



Universidad San Gregorio de Portoviejo

Carrera de Ciencias de la Computación.

Trabajo de Titulación.

Previo a la Obtención del Título de Ingeniero en Ciencias de la Computación

**Tema:**

Sistema Integrado para la automatización de los procesos del departamento de  
Talento Humano de la Universidad San Gregorio de Portoviejo

**Autor:**

Geovanny Andrés Linzán González

Tutora de Trabajo de Titulación

Ing. María Angelica Vélez Bermello.

Cantón Portoviejo – Provincia de Manabí – República del Ecuador.

2015

## **CERTIFICADO DE DIRECTORA DE TRABAJO DE TITULACIÓN**

Ing. Angélica Vélez Bermello, certifica que el Trabajo de Titulación: Sistema Integrado para la automatización de los procesos del departamento de Talento Humano de la Universidad San Gregorio de Portoviejo, es de la autoría de Geovanny Andrés Linzán González, el mismo que ha sido desarrollado bajo mi dirección.

---

Ing. Maria Angélica Vélez Bermello

Tutora de Trabajo de Titulación

## CERTIFICACIÓN DEL TRIBUNAL EXAMINADOR

El Trabajo de Investigación: Sistema Integrado para la automatización de los procesos del departamento de Talento Humano de la Universidad San Gregorio de Portoviejo, sometido a consideración del Tribunal por parte de la Ing. Sonia Párraga Muñoz, Coordinadora de la Carrera Ciencias de la Computación de la Universidad San Gregorio de Portoviejo, como requisito para la obtención del título de Ingeniero en Ciencias de la Computación.

### Tribunal

---

Ing. Alberto Balda Macias

Miembro del Tribunal

---

Ing. Sonia Párraga Muñoz

Miembro del Tribunal

---

Ing. Manuel Ponce Tubay

Miembro del Tribunal

---

Ing. Angélica Vélez Bermello

Tutora de Trabajo de Titulación

## **DECLARACIÓN DE AUTORÍA**

El Autor de la presente tesis, Geovanny Andrés Linzán González, declara que el trabajo de investigación titulado: Sistema Integrado para la automatización de los procesos del departamento de Talento Humano de la Universidad San Gregorio de Portoviejo, ha sido desarrollado en su totalidad por el investigador. Lo que doy fe para los fines legales correspondiente.

---

Geovanny Andrés Linzán González

Autor

## **AGRADECIMIENTO**

Ser grato, corresponde a reconocer desde un pequeño gesto hasta un verdadero acto de apoyo incondicional de quien o quienes en su momento me brindaron el apoyo necesario para el logro de esta meta alcanzada.

Gracias a mis padres; Ing. Geovanny Alfredo Linzán Villavicencio y la Sra. Rebeca Maria González Arteaga de Linzán que se convierten día a día en ese impulso motivador por verme realizado en cada uno de mis sueños.

Gracias a mi Abuelita que me guía y cuida desde el cielo.

Gracias a los familiares que siempre me apoyaron.

Gracias a los docentes que con sus conocimientos reafirmaron las ideas que llevaba y que hoy son mi patrimonio; mis conocimientos.

Gracias a la Universidad San Gregorio de Portoviejo, como institución de Educación Superior, en donde me formé como persona y profesional competente.

Geovanny Andrés Linzán González

## **DEDICATORIA**

Primero a Dios, por haberme dado fortaleza y ser fuente Divina del Saber

A mis padres, Geovanny y Rebeca, por ser ejemplos de perseverancia y constancia que los caracterizan y que me ha infundado siempre.

A ellos por sus consejos, por inculcarme valores y por la motivación constante que me permiten ser una persona de bien, pero sobre todas las cosas les dedico este trabajo por el amor que siempre me brindan.

A mis hermanas, Rebeca y Flormarina y a todas y cada una de las personas, que me brindaron su apoyo y orientación cuando más lo necesitaba en el transcurso de mi vida estudiantil y en la elaboración de este trabajo de titulación que es un gran paso en mi vida

Geovanny Andrés Linzán González

## RESUMEN

La realidad de las instituciones en la actualidad, frente a la evolución tecnológica constante, ameritan, ante las magníficas opciones que se pueden obtener de los sistemas informáticos, buscar, diseñar, desarrollar y poner a la disposición del Talento Humano, procesos creados por sus miembros para solucionar las necesidades de la institución. Para llevar a cabo el desarrollo de esta tesis se realizó el diagnóstico de necesidades del Departamento de Talento Humano de la Universidad Particular San Gregorio de Portoviejo, para luego del mismo y atendiendo criterios técnicos poder diseñar soluciones a los diferentes problemas que enfrenta, para lo cual fue necesario obtener toda la información, y cubriendo las necesidades que este requiere. Esta tesis fue basada en una herramienta informática que facilita el acceso a registros de control que se realizan diariamente en el Departamento de Talento Humano de la universidad.

Con la implementación del sistema Integrado, se logró: Acceder de forma inmediata a Roles de Pago, Permiso de vacaciones, Currículum resumido y completo del Personal, aportar a la toma de decisiones acertadas, simplificar tareas de emisión de registros, permitiendo gestionar otros procesos; Generar automáticamente Roles Mensuales con los datos almacenados en el sistema, Implementar la herramienta y asesorar al usuario final.

## **ABSTRACT**

The reality of the institutions currently face the constant technological evolution, warrant, to the magnificent options that are available from computer systems, search, design, develop and make available human talent, processes created by members to meet the needs of the institution. To carry out the development of this thesis needs assessment of the Department of Human Resource of the Private University San Gregorio de Portoviejo took place, then the same and attending technical criteria to design solutions to the various problems faced, for which It was necessary to obtain all the information and meeting the needs that it requires. This thesis was based on a computer tool that facilitates access to control registers are performed daily in the Human Resource Department of the university.

With the implementation of the integrated system, it was achieved: immediate access to Roles Pay, Vacation Leave, summarized and complete Curriculum Staff, contribute to making the right decisions, simplify tasks emission records, allowing manage other processes; Roles Monthly automatically generate the data stored in the system, implement the tool and advise the end user.

## INDICE

Certificado de directora de trabajo de titulación.....	II
Certificación del tribunal examinador.....	III
Declaración de autoría.....	IV
Agradecimiento.....	V
Dedicatoria.....	VI
Resumen.....	V
Abstract.....	VI
Indice.....	IX
Indice de figuras.....	XII
1. Tema.....	15
2. Antecedentes.....	15
3. Problematización.....	16
4. Justificación.....	18
5. Delimitación del problema.....	19
6. Objetivos.....	20
6.1. Objetivos General.....	20
6.2. Objetivos Específicos.....	20
7. Marco teórico.....	21
CAPÍTULO I.....	21
7.1. Analizar los requerimientos que involucran cada uno de los procesos que realiza el Departamento de Talento Humano de la USGP. ....	21
7.1.1. Estructura y Organización de la USGP.....	21
7.1.2. Departamento de Talento Humano.....	22
7.1.2.1. Procesos de Currículum.....	23
7.1.2.2. Procesos de Contratos.....	23
7.1.2.3. Procesos de Asistencia.....	24
7.1.2.4. Proceso Permisos y Vacaciones.....	24
7.1.2.5. Procesos de Rol de Pago.....	24
7.1.2.6. Proceso de Reportes.....	24
CAPÍTULO II.....	26
7.2. Diseñar la base de datos e interfaces del Sistema Integrado. ....	26

7.2.1.	Base de Datos .....	26
7.2.1.1.	Diseño de la Base de Datos.....	26
7.2.2.	Lenguaje SQL .....	27
7.2.3.	Workbench .....	27
7.2.4.	Cardinalidad .....	27
7.2.5.	Modelo Entidad-Relación .....	27
7.2.5.1.	Diagrama Entidad – Relación de la Base de Datos de Talento Humano.....	28
7.2.5.2.	Modelo Relacional.....	33
7.2.6.	Centos.....	42
7.2.7.	Interacción humano- computador .....	43
7.2.8.	Lenguaje de Programacion Visual Basic.NET.....	43
7.2.8.1.	Entorno de Desarrollo SharpDevelop.....	43
7.2.8.2.	Microsoft .Net Framework 4.5 .....	44
7.2.9.	Diseño de la Interfaz de Usuario.....	44
7.2.9.1.	Modelo Interfaz del Sistema Integrado.....	44
7.2.9.2.	Características de las Interfaces del Sistema Integrado .....	44
CAPITULO III .....		46
7.3.	Desarrollar los procesos del Sistema Integrado .....	46
7.3.1.	Inicio de Sesión .....	46
7.3.2.	Inicio del sistema.....	46
7.3.3.	Personal (Currículum y Contratos) .....	47
7.3.3.1.	Datos Personales .....	47
7.3.3.2.	Datos Administrativos .....	48
7.3.3.3.	Datos Profesionales.....	48
7.3.3.4.	Personal – Experiencia.....	49
7.3.4.	Asistencia General .....	50
7.3.4.1.	Leer Marcaciones.....	50
7.3.4.2.	Asistencia General – Calcular asistencia .....	50
7.3.4.3.	Feriados.....	51
7.3.5.	Asistencia Personal .....	52
7.3.5.1.	Horarios .....	52
7.3.5.2.	Asistencia.....	53
7.3.5.3.	Justificaciones.....	54
7.3.5.4.	Vacaciones .....	54
7.3.6.	Configuración Rol de Pago .....	55

7.3.7.	Generar Rol de Pago .....	56
7.3.8.	Modificar Rol de Pago .....	57
7.3.9.	Reportes.....	58
7.3.10.	Configuración.....	58
7.3.10.1.	Direcciones y Departamentos .....	58
7.3.10.2.	Cargos .....	59
7.3.10.3.	Países .....	59
7.3.10.4.	Instituciones Educativas .....	60
7.3.10.5.	Títulos Profesionales.....	61
7.3.10.6.	Contraseña .....	62
7.4.	Generar reportes de roles de pagos, vacaciones, permisos y vigencia de contratos.....	63
7.4.1.	MonoReport .....	63
7.4.2.	Generalidades.....	63
7.4.2.1.	Reporte del Personal .....	64
7.4.2.1.1.	Generación de Reporte del Personal .....	64
7.4.2.2.	Reportes Rol de pago.....	65
7.4.2.2.1.	Generación de Reportes Rol de Pago General.....	65
7.4.2.2.2.	Generación de reportes de Rol de Pago Personal .....	66
7.4.2.3.	Reportes de Contratos.....	66
7.4.2.3.1.	Generación de Reporte de Contratos .....	67
7.4.2.4.1.	Generación de reportes de Vacaciones .....	68
7.4.2.5.1.	Generación de Reportes de Asistencia.....	70
7.5.	Marco Teórico Ético y Legal.....	71
	Tipos de Contrato de la Universidad San Gregorio de Portoviejo .....	71
8.	Procedimiento Metodológico.....	76
8.1.	Modalidad de la investigación.....	76
8.2.	Tipo de investigación .....	76
8.3.	Métodos .....	77
8.3.1.	Método inductivo-deductivo .....	77
8.3.2.	Método descriptivo.....	77
8.3.3.	Método científico .....	77
8.3.4.	Método analítico-sintético.....	77
8.4.	Técnicas.....	77
8.4.1.	Técnica Bibliográfica .....	78
8.4.2.	Técnica de la entrevista,.....	78

8.5. Población .....	78
8.6. Muestra .....	78
9. Presupuesto .....	79
10. Conclusiones .....	80
11. Recomendaciones .....	81
12. Bibliografía .....	82
13. Anexos .....	85

## INDICE DE FIGURAS

FIGURA # 1: Organigrama Estructural de la USGP .....	22
FIGURA # 2: Esquema E-R Currículum .....	28
FIGURA # 3: Esquema E-R Asistencia .....	29
FIGURA # 4: Esquema E-R Contrato.....	30
FIGURA # 5: Esquema E-R Rol de Pago .....	31
FIGURA # 6: Esquema E-R Vacaciones .....	32
FIGURA # 7: Modelo relacional de Personal .....	34
FIGURA # 8: Modelo relacional de Departamento .....	34
FIGURA # 9: Modelo relacional de Reconocimiento.....	35
FIGURA # 10: Modelo relacional de Curso .....	35
FIGURA # 11: Modelo relacional de Libro .....	36
FIGURA # 12: Modelo relacional de Experiencia.....	36
FIGURA # 13: Modelo relacional de Institución Educativa.....	37
FIGURA # 14: Modelo relacional de Personal_has_Titulo.....	37
FIGURA # 15: Modelo relacional de Asistencia.....	38
FIGURA # 16: Modelo relacional de Marcación.....	38
FIGURA # 17: Modelo relacional de Permiso.....	39
FIGURA # 18: Modelo relacional de Horario .....	39
FIGURA # 19: Modelo relacional de Contrato.....	40
FIGURA # 20: Modelo relacional RolCabecera.....	40
FIGURA # 21: Modelo relacional RolDetalle .....	41
FIGURA # 22: Modelo relacional VacacionCabecera.....	41
FIGURA # 23: Modelo relacional VacacionDetalle .....	42
FIGURA # 24: Interfaz del sistema integrado .....	45
FIGURA # 25: Inicio de Sesion .....	46
FIGURA # 26: Inicio del Sistema.....	46
FIGURA # 27: Datos Personales .....	47
FIGURA # 28: Datos Administrativos.....	48
FIGURA # 29: Datos Profesionales .....	49
FIGURA # 30: Personal – Experiencia.....	49
FIGURA # 31: Asistencia General – Leer Marcaciones.....	50
FIGURA # 32: Calcular Asistencia.....	51
FIGURA # 33: Feriados .....	51
FIGURA # 34: Horarios.....	52
FIGURA # 35: Asistencia .....	53
FIGURA # 36: Justificaciones .....	54
FIGURA # 37: Vacaciones .....	55
FIGURA # 38: Configuración Rol de Pago .....	56
FIGURA # 39: Generar Rol de Pago .....	56

FIGURA # 40: Modificar Rol de Pago .....	57
FIGURA # 41: Direcciones y Departamentos .....	58
FIGURA # 42: Cargos .....	59
FIGURA # 43: Países.....	60
FIGURA # 44: Instituciones Educativas.....	60
FIGURA # 45: Títulos Profesionales.....	61
FIGURA # 46: Contraseña.....	62
FIGURA # 47: Reportes - Personal .....	64
FIGURA # 48: Reportes del Personal.....	65
FIGURA # 49: Generación de reportes Rol de Pago General .....	65
FIGURA # 50: Generación de reportes Rol de Pago Personal .....	66
FIGURA # 51: Reportes – Contrato.....	67
FIGURA # 52: Generación de Reporte de Contratos.....	67
FIGURA # 53: Reportes de Vacaciones .....	68
FIGURA # 54: Reportes de Vacaciones .....	69
FIGURA # 55: Reportes – Asistencia.....	69
FIGURA # 56: Generación de Reportes de Asistencia.....	70

## **1. Tema**

Sistema Integrado para la automatización de los procesos del departamento de Talento Humano de la Universidad San Gregorio de Portoviejo.

## **2. Antecedentes**

La existencia de la tecnología representa la creación de variedades de software que permitan realizar funciones específicas facilitando de forma ágil y segura los procesos. Es así que la Universidad San Gregorio de Portoviejo (USGP)<sup>1</sup>, en el mejoramiento continuo en el desarrollo institucional innovando los procesos administrativos contempla en su organigrama institucional, el Departamento de Talento Humano.

Este departamento cumple con las funciones de organizar el Talento Humano institucional, evidenciando los procesos del personal (administrativos y profesores) que a la fecha han generado varios intentos por automatizar cada uno de los procesos que se desarrollan en él, pero en ocasiones han sido fallidos y actualmente algunos de estos son realizados de forma manual, situación que significa la subutilización de un personal que puede con el sistema implementado realizar otras tareas. Por contemplar un desarrollo en el crecimiento institucional en relación de la infraestructura tecnológica necesaria que el departamento requiere, el presente trabajo de titulación enmarca la automatización de los

---

<sup>1</sup> USGP: Universidad San Gregorio de Portoviejo

procesos del Departamento de Talento Humano de la Universidad Particular San Gregorio de Portoviejo.

### **3. Problematización**

El departamento de Talento Humano por la ejecución de sus procesos se puede visualizar los siguientes problemas:

El primer problema que se identifica es, el registro de la jornada laboral por medio de un biométrico que no reflejan las faltas y atrasos del personal, lo que no contribuye a la elaboración del rol de pago de una manera más ágil y oportuna. En la actualidad, los registros de asistencia son descargados de una aplicación a una hoja de cálculo (Excel) con el objetivo de revisar diariamente empleado por empleado la asistencia marcadas al biométrico; posteriormente, se envía al Departamento Financiero un informe de los empleados con las multas respectivas por faltas y atrasos. Este procedimiento causa inconvenientes, ya que, por errores involuntarios se puede perjudicar o beneficiar al personal, es importante resaltar que el cálculo de los descuentos no debe ser realizado de manera manual, lo ideal es que se lo realice de manera automatizada y en el proceso no intervenga ninguna persona.

El segundo problema, tiene relación con los días de vacaciones, en este caso, el control es llevado a través de una hoja de cálculo donde los días son actualizados y cuando existen permisos con cargos a vacaciones este proceso tiende a errores que es responsabilidad del encargado de actualizar el archivo, quien se olvida de registrar los cambios referidos.

El tercer caso que se problematiza es, el referente a Contratos, por lo que cada mes se debe realizar un informe de los empleados y trabajadores a los que se le vence el contrato para luego ser enviado al Rector de la institución y se tomen las decisiones pertinentes, el inconveniente radica en la pérdida de tiempo que significa buscar la información en las carpetas físicas de los empleados y trabajadores, para obtener la fecha de vencimiento del contrato.

En lo referente con el currículum vitae del personal que labora en la universidad, es el cuarto caso problematizado, que consiste en que los datos del personal (profesores y administrativos) se encuentran archivados en carpetas físicas en el departamento de Talento Humano y lo adecuado sería, que estén en forma digital para que no haya acumulación de papeles y exista un mejor respaldo en caso de que se extravíe la documentación.

El quinto caso, tiene que ver con el tema de Registro de horarios presenta problemas a la hora de llevar un control sobre los horarios diarios ya que no hay conocimiento de los cambios de último momento, los cuales son realizados por circunstancias de diverso tipo.

#### **4. Justificación**

Es de suma importancia que el departamento de Talento Humano de la Universidad Particular San Gregorio de Portoviejo, cuente con un Software óptimo para el manejo de procesos de contratación, registros y asistencias del personal que laboran en este Centro de Estudios Superior, el mismo que debe ser totalmente automatizado, para que el personal encargado de ese departamento realice su trabajo de manera ágil, ahorrando tiempo y pueda llevar un registro adecuado sobre toda la información del personal.

Al integrar el biométrico con los roles de pago, se logrará la automatización de multas por atrasos e inasistencias del personal de la institución, lo cual permitirá la seguridad de que los descuentos sean calculados de manera automática, adicionalmente se elimina la intervención manual del encargado de Talento Humano para el cálculos de multas.

Al automatizar el proceso de control en el vencimiento de contratos, se logrará minimizar el tiempo y recurso humano invertido en buscar información en carpetas físicas de los empleados, ya que, se contará con la generación de informes dinámicos y con criterios de búsqueda de manera oportuna y eficiente. Además se podrá llevar un control de los contratos activos y vencidos de la institución.

El control automático de vacaciones permitirá mantener actualizada la información y ser consultada en cualquier momento por las personas encargadas del departamento de Talento Humano con un historial de los días gozados y por

gozar de los empleados, con esto se minimizara el tiempo invertido por el personal de Talento Humano.

Con el Currículum ingresado en el sistema se prevendrá tener acumulación de papeles y tener un mejor respaldo de la información referente a los empleados que laboran en la institución, así como también se podrán actualizar los datos de manera digital por el personal encargado.

Con la implementación del sistema de registro de los horarios del personal, se tendrá conocimiento de los cambios diarios realizados y así llevar un mejor control sobre las jornadas laborales, esto evitará inconvenientes al momento de realizar el cálculo de las horas trabajadas.

## **5. Delimitación del problema**

**Campo:** Software

**Área:** Administrativa

**Aspecto:** Sistema Integrado Automatizado.

**Tema:** Sistema Integrado para la automatización de los procesos del departamento de Talento Humano de la Universidad San Gregorio de Portoviejo.

El sistema tendrá las funcionalidades de rol de pago, control de contratos, asistencia, permisos y vacaciones, currículum vitae y el registro de horarios del personal de la Universidad San Gregorio de Portoviejo.

Se desarrollará un software con ambiente de escritorio

**Delimitación Espacial:** Departamento de Talento Humano, de la Universidad Particular San Gregorio de Portoviejo, ubicada en la Ciudad de Portoviejo en la Avenida Metropolitana y Avenida Olímpica.

**Delimitación Temporal:** La investigación se llevó a cabo desde el mes de febrero de 2015 hasta el mes de mayo de 2015.

## **6. Objetivos**

### **6.1. Objetivos General**

Desarrollar el Sistema Integrado para la automatización de los procesos del Departamento de Talento Humano de la Universidad San Gregorio de Portoviejo.

### **6.2. Objetivos Específicos**

- Analizar los requerimientos que involucran cada uno de los procesos que realiza el Departamento de Talento Humano de la USGP.
- Diseñar la base de datos e interfaces del Sistema Integrado.
- Desarrollar los procesos del Sistema Integrado
- Generar reportes de roles de pagos, vacaciones, permisos y vigencia de contratos.

## **7. Marco teórico**

### **CAPÍTULO I**

#### **7.1. Analizar los requerimientos que involucran cada uno de los procesos que realiza el Departamento de Talento Humano de la USGP.**

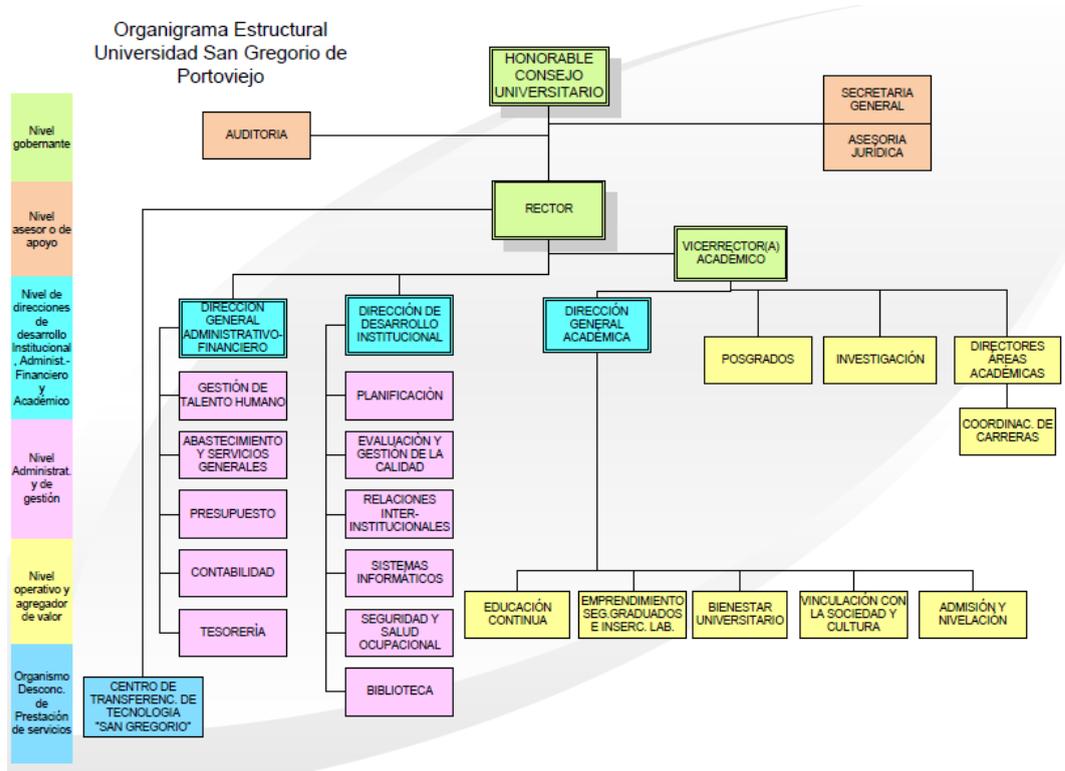
La ingeniería de requisitos del software es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software. (Pressman, 2002).

Para la aplicación de un software se debe cumplir con tareas para determinar el requerimiento del mismo, por lo que haciendo referencia a Pressman, es necesario que se reconozca el problema, luego se realice una evaluación utilizando la síntesis, pasamos posteriormente a un modelado con las especificaciones particulares de cada caso; al final se revisará su eficacia.

##### **7.1.1. Estructura y Organización de la USGP**

La Universidad San Gregorio de Portoviejo está estructurada por los niveles: Gobernante, Asesor, Direcciones de desarrollo institucional Administrativo-Financiero y Académico, Administrativo y de Gestión, Operativo y Agregado de valor y Organismo Descon. De Prestación de Servicios que cuenta con las consiguientes Unidades de Administración.

**FIGURA # 1: Organigrama Estructural de la USGP**



Fuente: Pagina web de la Universidad San Gregorio de Portoviejo

### 7.1.2. Departamento de Talento Humano

El Autor (Chiavenato, 2002) menciona que

“La gestión del talento humano es un área muy sensible a la mentalidad que predomina en las organizaciones. Es contingente y situacional, pues depende de aspectos como la cultura de cada organización, la estructura organizacional adoptada, las características del contexto ambiental, el negocio de la organización, la tecnología utilizada, los procesos internos y otra infinidad de variables importantes.”

El Departamento de Talento Humano de la USGP, requiere de un software, que ayude a agilizar los procesos que se ejecutan dentro del departamento, ya que, las aplicaciones que actualmente están utilizando hacen que los procesos de cálculos

sean lentos y tediosos y requieran de mucho personal. Se necesita almacenar la información personal de una manera más ágil. En la actualidad, estos procesos toman mucho tiempo porque son ingresados en una hoja electrónica y no permiten obtener resultados inmediatos, por lo que, estos procesos están sujetos a un término de tiempo demorado.

#### **7.1.2.1. Procesos de Currículum**

Dentro de los procesos de currículum es donde se lleva a cabo toda la información referente a los docentes, administrativos y auxiliares de servicios que laboran en la USGP, tales como sus datos personales, datos administrativos, datos profesionales, y experiencia, con esto se logra que el departamento de Talento Humano a mas de tener los curriculums en hojas físicas, también los tenga ingresados dentro del sistema integrado.

#### **7.1.2.2. Procesos de Contratos**

El Proceso de contrato atiende en el Sistema Integral los contratos de servicios profesionales que de acuerdo a Cabanellas, no es otra cosa que:

El contrato de Servicios Profesionales, es el contrato por el cual una persona se obliga frente a otra a ejecutar un trabajo o una empresa determinada, es un contrato por el cual una persona pone su actividad y sus talentos profesionales al servicio de otra persona por un tiempo determinado o indeterminado. (Cabanellas, 2006).

La información que proporciona este proceso tiene relación con la fecha de inicio del contrato y la fecha de finalización del mismo. También se puede elegir el tipo de contrato de cada servidor.

#### **7.1.2.3. Procesos de Asistencia**

Dentro de los requisitos y requerimientos se tiene un módulo de control de marcaciones y asistencia del personal administrativo, docente y auxiliar de servicio de la institución para así tener de manera eficaz las marcaciones realizadas diariamente por el personal que labora en la universidad y saber cuándo el mismo asiste, falta o llega con atraso a sus actividades laborales.

#### **7.1.2.4. Proceso Permisos y Vacaciones**

Dentro de este proceso se encuentra la información de los permisos respaldados por las justificaciones correspondientes, también se ingresan las fechas de vacaciones que le pertenecen a cada empleado con su respectiva fecha de salida y fecha de entrada de las mismas, además se podrá visualizar los días gozados y el saldo de días restantes que le quedan por gozar.

#### **7.1.2.5. Procesos de Rol de Pago**

Los procesos de rol de pago como su nombre lo indica es donde se realizaran los procesos que corresponden al rol de cada empleado que labora en la institución, en él se ingresarán los rubros con su fórmula respectiva, dependiendo del cálculo que se requiera que se haga, para poder realizar el procedimiento de manera automatizada para los valores de los rubros que son fijos.

#### **7.1.2.6. Proceso de Reportes**

Aquí es donde se ejecutan los reportes que se manejan en el departamento de acuerdo a las necesidades que se requieran, como la vigencia de los contratos que

muestra la fechas de inicio y fin del empleado, las asistencias diarias para poder tener un mejor control sobre horas laboradas, los roles de pago con los debidos rubros y sus descuentos, y el reporte de las vacaciones con la información de cada empleado o trabajador con su fecha de salida y entrada.

## **CAPÍTULO II**

### **7.2. Diseñar la base de datos e interfaces del Sistema Integrado.**

De acuerdo al diagnóstico realizado durante la determinación y requerimientos del departamento de Talento Humano en este capítulo se hablará sobre el diseño de la base de datos y además se contemplará la interfaz diseñada para el sistema integrado.

#### **7.2.1. Base de Datos**

Varios son los autores que expresan su opinión sobre la base de datos, pero es Osorio Rivera, que expresa que:

Una base de datos es una colección estructuradas de datos relacionados. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda o tan vasta como la de una tienda en línea, un portal, o la información generada en una red empresarial. (Fray León Osorio Rivera, 2008)

##### **7.2.1.1. Diseño de la Base de Datos**

La base de datos utilizada, se sirve del MySQL, potente gestor de bases de datos Open Source (Código libre) de fácil manejo, que proporciona una herramienta visual de diseño (workbench), de gran escalabilidad al momento de manejar cantidades enormes de información, y con un funcionamiento a gran velocidad, debido a su lenguaje SQL que permite realizar consultas de forma inmediata.

### **7.2.2. Lenguaje SQL**

El SQL es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales. Es un lenguaje declarativo: sólo hay que indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo hay que hacer cualquier acción sobre la base de datos. El SQL es un lenguaje muy parecido al lenguaje natural; concretamente, se parece al inglés, y es muy expresivo.” (Carme Martín Escofet, 2014)

### **7.2.3. Workbench**

MySQL Workbench es una herramienta visual unificada para los arquitectos de bases de datos, desarrolladores y administradores de bases de datos. MySQL Workbench proporciona el modelado de datos, desarrollo de SQL y herramientas completas de administración de la configuración del servidor, la administración de usuarios, copia de seguridad, y mucho más. MySQL Workbench está disponible en Windows, Linux y Mac OS X. (Pascual, 2014)

### **7.2.4. Cardinalidad**

“La Cardinalidad expresa un número mínimo y máximo de casos de entidad asociados con un caso de la entidad relacionada.” (Coronel, 2011). Las cardinalidades entre una relación de entidades pueden ser de uno a uno, uno a muchos o muchos a uno y de muchos a muchos.

### **7.2.5. Modelo Entidad-Relación**

El modelo de datos entidad relación está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica permitiendo la especificación de un esquema de la empresa que se representa la estructura lógica completa de una base de datos. El modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. (Nogueras, 2014).

### 7.2.5.1. Diagrama Entidad – Relación de la Base de Datos de Talento

#### Humano

La base de datos que tiene como Tabla principal, la entidad Personal, guarda información de docentes, administrativos y auxiliares de servicios, es decir, al Talento Humano que labora en la universidad, en lo referente a: Datos personales, profesionales y laborales. Esta entidad se relaciona con las entidades: Dirección, Departamento, Reconocimiento, Curso, Título, Libro, Experiencia, Institución Educativa y País. La tabla Personal lleva relación de Uno a Muchos con la entidad Personal\_has\_Título, que es la consecuencia de la relación de Muchos a Muchos entre la entidad Título con la entidad Institución Educativa, como se detalla en la figura # 2, correspondiente al Esquema E-R Currículum.

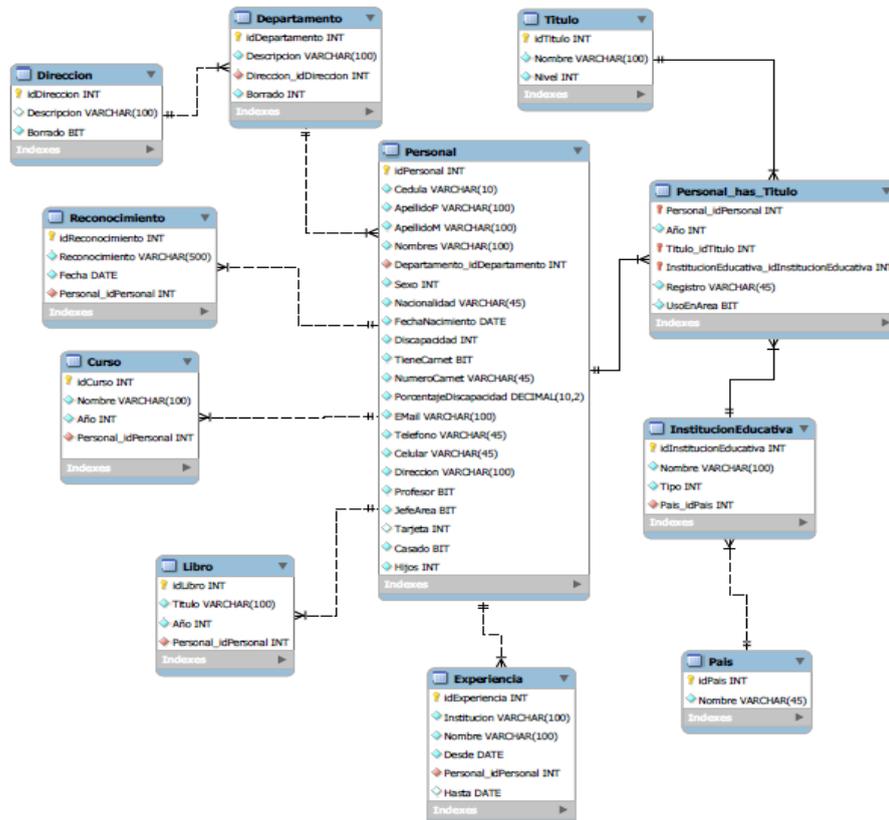
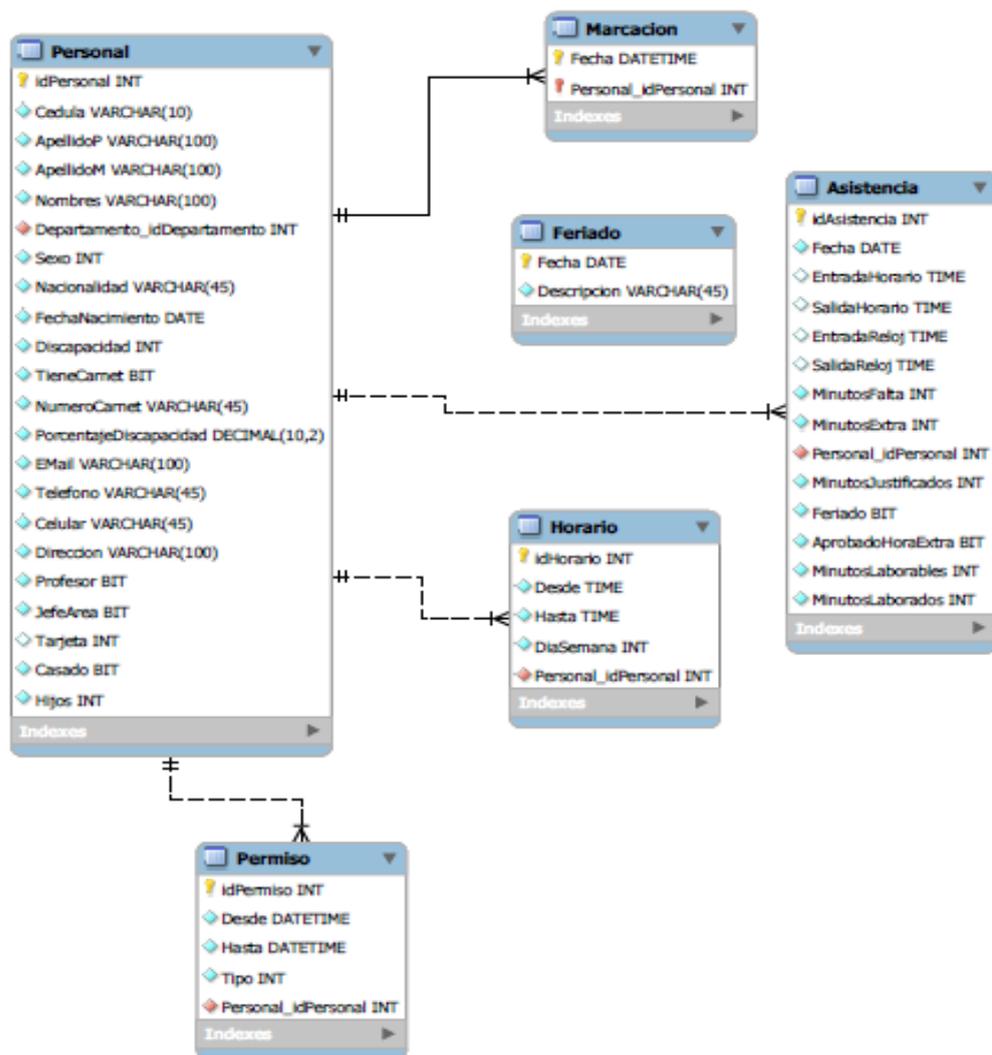


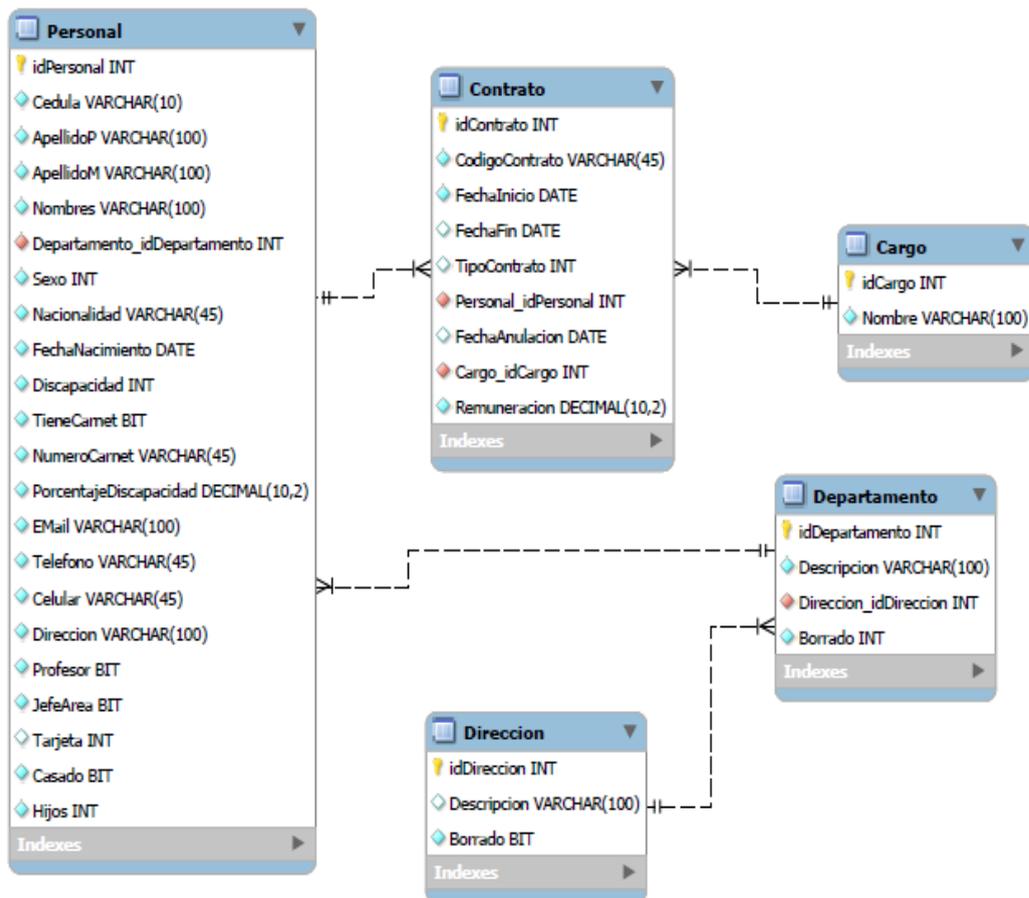
FIGURA # 2: Esquema E-R Currículum

Para el Esquema E-R Asistencia, la tabla Personal, estará relacionada de Uno a Muchos con las entidades; Marcación, Horario, Asistencia y Permiso. La figura #3 evidencia la entidad Feriado, sin relación con ninguna de las entidades detalladas anteriormente, porque en esta entidad o tabla se guardará únicamente el feriado de cada año. Esta entidad debe ser actualizada de acuerdo a los cronogramas de actividades de la universidad.



