



INSTITUTO TECNOLÓGICO  
“CORDILLERA”

**CARRERA ANÁLISIS DE SISTEMAS**

**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE  
UN SISTEMA INTEGRADO DE GESTIÓN FINANCIERA: MÓDULO DE  
NÓMINA.**

**Proyecto de investigación previo a la obtención del título de Tecnólogo en  
Análisis de Sistemas**

Autor: Cajamarca Sacoto Julio César

Tutor: Lic. Wilson Núñez

Quito, Abril 2015

---

## Aprobación del tutor y lector

En mi calidad de tutor del trabajo sobre el tema: AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.”, presentado por el ciudadano: **Cajamarca Sacoto Julio César**, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Abril de 2015

Lcdo. Wilson Núñez

**TUTOR**

Ing. Hugo Heredia

**LECTOR**

### **Declaratoria**

Declaro que la investigación es absolutamente original, autentica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

---

Julio César Cajamarca Sacoto

CC 1712049905

---

## Contrato de cesión sobre derechos propiedad intelectual

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante **JULIO CÉSAR CAJAMARCA SACOTO**, por sus propios y personales derechos, a quien en lo posterior se le denominará el "CEDENTE"; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el "CESIONARIO". Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

**PRIMERA: ANTECEDENTE.- a)** El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado "**CONTROL Y SEGUIMIENTO DE ACTIVIDADES MEDIANTE UNA MESA DE SERVICIOS, APLICANDO METODOLOGÍA ITIL PARA LA EMPRESA PARTNERS GROUP**", el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. **b)** Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

**SEGUNDA: CESIÓN Y TRANSFERENCIA.-** Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

**TERCERA: OBLIGACIÓN DEL CEDENTE.-** El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

**CUARTA: CUANTIA.-** La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

**QUINTA: PLAZO.-** La vigencia del presente contrato es indefinida.

**SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.-** Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el

español; y, g) La reconversión, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

**SÉPTIMA: ACEPTACIÓN.-** Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los seis días del mes de Abril del dos mil quince.

f) \_\_\_\_\_

C.C. N° 1712049905

**CEDENTE**

f) \_\_\_\_\_

Instituto Superior Tecnológico Cordillera

**CESIONARIO**

### **Agradecimiento**

Es merecido expresar un profundo agradecimiento a las personas que de alguna forma aportaron y son parte de su culminación, en particular va dirigido a mis padres y esposa, quienes me han apoyado día tras día. A mis profesores que han impartido sus conocimientos y experiencias, para formarme como un profesional, al Lcdo. Wilson Núñez que fue mi tutor de proyecto, quien supo creer en mi capacidad y orientarme sin interés alguno para culminar con éxito esta investigación. Así como también al Ing. Hugo Heredia que como lector final me brindo el impulso final para su culminación.

### **Dedicatoria**

La realización de este trabajo está dedicado a Dios a mis Padres y Esposa, por brindarme su apoyo constante dándome la fortaleza para continuar sin dudar un solo momento en mi inteligencia y capacidad.

---

## Índice general

### Contenido

Portada.....	¡Error! Marcador no definido.
Aprobación del tutor y lector .....	ii
Declaratoria .....	iii
Contrato de cesión sobre derechos propiedad intelectual .....	iv
Agradecimiento .....	viii
Dedicatoria .....	ix
Índice general .....	x
Índice de tablas.....	xv
Índice de figuras.....	xvii
Índice de anexos .....	xix
Resumen ejecutivo .....	xx
Abstract.....	xxi
Introducción .....	xxii
Capítulo I: Antecedentes .....	1
1.01 Contexto.....	1
1.02 Justificación .....	2
1.03 Definición del Problema central .....	3
1.03.1 Matriz de Análisis de Fuerzas.....	4
Capítulo II: Análisis de Involucrados .....	7
2.01 Requerimientos .....	7

---

<b>2.01.1 Descripción del sistema actual.</b> .....	7
<i>Alcance</i> .....	7
<b>2.01.3 Entrevistas.</b> .....	8
<b>2.01.4 Matriz de Requerimientos</b> .....	8
<b>2.01.5 Descripción Detallada</b> .....	9
<b>2.02 Mapeo de Involucrados</b> .....	14
<b>2.03 Matriz de Involucrados</b> .....	15
<b>Capítulo III: Problemas y Objetivos</b> .....	16
<b>3.01 Árbol de Problemas</b> .....	16
<b>3.02 Árbol de Objetivos</b> .....	17
<b>3.03 Diagrama de casos de uso</b> .....	18
<b>3.04 Casos de uso de realización</b> .....	18
<b>Caso: UC001</b> .....	18
<b>Caso: UC002</b> .....	19
<b>Caso: UC003</b> .....	20
<b>3.05 Diagramas de secuencia</b> .....	21
<b>3.06 Especificaciones de casos de uso</b> .....	22
<b>Capítulo IV: Análisis de Alternativas</b> .....	24
<b>4.01. Matriz de Análisis de Alternativas</b> .....	24
<b>4.02. Matriz de Impactos de Objetivos</b> .....	24
<b>4.0.3 Estándares para el Diseño de Clases</b> .....	24
<b>Formato de Clases-Interfaz</b> .....	24

<b>Atributos .....</b>	<b>24</b>
<b>Operaciones .....</b>	<b>25</b>
<b>4.04. Diagrama de clases.....</b>	<b>25</b>
<b>4.05. Modelo Lógico – Físico .....</b>	<b>25</b>
<b>Modelo Lógico .....</b>	<b>25</b>
<b>Modelo Físico.....</b>	<b>25</b>
<b>4.06. Diagrama de Componentes .....</b>	<b>25</b>
<b>4.07. Diagramas de Estrategias.....</b>	<b>26</b>
<b>4.08. Matriz de Marco Lógico .....</b>	<b>27</b>
<b>4.09. Vistas arquitectónicas .....</b>	<b>28</b>
<b>4.09.01. Vista lógica.....</b>	<b>28</b>
<b>4.09.02. Vista física.....</b>	<b>28</b>
<b>4.09.03. Vista de desarrollo .....</b>	<b>29</b>
<b>4.09.04. Vista de procesos .....</b>	<b>29</b>
<b>Capítulo V: Propuesta .....</b>	<b>30</b>
<b>5.01. Especificación de estándares de programación.....</b>	<b>30</b>
<b>Declaraciones de Variables .....</b>	<b>30</b>
<b>Descripción de la variable .....</b>	<b>30</b>
<b>Variables de Tipo Arreglo .....</b>	<b>31</b>
<b>Tipo de datos .....</b>	<b>31</b>
<b>Definición de Controles .....</b>	<b>31</b>
<b>Prefijo para el Control.....</b>	<b>31</b>

<b>Nombre descriptivo del Control .....</b>	<b>32</b>
<b>Declaración de variables, atributos y objetos .....</b>	<b>32</b>
<b>Declaración de Clases .....</b>	<b>32</b>
<b>Declaración de Métodos.....</b>	<b>33</b>
<b>Declaración de Funciones.....</b>	<b>33</b>
<b>Recomendaciones como no programar .....</b>	<b>34</b>
<b>5.02. Diseño de Interfaces de Usuario .....</b>	<b>35</b>
<b>5.03. Especificación de pruebas de unidad .....</b>	<b>42</b>
<b>5.04 Especificación de pruebas de aceptación .....</b>	<b>45</b>
<b>5.05. Especificación de pruebas de carga .....</b>	<b>46</b>
<b>5.06. Configuración del Ambiente mínima/ideal.....</b>	<b>49</b>
<b>Capítulo VI: Aspectos Administrativos .....</b>	<b>51</b>
<b>6.01. Recursos .....</b>	<b>51</b>
<b>    6.01.1 Recurso Humano.....</b>	<b>51</b>
<b>6.02. Presupuesto.....</b>	<b>52</b>
<b>6.03. Cronograma.....</b>	<b>52</b>
<b>Capítulo VII: Conclusiones y Recomendaciones.....</b>	<b>53</b>
<b>7.01. Conclusiones .....</b>	<b>53</b>
<b>7.02. Recomendaciones .....</b>	<b>53</b>
<b>ANEXOS .....</b>	<b>55</b>
<b>Manuales.....</b>	<b>68</b>
<b>    Manual de Instalación .....</b>	<b>68</b>

---

<b>Manual de usuario .....</b>	<b>96</b>
<b>Manual Técnico .....</b>	<b>109</b>
<i>Programas utilizados para el desarrollo de la aplicación: .....</i>	<i>109</i>
<b>Bibliografía .....</b>	<b>130</b>

---

## Índice de tablas

Tabla. 1 <i>Especificación del requerimiento funcional de Actualización inmediata de información del empleado</i> .....	9
Tabla. 2 <i>Especificación del requerimiento funcional Control de procesos para emitir descuentos</i> .....	10
Tabla. 3 <i>Especificación del requerimiento funcional Clasificación de tipos de empleados.</i> ...	11
Tabla. 4 <i>Especificación del requerimiento funcional Acceso restringido de usuarios al módulo de nómina.</i> .....	12
Tabla. 5 <i>Especificación del requerimiento no funcional La aplicación debe funcionar con cualquier navegador.</i> .....	13
Tabla. 6 <i>Especificación del requerimiento no funcional</i> .....	14
Tabla. 7.....	19
Tabla. 8 <i>Especificación del diagrama de realización de la Administración de nómina</i> .....	20
Tabla. 9 <i>Especificación del diagrama de realización Elaboración y consulta de reportes</i> ...	21
Tabla. 10 <i>Especificación de caso de uso Administración de nómina</i> .....	22
Tabla. 11 <i>Especificación de caso de uso Gestión y control de empleado</i> .....	22
Tabla. 12 <i>Especificación de caso de uso Elaboración y consulta de reportes.</i> .....	23
Tabla. 14 <i>Resumen del proyecto que destaca lo que se desea lograr</i> .....	27
Tabla. 15 <i>Detalle de la descripción de un variable</i> .....	30
Tabla. 16 <i>Detalle del tipo de datos de una variable</i> .....	31
Tabla. 17 <i>Detalle del nombre descriptivo para controles</i> .....	32
Tabla. 18 <i>Detalle de como declarar variables, atributos y objetos</i> .....	32
Tabla. 19 <i>Detalle de como declarar clases</i> .....	33
Tabla. 20 <i>Detalle de como declarar Métodos</i> .....	33
Tabla. 21 <i>Detalle de como declarar Funciones</i> .....	34
Tabla. 22 <i>Recomendaciones de que no se debe hacer en la programación</i> .....	34

---

Tabla. 23 Prueba de interface de usuario (estándares) .....	43
Tabla. 24 Pruebas de Reportes, resultados eficientes.....	43
Tabla. 25 Pruebas de compilación de Código.....	44
Tabla. 26 Pruebas de Almacenamiento de datos en la Base .....	44
Tabla. 27 Detalle de pruebas de aceptación en la Gestión y control del empleado .....	45
Tabla. 28 Detalle de pruebas de aceptación en la Administración de nómina .....	45
Tabla. 29 Detalle de pruebas de aceptación en la Elaboración y consulta de reportes.....	46
Tabla. 30 Detalle de un tipo de prueba de carga más baja .....	47
Tabla. 31 Detalle de un tipo de prueba de carga con un número mínimo de usuarios.....	48
Tabla. 32 Detalle de un tipo de prueba de carga con un número máximo de usuarios.....	48
Tabla. 33 Detalle de un tipo de prueba de carga con un número máximo de usuarios ejecutándose hasta llegar al colapso, se requiere descubrir los límites. ....	48
Tabla. 34 Recursos Humanos .....	51
Tabla. 35 Recursos Materiales .....	51
Tabla. 36 Detalle de Gastos realizados en el Proyecto (Presupuesto).....	52

---

## Índice de figuras

Figura. 1 Árbol de problemas.....	16
<i>Figura. 2</i> Árbol de objetivos.....	17
<i>Figura. 3.</i> Diagrama de casos de uso de gestión de nómina.....	18
<i>Figura. 4</i> CU001. Gestión y control del empleado.....	18
<i>Figura. 5</i> CU002 Administración de nómina.....	19
<i>Figura. 6</i> UC003 Elaboración y consulta de reportes.....	20
<i>Figura. 7</i> Diagrama de secuencia.....	21
Figura. 9 Diagrama de componentes.....	26
Figura. 10 Diagrama de estrategias.....	26
Figura. 11 Descripción lógica.....	28
Figura. 12 Descripción física.....	28
Figura. 13 Vista de desarrollo.....	29
Figura. 14 Vista de procesos.....	29
Figura. 15 Diseño general de interface de usuario.....	35
Figura. 16 Ingreso a los módulos.....	35
Figura. 17 Login de ingreso al módulo de nómina.....	36
Figura. 18 Interfaz del usuario.....	36
Figura. 19 Opciones de nómina.....	36
Figura. 20 Opciones base de referencia en nómina.....	37
Figura. 21 Interfaz de ingreso de país.....	37
Figura. 22 Interfaz de ingreso de ciudad.....	37
Figura. 23 Interfaz de ingreso de cargos.....	38
Figura. 24 Interfaz de ingreso de títulos.....	38
Figura. 25 Interfaz de ingreso de tipo de contrato.....	38
Figura. 26 Interfaz de ingreso de secciones y departamentos.....	39

---

Figura. 27 Interfaz de ingreso de conceptos.....	39
Figura. 28 Interfaz de ingreso de empleados.....	39
Figura. 29 Interfaz de asignación de conceptos.....	40
Figura. 30 Interfaz de opciones de procesos especiales .....	40
Figura. 31 Interfaz de transacciones de variable .....	40
Figura. 32 Interfaz de cierre de quincena .....	41
Figura. 33 Interfaz de asistencia del empleado .....	41
Figura. 34 Interfaz de selección de reporte .....	41
Figura. 35 Reporte Generado.....	42
Figura. 36 Tiempos propuestos para el desarrollo y la implementación del Sistema .....	52

---

## Índice de anexos

ANEXO A0 1 Matriz de fuerzas T.....	56
ANEXO A0 2 Entrevista.....	57
ANEXO A0 3 Matriz de requerimientos .....	58
ANEXO A0 4 Mapeo de involucrados .....	59
ANEXO A0 5 Matriz de Involucrados.....	61
ANEXO A0 6 Matriz de Análisis de Alternativas.....	62
ANEXO A0 7 Matriz de Impactos de Objetivos .....	63
<b>ANEXO A0 8 Diagrama de Clases .....</b>	<b>64</b>
ANEXO A0 9 Modelo lógico.....	65
ANEXO A0 10 Modelo Físico .....	67

---

## Resumen ejecutivo

Existen varias empresas que actualmente continúan realizando los procesos relacionados con la nómina de personal de forma manual es decir no cuentan con un sistema que les ayude a efectuar estos procesos de manera rápida y correcta, existen empresas que se dedican al desarrollo de software, pero que se evidencia que las interfaces del sistema en muchas ocasiones son poco amigables y en muchos casos sus aplicaciones son de difícil manejo para el usuario, la aplicación web en su módulo de nómina pretende ser una alternativa de fácil acceso, óptima y eficiente. Manteniendo las seguridades necesarias y permitiendo que al momento de ingresar un usuario a la aplicación tenga un menú dinámico y fácil de manejar que se amolde a los requerimientos del proceso de nómina.

La aplicación en su módulo de nómina permitirá realizar todos los procesos de nómina de forma eficiente optimizando tiempos de respuesta satisfaciendo las necesidades del usuario final.

---

## Abstract

There are several companies currently continue performing the processes related to payroll manually I not have a system to help them make these processes quickly and correctly, there are companies engaged in software development but is evidence that the system interfaces are often unfriendly and in many cases their applications are difficult to manage for the user, the web application on your payroll module intended as an alternative accessible, optimal and efficient.

Maintaining the necessary assurances and allowing when entering a user application has a dynamic and easy to handle menu that fits the requirements of the payroll process.

The application on your payroll module to perform all payroll processes efficiently optimizing response times to meet the needs of the end user.

---

## Introducción

Al verificar los problemas que se dan con respecto al realizar los procesos de nómina de forma manual, en las empresas se hace cada vez más necesario la automatización de procesos debido a que estas tienden a subir sus ingresos a medida que va creciendo la matriz productiva, los procesos manuales no son malos pero se podría implementar la aplicación para ayudar al ingreso, modificación, búsqueda, almacenamiento, manteniendo una constante actualización del aplicativo, brindándoles confianza, tiempo, rapidez y seguridad en todo sus procesos de esta manera se garantizaría el correcto funcionamiento del departamento de nómina en si siendo los principales beneficiarios los empleados de la empresa al contar con un sistema confiable y seguro.

---

## Capítulo I: Antecedentes

### 1.01 Contexto

En toda organización existe un área que se encarga del control del personal que se conoce como departamento de recursos humanos o talento humano, este tiene a cargo determinados trabajos de origen administrativos como la realización de contratos, selección de trabajadores, nómina y seguros sociales entre otros.

Una de las principales áreas dentro de la función de los recursos humanos es la administración de nómina, encargada de administrar y controlar los recursos monetarios destinados a los empleados, los procedimientos administrativos ejecutados en la sección nómina se realizan en base a los lineamientos de las leyes, reglamentos, estatutos, resoluciones, disposiciones administrativas, acciones del personal, y la normativa legal expedida a fin de lograr las metas propuestas por cada empresa, El concepto de nómina comprende muchos aspectos como son los sueldos, salarios, comisiones, indemnizaciones, vacaciones, bonos administrativos, premios, etc. (UTN, 2015)

Estos procesos en las empresas se han venido realizando de forma operativa en hojas de cálculo de Microsoft Excel llenando información como por ejemplo:

- Datos identificativos de la empresa.
- Datos básicos del trabajador, tipo de contrato, categoría y antigüedad.
- Periodo de liquidación al que corresponde dicha nómina.
- Detalle de las horas extra de trabajo.

- Líquido a recibir, de nómina a través del documento que acredita del pago de salarios cerrando los pagos pendientes al trabajador para el periodo estipulado.

Entre otros.

La nómina forma parte del corazón de la empresa en ella, no sólo se ve el salario del trabajador, sino se trabaja con las expectativas, compromisos, responsabilidad y las necesidades del capital humano de la empresa (Gutierrez, 2012). Es muy importante el realizar los pagos de nómina de manera correcta, así como el cumplimiento de los patrones con sus obligaciones, si una empresa no tiene este factor en orden, afectará de manera directa el ambiente laboral y disminuirá la productividad de sus empleados, por lo que la empresa necesita un sistema de nómina que contenga los requerimientos correctos y las especificaciones adecuadas, además de que sea funcional, confiable y que permita optimizar y llevar un registro correcto de los pago de los empleados.

## **1.02 Justificación**

La aplicación web en su "Módulo de nómina" permitirá maximizar la productividad del personal encargado de registrar y controlar los aspectos de la nómina de empleados minimizando y optimizando los tiempos de entrega de recursos de la empresa hacia los trabajadores como ingresos, egresos y total a pagar mediante la entrega de roles individuales y generales de pago.

El módulo de nómina facilita a los responsables de este proceso generar los registros de sus empleados, fechas de ingreso, pago y retiro así como la relación laboral que la empresa ha tenido con cada uno de sus empleados agilizando y precisando los

cálculos de liquidaciones de nóminas, todos estos registros y pagos que se almacenaran en una base de datos que servirá como reportes históricos o la realización de cálculos de la nómina, esta información debe ser de fácil acceso y esté disponible las actualizaciones de los ingresos y salidas de los empleados.

### **1.03 Definición del Problema central**

Para realizar los cálculos necesarios en el pago de nómina del personal asignado se necesita la información detallada de ingresos y egresos de los empleados la cual es llevada de forma manual en hojas de cálculo de Excel, esta información en ocasiones es incompleta, errónea o existe perdidas de la misma, es decir no se lleva un correcto control de esta información de los empleados, ocasionando el retraso en la generación de roles de pago provocando pérdida de tiempo para las personas que se encargan de generar el proceso, de igual manera cuando se requiere el informe o reporte no se dispone en el momento.

El inadecuado control y registro de ingresos y egresos de los empleados en definitiva son los errores que se comenten en el proceso de los registros de su nómina y generar los roles de pago e informes requeridos por los empleados.

### 1.03.1 Matriz de Análisis de Fuerzas.

En la matriz de fuerzas T analizamos la situación actual, la situación empeorada y la situación mejorada tomando en cuenta las fuerzas impulsadoras y las fuerzas bloqueadoras para mayor detalle ver ANEXO A01

#### Interpretación

Fuerza Impulsadora: **Coordinación entre departamentos administrativos de la empresa**

I = 2 La falta de coordinación entre departamentos administrativos de la empresa provoca demoras en la generación de reportes o a su vez roles de pago.

PC = 5 Tener la información necesaria para los reportes y roles a tiempo

Fuerza Impulsadora: **Registros completos y correctos de empleados**

I = 1 El tener los registros completos y correctos de los empleados permite optimizar la generación de reportes y roles

PC = 5 Disminución de tiempos de entrega de roles y reportes

Fuerza Impulsadora: **Utilización correcta de equipos**

I = 1 El buen uso de los equipos permitirá maximizar el desempeño en el departamento de nomina

PC = 4 Incrementar el desempeño general del departamento de nomina

Fuerza Impulsadora: **Acceso fácil e intuitivo al módulo de nomina**

I = 2 Al tener un intuitivo y fácil acceso al módulo de nómina los encargados

desempeñaran de mejor manera su labor

PC = 4 Agilizar la labor de los empleados

Fuerza Impulsadora: **Personal capacitado en los procesos de nómina.**

I = 3 Se logra la elaboración de reportes y entrega de roles de manera óptima.

PC = 4 Incremento de eficiencia en el departamento.

Fuerza Bloqueador: **No disponibilidad de información a tiempo**

I = 4 No se dispone de la información en las fechas indicadas para el proceso.

PC = 4 Al no tener la información a tiempo demora la entrega de cualquier tipo de reporte o informe requerido.

Fuerza Bloqueadora: **Registros incompletos de los empleados**

I = 3 Los registros de los empleados son incompletos

PC = 5 Los registros incompletos generan reportes erróneos de los empleados

Fuerza Bloqueadora: **Navegadores no actualizados**

I = 4 Es necesario la actualización general del equipo.

PC = 3 De no ser actualizados los sistemas de los equipos estos no trabajarán de manera adecuada y no permitirá ejecutar eficientemente la aplicación web.

Fuerza Bloqueadora: **No conocimiento básico de informática**

I = 4 Al no tener conocimientos básicos de informática no se tendrá un correcto uso de la aplicación.

---

PC = 4 Se necesitara brindar continuo soporte al personal de nómina.

Fuerza Bloqueadora: **Inexistencia de planes de capacitación al personal de nómina.**

I = 4 El personal de nómina no posee conocimientos actualizados para ejecutar su labor.

PC = 5 La capacitación al personal demorara la implementación de la aplicación.

---

## Capítulo II: Análisis de Involucrados

### 2.01 Requerimientos

#### 2.01.1 Descripción del sistema actual.

La empresa actualmente maneja la gestión de nómina de manera manual es decir todos los cálculos y registros son realizados en un documento en Excel por lo que se hace muy susceptible a errores sean estos por digitación o formulas mal planteadas para el cálculo en el proceso, el personal registra su asistencia de una manera no muy fiable, esto los realiza en formularios que luego son digitalizados de igual manera en documentos Excel para su respectivo control, es decir toda la información de los empleados se encuentra en hojas distintas de Excel esto provoca que al requerir la generación de un rol o reporte se tenga que buscar en todas las hojas la información requerida para la elaboración del mismo este proceso es demoroso provocando retrasos en las entregas de roles y molestias en los empleados.

#### 2.01.2 Visión.

Mediante la aplicación web en su módulo de nómina se conseguirá gestionar de manera apropiada los procesos de pagos de pago de nómina, permitiendo así el control y registro de ingresos y egresos, de una forma clara y concisa para salvaguardar los intereses de las personas involucradas directamente a la empresa y facilitar la obtención de reportes para la toma de decisiones posteriores.

#### *Alcance*

En el módulo de nómina se manejará el, salario total, deducciones, provisiones, realizará los cálculos necesarios para el procesamiento y generación de roles de pago generales e individuales.

Igualmente, un inventario de personal, donde se reflejen los datos principales del trabajador (Número de empleado, fecha de alta, fecha de antigüedad, nombre, cargo, entre otros) y su información personal (Dirección, Teléfono, Sexo, estado civil, fecha de nacimiento, entre otros), así como las condiciones de pago para el trabajador (Efectivo/Transferencia/Cheque).

### **2.01.3 Entrevistas.**

En la entrevista se realiza preguntas planteándose objetivos para realizar un posterior análisis y determinar requerimientos del usuario para un mayor detalle ver

ANEXO 02

### **2.01.4 Matriz de Requerimientos**

En la matriz de requerimientos establecemos los requerimientos funcionales y no funcionales para nuestro módulo para mayor detalle ver ANEXO A01.

### 2.01.5 Descripción Detallada

Tabla. 1 *Especificación del requerimiento funcional de Actualización inmediata de información del empleado*

**Identificador:** RQF001

<b>Descripción del Requerimiento:</b> Actualización inmediata de información del empleado.		<b>Estado</b>	<b>Análisis</b>
<b>Creado por:</b>	César Cajamarca	<b>Actualizado por:</b>	César Cajamarca
<b>Fecha de Creación:</b>	24/11/2014	<b>Fecha de Actualización:</b>	16/12/2014
<b>Tipo de Requerimiento:</b>	Crítico	<b>Tipo de Requerimiento:</b>	Funcional
<b>Datos de Entrada:</b>	Información del empleado. Nombres, Apellidos, Número de Cédula, Dirección, Teléfono Fijo, Teléfono Celular, Área de Trabajo, Cargo.		
<b>Descripción:</b>	Se registrara los datos completos del empleado y se almacenara en una base de datos con finalidad que esta sea accesible el momento que se la requiera.		
<b>Datos de Salida:</b>	Datos informativos del empleado		
<b>Resultados Esperados:</b>	Obtener la información actualizada del empleado de manera rápida.		
<b>Origen:</b>	Recursos Humanos		
<b>Dirigido a:</b>	Nómina		
<b>Prioridad:</b>	3		
<b>Requerimientos asociados:</b>	RQF001		
<b>ESPECIFICACIÓN:</b>			
<b>Pre condiciones:</b>	<ol style="list-style-type: none"> <li>1. Actualización de registros de información de los empleados</li> <li>2. Actualización de cargos y cambios realizados en nómina.</li> </ol>		
<b>Pos condiciones:</b>	<ol style="list-style-type: none"> <li>1. Utilización de la información bajo parámetros éticos.</li> </ol>		
<b>Criterios de Aceptación:</b>	Permite al usuario la visualización de la información completa del empleado.		

*Nota. En requerimiento asociado se refiere al requerimiento funcional 001*

Tabla. 2 Especificación del requerimiento funcional Control de procesos para emitir descuentos

<b>Identificador:</b>	RQF002	<b>Tipo de Requerimiento:</b>	Crítico	<b>Tipo de Requerimiento:</b>	Funcional
<b>Descripción del Requerimiento:</b> Control y registro de ingresos y egresos.		<b>Estado</b>	Análisis		
<b>Creado por:</b>	César Cajamarca	<b>Actualizado por:</b>	César Cajamarca		
<b>Fecha de Creación:</b>	24/11/2014	<b>Fecha de Actualización:</b>	16/12/2014		
<b>Datos de Entrada:</b>	Información del empleado. <ul style="list-style-type: none"> <li>• Ingresos mensuales</li> <li>• Egresos mensuales</li> <li>• Cálculos totales</li> </ul>				
<b>Descripción:</b>	Se registrara los datos completos del empleado y se almacenara en una base de datos con finalidad que esta sea accesible el momento que se la requiera.				
<b>Datos de Salida:</b>	Al registrar los descuentos de cada empleado se lleva un control más detallado de las fechas en que realizan.				
<b>Resultados Esperados:</b>	Tener un mejor control de la información de descuentos de todos los empleados				
<b>Origen:</b>	Contabilidad				
<b>Dirigido a:</b>	Remuneraciones				
<b>Prioridad:</b>	3				
<b>Requerimientos asociados:</b>	RQF001				
<b>ESPECIFICACIÓN:</b>					
<b>Pre condiciones:</b>	<ol style="list-style-type: none"> <li>1. Para ejecutar este requerimiento el empleado debe estar debidamente registrado.</li> <li>2. El encargado debe ingresar todos los ingresos y descuentos de una forma correcta.</li> <li>3. El encargado del proceso debe tener en cuenta la categorización de los descuentos para cada empleado.</li> </ol>				
<b>Pos condiciones:</b>	<ol style="list-style-type: none"> <li>1. Los descuentos se realizaran mediante las fechas establecidas por Recursos Humanos</li> </ol>				
<b>Criterios de Aceptación:</b>	Permite un mejor registro de ingresos y egresos a todos los empleados.				

*Nota. En requerimiento asociado se refiere al requerimiento funcional 001*

Tabla. 3 Especificación del requerimiento funcional Clasificación de tipos de empleados.

Descripción del Requerimiento: Clasificación de tipos de empleados.		Estado	Análisis
<b>Creado por:</b>	César Cajamarca	<b>Actualizado por:</b>	César Cajamarca
<b>Fecha de Creación:</b>	24/11/2014	<b>Fecha de Actualización:</b>	16/12/2014
<b>Identificador:</b>	RQF003		
<b>Tipo de Requerimiento:</b>	Crítico	<b>Tipo de Requerimiento:</b>	Funcional
<b>Datos de Entrada:</b>	Información del empleado. <ul style="list-style-type: none"> <li>• Tipos de Contratos establecidos con el empleado.</li> <li>• Tiempo de trabajo.</li> </ul>		
<b>Descripción:</b>	Se registrara los detalles del ingreso del trabajador siendo estos almacenados en la base de datos.		
<b>Datos de Salida:</b>	Identificación de los empleados por permanencia en la empresa y tipos de contrato.		
<b>Resultados Esperados:</b>	Tener un mejor control de la información de ingreso de todos los empleados		
<b>Origen:</b>	Recursos Humanos		
<b>Dirigido a:</b>	Remuneraciones y Contabilidad		
<b>Prioridad:</b>	3		
<b>Requerimientos asociados:</b>	RQF001		
<b>ESPECIFICACIÓN:</b>			
<b>Pre condiciones:</b>	<ol style="list-style-type: none"> <li>1. Para ejecutar este requerimiento el empleado debe estar debidamente registrado.</li> <li>2. El encargado debe ingresar todos los detalles de una forma correcta.</li> <li>3. El encargado del proceso debe tener en cuenta la categorización para cada empleado.</li> </ol>		
<b>Pos condiciones:</b>	<ol style="list-style-type: none"> <li>1. Identificar los Tipos de empleados de la empresa</li> </ol>		
<b>Criterios de Aceptación:</b>	Permite un mejor registro de categorización de los empleados.		

