

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS



DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SISTEMAS Y COMPUTACIÓN

DESARROLLO DE UN SISTEMA INFORMÁTICO PARA EL APOYO EN LAS
ACTIVIDADES DE ADMINISTRACIÓN DEL FINANCIAMIENTO DE LA FUNDACIÓN
NIÑEZ Y VIDA TIERRA DE HOMBRES ECUADOR

HERRERA MORA PABLO ANDRÉS

DIRECTOR: ING. FABIÁN DE LA CRUZ

QUITO, AGOSTO 2016

DIRECTOR:

ING. FABIÁN DE LA CRUZ

CORRECTORES:

ING. ANDRÉS JIMÉNEZ

ING. DAMIÁN NICOLALDE

AGRADECIMIENTO

Quiero agradecer a la Fundación Niñez y Vida Tierra de Hombres Ecuador por abrirme las puertas y permitirme ser parte aunque sea un momento de su labor diaria de ayuda. Especialmente quiero agradecer al Dr. Guillermo Ordoñez, director ejecutivo de la fundación y a todo el personal que siempre estuvo dispuesto a ayudarme incondicionalmente.

A mi director el Ingeniero Fabián De la Cruz por todo su apoyo y ayuda incondicional en el desarrollo del presente trabajo, pues sus consejos fueron muy importantes para solventar los diferentes obstáculos que se presentaron.

A mi equipo de lectores el Ingeniero Damián Nicolalde y el Ingeniero Andrés Jiménez por haber compartido sus conocimientos y sabiduría a lo largo de toda la vida universitaria.

Finalmente, quiero agradecer a la Pontificia Universidad Católica del Ecuador y en especial a la Facultad de Ingeniería no solo por la formación profesional sino, sobre todo, por infundir los valores éticos necesarios para el desarrollo profesional.

DEDICATORIA

Quiero dedicar el presente trabajo a mis padres. Pues sin su esfuerzo, dedicación y enseñanzas esto no habría sido posible.

TABLA DE CONTENIDOS

FUNDACIÓN NIÑEZ Y VIDA TIERRA DE HOMBRES ECUADOR.....	1
1.1. RESEÑA HISTÓRICA.....	2
1.2. MISIÓN.....	4
1.3. VISIÓN.....	5
1.4. OBJETIVOS.....	5
1.5. SITUACIÓN ACTUAL DE LA FUNDACIÓN	5
1.6. SITUACIÓN TECNOLÓGICA ACTUAL DE LA FUNDACIÓN.....	6
MARCO TEÓRICO.....	10
2.1. HERRAMIENTAS PARA EL DESARROLLO DEL PRODUCTO	11
2.1.1. LENGUAJE DE PROGRAMACIÓN JAVA	11
2.1.2. LENGUAJES DE MODELAMIENTO UML.....	13
2.1.3. POSTGRESQL.....	18
2.1.4. IDE ECLIPSE	20
2.1.5. SERVIDOR DE APLICACIONES JAVA EE, REDHAT JBOSS	21
2.2. METODOLOGÍA PARA EL DESARROLLO DEL PRODUCTO	22
2.2.1. EXTREME PROGRAMMING	22
2.2.1.1. INTRODUCCIÓN.....	22
2.2.1.2. CICLO DE VIDA.....	23
DESARROLLO DEL PRODUCTO	27
3.1. EXPLORACIÓN.....	28
3.1.1. RESUMEN DEL PROCESO DE ADMINISTRACIÓN DEL FINANCIAMIENTO.....	28
3.1.2. DEFINICIÓN DEL PROBLEMA	29
3.1.3. HISTORIA DE USUARIO	30
3.1.3.1. MODELO DE CASOS DE USO	42
3.2. PLANIFICACIÓN DE ENTREGAS	43
3.2.1. DIAGRAMA DE ACTIVIDADES.....	44
3.2.2. PLAN DEL PROYECTO.....	48
3.3. ITERACIONES	49
3.3.1. ITERACIÓN 1.....	49
3.3.1.1. DISEÑO LÓGICO DE INTERFACES PRIMERA ITERACIÓN.....	49
3.3.1.2. MODELO DE DATOS PRIMERA ITERACIÓN.....	53

3.3.1.3. CÓDIGO FUENTE PRIMERA ITERACIÓN.....	54
3.3.2. ITERACIÓN 2.....	57
3.3.2.1. DISEÑO LÓGICO DE INTERFACES SEGUNDA ITERACIÓN	57
3.3.2.2. MODELO DE DATOS SEGUNDA ITERACIÓN.....	58
3.3.2.3. CÓDIGO FUENTE SEGUNDA ITERACIÓN	59
a).....	60
3.3.3. ITERACIÓN 3.....	64
3.3.3.1. DISEÑO LÓGICO DE INTERFACES TERCERA ITERACIÓN	64
3.3.3.2. MODELO DE DATOS TERCERA ITERACIÓN	70
3.3.3.3. CÓDIGO FUENTE TERCERA ITERACIÓN	71
3.4. PRUEBAS Y MANTENIMIENTO.....	74
3.4.1. PLAN DE PRUEBAS DE ACEPTACIÓN	74
3.4.2. INFORME PARA IMPLANTACIÓN.....	77
3.5. CIERRE DEL PROYECTO	78
3.5.1. DOCUMENTACIÓN	79
3.5.1.1. MANUAL TÉCNICO.....	80
3.5.1.2. MANUAL DE USUARIO.....	80
3.5.1.3. MANUAL DE INSTALACIÓN	80
CONCLUSIONES Y RECOMENDACIONES	81
4.1. CONCLUSIONES	82
4.2. RECOMENDACIONES.....	82
ANEXO 1: PLAN DE ENTREGAS.....	84
ANEXO 2: MANUAL TÉCNICO.....	86
OBJETIVO	87
1. DESCRIPCIÓN DEL AMBIENTE DE DESARROLLO	87
2. INTERFAZ DE USUARIO	87
2.1. PÁGINA MAESTRA.....	87
2.2. PÁGINA LIGADA.....	88
3. MANEJO DE OBJETOS MAPEADOS DE LA BASE DE DATOS.....	88
3.1. INTERFAZ.....	88
3.2. IMPLEMENTACIÓN.....	89
4. CONTROLADORES.....	90
ANEXO 3: MANUAL DE USUARIO	93
OBJETIVO	94

1. AUSPICIANTE	94
1.1 INGRESAR.....	94
1.2 EDITAR	95
1.3 ELIMINAR.....	96
1.4 AÑADIR CUENTA BANCARIA.....	97
1.4.1 INGRESAR PRESUPUESTO	98
2. EMPLEADO	100
2.1 INGRESAR.....	100
2.2 ASIGNAR PROYECTO.....	101
3. PROYECTO	102
3.1 INGRESAR.....	103
3.2 ASIGNAR EMPLEADOS AL PROYECTO	104
3.3 CUENTA BANCARIA AL PROYECTO.....	105
4. INGRESAR CUENTA BANCARIA	106
5. ROLES DE PAGO	106
5.1 INGRESO DE ROL DE PAGO.....	107
5.2 IMPRESIÓN DE ROLES: CONSOLIDADO PERSONAL.....	109
5.3 IMPRESIÓN DE ROLES: PERSONAL.....	109
5.4 IMPRESIÓN DE ROLES: CONSOLIDADO GLOBAL POR CUENTA.....	110
6. INGRESO DE GASTOS.....	111
7. INFORME DE EJECUCIÓN DEL PRESUPUESTO.....	111
8. USUARIO.....	112
8.1 INGRESO	113
8.3 MODIFICACIÓN DE USUARIO	113
9. INGRESO Y SALIDA DEL APLICATIVO	114
9.2 INGRESAR AL SISTEMA	114
9.1 SALIR DEL SISTEMA.....	115
ANEXO 4: MANUAL DE INSTALACIÓN	117
OBJETIVO	118
1. ARCHIVOS REQUERIDOS	118
2. INSTALAR JDK	118
2.1 VARIABLES DE ENTORNO JAVA-HOME.....	121
3. INSTALAR MOTOR DE BASE DE DATOS POSTGRESQL 9.5	122
3.1 CARGAR LA BASE DE DATOS RESPALDO XXXXXX.BACKUP.....	125
5. INSTALAR EL APLICATIVO	126

BIBLIOGRAFÍA.....127

ÍNDICE DE TABLAS

TABLA 1 PROYECTOS DE LA FUNDACIÓN NIÑEZ Y VIDA TIERRA DE HOMBRES ECUADOR.....	4
TABLA 2 CARACTERÍSTICAS DE JAVA.....	13
TABLA 3 PRINCIPALES DIAGRAMAS DE UML.....	18
TABLA 4 PRINCIPALES CARACTERÍSTICAS DE POSTGRESQL.....	20
TABLA 5 PRINCIPALES REGLAS Y PRACTICAS DE XP.....	27
TABLA 6 PLANTILLA PARA HISTORIAS DE USUARIO.....	30
TABLA 7 ADMINISTRAR AUSPICIANTE.....	30
TABLA 8 INGRESAR AUSPICIANTE.....	31
TABLA 9 ACTUALIZAR AUSPICIANTE.....	31
TABLA 10 ELIMINAR AUSPICIANTE.....	31
TABLA 11 CONSULTAR AUSPICIANTE.....	32
TABLA 12 ADMINISTRAR ROLES.....	32
TABLA 13 INGRESAR ROLES.....	32
TABLA 14 ELIMINAR ROLES.....	33
TABLA 15 CONSULTAR ROLES.....	33
TABLA 16 ADMINISTRAR PROYECTOS.....	33
TABLA 17 INGRESAR PROYECTOS.....	34
TABLA 18 ACTUALIZAR PROYECTOS.....	34
TABLA 19 ELIMINAR PROYECTOS.....	34
TABLA 20 CONSULTAR ROLES.....	35
TABLA 21 ADMINISTRAR EMPLEADOS.....	35
TABLA 22 INGRESAR EMPLEADOS.....	35
TABLA 23 ACTUALIZAR EMPLEADOS.....	36
TABLA 24 ELIMINAR EMPLEADOS.....	36
TABLA 25 CONSULTAR EMPLEADOS.....	36
TABLA 26 ADMINISTRAR CUENTAS BANCARIAS.....	37
TABLA 27 INGRESAR CUENTAS BANCARIAS.....	37
TABLA 28 ACTUALIZAR CUENTAS BANCARIAS.....	38
TABLA 29 ELIMINAR CUENTAS BANCARIAS.....	38
TABLA 30 CONSULTAR EMPLEADOS.....	38
TABLA 31 ADMINISTRAR GASTOS.....	39
TABLA 32 INGRESAR GASTOS.....	39

TABLA 33 ELIMINAR GASTOS	39
TABLA 34 CONSULTAR GASTOS	40
TABLA 35 ADMINISTRAR USUARIOS.....	40
TABLA 36 INGRESAR USUARIOS.....	40
TABLA 37 ACTUALIZAR USUARIOS	41
TABLA 38 ELIMINAR USUARIOS	41
TABLA 39 CONSULTAR USUARIOS	41
TABLA 40 ADMINISTRAR USUARIOS.....	42
TABLA 41 PLAN DE PRUEBAS DE ACEPTACIÓN	77

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 ESQUEMA DE CONEXIÓN DE RED	7
ILUSTRACIÓN 2 MACROPROCESOS	7
ILUSTRACIÓN 3 ESQUEMA DESARROLLO INSTITUCIONAL	8
ILUSTRACIÓN 4 DIAGRAMA DE CASOS DE USO	43
ILUSTRACIÓN 5 DIAGRAMA DE ACTIVIDADES PARA ASIGNACIÓN DE EMPLEADOS A PROYECTOS	45
ILUSTRACIÓN 6 DIAGRAMA DE ACTIVIDADES PARA LA ASIGNACIÓN DE UNA CUENTA BANCARÍA A UN AUSPICIANTE	46
ILUSTRACIÓN 7 DIAGRAMA DE ACTIVIDADES PARA ASIGNAR EL FINANCIAMIENTO A UN PROYECTO.....	47
ILUSTRACIÓN 8 DIAGRAMA DE ACTIVIDADES PARA EL REGISTRO DE PAGO DE SUELDOS A EMPLEADOS	48
ILUSTRACIÓN 9 INTERFACES GRAFICAS AUSPICIANTE.....	50
ILUSTRACIÓN 10 INTERFACES GRAFICAS PROYECTOS	51
ILUSTRACIÓN 11 INTERFACES GRAFICAS EMPLEADO.....	52
ILUSTRACIÓN 12 INTERFACES GRAFICAS CUENTAS BANCARIAS	53
ILUSTRACIÓN 13 MODELO FÍSICO DE LA BASE DE DATOS PRIMERA ITERACIÓN.....	54
ILUSTRACIÓN 14 EXTRACTO CÓDIGO CUENTA BANCO PRIMERA ITERACIÓN	55
ILUSTRACIÓN 15 EXTRACTO CODIGO INTERFAZ DE SERVICIO PRIMERA ITERACIÓN	56
ILUSTRACIÓN 16 EXTRACTO CODIGO IMPLEMENTACIÓN INTERFAZ PRIMERA ITERACIÓN	57
ILUSTRACIÓN 17 INTERFACES GRAFICAS ROLES DE PAGO.....	58
ILUSTRACIÓN 18 MODELO FÍSICO DE LA BASE DE DATOS SEGUNDA ITERACIÓN	59
ILUSTRACIÓN 19 EXTRACTO CÓDIGO MAPEO DE TABLA ROL.....	60
ILUSTRACIÓN 20 EXTRACTO CÓDIGO INTERFACE SERVICIO ROL	61
ILUSTRACIÓN 21 EXTRACTO DE CODIGO IMPLEMENTACIÓN DE LA INTERFAZ DEL SERVICIO ROL	64
ILUSTRACIÓN 22 INTERFACES GRAFICAS GASTOS	65
ILUSTRACIÓN 23 INTERFACES GRAFICAS CONTROL DE USUARIOS	67
ILUSTRACIÓN 24 INTERFAZ GRÁFICA INFORMES	67

ILUSTRACIÓN 25 IMPRESIÓN DE ROLES DE PAGO	69
ILUSTRACIÓN 26 MODELO FÍSICO DE LA BASE DE DATOS PRIMERA ITERACIÓN.....	70
ILUSTRACIÓN 27 VISTA DINÁMICA EJECUCIÓN DEL PRESUPUESTO.....	71
ILUSTRACIÓN 28 FUNCIÓN PARA INSERCIÓN SE SALDO A LA FECHA	72
ILUSTRACIÓN 29 INSERCIÓN DE TRANSACCIONES.....	73
ILUSTRACIÓN 30 ANEXO 2: PÁGINA MAESTRA	87
ILUSTRACIÓN 31 ANEXO 2: PÁGINA LIGADA	88
ILUSTRACIÓN 32 ANEXO 2: DESCRIPCIÓN DE INTERFAZ	88
ILUSTRACIÓN 33 ANEXO 2: EJEMPLO DE IMPLEMENTACIÓN DE INTERFAZ DE SERVICIO	89
ILUSTRACIÓN 34 ANEXO 2: EXTRACTO CÓDIGO XHTML EMPLEADO	91
ILUSTRACIÓN 35 ANEXO 2: EXTRACTO DE CÓDIGO DEL CONTROLADOR EMPLEADO	92
ILUSTRACIÓN 36 ANEXO 3: MÓDULO AUSPICIANTE	94
ILUSTRACIÓN 37 ANEXO 3: NUEVO AUSPICIANTE	94
ILUSTRACIÓN 38 ANEXO 3: VALIDACIÓN NUEVO REGISTRO	95
ILUSTRACIÓN 39 ANEXO 3: EDICIÓN DE AUSPICIANTE.....	95
ILUSTRACIÓN 40 ANEXO 3: EDICIÓN CAMPOS AUSPICIANTE	96
ILUSTRACIÓN 41 ANEXO 3: VALIDACIÓN DE EDICIÓN.....	96
ILUSTRACIÓN 42 ANEXO 3: BOTÓN ELIMINAR AUSPICIANTE	96
ILUSTRACIÓN 43 ANEXO 3: CONFIRMACIÓN DE BORRADO	97
ILUSTRACIÓN 44 ANEXO 3: BOTÓN CUENTAS	97
ILUSTRACIÓN 45 ANEXO 3: SELECCIÓN DE CUENTA BANCARIA	97
ILUSTRACIÓN 46 ANEXO 3: VALIDACIÓN DE INGRESO DE CUENTA	98
ILUSTRACIÓN 47 ANEXO 3: BOTÓN INGRESAR PRESUPUESTO	98
ILUSTRACIÓN 48 ANEXO 4: LLENADO DE CAMPOS PARA INGRESO DE PRESUPUESTO.....	98
ILUSTRACIÓN 49 ANEXO 3: VENTANA DE CONFIRMACIÓN DE INGRESO DE PRESUPUESTO.....	99
ILUSTRACIÓN 50 ANEXO 3: VALIDACIÓN DEL INGRESO DEL PRESUPUESTO	99
ILUSTRACIÓN 51 ANEXO 3: MÓDULO EMPLEADO	100
ILUSTRACIÓN 52 ANEXO 3: INGRESO DE NUEVO EMPLEADO.....	100
ILUSTRACIÓN 53 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN NUEVO EMPLEADO .	101
ILUSTRACIÓN 54 ANEXO 3: ASIGNACIÓN DE PROYECTO DESDE EL MÓDULO DE EMPLEADOS	101

ILUSTRACIÓN 55 ANEXO 3: VALIDACIÓN DE LA ASIGNACIÓN DE UN PROYECTO A UN EMPLEADO.....	102
ILUSTRACIÓN 56 ANEXO 3: MÓDULO DE PROYECTOS	102
ILUSTRACIÓN 57 ANEXO 3: INGRESO DE NUEVO PROYECTO	103
ILUSTRACIÓN 58 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN NUEVO PROYECTO.	103
ILUSTRACIÓN 59 ANEXO 3: ASIGNACIÓN DE UN EMPLEADO A UN PROYECTO.....	104
ILUSTRACIÓN 60 ANEXO 3: VALIDACIÓN DE LA ASIGNACIÓN DE UN EMPLEADO A UN PROYECTO	104
ILUSTRACIÓN 61 ANEXO 3: ASIGNACIÓN DE CUENTA BANCARIA A UN PROYECTO.	105
ILUSTRACIÓN 62 ANEXO 3: VALIDACIÓN DE LA ASIGNACIÓN DE UNA CUENTA BANCARIA A UN PROYECTO.....	105
ILUSTRACIÓN 63 ANEXO 3: MÓDULO CUENTA BANCARIA. INGRESO NUEVA CUENTA	106
ILUSTRACIÓN 64 ANEXO 3: MÓDULO DE ROLES DE PAGO.....	106
ILUSTRACIÓN 65 ANEXO 3: BOTÓN ASIGNAR ROL DE PAGOS.....	107
ILUSTRACIÓN 66 ANEXO 3: VENTANA PARA EL INGRESO DE NUEVO ROL DE PAGO	107
ILUSTRACIÓN 67 ANEXO 3: CAMPOS PARA EL REGISTRO DE UN ROL DE PAGOS....	108
ILUSTRACIÓN 68 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN ROL DE PAGOS.....	108
ILUSTRACIÓN 69 ANEXO 3: IMPRESIÓN ROL CONSOLIDADO PERSONAL.....	109
ILUSTRACIÓN 70 ANEXO 3: IMPRESIÓN ROL DE PAGO PERSONAL.....	109
ILUSTRACIÓN 71 ANEXO 3: IMPRESIÓN ROL CONSOLIDADO GLOBAL.....	110
ILUSTRACIÓN 72 ANEXO 3: FORMATO DE IMPRESIÓN ROL CONSOLIDADO GLOBAL	110
ILUSTRACIÓN 73 ANEXO 3: INGRESO DE GASTOS.....	111
ILUSTRACIÓN 74 ANEXO 3: INFORME DE EJECUCIÓN DEL PRESUPUESTO	111
ILUSTRACIÓN 75 ANEXO 3: GRÁFICA DE EJECUCIÓN DE PRESUPUESTO.....	112
ILUSTRACIÓN 76 ANEXO 3: FORMATO DE IMPRESIÓN DEL INFORME DE EJECUCIÓN DEL PRESUPUESTO	112
ILUSTRACIÓN 77 ANEXO 3: MÓDULO DE USUARIOS.....	112
ILUSTRACIÓN 78 ANEXO 3: INGRESO DE NUEVO USUARIO	113
ILUSTRACIÓN 79 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN NUEVO USUARIO ...	113
ILUSTRACIÓN 80 ANEXO 3: EDICIÓN DE USUARIOS.....	113
ILUSTRACIÓN 81 ANEXO 3: PÁGINA PRINCIPAL.....	114
ILUSTRACIÓN 82 ANEXO 3: MENSAJE DE VALIDACIÓN DE CLAVE Y USUARIO.....	114
ILUSTRACIÓN 83 ANEXO 3: INGRESO EXITOSO AL SISTEMA	115

ILUSTRACIÓN 84 ANEXO 3: SALIR DEL SISTEMA	115
ILUSTRACIÓN 85 ANEXO 3: SALIDA EXITOSA DEL SISTEMA.....	116
ILUSTRACIÓN 86 ANEXO 4: ARCHIVOS REQUERIDOS PARA LA INSTALACIÓN.....	118
ILUSTRACIÓN 87 ANEXO 4: PANTALLA DE INSTALACIÓN JDK 1	118
ILUSTRACIÓN 88 ANEXO 4: PANTALLA DE INSTALACIÓN JDK 2	119
ILUSTRACIÓN 89 ANEXO 4: PANTALLA DE INSTALACIÓN JDK 3	119
ILUSTRACIÓN 90 ANEXO 4: SELECCIÓN CARPETA DESTINO JDK.....	120
ILUSTRACIÓN 91 ANEXO 4: FIN DE INSTALACIÓN JDK	120
ILUSTRACIÓN 92 ANEXO 4: SETEO DE LAS VARIABLES DE ENTORNO.....	121
ILUSTRACIÓN 93 ANEXO 4: INGRESO DEL REGISTRO	122
ILUSTRACIÓN 94 ANEXO 4: INSTALACIÓN POSTGRESQL 9.5.....	122
ILUSTRACIÓN 95 ANEXO 4: SELECCIÓN DEL DIRECTORIO POSTGRESQL 9.5.....	123
ILUSTRACIÓN 96 ANEXO 4: INGRESO DE CONTRASEÑA POSTGRESQL 9.5	124
ILUSTRACIÓN 97 ANEXO 4: FIN DE INSTALACIÓN POSTGRESQL 9.5.....	125
ILUSTRACIÓN 98 ANEXO 4: CARGA DE LA BASE DE DATOS EN EL MOTOR POSTGRESQL 9.5	125

CAPÍTULO: I

FUNDACIÓN NIÑEZ Y VIDA TIERRA DE HOMBRES ECUADOR

En el presente capítulo se abordan temas que permitirán conocer de mejor manera a la institución donde se realizará el presente proyecto, en otras palabras se realizará una introducción propiamente de la Fundación Niñez y Vida Tierra de Hombres Ecuador, su historia, sus objetivos, su situación actual. Para este fin primero iniciaremos con una breve reseña de la historia de la fundación. Seguido a esto se aborda la misión y la visión de la misma. Pasando por los objetivos y la situación actual de la fundación llegamos a la situación tecnológica actual de la misma con el fin de ubicar de mejor manera en qué condiciones se iniciará el presente proyecto informático

1.1. RESEÑA HISTÓRICA

En el año 1976, Jacobo Huber, de nacionalidad suiza, llega a la ciudad de Quito para trabajar como voluntario social; se integró al club Brethren, grupo que trabajaba apoyando a personas privadas de libertad. Este contacto que mantuvo Jacobo le permitió conocer de primera mano no solo la realidad de niñas y niños abandonados que dormían en las calles, sino también el estado de las y los inmigrantes indígenas y las dificultades de estas familias en sus labores diarias. (Campoverde Luna, 2004)

Jacobo, preocupado por los problemas que afrontan las familias de inmigrantes indígenas; se contacta con José Robayo, profesor de la Escuela de Trabajo Social de la Universidad Central del Ecuador, con el fin de organizar un trabajo de investigación socioeconómica de las personas de los sectores más pobres del centro histórico de Quito. (Campoverde Luna, 2004)

Este estudio, que contó con 51 casos de estudio, arrojó los siguientes datos:

- El 16% de los vendedores y vendedoras pertenecían al barrio del Placer, el 10 % al sector de Toctiuco y barrios cercanos al Mercado de San Roque.
- Eran personas con ingresos bajos aproximadamente 50 sucres diarios
- La mayoría tenían niños y niñas en edad preescolar y escolar
- La edad promedio del padre era de 40 años y las madres de 20 a 30 años
- En general eran familias completas (padre, madre, hijos). Muchas familias llevaban a sus hijos e hijas a su lugar de trabajo.
- Los niños y las niñas estaban expuestos a las difíciles condiciones climáticas por lo que con frecuencia se enfermaban de resfríos.

Concluida la investigación se ve oportuno crear una guardería; un lugar donde las madres y padres podían dejar a sus hijos e hijas durante su labor diaria y que satisfaga las necesidades de los niños y niñas referente a alimentación, cuidado y salud. Es entonces en el año de 1976 cuando se tomó la decisión de montar la guardería con el apoyo de la Fundación Terre des Hommes Lausanne de Suiza, en el barrio de Toctiuco, que está a una distancia de 30 minutos a pie del mercado central, es cercano al barrio del Placer y estaba mejor organizado que éste último. (Campoverde Luna, 2004)

La guardería Tierra de Hombres, en Toctiuco, inicia su trabajo en octubre de 1976, con Jacobo como director, Ana María como enfermera y Lupe Peña y Vilma Vargas para el trabajo directo con los niños y niñas. El centro se inaugura con la asistencia de 22 niños y niñas. Quienes para su ingreso debían tener entre los 2 y 6 años de edad, certificado de vacunas y ser de una familia de escasos recursos económicos. (Campoverde Luna, 2004)

Este proyecto de guardería se marca como el inicio de una serie de proyectos de ayuda social que la fundación ha desarrollado a lo largo de estos 40 años.

