

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

DESARROLLO DE UN SISTEMA DE ADMINISTRACIÓN DE NOMINA PARA UNA PYME.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

ESTEBAN SANTIAGO CASTILLO PILACUAN

esteban_castillo@hotmail.com

MAURICIO GONZALO OVIEDO IGLESIAS

magovi8@hotmail.com

DIRECTOR: ING. PAUL FERNANDO VILCA CHILQUINGA

pavich_2500@yahoo.es

Quito, Marzo 2011

DECLARACIÓN

Nosotros, Esteban Santiago Castillo Pilacúan y Gonzalo Mauricio Oviedo Iglesias, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en éste documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a éste trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Esteban Santiago Castillo Pilacúan

Mauricio Gonzalo Oviedo Iglesias

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Esteban Santiago Castillo Pilacúan y Mauricio Gonzalo Oviedo Iglesias, bajo mi supervisión.

**Ing. Paúl Fernando Vilca
Chiliquinga
DIRECTOR DE PROYECTO**

AGRADECIMIENTO

Agradezco a Dios todopoderoso que me ayudado a tener salud y que siempre me ha dado la fuerza para continuar con las metas que tengo en mi vida.

Agradezco en especial a mis padres y hermanos que en todo momento me han apoyado para seguir adelante.

A mi sobrina Dayana que es la luz que impulsa mi vida.

Agradezco a una persona especial Marthita, quién fue mi impulsó para seguir siempre adelante y a no decaer de mis sueños y metas.

A la Escuela Politécnica Nacional y a mis amigos con quienes compartimos momentos inolvidables.

Al Ing. Paúl Vilca director de nuestra tesis, quien hizo posible este trabajo.

Esteban

Agradezco a mi familia por apoyarme siempre en cada momento de mi vida a mi hermana Katty y a mí cuñado Fabricio quienes con esfuerzo y dedicación me ayudaron a culminar mi carrera universitaria.

A Santiago un excelente compañero de la U, gran participe de este proyecto.

A la poli y a los amigos que hice ahí que pasamos momentos inolvidables haciendo comfortable el paso por la u.

Al Ing. Paul Vilca que nos brindo gran apoyo con la tesis.

Mauricio

DEDICATORIA

Dedico esta tesis el regalo más grande que Dios me ha dado, a mis padres, por apoyarme siempre y tenerme mucha paciencia.

A mis hermanos, por su confianza y fe, que refleja el profundo amor que compartimos siempre, por el apoyo y ánimo incondicional que me brindan día a día para alcanzar nuevas metas.

Esteban

Dedico este trabajo a mi hermana amada Katty Oviedo que sin ella no lo hubiera logrado, con su paciencia y apoyo, me dio cada día ánimos para seguir.

Mauricio

CONTENIDO

CAPITULO I. PLANTEAMIENTO DEL PROBLEMA	1
1.1 DESCRIPCIÓN DEL PROBLEMA.	2
1.2 DETERMINACIÓN DE LA METODOLOGÍA DE DESARROLLO A UTILIZAR.	6
1.2.1 <i>SElección de la metodología de desarrollo.</i>	6
1.2.1.1 Análisis de Metodologías de Desarrollo.	6
1.2.1.2 Descripción de Metodologías de Desarrollo.	9
1.2.1.2.1 RUP (Rational Unified Process)	9
1.2.1.2.2 Microsoft Solution Framework (MSF)	10
1.2.1.2.3 XP (Xtreme Programming)	12
1.2.1.2.4 RAD (Desarrollo Rápido de Aplicaciones).....	13
1.2.1.3 Justificación de la Metodología seleccionada.	14
1.2.1.3.1 Análisis comparativo de las metodologías revisadas.	14
1.2.1.3.2 Evaluación y selección de la metodología a usarse para el proyecto.....	15
1.3 DETERMINACIÓN DE LA HERRAMIENTA A UTILIZAR.	16
1.3.1 <i>Elección de la herramienta de desarrollo.</i>	16
1.2.1.1 Descripción de herramientas tipo front end y back end	16
1.2.1.2 Selección de la herramienta de desarrollo.....	19
CAPITULO II. DESARROLLO DEL SISTEMA DE ADMINISTRACION DE NOMINA	20
2.1 ANALISIS DEL SISTEMA.	20
2.1.1 <i>REQUERIMIENTOS</i>	20
2.2.1.1 Documento Visión [].....	20
2.1.1.1.1 Propósito.....	20
2.1.1.1.2 Alcance.....	20
2.1.1.1.3 Determinación de los requisitos funcionales	21
2.1.1.1.4 Restricciones.	21
2.1.1.1.5 Descripción global del producto.	21
2.1.1.1.6 Terminología.....	22
2.1.2 <i>MODELO DEL NEGOCIO</i>	25
2.2.1.1 Diccionario de Actores	26
2.2.1.2 Catálogo de Casos de Uso del Negocio	26
2.1.3 <i>MODELO DE CASOS DE USO DEL SISTEMA []</i>	28
2.2.1.1 Diagrama Completo del Sistema	28
2.2.1.2 Módulo Administrar Accesos	29
2.1.3.2.1 CU:01 Gestionar Perfiles.	29
2.1.3.2.2 CU:02 Gestionar Usuarios.	31
2.2.1.3 Módulo Configurar Parámetros	33
2.1.3.3.1 CU:03 Gestionar Tipo de Contrato.	34

2.1.3.3.2	CU:04 Gestionar Banco	36
2.1.3.3.3	CU:05 Gestionar Tipo de Pago	38
2.1.3.3.4	CU:06 Gestionar Cargo.....	41
2.2.1.4	Módulo Gestionar Empleado	43
2.1.3.4.1	CU:07 Gestionar Empleado	43
2.2.1.5	Módulo Generar Rol de Pago	45
2.1.3.5.1	CU:08 Gestionar Cierre de Mes.....	45
2.1.3.5.2	CU: 09 Gestionar Vacaciones	48
2.1.3.5.3	CU:10 Gestionar Horas Extras.....	49
2.1.3.5.4	CU:11 Gestionar Rol de Pago	51
2.1.3.5.5	CU:12 Gestionar Beneficios.....	52
2.2.1.6	Módulo Generar Finiquito.....	53
2.1.3.6.1	CU:13 Generar Finiquito	53
2.2.1.7	Módulo Generar Reportes	54
2.1.3.7.1	CU:14 Generar Reportes	54
2.1.4	<i>DIAGRAMAS DE COLABORACION.....</i>	<i>57</i>
2.2.1.1	Módulo Administrar Accesos	57
2.1.4.1.1	CU:01 Gestionar Perfiles	57
2.1.4.1.2	CU:02 Gestionar Usuarios	58
2.2.1.2	Módulo Configurar Parámetros	60
2.1.4.2.1	CU:03 Gestionar Tipo de Contrato	60
2.1.4.2.2	CU:04 Gestionar Banco	61
2.1.4.2.3	CU:05 Gestionar Tipo de Pago	63
2.1.4.2.4	CU:06 Gestionar Cargo.....	64
2.2.1.3	Módulo Gestionar Empleado	66
2.1.4.3.1	CU:07 Gestionar Empleado	66
2.2.1.4	Módulo Generar Rol de Pago	67
2.1.4.4.1	CU:08 Gestionar Cierre de Mes.....	67
2.1.4.4.2	CU:09 Gestionar Vacaciones	69
2.1.4.4.3	CU:10 Gestionar Horas Extras.....	70
2.1.4.4.4	CU:11 Gestionar Rol de Pago	71
2.1.4.4.5	CU:12 Gestionar Beneficios.....	71
2.2.1.5	Módulo Generar Finiquito.....	72
2.1.4.5.1	CU:13 Generar Finiquito	72
2.2.1.6	Módulo Generar Reportes	74
2.1.4.6.1	CU:14 Generar Reportes	74
2.2	DISEÑO DEL SISTEMA	75
2.2.1	<i>DIAGRAMAS DE SECUENCIA. [].....</i>	<i>75</i>
2.2.1.1	Módulo Administrar Accesos	75
2.2.1.1.1	CU:01 Gestionar Perfiles	75
2.2.1.1.2	CU:02 Gestionar Usuarios	76

2.2.1.2	Módulo Configurar Parámetros.	77
2.2.1.2.1	CU:03 Gestionar Tipo de Contrato.	77
2.2.1.2.2	CU:04 Gestionar Banco.	78
2.2.1.2.3	CU:05 Gestionar Tipo de Pago.	78
2.2.1.2.4	CU:06 Gestionar Cargo.	79
2.2.1.3	Módulo Gestionar Empleado.	80
2.2.1.3.1	CU:07 Gestionar Empleado.	80
2.2.1.4	Módulo Generar Rol de Pago.	81
2.2.1.4.1	CU:08 Gestionar Cierre de Mes.	81
2.2.1.4.2	CU:09 Gestionar Vacaciones.	82
2.2.1.4.3	CU:10 Gestionar Horas Extras.	83
2.2.1.4.4	CU:11 Gestionar Rol de Pago.	83
2.2.1.4.5	CU:12 Gestionar Beneficios.	84
2.2.1.5	Módulo Generar Finiquito.	84
2.2.1.5.1	CU:13 Generar Finiquito.	84
2.2.1.6	Módulo Generar Reportes.	85
2.2.1.6.1	CU:14 Generar Reportes.	85
2.2.2	<i>DISEÑO DEL DIAGRAMA DE CLASES.</i>	<i>87</i>
2.2.3	<i>DISEÑO DEL MODELO LOGICO E/R.</i>	<i>91</i>
2.2.4	<i>DISEÑO DEL MODELO E/R FISICO.</i>	<i>96</i>
2.2.5	<i>DISEÑO DE COMPONENTES.</i>	<i>100</i>
2.2.3.1	Módulos del Sistema.	100
2.2.3.2	Módulo Administrar Accesos.	100
2.2.3.3	Módulo Configurar Parámetros.	101
2.2.3.4	Módulo Gestionar Empleado.	101
2.2.3.5	Módulo Generar Rol de Pago.	102
2.2.3.6	Módulo Generar Finiquito.	102
2.2.3.7	Módulo Generar Reportes.	103
2.2.6	<i>DISEÑO DE LA ARQUITECTURA.</i>	<i>103</i>
2.2.3.1	Arquitectura Web.	103
2.2.7	<i>DISEÑO DE LA JERARQUIA DE MENU DEL SISTEMA.</i>	<i>105</i>
2.3	IMPLEMENTACIÓN DEL SISTEMA.	106
2.3.1	<i>Estandares DE programacion.</i>	<i>106</i>
2.3.3.1	Estándar para nombrar clases.	106
2.3.3.2	Estándar para declaración de controles.	106
2.3.3.3	Estándar para declaración de variables.	107
2.3.3.4	Estándar para la base de datos.	108
2.3.3.5	Estándar para diseñar pantallas.	108
2.3.2	<i>REQUERIMIENTOS DE HARDWARE.</i>	<i>109</i>
CAPITULO III. APLICACIÓN A UN CASO DE ESTUDIO		111

3.1	PREPARACIÓN DEL CASO DE ESTUDIO	111
3.2	IMPLEMENTACION Y PRUEBAS DEL CASO DE ESTUDIO.....	111
3.2.1	<i>Casos de Prueba</i>	112
3.2.3.1	Prueba del caso de uso: Gestionar Perfiles.	112
3.2.3.2	Prueba del caso de uso: Gestionar Usuarios	113
3.2.3.3	Prueba del caso de uso: Gestionar Tipo de Contrato	114
3.2.3.4	Prueba del caso de uso: Gestionar Bancos.....	115
3.2.3.5	Prueba del caso de uso: Gestionar Tipo de Pago	116
3.2.3.6	Prueba del caso de uso: Gestionar Cargo.....	117
3.2.3.7	Prueba del caso de uso: Gestionar Empleado	118
3.2.3.8	Prueba del caso de uso: Gestionar Cierre de Mes.....	119
3.2.3.9	Prueba del caso de uso: Gestionar Vacaciones	120
3.2.3.10	Prueba del caso de uso: Gestionar Horas Extras	121
3.2.3.11	Prueba del caso de uso: Gestionar Rol de Pago	122
3.2.3.12	Prueba del caso de uso: Gestionar Beneficios.....	122
3.2.3.13	Prueba del caso de uso: Generar Finiquito	123
3.2.3.14	Prueba del caso de uso: Gestionar Reportes.....	124
3.2.2	<i>PruebaS DE USABILIDAD</i>	125
3.2.1.1	Objetivos	125
3.2.1.2	Usuarios de pruebas.....	125
3.2.1.3	Ambiente de pruebas.....	125
3.2.1.4	Plan de pruebas.....	126
3.2.1.5	Resultados de la evaluación	126
CAPITULO IV.	CONCLUSIONES Y RECOMENDACIONES.	129
4.1	CONCLUSIONES.....	129
4.2	RECOMENDACIONES.	131

ÍNDICE DE TABLAS

CAPITULO I

TABLA 1 - 1 :	DIFERENCIAS ENTRE METODOLOGÍAS ÁGILES Y TRADICIONALES.....	8
TABLA 1 - 2 :	COMPARACIÓN DE CRITERIOS DE LAS METODOLOGÍAS	15
TABLA 1 - 3 :	PONDERACIÓN DE LAS METODOLOGÍAS.....	16

CAPITULO II

TABLA 2 - 1:	DICCIONARIO DE ACTORES	26
TABLA 2 - 2:	CATÁLOGOS DE CASOS DE USO DEL NEGOCIO	27
TABLA 2 - 3:	CU:01 GESTIONAR PERFILES.....	31
TABLA 2 - 4:	CU:02 GESTIONAR USUARIOS	33
TABLA 2 - 5:	CU:03 GESTIONAR TIPO DE CONTRATO	36
TABLA 2 - 6:	CU:04 GESTIONAR BANCO	38
TABLA 2 - 7:	CU:05 GESTIONAR TIPO DE PAGO	41
TABLA 2 - 8:	CU:06 GESTIONAR CARGO	43
TABLA 2 - 9:	CU:06 CU:07 GESTIONAR EMPLEADO	45
TABLA 2 - 10:	CU:08 GESTIONAR CIERRE DE MES.....	48
TABLA 2 - 11:	CU:09 GESTIONAR VACACIONES	49
TABLA 2 - 12:	CU:10 GESTIONAR HORAS EXTRAS.....	51
TABLA 2 - 13:	CU:11 GESTIONAR ROL DE PAGO	52
TABLA 2 - 14:	CU:12 GESTIONAR BENEFICIOS	53
TABLA 2 - 15:	CU:13 GESTIONAR FINIQUITO	54
TABLA 2 - 16:	CU:14 GENERAR REPORTE.....	56
TABLA 2 - 17:	FLUJO DE SUCEOS:01 GESTIONAR PERFILES	58
TABLA 2 - 18:	FLUJO DE SUCEOS:02 GESTIONAR USUARIOS.....	60
TABLA 2 - 19:	FLUJO DE SUCEOS:03 GESTIONAR TIPO DE CONTRATO	61
TABLA 2 - 20:	FLUJO DE SUCEOS:04 GESTIONAR BANCO	63
TABLA 2 - 21:	FLUJO DE SUCEOS:05 GESTIONAR TIPO DE PAGO.....	64
TABLA 2 - 22:	FLUJO DE SUCEOS:06 GESTIONAR CARGO	66
TABLA 2 - 23:	FLUJO DE SUCEOS:07 GESTIONAR EMPLEADO	67
TABLA 2 - 24:	FLUJO DE SUCEOS:08 GESTIONAR CIERRE DE MES	69
TABLA 2 - 25:	FLUJO DE SUCEOS:09 GESTIONAR VACACIONES.....	70
TABLA 2 - 26:	FLUJO DE SUCEOS:10 GESTIONAR HORAS EXTRAS	71
TABLA 2 - 27:	FLUJO DE SUCEOS:11 GESTIONAR ROL DE PAGO.....	71
TABLA 2 - 28:	FLUJO DE SUCEOS:12 GESTIONAR BENEFICIOS.....	72

TABLA 2 - 29:	FLUJO DE SUCEOS:13 GENERAR FINIQUITO.....	74
TABLA 2 - 30:	FLUJO DE SUCEOS:14 GENERAR REPORTES	74
TABLA 2 - 31:	DIAGRAMA DE CLASES.....	90
TABLA 2 - 32:	RELACIÓN DIAGRAMA DE CLASES – MODELO E/R.....	91
TABLA 2 - 33:	ESTÁNDAR PARA NOMBRAR CLASES	106
TABLA 2 - 34:	ABREVIATURAS PARA DECLARACIÓN DE CONTROLES.....	107
TABLA 2 - 35:	ESTÁNDAR PARA NOMBRAR VARIABLES	108
TABLA 2 - 36:	ESTÁNDAR PARA LA BASE DE DATOS	108
TABLA 2 - 37:	CRITERIOS PARA DISEÑAR PANTALLAS	109
TABLA 2 - 38:	REQUERIMIENTOS MÍNIMOS DE HARDWARE DEL SERVIDOR	109
TABLA 2 - 39:	REQUERIMIENTOS MÍNIMOS DE HARDWARE DEL CLIENTE.....	110
TABLA 2 - 40:	CASO DE ESTUDIO: INSTALACIÓN DEL SERVIDOR.....	111
TABLA 2 - 41:	INTERPRETACIÓN DE LA RESPUESTAS A LAS PREGUNTAS DE USABILIDAD.....	128

CAPITULO III

TABLA 3 - 1:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:01.- GESTIONAR PERFILES	113
TABLA 3 - 2:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:02.- GESTIONAR USUARIOS.....	114
TABLA 3 - 3:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:03.- GESTIONAR TIPO DE CONTRATO	115
TABLA 3 - 4:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:04.- GESTIONAR BANCOS	116
TABLA 3 - 5:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:05.- GESTIONAR TIPO DE PAGO.....	117
TABLA 3 - 6:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:06.- GESTIONAR CARGO	118
TABLA 3 - 7:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:07.- GESTIONAR EMPLEADO	119
TABLA 3 - 8:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:08.- GESTIONAR CIERRE DE MES	120
TABLA 3 - 9:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:09.- GESTIONAR VACACIONES.....	121
TABLA 3 - 10:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:10.- GESTIONAR HORAS EXTRAS	122
TABLA 3 - 11:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:11.- GESTIONAR HORAS EXTRAS	122
TABLA 3 - 12:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:12.- GESTIONAR BENEFICIOS.....	123
TABLA 3 - 13:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:13.- GENERAR FINIQUITO	124
TABLA 3 - 14:	DESCRIPCIÓN DEL CASO DE PRUEBA CU:14.- GESTIONAR REPORTES	125

ÍNDICE DE FIGURAS

CAPITULO I

FIGURA 1 - 1 :	CICLO DE VIDA RUP []	10
FIGURA 1 - 2 :	CICLO DE VIDA MSF []	11

CAPITULO II

FIGURA 2 - 1:	MODELO DEL NEGOCIO	25
FIGURA 2 - 2:	DIAGRAMA COMPLETO DEL SISTEMA.....	28
FIGURA 2 - 3:	CU:01 GESTIONAR PERFILES.....	29
FIGURA 2 - 4:	CU:02 GESTIONAR USUARIOS	31
FIGURA 2 - 5:	CU:03 GESTIONAR TIPO DE CONTRATO	34
FIGURA 2 - 6:	CU:04 GESTIONAR BANCO	36
FIGURA 2 - 7:	CU:05 GESTIONAR TIPO DE PAGO	38
FIGURA 2 - 8:	CU:06 GESTIONAR CARGO	41
FIGURA 2 - 9:	CU:07 GESTIONAR EMPLEADO.....	43
FIGURA 2 - 10:	CU:08 GESTIONAR CIERRE DE MES.....	46
FIGURA 2 - 11:	CU:09 GESTIONAR VACACIONES	48
FIGURA 2 - 12:	CU:10 GESTIONAR HORAS EXTRAS.....	49
FIGURA 2 - 13:	CU:11 GESTIONAR ROL DE PAGO	51
FIGURA 2 - 14:	CU:12 GESTIONAR BENEFICIOS	52
FIGURA 2 - 15:	CU:13 GENERAR FINIQUITO	53
FIGURA 2 - 16:	CU:14 GENERAR REPORTES.....	55
FIGURA 2 - 17:	DIAGRAMA COLABORACIÓN:01 GESTIONAR PERFILES	57
FIGURA 2 - 18:	DIAGRAMA COLABORACIÓN:02 GESTIONAR USUARIOS.....	59
FIGURA 2 - 19:	DIAGRAMA COLABORACIÓN:03 GESTIONAR TIPO DE CONTRATO	60
FIGURA 2 - 20:	DIAGRAMA COLABORACIÓN:04 GESTIONAR BANCO	62
FIGURA 2 - 21:	DIAGRAMA COLABORACIÓN:05 GESTIONAR TIPO DE PAGO.....	63
FIGURA 2 - 22:	DIAGRAMA COLABORACIÓN:06 GESTIONAR CARGO	65
FIGURA 2 - 23:	DIAGRAMA COLABORACIÓN:07 GESTIONAR EMPLEADO.....	66
FIGURA 2 - 24:	DIAGRAMA COLABORACIÓN:08 GESTIONAR CIERRE DE MES.....	68
FIGURA 2 - 25:	DIAGRAMA COLABORACIÓN:09 GESTIONAR VACACIONES	69
FIGURA 2 - 26:	DIAGRAMA COLABORACIÓN:10 GESTIONAR HORAS EXTRAS	70

FIGURA 2 - 27:	DIAGRAMA COLABORACIÓN:11 GESTIONAR ROL DE PAGO	71
FIGURA 2 - 28:	DIAGRAMA COLABORACIÓN:12 GESTIONAR BENEFICIOS	72
FIGURA 2 - 29:	DIAGRAMA COLABORACIÓN:13 GENERAR FINIQUITO	73
FIGURA 2 - 30:	DIAGRAMA COLABORACIÓN:14 GENERAR REPORTES	74
FIGURA 2 - 31:	DIAGRAMA SECUENCIA:01 GESTIONAR PERFILES	76
FIGURA 2 - 32:	DIAGRAMA SECUENCIA:02 GESTIONAR USUARIOS.....	76
FIGURA 2 - 33:	DIAGRAMA SECUENCIA:03 GESTIONAR TIPO DE CONTRATO	77
FIGURA 2 - 34:	DIAGRAMA SECUENCIA:04 GESTIONAR BANCO	78
FIGURA 2 - 35:	DIAGRAMA SECUENCIA:05 GESTIONAR TIPO DE PAGO.....	79
FIGURA 2 - 36:	DIAGRAMA SECUENCIA:06 GESTIONAR CARGO	80
FIGURA 2 - 37:	DIAGRAMA SECUENCIA:07 GESTIONAR EMPLEADO	81
FIGURA 2 - 38:	DIAGRAMA SECUENCIA:08 GESTIONAR CIERRE DE MES	82
FIGURA 2 - 39:	DIAGRAMA SECUENCIA:09 GESTIONAR VACACIONES.....	83
FIGURA 2 - 40:	DIAGRAMA SECUENCIA:10 GESTIONAR HORAS EXTRAS	83
FIGURA 2 - 41:	DIAGRAMA SECUENCIA:11 GESTIONAR ROL DE PAGO	84
FIGURA 2 - 42:	DIAGRAMA SECUENCIA:12 GESTIONAR BENEFICIOS.....	84
FIGURA 2 - 43:	DIAGRAMA SECUENCIA:13 GENERAR FINIQUITO	85
FIGURA 2 - 44:	DIAGRAMA SECUENCIA:14 GENERAR REPORTES	86
FIGURA 2 - 45:	MODELO LÓGICO ENTIDAD / RELACIÓN	95
FIGURA 2 - 46:	MODELO FÍSICO ENTIDAD / RELACIÓN	99
FIGURA 2 - 47:	DIAGRAMA DE COMPONENTES: MÓDULOS DEL SISTEMA	100
FIGURA 2 - 48:	DIAGRAMA DE COMPONENTES: MÓDULO ADMINISTRAR ACCESOS.....	101
FIGURA 2 - 49:	DIAGRAMA DE COMPONENTES: MÓDULO CONFIGURAR PARÁMETROS	101
FIGURA 2 - 50:	DIAGRAMA DE COMPONENTES: MÓDULO GESTIONAR EMPLEADO	102
FIGURA 2 - 51:	DIAGRAMA DE COMPONENTES: MÓDULO GENERAR ROL DE PAGO	102
FIGURA 2 - 52:	DIAGRAMA DE COMPONENTES: MÓDULO GENERAR FINIQUITO.....	102
FIGURA 2 - 53:	DIAGRAMA DE COMPONENTES: MÓDULO GENERAR REPORTES	103
FIGURA 2 - 54:	DISEÑO DE LA ARQUITECTURA: ARQUITECTURA WEB[].....	103
FIGURA 2 - 55:	DISEÑO DE LA JERARQUÍA DEL MENÚ DEL SISTEMA	105

INTRODUCCION

El presente trabajo tiene como objetivo principal desarrollar un sistema de Administración de Nómina para una PYME, el cual sea amigable con el usuario, práctico, confiable y que brinde seguridad de su información; además de que si existe un cambio en los cálculos relacionados con la nómina sea muy fácil y parametrizable el mantenimiento.

A continuación se detallará en forma breve y concisa el contenido de cada uno de los capítulos correspondientes al presente documento:

CAPITULO 1. PLANTEAMIENTO DEL PROBLEMA

La administración de nómina es el proceso de diseñar, organizar, implementar y controlar un sistema de pago veraz y oportuno de las remuneraciones y compensaciones al personal de la empresa, dando cumplimiento a las obligaciones laborales, fiscales y de seguridad social que a las que establecen las leyes del país.

El objetivo del proyecto es desarrollar una aplicación que resuelva de manera efectiva y eficaz todos los procesos de nómina para que las empresas puedan seguir manteniéndose, competitivamente en el mercado en el que se desenvuelven, y sobre todo porque el personal humano es el aspecto más importante de toda empresa y tienen el compromiso de satisfacer estas necesidades.

De acuerdo al análisis realizado se ha establecido que la metodología que compete a las necesidades del proyecto es la metodología RUP (Proceso Unificado de Desarrollo).

Cabe recalcar que RUP, define una gran cantidad de tareas y desarrollo de documentación, pero a pesar de esto RUP es una metodología adaptable a la

naturaleza de cada proyecto, permitiéndonos de esta manera utilizar la documentación necesaria.

Debido a que el presente proyecto se puede considerar pequeño, nos enfocaremos en los flujos de trabajo correspondientes a las disciplinas de SW; Requisitos, Análisis, Diseño, Implementación y Pruebas, prescindiendo del flujo de trabajo correspondiente a la gestión del proyecto.

CAPITULO 2. DESARROLLO DE LA APLICACIÓN

El propósito de éste documento es identificar los requisitos y restricciones del Sistema de Administración de nómina para una PYME. El documento se centra en la funcionalidad requerida por la empresa Beehunting en donde se va a implementar el sistema.

Esta funcionalidad se centra principalmente en la generación de roles de pagos y el cálculo de todos los procesos de nómina, además de la generación automática de finiquitos.

A continuación se describe detalladamente los requerimientos funcionales a través de diagramas de casos de uso y por cada caso de uso se realizó la respectiva especificación o descripción. Los casos de uso principales los siguientes:

- Administrar (Perfiles, Usuarios)
- Configurar Parámetros (Banco, Tipo de contrato, Tipo de Pago, Cargo).
- Configurar Empleado (Empleado).
- Generar Rol de Pago (Cierre mes, Vacaciones, Horas Extras, Beneficios de Ley)
- Generar Finiquito (Generar Finiquito y Reporte).
- Generar Reportes (Reporte Rol de Pago)

Para cada caso de uso se crearon diagramas de clase de análisis, identificando de esta manera las clases necesarias para el desarrollo de los casos de uso antes definidos.

A continuación se describen la interacción entre los objetos mediante diagramas de colaboración, por cada diagrama de colaboración se realiza la descripción del flujo de sucesos.

Para el caso del diseño de casos de uso se realizaron los diagramas de actividades y los diagramas de secuencia.

A través del Diseño se ha modelado un esquema, de manera que defina la arquitectura del sistema con el fin de satisfacer los requerimientos funcionales y no funcionales planteados.

La arquitectura del sistema correspondiente a la aplicación desarrollada consiste en una arquitectura web de 3 capas: la capa de presentación, la capa de lógica y la capa de datos.

CAPITULO 3. APLICACIÓN A UN CASO DE ESTUDIO

El caso de estudio se refiere a la implantación del sistema de administración de nómina para una PYME en la empresa Beehunting S.A.

El objetivo de realizar las pruebas de aplicación es el de validar y verificar que los requerimientos satisfacen el producto desarrollado, para esto se crearon pruebas de aplicación por cada caso de uso.

Por cada caso de uso se realizaron las respectivas pruebas y con los resultados obtenidos de las pruebas, se realizaron los cambios pertinentes y necesarios para asegurar que la aplicación cumpla con los planteamientos establecidos para cada caso de uso.

CAPITULO 4. CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES:

- ◆ La metodología de Proceso Unificado Rational (RUP) es una de las metodologías con más amplia disponibilidad de información técnica y práctica así como de las herramientas que permiten implementarla, lo que ayuda en la utilización en el desarrollo de sistemas.
- ◆ La ejecución de pruebas de usabilidad de un software son muy importantes debido a que nos permiten establecer el grado de aceptación de los usuarios con el sistema y el éxito que tendrá el mismo.
- ◆ El realizar un buen diseño minimiza el riesgo de no cumplir con los requerimientos funcionales y no funcionales del sistema, debido a que es la base para que un proyecto de desarrollo funcione y si no se tiene claro que se debe hacer es imposible que el cliente al final quede satisfecho.

RECOMENDACIONES:

- ◆ Es recomendable utilizar RUP para el desarrollo de sistemas especialmente aquellos de gran complejidad y criticidad debido a las facilidades que provee en la distribución de los flujos de trabajo.
- ◆ Es recomendable que durante la fase de análisis se tenga un alto grado de interacción entre los técnicos y los usuarios, de esta forma se podrá obtener un mayor detalle de los requerimientos funcionales del sistema.
- ◆ Se debe tratar de usar al máximo estándares a nivel de todas las etapas del desarrollo de un sistema, esto permite mayor portabilidad al mismo y facilita el mantenimiento.

CAPITULO I.

PLANTEAMIENTO DEL PROBLEMA

ANTECEDENTES.

BEEHUNTING S.A. ofrece un portafolio de servicios completos en Talento Humano, entre los cuáles se destacan:

- Servicios de publicación de anuncios laborales en línea.
- Servicios de personal a nivel administrativo y ejecutivo.
- Servicios de capacitación Incompany y personalizado a las necesidades de cada empresa.
- Servicios de consultoría en temas de Talento Humano tales como optimización de cultura organizacional y actitudes, Employer Branding, manual de seguridad, clima laboral, team builders, análisis de desempeño, valoración de cargos, entre otros.

La empresa cuenta con personal altamente calificado tanto de especialistas nacionales y extranjeros. Su compromiso es brindar servicios de calidad, adecuados a nuestra cultura e idiosincrasia.

A través de la división BeeMarketing, se ofrece el diseño y asesoramiento para campañas de E-Marketing, marketing relacional, servicios de Relaciones Públicas, Organización de eventos, entre otros.

Han pasado 15 años desde la creación de la empresa, el cual se crea un portal de ofertas de empleo www.porfinempleo.com, cuyo fin es reclutar personal para que sea utilizado como herramienta para recursos humanos.

Para ser competitiva en el mercado, este año realizaron un rediseño de su portal, el cual incluye la depuración total del sitio, contando con el esfuerzo y dedicación de su equipo de trabajo, entregando un producto terminado de categoría internacional.

La empresa actualmente lleva todos sus procesos de administración de nómina a través de métodos manuales como son: hojas de cálculo, el cual genera mayor tiempo en el proceso e imposibilita que sus empleados puedan revisar su información con mayor facilidad y rapidez.

La empresa a lo largo del tiempo ha adquirido experiencia y ha conseguido que las empresas confíen en ella, por lo que tiene la meta de optimizar todos sus procesos, por tal motivo requieren implementar un sistema interno de administración de nómina de empleados.

A continuación se verá la descripción del problema.

1.1 DESCRIPCIÓN DEL PROBLEMA.

1.1.1 LA EMPRESA.

Nombre: BEE HUNTING S.A.

Dirección: Veintimilla 1106 y Amazonas, Of. 601.

Teléfonos: (2) 2524619 / 2561223 / 22508325.

Ciudad: Quito

País: Ecuador

Mail de contacto: clientes@porfinempleo.com

Creación de la empresa: Año 1996

Objetivo: Lograr un cambio de actitud auténtico y definitivo por parte del personal para lograr una mayor productividad en el trabajo.

Conseguir el compromiso de superación personal y profesional por parte de los participantes en forma inmediata y mediante la liberación de condicionamientos y creencias que no les permiten alcanzar su máximo potencial. Dar a conocer la propuesta "Actitud diferente, resultados diferentes", exclusivamente desarrollada por el equipo profesional de BeeHunting S.A., como una alternativa válida para la identificación de los obstáculos que se interponen a la actitud adecuada y a la

efectiva toma de decisiones. Desatar el verdadero potencial creativo en los participantes, evitar la sistematización deshumanizante, y promover la atención absoluta del sujeto, libre de condicionamientos y distracciones contraproducentes.

El compromiso es brindar servicios de calidad, adecuados a nuestra cultura e idiosincrasia.

Misión

Proveer valor agregado en servicios relacionados con la gestión del talento humano a nuestros clientes, con los más altos estándares de calidad, en forma eficiente y eficaz con innovación constante.

Visión

Nuestra visión se enfoca en hacer de BeeHunting la Consultora de Talento Humano más grande del Ecuador, obteniendo estándares de calidad que nos permitan estar a la vanguardia e innovar permanentemente en las mejores prácticas del talento humano, tanto internamente como en todos los sectores de actividad y proyectarnos hacia el exterior.

1.1.2 SITUACIÓN ACTUAL DE LA EMPRESA.

La administración de nómina es el proceso de diseñar, organizar, implementar y controlar un sistema de pago veraz y oportuno de las remuneraciones y compensaciones al personal de la empresa, dando cumplimiento a las obligaciones laborales, fiscales y de seguridad social que a las que establecen las leyes del país, actualmente todos los procesos se llevan de forma manual (Hojas de cálculo excel), provocando tiempo de demoras, información no tan accesible.

Por tal motivo la empresa necesita optimizar recursos y llevar un mejor control de sus procesos e información; para ello se identificará la solución de software más óptima en la cual se aplicará los conocimientos académicos adquiridos durante todo el proceso de estudio.

La empresa cuenta con una página web muy bien diseñada y también con un portal de empleo, sin embargo se ha determinado que la empresa tiene algunas falencias en cuanto a los procesos internos los cuales se describen a continuación:

- Debido a que el recurso humano es el material más importante de las organizaciones, se detecta la necesidad de contar con un sistema automatizado para el control de personal y generación de rol de pagos mensual para sus empleados.
- La empresa cuenta actualmente con un sistema de nómina que se encuentra obsoleto y desactualizado siendo uno de los motivos para no utilizar este sistema.
- No existe seguimiento y control de los procesos, ni historiales sobre la información de sus empleados.
- Al momento de que un empleado sale por algún motivo de la empresa, no se cuenta con un proceso de finiquitos automatizado.
- Es importante señalar que los parámetros de nómina en el transcurso del tiempo se modifican continuamente, el sistema con que cuentan actualmente no permite actualizaciones fáciles de implementar y toman mucho tiempo y dinero realizar actualizaciones de su sistema.
- Actualmente no existen reportes de los roles de pagos generados automáticamente por el sistema, los cuales deben realizarlos en hojas de cálculo Excel.

Todas estas falencias que tiene la empresa deben ser resueltas de manera efectiva y eficaz para que pueda seguir manteniéndose, competitivamente en el mercado en el que esta se desenvuelve, y sobre todo por el aspecto más

importante de todos que es el compromiso que tiene de satisfacer las necesidades de su personal humano de una forma eficiente.

La gran mayoría de estos problemas pueden ser resueltos a través de la implementación un sistema de nómina el cual controle todos los procesos descritos anteriormente, y llevar un correcto seguimiento mensual de todos sus empleados.

1.2 DETERMINACIÓN DE LA METODOLOGÍA DE DESARROLLO A UTILIZAR.

Como en todo desarrollo de software es necesario determinar el proceso o metodología de desarrollo a seguirse para alcanzar, desde un punto de vista de ingeniería, dicho objetivo.

A continuación se detallará los factores tomados en cuenta para la elección tanto de la metodología o proceso de desarrollo como las herramientas de desarrollo usadas para el presente proyecto.