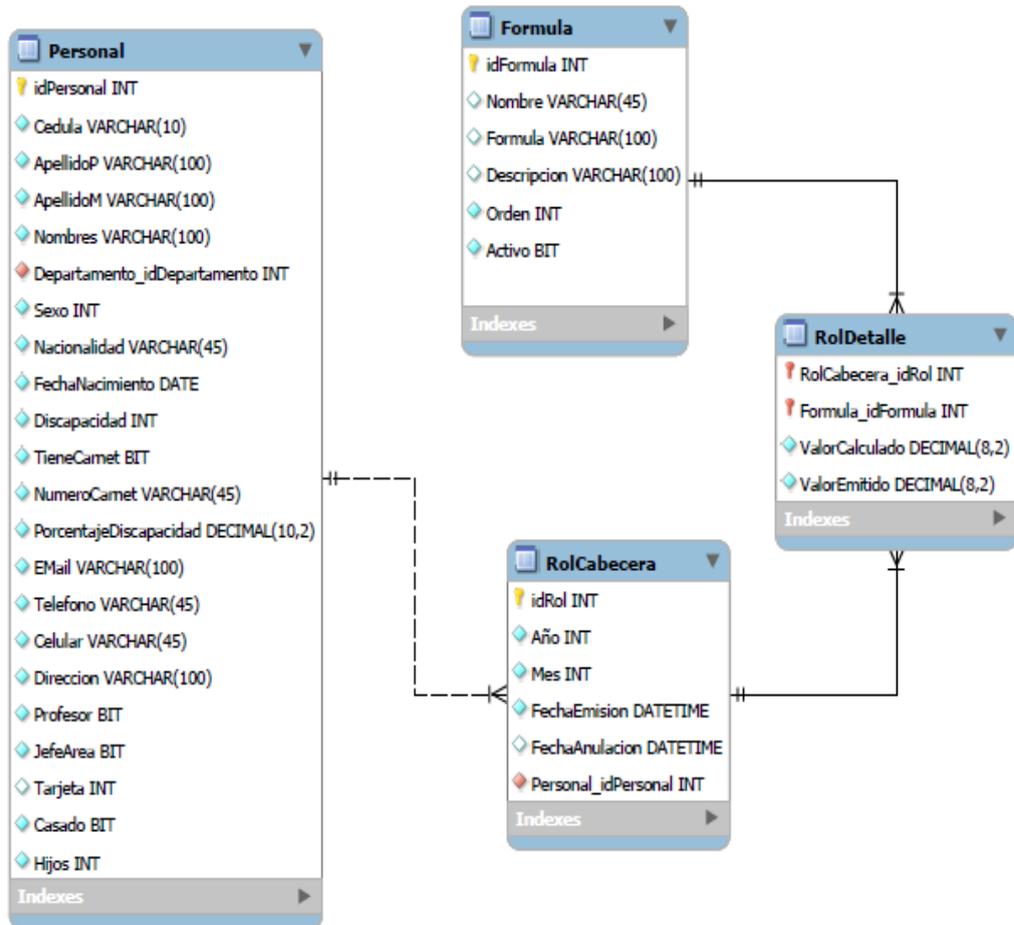
**FIGURA # 3: Esquema E-R Asistencia**

En el Esquema E-R de Contratos que se muestra en la figura #4 existe una relación de Uno a Muchos entre la entidad Personal con la entidad Contrato, la misma que se relaciona con la entidad Cargo de Muchos a Uno. Aunque las entidades Departamento y Dirección corresponden de manera más directa al Esquema E-R Currículum, estas también intervienen en el Esquema E-R Contrato.



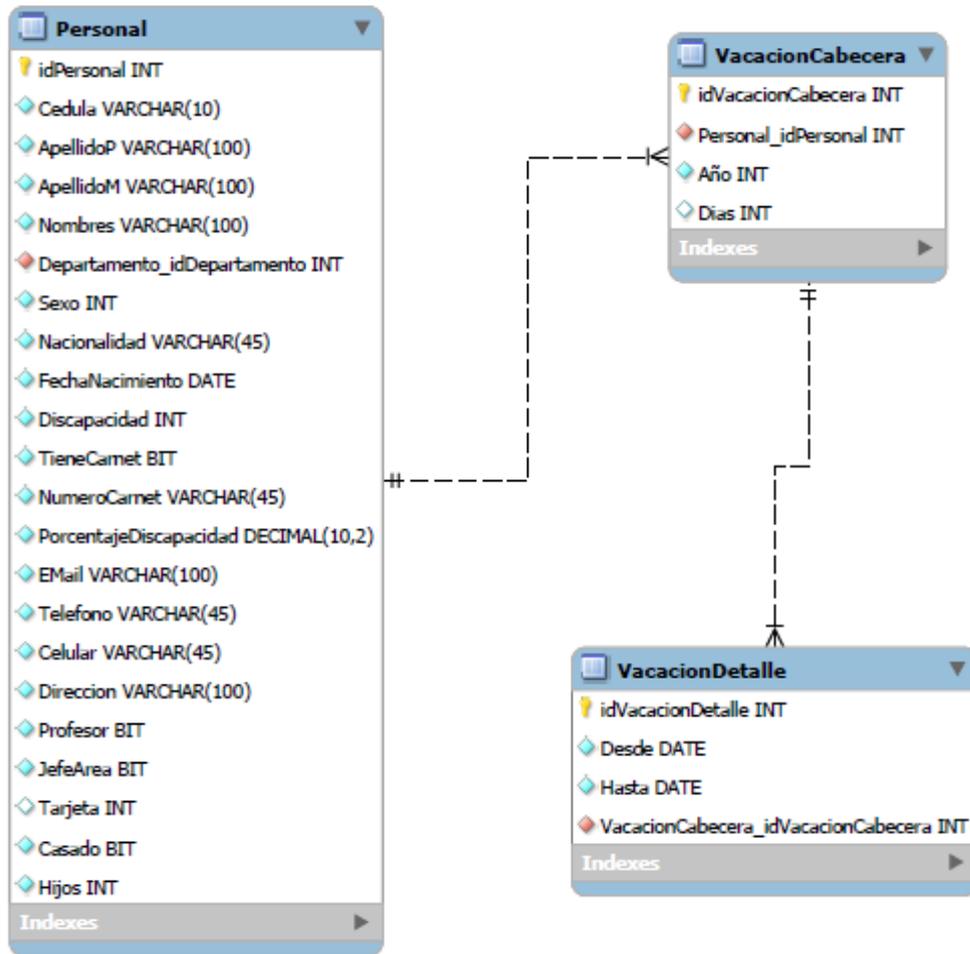
**FIGURA # 4: Esquema E-R Contrato**

La relación existente entre las entidades o tablas que corresponden al Esquema E-R Rol de Pago como se observa en la figura #5 es: de Uno a Muchos, entre Personal y RolCabecera; de Uno a Muchos entre las entidades RolCabecera y RolDetalle, y de Uno a Muchos entre las entidades Fórmula y RolDetalle.



**FIGURA # 5: Esquema E-R Rol de Pago**

La relación existente entre las entidades o tablas que corresponden al Esquema E-R Vacaciones que se muestra en la figura #6 es: de Uno a Muchos, entre Personal y VacacionCabecera; de Uno a Muchos entre las entidades VacacionDetalle y VacacionCabecera.



**FIGURA # 6: Esquema E-R Vacaciones**

### 7.2.5.2. Modelo Relacional

En el modelo relacional se basa en el concepto matemático de relación. En este modelo, la información se representa en forma de “tablas” o relaciones, donde cada fila de la tabla se interpreta como una relación ordenada de valores (un conjunto de valores relacionados entre sí) (López R, 2011-2012)

En el Modelo del Sistema Integral Automatizado para el Departamento de Talento Humano, se utilizó entidades con nombres que identifican las tablas y estas son reconocidas por su primary key (PK<sup>2</sup>), que es única de cada una de ellas.

Este modelo relacional, que corresponde a la entidad de Personal, contiene veintiún atributos propios, lo que significa, creados para esta entidad y en la figura #7 se resalta un atributo heredado de la entidad Departamento mediante una FK<sup>3</sup>. Esta relación está constituida por el tipo de información que puede ser: INT, BIT, VARCHAR, DECIMAL y DATE que forman parte de la relación dando a esta la característica que la hace particular.

El tipo de dato INT, corresponde a los valores numéricos, el tipo de dato BIT, corresponde a números binarios que son 1 y 0; por su parte el tipo de dato VARCHAR, puede ser tanto de letra como de número o mixto, que es cuando más se utiliza. El tipo de dato DECIMAL, es para números decimales y el tipo de dato DATE es para fechas.

Los valores que tienen una Viñeta o Visto marcado, en la opción NN, son valores que no pueden estar vacíos

---

<sup>2</sup> PK: Primary Key (Clave Primaria)

<sup>3</sup> FK: Foreign Key (Clave Foránea)

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idPersonal	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Cedula	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ApellidoP	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ApellidoM	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nombres	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Departamento_idDepartamento	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sexo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nacionalidad	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FechaNacimiento	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Discapacidad	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TieneCarnet	BIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NumeroCarnet	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PorcentajeDiscapacidad	DECIMAL(10,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Email	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Telefono	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Celular	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Direccion	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Profesor	BIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
JefeArea	BIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tarjeta	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Casado	BIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hijos	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**FIGURA # 7: Modelo relacional de Personal**

La identidad de Departamento, consta de un modelo relacional, que contiene tres atributos propios, que corresponden a IdDepartamento, Descripción y Borrado y un atributo heredado de la entidad Dirección, que es Direccion\_IdDireccion, lo que indica que cada vez que se ingrese una dirección será heredada al departamento correspondiente (figura#8). En este modelo relacional podemos identificar la utilización de dos tipos de información, INT y VARCHAR, porque solo requiere de números y letras.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idDepartamento	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Descripcion	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Direccion_idDireccion	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Borrado	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 8: Modelo relacional de Departamento**

Este modelo, que corresponde a la tabla Reconocimiento lleva consigo tres atributos propios y uno heredado. Por su parte los atributos propios son: IdReconocimiento, Reconocimiento y Fecha y el atributo heredado es Personal\_IdPersonal (figura #9)

En éste modelo se identifica la utilización de tres tipos de información; INT, VARCHAR y DATE, ya que, requiere de números, letras y fechas.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Def
idReconocimiento	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
Reconocimiento	VARCHAR(500)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Fecha	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				

**FIGURA # 9: Modelo relacional de Reconocimiento**

El modelo relacional de la entidad Curso nos presenta tres atributos propios, que son: IdCurso, Nombre y Año; y un atributo heredado que corresponde a Personal\_IdPersonal. Esta tabla alimenta al igual que las anteriores y otros sistemas que agregan datos a otras entidades. (Figura#10)

En este modelo relacional se logra identificar la utilización de dos tipos de información, INT y VARCHAR, porque solo requiere de números y letras.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Def
idCurso	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
Nombre	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Año	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				

**FIGURA # 10: Modelo relacional de Curso**

La Tabla de Libro, consta de un modelo relacional, que contiene tres atributos propios, que corresponden a IdLibro, Título y Año y un atributo heredado de la entidad Personal, que es Persona\_IdPersonal, lo que indica que cada vez que se ingrese un libro se genera una información a Personal\_IdPersonal (figura#11) Igual que la relación anterior, ésta utiliza dos tipos de información, INT y VARCHAR, porque solo requiere números y letras.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Def
idLibro	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
Título	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Año	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				

**FIGURA # 11: Modelo relacional de Libro**

El modelo relacional de la entidad Experiencia, presenta cinco atributos propios, que son: IdExperiencia, Institución, Nombre, Desde y Hasta; y un atributo heredado que corresponde a Personal\_IdPersonal. (Figura#12)

En éste modelo se identifican tres tipos de información, INT, VARCHAR y DATE, este último proporciona fechas.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Def
idExperiencia	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
Institucion	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Nombre	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Desde	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
Hasta	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

**FIGURA # 12: Modelo relacional de Experiencia**

La tabla que constituye el modelo relacional de Institución Educativa, contiene tres atributos propios y un atributo heredado, que son; IdInstitucionEducativa, Nombre, Tipo y Pais\_IdPais, respectivamente en la figura#13.

En este modelo relacional podemos identificar la utilización de dos tipos de información, INT y VARCHAR.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idInstitucionEducativa	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Nombre	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Tipo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Pais_idPais	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 13: Modelo relacional de Institución Educativa**

La identidad de Personal\_has\_Título, consta de un modelo relacional, que contiene tres atributos propios, que corresponden a Año, Registro y UsoEnArea, y tres atributos heredados de las entidades Personal\_IdPersonal, Titulo\_IdTitulo e InstituciónEducativa\_IdInstitucionEducativa. Las dos últimas nombradas tienen una relación de Muchos a Muchos, lo que causa la necesidad de crear la tabla Personal\_has\_Titulo como se observa en la figura#14. Aquí se identifica tres tipos de información, INT, VARCHAR y BIT. Los atributos heredados son de tipo INT.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
Personal_idPersonal	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Año	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Titulo_idTitulo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
InstitucionEducativa_idInstitud...	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Registro	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
UsoEnArea	BIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 14: Modelo relacional de Personal\_has\_Titulo.**

Como se muestra en la figura #15 el modelo relacional Asistencia, consta de trece atributos propios y un atributo heredado llamado Personal\_idPersonal lo que significa que los valores necesarios de la asistencia serán heredados a la entidad personal. Esta relación está constituida por cuatro tipos de información: INT, DATE, TIME y BIT que forman parte de la relación. El TIME, corresponde a los valores numéricos que expresan tiempo.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idAsistenda	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Fecha	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
EntradaHorario	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SalidaHorario	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EntradaReloj	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SalidaReloj	TIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MinutosFalta	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
MinutosExtra	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
MinutosJustificados	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Feriado	BIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
AprobadoHoraExtra	BIT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
MinutosLaborables	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
MinutosLaborados	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 15: Modelo relacional de Asistencia.**

En el modelo relacional Marcación consta de un atributo propio llamado Fecha que es considerado como su PK y su atributo heredado Personal\_idPersonal significa que las marcaciones serán heredadas automáticamente a la entidad Personal como lo indica en la figura #16. Esta relación está constituida por dos tipos de información: INT y DATETIME

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
Fecha	DATETIME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Personal_idPersonal	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 16: Modelo relacional de Marcación**

Como se puede apreciar en la figura # 17 el modelo relacional Permiso consta de cuatro atributos propios y de un atributo heredado llamado Personal\_idPersonal. Quiere decir que la información de los permisos ingresada también será reflejada en la entidad personal. Esta relación está constituida por dos tipos de información: INT y DATETIME

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idPermiso	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Desde	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Hasta	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Tipo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 17: Modelo relacional de Permiso**

En el modelo relacional Horario observamos que contiene cuatro atributos propios que es donde se verán los horarios del personal y un heredado llamado Personal\_idPersonal como se observa en la figura #18. Esta relación está constituida por dos tipos de información: INT y TIME, que corresponde a valores numéricos y tiempo.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idHorario	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Desde	TIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Hasta	TIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
DiaSemana	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 18: Modelo relacional de Horario**

Como se aprecia en la figura #19 el modelo relacional Contrato está compuesto por siete atributos propios y dos atributos heredados llamados Personal\_idPersonal y Cargo\_idCargo, lo que quiere decir que la entidad Contratos hereda información necesaria de la entidad Personal y de la Entidad Cargo. Esta relación está constituida por cuatro tipos de información: INT, VARCHAR, DATE y DECIMAL

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idContrato	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CodigoContrato	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FechaInicio	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FechaFin	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TipoContrato	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FechaAnulacion	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cargo_idCargo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Remuneracion	DECIMAL(10,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**FIGURA # 19: Modelo relacional de Contrato**

El modelo Relacional RolCabecera contiene cinco atributos propios y un atributo heredado llamado Personal\_idPersonal, como se observa en la figura #20. Esta entidad tiene una relación de muchos a muchos con la entidad Formula lo que causa un rompimiento creando así la entidad RolDetalle, figura #21. La relación Rol\_Cabecera está constituida por dos tipos de información: INT y DATETIME.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idRol	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Año	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Mes	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
FechaEmision	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
FechaAnulacion	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 20: Modelo relacional RolCabecera**

En la figura #21 modelo relacional RolDetalle consta de dos atributos propios y dos atributos heredados por el producto del rompiendo de relaciones de muchos a muchos como se explica en la figura #20. Este modelo relacional está constituido por dos tipos de información: INT y DECIMAL.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
RolCabecera_idRol	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Formula_idFormula	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
ValorCalculado	DECIMAL(8,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
ValorEmitido	DECIMAL(8,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 21: Modelo relacional RolDetalle**

En la figura #22, el modelo relacional VacacionCabecera está compuesto por tres atributos propios y un atributo heredado llamado Personal\_idPersonal. Esta relación consta de un solo tipo de información que corresponde a INT, con valores numéricos.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idVacacionCabecera	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Personal_idPersonal	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Año	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Dias	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**FIGURA # 22: Modelo relacional VacacionCabecera**

En la figura #23 la entidad VacacionDetalle consta de un modelo relacional de tres atributos propios y un atributo heredado de la entidad VacacionCabecera llamado VacacionCabecera\_idVacacionCabecera que significa que la entidad VacacionCabecera tendrá conocimiento sobre las fechas correspondiente a las vacaciones del personal y además se activa la fecha en que debe ingresar a laborar.

Esta relación consta de dos tipos de información que corresponde a INT y DATE, con valores numéricos y fechas.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
idVacacionDetalle	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Desde	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Hasta	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
VacacionCabecera_idVacacion..	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>				

**FIGURA # 23: Modelo relacional VacacionDetalle**

### 7.2.6. Centos

“CentOS es una distribución de Red Hat para Red Hat Enterprise Linux (RHEL). Principalmente cambia los paquetes para eliminar la marca del proveedor, es sin costo y libre de redistribuir.” (CentOS; Hughes, Jhonny;, 2015)

En el sistema operativo Centos es donde se alojará la base de datos con sus respectivas entidades y relaciones.

### **7.2.7. Interacción humano- computador**

La interacción humano-computador o la comunicación entre el ser humano y el computador, es el intercambio de información en donde interviene la interfaz utilizada con el usuario. Quien utilice este sistema está limitado a las opciones desplegadas. El usuario no necesita conocer el sistema, pero tiene que saber qué tarea se deba realizar.

Los menús no dependen del hardware sino del software, el menú establece el necesario uso del ratón para explorar y del teclado para ingresar información nueva. La consistencia es importante en el diseño de una interfaz de menú porque ayuda a su manejo. Los menús se pueden anidar dentro de otro para llevar a un usuario a las opciones de un programa. Los menús anidados permiten a la pantalla aparecer menos desordenada, la cual es consistente con el adecuado diseño. Los menús anidados también pueden mover rápidamente a los usuarios a través del programa.

### **7.2.8. Lenguaje de Programacion Visual Basic.NET**

Visual Basic .NET es un pilar de .NET Framework y otro paso hacia adelante en la evolución del lenguaje. Es un lenguaje de programación de alto nivel de .NET Framework y proporciona el punto de entrada más fácil a .NET. (Microsoft, 2015)

#### **7.2.8.1. Entorno de Desarrollo SharpDevelop**

La fuente de consulta que permitió desarrollar el Sistema Integrado para esta investigación es (ic#code, 2012), en él se define a SharpDevelop como el entorno

de desarrollo integrado libre para los lenguajes de programación C#, Visual Basic.NET, F#, Python, Ruby, Boo y C++.

#### **7.2.8.2. Microsoft ■Net Framework 4.5**

.NET Framework es un entorno de ejecución administrado que proporciona diversos servicios a las aplicaciones en ejecución. Consta de dos componentes principales: Common Language Runtime (CLR), que es el motor de ejecución que controla las aplicaciones en ejecución, y la biblioteca de clases de .NET Framework, que proporciona una biblioteca de código probado y reutilizable al que pueden llamar los desarrolladores desde sus propias aplicaciones (Microsoft, 2015)

#### **7.2.9. Diseño de la Interfaz de Usuario**

Roger Pressman, en la séptima edición de su obra, editada por Mc Graw-Hill, en 2010, expresa:

El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora. Siguiendo un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de esta y luego crea una plantilla de pantalla que constituye la base del prototipo de la interfaz de usuario (Presman, 2010)

##### **7.2.9.1. Modelo Interfaz del Sistema Integrado**

El modelo de la interfaz utilizada en el Sistema Integrado diseñado es de Tipo Menú porque proporciona al usuario una lista de pantalla de las selecciones disponibles.

##### **7.2.9.2. Características de las Interfaces del Sistema Integrado**

Como se puede apreciar en la figura #24 las interfaces que en número son ocho las cuales conforman el Sistema Integral desarrollado, cuentan con las mismas características físicas por lo que su operatividad es similar, están diseñadas para

que se adapte de acuerdo a los pixeles del monitor, volviéndolo de fácil adaptabilidad.

El diseño es técnico porque cuenta con colores que no afectan la visión siendo estos agradables para el usuario y van desde los colores de plomo y blanco, con color terroso poco intenso en los bordes. El Menú está situado al lado izquierdo de la pantalla siendo esta una variante en relación con otros programas y se lo ha realizado para darle la condición innovadora.



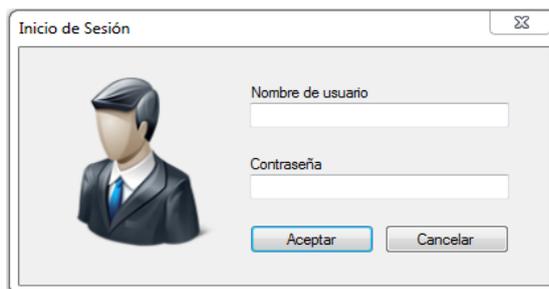
**FIGURA # 24: Interfaz del sistema integrado**

## CAPITULO III

### 7.3. Desarrollar los procesos del Sistema Integrado

#### 7.3.1. Inicio de Sesión

El Sistema Integrado Automatizado contiene un Inicio de Sesión en el cual se ingresará el usuario y contraseña correctos para poder acceder al software.



**FIGURA # 25: Inicio de Sesion**

#### 7.3.2. Inicio del sistema

Este Inicio cuenta con una pequeña portada antes de poder gustar de las opciones que nos ofrece el sistema.



**FIGURA # 26: Inicio del Sistema**

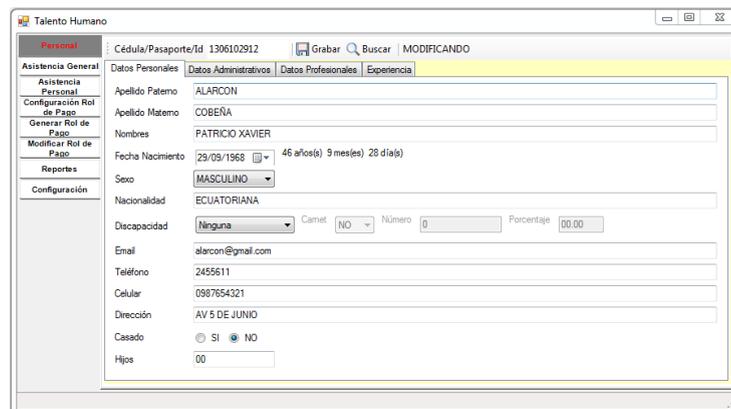
### 7.3.3. Personal (Currículum y Contratos)

En Personal es donde se ingresara todo lo referente al Personal, como son:

**Datos Personales, Datos Administrativos, Datos Profesionales y Experiencia**

#### 7.3.3.1. Datos Personales

En Datos Personales es donde se visualizan o almacenan (en caso de que no estén registrados) por medio de su Identificación, Cédula o Pasaporte, los datos referente al personal Docente, Administrativo y Auxiliares de Servicios que laboran en la institución, y estos son: Apellido paterno, apellido materno, nombres, fecha de nacimiento, si tiene o no alguna discapacidad (si la tiene da acceso a ingresar si tiene o no carnet, si tiene se puede ingresar el número del carnet y el porcentaje de la discapacidad, además se puede conocer el E-mail del empleado o trabajador, su contacto telefónico, la dirección de su domicilio, su estado civil y el número de hijos.



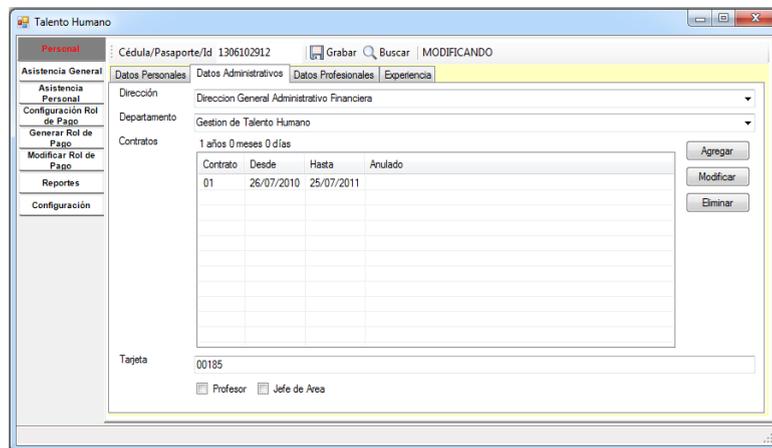
The screenshot shows a web application window titled 'Talento Humano'. The main content area is a form for 'Datos Personales' (Personal Data). The form is currently in 'MODIFICANDO' (Editing) mode. The user's ID is 'Cédula/Pasaporte/Id 1306102912'. The form fields are as follows:

Field	Value
Apellido Paterno	ALARCON
Apellido Materno	COBEÑA
Nombres	PATRICIO XAVIER
Fecha Nacimiento	29/09/1968 (46 años(s) 9 mes(es) 28 día(s))
Sexo	MASCULINO
Nacionalidad	ECUATORIANA
Discapacidad	Ninguna
Carnet	NO
Número	0
Porcentaje	00.00
Email	alarcon@gmail.com
Teléfono	2455611
Celular	0987654321
Dirección	AV 5 DE JUNIO
Casado	NO
Hijos	00

**FIGURA # 27: Datos Personales**

### 7.3.3.2. Datos Administrativos

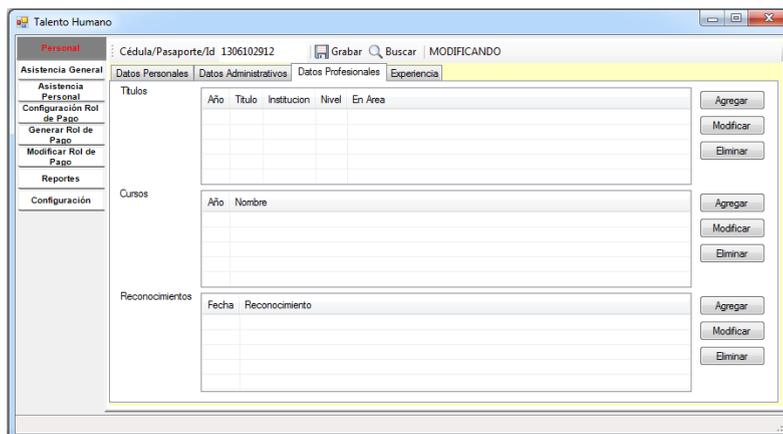
Datos Administrativos se despliega para conocer o ingresar información referente a la Dirección Administrativo y Departamento donde está situado el personal encontrándose dentro de ésta, información sobre el Contrato del personal.



**FIGURA # 28: Datos Administrativos**

### 7.3.3.3. Datos Profesionales

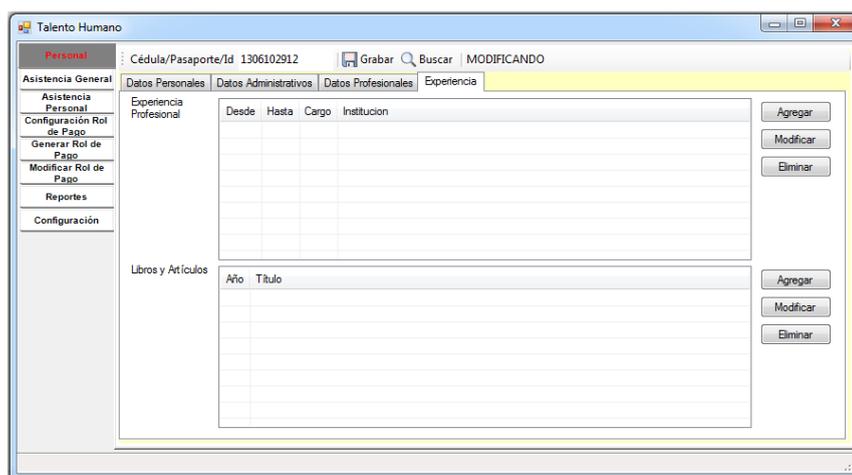
En Datos Profesionales, se podrán conocer o añadir datos sobre tres campos que encontramos, y son: Títulos (Información sobre los títulos que contenga el personal), Cursos (Cursos que hayan realizado) y Reconocimientos (Si ha obtenido algunos méritos). Todas estas opciones permiten agregar, modificar o eliminar información en estos campos.



**FIGURA # 29: Datos Profesionales**

### 7.3.3.4. Personal – Experiencia

Experiencia, se encuentra compuesta por Experiencia Profesional y Libros. En experiencia Profesional se puede agregar, modificar y eliminar información sobre su experiencia laboral, esto es, los trabajos que ha tenido el personal durante su vida profesional y en la opción Libros se puede agregar si ha publicado uno o varios libros.



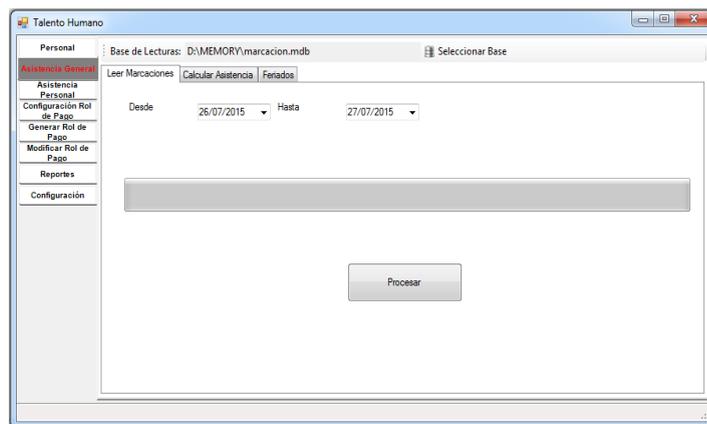
**FIGURA # 30: Personal – Experiencia**

### 7.3.4. Asistencia General

En Asistencia General está compuesto por tres pestañas llamadas: **Leer Marcaciones, Calcular Asistencia y Feriados.**

#### 7.3.4.1. Leer Marcaciones

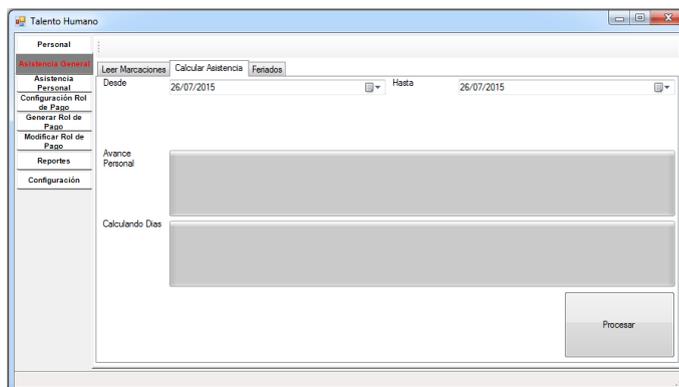
Para tener acceso, primero debe darse clic en Seleccione la Base, que permitirá seleccionar la base diseñada en Microsoft Access que es extraída directamente del dispositivo Biométrico, luego, se escogen las fechas desde y hasta, en las que se quiera realizar la lectura. Estos resultados se verán reflejados en Asistencia Personal en la pestaña Asistencia



**FIGURA # 31: Asistencia General – Leer Marcaciones**

#### 7.3.4.2. Asistencia General – Calcular asistencia

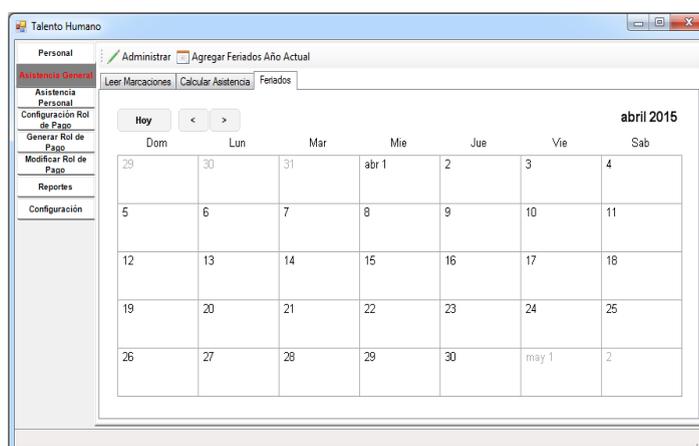
Para calcular la asistencia es necesario que se procesen las lecturas de las marcaciones, En la pestaña **Calcular Asistencia**, seleccionamos las fechas Desde y Hasta cuando se quiere ejecutar el cálculo de la asistencia del personal. Luego de escogido el término de tiempo se solicita procesar que es una opción que contiene este diseño dentro de la pestaña Calcular Asistencia



**FIGURA # 32: Calcular Asistencia**

### 7.3.4.3. Feriados

En **Feriados** se añadirán los días de feriados dependiendo del cronograma de la universidad y las fechas festivas que contenga el presente año. Esta ventana se despliega haciendo posible que un calendario mensual nos proporcione la información, pero, también permite ingresar datos que se reflejen cuando se consulta en él sobre los feriados estipulados por la ley y por las autoridades de la universidad.



**FIGURA # 33: Feriados**

### 7.3.5. Asistencia Personal

Asistencia Personal está compuesto por 4 pestañas: **Horarios, Asistencias, Justificaciones y Vacaciones.**

#### 7.3.5.1. Horarios

En Horarios se permite ingresar, el horario del personal, donde se indica cuantas horas de trabajo tienen a la semana incluyendo los fines de semana en caso de que labore uno de esos días. Este permite añadir varios horarios por día, dependiendo de las horas que contenga cada docente, administrativo o auxiliar de servicio.

The screenshot shows a web application window titled 'Talento Humano'. The main content area is divided into tabs: 'Horarios', 'Asistencia', 'Justificaciones', and 'Vacaciones'. The 'Horarios' tab is active. At the top, there is a search bar with the text 'Cédula' and a search icon, followed by date filters 'Desde 26/07/2015' and 'Hasta 26/07/2015', and a 'Refrescar' button. Below the tabs, there is a 'Grabar' button and a table with the following structure:

Day	Hours/Minutes	Buttons
Domingo	0 Horas 0 minutos	+ -
Lunes	0 Horas 0 minutos	+ -
Martes	0 Horas 0 minutos	+ -
Miércoles	0 Horas 0 minutos	+ -
Jueves	0 Horas 0 minutos	+ -
Viernes	0 Horas 0 minutos	+ -
Sábado	0 Horas 0 minutos	+ -

**FIGURA # 34: Horarios**

### 7.3.5.2. Asistencia

En **Asistencia** se podrá observar y constatar las horas laboradas cada día con el tiempo de retraso, si fuera el caso, y el tiempo justificado en caso de que lo haya.

Los días que se resaltan con fuente color rojo son días en los cuales el personal no debe asistir de acuerdo a su horario estipulado o pueden ser días feriados.

También esta ventana nos permite observar las marcaciones que el personal realiza en el dispositivo biométrico, es decir, son reflejadas en la parte inferior de la ventana abierta, esto se debe a que ya se realizó una **lectura de marcaciones**.

Mes / Día	Tiempo Laborable	Tiempo Laborado	Tiempo Falta	Tiempo Justificado
07/27	7 horas 0 minutos	2 horas 20 minutos	4 horas 40 minutos	0 horas 0 minutos
07/28	10 horas 0 minutos	0 horas 52 minutos	9 horas 8 minutos	0 horas 0 minutos
07/29	8 horas 0 minutos	4 horas 38 minutos	3 horas 22 minutos	0 horas 22 minutos
07/30	0 horas 0 minutos			
07/31	0 horas 0 minutos			
08/01	8 horas 0 minutos	0 horas 0 minutos	8 horas 0 minutos	0 horas 0 minutos
08/02	8 horas 0 minutos	0 horas 0 minutos	8 horas 0 minutos	0 horas 0 minutos
08/03	7 horas 0 minutos	3 horas 0 minutos	4 horas 0 minutos	0 horas 0 minutos
08/04	10 horas 0 minutos	1 horas 22 minutos	8 horas 38 minutos	0 horas 0 minutos
08/05	8 horas 0 minutos	1 horas 12 minutos	6 horas 48 minutos	0 horas 0 minutos

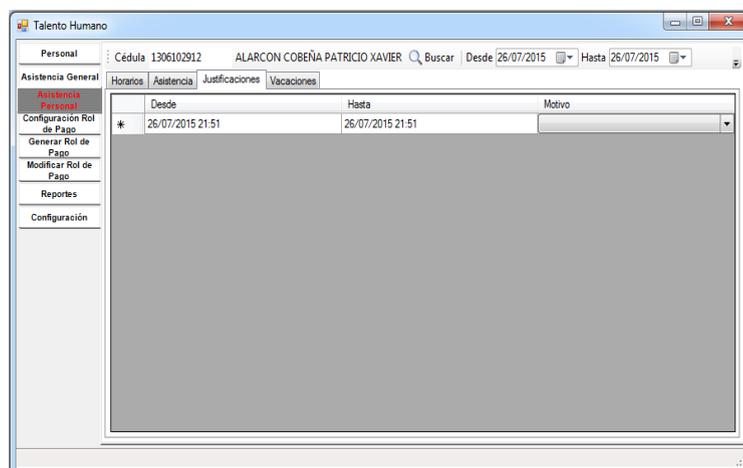
Entrada Reloj	Salida Reloj	Marca Entrada	Marca Salida	Tiempo Laborable	Tiempo Laborado	Tiempo Falta	Tiempo Justificado	Tiempo Extra	Aprobar Horas Extra
07:00	11:00	00:00	00:00	4:00	0:00	4:00	0:00	0:00	<input type="checkbox"/>
15:00	18:00	15:40	19:25	3:00	2:20	0:40	0:00	0:00	<input type="checkbox"/>

**FIGURA # 35: Asistencia**

### 7.3.5.3. Justificaciones

En la pestaña **Justificaciones** se permitirá realizar las respectivas justificaciones del personal por parte de la analista de Talento Humano, luego del cumplimiento del proceso interno correspondiente a este aspecto.

El registro de las justificaciones que incluye el sistema contempla las fechas Desde y Hasta, además, se presenta la opción Motivo, que ubica las razones por las cuales el empleado tiene su permiso, que hace referencia a las asistencias a Seminarios, representaciones u otros considerados por la autoridad.