A continuación se muestra un resumen de los principales proyectos realizados:

NOMBRE	AÑO	DESCRIPCIÓN
Centro infantil Tierra de Hombres	1976 - Actualidad.	Acoge a niños y niñas de 2 a 6 años de edad, en condiciones de riesgo y pertenecen a familias en situación económica y social crítica.
Programa Quiero Vivir	1978 - 2006	En este programa participan niños, niñas y jóvenes hasta los 17 años de edad. Se especializa en la asistencia médica, apoyo económico y acompañamiento social a niños y niñas que requieren tratamiento médico emergente.
Programa de Salud Comunitaria	1983 - 1995	El objetivo de este programa era prestar atención primaria en salud a niños, niñas y familias.
Centros Autogestionados¹: Centro social Toctiuco, Nueva vida, Ayllus, Semilla, Crecer y Vivir	1988 - 2015	Participan niños, niñas, jóvenes y familias con capacidad de organización.
Programa con niños, niñas, jóvenes y familias recicladoras	1988 - 2003	Se especializa en la asistencia médica, apoyo económico a niños y niñas de 2 a 17 años de edad.
El Centro Experimental Refugio de los Sueños	1990 - Actualidad	Es un programa que previene la callejización de la niñez y juventud del barrio de Toctiuco, mediante talleres ocupacionales de carpintería y de panadería,

¹ Centros con apoyo logístico de la fundación Niñez y Vida Tierra de Hombres Ecuador.

		actividades recreativas y de refuerzo escolar.
Programa de Salud Materno-Infantil. (Guamote)	2001 - 2006	Participa la población del cantón de Guamote, Chimborazo. Es un programa en el que se trabaja conjuntamente con el gobierno local del cantón. Su objetivo es reducir significativamente las enfermedades y muerte de las mujeres embarazadas, niños y niñas menores de 5 años.
Programa de Salud Materno-Infantil. (Sto. Domingo de los Tsáchilas)	2006 - 2014	Al igual que su predecesor de Guamote, este proyecto busca mejorar el acceso a los servicios de salud de las personas que viven en los Asentamientos de la Provincia de Santo Domingo de los Tsáchilas.
Programa de Salud Sexual y Reproductiva (Sto. Domingo de los Tsáchilas)	2015 - Actualidad	El objetivo principal de este proyecto, que se desarrolla en la provincia de Sto. Domingo de los Tsáchilas, es brindar capacitación y apoyo directo a las madres adolescentes principalmente en temas de salud reproductiva.
<p>FUENTES: Entrevista y datos tomados de la publicación (Campoverde Luna, 2004, pág. 24) ELABORADO POR: Pablo Herrera, 2016.</p>		

Tabla 1 PROYECTOS DE LA FUNDACIÓN NIÑEZ Y VIDA TIERRA DE HOMBRES ECUADOR

1.2. MISIÓN

La Fundación Niñez y Vida Tierra de Hombres Ecuador declara su misión de la siguiente manera:

Somos una organización no gubernamental ecuatoriana que busca fortalecer capacidades locales de gestión; facilitar y compartir procesos para la aplicación de una propuesta educativa basada en el respeto a procesos de vida a favor de niños, niñas, jóvenes y sus familias, excluidos de derechos, afectados por formas diversas de maltrato y para el desarrollo de actividades culturalmente aceptadas para el mejoramiento de la situación de salud, especialmente en materno infantil (FUNDACION NIÑEZ Y VIDA, 2016).

1.3. VISIÓN

La Fundación Niñez y Vida Tierra de Hombres Ecuador declara su visión de la siguiente manera:

Institución dinámica y sostenible, valorada a nivel nacional e internacional que, a través de la aplicación de una propuesta integral de desarrollo sustentada en actividades de educación y salud basada en el respeto a procesos naturales de vida, lidera acciones conducentes a que niños/as y adolescentes ecuatorianos pobres y sus familias ejerzan sus derechos y alcancen plena autonomía de vida. (FUNDACION NIÑEZ Y VIDA, 2016)

1.4. OBJETIVOS

1.4.1 “Generar y respetar los procesos a favor de la vida, en especial de los niños más necesitados y de sus familias, compartiendo experiencias, oportunidades, y condiciones adecuadas a través de acciones concretas” (FUNDACION NIÑEZ Y VIDA, 2016).

1.4.2 “Gestionar e implementar procesos sostenibles de asistencia técnica y capacitación a niños, niñas, jóvenes y sus familias de menores recursos económicos en las áreas de educación, salud y desarrollo humano” (FUNDACION NIÑEZ Y VIDA, 2016).

1.5. SITUACIÓN ACTUAL DE LA FUNDACIÓN

Actualmente, “la Fundación Niñez y Vida es una organización no gubernamental, sin fines de lucro, con personería jurídica otorgada por el ministerio de Bienestar Social mediante Acuerdo Ministerial No. 0303 y actualizado con Acuerdo Ministerial No.0084” (FUNDACION NIÑEZ Y VIDA, 2016).

Actualmente la fundación es miembro del “Movimiento Internacional Terre des Hommes” con sede en Suiza. Además, se encuentra a cargo de tres programas: Centro Experimental Refugio de los Sueños, Programa de Salud Sexual y Reproductiva, y el Centro Infantil Tierra de Hombres. Inmersos en dichos proyectos se encuentran 14 personas, de las cuales 4 a tiempo completo y un empleado a tiempo parcial desarrollan sus actividades en la oficina central, ubicada en la ciudad de Quito (FUNDACION NIÑEZ Y VIDA, 2016).

1.6. SITUACIÓN TECNOLÓGICA ACTUAL DE LA FUNDACIÓN

Actualmente (Enero, 2016) fecha en la que se realiza el presente trabajo de investigación, la Fundación Niñez y vida cuenta con las siguientes características tecnológicas, en su oficina central, ubicada en Quito.

1 Computador de escritorio:

- Marca: Xtratech
- Procesador: AMD Phenom 8650 TRIPLE CORE 2.30 GHz
- RAM: 3.0 GB
- Sistema Operativo: WINDOWS 7 Profesional 32 bits.
- Disco duro: 500 GB.

1 Computador de escritorio:

- Marca: Xtratech
- Procesador: Intel Core 2 DUO E7500 2.93 GHz.
- RAM: 2.0 GB.
- Sistema Operativo: WINDOWS 7 Profesional 32 bits.
- Disco duro: 500.

1 Computador de escritorio:

- Marca: Xtratech
- Procesador: Intel Core 2 DUO E7500 2.93 GHz
- RAM: 4.0 GB.
- Sistema Operativo: WINDOWS 7 Profesional 32 bits.
- Disco duro: 500 GB.

1 Computador de portátil:

- Marca:
- Procesador: Intel Core i5-2520M 2.50HGz.
- RAM: 8GB.
- Sistema Operativo: WINDOWS 7 Enterprise.
- Disco duro: 230 GB.

3 Impresoras:

- EPSON TX210.
- DESCKJET 950C.
- LEXMARK

Equipos de red:

- Router WiFi CNT HUAWEI HG531S.
- Modem.

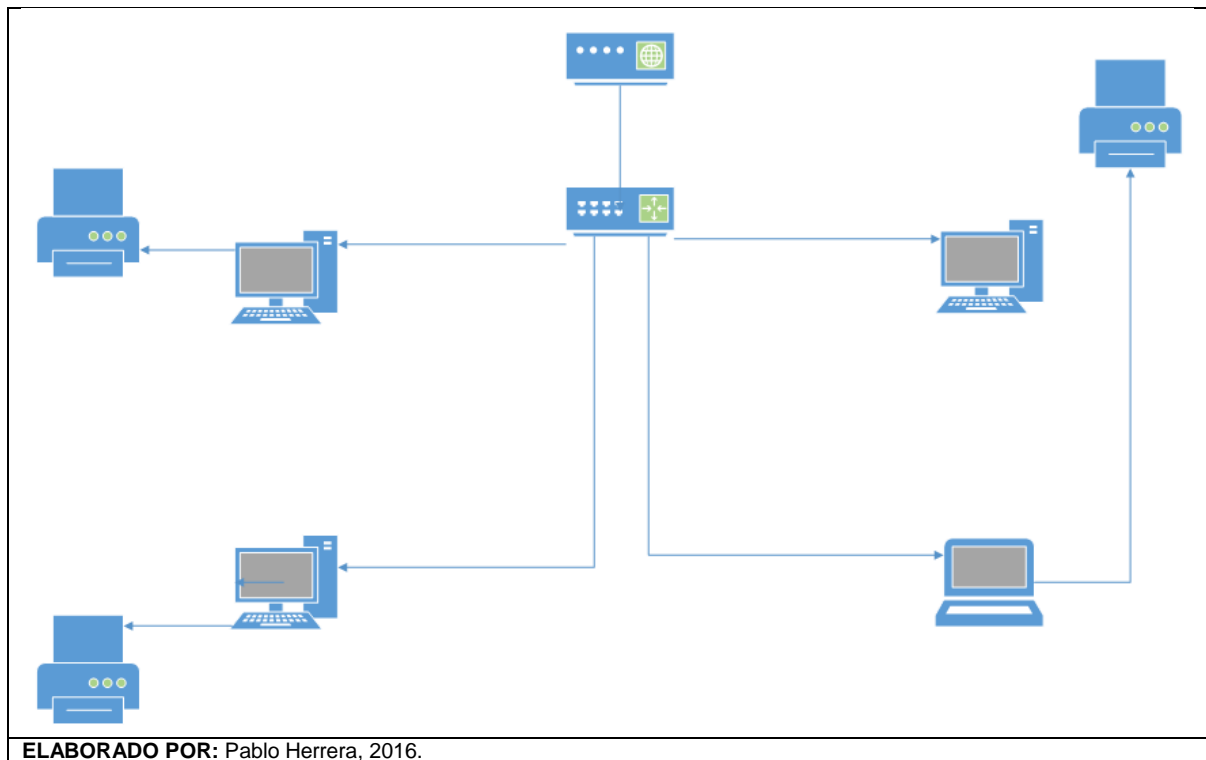


Ilustración 1 ESQUEMA DE CONEXIÓN DE RED

1.8. MAPA DE PROCESOS

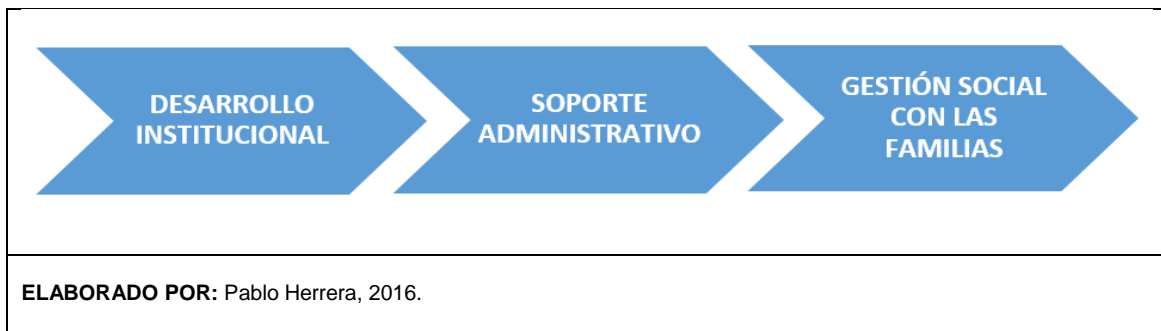
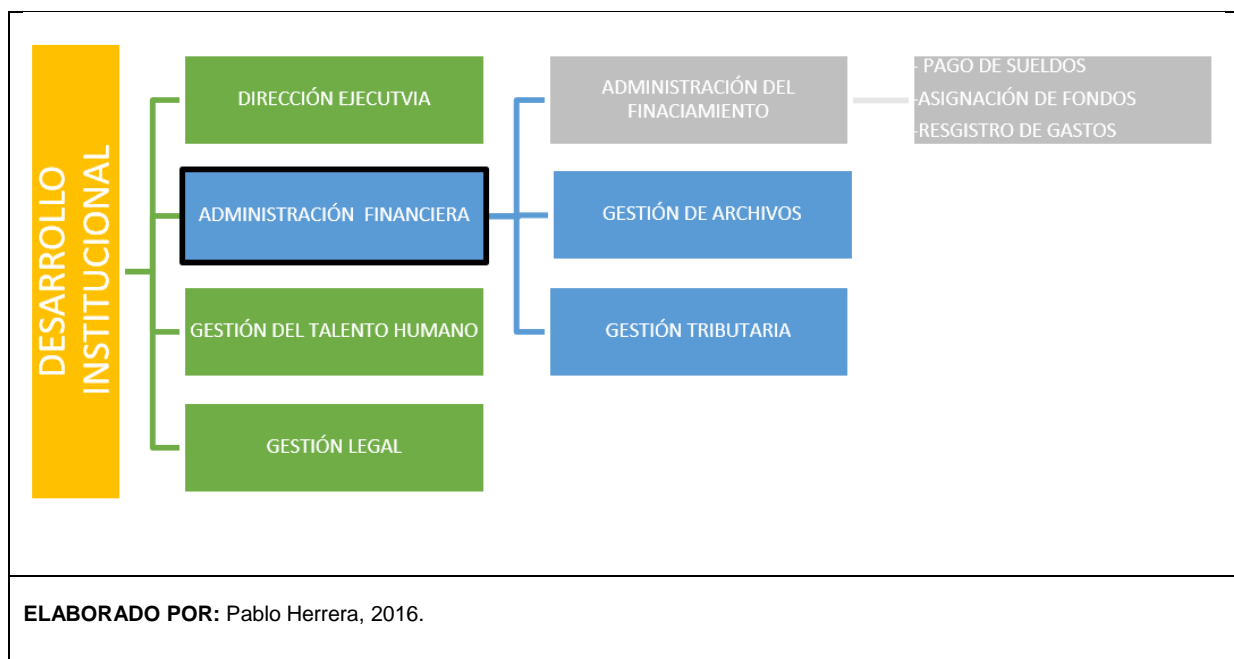


Ilustración 2 MACROPROCESOS

1.8.1 DESARROLLO INSTITUCIONAL

Para que el macroproceso de Gestión Social con las Familias pueda llevarse a cabo se estableció el macroproceso de Desarrollo Institucional, en el que se articulan los procesos de Dirección Ejecutiva, Gestión del Talento Humano y el de Administración y Finanzas.

El presente trabajo de disertación se centrará en el macroproceso de Gestión de Apoyo, donde se encuentra el área de Administración y Finanzas. A continuación se muestra el macroproceso de Desarrollo Institucional:



ELABORADO POR: Pablo Herrera, 2016.

Ilustración 3 ESQUEMA DESARROLLO INSTITUCIONAL

1.8.1.1 DIRECCIÓN EJECUTIVA

Se encarga de supervisar el buen funcionamiento de los proyectos de la fundación, realizar reuniones periódicas de trabajo con los responsables de cada proyecto para brindar acompañamiento o asesoría a los coordinadores de los proyectos para la toma de decisiones y soluciones a ciertos problemas complejos. También se ocupa de la recaudación de fondos con la presentación de nuevos proyectos, coordina el trabajo con los departamentos de contabilidad y recursos humanos, y paralelo a esto vigila el buen desarrollo de los programas. Una actividad muy importante es atender a las personas que requieren información de la fundación, asistir a eventos de relaciones públicas a fin de mantener contacto con autoridades del gobierno u otros donantes.

1.8.1.2 ADMINISTRACIÓN FINANCIERA

Es la encargada de todo el manejo financiero de la fundación. Esto encierra un buen control presupuestario a fin de que el dinero sea invertido en las actividades planificadas como son la entrega de fondos de funcionamiento a los proyectos, pago de sueldos, y control de los gastos. Brindar información financiera requerida por los donantes de acuerdo a sus requerimientos. Cumplir con los requerimientos del gobierno como son los pagos fiscales y laborales. Estar al día en el pago de los servicios básicos. Existencia de un archivo físico de la documentación de la fundación y control del mismo. Realizar pagos, emitir cheques soportando los comprobantes con la debida documentación.

1.8.1.3 GESTIÓN DEL TALENTO HUMANO

Se encarga de la contratación del personal de la fundación, afiliación al I. E .S. S., control de horarios de trabajo, y vacaciones. Brindar atención adecuada al personal a fin de que se sientan a gusto en su trabajo. Reuniones individuales con el personal a fin de motivarlos a capacitarse para mejoramiento de sus habilidades.

1.8.1.4 GESTIÓN LEGAL

Se encarga de realizar los trámites legales necesarios para el buen funcionamiento de la fundación. Cuando hay nuevo personal contratado realiza la legalización de los contratos de trabajo en el Ministerio de Trabajo, y cuando hay personal que deja de trabajar realiza las actas de finiquito, está atento a cualquier necesidad que surja en cualquier momento.

CAPÍTULO: II

MARCO TEÓRICO

En el presente capítulo se abordará temas referentes a las herramientas que se usarán en el desarrollo del proyecto informático. Primero, se presenta el lenguaje de programación Java; para el cual se realiza una breve introducción a su historia y se denotan algunas de sus características. Segundo, se analiza UML que es una herramienta de modelado el cual permitirá visualizar sistémicamente el proyecto. Tercero, se pasa a explicar sobre el sistema gestor de bases de datos PostgreSQL, en el cual se analiza sus principales características así como su estado actual. Cuarto, el análisis se centra en brindar una visión breve de las herramientas tanto IDE Eclipse como el servidor de aplicaciones JBoss. Finalmente, se tratará sobre la metodología seleccionada para el desarrollo del proyecto, Extreme Programming, para la cual se iniciará con una breve explicación del origen y sus principales características; pasando después, por los principios fundamentales de la metodología para analizar el ciclo de vida de la misma; y finalmente, se expone unas reglas y prácticas sugeridas para que el trabajo bajo los lineamientos de Extreme Programming sea satisfactorio.

2.1. HERRAMIENTAS PARA EL DESARROLLO DEL PRODUCTO

2.1.1. LENGUAJE DE PROGRAMACIÓN JAVA

Como nos dice Jorge Martínez, en su publicación sobre Fundamentos de programación Java, los lenguajes de programación pueden ser entendidos como:

Idiomas artificiales diseñados para expresar cálculos o procesos que serán llevados a cabo por ordenadores. Un lenguaje de programación está formado por un conjunto de palabras reservadas, símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (Martínez Ladrón de Guevara, s.f., pág. 1).

Estos lenguajes de programación nos permiten crear programas computacionales que controlen el comportamiento físico y lógico de una máquina con el fin de automatizar o facilitar la ejecución de tareas específicas. (Martínez Ladrón de Guevara, s.f.).

Una clasificación breve de los lenguajes de programación podría ser: los lenguajes de bajo nivel, los cuales son utilizados para controlar todas las partes de los equipos electrónicos y que dependen directamente del tipo de máquina; los lenguajes de alto nivel, que son usados para crear programas computacionales y que no dependen directamente del tipo de máquina (Morales, 2014).

Dentro de los lenguajes de alto nivel podemos encontrar algunos como C++, PHP, Java, entre otros. En el presente trabajo se usará el lenguaje de programación Java, por lo cual a continuación se presenta una breve introducción al mismo:

Cuando Bill Joy y su equipo estaban inmersos en el desarrollo del proyecto "GREEN"² se dieron cuenta de las limitaciones de trabajar con el lenguaje C++ pues el mismo mostraba problemas de incompatibilidad con diferentes arquitecturas físicas³, así como el nivel de la adaptación de la interfaz gráfica de las aplicaciones y de la interconexión entre los diferentes dispositivos. Debido a esto se vio necesario crear un nuevo lenguaje de programación, alrededor de una nueva plataforma de desarrollo. (Groussard, 2012)

A cargo de la creación de este nuevo lenguaje estuvieron dos desarrolladores, de SUN MICROSYSTEM, James Gosling y Patrick Naughton quienes basándose en las características de otros lenguajes⁴ obtuvieron un producto para el desarrollo de aplicaciones seguras, distribuidas, y portables; el cual para 1991 sería bautizado, luego de varios apodos⁵, como Java en referencia a los muchas tazas de café tomadas por los programadores. (Groussard, 2012)

Según Groussard (2012) "SUN caracteriza a Java como un lenguaje sencillo, orientado a objetos, distribuido, interpretado, robusto, securizado, independiente de las arquitecturas, portable, eficaz, multihilo, y dinámico" (Pág. 12).

² Desarrollo de aplicaciones destinadas a una amplia variedad de periféricos y sistemas transportables.

³ Procesadores, tamaños de memoria, etc.

⁴ C++, Eiffel, SmallTalk, Objective C, Cedar/Mesa, Ada, Perl.

⁵ C++-- (C++ sin defectos); OAK.

A continuación se muestra una tabla detallando cada una de las características antes mencionadas sobre Java:

CARACTERÍSTICA	DESCRIPCIÓN
SENCILLO	<ul style="list-style-type: none"> • Solo existen tres tipos de primitivos: Los numérico (enteros y reales), el tipo carácter y el tipo booleano. • El programador no debe preocuparse por la gestión de memoria (“Garbage Collector”). • Solo se autoriza la herencia simple pero aporta un mecanismo de simulación de herencia múltiple mediante la implementación de interfaces.
DISTRIBUIDO	<ul style="list-style-type: none"> • Se implementan los principales protocolos de acceso de red (FTP, HTTP). • Se puede configurar el JRE con el fin de controlar el acceso a la red de sus aplicaciones.
INDEPENDIENTE DE ARQUITECTURAS	<ul style="list-style-type: none"> • El compilador genera bytecode⁶ que es independiente de cualquier arquitectura, de todo sistema operativo y de todo dispositivo de gestión de la GUI⁷. • Solo es necesario disponer de la máquina virtual específica de su plataforma para hacer funcionar un programa Java.
	<ul style="list-style-type: none"> • Es portable pues se trata de un

⁶ Lenguaje binario intermedio.

⁷ Interfaz Gráfica de Usuario.

CARACTERÍSTICA	DESCRIPCIÓN
PORTABLE	<p>lenguaje interpretado.</p> <ul style="list-style-type: none"> Los datos primitivos⁸ de Java tienen el mismo tamaño, sea cual sea la plataforma de ejecución.
EFICAZ	<ul style="list-style-type: none"> Java pone en marcha un proceso de optimización de la interpretación de sus programas conocido como JIT (Just In Time). El cual compila el bytecode de Java en código nativo en tiempo de ejecución con el fin de alcanzar igual rendimiento que un programa escrito en C o C++.
MULTITAREA	<ul style="list-style-type: none"> Se permite la ejecución simultánea de varias tareas con el fin de aumentar la velocidad de las aplicaciones ya sea compartiendo el tiempo del CPU, o repartiendo las tareas entre varios procesadores.
DINÁMICO	<ul style="list-style-type: none"> Es posible editar una o varias clases sin tener que efectuar una actualización de estas modificaciones en todo el conjunto del programa. La comprobación de existencia de las clases solo se hace en el momento de la compilación y la llamada al código, solo en la ejecución del programa. Logrando con esto aplicaciones más ligeras en tamaño de memoria.
<p>FUENTES: (Groussard, 2012) ELABORADO POR: Pablo Herrera, 2016</p>	

Tabla 2 CARACTERÍSTICAS DE JAVA

2.1.2. LENGUAJES DE MODELAMIENTO UML

Para poder iniciar el entendimiento de UML debemos tener claro lo que es el modelado de software. Este concepto puede ser entendido, según el Object Management Group (2015), como: “el diseño, previo a la codificación, de una aplicación de software”.

⁸ Numérico, carácter y booleano.

Este modelamiento de software resulta una parte esencial en proyectos grandes de desarrollo de software pero también resulta una herramienta útil en proyectos medianos y pequeños ya que el mismo juega un rol analítico en el desarrollo. Además, el modelamiento es un mecanismo que, los proyectos exitosos usan, para asegurarse de que: las funcionalidades del negocio estén completas y correctas, de que las necesidades de los usuarios sean cubiertas, que el diseño del programa soporte la escalabilidad⁹, seguridad, robustez, entre otras características que minimizan la necesidad de correcciones, que en la mayoría de casos resulta muy costos después de la implementación. (Object Management Group , 2015).

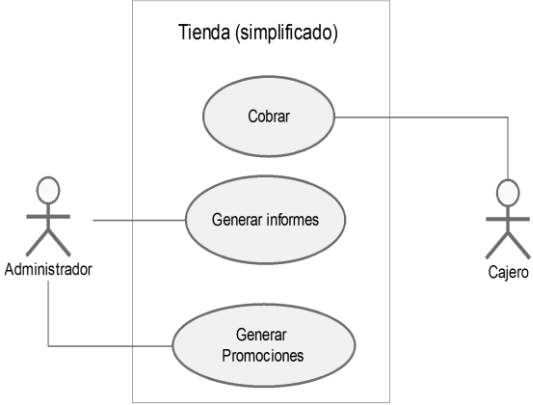
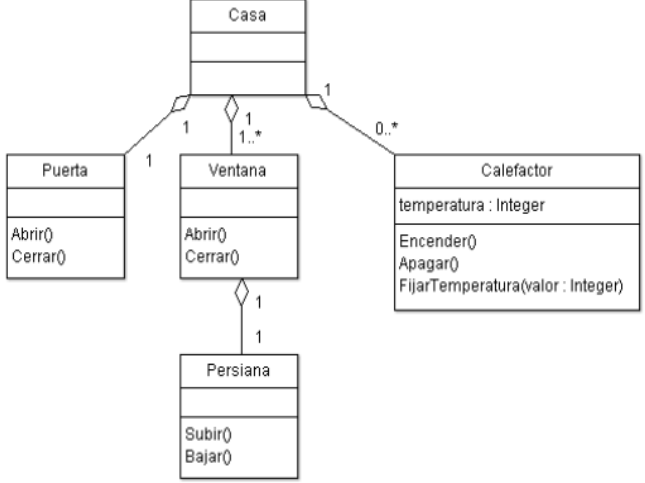
El Lenguaje de Modelado Unificado o UML, por sus siglas en Inglés, nos permite especificar, visualizar y documentar modelos de sistemas de software, incluyendo su estructura y diseño, de una manera que reúna todos los requerimientos (Object Management Group , 2015).

UML no es un método de desarrollo de software, lo que implica que no sirve para determinar sistemáticamente que pasos seguir sino que nos ayuda a visualizar el diseño y así hacerlo más asequible para otros (Umbrello UML Modeller Autores, 2013).

Como UML se compone de muchos elementos que representan alguna parte o punto de vista del proyecto. A continuación se muestra una tabla con los principales diagramas UML.

NOMBRE	DESCRIPCIÓN	EJEMPLO
---------------	--------------------	----------------

⁹ Capacidad de manejar crecimiento continuo sin perder su calidad en los servicios.

NOMBRE	DESCRIPCIÓN	EJEMPLO
<p>DIAGRAMA DE CASOS DE USO</p>	<ul style="list-style-type: none"> Muestra a los actores¹⁰, las situaciones que se producen cuando utilizan el sistema y sus relaciones. Nos ayudan a facilitar la comunicación con los futuros usuarios del sistema. Describen qué hace el sistema mas no el cómo. 	
<p>DIAGRAMA DE CLASES</p>	<ul style="list-style-type: none"> Muestra las diferentes clases que componen un sistema y como estas se relacionan. Muestra los atributos y métodos de las clases. 	

¹⁰ Otros usuarios del Sistema.