1.2.1 SELECCIÓN DE LA METODOLOGÍA DE DESARROLLO.

Para la determinación de la metodología se tomara en cuenta ciertos criterios relacionados con la empresa, que nos permitirán determinar la metodología más apropiada para continuar con el desarrollo e implementación del sistema.

Para ello se detallará un análisis de las metodologías de desarrollo candidatas a usarse en el presente proyecto, para posteriormente, en base a éstas metodologías, elegir la que será usada durante el presente proyecto.

1.2.1.1 Análisis de Metodologías de Desarrollo.

Definición

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Metodologías

Existen varios criterios para la elección de una metodología de desarrollo.

Paradigma de Desarrollo: Uno de estos criterios es el paradigma de desarrollo, mediante el cual podemos clasificar las metodologías en dos grupos.

- ◆ Metodologías Orientadas a Objetos.
- ◆ Metodologías Estructuradas.

Las metodologías usadas actualmente, tomando en cuenta éste criterio, son las metodologías Orientadas a Objetos.

Cantidad de formalidad o documentación requerida: Otro criterio aplicado es la cantidad de formalidad o documentación requerida por cada metodología, es decir podemos clasificar las metodologías en:

- ◆ Metodologías ágiles.
- ◆ Metodologías Tradicionales (o denominada Metodologías Pesadas, o Peso Pesado).

Las metodologías ágiles de desarrollo son caracterizadas por un enfoque altamente iterativo usando una mínima documentación y formalismo lo cual lo hace ideal para proyectos pequeños y menos complejos. Algunos de los enfoques ágiles como XP, le dan un mayor valor a la producción de software, mejor que la creación de documentación extensiva.

Por el contrario disponemos también de metodologías pesadas de desarrollo en los cuales se requiere de gran cantidad de documentación y formalismo, por lo que son aplicados en proyectos grandes y complejos.

De esta manera podríamos decidir escoger una metodología en base a la complejidad del proyecto, aunque muchas veces no siempre se aplica éste criterio ya que como se mencionó anteriormente la metodología debería adaptarse a la naturaleza del proyecto.

A continuación se detalla las principales diferencias entre las metodologías ágiles y tradicionales:

Metodologías Ágiles	Metodologías Tradicionales
---------------------	----------------------------

Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el Proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1 - 1 : Diferencias entre metodologías ágiles y tradicionales

Aplicando estos criterios podemos empezar a discernir la metodología que podría aplicarse al proyecto, es decir podríamos definir bajo que paradigma de desarrollo nos enfocaremos y la cantidad de formalidad requerida.

Previo a definir la metodología de desarrollo a aplicarse en este proyecto, se realizará una rápida descripción de varias metodologías, las cuales se ajustan a las necesidades del proyecto.

Posteriormente se procede a analizar la metodología que mejor se ajusta a las necesidades del presente proyecto y justificar la misma.

1.2.1.2 Descripción de Metodologías de Desarrollo.

Dentro del mundo de desarrollo de SW, existen varias metodologías que ofrecen diferentes contextos que se pueden seguir en cuanto a la ejecución de un proyecto.

Para efectos del presente proyecto, con la ayuda de artículos de libros e Internet, se presenta un cuadro comparativo con las metodologías de desarrollo de Software más utilizadas, de donde se tomara la más adecuada para continuar con el desarrollo e implementación del sistema objeto del presente proyecto.

1.2.1.2.1 RUP (*Rational Unified Process*)

RUP, es la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

“RUP^[1] es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto.

Contiene 3 características esenciales, está dirigido por los Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.”

El ciclo de vida de RUP

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades.

[1] Metodología RUP

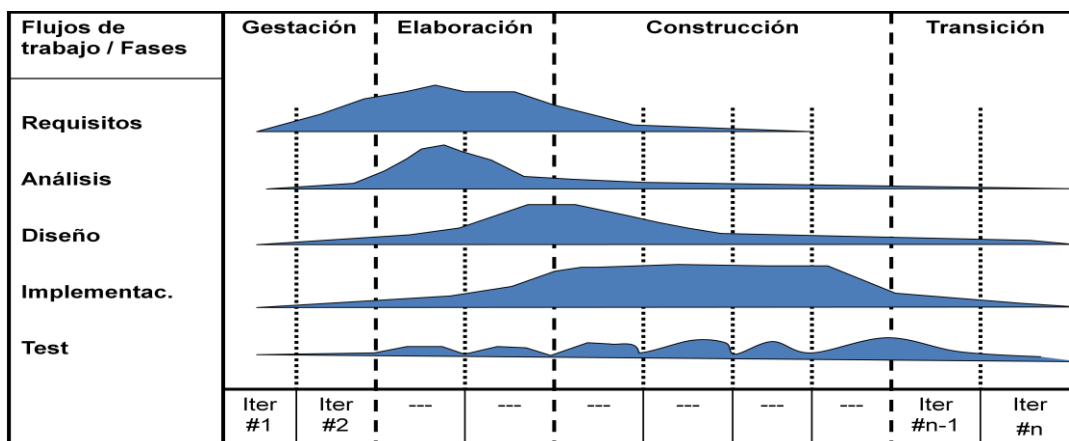


Figura 1 - 1 : Ciclo de Vida RUP [2]

En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades

- **Inicio:** Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

1.2.1.2.2 Microsoft Solution Framework (MSF)

“MSF [3] es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el

[2] Fases dentro de un ciclo de desarrollo

[3] Metodología MSF

desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.”

Además éste modelo provee un continuo control para las actividades del proyecto relacionadas con el diseño, a través del diseño conceptual, el diseño lógico y el diseño físico, de la aplicación que se está construyendo. Las fases y los documentos del diseño conceptual, lógico y físico, proveen tres perspectivas diferentes para cada una de las tres audiencias: los usuarios, el equipo y los desarrolladores.

Por lo tanto, el uso de éste modelo ayuda a garantizar que una aplicación no se desarrolle sólo para satisfacer una necesidad tecnológica sino también para cubrir las necesidades del negocio y de los usuarios.



Figura 1 - 2 : Ciclo de Vida MSF [4]

Como se observa en la figura las fases de MSF son:

- ◆ Previsión.
- ◆ Planeamiento.

[4] Ciclo de Vida MSF

- ◆ Desarrollo.
- ◆ Estabilización.
- ◆ Implementación.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

1.2.1.2.3 XP (Xtreme Programming)

“XP^[5] se basa en la idea de que existen cuatro variables que guían el desarrollo de sistemas: Costo, Tiempo, Calidad y Alcance. La manera de encarar los desarrollos avalados por éste modelo de desarrollo es permitir a las fuerzas externas (gerencia, clientes) manejar hasta tres de éstas variables, quedando el control de la restante en manos del equipo de desarrollo.”

Las características principales son:

- Desarrollo en iteraciones, en las que se añaden mejoras, generando las distintas versiones.
- Pruebas unitarias continuas, comprobando que la aplicación mantenga sus funcionalidades.
- Programación en parejas, haciendo que cada tarea sea desarrolle en parejas, permitiendo que el código se discuta y revise mientras se programa.
- Interacción entre los desarrolladores y el usuario, que ha de ser lo suficientemente frecuente para economizar el esfuerzo de ambas partes.
- Corrección de los errores antes de cada iteración, haciendo que la versión resultante de una iteración sea estable.

[5] Metodologías de desarrollo ligeras: XP Xtreme Programing

- Refactorización del código, haciéndolo más legible y mantenible, pero garantizando su funcionamiento manteniendo las pruebas unitarias.
- Propiedad del código compartida, de modo que cualquier integrante del proyecto pueda modificar código hecho por otro. Otra vez la existencia de errores se comprueba mediante las pruebas.
- Simplicidad del diseño, haciendo que sea fácil añadir las nuevas funcionalidades.

1.2.1.2.4 RAD (Desarrollo Rápido de Aplicaciones)

“RAD^[6] propone un proceso de desarrollo de software que permite que se creen sistemas de información utilizables en un periodo de tiempo entre 60 a 90 días. RAD es un ciclo de desarrollo diseñado para crear aplicaciones de alta calidad de las que acontecen en corporaciones grandes.”

Etapas del ciclo RAD

A continuación se detalla las etapas de la metodología RAD:

Etapa de planificación de los requisitos:

Esta etapa requiere que usuarios con un vasto conocimiento de los procesos de la compañía determinen cuales serán las funciones del sistema. Debe darse una discusión estructurada sobre los problemas de la compañía que necesitan solución.

Etapa de diseño:

Esta consiste de un análisis detallado de las actividades de la compañía en relación al sistema propuesto. Los usuarios participan activamente en talleres bajo la tutela de profesionales de la informática. En ellos descomponen funciones y definen entidades asociadas con el sistema. Una vez se completa el análisis se

[6] Metodologías de desarrollo ligeras: RAD Desarrollo de Aplicaciones Rápida

crean los diagramas que definen las alteraciones entre los procesos y la data. Al finalizar el análisis se traza el diseño del sistema. Se desarrollan los procedimientos y los esquemas de pantallas. Los prototipos de procedimientos críticos se construyen y se repasan y el plan para implementar el sistema se prepara.

Construcción:

En la etapa de construcción el equipo de desarrolladores trabajando de cerca con los usuarios finaliza el diseño y la construcción del sistema. La construcción de la aplicación consiste de una serie de pasos donde los usuarios tienen la oportunidad de afirmar los requisitos y repasar los resultados. Las pruebas al sistema se llevan a cabo durante esta etapa. También se crea la documentación y las instrucciones necesarias para manejar la nueva aplicación, rutinas y procedimientos para operar el sistema.

Implementación:

Esta etapa envuelve la implementación del nuevo producto y el manejo del cambio del viejo al nuevo sistema. Se hacen pruebas comprensivas y se adiestran los usuarios. Los cambios organizacionales y la operación del nuevo sistema se hacen en paralelo con el viejo sistema hasta que el nuevo se establezca completamente.

1.2.1.3 Justificación de la Metodología seleccionada.

A continuación realizaremos una comparación de las metodologías revisadas enfocándonos en el presente proyecto, analizando por cada una de éstas sus ventajas y desventajas.

1.2.1.3.1 Análisis comparativo de las metodologías revisadas.

Se detallará varios criterios, enfocados en el presente proyecto, los cuales nos permitirán establecer diferencias, ventajas y bondades de entre las distintas metodologías escogidas, y en base a esto determinar cuál es la que más se adapta al presente proyecto.

CRITERIOS	RUP	MSF	XP	RAD
RESISTENCIA A CAMBIOS	Cierta resistencia a los cambios	Cierta resistencia a los cambios	Preparado para cambios durante el proyecto	Apto para cambios durante el proyecto
ARTEFACTOS	Mas artefactos	Mas artefactos	Pocos artefactos	Pocos Artefactos
PROYECTOS/ PLAZOS	Más adaptable para proyectos de largo plazo.	Adaptable para proyectos de cualquier dimensión.	Recomendado para proyectos de corto plazo.	Recomendado para proyectos de corto plazo.
COMUNICACIÓN EFECTIVA	El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
GRUPO Y SITIO DE TRABAJO	Grupos grandes y posiblemente distribuidos	Grupos grandes y pequeños y trabajando en el mismo sitio	Grupos pequeños y trabajando en el mismo sitio	Grupos pequeños y trabajando en el mismo sitio

Tabla 1 - 2 : Comparación de criterios de las metodologías

1.2.1.3.2 Evaluación y selección de la metodología a usarse para el proyecto.

A continuación se procederá a escoger la metodología que será usada durante el desarrollo del presente proyecto.

Evaluación de la metodología

La Tabla 1-3 recoge algunos criterios de las metodologías investigadas. Estos criterios no sólo se refieren al proceso en sí, sino también al contexto del equipo y su organización.

Criterios	%	RUP	MSF	RAD	XP
	%	%	%	%	%
Grado de conocimiento	20	15	10	5	10
Soporte orientada a objetos	10	10	10	10	10
Adaptable a cambios	15	10	15	10	15
Basado en casos de uso	10	10	5	10	5
Posee documentación adecuada	15	15	15	15	8
Facilita la integración entre las etapas de desarrollo	10	10	10	10	10
Relación con UML	10	10	7	7	7
Permite desarrollar software sobre cualquier tecnología	10	10	10	10	10
	100%	90%	82%	77%	75%

Tabla 1 - 3 : Ponderación de las metodologías

Por el análisis realizado en la tabla 1-3 determinamos que la metodología RUP cumple con la puntuación más alta (90%) y por tanto es la escogida como la metodología para el desarrollo del presente proyecto.

1.3 DETERMINACIÓN DE LA HERRAMIENTA A UTILIZAR.

1.3.1 ELECCIÓN DE LA HERRAMIENTA DE DESARROLLO.

Para la determinación de las herramientas a utilizar se ha considerado los requerimientos de la empresa, la metodología y el conocimiento que se tiene de las mismas, razón por la cual en esta sección solo se hará una breve descripción y los criterios por los cuales se las ha seleccionado.

1.2.1.1 Descripción de herramientas tipo front end y back end

A continuación se presenta una breve descripción de estas:

Front-End [7] - Las herramientas de tipo front-end automatizan las primeras tareas del proceso de desarrollo de sistemas, incluidas el análisis de requerimientos y el diseño lógico. A menudo estas herramientas soportan la preparación de modelos gráficos, como diagramas de flujo de datos, que documentan procesos y actividades.

Back-End.- Las herramientas de tipo back-end brindan apoyo en la formulación de la lógica del programa, de algoritmos de procesamiento y otros detalles relacionados con el procesamiento por computadora. En algunas ocasiones, estas herramientas se conocen como herramientas de programación asistidas por computadora, ya que ayudan a preparar el software para la computadora y el código del programa.

HERRAMIENTA PARA MODELADO UML

Rational Rose

Rational Rose^[8] esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Rational Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

HERRAMIENTA PARA MODELADO DE BASE DE DATOS

Power Designer

[7] Front-End, Back-End
[8] Rational Rose

“Power Designer^[9] desarrollado por Sybase cuenta con herramientas para la creación y control de diagramas como la representación de los flujos de entradas y salidas en un proceso, define las reglas de uso para procesos, almacenamiento de datos, entidades externas, y flujos de datos.”

Permite a los desarrolladores: diseñar modelos de bases de datos físicas o crearlas instantáneamente a través de la ingeniería de reversa de bases de datos existentes, generar, documentar y mantener bases de datos, generar rápidamente objetos de aplicación y componentes de datos.

Proporciona Ingeniería hacia delante e Ingeniería Inversa, es decir, visualiza estructuras de bases de datos existentes directamente vía ODBC o usando archivos de script DDL. Genera el modelo conceptual del modelo físico. Retro-documentación de bases de datos existentes. Re-orientación de la base de datos existente a un DBMS diferente.

HERRAMIENTA DE DESARROLLO DE APLICACIONES

Visual C# .Net 2005

Visual C# .Net 2005 es parte de la suite Visual Studio 2005 de Microsoft la cual permite construir aplicaciones multi-capas, Web, dispositivos móviles, clientes inteligentes.

Posee nuevas herramientas de diseño y comunicación de bases de datos visuales e integrados que facilitan la elaboración de aplicaciones fiables, escalables, orientadas a los datos para SQL Server y otras bases de datos.

ADMINISTRADOR DE BASE DE DATOS

MS Sql Server 005

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-

[9] Power Designer

SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o PostgreSQL o MySQL.

1.2.1.2 Selección de la herramienta de desarrollo

Por las características descritas anteriormente y por el conocimiento adquirido en el transcurso de pregrado elegimos utilizar las siguientes herramientas de desarrollo.

La herramienta para modelado UML que se ha elegido es Rational Rose dado que la metodología elegida para el presente proyecto tiene como uno de sus objetivos seleccionar herramientas con la que se tenga más experiencia y permita mayor agilidad para el desarrollo del modelado que posteriormente nos permita desarrollar el sistema a un corto plazo y por las mismas razones se ha seleccionado la herramienta Power Designer ya que posee características propias que nos facilitara cumplir los objetivos.

Para la selección de la herramienta de desarrollo del sistema se elegido la herramienta Visual Studio 2005 de Microsoft el mismo que contiene el lenguaje de programación C# que es orientado a objetos, la selección del mismo se da debido a que C# usando Visual Studio y sus herramientas proporciona mas libertades que otras herramientas.

Debido a que el costo de las licencias de un administrador de bases de datos son relativamente elevados y la empresa no tiene asignado presupuesto para estos recursos para la adquisición de estas, hemos seleccionado la versión Microsoft Sql Express ya que es una versión gratuita y permite cumplir con los requerimientos de la metodología anteriormente expuesta.

CAPITULO II.

DESARROLLO DEL SISTEMA DE ADMINISTRACION DE NOMINA

2.1 ANALISIS DEL SISTEMA.

Dado que la metodología seleccionada para el desarrollo del proyecto es RUP, se utilizará los requerimientos (flujo de trabajo) en la parte de análisis para una mejor comprensión.

2.1.1 REQUERIMIENTOS

2.2.1.1 Documento Visión [¹⁰]

2.1.1.1.1 Propósito.

El propósito de éste documento es identificar el alcance, los requisitos y restricciones del Sistema de Administración de nómina para una PYME. El documento se centra en la funcionalidad requerida por la empresa Beehunting en donde se va a implementar el sistema.

Las funciones principales son la generación de roles de pagos y el cálculo de todos los procesos de nómina, además de la generación automática de finiquitos.

Los detalles de cómo el sistema cubre estos requerimientos se pueden observar más detalladamente en los casos de uso del sistema.

2.1.1.1.2 Alcance.

El sistema nos permitirá:

- Ingresar los empleados que prestan sus servicios en la empresa.

[10] Documento Visión

- Generar un rol de pagos mensual para todos los empleados de la empresa de una manera rápida, segura y eficaz el cuál reducirá de manera significativa el proceso normal de personal de recursos humanos.
- Generar el respectivo finiquito, calculando todos los valores que por ley le corresponden al empleado.

2.1.1.1.3 Determinación de los requisitos funcionales

El sistema a desarrollarse debe realizar las siguientes funciones:

- Permitir la creación, modificación y eliminación de usuarios del sistema.
- Permitir la creación, modificación y eliminación de datos personales del empleado.
- Permitir la creación, modificación y eliminación de datos laborales del empleado (fecha de ingreso, fecha de salida, datos IESS).
- Permitir la generación de rol de pagos mensual para los empleados.
- Permitir guardar la información ingresada para mantener datos históricos para consultas y respaldos de la empresa.
- Permitir consultar e imprimir los roles de pagos de un empleado.
- Permitir generar, consultar e imprimir el finiquito de un empleado que sale de la empresa.

2.1.1.1.4 Restricciones.

- El sistema no emitirá ningún formato o documento de contrato de trabajo o algún proceso específico para este tema.
- El sistema no realizará un control de horarios de entrada y salida de los empleados.
- Los roles de pago se generarán al final del mes.
- El sistema no generará archivos de texto para enviar a los diferentes bancos para la acreditación de la remuneración del empleado.
- Cualquier funcionalidad que no se encuentre especificada en este documento no será implementada.

2.1.1.1.5 Descripción global del producto.

El producto a desarrollar es un sistema dirigido para las empresas que necesitan optimizar el tiempo de administración de nómina facilitando el trabajo al personal de recursos humanos y posibles errores que puedan darse en el proceso de generación de roles de pagos, además de guardar información histórica y confiable.

El sistema contará con interfaces amigables para el usuario y de fácil uso, con el objetivo de que haya mejor interacción entre el sistema y el usuario, por lo cual se realizará pruebas con varios usuarios; además se realizará la respectiva implementación del sistema en un caso de estudio.

2.1.1.1.6 Terminología.

Los siguientes términos se relacionan con el problema a tratar:

Rol de Pagos: Es un documento contable, el cual permite conocer los ingresos, egresos, saldos y total líquido a recibir por el empleado, los porcentajes y valores han sido fijados por el Ministerio de Trabajo del Gobierno Ecuatoriano . Consta de 2 partes:

Ingresos:

- Sueldo Nominal
- Días
- Sueldo ganado (sueldo en contrato por el número de días)
- Horas Extras
 - Suplementarias: después de 8 horas normales de trabajo.
 - Extraordinarias: sábados, domingos, días cívicos y festivos.
 - Nocturnas/Especiales: en horario nocturno.
- Comisiones
- Otros
 - Bonos
 - Premios

- Gratificación
- Rol de Provisiones:
 - Décimo tercer
 - Se paga el 24 de diciembre a las 12 p.m.
 - Cálculo: Total ganado en el años/ 12
 - Desde el 1º de diciembre al 30 de noviembre
 - No grava IESS, pero si grava Impuesto a la Renta
 - Décimo cuarto:
 - Salario mínimo vital (\$150,00)
 - Del 1 de septiembre hasta el 31 de agosto, tanto régimen Costa-Sierra
 - No grava IESS, pero si grava Impuesto a la Renta.
 - Vacaciones:
 - Por ley el empleado tiene derecho a 15 días siempre y cuando tenga más de 1 año de trabajo en la empresa.
 - Son pagadas
 - Cálculo : Total ganado/24, (la mitad de décimo tercero)
 - Puede acumular tres años y cuando tenga más de un año TIENE QUE SALIR, sino no goza, pierde TODAS LAS VACACIONES
 - Los días adicionales puede negociar
 - Fondos de reserva:
 - Aporte que se deposita en el IESS no se le da al empleado, pasado el 1º año,
 - Es la doceava parte
 - Para su devolución, pasado los tres años puede retirar.

Egresos:

- Aporte IESS: Aporte que el empleador tiene que pagar al IESS como aporte personal del trabajador y corresponde al 9,35% del total de sus ingresos mensuales.
- Impuesto a la Renta : Es el impuesto que se debe cancelar sobre los ingresos o rentas, producto de actividades personales, comerciales, industriales, agrícolas, y en general actividades económicas y aún sobre

ingresos gratuitos, percibidos durante un año, luego de descontar los costos y gastos incurridos para obtener o conservar dichas rentas.

Liquidación ó Finiquito: Es un documento por el cual quedan canceladas todos los contactos entre el empleado y las empresas, tanto laborales como económicas.

Existen dos formas de liquidación:

- Por Renuncia: Se paga un rubro llamado desahucio, que corresponde el 5% del sueldo multiplicado por el número de años de servicio.
- Por Despido: Se paga un rubro llamado por despido intempestivo, que corresponde el 25% del sueldo multiplicado por el número de años de servicio.

2.1.2 MODELO DEL NEGOCIO

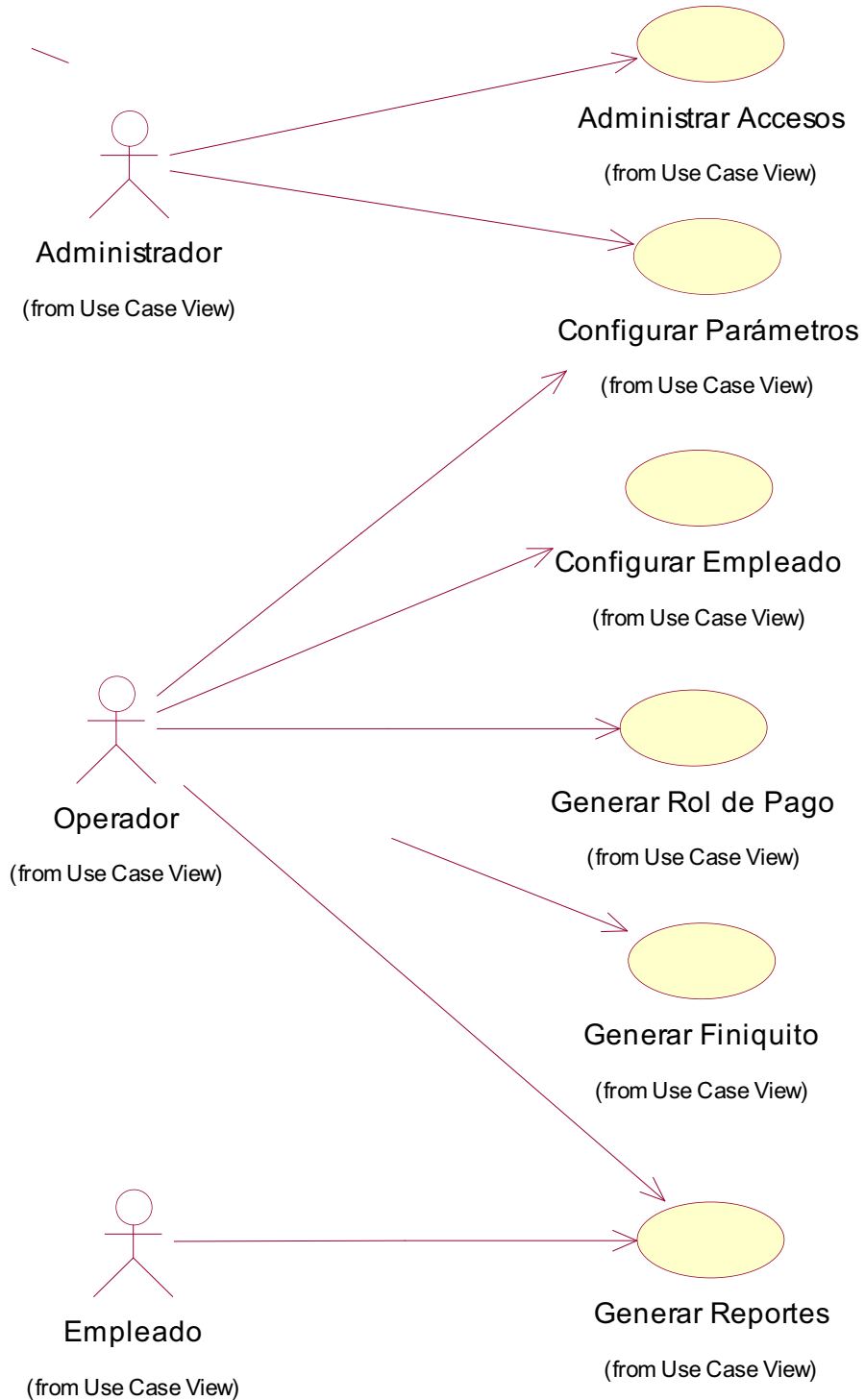


Figura 2 - 1: Modelo del Negocio

2.2.1.1 Diccionario de Actores

A continuación se detalla los actores involucrados en el sistema:

ACTOR	DESCRIPCION
Administrador	Persona que se encarga de Administrar el Sistema y hacer el mantenimiento de tablas, según los cambios que se dan; por tanto no tiene restricciones en el sistema.
Operador	Persona que genera los trámites del empleado como son: rol de pagos, rol de provisiones, ingreso de datos de un nuevo empleado, finiquitos.
Empleado	Es la persona que usa el sistema únicamente para Generar Reportes de su rol de pagos correspondiente actual o roles de pagos anteriores.

Tabla 2 - 1: Diccionario de Actores

2.2.1.2 Catálogo de Casos de Uso del Negocio

CASOS DE USO	ACTOR	DESCRIPCION
Administrar Accesos	Administrador	Permite crear, eliminar y modificar los usuarios que utilizaran el sistema, permisos de acceso de cada usuario, y mantenimiento del mismo.
Configurar parámetros	Operador	Permite crear, eliminar, modificar los parámetros que se utilizarán en el sistema para la generación del rol de pagos.
Configurar Empleado	Operador	Permite crear, eliminar, modificar los empleados del sistema con todas sus características.
Generar Rol de Pago	Operador	Permite la generación del rol de pagos con todos sus beneficios correspondientes.

Generar Finiquito	Operador	Permite generar automáticamente el finiquito correspondiente al empleado.
Generar Reportes	Empleado	Permite generar reportes de sus roles de pagos.

Tabla 2 - 2: Catálogos de casos de uso del negocio

2.1.3 MODELO DE CASOS DE USO DEL SISTEMA [11]

2.2.1.1 Diagrama Completo del Sistema

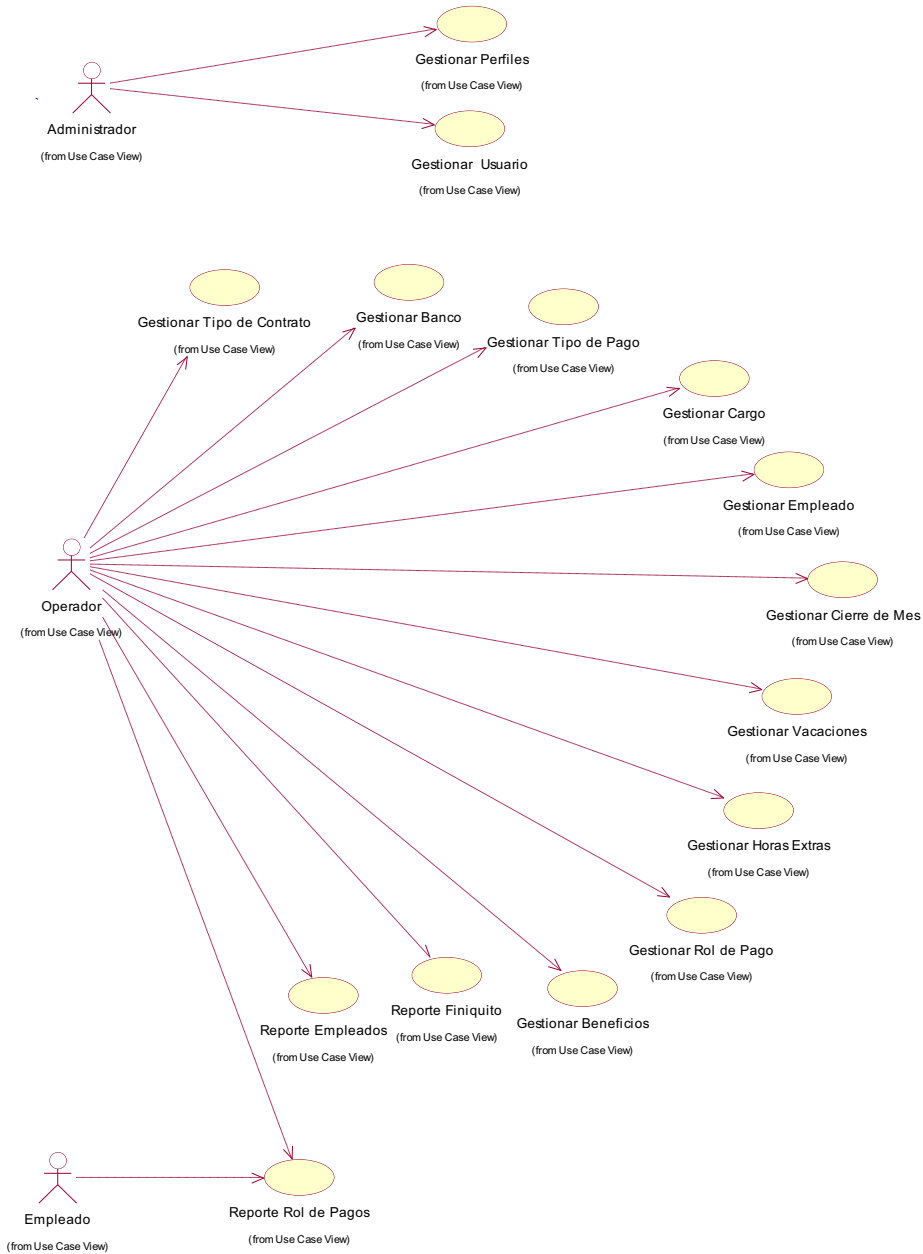


Figura 2 - 2: Diagrama Completo del Sistema

2.2.1.2 Módulo Administrar Accesos

2.1.3.2.1 CU:01 Gestionar Perfiles.

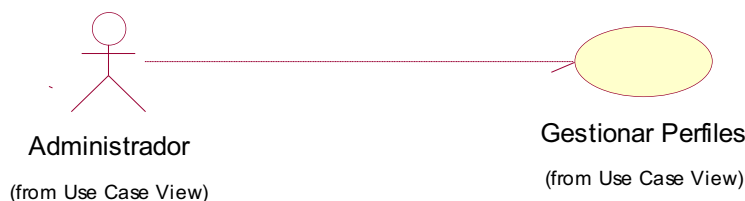


Figura 2 - 3: CU:01 Gestionar Perfiles

Gestionar Perfiles		
Descripción general: Nos permite crear, eliminar, modificar los perfiles que existen en el sistema y administrar el acceso que tiene cada usuario del sistema.		
Actor: Administrador		
Pre-Condiciones: Tener el acceso a crear perfiles.		
Pos-Condiciones: Los perfiles quedan registrados y podrán ser modificados por usuarios que tengan a acceso a este menú.		
Flujo: Nuevo Perfil		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista de perfiles.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos regresa a la lista de perfiles.

2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a la lista de perfiles.
Flujo: Eliminar Perfil		
Nº	ACTOR	SISTEMA
1	Selecciona el perfil y elige la opción eliminar.	
2		Presenta la información del perfil.
3		Despliega una pantalla de confirmación del proceso de eliminación.
4	Elige la opción Aceptar.	
5		Borra el registro correspondiente y regresa a la lista de perfiles.
Flujo Alternativo:		
4	Elige la opción Cancelar.	
5		No borra el registro.
Flujo: Modificar Perfil		
Nº	ACTOR	SISTEMA
1	Selecciona el perfil.	
2		Presenta la información del perfil.
3	Llena la información a modificar en los campos necesarios.	Valida los datos ingresados y si existe un error despliega un mensaje.
4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de perfiles.

Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos.
4	Elige la opción Cancelar.	
5		No modifica los datos del perfil y regresa a la lista de perfiles.
Flujo: Asignar Menús al Perfil		
Nº	ACTOR	SISTEMA
1	Selecciona el perfil	
2		Presenta la información de los menús.
3	Elige la opción Agregar.	
4		Agrega menús al perfil.
Flujo Alternativo:		
4		Valida si los datos están duplicados.
3	Elige la opción Eliminar.	
4		Elimina menús al perfil.

Tabla 2 - 3: CU:01 Gestionar Perfiles

2.1.3.2.2 CU:02 Gestionar Usuarios.

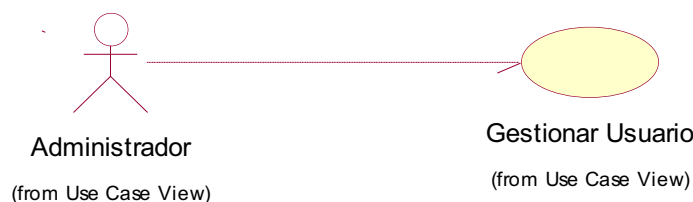


Figura 2 - 4: CU:02 Gestionar Usuarios

Gestionar Usuarios

Descripción general: Nos permite crear, modificar y cambiar contraseña de los usuarios del sistema y sus permisos de acceso.		
Actor: Administrador		
Pre-Condiciones: Tener privilegios para gestionar usuarios y tener al menos un perfil creado.		
Pos-Condiciones: Los usuarios quedan registrados y estos tendrán acceso al sistema según su perfil asignado.		
Flujo: Nuevo Usuario		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos y asigna un perfil al nuevo usuario.	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista de usuarios.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos regresa a la lista de usuarios.
2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a lista de usuarios.
Flujo: Modificar Usuario		
Nº	ACTOR	SISTEMA
1	Selecciona el usuario	
2		Presenta la información del usuario.
3	Llena la información a modificar en los campos necesarios.	Valida los datos ingresados y si existe un error despliega un mensaje.

4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de usuarios.
Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos regresa a la lista de usuarios.
4	Elige la opción Cancelar.	
5		No modifica los datos del usuario y regresa a la lista de usuarios.
Flujo: Cambiar Contraseña		
Nº	ACTOR	SISTEMA
1	Selecciona el usuario y cambia la contraseña	
2		Presenta la pantalla de cambio de contraseña.
3	Elige la opción Aceptar.	
4		La contraseña se modifica y regresa a la lista de usuarios.
Flujo Alternativo:		
4		Valida si los datos están duplicados.
3	Elige la opción Cancelar.	
4		No modifica la contraseña del usuario y regresa a la lista de usuarios.