*Nota. En requerimiento asociado se refiere al requerimiento funcional 001*

Tabla. 4 *Especificación del requerimiento funcional Acceso restringido de usuarios al módulo de nómina.*

<b>Identificador:</b>	RQF004	<b>Tipo de Requerimiento:</b>	Crítico	<b>Tipo de Requerimiento:</b>	Funcional
<b>Descripción del Requerimiento:</b> Acceso restringido de usuarios al módulo de nómina.		<b>Estado</b>	Análisis		
<b>Creado por:</b>	César Cajamarca	<b>Actualizado por:</b>	César Cajamarca		
<b>Fecha de Creación:</b>	24/11/2014	<b>Fecha de Actualización:</b>	16/12/2014		
<b>Datos de Entrada:</b>	Información de los usuarios del módulo de nómina. <ul style="list-style-type: none"> <li>• Cargo que desempeña.</li> <li>• Nivel de uso del modulo</li> </ul>				
<b>Descripción:</b>	Se registrara la información en la base de datos de usuarios del módulo.				
<b>Datos de Salida:</b>	Se admite el ingreso al módulo por niveles de usuarios.				
<b>Resultados Esperados:</b>	Tener un mejor control y seguridad de la información y proceso para la generación de pagos de nómina.				
<b>Origen:</b>	Recursos Humanos				
<b>Dirigido a:</b>	Remuneraciones y Contabilidad				
<b>Prioridad:</b>	3				
<b>Requerimientos asociados:</b>	RQF001, RQF002				
<b>ESPECIFICACIÓN:</b>					
<b>Pre condiciones:</b>	<ol style="list-style-type: none"> <li>1. Para ejecutar este requerimiento el empleado debe estar debidamente registrado.</li> <li>2. El encargado debe ingresar todos los detalles de una forma correcta.</li> <li>3. El encargado del proceso debe tener en cuenta el grado de responsabilidad y cargo del empleado registrado.</li> </ol>				
<b>Pos condiciones:</b>	<ol style="list-style-type: none"> <li>1. Clasificación por niveles de responsabilidad en el uso del módulo de nómina.</li> </ol>				
<b>Criterios de Aceptación:</b>	Brinda mayor seguridad en el manejo de información para el proceso de pagos de nómina.				

*Nota. En requerimiento asociado se refiere a los requerimientos funcionales 001 y 002*

Tabla. 5 Especificación del requerimiento no funcional La aplicación debe funcionar con cualquier navegador.

Descripción del Requerimiento: La aplicación debe funcionar en cualquier navegador		Estado	Análisis
<b>Creado por:</b>	César Cajamarca	<b>Actualizado por:</b>	César Cajamarca
<b>Fecha de Creación:</b>	24/11/2014	<b>Fecha de Actualización:</b>	16/12/2014
<b>Identificador:</b>	RQF005	<b>Tipo de Requerimiento:</b>	No Funcional
<b>Datos de Entrada:</b>	La dirección de url debe estar bien identificada		
<b>Descripción:</b>	Al escribir la dirección del url se abrirá la aplicación y podrán empezar a realizar los procesos		
<b>Datos de Salida:</b>	Página de inicio de la aplicación		
<b>Resultados Esperados:</b>	Tener un mejor control de la información por medio de la aplicación.		
<b>Origen:</b>	Recursos humanos		
<b>Dirigido a:</b>	Remuneraciones y Contabilidad		
<b>Prioridad:</b>	3		
<b>Requerimientos asociados:</b>	RQF001, RQF002		
<b>ESPECIFICACIÓN:</b>			
<b>Pre condiciones:</b>	1. Deberá tener un navegador instalado en su equipo.		
<b>Pos condiciones:</b>	1. El encargado podrá ingresar a la aplicación desde cualquier navegador		
<b>Criterios de Aceptación:</b>	Permite un buen manejo de información.		

Nota. En requerimiento asociado se refiere a los requerimientos funcionales 001 y 002

Tabla. 6 Especificación del requerimiento no funcional

<b>Identificador:</b>	RQF006	<b>Tipo de Requerimiento:</b>	Medio	<b>Tipo de Requerimiento:</b>	No Funcional
<b>Datos de Entrada:</b>	Fácil ingreso de información				

<b>Descripción del Requerimiento:</b> La interfaz debe ser sencilla, de fácil ingreso y recepción de datos		<b>Estado</b>	<b>Análisis</b>
<b>Creado por:</b>	César Cajamarca	<b>Actualizado por:</b>	César Cajamarca
<b>Fecha de Creación:</b>	24/11/2014	<b>Fecha de Actualización:</b>	16/12/2014
<b>Descripción:</b>	La interfaz de la aplicación será simple e intuitiva para el ingreso y salida de información.		
<b>Datos de Salida:</b>	Resultados de cálculos, generación de roles y reportes de pago		
<b>Resultados Esperados:</b>	Obtener de manera fácil, rápida, segura y eficiente los pagos de nomina		
<b>Origen:</b>	Recursos humanos		
<b>Dirigido a:</b>	Remuneraciones y Contabilidad		
<b>Prioridad:</b>	3		
<b>Requerimientos asociados:</b>	RQF001, RQF002		
<b>ESPECIFICACIÓN:</b>			
<b>Pre condiciones:</b>	1. Deberá tener un navegador instalado en su equipo.		
<b>Pos condiciones:</b>	2. El encargado podrá ingresar a la aplicación desde cualquier navegador		
<b>Criterios de Aceptación:</b>	Permite un buen manejo de información.		

*Nota. En requerimiento asociado se refiere a los requerimientos funcionales 001 y 002*

## 2.02 Mapeo de Involucrados

En el mapeo de involucrados se establece las personas involucradas en el proceso.

### Involucrados directos:

- Director de Recursos Humanos
- Analista de Remuneraciones
- Supervisor de Contabilidad

Se consideran involucrados directos por estar relacionados directamente con el proceso de nómina, es participan o contribuyen en el proceso de elaboración de roles y reportes.

**Involucrados Indirectos:**

- Empleados

Los empleados son Involucrados Indirectos por que no participan o no contribuyen en el proceso de elaboración de roles y reportes.

Para mayor detalle ver ANEXO A04.

**2.03 Matriz de Involucrados**

En la matriz de involucrados establecemos las personas

involucradas en el proceso de nómina identificando intereses sobre el problema, problemas percibidos, recursos, mandatos y capacidades, interés sobre el proyecto,

Para mayor detalle ver ANEXO A05.

## Capítulo III: Problemas y Objetivos

### 3.01 Árbol de Problemas

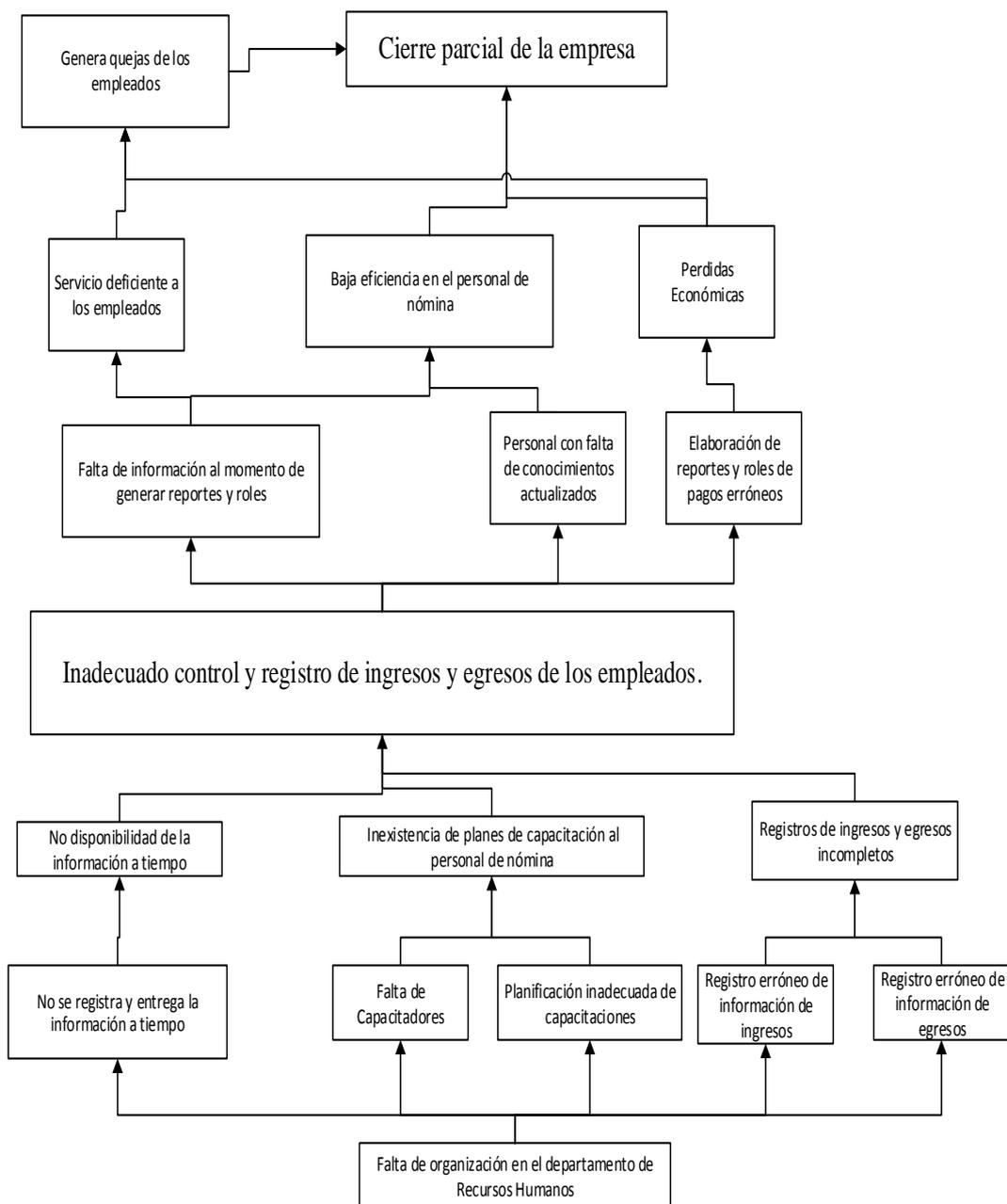


Figura. 1 Árbol de problemas.

El árbol de problemas tiene como eje fundamental la problemática que se va resolver, aquí se detalla los problemas detectados con sus causas y efectos.

### 3.02 Árbol de Objetivos

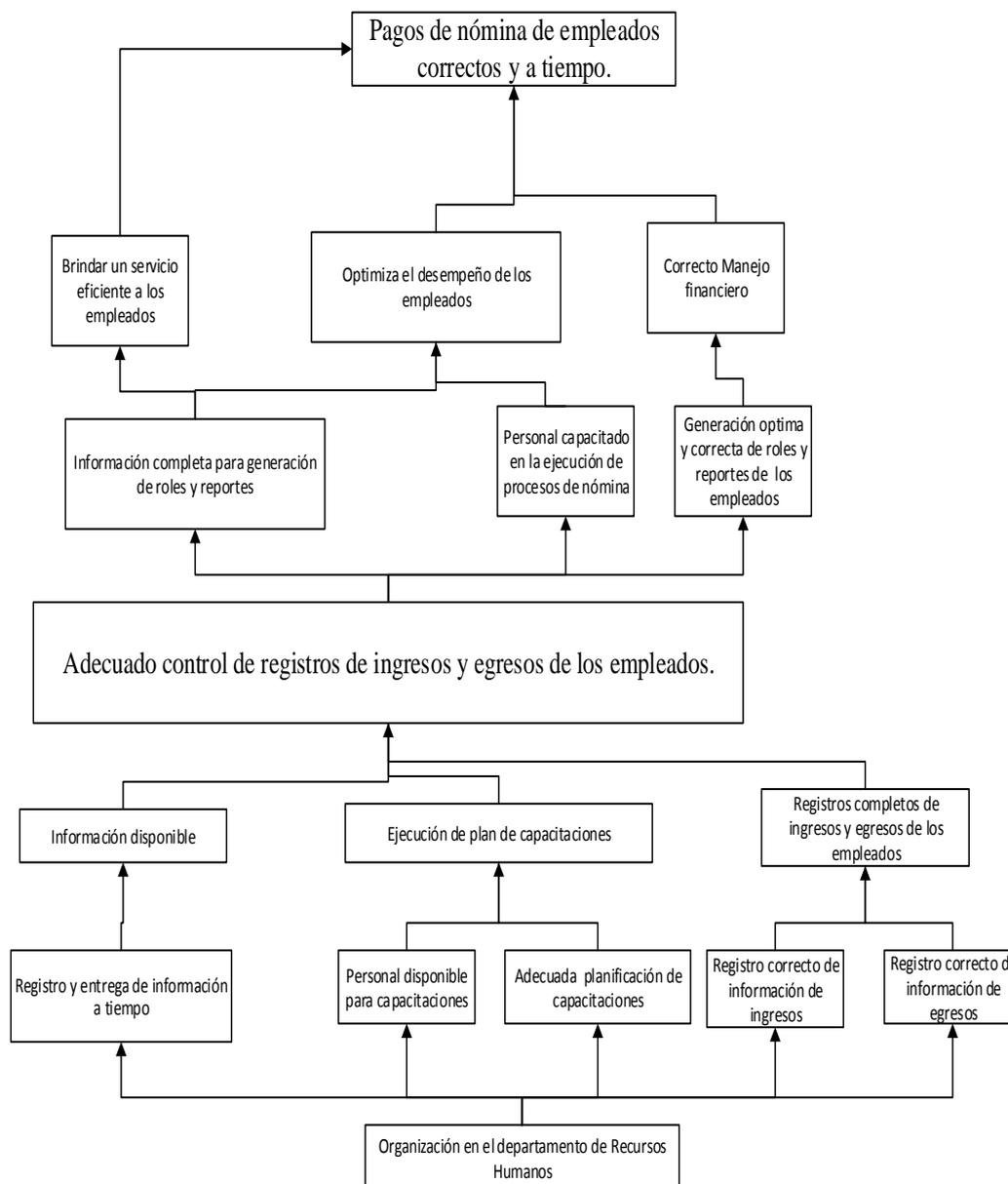


Figura. 2 Árbol de objetivos.

El árbol de objetivos como su nombre lo indica es donde se detalla los objetivos del proyecto tomando como punto inicial el propósito del mismo.



Tabla. 7

Especificación del diagrama de realización Gestión y control del empleado.

CASO DE USO DE GESTION Y CONTROL DEL EMPLEADO	
<b>Nombre</b>	Gestión y control del empleado
<b>Identificador</b>	UC001
<b>Responsabilidades</b>	Registro y verificación del
<b>Tipo</b>	Sistema
<b>Referencias</b>	UC001
<b>Casos de Uso</b>	
<b>Referencias</b>	
<b>Requisitos</b>	RF001, RF002, RF003
<b>PRECONDICIONES</b>	
1.- Se debe ingresar previamente la información del empleado.	
2.- Los datos ingresados deben ser verificados.	
<b>POSCONDICIONES</b>	
1.- Registro detallado del empleado	
<b>SALIDAS PANTALLA</b>	
Interfaz del administrador con varios campos para ingresar información detallada de cada Usuario y Opciones para elegir la categoría a la que pertenece, además el estado en el que se encuentra.	

**Caso: UC002**

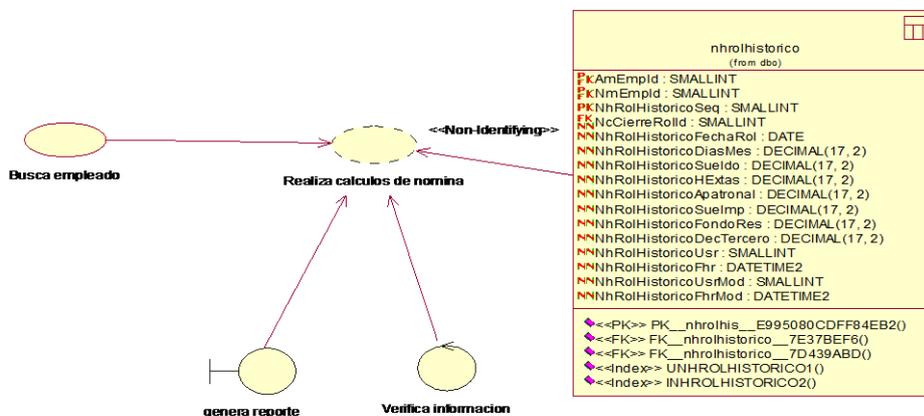


Figura. 5 CU002 Administración de nómina.

El caso de uso muestra el proceso para realizar los cálculos de nómina.

Tabla. 8 Especificación del diagrama de realización de la Administración de nómina.

CASO DE USO ADMINISTRACIÓN DE NÓMINA	
<b>Nombre</b>	Administración de nómina
<b>Identificador</b>	UC002
<b>Responsabilidades</b>	Verificar la información necesaria para los cálculos. Realizar los cálculos necesarios para generar el reporte de pagos
<b>Tipo</b>	Sistema
<b>Referencias Casos de Uso</b>	UC001
<b>Referencias Requisitos</b>	RF001
<b>PRECONDICIONES</b>	
1.- Ingreso de información detallada del empleado. 2.- Verificar información del empleado.	
<b>POSCONDICIONES</b>	
1.- Visualización de información detallada del empleado	
<b>SALIDAS PANTALLA</b>	
Interfaz con la información detallada de los empleados solicitados	

**Caso: UC003**

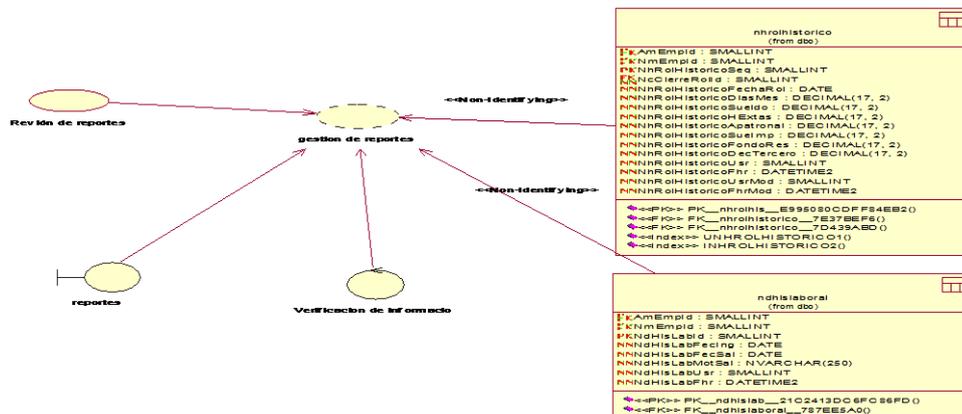


Figura. 6 UC003 Elaboración y consulta de reportes.

El caso de uso muestra el proceso de generación de reportes y roles para el empleado.

Tabla. 9 Especificación del diagrama de realización Elaboración y consulta de reportes.

### CASO DE USO ELABORACIÓN Y CONSULTA DE REPORTES

<b>Nombre</b>	Elaboración y consulta de reportes.
<b>Identificador</b>	UC003
<b>Responsabilidades</b>	Verificar la información necesaria para los cálculos. Realizar los cálculos necesarios para generar el reporte de pagos
<b>Tipo</b>	Sistema
<b>Referencias</b>	
<b>Casos de Uso</b>	UC001, UC002
<b>Referencias</b>	
<b>Requisitos</b>	RF001, RF002

#### PRECONDICIONES

- 1.- Ingreso de información detallada del empleado.
- 2.- Verificar información del empleado.

#### POSCONDICIONES

- 1.- Visualización de información detallada del empleado

#### SALIDAS PANTALLA

Visualización de reportes y roles de pago

### 3.05 Diagramas de secuencia.

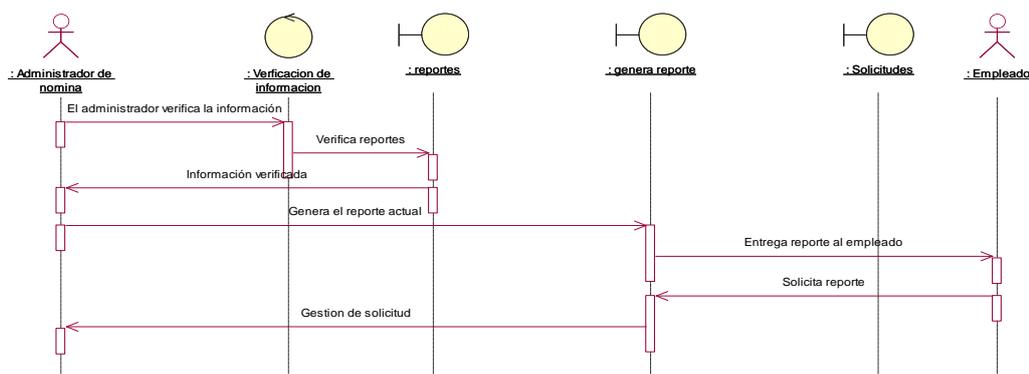


Figura. 7 Diagrama de secuencia.

En este proceso el administrador de nómina verifica la información del empleado, reportes anteriores y procede a generar el reporte para ser entregado al empleado,

cuando el empleado solicita un reporte primero se gestiona la solicitud para que luego el administrador de nómina repita el ciclo anterior.

### 3.06 Especificaciones de casos de uso

Tabla. 10 *Especificación de caso de uso Administración de nómina*

ESPECIFICACIÓN DE ADMINISTRACIÓN DE NÓMINA	
<b>Caso de uso</b>	<b>Administración de nomina</b>
<b>Identificador</b>	<b>Caso: UC002</b>
<b>CURSO TÍPICO DE EVENTOS</b>	
Usuario	Sistema
<ol style="list-style-type: none"> <li>1. Ingresa al módulo</li> <li>2. Revisa y valida registros</li> <li>3. Genera comprobantes</li> <li>4. Entrega reportes</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema realiza una búsqueda del empleado en la base de datos y muestra todos los registros necesarios para el cálculo de nómina.</li> </ol>
<b>CURSOS ALTERNATIVOS</b>	
El empleado puede solicitar el reporte, para lo cual se realiza los siguientes pasos:	
<ol style="list-style-type: none"> <li>2. Se valida solicitud</li> <li>3. El administrador de nómina realiza los pasos anteriormente mencionados.</li> </ol>	

Tabla. 11 *Especificación de caso de uso Gestión y control de empleado*

ESPECIFICACIÓN DE GESTIÓN Y CONTROL DE EMPLEADO	
<b>Caso de uso</b>	<b>Gestión y control de Empleado</b>
<b>Identificador</b>	<b>Caso: UC001</b>
<b>CURSO TÍPICO DE EVENTOS</b>	
Usuario	Sistema
<ol style="list-style-type: none"> <li>1. El administrador de nómina ingresa toda la información del nuevo empleado</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema guarda la información ingresada en la base de datos para posterior uso.</li> </ol>
<b>CURSOS ALTERNATIVOS</b>	
De tratarse de un reingreso el administrador de nómina revisa la información existente y la actualiza	

Tabla. 12 *Especificación de caso de uso Elaboración y consulta de reportes.*

ESPECIFICACIÓN DE ELABORACIÓN Y CONSULTA DE REPORTE	
<b>Caso de uso</b>	<b>Elaboración y consulta de reportes</b>
<b>Identificador</b>	<b>Caso: UC003</b>
<b>CURSO TÍPICO DE EVENTOS</b>	
Usuario	Sistema
<ul style="list-style-type: none"> <li>- El usuario realiza la consulta del proceso o empleado requerido</li> </ul>	<ul style="list-style-type: none"> <li>- El sistema realiza la búsqueda del empleado y muestra los registros de para la realización de reportes o consultas de reportes realizados anteriormente.</li> </ul>
<b>CURSOS ALTERNATIVOS</b>	
De tratarse de consultas a reportes anteriores estas se realizan en tipo histórico.	

---

## Capítulo IV: Análisis de Alternativas

### 4.01. Matriz de Análisis de Alternativas

Mediante la matriz de análisis de alternativas analizamos la factibilidad y el impacto.

Ver Anexo 3

### 4.02. Matriz de Impactos de Objetivos

Mediante esta matriz analizamos la factibilidad y los impactos del proyecto.

Ver Anexo 4

### 4.03 Estándares para el Diseño de Clases

#### Formato de Clases-Interfaz

Al nombrar a las clases la primera letra debe ser escrita con letra mayúscula, se debe usar sustantivos ejemplo: Nomina.

Para las interfaces siempre se debe usar un sustantivo mas adjetivo ejemplo:

CampoSimple o calificativos ejemplo: Editable

El nombre de la clase o interfaz debe estar en singular y comenzar con mayúscula, no estar abreviado.

En caso de estar compuesto por más de una palabra, las mismas se concatenaran y cada una de ellas comenzará con mayúsculas. (MAG, 2014)

#### Atributos

Deben escribirse en minúscula si constan de una palabra. Al poseer más de Una palabra, la primera ha de seguir el formato descrito anteriormente, pero Las palabras posteriores deben comenzar con cada primera letra en Mayúscula.

Detallar el tipo de dato de los atributos de las clases.

Detallar la visibilidad o el modo de acceso, es decir, si es pública (+), privada (-) o protegida (#).

Al detallar un atributo "static", será representado por texto subrayado.

### **Operaciones**

Deben escribirse en minúscula si constan de una palabra. Al poseer más de una palabra, la primera ha de seguir el formato descrito anteriormente, pero las palabras posteriores deben comenzar con cada primera letra en Mayúscula.

Dentro de las operaciones debe encontrarse el constructor de la clase.

Se debe especificar el tipo de dato, tanto de los parámetros como el retorno.

(Adriana, 2013)

#### **4.04. Diagrama de clases**

El diagrama describe la estructura de un sistema mostrando sus clases, orientados a objetos, de este diagrama parte para que se pueda modelar el Físico, Lógico y generar el script para a base de datos para mayor detalle ver anexo 8.

#### **4.05. Modelo Lógico – Físico**

##### **Modelo Lógico**

Se observa la estructura de la base de datos, adaptando al modelo de datos a utilizar transformando las entidades y relaciones en tablas para mayor detalle ver Anexo 9.

##### **Modelo Físico**

Se observa las tablas que componen la estructura de la base de datos para mayor detalle ver Anexo 7.

#### **4.06. Diagrama de Componentes**

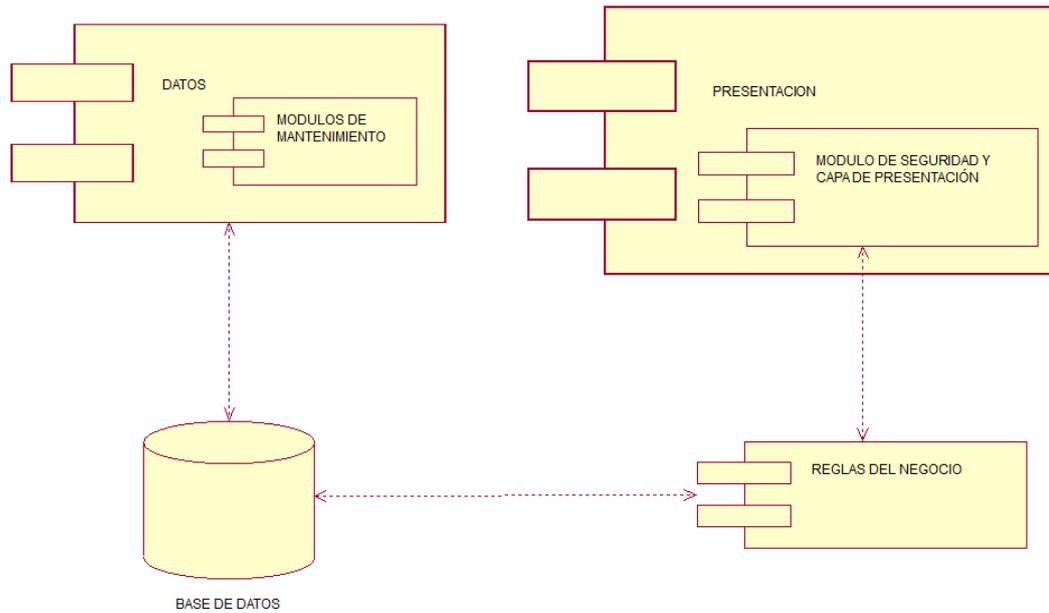


Figura. 8 Diagrama de componentes

En el diagrama observamos los componentes que actúan en el funcionamiento del sistema.