NOMBRE	DESCRIPCIÓN	EJEMPLO
<p>DIAGRAMA DE SECUENCIA</p>	<ul style="list-style-type: none"> Nos ayudan a visualizar el intercambio de mensajes en un momento dado. El eje del tiempo está situado de forma vertical donde los mensajes son intercambiados entre objetos mediante el uso de flechas horizontales con los nombres de las operaciones y parámetros. 	<p>INSERTAR VIAJE</p> <pre> sequenceDiagram actor Admin as ADMINISTRADOR (<CASOS_DE_USO>) participant GuiManejoViaje as :GuiManejoViaje participant ViajeDP as :ViajeDP participant ViajeMD as :ViajeMD Admin->>GuiManejoViaje: SELECCIONA ADMINISTRAR VIAJE activate GuiManejoViaje GuiManejoViaje->>GuiManejoViaje: GuiManejoViaje() deactivate GuiManejoViaje Admin->>GuiManejoViaje: INGRESA EL CODIGO DEL VIAJE activate GuiManejoViaje GuiManejoViaje->>ViajeDP: setCodigoViaje() activate ViajeDP ViajeDP->>ViajeDP: validarCodigo() ViajeDP->>ViajeMD: consultarPorCodigo() activate ViajeMD ViajeMD-->>ViajeDP: RETORNA EL OBJETO deactivate ViajeMD deactivate ViajeDP GuiManejoViaje->>GuiManejoViaje: cargarDatos() deactivate GuiManejoViaje Admin->>GuiManejoViaje: INGRESA OTROS DATOS activate GuiManejoViaje Admin->>GuiManejoViaje: PRESIONA BOTON INSERTAR activate GuiManejoViaje GuiManejoViaje->>ViajeDP: ingresarViajeDP() activate ViajeDP ViajeDP->>ViajeMD: insertarViajeMD() activate ViajeMD deactivate ViajeMD deactivate ViajeDP deactivate GuiManejoViaje </pre>
<p>DIAGRAMA DE ESTADO</p>	<ul style="list-style-type: none"> Muestra los diferentes estados de un objeto durante toda su vida así como los estímulos que provocan tales cambios. 	<p>CAJERO AUTOMÁTICO.</p> <pre> stateDiagram-v2 [*] --> LIBRE LIBRE --> LIBRE: Retirar Tarjeta LIBRE --> OCUPADO: Inserta Tarjeta LIBRE --> OCUPADO: Time out OCUPADO --> LIBRE: Retirar Tarjeta OCUPADO --> OCUPADO: verificar datos OCUPADO --> Procesando Transac: verificar datos Procesando Transac --> LIBRE: cancelar Procesando Transac --> OCUPADO: Error Datos Procesando Transac --> Transaccion Correcta: confirmacion Transaccion Correcta --> LIBRE: Retiro Tarjeta </pre>

NOMBRE	DESCRIPCIÓN	EJEMPLO
<p>DIAGRAMA DE ACTIVIDAD</p>	<ul style="list-style-type: none"> • Describen secuencialmente las actividades en un sistema. • Las actividades están unidas claramente a un objeto. 	<pre> graph TD Inicio((Inicio)) --> A[COMPROBACIÓN Verifica si el espacio ya existe] A -- "si no existe" --> B[Se le asigna la locación] A -- "si existe" --> C[Verifica cuál es la locación a la que corresponde] C --> D[Constata si los datos están cargados] D -- "si no están cargados" --> E[Carga la información] D -- "si están cargados" --> Fin((Fin)) B --> E </pre>
<p>DIAGRAMA DE PAQUETES</p>	<ul style="list-style-type: none"> • Provee una visión macro de la agrupación de las clases y sus relaciones. 	<pre> classDiagram package ComidaRapida["Comida Rapida (Logica de Negocio)"] { ProcesosPedido ProcesosRepartidor ProcesosMercadeo } package ComidaRapidaModeloDatos["Comida Rapida (Modelo de Datos)"] { PedidoComida PedidoCaja PedidoTelefonico Repartidor Combo Producto } PedidoComida < -- PedidoCaja PedidoComida < -- PedidoTelefonico PedidoComida o-- "0..*" Combo PedidoComida o-- "0..*" Producto Repartidor "1" -- "0..*" PedidoTelefonico Combo o-- "1..*" Producto </pre>

NOMBRE	DESCRIPCIÓN	EJEMPLO
<p>DIAGRAMA DE DESPLIEGUE</p>	<ul style="list-style-type: none"> Este tipo de diagramas muestra la topología sobre la cual se ejecuta el sistema informático. 	<pre> graph TD User((User)) --- Keyboard[Keyboard/monitor] --- Workstation[Workstation] subgraph Workstation WebBrowser[Web browser] end subgraph WebServer[Web Server] Presentation[Presentation layer (web interface)] DatabaseInterface[Database interface] LogFile[Log file] Presentation <--> DatabaseInterface DatabaseInterface --- LogFile end subgraph DatabaseServer[Database Server] MySQL[MySQL database] end WebBrowser -- HTTP/HTTPS connection --- Presentation DatabaseInterface -- TCP/IP or local socket --- MySQL </pre>
<p>FUENTES: (Umbrello UML Modeller Autores, 2013) ELABORADO POR: Pablo Herrera, 2016</p>		

Tabla 3 PRINCIPALES DIAGRAMAS DE UML

2.1.3. POSTGRESQL

Según Nevado V. (2010) un sistema gestor de base de datos (SGBD) puede ser visto como un sistema de software que permite:

- La definición de los datos tanto a nivel físico como a nivel lógico y externo
- Insertar, modificar, eliminar y consultar los datos; es decir la manipulación de los datos contenidos en una base de datos
- Garantiza la integridad de los datos
- Controla el acceso y la seguridad de los datos contenidos en las bases de datos

Según los desarrolladores de PostgreSQL (s.f.), en su página web oficial, se puede ver a PostgreSQL como un poderoso sistema gestor de base de datos objeto-

relacional¹¹ cuya distribución se hace bajo los lineamientos de la licencia BSD¹². Sus casi 16 años de activo desarrollo le han permitido ganarse una fuerte reputación por confiabilidad, integridad de datos, estabilidad, potencia, administración sencilla, entre otros. Además, este gestor corre en todos los principales sistemas operativos¹³ (PostgreSQL Global Development Group, s.f.).

Actualmente, Enero del 2016, la última versión que se maneja de PostgreSQL es 9.5.0 disponible, en el sitio web oficial¹⁴, para Windows, Linux y Mac OS X.

A continuación se presenta una tabla con las principales características de PostgreSQL:

TIPO	CARACTERÍSTICA
GENERALES	<ul style="list-style-type: none">• Es un sistema gestor de bases de datos 100% ACID¹⁵, por sus siglas en Inglés, Atomicity, Consistency, Isolation and Durability• Integridad referencial• Copias de seguridad en caliente¹⁶• Acceso encriptado vía SSL¹⁷

¹¹ Es una extensión de la base de datos relacional tradicional, a la cual se le proporcionan características de la programación orientada a objetos.

¹² Licencia Berkeley Software Distribution, para más información visitar: <https://www.gnu.org/licenses/license-list.html#OriginalBSD>

¹³ UNIX, Windows y Linux.

¹⁴ <http://www.enterprisedb.com/products-services-training/pgdownload>

¹⁵ **A**tomicidad, **C**onsistencia, **A**islamiento y **D**urabilidad.

¹⁶ Respaldo cuando aún los datos están accesibles para el usuario.

¹⁷ *Secure Sockets Layer*, capa de conexión segura.

TIPO	CARACTERÍSTICA
DESARROLLO	<ul style="list-style-type: none"> • Funciones y procedimientos almacenados en varios lenguajes de programación • Posibilidad para definir nuevos tipos de datos • APIs¹⁸ para programación
SQL	<ul style="list-style-type: none"> • Llaves primarias y foráneas, columnas autoincrementales, índices compuestos • Sub consultas, Joins, Vistas • Herencia de tablas
LÍMITES	<ul style="list-style-type: none"> • Tamaño máximo de tabla, 32 TB • Tamaño máximo de fila, 1.6 TB • Tamaños máximo de campo, 1 GB • Ilimitado número de filas por tabla • Número máximo de columnas por tabla, 250 a 1600 dependiendo del tipo • Número ilimitado de índices por tabla
<p>FUENTES: (Martinez, 2010) ELABORADO POR: Pablo Herrera, 2016</p>	

Tabla 4 PRINCIPALES CARACTERÍSTICAS DE POSTGRESQL

2.1.4. IDE ECLIPSE

¹⁸ Application Programming Interface, funciones de uso general para abstraer la programación.

Una plataforma para el desarrollo de software puede ser visto como un entorno común en el cual se desenvuelve la construcción de un grupo definido de aplicaciones informáticas. Además, se debe entender que al definir una plataforma de desarrollo de software estamos definiendo también la arquitectura, el sistema operativo, lenguaje de programación, con los cuales funcionara esta plataforma (Gallardo, 2012).

Para Gallardo (2012), consultor independiente de software y escritor de artículos técnicos en español en la página de IBM, el IDE¹⁹ Eclipse puede ser visto con una plataforma de desarrollo de aplicaciones informáticas de código abierto basada en Java (Gallardo, 2012).

Mientras la mayoría de usuarios utilizan Eclipse como un IDE de Java, este también puede ser usado para el desarrollo de aplicaciones en diferentes lenguajes de programación como C/C++ y COBOL, mediante el uso de complementos (Gallardo, 2012).

Para la fecha donde se realiza este trabajo de disertación, la versión más actual del IDE Eclipse es 4.5.1 disponible para Windows, Linux y Mac OS X.

2.1.5. SERVIDOR DE APLICACIONES JAVA EE, REDHAT JBOSS

Un servidor de aplicaciones puede ser visto como “una herramienta basada en componentes que residen en la capa intermedia, de la arquitectura de tres capas, que permite construir, desplegar y administrar aplicaciones de una forma ordenada” (Almeida Moyano & Imbacuán López, 2009).

JBoss Application Server o Jboss AS puede ser visto un proyecto de código abierto basado en J2EE²⁰ e implementado netamente en Java. Este proyecto nos permite el desarrollo y despliegue de aplicaciones web que involucran el acceso a bases de datos. (Almeida Moyano & Imbacuán López, 2009)

En Abril del 2006 la compañía RedHat adquiere el proyecto JBoss por una suma de 350 millones de dólares más 70 millones si el desempeño de JBoss es satisfactorio (SG Software Guru, 2006).

Debido a las posibles confusiones que podrían surgir RedHat decide usar el nombre JBoss solo para las versiones de pago y es por esta razón que surge Wildfly (proyecto comunitario JBoss). El cual dispone actualmente de la versión 10 de su servidor de aplicaciones. Cabe aclarar que la última versión de JBoss AS, la versión 7, está aún disponible para la descarga pero no tiene ningún tipo de soporte técnico.

Según Almeida Moyano & Imbacuán López (2009) Entre las principales características de JBoss 7 podemos citar lo siguiente:

¹⁹ Integrated Development Environment o entorno de desarrollo integrado.

²⁰ Java 2 Platform, Enterprise Edition.

- Implementa y ejecuta aplicaciones J2EE
- Es de código abierto
- Es de distribución libre
- Soporta IP V6
- Soporta servidores virtuales
- Capacidad para EJB 3.0
- Soporte de conexión para IDE eclipse

2.2. METODOLOGÍA PARA EL DESARROLLO DEL PRODUCTO

Cuando se habla de una **metodología de desarrollo de software** estamos hablando, según Cendejas V. (2014), “de un enfoque estructurado que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseños y guías de procesos”.

Es decir que una metodología de software se refiere a un marco de trabajo que nos ayuda a estructurar, planificar y controlar los procesos en el desarrollo de un sistema informático.

2.2.1. EXTREME PROGRAMMING

2.2.1.1. INTRODUCCIÓN

Para Beck (2012), Extreme Programming puede ser visto como “un estilo de desarrollo de software enfocado en la aplicación de excelentes técnicas de programación, comunicación clara y trabajo en equipo lo cual nos permite lograr cosas que anteriormente no podíamos ni imaginar” (Pág. 2).

Extreme Programming o XP, surge como una respuesta a las metodologías pesadas que necesitaban documentación y elementos que muchas veces resultaban innecesarios. Tiene su origen en los años 90 donde es usada por Kent Beck para un proyecto denominado C3²¹ para la firma Chrysler, en marzo de 1996 (Navarrete Jara, 2013).

XP está basada en cuatro valores fundamentales para un proyecto exitoso:

1. **Comunicación.** Comunicación permanente entre todas las partes interesadas incluyendo al cliente, desarrolladores y a los coordinadores o gerentes con el fin de resolver dudas y establecer prioridades para que de esta manera se pueda ajustar la agenda del avance del proyecto (Navarrete Jara, 2013).
2. **Simplicidad.** XP exige simplicidad en el diseño, en el código, en los procesos, etc. Con la finalidad de que todos puedan comprender y mejorar el trabajo (Navarrete Jara, 2013).

²¹ Chrysler Cromprehensive Compensation.

3. **Retroalimentación.** La retroalimentación de todas las partes es fundamental para que se pueda ajustar las funcionalidades a las necesidades reales del cliente así como también permite hallar posible errores de una manera más rápida (Navarrete Jara, 2013).
4. **Coraje.** Implica la habilidad para tomar decisiones duras, esto quiere decir, que sin importar el tiempo que nos demoremos en tarea si esta esta defectuosa corregirla sin importar si es necesario repetirla (Navarrete Jara, 2013).

2.2.1.2. CICLO DE VIDA

Entiéndase como ciclo de vida del software a la sucesión de etapas por las que un desarrollo de software debe atravesar comenzando desde que la idea del proyecto es concebida hasta que se finaliza su uso. Dentro de estas etapas se incluirá una serie de actividades que tienen el fin de construir, brindar mantenimiento y documentar el producto informático final (Instituto Nacional de Tecnologías de las Comunicaciones, 2009).

2.2.1.2.1. EXPLORACIÓN

En esta etapa el cliente define las historias de usuario con el fin de que los desarrolladores puedan estimar tiempos de desarrollo para cada una de las entregas (Navarrete Jara, 2013).

Las historias de usuario pueden ser entendidas como las diagramas de casos de uso pero se diferencian en que con ellas se pueden crear estimaciones de tiempo como datos de entrada para la reunión de planificación de lanzamientos (Wells, 2009).

2.2.1.2.2. PLANIFICACIÓN

Es una fase de corta duración donde el cliente junto con los gerentes y equipo de desarrollo establecen prioridades de implementación de las historias de usuario. Esta fase consiste en una o varias reuniones (Navarrete Jara, 2013).

2.2.1.2.3. ITERACIONES

En esta etapa se desarrollan las funcionalidades del sistema generando un entregable funcional al cabo de cada iteración. Además, estas iteraciones facilitan medir el avance general del proyecto (Navarrete Jara, 2013).

En el presente trabajo de disertación el entregable constara de una breve vista a los diseños de interfaces gráficas de usuario, el modelo de datos y un extracto del código fuente.

2.2.1.2.4. PRODUCCIÓN

La etapa de producción se da cuando hay una iteración sin errores, para la cual se deben implementar pruebas adicionales antes de que el sistema sea trasladado al entorno del cliente (Wells, 2009).

2.2.1.2.5. MANTENIMIENTO

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para lo cual se requiere tareas de soporte al cliente (Wells, 2009).

2.2.1.2.6. MUERTE DEL PROYECTO

Se da cuando el cliente no tiene más historias que deban ser incluidas. Es entonces cuando se genera el documento final del sistema y no se realizan más cambios en la arquitectura. Cabe destacar que esto también puede suceder cuando el proyecto no genera los beneficios esperados o no hay presupuesto para mantener el mismo (Wells, 2009).

2.2.1.3. REGLAS Y PRÁCTICAS

A continuación se presenta un cuadro con las recomendaciones de las reglas y prácticas de XP:

GRUPO	CONCEPTO	EXPLICACIÓN
	<p>Historias usuario</p>	<ul style="list-style-type: none"> • Sustituye el documento de especificación funcional, son escritas por el cliente. • Deben tener el mínimo detalle para que los desarrolladores puedan realizar una estimación de tiempo.

GRUPO	CONCEPTO	EXPLICACIÓN
PLANIFICACIÓN	Release Plan	<ul style="list-style-type: none"> Establece que historias de usuario se agrupan por cada entrega y además el orden de las mismas.
	Iteration Plan	<ul style="list-style-type: none"> Se debe establecer pruebas de aceptación para cada historia de usuario.
	Stand-Up Meeting	<ul style="list-style-type: none"> Reuniones muy cortas cuyo objetivo es mantener la comunicación entre el equipo y compartir problemas y soluciones.
PLANIFICACIÓN	Simplicidad	<ul style="list-style-type: none"> Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración.
	Spike	<ul style="list-style-type: none"> Se refiere al uso de pequeños programas de prueba con el fin de mitigar los problemas que surgen cuando es difícil estimar tiempos o

GRUPO	CONCEPTO	EXPLICACIÓN
DISEÑO		aparecen problemas técnicos.
	Refactoring	<ul style="list-style-type: none"> Se refiere a recodificar parte del código con el fin de simplificar el programa sin cambiar su funcionalidad.
	Metáfora del Sistema	<ul style="list-style-type: none"> Se crea la arquitectura del sistema para que todo el grupo de desarrollo tenga una misma visión del proyecto.
	Tarjeta CRC	<ul style="list-style-type: none"> Se divide en tres partes en las cuales se escribe la clase, las responsabilidades y el colaborador que es otra clase con la que interactúa para cumplir su o sus objetivos funcionales.
DISEÑO	Disponibilidad del cliente	<ul style="list-style-type: none"> El cliente debe estar disponible durante todo el proyecto.
	Uso de estándares	<ul style="list-style-type: none"> Facilita la recodificación y hace al código más entendible para todo el equipo.
	Programación dirigida por las pruebas	<ul style="list-style-type: none"> Se recomienda primero escribir las pruebas que el sistema debe pasar.
	40 hora semanales de	<ul style="list-style-type: none"> No sobrecargar el trabajo

GRUPO	CONCEPTO	EXPLICACIÓN
	trabajo	del equipo.
FUENTES: (Navarrete Jara, 2013) ELABORADO POR: Pablo Herrera, 2016.		

Tabla 5 PRINCIPALES REGLAS Y PRACTICAS DE XP

CAPÍTULO: III

DESARROLLO DEL PRODUCTO

En el presente capítulo se aborda propiamente el problema a solventar dentro del proyecto informático. Primero, se inicia con un breve análisis de los principales procesos que engloban el funcionamiento de la fundación. Segundo, se plasma las diferentes funcionalidades, que el sistema informático deberá contener, mediante el uso de historias de usuario. Tercero, se realiza un análisis para la división de las tres entregas de las cuales estará conformado el proyecto, asignando a cada entrega las funcionalidades requeridas siempre teniendo en cuenta el nivel de complejidad de la misma. Cuarto, se presenta una breve descripción de los diferentes entregables dentro de cada iteración. Quinto, se presenta un informe para la implantación del sistema así como un esquema para la validación de las distintas funcionalidades del proyecto informático. Finalmente, se presenta tres manuales desarrollados con la finalidad de ubicar de mejor manera al lector dentro de la lógica del funcionamiento del sistema.

3.1. EXPLORACIÓN

3.1.1. RESUMEN DEL PROCESO DE ADMINISTRACIÓN DEL FINANCIAMIENTO

La Fundación Niñez y Vida Tierra de Hombres Ecuador funciona mediante los fondos o donaciones que realizan terceros a la misma. Actualmente la fundación maneja dos cuentas: una para todos los aportes provenientes de todas los donantes ya sean entidades naturales o jurídicas, públicas o privadas; y una especial para los fondos provenientes de Suiza (T.D.H. Suiza y F.E.D.E.V.A.C.O.) los mismos que deben ser tratados de una manera particular sin ser mezclados con las otras donaciones.

Estos fondos serán destinados tanto para el financiamiento de diferentes proyectos que dirige la fundación como para el funcionamiento del departamento administrativo y financiero de la misma.

Actualmente los fondos son destinados para el funcionamiento de:

- Proyecto de Salud Materno Infantil.
- Proyecto CECAPI.
- Departamento administrativo y financiero.
- Proyecto refugio de los sueños.

Resulta necesario puntualizar que para el pago de los sueldos y gastos de funcionamiento que tiene la fundación, el departamento financiero debe adaptar al presupuesto entregado por los donantes, motivo por el cual muchas veces el pago de un sueldo o gasto se divide para que pueda ser cubierto, en su totalidad o parte de este, por la cuenta de las donaciones de Suiza o por la cuenta de Otras donaciones.

Por lo anteriormente dicho, no debe resultar raro que una misma persona tenga dos roles de pago; uno en el proyecto de CECAPI y Refugio de los sueños y otro en el proyecto de Salud Materno Infantil. Con el fin de que el rubro total del sueldo o gasto sea cubierto.

3.1.2. DEFINICIÓN DEL PROBLEMA

Para llevar el control de los fondos asignados por los donantes en el departamento financiero de manejan archivos de Microsoft Excel, en los cuales se registra lo siguiente:

PARA ROLES DE PAGO:

- ROL REFUGIO y CECAPI - (MES).xlsx

Este archivo contiene registro de los roles de pago de los sueldos de los empleados de los proyectos de CECAPI, Refugio de los sueños y empleados del departamento administrativo y financiero.

- ROL SALUD – (MES).xlsx

Este archivo contiene registro de los roles de pago de los sueldos de los empleados de los proyecto de Salud Materno Infantil y empleados del departamento administrativo y financiero.

Mediante la información contenida en los archivos citados anteriormente se logra generar los siguientes informes:

- a) Rol de pagos individual
- b) Rol de pagos consolidado individual
- c) Rol de pagos consolidado general

PROBLEMA:

Al manejar archivos de Microsoft Excel los empleados del departamento financiero se ven obligados a crear un archivo para cada mes del año y actualizar los campos que

necesiten ser actualizados lo cual resulta en un problema cuando se quiere buscar información específica; pues para realizar las búsquedas primero se debe ubicar el archivo y luego ubicar la hoja para después ubicar el registro deseado.

3.1.3. HISTORIA DE USUARIO

Para el desarrollo de las Historias de Usuario (HU) se usará la siguiente plantilla:

HISTORIA DE USUARIO	
NÚMERO:	USUARIO:
NOMBRE DE HISTORIA:	
PRIORIDAD EN NEGOCIO: (Alta, Media, Baja)	RIESGO EN DESARROLLO: (Alta, Media, Baja)
PUNTOS ESTIMADOS.	ITERACIÓN ASIGNADA:
DESCRIPCIÓN:	
OBSERVACIÓN:	

Tabla 6 PLANTILLA PARA HISTORIAS DE USUARIO

Las Historias de Usuario se dividen en los siguientes módulos:

1. F0: Administrar Auspiciantes

HISTORIA DE USUARIO	
NÚMERO: 01	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Auspiciante	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite administrar los Auspiciantes de los proyectos de la fundación	
OBSERVACIÓN:	

Tabla 7 ADMINISTRAR AUSPICIANTE

1.1. Ingresar Auspiciante

HISTORIA DE USUARIO	
NÚMERO: 02	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Ingresar Auspiciante	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite ingresar Auspiciantes de los proyectos de la fundación	
OBSERVACIÓN:	

Tabla 8 INGRESAR AUSPICIANTE

1.2. Actualizar Auspiciante

HISTORIA DE USUARIO	
NÚMERO: 03	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Actualizar Auspiciante	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite modificar los datos de los Auspiciantes registrados	
OBSERVACIÓN:	

Tabla 9 ACTUALIZAR AUSPICIANTE

1.3. Eliminar Auspiciante

HISTORIA DE USUARIO	
NÚMERO: 04	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Eliminar Auspiciante	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite eliminar los Auspiciantes registrados	
OBSERVACIÓN:	

Tabla 10 ELIMINAR AUSPICIANTE

1.4. Consultar Auspiciante

HISTORIA DE USUARIO

NÚMERO: 05	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Consultar Auspiciante	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite consultar los datos de los Auspiciantes registrados	
OBSERVACIÓN: Se debe tener una consulta general a la cual se le aplicará filtros para la selección de registros.	

Tabla 11 CONSULTAR AUSPICIANTE

2. F1: Administrar Roles

HISTORIA DE USUARIO	
NÚMERO: 06	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Roles	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Alta
ITERACIÓN ASIGNADA: Segunda	
DESCRIPCIÓN: Permite administrar la información de los Roles de pagos de todos los proyectos administrados por la fundación.	
OBSERVACIÓN:	

Tabla 12 ADMINISTRAR ROLES

2.1. Ingresar Roles

HISTORIA DE USUARIO	
NÚMERO: 07	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Ingresar Roles	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Alta
ITERACIÓN ASIGNADA: Segunda	
DESCRIPCIÓN: Permite ingresar Roles de pagos de todos los proyectos administrados por la fundación.	
OBSERVACIÓN:	

Tabla 13 INGRESAR ROLES

2.2. Eliminar Roles

HISTORIA DE USUARIO	
NÚMERO: 08	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Eliminar Roles	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Alta
ITERACIÓN ASIGNADA: Segunda	
DESCRIPCIÓN: Permite eliminar los roles de pago registrados	
OBSERVACIÓN: Al eliminar un Rol de debe ingresar un ajuste para que el presupuesto no se vea afectado.	

Tabla 14 ELIMINAR ROLES

2.3. Consultar Roles

HISTORIA DE USUARIO	
NÚMERO: 09	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Consultar Roles	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Alta
ITERACIÓN ASIGNADA: Segunda	
DESCRIPCIÓN: Permite consultar los datos de los Roles de pago registrados	
OBSERVACIÓN: Se debe tener una consulta general a la cual se le aplicará filtros para la selección de registros.	

Tabla 15 CONSULTAR ROLES

3. F2: Administrar Proyectos

HISTORIA DE USUARIO	
NÚMERO: 10	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Proyectos	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite administrar los Proyectos administrados por la fundación	
OBSERVACIÓN:	

Tabla 16 ADMINISTRAR PROYECTOS

3.1. Ingresar Proyectos

HISTORIA DE USUARIO	
NÚMERO: 11	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Ingresar Proyectos	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite ingresar Proyectos administrados por la fundación	
OBSERVACIÓN:	

Tabla 17 INGRESAR PROYECTOS

3.2. Actualizar Proyectos

HISTORIA DE USUARIO	
NÚMERO: 12	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Actualizar Proyectos	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite actualizar la información de los Proyectos administrados por la fundación	
OBSERVACIÓN:	

Tabla 18 ACTUALIZAR PROYECTOS

3.3. Eliminar Proyectos

HISTORIA DE USUARIO	
NÚMERO: 13	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Eliminar Proyectos	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite eliminar los Proyectos registrados	
OBSERVACIÓN:	

Tabla 19 ELIMINAR PROYECTOS

3.4. Consultar Proyectos

HISTORIA DE USUARIO	
NÚMERO: 14	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Consultar Roles	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite consultar los datos de los Proyectos registrados	
OBSERVACIÓN: Se debe tener una consulta general a la cual se le aplicará filtros para la selección de registros	

Tabla 20 CONSULTAR ROLES

4. F3: Administrar Empleados

HISTORIA DE USUARIO	
NÚMERO: 15	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Empleados	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite administrar los Empleados que trabajan en la fundación	
OBSERVACIÓN:	

Tabla 21 ADMINISTRAR EMPLEADOS

4.1. Ingresar Empleados

HISTORIA DE USUARIO	
NÚMERO: 16	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Ingresar Empleados	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite ingresar Empleados	
OBSERVACIÓN:	

Tabla 22 INGRESAR EMPLEADOS

4.2. Actualizar Empleados

HISTORIA DE USUARIO	
NÚMERO: 17	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Actualizar Empleados	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite actualizar la información de los Empleados registrados	
OBSERVACIÓN:	

Tabla 23 ACTUALIZAR EMPLEADOS

4.3. Eliminar Empleados

HISTORIA DE USUARIO	
NÚMERO: 18	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Eliminar Empleados	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite eliminar los Empleados registrados	
OBSERVACIÓN:	

Tabla 24 ELIMINAR EMPLEADOS

4.4. Consultar Empleados

HISTORIA DE USUARIO	
NÚMERO: 19	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Consultar Empleados	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite consultar los datos de los Empleados registrados	
OBSERVACIÓN: Se debe tener una consulta general a la cual se le aplicará filtros para la selección de registros.	

Tabla 25 CONSULTAR EMPLEADOS

5. F4: Administrar Cuentas Bancarias

HISTORIA DE USUARIO	
NÚMERO: 20	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Cuentas Bancarias	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite administrar los datos de las Cuentas Bancarias con las que trabaja la fundación	
OBSERVACIÓN: Administrar los datos de las cuentas bancarias se entenderá como el registro de la información necesaria para realizar los informes de ejecución de presupuesto así como para facilitar a los empleados del departamento contable la asignación de la carga financiera en el proceso de registro de Roles de pago y Registro de Gastos. Es necesario recalcar que esta función no busca remplazar o emular las actividades de una entidad financiera propiamente conocida como Banco, solamente busca facilitar el control de la ejecución del presupuesto de la Fundación Niñez y Vida Tierra de Hombres Ecuador.	