Tabla 2 - 4: CU:02 Gestionar Usuarios

2.2.1.3 Módulo Configurar Parámetros

2.1.3.3.1 CU:03 Gestionar Tipo de Contrato.



Figura 2 - 5: CU:03 Gestionar Tipo de Contrato

Gestionar Tipo de Contrato		
Descripción general: Nos permite crear, eliminar, modificar los tipos de contratos que existen en el sistema.		
Actor: Operador		
Pre-Condiciones: Tener privilegios para gestionar tipo de contrato.		
Pos-Condiciones: Los tipos de contrato quedan registrados.		
Flujo: Nuevo Tipo de Contrato		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos.	Valida los datos ingresados.
2	Elige la opción Guardar.	
3		Almacena los datos.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos regresa a la lista de tipos de contratos.
2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a la lista de tipos de contratos.
Flujo: Eliminar Tipo de Contrato		

N°	ACTOR	SISTEMA
1	Selecciona el tipo de contrato y elige la opción eliminar.	
2		Presenta la información del tipo de contrato.
3		Despliega una pantalla de confirmación del proceso de eliminación.
4	Elige la opción Aceptar.	
5		Borra el registro correspondiente y regresa a la lista de tipos de contratos.
Flujo Alternativo:		
4	Elige la opción Cancelar.	
5		No borra el registro y regresa a la lista de tipos de contratos.
Flujo: Modificar Tipo de Contrato		
N°	ACTOR	SISTEMA
1	Selecciona el tipo de contrato.	
2		Presenta la información del tipo de contrato.
3	Llena la información a modificar en los campos necesarios.	Valida los datos ingresados y si existe un error despliega un mensaje.
4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de tipos de contratos.
Flujo Alternativo:		
5		Despliega un mensaje de error, si los

		datos están duplicados o erróneos.
4	Elige la opción Cancelar.	
5		No modifica los datos del tipo de contrato y regresa a la lista de tipos de contratos.

Tabla 2 - 5: CU:03 Gestionar Tipo de Contrato

2.1.3.3.2 CU:04 Gestionar Banco.



Figura 2 - 6: CU:04 Gestionar Banco

Gestionar Banco		
Descripción general: Nos permite crear, eliminar, modificar los bancos que existen en el sistema.		
Actor: Operador		
Pre-Condiciones: Tener privilegios para gestionar banco.		
Pos-Condiciones: Los bancos quedan registrados y se podrán utilizar en la creación del empleado.		
Flujo: Nuevo Banco		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos.	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista

		de bancos.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos.
2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a la lista de bancos.
Flujo: Eliminar Banco		
Nº	ACTOR	SISTEMA
1	Selecciona el banco y elige la opción eliminar.	
2		Presenta la información del banco.
3		Despliega una pantalla de confirmación del proceso de eliminación.
4	Elige la opción Aceptar.	
5		Borra el registro correspondiente y regresa a la lista de bancos.
Flujo Alternativo:		
4	Elige la opción Cancelar.	
5		No borra el registro y regresa a la lista de bancos.
Flujo: Modificar Banco		
Nº	ACTOR	SISTEMA
1	Selecciona el banco.	
2		Presenta la información del banco.
3	Llena la información a modificar en	Valida los datos ingresados y si existe

	los campos necesarios.	un error despliega un mensaje.
4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de bancos.
Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos.
4	Elige la opción Cancelar.	
5		No modifica los datos del banco y regresa a la lista de bancos.

Tabla 2 - 6: CU:04 Gestionar Banco

2.1.3.3.3 CU:05 Gestionar Tipo de Pago.

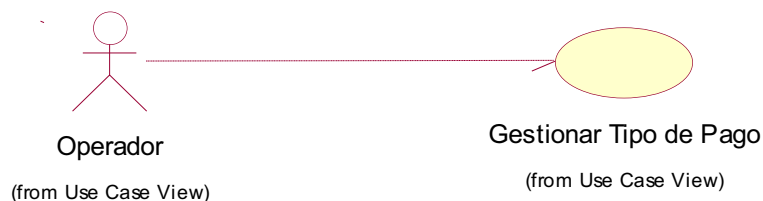


Figura 2 - 7: CU:05 Gestionar Tipo de Pago

Gestionar Tipo de Pago
Descripción general: Nos permite crear, eliminar, modificar los tipos de pago o rubros que tiene un rol de pagos y administrar los parámetros o fórmulas que debe contener el tipo de pago.
Actor: Operador
Pre-Condiciones: Tener privilegios para gestionar tipo de pago.
Pos-Condiciones: Los tipos de pago quedan registrados y se podrán utilizar en la generación del rol de pagos.

Flujo: Nuevo Tipo de Pago		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos.	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista de tipos de pago.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos.
2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a la lista de tipos de pago.
Flujo: Eliminar Tipo de Pago		
Nº	ACTOR	SISTEMA
1	Selecciona el tipo de pago y elige la opción eliminar.	
2		Presenta la información del tipo de pago.
3		Despliega una pantalla de confirmación del proceso de eliminación.
4	Elige la opción Aceptar.	
5		Borra el registro correspondiente y regresa a la lista de tipos de pago.
Flujo Alternativo:		
4	Elige la opción Cancelar.	

5		No borra el registro y regresa a la lista de tipos de pago.
Flujo: Modificar Tipo de Pago		
N°	ACTOR	SISTEMA
1	Selecciona el tipo de pago.	
2		Presenta la información del tipo de pago.
3	Llena la información a modificar en los campos necesarios.	Valida los datos ingresados y si existe un error despliega un mensaje.
4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de tipos de pago.
Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos regresa a la lista de tipos de pago.
4	Elige la opción Cancelar.	
5		No modifica los datos del tipo de pago y regresa a la lista de tipos de pago.
Flujo: Modificar Parámetros		
N°	ACTOR	SISTEMA
1	Selecciona el tipo de pago y escogemos los parámetros.	
2		Presenta la información de los parámetros.
4	Elige la opción Agregar.	
5		Agrega los parámetros al tipo de pago.

Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos.
4	Elige la opción Quitar.	
5		Quita los parámetros al tipo de pago.

Tabla 2 - 7: CU:05 Gestionar Tipo de Pago

2.1.3.3.4 CU:06 Gestionar Cargo.



Figura 2 - 8: CU:06 Gestionar Cargo

Gestionar Cargo		
Descripción general: Nos permite crear, eliminar, modificar los cargos que existen en el sistema.		
Actor: Operador		
Pre-Condiciones: Tener privilegios para gestionar cargo.		
Pos-Condiciones: Los cargos quedan registrados y se podrán utilizar en la creación del empleado.		
Flujo: Nuevo Cargo		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos.	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista

		de cargos.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos.
2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a la lista de cargos.
Flujo: Eliminar Cargo		
Nº	ACTOR	SISTEMA
1	Selecciona el cargo y elige la opción eliminar.	
2		Presenta la información del cargo.
3		Despliega una pantalla de confirmación del proceso de eliminación.
4	Elige la opción Aceptar.	
5		Borra el registro correspondiente y regresa a la lista de cargos.
Flujo Alternativo:		
4	Elige la opción Cancelar.	
5		No borra el registro y regresa a la lista de cargos.
Flujo: Modificar Cargo		
Nº	ACTOR	SISTEMA
1	Selecciona el cargo.	
2		Presenta la información del cargo.
3	Llena la información a modificar en	Valida los datos ingresados y si existe

	los campos necesarios.	un error despliega un mensaje.
4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de cargos.
Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos regresa a la lista de cargos.
4	Elige la opción Cancelar.	
5		No modifica los datos del cargo y regresa a la lista de cargos.

Tabla 2 - 8: CU:06 Gestionar Cargo

2.2.1.4 Módulo Gestionar Empleado

2.1.3.4.1 CU:07 Gestionar Empleado.



Figura 2 - 9: CU:07 Gestionar Empleado

Gestionar Empleado
Descripción general: Nos permite crear, modificar, cambiar contraseña a los empleados de la empresa, además de sus cargas familiares.
Actor: Operador
Pre-Condiciones: Tener creado los parámetros necesarios para gestionar un empleado, tener creado perfiles en el sistema.

Pos-Condiciones: Los empleados quedan registrados y no podrán ser eliminados del sistema.		
Flujo: Nuevo Empleado		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos.	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista de empleados.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos.
2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a la lista de empleados.
Flujo: Modificar Empleado		
Nº	ACTOR	SISTEMA
1	Selecciona el empleado.	
2		Presenta la información del empleado.
3	Llena la información a modificar en los campos necesarios.	Valida los datos ingresados y si existe un error despliega un mensaje.
4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de empleados.
Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos

		regresa a la lista de empleados.
4	Elige la opción Cancelar.	
5		No modifica los datos del empleado y regresa a la lista de empleados.
Flujo: Cambiar Contraseña		
N°	ACTOR	SISTEMA
1	Selecciona el empleado y cambia la contraseña	
2		Presenta la pantalla de cambio de contraseña
3	Elige la opción Aceptar.	
4		La contraseña se modifica y regresa a la lista de empleados.
Flujo Alternativo:		
4		Valida si los datos están duplicados.
3	Elige la opción Cancelar.	
4		No modifica la contraseña del empleado y regresa a la lista de empleados.

Tabla 2 - 9: CU:06 CU:07 Gestionar Empleado

2.2.1.5 Módulo Generar Rol de Pago

2.1.3.5.1 CU:08 Gestionar Cierre de Mes.



Figura 2 - 10: CU:08 Gestionar Cierre de Mes

Gestionar Cierre de Mes		
Descripción general: Nos permite abrir un mes y cerrarlo para poder generar un rol de pagos.		
Actor: Operador		
Pre-Condiciones: Tipos de pago o rubros configurados para los roles de pago.		
Pos-Condiciones: Los cierre de mes quedan registrados y no podrán ser eliminados.		
Flujo: Nuevo Cierre de Mes		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos.	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista de cierres de mes.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos.
2	Elige la opción Cancelar.	
3		No guarda los datos y regresa a la lista de cierres de mes.
Flujo: Modificar Cierre de Mes		

Nº	ACTOR	SISTEMA
1	Selecciona el cierre de mes.	
2		Presenta la información del cierre de mes.
3	Llena la información a modificar en los campos necesarios.	Valida los datos ingresados y si existe un error despliega un mensaje.
4	Elige la opción Actualizar.	
5		Almacena los datos modificados y regresa a la lista de cierres de mes.
Flujo Alternativo:		
5		Despliega un mensaje de error, si los datos están duplicados o erróneos regresa a la lista de cierres de mes.
4	Elige la opción Cancelar.	
5		No modifica los datos y regresa a la lista de cierres de mes.
Flujo: Agregar Rubros		
Nº	ACTOR	SISTEMA
1	Selecciona el cierre de mes.	
2		Presenta la información del cierre de mes.
3	Agregar rubros o tipos de pago.	
4	Elige la opción Guardar.	
5		Almacena los datos modificados y regresa a la lista de cierres de mes.
Flujo Alternativo:		
3	Quitar rubros o tipos de pago.	

5		No almacena los rubros y regresa a la lista de cierres de mes.
---	--	--

Tabla 2 - 10: CU:08 Gestionar Cierre de Mes

2.1.3.5.2 CU: 09 Gestionar Vacaciones.



Figura 2 - 11: CU:09 Gestionar Vacaciones

Gestionar Vacaciones		
Descripción general: Nos permite crear las vacaciones que toma un empleado, y calcularlo en la generación del rol de pagos.		
Actor: Operador		
Pre-Condiciones: Empleado registrado, datos de contrato registrado.		
Pos-Condiciones: Las vacaciones quedan registradas y no se podrán modificar ni eliminar los registros.		
Flujo: Nueva Vacación		
Nº	ACTOR	SISTEMA
1	Escoger empleado y llenar los campos requeridos.	Valida los datos ingresados y si existe un error despliega un mensaje.
2	Elige la opción Guardar.	
3		Almacena los datos y regresa a la lista de vacaciones del empleado.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los

		datos están duplicados o erróneos y regresa a la lista de vacaciones del empleado.
2	Se cancela el ingreso de vacaciones, saliendo del formulario sin hacer clic en el botón Guardar.	
3		No guarda los datos y regresa a la lista de vacaciones del empleado.

Tabla 2 - 11: CU:09 Gestionar Vacaciones

2.1.3.5.3 CU:10 Gestionar Horas Extras.



Figura 2 - 12: CU:10 Gestionar Horas Extras

Gestionar Horas Extras		
Descripción general: Nos permite crear y modificar las horas extras de un empleado, y calcularlo en la generación del rol de pagos.		
Actor: Operador		
Pre-Condiciones: Empleado registrado.		
Pos-Condiciones: Las horas extras quedan registradas y no se podrán eliminar los registros.		
Flujo: Nueva Hora Extra		
Nº	ACTOR	SISTEMA
1	Escoger empleado y llenar los campos requeridos.	Valida los datos ingresados y si existe un error despliega un mensaje.

2	Elige la opción Guardar.	
3		Almacena los datos, calculando si es hora extra u hora suplementaria y regresa a la lista de horas extras del empleado.
Flujo Alternativo:		
3		Despliega un mensaje de error, si los datos están duplicados o erróneos y regresa a la lista de horas extras del empleado.
2	Se cancela el ingreso de horas extras, saliendo del formulario sin hacer clic en el botón Guardar.	
3		No guarda los datos y regresa a la lista de horas extras del empleado.
Flujo: Eliminar Hora Extra		
Nº	ACTOR	SISTEMA
1	Selecciona la hora extra y elige la opción eliminar.	
2		Presenta la información de la hora extra.
3		Despliega una pantalla de confirmación del proceso de eliminación.
4	Elige la opción Aceptar.	
5		Borra el registro correspondiente y regresa a la lista de horas extras del empleado.

Tabla 2 - 12: CU:10 Gestionar Horas Extras

2.1.3.5.4 CU:11 Gestionar Rol de Pago.

Figura 2 - 13: CU:11 Gestionar Rol de Pago

Gestionar Rol de Pago		
Descripción general: Nos permite generar el rol de pagos para todos los empleados de la empresa.		
Actor: Operador		
Pre-Condiciones: Tener privilegios para gestionar rol de pago, tener abierto un cierre de mes del cual se va a generar el rol.		
Pos-Condiciones: Los roles de pago quedan registrados, no podrán ser eliminados y solo podrán ser modificados hasta el cierre del mismo.		
Flujo: Generar Rol de Pagos		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos escogiendo el año y el mes a generarse.	
2	Elige la opción Generar.	
3		Almacena los datos.
Flujo Alternativo:		
3		Despliega un mensaje de error, si no existe un cierre de mes abierto.
2	Se cancela la generación del rol,	

	saliendo del formulario sin hacer clic en el botón Generar.	
3		Regresa a la pantalla inicial de rol de pagos.

Tabla 2 - 13: CU:11 Gestionar Rol de Pago

2.1.3.5.5 CU:12 Gestionar Beneficios.



Figura 2 - 14: CU:12 Gestionar Beneficios

Gestionar Beneficios		
Descripción general: Nos permite generar los beneficios de ley (décimo tercer y cuarto sueldo, utilidades) para todos los empleados de la empresa.		
Actor: Operador		
Pre-Condiciones: Tener datos de los empleados, tener el tipo de pago o rubro a generarse, tener abierto un cierre de mes del cual se va a generar los beneficios.		
Pos-Condiciones: Los beneficios quedan registrados, no podrán ser eliminados y solo podrán ser modificados hasta el cierre del mismo.		
Flujo: Generar Beneficios		
Nº	ACTOR	SISTEMA
1	Escoge el beneficio a generarse.	
2	Elige la opción Generar.	
3		Almacena los datos.
Flujo Alternativo:		

3		Despliega un mensaje de error, si no existe un cierre de mes abierto.
2	Se cancela la generación de beneficios, saliendo del formulario sin hacer clic en el botón Generar.	
3		Regresa a la pantalla inicial de beneficios.

Tabla 2 - 14: CU:12 Gestionar Beneficios

2.2.1.6 Módulo Generar Finiquito

2.1.3.6.1 CU:13 Generar Finiquito.



Figura 2 - 15: CU:13 Generar Finiquito

Generar Finiquito		
Descripción general: Nos permite generar automáticamente el finiquito correspondiente a un empleado.		
Actor: Operador		
Pre-Condiciones: Tener datos de los empleados con fecha de salida, tener roles de pagos generados.		
Pos-Condiciones: El finiquito queda registrado y no podrá ser eliminado.		
Flujo: Generar Finiquito		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos.	

2	Elige la opción Generar.	
3		Almacena los datos.
4		Genera reporte de finiquito
Flujo Alternativo:		
2	Se cancela la generación de finiquito, saliendo del formulario sin hacer clic en el botón Generar.	
3		Regresa a la pantalla inicial de generar finiquitos.
Flujo: Modificar Finiquito		
Nº	ACTOR	SISTEMA
1	Llenar los campos requeridos y modifica el finiquito.	
2	Elige la opción Guardar.	
3		Almacena los datos.
4		Genera reporte de finiquito
Flujo Alternativo:		
2	Se cancela la generación de finiquito, saliendo del formulario sin hacer clic en el botón Generar.	
3		Regresa a la pantalla inicial de generar finiquitos..

Tabla 2 - 15: CU:13 Gestionar Finiquito

2.2.1.7 Módulo Generar Reportes

2.1.3.7.1 CU:14 Generar Reportes.

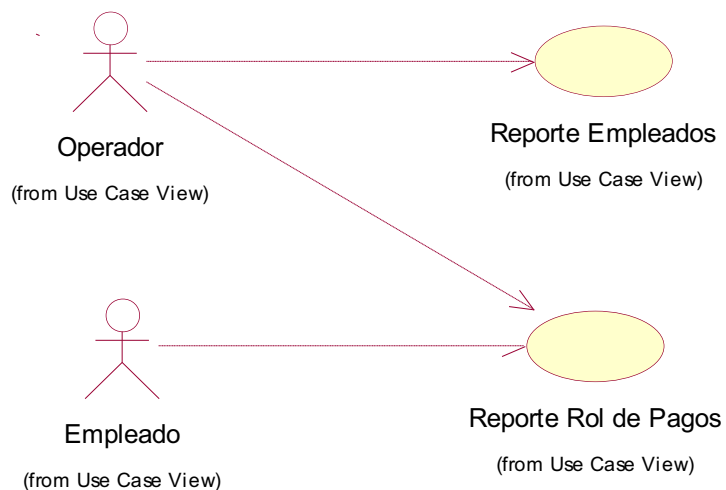


Figura 2 - 16: CU:14 Generar Reportes

Generar Reportes		
Descripción general: Permite al empleado y operador visualizar sus reportes de rol de pagos en el momento que lo requiera e imprimirlo para cualquier uso, y también sacar un reporte de empleados.		
Actor: Empleado/Operador		
Pre-Condiciones: Tener datos de los empleados, tener roles de pagos generados.		
Pos-Condiciones: Se puede imprimir o visualizar el reporte las veces que se lo requiera.		
Flujo: Generar Reporte Rol de Pago		
Nº	ACTOR	SISTEMA
1	Escoge el empleado o empleados a generar el reporte.	
2	Elige la opción Generar	Despliega el reporte de acuerdo a la opción escogida, con los campos de información necesaria.
3	Elige la opción Imprimir.	

4		Muestra en pantalla una vista previa del reporte.
5	Imprime el reporte	Envía a imprimir el reporte.
Flujo: Generar Reporte Empleado		
N°	ACTOR	SISTEMA
1	Escoge los filtros y condiciones de búsqueda.	
2	Elige la opción Buscar	Despliega el reporte de acuerdo a los filtros de búsqueda, con los campos de información necesaria.
3	Elige la opción Imprimir.	
4		Muestra en pantalla una vista previa del reporte.
5	Imprime el reporte	Envía a imprimir el reporte.

Tabla 2 - 16: CU:14 Generar Reportes

2.1.4 DIAGRAMAS DE COLABORACION

2.2.1.1 Módulo Administrar Accesos

2.1.4.1.1 CU:01 Gestionar Perfiles.

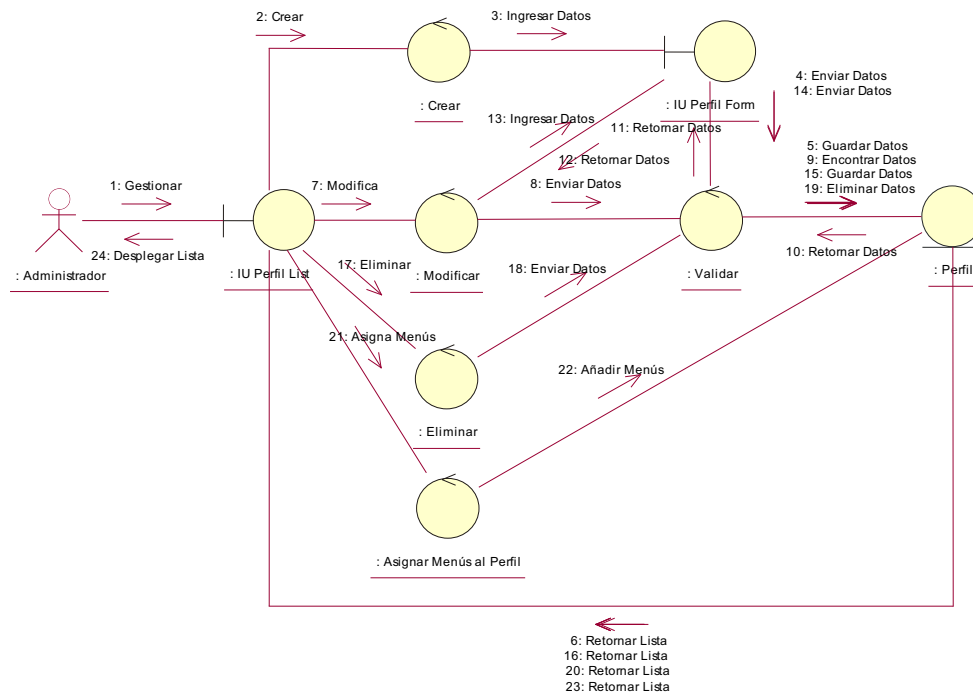


Figura 2 - 17: Diagrama Colaboración:01 Gestionar Perfiles

Flujo de Sucesos

El usuario Administrador solicita al objeto IUPerfilList y este a su vez al objeto Crear el formulario de ingreso de un nuevo perfil(1,2). El usuario Administrador por medio del objeto IUPerfilForm ingresa los datos del nuevo perfil (3), IUPerfilForm solicita al objeto Validar que añada la información del nuevo perfil (4), en caso de ser válida la información, el objeto Validar solicita al objeto Perfil guarde la información (5). El objeto Perfil retorna la lista de perfiles al objeto IUPerfilList mostrando la lista de perfiles (6). El usuario Administrador solicita al objeto IUPerfilList y este a su vez al objeto Modificar el formulario para modificar un perfil (7). El objeto Validar solicita al objeto Perfil encuentre el registro solicitado por medio de su código, el

objeto Perfil retorna el registro solicitado al objeto Validar y este a su vez al objeto IUPerfilForm y este a su vez la despliega al usuario Administrador (8 ,9 ,10 ,11 ,12). El usuario Administrador por medio del objeto IUPerfilForm ingresa los datos del perfil (13), IUPerfilForm solicita al objeto Validar que añada la información del perfil (14), en caso de ser válida la información, el objeto Validar solicita al objeto Perfil guarde la información (15). El objeto Perfil retorna la lista de perfiles al objeto IUPerfilList mostrando la lista de perfiles (16). El usuario Administrador solicita al objeto IUPerfilList y este al objeto Eliminar para eliminar los datos de un perfil (17). El objeto Validar envía el código del perfil a ser eliminada al objeto Perfil el cual elimina dicho registro (18,19). El objeto Perfil retorna la lista de perfiles al objeto IUPerfilList mostrando la lista de perfiles al usuario Administrador (20). El usuario Administrador solicita al objeto IUPerfilList y este al objeto Asignar Menús al Perfil para asignar un menú al perfil seleccionado (21). El objeto Perfil retorna la lista de perfiles al objeto IUPerfilList mostrando la lista de perfiles al usuario Administrador (22,24).

Tabla 2 - 17: Flujo de Sucesos:01 Gestionar Perfiles

2.1.4.1.2 CU:02 Gestionar Usuarios.

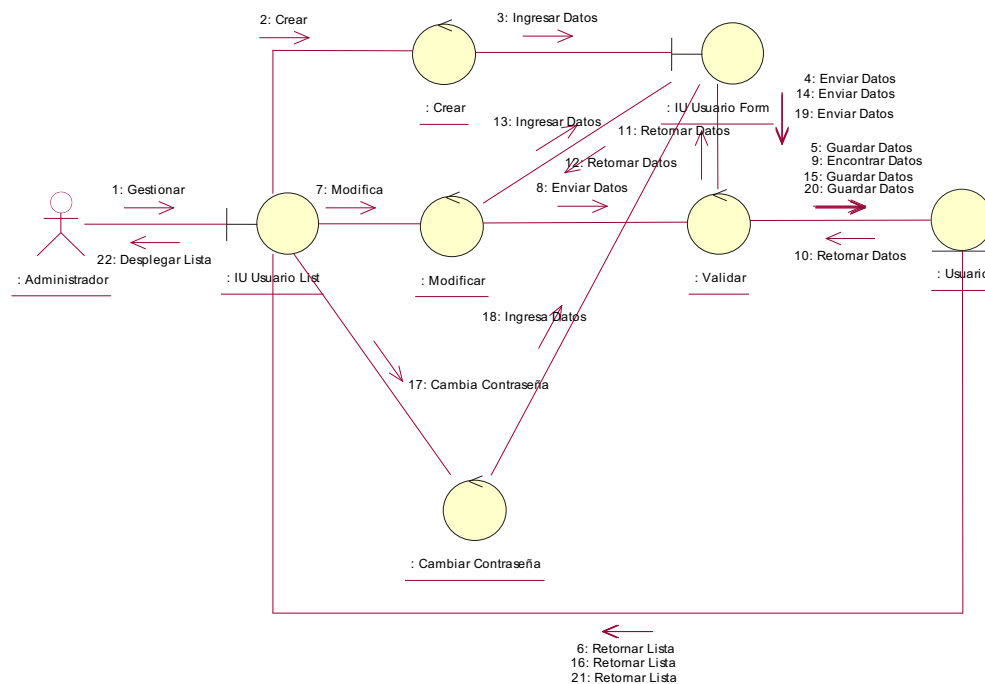


Figura 2 - 18: Diagrama Colaboración:02 Gestionar Usuarios

Flujo de Sucesos

El usuario Administrador solicita al objeto IUUsuarioList y este a su vez al objeto Crear el formulario de ingreso de un nuevo usuario(1,2). El usuario Administrador por medio del objeto IUUsuarioForm ingresa los datos del nuevo usuario (3), IUUsuarioForm solicita al objeto Validar que añada la información del nuevo usuario (4), en caso de ser válida la información, el objeto Validar solicita al objeto Usuario guarde la información (5). El objeto Usuario retorna la lista de usuarios al objeto IUUsuarioList mostrando la lista de usuarios (6). El usuario Administrador solicita al objeto IUUsuarioList y este a su vez al objeto Modificar el formulario para modificar un usuario (7). El objeto Validar solicita al objeto Usuario encuentre el registro solicitado por medio de su código, el objeto Usuario retorna el registro solicitado al objeto Validar y este a su vez al objeto IUUsuarioForm y este a su vez la despliega al usuario Administrador (8 ,9 ,10 ,11 ,12). El usuario Administrador por medio del objeto IUUsuarioForm ingresa los datos del usuario (13), IUUsuarioForm solicita al objeto Validar que añada la

información del usuario (14), en caso de ser válida la información, el objeto Validar solicita al objeto Usuario guarde la información (15). El objeto Usuario retorna la lista de usuarios al objeto IUUsuarioList mostrando la lista de usuarios (16). El usuario Administrador solicita al objeto IUUsuarioList y este al objeto CambiarContraseña para cambiar la contraseña de un usuario (17,18), IUUsuarioForm solicita al objeto Validar que añada la información de la contraseña(19), en caso de ser válida la información, el objeto Validar solicita al objeto Usuario guarde la información (20).El objeto Usuario retorna la lista de usuarios al objeto IUUsuarioList mostrando la lista de usuarios al usuario Administrador (21,22).

Tabla 2 - 18: Flujo de Sucesos:02 Gestionar Usuarios

2.2.1.2 Módulo Configurar Parámetros

2.1.4.2.1 CU:03 Gestionar Tipo de Contrato.

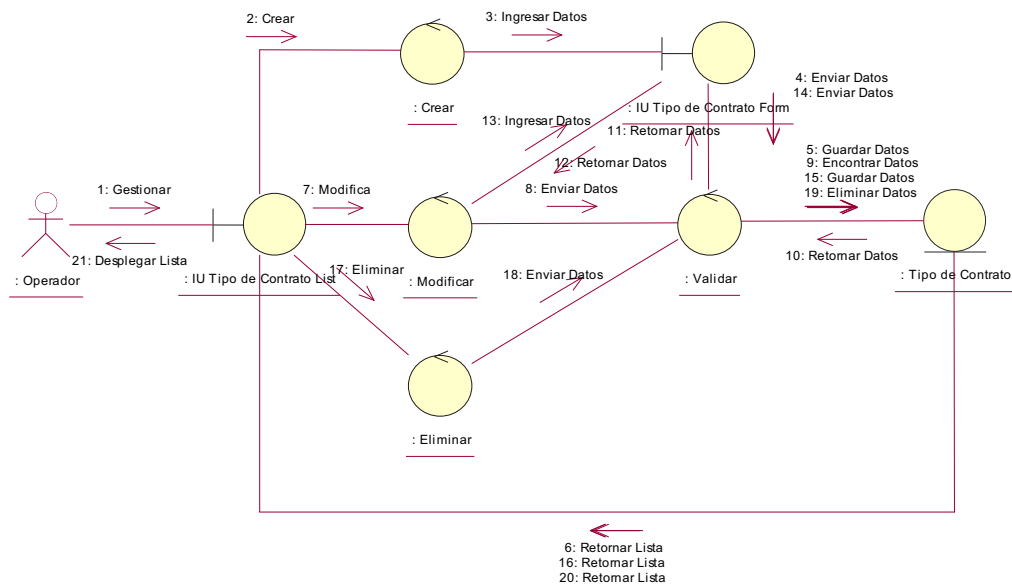


Figura 2 - 19: Diagrama Colaboración:03 Gestionar Tipo de Contrato

Flujo de Sucesos

El usuario Operador solicita al objeto IUTipodeContratoList y este a su vez al objeto Crear el formulario de ingreso de un nuevo tipo de

contrato(1,2). El usuario Operador por medio del objeto IUTipodeContratoForm ingresa los datos del nuevo tipo de contrato (3), IUTipodeContratoForm solicita al objeto Validar que añada la información del nuevo tipo de contrato (4), en caso de ser válida la información, el objeto Validar solicita al objeto TipodeContrato guarde la información (5). El objeto TipodeContrato retorna la lista de tipo de contratos al objeto IUTipodeContratoList mostrando la lista de tipo de contratos (6). El usuario Operador solicita al objeto IUTipodeContratoList y este a su vez al objeto Modificar el formulario para modificar un tipo de contrato (7). El objeto Validar solicita al objeto TipodeContrato encuentre el registro solicitado por medio de su código, el objeto TipodeContrato retorna el registro solicitado al objeto Validar y este a su vez al objeto IUTipodeContratoForm y este a su vez la despliega al usuario Operador (8 ,9 ,10 ,11 ,12). El usuario Operador por medio del objeto IUTipodeContratoForm ingresa los datos del tipo de contrato (13), IUTipodeContratoForm solicita al objeto Validar que añada la información del tipo de contrato (14), en caso de ser válida la información, el objeto Validar solicita al objeto TipodeContrato guarde la información (15). El objeto TipodeContrato retorna la lista de tipo de contratos al objeto IUTipodeContratoList mostrando la lista de tipo de contratos (16). El usuario Operador solicita al objeto IUTipodeContratoList y este al objeto Eliminar para eliminar los datos de un tipo de contrato (17). El objeto Validar envía el código del tipo de contrato a ser eliminada al objeto TipodeContrato el cual elimina dicho registro (18,19). El objeto TipodeContrato retorna la lista de tipo de contratos al objeto IUTipodeContratoList mostrando la lista de tipo de contratos al usuario Operador (20,21).

Tabla 2 - 19: Flujo de Sucesos:03 Gestionar Tipo de Contrato

2.1.4.2.2 CU:04 Gestionar Banco.

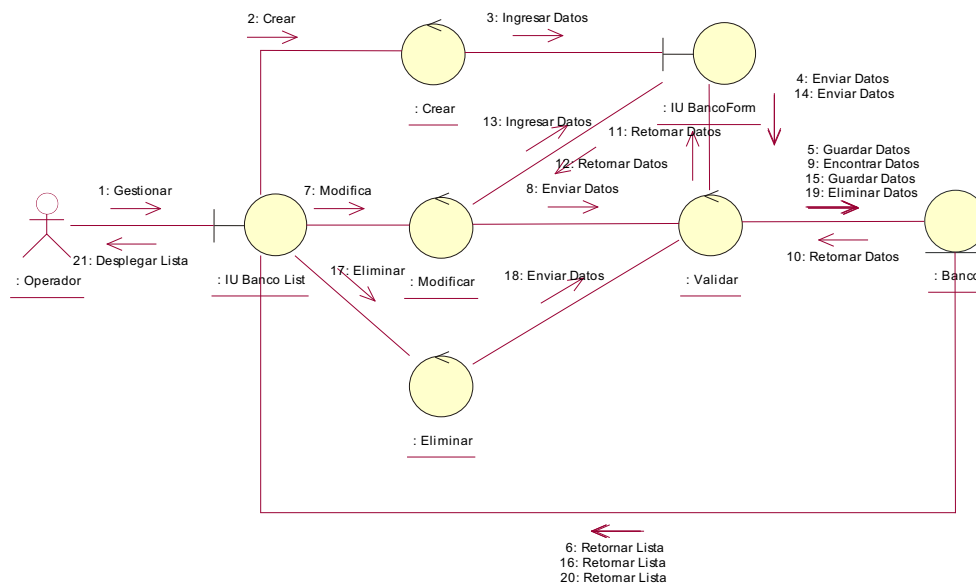


Figura 2 - 20: Diagrama Colaboración:04 Gestionar Banco

Flujo de Sucesos

El usuario Operador solicita al objeto IUBancoList y este a su vez al objeto Crear el formulario de ingreso de un nuevo banco(1,2). El usuario Operador por medio del objeto IUBancoForm ingresa los datos del nuevo banco (3), IUBancoForm solicita al objeto Validar que añada la información del nuevo banco (4), en caso de ser válida la información, el objeto Validar solicita al objeto Banco guarde la información (5). El objeto Banco retorna la lista de bancos al objeto IUBancoList mostrando la lista de bancos (6). El usuario Operador solicita al objeto IUBancoList y este a su vez al objeto Modificar el formulario para modificar un banco (7). El objeto Validar solicita al objeto Banco encuentre el registro solicitado por medio de su código, el objeto Banco retorna el registro solicitado al objeto Validar y este a su vez al objeto IUBancoForm y este a su vez la despliega al usuario Operador (8 ,9 ,10 ,11 ,12). El usuario Operador por medio del objeto IUBancoForm ingresa los datos del banco (13), IUBancoForm solicita al objeto Validar que añada la información del banco (14), en caso de ser válida la información, el objeto Validar solicita al objeto Banco guarde la información (15). El objeto Banco retorna la lista de bancos

al objeto IUBancoList mostrando la lista de bancos (16). El usuario Operador solicita al objeto IUBancoList y este al objeto Eliminar para eliminar los datos de un banco (17). El objeto Validar envía el código del banco a ser eliminada al objeto Banco el cual elimina dicho registro (18,19). El objeto Banco retorna la lista de bancos al objeto IUBancoList mostrando la lista de bancos al usuario Operador (20,21).

Tabla 2 - 20: Flujo de Sucesos:04 Gestionar Banco

2.1.4.2.3 CU:05 Gestionar Tipo de Pago.

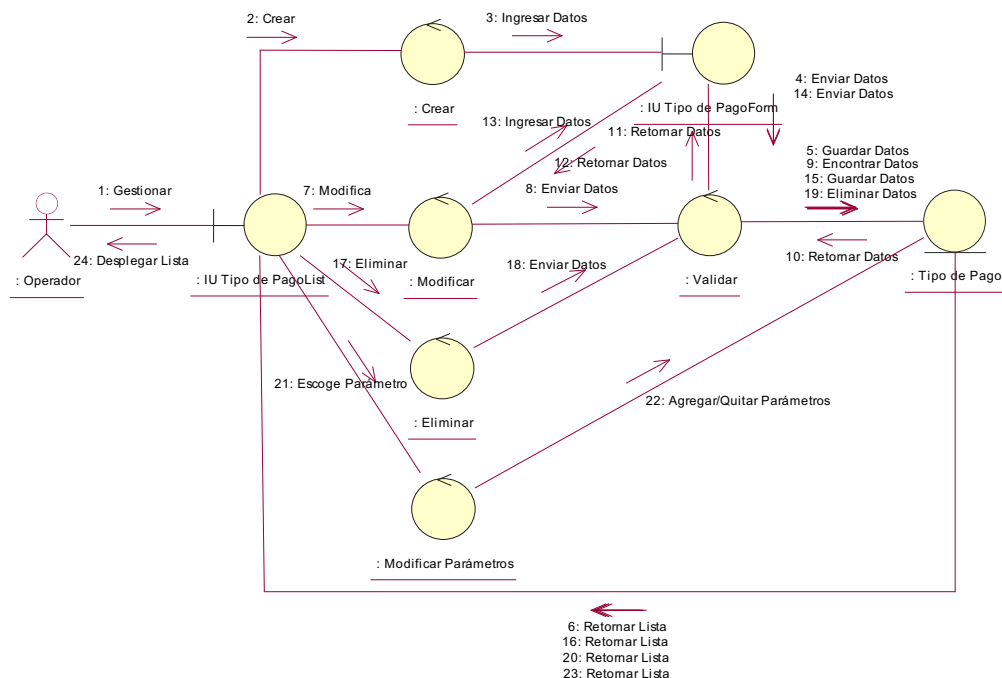


Figura 2 - 21: Diagrama Colaboración:05 Gestionar Tipo de Pago

Flujo de Sucesos

El usuario Operador solicita al objeto IUTipodePagoList y este a su vez al objeto Crear el formulario de ingreso de un nuevo tipo de pago(1,2). El usuario Operador por medio del objeto IUTipodePagoForm ingresa los datos del nuevo tipo de pago (3), IUTipodePagoForm solicita al objeto Validar que añada la información del nuevo tipo de pago (4), en caso de ser válida la información, el objeto Validar solicita al objeto TipodePago guarde la información (5).