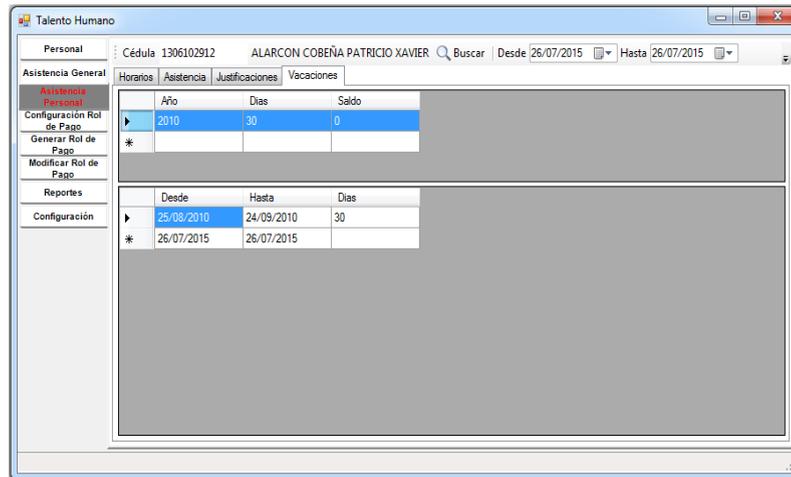


**FIGURA # 36: Justificaciones**

### 7.3.5.4. Vacaciones

En la pestaña **Vacaciones**, se aprecia en la parte superior el año, los días y el saldo que tiene de vacaciones el personal consultado, mientras que, en la parte inferior se puede apreciar y constatar la fecha salida y de reintegro del personal, que ubica de manera automática los días de vacaciones a los que tiene derecho.

Esta ventana es de exclusiva responsabilidad del Departamento de Talento Humano, por lo tanto, la información que se ingresa está enmarcada dentro de los procesos internos.



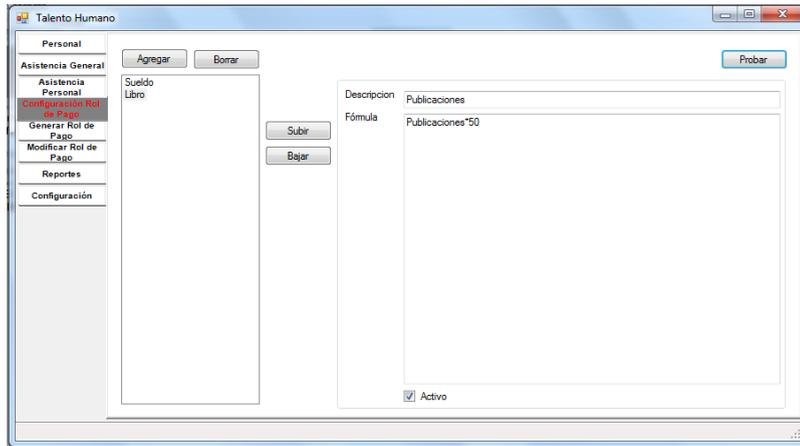
**FIGURA # 37: Vacaciones**

### 7.3.6. Configuración Rol de Pago

La **Configuración de Rol de Pago**, es la opción que permite generar cada mes los roles de pago, aquí se pondrán los rubros correspondientes a cada empleado o trabajador y los descuentos apropiados con sus respectivas fórmulas. Incluye las opciones Subir y Bajar que permitirán cambiar el orden de los rubros de acuerdo a los requerimientos del Departamento de Talento Humano, concordante con las disposiciones del Área Financiera.

Además cuenta con la opción Descripción en donde se describe mejor el nombre del rubro. Todo lo que se ingrese en esta Ventana se verá reflejado en

### **Modificar Rol de Pago**

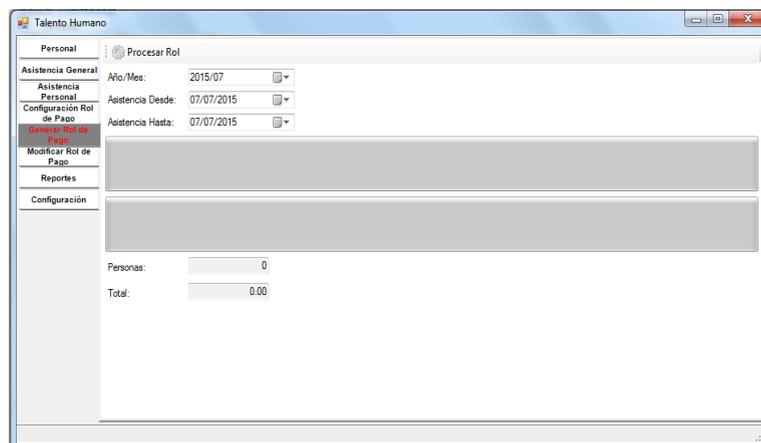


**FIGURA # 38: Configuración Rol de Pago**

### 7.3.7. Generar Rol de Pago

En **Generar Rol de pago** es donde se pondrá la fecha Desde y Hasta correspondiente al mes que permita poder procesar los roles de todo el personal de la institución. En la parte inferior se podrá observar el número de personas a las que se les ha generado su rol y el Valor que representa los roles generados.

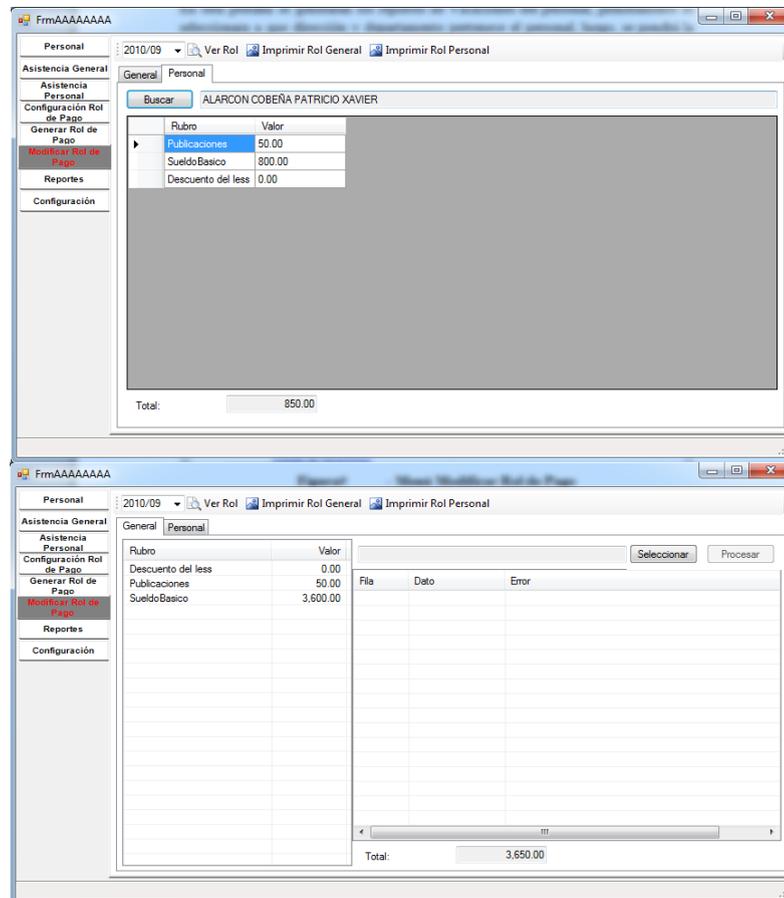
Posterior a esto se accederá al menú **Modificar Rol de Pago**. Aquí también se podrá imprimir el Rol dando clic en el botón Imprimir Rol.



**FIGURA # 39: Generar Rol de Pago**

### 7.3.8. Modificar Rol de Pago

En Modificar Rol de Pago nos encontramos con: Rol de Pago General y Rol de Pago Personal, en donde se podrá observar todos los rubros y descuentos que fueron puestos anteriormente en Configuración Rol de Pago. Esta ventana también nos permite añadir los descuentos que no son fijos para cada personal de trabajo, es decir, los rubros que tienen un valor diferente para cada empleado. Aquí también se podrá generar un reporte sobre los roles de manera general y de manera personal.



**FIGURA # 40: Modificar Rol de Pago**

### 7.3.9. Reportes

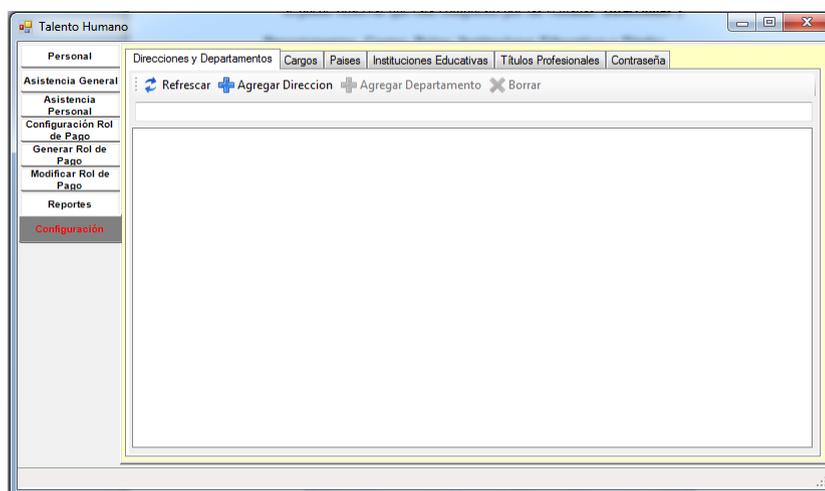
Los reportes están compuestos de cuatro Pestañas que resultan de la información ingresada anteriormente que han sido detalladas de manera minuciosa, de la misma manera que será descrita en **Reportes** en el **Capítulo IV**, que está dedicado puntualmente a esta temática.

### 7.3.10. Configuración

Se puede observar que está compuesto por las ventanas: **Direcciones y Departamentos**, **Cargos**, **Países**, **Instituciones Educativas** y **Títulos Profesionales**.

#### 7.3.10.1. Direcciones y Departamentos

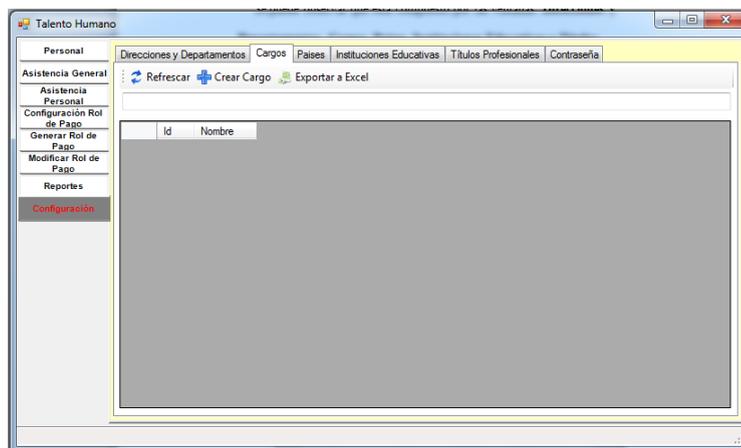
En **Direcciones y Departamentos** se podrá agregar las direcciones con sus respectivos departamentos, en caso de que exista error al ingresar un departamento o dirección exista la opción de Borrar.



**FIGURA # 41: Direcciones y Departamentos**

### 7.3.10.2. Cargos

En **Cargos** se encuentran agregados los cargos del personal, que los identifica de sus labores al personal, estos pueden ser exportados a una hoja de cálculo en Excel si hubiera necesidad de hacerlo. El sistema permite crear nuevos cargos en caso de ser necesario y permite luego refrescar la información nueva generada.

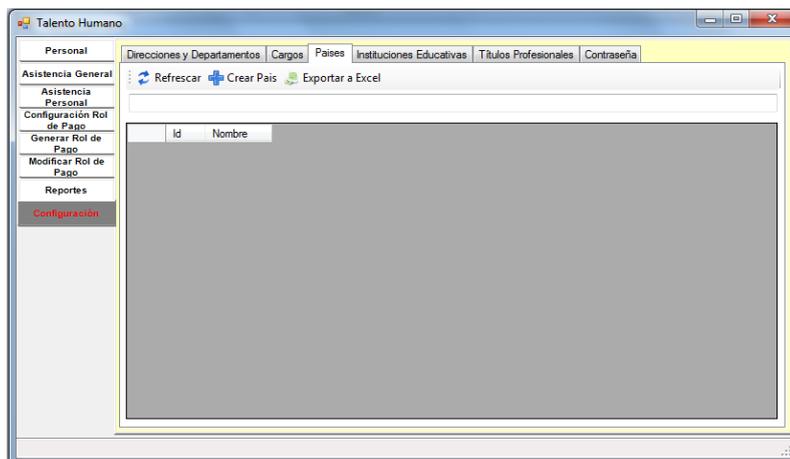


**FIGURA # 42: Cargos**

### 7.3.10.3. Países

En **Países**, se encuentran agregados los países de origen de cada miembro que forma parte del personal, esta información puede ser exportada a una hoja de cálculo en Excel si hubiera necesidad de hacerlo.

El sistema permite crear nuevos países en caso de ser necesario y permite luego refrescar la información nueva generada.

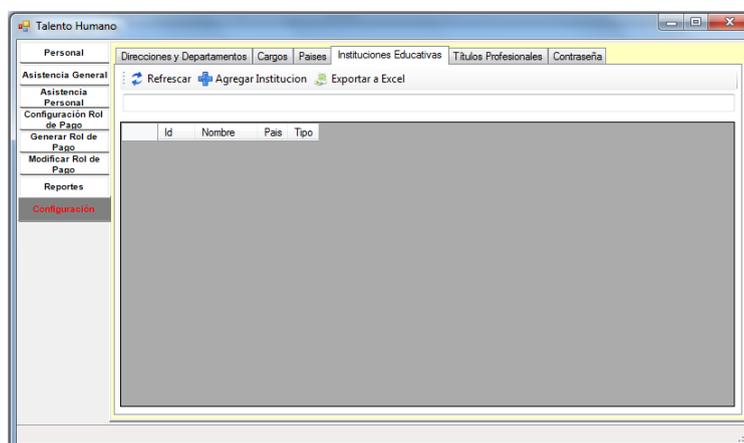


**FIGURA # 43: Países**

#### 7.3.10.4. Instituciones Educativas

En **Instituciones Educativas** en donde el personal haya realizado sus estudios en todos sus niveles, se registra en cada uno de los casos o instituciones el país al que esta pertenece. Esta información puede ser exportada a una hoja de cálculo en Excel si hubiera necesidad de hacerlo.

El sistema permite agregar instituciones en caso de ser necesario y permite luego refrescar la información nueva generada.

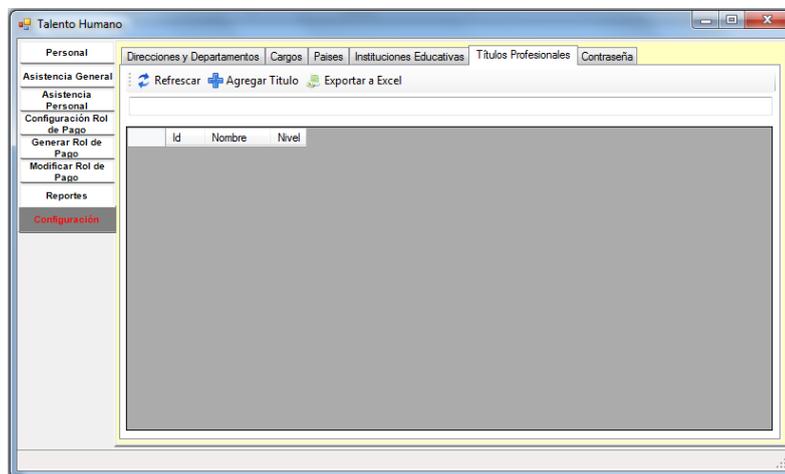


**FIGURA # 44: Instituciones Educativas**

### 7.3.10.5. Títulos Profesionales

En **Títulos Profesionales** están ingresados los títulos profesionales que tenga el personal. Exista la opción para poder ingresar el Nivel a que corresponde el título profesional. Esta información puede ser exportada a una hoja de cálculo en Excel si hubiera necesidad de hacerlo.

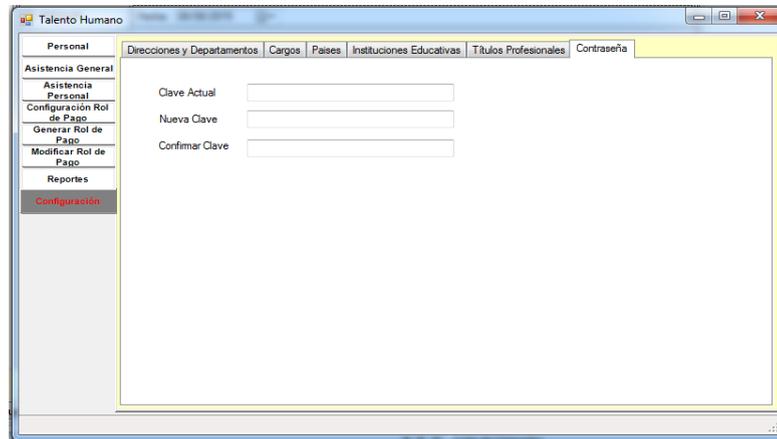
El sistema permite agregar títulos en caso de ser necesario y permite luego refrescar la información nueva generada.



**FIGURA # 45: Títulos Profesionales**

### 7.3.10.6. Contraseña

En contraseña se podrá cambiar la contraseña de usuario para obtener una mayor seguridad y más privacidad para la información que se maneja en el sistema.



The screenshot displays a web application window titled "Talento Humano". The interface is divided into a sidebar menu on the left and a main content area. The sidebar menu includes the following items: "Personal", "Asistencia General", "Asistencia Personal", "Configuración Rol de Pago", "Generar Rol de Pago", "Modificar Rol de Pago", "Reportes", and "Configuración" (highlighted in red). The main content area has a tabbed interface with tabs for "Direcciones y Departamentos", "Cargos", "Paises", "Instituciones Educativas", "Títulos Profesionales", and "Contraseña" (selected). Below the tabs, there are three text input fields: "Clave Actual", "Nueva Clave", and "Confirmar Clave".

**FIGURA # 46: Contraseña**

## CAPÍTULO IV

### **7.4. Generar reportes de roles de pagos, vacaciones, permisos y vigencia de contratos.**

#### **7.4.1. MonoReport**

El sitio web de MonoReport, nos proporciona el concepto de esta herramienta, en inglés, que ha sido traducido por el autor de la tesis con la ayuda del traductor google, lo que da como resultado que:

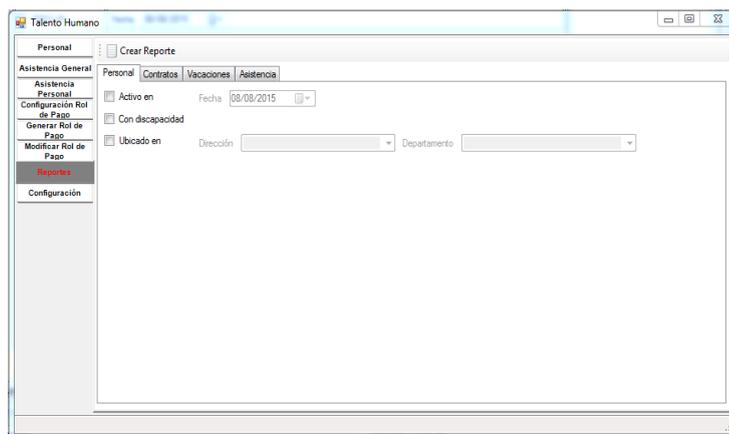
MonoReport es un generador de informes desarrollado en la plataforma .NET (C #). MonoReport trabaja con .Net Framework 2.0 o superior (marco Mono será apoyado en breve). MonoReport redistribuible en tiempo de ejecución es de bibliotecas libres. Así MonoReport podría ser utilizado en cualquier entorno que pueden utilizar las bibliotecas de .NET Framework. (MonoReport, 2008)

#### **7.4.2. Generalidades**

En el sistema se puede constatar la lista de reportes que genera el Sistema Integrado para hacer más ágil el trabajo que realiza el Departamento de Talento Humano de la Institucion, y entre los que encontramos tenemos: Roles de Pago General y Personal, Reportes de Vacaciones de cada empleado y trabajador, Reportes de Asistencia, y los Reportes de Contratos.

### 7.4.2.1. Reporte del Personal

En reportes Personal se realizara reportes sobre el personal de trabajo que laboran en la universidad. Aquí se seleccionará la fecha desde cuando está activo el personal, en la opción Con Discapacidad se seleccionará en caso de que se quiera incluir en el reporte los empleados con discapacidad. También se pondrá la ubicación de dirección y departamento donde se encuentren.



**FIGURA # 47: Reportes - Personal**

#### 7.4.2.1.1. Generación de Reporte del Personal

El reporte del personal comprende la información de cada persona, esto se refiere a: Datos personales (número de cédula, nombres, apellidos, sexo, discapacidad, entre otros). Con esta información se pueden ubicar los perfiles.

Listado de Personal						
Departamento: Gestión de Talento Humano						
1	130743003	COBENIA	PATRICIO XAVIER	ECUATORIANA	09/05/76	03PATRICIO@ANUPROF.EDU.EC
2	130743003	SAZ	MARCO VINICIO	ECUATORIANA	08/12/72	03MARCO@ANUPROF.EDU.EC
3	130913003	ROLDAN	SAUL VICENTE	ECUATORIANA	07/14/76	03SAUL@ANUPROF.EDU.EC
4	130913003	LEON	MANUEL ENRIQUE	ECUATORIANA	04/07/76	03MANUEL@ANUPROF.EDU.EC
5	Total: Gestión de Talento Humano					

**FIGURA # 48: Reportes del Personal**

### 7.4.2.2. Reportes Rol de pago

Como ya se explicó en la figuras #36 y #37, para generar un rol simplemente se tiene que dar clic en el botón imprimir rol General o imprimir Rol Personal.

#### 7.4.2.2.1. Generación de Reportes Rol de Pago General

En el Rol de Pago General contiene información sobre los rubros ingresados por la Analista de Talento Humano y que corresponde a cada empleado o trabajador y dentro de este podremos apreciar los descuentos correspondientes a rubros determinados y que varían dependiendo del personal.

Rol de Pago			
Empleado	Salario	Descuentos	Salario Neto
130743003 ALARCON COBENIA PATRICIO XAVIER	800.00	30.00	770.00
130743003 ALARCON COBENIA PATRICIO XAVIER	900.00	0.00	900.00
130913003 ALVARADO ROLDAN SAUL VICENTE	1,300.00	0.00	1,300.00
130913003 ANDRADE LEON MANUEL ENRIQUE	700.00	0.00	700.00
Total:		2,700.00	900.00

**FIGURA # 49: Generación de reportes Rol de Pago General**

#### 7.4.2.2.2. Generación de reportes de Rol de Pago Personal

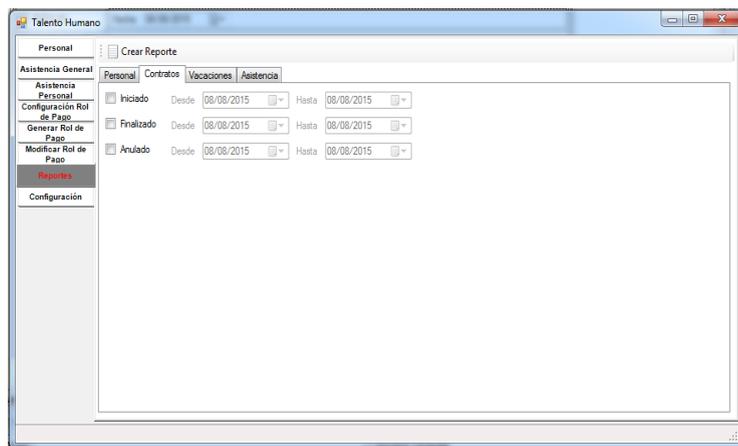
En el Rol de Pago Personal se puede observar la información correspondiente a cada empleado o trabajador y se presenta de manera vertical que permite observar de mejor forma el detalle de cada trabajador y empleado. Los roles General y personal suman el primero por rubro y el segundo totales.

ROL DE PAGO	
1	
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19
21	20
22	21
23	22
24	23
25	24
26	25
27	26
28	27
29	28
30	29
31	30
32	31
33	32
34	33
35	34
36	35
37	36
38	37
39	38
40	39
41	40
42	41
43	42
44	43
45	44
46	45
47	46
48	47
49	48
50	49
51	50
52	51
53	52
54	53
55	54
56	55
57	56
58	57
59	58
60	59
61	60
62	61
63	62
64	63
65	64
66	65
67	66
68	67
69	68
70	69
71	70
72	71
73	72
74	73
75	74
76	75
77	76
78	77
79	78
80	79
81	80
82	81
83	82
84	83
85	84
86	85
87	86
88	87
89	88
90	89
91	90
92	91
93	92
94	93
95	94
96	95
97	96
98	97
99	98
100	99

FIGURA # 50: Generación de reportes Rol de Pago Personal

#### 7.4.2.3. Reportes de Contratos

En ella se podrán realizar reportes referentes a lo que compete a los contratos del personal para ello se definirán las fechas de inicio y final de los contratos del personal que se querrá observar en el reporte. En caso de que también se quiera ver los contratos anulado se seleccionará la opción Anulado.



**FIGURA # 51: Reportes – Contrato**

#### 7.4.2.3.1. Generación de Reporte de Contratos

El reporte de Contratos nos ayuda a conocer la información correspondiente al ingreso del personal, tipo de contrato, el cargo que ostenta, el departamento donde labora y la finalización del contrato.

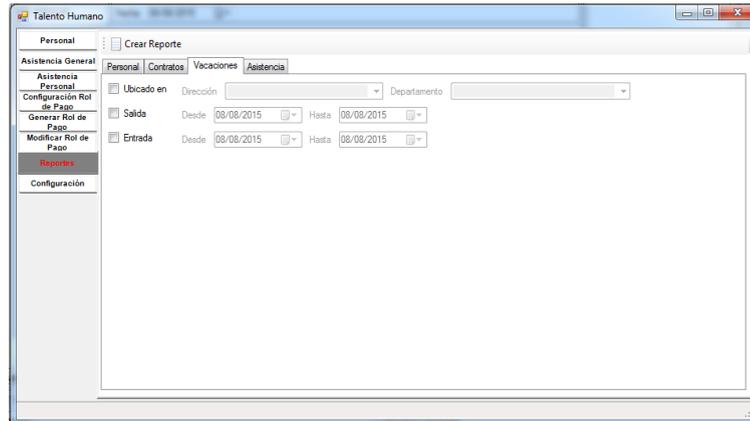
The screenshot shows a Microsoft Excel spreadsheet with the title 'Listado de Contratos'. The spreadsheet has columns for ID, Nombre, Apellido Paterno, Apellido Materno, Departamento, Cargo, Fecha de Inicio, Fecha de Finalización, and Estado. A single data row is visible with the following information:

ID	Nombre	Apellido Paterno	Apellido Materno	Departamento	Cargo	Fecha de Inicio	Fecha de Finalización	Estado
1306102912	ALARCON	COBEÑA	PATRICIO XAVIER	Dirección General de Gestión de Talento		25/07/2010	24/07/2011	DOCENTE - Con Relación de Des

**FIGURA # 52: Generación de Reporte de Contratos**

#### 7.4.2.4. Reportes de Vacaciones

Los reportes de vacaciones constaran de la fecha de salida a los días de gozo de cada empleado, y la fecha de reintegración a la institución de los mismos.



**FIGURA # 53: Reportes de Vacaciones**

##### 7.4.2.4.1. Generación de reportes de Vacaciones

Este reporte que formará parte del registro de cada empleado o trabajador es importante porque mantiene actualizado la información del Departamento de Talento Humano que solo con un clic puede acceder a una información que no estará solamente en físico como es actualmente. Este reporte consta de Fecha de salida a Vacaciones y la Fecha de Entrada, con la posibilidad de conocer los días de saldo o por gozar de cada personal.

Nombre	Periodo	Días	Total Utilizados
ALARCON COBENA PATRICIA	2010	30	26
ALAVA LAZ MARCO VINICIO	2010	30	27

**FIGURA # 54: Reportes de Vacaciones**

#### 7.4.2.5. Reportes de Asistencia

Aquí se generarán reportes de asistencia dependiendo de las opciones que uno haya seleccionado, se deberá escoger las fechas desde que se encuentra activo, luego su ubicación, y los campos en los que se requiera observar la información en el reporte.

**FIGURA # 55: Reportes – Asistencia**

### 7.4.2.5.1. Generación de Reportes de Asistencia

El reporte de la asistencia del personal comprende el detalle del tiempo laborable y laborado, y proporciona la información sobre las faltas, atrasos y justificaciones, por lo que, resulta eficiente y rápido.

Asistencia								
Fecha	Horario de entrada	Horario de salida						
25/07/2010		0:00	0:00	0:00	0:00	0:00	0:00	0:00
26/07/2010	7:00	12:00	0:00	0:00	5:00	0:00	5:00	0:00
	16:00	19:00	18:20	20:15	3:00	0:40	2:20	0:00
27/07/2010	8:00	13:00	0:00	0:00	4:00	0:00	4:00	0:00
	15:00	19:00	15:40	19:25	4:00	3:20	0:40	0:00
28/07/2010	7:00	13:00	0:00	0:00	4:00	0:00	4:00	0:00
	15:00	19:00	15:40	19:25	4:00	3:20	0:40	0:00
29/07/2010	7:00	13:00	0:00	0:00	4:00	0:00	4:00	0:00
	15:00	19:00	15:40	19:25	4:00	3:20	0:40	0:00
30/07/2010	7:00	13:00	0:00	0:00	4:00	0:00	4:00	0:00
	15:00	19:00	15:40	19:25	4:00	3:20	0:40	0:00
31/07/2010	8:00	13:00	0:00	0:00	5:00	0:00	5:00	0:00
	16:00	19:00	0:00	0:00	3:00	0:00	3:00	0:00

FIGURA # 56: Generación de Reportes de Asistencia

## **7.5. Marco Teórico Ético y Legal**

### **Tipos de Contrato de la Universidad San Gregorio de Portoviejo**

De acuerdo a lo que se encuentra establecido en Reglamento Interno de Trabajo de la Universidad san Greogrio de Portoviejo, en el Capítulo III, De la Celebración, duración y terminación del Contrato de Trabajo, artículo 7, queda establecido que todos los contratos de trabajos que celebre la Universidad serán por escrito y estarán sujetos a lo que prescribe el Reglamento Interno y el Código de Trabajo.

**Artículo 7.1**, establece los tipos de Contratos, siendo estos:

**Contrato a Prueba.-** El mismo que tendrá una duración de hasta 90 días, tiempo en el cual el empleador y trabajadores podrán darlo por terminado, por convenir a sus intereses, sujetándose estrictamente a lo que determina el Art. 15 de Código del Trabajo en vigencia

**Contrato a Plazo Fijo.-** Se entenderá como contrato a plazo fijo, aquel que tenga una duración mínima de un año y no renovable de conformidad a lo que establece el Art. 184 inciso segundo, con las excepciones establecidas en los literales a) b) c) d) e) f) g) y h) del Art.14 del Código del Trabajo.

**Contrato Indefinido.-** Se entenderá como contrato de tiempo indefinido aquellos contratos que por no existir objeción por ninguna de las partes en el tiempo mínimo de un año de labores ininterrumpidamente sobrepasan este tiempo.

**Contrato Eventual.-** Es el que se celebrara para atender una mayor demanda de servicios en actividades habituales del empleador, el contrato eventual no podrá tener una duración mayor de ciento ochenta días continuos, dentro de un lapso de trescientos sesente y cinco días.

### **Artículo 7.2 de la terminación de los contratos**

Los contratos celebrados entre la Universidad San Gregorio de Portoviejo; y, sus empleados y trabajadores podrá terminar por:

- a) Desahucio de conformidad al Art. 184 del Código de Trabajo.
- b) Visto bueno propuesto por la empleadora, de conformidad al Art.172 del Código de Trabajo.
- c) Visto bueno propuesto por el/la empleado(a) y trabajador (a), de conformidad al Art. 173 del mismo cuerpo de Ley; y,
- d) Por las casuales determinadas en el Art. 169 del Código del Trabajo.

### **Capítulo IV jornadas de trabajo, horarios de labor, asistencia del personal y vacaciones.**

#### **Artículo 10**

Las jornadas de labor se establecerán de acuerdo a lo determinado en el Art. 47 del Código del Trabajo, esto es una jornada máxima de OCHO HORAS DIARIAS, sin exceder las CUARENTA HORAS SEMANALES.

Los turnos de labor los determinara la universidad y los comunicará con la debida anticipación a sus empleados y trabajadores, los horarios, turnos y modificaciones serán previamente aprobados por la Dirección Regional del Trabajo. En lo referente a vacaciones, los/las empleados(as) y trabajadores(as) tendrán derecho a gozar anualmente de un período ininterrumpido de quince días de descanso, incluido los días no laborales, de conformidad a lo dispuesto en los Arts. 69 a 77 del Código de Trabajo.

No habiendo señalamiento en el contrato escrito del período en el que el/la empleado(a) y trabajador(a) comenzara a gozar las vacaciones, la universidad hara conocer al empleado(a) o trabajador(a) con tres meses de anticipación, el periodo en el que concederá las vacaciones, esto se lo realizará a través de un cronograma de vacaciones para sus empleados (as) y trabajadores (as), en caso que el/la empleado (a) o trabajador (a) tenga derecho a un número mayor de 15 días, la Universidad tendrá la libertad de decidir si el /la empleado(a) o trabajador(a) toma descanso por ese periodo o se cancela el valor correspondiente.

### **Artículo 12**

La Universidad San Gregorio de Portoviejo exigirá a los/las empleados(as) y trabajadores(as) que presten sus servicios, la incorporación a sus puestos de trabajo exactamente a la hora fijada para sus respectivas labores. El personal deberá estar en sus puestos de trabajo a la hora que inicie su labor y permanecerá en ellos hasta la hora de salida.

### **Artículo 13**

Los/las empleados (as) y trabajadores (as) están obligados a registrar la hora de entrada y salida de su trabajo, de acuerdo con el sistema establecido por la universidad. Los/las empleados (as) y trabajadores (as) que no hubieran registrado su asistencia, deberán reportarse directamente ante su empleador o ante el encargado o jefe de Gestión de Talento Humano. En caso de no hacerlo con un atraso de sesenta minutos, se considerara como si hubiera fallado al trabajo en la mañana o en la tarde, en la que no hubiera registrado su ingreso.

#### **Artículo 14**

El personal no debe abandonar sus labores antes del término de la jornada, salvo que tenga permiso del jefe inmediato; y, que este sea ratificado por el jefe de Gestión de Talento Humano. En el caso de comprobarse una salida injustificada, el contraventor será multado hasta con el 10% de su remuneración diaria. En caso de haberse comprobado más de tres abandonos injustificados, esto será considerado como faltas repetidas e injustificadas de acuerdo al Art. 172 numeral 1 y 2 del Código del Trabajo, por lo que será casual suficiente para solicitar el visto bueno ante la Inspectoría del Trabajo.

#### **Artículo 15**

Es obligación de los/las empleados (as) y trabajadores (as) que no puedan concurrir al trabajo, comunicar este hecho a su jefe inmediato o jefe de Gestión de Talento Humano solicitando el permiso correspondiente, o en su defecto comunicarlo directamente a su empleador, a fin de proveer el reemplazo que fuere necesario.

#### **Artículo 16**

Todo trabajo o prestación de servicio efectuado fuera de la jornada cotidiana de servicio efectuado fuera de la jornada cotidiana de labor, deberá ser previamente acordada con los trabajadores y los representantes de la Universidad San Gregorio de Portoviejo, y su pago será abonado conforme al Código del Trabajo. Si por causas provenientes de fuerza mayor o caso fortuito, tales como suspensión de energía eléctrica, fenómenos atmosféricos, incendios, paros de transporte que interrumpan el trabajo y la prestación de servicios de la Universidad San Gregorio

de Portoviejo, la universidad podrá pedir la recuperación de las horas perdidas, sin considerar ese trabajo como horas suplementarias o extraordinarias de conformidad al Art. 60 del Código de Trabajo.

### **Artículo 17**

Los permisos por asuntos personales pueden ser descontados de las vacaciones anuales correspondientes o de la remuneración anual correspondiente o de la remuneración a criterio del empleador. Los mencionados permisos para atender asuntos de tipo personal o familiar y que puede ir desde menos de una hora hasta días completos de trabajo, serán concedidos por el jefe de gestión del talento humano de la universidad en situaciones de calamidad doméstica, las mismas que serán consideradas como tales, en los siguientes casos:

Situaciones familiares o personales

Matrimonio

Nacimiento de hijos

Permiso con Cargo a Vacaciones

El Sistema Integral del Departamento de Talento Humano de la USGP, aporta la información correspondiente mediante los reportes que emite para que las autoridades, Jefes de División o Departamentos puedan cumplir con sus funciones, por lo que es coherente y de utilidad para el cumplimiento de lo establecido en el Marco legal correspondiente.

## **8. Procedimiento Metodológico**

### **8.1. Modalidad de la investigación**

La presente investigación estuvo basada en la modalidad de Campo, “in situ”; ya que, a las personas involucradas en la investigación, responsable y analista del departamento del Talento humano de la Universidad San Gregorio de Portoviejo, se les realizó las indagaciones correspondientes mediante entrevistas para recolectar la información necesaria.

**Modalidad bibliográfica.-** Se aplicó lectura científica tanto en texto, como en internet, de tal manera que se pudo recabar información científica que se relacionó con el problema de investigación para obtener un conocimiento nuevo.

### **8.2. Tipo de investigación**

Esta investigación fue de tipo Descriptivo en donde el fenómeno o problema se analizó en la ciudad de Portoviejo con los involucrados con el desarrollo tecnológico de la Universidad San Gregorio de Portoviejo, que laboran en ella con responsabilidad que implica el desarrollo de un Sistema Integral Automatizado, en este momento se permite la aplicación del conocimiento adquirido como son; la descripción, la explicación, el control para culminar en la aplicación que puede suceder en lo posterior.

### **8.3. Métodos**

Para el desarrollo de esta investigación se utilizará los siguientes métodos:

#### **8.3.1. Método inductivo-deductivo**

Inductivo, porque analizará aspectos particulares para llegar a los generales, y Deductivo, ya que se parte de lo general a lo particular.

#### **8.3.2. Método descriptivo**

Porque se describe la problemática planteada de la investigación.

#### **8.3.3. Método científico**

Ya que permite la producción de conocimientos en las ciencias, se basará en la medición, sujetos o principios específicos del razonamiento.

#### **8.3.4. Método analítico-sintético**

Porque se obtiene un análisis integral del problema. Toda la información recopilada mediante el alcance de los objetivos y será sintetizado mediante conclusiones al final de la investigación.

### **8.4. Técnicas**

Se aplicaron las siguientes técnicas:

#### **8.4.1. Técnica Bibliográfica**

A través de la recopilación de información científica valida que permitió disponer de los referentes teóricos con respecto al Sistema Integral Automatizado.

#### **8.4.2. Técnica de la entrevista,**

Ya sé que entrevistó a la responsable del departamento del Talento Humano y a la Analista del mismo.

#### **8.5. Población**

Para el desarrollo del Sistema se consideró al personal del Departamento del Talento Humano de la Universidad San Gregorio de Portoviejo.

#### **8.6. Muestra**

Se tomó en cuenta la totalidad de la población.

## 9. Presupuesto

El presupuesto requerido para la adquisición del software para la instalación del Sistema Integrado para la realización de las tareas del Departamento de Talento Humano de la Universidad San Gregorio de Portoviejo, contiene los rubros que se detallan a continuación:

<b>Nombre de la Tarea</b>	<b>Descripcion</b>	<b>Costo USD</b>
2 Computadoras	Intel Core i3	\$1,100,00
4 Biométrico Huella Dactilar	Biotech	\$1,200,00
1 Servidor	Computadora HP	\$3,920,00
Instalacion de puntos red	4 puntos de red incluido material	\$400.00
Total		\$6,620.00

## **10. Conclusiones**

En la realización del presente trabajo de investigación se llegó a las siguientes conclusiones:

El Sistema Integrado desarrollado para la realización de las tareas en el Departamento de Talento Humano de la Universidad San Gregorio de Portoviejo, permite acceder a datos de forma inmediata y en menor tiempo y aporta en la toma de decisiones en base a registros, informes y reportes que este genera.

La implementación del Sistema, permitirá generar informes para elaborar roles y agilizar la emisión de reportes para el Departamento de Talento Humano, así como para los Departamentos de Financiero y Jurídico permitiendo la gestión de otros procesos.

La generación automática de reportes facilita obtener informes de los datos almacenados en el Sistema Integral con relación a cada empleado y/o trabajador.