Tabla 26 ADMINISTRAR CUENTAS BANCARIAS

5.1. Ingresar Cuentas Bancarias

HISTORIA DE USUARIO	
NÚMERO: 21	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Ingresar Cuentas Bancarias	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite registrar los datos de las cuentas bancarias con las que trabaja la fundación	
OBSERVACIÓN:	

Tabla 27 INGRESAR CUENTAS BANCARIAS

5.2. Actualizar Cuentas Bancarias

HISTORIA DE USUARIO	
NÚMERO: 22	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Actualizar Cuentas Bancarias	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite actualizar la información de las Cuentas Bancarias registradas	
OBSERVACIÓN:	

Tabla 28 ACTUALIZAR CUENTAS BANCARIAS

5.3. Eliminar Cuentas Bancarias

HISTORIA DE USUARIO	
NÚMERO: 23	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Eliminar Cuentas Bancarias	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite eliminar las Cuentas Bancarias registradas	
OBSERVACIÓN:	

Tabla 29 ELIMINAR CUENTAS BANCARIAS

5.4. Consultar Cuentas Bancarias

HISTORIA DE USUARIO	
NÚMERO: 24	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Consultar Empleados	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite consultar los datos de las Cuentas Bancarias registradas	
OBSERVACIÓN: Se debe tener una consulta general a la cual se le aplicará filtros para la selección de registros	

Tabla 30 CONSULTAR EMPLEADOS

6. F5: Administrar Gastos

HISTORIA DE USUARIO	
NÚMERO: 25	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Gastos	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite administrar la información de los gastos producidos en todos los proyectos administrados por la fundación	
OBSERVACIÓN:	

Tabla 31 ADMINISTRAR GASTOS

6.1. Ingresar Gastos

HISTORIA DE USUARIO	
NÚMERO: 26	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Ingresar Gastos	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite ingresar Gatos realizados en todos los proyectos administrados por la fundación	
OBSERVACIÓN:	

Tabla 32 INGRESAR GASTOS

6.2. Eliminar Gastos

HISTORIA DE USUARIO	
NÚMERO: 27	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Eliminar Gastos	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Alta
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite eliminar los gastos registrados	
OBSERVACIÓN: Al eliminar registro de un gasto se debe ingresar un ajuste para que el presupuesto no se vea afectado.	

Tabla 33 ELIMINAR GASTOS

6.3. Consultar Gastos

HISTORIA DE USUARIO	
NÚMERO: 28	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Consultar Gastos	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Baja
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite consultar los datos de los gastos registrados	
OBSERVACIÓN: Se debe tener una consulta general a la cual se le aplicará filtros para la selección de registros.	

Tabla 34 CONSULTAR GASTOS

7. F6: Administrar Usuarios

HISTORIA DE USUARIO	
NÚMERO: 29	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Usuarios	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite administrar los Usuarios que trabajan usan el sistema	
OBSERVACIÓN: Solo se agregara dos tipos de usuarios: el súper usuario y el usuario normal	

Tabla 35 ADMINISTRAR USUARIOS

7.1. Ingresar Usuarios

HISTORIA DE USUARIO	
NÚMERO: 30	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Ingresar Usuarios	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite ingresar Usuarios	
OBSERVACIÓN:	

Tabla 36 INGRESAR USUARIOS

7.2. Actualizar Usuarios

HISTORIA DE USUARIO	
NÚMERO: 31	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Actualizar Usuarios	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite actualizar la información de los Usuarios registrados	
OBSERVACIÓN:	

Tabla 37 ACTUALIZAR USUARIOS

7.3. Eliminar Usuarios

HISTORIA DE USUARIO	
NÚMERO: 32	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Eliminar Usuarios	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite eliminar los Usuarios registrados	
OBSERVACIÓN:	

Tabla 38 ELIMINAR USUARIOS

7.4. Consultar Usuarios

HISTORIA DE USUARIO	
NÚMERO: 33	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Consultar Usuarios	
PRIORIDAD EN NEGOCIO: Media	RIESGO EN DESARROLLO: Media
ITERACIÓN ASIGNADA: Primera	
DESCRIPCIÓN: Permite consultar los datos de los Usuarios registrados	
OBSERVACIÓN: Se debe tener una consulta general a la cual se le aplicará filtros para la selección de registros	

Tabla 39 CONSULTAR USUARIOS

8. F6: Informe de ejecución del presupuesto

HISTORIA DE USUARIO	
NÚMERO: 34	USUARIO: Usuario Final Financiero
NOMBRE DE HISTORIA: Administrar Usuarios	
PRIORIDAD EN NEGOCIO: Alta	RIESGO EN DESARROLLO: Alta
ITERACIÓN ASIGNADA: Tercera	
DESCRIPCIÓN: Permite validar el nivel de ejecución del presupuesto asignado	
OBSERVACIÓN:	

Tabla 40 ADMINISTRAR USUARIOS

3.1.3.1. MODELO DE CASOS DE USO

En el siguiente diagrama de casos de uso se puede observar que la administración del sistema recae sobre el usuario final que básicamente tiene un perfil de Administrador Financiero. Dicho usuario estará habilitado para la administración de Auspiciantes, Roles de pago, Proyectos, Empleados y Cuentas bancarias.

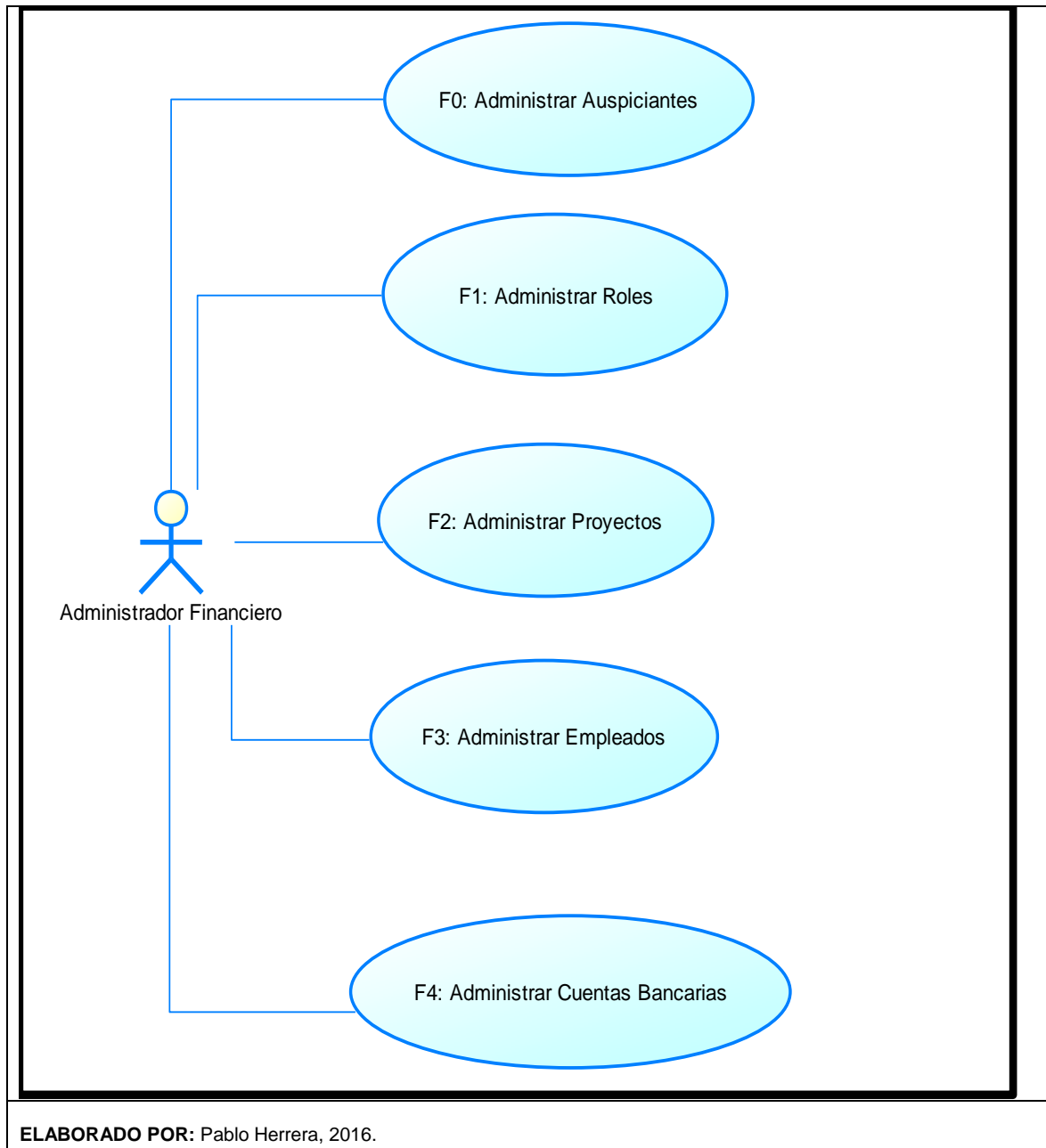


Ilustración 4 DIAGRAMA DE CASOS DE USO

3.2. PLANIFICACIÓN DE ENTREGAS

El presente proyecto de disertación se lo ha dividido en tres partes, tres iteraciones, dentro de las cuales se desarrollará las diferentes funcionalidades del sistema. Cada una de las iteraciones contará con tres entregables:

- A. Diseños lógico de interfaces**, en el cual se presentará mediante capturas de pantalla el diseño de las diferentes pantallas del sistema informático.

- B. Modelo de Datos**, en el cual se podrá observar el modelo físico de la base de datos utilizada para la iteración.

- C. Código Fuente**, el cual estará compuesto por las partes, que para el autor, representan los más interesante del desarrollo de la iteración.

3.2.1. DIAGRAMA DE ACTIVIDADES

3.2.1.1. Asignar Empleados a Proyectos

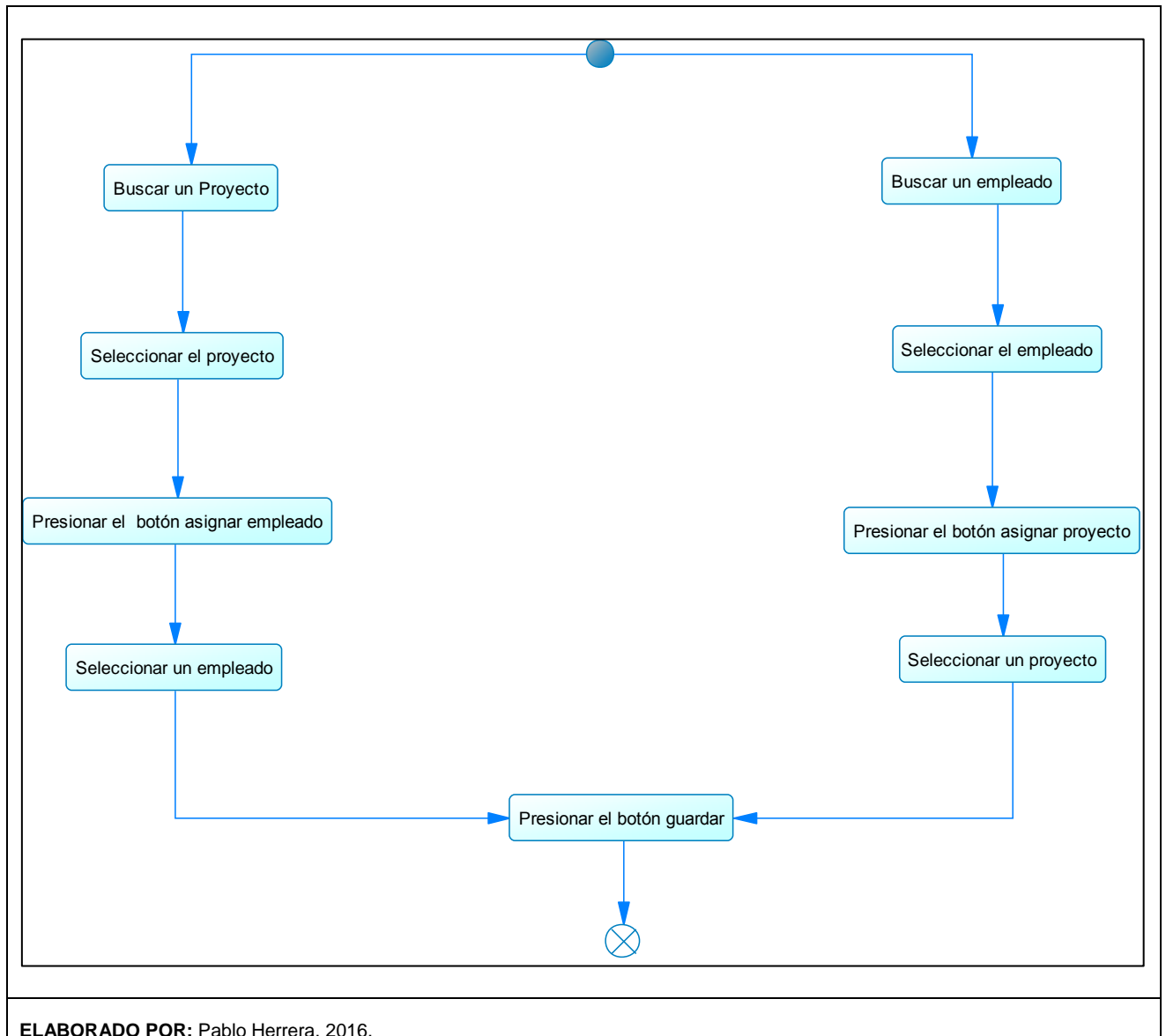


Ilustración 5 DIAGRAMA DE ACTIVIDADES PARA ASIGNACIÓN DE EMPLEADOS A PROYECTOS

3.2.1.2. Asignar Cuenta a Auspiciante

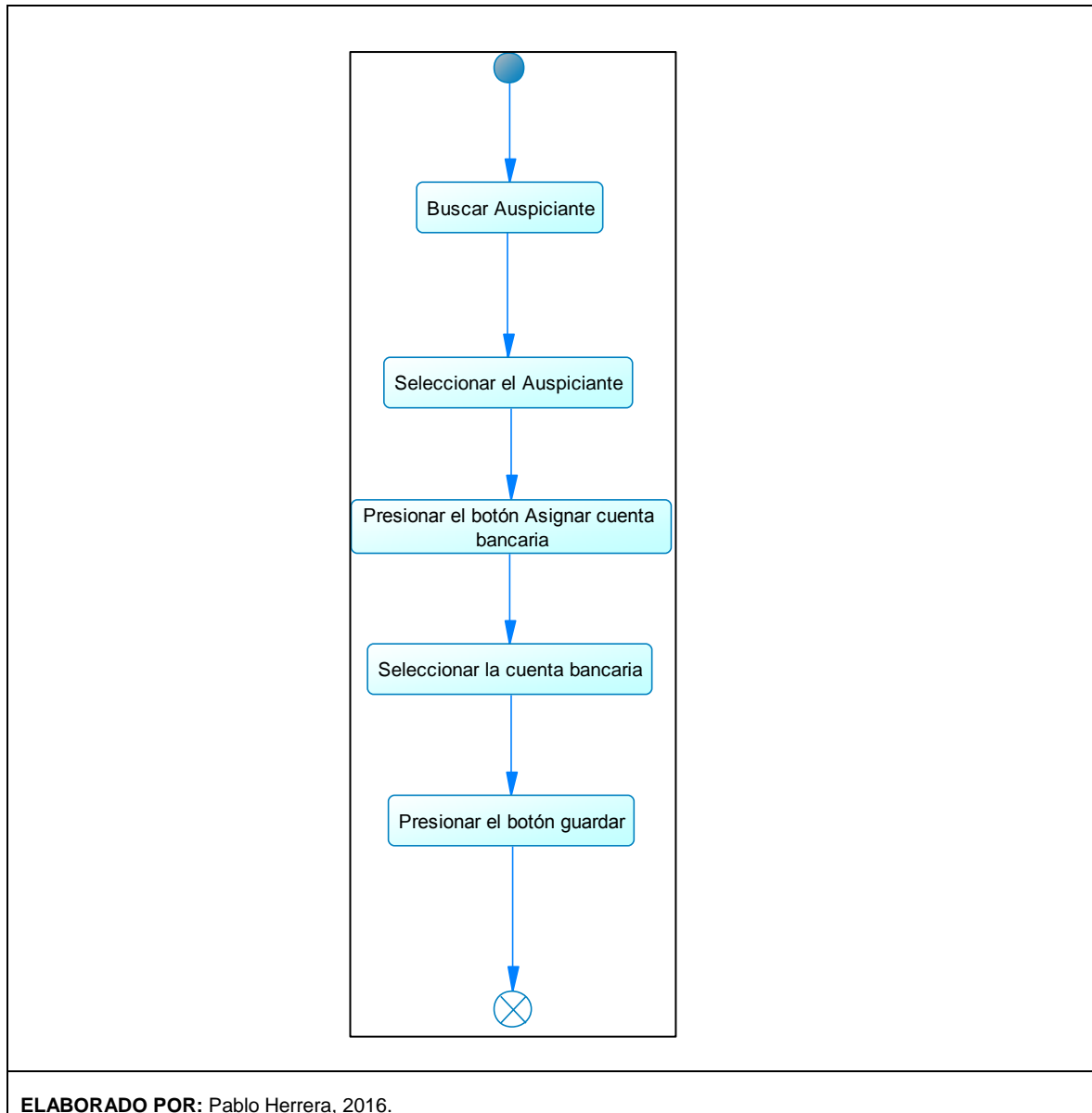


Ilustración 6 DIAGRAMA DE ACTIVIDADES PARA LA ASIGNACIÓN DE UNA CUENTA BANCARÍA A UN AUSPICIANTE

3.2.1.3. Asignar financiamiento a un proyecto

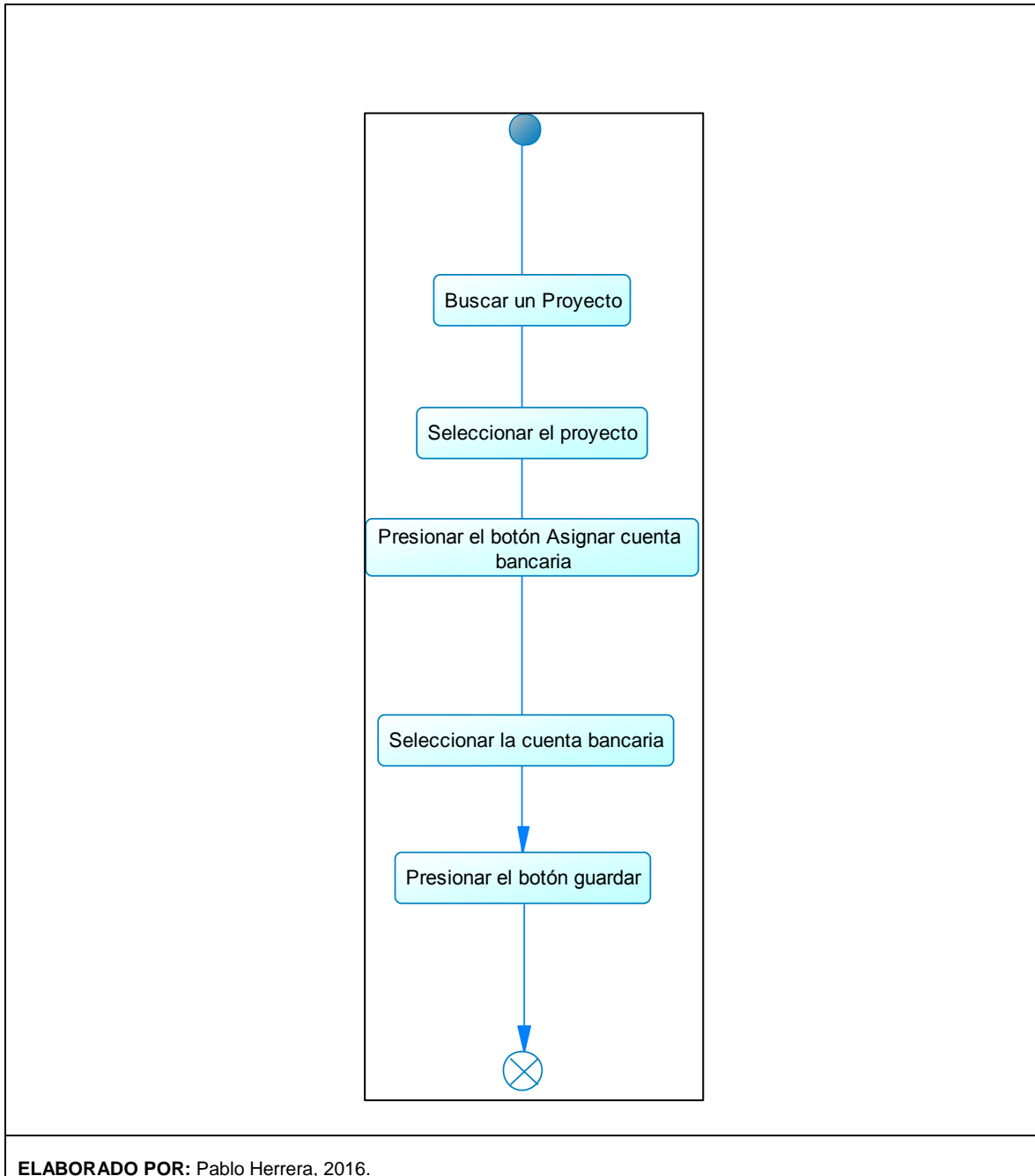


Ilustración 7 DIAGRAMA DE ACTIVIDADES PARA ASIGNAR EL FINANCIAMIENTO A UN PROYECTO

3.2.1.4. Registrar pago de sueldos

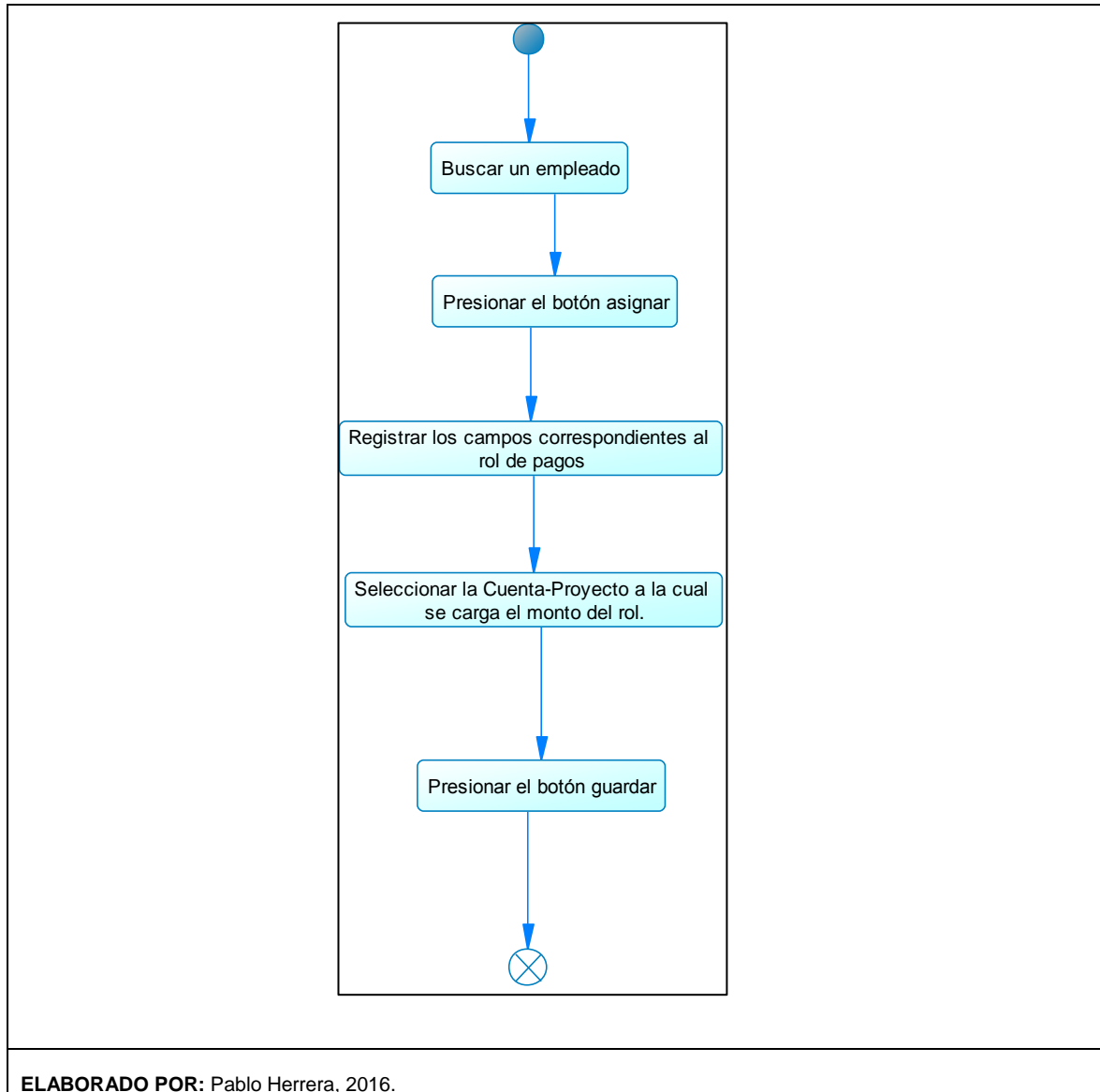


Ilustración 8 DIAGRAMA DE ACTIVIDADES PARA EL REGISTRO DE PAGO DE SUELDOS A EMPLEADOS

3.2.2. PLAN DEL PROYECTO.

En el siguiente documento se puede visualizar como se planifico las entrega de las diferentes historias de usuario que se solicitaron para el presente proyecto.

Para más detalles ver anexo 1: Plan de Entregas

3.3. ITERACIONES

3.3.1. ITERACIÓN 1


En la presente iteración se trabajará en el desarrollo de los módulos: Auspiciantes, Proyectos, Empleados, Cuentas Bancarias. Para los cuales se habilitará la funcionalidad para el ingreso, modificación, eliminación y filtros para la selección de registros.

3.3.1.1. DISEÑO LÓGICO DE INTERFACES PRIMERA ITERACIÓN

En las siguientes capturas de pantalla se podrá observar que en la presente iteración se desarrolló interfaces gráficas en las cuales se incluye un botón, en la parte superior izquierda, para el ingreso de nuevos registros; además, se incluye en todas las tablas la funcionalidad de filtros que permiten al usuario navegar por la información con mayor rapidez.

Se trabajó pensando en la experiencia del usuario mediante el uso de iconos, para las funcionalidades de la página, fáciles de reconocer; además, los colores seleccionados no solo se adaptan a los colores usados por la fundación, los mismos no producen distracciones innecesarias.

a) AUSPICIANTES



MODULO: AUSPICIANTES

[+ Nuevo](#)

Lista de auspiciantes. Columnas

CÓDIGO	NOMBRE	NACIONALIDAD	TELEFONO	E - MAIL	ESTADO	CUENTAS B.
TDH-SUIZASUIZA	TDH-SUIZA	SUIZA		TDHSUIZA@MAI	Activo	<input type="checkbox"/>
FLORENCE DE GOUMOENCE SUIZA	FLORENCE DE GOUMOENCE	SUIZA		FGOUMOENCE@	Inactivo	<input type="checkbox"/>
SUSETTE GOFF NORTE AMERICANA	SUSETTE GOFF	NORTE AMERICANA	12312	SGOFF@MAIL.C	Activo	<input type="checkbox"/>

Creación Nuevo Auspiciante

Código:

Nombre: *

Nacionalidad: *

Telefono:

E - MAIL: *

Estado del auspiciante: - Seleccione -

[Guardar](#)
[Cancelar](#)

- Seleccione -

Activo

Inactivo

Asignar Cuenta Banacaria a FLORENCE DE GOUMOENCE, SUIZA

BANCO	CODIGO	DESCRIPCION	CUENTAS B.
PRODUBANCO	18	DONACIONES 181	<input type="checkbox"/> Eliminar <input checked="" type="checkbox"/> INGRESAR PRESUPUESTO

[NUEVO](#)

Ingreso Presupuesto

Cuenta:

Año:

Monto:

[GUARDAR](#)

Ingreso Nueva Cuenta


Cuentas:

[GUARDAR](#) [CANCELAR](#)

ELABORADO POR: Pablo Herrera, 2016.