#### 4.07. Diagramas de Estrategias

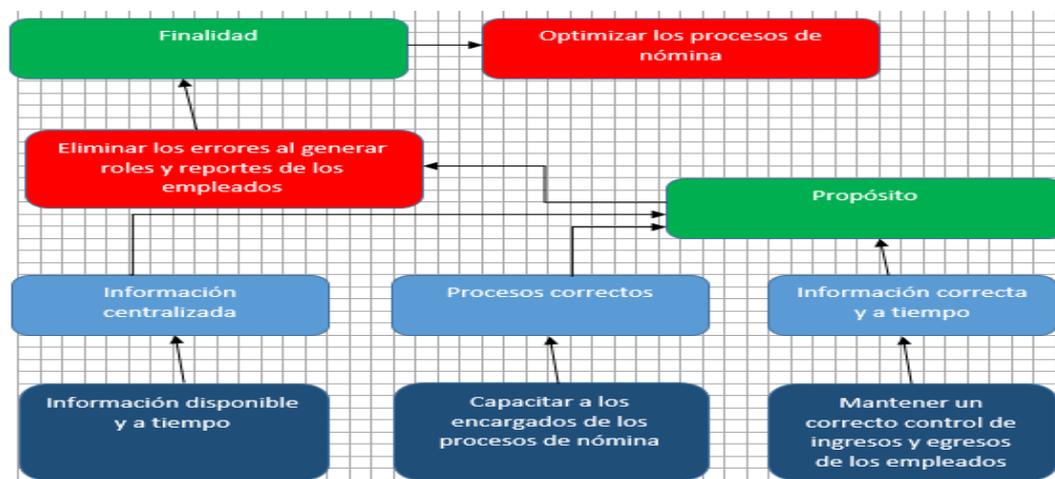


Figura. 9 Diagrama de estrategias

#### 4.08. Matriz de Marco Lógico

Tabla. 13 *Resumen del proyecto que destaca lo que se desea lograr*

Resumen narrativo de objetos	Indicadores	Medios de verificación	Supuestos
Fin: Optimizar los procesos de nómina	Proceso de obtención de roles y reportes de manera eficiente.	Fácil obtención de la información de todos los empleados involucrados en el proceso de nómina.	Inducción a empleados de la empresa a que utilicen herramientas tecnológicas.
Propósito: Eliminar los errores al generar roles y reportes de los empleados	Optimización de tiempos y aumento en la eficiencia de los empleados.	Se optimiza los recursos y mejorar los procesos.	Aumento en la eficiencia y satisfacción de los empleados
Componentes: 1. Sistema fácil de utilizar y agradable a la vista del usuario.	Realizar pruebas a la aplicación con usuarios múltiples comprobando la complejidad de su uso.	Realizar modelos que muestre las mejoras que se está logrando	No contemplar las restricciones de la aplicación.
Actividad: 1. Analizar el proceso de generación de roles y reportes. 2. Levantamiento de requerimientos. 3. Realización de Base de Datos y desarrollo de la aplicación.	Claridad de los requerimientos al realizar los procesos en la aplicación	Documentar todo aquello que se esté realizando según los avances del proyecto.	El tiempo es demasiado corto para desarrollar la aplicación en su totalidad.

*Nota. La tabla muestra los objetivos del proyecto es decir lo que se desea alcanzar*

## 4.09. Vistas arquitectónicas

### 4.09.01. Vista lógica.

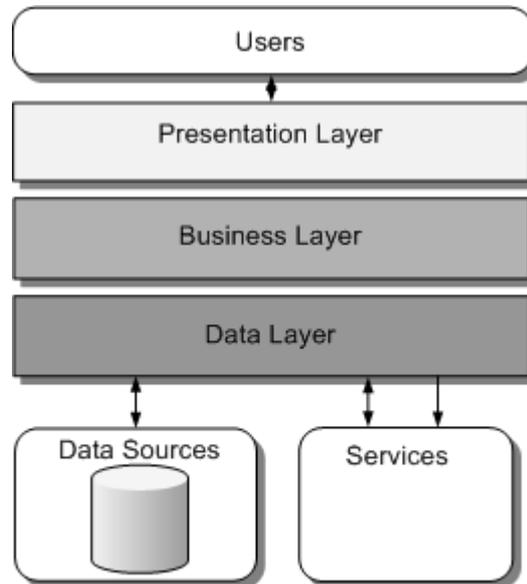


Figura. 10 Descripción lógica

(Gonzales, 2014)

### 4.09.02. Vista física

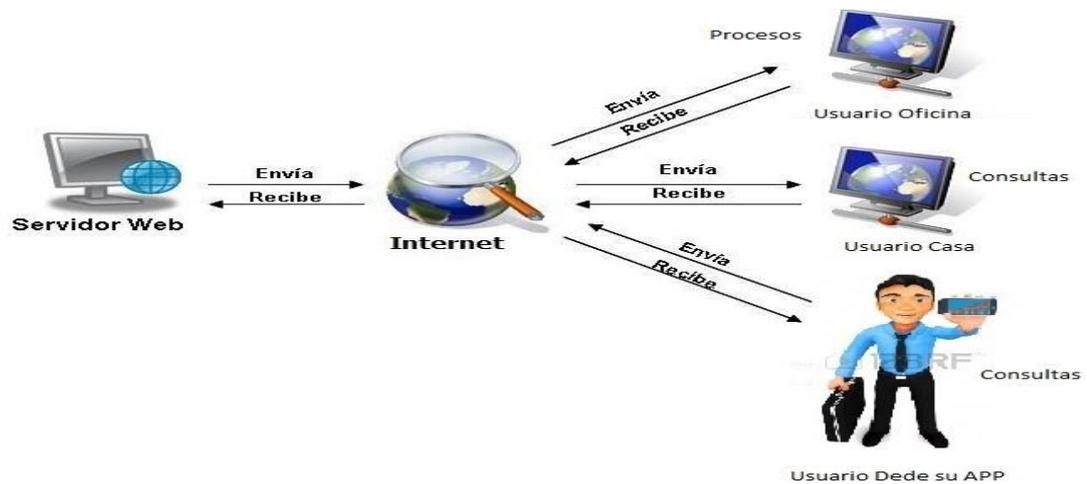


Figura. 11 Descripción física

(Ruiz, 2014)

#### 4.09.03. Vista de desarrollo

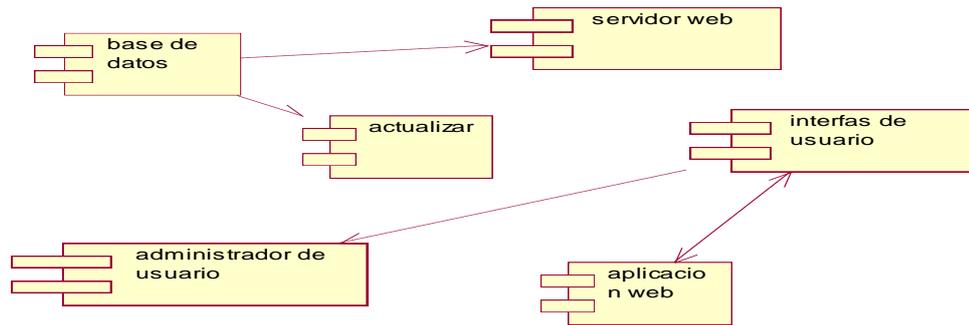


Figura. 12 Vista de desarrollo

#### 4.09.04. Vista de procesos

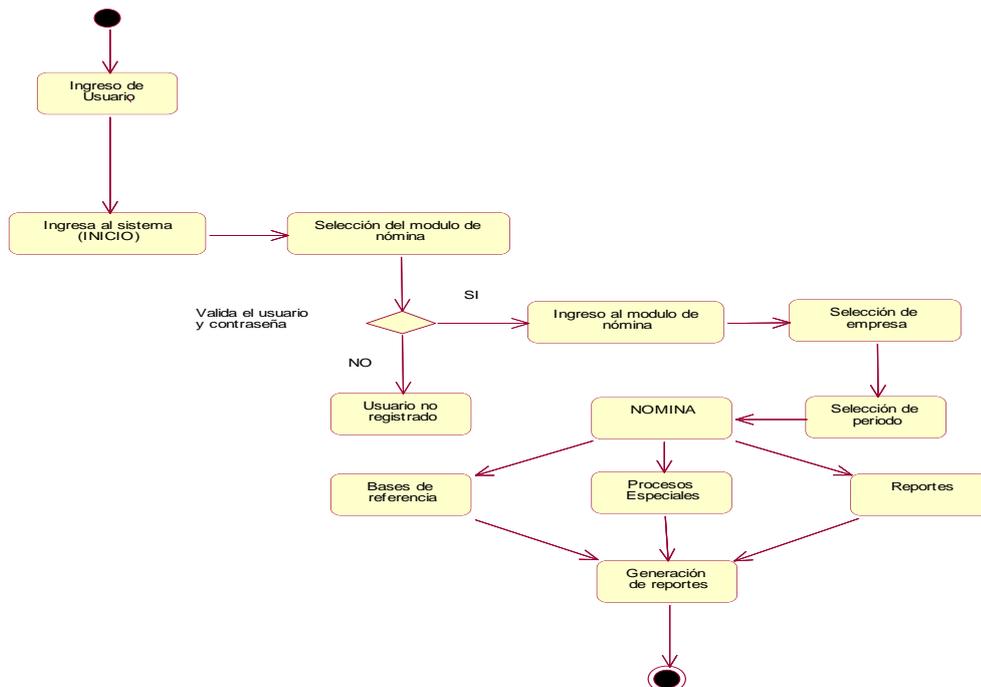


Figura. 13 Vista de procesos

El proceso de la figura muestra el ingreso al sistema para la generación de reportes desde iniciando por el ingreso de usuario y contraseña hasta la generación final del reporte.

## Capítulo V: Propuesta

### 5.01. Especificación de estándares de programación

En esta parte se reglamentara el código fuente a utilizar en el desarrollo de la aplicación.

#### Declaraciones de Variables

Las variables deben ajustarse al requerimiento para el que sean creadas

El mnemotécnico definido se establece tomando en consideración principalmente lo siguiente:

- La longitud debe ser lo más recomendable posible.
- El tipo de dato al que pertenece la variable.

Por lo tanto la estructura de la variable es:

Siendo el nombre que identifica a la variable: \$a

Tabla. 14 *Detalle de la descripción de un variable*

Estructura	Descripción de la Variable
LONGITUD. MAX.	← 1 →← 16 →
FORMATO	Todo con minúsculas
EJEMPLO	\$meses, \$a = 1; \$cadena="Hola amigo";

#### Descripción de la variable

El nombre que se le asignará a la variable para que se le identifique y deberá estar asociada al motivo para la cual se le declara.

**Ejemplo:** \$a = 1; \$cadena="Hola amigo"; \$suma = "suma";

## Variables de Tipo Arreglo

En el caso de las definiciones de arreglos de elementos se declarara la variable con el prefijo de "lista", el cual nos dará entender que se trata de una variable del tipo arreglo la cual contendrá de cero a más datos, según el tamaño declarado.

**Ejemplos:** lista\_meses

```
$lista_meses=
```

```
array("ENE","FEB","MAR","ABR","MAY","JUN","JUL","AGO","SEP","OCT","NOV","DIC");
```

## Tipo de datos

Tabla. 15 *Detalle del tipo de datos de una variable*

Tipo de variable	Mnemónico	Descripción
Byte	By	Entero de 8 bits sin signo.
Integer	In	Entero de 32 bits con signo.
Char	Ch	Un carácter UNICODE de 16 bits
String	St	Cadena de caracteres
Date	Dt	Formato de fecha/hora
Boolean	Bl	Valor lógico: verdadero y falso
Float	Fl	Coma flotantes, 11-12 dígitos significativos.
Double	Db	Coma flotante, 64 bits (15-16 dígitos significativos)
Object	Ob	Objeto genérico

## Definición de Controles

Para poder determinar el nombre de un control dentro de cualquier aplicación de tipo visual, se procede a identificar el tipo al cual pertenece y la función que cumple dentro de la aplicación.

## Prefijo para el Control

El prefijo del control será determinado mediante tres caracteres que estarán conformados por las consonantes más representativas del control, es así, por

ejemplo; el control Button, estará asociado al prefijo btn.

### Nombre descriptivo del Control

Formado por la descripción de la función que lleva a cabo el control, esta debe ser descrita en forma específica y clara.

Tabla. 16 *Detalle del nombre descriptivo para controles*

Tipo de control	Prefijo	Ejemplo
Label	lbl	LblNombre
TextBox	txt	TxtApellido
Button	btn	BtnLogin
RadioButton	rdo	RdoSeleccion
CheckBox	chk	chkRuta1
DropDownList	cmb	cmbDocumentos

### Declaración de variables, atributos y objetos

Se debe declarar una variable por línea.

Tabla. 17 *Detalle de como declarar variables, atributos y objetos*

Título	Descripción
Sintaxis	<b>[Tipo Variable] [Nombre de la Variable]</b>
Descripción	Las variables o atributo tendrán una longitud máxima de 30 caracteres. El nombre de la variable puede incluir más de un sustantivo los cuales se escribirán juntos. En caso de tener variables que puedan tener nombres iguales, se le agregará un número asociado (si está dentro de un mismo método será correlativo).
Observaciones	En la declaración de variables o atributos no se deberá utilizar caracteres como: <ul style="list-style-type: none"> <li>• Letra Ñ o ñ.</li> <li>• Caracteres especiales ¡ ^, #, \$, %, &amp;, /, (, ), ¿, ‘, +, -, *, {, }, [, ].</li> <li>• Caracteres tildados: á, é, í, ó, ú.</li> </ul>
Ejemplo	Public String nombre El ejemplo indica una variable o atributo que guardará un nombre.

### Declaración de Clases

Tabla. 18 *Detalle de como declarar clases*

Título	Descripción
Sintaxis	[Tipo] Class [Nombre de Clase]
Descripción	El nombre de las clases tendrá una longitud máxima de 30 caracteres y las primeras letras de todas las palabras estarán en mayúsculas. Tipo se refiere a si la clase será: Private, Public o Protected.
Observaciones	En la declaración de clases no se deberá utilizar caracteres como: Letra Ñ o ñ. Caracteres especiales ¡ ^, #, \$, %, &, /, (, ), ÿ, ‘, +, -, *, {, }, [, ], _. Caracteres tildados: á, é, í, ó, ú.
Ejemplo	Private Class Estudiante  Indica una clase Estudiante

### Declaración de Métodos

Tabla. 19 *Detalle de como declarar Métodos*

Título	Descripción
Sintaxis	nombreProcedim[(ListaParámetros)]
Descripción	El nombre del método constará hasta de 25 caracteres. La primera letra de la primera palabra del nombre será escrita en minúscula y las siguientes palabras empezarán con letra mayúscula.
Observaciones	En la declaración de métodos no se deberá utilizar caracteres como: Letra Ñ o ñ. Caracteres especiales ¡ ^, #, \$, %, &, /, (, ), ÿ, ‘, +, -, *, {, }, [, ], _. Caracteres tildados: á, é, í, ó, ú.
Ejemplo	Protected calcularNotas(String nota)  Indica un método calcularNota que recibe una variable por valor de tipo string al ámbito de la clase

### Declaración de Funciones

Tabla. 20 *Detalle de como declarar Funciones*

Título	Descripción
<b>Sintaxis</b>	[TipoDato] nombreFuncion[(ListaParámetros)]
	El nombre del objeto constará hasta de 25 caracteres, no es necesario colocar un nombre que indique la clase a la cual pertenece.
<b>Descripción</b>	La primera letra de la primera palabra del nombre será escrita en mayúsculas
	El tipo de dato de retorno se coloca al final y será obligatorio colocarlo.
<b>Observaciones</b>	En la declaración de objetos no se deberá utilizar caracteres como: Letra Ñ o ñ. Caracteres especiales ; ^, #, \$, %, &, /, (, ), & , ' , +, -, *, {, }, [, ], _. Caracteres tildados: á, é, í, ó, ú.
<b>Ejemplo</b>	Public int sumar(int A, int B)
	Indica una función que suma dos variables enteras

### Recomendaciones como no programar

Tabla. 21 *Recomendaciones de que no se debe hacer en la programación*

COMO NO PROGRAMAR ERRORES
Ignorar los mensajes de error
Código sin pensar
No copilar el código
No escribir comentarios
Ignorar las normas de presentación y programación
No realizar la documentación al finalizar
No utilizar el depurador y otras herramientas
No pedir ayuda
No aislar un problema
Cometer errores de ortografía
Copias de practica
Nunca dejes el trabajo para el final

## 5.02. Diseño de Interfaces de Usuario

El diseño de la interfaz de usuario es el proceso de determinar los distintos componentes, tanto de hardware como de software, sus características y su disposición, que se utilizarán para interactuar con una serie de usuarios determinados en un medio ambiente determinado.



Figura. 14 Diseño general de interface de usuario

La figura muestra la interface que el usuario ve al momento del ingreso al URL del sistema.

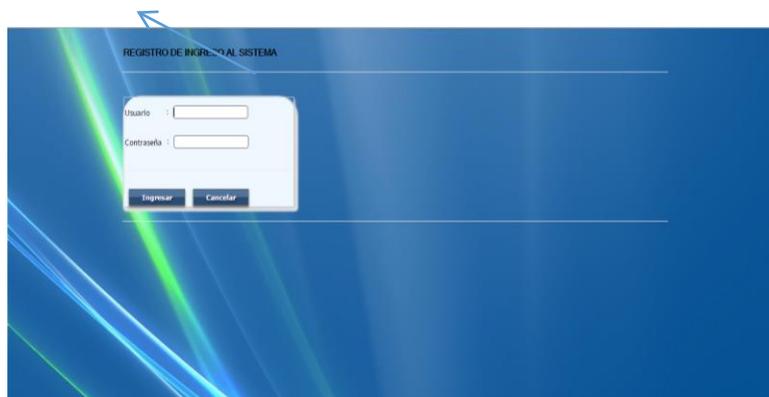


Figura. 15 Ingreso a los módulos

La figura muestra la ventana en donde se debe ingresar el usuario y contraseña.

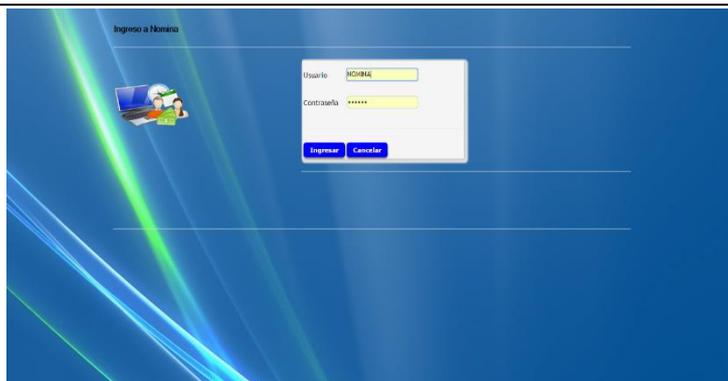


Figura. 16 Login de ingreso al módulo de nómina

La figura muestra el ingreso de contraseña del usuario.

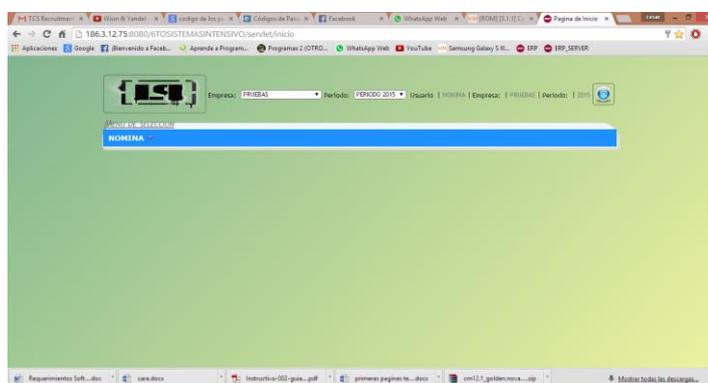


Figura. 17 Interfaz del usuario

La figura muestra la interfaz luego de ingresar la contraseña.

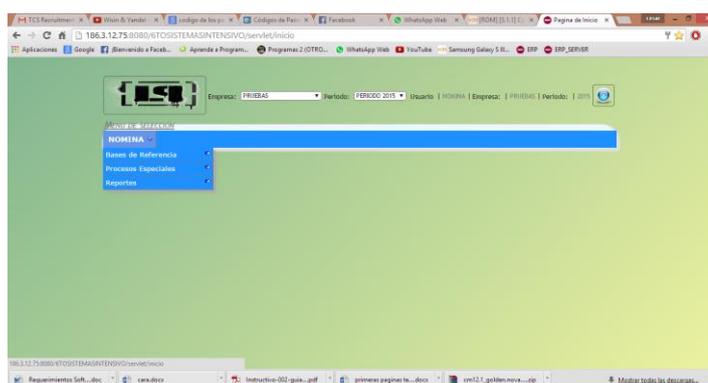


Figura. 18 Opciones de nómina

La figura muestra las opciones que despliega al seleccionar NÓMINA.

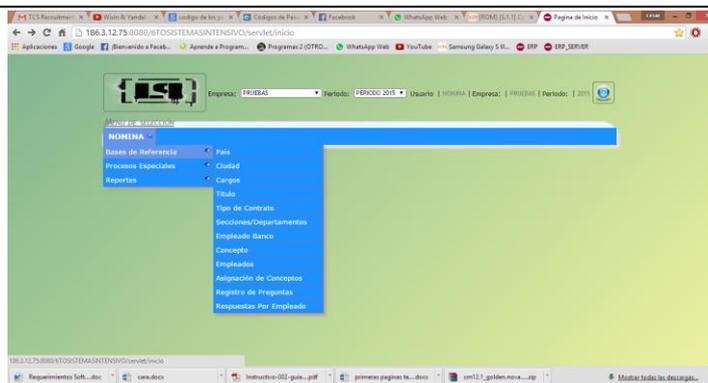


Figura. 19 Opciones base de referencia en nómina

La figura muestra las opciones al seleccionar Base de referencia en la opción NOMINA.

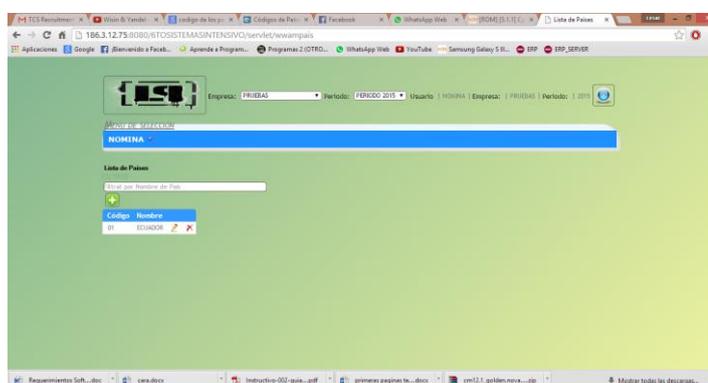


Figura. 20 Interfaz de ingreso de país

La figura muestra la interfaz de ingreso de un nuevo país.

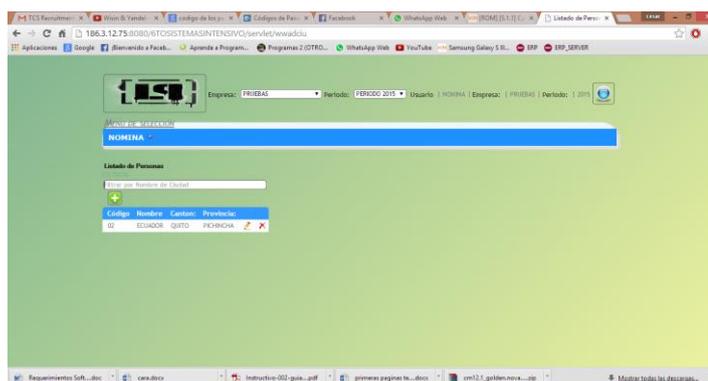


Figura. 21 Interfaz de ingreso de ciudad

La figura muestra la interfaz de ingreso de una nueva ciudad.

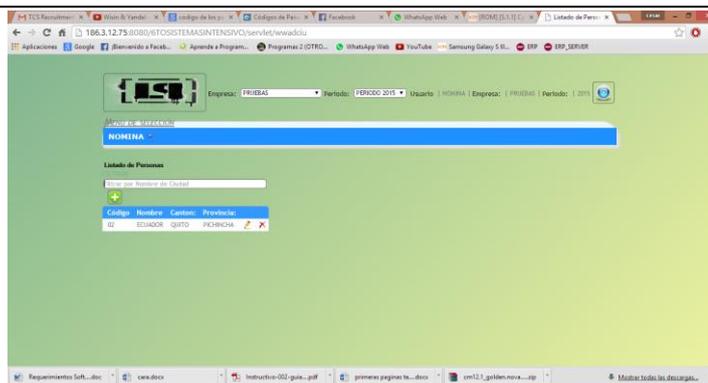


Figura. 22 Interfaz de ingreso de cargos

La figura muestra la interfaz de ingreso de un nuevo cargo.

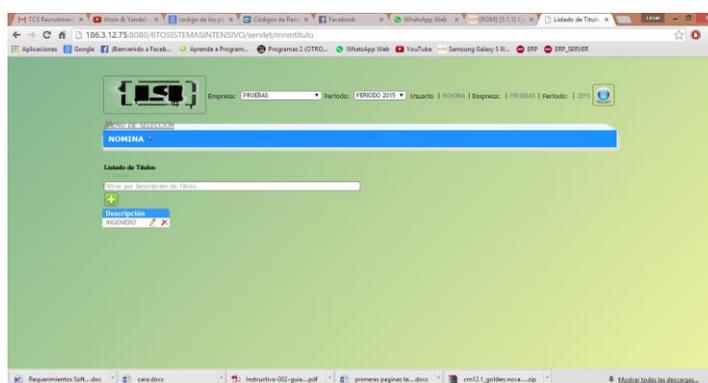


Figura. 23 Interfaz de ingreso de títulos

La figura muestra la interfaz para el ingreso de nuevos títulos.

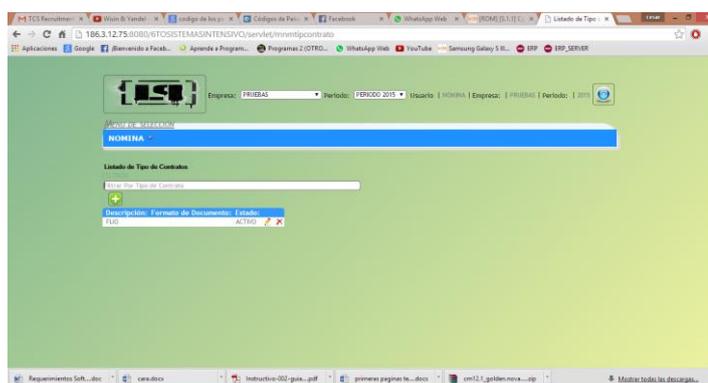


Figura. 24 Interfaz de ingreso de tipo de contrato

La figura muestra la interfaz para el de ingreso de los tipos de contrato.

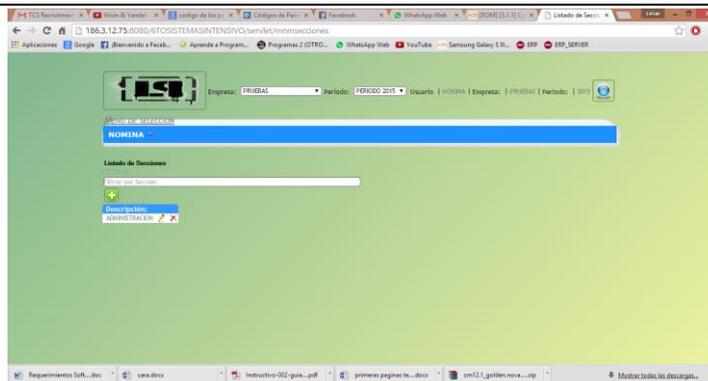


Figura. 25 Interfaz de ingreso de secciones y departamentos

La figura muestra la interfaz para el ingreso de secciones y departamentos.

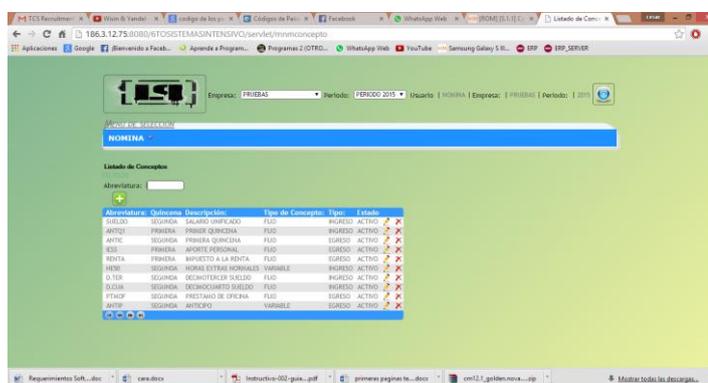


Figura. 26 Interfaz de ingreso de conceptos

La figura muestra la interfaz para el ingreso de nuevos conceptos.

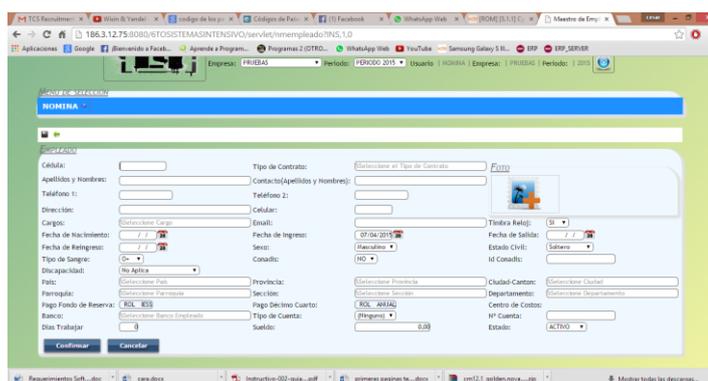


Figura. 27 Interfaz de ingreso de empleados

La figura muestra la interfaz para el ingreso de nuevos empleados.

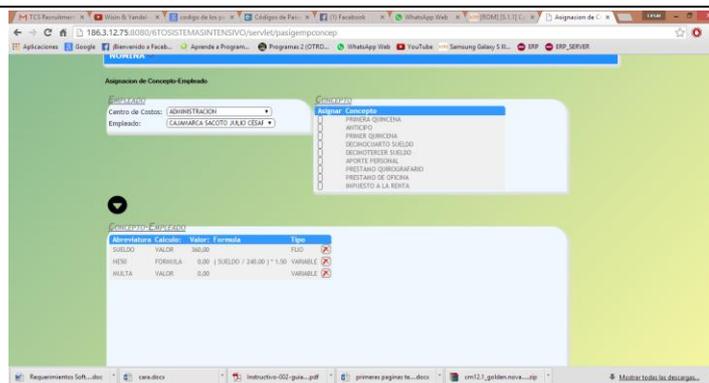


Figura. 28 Interfaz de asignación de conceptos

La figura muestra la interfaz para realizar la asignación de conceptos.

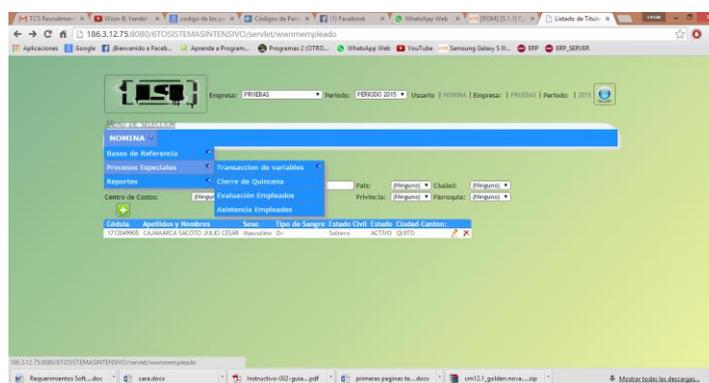


Figura. 29 Interfaz de opciones de procesos especiales

La figura muestra la interfaz para la selección de opciones en Procesos Especiales.