El objeto TipodePago retorna la lista de tipo de pagos al objeto IUTipodePagoList mostrando la lista de tipo de pagos (6). El usuario Operador solicita al objeto IUTipodePagoList y este a su vez al objeto Modificar el formulario para modificar un tipo de pago (7). El objeto Validar solicita al objeto TipodePago encuentre el registro solicitado por medio de su código, el objeto TipodePago retorna el registro solicitado al objeto Validar y este a su vez al objeto IUTipodePagoForm y este a su vez la despliega al usuario Operador (8 ,9 ,10 ,11 ,12). El usuario Operador por medio del objeto IUTipodePagoForm ingresa los datos del tipo de pago (13), IUTipodePagoForm solicita al objeto Validar que añada la información del tipo de pago (14), en caso de ser válida la información, el objeto Validar solicita al objeto TipodePago guarde la información (15). El objeto TipodePago retorna la lista de tipo de pagos al objeto IUTipodePagoList mostrando la lista de tipo de pagos (16). El usuario Operador solicita al objeto IUTipodePagoList y este al objeto Eliminar para eliminar los datos de un tipo de pago (17). El objeto Validar envía el código del tipo de pago a ser eliminada al objeto Banco el cual elimina dicho registro (18,19). El objeto Banco retorna la lista de tipo de pagos al objeto IUTipodePagoList mostrando la lista de tipo de pagos al usuario Operador (20). El usuario Operador solicita al objeto IUTipodePagoList y este al objeto ModificarParámetros para asignar un parámetro al tipo de pago seleccionado (21). El objeto TipodePago retorna la lista de tipo de pagos al objeto IUTipodePagoList mostrando la lista de tipo de pagos al usuario Operador (22,24).

Tabla 2 - 21: Flujo de Sucesos:05 Gestionar Tipo de Pago

2.1.4.2.4 CU:06 Gestionar Cargo.

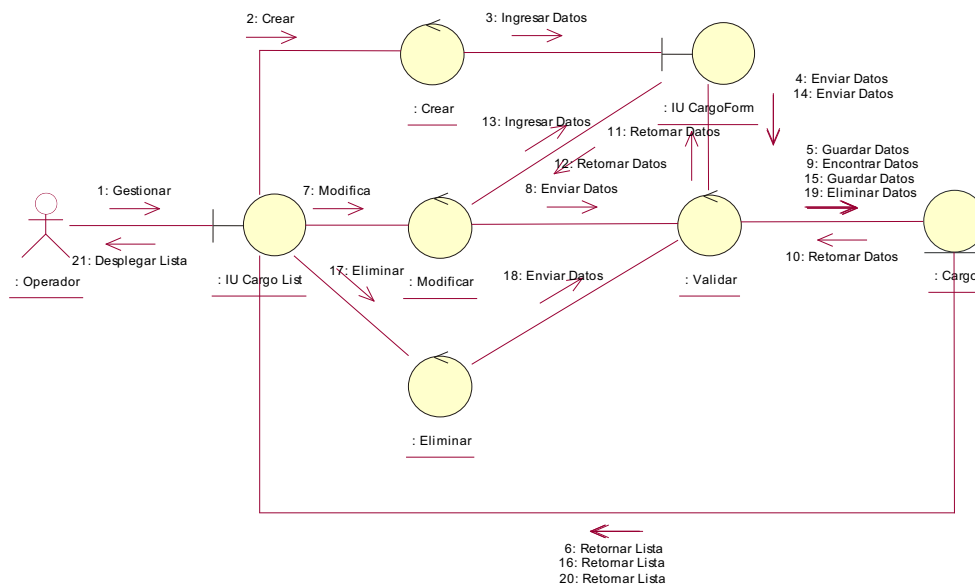


Figura 2 - 22: Diagrama Colaboración:06 Gestionar Cargo

Flujo de Sucesos

El usuario Operador solicita al objeto IUCargoList y este a su vez al objeto Crear el formulario de ingreso de un nuevo cargo(1,2). El usuario Operador por medio del objeto IUCargoForm ingresa los datos del nuevo banco (3), IUCargoForm solicita al objeto Validar que añada la información del nuevo cargo (4), en caso de ser válida la información, el objeto Validar solicita al objeto Cargo que guarde la información (5). El objeto Cargo retorna la lista de cargos al objeto IUCargoList mostrando la lista de cargos (6). El usuario Operador solicita al objeto IUCargoList y este a su vez al objeto Modificar el formulario para modificar un cargo (7). El objeto Validar solicita al objeto Cargo que encuentre el registro solicitado por medio de su código, el objeto Cargo retorna el registro solicitado al objeto Validar y este a su vez al objeto IUCargoForm y este a su vez la despliega al usuario Operador (8 ,9 ,10 ,11 ,12). El usuario Operador por medio del objeto IUCargoForm ingresa los datos del cargo (13), IUCargoForm solicita al objeto Validar que añada la información del cargo (14), en caso de ser válida la información, el objeto Validar solicita al objeto Cargo que guarde la información (15). El objeto Cargo retorna la lista de cargos al objeto

IUCargoList mostrando la lista de cargos (16). El usuario Operador solicita al objeto IUCargoList y este al objeto Eliminar para eliminar los datos de un cargo (17). El objeto Validar envía el código del cargo a ser eliminado al objeto Cargo el cual elimina dicho registro (18,19). El objeto Cargo retorna la lista de cargos al objeto IUCargoList mostrando la lista de cargos al usuario Operador (20,21).

Tabla 2 - 22: Flujo de Sucesos:06 Gestionar Cargo

2.2.1.3 Módulo Gestionar Empleado

2.1.4.3.1 CU:07 Gestionar Empleado.

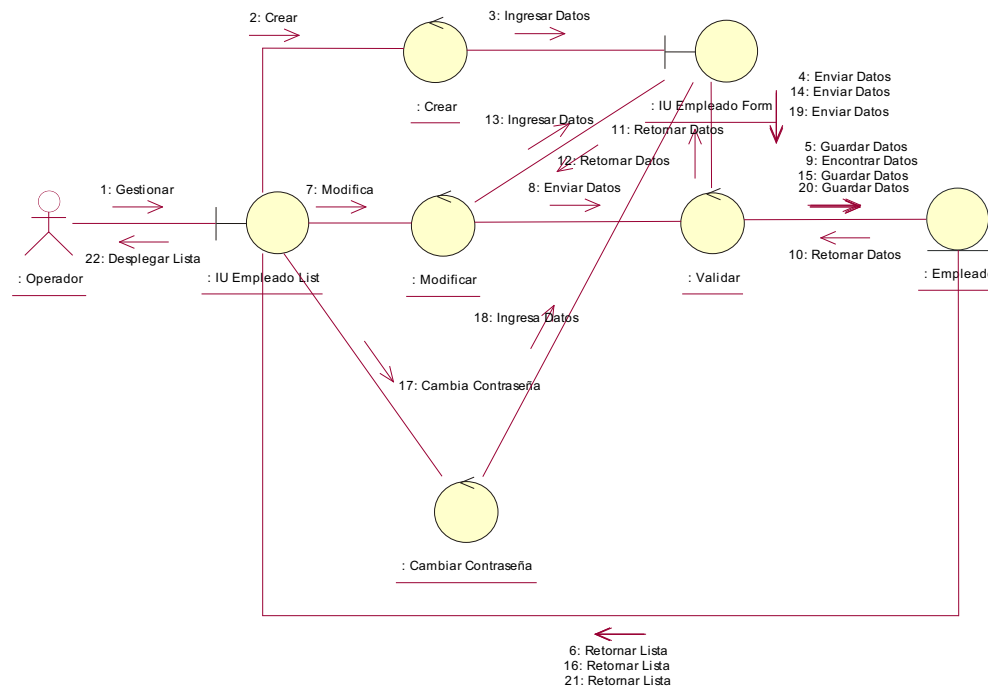


Figura 2 - 23: Diagrama Colaboración:07 Gestionar Empleado

Flujo de Sucesos
<p>El usuario Operador solicita al objeto IUEmpleadoList y este a su vez al objeto Crear el formulario de ingreso de un nuevo empleado(1,2). El usuario Operador por medio del objeto IUEmpleadoForm ingresa los datos del nuevo empleado (3), IUEmpleadoForm solicita al objeto Validar que añada la información del nuevo empleado (4), en caso de</p>

ser válida la información, el objeto Validar solicita al objeto Empleado guarde la información (5). El objeto Empleado retorna la lista de empleados al objeto IUEmpleadoList mostrando la lista de empleados (6). El usuario Operador solicita al objeto IUEmpleadoList y este a su vez al objeto Modificar el formulario para modificar un empleado (7). El objeto Validar solicita al objeto Empleado encuentre el registro solicitado por medio de su código, el objeto Empleado retorna el registro solicitado al objeto Validar y este a su vez al objeto IUEmpleadoForm y este a su vez la despliega al usuario Operador (8 ,9 ,10 ,11 ,12). El usuario Operador por medio del objeto IUEmpleadoForm ingresa los datos del empleado (13), IUEmpleadoForm solicita al objeto Validar que añada la información del empleado (14), en caso de ser válida la información, el objeto Validar solicita al objeto Empleado guarde la información (15). El objeto Empleado retorna la lista de empleados al objeto IUEmpleadoList mostrando la lista de empleados (16). El usuario Operador solicita al objeto IUEmpleadoList y este al objeto CambiarContraseña para cambiar la contraseña de un empleado (17,18), IUEmpleadoForm solicita al objeto Validar que añada la información de la contraseña(19), en caso de ser válida la información, el objeto Validar solicita al objeto Empleado guarde la información (20).El objeto Empleado retorna la lista de empleados al objeto IUEmpleadoList mostrando la lista de empleados al usuario Operador (21,22).

Tabla 2 - 23: Flujo de Sucesos:07 Gestionar Empleado

2.2.1.4 Módulo Generar Rol de Pago

2.1.4.4.1 CU:08 Gestionar Cierre de Mes.

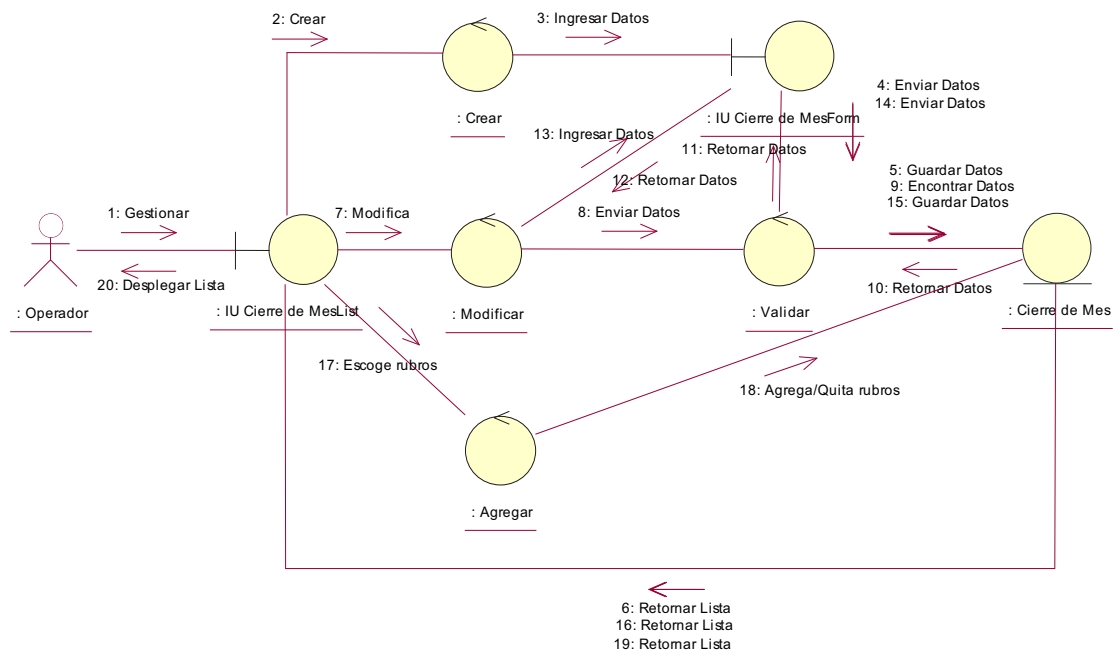


Figura 2 - 24: Diagrama Colaboración:08 Gestionar Cierre de Mes

Flujo de Sucesos

El usuario Operador solicita al objeto IUCierredeMesList y este a su vez al objeto Crear el formulario de ingreso de un nuevo cierre de mes(1,2). El usuario Operador por medio del objeto IUCierredeMesForm ingresa los datos del nuevo cierre de mes (3), IUCierredeMesForm solicita al objeto Validar que añada la información del nuevo cierre de mes (4), en caso de ser válida la información, el objeto Validar solicita al objeto CierredeMes guarde la información (5). El objeto CierredeMes retorna la lista de cierre de mes al objeto IUCierredeMesList mostrando la lista de cierre de mes (6). El usuario Operador solicita al objeto IUCierredeMesList y este a su vez al objeto Modificar el formulario para modificar un cierre de mes (7). El objeto Validar solicita al objeto CierredeMes encuentre el registro solicitado por medio de su código, el objeto CierredeMes retorna el registro solicitado al objeto Validar y este a su vez al objeto IUCierredeMesForm y este a su vez la despliega al usuario Operador (8 ,9 ,10 ,11 ,12). El usuario Operador por medio del objeto

IUCierredeMesForm ingresa los datos del usuario (13), IUCierredeMesForm solicita al objeto Validar que añada la información del cierre de mes (14), en caso de ser válida la información, el objeto Validar solicita al objeto CierredeMes guarde la información (15). El objeto CierredeMes retorna la lista de cierre de mes al objeto IUCierredeMesList (16). El usuario Operador solicita al objeto IUCierredeMesList y este al objeto Agregar para asignar un rubro al cierre de mes seleccionado (17,18). El objeto Cierre de Mes retorna la lista de cierres de mes al objeto IUCierredeMesList mostrando la lista de cierres de mes al usuario Operador (19,20).

Tabla 2 - 24: Flujo de Sucesos:08 Gestionar Cierre de Mes

2.1.4.4.2 CU:09 Gestionar Vacaciones.

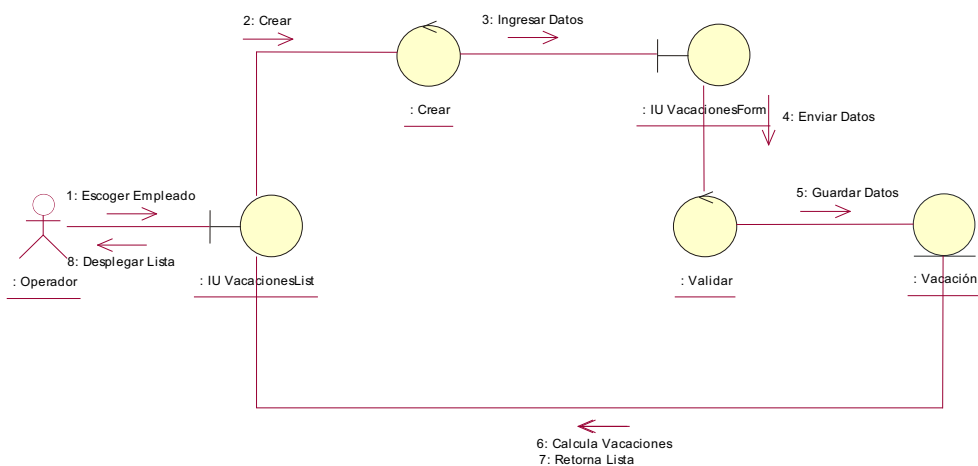


Figura 2 - 25: Diagrama Colaboración:09 Gestionar Vacaciones

Flujo de Sucesos

El usuario Operador escoge un empleado y lista las vacaciones en el objeto IUVacacionesList(1). El usuario Operador solicita al objeto IUVacacionesList y este a su vez al objeto Crear el formulario de ingreso de una nueva vacación(2). El usuario Operador por medio del objeto IUVacacionesForm ingresa los datos de la nueva vacación (3), IUVacacionesForm solicita al objeto Validar que añada la información de la nueva vacación (4), en caso de ser válida la información, el

objeto Validar solicita al objeto Vacación guarde la información (5). El objeto Vacación calcula las vacaciones del empleado y retorna al objeto IUVacacionesList (6,7,8).

Tabla 2 - 25: Flujo de Sucesos:09 Gestionar Vacaciones

2.1.4.4.3 CU:10 Gestionar Horas Extras.

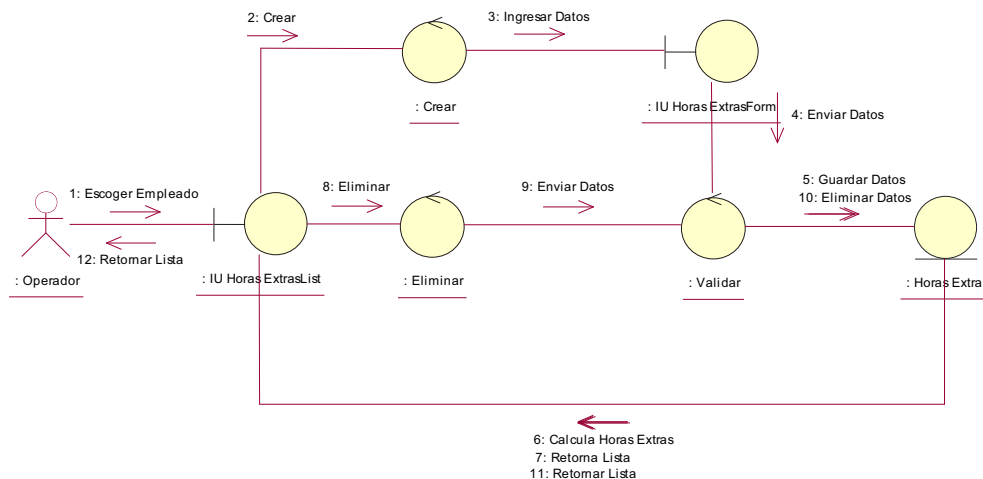


Figura 2 - 26: Diagrama Colaboración:10 Gestionar Horas Extras

Flujo de Sucesos

El usuario Operador escoge un empleado y lista las horas extras en el objeto IUHorasExtrasList(1). El usuario Operador solicita al objeto IUHorasExtrasList y este a su vez al objeto Crear el formulario de ingreso de una nueva hora extra(2). El usuario Operador por medio del objeto IUHorasExtrasForm ingresa los datos de la nueva hora extra (3), IUHorasExtrasForm solicita al objeto Validar que añada la información de la nueva hora extra (4), en caso de ser válida la información, el objeto Validar solicita al objeto HoraExtras guarde la información (5). El objeto HoraExtras calcula las horas suplementarias o extraordinarias del empleado y retorna al objeto IUHorasExtrasList (6,7). El usuario Operador solicita al objeto IUHorasExtrasList y este al objeto Eliminar para eliminar los datos de una hora extra(8). El objeto Validar envía el código de la hora extra a ser eliminada al objeto HoraExtras el cual elimina dicho registro (9,10). El objeto HoraExtras

retorna la lista de horas extras al objeto IUHorasExtrasList mostrando la lista de horas extras al usuario Operador (11,12).

Tabla 2 - 26: Flujo de Sucesos:10 Gestionar Horas Extras

2.1.4.4.4 CU:11 Gestionar Rol de Pago.

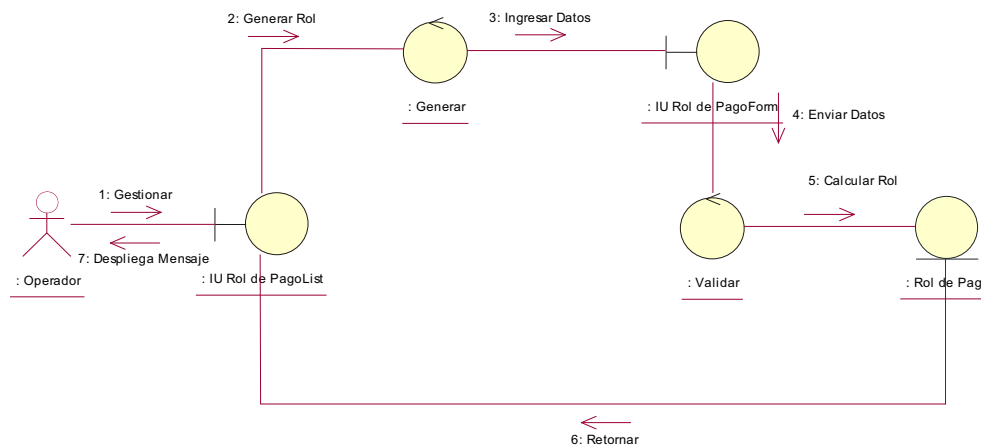


Figura 2 - 27: Diagrama Colaboración:11 Gestionar Rol de Pago

Flujo de Sucesos

El usuario Operador solicita al objeto IU Rol de PagoList y este a su vez al objeto Generar el formulario de ingreso de un nuevo rol de pagos(2). El usuario Operador por medio del objeto IU Rol de PagoForm ingresa los datos del rol de pagos(3), IU Rol de PagoForm solicita al objeto Validar que añada la información del rol de pagos (4), en caso de ser válida la información, el objeto Validar solicita al objeto Rol de Pago guarde la información (5). El objeto Rol de Pago calcula el rol de pagos de cada empleado y retorna al objeto IU Rol de PagoList (6). El objeto IU Rol de PagoList genera un mensaje de éxito o error respectivo de la planilla de pagos (7).

Tabla 2 - 27: Flujo de Sucesos:11 Gestionar Rol de Pago

2.1.4.4.5 CU:12 Gestionar Beneficios.

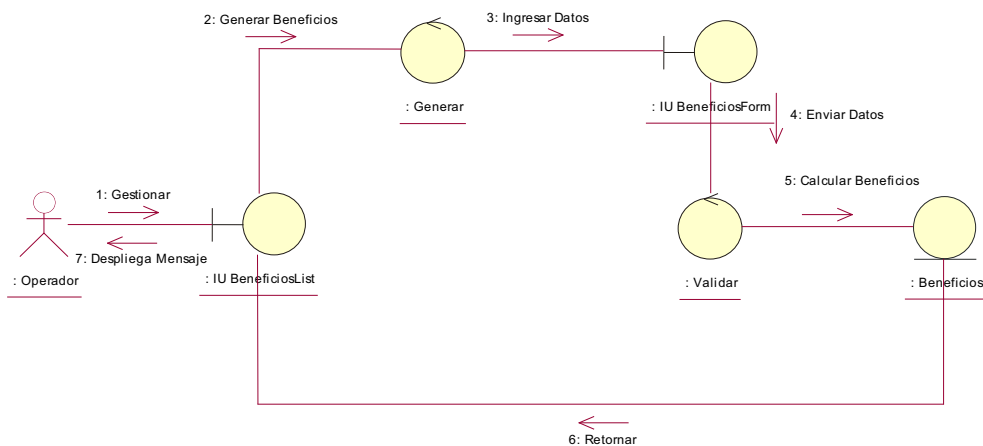


Figura 2 - 28: Diagrama Colaboración:12 Gestionar Beneficios

Flujo de Sucesos

El usuario Operador solicita al objeto IUBeneficiosList y este a su vez al objeto Generar el formulario de ingreso de beneficios(2). El usuario Operador por medio del objeto IUBeneficiosForm ingresa los datos de beneficios(3), IUBeneficiosForm solicita al objeto Validar que añada la información de beneficios(4), en caso de ser válida la información, el objeto Validar solicita al objeto Beneficios guarde la información (5). El objeto Beneficios calcula el los beneficios para cada empleado y retorna al objeto IUBeneficiosList (6). El objeto IUBeneficiosList genera un mensaje de éxito o error respectivo del cálculo de beneficios(7).

Tabla 2 - 28: Flujo de Sucesos:12 Gestionar Beneficios

2.2.1.5 Módulo Generar Finiquito

2.1.4.5.1 CU:13 Generar Finiquito.

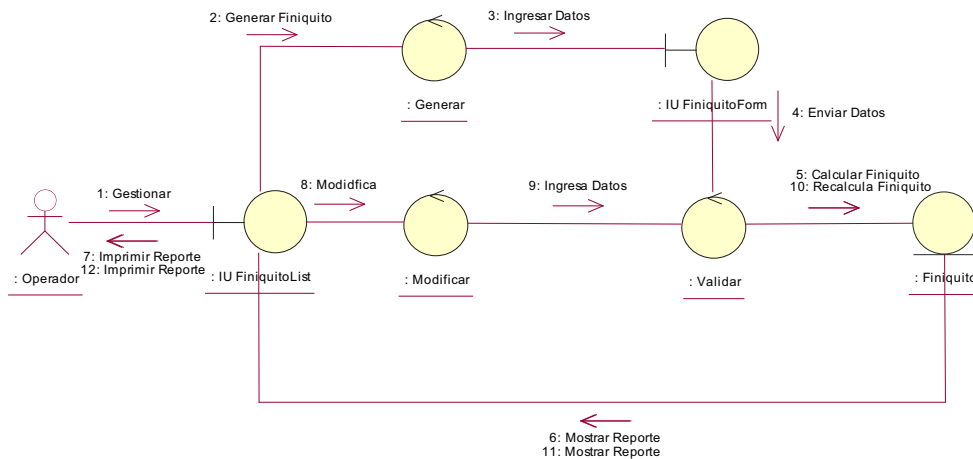


Figura 2 - 29: Diagrama Colaboración:13 Generar Finiquito

Flujo de Sucesos

El usuario Operador solicita al objeto IUFiniquitoList y este a su vez al objeto Generar, generar el formulario de ingreso de finiquitos(2). El usuario Operador por medio del objeto IUFiniquitoForm ingresa los datos de finiquito(3), IUFiniquitoForm solicita al objeto Validar que añada la información de finiquito(4), en caso de ser válida la información, el objeto Validar solicita al objeto Finiquito guarde la información (5). El objeto Finiquito calcula el finiquito del empleado seleccionado y retorna al objeto IUFiniquitoList mostrando el reporte de finiquito(6). El usuario Operador solicita al objeto IUFiniquitoList imprimir el reporte de finiquito(7) del empleado. El usuario Operador solicita al objeto IUFiniquitoList y este a su vez al objeto Modificar el formulario de ingreso de finiquitos(8). El usuario Operador por medio del objeto IUFiniquitoForm ingresa los datos de finiquito, IUFiniquitoForm solicita al objeto Validar que añada la información de finiquito(9), en caso de ser válida la información, el objeto Validar solicita al objeto Finiquito guarde la información (10). El objeto Finiquito recalcula el finiquito del empleado seleccionado y retorna al objeto IUFiniquitoList mostrando el reporte de finiquito(11). El usuario Operador solicita al objeto IUFiniquitoList imprimir el reporte de finiquito(12) del empleado.