Ilustración 9 INTERFACES GRAFICAS AUSPICIANTE

b) PROYECTOS



MODULO: PROYECTOS

+ Nuevo

Lista de proyectos. Columnas							
NOMBRE	CIUDAD	ESTADO	F. INICIO	F. FIN	Empleados	CUENTAS B.	
PROGRAMA SALUD MATERNO INFANTIL	STO. DOMINGO DE LOS TSACHILAS	En proceso	2007-01-01	2017-12-31	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
REFUGIO DE LOS SUEÑOS	QUITO	En proceso	1998-01-01	2020-12-31	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CECAPI - CENTRO INFANTIL TDH	QUITO	En proceso	1977-01-01	2017-12-31	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Creación Nuevo Proyecto

Código:

Nombre: *

Ciudad: *

Fecha de Inicio: *

Fecha de Fin: *

Estado del proyecto: - Seleccione

- Seleccione -
Finalizado
En proceso
Suspendido
Negado
En espera

Guardar
Cancelar

Asignar Cuenta Bancaria PROGRAMA SALUD MATERNO INFANTIL, STO. DOMINGO DE LOS TSACHILAS

BANCO	CODIGO	DESCRIPCION	
PRODUBANCO	76	SUIZA 7621	Eliminar

NUEVO

Ingreso Nueva Cuenta

Cuentas: PRODUBANCO

GUARDAR
PRODUBANCO - 18

Asignar Empleado a PROGRAMA SALUD MATERNO INFANTIL, STO. DOMINGO DE LOS TSACHILAS

N. ID	NOMBRE	APELLIDO	
No records found.			

NUEVO

Ingreso Nuevo Proyecto

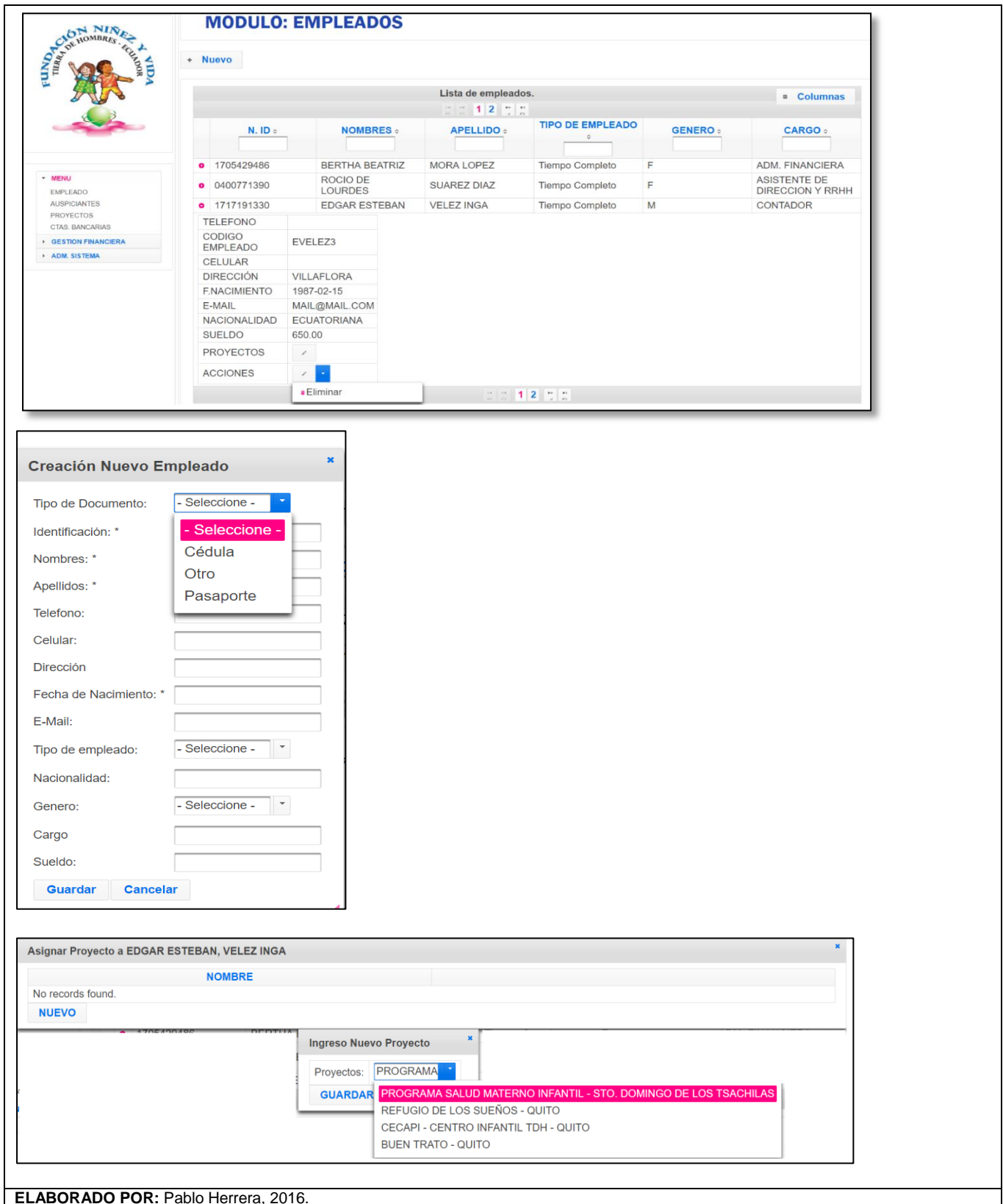
Proyectos: BERTHA BE

GUARDAR
BERTHA BEATRIZ - MORA LOPEZ
ROCIO DE LOURDES - SUAREZ DIAZ
EDGAR ESTEBAN - VELEZ INGA
CESAR EDMUNDO - AYALA TORRES
LUIS GUILLERMO - ORDOÑEZ LOPEZ

ELABORADO POR: Pablo Herrera, 2016.

Ilustración 10 INTERFACES GRAFICAS PROYECTOS

c) EMPLEADO



ELABORADO POR: Pablo Herrera, 2016.

Ilustración 11 INTERFACES GRAFICAS EMPLEADO

d) CUENTAS BANCARIAS

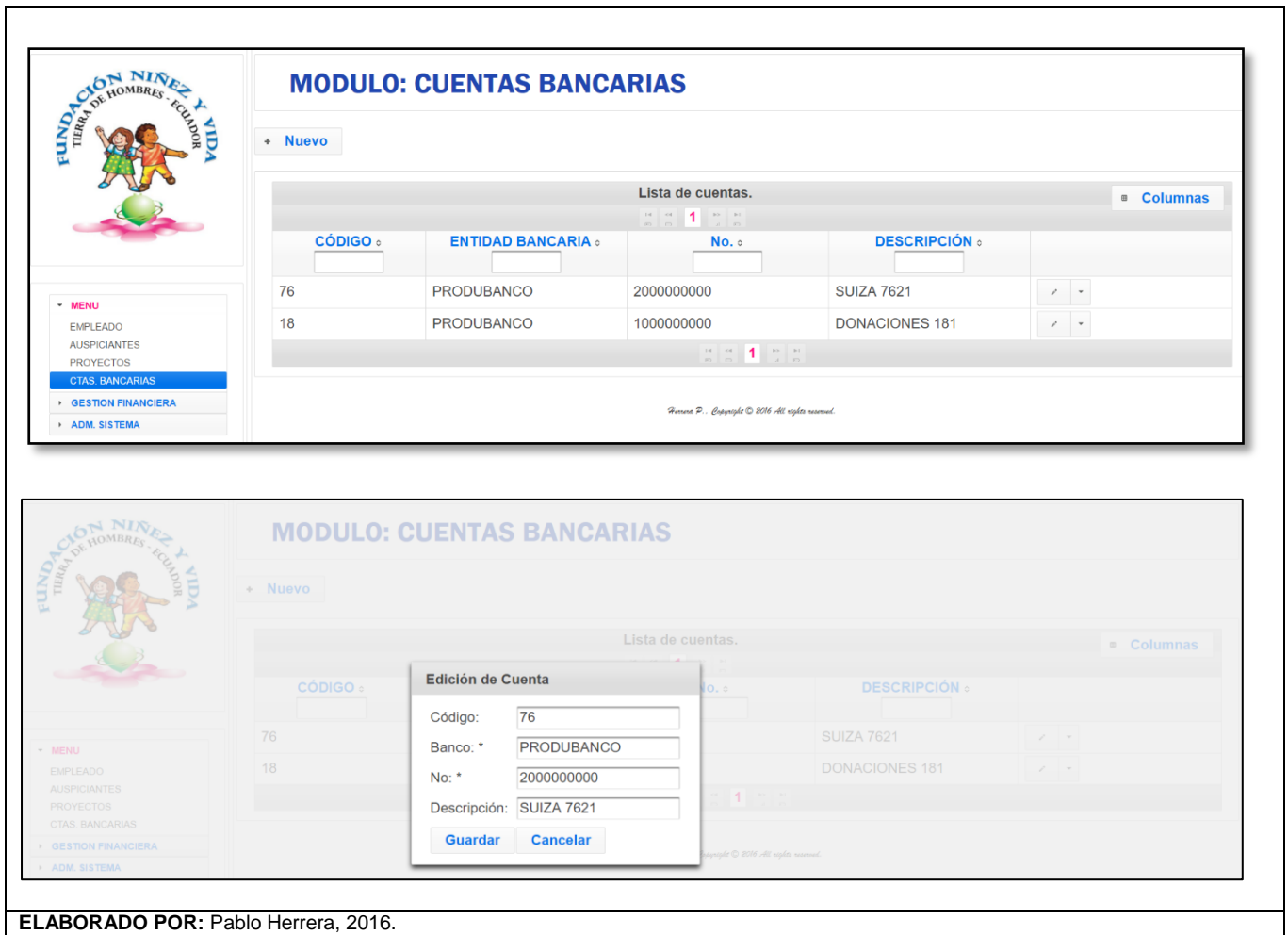
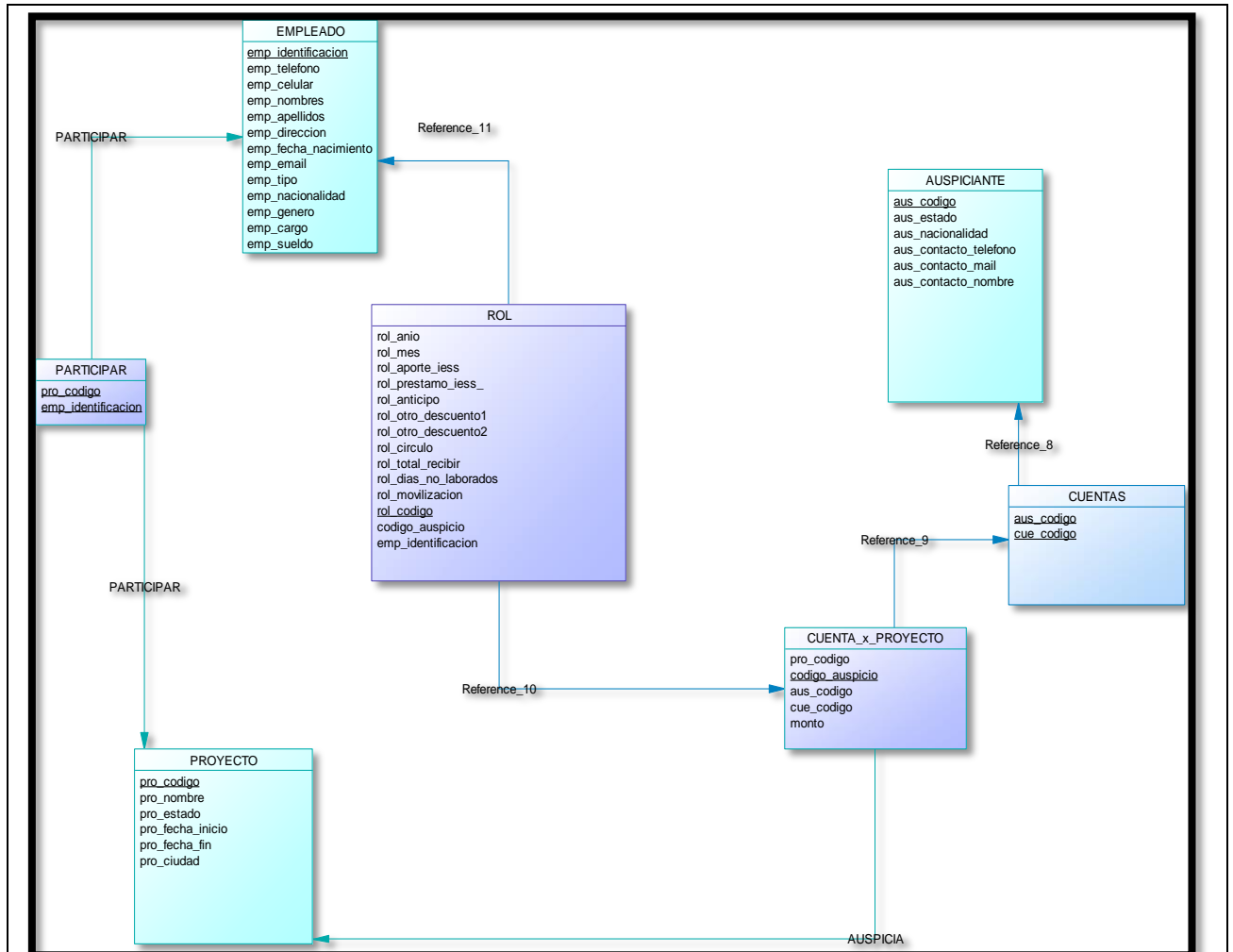


Ilustración 12 INTERFACES GRAFICAS CUENTAS BANCARIAS

3.3.1.2. MODELO DE DATOS PRIMERA ITERACIÓN

El siguiente modelo físico de la base de datos nos permite observar cómo se procedió con el desarrolló del aplicativo. En el mismo se puede ver que se incluyó los módulos de Empleados, Auspiciantes, Cuentas Bancarias, Proyecto y Rol de Pago, el cual se lo desarrollará en las siguientes iteraciones pero permite conectar los demás módulos.



ELABORADO POR: Pablo Herrera, 2016.

Ilustración 13 MODELO FÍSICO DE LA BASE DE DATOS PRIMERA ITERACIÓN

3.3.1.3. CÓDIGO FUENTE PRIMERA ITERACIÓN

A continuación se muestra extractos del código fuente donde se puede apreciar cómo se manejó el mapeo de las tablas de la base de datos, la definición de los métodos mediante el uso de interfaces y además la implementación de dichos métodos descritos en la interfaz.

a) Cuenta bancaria: extracto del código de la clase que mapea la tabla CUENTA_BANCO.

```

package fnv.modelo;

import java.io.Serializable;
import javax.persistence.*;
import java.math.BigDecimal;
import java.util.List;
/**
 * The persistent class for the cuenta_banco database table.
 *
 */
@Entity
@Table(name="cuenta_banco")
@NamedQuery(name="CuentaBanco.findAll", query="SELECT c FROM CuentaBanco c")

public class CuentaBanco implements Serializable {

    @Id //Denota Primary Key
    @Column(name="cue_codigo") //Nombre de la columna en la base de datos
    private long cueCodigo; //Nombre del atributo en la clase java
    /*
     * Todos los atributos
     */
    //bi-directional many-to-many association to Auspiciante
    @ManyToMany //tipo de relación
    @JoinTable( //como se relacionan las tablas
        name="auspiciante_x_cuenta"
        , joinColumns={
            @JoinColumn(name="cue_codigo")
        }
        , inverseJoinColumns={
            @JoinColumn(name="aus_codigo")
        }
    )

    public CuentaBanco() {
    }

    /*
     * Getter y Setters para la encapsulación.
     */
}

```

Ilustración 14 EXTRACTO CÓDIGO CUENTA BANCO PRIMERA ITERACIÓN

b) Extracto del código de la interfaz del servicio y la implementación del mismo.

Contiene la descripción de los métodos que se usarán para manejar la clase.

```
package fnv.service;

import java.util.List;

import javax.ejb.Local;

import fnv.model.CuentaBanco;

@Local
public interface CuentaBancoServicio {

    List<CuentaBanco> findAll();

    void crearCuentaBanco(CuentaBanco cuenta) throws Exception;

    void actualizarCuentaBanco(CuentaBanco cuenta);

    void eliminarCuentaBanco(long codigoCuentaBanco);

    CuentaBanco buscarCuentaBancoPorCodigo(long codigo);

}
```

Ilustración 15 EXTRACTO CODIGO INTERFAZ DE SERVICIO PRIMERA ITERACIÓN

Contiene la implementación de la interfaz ya descrita.

```
package fnv.service.impl;

import java.util.List;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;
import fnv.model.CuentaBanco;
import fnv.service.CuentaBancoServicio;

@Stateless
public class CuentaBancoServicioImpl implements CuentaBancoServicio {

    public CuentaBancoServicioImpl() {

    }

    @PersistenceContext(unitName = "fnv")
    protected EntityManager em;

    @SuppressWarnings("unchecked")
    @Override
    public List<CuentaBanco> findAll() {

        // return empleadoDao.findAll();
        Query = em.createQuery("SELECT e FROM CuentaBanco e");
        return (List<CuentaBanco>) query.getResultList();

    }

}
```

```
@Override
public void crearCuentaBanco(CuentaBanco empleado) throws Exception {
    em.persist(empleado);
}

@Override
public void actualizarCuentaBanco(CuentaBanco empleado) {
    em.merge(empleado);
}

@Override
public void eliminarCuentaBanco(long codigoCuentaBanco) {
    em.remove(buscarCuentaBancoPorCodigo(codigoCuentaBanco));
}

@Override
public CuentaBanco buscarCuentaBancoPorCodigo(long codigo) {
    // return empleadoDao.findByPk(codigo);
    return em.find(fnv.model.CuentaBanco.class, codigo);
}
}
```

Ilustración 16 EXTRACTO CODIGO IMPLEMENTACIÓN INTERFAZ PRIMERA ITERACIÓN

3.3.2. ITERACIÓN 2

En la presente iteración se trabajará en el desarrollo del módulo: Roles de Pago, para el cual se habilitará la funcionalidad para el ingreso, eliminación y filtros para la selección de registros. Es importante destacar que la funcionalidad de edición en este módulo no se la habilitará pues una vez ingresado el registro solo se podrá eliminarlo generando en el sistema el ingreso de un ajuste por eliminación de Rol de Pago con el fin de que el presupuesto no se vea afectado.

3.3.2.1. DISEÑO LÓGICO DE INTERFACES SEGUNDA ITERACIÓN

En este módulo se precisó incluir el ingreso variable de las parametrizaciones del aporte personal y patronal para el IESS (Instituto Ecuatoriano de Seguridad Social). Pero se debe desatacar que se predefinió con el usuario los valores que por defecto tendrán estas parametrizaciones.

Adicionalmente se acordó incluir un gráfico de guía con el estado de ejecución del presupuesto para cada una de las cuentas bancarias registradas, con el fin de facilitar al administrador financiero la carga de los sueldos a cada una de las cuentas bancarias registradas.

a)Roles de pago




MODULO: ROLES DE PAGO

MES: 6 AÑO: 2016 CONSOLIDADO

Lista de empleados

N. ID	NOMBRES	SUELDO	
1706566286	LUIS GUILLERMO, ORDOÑEZ LOPEZ	140.00	Asignar
1717191330	EDGAR ESTEBAN, VELEZ INGA	140.00	Asignar

Registro de Rol de Pagos al mes 6 del año 2016



Roles de pagos ORDOÑEZ LOPEZ, LUIS GUILLERMO . Total del sueldo asignado 0 / 140.00

S. Unificado	Movilización	T. Ingresos	Aporte IESS	Anticipo	T. Recibir	Cuenta
No records found.						

Registro de Rol de Pagos al mes 6 de



Registro de Nuevo Rol

Total Unificado: * 0

Movilización: 0

IESS PERSONAL 9% 45% 0

IESS PATRONAL 12% 15% 0

Descuento 3: 0

Prestamo IESS: 0

Anticipo: 0

Otros Desc. 1: 0

Otros Desc. 2: 0

Imp. Renta: 0

Total a Recibir: 0

Aportado por: 76- PROGRÁ

76- PROGRAMA SALUD MATERNO INFANTIL. STO. DOMINGO DE LOS TSACHILAS

18- REFUGIO DE LOS SUEÑOS QUITO

18- CECAPI - CENTRO INFANTIL TDH QUITO

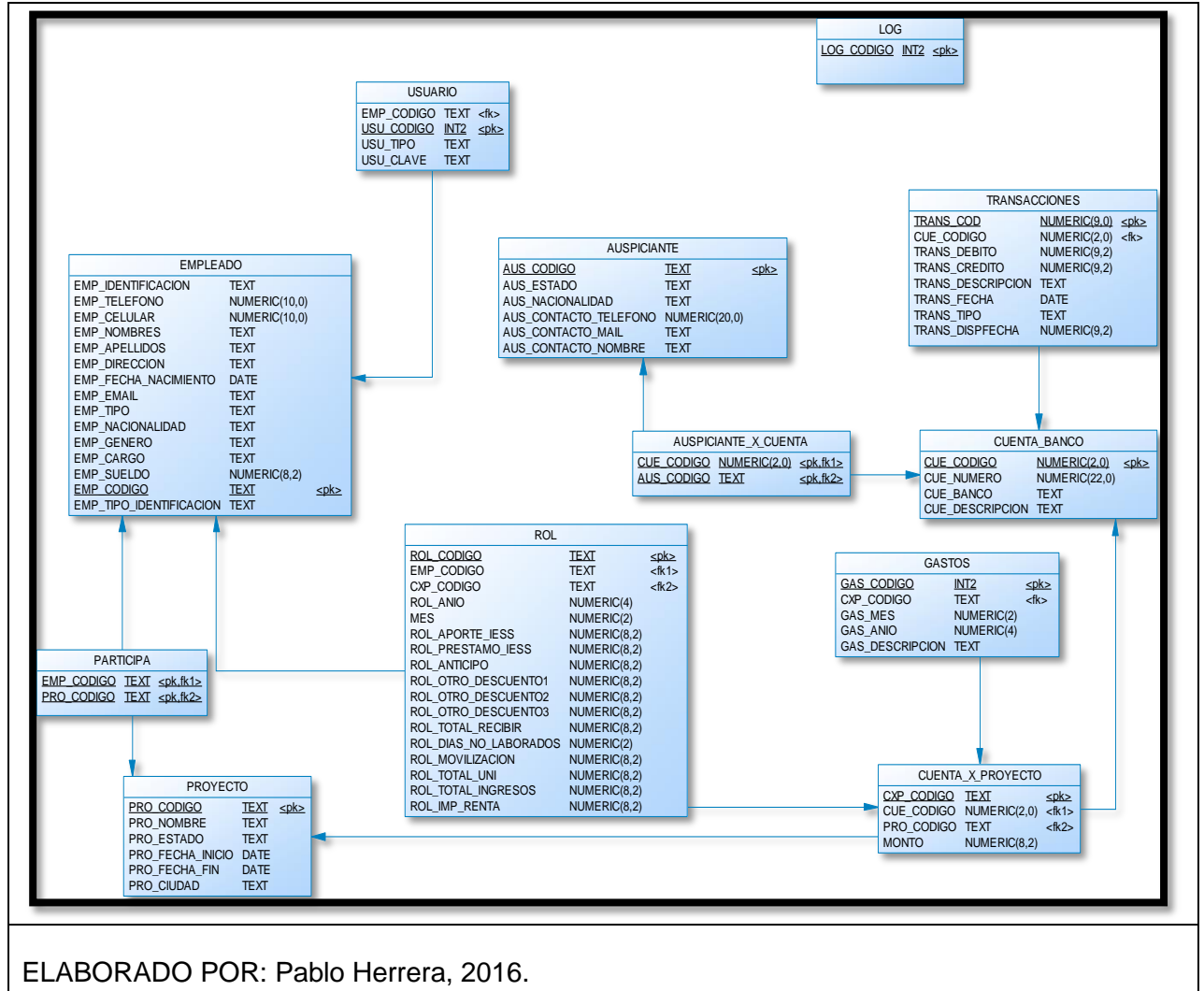
ELABORADO POR: Pablo Herrera, 2016.

Ilustración 17 INTERFACES GRAFICAS ROLES DE PAGO

3.3.2.2. MODELO DE DATOS SEGUNDA ITERACIÓN

Para la presente iteración se puede denotar la inclusión de las tablas para el manejo de usuario, el registro de transacciones y la tabla para la auditoria del sistema,

LOG, la cual será desarrollada más adelante. Adicionalmente se puede observar la tabla AUSPICIANTE_X_CUENTA la cual permite tener un control sobre las cuentas bancarias en las que el auspiciante realiza sus aportes.



ELABORADO POR: Pablo Herrera, 2016.