La investigación realizada fue beneficiosa para el autor de este trabajo por los conocimientos adquiridos durante el desarrollo y para la Institución por el producto elaborado el cual da respuesta a los requerimientos del Departamento de Talento Humano para la automatización de sus procesos agilizando el trabajo del personal.

## **11. Recomendaciones**

Después de las conclusiones planteadas se recomienda lo siguiente:

Se debe validar el Sistema Integral automatizado por parte de las autoridades de la universidad para alcanzar la actualización tecnológica.

El manual técnico es destinado exclusivamente para el personal del Departamento Tecnológico ya que permitirá conocer la funcionalidad de la aplicación

El manual de usuario es destinado para los usuarios del Departamento del Talento Humano, ya que permitirá llevar el correcto manejo del Sistema Integral Automatizado.

El personal que utilice el Sistema debe ser capacitado para sacarle provecho a todas las ventajas que el sistema ofrece.

## 12. Bibliografía

Silberschatz, A., & F. Korth, H. (2002). *Fundamentos de Base de Datos*. Aravaca (Madrid): McGraw-Hill.

Andreu, J. (2011). Gestión de servidores web. En J. Andreu, *Servicios en red* (pág. 152). Editex.

Cabanellas, G. (2006). *Diccionario Jurídico Elemental*. Heliasta.

Caguana, L. (31 de Octubre de 2012). <http://6tocontabilida.blogspot.com>.  
Obtenido de [http://6tocontabilida.blogspot.com/2012/10/v-behaviorurldefaultvmlo\\_31.html](http://6tocontabilida.blogspot.com/2012/10/v-behaviorurldefaultvmlo_31.html)

Canchala Fernández, L. A. (s.f.). [msdn.microsoft.com](http://msdn.microsoft.com). Recuperado el sabado de mayo de 2015, de [msdn.microsoft.com](http://msdn.microsoft.com): <https://msdn.microsoft.com/es-es/library/bb972232.aspx>

Carme Martín Escofet. (2014). *Lenguaje SQL*. Obtenido de <http://ocw.uoc.edu>:  
[http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\\_M2109\\_02149.pdf](http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02149.pdf)

CentOS; Hughes, Jhonny;. (31 de Marzo de 2015). *CentOS*. Obtenido de <http://wiki.centos.org/>

Chiavenato, I. (2002). *Gestión del Talento Humano*. McGrawhill.

[community.sharpdevelop.net](http://community.sharpdevelop.net). (s.f.). <http://community.sharpdevelop.net>.  
Recuperado el 07 de 05 de 2015, de <http://community.sharpdevelop.net>:  
<http://community.sharpdevelop.net/blogs/mattward/articles/FeatureTourCreatingAWindowsApplication.aspx&usg=ALkJrhhbh5IoIJoAIqgolotD9Q7Xgp6gQ>

Coronel, C. (2011). *Base de datos, Diseño, Implementacion y Administracion*. Mexico: Cengage Learning Editores, S.A.

Fray León Osorio Rivera. (2008). *Base de Datos Relacionales*. ITM.

Gilfillan, I. (2003). *La Biblia MySQL*. Anaya Multimedia.

Hobbs, L. (1999). *Diseñar su propia Página Web* (Sengunda ed.). (V. P. Moreno, Trad.) Barcelona, España: Marcombo.

Ian Gilfillan. (2010). *La Biblia de Mysql*. Anaya.

- ic#code. (2012). *www.icsharpcode.net*. Retrieved Mayo 15, 2015, from [www.icsharpcode.net/OpenSource/SD/Default.aspx](http://www.icsharpcode.net/OpenSource/SD/Default.aspx)
- López R, J. R. (2011-2012). *Curso de Bases de Datos Documentais.*, (pág. 1).
- Microsoft. (2015). <https://msdn.microsoft.com>. Recuperado el 15 de Mayo de 2015, de [https://msdn.microsoft.com/es-es/library/hh425099\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/hh425099(v=vs.110).aspx)
- Microsoft. (2015). *Introduccion*. Obtenido de [https://msdn.microsoft.com/es-es/library/aa711604\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa711604(v=vs.71).aspx)
- MonoReport. (2008). *MonoReport*. Retrieved Mayo 15, 2015, from <http://www.monoreport.com/index.aspx>
- msdn.microsoft.com. (s.f.). <https://msdn.microsoft.com/>. Recuperado el 07 de 05 de 2015, de <https://msdn.microsoft.com/>: <https://msdn.microsoft.com/es-es/library/hh425099%28v=vs.110%29.aspx>
- Nogueras, J. D. (2014). *Sistemas de infromacion y base de datos en consumo*. España - Malaga: ic editorial.
- Pascual, A. Q. (30 de Octubre de 2014). <https://alejandroquiropascual.wordpress.com>. Obtenido de <https://alejandroquiropascual.wordpress.com>: <https://alejandroquiropascual.wordpress.com/2014/10/30/mysql-workbench/>
- Presman, R. (2010). *Ingeniería del software, un enfoque práctico. Séptima Edición*. Mc Graw-Hill.
- Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico*. McGraw-Hill.
- Quidonoz, P. C. (19 de abril de 2009). *Blog de Informática, Educación Tecnológica y TICs*. Obtenido de <http://profecarolinaquinodoz.com/principal/?tag=concepto-de-redes-clienteservidor>
- Quilaguy, W. (12 de Octubre de 2014). *Ingeniería de Software*. Obtenido de Modelo de desarrollo: <https://sites.google.com/site/ingenieriadesoftwareqw/home/modelos-de-desarrollo>
- Ramez, E., & Shamkant B, N. (2007). *Fundamentos de Sistemas de Bases de Datos*. España: PEARSON EDUCACIÓN S.A.

Santo, V. (2013). *Diccionario de ciencias jurídicas, políticas, sociales y de economía*. 20XII Grupo editorial.

Thill Eduardo. (2011). *BIOMETRÍAS 2*.

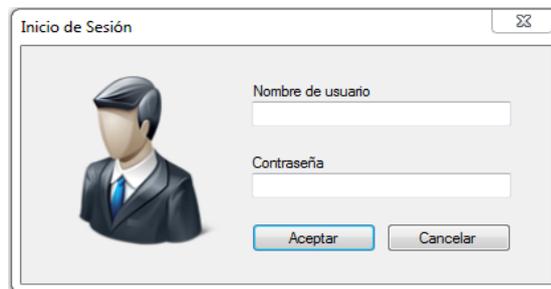
Werther, W. (2000).

## 13. Anexos

### MANUAL DE USUARIO

#### Inicio de sesión

El sistema cuenta con un inicio de sesión para asegurarse que solo la persona encargada lo manipule. Si se ingresa un usuario equivocado o una contraseña equivocada se mostrara un mensaje diciendo “usuario o contraseña incorrecta”



The screenshot shows a login window titled "Inicio de Sesión". On the left side, there is a 3D illustration of a man in a dark suit, white shirt, and blue tie. To the right of the illustration, there are two text input fields: the top one is labeled "Nombre de usuario" and the bottom one is labeled "Contraseña". Below these fields are two buttons: "Aceptar" (highlighted in blue) and "Cancelar". The window has a standard Windows-style title bar with a close button in the top right corner.

#### Inicio del sistema

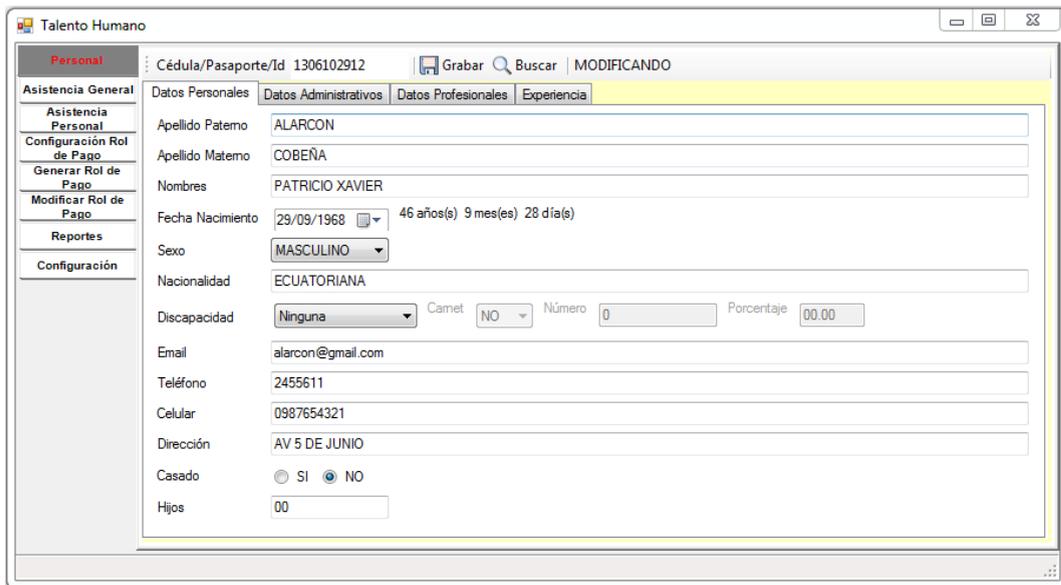
Antes de gustar a las opciones que ofrece el sistema se podrá apreciar una ventana de inicio, luego podremos dar clic en **Personal**.



## MENÚ PERSONAL

### Pestaña Datos Personales

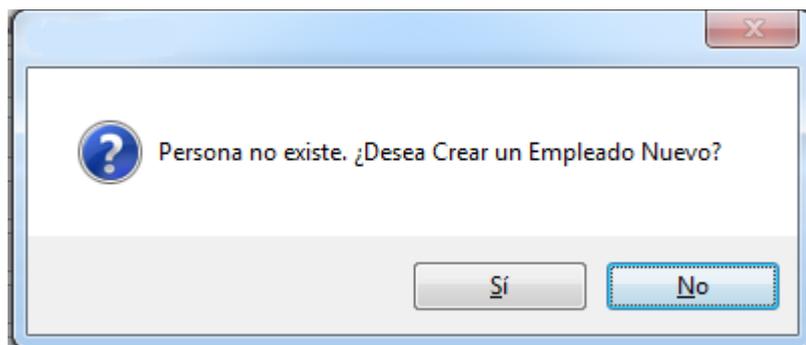
Comenzamos por el primer menú llamado Personal y su pestaña Datos Personales que es donde se ingresaran los datos del personal administrativo, docente y auxiliar de servicio.



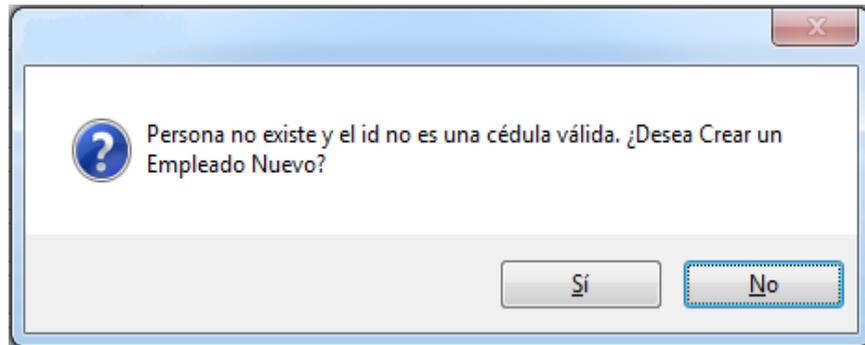
The screenshot shows a web application window titled 'Talento Humano'. The main content area is the 'Datos Personales' tab, which is highlighted in yellow. The user's ID is '1306102912'. The form contains the following fields:

Apellido Paterno	ALARCON
Apellido Materno	COBEÑA
Nombres	PATRICIO XAVIER
Fecha Nacimiento	29/09/1968 (46 años(s) 9 mes(es) 28 día(s))
Sexo	MASCULINO
Nacionalidad	ECUATORIANA
Discapacidad	Ninguna
Camet	NO
Número	0
Porcentaje	00.00
Email	alarcon@gmail.com
Teléfono	2455611
Celular	0987654321
Dirección	AV 5 DE JUNIO
Casado	<input type="radio"/> SI <input checked="" type="radio"/> NO
Hijos	00

Primero se ingresará su Cédula, Pasaporte o ID en caso de que la persona no esté registrada y su cedula sea un numero valido se mostrará una ventana con el siguiente mensaje:

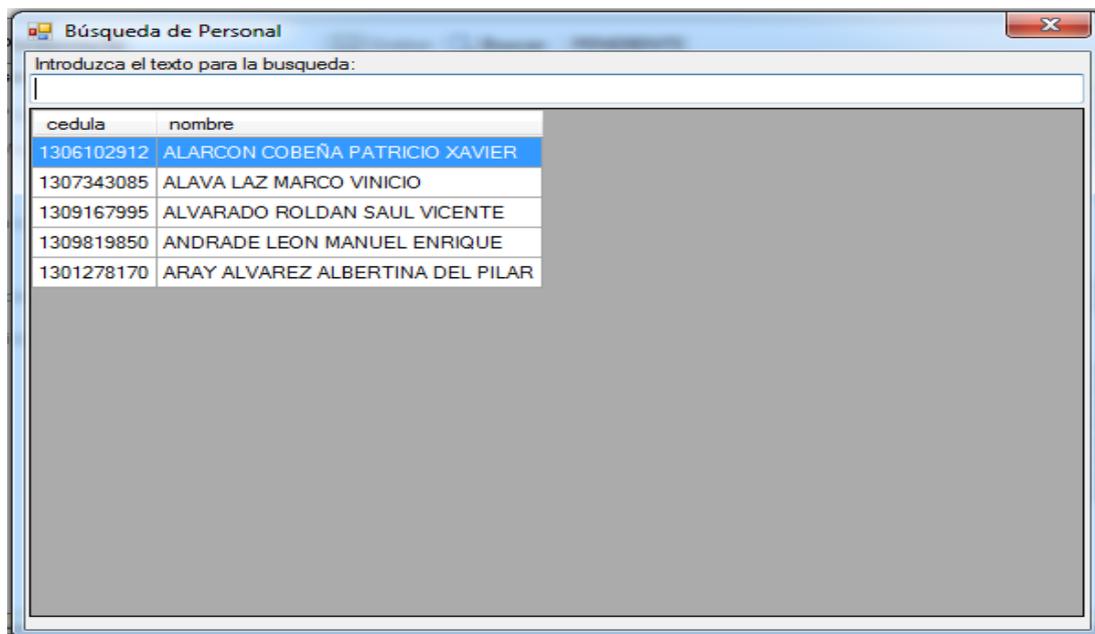


En el caso de que la persona no tenga cédula pero contenga un número de pasaporte se mostrará este mensaje:



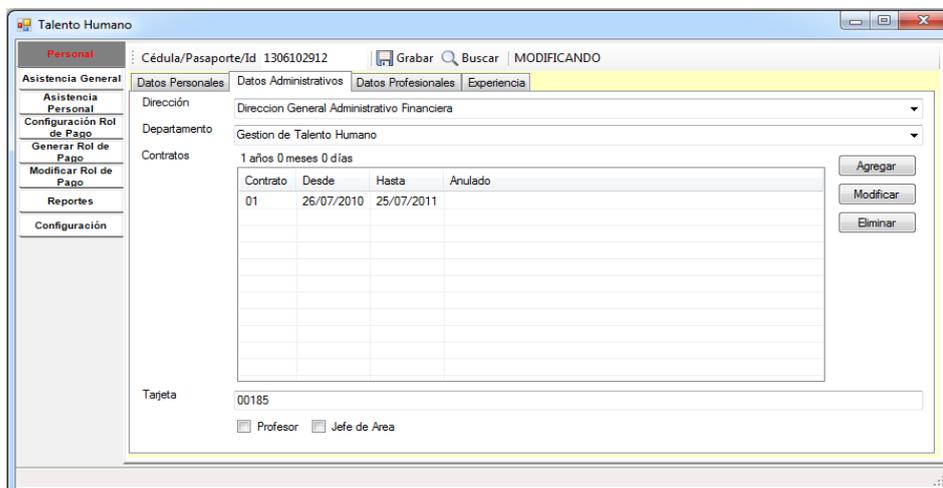
Procedemos a dar clic en si en caso de que se quiera ingresar el nuevo empleado. Una vez que estén todos los campos llenos del nuevo empleado se dará clic en **Guardar**.

Para realizar búsquedas del personal se dará clic en el botón **Buscar**, donde se abrirá la ventana de **Búsqueda de Personal** en ella se podrán realizar búsquedas por medio del número de cédula o nombre del personal. Una vez encontrada la persona doble clic para poder ver sus datos o poder modificarlos en caso de que sea necesario.



### **Pestaña Datos Administrativos**

En esta pestaña se agregará la **Dirección** y **Departamento** en donde se encuentren los empleados. También contará con el campo **Contratos** que es donde se agregarán, modificarán y eliminarán los contratos del personal. También se ingresará el número de tarjeta de cada uno de ellos. Encontramos dos checkbox que son Profesor y Jefe de Área que serán seleccionados de acuerdo al cargo que tenga el personal.



Para añadir un nuevo contrato daremos clic en el botón Agregar se abrirá la siguiente ventana:

Aquí se ingresará el código del empleado, su fecha de inicio y fin del contrato, su cargo, remuneración, Fecha Anulación (en caso de que su contrato sea anulado) y por último el tipo de contrato que este tendrá. Una vez llenado los campos daremos clic en **Aceptar**.

Una vez agregado todos los campos de la pestaña **Datos Administrativos** clic en **Guardar**. Si se quiere modificar el contrato se seleccionará el contrato y se

dará clic en **Modificar** para proceder con su modificación en caso de errores. El proceso para la eliminación de contrato se selecciona y clic en **Eliminar**.

Contrato	Desde	Hasta	Anulado
01	26/07/2010	25/07/2011	

## Pestaña Datos Profesionales

En esta pestaña se encuentran los Datos Profesionales del personal, aquí se podrá ingresar los títulos, cursos y reconocimientos que el/la empleado haya obtenido en el transcurso de su vida profesional.

Año	Titulo	Institucion	Nivel	En Area

Año	Nombre

Fecha	Reconocimiento

Para agregar un **Título**, daremos clic en **Agregar**.

Año	Título	Institucion	Nivel	En Area
1990	ING EN MARKETING	UNIVERSIDAD SAN GREGORIO DE PORTOVIEJO	3	NO

Títulos

Agregar

Modificar

Eliminar

Donde se nos abrirá la siguiente Ventana:

Títulos

Año: 2015

Título: ING EN SISTEMAS

Institución: UNIVERSIDAD TECNICA DE MANABI

Registro:

EnArea

Aceptar Cancelar

Aquí se agregarán los datos respecto al **Año** en que se obtuvo el título, el nombre del **Título**, la **Institución** donde lo obtuvo y por último el **Registro** de la Senescyt, una vez agregados clic en **Aceptar**. En caso de que se quiera modificar se selecciona el **Título** ya ingresado y se dará clic en **Modificar**.

Para Agregar los **Cursos** daremos clic en **Agregar**.

Año	Nombre
2000	COMPUTACION BASICA 1

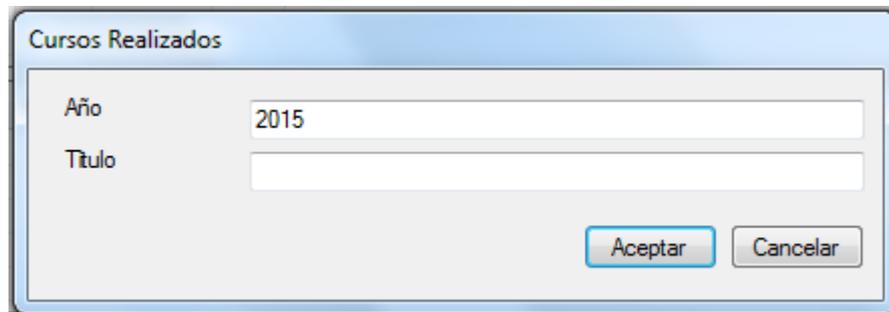
Cursos

Agregar

Modificar

Eliminar

Aquí se ingresarán los Cursos obtenidos por el empleado. El año en que lo realizó y el título del Curso, una vez ingresada la información clic en Aceptar. En caso de que se quiera modificar se selecciona el Curso ya agregado y se dará clic en **Modificar**.



Formulario de "Cursos Realizados" con los siguientes campos:

- Año: 2015
- Título: (campo vacío)
- Botones: Aceptar, Cancelar

Para añadir los **Reconocimientos** clic en **Agregar**.

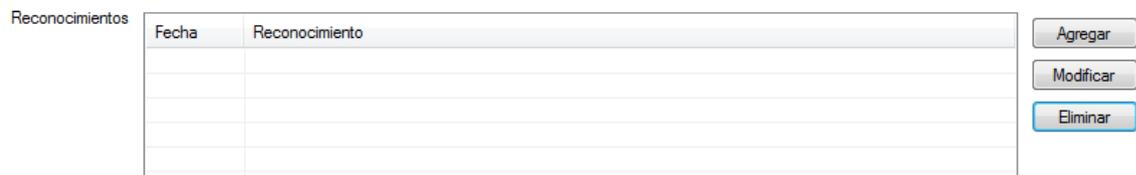


Tabla de Reconocimientos:

Fecha	Reconocimiento

Botones: Agregar, Modificar, Eliminar

Aquí se añadirán los reconocimientos obtenidos por el empleado. El año en que los obtuvo y el nombre del reconocimiento, una vez ingresada la información se dará clic en **Aceptar**. En caso de que se quiera modificar se seleccionará el **Reconocimiento** ya agregado y se dará clic en **Modificar**.

Reconocimientos Recibidos

Fecha: 08/07/2015

Nombre:

Aceptar Cancelar

## Pestaña Experiencia

En esta pestaña se pueden apreciar 2 dos campos, **Experiencia Profesional** que es donde se ingresará la experiencia del empleado desde que comenzó con su vida profesional, libros y artículos en donde se agregarán los libros o artículos publicados por el empleado en caso de que haya publicado uno.

Talento Humano

Cédula/Pasaporte/Id 1306102912 Grabar Buscar MODIFICANDO

Personal

Asistencia General

Asistencia Personal

Configuración Rol de Pago

Generar Rol de Pago

Modificar Rol de Pago

Reportes

Configuración

Datos Personales | Datos Administrativos | Datos Profesionales | Experiencia

Experiencia Profesional

Desde	Hasta	Cargo	Institucion

Libros y Artículos

Año	Título

Agregar Modificar Eliminar

Para añadir datos en el campo Experiencia Personal se dará clic en **Agregar**.

Experiencia  
Profesional

Desde	Hasta	Cargo	Institucion
06/07/1992	05/08/1993	DOCENTE	COLEGIO SANTO TOMAS

Agregar

Modificar

Eliminar

Aquí se ingresarán los datos de acuerdo a la fecha que estuvo en sus anteriores trabajos, el lugar donde trabajo y el cargo que tenía. Una vez ingresada la información se dará clic en el botón **Aceptar**. Si se requiere modificar se seleccionará la Experiencia Profesional agregada y se dará clic en el botón **Modificar**.

Experiencia

Desde: 03/05/1988

Hasta: 02/05/1989

Institución: UNIDAD EDUCATIVA ARCO IRIS

Cargo: DOCENTE

Aceptar Cancelar

Para añadir datos en el campo **Libro** se dará clic en el botón **Agregar**.

Libros

Año	Título
2015	MARKETING 1

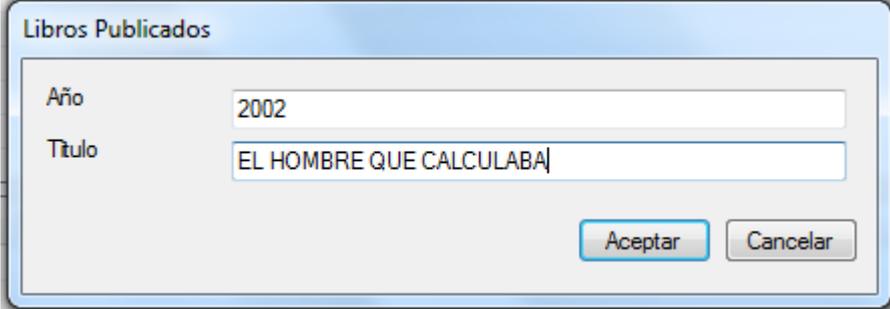
Agregar

Modificar

Eliminar

Aquí se escribirá el año en el que se publicó el libro y el título del mismo, una vez que estén llenos los campos clic en **Aceptar**. En caso de querer modificar la

información se seleccionará la información ingresada y se dará clic en el botón **Modificar**.



Libros Publicados

Año: 2002

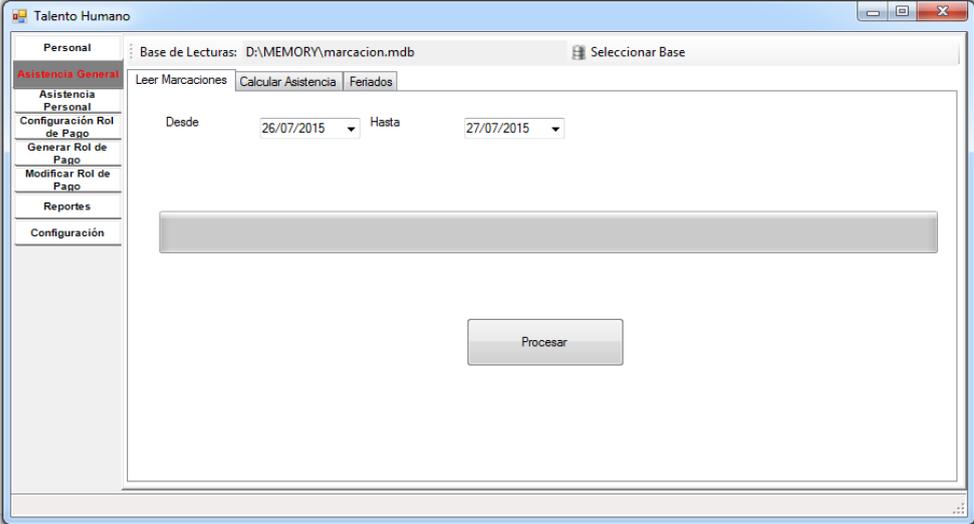
Titulo: EL HOMBRE QUE CALCULABA

Aceptar Cancelar

## MENÚ ASISTENCIA GENERAL

### Pestaña Leer Marcaciones

Continuamos con el segundo menú llamado Asistencia General que consta de tres pestañas la primera de ellas es Leer Marcaciones.



Talento Humano

Personal Base de Lecturas: D:\MEMORY\marcacion.mdb Seleccionar Base

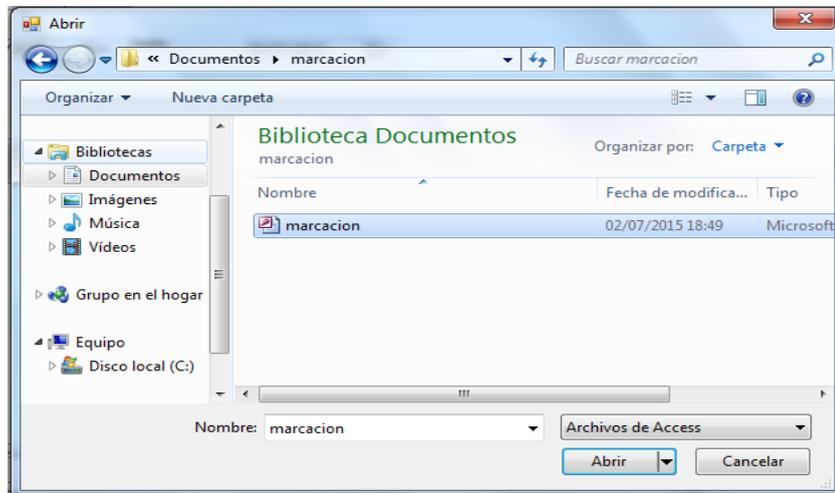
Asistencia General

Leer Marcaciones Calcular Asistencia Feriados

Desde: 26/07/2015 Hasta: 27/07/2015

Procesar

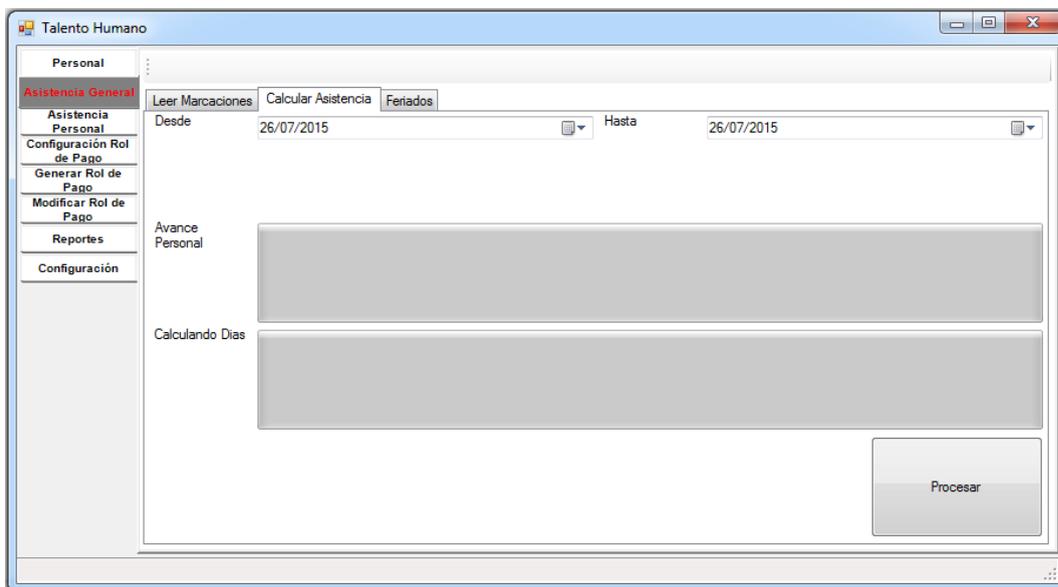
Para tener acceso, primero debe darse clic en **Seleccione la Base**, se abrirá una ventana en donde se permitirá seleccionar la base dependiendo de dónde esté guardada, en este caso se encuentra en **Biblioteca- Documentos**, cabe recalcar que esta base está diseñada en Microsoft Access y es extraída directamente del dispositivo Biométrico.



Luego, se escogen las fechas desde y hasta, en las que se quiera realizar la lectura, una vez escogidas las fechas clic en **Procesar**. Estos resultados se verán reflejados en el menú **Asistencia Personal** en la pestaña **Asistencia**.

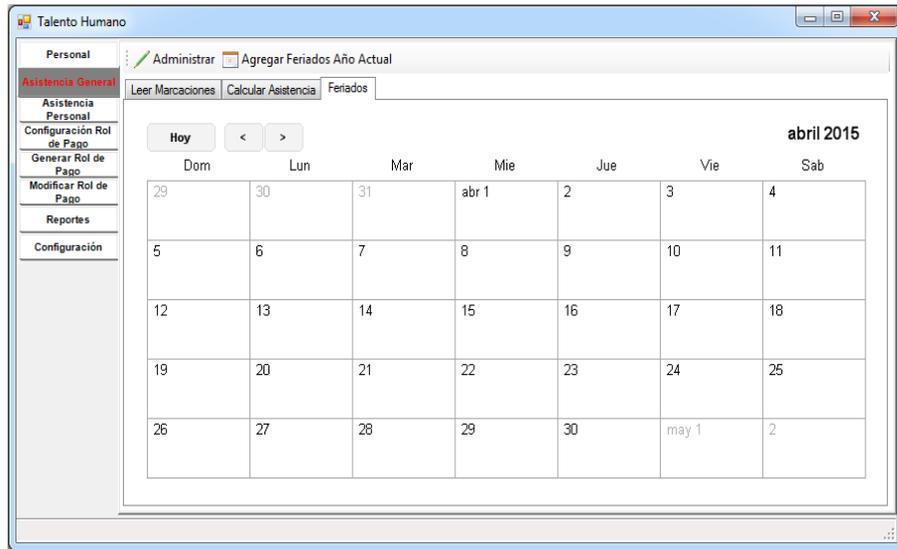
### **Pestaña Calcular Asistencia**

Aquí se calculará la asistencia del personal, Para calcular la asistencia es necesario que se procesen las lecturas de las marcaciones, En la pestaña **Calcular Asistencia**, seleccionamos las fechas desde y hasta cuando se quiere ejecutar el cálculo de la asistencia del personal. Luego de escogido el término de tiempo se solicita procesar que es una opción que contiene este diseño dentro de la pestaña Calcular Asistencia.

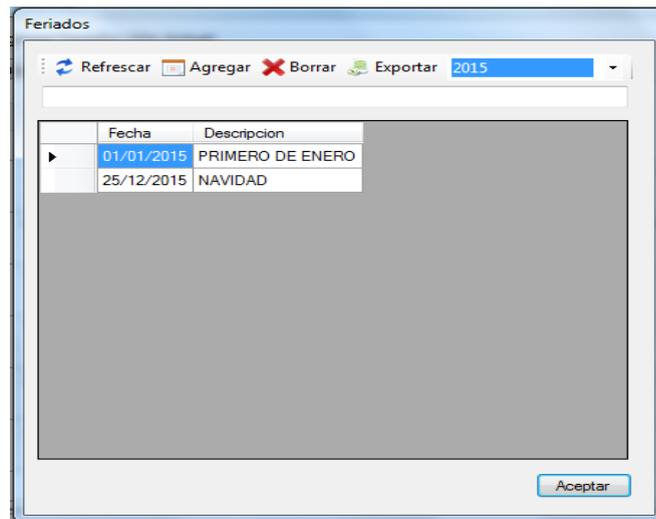


## Pestaña Feriados

En la pestaña Feriados se añadirá los días de feriados dependiendo del cronograma de la universidad y las fechas festivas que contenga el presente año. Esta ventana se despliega haciendo posible que un calendario mensual nos proporcione la información, pero, también permite ingresar datos que se reflejen cuando se consulta en él sobre los feriados estipulados por la ley y por las autoridades de la universidad. Esta opción mejora el sistema que se programará para que no se marquen los feriados como inasistencias.

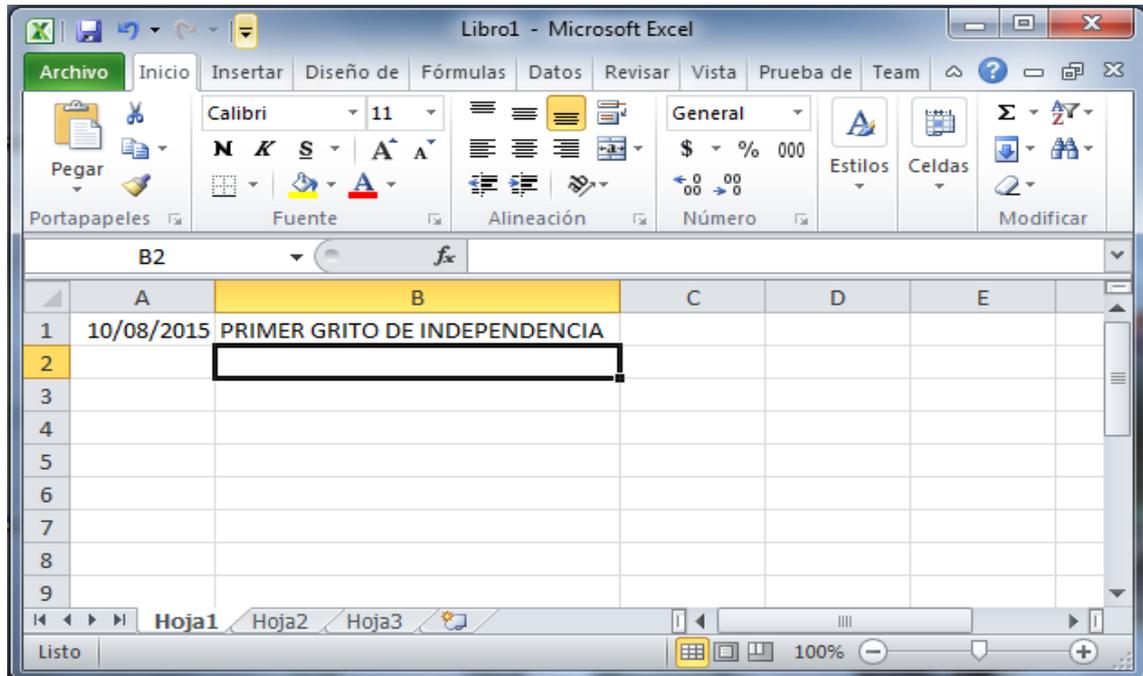


Para poder ingresar los Feriados de manera manual se hará clic en el botón Administrar, en donde se abrirá la siguiente ventana, aquí se seleccionará el año y se ingresará en la primera columna la fecha del feriado y en la segunda columna la descripción del mismo, en caso de tener errores se puede borrar seleccionándolo y dándole clic en borrar, también se puede exportar a una hoja de cálculo en Excel.

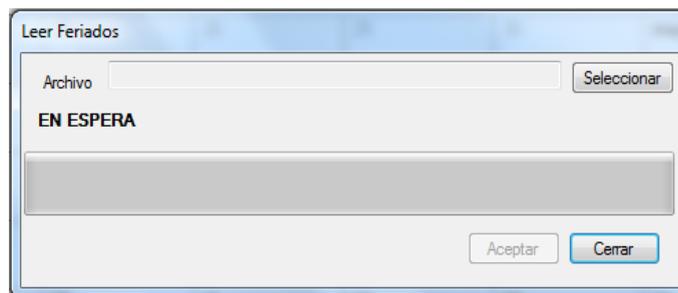


Si se desea importar la fecha de los feriados, primero se deberá abrir Microsoft Excel, una vez abierto, daremos clic en la primera columna en donde se pondrá la

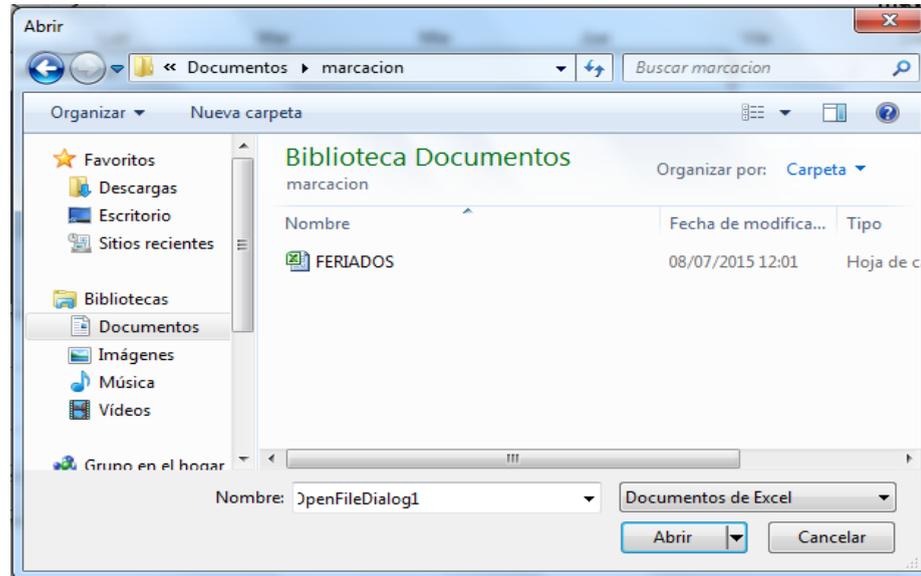
fecha del feriado, luego, daremos clic en la segunda columna en donde pondremos la descripción del feriado, una vez realizado esto la guardamos.



Una vez guardado el archivo con todos los feriados del correspondiente año, volvemos a la pestaña Feriado y daremos clic en **Añadir Feriados Año Actual** y se nos abrirá la siguiente ventana.



En donde se le dará clic en Seleccionar y se abrirá la ventana en donde podremos seleccionar el documento ya antes realizado.



Lo Seleccionamos y le damos clic en abrir, hecho esto, procederemos a dar clic en **Aceptar**, y listo el feriado del año actual estará ingresado.

## MENÚ ASISTENCIA PERSONAL

### Pestaña Horarios

Este menú está compuesto por 4 pestañas que son: horarios, asistencias, justificaciones y vacaciones.