Tabla 2 - 29: Flujo de Sucesos:13 Generar Finiquito

2.2.1.6 Módulo Generar Reportes

2.1.4.6.1 CU:14 Generar Reportes.

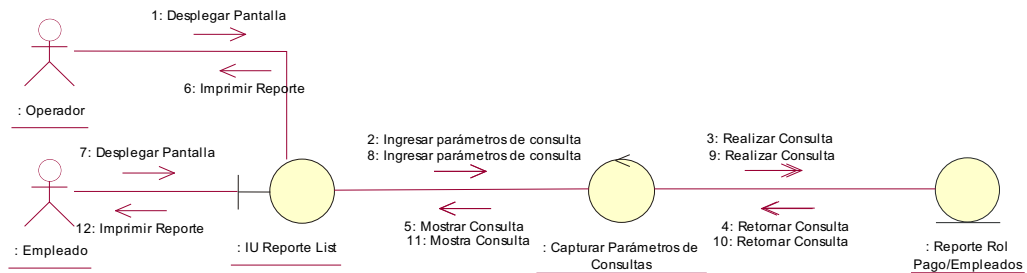


Figura 2 - 30: Diagrama Colaboración:14 Generar Reportes

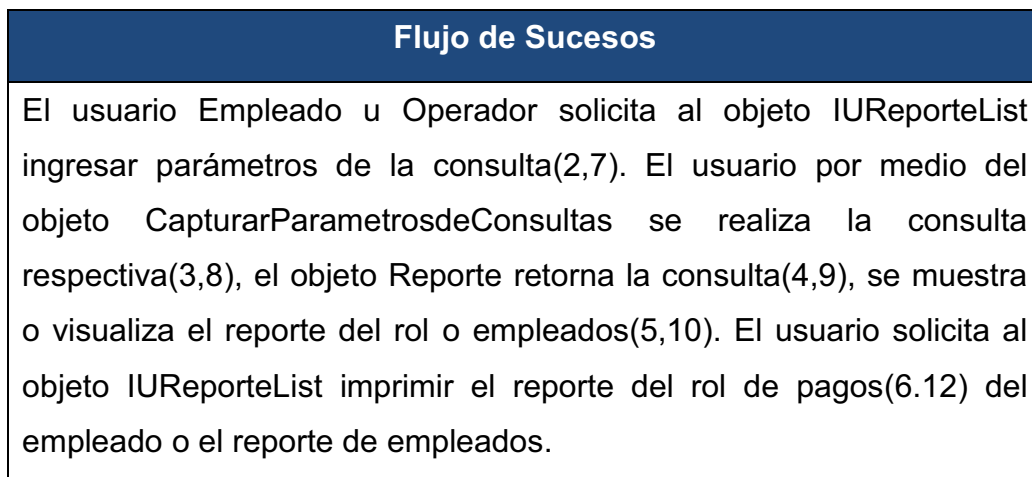


Tabla 2 - 30: Flujo de Sucesos:14 Generar Reportes

2.2 DISEÑO DEL SISTEMA.

2.2.1 DIAGRAMAS DE SECUENCIA. [12]

2.2.1.1 Módulo Administrar Accesos.

2.2.1.1.1 CU:01 Gestionar Perfiles.

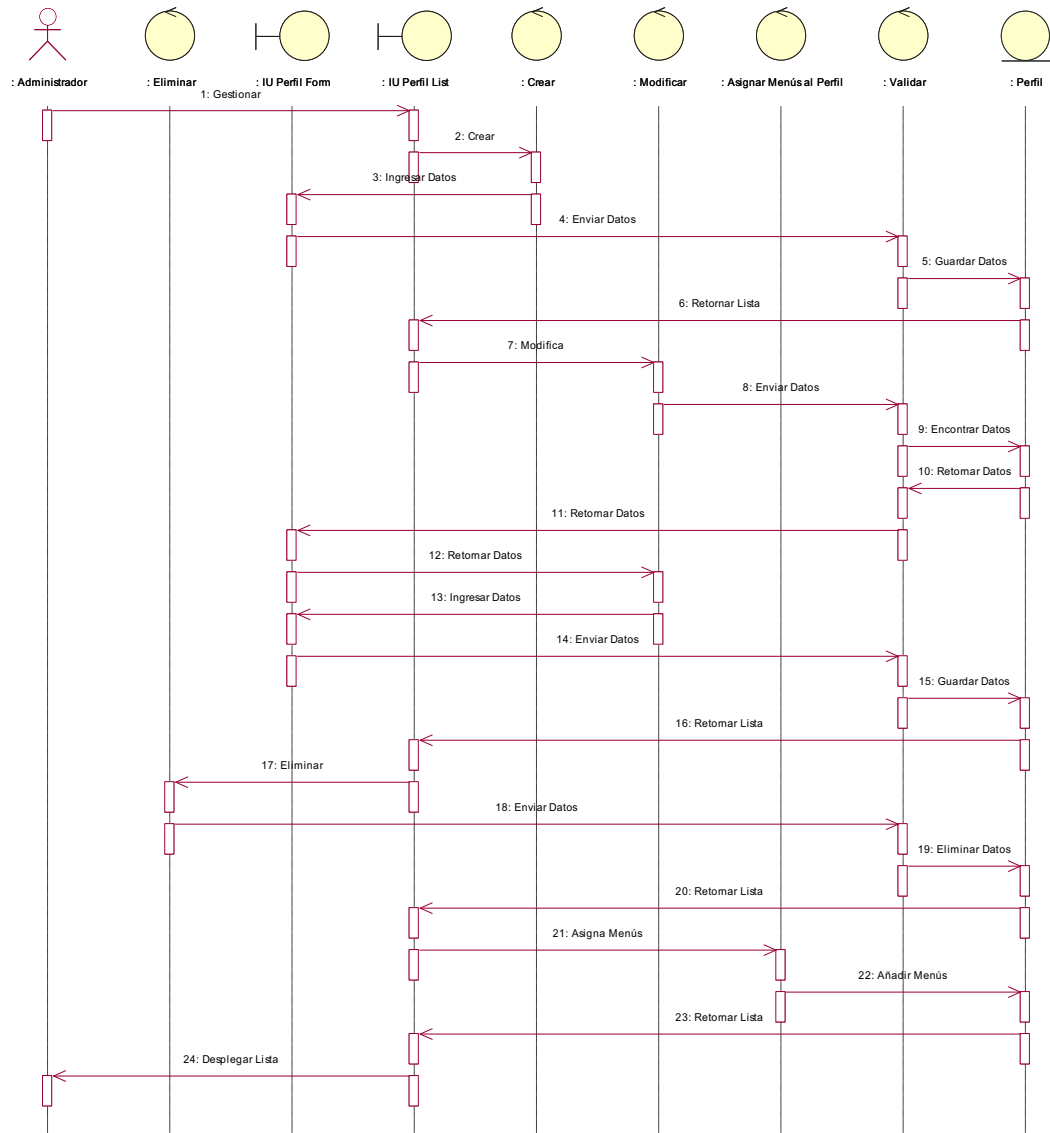


Figura 2 - 31: Diagrama Secuencia:01 Gestionar Perfiles

2.2.1.1.2 CU:02 Gestionar Usuarios.

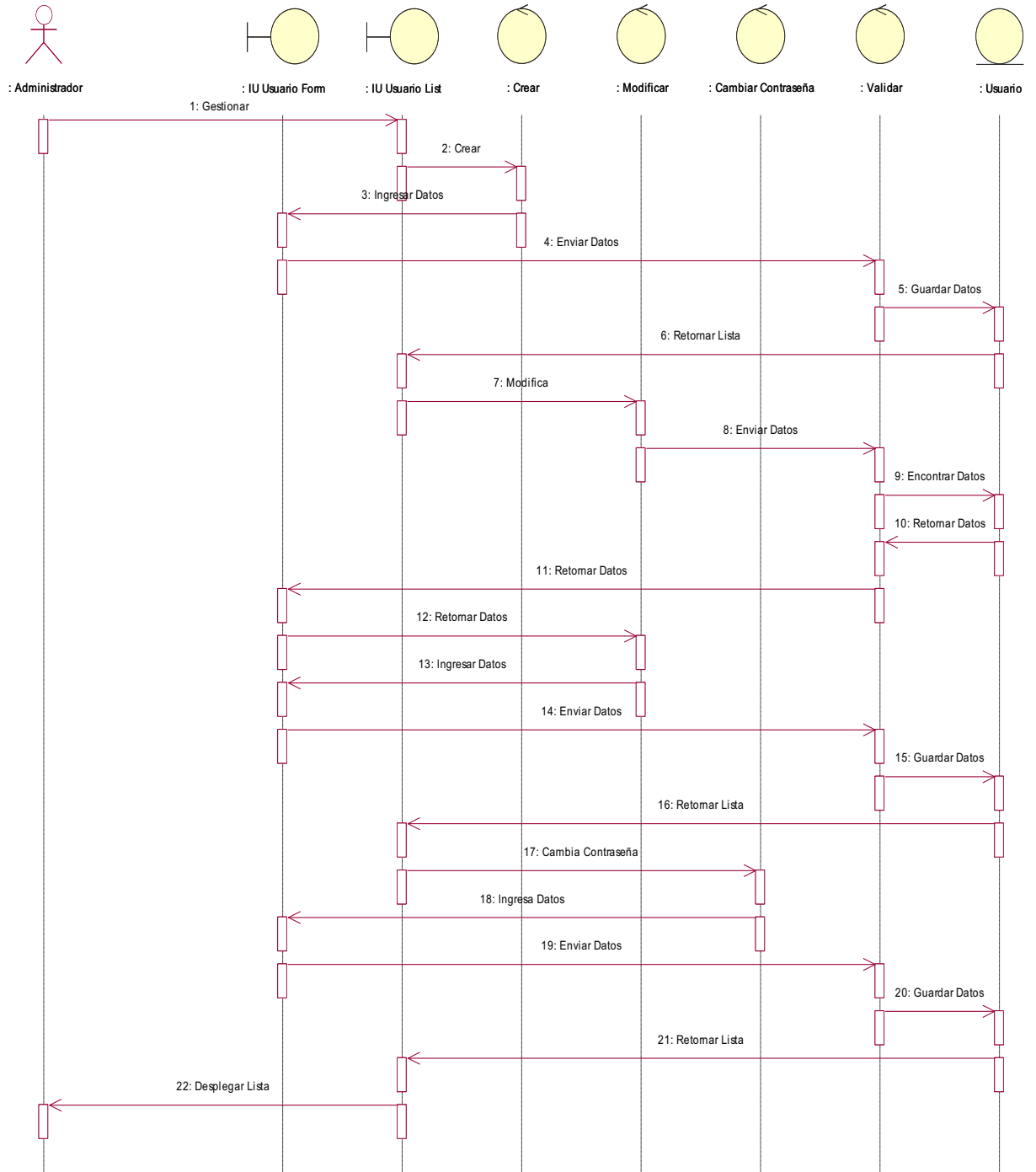


Figura 2 - 32: Diagrama Secuencia:02 Gestionar Usuarios

2.2.1.2 Módulo Configurar Parámetros.

2.2.1.2.1 CU:03 Gestionar Tipo de Contrato.

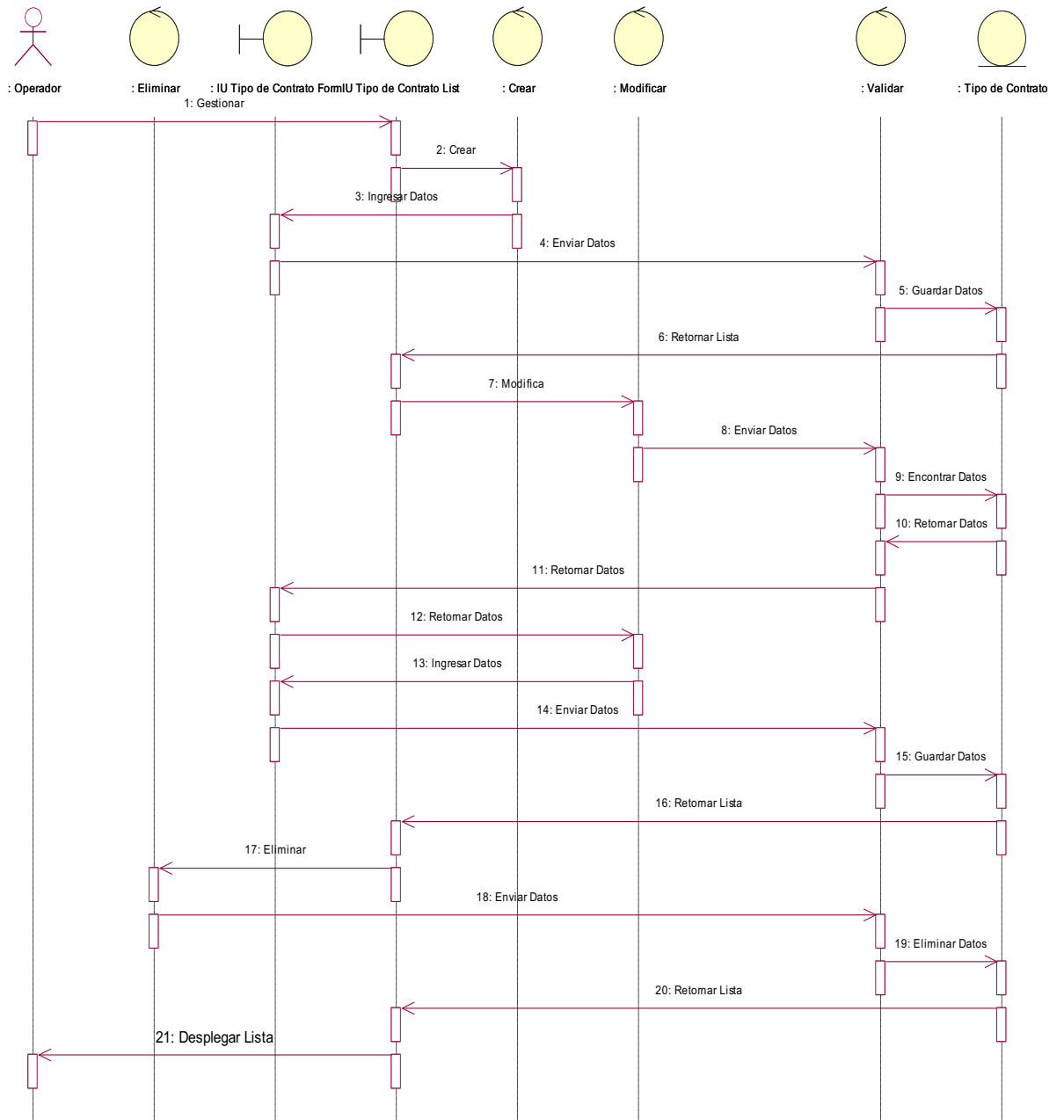


Figura 2 - 33: Diagrama Secuencia:03 Gestionar Tipo de Contrato

2.2.1.2.2 CU:04 Gestionar Banco.

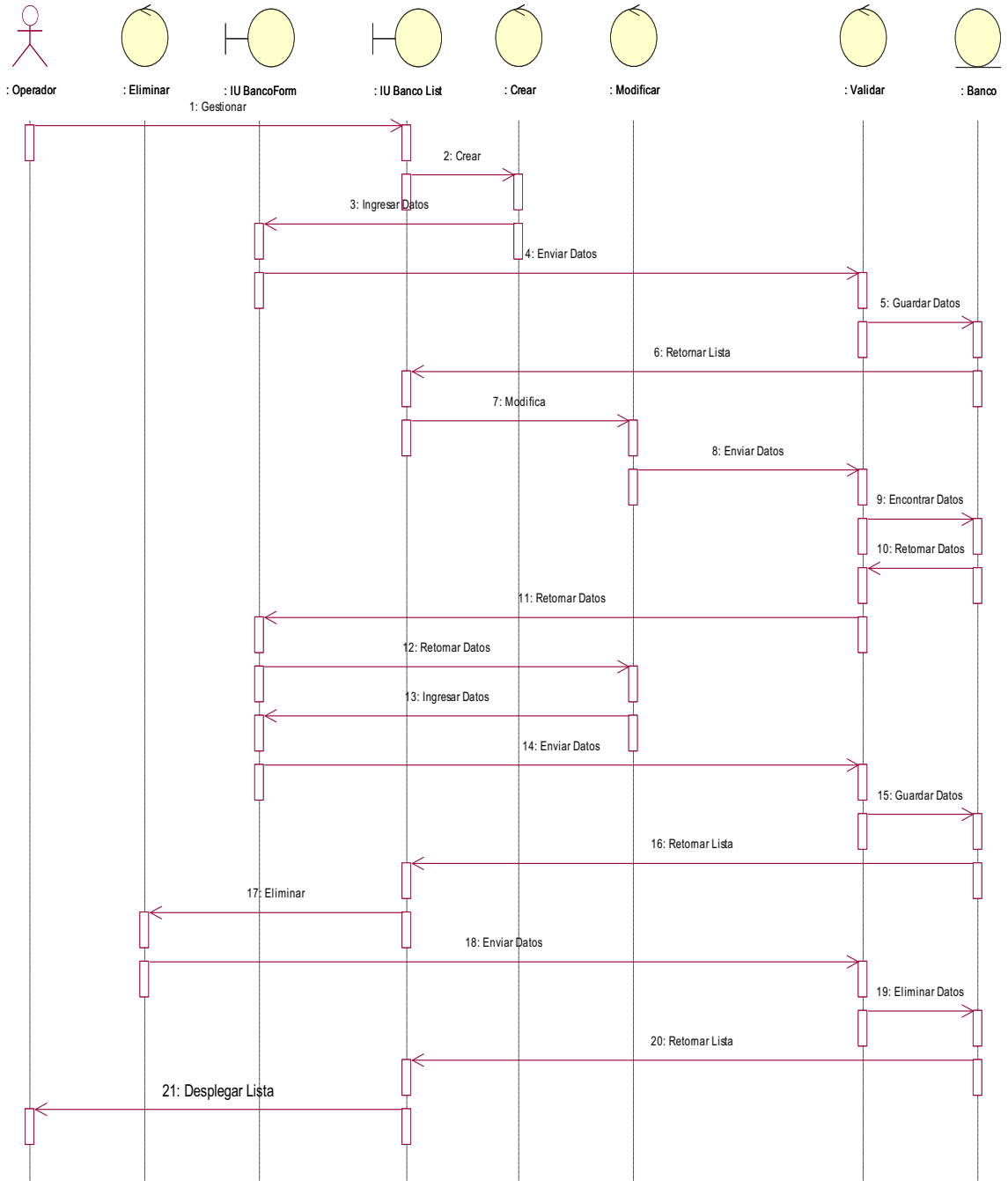


Figura 2 - 34: Diagrama Secuencia:04 Gestionar Banco

2.2.1.2.3 CU:05 Gestionar Tipo de Pago.

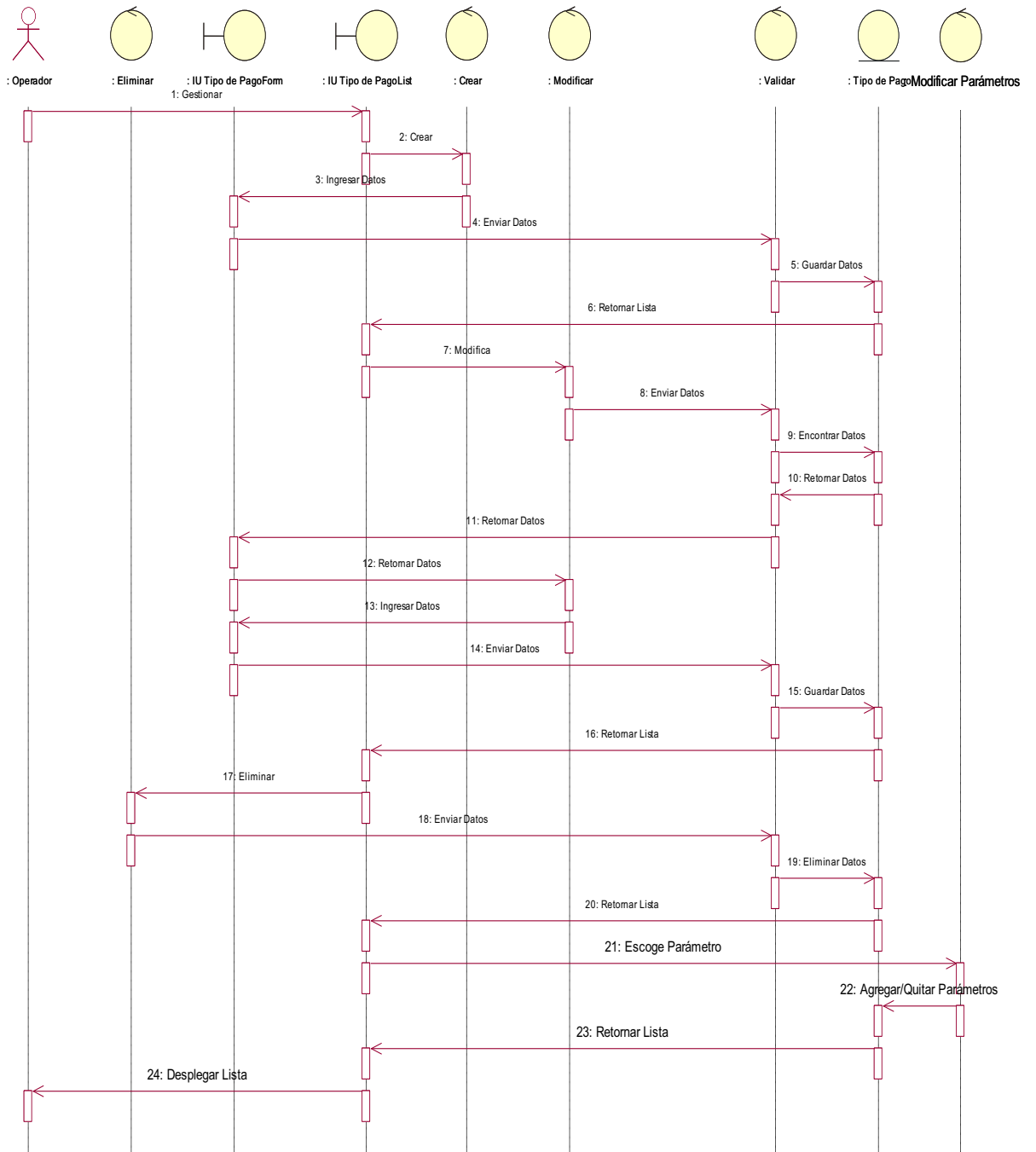


Figura 2 - 35: Diagrama Secuencia:05 Gestionar Tipo de Pago

2.2.1.2.4 CU:06 Gestionar Cargo.

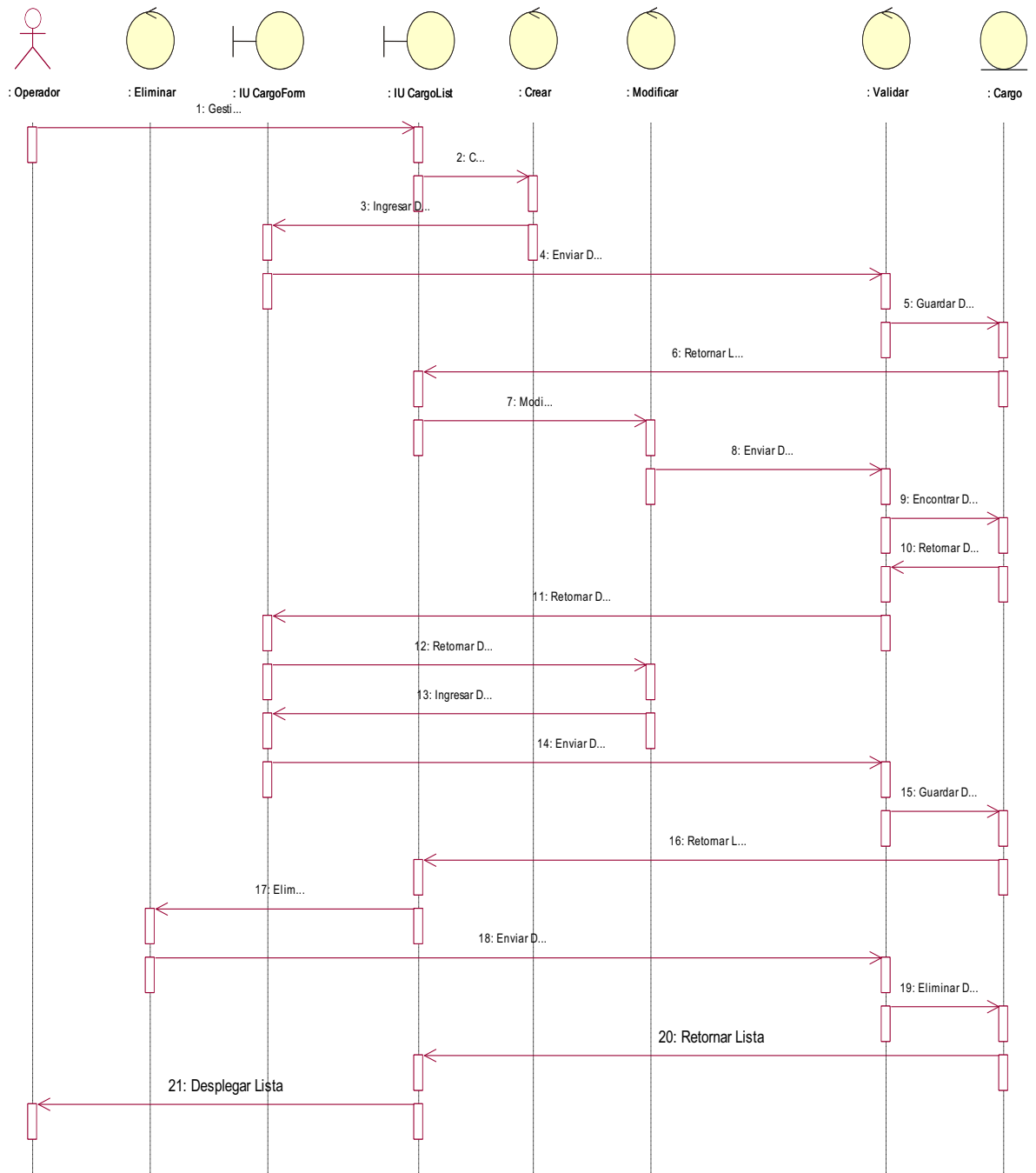


Figura 2 - 36: Diagrama Secuencia:06 Gestionar Cargo

2.2.1.3 Módulo Gestionar Empleado.

2.2.1.3.1 CU:07 Gestionar Empleado.

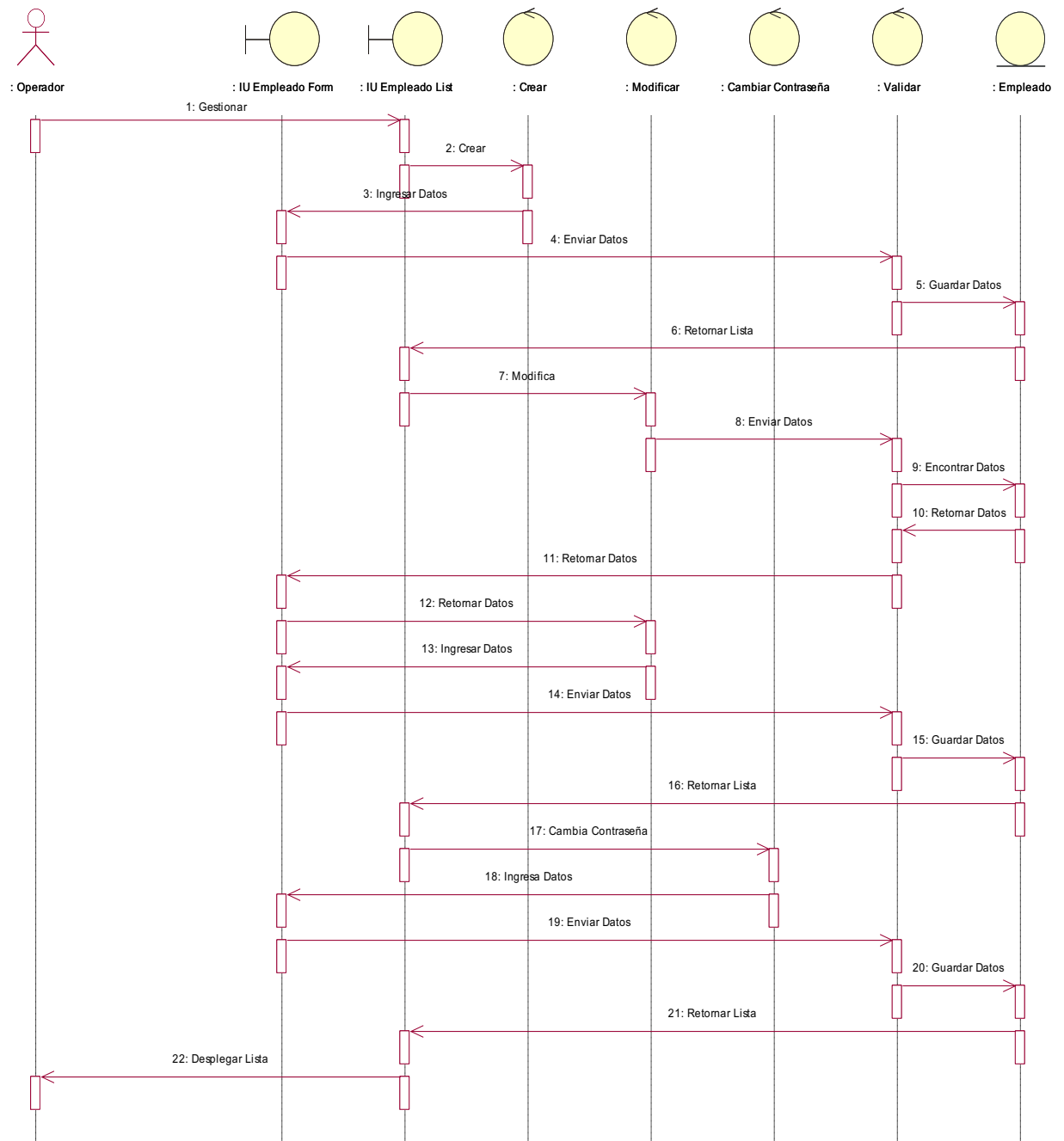


Figura 2 - 37: Diagrama Secuencia:07 Gestionar Empleado

2.2.1.4 Módulo Generar Rol de Pago.

2.2.1.4.1 CU:08 Gestionar Cierre de Mes.

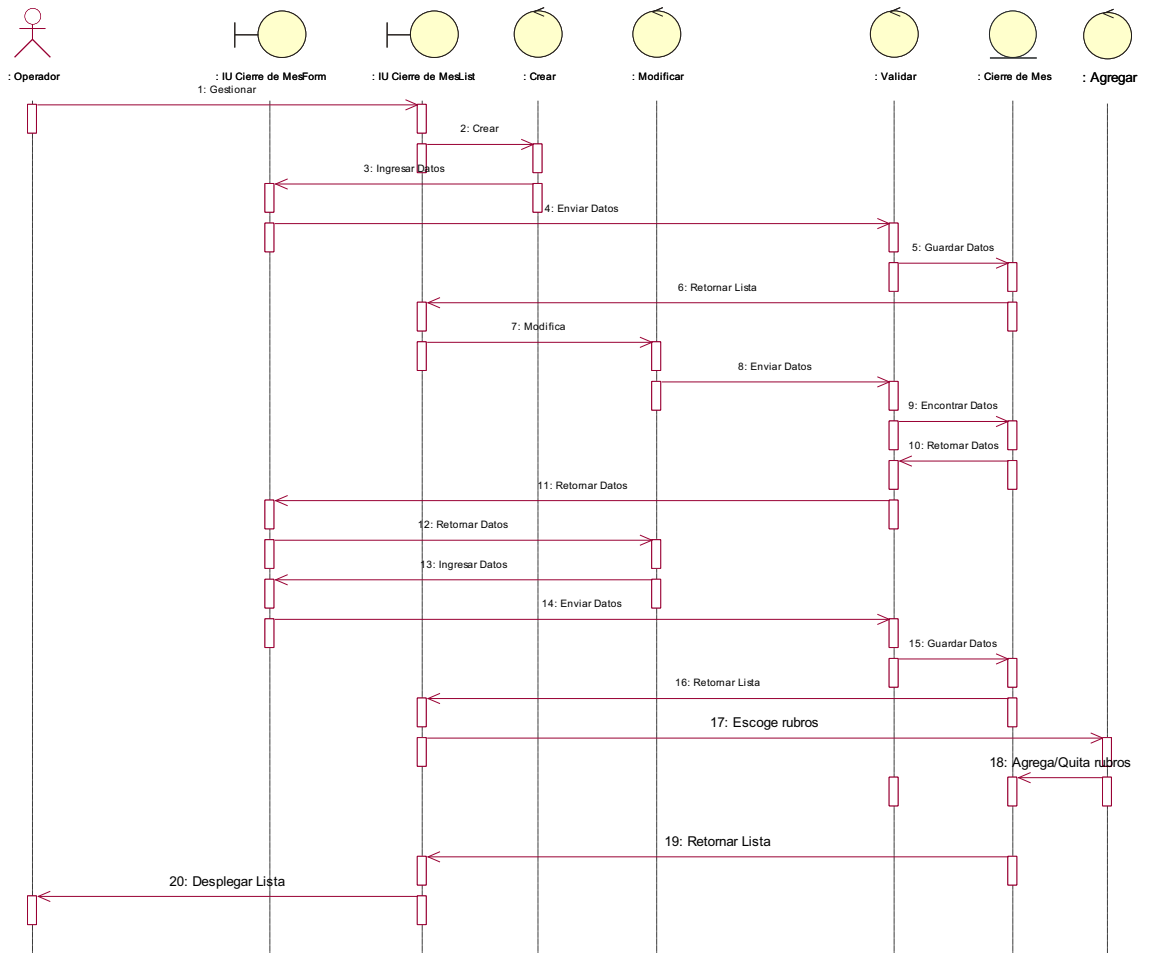


Figura 2 - 38: Diagrama Secuencia:08 Gestionar Cierre de Mes

2.2.1.4.2 CU:09 Gestionar Vacaciones.

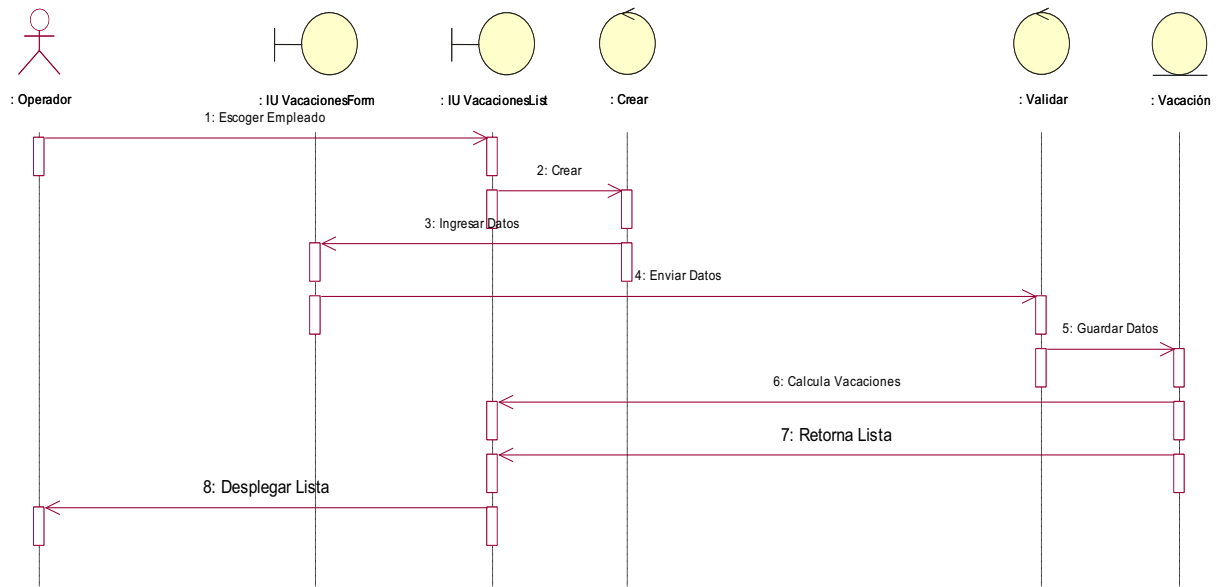


Figura 2 - 39: Diagrama Secuencia:09 Gestionar Vacaciones

2.2.1.4.3 CU:10 Gestionar Horas Extras.

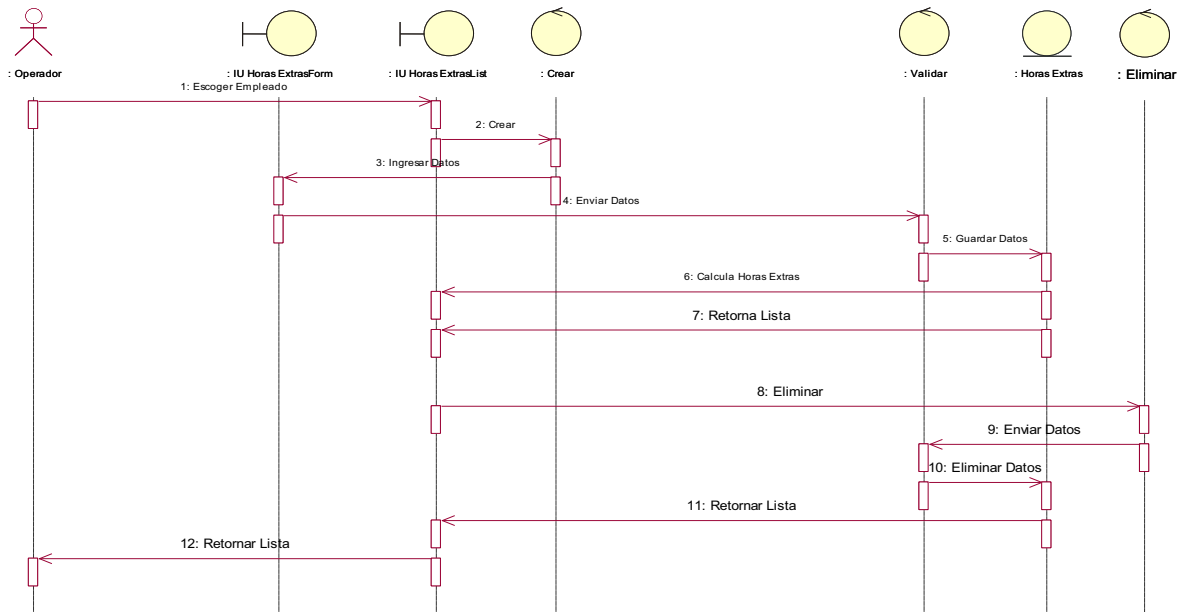


Figura 2 - 40: Diagrama Secuencia:10 Gestionar Horas Extras

2.2.1.4.4 CU:11 Gestionar Rol de Pago.

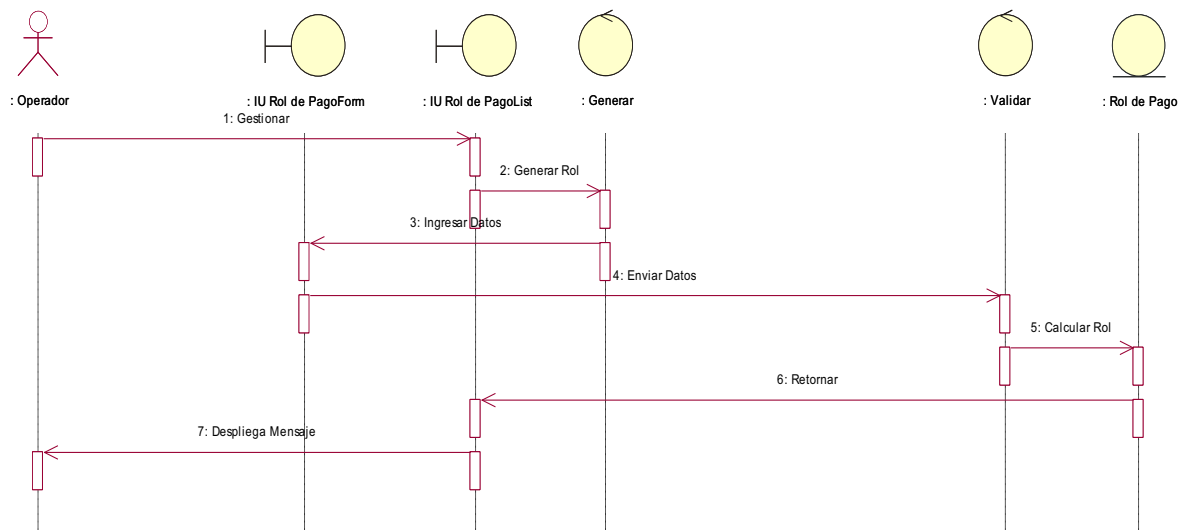


Figura 2 - 41: Diagrama Secuencia:11 Gestionar Rol de Pago