Ilustración 18 MODELO FÍSICO DE LA BASE DE DATOS SEGUNDA ITERACIÓN

3.3.2.3. CÓDIGO FUENTE SEGUNDA ITERACIÓN

En la siguiente sección se puede observar un extracto del código fuente utilizado para el mapeo de la tabla ROL (Rol de Pagos) así como la definición de los métodos y su respectiva implementación. Adicionalmente se puede notar la generación del ingreso de ajuste por eliminación de rol.

a) Clase modelo para el mapeo de la tabla ROL.

```

package fnv.modelo;

import java.io.Serializable;
import javax.persistence.*;
import java.math.BigDecimal;

/**
 * The persistent class for the rol database table.
 */
@Entity
@NamedQuery(name = "Rol.findAll", query = "SELECT r FROM Rol r")
public class Rol implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @Column(name = "rol_codigo")
    private String rolCodigo;

    /*
    *Todos los atributos
    */

    // bi-directional many-to-one association to CuentaXProyecto
    @ManyToOne
    @JoinColumn(name = "cxp_codigo")
    private CuentaXProyecto;

    // bi-directional many-to-one association to Empleado
    @ManyToOne
    @JoinColumn(name = "emp_codigo")
    private Empleado;

    //Constructor
    public Rol() {
        cuentaXProyecto = new CuentaXProyecto();
        this.setRolTotalUni(new BigDecimal("0"));
        this.setRolAporteless(new BigDecimal("0"));
        this.setRolMovilizacion(new BigDecimal("0"));
        this.setRolOtroDescuento3(new BigDecimal("0"));
        this.setRolOtroDescuento2(new BigDecimal("0"));
        this.setRolOtroDescuento1(new BigDecimal("0"));
        this.setRolImpRenta(new BigDecimal("0"));
        this.setRolAnticipo(new BigDecimal("0"));
        this.setRolTotalRecibir(new BigDecimal("0"));
        this.setRolTotalIngresos(new BigDecimal("0"));
        this.setRolPrestamoless(new BigDecimal("0"));
        this.setRolDiasNoLaborados(new BigDecimal("0"));
        this.setRolAportePatronal(new BigDecimal("0"));
    }

    //Getters and Setters para encapsular los atributos
}

```

Ilustración 19 EXTRACTO CÓDIGO MAPEO DE TABLA ROL

b) Interface del servicio ROL

```
package fnv.service;

import java.math.BigDecimal;
import java.util.List;

import javax.ejb.Local;

import fnv.model.CuentaXProyecto;
import fnv.model.Rol;
import fnv.model.VConsolidadoRol;

@Local
public interface RolServicio {
    List<Rol> findAll();
    void crearRol(Rol rol);
    void actualizarRol(Rol rol);
    void eliminarRol(String codigoRol);
    Rol buscarRolPorCodigo(String codigo);
    List<Rol> getRolesEmpleado(BigDecimal anio, BigDecimal mes, String emplIdentificacion);
    List<CuentaXProyecto> getCuentasXProyecto();
    void registroTransaccion(Rol rol);
    List<VConsolidadoRol> getConsolidado(BigDecimal anio, BigDecimal mes, long codigoCuenta);
}
```

Ilustración 20 EXTRACTO CÓDIGO INTERFACE SERVICIO ROL

c) Implementación de la interfaz del servicio ROL

```
package fnv.service.impl;
```

```
import java.math.BigDecimal;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;
import fnv.model.CuentaXProyecto;
import fnv.model.Rol;
import fnv.model.Transaccion;
import fnv.model.VConsolidadoRol;
import fnv.service.RolServicio;

@Stateless
public class RolServicioImpl implements RolServicio {

    public RolServicioImpl() {

    }

    @PersistenceContext(unitName = "fnv")
    protected EntityManager em;

    @SuppressWarnings("unchecked")
    @Override
    public List<Rol> findAll() {

        Query = em.createQuery("SELECT r FROM Rol r");
        return (List<Rol>) query.getResultList();
    }

    @Override
    public void crearRol(Rol rol) {
        try {
            String cueCodigo = "";

            cueCodigo = rol.getCuentaXProyecto().getCxpCodigo().substring(
                (rol.getCuentaXProyecto().getCxpCodigo().length() - 2),
                rol.getCuentaXProyecto().getCxpCodigo().length());
            rol.getCuentaXProyecto().getCuentaBanco().setCueCodigo(Long.parseLong(cueCodigo));

            em.persist(rol);

            registroTransaccion(rol);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @SuppressWarnings("unchecked")
    public void registroTransaccion(Rol rol) {
        try {

            Query = em.createQuery("SELECT t FROM Transaccion t");

            int size = ((List<Transaccion>) query.getResultList()).size();
            Transaccion trans = new Transaccion();
            trans.setTransCod(size + 1);
            trans.setTransDebito(rol.getRolTotalIngresos().add(rol.getRolAportePatronal()));
            trans.setCuentaBanco(rol.getCuentaXProyecto().getCuentaBanco());
            // fecha se setea en la base
            trans.setTransCredito(new BigDecimal("0"));
            trans.setTransDescripcion(
                "Rol de pagos: " + rol.getMes() + "-" + rol.getRolAnio() + " " +
                rol.getEmpleado().getEmpCodigo());
            trans.setTransTipo("debRol " + rol.getMes() + "-" + rol.getRolAnio());
        }
    }
}
```

```

        em.persist(trans);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@SuppressWarnings("unchecked")
private void registroAjusteTransaccion(Rol rol) {
    try {
        Calendar fecha = new GregorianCalendar();
        rol.setRolAnio(new BigDecimal("'" + fecha.get(Calendar.YEAR)));
        Query = em.createQuery("SELECT t FROM Transaccion t");

        int size = ((List<Transaccion>) query.getResultList()).size();
        Transaccion trans = new Transaccion();
        trans.setTransCod(size + 1);
        trans.setTransCredito(rol.getRolTotalIngresos());
        trans.setCuentaBanco(rol.getCuentaXProyecto().getCuentaBanco());
        trans.setTransFecha(fecha.getTime());
        trans.setTransDebito(new BigDecimal("0"));
        trans.setTransDescripcion("Ajuste Rol de pagos: " + rol.getEmpleado().getEmpCodigo());
        trans.setTransTipo("credDelRol " + rol.getMes() + "-" + rol.getRolAnio());
        em.persist(trans);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@Override
public void actualizarRol(Rol rol) {
    em.merge(rol);
}

@Override
public void eliminarRol(String codigoRol) {
    Rol = buscarRolPorCodigo(codigoRol);
    em.remove(rol);
    registroAjusteTransaccion(rol);
}

@Override
public Rol buscarRolPorCodigo(String codigo) {
    return em.find(fnv.model.Rol.class, codigo);
}

@SuppressWarnings("unchecked")
@Override
public List<Rol> getRolesEmpleado(BigDecimal anio, BigDecimal mes, String emplIdentificacion) {
    Query = em.createQuery("SELECT r FROM Rol r join r.empleado e where e.emplIdentificacion = '"
        + emplIdentificacion + "'" + "and r.rolAnio = " + anio + " and r.mes = " + mes + "'");
    return (List<Rol>) query.getResultList();
}

@SuppressWarnings("unchecked")
@Override
public List<CuentaXProyecto> getCuentasXProyecto() {
    // return empleadoDao.findAll();
    Query = em.createQuery("SELECT e FROM CuentaXProyecto e");
    return (List<CuentaXProyecto>) query.getResultList();
}

@SuppressWarnings("unchecked")
@Override
public List<VConsolidadoRol> getConsolidado(BigDecimal anio, BigDecimal mes, long codigoCuenta) {

```

```
// return empleadoDao.findAll();  
  
Query = em.createQuery("SELECT v FROM VConsolidadoRol v where v.rolAnio =" + anio + " and  
v.mes =" + mes  
+ " AND v.cxp LIKE '%" + codigoCuenta + "%");  
return (List<VConsolidadoRol>) query.getResultList();  
}  
}
```

Ilustración 21 EXTRACTO DE CODIGO IMPLEMENTACIÓN DE LA INTERFAZ DEL SERVICIO ROL

NOTA.- El código en la línea punteada representa el ingreso del ajuste para la eliminación de un rol con el fin de que el presupuesto no se vea afectado.

3.3.3. ITERACIÓN 3

En la presente iteración se trabajó en el desarrollo de los módulos de gastos, usuarios y además en la generación de informes. Se habilitó las funcionalidades para el ingreso, modificación, eliminación y filtrado de los registros tanto para gastos como para usuarios.

3.3.3.1. DISEÑO LÓGICO DE INTERFACES TERCERA ITERACIÓN

En esta sección se puede observar principalmente los formatos elegidos para la presentación de informes, los cuales se asemejan mucho a los formatos ya existentes. En cuanto se refiere al control de acceso a la aplicación se colocó los campos respectivos para la autenticación del usuario; adicionalmente se destaca que el registro de nuevos usuarios no se lo incluye en la pantalla de inicio pues los mismos deberán ser ingresados internamente en el módulo de usuarios con el usuario que cuente con el permiso respectivo.

a) Gastos

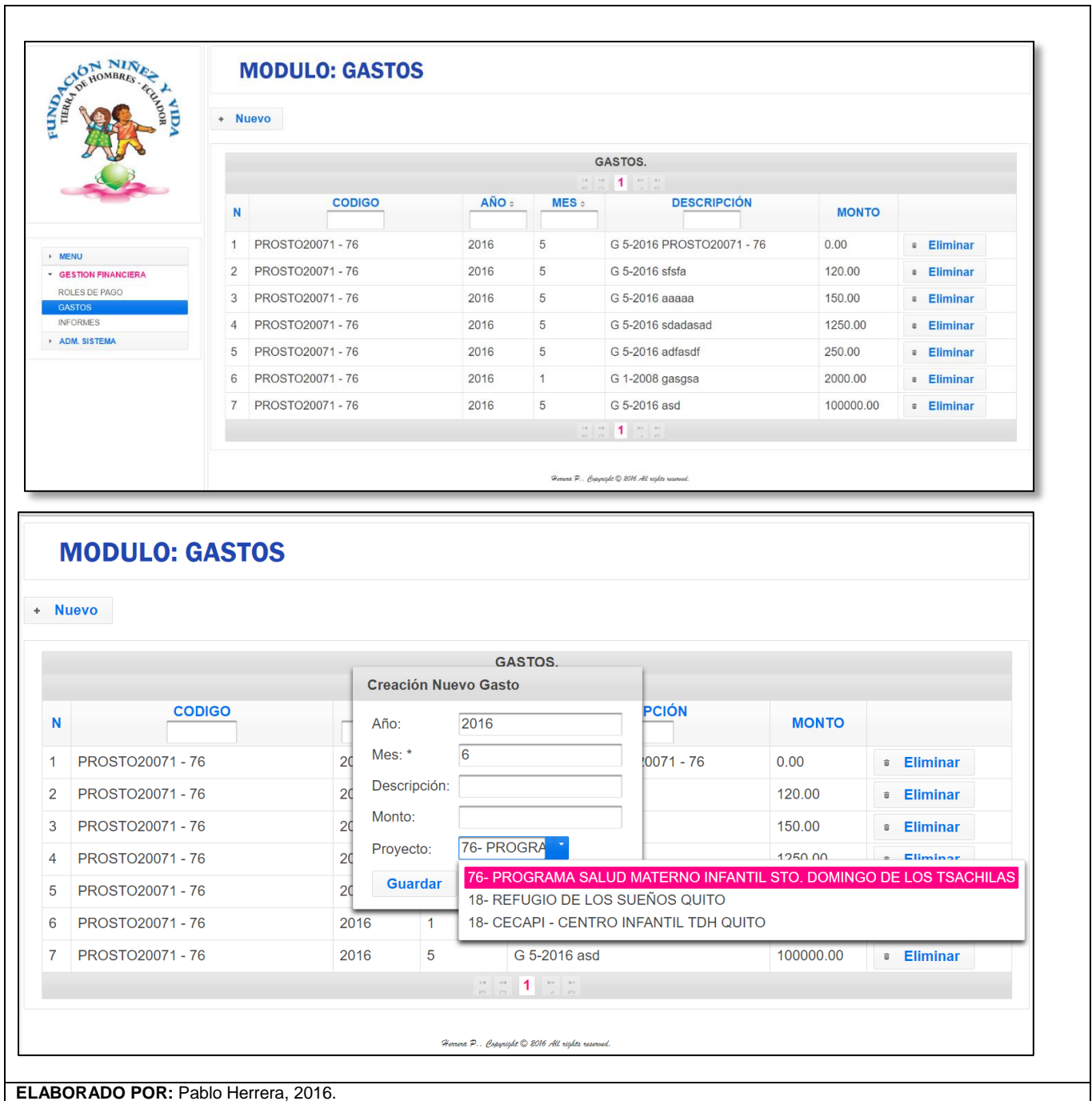


Ilustración 22 INTERFACES GRAFICAS GASTOS

b) Control de Usuarios



Usuario:

Clave:

[Ingresar](#)

Ferrera P. Copyright © 2016. All rights reserved.

*La escuela
intruye
y la familia
educa*






- [MENU](#)
- [GESTION FINANCIERA](#)
- [ADM. SISTEMA](#)
- [USUARIOS](#)

Bienvenido (a) BMORA2

[Logout](#)

Ferrera P. Copyright © 2016. All rights reserved.

MODULO: USUARIOS

[+ Nuevo](#)

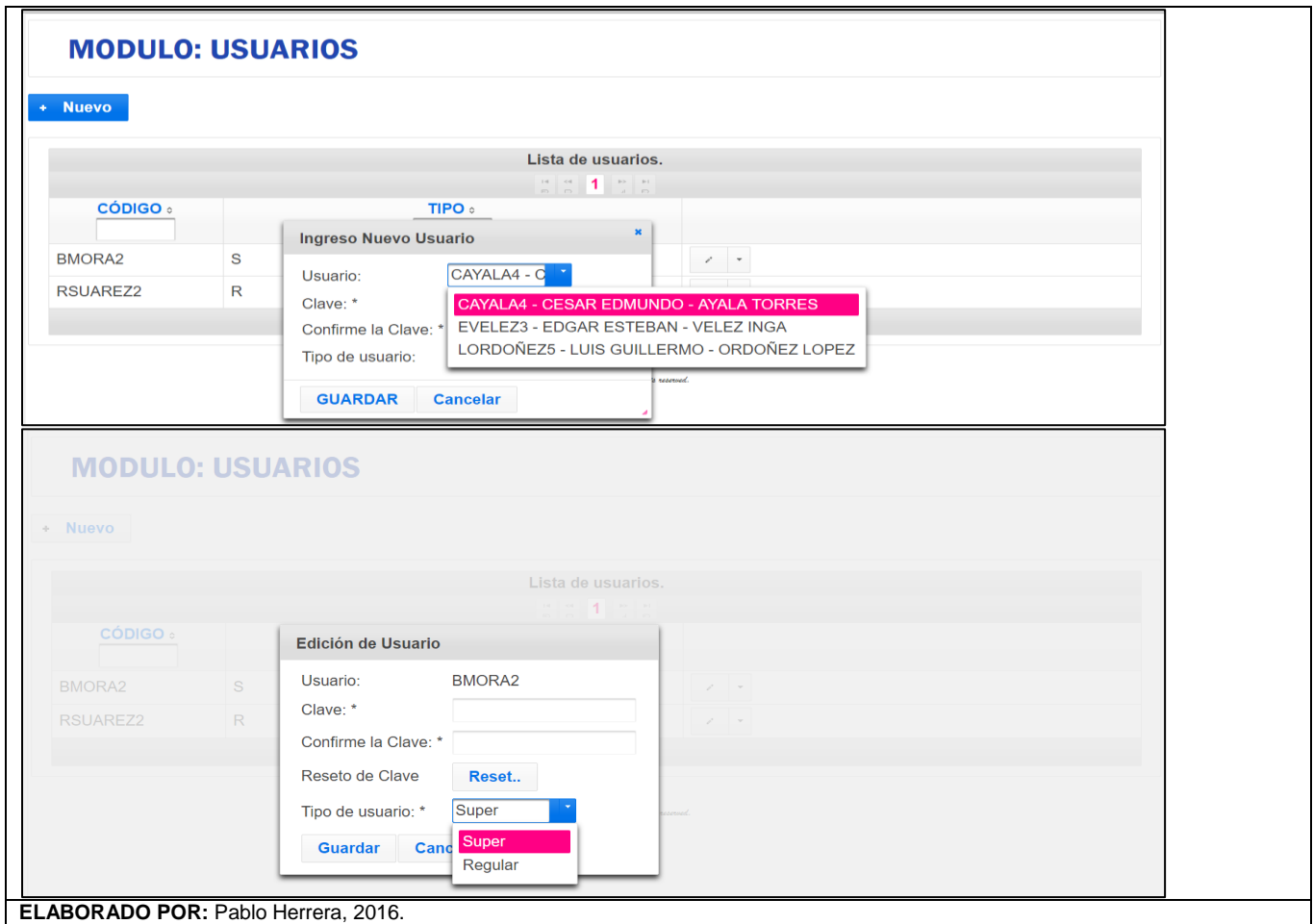
Lista de usuarios.

1

CÓDIGO	TIPO	
BMORA2	S	✓
RSUAREZ2	R	✓

1

Ferrera P. Copyright © 2016. All rights reserved.



ELABORADO POR: Pablo Herrera, 2016.

Ilustración 23 INTERFACES GRAFICAS CONTROL DE USUARIOS

c) Informes



ELABORADO POR: Pablo Herrera, 2016.

Ilustración 24 INTERFAZ GRÁFICA INFORMES

b) Impresión de roles

MODULO: ROLES DE PAGO

MES: 5 AÑO: 2016 CONSOLIDADO

Registro de Rol de Pagos al mes 5 del año 2016

SELECCIONE LA CUENTA A CONSULTAR: 76 - PRODUBANCO- 2000000000

EMPLEADO	TOTAL UNIFICADO	APORTE PERSONAL IESS	APORTE PATRONAL IESS	ANTICIPO	DIAS N.L.	MOVILIZACION	DESCUENTO 1	DESCUENTO 2	DESCUENTO 3	IMP. RENTA	PRESTAMO IESS	TOTAL INGRESOS	TOTAL A RECIBIR
CESAR EDUARDO	675.33	63.82	0.00	0.00	0	0.00	0.00	0.00	82.45	0.00	50.63	675.33	478.43
LUIS GUILLERMO	2475.00	233.89	0.00	0.00	0	0.00	10.00	53.41	280.00	86.48	0.00	2475.00	1831.22
BERTHA BEATRIZ	850.00	80.33	0.00	0.00	0	0.00	0.00	0.00	250.00	0.00	0.00	850.00	519.68
RODIO DE LOURDES	472.18	44.00	0.00	0.00	0	5.30	10.00	0.00	280.00	0.00	5.22	472.18	137.56
Totales:	\$4472.51	\$5.00	\$422.04	\$0.00	\$0	\$5.30	\$20.00	\$53.41	\$972.45	\$86.48	\$55.85	\$4472.51	\$2966.89

Imprimir

Total: 1 página

Guardar Cancelar

Destino: Guardar como PDF

Páginas: Todos

Diseño: Horizontal

Tamaño del papel: A4

Márgenes: Predeterminado

Opciones:

- Simplificar página
- Encabezados y pies de página
- Gráficos en segundo plano

Registro de Rol de Pagos al mes 5 del año 2016

SELECCIONE LA CUENTA A CONSULTAR: 76 - PRODUBANCO- 2000000000

EMPLEADO	TOTAL UNIFICADO	APORTE PERSONAL IESS	APORTE PATRONAL IESS	ANTICIPO	DIAS N.L.	MOVILIZACION	DESCUENTO 1	DESCUENTO 2	DESCUENTO 3	IMP. RENTA	PRESTAMO IESS	TOTAL INGRESOS	TOTAL A RECIBIR
CESAR EDUARDO	675.33	63.82	0.00	0.00	0	0.00	0.00	0.00	82.45	0.00	50.63	675.33	478.43
LUIS GUILLERMO	2475.00	233.89	0.00	0.00	0	0.00	10.00	53.41	280.00	86.48	0.00	2475.00	1831.22
BERTHA BEATRIZ	850.00	80.33	0.00	0.00	0	0.00	0.00	0.00	250.00	0.00	0.00	850.00	519.68
RODIO DE LOURDES	472.18	44.00	0.00	0.00	0	5.30	10.00	0.00	280.00	0.00	5.22	472.18	137.56
Totales:	\$4472.51	\$5.00	\$422.04	\$0.00	\$0	\$5.30	\$20.00	\$53.41	\$972.45	\$86.48	\$55.85	\$4472.51	\$2966.89

Print

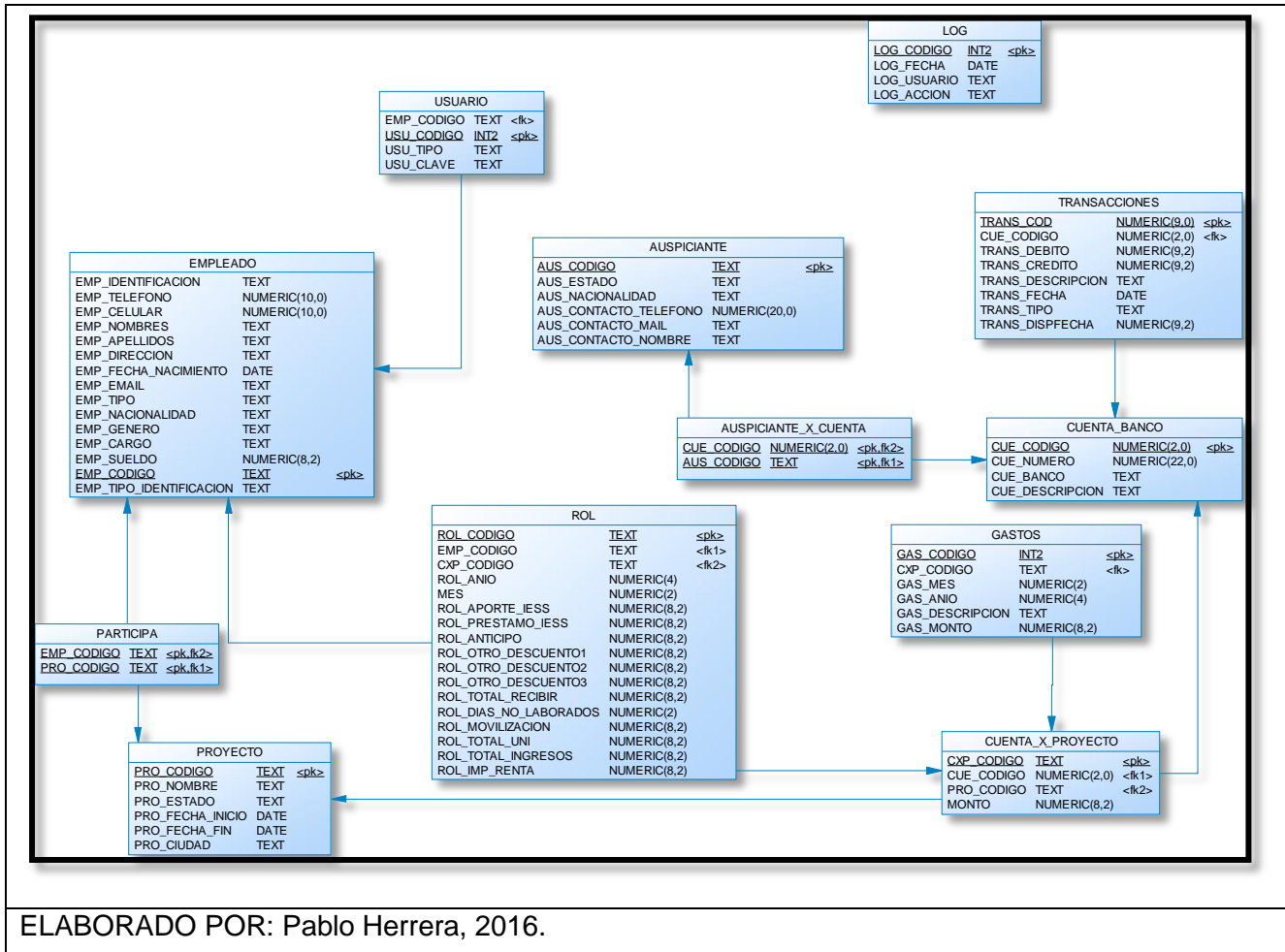


ELABORADO POR: Pablo Herrera, 2016.

Ilustración 25 IMPRESIÓN DE ROLES DE PAGO

3.3.3.2. MODELO DE DATOS TERCERA ITERACIÓN

Para la tercera y última iteración se desarrolla la tabla de auditoria del sistema así como la tabla de usuario y la tabla de registro de gastos.



ELABORADO POR: Pablo Herrera, 2016.

Ilustración 26 MODELO FÍSICO DE LA BASE DE DATOS PRIMERA ITERACIÓN

3.3.3.3. CÓDIGO FUENTE TERCERA ITERACIÓN

Para la tercera y última iteración se presenta un poco del desarrollo realizado en la base de datos tanto para la obtención de saldos a la fecha como para los informes de ejecución del presupuesto. Además, se presenta la implementación de la inserción en la tabla transacciones cuando se registra un rol de pagos.

a) Creación de la vista **dinámica** para ejecución del presupuesto usada en el informe de ejecución del presupuesto.

```
-- Function: public.fn_crea_vista_ejpres(text)
-- DROP FUNCTION public.fn_crea_vista_ejpres(text);
CREATE OR REPLACE FUNCTION public.fn_crea_vista_ejpres(text)
  RETURNS void AS
$BODY$
  DECLARE
    v_param alias for $1;
    script_crear_vista text;
BEGIN
    script_crear_vista = 'CREATE OR REPLACE VIEW public.v_ejec_pres AS
    SELECT A.CUENTA CUENTA, A.PRESUPUESTO PRESUPUESTO, B.DEBITOS DEBITOS,
(100*((B.DEBITOS)/ A.PRESUPUESTO)) EJECUTADO
FROM(
SELECT cue_codigo CUENTA, SUM(trans_credito) AS PRESUPUESTO
FROM public.transacciones
WHERE trans_descripcion LIKE "%|| v_param ||%" AND trans_descripcion NOT LIKE "%AJ%"
GROUP BY cue_codigo) AS A,
(SELECT cue_codigo CUENTA, SUM(trans_debito) AS DEBITOS
FROM public.transacciones
WHERE trans_descripcion LIKE "%|| v_param ||%" AND trans_descripcion NOT LIKE "%AJ%"
GROUP BY cue_codigo) AS B
WHERE A.CUENTA = B.CUENTA';
    execute script_crear_vista;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.fn_crea_vista_ejpres(text)
  OWNER TO postgres;
```

Ilustración 27 VISTA DINÁMICA EJECUCIÓN DEL PRESUPUESTO

b) Función la para inserción del saldo disponible a la fecha. Usado en la asignación de nuevos roles de pago.

```
-- Function: public.insertar_saldo_disp_fecha()
-- DROP FUNCTION public.insertar_saldo_disp_fecha();
CREATE OR REPLACE FUNCTION public.insertar_saldo_disp_fecha()
    RETURNS void AS
$BODY$
    DECLARE
    creditos NUMERIC(9,2);
    debitos NUMERIC(9,2);
    saldo NUMERIC(9,2);
    BEGIN
        SELECT (SELECT SUM(trans_credito) FROM TRANSACCIONES) INTO creditos;
        SELECT (SELECT SUM(trans_debito) FROM TRANSACCIONES) INTO debitos;
        saldo := creditos - debitos;
        UPDATE public.transacciones
        SET trans_dispfecha = saldo
        WHERE public.transacciones.trans_cod = 1;
    END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.insertar_saldo_disp_fecha()
    OWNER TO postgres;
```

Ilustración 28 FUNCIÓN PARA INSERCIÓN SE SALDO A LA FECHA

c) Función para la inserción de transacciones en la tabla Transacciones. Usada en la inserción de gastos, roles y ajustes por eliminación de roles y gastos.

```
--Function: public.insertar_transaccion(numeric, numeric, numeric, numeric, text, date, text, numeric)
-- DROP FUNCTION public.insertar_transaccion(numeric, numeric, numeric, numeric, text, date, text, numeric);
CREATE OR REPLACE FUNCTION public.insertar_transaccion(
    trans_cod numeric,
    cue_codigo numeric,
    trans_debito numeric,
    trans_credito numeric,
    trans_descripcion text,
    trans_fecha date,
    trans_tipo text,
    trans_dispfecha numeric)
RETURNS void AS
$BODY$
BEGIN
    INSERT INTO transacciones VALUES (trans_cod, cue_codigo,
        trans_debito, trans_credito, trans_descripcion,
        trans_fecha, trans_tipo, trans_dispfecha);
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.insertar_transaccion(numeric, numeric, numeric, numeric, text, date, text, numeric)
OWNER TO postgres;
```

Ilustración 29 INSERCIÓN DE TRANSACCIONES

3.4. PRUEBAS Y MANTENIMIENTO

En la presente sección se muestra el plan de pruebas de aceptación que se aplicará al sistema, el informe para la posible implantación del sistema y además se explicará el uso de los distintos manuales que se presentan en la sección de anexos.

3.4.1. PLAN DE PRUEBAS DE ACEPTACIÓN

A continuación se presenta el esquema utilizado para la validación de las funcionalidades del sistema informático. Donde se entenderá:

- a) **N. HU.** Como el número de Historia de Usuario usada en la validación.
- b) **HU.** Como el nombre de la historia de usuario utilizada en la validación.
- c) **N. EVENTO.** Como el número del evento que se probará.
- d) **EVENTO A PROBAR.** Como una breve descripción del evento que se probará.
- e) **RESULTADO ESPERADO.** Como el resultado que se espera se genere al reproducir el evento a probar.