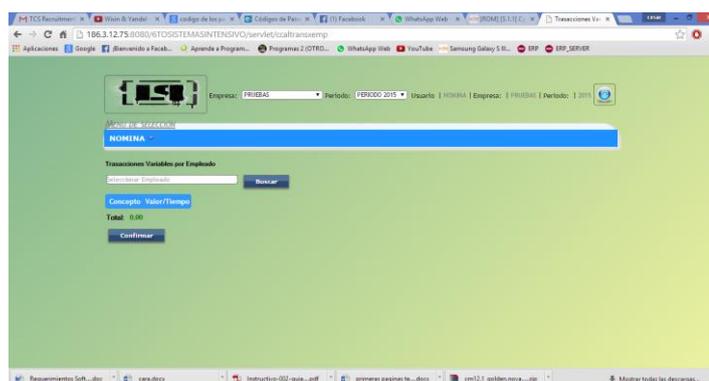


Figura. 30 Interfaz de transacciones de variable

La interfaz muestra la selección de una transacción variable.



Figura. 31 Interfaz de cierre de quincena

La figura muestra la interfaz en la que se realiza el cierre de quincena para ejecutar de cálculo de nomina

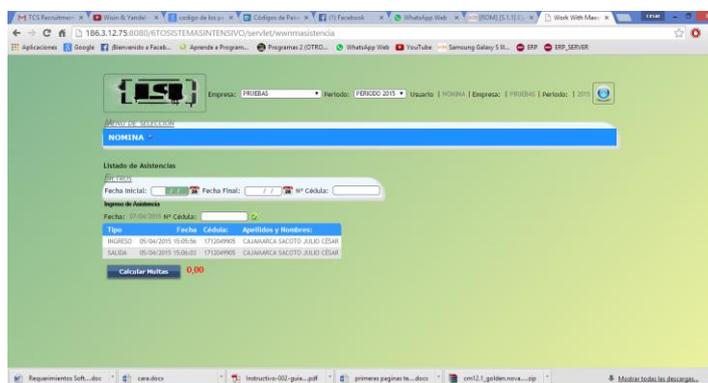


Figura. 32 Interfaz de asistencia del empleado

La figura muestra la interfaz en la que el usuario registra su asistencia (entrada y salida).



Figura. 33 Interfaz de selección de reporte

La figura muestra la interfaz para la generación de un rol de pago.



Tabla. 22 Prueba de interface de usuario (estándares)

Identificador de la Prueba:		PRU_UNI001
<b>Método a Probar</b>	<b>Interface</b>	
<b>Objetivo de la Prueba</b>	Examinar las posibles fallas en el manejo de la interface y corregirlos, revisar estándares para facilitar la navegación del usuario.	
<b>Datos de Entrada:</b>		
Datos Generales en los diferentes Formularios		
<b>Resultados Esperados</b>		
Encontrar errores que podrían detener el funcionamiento del sistema		
<b>Comentarios</b>		
El usuario ingresa sin problema al sistema.		
El sistema funciona con normalidad.		
El usuario domina la interfaz rápidamente.		

Tabla. 23 Pruebas de Reportes, resultados eficientes

Identificador de la Prueba:		PRU_UNI002
<b>Método a Probar</b>	<b>Reportes</b>	
<b>Objetivo de la Prueba</b>	Verificar que todos los resultados esperados en el proceso sean los adecuados y correctos.	
<b>Datos de Entrada:</b>		
Datos Generales en los diferentes Formularios		
<b>Resultados Esperados</b>		
Corregir errores posibles al generar reportes		
<b>Comentarios</b>		
Se realiza pruebas en todos los reportes encontrando errores únicamente de ortografía		

Tabla. 24 *Pruebas de compilación de Código*

Identificador de la Prueba:	PRU_UNI003
<b>Método a Probar</b>	<b>Pruebas de Código - Compilación</b>
<b>Objetivo de la Prueba</b>	Evaluar los resultados obtenidos y analizar los errores del código encontrados
<b>Datos de Entrada:</b> Compilación proceso paso a paso.	
<b>Resultados Esperados</b> Mantener el sistema en ejecución, corregir y evitar errores al compilar.	
<b>Comentarios</b> El código copila sin errores	

Tabla. 25 *Pruebas de Almacenamiento de datos en la Base*

Identificador de la Prueba:	PRU_UNI004
<b>Método a Probar</b>	<b>Almacenamiento de datos en la base</b>
<b>Objetivo de la Prueba</b>	Evaluar si los datos ingresados en los mantenimientos están siendo manejados de la manera adecuada.
<b>Datos de Entrada:</b> Datos generales en todos los formularios de Mantenimiento	
<b>Resultados Esperados</b> Obtener datos consistentes y coherentes.	
<b>Comentarios</b> Con esta prueba inclusive se verifica encriptación de contraseñas y que los valores sean reales al presentar en los diferentes reportes	

## 5.04 Especificación de pruebas de aceptación

Tabla. 26 *Detalle de pruebas de aceptación en la Gestión y control del empleado*

<b>Identificador de la Prueba:</b>	<b>PRU_ACE001</b>
<b>Caso de Uso</b>	<b>CU001</b>
<b>Tipo de Usuario</b>	<b>Administrador</b>
<b>Objetivo de la Prueba</b>	Probar el funcionamiento del proceso general de Gestión y control del empleado.
<b>Secuencia de Eventos</b>	Login de usuario, ingresar información del nuevo empleado, Ejecutar cambios, Eliminar.
<b>Resultados Esperados</b>	Que no tenga inconsistencias con respecto a datos guardados, validaciones y seguridad en la información.
<b>Comentarios</b>	Se realiza las pruebas en el sistema ingresando un nuevo empleado y realizando el mantenimiento a la tabla.
<b>Estado Aceptado/No aceptado</b>	Aceptado

Tabla. 27 *Detalle de pruebas de aceptación en la Administración de nómina*

<b>Identificador de la Prueba:</b>	<b>PRU_ACE002</b>
<b>Caso de Uso</b>	<b>CU002</b>
<b>Tipo de Usuario</b>	<b>Administrador</b>
<b>Objetivo de la Prueba</b>	Probar el funcionamiento del proceso de Administración de nómina
<b>Secuencia de Eventos</b>	Login de usuario, verificación de información de empleado, realiza proceso de nómina, revisa ingresos y egresos, cierra el pago de empleado, modificar, Guardar.
<b>Resultados Esperados</b>	Que no tenga inconsistencias con respecto a guardados, validaciones y seguridad en la información.
<b>Comentarios</b>	Se realiza las pruebas en el sistema ingresando registros nuevos, consulta y reportes.
<b>Estado Aceptado/No aceptado</b>	Aceptado

Tabla. 28 Detalle de pruebas de aceptación en la Elaboración y consulta de reportes.

<b>Identificador de la Prueba:</b>	<b>PRU_ACE003</b>
<b>Caso de Uso</b>	<b>CU003</b>
<b>Tipo de Usuario</b>	<b>Administrador</b>
<b>Objetivo de la Prueba</b>	Probar el funcionamiento del proceso de Elaboración y consulta de reportes.
<b>Secuencia de Eventos</b>	Login de usuario, ingresar a nómina, consultar reportes, Ejecutar cambios, Eliminar, Guardar. Login de usuario, ingresar bodegas, recursos, sacar reportes, cerrar sesión
<b>Resultados Esperados</b>	Que no tenga inconsistencias con respecto a guardados, validaciones y seguridad en la información.
<b>Comentarios</b>	Se realiza las pruebas en el sistema ingresando registros nuevos, eliminando, modificando, consultando y sacando reportes.
<b>Estado Aceptado/No aceptado</b>	Aceptado

### 5.05. Especificación de pruebas de carga

Estas pruebas de rendimiento se pueden realizar tanto en las plataformas de prueba del desarrollo como, opcionalmente, en la plataforma de producción del cliente. En cualquier caso, el resultado obtenido consiste en una serie de informes que reflejan el rendimiento del sistema en distintos escenarios.

Ha de tenerse en cuenta que en estas pruebas se presentan factores que pueden influir en los resultados obtenidos como la topología de red, la configuración de los servidores...

Estas pruebas no pretenden optimizar todos estos factores sino sólo medir el rendimiento de las aplicaciones entregadas en su ubicación establecida.

Los tipos de pruebas de rendimiento que habitualmente pueden ponerse en marcha son los siguientes:

**Prueba normal.** Permite establecer los tiempos medios de respuesta cuando sólo un usuario está conectado a la aplicación.

Esta prueba pretende establecer una referencia futura para posteriores comparaciones así como medir unitariamente el software entregado.

**Prueba con número mínimo de usuarios.** Se realizan las pruebas del sistema con el número de usuarios mínimos concurrentes establecido.

**Prueba con número máximo de usuarios.** Se realizan las pruebas del sistema con el número de usuarios máximo concurrentes establecido.

**Prueba de número máximo soportado de usuarios.** Se busca encontrar cuál es el límite del sistema. (Catarina, 2014)

Tabla. 29 *Detalle de un tipo de prueba de carga más baja*

Identificador de la Prueba:		PRCA001
<b>Tipo de Prueba</b>	Prueba normal (Prueba de Carga)	
<b>Objetivo de la Prueba</b>	Verificar el funcionamiento del sistema cuando sólo un usuario está conectado a la aplicación.	
<b>Descripción:</b>	Esta prueba pretende establecer una referencia futura para posteriores comparaciones así como medir unitariamente el software entregado.	
<b>Resultados Esperados</b>	Hacer que los procesos del sistema sean óptimos y tengan buenos tiempos de respuesta.	
<b>Comentarios</b>	El sistema funciona eficientemente con una carga baja de usuarios	

Tabla. 30 *Detalle de un tipo de prueba de carga con un número mínimo de usuarios*

<b>Identificador de la Prueba:</b>	<b>PRCA002</b>
<b>Tipo de Prueba</b>	<b>Prueba con número mínimo de usuarios</b>
<b>Objetivo de la Prueba</b>	Conocer si los procesos se están efectuando con normalidad y sin problemas ya con algunos usuarios.
<b>Descripción:</b>	Se realizan las pruebas del sistema con el número de usuarios mínimos concurrentes establecido.
<b>Resultados Esperados</b>	Validar la funcionalidad del sistema con un mínimo de usuarios logueados.
<b>Comentarios</b>	Con una cantidad mínima de usuarios el sistema responde eficientemente

Tabla. 31 *Detalle de un tipo de prueba de carga con un número máximo de usuarios*

<b>Identificador de la Prueba:</b>	<b>PRCA003</b>
<b>Tipo de Prueba</b>	<b>Prueba con número máximo de usuarios</b>
<b>Objetivo de la Prueba</b>	Establecer los tiempos de respuesta cuando una gran cantidad de usuarios están conectados a la aplicación.
<b>Descripción:</b>	Se realizan las pruebas del sistema con el número de usuarios máximo concurrentes establecido
<b>Resultados Esperados</b>	Hacer que los procesos del sistema sean óptimos y tengan buenos tiempos de respuesta cuando tengamos una gran cantidad de usuarios conectados.
<b>Comentarios</b>	El sistema funciona normalmente con una cantidad considerable de usuarios

Tabla. 32 *Detalle de un tipo de prueba de carga con un número máximo de usuarios ejecutándose hasta llegar al colapso, se requiere descubrir los límites.*

<b>Identificador de la Prueba:</b>	<b>PRCA004</b>
<b>Tipo de Prueba</b>	<b>Prueba de número máximo soportado de usuarios</b>
<b>Objetivo de la Prueba</b>	Se busca encontrar cuál es el límite del sistema.
<b>Descripción:</b>	Se realizan las pruebas del sistema con el número de usuarios máximo concurrentes establecido hasta llegar al colapso.
<b>Resultados Esperados</b>	Encontrar los límites que tiene el sistema en sobrecarga.
<b>Comentarios</b>	

### 5.06. Configuración del Ambiente mínima/ideal

Para casos de Aplicaciones desarrollados en un ambiente cliente servidor, un cuarto de máquinas envuelve los siguientes factores a tomar en consideración: localización, diseño, hardware, software, fuente de energía, temperatura, humedad, recuperación de desastres, seguridad. El planeamiento adecuado seguro de la localización y el diseño particular son los primeros pasos para crear un ambiente seguro.

Otras consideraciones a tener en cuenta son los insumos básicos con lo que la sala de servidores debe contar (enchufes, espacio, disponibilidad de red).

Un punto importante a considerar en cuanto al ambiente de la sala de servidores, es la humedad. Altos niveles de humedad pueden causar condensación y bajos niveles pueden causar electrostática. Adicionalmente, la sala de servidores debe contar con detectores de humo y agua, así como también UPS para proteger a los equipos de cortes/picos de electricidad.

---

El ambiente ideal para que nuestra aplicación en la web corra efectivamente sería:

- Contratar un Web Hosting confiable y que permanezca siempre en línea.
- Contratar en la oficina un servicio de Internet que sea siempre estable.
- Contratar un ancho de banda adecuada para tener una navegación óptima.
- Mantener la infraestructura de la oficina en buenas condiciones.

## Capítulo VI: Aspectos Administrativos

### 6.01. Recursos

#### 6.01.1 Recurso Humano

Tabla. 33 *Recursos Humanos*

RECURSOS HUMANOS			
Humano	Nombre	Actividad	Responsabilidad
Tutor	Lcdo. Wilson Nuñez	Guiar el desarrollo del proyecto.	Revisar el progreso del desarrollo del sistema y la documentación.
Lector	Ing. Hugo Heredia	Revisar el desarrollo del proyecto.	Asegurar que el desarrollo del proyecto elaborado cumple con las normas establecidas.

Tabla. 34 *Recursos Materiales*

RECURSOS MATERIALES			
MATERIAL	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
Computador	1	900	900
Hojas de papel bond	600	0.02	12
Esfero	3	0.5	1.5
Lápiz	1	0.3	0.3
Borrador	2	0.25	0.5
Grapas	100	0.01	1
Servicios Básicos	600	0.3	180
Servicios de internet	600	0.1	60
Alimentación	60	2	120
Transporte	60	2.5	150
Total			1425.3



---

## Capítulo VII: Conclusiones y Recomendaciones

### 7.01. Conclusiones

El presente proyecto nos ha permitido desarrollar una Aplicación Web de un sistema financiero aplicando los conocimientos de diseño y desarrollo de Software adquiridos a lo largo de nuestra carrera, profundizándonos en la investigación y análisis.

El desarrollo de este proyecto se cumplido satisfactoriamente, permitiendo automatizar los procesos de Nómina, consiguiendo optimizar tiempos y eficiencia en los empleados.

La aplicación web al contar con un diseño amigable, interactivo y de fácil acceso cumpliendo con los requerimientos del empleado permite al usuario interactuar de manera rápida y eficiente en el desenvolvimiento de sus tareas diarias.

Luego de realizar las pruebas en la aplicación se establece que logra generar los reportes y roles de manera eficiente y correcta, optimizando el tiempo de respuesta del personal del departamento.

Se consigue también mantener la información almacenada en una base de datos siendo esta de fácil acceso cuando el usuario lo requiera.

### 7.02. Recomendaciones

- Para una correcta utilización de la aplicación el usuario debe leer los manuales correspondientes.
  - Todos los empleados encargados de los procesos deben tener un usuario y contraseña de ingreso al sistema para lo cual deben ser registrados correctamente.
-

- Es importante capacitar al personal de nómina en el manejo de la aplicación para evitar demoras en los procesos.
- Planificar y ejecutar actualizaciones.
- Realizar registros completos de los usuarios

# ANEXOS

## ANEXO A0 1 Matriz de fuerzas T

ANÁLISIS DE FUERZAS T					
Situación Empeorada	Situación Actual				Situación Mejorada
Cierre parcial de la empresa	Inadecuado control y registro de ingresos y egresos de los empleados.				Pagos de nómina de empleados correctos y a tiempo.
Fuerzas Impulsadoras	I	P C	I	P C	Fuerzas Bloqueadoras
Coordinación entre departamentos administrativos de la empresa	2	5	4	4	No disponibilidad de información a tiempo
Registros completos y correctos de empleados	1	5	3	5	Registros incompletos de los empleados
Utilización correcta de equipos	1	4	4	3	Navegadores no actualizados
Acceso fácil e intuitivo al módulo de nómina	2	4	4	4	No conocimiento básico de informática
Personal capacitado en los procesos de nómina.	3	4	3	5	Inexistencia de planes de capacitación al personal de nómina.

## ANEXO A0 2 Entrevista

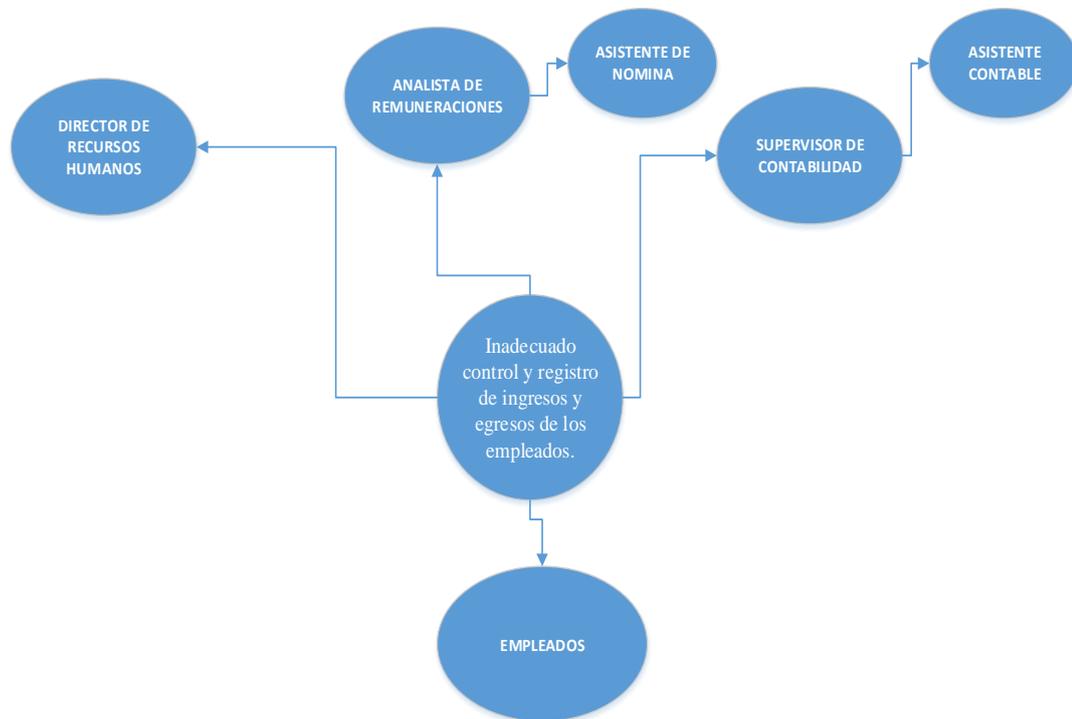
<b>DISEÑO ENTREVISTA</b>		
<b>identificador: 001</b>		
<b>Preguntas</b>	<b>Objetivos</b>	<b>Análisis posterior</b>
¿Cree usted que la gestión de nómina se realiza eficazmente?	Evaluar la situación actual de gestión de nómina	Los usuarios necesitan mayor control sobre la información de ingresos y egresos de los empleados. Quieren obtener datos confiables para los cálculos de nómina. El departamento encargado de nómina requiere tener la información a tiempo.
¿Cuáles son algunos de los problemas que experimenta al realizar el proceso de gestión de nómina?	Identificar los problemas actuales en la gestión de nómina.	Se necesita identificar empleados fijos de empleados por contrato para cálculos de nómina. Se requiere que el acceso a este módulo sea restringido por privilegios de usuario.
¿Quiénes tendrán acceso al módulo de nómina?	Obtener el listado de los usuarios que utilizara el módulo de nómina.	Departamentos que utilizarían el modulo: Recursos Humanos Remuneraciones Contabilidad
¿Qué sistema o aplicaciones utilizan hoy en día para nomina?	Enumerar los sistemas/aplicaciones que utilizan para gestión de nómina.	Los sistemas/aplicaciones con las que laboran actualmente son: Excel Word
Cómo considera usted al mejoramiento de gestión de nómina ¿no es necesario, si es necesario o le da lo mismo?	Evaluar la necesidad del mejoramiento de gestión nómina	Se requiere el inmediato mejoramiento de gestión de nómina para optimizar el trabajo en los departamentos involucrados

## ANEXO A0 3 Matriz de requerimientos

### MATRIZ DE REQUERIMIENTOS

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
<b>REQUERIMIENTOS FUNCIONALES</b>						
<b>RQF001</b>	Actualización inmediata de información del empleado.	Recursos Humanos	Alta	usuario	Análisis	Recursos Humanos, Remuneraciones, Contabilidad
<b>RQF002</b>	Control y registro de ingresos y egresos.	Remuneraciones	Alta	Usuario	Análisis	Departamento de Remuneraciones, Contabilidad
<b>RQF003</b>	Clasificación de tipos de empleados.	Remuneraciones	Alta	Usuario	Análisis	Departamento de Remuneraciones
<b>RQF004</b>	Acceso restringido de usuarios al módulo de nómina.	Recursos Humanos	Alta	Usuario	Análisis	Recursos Humanos, Remuneraciones, Contabilidad
<b>REQUERIMIENTOS NO FUNCIONALES</b>						
<b>RQNF001</b>	La aplicación debe funcionar en cualquier navegador	Recursos Humanos	Mediana	Sistema	En revisión	Recursos Humanos, Remuneraciones, Contabilidad
<b>RQNF002</b>	La interfaz debe ser sencilla, de fácil ingreso y recepción de datos	Remuneraciones	Mediana	Sistema	En revisión	Recursos Humanos, Remuneraciones, Contabilidad

## ANEXO A0 4 Mapeo de involucrados





## ANEXO A0 5 Matriz de Involucrados

Actores Involucrados	Intereses sobre el problema central	Problemas Percibidos	Recursos, Mandatos y Capacidades	Intereses sobre el Proyecto	Conflictos Potenciales
Director de Recursos Humanos	Controla toda la información generada por todos los empleados de la empresa así como de los procesos del departamento.	La información generada no tiene alguna base de datos donde almacenarse por la cual puede extraviarse o ser borrada.	Reglamentos internos del departamento establecidos por los empleados y aprobadas por las leyes del Ecuador.	Acceder a la información línea	Ninguno
Analista de remuneraciones	Controla todos los descuentos generados a todos los empleados de la empresa en las fechas determinada por el departamento.	La información de egresos que es generada no tiene control de seguridad por lo cual puede ser manipulada por otra persona el departamento.	Reglamentos internos del departamento establecidos por los empleados y aprobadas por las leyes del Ecuador.	Permite registrar los datos de ingresos y egresos generados por los movimientos de los empleados.	Ninguno
Supervisor de Contabilidad	Revisa que los procesos de descuentos establecidos por el departamento se realicen de una forma correcta.	Todo el proceso de revisión se hace de forma manual y luego se lo pasa a una hoja de cálculo de Microsoft Excel	Reglamentos internos del departamento establecidos por los empleados y aprobadas por las leyes del Ecuador.	Permite llevar o establecer un control de cada descuento a los empleados generados semanalmente.	Ninguno
Empleados	Genera solicitudes de reportes.	Los reportes solicitados no le son entregados a tiempo	Reglamentos internos del departamento establecidos por los empleados y aprobadas por las leyes del Ecuador.	Generación rápida y eficiente de los reportes solicitados	Ninguno

## ANEXO A0 6 Matriz de Análisis de Alternativas

Objetivos	Factibilidad de Lograse	Impacto en Género	Impacto Ambiental	Relevancia	Sostenibilidad	Total
	<b>(A-M-B)</b>	<b>(A-M-B)</b>	<b>(A-M-B)</b>	<b>(A-M-B)</b>	<b>(A-M-B)</b>	<b>puntos</b>
	<b>( 4-2-1)</b>	<b>( 4-2-1)</b>	<b>( 4-2-1)</b>	<b>( 4-2-1)</b>	<b>( 4-2-1)</b>	
	Los beneficios son mayores que los costos	Puede ser utilizado por cualquier genero	Fomenta el reciclaje	Responde a las expectativas de los beneficiarios	Fortalece a los usuarios involucrados	89
	Es aceptable y conveniente para los beneficiarios	Incrementa el nivel educativo de la mujer	Mejora el entorno social	Es una prioridad sentida por los beneficiarios	Fortalece la Organización local	0 a 10
	Existe tecnología adecuada para su realización	Lo podrán manejar diversidad de género, sin excepción	Mejora el entorno cultural	Beneficia a grupos de mayor carencia y vulnerabilidad	Fortalece la participación de los beneficiarios y población local	BAJA
	Se minimizará tiempo		Protege el uso de los recursos	Los beneficios son deseados por los beneficiarios		7 a 11
	El tiempo para el desarrollo es el adecuado		Favorece la educación ambiental	Los usuarios quedarán satisfechos		MEDIA BAJA
						12 a 16
						MEDIA ALTA
Puntos						17 a 25
						ALTA
	25	13	15	25	11	

## ANEXO A0 7 Matriz de Impactos de Objetivos

Objetivos	Impacto sobre el propósito	Factibilidad Técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categoría
Llevar una correcta información de los empleados	5	5	3	3	2	18	Alta
Aumentar la eficiencia y productividad en los empleados	5	5	3	3	2	18	Alta
Reducir tiempos en los procesos	5	4	4	3	2	18	Alta
Facilitar la elaboración de reportes	5	4	3	3	2	17	Media Alta
Obtener información rápida y fiable	5	5	2	2	2	16	Alta
<b>TOTAL:</b>	<b>25</b>	<b>23</b>	<b>15</b>	<b>14</b>	<b>10</b>	<b>87</b>	









## Manuales

### Manual de Instalación

#### Instalación de WAMPServer.

El proceso está compuesto de 11 sencillos pasos:

1°. Descargamos desde la web <http://www.wampserver.es/> el archivo de instalación.

En el momento en el que se realiza este artículo, la versión que tenemos disponible es la 2.2 de WAMPServer, lo que tras finalizar el proceso de instalación al que Windows nos tienen acostumbrados nos habrá instalado la versión 5.4.3 de PHP, 2.2.22 de Apache y 5.5.24 de MySQL.

2°. Iniciamos el proceso de instalación y nos encontramos la siguiente ventana.



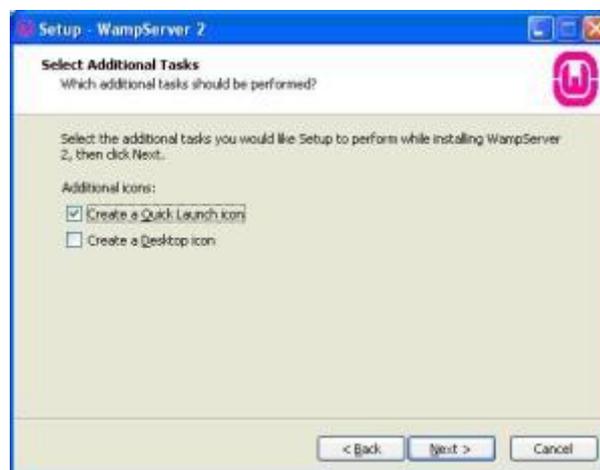
3°. Pulsamos sobre el botón "Next" y nos encontramos con la licencia de WAMPServer que tenemos que aceptar.



4°. En la siguiente ventana elegimos el directorio sobre el que queremos instalar nuestro servidor.



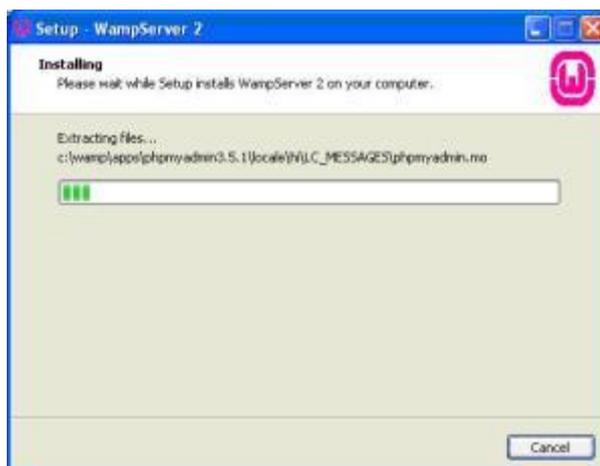
5°. Tras pulsar “Next” de nuevo, nos pide que seleccionemos que iconos queremos crear. Yo dejaré la opción que viene por defecto “Create a Quick Launch icon” que para los que no sepan lo que hace, nos crea una icono de acceso rápido al que podemos acceder desde el botón de Inicio (abajo a la izquierda de vuestra ventana Windows). La opción “Create a Desktop icon” nos crearía si fuera seleccionado un acceso directo al programa en el escritorio.



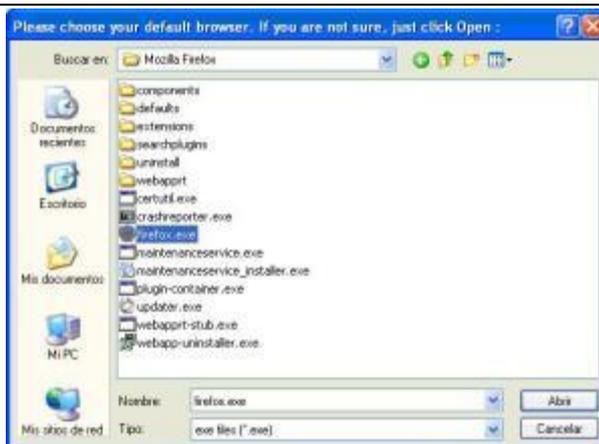
6°. Pulsamos “Next”, y en la siguiente ventana tan solo encontramos un resumen de lo indicado en los pasos anteriores. En mi caso se instalará en el directorio c:\wamp y me creará un acceso desde el menú Inicio.



7°. Pulsamos en botón “Install” y comienza el proceso de instalación como se muestra en la imagen siguiente.