2.2.1.4.5 CU:12 Gestionar Beneficios.

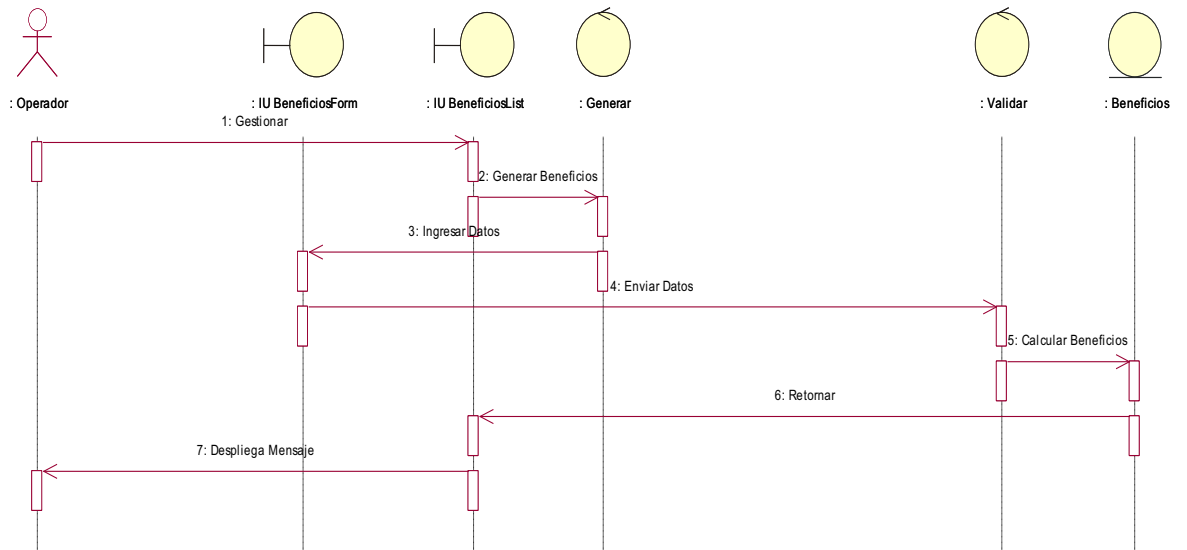


Figura 2 - 42: Diagrama Secuencia:12 Gestionar Beneficios

2.2.1.5 Módulo Generar Finiquito.

2.2.1.5.1 CU:13 Generar Finiquito.

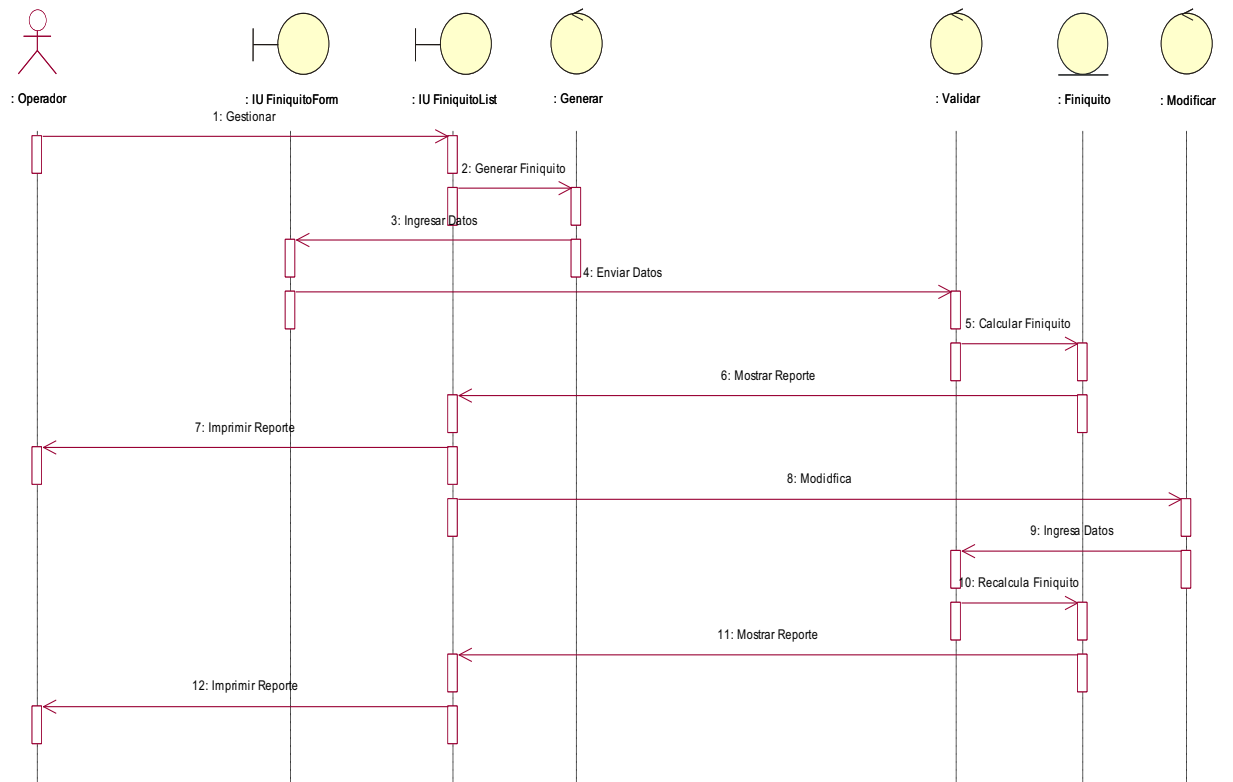


Figura 2 - 43: Diagrama Secuencia:13 Generar Finiquito

2.2.1.6 Módulo Generar Reportes.

2.2.1.6.1 CU:14 Generar Reportes.

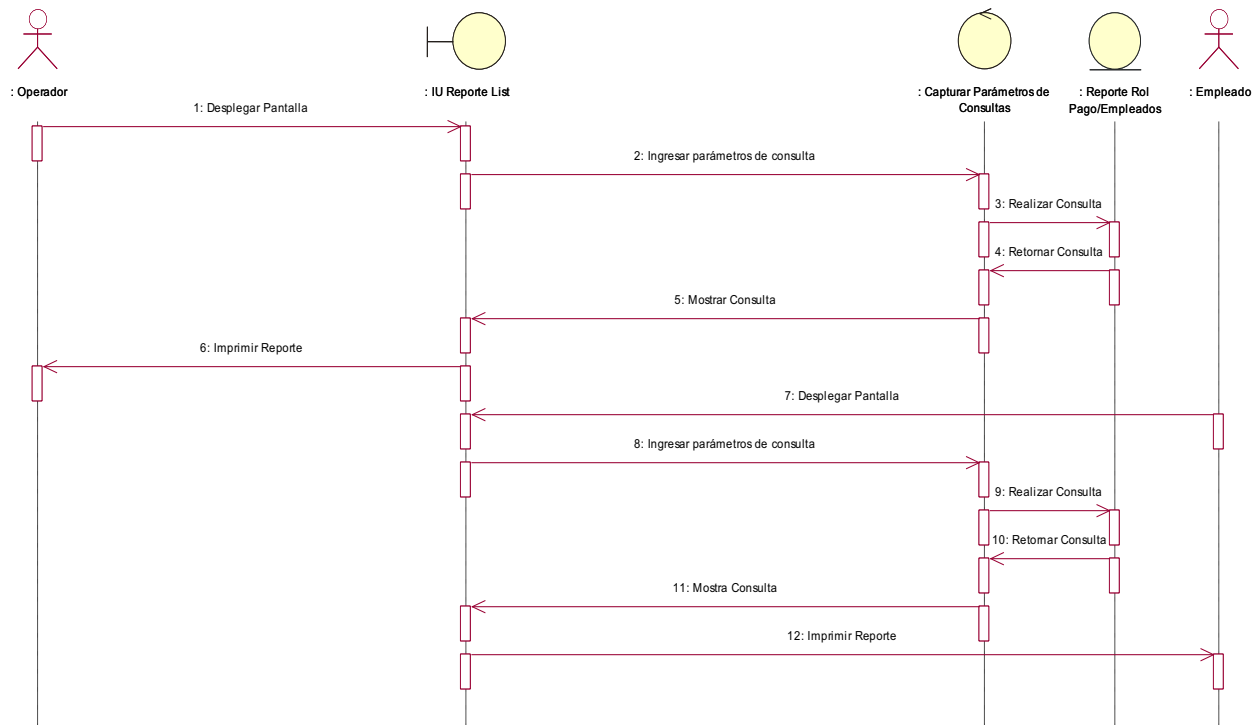
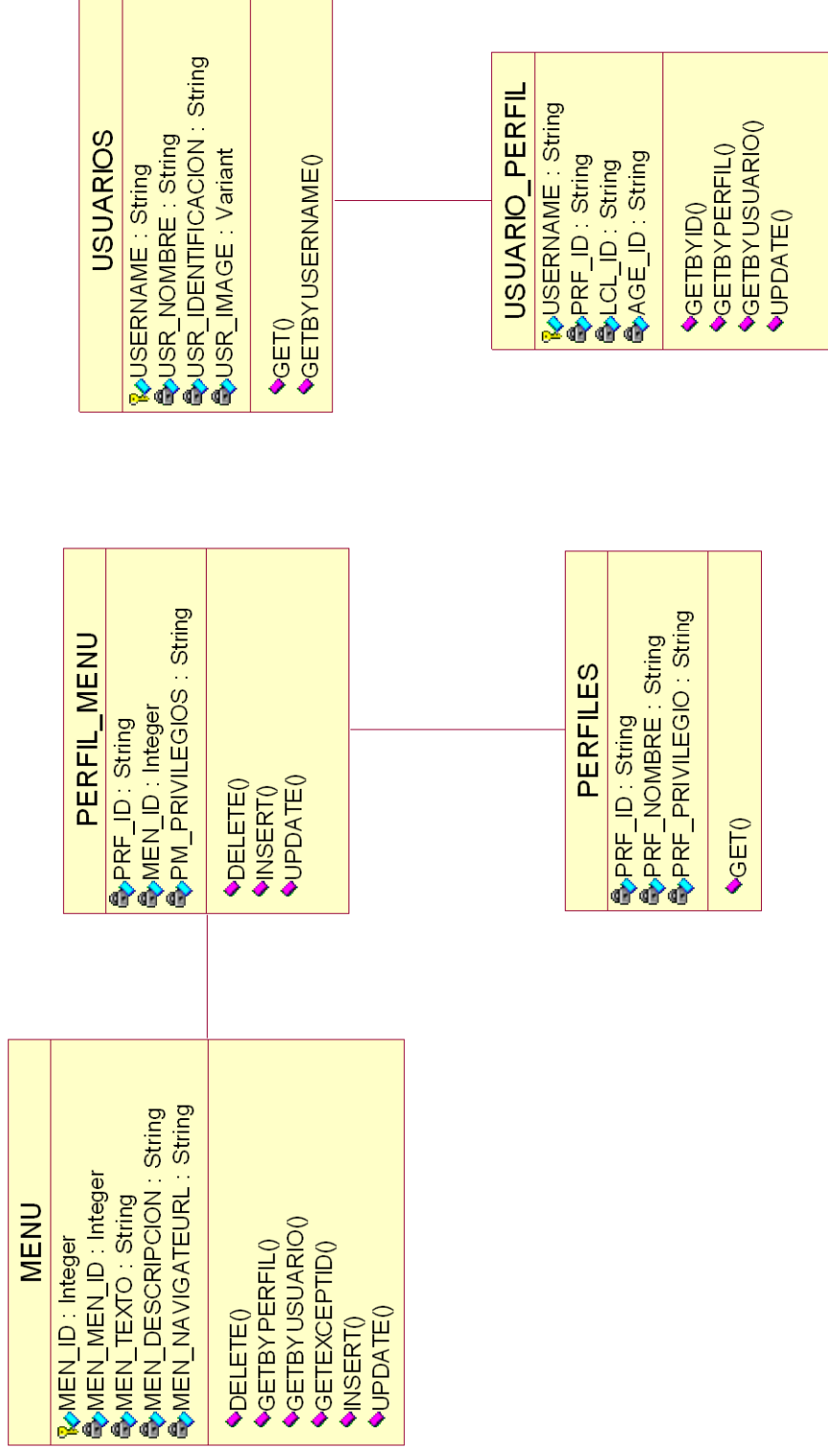
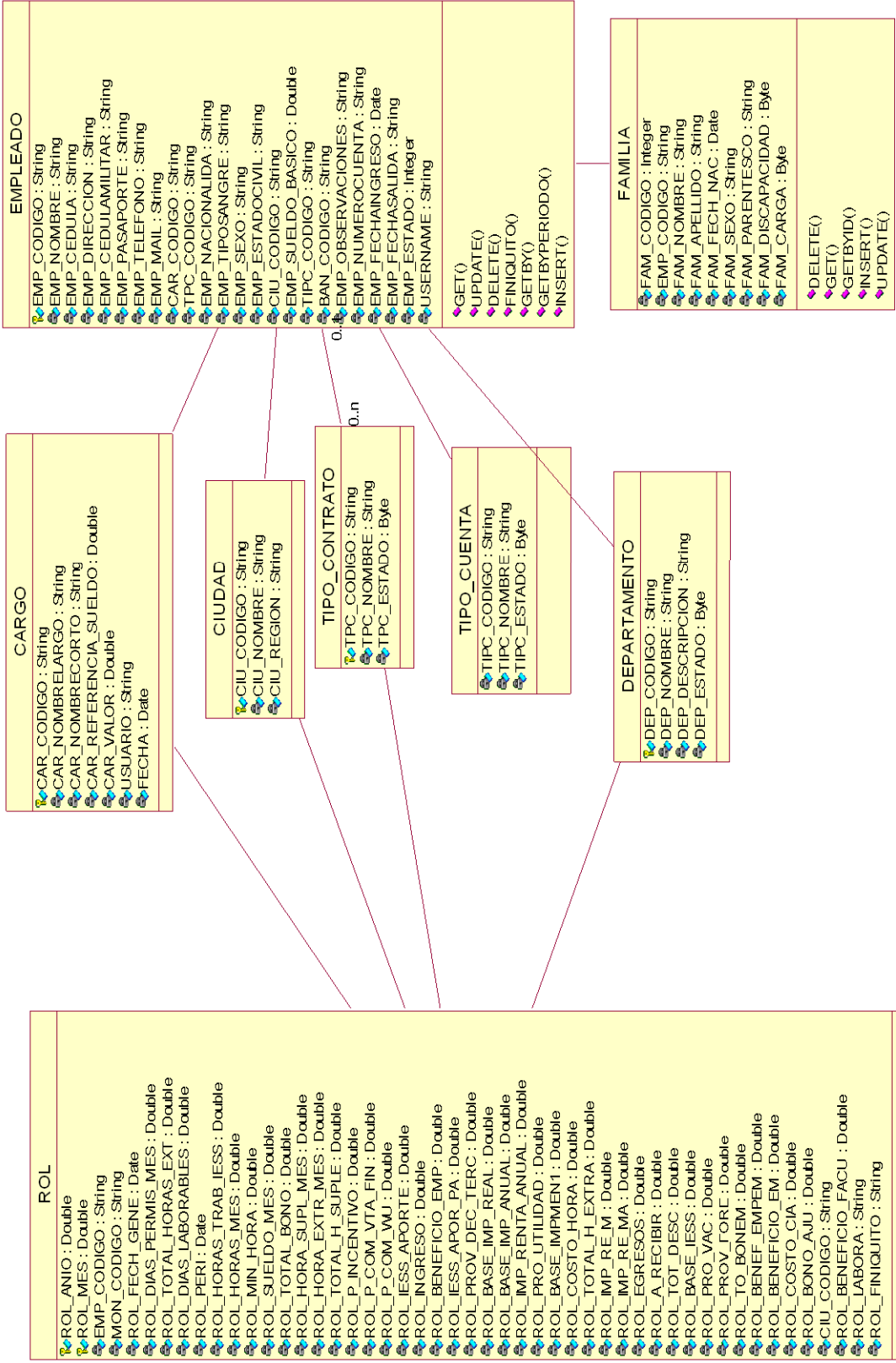


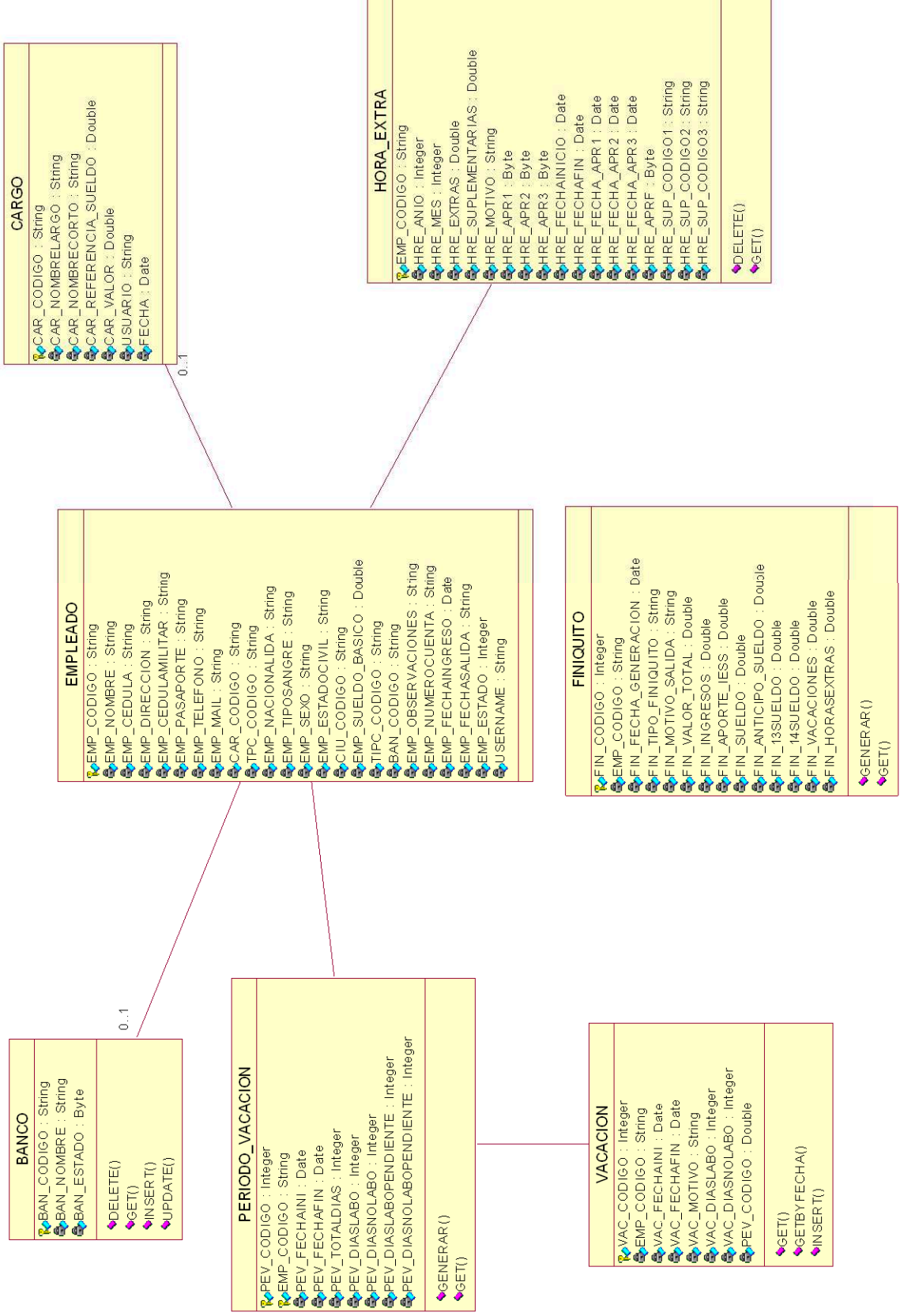
Figura 2 - 44: Diagrama Secuencia:14 Generar Reportes

2.2.2 DISEÑO DEL DIAGRAMA DE CLASES.

A continuación se presenta el diagrama de clases del sistema, para un mejor entendimiento y visualización se la dividió en varias partes:







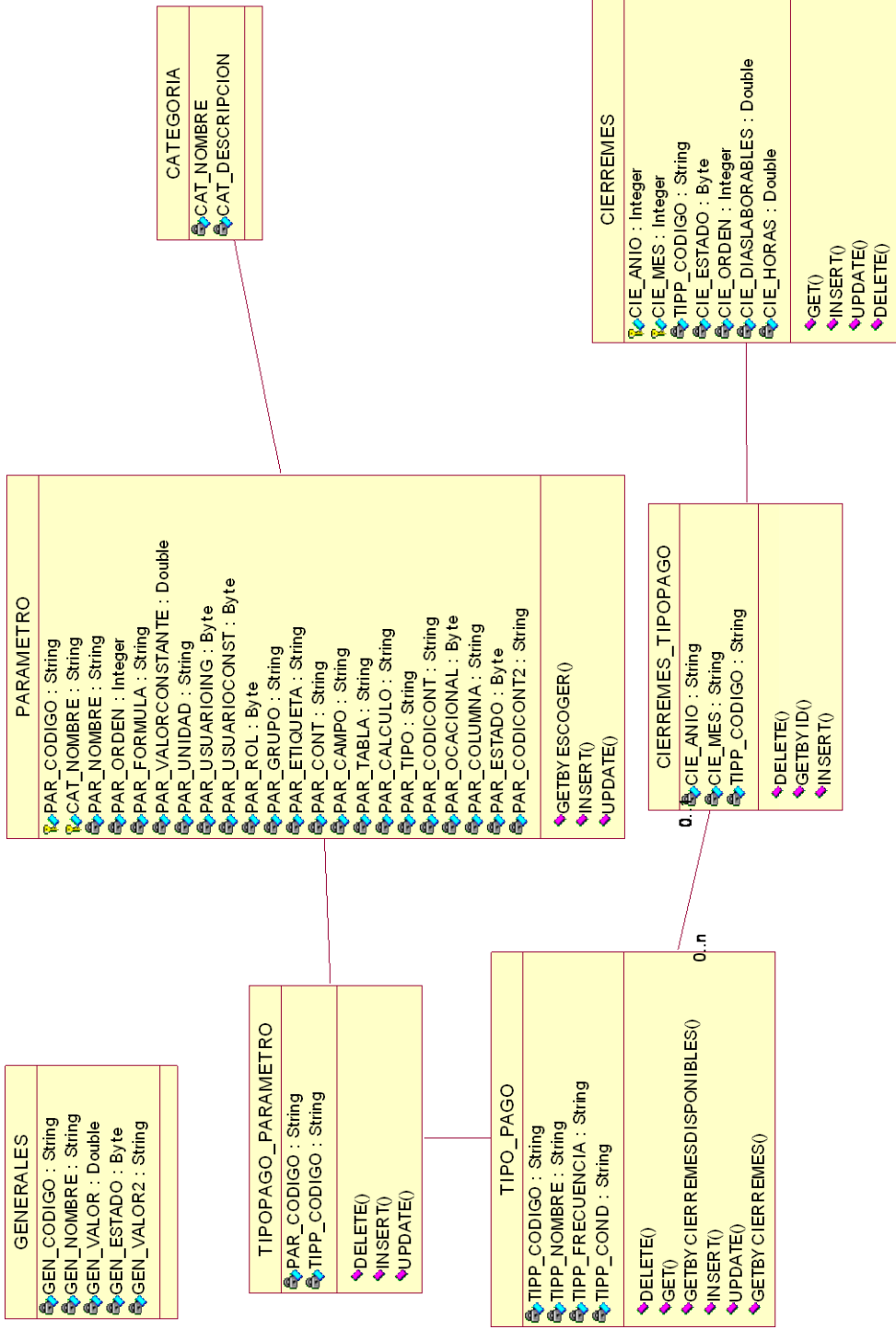


Tabla 2 - 31: Diagrama de clases

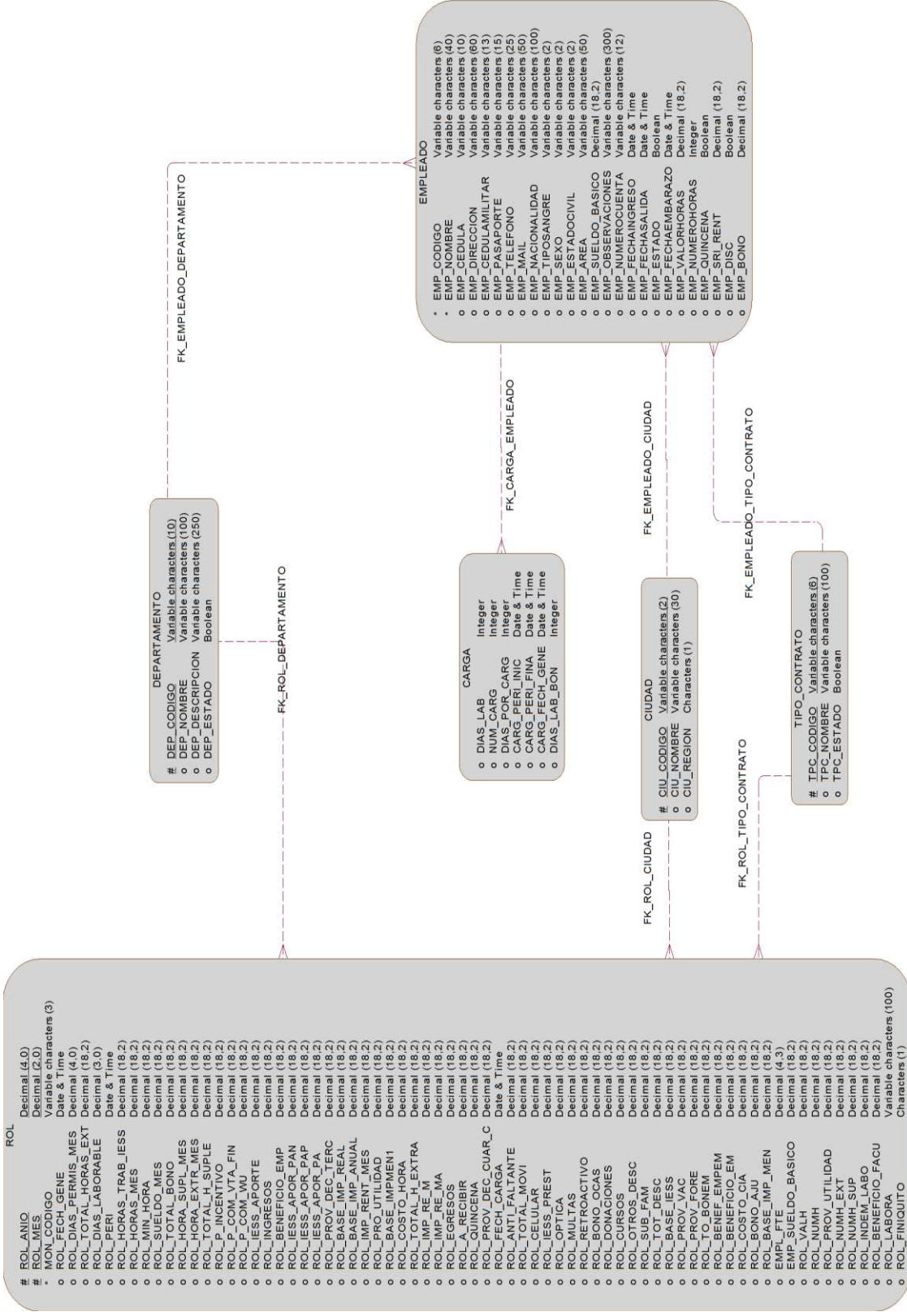
2.2.3 DISEÑO DEL MODELO LOGICO E/R.

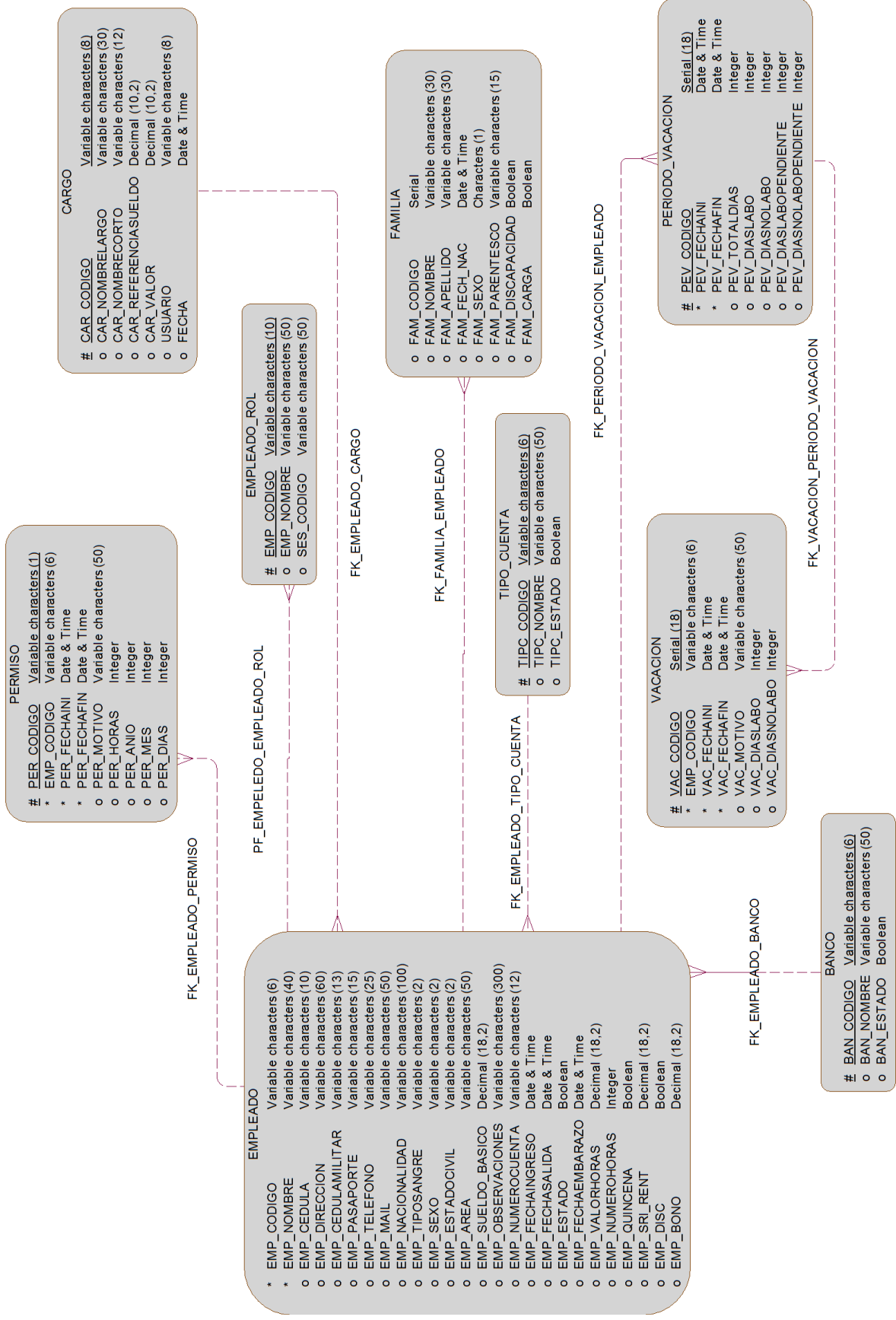
Por ser un diseño orientado a objetos, se presenta a continuación el diagrama entidad – relación, y las tablas relacionadas entre sí, las mismas que se construyen en base al diagrama de clases.

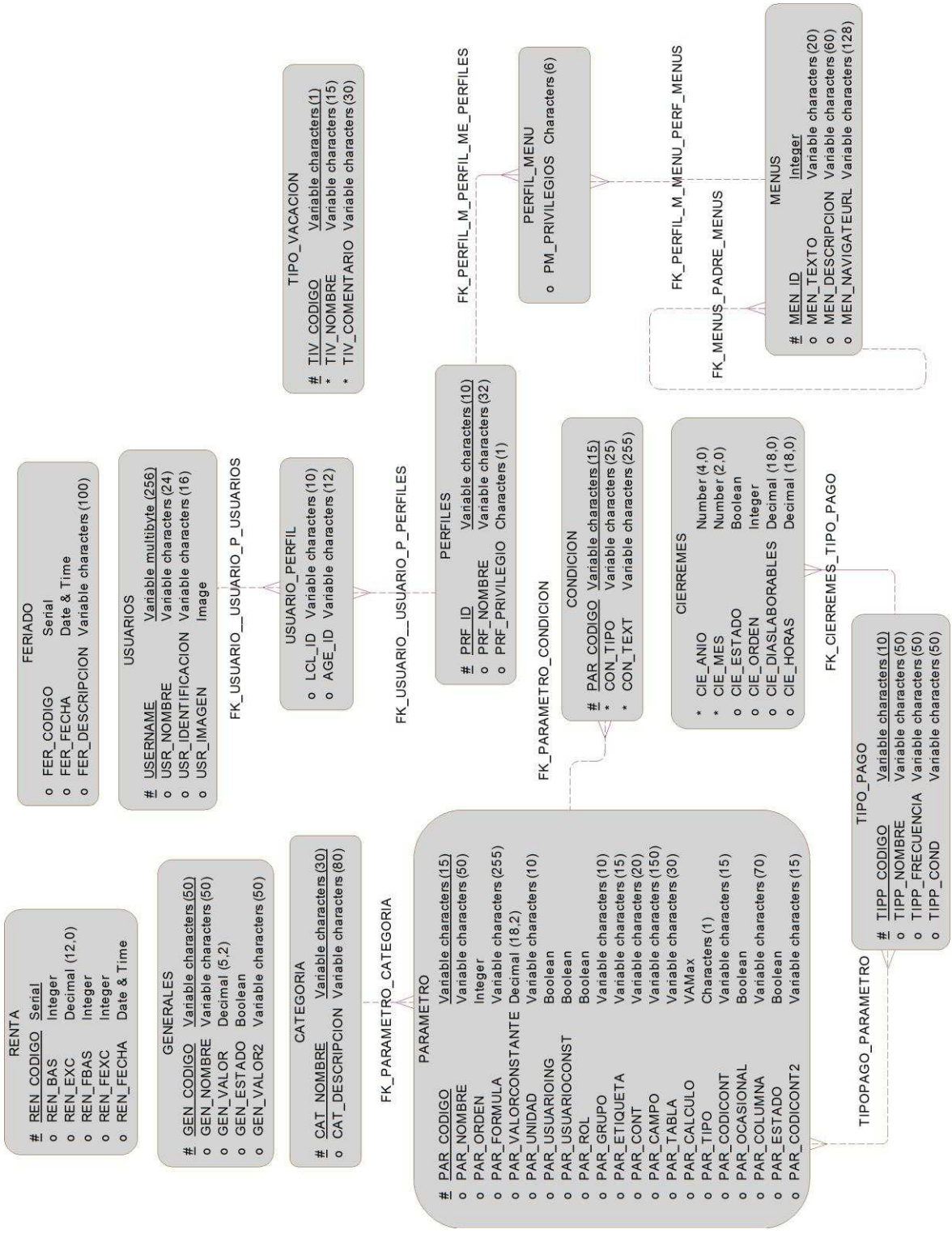
Clase	Tabla
ADICIONAL	ADICIONAL
BANCO	BANCO
CARGA	CARGA
CARGO	CARGO
CATEGORIA	CATEGORIA
CIERREMES	CIERREMES
CIERREMES_TIPOPAGO	CIERREMES_TIPOPAGO
CIUDAD	CIUDAD
CONDICION	CONDICION
DEPARTAMENTO	DEPARTAMENTO
EMPLEADO	EMPLEADO
EMPLEADO_ROL	EMPLEADO_ROL
EMPRESA	EMPRESA
FAMILIA	FAMILIA
FERIADO	FERIADO
FINIQUITO	FINIQUITO
GASTOS_SRI	GASTOS_SRI
GENERALES	GENERALES
HORA_EXTRA	HORA_EXTRA
LOGS	LOGS
MENUS	MENUS
PAIS	PAIS
PARAMETRO	PARAMETRO
PERFIL_MENU	PERFIL_MENU
PERFILES	PERFILES
PERIODO_VACACION	PERIODO_VACACION
PERMISO	PERMISO
RENTA	RENTA
ROL	ROL
TIPO_CONTRATO	TIPO_CONTRATO
TIPO_CUENTA	TIPO_CUENTA
TIPO_PAGO	TIPO_PAGO
TIPO_VACACION	TIPO_VACACION
TIPOPAGO_PARAMETRO	TIPOPAGO_PARAMETRO
TIPOS_LOG	TIPOS_LOG
USUARIO_PERFIL	USUARIO_PERFIL
USUARIOS	USUARIOS
UTILIDAD	UTILIDAD
VACACION	VACACION

Tabla 2 - 32: Relación diagrama de clases – Modelo E/R

DISEÑO DEL MODELO LOGICO E/R.