N.HU	HU.	N. EVENTO	EVENTO A PROBAR	RESULTADO ESPERADO
1	Administración de Auspiciantes.	1	Ingresar a la pantalla de administración de auspiciantes	Validar que el sistema permita visualizar al usuario los siguientes campos: (campos para ingreso de datos); tabla de registros; los siguientes botones: cancelar, insertar, eliminar, actualizar, buscar.
		2	Botón Actualizar	Verificar que sólo se habilite cuando existan registros. Verificar que al arrancar la pantalla de administración de auspiciantes se encuentre desactivado. Verificar que al presionar se muestre un panel de tipo modal para la edición de los campos.
		3		Validar que al presionar se dirija hacia la pantalla

N.HU	HU.	N. EVENTO	EVENTO A PROBAR	RESULTADO ESPERADO
			Botón Cancelar	principal. Verificar que todo el tiempo se encuentre activado.
		4	Botón Ingresar	Validar que al ser presionado se limpien los campos y se habiliten.
		5	Botón Eliminar	Validar que al ser presionado se muestre un dialogo de confirmación.
		6	Botón Buscar	Validar que al ser presionado se carguen los campos con el resultado de la búsqueda.
		7	Tabla de registros	Validar que la tabla se cargue al cargar la pantalla de administración de auspiciantes. Validar que se actualice cuando se realice algún cambio en la base de datos.
6	Administración de Roles	8	Ingresar a la pantalla de administración de roles	Validar que el sistema permita visualizar al usuario los siguientes campos: (campos para ingreso de datos); tabla de registros; los siguientes botones: cancelar, insertar, eliminar, actualizar, buscar.
		9	Botón Actualizar	Verificar que sólo se habilite cuando existan registros. Verificar que al arrancar la pantalla de administración de auspiciantes se encuentre desactivado. Verificar que al presionar se muestre un panel de tipo modal para la edición de los campos.
		10	Botón Cancelar	Validar que al presionar se direcciona hacia la pantalla principal. Verificar que todo el tiempo se encuentre activado.
		11	Botón Ingresar	Validar que al ser presionado se limpien los campos y se habiliten.
		12	Botón Eliminar	Validar que al ser presionado se muestre un dialogo de confirmación.

N.HU	HU.	N. EVENTO	EVENTO A PROBAR	RESULTADO ESPERADO
		13	Botón Buscar	Validar que al ser presionado se carguen los campos con el resultado de la búsqueda.
		14	Tabla de registros	Validar que la tabla se cargue al cargar la pantalla de administración de auspiciantes. Validar que se actualice cuando se realice algún cambio en la base de datos.
11	Administración de Proyectos	15	Ingresar a la pantalla de administración de proyectos	Validar que el sistema permita visualizar al usuario los siguientes campos: (campos para ingreso de datos); tabla de registros; los siguientes botones: cancelar, insertar, eliminar, actualizar, buscar.
		16	Botón Actualizar	Verificar que sólo se habilite cuando existan registros. Verificar que al arrancar la pantalla de administración de auspiciantes se encuentre desactivado. Verificar que al presionar se muestre un panel de tipo modal para la edición de los campos.
		17	Botón Cancelar	Validar que al presionar se direcciona hacia la pantalla principal. Verificar que todo el tiempo se encuentre activado.
		18	Botón Ingresar	Validar que al ser presionado se limpien los campos y se habiliten.
		19	Botón Eliminar	Validar que al ser presionado se muestre un dialogo de confirmación.
		20	Botón Buscar	Validar que al ser presionado se carguen los campos con el resultado de la búsqueda.
		21	Tabla de registros	Validar que la tabla se cargue al cargar la pantalla de administración de auspiciantes. Validar que se actualice cuando se realice algún cambio en la base de datos.

N.HU	HU.	N. EVENTO	EVENTO A PROBAR	RESULTADO ESPERADO
16	Administración de Empleados	22	Ingresar a la pantalla de administración de empleados	Validar que el sistema permita visualizar al usuario los siguientes campos: (campos para ingreso de datos); tabla de registros; los siguientes botones: cancelar, insertar, eliminar, actualizar, buscar.
		23	Botón Actualizar	<p>Verificar que sólo se habilite cuando existan registros.</p> <p>Verificar que al arrancar la pantalla de administración de auspiciantes se encuentre desactivado.</p> <p>Verificar que al presionar se muestre un panel de tipo modal para la edición de los campos.</p>
		24	Botón Cancelar	<p>Validar que al presionar se direcciona hacia la pantalla principal.</p> <p>Verificar que todo el tiempo se encuentre activado.</p>
		25	Botón Ingresar	Validar que al ser presionado se limpien los campos y se habiliten.
		26	Botón Eliminar	Validar que al ser presionado se muestre un dialogo de confirmación.
		27	Botón Buscar	Validar que al ser presionado se carguen los campos con el resultado de la búsqueda.
		28	Tabla de registros	<p>Validar que la tabla se cargue al cargar la pantalla de administración de auspiciantes.</p> <p>Validar que se actualice cuando se realice algún cambio en la base de datos.</p>

Tabla 41 PLAN DE PRUEBAS DE ACEPTACIÓN

3.4.2. INFORME PARA IMPLANTACIÓN

El presente proyecto de software está orientado hacia sistemas computacionales que cuenten con las siguientes características:

1. PC de escritorio o a su vez equipos portátiles que cuenten con:
 - a. Una resolución de pantalla mínima de 1600 x 900 pixeles
 - b. Memoria RAM recomendada 1GB para terminales y 4 GB mínimo para el servidor
 - c. Espacio de disco duro 80 GB
 - d. Al haber realizado las pruebas con el navegador Google Chrome Versión 51 se recomienda manejar el aplicativo con dicha versión

2. El equipo que funcionará como servidor de aplicaciones deberá contar con lo siguiente:
 - a. Motor de base de datos PostgreSQL en la versión 9.5
 - b. Servidor de aplicaciones Wildfly (JBoss comunitario) en la versión 8
 - c. JDK Java en la versión 1.8 instalada

3.5. CIERRE DEL PROYECTO

Como parte fundamental del cierre del proyecto se preparó el siguiente análisis donde se detalla brevemente los problemas encontrados durante el desarrollo así como las cosas que se realizaron bien.

Dentro de los principales problemas que se pudo detectar durante el desarrollo del trabajo se podría citar los siguiente:

- a) Para la selección del servidor de aplicaciones no se tuvo en cuenta el cambio que el proveedor realizó, cortando el soporte gratuito que tenía. Motivo por el cual la preparación del ambiente sufrió un leve retraso.
- b) Una falla en el análisis para la coordinación de los horarios entre los usuario y el autor del presente trabajo fue determinante y produjo un leve retraso en el análisis y validación de las distintas funcionalidades del sistema informático.

Dentro de lo que se hizo bien durante el desarrollo del aplicativo se puede destacar lo siguiente:

- a) La correcta selección del lenguaje de programación aceleró el desarrollo del sistema pues Java es un lenguaje del dominio del autor motivo por el cual los posibles retrasos, previamente descritos, fueron solventados ahorrando tiempo en la codificación del sistema.
- b) Dentro de la codificación la correcta estructuración de las interfaces, usadas para la definición de los métodos, fue un punto importante pues permitió la reutilización del código de manera más ágil.
- c) El motor de base de datos seleccionado fue un punto importante dentro del desarrollo pues el mismo cuenta con mucha documentación lo cual facilitó la codificación de programas dentro del mismo.

Adicionalmente, para el cierre del presente proyecto de disertación se ha preparado tres documentos que servirán de guía tanto para el uso del aplicativo como para entender la lógica de la programación aplicada. Además, se puede ver en el manual de instalación los pasos seguidos para la puesta en marcha del sistema informático.

3.5.1. DOCUMENTACIÓN

A continuación se presenta tres documentos que funcionarían como una guía base tanto para el uso del aplicativo como para el entendimiento de la lógica de programación que se utilizó durante el desarrollo del proyecto.

3.5.1.1. MANUAL TÉCNICO

El presente manual contiene una breve descripción de la lógica de programación utilizada por el desarrollador del sistema informático y además una breve descripción de las tecnologías utilizadas en la codificación del mismo con el fin de ubicar de mejor manera al lector, que deberá poseer conocimientos técnicos, dentro del funcionamiento e integración de los diferentes componentes del sistema.

Para más detalles ver el anexo 2: Manual Técnico

3.5.1.2. MANUAL DE USUARIO.

El presente manual tiene como finalidad explicar de una manera gráfica la dinámica del funcionamiento de las diferentes opciones del sistema. Pasando desde el ingreso al sistema hasta el registro de roles de pago. Al ser un manual dirigido al usuario final del sistema dentro del mismo no se incluyó definiciones técnicas del funcionamiento pero si se incluyó de manera gráfica los pasos a seguir para la ejecución de cada una de las opciones.

Para más detalles ver el anexo 3: Manual De Usuario

3.5.1.3 MANUAL DE INSTALACIÓN

El siguiente manual está orientado a usuarios con conocimientos básicos sobre tecnología. Dentro del mismo se puede observar metódicamente los pasos seguidos por el autor del proyecto para la preparación del ambiente para la puesta en marcha del sistema.

Ver Anexo 4: Manual De Instalación

CAPÍTULO: IV

CONCLUSIONES Y RECOMENDACIONES

Al finalizar el presente proyecto informático es importante resaltar ciertas precisiones bajo los siguientes criterios:

4.1. CONCLUSIONES

- El trabajo conjunto del autor del presente trabajo y los responsables del departamento financiero facilitó la comprensión de los procesos financieros que intervienen en el control del presupuesto. De esta manera se lograron identificar los principales problemas y los puntos más críticos del proceso con el fin de optimizar el uso de los recursos tecnológicos y el talento humano.
- Los conocimientos financieros adquiridos en la institución educativa de la que fue parte el autor facilitaron la identificación de los principales problemas así como el enfoque más apropiado de la solución tecnológica. Quedando demostrado que los profesionales de la ingeniería de sistemas computacionales deben poseer, además de sus conocimientos técnicos, la facilidad de adquirir nuevos conocimientos y adaptarlos a las soluciones tecnológicas que sean requeridas.
- El uso de hojas de cálculo para manejar toda la administración de una fundación de tamaño medio a largo plazo, llega a ser contraproducente pues la búsqueda de información o la generación de informes resulta, muchas veces, una tarea titánica pues es necesario buscar en diferentes hojas y obtener manualmente los cálculos. Además, la redundancia de la información es evidente pues cada hoja de cálculo debe contener toda la información a pesar de que ésta se repita. Esto último resulta en un mal uso de los recursos tecnológicos.
- Los conocimientos técnicos adquiridos durante la carrera fueron fundamentales para el desarrollo del proyecto; pues sin los conocimientos de Nuevas Técnicas de Programación no se habría podido analizar la experiencia del usuario al usar el sistema; sin los conocimientos de Bases de Datos no habría podido generar las funciones para el cálculo de los saldos; sin los conocimientos de Ingeniería de Software no se habría sido posible documentar formalmente el proyecto; sin los conocimientos adquiridos en Programación Avanzada no habría sido posible pensar en el desarrollo multicapas y en el uso de técnicas para el ágil desarrollo de aplicativos relacionados con bases de datos; y por último pero no menos importante sin los conocimientos de Programación Orientada a Objetos no se habría podido desarrollar el sistema pues los mismo fueron la base para la transformación del problema en solución tecnológica.
- Se evidenció la importancia de escoger, para el desarrollo del proyecto, motores de base de datos, frameworks de desarrollo, IDEs de desarrollo, que cuenten con una amplia documentación pues esto resulta muy importante cuando se necesita resolver problemas que pueden ir desde preparación del ambiente de desarrollo hasta la ejecución del producto.

4.2. RECOMENDACIONES

- Es recomendable que las fundaciones sin fines de lucro, que buscan ayudar a la comunidad, deberían ocupar un lugar especial en las asignaciones no solo de temas de disertación sino también en el desarrollo de proyectos con la comunidad con el fin de que cada estudiante pueda aportar con un grano de arena en esta labor muy importante.

- Se recomienda implementar un estudio más detallado que permita analizar de mejor manera la integración de todos los aplicativos que el departamento contable utiliza para la administración de la fundación.

- Es recomendable tener muy en cuenta el perfil de la institución para la que se está desarrollando un producto de software pues no es lo mismo desarrollar para fundaciones sin fines de lucro, las cuales obtienen sus fondos de donaciones principalmente, que desarrollar para instituciones que cuentan con financiamiento propio; ya que estas últimas podrían tener menos problemas en costear el soporte técnico o en la renovación de equipos.

- Se recomienda continuar con el presente proyecto, creando e integrando un módulo que permita registrar los documentos financieros producidos con el fin de mejorar la administración de los mismos.

- Se recomienda trabajar en el desarrollo de un módulo que permita administrar las campañas de promoción que la fundación tiene para sus diferentes proyectos de ayuda a la comunidad con el fin de incrementar la ayuda recibida.

ANEXO 1: PLAN DE ENTREGAS

PLAN DE ENTREGAS DEL PROYECTO DE DISERTACIÓN
AUTOR: PABLO HERRERA
METODOLOGÍA USADA: Extremme Programming
NUMERO DE ITERACIONES: 3

IT/DETALLE	MES DE DESARROLLO: 01 DE MARZO - 30 JUNIO DEL 2016.			
	HISTORIA N°	DESCRIPCIÓN	FECHA ENTREGA	ENTREGABLE
ITERACIÓN 1	HU N° 2	Ingresar Auspiciante	15/03/2016	Diseño lógico de interfaces primera iteración Modelo de datos primera iteración Código fuente primera iteración
	HU N° 3	Actualizar Auspiciante		
	HU N° 4	Eliminar Auspiciante		
	HU N° 5	Consultar Auspiciante		
	HU N° 11	Ingresar Proyectos	20/03/2016	
	HU N° 12	Actualizar Proyectos		
	HU N° 13	Eliminar Proyectos		
	HU N° 14	Consultar Proyectos		
	HU N° 16	Ingresar Empleados	25/03/2016	
	HU N° 17	Actualizar Empleados		
	HU N° 18	Eliminar empleados		
	HU N° 19	Consultar Empleados		
	HU N° 21	Ingresar Cuentas Bancarias	30/04/2016	
	HU N° 22	Actualizar Cuentas Bancarias		
HU N° 23	Eliminar Cuentas Bancarias			
HU N° 24	Consultar Cuentas Bancarias			

IT/DETALLE	MES DE DESARROLLO: 01 DE MARZO - 30 JUNIO DEL 2016.			
	HISTORIA N°	DESCRIPCIÓN	FECHA ENTREGA	ENTREGABLE
ITERACIÓN 2	HU N° 7	Ingresar Roles de pago	20/04/2016	Diseño lógico de interfaces primera iteración Modelo de datos primera iteración. Código fuente primera iteración.
	HU N° 8	Eliminar Roles de pago		
	HU N° 9	Consultar Roles de pago		
ITERACIÓN 3	HU N° 26	Ingresar Gastos	15/06/2016	Diseño lógico de interfaces primera iteración Modelo de datos primera iteración Código fuente primera iteración
	HU N° 27	Eliminar Gastos		
	HU N° 28	Consultar Gastos		
	HU N° 30	Ingresar Usuarios	20/06/2016	
	HU N° 31	Actualizar Usuarios		
	HU N° 32	Eliminar Usuarios		
	HU N° 33	Consultar Usuarios		
HU N° 34	Inf. Ejecución del presupuesto			

ANEXO 2: MANUAL TÉCNICO

Contenido

OBJETIVO.....	87
1. DESCRIPCIÓN DEL AMBIENTE DE DESARROLLO.....	87
2. INTERFAZ DE USUARIO.....	87
2.1 PÁGINA MAESTRA.....	87
2.2. PÁGINA LIGADA.....	88
3. MANEJO DE OBJETOS MAPEADOS DE LA BASE DE DATOS.....	88
3.1 INTERFAZ.....	88
3.2 IMPLEMENTACIÓN.....	88
4. CONTROLADORES.....	90

Autor: Pablo Herrera

Quito, 04 de julio del 2016

OBJETIVO

El objetivo del presente manual es ubicar un poco al lector en la lógica del desarrollo de aplicativo

1. DESCRIPCIÓN DEL AMBIENTE DE DESARROLLO

- El aplicativo se desarrolló usando la tecnología Java EE
- Se utilizó el IDE de desarrollo Eclipse en su versión MARS
- En el presente trabajo se utilizó Hibernate para el mapeo de clases

2. INTERFAZ DE USUARIO

2.1 PÁGINA MAESTRA

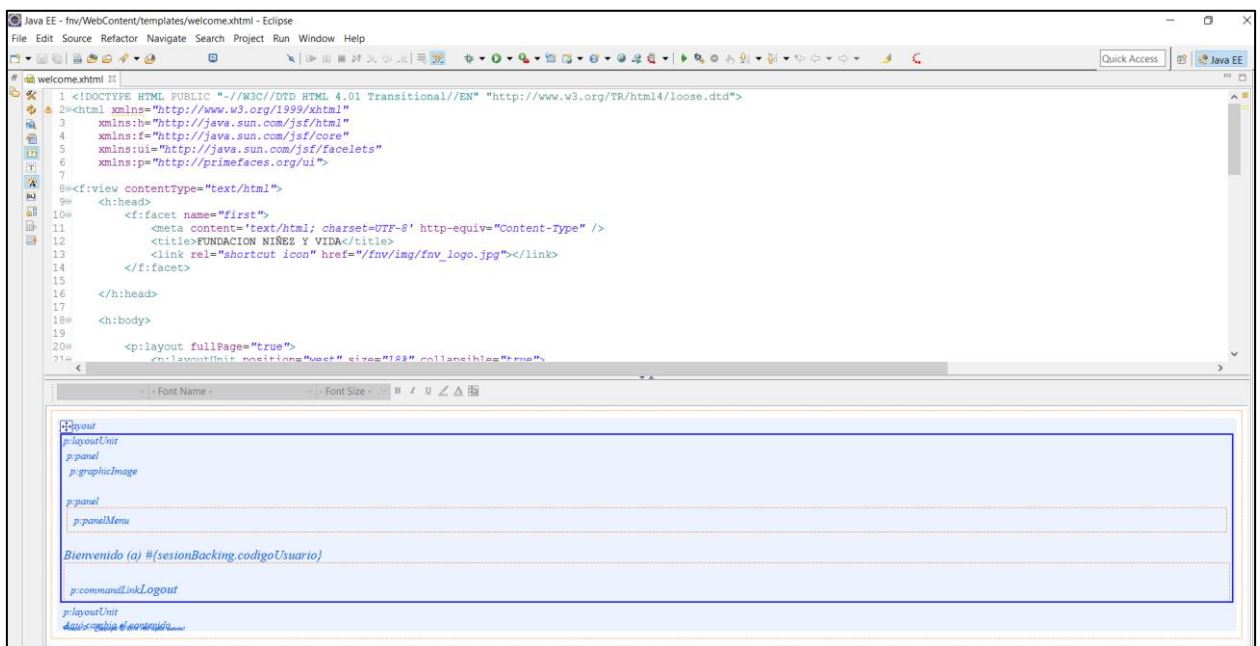


Ilustración 30 ANEXO 2: PÁGINA MAESTRA

Se usó un template para el contenido de la parte izquierda del aplicativo.

2.2. PÁGINA LIGADA



```
1<html xmlns="http://www.w3.org/1999/xhtml"
2  xmlns:ui="http://java.sun.com/jsf/facelets"
3  xmlns:f="http://java.sun.com/jsf/core"
4  xmlns:h="http://java.sun.com/jsf/html"
5  xmlns:p="http://primefaces.org/ui">
6
7<!--
8  Replace path to template, title, header and body
9  with actual data.
10 -->
11<ui:composition template="/templates/welcome.xhtml">
12  <ui:define name="pageTitle"></ui:define>
13  <ui:define name="pageHeader"></ui:define>
14  <ui:define name="contenido">
15    <div style="font: Goudy Stout">
16      <p:panel
17        style="font-family: 'Franklin Gothic Medium', 'Franklin Gothic', 'ITC Franklin Gothic', Arial, sans-serif;
18        font-size: 200%;
19        font-style: normal;
20        font-variant: normal;
21        font-weight: 500;
22        line-height: 26.4px;">
23        <font color="2046B0">MODULO: EMPLEADOS</font>
24      </p:panel>
```

Ilustración 31 ANEXO 2: PÁGINA LIGADA

Cada página xhtml está ligada al template mediante el uso del tag **<ui:define name="contenido">**

3. MANEJO DE OBJETOS MAPEADOS DE LA BASE DE DATOS

3.1 INTERFAZ

Contiene la descripción de los métodos que se usaran para manejar la clase.

```
package fnv.service;

import java.util.List;

import javax.ejb.Local;

import fnv.model.CuentaBanco;

@Local
public interface CuentaBancoServicio {

    List<CuentaBanco> findAll();
    void crearCuentaBanco(CuentaBanco cuenta) throws Exception;
    void actualizarCuentaBanco(CuentaBanco cuenta);
    void eliminarCuentaBanco(long codigoCuentaBanco);
    CuentaBanco buscarCuentaBancoPorCodigo(long codigo);
}
```

Ilustración 32 ANEXO 2: DESCRIPCIÓN DE INTERFAZ

3.2 IMPLEMENTACIÓN

Contiene la implementación de la interfaz ya descrita

```
package fnv.service.impl;
import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;
import fnv.model.CuentaBanco;
import fnv.service.CuentaBancoServicio;
@Stateless
public class CuentaBancoServicioImpl implements
CuentaBancoServicio {
    public CuentaBancoServicioImpl() {

    }
    @PersistenceContext(unitName = "fnv")
    protected EntityManager em;

    @SuppressWarnings("unchecked")
    @Override
    public List<CuentaBanco> findAll() {
        // return empleadoDao.findAll();
        Query query = em.createQuery("SELECT e FROM CuentaBanco
e");
        return (List<CuentaBanco>) query.getResultList();
    }

    @Override
    public void crearCuentaBanco(CuentaBanco empleado) throws
Exception {em.persist(empleado); }
    @Override
    public void actualizarCuentaBanco(CuentaBanco empleado) {
        em.merge(empleado); }
    @Override
    public void eliminarCuentaBanco(long codigoCuentaBanco) {
        em.remove(buscarCuentaBancoPorCodigo(codigoCuentaBanco));
    }
    @Override
    public CuentaBanco buscarCuentaBancoPorCodigo(long codigo) {
        // return empleadoDao.findByPk(codigo);
        return em.find(fnv.model.CuentaBanco.class, codigo);
    }
}
```

Ilustración 33 ANEXO 2: EJEMPLO DE IMPLEMENTACIÓN DE INTERFAZ DE SERVICIO

4. CONTROLADORES

En el proyecto se los conocerá como los archivos **Backing** los cuales crean la interacción directa con el usuario.

Extracto del ingreso de un nuevo Empleado

```
<h:form id="frmNuevoEmpleado">
    <p:dialog id="dialogoNuevoEmpleado"
widgetVar="dlgNuevoEmpleado"
        header="Creación Nuevo Empleado" >
        <h:panelGrid columns="2"
cellpadding="5" style="font-size: 80%">
            <h:outputText value="Tipo de
Documento: " />
            <p:selectOneMenu
id="somTipoDocumento"
value="#{empleadoBacking.empleado.empTipoIdentificacion}"
style="width:125px">
                <f:selectItem itemLabel="- Seleccione
-" itemValue="" />
                <f:selectItem itemLabel="Cédula"
itemValue="Cédula" />
                <f:selectItem itemLabel="Otro"
itemValue="Otro" />
                <f:selectItem itemLabel="Pasaporte"
itemValue="Pasaporte" />
            </p:selectOneMenu>
            <h:outputText value="Identificación:
*" />
            <p:inputText
value="#{empleadoBacking.empleado.empIdentificacion}" />
            <h:outputText value="Nombres: *" />
            <p:inputText
value="#{empleadoBacking.empleado.empNombres}" />
            <h:outputText value="Apellidos: *" />
            <p:inputText
value="#{empleadoBacking.empleado.empApellidos}" />
            <h:outputText value="Telefono: " />
            <p:inputText
value="#{empleadoBacking.empleado.empTelefono}" />
            <h:outputText value="Celular: " />
            <p:inputText
value="#{empleadoBacking.empleado.empCelular}" />
            <h:outputText value="Dirección" />
            <p:inputText
value="#{empleadoBacking.empleado.empDireccion}" />
            <h:outputText value="Fecha de
Nacimiento: *" />
            <p:calendar
value="#{empleadoBacking.empleado.empFechaNacimiento}" pattern="dd-MM-
yyyy" mask="true" />
            <h:outputText value="E-Mail: " />
            <p:inputText id="email1"
value="#{empleadoBacking.empleado.empEmail}" />
        </h:panelGrid>
    </p:dialog>
</h:form>
```

```

        <f:validateRegex pattern="^[_A-Za-z0-9-
\+]+\(\.[_A-Za-z0-9-]+\)*@[A-Za-z0-9-]+\(\.[_A-Za-z]{2,})$"
/>
        </p:inputText>
        <h:outputText value="Tipo de
empleado: " />
        <p:selectOneMenu id="somTipoEmpleado"
value="#{empleadoBacking.empleado.empTipo}" style="width:125px">
        <f:selectItem itemLabel="- Seleccion
-" itemValue="" />
        <f:selectItem itemLabel="Tiempo
Completo" itemValue="Tiempo Completo" />
        <f:selectItem itemLabel="Tiempo
Parcial" itemValue="Tiempo Parcial" />
        </p:selectOneMenu>
        <h:outputText value="Nacionalidad: "
/>
        <p:inputText
value="#{empleadoBacking.empleado.empNacionalidad}" />
        <h:outputText value="Genero: " />
        <p:selectOneMenu id="somGenero"
value="#{empleadoBacking.empleado.empGenero}" style="width:125px">
        <f:selectItem itemLabel="- Seleccion
-" itemValue="" />
        <f:selectItem itemLabel="Masculino"
itemValue="M" />
        <f:selectItem itemLabel="Femenino"
itemValue="F" />
</p:selectOneMenu>
        <h:outputText value="Cargo" />
        <p:inputText
value="#{empleadoBacking.empleado.empCargo}" />
        <h:outputText value="Sueldo: " />
        <p:inputText
value="#{empleadoBacking.empleado.empSueldo}" />
        <f:facet name="footer">
        <p:commandButton
id="btnGuardarNuevo" value="Guardar"
        actionListener="#{empleadoBacking.crearEmpleado()}"
        update="frmNuevoEmpleado :formEdit" />
        <p:commandButton
id="btnCancelarNuevo" value="Cancelar"
        oncomplete="PF('dlgNuevoEmpleado').hide()" />
        </f:facet>
        </h:panelGrid>
    </p:dialog>
</h:form>

```

Ilustración 34 ANEXO 2: EXTRACTO CÓDIGO XHTML EMPLEADO

Controlador de la página, extracto del ingreso de un nuevo Empleado.

```

public void crearEmpleado() {
    if(!this.empleado.getEmpApellidos().equals("")){
        String[] apellidos = new String[3];
        apellidos = this.empleado.getEmpApellidos().split(" ");
        String empCodigo =
this.empleado.getEmpNombres().substring(0,
1).toUpperCase()+apellidos[0].toUpperCase()+ (this.empleadoServicio.findAll().size()+1);
        empleado.setEmpCodigo(empCodigo);
    }

    if(this.empleado.getEmpTipoIdentificacion().equals("Cédula")){
        if(validadorDeCedula(empleado.getEmpIdentificacion())){
            empleadoServicio.crearEmpleado(empleado);
            log.setLogAccion("Crear Auspiciante: "
+empleado.getEmpCodigo());
            logServicio.registroLog(log);
            empleado = new Empleado();
            empleados = empleadoServicio.findAll();
            FacesMessage msg = new
FacesMessage(FacesMessage.SEVERITY_INFO,
                "Creación Exitosa",
                "Se ha creado un nuevo Empleado
 exitosamente!");

            FacesContext.getCurrentInstance().addMessage(null, msg);
        }
        else{
            FacesMessage msg = new
FacesMessage(FacesMessage.SEVERITY_INFO,
                "ERROR ",
                "Ingrese una cedula valida!");

            FacesContext.getCurrentInstance().addMessage(null, msg);
        }
    }
    else{
        empleadoServicio.crearEmpleado(empleado);
        log.setLogAccion("Crear Auspiciante: "
+empleado.getEmpCodigo());
        logServicio.registroLog(log);
        empleado = new Empleado();
        empleados = empleadoServicio.findAll();
        FacesMessage msg = new
FacesMessage(FacesMessage.SEVERITY_INFO,
                "Creación Exitosa",
                "Se ha creado un nuevo Empleado
 exitosamente!");
        FacesContext.getCurrentInstance().addMessage(null,
msg);
    }
}
}

```

Ilustración 35 ANEXO 2: EXTRACTO DE CÓDIGO DEL CONTROLADOR EMPLEADO

ANEXO 3: MANUAL DE USUARIO

CONTENIDO

OBJETIVO.....	94
1. AUSPICIANTE.....	94
1.1 INGRESAR.....	94
1.2 EDITAR.....	95
1.3 ELIMINAR.....	96
1.4 AÑADIR CUENTA BANCARIA.....	97
2. EMPLEADO.....	100
2.1 INGRESAR.....	100
2.2 ASIGNAR PROYECTO.....	101
3. PROYECTO.....	102
3.1 INGRESAR.....	103
3.2 ASIGNAR EMPLEADOS AL PROYECTO.....	104
3.3 CUENTA BANCARIA AL PROYECTO.....	105
4. INGRESAR CUENTA BANCARIA.....	106
5. ROLES DE PAGO.....	106
5.1 INGRESO DE ROL DE PAGO.....	107
5.2 IMPRESIÓN DE ROLES: CONSOLIDADO PERSONAL.....	109
5.3 IMPRESIÓN DE ROLES: PERSONAL.....	109
5.4 IMPRESIÓN DE ROLES: CONSOLIDADO GLOBAL POR CUENTA.....	110
6. INGRESO DE GASTOS.....	111
7. INFORME DE EJECUCIÓN DEL PRESUPUESTO.....	111
8. USUARIO.....	112
8.1 INGRESO.....	113
8.3 MODIFICACIÓN DE USUARIO.....	113
9. INGRESO Y SALIDA DEL APLICATIVO.....	114
9.2 INGRESAR AL SISTEMA.....	114
9.1 SALIR DEL SISTEMA.....	115

Autor: Pablo Herrera

Quito, 04 de julio del 2016

OBJETIVO

El objetivo de este manual es brindar al usuario una guía para el uso de las funciones del aplicativo.