8°. Os saldrá una ventana como la que mostramos en la que tenemos que buscar el directorio donde tenemos instalado nuestro navegador y seleccionarlo, si tenéis más de uno en vuestro Windows seleccionar el que utilizáis con mayor frecuencia.



9°. En caso de que os salte una alerta de seguridad de Windows solicitando permisos para Apache, se le concede el permiso y lo desbloqueamos.

En la imagen siguiente vemos como configurar los parámetros de SMTP, en el que tan solo escribimos "localhost" y un correo electrónico.



10°. Pulsamos "Next" y nos encontramos con la última ventana del proceso de instalación de WAMPServer, en el que marcaremos la casilla "Launch WampServer 2 now" si deseamos que se nos ejecute automáticamente tras finalizar la instalación, y pulsamos "Finalizar".



**11°.** Ya tenemos instalado correctamente la infraestructura necesaria para poner en funcionamiento nuestras páginas web's de forma local.

Accediendo desde el navegador a la siguiente dirección <http://localhost> nos encontramos con la siguiente imagen que nos muestra que WAMPServer esta correctamente instalado y funcionando.

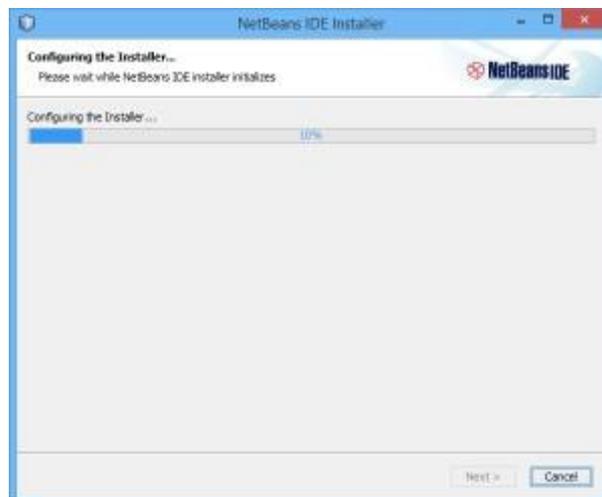


## INSTALACIÓN DE NETBEANS (IDE).

1. Para comenzar la instalación tendremos que buscar el ejecutable descargado (se guardara en la carpeta predeterminada de su buscador) y pulsar doble click sobre el ejecutable. Una vez comenzado, aparecerá una ventana que mostrara el estado de la instalación. En este paso no hay que hacer nada, solo esperar.



2. Tras un rato de espera, se abrirá algo parecido al primer paso, en este paso tampoco tendremos que hacer algo, solo esperar hasta que finalice.

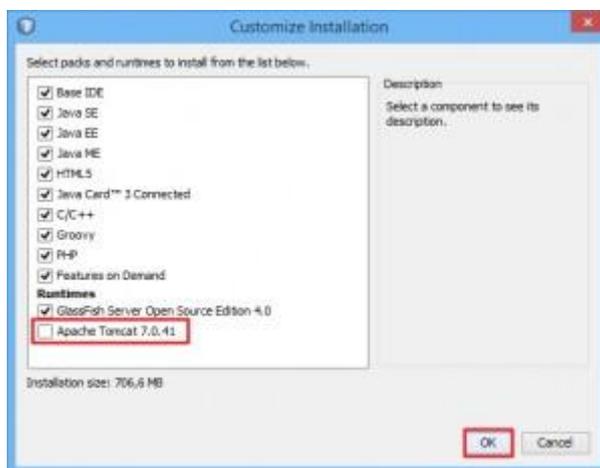


3. Una vez haya terminado, le aparecerá una ventana, en aquella ventana podremos elegir si deseamos personalizar nuestra instalación y elegir lo que instalar. Para hacer

eso pulsaremos sobre "Customize...".



4. Según lo que deseemos instalar lo seleccionaremos o no, en nuestro caso elegiremos instalar Apache Tomcat con la misma instalación. Lo seleccionamos y pulsamos "Ok".



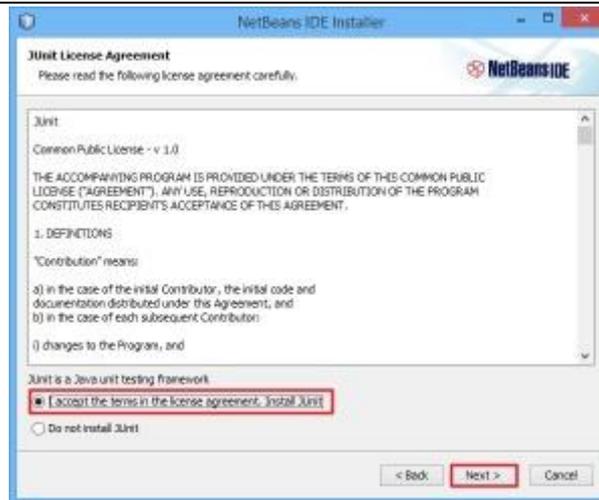
5. Al pulsar "Ok" nos volverá a mostrar la ventana #3. Para continuar con la instalación pulsaremos "Next >".



6. En la siguiente ventana nos mostrara los acuerdos de licencia, una vez leídos, marcaremos “I accept the terms in the license agreement” y pulsaremos “Next >”.



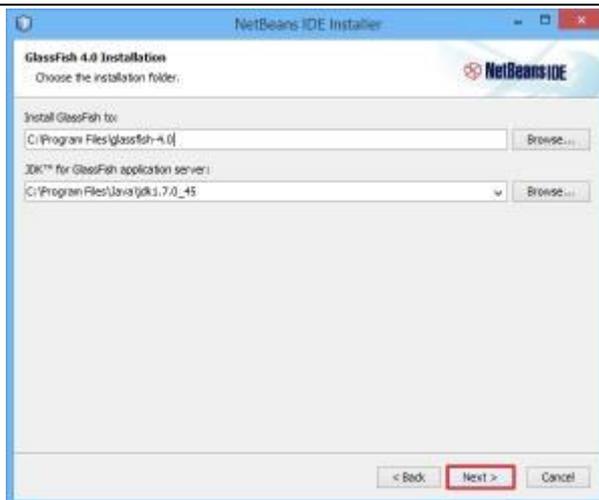
7. En la siguiente ventana nos aparecerán los acuerdos de licencia que tiene la aplicación “Java Unit”. Volvemos a marcar “I accept the terms in the license agreement. Install JUnit” y a continuación pulsaremos “Next >”.



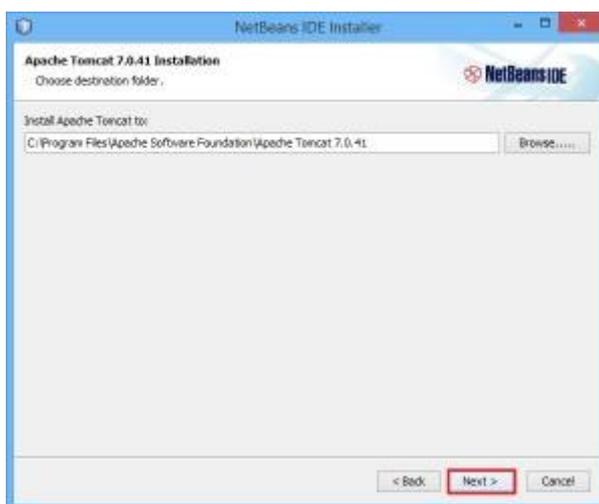
8. Si queremos cambiar el lugar de instalación este será el momento ( si usted lo desea ), en nuestro caso, lo instalaremos donde NetBeans sugiere instalarlo. Y pulsaremos “Next >” sin realizar ningún cambio.



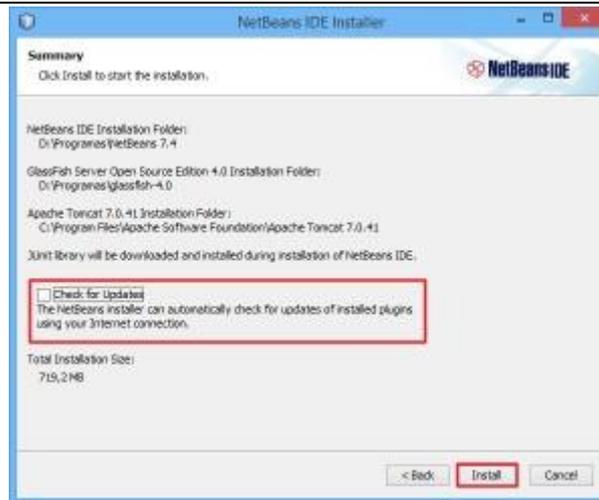
9. Lo mismo haremos con “GlassFish”. No cambiaremos nada y únicamente pulsaremos “Next >”.



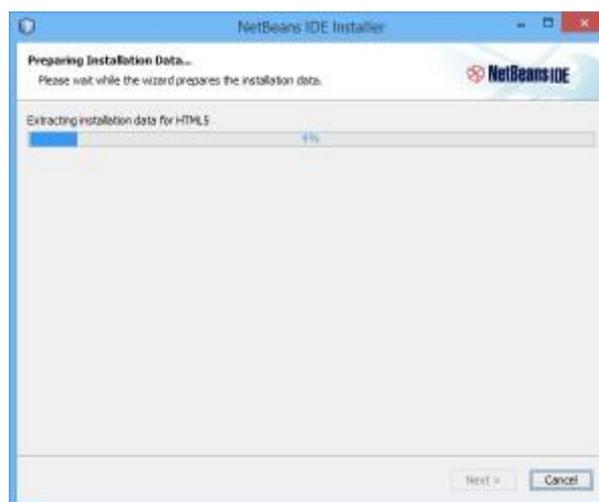
10. Por último, haremos los mismos pasos con “Apache”.



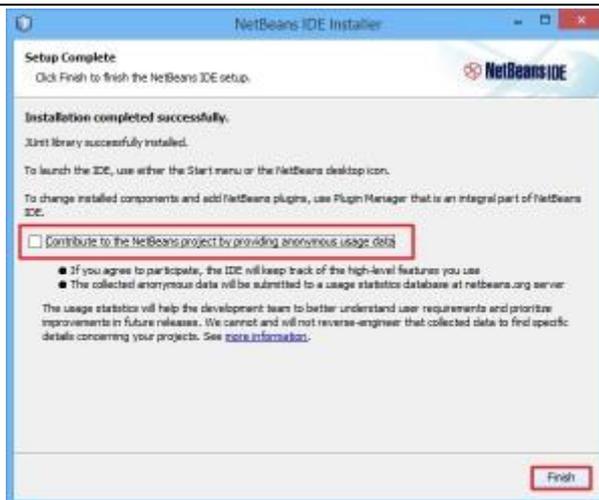
11. Antes de comenzar la instalación final, nos preguntara si deseamos que NetBeans busque actualizaciones automáticamente por nosotros. Según lo que le convenga marcaran o no la casilla “Check for Updates” ( Buscar actualizaciones ). Una vez decidido, pulsaremos “Install”.



12. Tras comenzar la instalación final, veremos una barra de instalación, como con cualquier otro programa.



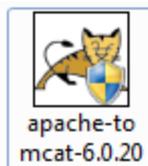
13. Una vez finalizada la instalación, nos aparecerá una ventana en la que NetBeans nos pregunta si deseamos contribuir con datos anónimos del programa para mejorar su uso y estadísticas. Según como usted deseé, lo dejara marcado o no. Y por último pulsaremos "Finish".



14. Y ya podrá comenzar a disfrutar de NetBeans

## Instalar el Apache Tomcat 6.x

Ejecutaremos para la instalación.



Icono de instalador del Tomcat 6.x

Nos da la bienvenida al instalador, le damos clic en NEXT:



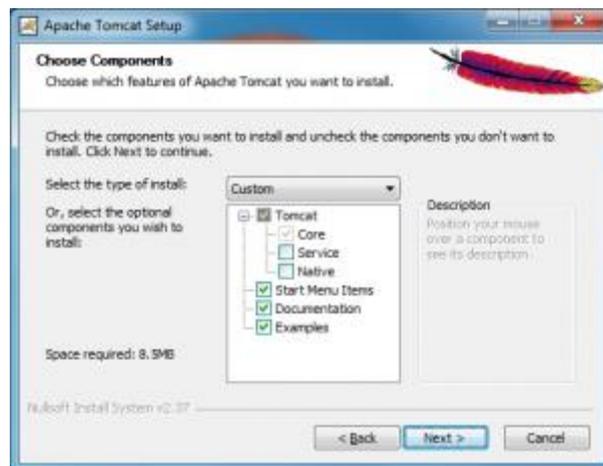
Bienvenida del instalador de Tomcat

Nos presenta el acuerdo de licencia de uso para APACHE TOMCAT, le damos clic en I AGREE:



Acuerdo de Licencia de uso de Tomcat

Nos pregunta, que todo va a instalar del TOMCAT, por default NO incluye la opción de SERVICE (que solo debemos de activar si queremos que cuando arranque windows, arranque el tomcat,, útil, si ya lo estamos ocupando en producción, o un ambiente de desarrollo compartido), tampoco incluye la opción de NATIVE (que instala el TOMCAT por medio de DLLs, que supuestamente brindan al TOMCAT un mejor desempeño, si estamos en desarrollo, pues esto no interesa tanto, pero si estamos en un ambiente de producción, seamos sinceros, lo mejor seria instalarlo en UNIX, porque JAVA en WINDOWS no experimenta TODO SU POTENCIAL), la última opción que no viene por default, es la de los EXAMPLES (estos son ejemplos de código, para hacer nuestros primeros experimentos en TOMCAT, es igual si lo palomean o no), le damos clic en NEXT:



### Opciones de instalación de Tomcat

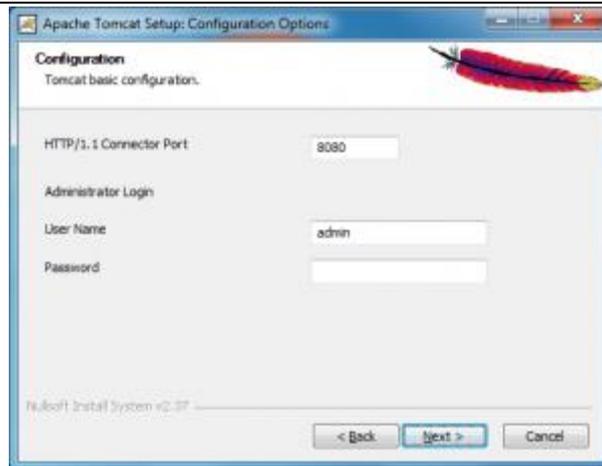
Nos da a escoger una carpeta para la instalación del TOMCAT. Y aquí viene el supertruquisimo, si escoges la carpeta por default (dentro de archivos de programa), NO PODRAS LEVANTAR SERVICIOS EN WINDOWS VISTA O WINDOWS 7, ya que acorde a la posición de Microsoft, todo eso que podíamos hacer en XP, como que no les gustaba más. Ahora escogeremos una carpeta en C: (raíz), para poder

levantar los servicios SIN PROBLEMAS, si no existe crearemos una (tal como lo hicimos con el EasyPHP:<http://profesor.antonio.com.mx/?p=92>), y le damos clic en NEXT:



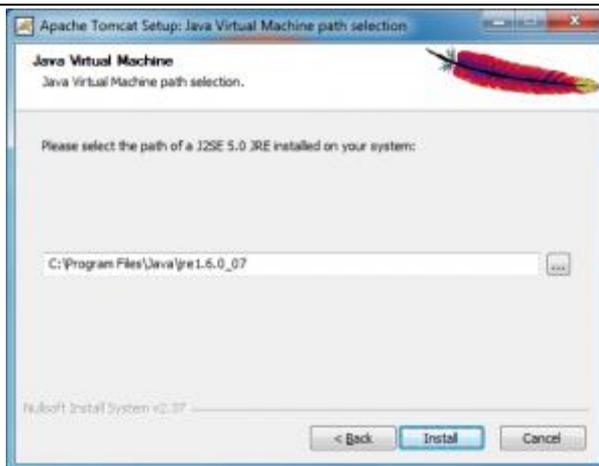
#### Directorio de instalación del Tomcat

Primero nos pregunta el puerto en el cual levantara el servicio del TOMCAT, por default tiene el 8080, pero podemos ocupar otros puertos como el 80, 82 u 85, dependiendo de cual tenemos libres, también nos pide el password del Admin, para poder dar de alta y baja servicios o el mismo TOMCAT de manera remota. Cabe aclarar que si estamos en desarrollo, pues eso resulta efímero. Si estamos en producción, si hay que ponerle password, algo rudo, porque desde el Admin podemos BORRAR TODO UN PROYECTO del mismo, y NO LO MANDA A LA PAPELERA, abusados. La recomendación es ponerle password, pero remover la carpeta del entorno y solo pegarla bajo petición, pero es MUY PELIGROSA dicha carpeta. Le damos clic en NEXT:



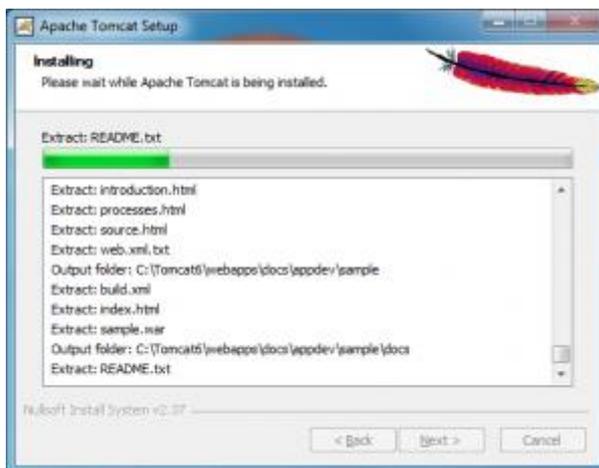
### Password del Admin para el Tomcat

Nos presenta la ruta del JRE que va a ocupar la instalación. Aquí hay que estar muy abusados, ya que una GRAN DESVENTAJA del java, es que es muy celoso de su JVM (JAVA VIRTUAL MACHINE), Si instalaste y compilaste bajo “equis” versión, tu aplicación es probable que se haya quedado casada con esa versión 😞 Lo cual dificulta mucho esto, ya que hay algunos programas que cuando se instalan, instalan su propia JVM, SI, cada uno instala su propia JVM, y se vuelve un gran batidillo de JVM. ¿Como saber cuántas máquinas virtuales tengo? Sencillo, dentro del panel de control, está la opción de JAVA, ahí puedes ver cuantas tienes, de preferencia es lo mejor tener 1 SOLA. Ya que de otra forma el TOMCAT, puede alegar que la JVM no esta compilando, le damos clic en INSTALL:



Ruta del JRE de JAVA para usar por el Tomcat

Nos presenta el avance de la instalación:



Progreso de instalación del Tomcat

Nos presenta la imagen del éxito de instalación del TOMCAT, pero antes de ejecutarlo, todavía hay que hacer algunas cosas con JAVA (ni modo, por eso no me gusta JAVA), damos clic en FINISH:



Fin de la instalación del Tomcat

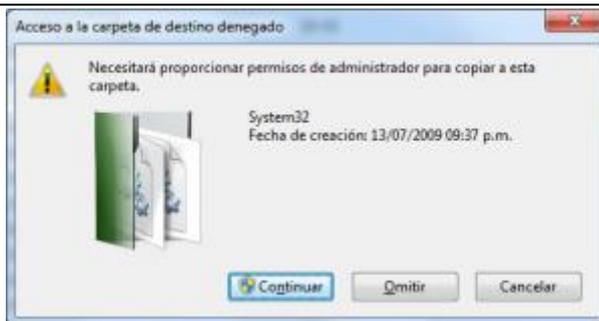
Nos iremos a la carpeta donde esta instalada la JVM de JAVA, y dentro de la carpeta JDKxxx, estara la carpeta BIN, ahi buscaremos y copiaremos el archivo:

**msvcr71.dll**



Archivo msvcr71.dll para el Tomcat

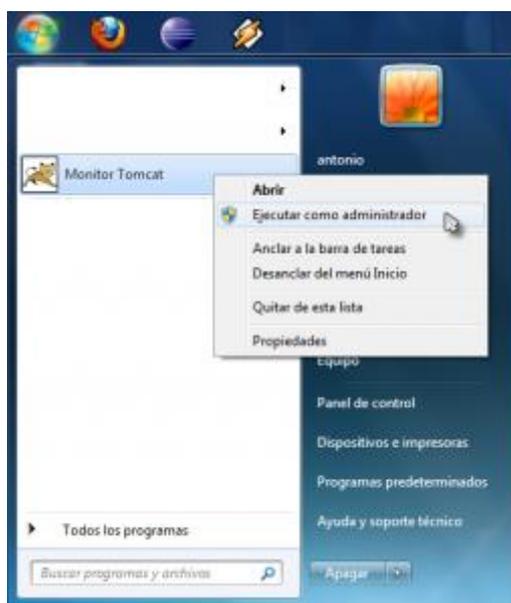
Dicha DLL, la pegaremos dentro de windows, en system32, es la solución más rápida, otra, es que entremos a las variables del sistema, y extendamos la búsqueda de dll hasta la carpeta especificada de JAVA, debe ser en las variables de sistema, no de usuario, etc. Nos va a pedir la verificación humana para pegar el archivo:



Confirmación humana para pegar el archivo

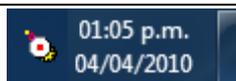
Ahora viene otro supertruquisimo, que es como levantar el TOMCAT a capricho en WINDOWS VISTA O WINDOWS 7, ya que de buenas a primeras, Microsoft, NO DEJA, por lo que ya platicamos anteriormente.

Debemos buscar el icono del MONITOR TOMCAT, dar un clic derecho y escoger EJECUTAR COMO ADMINISTRADOR, nos pedirá la confirmación humana:



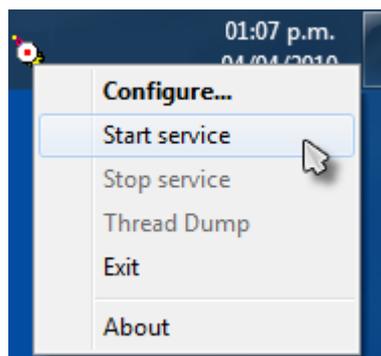
Iniciar como Administrador de windows el Tomcat

Entonces, se colocara un iconito de pluma con un CUADRITO ROJO en la barra de tareas, que indica que el TOMCAT, esta DETENIDO:



Icono del Monitor del Tomcat (apagado el servicio)

Para INICIAR el servicio del TOMCAT, daremos clic derecho sobre el icono, y escoger la opción de STAR SERVICE (obvio):



Iniciar el servicio del Tomcat

Nos dirá, que el TOMCAT, está TRATANDO DE INICIAR, en este punto, tal vez te pueda salir un mensaje del firewall de Windows o del firewall de tu antivirus, hay que darle permisos:



Iniciando el servicio del Tomcat

Si todo lo hicieron bien, pues verán que el iconito de la pluma cambio con un TRIANGULO VERDE en la barra de tareas, que indica que el TOMCAT, esta INICIADO el servicio:

01:23 p.m.  
04/04/2010

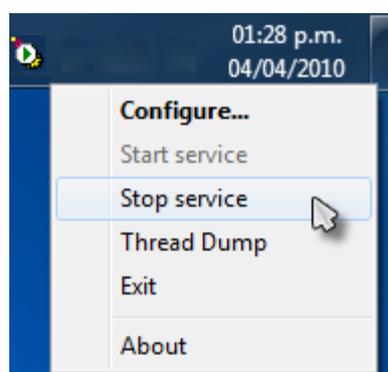
Icono del monitor de Tomcat (iniciado el servicio)

Si utilizaste todos los parámetros que mencionamos, pues podemos entrar al navegador la dirección del TOMCAT: <http://localhost:8080/> y ver el “gatito”:



Página principal del Apache Tomcat

Para DETENER el servicio del TOMCAT, daremos clic derecho sobre el icono, y escoger la opción de STOP SERVICE (obvio):



Detener el servicio del Tomcat

Nos dirá, que el TOMCAT, esta TRATANDO DE DETENER el servicio:



Aviso de apagado del servicio del Tomcat

Aquí hay algo importante que decir, como el TOMCAT es un servicio, y para evitar código malicioso, tiene la política de cuando inicia el servicio, carga todas las librerías (.CLASS o .JAR) que se encuentren en él. Si agregas un .CLASS o un .JAR, tendrás que DETENER el TOMCAT y volverlo a INICIAR, para que las reconozca, esto puede ser algo molesto, sobre todo en ambientes de producción, donde al DETENER el TOMCAT, tumbas a todos los usuarios conectados, algunas empresas de hosting, hacen esto de manera automática en la madrugada, si contratas un outsourcing de hosting, tienes que ver si ellos hacen esto, o te habilitan esa opción de manera remota. OJO con eso.

Ahora veamos cómo está la carpeta del TOMCAT:



Carpeta conf para configurar el inicio del Tomcat

Dentro del TOMCAT, hay una carpeta: CONF, ahí el archivo más importante es el de:

**server.xml**

Este archivo, nos sirve para configurar el TOMCAT, si corre en LOCALHOST, en una IP dentro de una Intranet, o en una IP Homologada, así como un DOMINIO o SUBDOMINIO, hay que abrir el archivo, modificarlo, detener e iniciar el TOMCAT, para reconocer cualquier cambio.

Aquí la carpeta importante es WEBAPPS, cada carpeta dentro de esta, representa para el TOMCAT un PROYECTO, por ejemplo si tenemos el proyecto "practicaflex", con una carpeta de igual nombre dentro de WEBAPPS, la ruta para acceder a él, desde web, sería: <http://localhost:8080/practicaflex/>

Dentro de TODO PROYECTO en TOMCAT, debemos tener una carpeta que se

**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.**

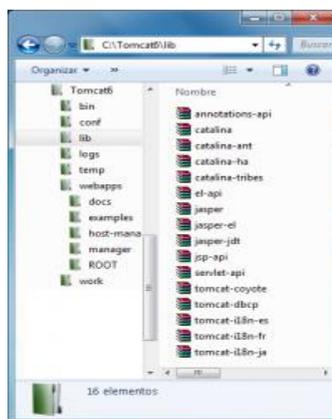
llama WEB-INF Es la UNICA carpeta que se estila en MAYUSCULAS, y es la UNICA carpeta que no es accesible mediante el navegador, inclusive al Administrador del TOMCAT.

Dentro de WEB-INF, debemos tener 2 carpetas siempre:

LIB: Que es una carpeta para poner archivos JAR que se vuelven EXCLUSIVOS del proyecto en donde se encuentran.

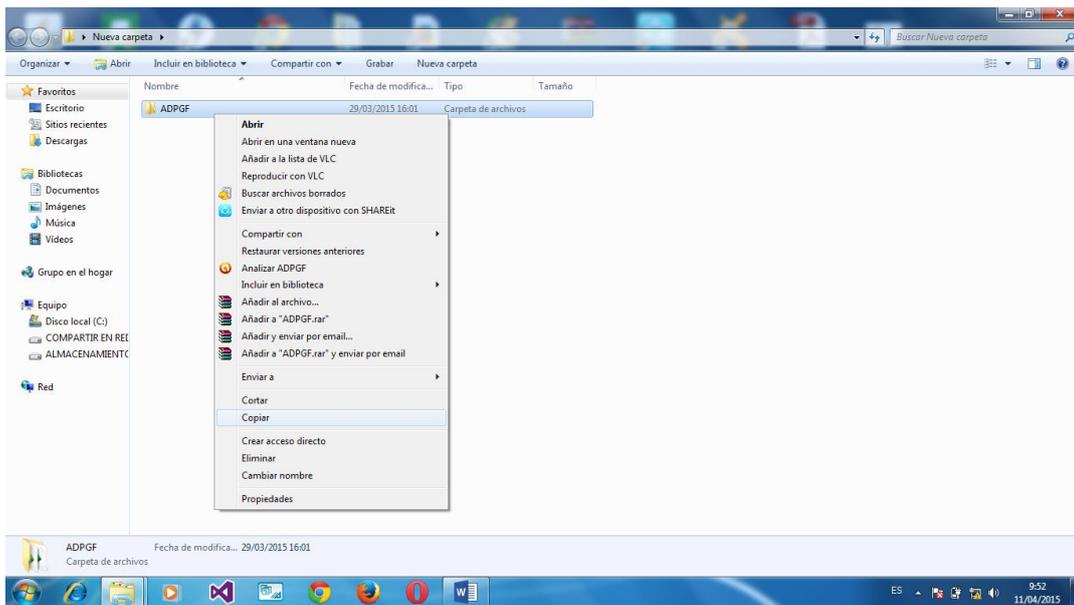
CLASSES: Que es una carpeta para poner los archivos CLASS que se vuelven EXCLUSIVOS del proyecto en donde se encuentran.

Si queremos tener archivos JAR que se van a ocupar en TODOS LOS PROYECTOS DEL TOMCAT, pues debemos colocarla en la carpeta LIB que se encuentra directamente bajo la carpeta raíz del TOMCAT:



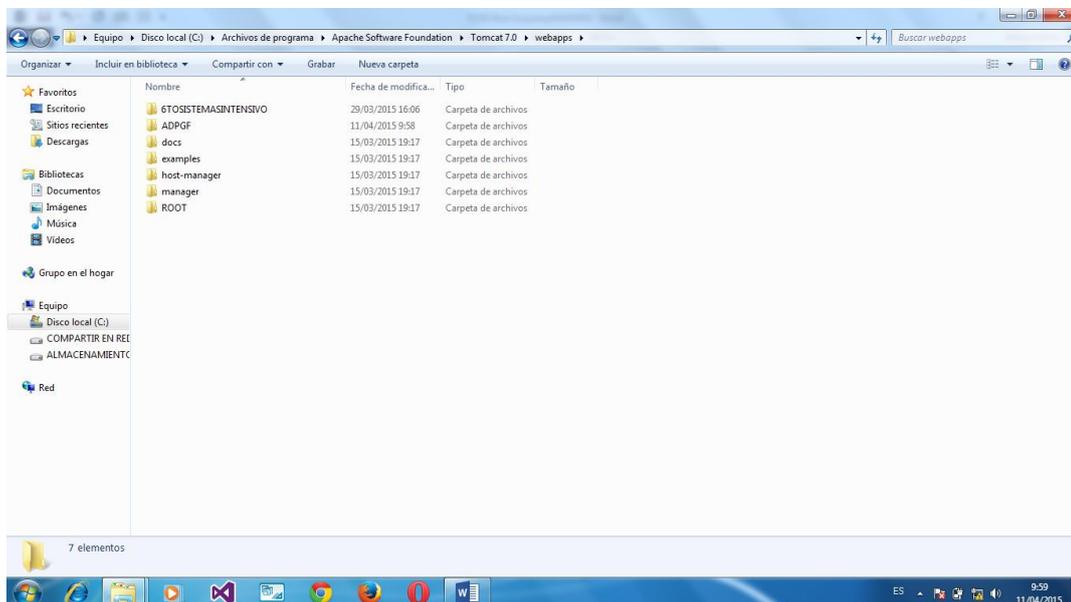
Carpeta lib para colocar los JAR compartidos del Tomcat

Un buen ejemplo serían los JDBC para conectarlos con nuestra base... u otros que es necesario tenerlos "a la mano" de cualquier proyecto de TOMCAT.