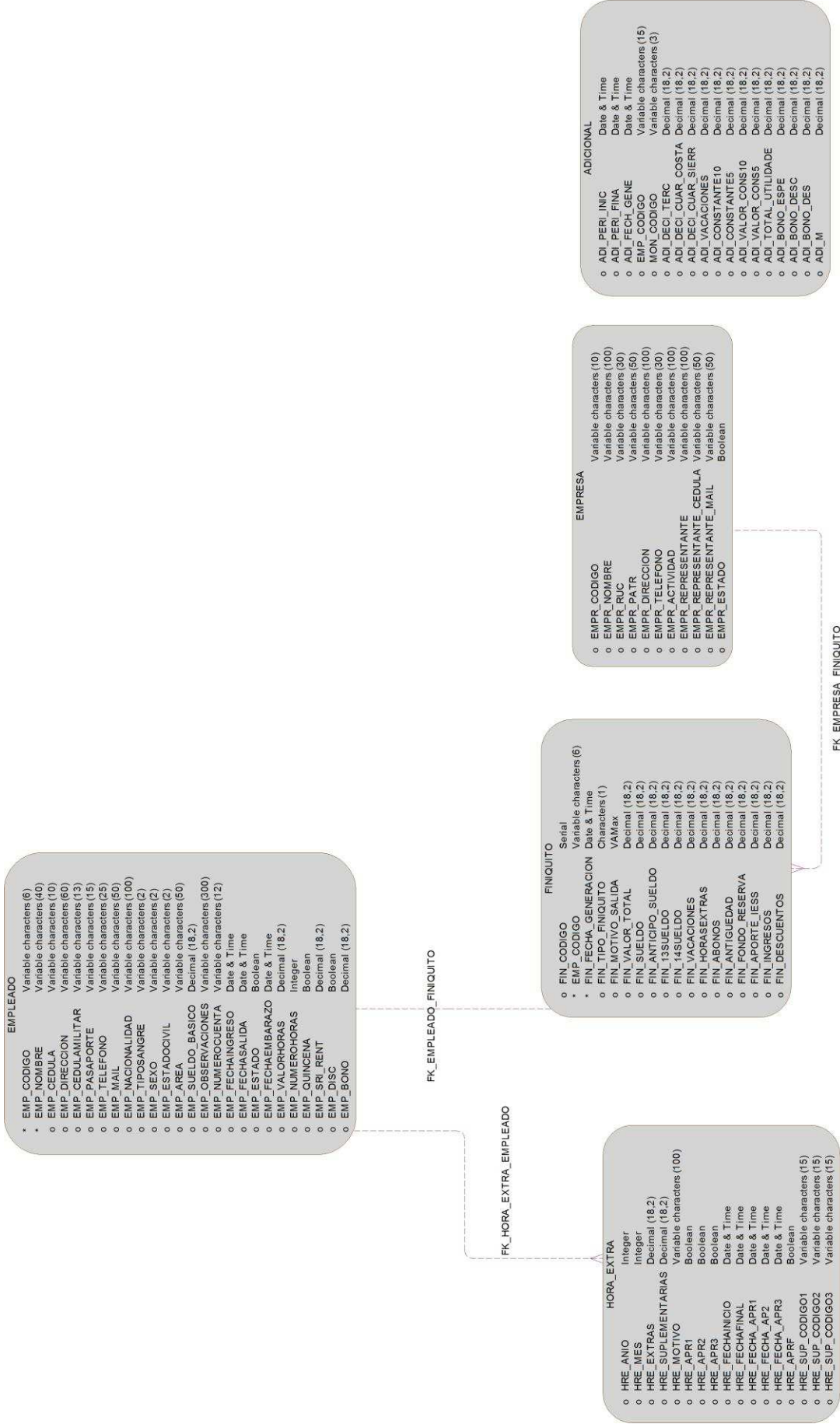
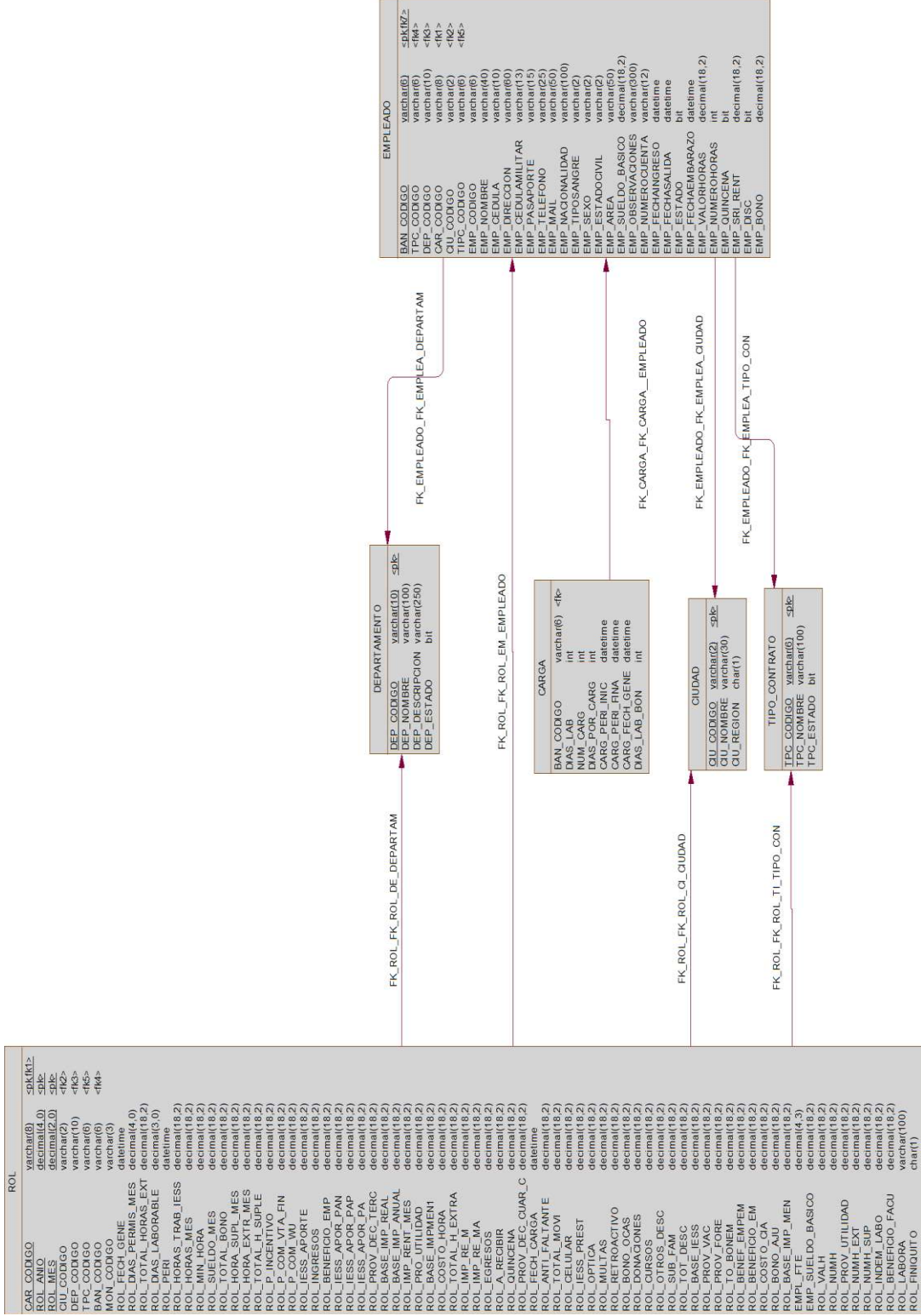
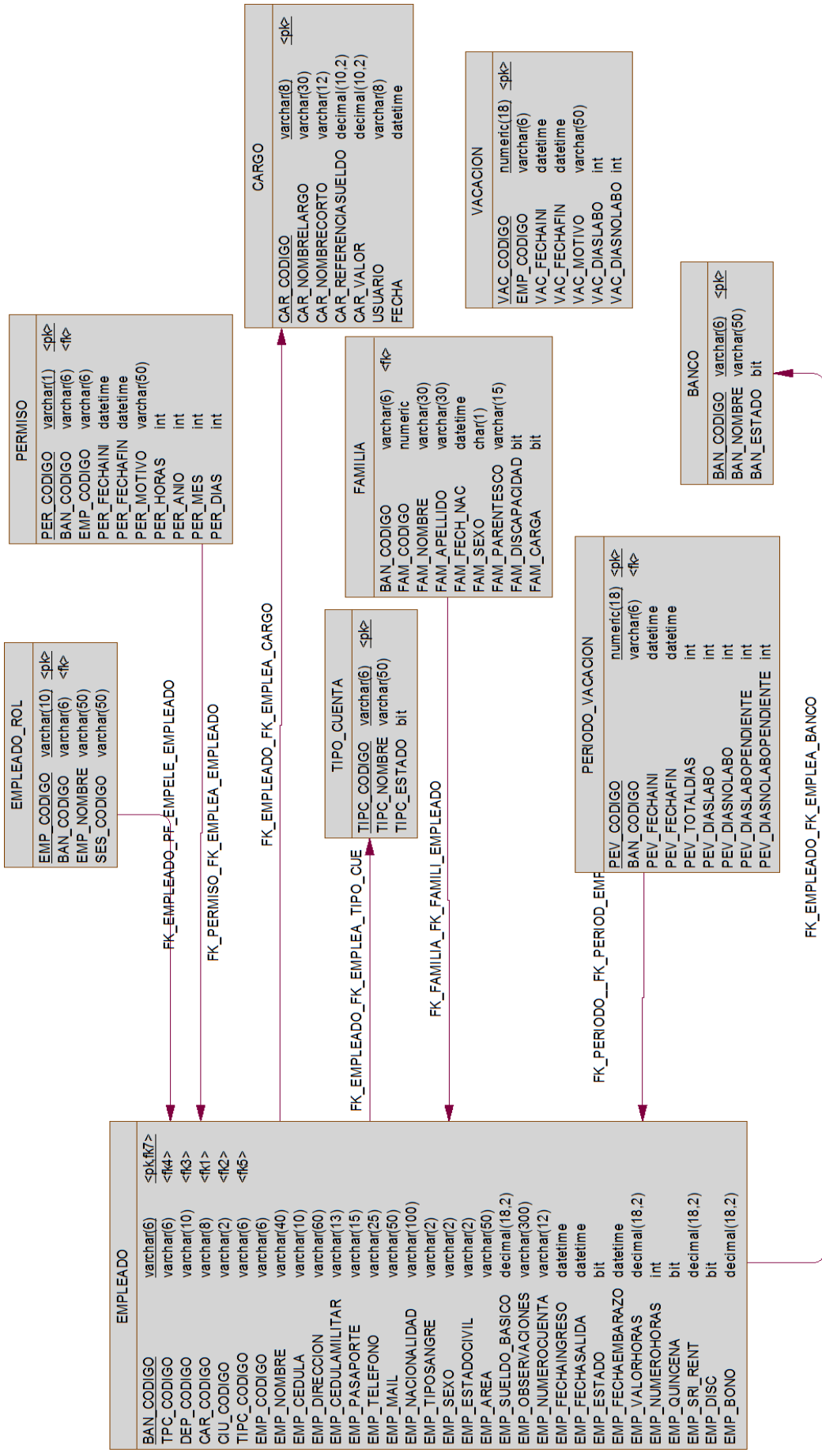


Figura 2 - 45: Modelo Lógico Entidad / Relación

2.2.4 DISEÑO DEL MODELO E/R FISICO.





EMPLEADO	
BAN_CODIGO	varchar(6)
TPC_CODIGO	varchar(6) <fk4>
DEP_CODIGO	varchar(10) <fk3>
CAR_CODIGO	varchar(8) <fk1>
CIU_CODIGO	varchar(2) <fk2>
TIPC_CODIGO	varchar(6) <fk5>
EMP_CODIGO	varchar(6)
EMP_NOMBRE	varchar(40)
EMP_CEDULA	varchar(10)
EMP_DIRECCION	varchar(60)
EMP_CEDULAMILITAR	varchar(13)
EMP_PASAPORTE	varchar(15)
EMP_TELEFONO	varchar(25)
EMP_MAIL	varchar(50)
EMP_NACIONALIDAD	varchar(100)
EMP_TIPOSANGRE	varchar(2)
EMP_SEXO	varchar(2)
EMP_ESTADOCIVIL	varchar(2)
EMP_AREA	varchar(50)
EMP_SUELDO_BASICO	decimal(18,2)
EMP_OBSERVACIONES	varchar(300)
EMP_NUMEROCUENTA	varchar(12)
EMP_FECHAINGRESO	datetime
EMP_FECHASALIDA	datetime
EMP_ESTADO	bit
EMP_FECHAEMBARAZO	datetime
EMP_VALORHORAS	decimal(18,2)
EMP_NUMEROHORAS	int
EMP_QUINCENA	bit
EMP_SRI_RENT	decimal(18,2)
EMP_DISC	bit
EMP_BONO	decimal(18,2)

FK_HORA_EXT_FK_HORA_E_EMPLEADO

FK_EMPLEADO_FK_EMPLEA_FINIQUIT

HORA_EXTRA	
BAN_CODIGO	varchar(6) <fk>
HRE_ANIO	int
HRE_MES	int
HRE_EXTRAS	decimal(18,2)
HRE_SUPLEMENTARIAS	decimal(18,2)
HRE_MOTIVO	varchar(100)
HRE_APR1	bit
HRE_APR2	bit
HRE_APR3	bit
HRE_FECHAINICIO	datetime
HRE_FECHAFINAL	datetime
HRE_FECHA_APR1	datetime
HRE_FECHA_APR2	datetime
HRE_FECHA_APR3	datetime
HRE_APRF	bit
HRE_SUP_CODIGO01	varchar(15)
HRE_SUP_CODIGO02	varchar(15)
HRE_SUP_CODIGO03	varchar(15)

FINIQUITO	
FIN_CODIGO	numeric
EMP_CODIGO	varchar(6)
FIN_FECHA_GENERACION	datetime
FIN_TIPO_FINIQUITO	char(1)
FIN_MOTIVO_SALIDA	VAMax
FIN_VALOR_TOTAL	decimal(18,2)
FIN_SUELDO	decimal(18,2)
FIN_ANTICIPO_SUELDO	decimal(18,2)
FIN_13SUELDO	decimal(18,2)
FIN_14SUELDO	decimal(18,2)
FIN_VACACIONES	decimal(18,2)
FIN_HORASEXTRAS	decimal(18,2)
FIN_ABONOS	decimal(18,2)
FIN_ANTIQUEDAD	decimal(18,2)
FIN_FONDO_RESERVA	decimal(18,2)
FIN_APORTE_RESS	decimal(18,2)
FIN_INGRESOS	decimal(18,2)
FIN_DESCUENTOS	decimal(18,2)

FK_FINIQUIT_FK_EMPRES_EMPRESA

EMPRESA	
EMPR_CODIGO	varchar(10)
EMPR_NOMBRE	varchar(100)
EMPR_RUC	varchar(30)
EMPR_PATR	varchar(50)
EMPR_DIRECCION	varchar(100)
EMPR_TELEFONO	varchar(30)
EMPR_ACTIVIDAD	varchar(100)
EMPR_REPRESENTANTE	varchar(100)
EMPR_REPRESENTANTE_CEDULA	varchar(50)
EMPR_REPRESENTANTE_MAIL	varchar(50)
EMPR_ESTADO	bit

ADICIONAL	
ADL_PERL_INIC	datetime
ADL_PERL_FINA	datetime
ADL_FECH_GENE	datetime
EMP_CODIGO	varchar(15)
MON_CODIGO	varchar(3)
ADL_DECI_TERC	decimal(18,2)
ADL_DECI_CUAR_COSTA	decimal(18,2)
ADL_DECI_CUAR_SIERR	decimal(18,2)
ADL_VACACIONES	decimal(18,2)
ADL_CONSTANTE10	decimal(18,2)
ADL_CONSTANTE5	decimal(18,2)
ADL_VALOR_CONS10	decimal(18,2)
ADL_VALOR_CONS5	decimal(18,2)
ADL_TOTAL_UTILIDADE	decimal(18,2)
ADL_BONO_ESPE	decimal(18,2)
ADL_BONO_DESC	decimal(18,2)
ADL_BONO_DES	decimal(18,2)
ADL_M	decimal(18,2)

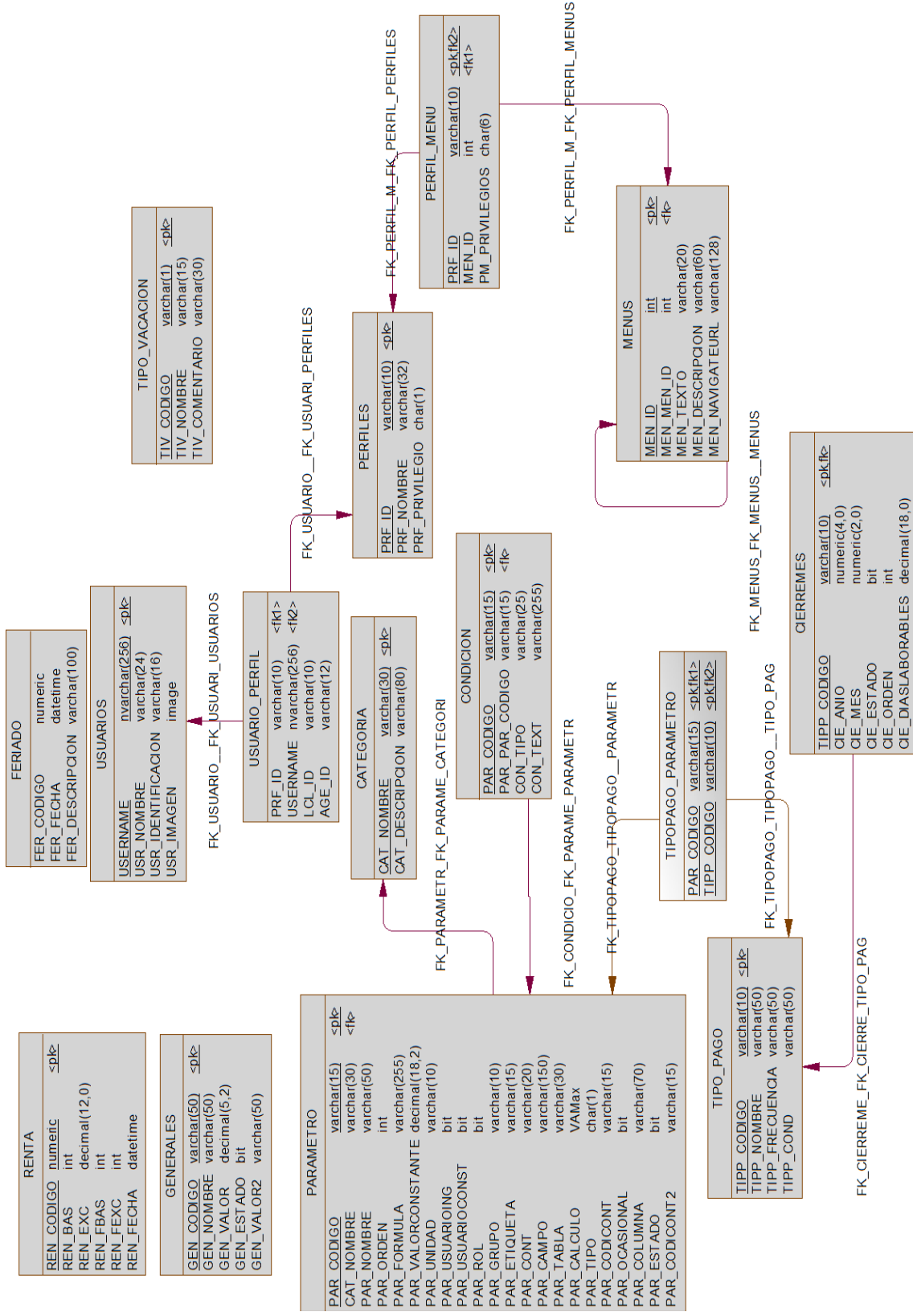


Figura 2 - 46: Modelo Físico Entidad / Relación

2.2.5 DISEÑO DE COMPONENTES.

Mediante los siguientes diagramas se verá la organización de los componentes (interfases, ddl's, módulos, paquetes, archivos, código fuente, etc.) del sistema y su dependencia con diferentes partes del mismo.

Cada módulo representa una parte del sistema.

2.2.3.1 Módulos del Sistema.

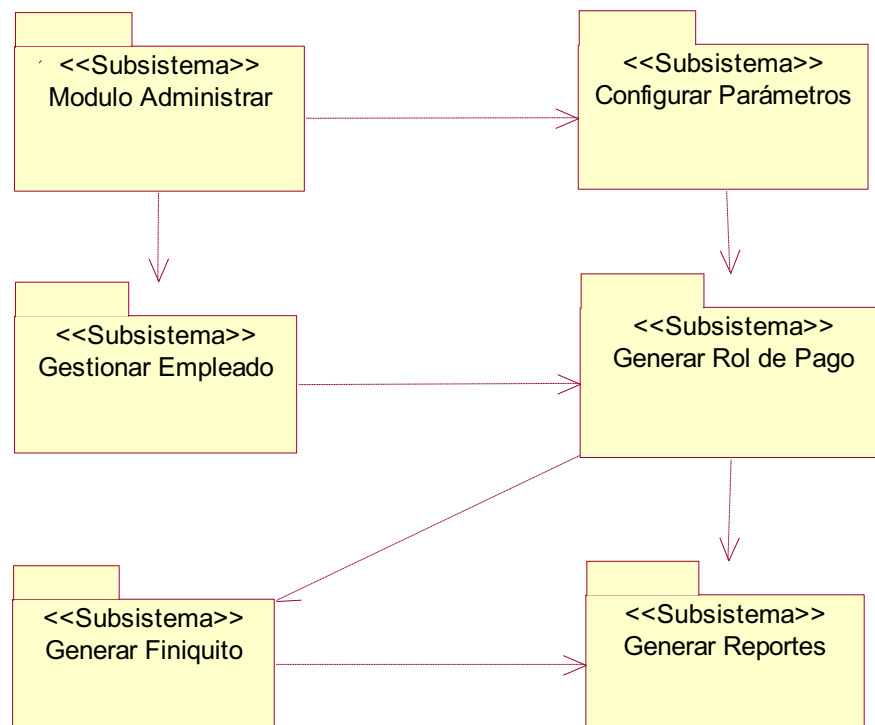


Figura 2 - 47: Diagrama de Componentes: Módulos del Sistema

2.2.3.2 Módulo Administrar Accesos.

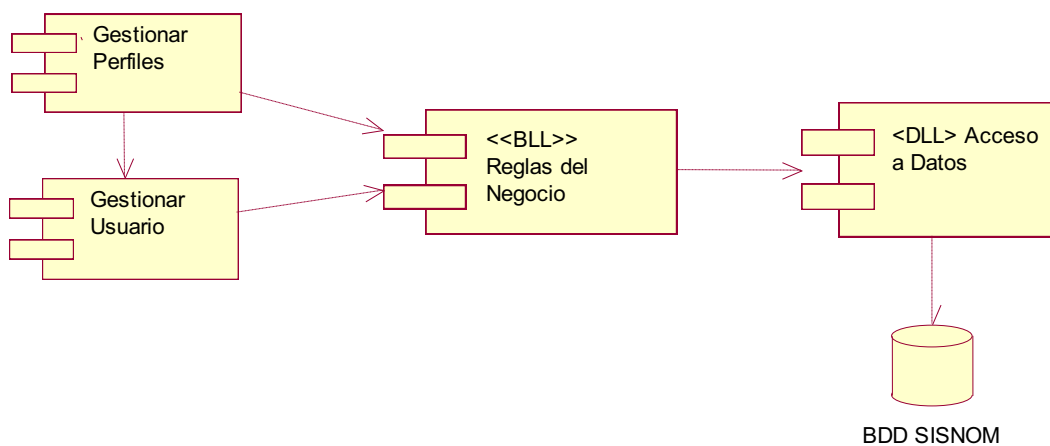


Figura 2 - 48: Diagrama de Componentes: Módulo Administrar Accesos

2.2.3.3 Módulo Configurar Parámetros.

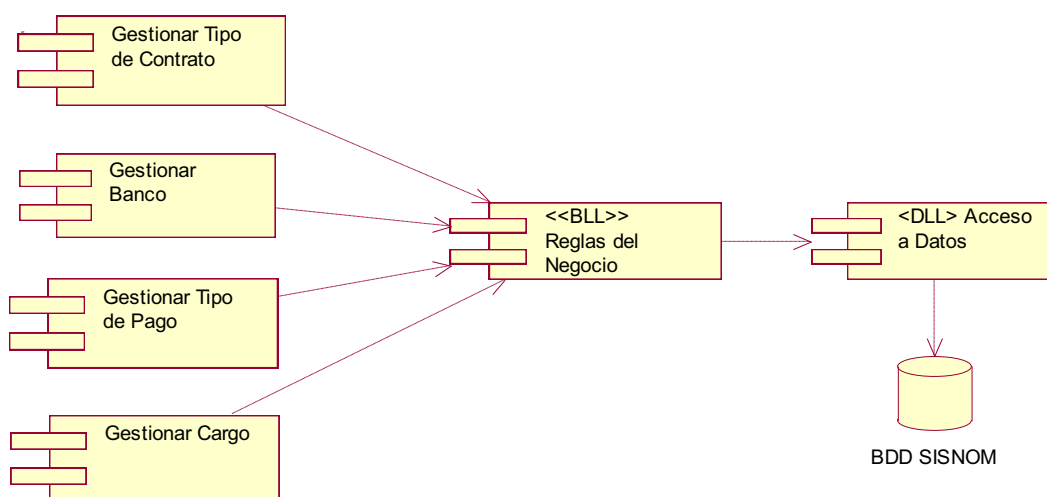


Figura 2 - 49: Diagrama de Componentes: Módulo Configurar Parámetros

2.2.3.4 Módulo Gestionar Empleado.

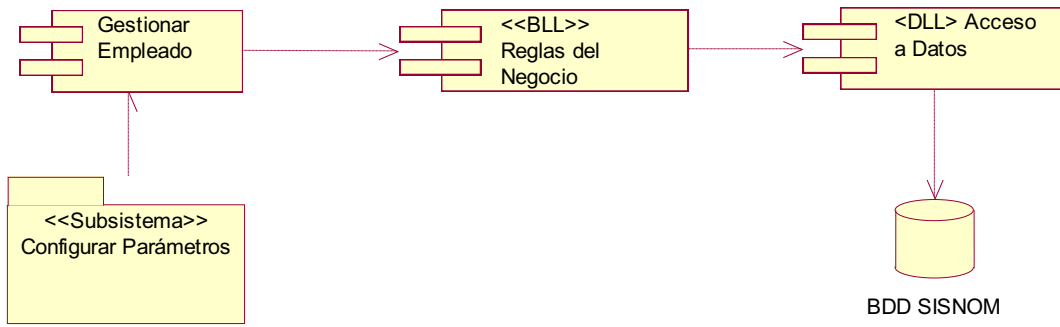


Figura 2 - 50: Diagrama de Componentes: Módulo Gestionar Empleado

2.2.3.5 Módulo Generar Rol de Pago.

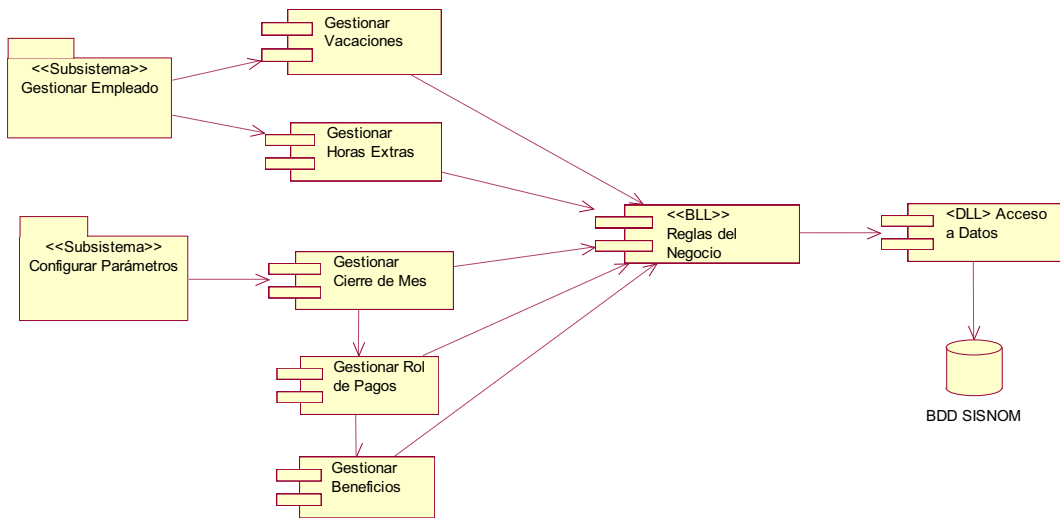


Figura 2 - 51: Diagrama de Componentes: Módulo Generar Rol de Pago

2.2.3.6 Módulo Generar Finiquito.

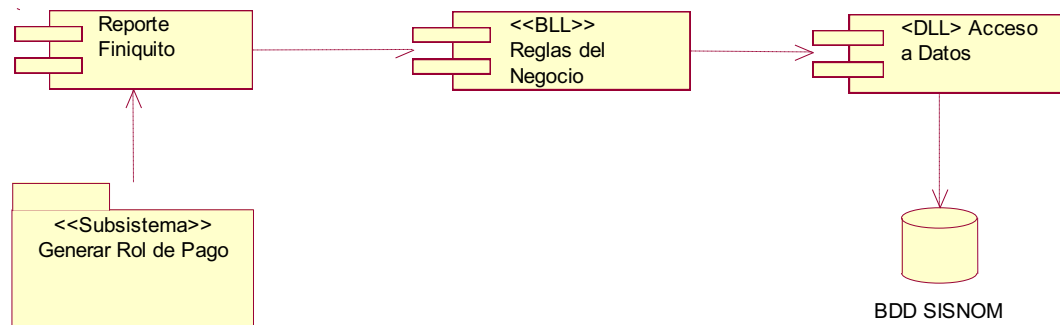


Figura 2 - 52: Diagrama de Componentes: Módulo Generar Finiquito

2.2.3.7 Módulo Generar Reportes.

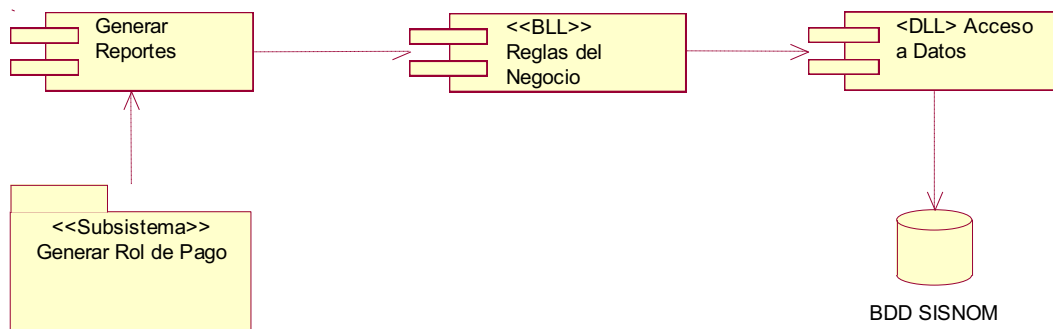


Figura 2 - 53: Diagrama de Componentes: Módulo Generar Reportes

2.2.6 DISEÑO DE LA ARQUITECTURA.

Para la implementación del Sistema de Administración de Nómina para una PYME (SANPYME) se va a utilizar una **Arquitectura Web**.

2.2.3.1 Arquitectura Web.

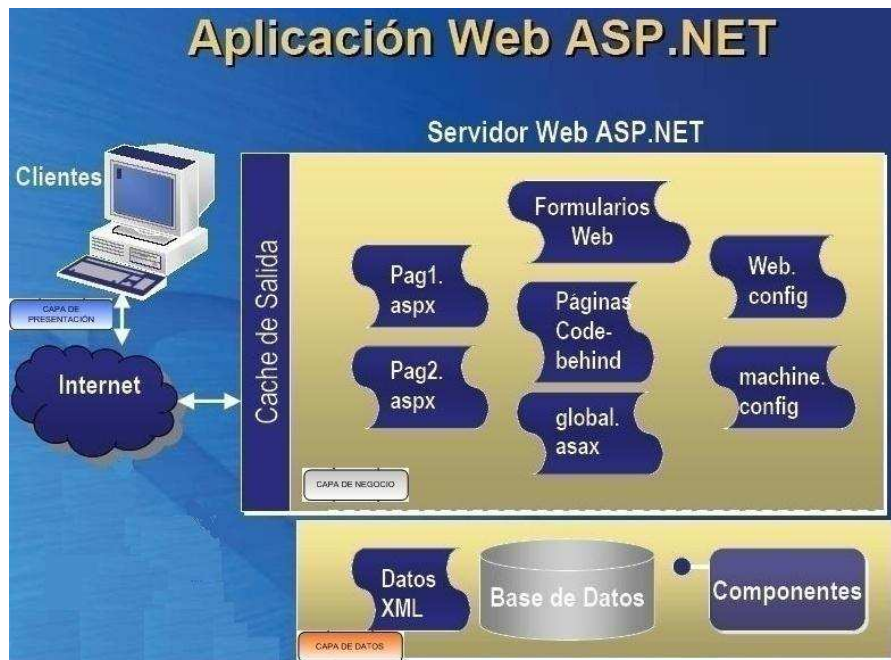


Figura 2 - 54: Diseño de la arquitectura: Arquitectura Web^[13]

[13] Arquitectura Web

- **Capa de Presentación:** Es la capa que presenta la interfaz del sistema al usuario e interactúa con el mismo, permitiéndole ingresar datos y recibir datos.
- **Capa de Negocio:** Es la capa que realiza los procesos una vez ingresado los datos necesarios, utilizando reglas empresariales o del negocio ya implementadas. Las reglas del negocio implementan la lógica empresarial de la aplicación.
- **Capa de Datos:** Es la capa que realiza el acceso al Origen de Datos que requiere la capa de negocio, es capa independiente de componentes lógicos de acceso a datos, ya que de este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.

2.2.7 DISEÑO DE LA JERARQUIA DE MENU DEL SISTEMA.

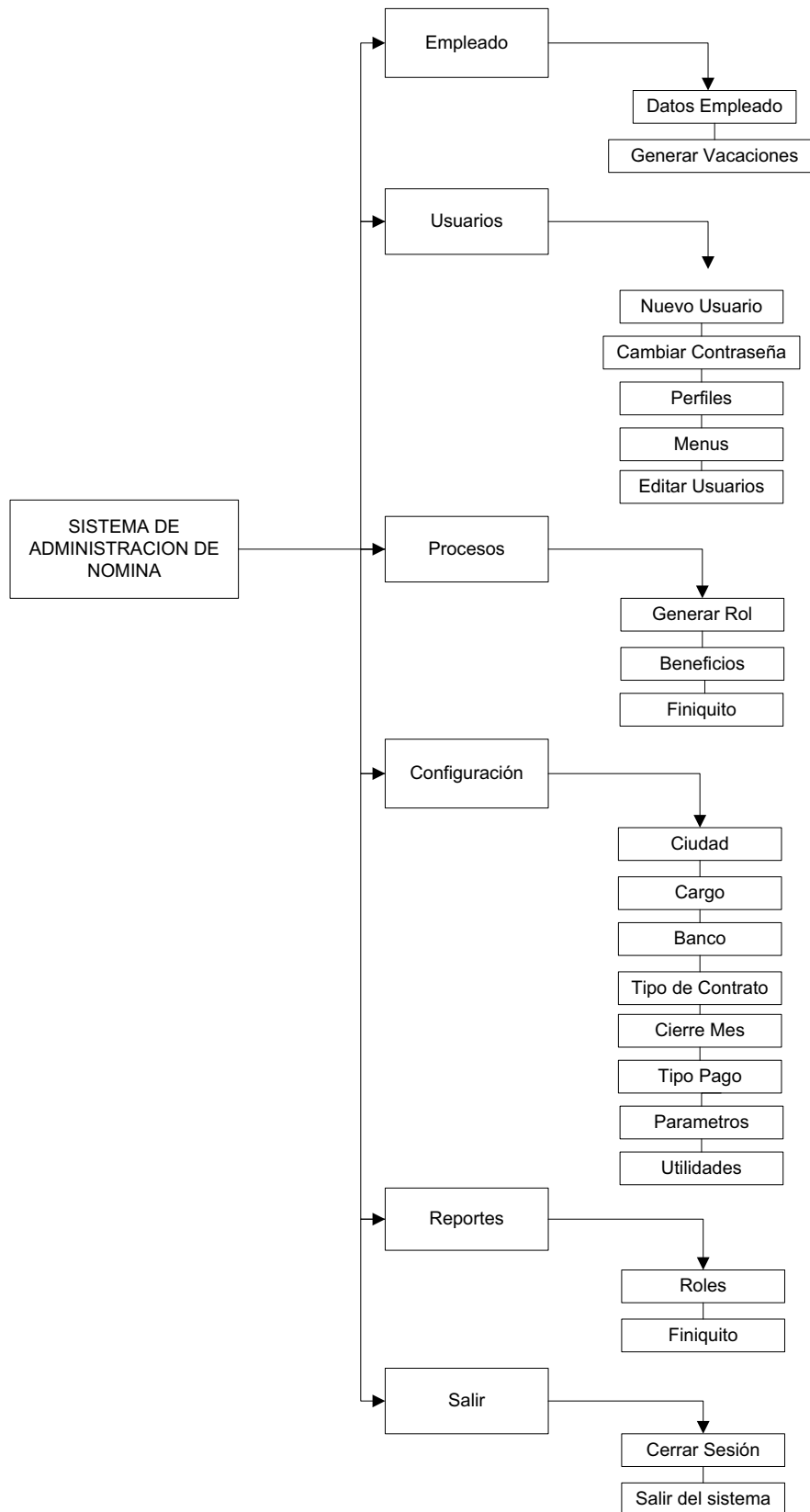


Figura 2 - 55: Diseño de la Jerarquía del Menú del Sistema

2.3 IMPLEMENTACIÓN DEL SISTEMA.

2.3.1 ESTANDARES DE PROGRAMACION.

Corresponde ahora especificar los estándares a usar para: nominar clases, nominar atributos de una clase, nominar operaciones de una clase, diseño de pantallas, diseño de botones y codificación.

Es importante tener un estándar de programación, con el objetivo de tener un desarrollo de fácil mantenimiento, escalable y consistente de datos.

A continuación se definen los estándares de programación:

2.3.3.1 Estándar para nombrar clases.

Para nombrar las clases se seguirán los siguientes criterios:

- ◆ Se utilizará sustantivos o palabras que la describan completamente.
- ◆ Se usará letras minúsculas.
- ◆ Si se necesitará de dos sustantivos para describir mejor la funcionalidad de la clase, se usará los dos sustantivos unidos.
- ◆ Los nombres de las clases estarán precedidas por 2 letras las cuales identificarán la clases y como separador el guión bajo (_) y luego por 3 letras identificando la capa del modelo, las cuales son:

Tipo de Clase	Identificador
Clase de Datos	cl_dal
Clase de Lógica	cl_bll

Tabla 2 - 33: Estándar para nombrar clases

Ejemplo:

- cl_dal_Banco.cs
- cl_bll_Banco.cs

2.3.3.2 Estándar para declaración de controles.

Para nombrar o declarar controles, se seguirá los criterios detallados anteriormente con la diferencia de que estarán precedidos por 3 letras que identificarán al control usado.

Los tipos de controles más usados se describen a continuación.

Control	Prefijo	Prefijo+Nombre
Form	frm	frm_Nombre
Label	lbl	lblNombre
TextBox	txt	txtNombre
GridView	gvi	gviNombre
Button	btn	btnNombre
ImageButton	imb	imbNombre
DropDownList	ddl	ddlNombre
ListBox	lst	lstNombre
DataList	dtl	dtlNombre
Checkbox	chk	chkNombre
CheckBoxList	cbl	cblNombre
RadioButton	rdo	rdoNombre
RadioButtonList	rbl	rblNombre
Image	img	imgNombre
Panel	pnl	pnlNombre
Table	tbl	tblNombre
TableLayoutPanel	tblp	tblpNombre
Validators	val	valNombre

Tabla 2 - 34: Abreviaturas para declaración de controles

2.3.3.3 Estándar para declaración de variables.

Para las variables de los métodos se utilizará:

Camel: El primer carácter de todas las palabras, excepto el de la primera palabra son mayúsculas y los otros caracteres son minúsculas.