La pestaña horarios es donde se ingresará el horario del empleado dependiendo de las horas que deba cumplir.

Para agregar el horario se dará clic en el botón Más, luego, procederemos a ingresar las respectivas horas de trabajo, el mismo se aplicará con los siguientes días, ya ingresados los horarios se dará clic en el botón Grabar.

Lunes	00:00	00:00	0 horas 0 minutos	+ -
-------	-------	-------	-------------------	-----

Aquí también se podrá buscar el horario de otra persona dando clic en el botón buscar como ya se explicó antes.

FmAAAAA  
 Cédula 1306102912 ALARCON COBEÑA PATRICIO XAVIER  
 Desde 09/07/2015 Hasta 09/07/2015 Refrescar

Horarios Asistencia Justificaciones Vacaciones

41 horas 0 minutos

Domingo			0 Horas 0 minutos	+ -
Lunes	07:00	12:00	8 horas 0 minutos	+ -
	16:00	19:00		
Martes	08:00	12:00	8 horas 0 minutos	+ -
	15:00	19:00		
Miércoles	07:00	11:00	7 horas 0 minutos	+ -
	15:00	18:00		
Jueves	07:00	13:00	10 horas 0 minutos	+ -
	15:00	19:00		
Viernes	08:00	13:00	8 horas 0 minutos	+ -
	16:00	19:00		
Sábado			0 Horas 0 minutos	+ -

## Pestaña Asistencia

En esta pestaña se podrá observar la asistencia del personal diaria con su tiempo laborado, tiempo falta y tiempo justificado en caso de que tenga la respectiva justificación.

FrmAAAAA

Personal : Cédula 1306102912 ALARCON COBEÑA PATRICIO XAVIER Buscar Desde 27/07/2010 Hasta 26/08/2010 Refrescar

Asistencia General Horarios Asistencia Justificaciones Vacaciones

Mes /Dia	Tiempo Laborable	Tiempo Laborado	Tiempo Falta	Tiempo Justificado
07/27	7 horas 0 minutos	2 horas 20 minutos	4 horas 40 minutos	0 horas 0 minutos
07/28	10 horas 0 minutos	0 horas 52 minutos	9 horas 8 minutos	0 horas 0 minutos
07/29	8 horas 0 minutos	4 horas 38 minutos	3 horas 22 minutos	0 horas 22 minutos
07/30	0 horas 0 minutos			
07/31	0 horas 0 minutos			
08/01	8 horas 0 minutos	0 horas 0 minutos	8 horas 0 minutos	0 horas 0 minutos
08/02	8 horas 0 minutos	0 horas 0 minutos	8 horas 0 minutos	0 horas 0 minutos
08/03	7 horas 0 minutos	3 horas 0 minutos	4 horas 0 minutos	0 horas 0 minutos
08/04	10 horas 0 minutos	1 horas 22 minutos	8 horas 38 minutos	0 horas 0 minutos
08/05	8 horas 0 minutos	1 horas 12 minutos	6 horas 48 minutos	0 horas 0 minutos

Entrada Reloj	Salida Reloj	Marca Entrada	Marca Salida	Tiempo Laborable	Tiempo Laborado	Tiempo Falta	Tiempo Justificado	Tiempo Extra	Aprobar Horas Extra
07:00	11:00	00:00	00:00	4:00	0:00	4:00	0:00	0:00	<input type="checkbox"/>
15:00	18:00	15:40	19:25	3:00	2:20	0:40	0:00	0:00	<input type="checkbox"/>

Recalcular

En esta sección se selecciona la fecha en la que se quiera visualizar el Tiempo laborado, cabe recalcar que la fecha escogida tiene que estar en el rango de la fecha seleccionada en el menú **Asistencia General – Pestaña Leer Marcaciones**.

Desde 09/07/2015 Hasta 09/07/2015 Refrescar

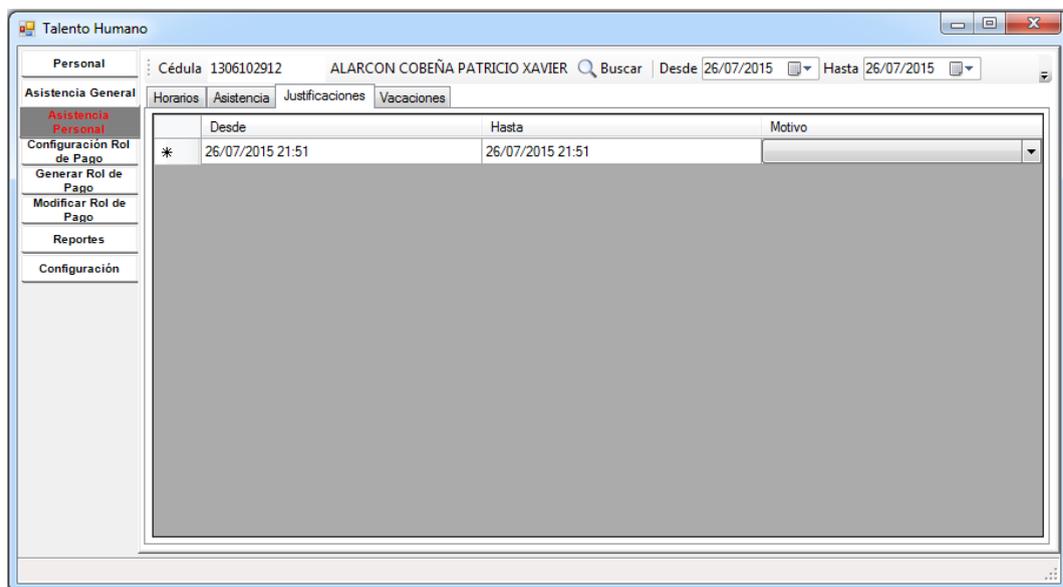
En la parte inferior se refleja la entrada y salida del reloj que tenga el empleado, además se ven las marcaciones de entrada y salida del personal, el tiempo laborado, el tiempo falta, tiempo justificado y el tiempo extra. También contiene Aprobar Horas Extra en caso de que el personal labora horas demás.

Entrada Reloj	Salida Reloj	Marca Entrada	Marca Salida	Tiempo Laborable	Tiempo Laborado	Tiempo Falta	Tiempo Justificado	Tiempo Extra	Aprobar Horas Extra
07:00	11:00	00:00	00:00	4:00	0:00	4:00	0:00	0:00	<input type="checkbox"/>
15:00	18:00	15:40	19:25	3:00	2:20	0:40	0:00	0:00	<input type="checkbox"/>

Recalcular

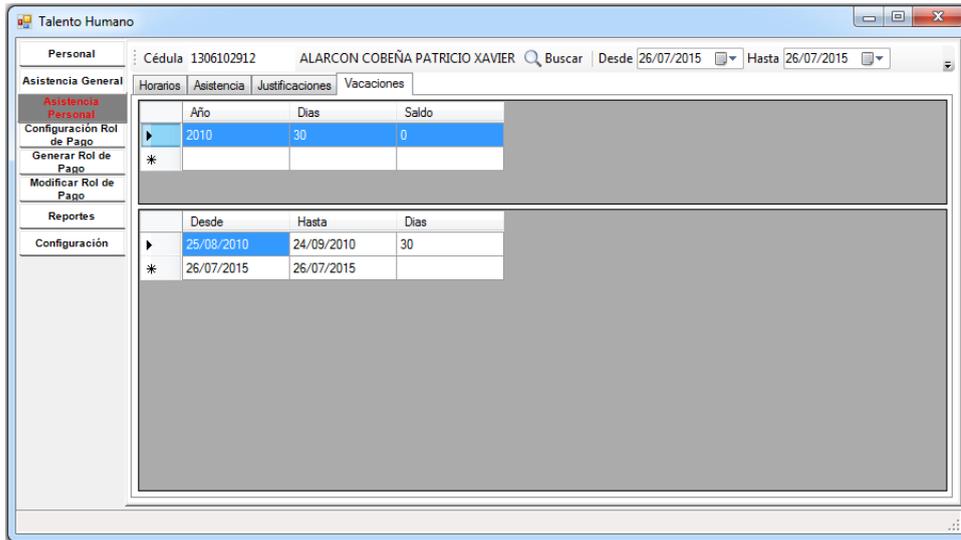
## Pestaña Justificaciones

En esta pestaña se podrán añadir las debidas justificaciones al personal, una vez agregadas las justificaciones se tiene que volver a calcular para que la justificación sea reflejada en la pestaña Asistencia en la parte inferior.



## Pestaña Vacaciones

En esta pestaña se agregarán las vacaciones del personal.



En la parte superior se escribirá el año, días que tiene por gozar y el saldo se reflejará el valor de los días que le quedarán por gozar

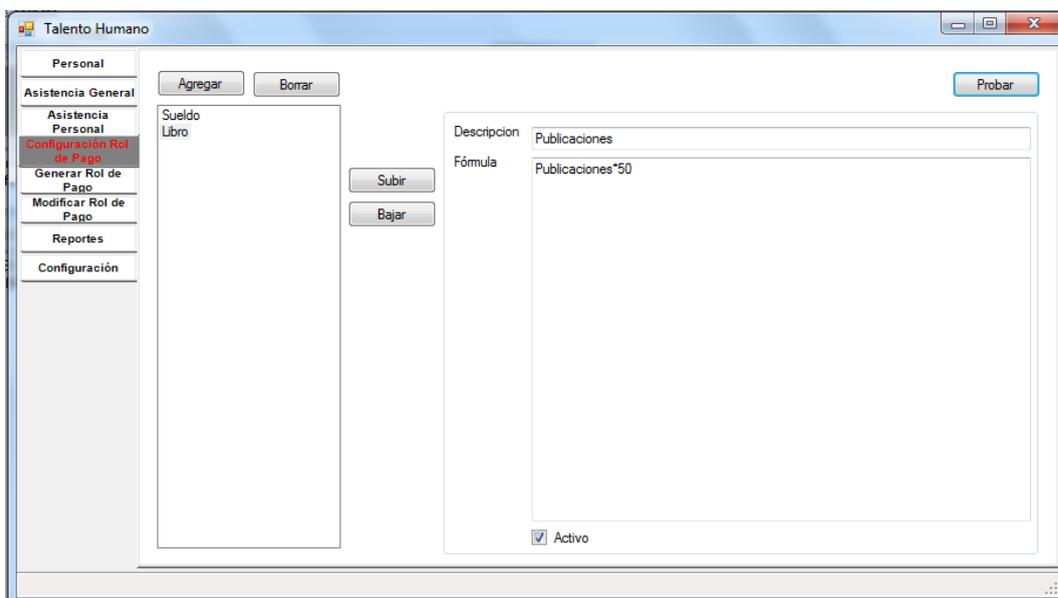
Año	Días	Saldo
▶ 2010	30	0
*		

En la parte inferior se escribirá la fecha desde que tendrá sus vacaciones y hasta cuándo podrá gozar de las mismas. Y en la parte donde dice Días se podrá ver cuantos fueron los días que tuvo.

Desde	Hasta	Días
▶ 25/08/2010	24/09/2010	30
* 07/07/2015	07/07/2015	

## MENÚ CONFIGURACIÓN ROL DE PAGO

En este menú es donde se agregarán y configurara el rol, para añadir rubros se dará clic en Agregar, una vez que se ha agregado damos clic para poder cambiarle el nombre dependiendo del rubro que se vaya agregar, en este caso le pondremos Libro, el cual se refiere a las publicaciones de libro que tenga el personal, Luego, lo seleccionamos y nos dirigimos a Descripción, ahí se escribirá una descripción el rubro ya agregado, le pondremos Publicaciones. Posteriormente iremos a Fórmula y aquí es donde se agregará una fórmula de Libro que será  $\text{Publicaciones} * 50$  con esto se quiere decir que a cada persona que tenga una publicación de Libro se le agregará \$50 más a su remuneración.



Después se dará clic en **Probar** y se abrirá una ventana con un listado de rubros que el sistema ya tendrá ingresado, ahí es donde se podrá verificar que la fórmula puesta esté bien.

Probar Rol de Pago

Variable	Valor
SueldoBasico	0
MinutosLaborados	0
MinutosFalta	0
MinutosExtra	0
MinutosJustificados	0
JefeArea	0
NivelMaestria	0
Publicaciones	1
Profesor	0
Casado	0

Descripcion	Nombre	Valor
Sueldo Basico	Sueldo	0.00
Publicaciones	Libro	50.00

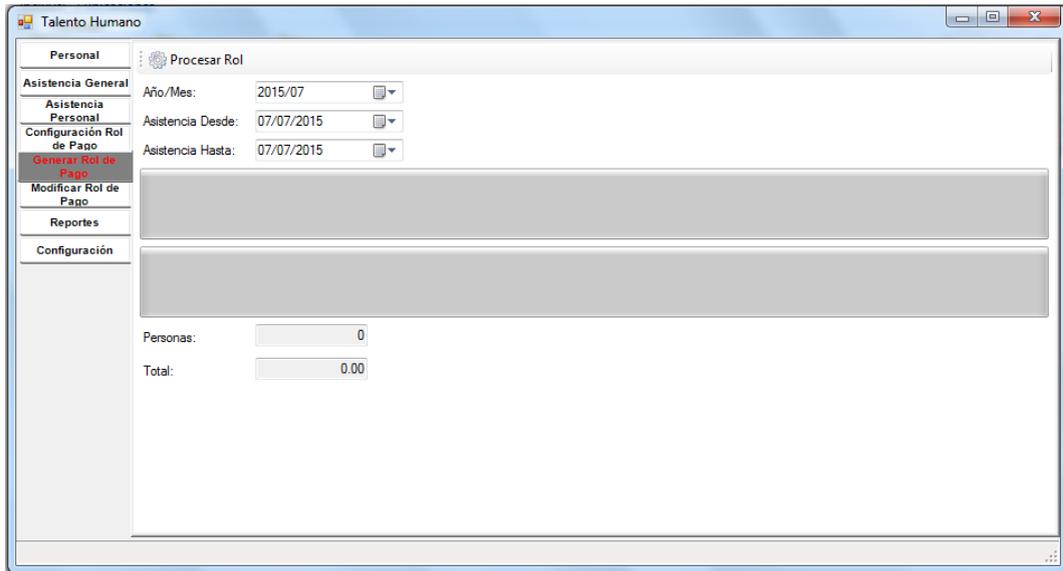
Calcular      Aceptar

Como se podrá ver se ha puesto en el campo Publicaciones el número 1 que significa que tiene 1 publicación lo que da un total de \$50 más a la remuneración, lo que nos indica que está bien la fórmula.

Una vez constatado que la fórmula está bien continuamos al siguiente menú llamado **Generar Rol de Pago**.

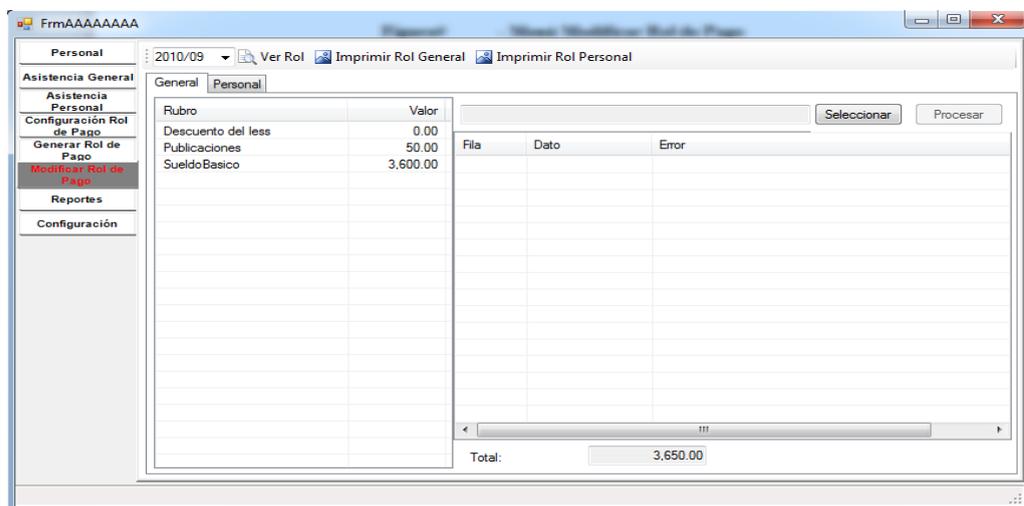
### MENÚ GENERAR ROL DE PAGO

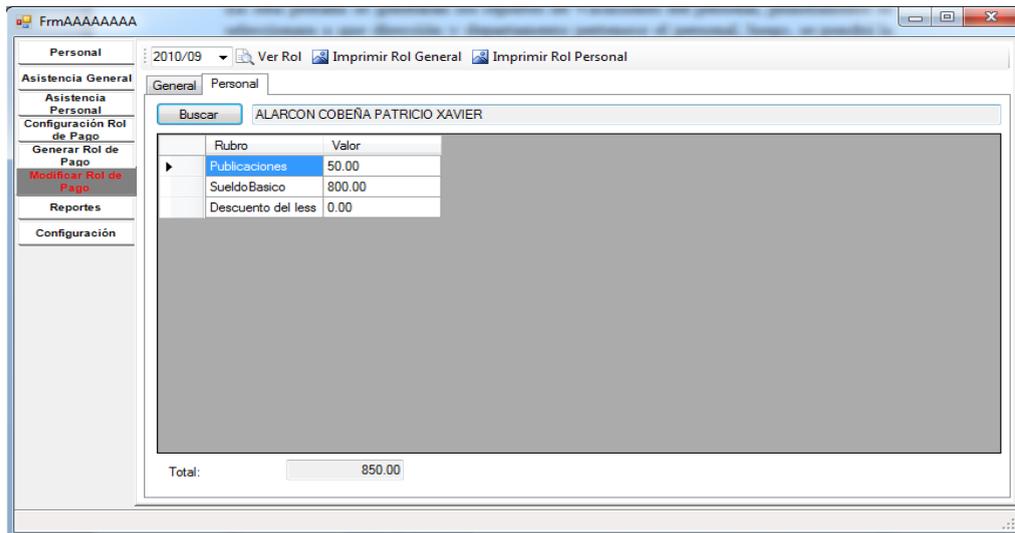
En esta pestaña es donde se generará el rol del pago de todo el personal en general. Para realizarlo se pondrá la fecha Desde y Hasta correspondiente al mes que permita poder procesar los roles de todo el personal de la institución. Una vez hecho esto se dará clic en Procesar Rol. Una vez que se procese la información se podrá observar a cuantas personas se realizó la acción y un valor total de todos.



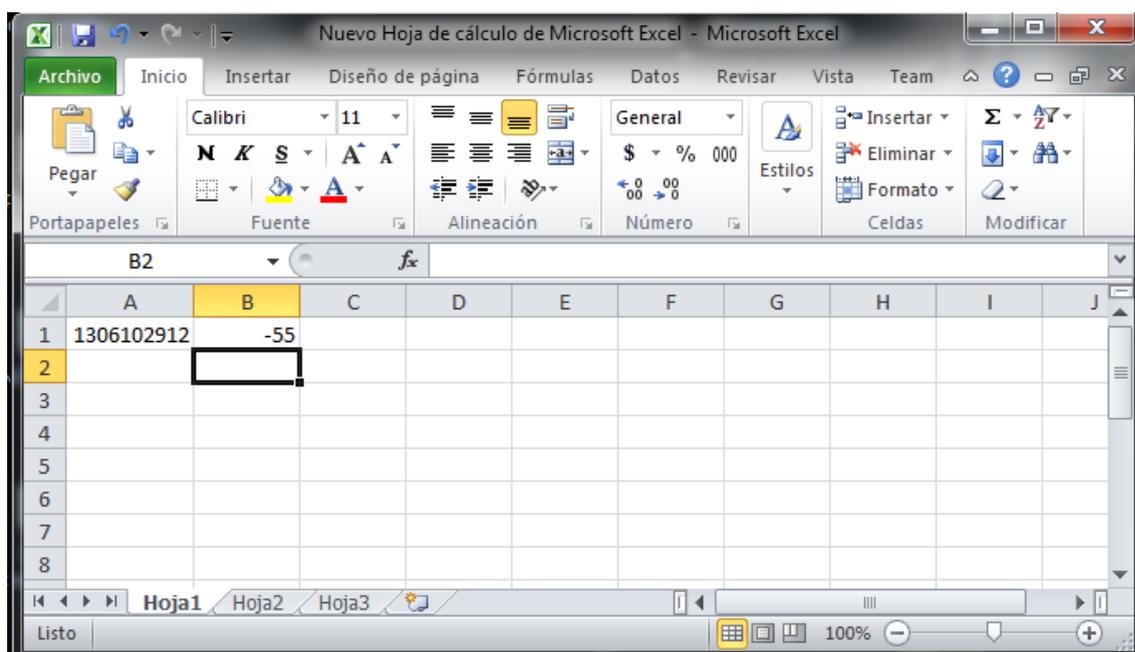
## MENÚ MODIFICAR ROL DE PAGO

En el menú Modificar Rol de Pago nos encontramos con: Rol de Pago General y Rol de Pago Personal, en donde se podrá observar todos los rubros y descuentos que fueron puestos anteriormente en el menú Configuración Rol de Pago.

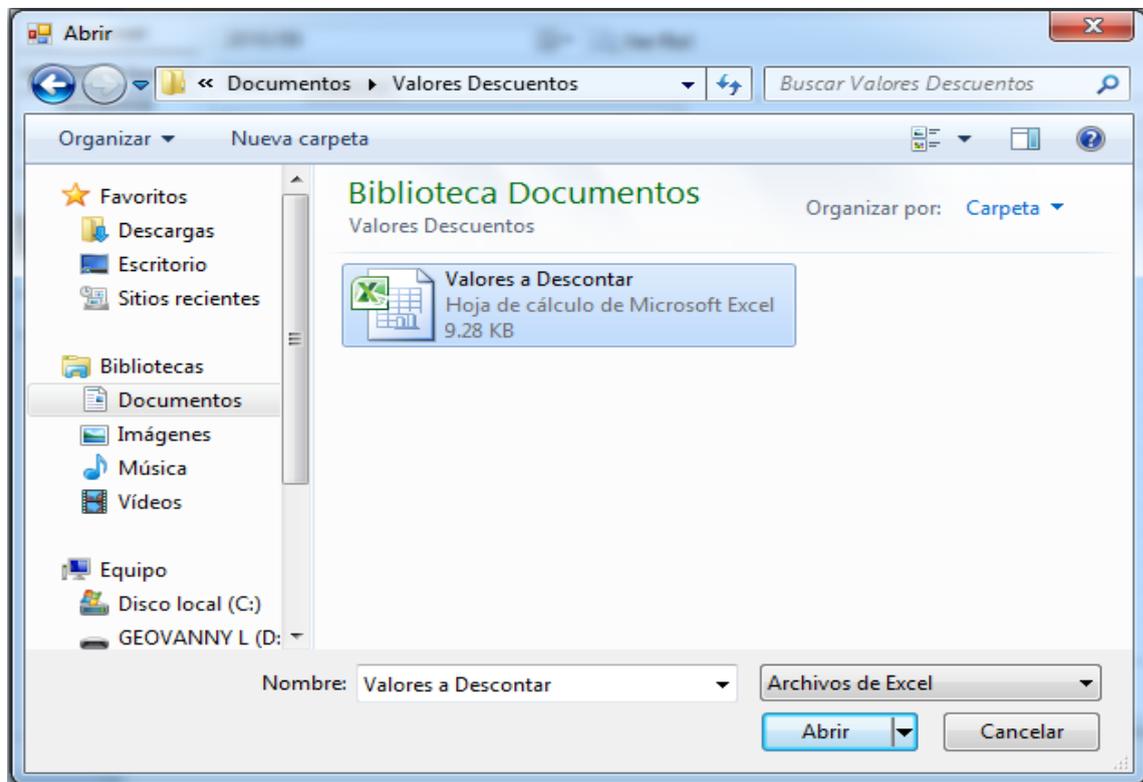




Habr  rubros de descuentos que estar  vac os puesto, como es el caso de los descuento de los pr stamos del Iess son rubros de valor desconocido y su valor de descuento var a entre las personas, por eso si se requiere realizar un descuento a este tipo rubros, se tendr  que crear en una hoja de Excel una columna con el n mero de celda de la persona que tenga este descuento y en otra columna el valor a descontar as  como se muestra en la siguiente imagen:



Una vez realizado esto procederemos a ir a la ventana **General** donde seleccionaremos el rubro **y** daremos clic en **Seleccionar**. Se nos abrirá una ventana que nos permitirá escoger la hoja de cálculo en Excel donde pusimos el valor a descontar.



En este caso se encuentra en **Documentos – Valores Descuentos**. La seleccionamos y se dará clic en **Abrir**.



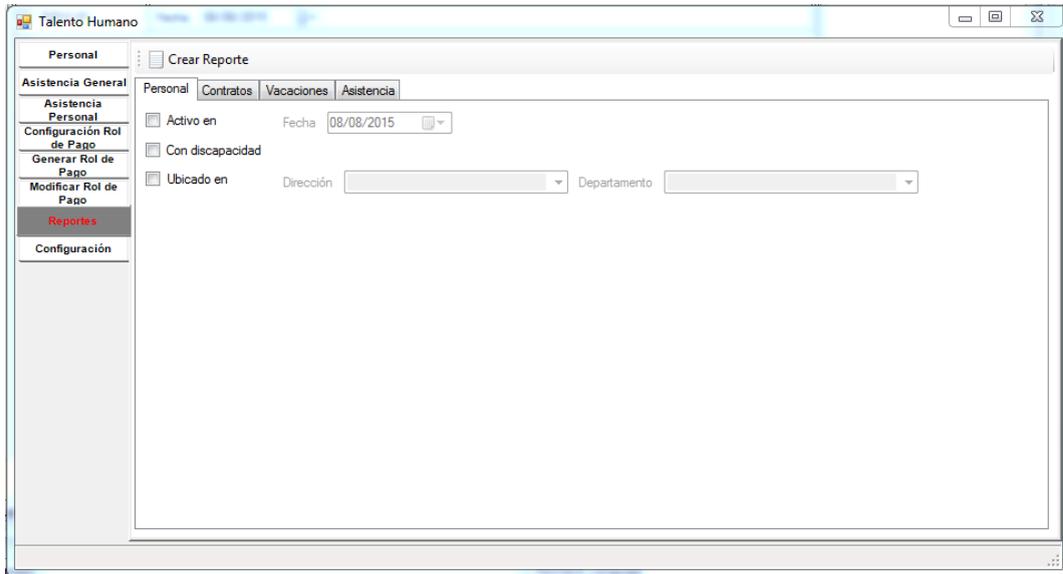
Ya se puede apreciar el valor de descuento que le dimos a ese rubro y a esa persona. También se puede apreciar dos botones que son Imprimir Rol General e Imprimir Rol Personal que se utilizarán para generar reportes de los roles de pago de los empleados dependiendo de la opción que elija.

## **MENÚ REPORTES**

Aquí en este menú se encontrará compuesto de 4 pestañas que son **Personal, Contratos, Vacaciones y Asistencia**

### **Pestaña Personal**

En esta pestaña es donde se realizará un reporte sobre el personal que labora en la universidad, para poder crear el reporte, primero se hará clic en el checkbox **Activo en** y posteriormente se pondrá la fecha desde cuándo se encuentra activo el personal. También se podrá seleccionar la opción Con Discapacidad para poder visualizar un listado de las personas que tengan discapacidad. Por último se podrá seleccionar la dirección y departamento de donde se quiera realizar el reporte del personal. Cuando ya se hayan seleccionado los campos se procederá a dar clic en el botón **Crear Reporte**.



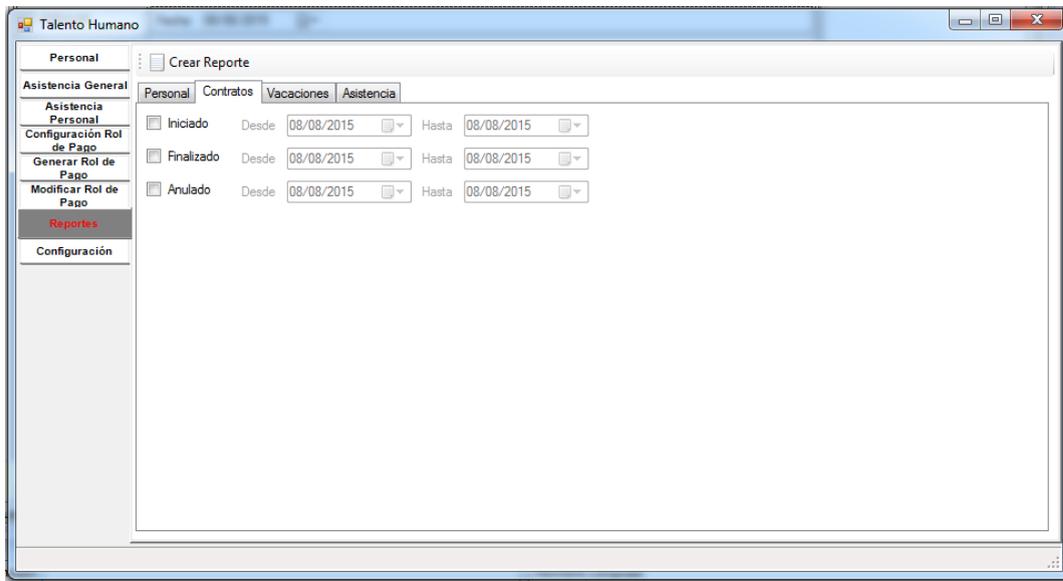
Luego se mostrará el reporte en una hoja de cálculo en Excel o en Openoffice Calc, dependiendo de cual se tenga instalado, donde se podrá apreciar el reporte del **Listado del Personal**

Cedula/ID	Apellido Paterno	Apellido Materno	Nombres	Nacionalidad	Fecha Nacimiento	Discapacidad	Carnet	Porcentaje Discapacidad	Email
1306102912	ALARCON	COBEÑA	PATRICIO XAVIER	ECUATORIANA	09/29/68	Ninguna	0	0	
1307343085	ALAVA	LAZ	MARCO VINICIO	ECUATORIANA	06/12/72	Ninguna	0	0	
1314078864	LINZAN	GONZALEZ	GEOVANNY ANDRES	ECUATORIANO	08/09/92	Ninguna		0	
<b>Total</b>	<b>Gestion de Talento Humano</b>		<b>3</b>						

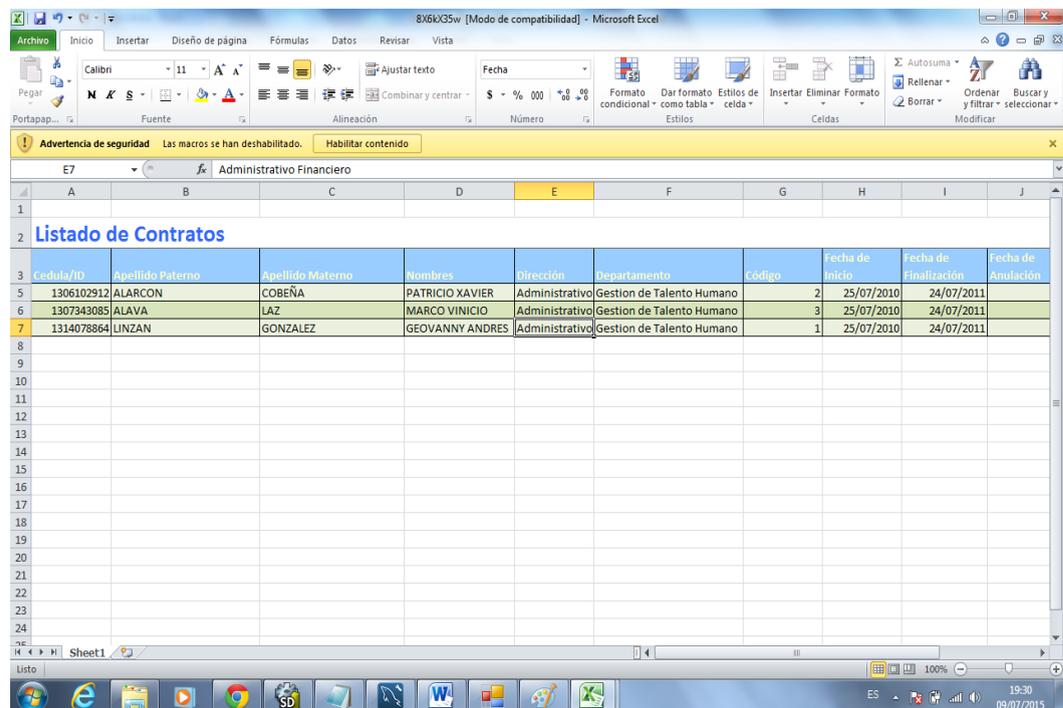
## Pestaña Contratos

En ella se podrán realizar reportes referentes a lo que compete a los contratos del personal para ello se tendrá que dar clic en **Iniciado** y luego en **Finalizado** en

estos 2 campos se definirá la fecha de inicio y final de los contratos del personal que se querrá observar en el reporte. En caso de que también se quiera ver los contratos anulado se seleccionará la opción Anulado para así poder ver el personal con los contratos anulados.



Una vez hecho esto se dará clic en el Botón **Crear Reporte**.



## Pestaña Vacaciones

En esta pestaña se generarán los reportes de Vacaciones del personal, primeramente se seleccionará a que dirección y departamento pertenece el personal, luego, se pondrá la salida y entrada del personal.

The screenshot shows a web application window titled 'Talento Humano'. On the left is a sidebar menu with the following items: 'Personal', 'Asistencia General', 'Asistencia Personal', 'Configuración Rol de Pago', 'Generar Rol de Pago', 'Modificar Rol de Pago', 'Reportes' (highlighted in red), and 'Configuración'. The main content area has a 'Crear Reporte' button at the top left. Below it are four tabs: 'Personal', 'Contratos', 'Vacaciones', and 'Asistencia'. The 'Vacaciones' tab is active. The form contains the following fields:

- 'Ubicado en' section with two dropdown menus: 'Dirección' and 'Departamento'.
- 'Salida' section with 'Desde' and 'Hasta' date pickers, both set to '08/08/2015'.
- 'Entrada' section with 'Desde' and 'Hasta' date pickers, both set to '08/08/2015'.

Cuando ya se hayan seleccionado los campos y puesto las fechas correctas se dará clic en el botón Crear Reporte.

Listado de Vacaciones						
ALARCON COBEÑA PATRICIO XAVIER						
Nombres	ALARCON COBEÑA PATRICIO XAVIER	cédula/id	1306102912			
Periodo	2010	Días	30			
Desde	Hasta	Días Tomados				
	01/10/2010	30				
Total Utilizadas			30			
LINZAN GONZALEZ GEOVANNY ANDRES						
Nombres	LINZAN GONZALEZ GEOVANNY ANDRES	cédula/id	1314078864			
Periodo	2010	Días	30			
Desde	Hasta	Días Tomados				
	01/01/2011	30				
Total Utilizadas			30			

## Pestaña Asistencia

Aquí se generarán reportes de asistencia dependiendo de las opciones que uno haya seleccionado, primero se deberá escoger las fechas desde que se encuentra activo, luego su ubicación, y los campos en los que se requiera observar la información en el reporte.

**Talento Humano**

Personal

Asistencia General

Asistencia Personal

Configuración Rol de Pago

Generar Rol de Pago

Modificar Rol de Pago

**Reportes**

Configuración

Crear Reporte

Personal | Contratos | Vacaciones | **Asistencia**

Activo en Desde: 08/08/2015 Hasta: 08/08/2015

Ubicado en Dirección: Departamento:

Tiempo Laborable  SI  NO

Tiempo Laborado  SI  NO

Tiempo Falta  SI  NO

Tiempo Justificado  SI  NO

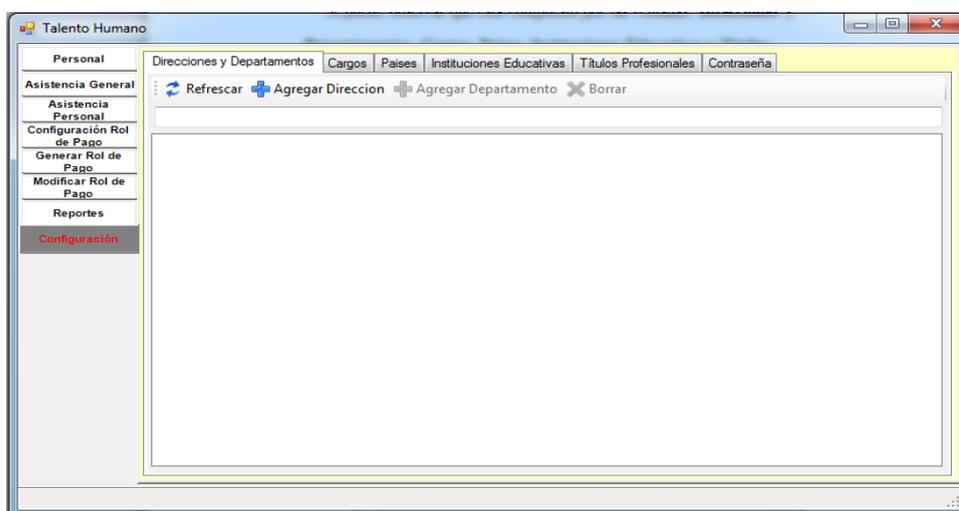
Tiempo Extra  SI  NO

## MENÚ CONFIGURACIÓN

En el menú Configuración se puede observar que está compuesto por las ventanas: **Direcciones y Departamentos**, **Cargos**, **Países**, **Instituciones Educativas** y **Títulos Profesionales**.

### Pestaña Direcciones y Departamentos

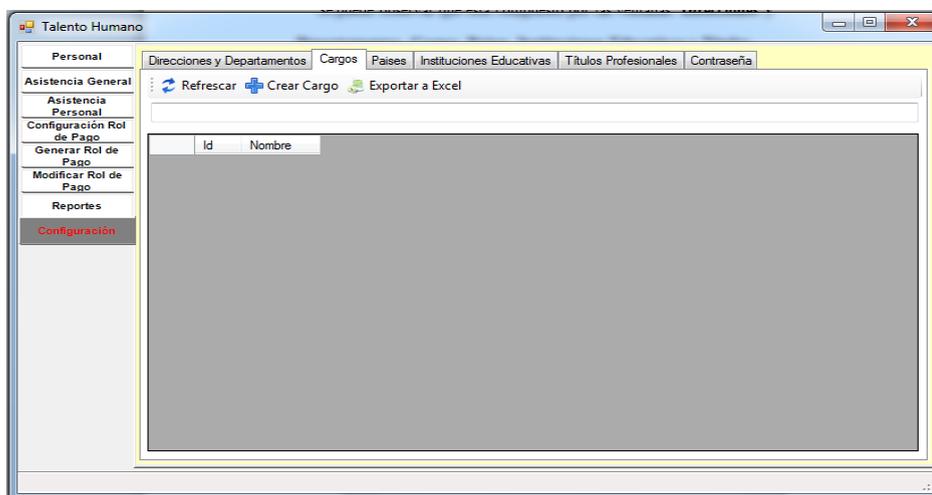
En la Pestaña **Direcciones y Departamentos** se podrá agregar las direcciones con sus respectivos departamentos, Si por error se escribe mal el nombre de la Dirección o Departamento se hará clic encima del nombre para poderle cambiar de nombre o en caso de que se cree departamentos o direcciones de más existe la opción de **Borrar**.



## Pestaña Cargos

En la pestaña denominada **Cargos** se encuentran agregados los cargos del personal, que los identifica de sus labores al personal, estos pueden ser exportados a una hoja de cálculo en Excel si hubiera necesidad de hacerlo.