## **Instalar la aplicación ADPGF (Administración De Procesos de Gestión Financiera).**

Ya después de haber instalado los complementos anteriores, tenemos que copiar la carpeta ADPGF.



Debemos dirigirnos a C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps y pegar la carpeta ADPGF.

Y listo ya está alojada la aplicación de forma local.



**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.**

## Manual de usuario

Automatización de los procesos financieros mediante un sistema integrado de gestión módulo

Nomina

### Ingreso al sistema

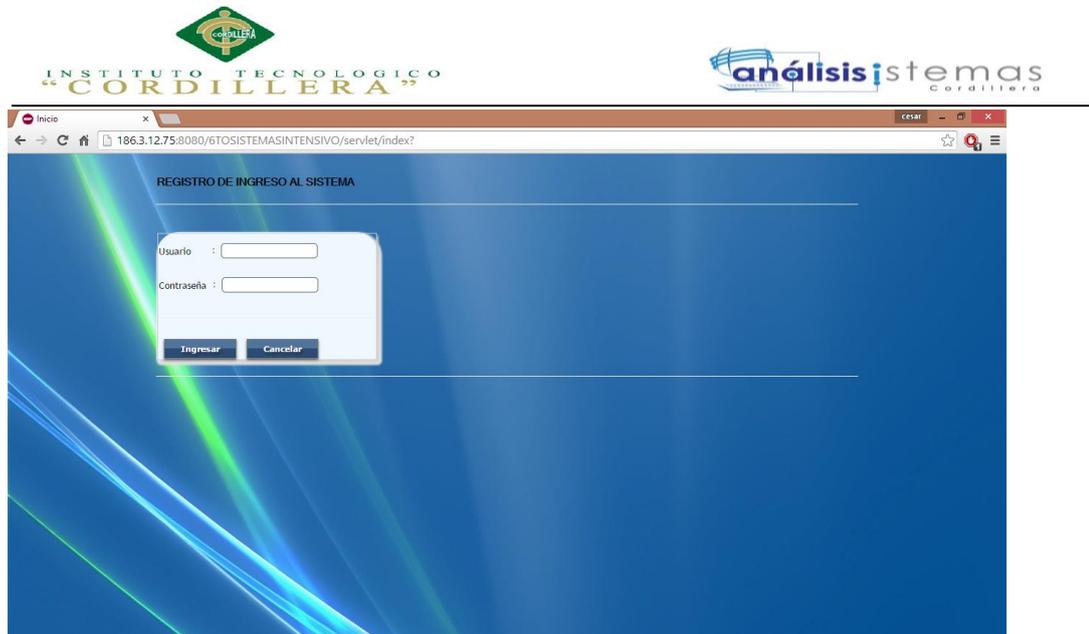
El ingreso al sistema se realiza desde un equipo con conexión a internet utilizando cualquier navegador e ingresando el link en donde se encuentra la aplicación web.

Url: <http://186.3.12.75:8080/6TOSISTEMASINTENSIVO/servlet/erp>



### Login de Usuarios

Presionando ingresar nos dirige a la siguiente ventana de menú de inicio.

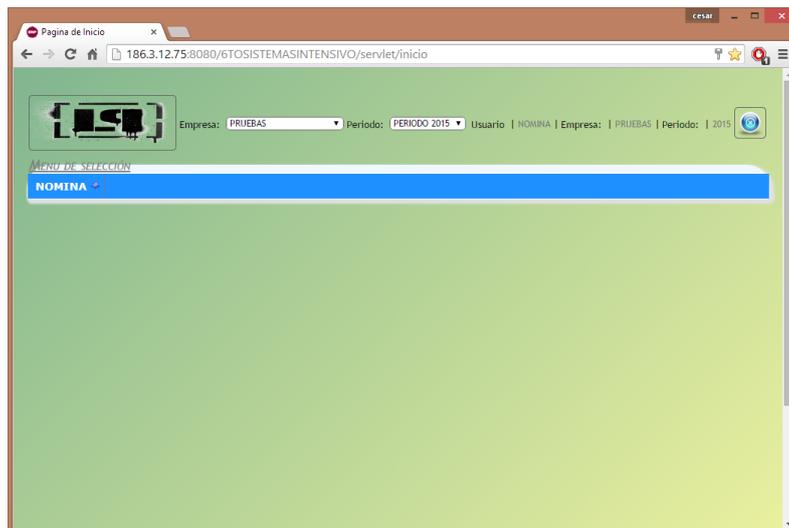


Para realizar el login se debe ingresar un usuario y contraseña proporcionado por el Administrador del sistema, esto no direcciona al menú del módulo nómina.

Para el módulo de Nomina el usuario y el password son

Usuario: NOMINA

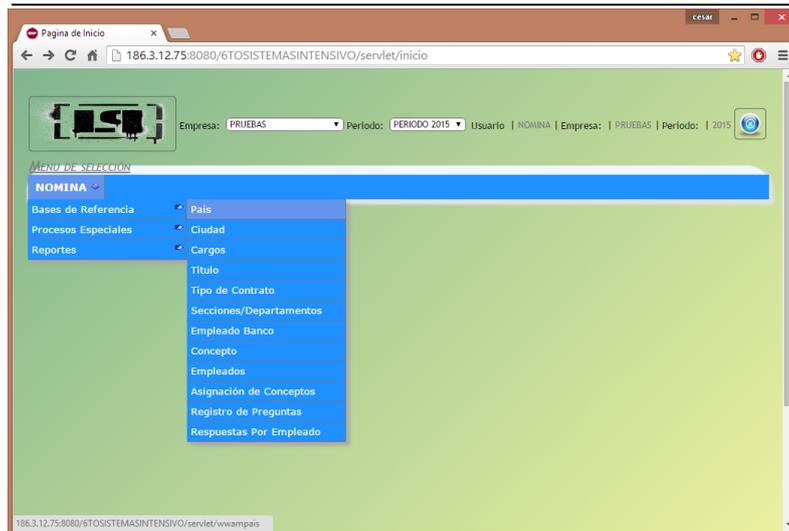
Pass: NOMINA



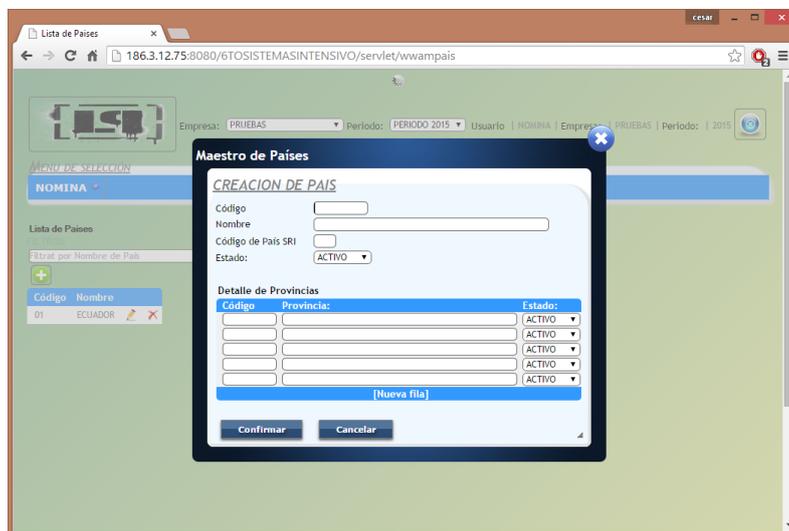
En este menú del módulo nómina se encontrara las diferentes opciones para realizar el proceso de generación de reportes y roles.

### Creación de país

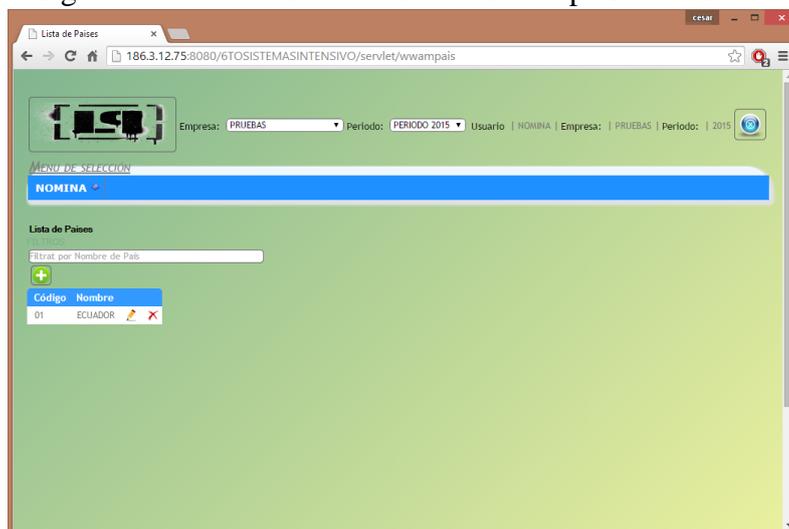
En el menú de selección seleccionamos NOMINA luego Bases de Referencia, por ultimo presionamos en País.



Al presionar País nos muestra el formulario para la creación de un nuevo País.



Llenamos el formulario con los datos solicitados para un nuevo país y nos mostrara la siguiente ventana con la información del país creado.



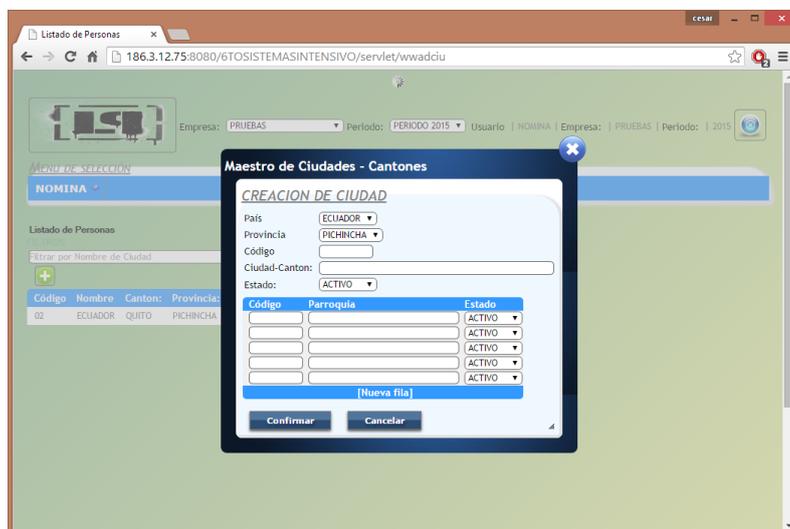
**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.**

En esta ventana observamos y podemos seleccionar el país creado.

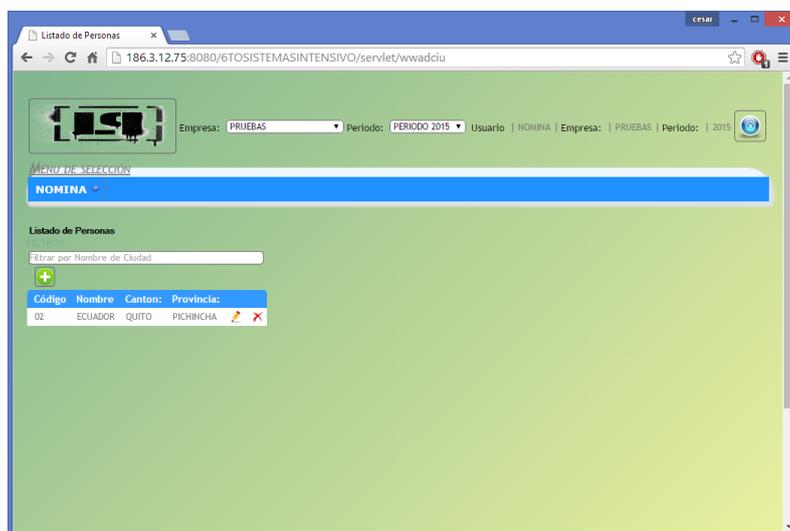
### Creación de Ciudad

En el menú de selección seleccionamos NOMINA luego Bases de Referencia, por último presionamos en Ciudad.

Y nos mostrará el formulario de creación de Ciudades.

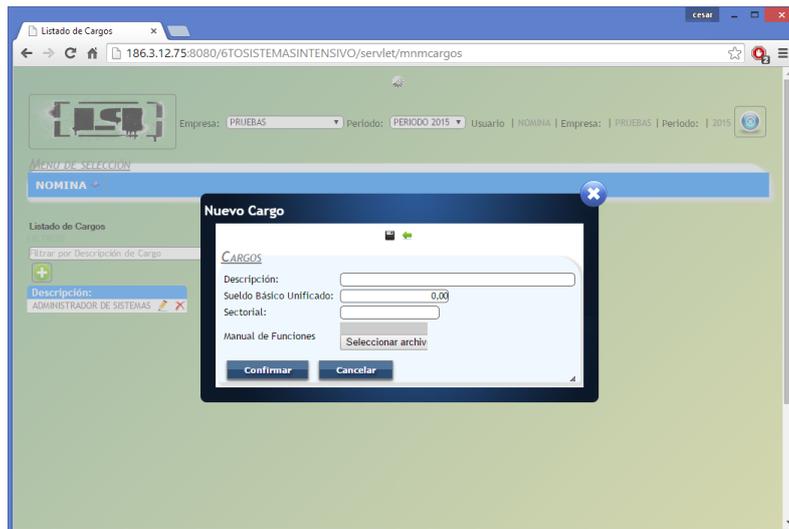


Se ingresan los datos solicitados en el formulario y se presiona confirmar, se crea la ciudad y nos mostrará la ventana en la que se puede seleccionar la ciudad creada.

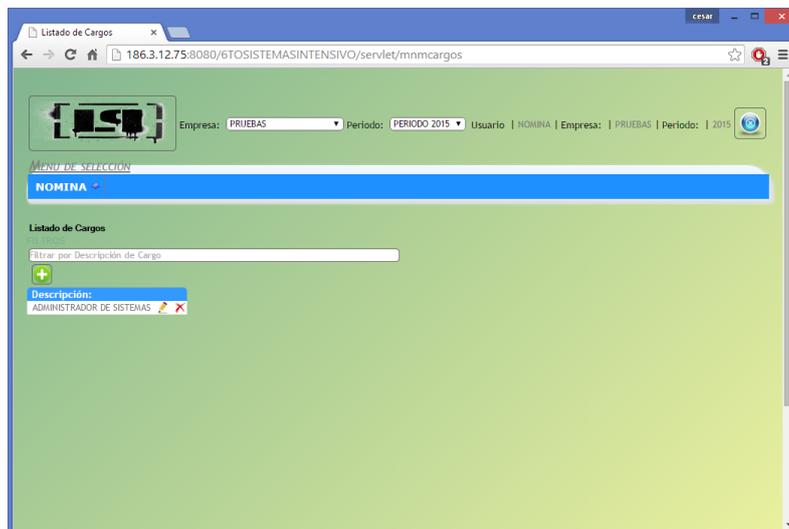


### Creación de Cargos

En el menú de selección seleccionamos NOMINA luego Bases de Referencia, por último presionamos en Cargos.



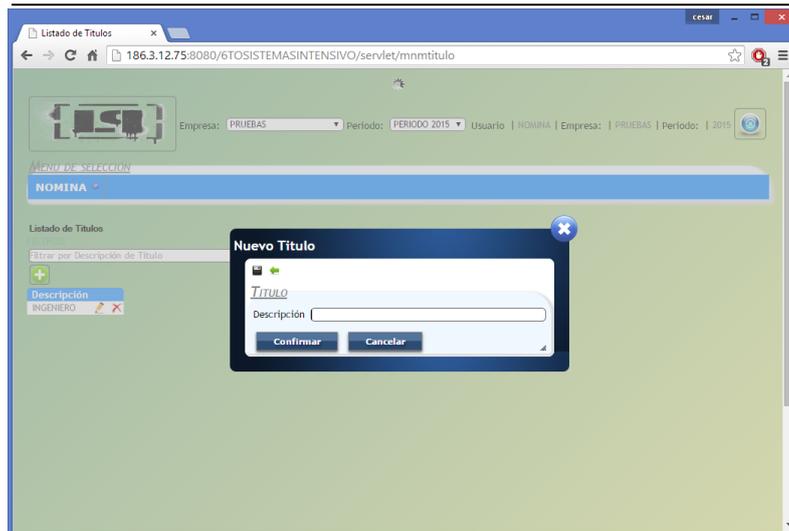
Llenamos los campos solicitados y presionamos confirmar



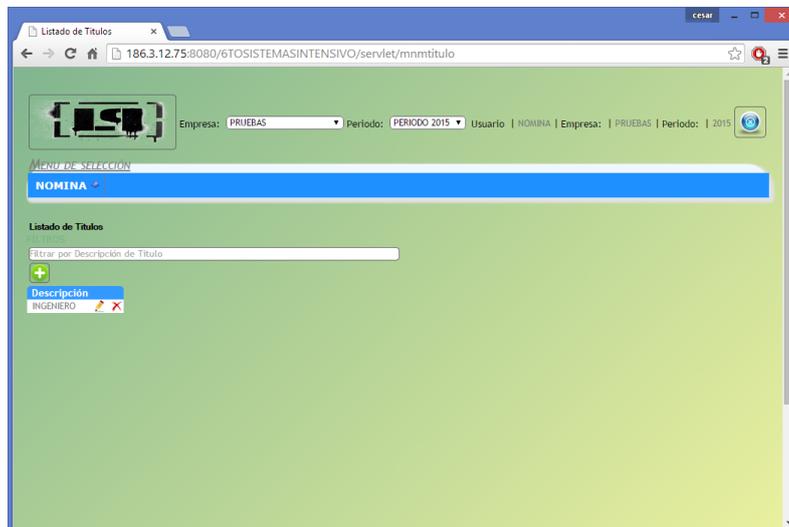
Esta interfaz nos muestra el cargo creado.

### Creación de Título

En el menú de selección seleccionamos NOMINA luego Bases de Referencia, por ultimo presionamos en Título.



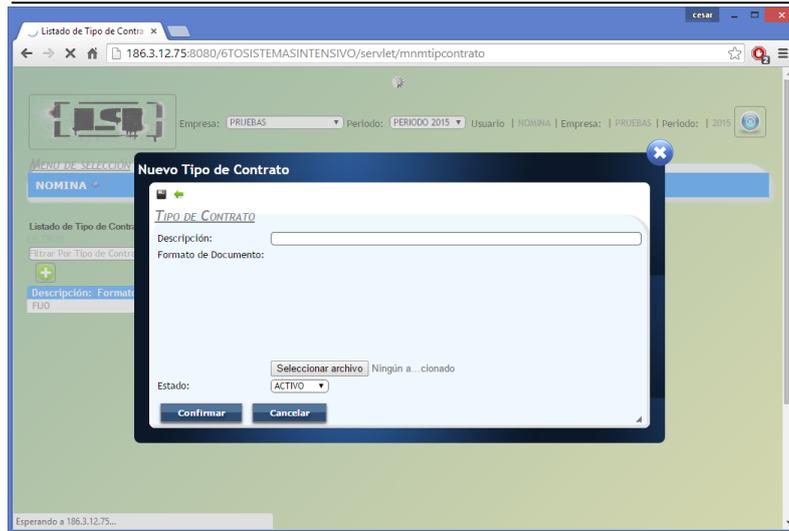
Llenamos los datos requeridos y confirmamos, nos mostrara la interfaz con el título creado.



Interfaz con el título creado

Para las siguientes opciones de base de referencia se realiza el mismo procedimiento, a continuación se muestra la interfaces de creación de los elementos de Base de Referencia.

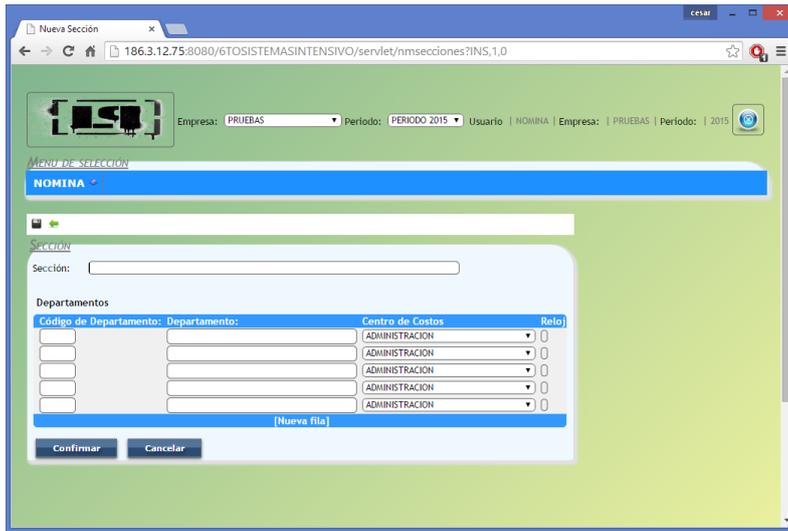
### Creación de tipo de contrato



---

**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.**

## Creación de secciones y departamentos



Empresa: PRIEBAS | Período: PERIODO 2015 | Usuario | NOMINA | Empresa: PRIEBAS | Período: 2015

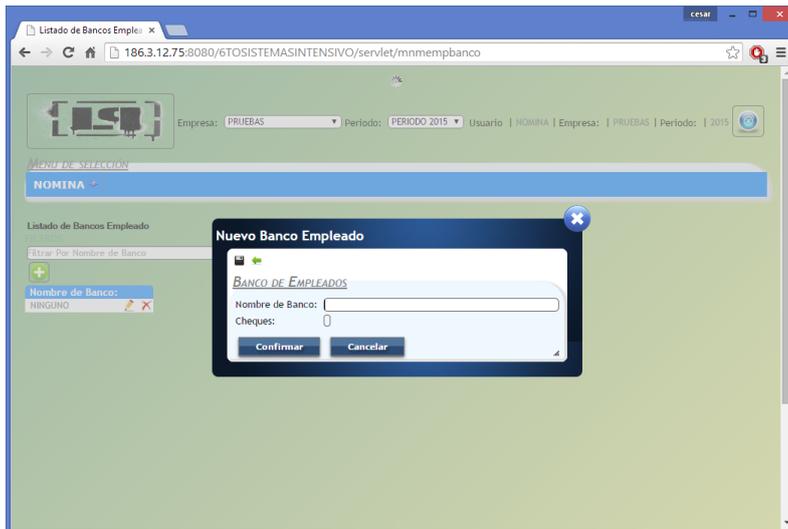
Sección:

Código de Departamento	Departamento	Centro de Costos	Reloj
		ADMINISTRACION	0

[Nueva fila]

Confirmar Cancelar

## Creación de Bancos de empleados



Empresa: PRIEBAS | Período: PERIODO 2015 | Usuario | NOMINA | Empresa: PRIEBAS | Período: 2015

Nombre de Banco: NINGUNO

**Nuevo Banco Empleado**

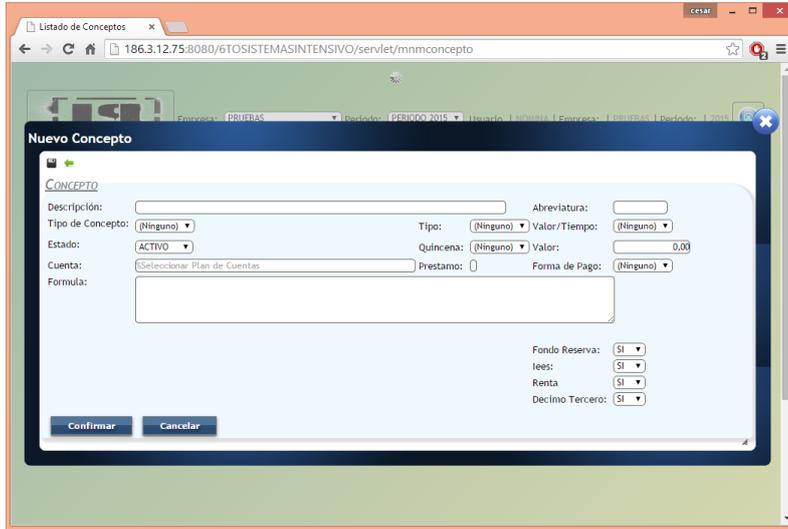
BANCO DE EMPLEADOS

Nombre de Banco:

Cheques: 0

Confirmar Cancelar

## Creación de Conceptos



Lista de Conceptos

186.3.12.75:8080/6TOSISTEMASINTENSIVO/servlet/mnmconcepto

Empresa: PRIEBAS | Período: PERIODO 2015 | Usuario: NOMINA | Empresa: PRIEBAS | Período: 2015

### Nuevo Concepto

CONCEPTO

Descripción:

Tipo de Concepto: (Ninguno) Tipo: (Ninguno) Valor/Tiempo: (Ninguno)

Estado: ACTIVO Quincena: (Ninguno) Valor: 0,00

Cuenta: (Seleccionar Plan de Cuentas) Prestamo:  Forma de Pago: (Ninguno)

Formula:

Fondo Reserva: SI

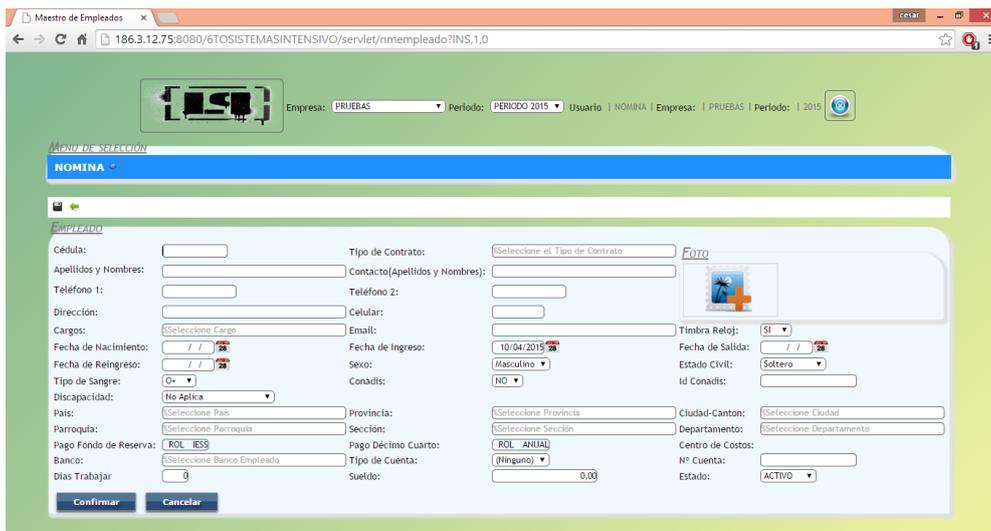
Iees: SI

Renta: SI

Decimo Tercero: SI

Confirmar Cancelar

## Creación de Empleados



Maestro de Empleados

186.3.12.75:8080/6TOSISTEMASINTENSIVO/servlet/nmempleado?INS,1,0

Empresa: PRIEBAS | Período: PERIODO 2015 | Usuario: NOMINA | Empresa: PRIEBAS | Período: 2015

NOMINA

### EMPLEADO

Cédula:

Apellidos y Nombres:

Teléfono 1:

Dirección:

Cargos: (Seleccionar Cargo)

Fecha de Nacimiento:

Fecha de Reingreso:

Tipo de Sangre: O-

Discapacidad: No Aplica

País: (Seleccionar País)

Párrquia: (Seleccionar Párrquia)

Pago Fondo de Reserva: ROL IESS

Bancos: (Seleccionar Banco Empleado)

Dias Trabajar:

Tipo de Contrato: (Seleccionar el Tipo de Contrato)

Contacto (Apellidos y Nombres):

Teléfono 2:

Celular:

Email:

Fecha de Ingreso: 10/04/2015

Sexo: Masculino

Conadis: NO

Provincia: (Seleccionar Provincia)

Sección: (Seleccionar Sección)

Pago Décimo Cuarto: ROL ANUAL

Tipo de Cuenta: (Ninguno)

Sueldo: 0,00

FOTO

Timbra Reloj: SI

Fecha de Salida:

Estado Civil: Soltero

Id Conadis:

Ciudad-Canton: (Seleccionar Ciudad)

Departamento: (Seleccionar Departamento)

Centro de Costos:

N° Cuenta:

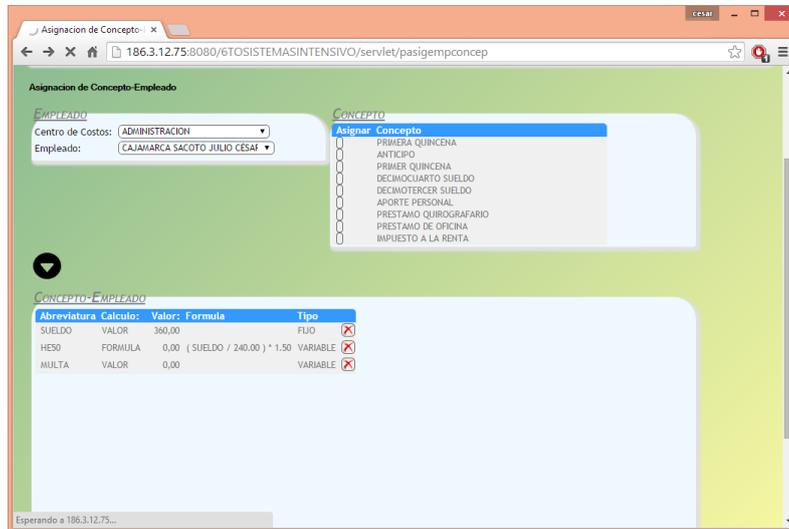
Estado: ACTIVO

Confirmar Cancelar

**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.**

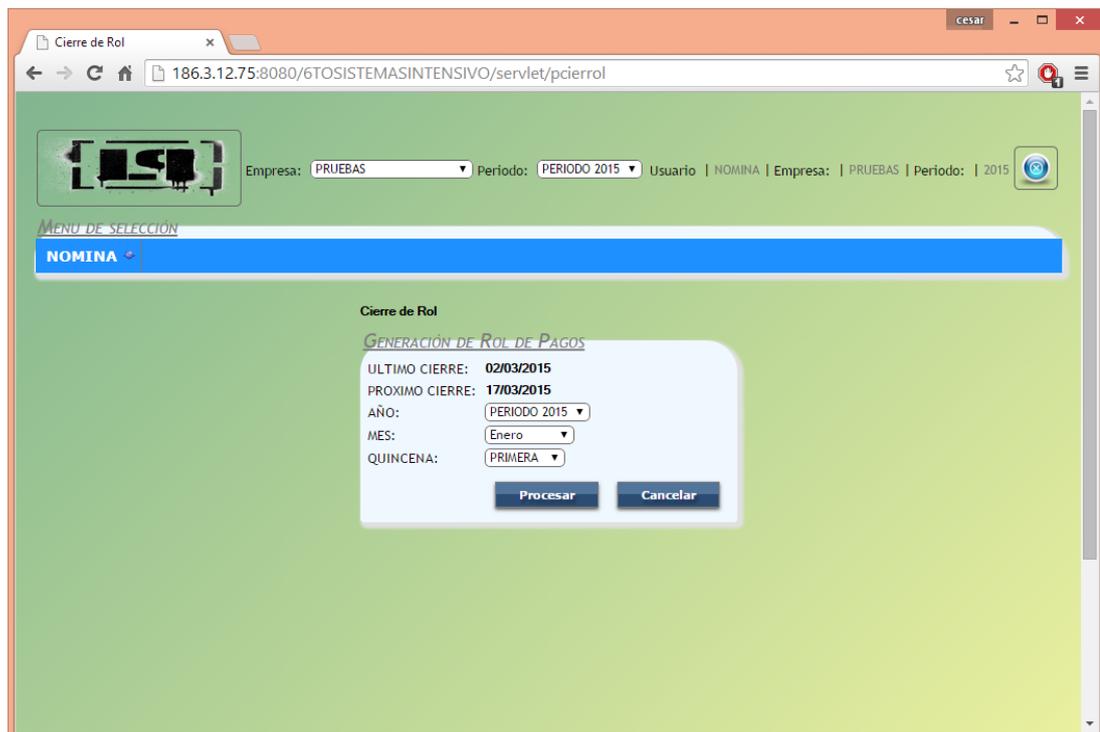
## Asignación de conceptos

Seleccionamos el empleado y asignamos conceptos

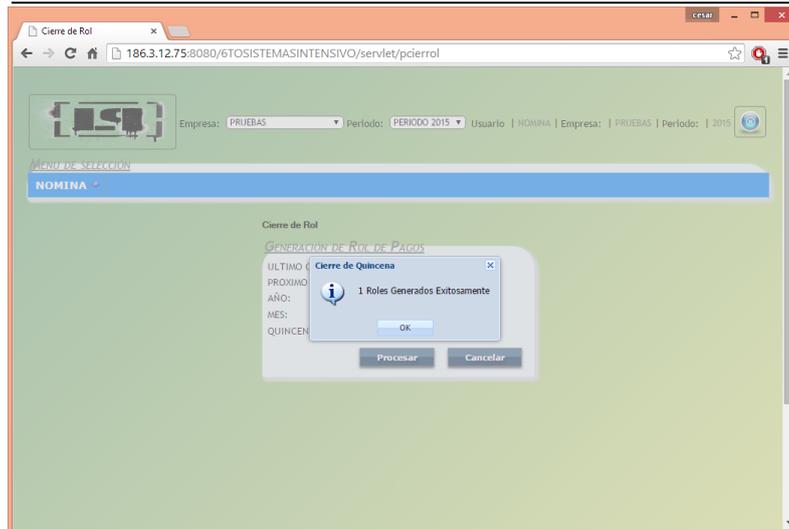


## Cierre de quincena

En la opción de Procesos Especiales de NOMINA seleccionamos cierre de quincena



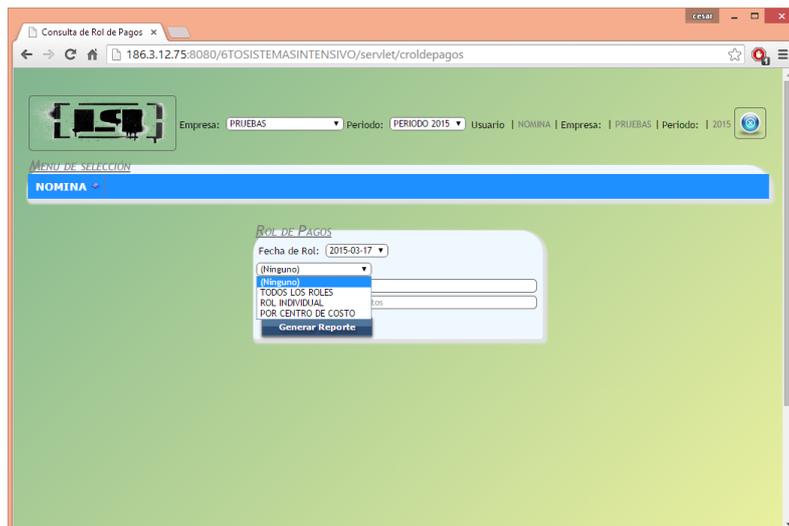
Seleccionamos la primera o segunda quincena según lo requerido y Procesásemos.



Aparecerá un mensaje que indica que se a generado correctamente el rol.

### Consulta de Roles

Para realizar una consulta de Roles y el empleado.



Se genera el rol del empleado seleccionado

Fecha: 10/04/15 20:33:15

Pag. 1 de 1



## COMPROBANTE DE PAGO

PERIODO: 02/03/2015 FECHA DE PAGO: 17/03/2015  
 EMPLEADO: CAJAMARCA SACOTO JULIO CESAR TELÉFONO: 2695820  
 DIRECCIÓN: LA ARCADIA  
 C.COSTOS: ADMINISTRACION

INGRESOS		EGRESOS	
CONCEPTO	VALOR	CONCEPTO	VALOR
SALARIO UNIFICADO	360.00		0.00

TOTAL INGRESOS: 360.00  
 TOTAL EGRESOS: 0.00  
 LIQUIDO A RECIBIR: 360.00

CI: 171204905  
 RECIBI CONFORME

Rol generado



---

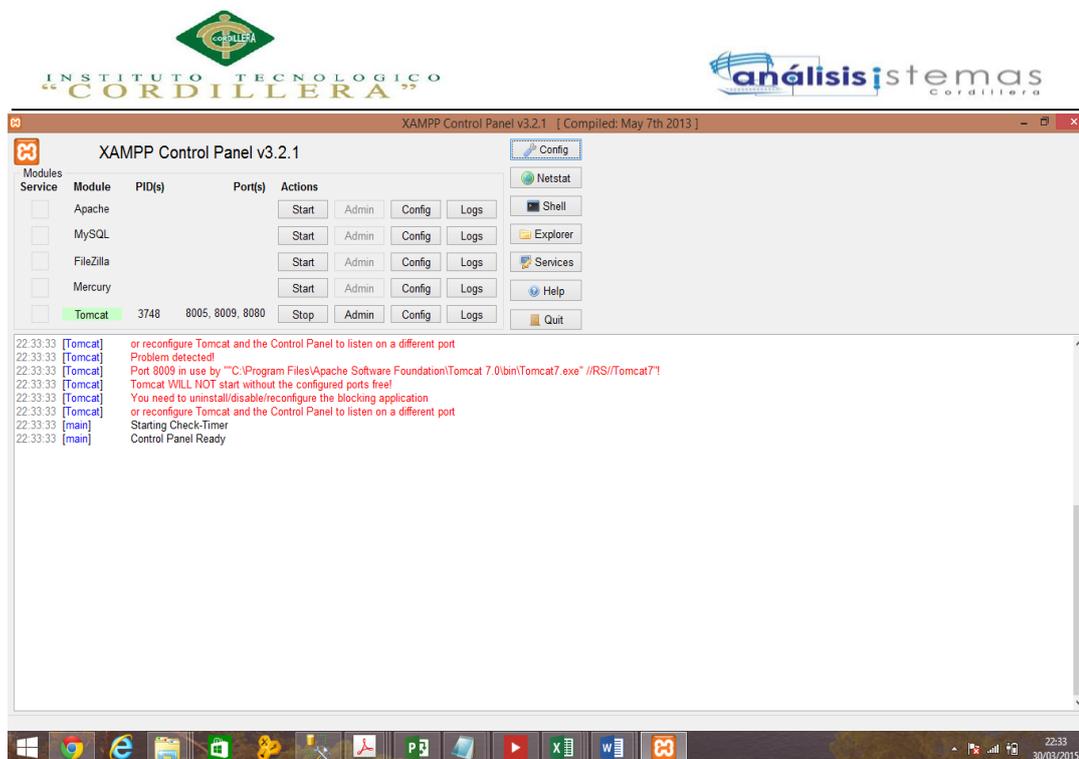
## Manual Técnico

### *Programas utilizados para el desarrollo de la aplicación:*

**XAMPP**-.Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.XAMPP, incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD ó FileZilla FTP Serve, etc. entre muchas cosas más.

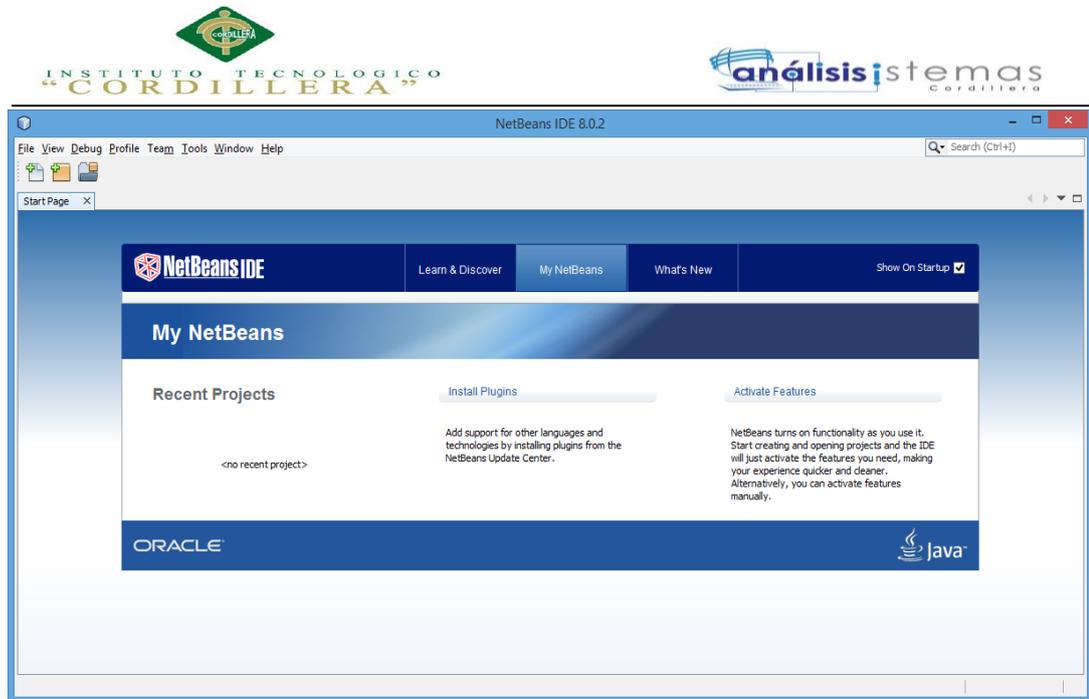
XAMPP es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programación por ejemplo) en tu propio ordenador sin necesidad de tener que acceder a internet.

XAMPP provee de una configuración totalmente funcional desde el momento que se lo instala, sin embargo es bueno acotar que la seguridad de datos no es su punto fuerte, por lo cual no es suficientemente seguro para ambientes grandes o de producción.

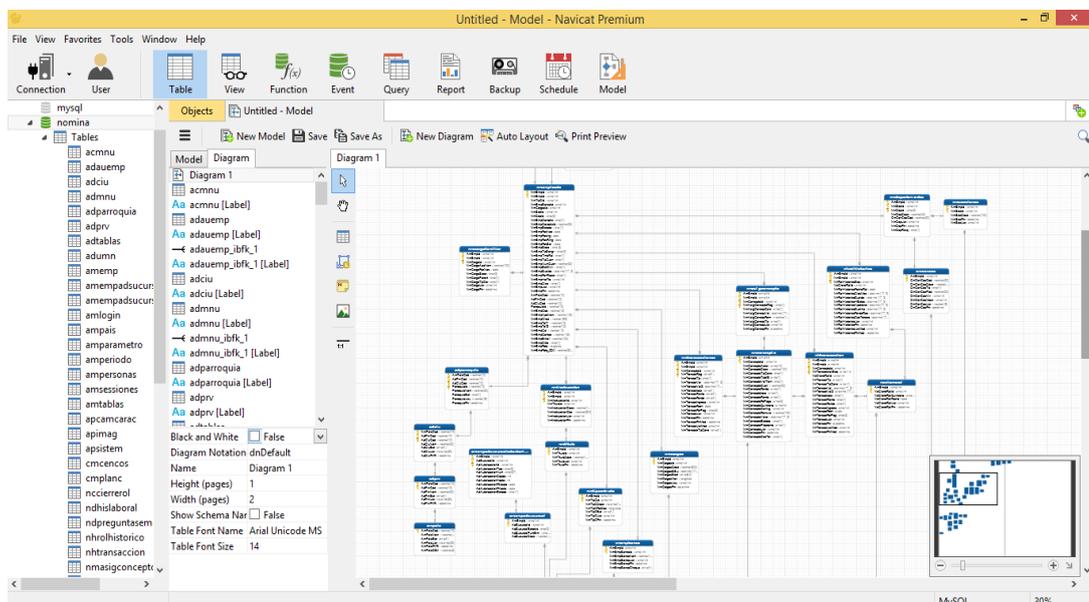


**NETBEANS:** es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).



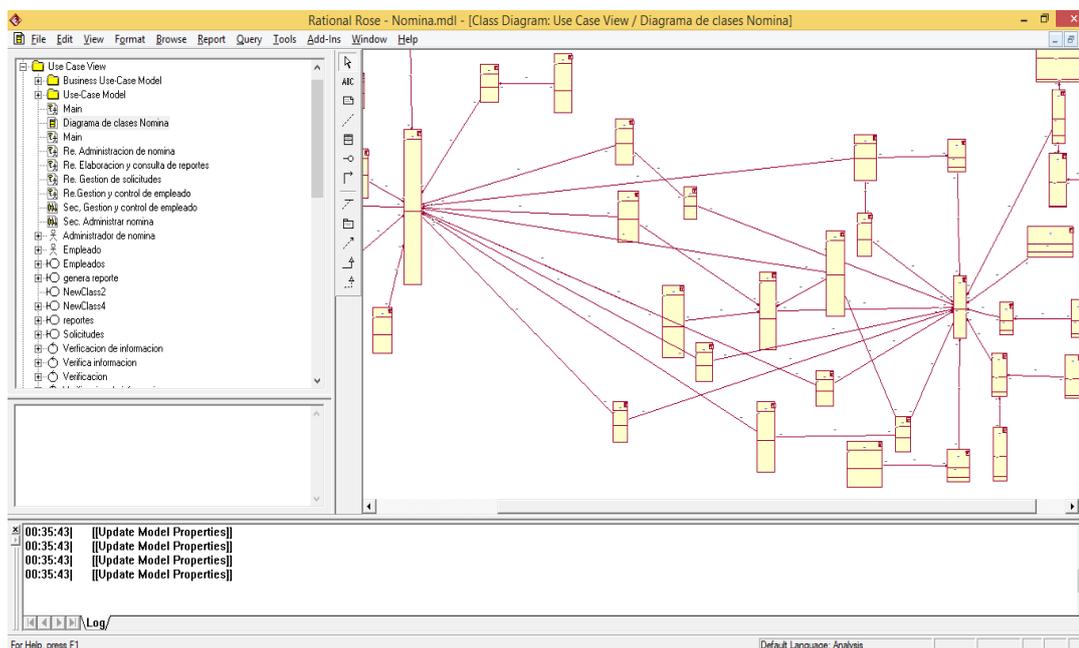
**NAVICAT:** es un administrador gráfico de base de datos y un software de desarrollo producido por PremiumSoft CyberTech Ltd. para MySQL, MariaDB, Oracle, SQLite, PostgreSQL y Microsoft SQL Server. Cuenta con un Explorador como interfaz gráfica de usuario soportando múltiples conexiones para bases de datos locales y remotas.



**RATIONAL ROSE:** Es actualmente conocida como una familia de software de IBM para el despliegue, diseño, construcción, pruebas y administración de proyectos

**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.**

en el proceso desarrollo de software.



### Scrip de la base de datos utilizada

```

CREATE TABLE `acmnu` (
  `AcMnuCod` bigint(20) NOT NULL,
  `AcMnuDes` varchar(40) DEFAULT NULL,
  `AcMnuMPad` bigint(20) NOT NULL,
  `AcMnuEst` char(1) DEFAULT NULL,
  `AcMnuUsr` varchar(20) NOT NULL,
  `AcMnuFHR` datetime NOT NULL,
  PRIMARY KEY (`AcMnuCod`),
  KEY `IACMNU10` (`AcMnuDes`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Table structure for adauemp
-----
DROP TABLE IF EXISTS `adauemp`;
CREATE TABLE `adauemp` (
  `AdAUEmpUsu` int(11) NOT NULL,
  `AmEmpId` smallint(6) NOT NULL,
  `AdAUEmpFhr` datetime NOT NULL,
  `AdAUEmpUsr` varchar(20) NOT NULL,
  PRIMARY KEY (`AdAUEmpUsu`,`AmEmpId`),
  KEY `IADAUEMP1` (`AmEmpId`) USING BTREE,
  CONSTRAINT `adauemp_ibfk_1` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

**AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN MÓDULO NOMINA.**

-- Table structure for adciu

```

-----
DROP TABLE IF EXISTS `adciu`;
CREATE TABLE `adciu` (
  `AmPaisCod` varchar(10) NOT NULL,
  `AdPrvCod` varchar(10) NOT NULL,
  `AdCiuCod` varchar(10) NOT NULL,
  `AdCiuNom` varchar(50) NOT NULL,
  `AdCiuEst` char(1) NOT NULL,
  `AdCiuUsr` varchar(20) NOT NULL,
  `AdCiuFHR` datetime NOT NULL,
  PRIMARY KEY (`AmPaisCod`,`AdPrvCod`,`AdCiuCod`),
  CONSTRAINT `IADCIU1` FOREIGN KEY (`AmPaisCod`,`AdPrvCod`)
REFERENCES `adprv` (`AmPaisCod`,`AdPrvCod`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

-- Table structure for admnu

```

-----
DROP TABLE IF EXISTS `admnu`;
CREATE TABLE `admnu` (
  `AcMnuCod` bigint(20) NOT NULL,
  `AdMnuCod` bigint(20) NOT NULL,
  `AdMnuDes` varchar(40) NOT NULL,
  `AdMnuProg` varchar(40) NOT NULL,
  `AdmnuSub` char(1) NOT NULL,
  `AdMnuArg` varchar(60) DEFAULT NULL,
  `AdMnuTip` char(3) DEFAULT NULL,
  `AdMnuPath` varchar(512) DEFAULT NULL,
  `AdMnuEst` char(1) DEFAULT NULL,
  `AdMnuAut` char(1) DEFAULT NULL,
  `AdMnuUsr` varchar(20) NOT NULL,
  `AdMnuFHR` datetime NOT NULL,
  PRIMARY KEY (`AcMnuCod`,`AdMnuCod`),
  KEY `IDMNU02` (`AdMnuProg`) USING BTREE,
  KEY `IADMNU10` (`AcMnuCod`,`AdMnuTip`) USING BTREE,
  CONSTRAINT `admnu_ibfk_1` FOREIGN KEY (`AcMnuCod`) REFERENCES
`acmnu` (`AcMnuCod`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

-- Table structure for adparroquia

```

-----
DROP TABLE IF EXISTS `adparroquia`;
CREATE TABLE `adparroquia` (
  `AmPaisCod` varchar(10) NOT NULL,
  `AdPrvCod` varchar(10) NOT NULL,
  `AdCiuCod` varchar(10) NOT NULL,
  `ParroquiaId` varchar(10) NOT NULL,
  `ParroquiaNom` varchar(50) NOT NULL,
  `ParroquiaEst` char(1) NOT NULL,

```

```

`ParroquiaUsu` varchar(20) NOT NULL,
`ParroquiaFhr` datetime NOT NULL,
PRIMARY KEY (`AmPaisCod`,`AdPrvCod`,`AdCiuCod`,`ParroquiaId`),
CONSTRAINT `IADCIUADPARROQUIA1` FOREIGN KEY (`AmPaisCod`,
`AdPrvCod`,`AdCiuCod`) REFERENCES `adciu` (`AmPaisCod`,`AdPrvCod`,
`AdCiuCod`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for adprv
-----
DROP TABLE IF EXISTS `adprv`;
CREATE TABLE `adprv` (
  `AmPaisCod` varchar(10) NOT NULL,
  `AdPrvCod` varchar(10) NOT NULL,
  `AdPrvNom` varchar(50) NOT NULL,
  `AdPrvEst` char(1) NOT NULL,
  `AdPrvUsr` varchar(20) NOT NULL,
  `AdPrvFHR` datetime NOT NULL,
  PRIMARY KEY (`AmPaisCod`,`AdPrvCod`),
  CONSTRAINT `IAMPAISADPRV1` FOREIGN KEY (`AmPaisCod`)
REFERENCES `ampais` (`AmPaisCod`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for adtablas
-----
DROP TABLE IF EXISTS `adtablas`;
CREATE TABLE `adtablas` (
  `AmEmpId` smallint(6) NOT NULL,
  `AmTablasId` varchar(15) NOT NULL,
  `AdTablasId` varchar(15) NOT NULL,
  `AdTablasDesc` varchar(40) NOT NULL,
  `AdTablasEst` char(1) NOT NULL,
  `AdTablasUsr` varchar(20) NOT NULL,
  `AdTablasFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`AmTablasId`,`AdTablasId`),
  CONSTRAINT `IAMTABLASADTABLAS1` FOREIGN KEY (`AmEmpId`,
`AmTablasId`) REFERENCES `amtablas` (`AmEmpId`,`AmTablasId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for adumn
-----
DROP TABLE IF EXISTS `adumn`;
CREATE TABLE `adumn` (
  `AmEmpId` smallint(6) NOT NULL,
  `AmPersonasId` int(11) NOT NULL,
  `AcMnuCod` bigint(20) NOT NULL,
  `AdMnuCod` bigint(20) NOT NULL,
  `AdUMnModIN` char(1) DEFAULT NULL,
  `AdUMnModUP` char(1) DEFAULT NULL,

```

```

`AdUMnModDE` char(1) DEFAULT NULL,
`AdUMnUsr` varchar(20) NOT NULL,
`AdUMnFhr` datetime NOT NULL,
PRIMARY KEY (`AmEmpId`,`AmPersonasId`,`AcMnuCod`,`AdMnuCod`),
KEY `IADUMN1` (`AcMnuCod`,`AdMnuCod`) USING BTREE,
KEY `IADUMN2` (`AmPersonasId`) USING BTREE,
CONSTRAINT `adumn_ibfk_1` FOREIGN KEY (`AcMnuCod`,`AdMnuCod`)
REFERENCES `admnu` (`AcMnuCod`,`AdMnuCod`),
CONSTRAINT `adumn_ibfk_2` FOREIGN KEY (`AmPersonasId`)
REFERENCES `ampersonas` (`AmPersonasId`),
CONSTRAINT `adumn_ibfk_3` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for amemp
-----

```

```

DROP TABLE IF EXISTS `amemp`;
CREATE TABLE `amemp` (
  `AmEmpId` smallint(6) NOT NULL AUTO_INCREMENT,
  `AmEmpNomb` varchar(100) NOT NULL,
  `AmEmpRepN` varchar(40) NOT NULL,
  `AmEmpReCed` char(10) NOT NULL,
  `AmEmpRuc` char(13) NOT NULL,
  `AmEmpDir` varchar(200) NOT NULL,
  `AmEmpEmail` varchar(100) NOT NULL,
  `AmEmpTelf` varchar(15) NOT NULL,
  `AmEmpRazS` varchar(100) NOT NULL,
  `AmEmpConNo` varchar(100) NOT NULL,
  `AmEmpConCed` char(10) NOT NULL,
  `AmEmpConEma` varchar(100) NOT NULL,
  `AmEmpImg` longblob NOT NULL,
  `AmEmpImg_GXI` varchar(2048) DEFAULT NULL,
  `AmEmpTipoEmici` char(1) DEFAULT NULL,
  `AmEmpAmbiente` char(1) DEFAULT NULL,
  PRIMARY KEY (`AmEmpId`),
  KEY `UAMEMP` (`AmEmpNomb`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for amempadsucursal
-----

```

```

DROP TABLE IF EXISTS `amempadsucursal`;
CREATE TABLE `amempadsucursal` (
  `AmEmpId` smallint(6) NOT NULL,
  `AdSucursalId` smallint(6) NOT NULL,
  `AdSucursalEstable` char(3) NOT NULL,
  `AdSucursalPuntEmi` char(3) NOT NULL,
  `AdSucursalDesc` varchar(40) NOT NULL,
  PRIMARY KEY (`AmEmpId`,`AdSucursalId`),
  CONSTRAINT `IAMEMPADSUCURSALI` FOREIGN KEY (`AmEmpId`)

```

```

REFERENCES `amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for amempadsucursaladaautorizacion
-----
DROP TABLE IF EXISTS `amempadsucursaladaautorizacion`;
CREATE TABLE `amempadsucursaladaautorizacion` (
  `AmEmpId` smallint(6) NOT NULL,
  `AdSucursalId` smallint(6) NOT NULL,
  `AdAutorizacionId` smallint(6) NOT NULL,
  `AdAutorizacionTipo` char(3) NOT NULL,
  `AdAutorizacionNum` char(37) NOT NULL,
  `AdAutorizacionDesde` int(11) NOT NULL,
  `AdAutorizacionHasta` int(11) NOT NULL,
  `AdAutorizacionFDesde` date NOT NULL,
  `AdAutorizacionFHasta` date NOT NULL,
  `AdAutorizacionEstado` char(1) NOT NULL,
  PRIMARY KEY (`AmEmpId`,`AdSucursalId`,`AdAutorizacionId`),
  CONSTRAINT `IAMEMPADSUCURSALADAUTORIZACIO1` FOREIGN KEY
(`AmEmpId`,`AdSucursalId`) REFERENCES `amempadsucursal` (`AmEmpId`,
`AdSucursalId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for amlogin
-----
DROP TABLE IF EXISTS `amlogin`;
CREATE TABLE `amlogin` (
  `AmPersonasId` int(11) NOT NULL,
  `AmLoginId` char(1) NOT NULL,
  `AmLoginUsu` varchar(20) NOT NULL,
  `AmLoginContra` varchar(128) NOT NULL,
  `AmLoginTipo` char(1) NOT NULL,
  `AmLoginEst` char(1) NOT NULL,
  `AmLoginUsr` varchar(20) NOT NULL,
  `AmLoginFhr` datetime NOT NULL,
  PRIMARY KEY (`AmPersonasId`,`AmLoginId`),
  CONSTRAINT `amlogin_ibfk_1` FOREIGN KEY (`AmPersonasId`)
REFERENCES `ampersonas` (`AmPersonasId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for ampais
-----
DROP TABLE IF EXISTS `ampais`;
CREATE TABLE `ampais` (
  `AmPaisCod` varchar(10) NOT NULL,
  `AmPaisNom` varchar(50) NOT NULL,
  `AmPaisEst` char(1) NOT NULL,
  `AmPaisUsr` varchar(20) NOT NULL,
  `AmPaisFHR` datetime NOT NULL,

```

```

`AmPaisCSri` varchar(3) DEFAULT NULL,
PRIMARY KEY (`AmPaisCod`),
KEY `IAMPAIS10` (`AmPaisNom`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for amparametro
-----
DROP TABLE IF EXISTS `amparametro`;
CREATE TABLE `amparametro` (
  `AmEmpId` smallint(6) NOT NULL,
  `AmPeriodoId` smallint(6) NOT NULL,
  `AmParametroId` smallint(6) NOT NULL,
  `AmParametroModulo` char(3) NOT NULL,
  `AmParametroTipo` char(3) NOT NULL,
  `AmParametroDesc` varchar(40) NOT NULL,
  `AmParametroValor` varchar(40) NOT NULL,
  `AmParametroUsu` smallint(6) NOT NULL,
  `AmParametroFecha` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`AmPeriodoId`,`AmParametroId`),
  CONSTRAINT `IAMPARAMETRO1` FOREIGN KEY (`AmEmpId`,
`AmPeriodoId`) REFERENCES `amperiodo` (`AmEmpId`,`AmPeriodoId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for amperiodo
-----
DROP TABLE IF EXISTS `amperiodo`;
CREATE TABLE `amperiodo` (
  `AmEmpId` smallint(6) NOT NULL,
  `AmPeriodoId` smallint(6) NOT NULL,
  `AmPeriodoDesc` varchar(40) NOT NULL,
  `AmPeriodoEst` char(1) NOT NULL,
  `AmPeriodoFini` date NOT NULL,
  `AmPeriodoFFin` date NOT NULL,
  `AmPeriodoUsr` varchar(20) NOT NULL,
  `AmPeriodoFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`AmPeriodoId`),
  KEY `UAMPERIODO1` (`AmEmpId`,`AmPeriodoDesc`) USING BTREE,
  CONSTRAINT `amperiodo_ibfk_1` FOREIGN KEY (`AmEmpId`)
REFERENCES `amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for ampersonas
-----
DROP TABLE IF EXISTS `ampersonas`;
CREATE TABLE `ampersonas` (
  `AmPersonasId` int(11) NOT NULL,
  `AmPersonasNom` varchar(100) NOT NULL,
  `AmPersonasIden` varchar(20) NOT NULL,
  `AmPersonasDirec` varchar(200) NOT NULL,

```