Ejemplo:

Variable
nombreVariable

Tabla 2 - 35: Estándar para nombrar variables

2.3.3.4 Estándar para la base de datos.

- **Tabla:** Para la tabla se utilizara el nombre de la entidad todo en mayúscula
- **Columnas:** Para las columnas se utilizara las tres primeras letras de la tabla, seguido de un guión bajo (_) luego un nombre descriptivo del campo; todo esto en mayúsculas.
- **Procedimiento Almacenado (PA o SP):** Para el procedimiento almacenado se utilizará la abreviación seguido de un guión bajo y a continuación el nombre del procedimiento todo en mayúscula.
- **Funciones (FN):** Para las funciones se utilizará la abreviación seguido de un guión bajo y a continuación el nombre de la función.

Tipo	Ejemplo
Tabla	EMPLEADO
Columna	EMP_CODIGO
Procedimiento Almacenado	pa_EmpleadoGet
Función	fn_dielasemana

Tabla 2 - 36: Estándar para la base de datos

2.3.3.5 Estándar para diseñar pantallas.

Usaremos una serie de criterios generales que permitan estandarizar el diseño de pantallas del módulo, entre estos tenemos:

- ◆ Color.
- ◆ Brillo.
- ◆ Style.
- ◆ Gráficos.

Criterio	Características
Color	Se usará un color amarillo para el fondo de pantalla.
Brillo	No presenta ningún tipo de brillo.
Style	Se utilizarán hojas de estilo para todos los controles y objetos de diseño.
Gráficos	Se utilizarán gráficos para creación, eliminación y modificación de datos.

Tabla 2 - 37: Criterios para diseñar pantallas

2.3.2 REQUERIMIENTOS DE HARDWARE.

Las características mínimas del hardware que se necesita son las siguientes:

Servidor

Tipo	Componente	Característica
Hardware	Disco Duro	60 GB
	Memoria RAM	2 GB
	Procesador	Intel Pentium 4 de 2.3 Ghz
	Tarjeta de red	10/100
Software	Sistema Operativo	Windows 2003 server, Windows XP
	Base de Datos	Microsoft SQL Server Express 2005
	Otros	Framework 2.0
		Windows Installer 3_1

Tabla 2 - 38: Requerimientos mínimos de hardware del servidor

Cliente

Tipo	Componente	Característica
Hardware	Disco Duro	20 GB
	Memoria RAM	512 MB
	Procesador	Intel Celeron - 1.5 MHz
	Tarjeta de red	10/100
Software	Sistema Operativo	Windows XP
	Otros	Framework 2.0

Tabla 2 - 39: Requerimientos mínimos de hardware del cliente

CAPITULO III.

APLICACIÓN A UN CASO DE ESTUDIO

3.1 PREPARACIÓN DEL CASO DE ESTUDIO.

El caso de estudio se refiere a la implantación del sistema de administración de nómina para una PYME en la empresa Beehunting S.A. bajo las siguientes condiciones:

- Se realizó la instalación del sistema en un servidor con las siguientes características:

Tipo	Componente	Característica
Hardware	Disco Duro	180 GB
	Memoria RAM	4 GB
	Procesador	Intel Core 2 Duo de 3 Ghz
	Tarjeta de red	10/100
Software	Sistema Operativo	Windows 2003 server
	Base de Datos	Microsoft SQL Server Express 2005
	Otros	Framework 2.0
		Windows Installer 3_1

Tabla 2 - 40: Caso de estudio: Instalación del servidor

- Se realizó pruebas de conexión desde las máquinas clientes hacia el servidor; verificando el correcto funcionamiento del mismo.
- Se realizó la respectiva capacitación a cada uno de los usuarios que van a utilizar el sistema; con la entrega respectiva de todos los manuales correspondientes.

3.2 IMPLEMENTACION Y PRUEBAS DEL CASO DE ESTUDIO.

Las pruebas son un elemento esencial en el ciclo de desarrollo de software y consiste en una revisión final de los requerimientos, análisis, diseño y codificación.

El objetivo de las pruebas es encontrar fallas o errores para luego hacer una depuración del módulo y así asegurar que el producto ha sido desarrollado de acuerdo a los requerimientos y que los errores de la implementación han sido detectados.

3.2.1 CASOS DE PRUEBA.

Los casos de uso se consideran la guía para todo el proceso de desarrollo de software, por tanto en esta fase también serán utilizados como punto de partida.

En algunos casos de uso intervienen varios componentes, entonces los casos de prueba permitirán probar tanto la funcionalidad del módulo como la integración de los componentes.

3.2.3.1 Prueba del caso de uso: Gestionar Perfiles.

Prueba de Unidad: Gestionar Perfiles	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Administrar	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de perfiles y asignaciones de permisos de menús de un usuario.	
Datos de Ingreso: Datos perfiles y asignación de menús.	
Procedimiento de prueba:	
- Ingresar Nuevo Perfil:	

<ul style="list-style-type: none"> • Elegir la opción Nuevo. • Ingresar todos los datos en los campos del formulario. • Clic en el botón Guardar. <p>- Editar Perfil:</p> <ul style="list-style-type: none"> • Seleccionar el registro del perfil a ser modificado. • Clic en el botón Editar. • Modificar los datos necesarios. • Clic en el botón Actualizar. <p>- Eliminar Perfil:</p> <ul style="list-style-type: none"> • Seleccionar el registro del perfil a ser eliminado del sistema. • Clic en el botón Eliminar. • Clic en el botón Aceptar para confirmar la eliminación. <p>- Asignar Menús al Perfil:</p> <ul style="list-style-type: none"> • Seleccionar el registro del perfil. • Seleccionar el registro del menú. • Clic en el botón Agregar. •
<p>Resultados:</p> <p>- Ingresar Nuevo Perfil:</p> <ul style="list-style-type: none"> • Se verifica en la lista de perfiles que el nuevo registro ha sido creado exitosamente. <p>- Editar Perfil:</p> <ul style="list-style-type: none"> • Se verifica en la lista de perfiles que la información del registro se ha modificado exitosamente. <p>- Eliminar Perfil:</p> <ul style="list-style-type: none"> • Se verifica en la lista de perfiles que el registro ya no se muestra por tanto ha sido eliminado exitosamente. <p>- Asignar Menús al Perfil:</p> <ul style="list-style-type: none"> • Se verifica cuando se ingresa con el perfil se agregado los menús asignados.

Tabla 3 - 1: Descripción del caso de prueba CU:01.- Gestionar Perfiles

3.2.3.2 Prueba del caso de uso: Gestionar Usuarios

Prueba de Unidad: Gestionar Usuarios	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Administrar	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de	

usuarios del sistema.
Datos de Ingreso: Datos personales de usuario.
<p>Procedimiento de prueba:</p> <p>- Ingresar Nuevo Usuario:</p> <ul style="list-style-type: none"> • Elegir la opción Nuevo. • Elegir el perfil de usuario. • Ingresar todos los datos del usuario en los campos del formulario. • Clic en el botón Guardar. <p>- Editar Usuario:</p> <ul style="list-style-type: none"> • Seleccionar la pantalla del perfil a modificarse en el sistema. • Clic en el botón Editar. • Modificar los datos necesarios. • Clic en el botón Actualizar. <p>- Cambiar Contraseña:</p> <ul style="list-style-type: none"> • Seleccionar el registro del usuario. • Clic en el botón Cambiar Clave. • Digitar la nueva clave. • Clic en el botón Aceptar.
<p>Resultados:</p> <p>- Ingresar Nuevo Usuario:</p> <ul style="list-style-type: none"> • Se verifica en la lista de usuarios que el nuevo registro ha sido creado exitosamente. <p>- Editar Usuario:</p> <ul style="list-style-type: none"> • Se verifica en la lista de usuarios que la información del registro se ha modificado exitosamente. <p>- Cambiar Contraseña:</p> <ul style="list-style-type: none"> • Se verifica al ingresar nuevamente al sistema que se ha cambiado la contraseña exitosamente.

Tabla 3 - 2: Descripción del caso de prueba CU:02.- Gestionar Usuarios

3.2.3.3 Prueba del caso de uso: Gestionar Tipo de Contrato

Prueba de Unidad: Gestionar Tipo de Contrato	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Configurar Parámetros	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de	

tipos de contratos en el sistema.
Datos de Ingreso: Datos de tipos de contratos.
<p>Procedimiento de prueba:</p> <p>- Ingresar Nuevo Tipo de Contrato:</p> <ul style="list-style-type: none"> • Elegir la opción Nuevo. • Ingresar todos los datos del tipo de contrato en los campos del formulario. • Clic en el botón Guardar. <p>- Editar Tipo de Contrato:</p> <ul style="list-style-type: none"> • Seleccionar el registro del tipo de contrato a ser modificado en el sistema. • Clic en el botón Editar. • Modificar los datos necesarios. • Clic en el botón Actualizar. <p>- Eliminar Tipo de Contrato:</p> <ul style="list-style-type: none"> • Seleccionar el registro del tipo de contrato a ser eliminado en el sistema. • Clic en el botón Eliminar. • Clic en el botón Aceptar para confirmar la eliminación.
<p>Resultados:</p> <p>- Ingresar Nuevo Tipo de Contrato:</p> <ul style="list-style-type: none"> • Se verifica en la lista de tipos de contrato que el nuevo registro ha sido creado exitosamente. <p>- Editar Tipo de Contrato:</p> <ul style="list-style-type: none"> • Se verifica en la lista de tipos de contrato que la información del registro se ha modificado exitosamente. <p>- Eliminar Tipo de Contrato:</p> <ul style="list-style-type: none"> • Se verifica en la lista de tipos de contrato que el registro ya no se muestra por tanto ha sido eliminado exitosamente.

Tabla 3 - 3: Descripción del caso de prueba CU:03.- Gestionar Tipo de Contrato

3.2.3.4 Prueba del caso de uso: Gestionar Bancos

Prueba de Unidad: Gestionar Bancos	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Configurar Parámetros	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de bancos en el sistema.	

Datos de Ingreso: Datos de bancos.
<p>Procedimiento de prueba:</p> <p>- Ingresar Nuevo Tipo de Contrato:</p> <ul style="list-style-type: none"> • Elegir la opción Nuevo. • Ingresar todos los datos del banco en los campos del formulario. • Clic en el botón Guardar. <p>- Editar Tipo de Contrato:</p> <ul style="list-style-type: none"> • Seleccionar el registro del banco a ser modificado en el sistema. • Clic en el botón Editar. • Modificar los datos necesarios. • Clic en el botón Actualizar. <p>- Eliminar Tipo de Contrato:</p> <ul style="list-style-type: none"> • Seleccionar el registro del banco a ser eliminado en el sistema. • Clic en el botón Eliminar. • Clic en el botón Aceptar para confirmar la eliminación.
<p>Resultados:</p> <p>- Ingresar Nuevo Banco:</p> <ul style="list-style-type: none"> • Se verifica en la lista de bancos que el nuevo registro ha sido creado exitosamente. <p>- Editar Banco:</p> <ul style="list-style-type: none"> • Se verifica en la lista de bancos que la información del registro se ha modificado exitosamente. <p>- Eliminar Banco:</p> <ul style="list-style-type: none"> • Se verifica en la lista de bancos que el registro ya no se muestra por tanto ha sido eliminado exitosamente.

Tabla 3 - 4: Descripción del caso de prueba CU:04.- Gestionar Bancos

3.2.3.5 Prueba del caso de uso: Gestionar Tipo de Pago

Prueba de Unidad: Gestionar Tipo de Pago	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Configurar Parámetros	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de tipos de pago en el sistema.	
Datos de Ingreso: Datos de tipos de pago con sus parámetros de rol de pagos.	

Procedimiento de prueba:**- Ingresar Nuevo Tipo de Pago:**

- Elegir la opción Nuevo.
- Ingresar todos los datos del tipo de pago en los campos del formulario.
- Clic en el botón Guardar.

- Editar Tipo de Pago:

- Seleccionar el registro del tipo de pago a ser modificado en el sistema.
- Clic en el botón Editar.
- Modificar los datos necesarios.
- Clic en el botón Actualizar.

- Eliminar Tipo de Pago:

- Seleccionar el registro del tipo de pago a ser eliminado en el sistema.
- Clic en el botón Eliminar.
- Clic en el botón Aceptar para confirmar la eliminación.

- Modificar Parámetros:

- Seleccionar el registro del tipo de pago.
- Seleccionar los parámetros del rol de pagos.
- Agregamos los parámetros que afectan al tipo de pago.

Resultados:**- Ingresar Nuevo Tipo de Pago:**

- Se verifica en la lista de tipos de pago que el nuevo registro ha sido creado exitosamente.

- Editar Tipo de Pago:

- Se verifica en la lista de tipos de pago que la información del registro se ha modificado exitosamente.

- Eliminar Tipo de Pago:

- Se verifica en la lista de tipos de pago que el registro ya no se muestra por tanto ha sido eliminado exitosamente.

- Modificar Parámetros:

- Se verifica en la lista de parámetros seleccionados que los registros se agregan exitosamente.

Tabla 3 - 5: Descripción del caso de prueba CU:05.- Gestionar Tipo de Pago

3.2.3.6 Prueba del caso de uso: Gestionar Cargo

Prueba de Unidad: Gestionar Cargo	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	

Responsable: Esteban Castillo y Mauricio Oviedo
Ubicación: Módulo Configurar Parámetros
Objetivo: Comprobar el funcionamiento y desempeño de la administración de cargos en el sistema.
Datos de Ingreso: Datos de cargos de un empleado.
<p>Procedimiento de prueba:</p> <p>- Ingresar Nuevo Cargo:</p> <ul style="list-style-type: none"> • Elegir la opción Nuevo. • Ingresar todos los datos del cargo en los campos del formulario. • Clic en el botón Guardar. <p>- Editar Cargo:</p> <ul style="list-style-type: none"> • Seleccionar el registro del cargo a ser modificado en el sistema. • Clic en el botón Editar. • Modificar los datos necesarios. • Clic en el botón Actualizar. <p>- Eliminar Cargo:</p> <ul style="list-style-type: none"> • Seleccionar el registro del cargo a ser eliminado en el sistema. • Clic en el botón Eliminar. • Clic en el botón Aceptar para confirmar la eliminación.
<p>Resultados:</p> <p>- Ingresar Nuevo Cargo:</p> <ul style="list-style-type: none"> • Se verifica en la lista de cargos que el nuevo registro ha sido creado exitosamente. <p>- Editar Cargo:</p> <ul style="list-style-type: none"> • Se verifica en la lista de cargos que la información del registro se ha modificado exitosamente. <p>- Eliminar Cargo:</p> <ul style="list-style-type: none"> • Se verifica en la lista de cargos que el registro ya no se muestra por tanto ha sido eliminado exitosamente.

Tabla 3 - 6: Descripción del caso de prueba CU:06.- Gestionar Cargo

3.2.3.7 Prueba del caso de uso: Gestionar Empleado

Prueba de Unidad: Gestionar Empleado	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	

Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Gestionar Empleado	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de empleados en el sistema.	
Datos de Ingreso: Datos de un empleado, cargas familiares.	
Procedimiento de prueba:	
<p>- Ingresar Nuevo Empleado:</p> <ul style="list-style-type: none"> • Elegir la opción Nuevo. • Ingresar todos los datos del cargo en los campos del formulario. • Clic en el botón Guardar. <p>- Editar Empleado:</p> <ul style="list-style-type: none"> • Seleccionar el registro del empleado a ser modificado en el sistema. • Clic en el botón Editar. • Modificar los datos necesarios. • Clic en el botón Actualizar. <p>- Cambiar Contraseña:</p> <ul style="list-style-type: none"> • Seleccionar el registro del empleado. • Clic en el botón Cambiar Clave. • Digitar la nueva clave. • Clic en el botón Aceptar. 	
Resultados:	
<p>- Ingresar Nuevo Empleado:</p> <ul style="list-style-type: none"> • Se verifica en la lista de empleados que el nuevo registro ha sido creado exitosamente. <p>- Editar Empleado:</p> <ul style="list-style-type: none"> • Se verifica en la lista de empleados que la información del registro se ha modificado exitosamente. <p>- Cambiar Contraseña:</p> <ul style="list-style-type: none"> • Se verifica al ingresar nuevamente al sistema que se ha cambiado la contraseña exitosamente. 	

Tabla 3 - 7: Descripción del caso de prueba CU:07.- Gestionar Empleado

3.2.3.8 Prueba del caso de uso: Gestionar Cierre de Mes

Prueba de Unidad: Gestionar Cierre de Mes	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	

Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Generar Rol de Pagos	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de cierres de mes en el sistema con sus respectivas validaciones.	
Datos de Ingreso: Datos de cierre de mes.	
Procedimiento de prueba: <ul style="list-style-type: none"> - Ingresar Nuevo Cierre de Mes: <ul style="list-style-type: none"> • Elegir la opción Nuevo. • Ingresar todos los datos del cierre de mes en los campos del formulario. • Clic en el botón Guardar. - Editar Cierre de Mes: <ul style="list-style-type: none"> • Seleccionar el registro del cierre de mes a ser modificado en el sistema. • Clic en el botón Editar. • Modificar los datos necesarios. • Clic en el botón Actualizar. - Agregar Rubros: <ul style="list-style-type: none"> • Seleccionar el registro del cierre de mes. • Agregar rubros desde la lista de tipos de pago disponibles a la lista de tipos de pago seleccionado. • Clic en el botón Agregar. Clic en el botón Guardar.	
Resultados: <ul style="list-style-type: none"> - Ingresar Nuevo Cierre de Mes: <ul style="list-style-type: none"> • Se verifica en la lista de cierres de mes que el nuevo registro ha sido creado exitosamente. - Editar Cierre de Mes: <ul style="list-style-type: none"> • Se verifica en la lista de cierres de mes que la información del registro se ha modificado exitosamente. - Agregar Rubros: <ul style="list-style-type: none"> • Se verifica en la lista de tipos de pago seleccionados se encuentren los rubros a pagarse en ese mes. 	

Tabla 3 - 8: Descripción del caso de prueba CU:08.- Gestionar Cierre de Mes

3.2.3.9 Prueba del caso de uso: Gestionar Vacaciones

Prueba de Unidad: Gestionar Vacaciones	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	

Responsable: Esteban Castillo y Mauricio Oviedo
Ubicación: Módulo Generar Rol de Pagos
Objetivo: Comprobar el funcionamiento y desempeño de la administración de vacaciones de un empleado en el sistema.
Datos de Ingreso: Datos de vacaciones.
Procedimiento de prueba: - Ingresar Nueva Vacación: <ul style="list-style-type: none"> • Seleccionar al empleado. • Elegir la opción Nuevo. • Ingresar todos los datos de vacaciones en los campos del formulario. • Clic en el botón Guardar.
Resultados: - Ingresar Nueva Vacación: <ul style="list-style-type: none"> • Se verifica en la lista de vacaciones que el nuevo registro ha sido creado exitosamente.

Tabla 3 - 9: Descripción del caso de prueba CU:09.- Gestionar Vacaciones

3.2.3.10 Prueba del caso de uso: Gestionar Horas Extras

Prueba de Unidad: Gestionar Horas Extras	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Generar Rol de Pagos	
Objetivo: Comprobar el funcionamiento y desempeño de la administración de horas extras de un empleado en el sistema.	
Datos de Ingreso: Datos de horas extras.	
Procedimiento de prueba: - Ingresar Nueva Hora Extra: <ul style="list-style-type: none"> • Seleccionar al empleado. • Elegir la opción Nuevo. • Ingresar todos los datos de hora extras en los campos del formulario. • Clic en el botón Guardar. 	

<p>Eliminar Hora Extra:</p> <ul style="list-style-type: none"> • Seleccionar el registro de hora extra a ser eliminada en el sistema. • Clic en el botón Eliminar. • Clic en el botón Aceptar para confirmar la eliminación.
<p>Resultados:</p> <p>- Ingresar Nueva Hora Extra:</p> <ul style="list-style-type: none"> • Se verifica en la lista de horas extras que el nuevo registro ha sido creado exitosamente. <p>- Eliminar Hora Extra:</p> <ul style="list-style-type: none"> • Se verifica en la lista de extras que el registro ya no se muestra por tanto ha sido eliminado exitosamente.

Tabla 3 - 10: Descripción del caso de prueba CU:10.- Gestionar Horas Extras

3.2.3.11 Prueba del caso de uso: Gestionar Rol de Pago

Prueba de Unidad: Gestionar Rol de Pago	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Generar Rol de Pagos	
Objetivo: Comprobar el correcto funcionamiento de la generación del rol de pagos en el sistema.	
Datos de Ingreso: Datos del rol de pagos.	
<p>Procedimiento de prueba:</p> <p>- Generar Rol de Pagos:</p> <ul style="list-style-type: none"> • Seleccionar el mes y el año el cual se va generar el rol. • Clic en el botón Generar. 	
<p>Resultados:</p> <p>- Generar Rol de Pagos:</p> <ul style="list-style-type: none"> • Se genera el rol de pagos de todos los empleados de la empresa y lo verificamos en el reporte del rol respectivo de cada empleado. 	

Tabla 3 - 11: Descripción del caso de prueba CU:11.- Gestionar Horas Extras

3.2.3.12 Prueba del caso de uso: Gestionar Beneficios

Prueba de Unidad: Gestionar Beneficios	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Generar Rol de Pagos	
Objetivo: Comprobar el correcto funcionamiento de la generación beneficios de ley en el sistema.	
Datos de Ingreso: Datos del beneficios.	
Procedimiento de prueba:	
<p>- Generar Beneficios:</p> <ul style="list-style-type: none"> • Seleccionar el mes y el año el cual se va generar los beneficios. • Clic en el botón Generar. 	
Resultados:	
<p>- Generar Beneficios:</p> <ul style="list-style-type: none"> • Se genera los beneficios para todos los empleados de la empresa y lo verificamos en el reporte del rol respectivo de cada empleado. 	

Tabla 3 - 12: Descripción del caso de prueba CU:12.- Gestionar Beneficios

3.2.3.13 Prueba del caso de uso: Generar Finiquito

Prueba de Unidad: Generar Finiquito	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Generar Finiquito	
Objetivo: Comprobar el correcto funcionamiento de la generación automática del finiquito de un empleado en el sistema.	
Datos de Ingreso: Datos de finiquito.	
Procedimiento de prueba:	
<p>- Generar Finiquito:</p> <ul style="list-style-type: none"> • Seleccionar el empleado a generarse el finiquito. • Clic en el botón Generar. 	

<p>- Modificar Finiquito:</p> <ul style="list-style-type: none"> • Seleccionar el empleado a generarse el finiquito. • Clic en el botón Generar.
<p>Resultados:</p> <p>- Generar Finiquito:</p> <ul style="list-style-type: none"> • Se genera el finiquito del empleado y se verifica cuando se presentan los valores en el reporte de finiquitos de acuerdo al formato del Ministerio de Trabajo. <p>- Modificar Finiquito:</p> <ul style="list-style-type: none"> • Se genera nuevamente el finiquito del empleado calculando los valores a la nueva fecha de generación y se verifica cuando se presentan los valores en el reporte de finiquitos de acuerdo al formato del Ministerio de Trabajo.

Tabla 3 - 13: Descripción del caso de prueba CU:13.- Generar Finiquito

3.2.3.14 Prueba del caso de uso: Gestionar Reportes

Prueba de Unidad: Reporte Rol de Pago	
Proyecto:	Sistema de Administración de Nómina para una PYME
Fecha de realización: 30-01-2011	
Responsable: Esteban Castillo y Mauricio Oviedo	
Ubicación: Módulo Generar Reportes	
Objetivo: Comprobar la planilla de pago del empleado mediante un reporte	
Datos de Ingreso: Datos del empleado y del período de generación del rol de pagos.	
Procedimiento de prueba:	
<p>- Generar Reporte Rol de Pagos:</p> <ul style="list-style-type: none"> • Seleccionar el o los empleados y el periodo a generarse. • Clic en el botón Generar. • Imprimir Reporte <p>- Generar Reporte Empleados:</p> <ul style="list-style-type: none"> • Seleccionar los filtros de búsqueda. • Clic en el botón Generar. • Imprimir Reporte. 	
Resultados:	
<p>- Generar Reporte Rol de Pagos:</p> <ul style="list-style-type: none"> • Se genera una planilla de rol de pago con sus valores detallados para cada 	

- empleado.
- Generar Reporte Empleados:**
- Se genera el reporte de empleados.

Tabla 3 - 14: Descripción del caso de prueba CU:14.- Gestionar Reportes

3.2.2 PRUEBAS DE USABILIDAD.

3.2.1.1 Objetivos

Estas pruebas están orientadas a probar la usabilidad del sistema. Esto se refiere a probar la facilidad con la cual los usuarios de una aplicación la pueden operar.

En nuestro caso, los objetivos principales serán:

- Determinar si un usuario puede utilizar la aplicación completado satisfactoriamente los procesos.
- Determinar si la interfaz del usuario es lo suficientemente intuitiva tanto para usuarios que tienen experiencia en aplicaciones de software como para aquellos que no la tienen.
- Determinar si la aplicación requiere modificaciones para que cumpla los objetivos anteriores.

3.2.1.2 Usuarios de pruebas

Las pruebas de usabilidad se realizaron en 3 usuarios; el 1 usuario es la persona encargada de llevar el proceso de nómina de la empresa Beehunting y los otros 2 usuarios son personas que forman parte de la empresa en el área de contabilidad; siendo parte de la población objetivo del sistema; por lo tanto existe un mayor grado de validez en la evaluación.

3.2.1.3 Ambiente de pruebas

La evaluación se llevó a cabo en un lapso de dos días. El lugar donde se realizaron fue en la empresa Beehunting; en cada computadora que opera cada usuario normalmente.

3.2.1.4 Plan de pruebas

Antes de comenzar, a cada usuario se le explicó cuál sería su tarea a realizar y se le dio una pequeña introducción acerca del manejo del sistema. Una vez iniciada la prueba, no se brindó ayuda a los usuarios excepto de que ellos lo preguntaran o que llevaran mucho tiempo sin poder resolver un problema.

Los casos de uso representan tareas específicas que cada usuario debe realizar y sobre las cuales se evaluaron los siguientes factores:

- Tiempo necesario para completar la tarea
- Número de errores
- Número de preguntas hechas para solucionar un error o duda
- Si el usuario consultó la ayuda o el manual de usuario

Después de realizar sus tareas correspondientes, los usuarios contestaron un breve cuestionario para completar la evaluación.

Este cuestionario se encuentra compuesto por varias preguntas orientadas a la amigabilidad del sistema con el usuario.

Todos los resultados de cada usuario se encuentran junto con sus respectivos cuestionarios, en el Anexo Pruebas de Usabilidad.

3.2.1.5 Resultados de la evaluación

Con los resultados de estas evaluaciones, es posible derivar conclusiones importantes, sobre todo orientadas al mejoramiento del software para que su operación se adapte más a las costumbres de usuarios reales. De todas las sugerencias sobre los cambios que podrían realizarse al sistema para que esta fuera más intuitiva o amigable; las detallamos a continuación:

- Que la tecla “Enter” sea una alternativa para realizar cualquier proceso de una pantalla que tiene un solo botón. Su justificación está en el hecho de que la mayoría de aplicaciones en la web implementan esta opción, y tiene mucha familiaridad con la mayoría de usuarios.

- Que los mensajes en todo el sistema tengan el mismo estándar o formato, y que los mensajes de error se diferencien con los mensajes de éxito. La justificación se da para diferenciar a simple vista, cuando es un mensaje de error ya que cuando el usuario ve un mensaje cree siempre que es por algún error que cometió en el sistema.

A continuación se detalla la lista de preguntas con la respectiva interpretación de las respuestas dando una calificación entre el rango de 0 a 3 siendo las puntuaciones las siguientes:

0 = Malo

1= Regular

2 = Muy Bueno

3 = Excelente

Pregunta	Interpretación	Promedio
En general el sistema: Me agradó/Me incomodó	El sistema les agradó bastante al grupo que realizó las pruebas.	2.3
Los íconos de los botones: Representan su acción/No representan su acción	Los íconos son bastante representativos y cada acción está bien representada.	3.0
La interacción con el sistema es Intuitiva/Confusa	La interacción con el sistema es muy intuitiva, siempre y cuando se domine los procesos de nómina de empleados.	2.3
Los mensajes del sistema: Me sirvieron/Me confundieron	Los mensajes que genera el sistema son claros y concisos, sin embargo hay que realizar una diferenciación entre los mensajes de error con los de proceso exitoso.	2.0
El propósito del sistema me	El grupo de usuarios considera	3.0

parece: Útil/Inútil	que el propósito del sistema es muy interesante y bastante útil.	
Las opciones que brinda el sistema son: Pocas/Demasiadas	El grupo de usuarios considera que el sistema proporciona el suficiente número de opciones.	3.0
La disponibilidad del sistema mediante el Web me parece: Útil/Inútil	El grupo de usuarios considera que la disponibilidad y el acceso al sistema vía Web es de gran utilidad.	3.0
La ayuda/manual de usuario proporcionado por el sistema es: Informativa/Confusa	La información que contiene la ayuda y manual de usuario son muy útiles, aunque la falencia es que muy poco son leídas.	2.7
Una vez que el sistema este operando, lo usaría: Frecuentemente/Nunca	El grupo usaría el sistema con frecuente regularidad, en especial al final de cada mes.	3.0
En términos generales, el funcionamiento del sistema me parece: Bueno/Malo	En general, el grupo considera que el funcionamiento del sistema es bueno y principalmente mejora el tiempo del proceso de nómina.	3.0

Tabla 2 - 41: Interpretación de la respuestas a las preguntas de usabilidad

CAPITULO IV.

CONCLUSIONES Y RECOMENDACIONES.

4.1 CONCLUSIONES.

- ◆ La metodología de Proceso Unificado Rational (RUP) es una de las metodologías con más amplia disponibilidad de información técnica y práctica así como de las herramientas que permiten implementarla, lo que ayuda en la utilización en el desarrollo de sistemas.
- ◆ Para el desarrollo del sistema fue necesario recurrir a expertos en el área, así como de material técnico que de una u otra manera nos ayudaron a entender y comprender como se va a plantear la solución al problema, de lo que se puede deducir que para un correcto levantamiento de requerimientos se debe trabajar con los usuarios que conocen el proceso y con expertos del mismo.
- ◆ La ejecución de pruebas de usabilidad de un software son muy importantes debido a que nos permiten establecer el grado de aceptación de los usuarios con el sistema y el éxito que tendrá el mismo.
- ◆ El realizar un buen diseño minimiza el riesgo de no cumplir con los requerimientos funcionales y no funcionales del sistema, debido a que es la base para que un proyecto de desarrollo funcione y si no se tiene claro que se debe hacer es imposible que el cliente al final quede satisfecho.
- ◆ El sistema desarrollado contribuye tanto al ahorro de costos de la empresa como ayudar al medio ambiente y evitar el uso de papel, al ser opcional la impresión del rol de pagos, debido a que cuando el empleado lo requiera los podrá consultar vía browser, teniendo la opción de imprimirlo y hacerlo sellar en recursos humanos.

- ◆ El sistema de nómina optimizará el almacenamiento de la información debido a que se almacenará en una base de datos, la cual permitirá obtener respaldos, proteger la información y mantener históricos de sus empleados y pagos realizados.

4.2 RECOMENDACIONES.

- ◆ Se recomienda analizar a fondo la complejidad del sistema, antes de seleccionar una metodología de desarrollo, para ver si resulta o no aconsejable el uso de esta para el desarrollo del sistema a implementarse.
- ◆ Es recomendable utilizar RUP para el desarrollo de sistemas especialmente aquellos de gran complejidad y criticidad debido a las facilidades que provee en la distribución de los flujos de trabajo.
- ◆ Es recomendable que durante la fase de análisis se tenga un alto grado de interacción entre los técnicos y los usuarios, de esta forma se podrá obtener un mayor detalle de los requerimientos funcionales del sistema.
- ◆ Se recomienda que el servidor donde se vaya a instalar la aplicación cumpla al menos los requerimientos mínimos de hardware indicados en la documentación para un buen desempeño del sistema SISNOM.
- ◆ Se debe tratar de usar al máximo estándares a nivel de todas las etapas del desarrollo de un sistema, esto permite mayor portabilidad al mismo y facilita el mantenimiento.
- ◆ Se recomienda continuar con la implementación de otras funcionalidades de nómina que permitan completar todo el proceso de nómina.

BIBLIOGRAFÍA

1. Metodología RUP
<http://www.scribd.com/doc/297224/RUP>
2. Gráfico Ciclo de Vida Pág 11
JACOBSON Ivar, BOOCH Grady, RUMBAUGH James. EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE. Editorial Pearson Education. 2000.
3. Metodología MSF
http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf
4. Ciclo de Vida MSF
<http://arevalomaria.wordpress.com/2010/10/20/estableciendo-ciclo-de-vida-del-proyecto-con-msf/>
5. Metodologías de desarrollo ligeras: XP Xtreme Programing
[http://brconsulting.info/portal/articulos/metodologias-de-desarrollo/extreme-programing---xp.html](http://brconsulting.info/portal/articulos/metodologias-de-desarrollo/extreme-programming---xp.html)
6. Metodologías de desarrollo ligeras: RAD Desarrollo de Aplicaciones Rápida.
<http://members.fortunecity.com/miadinellie/TrabajoFinal.htm>
7. Front-End, Back-End.
<http://www.fce.unju.edu.ar/download/si/unidadvi.pdf>
8. Rational Rose
<http://dmi.uib.es/%7Elabsoft/Labsg/trabajo%20herramientas%20CASE.pdf>
9. Power Designer.
<http://www.sybase.com/>

10. Documento Visión.

<http://chris.queque.com/blog/2008/08/documento-vision/>

11. Casos de Uso.

KROLL Per , KRUCHTEN Philippe , “Rational Unified Process Made Easy: A Practitioner's Guide to the RUP”, Addison Wesley, 2003

12. Diagramas de Secuencia.

ARLOW Jim, NEUSTADT Ila. UML AND THE UNIFIED PROCESS PRACTICAL OBJECT-ORIENTED ANALYSIS A DESING. Addison-Wesley. Estados Unidos de América. 2002

13. Arquitectura Web.

<http://www.milbits.com/916/asp-net-aplicaciones-instalar.html>

GLOSARIO DE TERMINOS

- PYMES:** La sigla PyME (también se puede encontrar como PYME) es la que se utiliza para hacer referencia a las pequeñas y medianas empresas que existen en el mercado de un país.
- Trabajador:** La persona que se obliga a la prestación del servicio o a la ejecución de la obra se denomina trabajador y puede ser empleado u obrero.
- Empleador:** La persona o entidad, de cualquier clase que fuere, por cuenta u orden de la cual se ejecuta la obra o a quien se presta el servicio, se denomina empresario o empleador.
- Contrato de trabajo:** Es aquel celebrado individualmente entre un trabajador y un empleador.
- Décimo Tercer Sueldo:** El décimo tercer sueldo o remuneración de los servidores públicos y trabajadores de las entidades y organismos contemplados en el Art. 101 de la Ley, consiste en una remuneración mensual unificada adicional, que deberá ser pagada hasta el 24 de diciembre de cada año.
- Décimo Cuarto Sueldo:** El décimo cuarto sueldo o remuneración de los servidores públicos y trabajadores de las entidades y organismos contempladas en el Art. 101 de la Ley, consiste en una remuneración básica mínima unificada que será pagada hasta el 15 de abril de cada año, en las regiones de la costa e insular; y, hasta el 15 de septiembre en las regiones de la sierra y oriente, de conformidad con el Art. 113 del Código del Trabajo.
- Rol de Pagos:** Es un documento contable, de carácter interno, en el que se registran los valores que constituyen ingresos y deducciones

para el trabajador por concepto de remuneraciones. Para la empresa estos rubros representan gastos o costos según sea el caso, obligaciones con los trabajadores y con terceros. Estos valores deberán ser entregados por lo general en forma mensual al trabajador, en pago por sus servicios prestados dentro de la institución.

Finiquito: Recibo mediante el cual se da por terminada la relación laboral y el trabajador certifica que su empleador le ha liquidado cuantas obligaciones tenía con él en razón a dicha relación.

Horas Suplementarias: Las horas suplementarias se deberá pagar la remuneración correspondiente con un más 50% de recargo hasta las 24h00, de ahí en adelante se deberá pagar con un 100% de recargo hasta las 06h00.

Horas Extraordinarias: El trabajo que se ejecute el sábado, domingo o díasfestivos será pagado con el 100% de recargo.

.NET: Proyecto de Microsoft, nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.

Ajax: Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

RUP: Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la

metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

UML:

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.