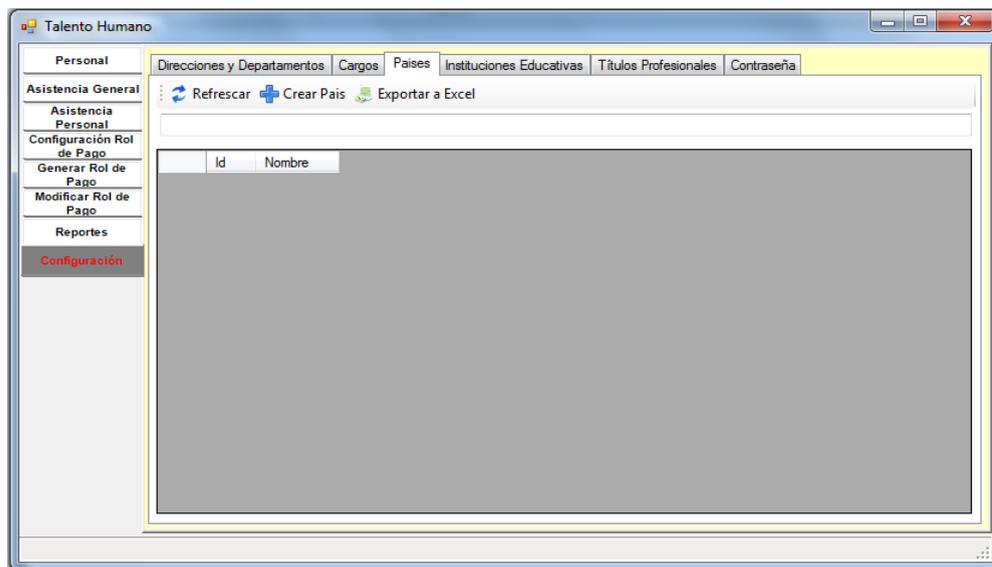
El sistema permite crear nuevos cargos en caso de ser necesario y permite luego refrescar la información nueva generada.



## Pestaña País

En la pestaña denominada **Países**, se encuentran agregados los países de origen de cada miembro que forma parte del personal, esta información puede ser exportada a una hoja de cálculo en Excel si hubiera necesidad de hacerlo.

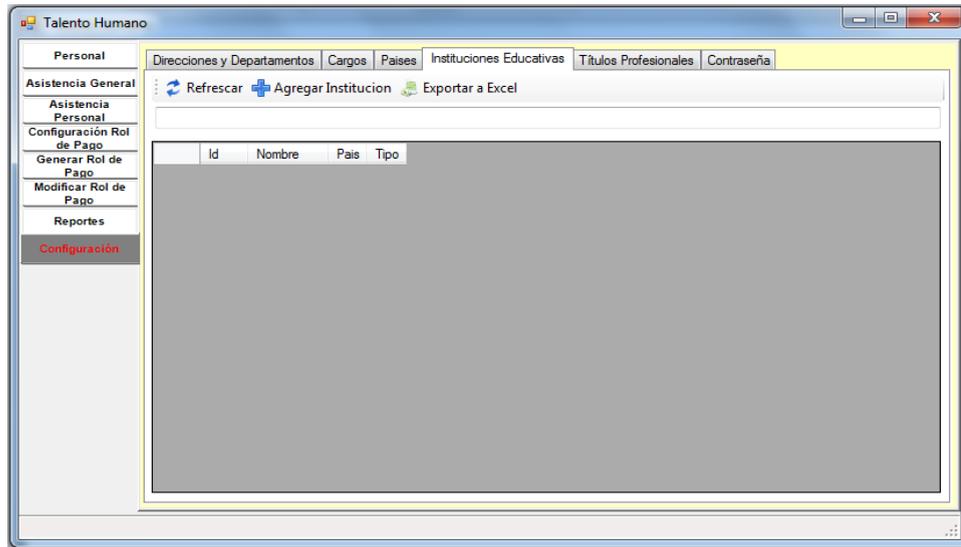
El sistema permite crear nuevos países en caso de ser necesario y permite luego refrescar la información nueva generada.



### **Pestaña Instituciones Educativas**

En esta pestaña se añadirán las **Instituciones Educativas** en donde el personal haya realizado sus estudios en todos sus niveles, se registra en cada uno de los casos o instituciones del país al que esta pertenece. Esta información puede ser exportada a una hoja de cálculo en Excel si hubiera necesidad de hacerlo.

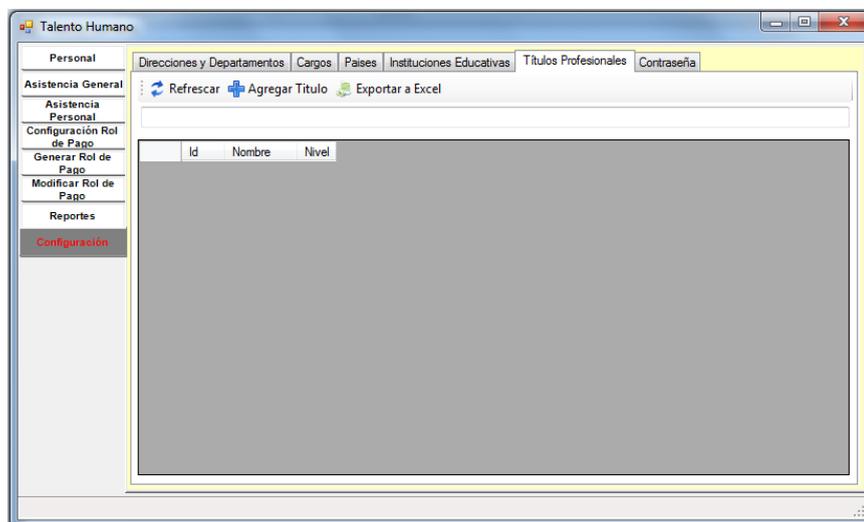
El sistema permite agregar instituciones en caso de ser necesario y permite luego refrescar la información nueva generada.



## Pestaña Títulos Profesionales

En la pestaña **Títulos Profesionales** están ingresados los títulos profesionales que tenga el personal. Exista la opción para poder ingresar el Nivel a que corresponde el título profesional. Esta información puede ser exportada a una hoja de cálculo en Excel si hubiera necesidad de hacerlo.

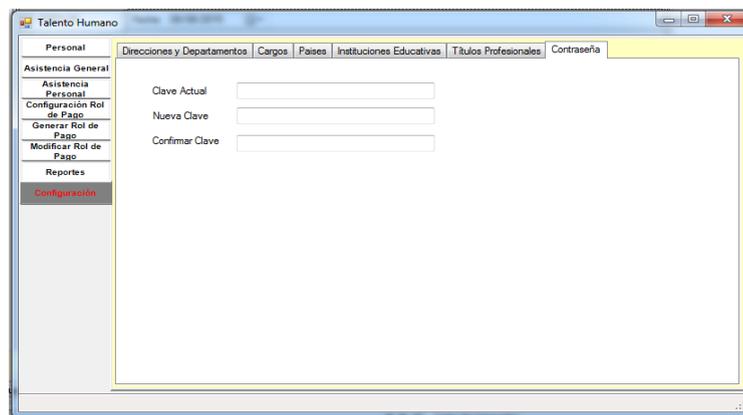
El sistema permite agregar títulos en caso de ser necesario y permite luego refrescar la información nueva generada.



## Pestaña Contraseña

En esta pestaña se podrá configurar lo que es la contraseña del usuario que haya ingresado al sistema, se puede cambiar la contraseña las veces que el usuario crea conveniente.

En Clave Actual se escribirá la contraseña que se esta utilizando, En Nueva Clave se escribirá la contraseña por la cual se quiere cambiar la anterior y en Confirmar Clave se escribirá de nuevo la nueva contraseña para asegurarse de que esta bien.



The screenshot shows a web application window titled "Talentos Humanos". The interface has a sidebar menu on the left with the following items: Personal, Asistencia General, Asistencia Personal, Configuración Rol de Pago, Generar Rol de Pago, Modificar Rol de Pago, Reportes, and Configuración. The main content area is titled "Contraseña" and contains three input fields: "Clave Actual", "Nueva Clave", and "Confirmar Clave".

# MANUAL TÉCNICO

## Base de Datos

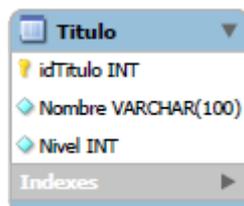
### Tablas y su Ejecución en My SQL

#### Sentencia SQL de la Entidad Dirección



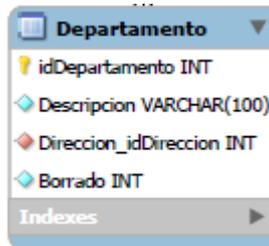
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Direccion` (  
  `idDireccion` INT NOT NULL AUTO_INCREMENT,  
  `Descripcion` VARCHAR(100) NULL,  
  `Borrado` BIT NOT NULL DEFAULT 0,  
  PRIMARY KEY (`idDireccion`))  
ENGINE = InnoDB
```

#### Sentencia SQL de la Entidad Titulo



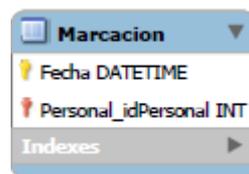
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Titulo` (  
  `idTitulo` INT NOT NULL AUTO_INCREMENT,  
  `Nombre` VARCHAR(100) NOT NULL,  
  `Nivel` INT NOT NULL,  
  PRIMARY KEY (`idTitulo`))  
ENGINE = InnoDB
```

## Sentencia SQL de la Entidad Departamento



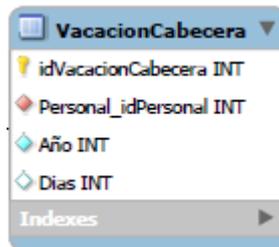
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Departamento` (  
  `idDepartamento` INT NOT NULL AUTO_INCREMENT,  
  `Descripcion` VARCHAR(100) NOT NULL,  
  `Direccion_idDireccion` INT NOT NULL,  
  `Borrado` INT NOT NULL DEFAULT 0,  
  PRIMARY KEY (`idDepartamento`),  
  INDEX `fk_Departamento_Direccion1_idx` (`Direccion_idDireccion` ASC),  
  CONSTRAINT `fk_Departamento_Direccion1`  
    FOREIGN KEY (`Direccion_idDireccion`)  
    REFERENCES `TalentoHumano`.`Direccion` (`idDireccion`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

## Sentencia SQL de la Entidad Marcacion



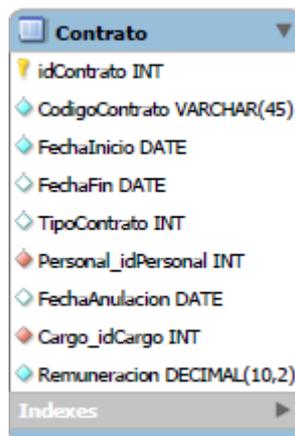
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Marcacion` (  
  `Fecha` DATETIME NOT NULL,  
  `Personal_idPersonal` INT NOT NULL,  
  PRIMARY KEY (`Fecha`, `Personal_idPersonal`),  
  INDEX `fk_Marcacion_Personal1_idx` (`Personal_idPersonal` ASC),  
  CONSTRAINT `fk_Marcacion_Personal1`  
    FOREIGN KEY (`Personal_idPersonal`)  
    REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

## Sentencia SQL de la Entidad VacacionCabecera



```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`VacacionCabecera` (  
  `idVacacionCabecera` INT NOT NULL AUTO_INCREMENT,  
  `Personal_idPersonal` INT NOT NULL,  
  `Año` INT NOT NULL,  
  `Dias` INT NULL,  
  PRIMARY KEY (`idVacacionCabecera`),  
  INDEX `fk_VacacionCabecera_Personal1_idx` (`Personal_idPersonal` ASC),  
  CONSTRAINT `fk_VacacionCabecera_Personal1`  
    FOREIGN KEY (`Personal_idPersonal`)  
    REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

## Sentencia SQL de la Entidad Contrato



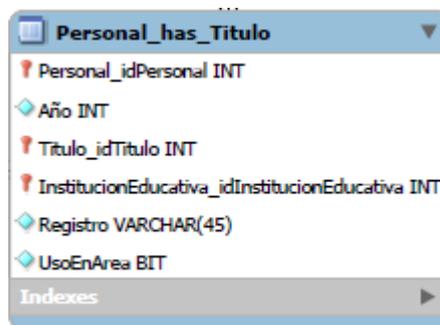
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Contrato` (  
  `idContrato` INT NOT NULL AUTO_INCREMENT,  
  `CodigoContrato` VARCHAR(45) NOT NULL,  
  `FechaInicio` DATE NOT NULL,  
  `FechaFin` DATE NULL,  
  `Personal_idPersonal` INT NOT NULL,  
  `Cargo_idCargo` INT NOT NULL,  
  `FechaAnulacion` DATE NULL,  
  `Remuneracion` DECIMAL(10,2) NULL,  
  PRIMARY KEY (`idContrato`),  
  INDEX `fk_Contrato_Personal1_idx` (`Personal_idPersonal` ASC),  
  INDEX `fk_Contrato_Cargo1_idx` (`Cargo_idCargo` ASC),  
  CONSTRAINT `fk_Contrato_Personal1`  
    FOREIGN KEY (`Personal_idPersonal`)  
    REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION),  
  CONSTRAINT `fk_Contrato_Cargo1`  
    FOREIGN KEY (`Cargo_idCargo`)  
    REFERENCES `TalentoHumano`.`Cargo` (`idCargo`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

```

`TipoContrato` INT NULL COMMENT '0 con relacion dependencia\n1 sin
relacion dependencia\n2 nombramiento',
`Personal_idPersonal` INT NOT NULL,
`FechaAnulacion` DATE NULL,
`Cargo_idCargo` INT NOT NULL,
`Remuneracion` DECIMAL(10,2) NOT NULL,
PRIMARY KEY (`idContrato`),
UNIQUE INDEX `CodigoContrato_UNIQUE` (`CodigoContrato` ASC),
INDEX `fk_Contrato_Personal1_idx` (`Personal_idPersonal` ASC),
INDEX `fk_Contrato_Cargo1_idx` (`Cargo_idCargo` ASC),
CONSTRAINT `fk_Contrato_Personal1`
  FOREIGN KEY (`Personal_idPersonal`)
  REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Contrato_Cargo1`
  FOREIGN KEY (`Cargo_idCargo`)
  REFERENCES `TalentoHumano`.`Cargo` (`idCargo`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencia SQL de la Entidad Personal\_has\_Titulo



```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Personal_has_Titulo` (
  `Personal_idPersonal` INT NOT NULL,
  `Año` INT NOT NULL,
  `Titulo_idTitulo` INT NOT NULL,
  `InstitucionEducativa_idInstitucionEducativa` INT NOT NULL,
  `Registro` VARCHAR(45) NOT NULL,
  `UsoEnArea` BIT NOT NULL DEFAULT 0,
  PRIMARY KEY (`Personal_idPersonal`, `Titulo_idTitulo`,
  `InstitucionEducativa_idInstitucionEducativa`),
  INDEX `fk_Personal_has_Titulo_Personal1_idx` (`Personal_idPersonal` ASC),
  INDEX `fk_Personal_has_Titulo_Titulo1_idx` (`Titulo_idTitulo` ASC),
  INDEX `fk_Personal_has_Titulo_InstitucionEducativa1_idx`

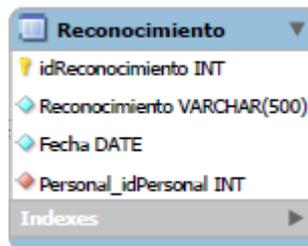
```

```

(`InstitucionEducativa_idInstitucionEducativa` ASC),
CONSTRAINT `fk_Personal_has_Titulo_Personal1`
  FOREIGN KEY (`Personal_idPersonal`)
  REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Personal_has_Titulo_Titulo1`
  FOREIGN KEY (`Titulo_idTitulo`)
  REFERENCES `TalentoHumano`.`Titulo` (`idTitulo`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Personal_has_Titulo_InstitucionEducativa1`
  FOREIGN KEY (`InstitucionEducativa_idInstitucionEducativa`)
  REFERENCES `TalentoHumano`.`InstitucionEducativa`
  (`idInstitucionEducativa`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencia SQL de la Entidad Reconocimiento

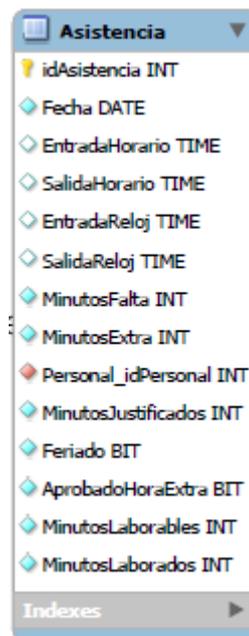


```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Reconocimiento` (
  `idReconocimiento` INT NOT NULL AUTO_INCREMENT,
  `Reconocimiento` VARCHAR(500) NOT NULL,
  `Fecha` DATE NOT NULL,
  `Personal_idPersonal` INT NOT NULL,
  PRIMARY KEY (`idReconocimiento`),
  INDEX `fechareconocimiento` (`Fecha` ASC),
  INDEX `fk_Reconocimiento_Personal1_idx` (`Personal_idPersonal` ASC),
  CONSTRAINT `fk_Reconocimiento_Personal1`
    FOREIGN KEY (`Personal_idPersonal`)
    REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

```

## Sentencia SQL de la Entidad Asistencia



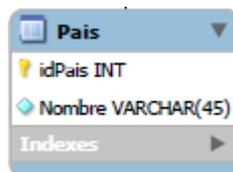
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Asistencia` (  
  `idAsistencia` INT NOT NULL AUTO_INCREMENT,  
  `Fecha` DATE NOT NULL,  
  `EntradaHorario` TIME NULL,  
  `SalidaHorario` TIME NULL,  
  `EntradaReloj` TIME NULL,  
  `SalidaReloj` TIME NULL,  
  `MinutosFalta` INT NOT NULL,  
  `MinutosExtra` INT NOT NULL,  
  `Personal_idPersonal` INT NOT NULL,  
  `MinutosJustificados` INT NOT NULL,  
  `Feriado` BIT NOT NULL,  
  `AprobadoHoraExtra` BIT NOT NULL,  
  `MinutosLaborables` INT NOT NULL,  
  `MinutosLaborados` INT NOT NULL,  
  PRIMARY KEY (`idAsistencia`),  
  INDEX `fk_Asistencia_Personal1_idx` (`Personal_idPersonal` ASC),  
  CONSTRAINT `fk_Asistencia_Personal1`  
    FOREIGN KEY (`Personal_idPersonal`)  
    REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

### Sentencia SQL de la Entidad Feriado



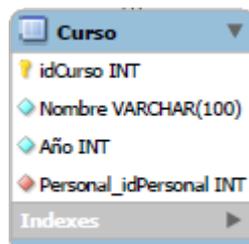
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Feriado` (  
  `Fecha` DATE NOT NULL,  
  `Descripcion` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Fecha`))  
ENGINE = InnoDB
```

### Sentencia SQL de la Entidad Pais



```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Pais` (  
  `idPais` INT NOT NULL AUTO_INCREMENT,  
  `Nombre` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idPais`))  
ENGINE = InnoDB
```

### Sentencia SQL de la Entidad Curso



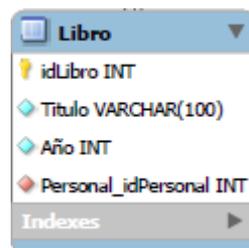
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Curso` (  
  `idCurso` INT NOT NULL AUTO_INCREMENT,  
  `Nombre` VARCHAR(100) NOT NULL,  
  `Año` INT NOT NULL,  
  `Personal_idPersonal` INT NOT NULL,
```

```

PRIMARY KEY (`idCurso`),
INDEX `fk_Curso_Personal1_idx` (`Personal_idPersonal` ASC),
CONSTRAINT `fk_Curso_Personal1`
  FOREIGN KEY (`Personal_idPersonal`)
  REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencia SQL de la Entidad Libro

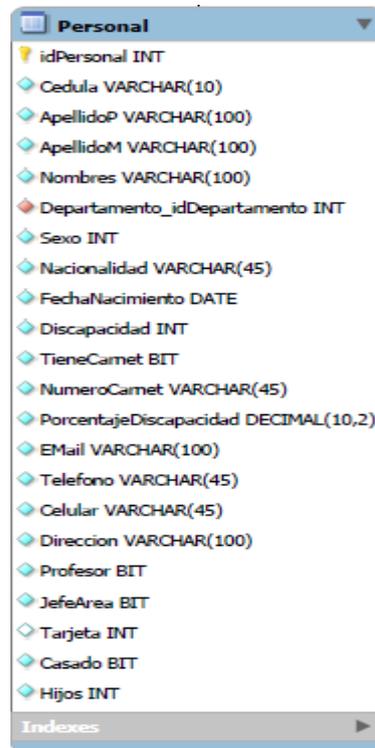


```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Libro` (
  `idLibro` INT NOT NULL AUTO_INCREMENT,
  `Titulo` VARCHAR(100) NOT NULL,
  `Año` INT NOT NULL,
  `Personal_idPersonal` INT NOT NULL,
  PRIMARY KEY (`idLibro`),
  INDEX `fk_Libro_Personal1_idx` (`Personal_idPersonal` ASC),
  CONSTRAINT `fk_Libro_Personal1`
    FOREIGN KEY (`Personal_idPersonal`)
    REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

```

## Sentencia SQL de la Entidad Personal



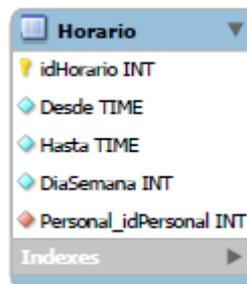
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Personal` (  
  `idPersonal` INT NOT NULL AUTO_INCREMENT,  
  `Cedula` VARCHAR(10) NOT NULL,  
  `ApellidoP` VARCHAR(100) NOT NULL,  
  `ApellidoM` VARCHAR(100) NOT NULL,  
  `Nombres` VARCHAR(100) NOT NULL,  
  `Departamento_idDepartamento` INT NOT NULL,  
  `Sexo` INT NOT NULL DEFAULT 0,  
  `Nacionalidad` VARCHAR(45) NOT NULL,  
  `FechaNacimiento` DATE NOT NULL,  
  `Discapacidad` INT NOT NULL DEFAULT 0 COMMENT '0 ninguna\n1  
fisica\n2 psicologica\n3 visual\n4 auditiva\n',  
  `TieneCarnet` BIT NOT NULL,  
  `NumeroCarnet` VARCHAR(45) NOT NULL,  
  `PorcentajeDiscapacidad` DECIMAL(10,2) NOT NULL,  
  `EMail` VARCHAR(100) NOT NULL,  
  `Telefono` VARCHAR(45) NOT NULL,  
  `Celular` VARCHAR(45) NOT NULL,  
  `Direccion` VARCHAR(100) NOT NULL,  
  `Profesor` BIT NOT NULL DEFAULT 0,  
  `JefeArea` BIT NOT NULL DEFAULT 0,  
  `Tarjeta` INT NULL DEFAULT 0,  
  `Casado` BIT NOT NULL DEFAULT 0,  
  `Hijos` INT NOT NULL DEFAULT 0,
```

```

PRIMARY KEY (`idPersonal`),
UNIQUE INDEX `cedulapersonal` (`Cedula` ASC),
INDEX `fk_Personal_Departamento1_idx` (`Departamento_idDepartamento`
ASC),
CONSTRAINT `fk_Personal_Departamento1`
FOREIGN KEY (`Departamento_idDepartamento`)
REFERENCES `TalentoHumano`.`Departamento` (`idDepartamento`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencia SQL de la Entidad Horario

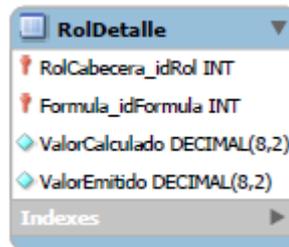


```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Horario` (
`idHorario` INT NOT NULL AUTO_INCREMENT,
`Desde` TIME NOT NULL,
`Hasta` TIME NOT NULL,
`DiaSemana` INT NOT NULL,
`Personal_idPersonal` INT NOT NULL,
PRIMARY KEY (`idHorario`),
INDEX `fk_Horario_Personal1_idx` (`Personal_idPersonal` ASC),
CONSTRAINT `fk_Horario_Personal1`
FOREIGN KEY (`Personal_idPersonal`)
REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB

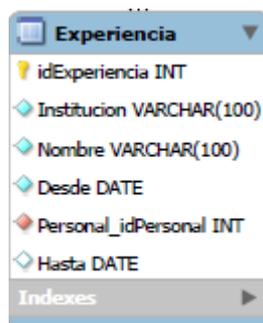
```

## Sentencia SQL de la Entidad RolDetalle



```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`RolDetalle` (  
  `RolCabecera_idRol` INT NOT NULL AUTO_INCREMENT,  
  `Formula_idFormula` INT NOT NULL,  
  `ValorCalculado` DECIMAL(8,2) NOT NULL DEFAULT 0,  
  `ValorEmitido` DECIMAL(8,2) NOT NULL DEFAULT 0,  
  PRIMARY KEY (`RolCabecera_idRol`, `Formula_idFormula`),  
  INDEX `fk_RolDetalle_Formula1_idx` (`Formula_idFormula` ASC),  
  CONSTRAINT `fk_RolDetalle_RolCabecera1`  
    FOREIGN KEY (`RolCabecera_idRol`)  
    REFERENCES `TalentoHumano`.`RolCabecera` (`idRol`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_RolDetalle_Formula1`  
    FOREIGN KEY (`Formula_idFormula`)  
    REFERENCES `TalentoHumano`.`Formula` (`idFormula`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

## Sentencia SQL de la Entidad Experiencia



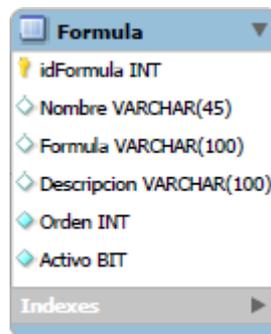
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Experiencia` (  
  `idExperiencia` INT NOT NULL AUTO_INCREMENT,  
  `Institucion` VARCHAR(100) NOT NULL,  
  `Nombre` VARCHAR(100) NOT NULL,
```

```

`Desde` DATE NOT NULL,
`Personal_idPersonal` INT NOT NULL,
`Hasta` DATE NULL,
PRIMARY KEY (`idExperiencia`),
INDEX `fk_Experiencia_Personal1_idx` (`Personal_idPersonal` ASC),
CONSTRAINT `fk_Experiencia_Personal1`
  FOREIGN KEY (`Personal_idPersonal`)
  REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencia SQL de la Entidad Formula

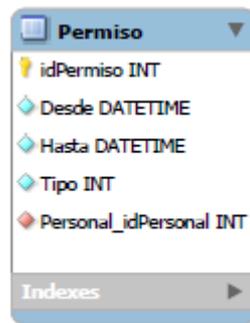


```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Formula` (
  `idFormula` INT NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(45) NULL,
  `Formula` VARCHAR(100) NULL DEFAULT 0,
  `Descripcion` VARCHAR(100) NULL,
  `Orden` INT NOT NULL DEFAULT 0,
  `Activo` BIT NOT NULL DEFAULT 1,
  PRIMARY KEY (`idFormula`),
  UNIQUE INDEX `Nombre_UNIQUE` (`Nombre` ASC))
ENGINE = InnoDB

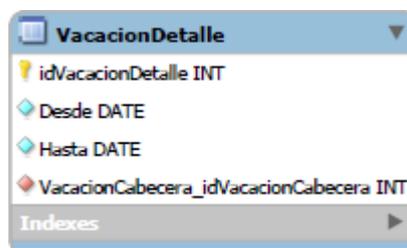
```

## Sentencia SQL de la Entidad Permiso



```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Permiso` (  
  `idPermiso` INT NOT NULL AUTO_INCREMENT,  
  `Desde` DATETIME NOT NULL,  
  `Hasta` DATETIME NOT NULL,  
  `Tipo` INT NOT NULL,  
  `Personal_idPersonal` INT NOT NULL,  
  PRIMARY KEY (`idPermiso`),  
  INDEX `fk_Permiso_Personal1_idx` (`Personal_idPersonal` ASC),  
  CONSTRAINT `fk_Permiso_Personal1`  
    FOREIGN KEY (`Personal_idPersonal`)  
    REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

## Sentencia SQL de la Entidad VacacionDetalle



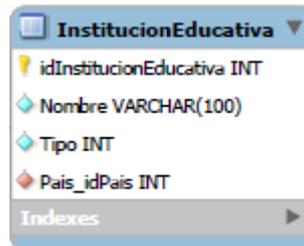
```
CREATE TABLE IF NOT EXISTS `TalentoHumano`.`VacacionDetalle` (  
  `idVacacionDetalle` INT NOT NULL AUTO_INCREMENT,  
  `Desde` DATE NOT NULL,  
  `Hasta` DATE NOT NULL,  
  `VacacionCabecera_idVacacionCabecera` INT NOT NULL,  
  PRIMARY KEY (`idVacacionDetalle`),  
  INDEX `fk_VacacionDetalle_VacacionCabecera1_idx`  
    (`VacacionCabecera_idVacacionCabecera` ASC),  
  CONSTRAINT `fk_VacacionDetalle_VacacionCabecera1`
```

```

FOREIGN KEY (`VacacionCabecera_idVacacionCabecera`)
REFERENCES `TalentoHumano`.`VacacionCabecera` (`idVacacionCabecera`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencias SQL de la Entidad InstitucionEducativa

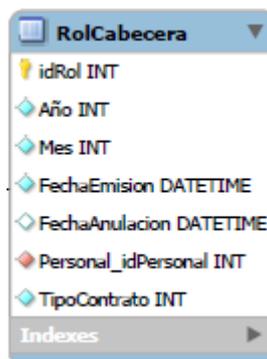


```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`InstitucionEducativa` (
  `idInstitucionEducativa` INT NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(100) NOT NULL,
  `Tipo` INT NOT NULL,
  `Pais_idPais` INT NOT NULL,
  PRIMARY KEY (`idInstitucionEducativa`),
  INDEX `fk_InstitucionEducativa_Pais1_idx` (`Pais_idPais` ASC),
  CONSTRAINT `fk_InstitucionEducativa_Pais1`
    FOREIGN KEY (`Pais_idPais`)
    REFERENCES `TalentoHumano`.`Pais` (`idPais`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencias SQL de la Entidad RolCabecera



```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`RolCabecera` (
  `idRol` INT NOT NULL AUTO_INCREMENT,

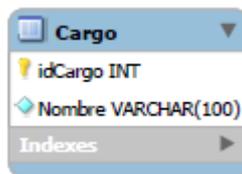
```

```

`Año` INT NOT NULL,
`Mes` INT NOT NULL,
`FechaEmision` DATETIME NOT NULL,
`FechaAnulacion` DATETIME NULL,
`Personal_idPersonal` INT NOT NULL,
`TipoContrato` INT NOT NULL DEFAULT 0,
PRIMARY KEY (`idRol`),
INDEX `fk_RolCabecera_Personal1_idx` (`Personal_idPersonal` ASC),
CONSTRAINT `fk_RolCabecera_Personal1`
  FOREIGN KEY (`Personal_idPersonal`)
  REFERENCES `TalentoHumano`.`Personal` (`idPersonal`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB

```

### Sentencias SQL de la Entidad Cargo



```

CREATE TABLE IF NOT EXISTS `TalentoHumano`.`Cargo` (
  `idCargo` INT NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`idCargo`))
ENGINE = InnoDB

```

### Código Fuente Clase Personal

```

Public Class Personal
  Inherits ClaseBase

  Enum TipoSexo
    Masculino
    Femenino
  End Enum

  Enum TipoDiscapacidad
    ninguna
    fisica
    psicologica
    visual

```

```
    auditiva
End Enum
```

```
Private pCedula As String
Private pContratos As List(Of Contrato)
Private pTitulos As List(Of PersonalTitulo)
Private pExperiencias As List(Of Experiencia)
Private pCursos As List(Of Curso)
Private pReconocimientos As List(Of Reconocimiento)
Private pLibros As List(Of Libro)
```

```
#Region "Propiedades"
```

```
Property ApellidoP As String = ""
Property ApellidoM As String = ""
Property Nombres As String = ""
Property Departamento As Departamento
Property Sexo As TipoSexo = TipoSexo.Masculino
Property Nacionalidad As String = ""
Property FechaNacimiento As Date = Now
Property Discapacidad As TipoDiscapacidad = TipoDiscapacidad.ninguna
Property TieneCarnet As Boolean = False
Property NumeroCarnet As String = ""
Property PorcentajeDiscapacidad As Decimal = 0
Property EMail As String = ""
Property Telefono As String = ""
Property Celular As String = ""
Property Direccion As String = ""
Property Tarjeta As Integer = 0
Property Profesor As Boolean = False
Property JefeArea As Boolean = False
Property Casado As Boolean
Property Hijos As Integer
```

```
Overloads ReadOnly Property Nombre As String
    Get
        Return ApellidoP + " " + ApellidoM + " " + Nombres
    End Get
```

```
End Property
```

```
Public Overrides Function Utilizado() As Boolean
    Return False
End Function
```

```
ReadOnly Property Cedula As String
    Get
        Return pCedula
    End Get
End Property
```

```
ReadOnly Property Contratos As List(Of Contrato)
    Get
```

```
        pContratos = New List(Of Contrato)
```

```
        Dim tabla As DataTable
```

```
        tabla = AccesoDatos.AbrirTabla("select idContrato, CodigoContrato, FechaInicio,
FechaFin, TipoContrato, FechaAnulacion, Cargo_idCargo, Remuneracion from Contrato
where Personal_idPersonal = " & Id & " order by FechaInicio", Nothing)
```

```
        For Each rst As DataRow In tabla.Rows
```

```
            Dim aux As New Contrato
```

```
            aux.pid = rst!idContrato
```

```
            aux.CodigoContrato = rst!CodigoContrato
```

```
            aux.FechaInicio = rst!FechaInicio
```

```
            aux.FechaFin = rst!FechaFin
```

```
            aux.Tipo = rst!TipoContrato
```

```
            aux.FechaAnulacion = If(IsDBNull(rst!FechaAnulacion), Nothing,
rst!fechaanulacion)
```

```
            aux.Cargo = New Cargo(rst!Cargo_idCargo)
```

```
            aux.Remuneracion = rst!Remuneracion
```

```
            pContratos.Add(aux)
```

```
        Next
```

```
        Return pContratos
```

```
    End Get
```

```
End Property
```

```
ReadOnly Property ContratoActual As Contrato
```

```
    Get
```

```
        For Each aux As Contrato In Contratos
```

```
            If Now.Date >= aux.FechaInicio And Now.Date <= aux.FechaFin Then
```

```
                Return aux
```

```
            End If
```

```
        Next
```

```
        Return Nothing
```

```
    End Get
```

```
End Property
```

```
ReadOnly Property Contrato(desde As Date) As Contrato
```

```
    Get
```

```
        For Each aux As Contrato In Contratos
```

```
            If desde >= aux.FechaInicio And desde <= aux.FechaFin Then
```

```
                Return aux
```

```
            End If
```

```
        Next
```

```
        Return Nothing
```

```
    End Get
```

```
End Property
```

```
ReadOnly Property Titulos As List(Of PersonalTitulo)
```

```
    Get
```

```
pTitulos = New List(Of PersonalTitulo)
```

```
Dim tabla As DataTable  
tabla = AccesoDatos.AbrirTabla("select Personal_idPersonal, Año, Titulo_idTitulo,  
InstitucionEducativa_idInstitucionEducativa, Registro, UsoEnArea from  
Personal_has_Titulo where Personal_idPersonal = @1 order by año", Nothing, Id)  
For Each rst As DataRow In tabla.Rows  
Dim aux As New PersonalTitulo  
aux.Año = rst!año  
aux.Titulo = New Titulo(rst!titulo_idtitulo)  
aux.Institucion  
= New Institucion(rst!InstitucionEducativa_idInstitucionEducativa)  
aux.Registro = rst!registro  
aux.EnArea = rst!UsoenArea  
pTitulos.Add(aux)  
Next  
Return pTitulos  
End Get  
End Property
```

```
ReadOnly Property Experiencias As List(Of Experiencia)  
Get
```

```
pExperiencias = New List(Of Experiencia)
```

```
Dim tabla As DataTable  
tabla = AccesoDatos.AbrirTabla("select idExperiencia, Nombre, Institucion, Desde,  
Hasta from Experiencia where Personal_idPersonal = @1 order by desde", Nothing, Id)  
For Each rst As DataRow In tabla.Rows  
Dim aux As New Experiencia  
aux.Nombre = rst!nombre  
aux.Institucion = rst!institucion  
aux.Desde = rst!desde  
aux.Hasta = rst!hasta  
aux.pid = rst!idexperiencia  
pExperiencias.Add(aux)  
Next  
Return pExperiencias  
End Get  
End Property
```

```
ReadOnly Property Cursos As List(Of Curso)  
Get
```

```
pCursos = New List(Of Curso)
```

```
Dim tabla As DataTable  
tabla = AccesoDatos.AbrirTabla("select idCurso, Nombre, Año,
```

```

Personal_idPersonal from Curso where Personal_idPersonal = @1 order by
año", Nothing, Id)
    For Each rst As DataRow In tabla.Rows
        Dim aux As New Curso
        aux.Año = rst!año
        aux.Nombre = rst!nombre
        aux.pid = rst!idCurso
        pCursos.Add(aux)
    Next

    Return pCursos
End Get
End Property

ReadOnly Property Reconocimientos As List(Of Reconocimiento)
Get

    pReconocimientos = New List(Of Reconocimiento)

    Dim tabla As DataTable
    tabla = AccesoDatos.AbrirTabla("select idReconocimiento, Reconocimiento,
Fecha, Personal_idPersonal from Reconocimiento where Personal_idPersonal = @1
order by fecha", Nothing, Id)
    For Each rst As DataRow In tabla.Rows
        Dim aux As New Reconocimiento
        aux.Fecha = rst!fecha
        aux.Nombre = rst!reconocimiento
        aux.pid = rst!idReconocimiento
        pReconocimientos.Add(aux)
    Next
    Return pReconocimientos
End Get
End Property

ReadOnly Property Vacaciones As List(Of Vacacion)
Get
    Dim aux As New List(Of Vacacion)
    For Each rst As DataRow In AccesoDatos.AbrirTabla("select idVacacionCabecera,
Personal_idPersonal, Año, Dias from VacacionCabecera where Personal_idPersonal =
@1 order by año", Nothing, Id).Rows
        Dim vac As New Vacacion()
        vac.pid = rst!idVacacionCabecera
        vac.Persona = Me
        vac.Año = rst!año
        vac.Dias = rst!dias
        aux.Add(vac)
    Next
    Return aux
End Get
End Property

```

ReadOnly Property Libros As List(Of Libro)

Get

pLibros = New List(Of Libro)

Dim tabla As DataTable

tabla = AccesoDatos.AbrirTabla("select idLibro, Titulo, Año, Personal\_idPersonal  
from Libro where Personal\_idPersonal = @1", Nothing, Id)

For Each rst As DataRow In tabla.Rows

Dim aux As New Libro

aux.Nombre = rst!titulo

aux.Año = rst!año

aux.pid = rst!idLibro

pLibros.Add(aux)

Next

Return pLibros

End Get

End Property

ReadOnly Property Horario() As Horario

Get

Dim tabla As DataTable

tabla = AccesoDatos.AbrirTabla("select Desde, Hasta,DiaSemana from Horario  
where Personal\_idPersonal = @1 order by DiaSemana, Desde", Nothing, Id)

Dim aux As New Horario

For Each rst As DataRow In tabla.Rows

aux.Semana(rst!Diasemana).Jornadas.Add(New Horario.Jornada(New Date(1900, 1, 1, CType(rst!desde, TimeSpan).Hours, CType(rst!desde, TimeSpan).Minutes, 0), New Date(1900, 1, 1, CType(rst!hasta, TimeSpan).Hours, CType(rst!hasta, TimeSpan).Minutes, 0)))

Next

Return aux

End Get

End Property

**#End Region**

**#Region "Constructores"**

Public Sub New(id As Integer)

MyBase.New(id, "", False)

End Sub

Public Sub New(cedula As String)

MyBase.New(0, "", False)

Dim tabla As DataTable

pCedula = cedula

tabla = AccesoDatos.AbrirTabla("select idPersonal, Cedula, ApellidoP, ApellidoM,

Nombres, Departamento\_idDepartamento, Sexo, Nacionalidad, FechaNacimiento, Discapacidad, TieneCarnet, NumeroCarnet, PorcentajeDiscapacidad, EMail, Telefono, Celular, Direccion, Tarjeta from Personal where Cedula = @1", Nothing, cedula)