```

`AmPersonasUbFi` varchar(1024) NOT NULL,
`AmPersonasTelf` varchar(15) NOT NULL,
`AmPersonasTelf2` varchar(15) NOT NULL,
`AmPersonasCel` varchar(10) NOT NULL,
`AmPersonasEmail` varchar(100) NOT NULL,
`AmPersonasContact` varchar(100) NOT NULL,
`AmPersonasEst` char(1) NOT NULL,
`AmPersonasUsr` varchar(20) NOT NULL,
`AmPersonasFhr` datetime NOT NULL,
PRIMARY KEY (`AmPersonasId`),
KEY `UAMPERSONAS1` (`AmPersonasNom`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for amsessiones
-----
DROP TABLE IF EXISTS `amsessiones`;
CREATE TABLE `amsessiones` (
  `AmSessionesLogin` int(11) NOT NULL,
  `AmSessionesIP` varchar(32) NOT NULL,
  `AmSessionesFhr` datetime NOT NULL,
  PRIMARY KEY (`AmSessionesLogin`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for amtablas
-----
DROP TABLE IF EXISTS `amtablas`;
CREATE TABLE `amtablas` (
  `AmEmpId` smallint(6) NOT NULL,
  `AmTablasId` varchar(15) NOT NULL,
  `AmTablasDesc` varchar(40) NOT NULL,
  `AmTablasUsr` varchar(20) NOT NULL,
  `AmTablasFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`AmTablasId`),
  CONSTRAINT `IAMTABLAS1` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for apcamcarac
-----
DROP TABLE IF EXISTS `apcamcarac`;
CREATE TABLE `apcamcarac` (
  `ApCamCaracId` smallint(6) NOT NULL,
  `ApCamCaracOri` char(3) NOT NULL,
  `ApCamCaracCam` char(3) NOT NULL,
  `ApCamCaracEst` char(1) NOT NULL,
  `ApCamCaracFhr` datetime NOT NULL,
  `ApCamCaracUsr` smallint(6) NOT NULL,
  PRIMARY KEY (`ApCamCaracId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

-----  
-- Table structure for apimag

-----  
DROP TABLE IF EXISTS `apimag`;  
CREATE TABLE `apimag` (  
  `ApImagId` smallint(6) NOT NULL,  
  `ApImagUpPath` mediumtext NOT NULL,  
  `ApImagUpNom` varchar(20) NOT NULL,  
  `ApImagUpExt` varchar(10) NOT NULL,  
  `ApImagInsPath` mediumtext NOT NULL,  
  `ApImagInsNom` varchar(20) NOT NULL,  
  `ApImagInsExt` varchar(10) NOT NULL,  
  `ApImagDelPath` mediumtext NOT NULL,  
  `ApImagDelNom` varchar(20) NOT NULL,  
  `ApImagDelExt` varchar(10) NOT NULL,  
  `ApImagUpAcc` varchar(40) NOT NULL,  
  `ApImagInsAcc` varchar(40) NOT NULL,  
  `ApImagDelAcc` varchar(40) NOT NULL,  
  PRIMARY KEY (`ApImagId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----  
-- Table structure for apsistem

-----  
DROP TABLE IF EXISTS `apsistem`;  
CREATE TABLE `apsistem` (  
  `ApSistemId` smallint(6) NOT NULL,  
  `ApSistemSite` varchar(60) NOT NULL,  
  `ApSistemKey` varchar(128) NOT NULL,  
  `ApSistemLeng` char(1) NOT NULL,  
  `ApSistemPathTemplate` mediumtext NOT NULL,  
  `ApSistemPathDocTemp` mediumtext NOT NULL,  
  `ApSistemNumUsuPer` varchar(300) NOT NULL,  
  `ApSistemTiemEsSess` smallint(6) NOT NULL,  
  `ApSistemUsuario` varchar(20) NOT NULL,  
  `ApSistemFhr` datetime NOT NULL,  
  `ApSistemPlanCuenProv` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`ApSistemId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----  
-- Table structure for cmcencos

-----  
DROP TABLE IF EXISTS `cmcencos`;  
CREATE TABLE `cmcencos` (  
  `AmEmpId` smallint(6) NOT NULL,  
  `CmCenCosCod` varchar(50) NOT NULL,  
  `CmCenCosDesc` varchar(60) NOT NULL,  
  `CmCenCosTip` char(1) NOT NULL,  
  `CmCenCosPad` varchar(50) NOT NULL,  
  `CmCenCosNiv` smallint(6) NOT NULL,

```

`CmCenCosMask` smallint(6) NOT NULL,
`CmCenCosUsr` varchar(20) NOT NULL,
`CmCenCosFhr` datetime NOT NULL,
PRIMARY KEY (`AmEmpId`,`CmCenCosCod`),
CONSTRAINT `ICMCENCOS1` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for cmplanc
-----
DROP TABLE IF EXISTS `cmplanc`;
CREATE TABLE `cmplanc` (
  `AmEmpId` smallint(6) NOT NULL,
  `AmPeriodoId` smallint(6) NOT NULL,
  `CmPlanCCuen` varchar(50) NOT NULL,
  `CmPlanCDesc` varchar(60) NOT NULL,
  `CmPlanCTCue` char(1) NOT NULL,
  `CmPlanCCueP` varchar(50) NOT NULL,
  `CmPlanCNive` smallint(6) NOT NULL,
  `CmPlanCMask` smallint(6) NOT NULL,
  `CmPlanCNat` char(1) NOT NULL,
  `CmPlanCUsr` varchar(20) NOT NULL,
  `CmPlanCFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`AmPeriodoId`,`CmPlanCCuen`),
  KEY `UCMPLANC` (`AmEmpId`,`CmPlanCCuen`,`AmPeriodoId`),
  KEY `UCMPLANC1` (`AmEmpId`,`CmPlanCNive`),
  CONSTRAINT `ICMPLANC1` FOREIGN KEY (`AmEmpId`,`AmPeriodoId`)
REFERENCES `amperiodo` (`AmEmpId`,`AmPeriodoId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nccierrerol
-----
DROP TABLE IF EXISTS `nccierrerol`;
CREATE TABLE `nccierrerol` (
  `AmEmpId` smallint(6) NOT NULL,
  `NcCierreRolId` smallint(6) NOT NULL,
  `NcCierreRolQuincena` smallint(6) NOT NULL,
  `NcCierreRolFecha` date NOT NULL,
  `NcCierreRolUsr` smallint(6) NOT NULL,
  `NcCierreRolFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NcCierreRolId`),
  CONSTRAINT `INCCIERREROL1` FOREIGN KEY (`AmEmpId`)
REFERENCES `amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for ndhislaboral
-----
DROP TABLE IF EXISTS `ndhislaboral`;
CREATE TABLE `ndhislaboral` (

```

```

`AmEmpId` smallint(6) NOT NULL,
`NmEmpId` smallint(6) NOT NULL,
`NdHisLabId` smallint(6) NOT NULL,
`NdHisLabFecIng` date NOT NULL,
`NdHisLabFecSal` date NOT NULL,
`NdHisLabMotSal` varchar(500) NOT NULL,
`NdHisLabUsr` smallint(6) NOT NULL,
`NdHisLabFhr` datetime NOT NULL,
PRIMARY KEY (`AmEmpId`,`NmEmpId`,`NdHisLabId`),
CONSTRAINT `INDHISLAB1` FOREIGN KEY (`AmEmpId`,`NmEmpId`)
REFERENCES `nmempleado` (`AmEmpId`,`NmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for ndpreguntasempleado
-----
DROP TABLE IF EXISTS `ndpreguntasempleado`;
CREATE TABLE `ndpreguntasempleado` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpId` smallint(6) NOT NULL,
  `NdPreguntasEmpleadoId` smallint(6) NOT NULL,
  `NmPreguntasId` smallint(6) NOT NULL,
  `NdPreguntasEmpleadoFech` datetime NOT NULL,
  `NdPreguntasEmpleadoRespue` char(3) NOT NULL,
  `NdPreguntasEmpleadoEst` char(1) NOT NULL,
  `NdPreguntasEmpleadoUsu` smallint(6) NOT NULL,
  `NdPreguntasEmpleadoFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmEmpId`,`NdPreguntasEmpleadoId`),
  KEY `INDPREGUNTASEMPLEADO1` (`NmPreguntasId`),
  CONSTRAINT `INDPREGUNTASEMPLEADO1` FOREIGN KEY
(`NmPreguntasId`) REFERENCES `nmpreguntas` (`NmPreguntasId`),
  CONSTRAINT `INDPREGUNTASEMPLEADO2` FOREIGN KEY (`AmEmpId`,
`NmEmpId`) REFERENCES `nmempleado` (`AmEmpId`,`NmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nhrolhistorico
-----
DROP TABLE IF EXISTS `nhrolhistorico`;
CREATE TABLE `nhrolhistorico` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpId` smallint(6) NOT NULL,
  `NhRolHistoricoSeq` smallint(6) NOT NULL,
  `NcCierreRolId` smallint(6) NOT NULL,
  `NhRolHistoricoFechaRol` date NOT NULL,
  `NhRolHistoricoDiasMes` decimal(17,2) NOT NULL,
  `NhRolHistoricoSueldo` decimal(17,2) NOT NULL,
  `NhRolHistoricoHExtas` decimal(17,2) NOT NULL,
  `NhRolHistoricoApatronal` decimal(17,2) NOT NULL,
  `NhRolHistoricoSueImp` decimal(17,2) NOT NULL,
  `NhRolHistoricoFondoRes` decimal(17,2) NOT NULL,

```

```

`NhRolHistoricoDecTercero` decimal(17,2) NOT NULL,
`NhRolHistoricoUsr` smallint(6) NOT NULL,
`NhRolHistoricoFhr` datetime NOT NULL,
`NhRolHistoricoUsrMod` smallint(6) NOT NULL,
`NhRolHistoricoFhrMod` datetime NOT NULL,
PRIMARY KEY (`AmEmpId`,`NmEmpId`,`NhRolHistoricoSeq`),
KEY `INHROLHISTORICO2` (`AmEmpId`,`NcCierreRolId`),
KEY `UNHROLHISTORICO1`
(`AmEmpId`,`NmEmpId`,`NhRolHistoricoFechaRol`),
CONSTRAINT `INHROLHISTORICO1` FOREIGN KEY (`AmEmpId`,
`NmEmpId`) REFERENCES `nmempleado` (`AmEmpId`,`NmEmpId`),
CONSTRAINT `INHROLHISTORICO2` FOREIGN KEY (`AmEmpId`,
`NcCierreRolId`) REFERENCES `nccierrerol` (`AmEmpId`,`NcCierreRolId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

-----  
-- Table structure for nhtransaccion  
-----

```

DROP TABLE IF EXISTS `nhtransaccion`;
CREATE TABLE `nhtransaccion` (
`AmEmpId` smallint(6) NOT NULL,
`NmEmpId` smallint(6) NOT NULL,
`NmConceptoId` smallint(6) NOT NULL,
`NhTransaccionSeq` smallint(6) NOT NULL,
`NcCierreRolId` smallint(6) NOT NULL,
`NhTransacTip` char(1) NOT NULL,
`NhTransacTipConc` char(1) NOT NULL,
`NhTransacVal` decimal(17,2) NOT NULL,
`NhTransacVal2` decimal(17,2) NOT NULL,
`NhTransacLess` char(1) NOT NULL,
`NhTransacRenta` char(1) NOT NULL,
`NhTransacFondo` char(1) NOT NULL,
`NhTransacIngreso` smallint(6) NOT NULL,
`NhTransacFech` date NOT NULL,
`NhTransacForPag` char(3) NOT NULL,
`NhTransacUsr` smallint(6) NOT NULL,
`NhTransacFhr` datetime NOT NULL,
`NhTransacUsrMod` smallint(6) NOT NULL,
`NhTransacFhrMod` datetime NOT NULL,
PRIMARY KEY
(`AmEmpId`,`NmEmpId`,`NmConceptoId`,`NhTransaccionSeq`),
KEY `INHTRANSACCION3` (`AmEmpId`,`NcCierreRolId`),
KEY `INHTRANSACCION1` (`AmEmpId`,`NmConceptoId`),
KEY `UNHTRANSACCION` (`AmEmpId`,`NmEmpId`,`NhTransaccionSeq`),
KEY `UNHTRANSACCION1` (`AmEmpId`,`NmEmpId`,`NhTransacFech`),
CONSTRAINT `INHTRANSACCION1` FOREIGN KEY (`AmEmpId`,
`NmConceptoId`) REFERENCES `nmconcepto` (`AmEmpId`,`NmConceptoId`),
CONSTRAINT `INHTRANSACCION2` FOREIGN KEY (`AmEmpId`,
`NmEmpId`) REFERENCES `nmempleado` (`AmEmpId`,`NmEmpId`),
CONSTRAINT `INHTRANSACCION3` FOREIGN KEY (`AmEmpId`,

```

```

`NcCierreRolId`) REFERENCES `nccierrol` (`AmEmpId`, `NcCierreRolId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nmasigconcepto
-----
DROP TABLE IF EXISTS `nmasigconcepto`;
CREATE TABLE `nmasigconcepto` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpId` smallint(6) NOT NULL,
  `NmConceptoId` smallint(6) NOT NULL,
  `NmAsigConceptoReg` char(1) NOT NULL,
  `NmAsigConcepCalc` char(1) NOT NULL,
  `NmAsigConcepVal` decimal(17,2) NOT NULL,
  `NmAsigConcepForm` varchar(100) NOT NULL,
  `NmAsigConcepTip` char(1) NOT NULL,
  `NmAsigConcepUsr` smallint(6) NOT NULL,
  `NmAsigConcepFhr` datetime NOT NULL,
  PRIMARY KEY
(`AmEmpId`, `NmEmpId`, `NmConceptoId`, `NmAsigConceptoReg`),
  KEY `INMASIGCONCEPTO1` (`AmEmpId`, `NmConceptoId`),
  CONSTRAINT `INMASIGCONCEPTO1` FOREIGN KEY (`AmEmpId`,
`NmConceptoId`) REFERENCES `nmconcepto` (`AmEmpId`, `NmConceptoId`),
  CONSTRAINT `INMASIGCONCEPTO2` FOREIGN KEY (`AmEmpId`,
`NmEmpId`) REFERENCES `nmempleado` (`AmEmpId`, `NmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nmcargafamiliar
-----
DROP TABLE IF EXISTS `nmcargafamiliar`;
CREATE TABLE `nmcargafamiliar` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpId` smallint(6) NOT NULL,
  `NmCargaId` smallint(6) NOT NULL,
  `NmCargaApeNom` varchar(100) NOT NULL,
  `NmCargaFecNac` date NOT NULL,
  `NmCargaSexo` char(3) NOT NULL,
  `NmCargaParent` char(1) NOT NULL,
  `NmCargaTipCar` char(1) NOT NULL,
  `NmCargaUsr` smallint(6) NOT NULL,
  `NmCargaFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`, `NmEmpId`, `NmCargaId`),
  CONSTRAINT `INMCARGA1` FOREIGN KEY (`AmEmpId`, `NmEmpId`)
REFERENCES `nmempleado` (`AmEmpId`, `NmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nmcargos
-----
DROP TABLE IF EXISTS `nmcargos`;
CREATE TABLE `nmcargos` (

```

```
`AmEmpId` smallint(6) NOT NULL,
`NmCargosId` smallint(6) NOT NULL,
`NmCargosDesc` varchar(200) NOT NULL,
`NmCargosSBU` decimal(17,2) NOT NULL,
`NmCargosSect` char(20) NOT NULL,
`NmCargosMan` longblob NOT NULL,
`NmCargosUsu` smallint(6) NOT NULL,
`NmCargosFhr` datetime NOT NULL,
PRIMARY KEY (`AmEmpId`,`NmCargosId`),
CONSTRAINT `INMCARGOS1` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----
-- Table structure for nmconcepto
```

```
-----
DROP TABLE IF EXISTS `nmconcepto`;
CREATE TABLE `nmconcepto` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmConceptoId` smallint(6) NOT NULL,
  `NmConceptoAbrev` char(10) NOT NULL,
  `NmConceptoDesc` varchar(100) NOT NULL,
  `NmConceptoTipConc` char(1) NOT NULL,
  `NmConceptoTipoIE` char(1) NOT NULL,
  `NmConceptoValTiem` char(1) NOT NULL,
  `NmConceptoCuen` varchar(30) NOT NULL,
  `NmConceptoFondo` char(1) NOT NULL,
  `NmConceptoIess` char(1) NOT NULL,
  `NmConceptoRenta` char(1) NOT NULL,
  `NmConceptoFoPago` char(3) NOT NULL,
  `NmConceptoQuincena` smallint(6) NOT NULL,
  `NmConceptoConIng` smallint(6) NOT NULL,
  `NmConceptoFormula` varchar(100) NOT NULL,
  `NmConceptoValor` decimal(17,2) NOT NULL,
  `NmConceptoEstado` char(1) NOT NULL,
  `NmConceptoPrestamo` char(1) NOT NULL,
  `NmConceptoUsr` smallint(6) NOT NULL,
  `NmConceptoFhr` datetime NOT NULL,
  `NmConceptoDecTer` char(1) NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmConceptoId`),
  KEY `UNMCONCEPTO1` (`AmEmpId`,`NmConceptoAbrev`),
  CONSTRAINT `INMCONCEPTO1` FOREIGN KEY (`AmEmpId`)
REFERENCES `amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----
-- Table structure for nmdepartamentos
```

```
-----
DROP TABLE IF EXISTS `nmdepartamentos`;
CREATE TABLE `nmdepartamentos` (
  `AmEmpId` smallint(6) NOT NULL,
```

```

`NmSecId` smallint(6) NOT NULL,
`NmDepId` char(6) NOT NULL,
`NmDepDesc` varchar(40) NOT NULL,
`CmCenCosCod` varchar(50) NOT NULL,
`NmDepUsr` smallint(6) NOT NULL,
`NmDepFhr` datetime NOT NULL,
`NmDepReloj` char(1) NOT NULL,
PRIMARY KEY (`AmEmpId`,`NmSecId`,`NmDepId`),
KEY `INMSECNMDEP1` (`AmEmpId`,`CmCenCosCod`),
CONSTRAINT `INMSECNMDEP1` FOREIGN KEY (`AmEmpId`,
`CmCenCosCod`) REFERENCES `cmcencos` (`AmEmpId`,`CmCenCosCod`),
CONSTRAINT `INMSECNMDEP2` FOREIGN KEY (`AmEmpId`,`NmSecId`)
REFERENCES `nmsecciones` (`AmEmpId`,`NmSecId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for nmempbanco

```

```

-----
DROP TABLE IF EXISTS `nmempbanco`;
CREATE TABLE `nmempbanco` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpBancoId` smallint(6) NOT NULL,
  `NmEmpBancoNom` varchar(100) NOT NULL,
  `NmEmpBancoUsr` smallint(6) NOT NULL,
  `NmEmpBancoFhr` datetime NOT NULL,
  `NmEmpBancoCheque` char(1) NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmEmpBancoId`),
  CONSTRAINT `INMEMPBANCO1` FOREIGN KEY (`AmEmpId`)
REFERENCES `amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for nmempleado

```

```

-----
DROP TABLE IF EXISTS `nmempleado`;
CREATE TABLE `nmempleado` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpId` smallint(6) NOT NULL,
  `NmTipCId` smallint(6) NOT NULL,
  `NmEmpBancoId` smallint(6) NOT NULL,
  `NmCargosId` smallint(6) NOT NULL,
  `NmSecId` smallint(6) NOT NULL,
  `NmDepId` char(6) NOT NULL,
  `NmEmpConadis` char(1) NOT NULL,
  `NmEmpConadisId` varchar(20) NOT NULL,
  `NmEmpEstado` char(1) NOT NULL,
  `NmEmpFecNac` date NOT NULL,
  `NmEmpFecIng` date NOT NULL,
  `NmEmpFecRIng` date NOT NULL,
  `NmEmpFecSal` date NOT NULL,
  `NmEmpSexo` char(3) NOT NULL,

```

```

`NmEmpTipSangr` char(5) NOT NULL,
`NmEmpTimbRel` char(1) NOT NULL,
`NmEmpTipCuen` char(1) NOT NULL,
`NmEmpNumCuen` varchar(20) NOT NULL,
`NmEmpEstCivil` char(1) NOT NULL,
`NmEmpSueldo` decimal(17,2) NOT NULL,
`NmEmpFonReser` char(1) NOT NULL,
`NmEmpHorTra` smallint(6) NOT NULL,
`NmEmpDisc` char(1) NOT NULL,
`NmEmpUsr` smallint(6) NOT NULL,
`NmEmpFhr` datetime NOT NULL,
`AmPaisCod` varchar(10) NOT NULL,
`AdPrvCod` varchar(10) NOT NULL,
`AdCiuCod` varchar(10) NOT NULL,
`ParroquiaId` varchar(10) NOT NULL,
`NmEmpCed` varchar(13) NOT NULL,
`NmEmpApeNom` varchar(100) NOT NULL,
`NmEmpDirec` varchar(200) NOT NULL,
`NmEmpTelf1` varchar(15) NOT NULL,
`NmEmpTelf2` varchar(15) NOT NULL,
`NmEmpCel` varchar(10) NOT NULL,
`NmEmpContac` varchar(100) NOT NULL,
`NmEmpEmail` varchar(100) NOT NULL,
`NmEmpD4to` char(1) NOT NULL,
`NmEmpFoto` longblob NOT NULL,
`NmEmpFoto_GXI` varchar(2048) DEFAULT NULL,
PRIMARY KEY (`AmEmpId`,`NmEmpId`),
KEY `INMEM3` (`AmEmpId`,`NmCargosId`),
KEY `INMEM4` (`AmEmpId`,`NmEmpBancoId`),
KEY `INMEM5` (`AmEmpId`,`NmTipCid`),
KEY `INMEM2` (`AmEmpId`,`NmSecId`,`NmDepId`),
KEY `INMEM1` (`AmPaisCod`,`AdPrvCod`,`AdCiuCod`,`ParroquiaId`),
KEY `UNMEMPLEADO1` (`AmEmpId`,`NmEmpApeNom`),
CONSTRAINT `INMEM1` FOREIGN KEY (`AmPaisCod`,`AdPrvCod`,`AdCiuCod`,`ParroquiaId`) REFERENCES `adparroquia` (`AmPaisCod`,`AdPrvCod`,`AdCiuCod`,`ParroquiaId`),
CONSTRAINT `INMEM2` FOREIGN KEY (`AmEmpId`,`NmSecId`,`NmDepId`) REFERENCES `nmdepartamentos` (`AmEmpId`,`NmSecId`,`NmDepId`),
CONSTRAINT `INMEM3` FOREIGN KEY (`AmEmpId`,`NmCargosId`) REFERENCES `nmcargos` (`AmEmpId`,`NmCargosId`),
CONSTRAINT `INMEM4` FOREIGN KEY (`AmEmpId`,`NmEmpBancoId`) REFERENCES `nmempbanco` (`AmEmpId`,`NmEmpBancoId`),
CONSTRAINT `INMEM5` FOREIGN KEY (`AmEmpId`,`NmTipCid`) REFERENCES `nmtipcontrato` (`AmEmpId`,`NmTipCid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

-----  
-- Table structure for nminstruccion  
-----

---

```

DROP TABLE IF EXISTS `nminstruccion`;
CREATE TABLE `nminstruccion` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpId` smallint(6) NOT NULL,
  `NmInstruccionId` smallint(6) NOT NULL,
  `NmTituloId` smallint(6) NOT NULL,
  `NmInstruccionDesc` varchar(100) NOT NULL,
  `NmInstruccionObs` varchar(200) NOT NULL,
  `NmInstruccionUsr` smallint(6) NOT NULL,
  `NmInstruccionFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmEmpId`,`NmInstruccionId`),
  KEY `INMINSTRUCCION1` (`AmEmpId`,`NmTituloId`),
  CONSTRAINT `INMINSTRUCCION1` FOREIGN KEY (`AmEmpId`,`NmTituloId`) REFERENCES `nmtitulo` (`AmEmpId`,`NmTituloId`),
  CONSTRAINT `INMINSTRUCCION2` FOREIGN KEY (`AmEmpId`,`NmEmpId`) REFERENCES `nmempleado` (`AmEmpId`,`NmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nmpreguntas
-----
DROP TABLE IF EXISTS `nmpreguntas`;
CREATE TABLE `nmpreguntas` (
  `NmPreguntasId` smallint(6) NOT NULL,
  `NmPreguntasDesc` varchar(200) NOT NULL,
  `NmPreguntasRespuesta` char(3) NOT NULL,
  `NmPreguntasEst` char(1) NOT NULL,
  `NmPreguntasUsr` smallint(6) NOT NULL,
  `NmPreguntasFhr` datetime NOT NULL,
  PRIMARY KEY (`NmPreguntasId`),
  KEY `UNMPREGUNTAS1` (`NmPreguntasDesc`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nmsecciones
-----
DROP TABLE IF EXISTS `nmsecciones`;
CREATE TABLE `nmsecciones` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmSecId` smallint(6) NOT NULL,
  `NmSecDesc` varchar(100) NOT NULL,
  `NmSecFhr` datetime NOT NULL,
  `NmSecUsr` smallint(6) NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmSecId`),
  CONSTRAINT `INMSEC1` FOREIGN KEY (`AmEmpId`) REFERENCES `amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
-- Table structure for nmsecnmdep
-----
DROP TABLE IF EXISTS `nmsecnmdep`;

```

---

```

CREATE TABLE `nmsecnmdep` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmSecId` smallint(6) NOT NULL,
  `NmDepId` smallint(6) NOT NULL,
  `NmDepDesc` varchar(40) NOT NULL,
  `CmCenCosCod` varchar(50) NOT NULL,
  `NmDepUsr` varchar(20) NOT NULL,
  `NmDepFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmSecId`,`NmDepId`),
  KEY `INMSECNMDEP1` (`AmEmpId`,`CmCenCosCod`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Table structure for nmtipcontrato
-----
DROP TABLE IF EXISTS `nmtipcontrato`;
CREATE TABLE `nmtipcontrato` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmTipCId` smallint(6) NOT NULL,
  `NmTipCDesc` varchar(100) NOT NULL,
  `NmTipCForDoc` longblob NOT NULL,
  `NmTipCEst` char(1) NOT NULL,
  `NmTipCUsr` smallint(6) NOT NULL,
  `NmTipCFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmTipCId`),
  CONSTRAINT `INMTIPC1` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Table structure for nmtitulo
-----
DROP TABLE IF EXISTS `nmtitulo`;
CREATE TABLE `nmtitulo` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmTituloId` smallint(6) NOT NULL,
  `NmTituloDesc` varchar(100) NOT NULL,
  `NmTituloUsr` smallint(6) NOT NULL,
  `NmTituloFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`NmTituloId`),
  CONSTRAINT `INMTITULO1` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Table structure for nmtransacciones
-----
DROP TABLE IF EXISTS `nmtransacciones`;
CREATE TABLE `nmtransacciones` (
  `AmEmpId` smallint(6) NOT NULL,
  `NmEmpId` smallint(6) NOT NULL,
  `NmConceptoId` smallint(6) NOT NULL,

```

```

`NmTransacReg` char(1) NOT NULL,
`NmTransacTip` char(1) NOT NULL,
`NmTransacVal` decimal(17,2) NOT NULL,
`NmTransacVal2` decimal(17,2) NOT NULL,
`NmTransacIess` char(1) NOT NULL,
`NmTransacRenta` char(1) NOT NULL,
`NmTransacFondo` char(1) NOT NULL,
`NmTransacIngreso` smallint(6) NOT NULL,
`NmTransacFech` date NOT NULL,
`NmTransacForPag` char(3) NOT NULL,
`NmTransacUsr` smallint(6) NOT NULL,
`NmTransacFhr` datetime NOT NULL,
`NmTransacFhrMod` datetime NOT NULL,
`NmTransacUsrMod` smallint(6) NOT NULL,
`NmTransacTipConc` char(1) NOT NULL,
PRIMARY KEY (`AmEmpId`,`NmEmpId`,`NmConceptoId`,`NmTransacReg`),
KEY `INMTRANSACCIONES1` (`AmEmpId`,`NmConceptoId`),
CONSTRAINT `INMTRANSACCIONES1` FOREIGN KEY (`AmEmpId`,
`NmConceptoId`) REFERENCES `nmconcepto` (`AmEmpId`,`NmConceptoId`),
CONSTRAINT `INMTRANSACCIONES2` FOREIGN KEY (`AmEmpId`,
`NmEmpId`) REFERENCES `nmempleado` (`AmEmpId`,`NmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

**Código utilizado en el desarrollo de la aplicación**

---

## Bibliografía

- Adriana, Z. (09 de 07 de 2013). *Slideshare*. Obtenido de <http://es.slideshare.net/adrianazamora/diseo-de-interfaz-importancia-y-proceso-24058181>
- Bueno, C. B. (2014). *Ingeniería del software II*. Obtenido de <http://ocw.unican.es/enseanzas-tecnicas/ingenieria-del-software-ii/materiales/tema1-pruebasSistemasSoftware.pdf>
- Catarina. (2014). *catarina.udlap.mx/*. Obtenido de [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/viveros\\_s\\_ca/capitulo6.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros_s_ca/capitulo6.pdf)
- Gonzales. (12 de 12 de 2014). *Juanjo's Blog*. Obtenido de <https://jjegonzalezf.wordpress.com/>
- Gutierrez, T. (08 de 10 de 2012). *ALTO NIVEL*. Obtenido de <http://www.altonivel.com.mx/24170-control-de-nomina-un-factor-que-alienta-tu-productividad.html>
- MAG. (2014). *Estandares de programación*. Obtenido de <http://sistemas.mag.go.cr/SoporteTecnico/Est%C3%A1ndares%20de%20Sistemas.pdf>
- Ruiz, G. (2014). *galeon.com*. Obtenido de <http://gabrielruiz.galeon.com/>
- UTN. (2015). *www.utn.edu.ec/*. Obtenido de [http://www.utn.edu.ec/financiero/?page\\_id=228](http://www.utn.edu.ec/financiero/?page_id=228)