For Each rst As DataRow In tabla.Rows

pid = rst!idPersonal

ApellidoP = rst!ApellidoP

ApellidoM = rst!ApellidoM

Nombres = rst!Nombres

Departamento = New Departamento(rst!Departamento\_idDepartamento)

Sexo = Sexo

Nacionalidad = rst!Nacionalidad

FechaNacimiento = rst!FechaNacimiento

Discapacidad = rst!Discapacidad

TieneCarnet = rst!TieneCarnet

NumeroCarnet = rst!NumeroCarnet

PorcentajeDiscapacidad = rst!PorcentajeDiscapacidad

EMail = rst!email

Telefono = rst!Telefono

Celular = rst!Celular

Direccion = rst!Direccion

Tarjeta = rst!tarjeta

Next

End Sub

**#End Region**

**#Region "Metodos"**

Public Overrides Sub Borrar()

End Sub

Public Function TieneContrato(fecha) As Boolean

Return AccesoDatos.AbrirEscalar("select count(\*) from contrato where Personal\_idPersonal = @1 and (FechaAnulacion is null and @2 between FechaInicio and FechaFin) or (@3 between FechaInicio and FechaAnulacion)", Nothing, Me, fecha, fecha)

End Function

Public Function TieneVacacion(fecha) As Boolean

Return AccesoDatos.AbrirEscalar("select count(\*) from VacacionCabecera inner join VacacionDetalle on VacacionCabecera.idVacacionCabecera = VacacionDetalle.VacacionCabecera\_idVacacionCabecera where Personal\_idPersonal = @1 and (@2 between Desde and Hasta) ", Nothing, Me, fecha)

End Function

Public Overrides Sub Guardar()

Dim trans As MySQLTransaction = Nothing

Sesion.Conexion.Open()

trans = Sesion.Conexion.BeginTransaction

If Nuevo Then

Try

```
AccesoDatos.RunSql("INSERT INTO Personal (Cedula, ApellidoP, ApellidoM,
Nombres, Departamento_idDepartamento, Sexo, Nacionalidad, FechaNacimiento,
Discapacidad, TieneCarnet, NumeroCarnet, PorcentajeDiscapacidad, EMail, Telefono,
Celular, Direccion, Profesor, JefeArea, Tarjeta, Casado, Hijos) VALUES (@Cedula,
@ApellidoP, @ApellidoM, @Nombres, @Departamento_idDepartamento, @Sexo,
@Nacionalidad, @FechaNacimiento, @Discapacidad, @TieneCarnet, @NumeroCarnet,
@PorcentajeDiscapacidad, @EMail, @Telefono, @Celular, @Direccion, @Profesor,
@JefeArea, @Tarjeta, @casado, @Hijo);", trans, Cedula, ApellidoP, ApellidoM, Nombres,
Departamento.Id, Sexo, Nacionalidad, FechaNacimiento, Discapacidad, TieneCarnet,
NumeroCarnet, PorcentajeDiscapacidad, EMail, Telefono, Celular, Direccion, Profesor,
JefeArea, Tarjeta, Casado, Hijos)
```

```
pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
```

Catch ex As Exception

```
If trans IsNot Nothing Then trans.Rollback()
```

```
Sesion.Conexion.Close()
```

Throw

End Try

Else

Try

```
AccesoDatos.RunSql("update Personal set Cedula = @Cedula, ApellidoP =
@ApellidoP, ApellidoM = @1, Nombres = @2, Departamento_idDepartamento = @3,
Sexo = @4, Nacionalidad = @5, FechaNacimiento = @6, Discapacidad = @7,
TieneCarnet = @8, NumeroCarnet = @9, PorcentajeDiscapacidad = @a, EMail = @b,
Telefono = @c, Celular = @d, Direccion = @e, Profesor = @f, JefeArea = @g, Tarjeta =
@h, Casado = @J, Hijos = @K where idPersonal = @i", trans, Cedula, ApellidoP,
ApellidoM, Nombres, Departamento.Id, Sexo, Nacionalidad, FechaNacimiento,
Discapacidad, TieneCarnet, NumeroCarnet, PorcentajeDiscapacidad, EMail, Telefono,
Celular, Direccion, Profesor, JefeArea, Tarjeta, Casado, Hijos, Id)
```

Catch ex As Exception

```
If trans IsNot Nothing Then trans.Rollback()
```

```
Sesion.Conexion.Close()
```

Throw

End Try

End If

If pContratos IsNot Nothing Then

```
Dim lista As String = "0"
```

```
For Each c As Contrato In pContratos
```

```
If c.Nuevo Then
```

```
AccesoDatos.RunSql("INSERT INTO Contrato (CodigoContrato, FechaInicio,
FechaFin, TipoContrato, Personal_idPersonal, FechaAnulacion, Cargo_idCargo,
Remuneracion) VALUES (@CodigoContrato, @FechaInicio, @FechaFin, @TipoContrato,
@Personal_idPersonal, @FechaAnulacion, @Cargo_idCargo, @Remuneracion);", trans,
c.CodigoContrato, c.FechaInicio, c.FechaFin, c.Tipo, Id, c.FechaAnulacion, c.Cargo,
c.Remuneracion)
```

```
c.pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
```

Else

```

        AccesoDatos.RunSql("update Contrato setCodigoContrato= @1, FechaInicio
= @2, FechaFin = @3, TipoContrato = @4, Personal_idPersonal = @5, FechaAnulacion =
@6, Cargo_idCargo = @7, Remuneracion = @8 where idContrato = @9", trans,
c.CodigoContrato, c.FechaInicio, c.FechaFin, c.Tipo, Id, c.FechaAnulacion, c.Cargo,
c.Remuneracion, c.Id)

```

```

        End If

```

```

        lista &= "," & c.Id

```

```

    Next

```

```

    AccesoDatos.RunSql("delete from Contrato where Personal_idPersonal = @1 and
idContrato not in (" & lista & ")", trans, Me)

```

```

End If

```

```

If pTitulos IsNot Nothing Then

```

```

    AccesoDatos.RunSql("delete from Personal_has_Titulo where
Personal_idPersonal = " & Id, trans)

```

```

    For Each c As PersonalTitulo In pTitulos

```

```

        'If c.Nuevo Then

```

```

            AccesoDatos.RunSql("INSERT INTO Personal_has_Titulo (Personal_idPersonal,
Año, Titulo_idTitulo, InstitucionEducativa_idInstitucionEducativa, Registro, UsoEnArea)
VALUES (@Personal_idPersonal, @Año, @Titulo_idTitulo,
@InstitucionEducativa_idInstitucionEducativa, @Registro, @enarea);", trans, Id, c.Año,
c.Titulo, c.Institucion, c.Registro, c.EnArea)

```

```

            'c.pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)

```

```

            'Else

```

```

                AccesoDatos.RunSql("update Personal_has_Titulo set Año = @1,
Titulo_idTitulo = @2, InstitucionEducativa_idInstitucionEducativa = @3, Registro = @4
where VALUES (@Personal_idPersonal, @Año, @Titulo_idTitulo,
@InstitucionEducativa_idInstitucionEducativa, @Registro);", trans, Id, c.Año, c.Titulo,
c.Institucion, c.Registro)

```

```

            'End If

```

```

            'lista &= "," & c.Id

```

```

        Next

```

```

End If

```

```

If pCursos IsNot Nothing Then

```

```

    Dim lista As String = "0"

```

```

    For Each c As Curso In pCursos

```

```

        If c.Nuevo Then

```

```

            AccesoDatos.RunSql("INSERT INTO Curso (Nombre, Año,
Personal_idPersonal) VALUES (@Nombre, @Año, @Personal_idPersonal)", trans,
c.Nombre, c.Año, Id)

```

```

            c.pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)

```

```

        Else

```

```

            AccesoDatos.RunSql("update Curso set Nombre = @n, Año = @a,
Personal_idPersonal = @p where idCurso = @9", trans, c.Nombre, c.Año, Id, c.Id)

```

```

        End If

```

```

        lista &= "," & c.Id

```

```

    Next

```

```

        AccesoDatos.RunSql("delete from Curso where Personal_idPersonal = @1 and
idCurso not in (" & lista & ")", trans, Me)
    End If

    If pReconocimientos IsNot Nothing Then
        Dim lista As String = "0"
        For Each c As Reconocimiento In pReconocimientos
            If c.Nuevo Then
                AccesoDatos.RunSql("INSERT INTO Reconocimiento (Reconocimiento, Fecha,
Personal_idPersonal) VALUES (@Nombre, @Año, @Personal_idPersonal)", trans,
c.Nombre, c.Fecha, Id)
                c.pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
            Else
                AccesoDatos.RunSql("update Reconocimiento set Reconocimiento = @n,
Fecha = @a, Personal_idPersonal = @p where idreconocimiento = @9", trans, c.Nombre,
c.Fecha, Id, c.Id)
            End If
            lista &= "," & c.Id
        Next
        AccesoDatos.RunSql("delete from reconocimiento where Personal_idPersonal =
@1 and idreconocimiento not in (" & lista & ")", trans, Me)
    End If

    If pExperiencias IsNot Nothing Then
        Dim lista As String = "0"
        For Each e As Experiencia In pExperiencias
            If e.Nuevo Then
                AccesoDatos.RunSql("insert into Experiencia (Nombre, Institucion, Desde,
Hasta, Personal_idPersonal) values (@Nombre, @Experiencia, @Desde , @Hasta, @55)",
trans, e.Nombre, e.Institucion, e.Desde, e.Hasta, Id)
                e.pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
            Else
                AccesoDatos.RunSql("update Experiencia set Nombre = @1, Institucion = @2,
Desde = @3, Hasta = @4, Personal_idPersonal = @5 where idExperiencia = @6", trans,
e.Nombre, e.Institucion, e.Desde, e.Hasta, Id, e.Id)
            End If
            lista &= "," & e.Id
        Next
    End If

    If pLibros IsNot Nothing Then
        Dim lista As String = "0"
        For Each e As Libro In pLibros
            If e.Nuevo Then
                AccesoDatos.RunSql("insert into Libro (Titulo, Año, Personal_idPersonal)
values (@Nombre, @Experiencia, @Desde )", trans, e.Nombre, e.Año, Id)
                e.pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
            Else
                AccesoDatos.RunSql("update Libro set Titulo = @1, Año = @2 where idLibro =
@6", trans, e.Nombre, e.Año, e.Id)
            End If
        Next
    End If

```

```

        End If
        lista &= "," & e.Id
    Next

End If

trans.Commit()
Sesion.Conexion.Close()
End Sub

ReadOnly Property MinutosTrabajados(desde As Date, hasta As Date) As Integer
Get
    Try
        Sesion.Conexion.Open()
        Return AccesoDatos.AbrirEscalar("select ifnull(sum(MinutosLaborados),0) from
Asistencia where Personal_idPersonal = @1 and Fecha between @2 and
@3", Nothing, Me, desde, hasta)
    Finally
        Sesion.Conexion.Close()
    End Try
    Return 0
End Get
End Property

ReadOnly Property MinutosFalta(desde As Date, hasta As Date) As Integer
Get
    Try
        Sesion.Conexion.Open()
        Return AccesoDatos.AbrirEscalar("select ifnull(sum(MinutosFalta),0) from
Asistencia where Personal_idPersonal = @1 and Fecha between @2 and
@3", Nothing, Me, desde, hasta)
    Finally
        Sesion.Conexion.Close()
    End Try
    Return 0
End Get
End Property

ReadOnly Property MinutosJustificados(desde As Date, hasta As Date) As Integer
Get
    Try
        Sesion.Conexion.Open()
        Return AccesoDatos.AbrirEscalar("select ifnull(sum(MinutosJustificados),0)
from Asistencia where Personal_idPersonal = @1 and Fecha between @2 and
@3", Nothing, Me, desde, hasta)
    Finally
        Sesion.Conexion.Close()
    End Try
    Return 0
End Get

```

End Property

ReadOnly Property MinutosExtra(desde As Date, hasta As Date) As Integer

Get

Try

Sesion.Conexion.Open()

Return AccesoDatos.AbrirEscalar("select ifnull(sum(MinutosExtra),0) from Asistencia where Personal\_idPersonal = @1 and Fecha between @2 and @3 and AprobadoHoraExtra = 1", Nothing, Me, desde, hasta)

Finally

Sesion.Conexion.Close()

End Try

Return 0

End Get

End Property

**#End Region**

Shared Function CedulaValida(cedula As String)

Return True

End Function

Friend Sub LlenarDatos(ByRef valores As ParametroRol, desde As Date, hasta As Date)

valores.JefeArea = JefeArea

valores.MinutosExtra = MinutosExtra(desde, hasta)

valores.MinutosFalta = MinutosFalta(desde, hasta)

valores.MinutosJustificados = MinutosJustificados(desde, hasta)

valores.MinutosLaborados = MinutosTrabajados(desde, hasta)

valores.NivelMaestria = 0

For Each t As PersonalTitulo In Titulos

If t.Titulo.Nivel = 3 And valores.NivelMaestria = 0 Then

valores.NivelMaestria = 1

Elseif t.Titulo.Nivel = 3 And t.EnArea Then

valores.NivelMaestria = 2

End If

If t.EnArea Then

valores.TituloEnCampo = True

End If

Next

valores.Publicaciones = Libros.Count

valores.Profesor = Profesor

valores.Casado = Casado

valores.Hijos = Hijos

Try

valores.SueldoBasico = Contrato(hasta).Remuneracion

Catch ex As Exception

valores.SueldoBasico = 0

End Try

```
End Sub
End Class
```

## Código Fuente Clase Asistencia

```
Public Class Asistencia
```

```
Inherits ClaseBase
```

```
Property HorarioEntrada As Date
```

```
Property HorarioSalida As Date
```

```
Property MarcaEntrada As Date
```

```
Property MarcaSalida As Date
```

```
Property Feriado As Boolean
```

```
Property TiempoTrabajado As TimeSpan
```

```
Property TiempoFalta As TimeSpan
```

```
Property TiempoJustificado As TimeSpan
```

```
Property TiempoExtra As TimeSpan = New TimeSpan(0)
```

```
Property Aprobado As Boolean = False
```

```
Property Persona As Personal
```

```
Property Fecha As Date
```

```
Public Sub New(Fecha As Date, Persona As Personal)
```

```
    MyBase.New(0, "", 0)
```

```
    Me.Persona = Persona
```

```
    Me.Fecha = Fecha
```

```
End Sub
```

```
ReadOnly Property TiempoLaborable As TimeSpan
```

```
Get
```

```
    Return HorarioSalida - HorarioEntrada
```

```
End Get
```

```
End Property
```

```
Public Overrides Sub Borrar()
```

```
End Sub
```

```
Public Overrides Sub Guardar()
```

```
    Sesion.Conexion.Open()
```

```
    If Nuevo Then
```

```
        AccesoDatos.RunSql("INSERT INTO Asistencia (Fecha, EntradaHorario,
SalidaHorario, EntradaReloj, SalidaReloj, MinutosFalta, MinutosExtra,
Personal_idPersonal, MinutosJustificados, Feriado, AprobadoHoraExtra,
MinutosLaborables, MinutosLaborados) VALUES (@Fecha, @EntradaHorario,
@SalidaHorario, @EntradaReloj, @SalidaReloj, @MinutosFalta, @MinutosExtra,
@Personal_idPersonal, @MinutosJustificados, @Feriado, @AprobadoHoraExtra,
```

```

@MinutosLaborables, @MinutosLaborados);" , Nothing, Fecha, IIf(HorarioEntrada
= Nothing, DBNull.Value, HorarioEntrada.TimeOfDay), IIf(HorarioSalida = Nothing,
DBNull.Value, HorarioSalida.TimeOfDay), IIf(IsDBNull(MarcaEntrada), DBNull.Value,
MarcaEntrada.TimeOfDay), IIf(IsDBNull(MarcaSalida), DBNull.Value,
MarcaSalida.TimeOfDay), TiempoFalta.TotalMinutes, TiempoExtra.TotalMinutes,
Persona, TiempoJustificado.TotalMinutes, Feriado, Aprobado,
TiempoLaborable.TotalMinutes, TiempoTrabajado.TotalMinutes)
    pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", Nothing)
Else
    AccesoDatos.RunSql("update Asistencia set Fecha = @1, EntradaHorario = @2,
SalidaHorario = @3, EntradaReloj = @4, SalidaReloj = @5, MinutosFalta = @6,
MinutosExtra = @7, Personal_idPersonal = @8, MinutosJustificados = @9, Feriado =
@10, AprobadoHoraExtra = @11, MinutosLaborables = @12, MinutosLaborados = @13
where idAsistencia = @14", Nothing, Fecha, HorarioEntrada, HorarioSalida,
MarcaEntrada, MarcaSalida, TiempoFalta.TotalMinutes, TiempoExtra.TotalMinutes,
Persona, TiempoJustificado.TotalMinutes, Feriado, Aprobado,
TiempoLaborable.TotalMinutes, TiempoTrabajado.TotalMinutes, Id)
End If
    Sesion.Conexion.Close()
End Sub

Public Overrides Function Utilizado() As Boolean
    Return False
End Function
End Class

```

## Código fuente Clase Cargo

```

Public Class Cargo

    Inherits ClaseBase
    Public Sub New(id As Integer, nombre As String)
        MyBase.New(id, nombre, False)
    End Sub
    Public Overrides Sub Borrar()

    End Sub

    Friend Sub New(id As Integer)
        MyBase.New(0, "", 0)
        Dim tabla As DataTable
        tabla = AccesoDatos.AbrirTabla("select idCargo, Nombre from Cargo where idCargo
= @1", Nothing, id)
        For Each rst In tabla.Rows

```

```

        pid = rst!idCargo
        Nombre = rst!Nombre
    Next
End Sub

Public Overrides Sub Guardar()
    Dim trans As MySqlConnection = Nothing
    Sesion.Conexion.Open()
    trans = Sesion.Conexion.BeginTransaction
    If Nuevo Then

        Try
            AccesoDatos.RunSql("INSERT INTO Cargo (Nombre) VALUES (@1);", trans,
Nombre)
            pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)

            'If pDepartamentos IsNot Nothing Then
            '    For Each x As Departamento In Me.Departamentos
            '        x.pdireccion = Me
            '        x.Guardar(trans)
            '    Next
            'End If
        Catch ex As Exception
            If trans IsNot Nothing Then trans.Rollback()
            Sesion.Conexion.Close()

            Throw
        End Try
    Else
        Try
            AccesoDatos.RunSql("update Cargo set Nombre= @1 where idCargo= @2",
trans, pnombre, pid)
        Catch ex As Exception
            If trans IsNot Nothing Then trans.Rollback()
            Sesion.Conexion.Close()
            Throw
        End Try
    End If

    trans.Commit()
    Sesion.Conexion.Close()
End Sub

Public Overrides Function Utilizado() As Boolean
    Return False
End Function
End Class

```

## Código fuente Clase Contrato

```
Public Class Cargo
```

```
    Inherits ClaseBase
```

```
    Public Sub New(id As Integer, nombre As String)
```

```
        MyBase.New(id, nombre, False)
```

```
    End Sub
```

```
    Public Overrides Sub Borrar()
```

```
    End Sub
```

```
    Friend Sub New(id As Integer)
```

```
        MyBase.New(0, "", 0)
```

```
        Dim tabla As DataTable
```

```
        tabla = AccesoDatos.AbrirTabla("select idCargo, Nombre from Cargo where idCargo  
= @1", Nothing, id)
```

```
        For Each rst In tabla.Rows
```

```
            pid = rst!idCargo
```

```
            Nombre = rst!Nombre
```

```
        Next
```

```
    End Sub
```

```
    Public Overrides Sub Guardar()
```

```
        Dim trans As MySQLTransaction = Nothing
```

```
        Sesion.Conexion.Open()
```

```
        trans = Sesion.Conexion.BeginTransaction
```

```
        If Nuevo Then
```

```
            Try
```

```
                AccesoDatos.RunSql("INSERT INTO Cargo (Nombre) VALUES (@1);", trans,  
Nombre)
```

```
                pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
```

```
                'If pDepartamentos IsNot Nothing Then
```

```
                ' For Each x As Departamento In Me.Departamentos
```

```
                '     x.pdireccion = Me
```

```
                '     x.Guardar(trans)
```

```
                ' Next
```

```
                'End If
```

```
            Catch ex As Exception
```

```
                If trans IsNot Nothing Then trans.Rollback()
```

```
                Sesion.Conexion.Close()
```

```
            Throw
```

```
        End Try
```

```
    Else
```

```
        Try
```

```

        AccesoDatos.RunSql("update Cargo set Nombre= @1 where idCargo= @2",
trans, pnombre, pid)
    Catch ex As Exception
        If trans IsNot Nothing Then trans.Rollback()
        Sesion.Conexion.Close()
        Throw
    End Try
End If

trans.Commit()
Sesion.Conexion.Close()
End Sub

Public Overrides Function Utilizado() As Boolean
    Return False
End Function
End Class

```

### Código fuente Clase Curso

```

Public Class Curso
    Inherits ClaseBase
    Implements ICloneable

    Property Año As Integer
    Friend Sub New()
        MyBase.New(0, "", False)
    End Sub
    Public Overrides Sub Borrar()

    End Sub

    Public Overrides Sub Guardar()

    End Sub

    Public Function Clone() As Object Implements System.ICloneable.Clone
        Return Me.MemberwiseClone
    End Function

    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
End Class

```

Public Class Departamento

Inherits ClaseBase

Friend pdireccion As Direccion

Friend Sub New(Direccion As Direccion, nombre As String)

MyBase.New(0, nombre, False)

pdireccion = Direccion

End Sub

Friend Sub New(id As Integer, Direccion As Direccion, nombre As String,  
borrado As Boolean)

MyBase.New(id, nombre, borrado)

pdireccion = Direccion

End Sub

Friend Sub New(id As Integer)

MyBase.New(id, "", False)

Dim tabla As DataTable

tabla = AccesoDatos.AbrirTabla("select idDepartamento, Descripcion,  
Direccion\_idDireccion, Borrado from Departamento where idDepartamento =  
@1", Nothing, id)

For Each rst In tabla.Rows

pdireccion = New Direccion(rst!Direccion\_idDireccion)

pid = rst!iddepartamento

Nombre = rst!descripcion

pborrado = rst!borrado

Next

End Sub

Public Overrides Sub Guardar()

Sesion.Conexion.Open()

Guardar(Nothing)

Sesion.Conexion.Close()

End Sub

Public Overrides Sub Borrar()

If Nuevo Then Exit Sub

Sesion.Conexion.Open()

AccesoDatos.RunSql("delete from Departamento where idDepartamento =  
@3;", Nothing, pid)

Sesion.Conexion.Close()

pborrado = True

End Sub

Public Overloads Sub Guardar(tran As MySqlTransaction)

If Nuevo Then

AccesoDatos.RunSql("INSERT INTO Departamento (Descripcion,  
Direccion\_idDireccion, Borrado) VALUES (@1, @2, 0);", tran, pnombre, pdireccion)

```

        pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", tran)
    Else
        AccesoDatos.RunSql("update Departamento set Descripcion=
@1, Direccion_idDireccion = @2, Borrado = 0 where idDepartamento = @3;", tran,
pnombre, pdireccion, pid)
    End If
End Sub
Public Overrides Function Utilizado() As Boolean
    Sesion.Conexion.Open()
    Try
        Return AccesoDatos.AbrirEscalar("select count(*) from personal inner join
departamento on departamento_iddepartamento = departamento.iddepartamento
inner join direccion on iddireccion = direccion_iddireccion where iddepartamento =
@1", Nothing, Me)
    Catch
        Return True
    Finally
        Sesion.Conexion.Close()
    End Try
End Function
#Region "Propiedades"
ReadOnly Property Direccion As Direccion
    Get
        Return pdireccion
    End Get
End Property
#End Region
End Class

```

### Código fuente Clase Dia Trabajo

```

Public Class DiaTrabajo

    Property Fecha As Date
    Property Jornadas As List(Of Horario.Jornada)
    Property Asistencia As List(Of Asistencia)
    Property Permisos As List(Of Horario.Jornada)
    Property Marcaciones As List(Of Horario.Jornada)
    Property Persona As Personal
    Property Feriado As Boolean

    Public Sub New()

    End Sub

```

```

Public Sub New(Persona As Personal, Dia As Date, feriado As Boolean)
    Fecha = Dia.Date

    If 1 = 2 Then
    Else
        Me.Persona = Persona
        Me.Feriado = feriado
        Jornadas = Persona.Horario.Semana(Weekday(Fecha)).Jornadas
        Asistencia = New List(Of Asistencia)

        Dim contador As Integer = 0
        Dim flag As Boolean = False
        Marcaciones = New List(Of Horario.Jornada)
        Dim marcac As Horario.Jornada = Nothing
        For Each rst As DataRow In AccesoDatos.AbrirTabla("select Fecha from Marcacion
where Personal_idPersonal = @1 and Fecha between @2 and @3 order by
Fecha", Nothing, Persona, Dia.Date, Dia.AddDays(1).Date).Rows

            flag = True
            If marcac Is Nothing Then
                marcac = New Horario.Jornada(rst!fecha, Nothing)

            Else
                marcac.Salida = rst!fecha
                Marcaciones.Add(marcac)
                marcac = Nothing
            End If
        Next

        If Jornadas.Count = 0 Or feriado Then
            Dim marcacion As New Asistencia(fecha, Persona)
            marcacion.TiempoExtra = New TimeSpan(0)
            For Each marca As Horario.Jornada In Marcaciones
                marcacion.TiempoExtra.Add(marca.Tiempo)
            Next
            Asistencia.Add(marcacion)
        Else
            For Each jornada As Horario.Jornada In Jornadas
                Dim marcacion As Asistencia
                Dim i As Integer
                marcacion = Nothing
                For l = i To Marcaciones.Count - 1
                    Dim j As Horario.Jornada
                    j = Marcaciones(l)
                    marcacion = CalcularFalta(jornada, j)
                    If marcacion IsNot Nothing Then

                        i = l + 1
                    Exit For
                Next
            Next
        End If
    End Sub

```

```

        End If
    Next
    If marcacion Is Nothing Then
        marcacion = New Asistencia(fecha, Persona)
        marcacion.HorarioEntrada = jornada.Entrada
        marcacion.HorarioSalida = jornada.Salida
        marcacion.TiempoTrabajado = New TimeSpan(0)
        marcacion.TiempoFalta = jornada.Salida - jornada.Entrada
    End If

    If marcacion.TiempoFalta.TotalMinutes > 0 Then
        CalcularJustificacion(marcacion)
    End If

    Asistencia.Add(marcacion)
Next
End If
End If
'If Not flag Then
' Dim x As New Horario.Jornada
' CalcularFalta(x)

'End If

End Sub

```

**Friend Function** CalcularFalta(Jornada As Horario.Jornada,  
marca As Horario.Jornada) As Asistencia

```

    If Not (marca.Salida.TimeOfDay <
Jornada.Entrada.TimeOfDay Or marca.Entrada.TimeOfDay >
Jornada.Salida.TimeOfDay) Then

```

```

        Dim s As New Asistencia(fecha, Persona)
        s.HorarioEntrada = Jornada.Entrada
        s.HorarioSalida = Jornada.Salida
        s.MarcaEntrada = marca.Entrada
        s.MarcaSalida = marca.Salida
        Dim entrada As TimeSpan, salida As TimeSpan
        If marca.Entrada.TimeOfDay > Jornada.Entrada.TimeOfDay Then
            entrada = marca.Entrada.TimeOfDay
        Else
            entrada = Jornada.Entrada.TimeOfDay
        End If

        If marca.Salida.TimeOfDay < Jornada.Salida.TimeOfDay Then
            salida = marca.Salida.TimeOfDay
        Else
            salida = Jornada.Salida.TimeOfDay

```

```

End If

s.TiempoTrabajado = salida - entrada
If Feriado Then
    s.TiempoFalta = New TimeSpan(0)
Else
    s.TiempoFalta = (Jornada.Salida -
Jornada.Entrada).Subtract(s.TiempoTrabajado)
    If s.TiempoFalta.Ticks < 0 Then s.TiempoFalta = New TimeSpan(0)
End If
Return s
End If

Return Nothing
End Function

Friend Sub CalcularJustificacion(ByRef asistencia As Asistencia)
    Dim justificadodesde As Date = Nothing
    Dim justificadohasta As Date = Nothing

    Dim desde As Date = New Date(Fecha.Year, Fecha.Month, Fecha.Day,
asistencia.HorarioEntrada.Hour, asistencia.HorarioEntrada.Minute, 0)
    Dim hasta As Date = New Date(Fecha.Year, Fecha.Month, Fecha.Day,
asistencia.HorarioSalida.Hour, asistencia.HorarioSalida.Minute, 0)

    With AccesoDatos.AbrirTabla("select Desde, Hasta from Permiso where
Personal_idPersonal= @1 and not (Hasta < @desde or Desde > @hasta)", Nothing,
Persona.Id, desde, hasta)
        If .Rows.Count = 0 Then
            Exit Sub
        Else
            If .Rows(0)!desde > desde Then
                justificadodesde = .Rows(0)!desde
            Else
                justificadodesde = desde
            End If

            If .Rows(0)!hasta < hasta Then
                justificadohasta = .Rows(0)!hasta
            Else
                justificadohasta = hasta
            End If

            asistencia.TiempoJustificado = justificadohasta - justificadodesde
            If asistencia.TiempoJustificado > asistencia.TiempoFalta Then
                asistencia.TiempoJustificado = asistencia.TiempoFalta
            End If
        End If
    End With
End Sub

```

```

ReadOnly Property TiempoTrabajado As TimeSpan
Get
    Dim aux As New TimeSpan(0)
    For Each x As Asistencia In Asistencia
        aux = aux.add(x.TiempoTrabajado)
    Next
    Return aux
End Get
End Property

```

```

ReadOnly Property TiempoFalta As TimeSpan
Get
    Dim aux As New TimeSpan(0)
    For Each x As Asistencia In Asistencia
        aux = aux.add(x.TiempoFalta)
    Next
    Return aux
End Get
End Property

```

```

ReadOnly Property TiempoJustificado As TimeSpan
Get
    Dim aux As New TimeSpan(0)
    For Each x As Asistencia In Asistencia
        aux = aux.add(x.TiempoJustificado)
    Next
    Return aux
End Get
End Property

```

```

ReadOnly Property TiempoExtra As TimeSpan
Get
    Dim aux As New TimeSpan(0)
    For Each x As Asistencia In Asistencia
        aux = aux.add(x.TiempoExtra)
    Next
    Return aux
End Get
End Property

```

```

ReadOnly Property TiempoLaborable As TimeSpan
Get
    Dim aux As New TimeSpan
    If Jornadas Is Nothing Then
        For Each asis As Asistencia In Asistencia
            aux = aux.Add(asis.TiempoLaborable)
        Next
    Else

```

```

        For Each x As Horario.Jornada In Jornadas
            aux = aux.Add(x.Tiempo)
        Next

    End If
    Return aux
End Get
End Property

Public Sub Guardar()
    For Each asis As Asistencia In Asistencia
        asis.Guardar()
    Next
End Sub
End Class

```

### Código fuente Clase Direccion

```

Public Class Direccion
    Inherits ClaseBase

    Friend pDepartamentos As List(Of Departamento)

    Friend Sub New(id As Integer, nombre As String, borrado As Boolean)
        MyBase.New(id, nombre, borrado)
    End Sub

    Friend Sub New(id As Integer)
        MyBase.New(id, "", False)
        Dim tabla As DataTable
        tabla = AccesoDatos.AbrirTabla("select idDireccion, Descripcion, Borrado from
Direccion where idDireccion = @1", Nothing, id)
        For Each rst In tabla.Rows
            pid = rst!iddireccion
            Nombre = rst!descripcion
            pborrado = rst!borrado
        Next
    End Sub
    Public Overrides Function Utilizado() As Boolean
        Sesion.Conexion.Open()
        Try
            Return AccesoDatos.AbrirEscalar("select count(*) from personal inner join
departamento on departamento_iddepartamento = departamento.iddepartamento
inner join direccion on iddireccion = direccion_iddireccion where iddireccion =
@1", Nothing, Me)
        Catch

```

```

        Return True
    Finally
        Sesion.Conexion.Close()
    End Try
End Function
Public Overrides Sub Guardar()
    Dim trans As MySqlConnection = Nothing
    Sesion.Conexion.Open()
    trans = Sesion.Conexion.BeginTransaction
    If Nuevo Then

        Try
            AccesoDatos.RunSql("INSERT INTO Direccion (Descripcion) VALUES (@1);",
trans, Nombre)
            pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)

            'If pDepartamentos IsNot Nothing Then
            ' For Each x As Departamento In Me.Departamentos
            '     x.pdireccion = Me
            '     x.Guardar(trans)
            ' Next
            'End If
        Catch ex As Exception
            If trans IsNot Nothing Then trans.Rollback()
            Sesion.Conexion.Close()

            Throw
        End Try
    Else
        Try
            AccesoDatos.RunSql("update Direccion set Borrado = 0, Descripcion = @1
where idDireccion = @2", trans, pnombre, pid)
            'AccesoDatos.RunSql("delete from Departamento where Direccion_idDireccion
= @1", trans, pid)
            'For Each x As Departamento In Me.Departamentos

            '     x.pdireccion = Me
            '     x.Guardar(trans)
            'Next
        Catch ex As Exception
            If trans IsNot Nothing Then trans.Rollback()
            Sesion.Conexion.Close()
            Throw
        End Try
    End If

    trans.Commit()
    Sesion.Conexion.Close()
End Sub

```

```

Public Overrides Sub Borrar()
    If Nuevo Then Exit Sub
    Sesion.Conexion.Open()
    AccesoDatos.RunSql("delete from Departamento where Direccion_idDireccion =
@2", Nothing, pid)
    AccesoDatos.RunSql("delete from Direccion where idDireccion = @2", Nothing, pid)
    Sesion.Conexion.Close()
    pborrado = True
End Sub

#Region "Propiedades"
ReadOnly Property Departamentos As List(Of Departamento)
    Get
        If pDepartamentos Is Nothing Then
            pDepartamentos = (New List(Of Departamento))
            If Not Nuevo Then
                For Each rst As DataRow In AccesoDatos.AbrirTabla("select idDepartamento,
Descripcion from Departamento where Direccion_idDireccion = @1 and borrado = 0
order by Descripcion", Nothing, Me).Rows
                    pDepartamentos.Add(New Departamento(rst!idDepartamento, Me,
rst!Descripcion, False))
                Next
            End If
        End If
    End Property

    Return pDepartamentos
End Get
End Property
#End Region
End Class

```

### Código Fuente Clase

```

Public Class Experiencia
    Inherits ClaseBase
    Implements ICloneable

    Property Desde As Date
    Property Hasta As Date
    Property Institucion As String

    Friend Sub New()
        MyBase.New(0, "", False)
    End Sub

```

```

    Friend Sub New(id As Integer, Nombre As String, institucion As String, desde As Date,
hasta As Date)
        MyBase.New(0, Nombre, False)
        Institucion = institucion
        Me.Desde = desde
        Me.Hasta = hasta
    End Sub

    Public Overrides Sub Borrar()

    End Sub

    Public Overrides Sub Guardar()

    End Sub
    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
    Public Function Clone() As Object Implements System.ICloneable.Clone
        Return Me.MemberwiseClone
    End Function
End Class

```

## Código Fuente Feriado

```

Public Class Feriado
    Property Fecha As Date
    Property Descripcion As String

    Friend Sub New(fecha As Date, descripcion As String)
        Me.Fecha = fecha
        Me.Descripcion = descripcion
    End Sub

    Friend Sub Guardar()
        Try
            Sesion.Conexion.Open()
            AccesoDatos.RunSql("INSERT INTO Feriado (Fecha, Descripcion) VALUES (@1,
@2);", Nothing, Fecha.Date, Descripcion)
        Catch ex As Exception
            Throw
        Finally
            Sesion.Conexion.Close()
        End Try
    End Sub

```

```

Friend Sub Actualizar(fechanatigua As Date)
    Try
        Sesion.Conexion.Open()
        AccesoDatos.RunSql("UPDATE Feriado SET Fecha = @FECHA, Descripcion =
@DESCRIPCION WHERE Fecha = @id", Nothing, Fecha.Date, Descripcion, fechanatigua)
    Catch ex As Exception
        Throw
    Finally
        Sesion.Conexion.Close()
    End Try
End Sub

Friend Sub Borrar()
    Try
        Sesion.Conexion.Open()
        AccesoDatos.RunSql("Delete from Feriado WHERE Fecha = @id", Nothing,
Fecha.Date)
    Catch ex As Exception
        Throw
    Finally
        Sesion.Conexion.Close()
    End Try
End Sub
End Class

```

### Código Fuente Formula

```

Public Class Formula
    Inherits ClaseBase

    Property Descripcion As String
    Property Formula As String = "0"
    Property Orden As Integer
    Property Activo As Boolean = True

    Public Function Validar() As Boolean
        Return False
    End Function

    Public Sub New()
        MyBase.New(0, 0, 0)
    End Sub

    Public Overrides Sub Borrar()
        If Nuevo Then Exit Sub
        Sesion.Conexion.Open()

```

```

Try
    Dim c As Integer = AccesoDatos.AbrirEscalar("select count(*) from RolDetalle
where Formula_idFormula = @1", Nothing, Id)
    If c = 0 Then
        AccesoDatos.RunSql("delete from Formula where idFormula = @1", Nothing,
Id)
    Else
        Throw New Exception("no se puede borrar")
    End If
Catch ex As Exception
    Throw
Finally
    Sesion.Conexion.Close()
End Try
End Sub
Public Overrides Function Utilizado() As Boolean
    Return False
End Function
Public Overrides Sub Guardar()
    Dim trans As MySQLTransaction = Nothing
    Sesion.Conexion.Open()
    trans = Sesion.Conexion.BeginTransaction
    If Nuevo Then

        Try
            AccesoDatos.RunSql("INSERT INTO Formula (Nombre, Formula, Descripcion,
Orden) VALUES (@1, @2, @3, @4);", trans, Nombre, Formula, Descripcion, Orden)
            pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)

        Catch ex As Exception
            If trans IsNot Nothing Then trans.Rollback()
            Sesion.Conexion.Close()

            Throw
        End Try
    Else
        Try
            AccesoDatos.RunSql("update Formula set Nombre= @1, Formula = @2,
Descripcion= @3, activo = @4, Orden = @5 where idFormula = @6;", trans, Nombre,
Formula, Descripcion, Activo, Orden, Id)
        Catch ex As Exception
            If trans IsNot Nothing Then trans.Rollback()
            Sesion.Conexion.Close()
            Throw
        End Try
    End If

    trans.Commit()
    Sesion.Conexion.Close()

```

```
End Sub
End Class
```

## Código fuente Horario

```
Public Class Horario
```

```
Friend Sub New()
```

```
End Sub
```

```
Enum DiaSemana
```

```
nada
```

```
Domingo
```

```
Lunes
```

```
Martes
```

```
Miercoles
```

```
Jueves
```

```
Viernes
```

```
End Enum
```

```
Private jornadas(7) As Dia
```

```
Public Class Dia
```

```
Property Jornadas As List(Of Jornada)
```

```
End Class
```

```
Public Class Jornada
```

```
Property Entrada As Date
```

```
Property Salida As Date
```

```
ReadOnly Property Tiempo As TimeSpan
```

```
Get
```

```
Return Salida - Entrada
```

```
End Get
```

```
End Property
```

```
Friend Sub New(desde As Date, hasta As Date)
```

```
Entrada = desde
```

```
Salida = hasta
```

```
End Sub
```

```
End Class
```

```
ReadOnly Property Semana(dia As DiaSemana) As Dia
```

```
Get
```

```

    If jornadas(dia) Is Nothing Then
        jornadas(dia) = New Dia
        jornadas(dia).Jornadas = New List(Of Jornada)
    End If
    Return jornadas(dia)
End Get

End Property

Public Sub Grabar(Persona As Personal)
    If Persona.Nuevo Then Exit Sub
    Sesion.Conexion.Open()
    Dim trasn As MySqlConnection = Sesion.Conexion.BeginTransaction
    Try
        AccesoDatos.RunSql("delete from Horario where Personal_idPersonal = @1",
        trasn, Persona.Id)
        For Dia As Integer = 1 To 7
            For Each h As Horario.Jornada In Semana(Dia).Jornadas
                AccesoDatos.RunSql("INSERT INTO Horario (Desde, Hasta, DiaSemana,
                Personal_idPersonal) VALUES (@1, @2, @3, @4)", trasn, h.Entrada, h.Salida, Dia,
                Persona.Id)
            Next
        Next
        trasn.Commit()
    Catch
        trasn.Rollback()
    Throw
    Finally
        Sesion.Conexion.Close()
    End Try
End Sub

End Class

```

### **Código Fuente Institucion Educativa**

```

Public Class Institucion
    Inherits ClaseBase

    Property Pais As Pais
    Property Tipo As Integer

    Enum Tipoinstitucion
        Escuela
        Colegio
        Universidad
    End Enum

```

```

    Instituto
End Enum

Friend Sub New(id As Integer)
    MyBase.New(id, "", False)
    Dim tabla As DataTable
    tabla = AccesoDatos.AbrirTabla("select idInstitucionEducativa, Nombre, Tipo,
Pais_idPais from InstitucionEducativa where idInstitucionEducativa = @1", Nothing, id)
    For Each rst In tabla.Rows
        Pais = New Pais(rst!Pais_idPais)
        pid = rst!idInstitucionEducativa
        Nombre = rst!Nombre
        Tipo = rst!tipo
    Next
End Sub

Friend Sub New(id As Integer, nombre As String, Pais As Pais, tipo As TipoInstitucion)
    MyBase.New(id, nombre, 0)
    Me.Pais = Pais
    Me.Tipo = tipo
End Sub

Public Sub New(id As Integer, nombre As String)
    MyBase.New(id, nombre, 0)
    Me.Pais = New Pais(1)
    Me.Tipo = 2
End Sub
Public Overrides Function Utilizado() As Boolean
    Return False
End Function
Public Overrides Sub Borrar()

End Sub

Public Overrides Sub Guardar()
    Dim trans As MySqlTransaction = Nothing
    Sesion.Conexion.Open()
    trans = Sesion.Conexion.BeginTransaction
    If Nuevo Then

        Try
            AccesoDatos.RunSql("INSERT INTO InstitucionEducativa (Nombre, Pais_idPais,
Tipo) VALUES (@1,@2,@3);", trans, Nombre, Pais, Tipo)
            pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)

            'If pDepartamentos IsNot Nothing Then
            ' For Each x As Departamento In Me.Departamentos
            '     x.pdireccion = Me
            '     x.Guardar(trans)
            ' Next
        End Try
    End If
End Sub

```

```

        'End If
    Catch ex As Exception
        If trans IsNot Nothing Then trans.Rollback()
        Sesion.Conexion.Close()

        Throw
    End Try
Else
    Try
        AccesosDatos.RunSql("update InstitucionEducativa set Nombre = @1,
Pais_idPais = @2, Tipo = @x where idInstitucionEducativa = @3", trans, pnombre, Pais,
Tipo, pid)
    Catch ex As Exception
        If trans IsNot Nothing Then trans.Rollback()
        Sesion.Conexion.Close()
        Throw
    End Try
End If

trans.Commit()
Sesion.Conexion.Close()
End Sub
End Class

```

## Código Fuente Libro

```

Public Class Libro
    Inherits ClaseBase
    Implements ICloneable

    Property Año As Integer
    Friend Sub New()
        MyBase.New(0, "", False)
    End Sub
    Public Overrides Sub Borrar()

    End Sub

    Public Overrides Sub Guardar()

    End Sub
    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
    Public Function Clone() As Object Implements System.ICloneable.Clone
        Return Me.MemberwiseClone
    End Function

```

```
End Function
End Class
```

## Código Fuente Pais

```
Public Class Pais
    Inherits ClaseBase
    Public Sub New(id As Integer, nombre As String)
        MyBase.New(id, nombre, 0)
    End Sub

    Friend Sub New(id As Integer)
        MyBase.New(id, "", False)
        Dim tabla As DataTable
        tabla = AccesoDatos.AbrirTabla("select idPais, Nombre from Pais where idPais =
@1", Nothing, id)
        For Each rst In tabla.Rows
            pid = rst!idpais
            Nombre = rst!Nombre
        Next
    End Sub
    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
    Public Overrides Sub Borrar()

End Sub

    Public Overrides Sub Guardar()
        Dim trans As MySqlTransaction = Nothing
        Sesion.Conexion.Open()
        trans = Sesion.Conexion.BeginTransaction
        If Nuevo Then

            Try
                AccesoDatos.RunSql("INSERT INTO Pais (Nombre) VALUES (@1);", trans,
Nombre)
                pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
            Catch ex As Exception
                If trans IsNot Nothing Then trans.Rollback()
                Sesion.Conexion.Close()

            Throw
        End Try
    Else
        Try
```

```
AccesoDatos.RunSql("update Pais set Nombre = @1 where idPais = @2", trans,  
pnombre, pid)
```

```
    Catch ex As Exception  
        If trans IsNot Nothing Then trans.Rollback()  
        Sesion.Conexion.Close()  
        Throw  
    End Try  
End If  
  
trans.Commit()  
Sesion.Conexion.Close()  
End Sub  
End Class
```

### Código Fuente Permiso

```
Public Class Permiso  
    Inherits ClaseBase  
  
    Enum Motivo  
        CertificadoMedicoParticular  
        CertificadoMedicoIESS  
        PermisoSinCargo  
    End Enum  
    Property Persona As Personal  
    Property Desde As Date  
    Property Hasta As Date  
    Property Tipo As Motivo  
    'Property Descripcion As String  
  
    Public Sub New()  
        MyBase.New(0, 0, 0)  
    End Sub  
    Public Overrides Function Utilizado() As Boolean  
        Return False  
    End Function  
    Public Overrides Sub Borrar()  
  
        Sesion.Conexion.Open()  
        AccesoDatos.RunSql("delete from Permiso where idPermiso = @1", Nothing, Id)  
        Sesion.Conexion.Close()  
    End Sub  
  
    Public Overrides Sub Guardar()  
        Dim trans As MySqlConnection = Nothing  
        Sesion.Conexion.Open()
```

```

trans = Sesion.Conexion.BeginTransaction
If Nuevo Then

    Try
        AccesoDatos.RunSql("insert Into Permiso ( Desde, Hasta, Tipo,
Personal_idPersonal) values (@Desde, @Hasta, @Tipo, @Personal_idPersonal)", trans,
Desde, Hasta, Tipo, Persona)
        pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)

        'If pDepartamentos IsNot Nothing Then
        ' For Each x As Departamento In Me.Departamentos
        '     x.pdireccion = Me
        '     x.Guardar(trans)
        ' Next
        'End If
    Catch ex As Exception
        If trans IsNot Nothing Then trans.Rollback()
        Sesion.Conexion.Close()

        Throw
    End Try
Else
    Try
        AccesoDatos.RunSql("update Permiso set Desde = @1, Hasta = @2, Tipo = @3
where idPermiso = @4", trans, Desde, Hasta, Tipo, Id)
    Catch ex As Exception
        If trans IsNot Nothing Then trans.Rollback()
        Sesion.Conexion.Close()
        Throw
    End Try
End If

trans.Commit()
Sesion.Conexion.Close()
End Sub
End Class

```

### Código Fuente Personal Titulo

```

Public Class PersonalTitulo
    Inherits ClaseBase
    Implements ICloneable

    Property Titulo As Titulo
    Property Institucion As Institucion
    Property Año As Integer

```

```

Property Registro As String = ""
Property EnArea As Boolean = False

Friend Sub New()
    MyBase.New(0, "", False)
End Sub
Public Overrides Sub Borrar()

End Sub

Public Overrides Sub Guardar()

End Sub
Public Overrides Function Utilizado() As Boolean
    Return False
End Function
Public Function Clone() As Object Implements System.ICloneable.Clone
    Return Me.MemberwiseClone
End Function
End Class

```

### **Código Fuente Reconocimiento**

```

Public Class Reconocimiento
    Inherits ClaseBase
    Implements ICloneable

    Friend Sub New()
        MyBase.New(0, "", False)
    End Sub
    Property Fecha As Date

    Public Overrides Sub Borrar()

    End Sub

    Public Overrides Sub Guardar()

    End Sub
    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
    Public Function Clone() As Object Implements System.ICloneable.Clone
        Return Me.MemberwiseClone
    End Function
End Class

```

## Código Fuente Rol Cabecera

```
Public Class RolCabecera
    Inherits ClaseBase
    Property Persona As Personal
    Property Año As Integer
    Property Mes As Integer
    Property FechaEmision As Date
    Property TipoContrato As Contrato.TipoContrato
    Public Sub New()
        MyBase.New(0, 0, 0)
    End Sub

    Public Overrides Sub Borrar()

    End Sub
    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
    Public Overrides Sub Guardar()
        Dim trans As MySqlConnection = Nothing
        Sesion.Conexion.Open()
        trans = Sesion.Conexion.BeginTransaction
        If Nuevo Then

            Try
                AccesosDatos.RunSql("INSERT INTO RolCabecera (Año, Mes, FechaEmision,
Personal_idPersonal, TipoContrato) VALUES (@1, @2, @3, @4,@5);", trans, Año, Mes,
Now, Persona, TipoContrato)
                pid = AccesosDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
                FechaEmision = Now

            Catch ex As Exception
                If trans IsNot Nothing Then trans.Rollback()
                Sesion.Conexion.Close()

            Throw
        End Try
    Else
        'Try
        ' AccesosDatos.RunSql("update Formula set Nombre= @1, Formula = @2,
Descripcion= @3, activo = @4, Orden = @5 where idFormula = @6;", trans, Nombre,
Formula, Descripcion, Activo, Orden, Id)
        'Catch ex As Exception
        ' If trans IsNot Nothing Then trans.Rollback()
        ' Sesion.Conexion.Close()
        ' Throw
        'End Try
    End Sub
End Class
```

```

End If

trans.Commit()
Sesion.Conexion.Close()

End Sub
End Class

```

### Código Fuente Clase Titulo

```

Public Class Titulo
    Inherits ClaseBase

    Property Nivel As Integer = 1

    Friend Sub New(id As Integer, nombre As String, Nivel As Integer, borrado As Boolean)
        MyBase.New(id, nombre, Borrado)
        Me.Nivel = Nivel
    End Sub

    Friend Sub New(id As Integer)
        MyBase.New(0, "", 0)
        Dim tabla As DataTable
        tabla = AccesoDatos.AbrirTabla("select idTitulo, Nombre, Nivel from Titulo where
idTitulo = @1", Nothing, id)
        For Each rst In tabla.Rows
            pid = rst!idtitulo
            Nombre = rst!Nombre
            Nivel = rst!nivel
        Next
    End Sub
    Public Overrides Sub Borrar()

    End Sub
    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
    Public Overrides Sub Guardar()
        Dim trans As MySQLTransaction = Nothing
        Sesion.Conexion.Open()
        trans = Sesion.Conexion.BeginTransaction
        If Nuevo Then

            Try
                AccesoDatos.RunSql("INSERT INTO Titulo (Nombre, Nivel) VALUES (@1, @2);",
trans, Nombre, Nivel)

```

```

        pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", trans)
    Catch ex As Exception
        If trans IsNot Nothing Then trans.Rollback()
        Sesion.Conexion.Close()

        Throw
    End Try
Else
    Try
        AccesoDatos.RunSql("update Titulo set Nombre = @1, Nivel = @2 where
idTitulo = @3", trans, pnombre, Nivel, pid)
    Catch ex As Exception
        If trans IsNot Nothing Then trans.Rollback()
        Sesion.Conexion.Close()
        Throw
    End Try
End If

    trans.Commit()
    Sesion.Conexion.Close()
End Sub
End Class

```

### Código Fuente Clase UsoVacaciones

```

Public Class UsoVacacion
    Inherits ClaseBase

    Property Desde As Date = Now
    Property Hasta As Date = Now
    Property Vacacion As Vacacion
    Friend Sub New(cabecera As Vacacion)
        MyBase.New(0, 0, 0)
        Me.Vacacion = cabecera
    End Sub

    Public Overrides Function Utilizado() As Boolean
        Return False
    End Function
    Public Overrides Sub Borrar()
        If Nuevo Then Exit Sub
        Sesion.Conexion.Open()
        AccesoDatos.RunSql("delete from VacacionDetalle where idVacacionDetalle =
@1", Nothing, Id)
        Sesion.Conexion.Close()
    End Sub

```

```

Public Overrides Sub Guardar()
    Sesion.Conexion.Open()
    If Nuevo Then

        AccesoDatos.RunSql("insert into VacacionDetalle(Desde, Hasta,
VacacionCabecera_idVacacionCabecera) values (@1,@2,@3)", Nothing, Desde, Hasta,
Vacacion)
        pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID() ", Nothing)
    Else
        AccesoDatos.RunSql("update VacacionDetalle set Desde = @1, Hasta = @2 where
idVacacionDetalle = @3", Nothing, Desde, Hasta, Id)
    End If
    Sesion.Conexion.Close()
End Sub

Function Dias() As Integer
    Return DateDiff(DateInterval.Day, Desde, Hasta)
End Function
End Class

```

### Código Fuente Clase Vacaciones

```

Public Class Vacacion
    Inherits ClaseBase

    Property Persona As Personal
    Property Año As Integer
    Property Dias As Integer

    Public Sub New()
        MyBase.New(0, 0, 0)
    End Sub

    Public Overrides Sub Borrar()
        Sesion.Conexion.Open()
        If Nuevo Then

            Else
                AccesoDatos.RunSql("delete from VacacionDetalle where
VacacionCabecera_idVacacionCabecera = @3", Nothing, Id)
                AccesoDatos.RunSql("delete from VacacionCabecera where
idVacacionCabecera = @3", Nothing, Id)
            End If
        Sesion.Conexion.Close()
    End Sub

```

```

Public Overrides Sub Guardar()
    Sesion.Conexion.Open()
    If Nuevo Then
        AccesoDatos.RunSql("insert into VacacionCabecera (Personal_idPersonal,
Año, Dias) values(@1,@2,@3)", Nothing, Persona, Año, Dias)
        pid = AccesoDatos.AbrirEscalar("SELECT LAST_INSERT_ID()
", Nothing)
    Else
        AccesoDatos.RunSql("update VacacionCabecera set Año = @1, Dias =
@2 where idVacacionCabecera = @3", Nothing, Año, Dias, Id)
    End If
    Sesion.Conexion.Close()
End Sub

```

```

ReadOnly Property Usos As List(Of UsoVacacion)
    Get
        Dim aux As New List(Of UsoVacacion)
        For Each rst As DataRow In AccesoDatos.AbrirTabla("select
idVacacionDetalle,desde, hasta from VacacionDetalle where
VacacionCabecera_idVacacionCabecera = @1", Nothing, Me).Rows
            Dim x As New UsoVacacion(Me)
            x.pid = rst!idVacacionDetalle
            x.Desde = rst!desde
            x.Hasta = rst!hasta
            aux.Add(x)
        Next
        Return aux
    End Get
End Property

```

```

Function Saldo() As Integer
    If Nuevo Then Return Dias
    Dim usados As Integer
    For Each uso As UsoVacacion In Usos
        usados += uso.Dias
    Next
    Return Dias - usados
End Function

```

```

Public Overrides Function Utilizado() As Boolean
    Return False
End Function
End Class

```

## Instalación de Linux Centos

La instalación se realiza desde el CD/DVD del ISO de Linux CentOS, seleccionados la opción “Install CentOS 7” y presionaremos el botón INTRO:



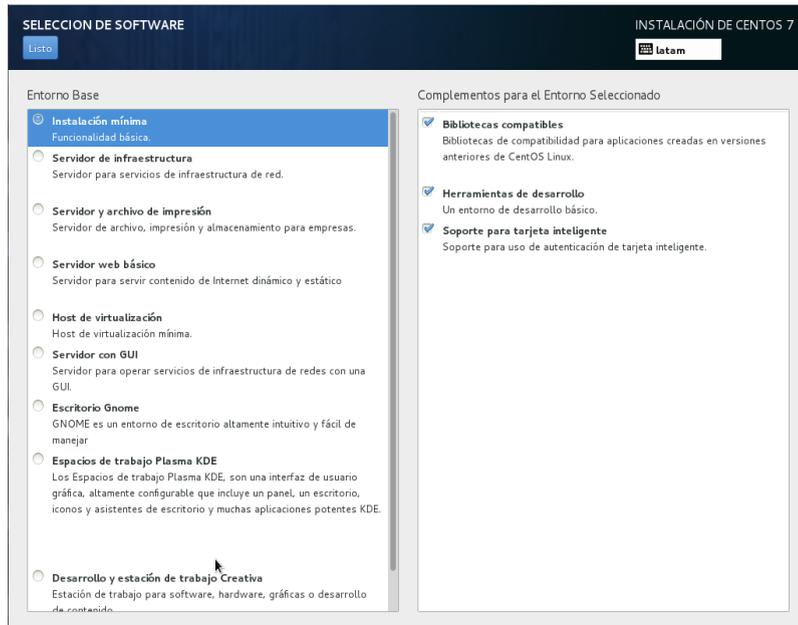
Seguido de esto se podrá observar la interfaz gráfica en donde se tendrá que seleccionar el idioma con el cual querramos que se instale el CentOS.



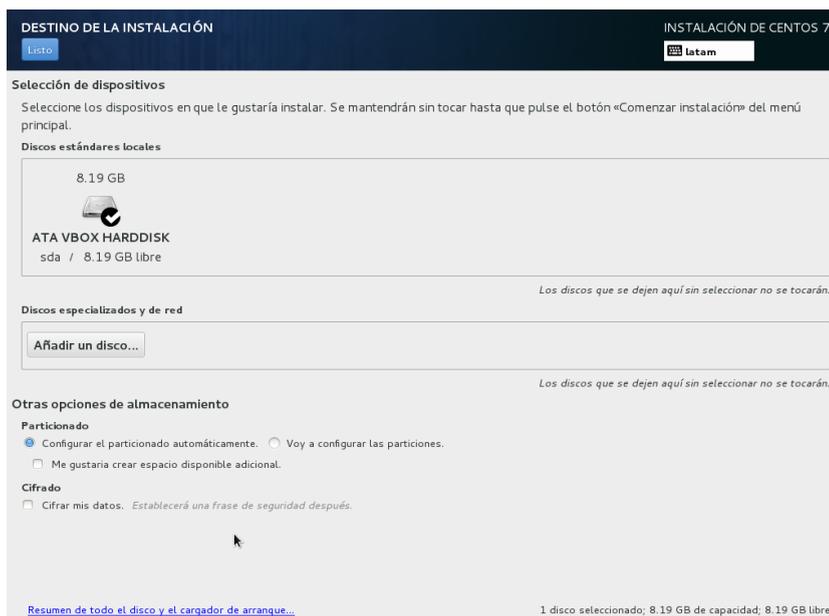
Luego de seleccionar el idioma se dará clic en **Continuar** y se nos mostrará una ventana donde podremos configurar algunas opciones que trae el CentOS, es importante recalcar que solo se configurará las opciones necesarias.



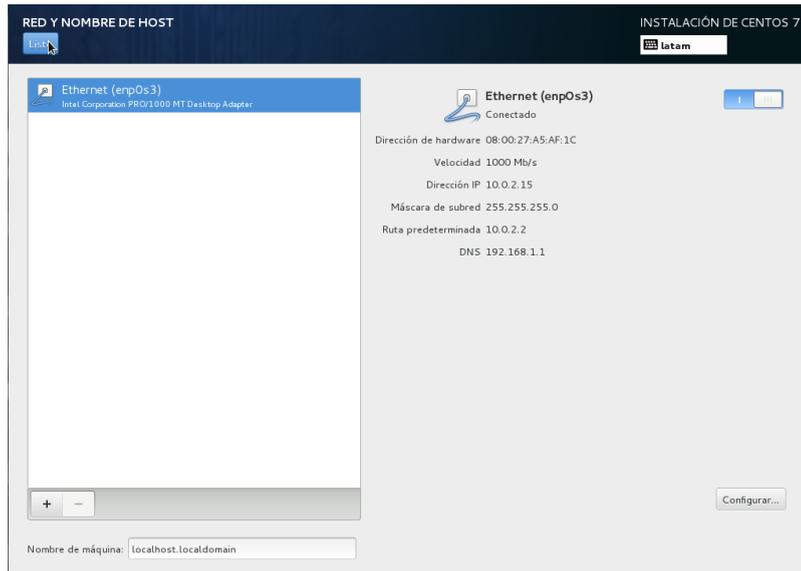
Selección de software es donde elegiremos las opciones que deseamos instalar para el CentOS.



En destino de la instalación es donde se indica el lugar donde será instalado el centos también se seccionará el tipo de particiones que necesite el servidor y los tipos de archivos. Por lo general ya viene con un espacio seleccionado.

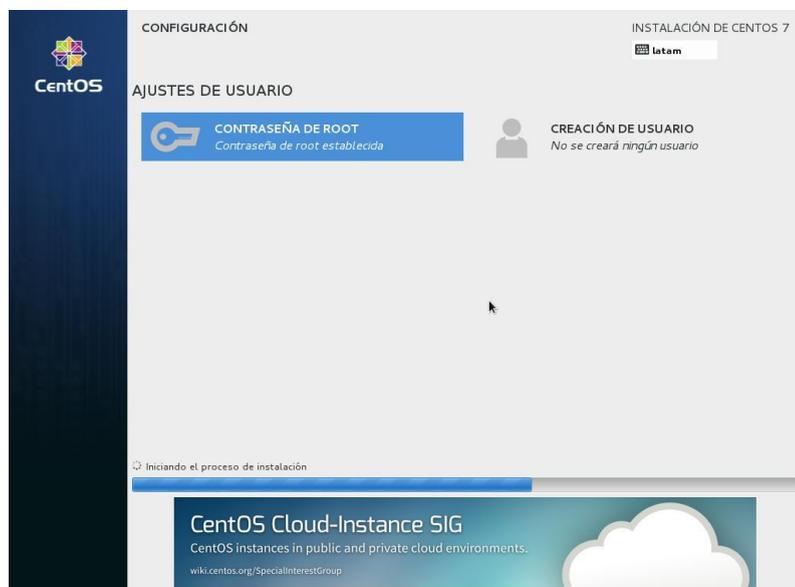


En Red y nombre se habilitara el Ethernet el cual nos dara las direcciones para poder navegar en internet desde el sistema operativo centos. Si se desea configurar las direcciones solo se da clic en configurar.



Una vez que se hayan realizado las configuraciones estará listo para ser instalado. Procederemos a dar clic en **Listo** y luego en **Comenzar instalación**.

Durante la instalación se debe guardar la contraseña root.



CONTRASEÑA ROOT

INSTALACIÓN DE CENTOS 7

Listo

La cuenta root se usa para administrar el sistema. Introduzca una contraseña para el usuario root.

Contraseña de root: [password field] Débil

Confirmar: [password field]

Luego ingresar un usuario y con su respectiva contraseña

CREAR USUARIO

INSTALACIÓN DE CENTOS 7

Listo

Nombre completo: Geovanny

Usuario: geovanny

Consejo: Mantenga su nombre de usuario con menos de 32 caracteres y no utilice espacios.

Hacer que este usuario sea administrador

Se requiere una contraseña para usar esta cuenta

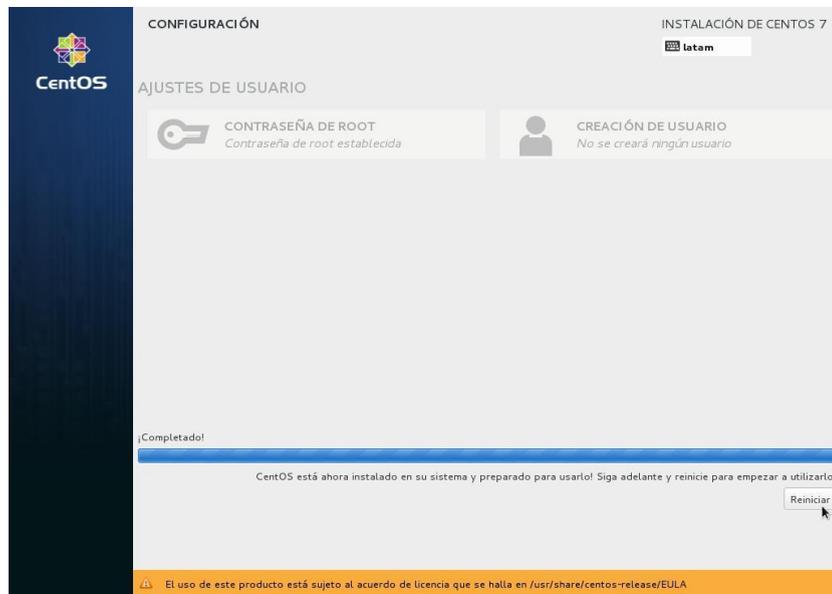
Contraseña: [password field] Débil

Confirmar contraseña: [password field]

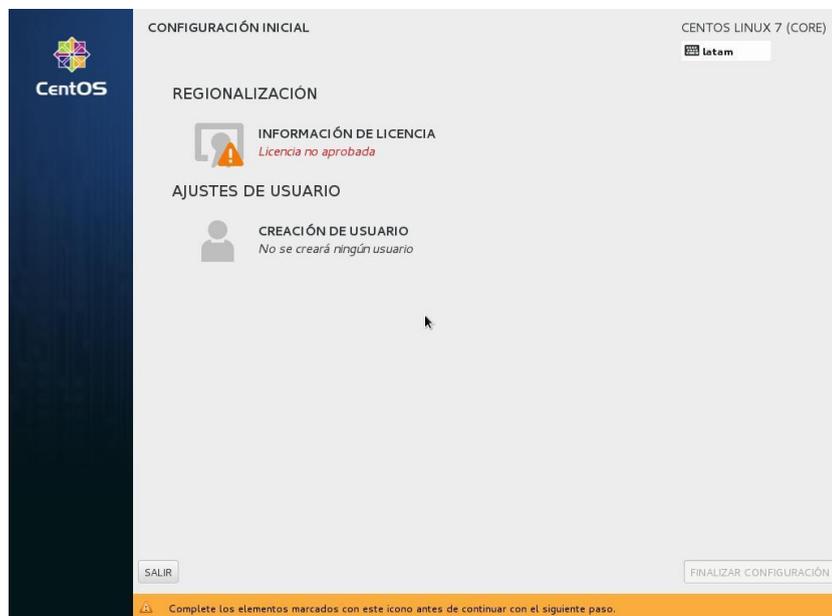
Avanzado...

Realizado esto daremos clic en **Listo** y esperaremos hasta que se haya cumplido correctamente la instalación.

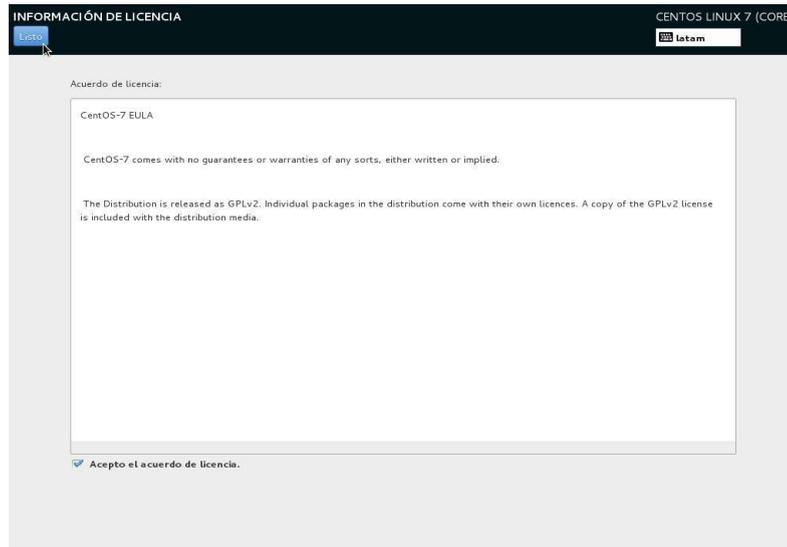
Después de la instalación, clic en **Reiniciar**.



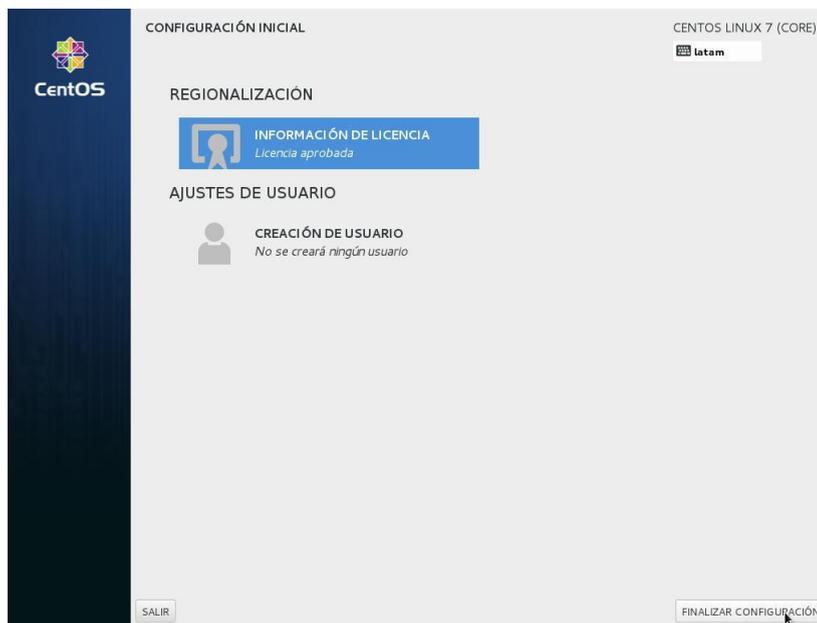
Cuando ya se nos reinicie se los abrirá la siguiente ventana donde se dará clic en **Información de Licencia**.



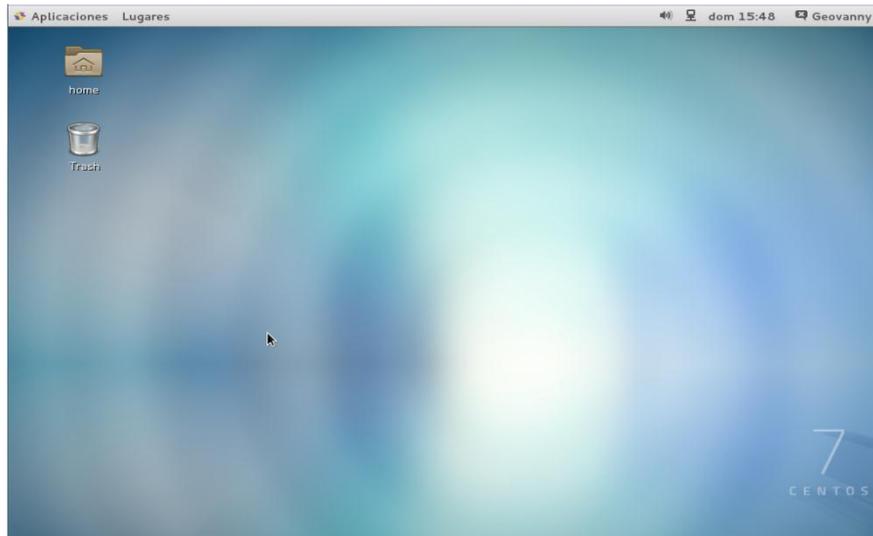
Luego se seleccionará la opción acepto los términos de licencia.



Daremos clic en listo y luego en **Finalizar Configuración**.



Una vez Hecho esto esperamos un momento ingresamos con nuestra contraseña y listo estaremos en el escritorio de Linux CentOS.



Ahora procederemos a abrir la terminal del centos e instalar las herramientas necesarias para que nos sirva de servidor.



Ahora ingresamos como usuario root con el comando “su root” para que se nos permita realizar instalaciones.

```
geovanny@localhost:/home/geovanny
Archivo Editar Ver Buscar Terminal Ayuda
[geovanny@localhost ~]$ su root
Contraseña:
[root@localhost geovanny]#
```

Se necesita instalar el repositorio MySQL para ello necesitamos el **wget** que no viene instalado, pero se lo puede instalar con el siguiente comando: *yum install wget*.

```
geovanny@localhost:/home/geovanny
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost geovanny]# yum install wget
Complementos cargados:fastestmirror, langpacks
base | 3.6 kB | 00:00
extras | 3.4 kB | 00:00
updates | 3.4 kB | 00:00
Loading mirror speeds from cached hostfile
* base: mirror.uta.edu.ec
* extras: mirror.uta.edu.ec
* updates: mirror.uta.edu.ec
Resolviendo dependencias
--> Ejecutando prueba de transacción
---> Paquete wget.x86_64 0:1.14-10.el7 debe ser actualizado
---> Paquete wget.x86_64 0:1.14-10.el7_0.1 debe ser una actualización
--> Resolución de dependencias finalizada

Dependencias resueltas

=====
Package      Arquitectura  Versión      Repositorio  Tamaño
=====
Actualizando:
wget       x86_64        1.14-10.el7_0.1  base         545 k
Resumen de la transacción
```

Una vez instalado actualizaremos el paquete escribimos **y** para confirmar que lo queremos actualizar y luego presionamos enter. Lo tendremos instalado y actualizado.

```

geovanny@localhost:/home/geovanny
Archivo Editar Ver Buscar Terminal Ayuda
advertencia:/var/cache/yum/x86_64/7/base/packages/wget-1.14-10.el7_0.1.x86_64.rpm
m: EncabezadoV3 RSA/SHA256 Signature, ID de clave f4a80eb5: NOKEY
Obteniendo clave desde file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importando llave GPG 0xF4A80EB5:
  Usuarioid : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
  Huella      : 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
  Paquete    : centos-release-7-0.1406.el7.centos.2.3.x86_64 (@anaconda)
  Desde      : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Está de acuerdo [s/N]:s
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Actualizando : wget-1.14-10.el7_0.1.x86_64 1/2
  Limpieza     : wget-1.14-10.el7.x86_64 2/2
  Comprobando  : wget-1.14-10.el7_0.1.x86_64 1/2
  Comprobando  : wget-1.14-10.el7.x86_64 2/2

Actualizado:
  wget.x86_64 0:1.14-10.el7_0.1

¡Listo!
[root@localhost geovanny]#

```

Ahora Descargaremos el paquete RPM del repositorio MySQL server con el siguiente comando: `wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm`

```

geovanny@localhost:/home/geovanny
Archivo Editar Ver Buscar Terminal Ayuda
wget.x86_64 0:1.14-10.el7_0.1

¡Listo!
[root@localhost geovanny]# wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
--2015-08-10 10:56:32-- http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
Resolviendo repo.mysql.com (repo.mysql.com)... 104.73.125.144
Conectando con repo.mysql.com (repo.mysql.com)[104.73.125.144]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 6140 (6,0K) [application/x-redhat-package-manager]
Grabando a: "mysql-community-release-el7-5.noarch.rpm"

100%[=====] 6.140 ---K/s en 0,02s

2015-08-10 10:56:33 (249 KB/s) - "mysql-community-release-el7-5.noarch.rpm" guardado [6140/6140]

```

Instalaremos el paquete rpm del repositorio de MySQL Server con el siguiente comando: `rpm -ivh mysql-community-release-el7-5.noarch.rpm`

```

[root@localhost geovanny]# rpm -ivh mysql-community-release-el7-5.noarch.rpm
Preparando... ##### [100%]
Actualizando / instalando...
  1:mysql-community-release-el7-5 ##### [100%]
[root@localhost geovanny]#

```

Para verificar si se han instalado los repositorios de MySQL pondremos el siguiente comando: `ls -l /etc/yum.repos.d/mysql-community*` el cual nos devolverá lo siguiente:

```
/etc/yum.repos.d/mysql-community.repo
```

```
/etc/yum.repos.d/mysql-community-source.repo
```

```
[root@localhost geovanny]# ls -l /etc/yum.repos.d/mysql-community*  
/etc/yum.repos.d/mysql-community.repo  
/etc/yum.repos.d/mysql-community-source.repo
```

Realizado esto ya podremos instalar el Mysql server con el este comando: `yum install mysql-server`

```
Installing:  
mysql                i686      5.1.67-1.el6_3      updates      896 k  
mysql-server         i686      5.1.67-1.el6_3      updates      8.8 M  
Installing for dependencies:  
mysql-libs           i686      5.1.67-1.el6_3      updates      1.2 M  
perl-DBD-MySQL       i686      4.013-3.el6         base         134 k  
  
Transaction Summary  
-----  
Install      4 Package(s)  
  
Total download size: 11 M  
Installed size: 31 M  
Is this ok [y/N]: y  
Downloading Packages:  
(1/4): mysql-5.1.67-1.el6_3.i686.rpm           | 896 kB   00:00  
(2/4): mysql-libs-5.1.67-1.el6_3.i686.rpm      | 1.2 MB   00:00  
(3/4): mysql-server-5.1.67-1.el6_3.i686.rpm   | 8.8 MB   00:05  
(4/4): perl-DBD-MySQL-4.013-3.el6.i686.rpm    | 134 kB   00:00  
-----  
Total                                           1.2 MB/s | 11 MB   00:09  
Running rpm_check_debug  
Running Transaction Test  
■
```

Nos pedirá una confirmación para añadir la llave GPG y escribiremos `s`

```
-----  
Total  
Obteniendo clave desde file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql  
Importando llave GPG 0x5072E1F5:  
  Usuarioid   : "MySQL Release Engineering <mysql-build@oss.oracle.com>"  
  Huella      : a4a9 4068 76fc bd3c 4567 70c8 8c71 8d3b 5072 e1f5  
  Paquete     : mysql-community-release-el7-5.noarch (installed)  
  Desde       : file:/etc/pki/rpm-gpg/RPM-GPG-KEY-mysql  
Está de acuerdo [s/N]: ■
```

Una vez instalado ya podremos iniciar el servicio con el comando: *systemctl start mysqld*.

Después que se haya realizado la instalación es conveniente que se configure la seguridad estableciendo una contraseña para el usuario root del MySQL

```
[root@localhost geovanny]# mysql_secure_installation
```

Se nos expondrá una asistente para realizar todos los cambios respectivos y configuraciones debidas

---

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current  
password for the root user. If you've just installed MySQL, and  
you haven't set the root password yet, the password will be blank,  
so you should just press enter here.

Enter current password for root (enter for none):  
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL  
root user without the proper authorisation.

Set root password? [Y/n] y  
New password:  
Re-enter new password:  
Password updated successfully!  
Reloading privilege tables..  
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone  
to log into MySQL without having to have a user account created for  
them. This is intended only for testing, and to make the installation  
go a bit smoother. You should remove them before moving into a  
production environment.

Remove anonymous users? [Y/n] y  
... Success!

Normally, root should only be allowed to connect from 'localhost'. This  
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n  
... skipping.

By default, MySQL comes with a database named 'test' that anyone can  
access. This is also intended only for testing, and should be removed  
before moving into a production environment.

Remove test database and access to it? [Y/n] y  
- Dropping test database...  
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist  
... Failed! Not critical, keep moving...  
- Removing privileges on test database...  
... Success!

Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.

Reload privilege tables now? [Y/n] y  
... Success!

All done! If you've completed all of the above steps, your MySQL

En el caso de que se quiera el acceso desde fuera del equipo del servidor donde se encuentra la base se debe abrir el cortafuegos y puerto de la base con el siguiente comando: *firewall-cmd --permanent --zone=public --add-service=mysql*

Para aplicar los cambios ejecutaremos el comando: *systemctl restart firewalld.service*

Si se desea permitir acceso desde otros equipos que estén en la misma red al usuario root u otro usuario: ejecutaremos el siguiente comando: `mysql -u root -p`

Nos pedirá la contraseña de usuario root que ingresamos anteriormente en las configuraciones, una vez que entremos introduciremos el siguiente comando: `grant all on *.* to root@'%' identified by 'contraseña' ;`

Para que los cambios se realicen utilizaremos este comando: `flush privileges;`

También si se desea se puede instalar el webmin que se puede hacer desde la página oficial con el webmin el control de la base será de mejor manejo y más fácil.