1. AUSPICIANTE

1.1 INGRESAR

The screenshot shows the 'MODULO: AUSPICIANTES' interface. On the left is a navigation menu with options like 'MENU', 'EMPLEADO', 'AUSPICIANTES', 'PROYECTOS', 'CTAS. BANCARIAS', 'GESTION FINANCIERA', and 'ADM. SISTEMA'. The main area displays a table titled 'Lista de auspiciantes.' with columns: CÓDIGO, NOMBRE, NACIONALIDAD, TELEFONO, E - MAIL, ESTADO, and CUENTAS B. A '+ Nuevo' button is highlighted with a red box. The table contains three rows of data.

CÓDIGO	NOMBRE	NACIONALIDAD	TELEFONO	E - MAIL	ESTADO	CUENTAS B.
TDH-SUIZASUIZA	TDH-SUIZA	SUIZA		TDHSUIZA@MAI	Activo	✓
FLORENCE DE GOUMOENCESUIZA	FLORENCE DE GOUMOENCE	SUIZA		FGOUMOENCE@	Inactivo	✓
SUSETTE GOFFNORTE AMERICANA	SUSETTE GOFF	NORTE AMERICANA	12312	SGOFF@MAIL.C	Activo	✓

Ilustración 36 ANEXO 3: MÓDULO AUSPICIANTE

Llenar todos los campos.

The screenshot shows the 'Creación Nuevo Auspiciante' form. It contains the following fields and values:

- Código:
- Nombre: * Prueba
- Nacionalidad: * Ecuatoriana
- Telefono: 0223242526
- E - MAIL: * prueba@mail.com
- Estado del auspiciante: Activo

At the bottom, there are 'Guardar' and 'Cancelar' buttons.

Ilustración 37 ANEXO 3: NUEVO AUSPICIANTE

MODULO: AUSPICIANTES

+ Nuevo

Lista de auspiciantes. Columnas

CÓDIGO	NOMBRE	NACIONALIDAD	TELEFONO	E - MAIL	ESTADO	CUENTAS B.	
TDH-ITALIAITALIA	TDH-ITALIA	ITALIA	111	TDHITALIA@MAI	Activo	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
FEDEVACO SUIZASUIZA	FEDEVACO SUIZA	SUIZA	11	FEDEVACO@MAI	Activo	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
Prueba Ecuatoriana	Prueba	Ecuatoriana	223242526	prueba@mail.com	Activo	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Bienvenido (a) BMORA2
[Logout](#)

© 2016. All rights reserved.

Ilustración 38 ANEXO 3: VALIDACIÓN NUEVO REGISTRO

1.2 EDITAR

MODULO: AUSPICIANTES

+ Nuevo

Lista de auspiciantes. Columnas

CÓDIGO	NOMBRE	NACIONALIDAD	TELEFONO	E - MAIL	ESTADO	CUENTAS B.	
TDH-ITALIAITALIA	TDH-ITALIA	ITALIA	111	TDHITALIA@MAI	Activo	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
FEDEVACO SUIZASUIZA	FEDEVACO SUIZA	SUIZA	11	FEDEVACO@MAI	Activo	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
Prueba Ecuatoriana	Prueba	Ecuatoriana	223242526	prueba@mail.com	Activo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>

Haz clic para editar el Auspiciante actual

Bienvenido (a) BMORA2
[Logout](#)

© 2016. All rights reserved.

Ilustración 39 ANEXO 3: EDICIÓN DE AUSPICIANTE

Edición de Auspiciante ✕

Código: Prueba Ecuatoriana

Nombre: *

Nacionalidad: *

Telefono:

E - MAIL: *

Estado del auspiciante: ▼

Ilustración 40 ANEXO 3: EDICIÓN CAMPOS AUSPICIANTE

MODULO: AUSPICIENTES

1 Actualización Exitosa
 Auspiciante: Prueba Ecuatoriana

+ Nuevo

Lista de auspiciantes. ■ Columnas

CÓDIGO	NOMBRE	NACIONALIDAD	TELEFONO	E - MAIL	ESTADO	CUENTAS B.	
TDH-ITALIAITALIA	TDH-ITALIA	ITALIA	111	TDHITALIA@MAI	Activo	✓	✓ -
FEDEVACO SUIZASUIZA	FEDEVACO SUIZA	SUIZA	11	FEDEVACO@MA	Activo	✓	✓ -
Prueba Ecuatoriana	Prueba	Ecuatoriana	223242526	prueba@mail.com	Inactivo	✓	✓ -

Bienvenido (a) BMORA2
© 2016 All rights reserved.

Ilustración 41 ANEXO 3: VALIDACIÓN DE EDICIÓN

1.3 ELIMINAR

MODULO: AUSPICIENTES

+ Nuevo

Lista de auspiciantes. ■ Columnas

CÓDIGO	NOMBRE	NACIONALIDAD	TELEFONO	E - MAIL	ESTADO	CUENTAS B.	
TDH-ITALIAITALIA	TDH-ITALIA	ITALIA	111	TDHITALIA@MAI	Activo	✓	✓ -
FEDEVACO SUIZASUIZA	FEDEVACO SUIZA	SUIZA	11	FEDEVACO@MA	Activo	✓	✓ -
Prueba Ecuatoriana	Prueba	Ecuatoriana	223242526	prueba@mail.com	Inactivo	✓	✓ -

Bienvenido (a) BMORA2
© 2016 All rights reserved.

Ilustración 42 ANEXO 3: BOTÓN ELIMINAR AUSPICIANTE



Ilustración 43 ANEXO 3: CONFIRMACIÓN DE BORRADO

1.4 AÑADIR CUENTA BANCARIA



Ilustración 44 ANEXO 3: BOTÓN CUENTAS



Ilustración 45 ANEXO 3: SELECCIÓN DE CUENTA BANCARIA



Ilustración 46 ANEXO 3: VALIDACIÓN DE INGRESO DE CUENTA

1.4.1 INGRESAR PRESUPUESTO

Funcionalidad que permite ingresar el monto que el auspiciante disponga depositar en dicha cuenta.



Ilustración 47 ANEXO 3: BOTÓN INGRESAR PRESUPUESTO

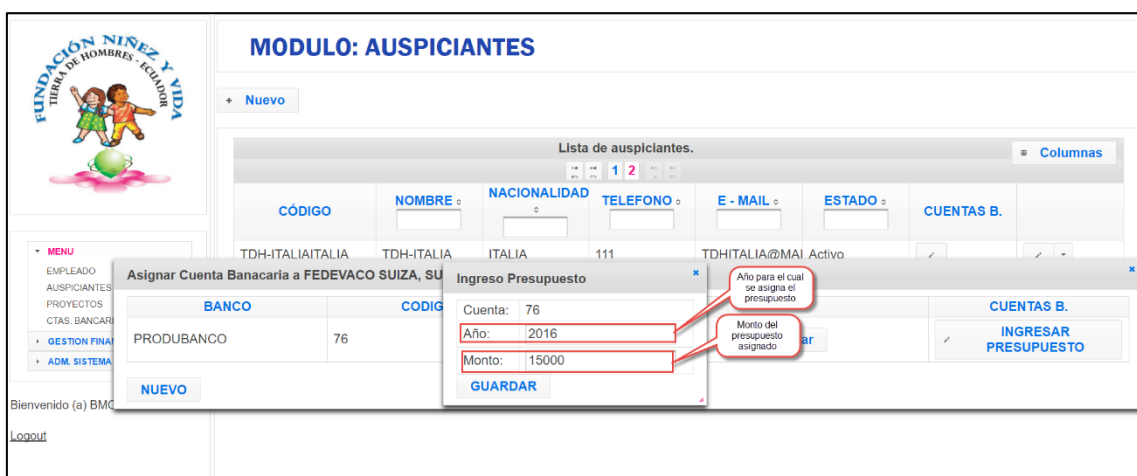


Ilustración 48 ANEXO 4: LLENADO DE CAMPOS PARA INGRESO DE PRESUPUESTO



Ilustración 49 ANEXO 3: VENTANA DE CONFIRMACIÓN DE INGRESO DE PRESUPUESTO

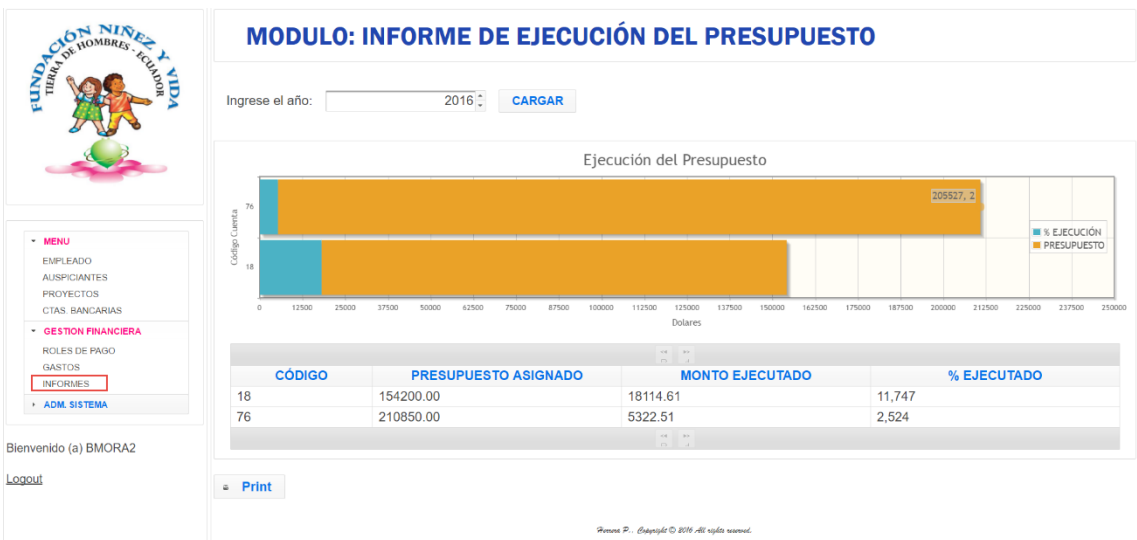


Ilustración 50 ANEXO 3: VALIDACIÓN DEL INGRESO DEL PRESUPUESTO

2. EMPLEADO

MÓDULO: EMPLEADOS

+ Nuevo

Lista de empleados. Columnas

N. ID	NOMBRES	APELLIDO	TIPO DE EMPLEADO	GENERO	CARGO
1704386836	CESAR EDMUNDO	AYALA TORRES	Tiempo Completo	M	DISEÑADOR DE MATERIAL DIDACTICO
1705429486	BERTHA BEATRIZ	MORA LOPEZ	Tiempo Completo	F	ADM. FINANCIERA
1717191330	EDGAR ESTEBAN	VELEZ INGA	Tiempo Completo	M	CONTADOR

Datos del empleado

TELEFONO
 CODIGO EMPLEADO: CAYALA4
 CELULAR
 DIRECCIÓN: QUITO SUR
 F.NACIMIENTO: 1960-01-01
 E-MAIL: MAIL@MAIL.COM
 NACIONALIDAD: ECUATORIANA
 SUELDO: 675.33

PROYECTOS:
 ACCIONES: **Eliminar / Editar**

Ilustración 51 ANEXO 3: MÓDULO EMPLEADO

2.1 INGRESAR

MÓDULO: EMPLEADOS

+ Nuevo

Creación Nuevo Empleado

Tipo de Documento: Pasaporte
 Identificación: prueba125
 Nombres: Pedro
 Apellidos: Acosta
 Telefono: 022222225
 Celular: 0984565656
 Dirección: Calle SN
 Fecha de Nacimiento: 12-05-1980
 E-Mail: pedroa@mail.com
 Tipo de empleado: Tiempo Complet
 Nacionalidad: Ecuatoriana
 Genero: Masculino
 Cargo: Contador
 Sueldo: 650.00

Guardar Cancelar

Ilustración 52 ANEXO 3: INGRESO DE NUEVO EMPLEADO



Ilustración 53 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN NUEVO EMPLEADO

2.2 ASIGNAR PROYECTO

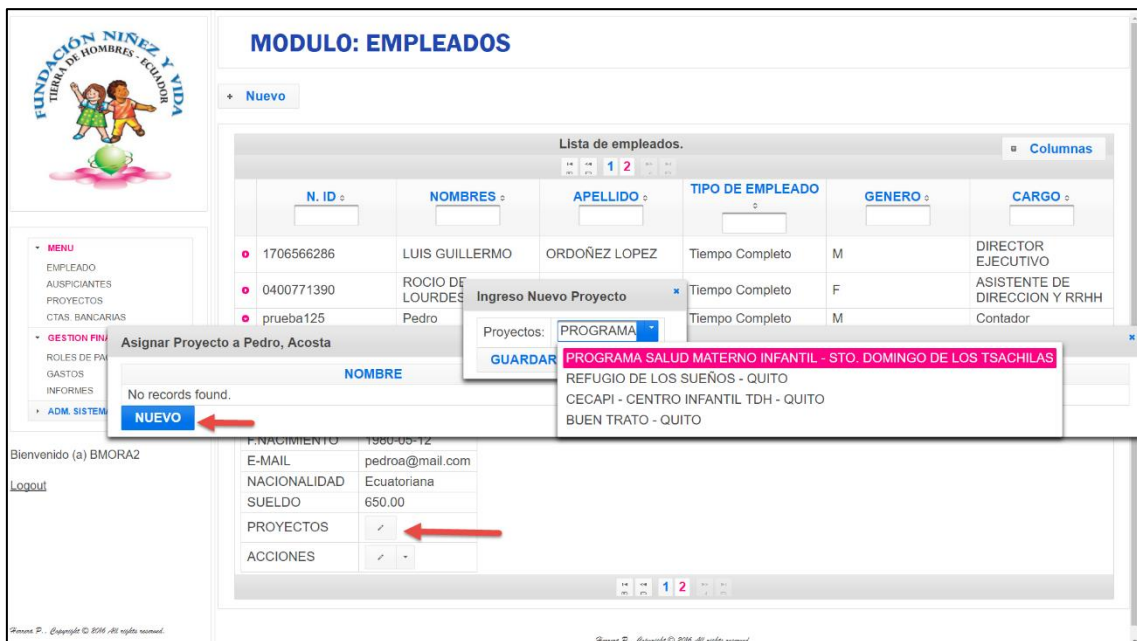


Ilustración 54 ANEXO 3: ASIGNACIÓN DE PROYECTO DESDE EL MÓDULO DE EMPLEADOS

MODULO: EMPLEADOS

+ Nuevo

Lista de empleados. Columnas

N. ID	NOMBRES	APELLIDO	TIPO DE EMPLEADO	GENERO	CARGO
1706566286	LUIS GUILLERMO	ORDOÑEZ LOPEZ	Tiempo Completo	M	DIRECTOR EJECUTIVO
0400771390	ROCIO DE LOURDES	SUAREZ DIAZ	Tiempo Completo	F	ASISTENTE DE DIRECCION Y RRHH
prueba125	Pedro	Acosta	Tiempo Completo	M	Contador

Asignar Proyecto a Pedro, Acosta

NOMBRE: PROGRAMA SALUD MATERNO INFANTIL Eliminar

NUEVO

Bienvenido (a) BMORA2

Logout

E-MAIL	pedroa@mail.com
NACIONALIDAD	Ecuatoriana
SUELDO	650.00
PROYECTOS	/
ACCIONES	/

Ilustración 55 ANEXO 3: VALIDACIÓN DE LA ASIGNACIÓN DE UN PROYECTO A UN EMPLEADO

3. PROYECTO

MODULO: PROYECTOS

+ Nuevo

Lista de proyectos. Columnas

NOMBRE	CIUDAD	ESTADO	F. INICIO	F. FIN	Empleados	CUENTAS B.	Eliminar / Editar
PROGRAMA SALUD MATERNO INFANTIL	STO. DOMINGO DE LOS TSACHILAS	En proceso	2007-01-01	2017-12-31	/	/	/
REFUGIO DE LOS SUEÑOS	QUITO	En proceso	1998-01-01	2020-12-31	/	/	/
CECAPI - CENTRO INFANTIL TDH	QUITO	En proceso	1977-01-01	2017-12-31	/	/	/

Bienvenido (a) BMORA2

Logout

Ilustración 56 ANEXO 3: MÓDULO DE PROYECTOS

3.1 INGRESAR

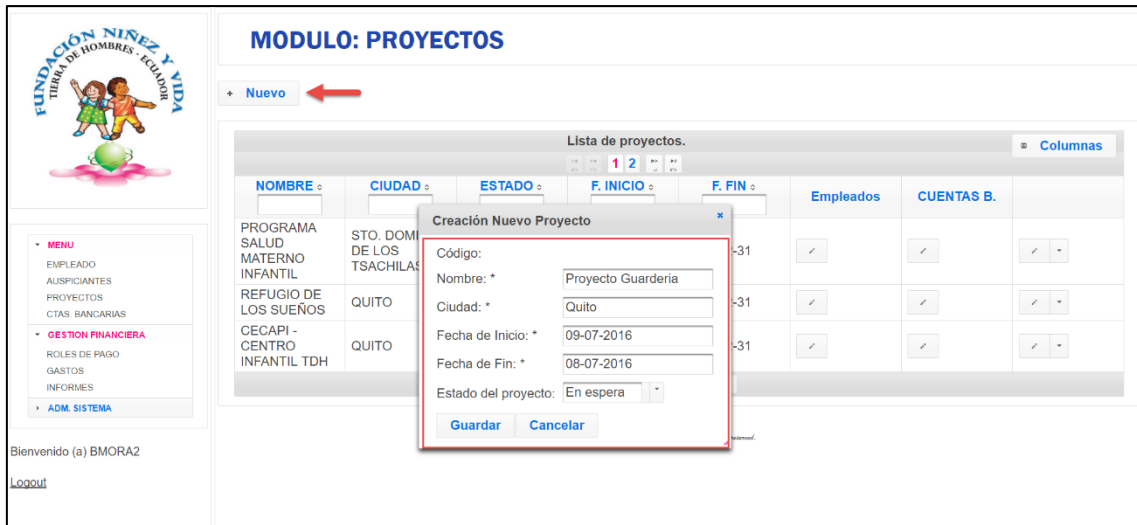


Ilustración 57 ANEXO 3: INGRESO DE NUEVO PROYECTO



Ilustración 58 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN NUEVO PROYECTO

3.2 ASIGNAR EMPLEADOS AL PROYECTO

MODULO: PROYECTOS

+ Nuevo

Lista de proyectos. Columnas

NOMBRE	CIUDAD	ESTADO	F. INICIO	F. FIN	Empleados	CUENTAS B.
BUEN TRATO	QUITO	Finalizado	2008-01-01	2010-12-31	✓	✓
Proyecto Guarderia	Quito	En espera	2016-07-09	2016-07-08	✓	✓

Ingreso Nuevo Proyecto

Proyectos: CESAR EDM

GUARDAR

CESAR EDMUNDO - AYALA TORRES
 BERTHA BEATRIZ - MORA LOPEZ
 EDGAR ESTEBAN - VELEZ INGA
 LUIS GUILLERMO - ORDOÑEZ LOPEZ
 ROCIO DE LOURDES - SUAREZ DIAZ
 Pedro - Acosta

Asignar Empleado a BUEN TRATO, QUITO

N. ID

No records found.

NUEVO

Bienvenido (a) BMORA2
 Logout

Ilustración 59 ANEXO 3: ASIGNACIÓN DE UN EMPLEADO A UN PROYECTO

MODULO: PROYECTOS

+ Nuevo

Lista de proyectos. Columnas

NOMBRE	CIUDAD	ESTADO	F. INICIO	F. FIN	Empleados	CUENTAS B.
BUEN TRATO	QUITO	Finalizado	2008-01-01	2010-12-31	✓	✓
Proyecto Guarderia	Quito	En espera	2016-07-09	2016-07-08	✓	✓

Asignar Empleado a BUEN TRATO, QUITO

N. ID	NOMBRE	APELLIDO
1705429486	BERTHA BEATRIZ	MORA LOPEZ

Eliminar

NUEVO

Bienvenido (a) Bmora2
 Logout

Ilustración 60 ANEXO 3: VALIDACIÓN DE LA ASIGNACIÓN DE UN EMPLEADO A UN PROYECTO

3.3 CUENTA BANCARIA AL PROYECTO

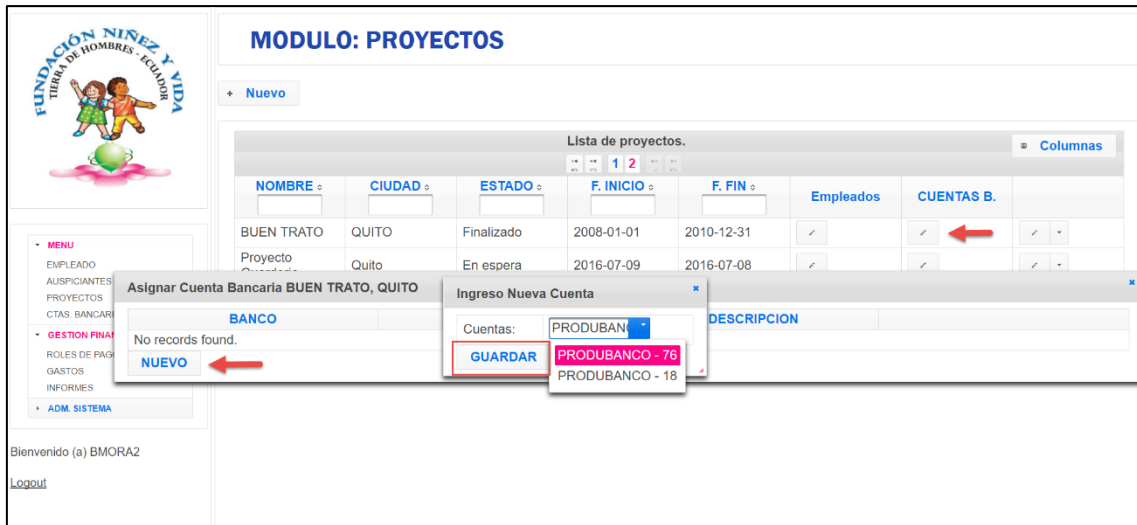


Ilustración 61 ANEXO 3: ASIGNACIÓN DE CUENTA BANCARIA A UN PROYECTO



Ilustración 62 ANEXO 3: VALIDACIÓN DE LA ASIGNACIÓN DE UNA CUENTA BANCARIA A UN PROYECTO

4. INGRESAR CUENTA BANCARIA



Ilustración 63 ANEXO 3: MÓDULO CUENTA BANCARIA. INGRESO NUEVA CUENTA

5. ROLES DE PAGO



Ilustración 64 ANEXO 3: MÓDULO DE ROLES DE PAGO

5.1 INGRESO DE ROL DE PAGO

MODULO: ROLES DE PAGO

MES: 6 AÑO: 2016 CONSOLIDADO

Lista de empleados

N. ID	NOMBRES	SUELDO	
1704386836	CESAR EDMUNDO, AYALA TORRES	675.33	Asignar
1705429486	BERTHA BEATRIZ, MORA LOPEZ	140.00	Asignar
1717191330	EDGAR ESTEBAN, VELEZ INGA	140.00	Asignar

Menú: EMPLEADO, AUSPICIANTES, PROYECTOS, CTAS. BANCARIAS, GESTION FINANCIERA (ROLES DE PAGO, GASTOS, INFORMES), ADM. SISTEMA

Bienvenido (a) BMORAZ
Logout!

Ilustración 65 ANEXO 3: BOTÓN ASIGNAR ROL DE PAGOS

Registro de Rol de Pagos al mes 6 del año 2016

SALDO DISPONIBLE

Gráfico descriptivo del estado del presupuesto asignado en las cuentas bancarias.

Monto del sueldo asignado

+ Nuevo Imprimir Consolidado

Roles de pagos AYALA TORRES, CESAR EDMUNDO. Total del sueldo asignado 0 / 675.33

S. Unificado	Movilización	T. Ingresos	Aporte IESS	Anticipo	T. Recibir	Cuenta
No records found.						

Ilustración 66 ANEXO 3: VENTANA PARA EL INGRESO DE NUEVO ROL DE PAGO

MODULO: ROLES DE PAGO

Registro de Rol de Pagos al mes 6 del año 2016

SALDO DISPONIBLE

Roles de pagos AYALA

Registro de Nuevo Rol

Total Unificado: * 650

Movilización: 0

IESS PERSONAL 9% 45% 61.4250

IESS PATRONAL 12% 15% 78.9750

Descuento 3: 0

Prestamo IESS: 0

Anticipo: 0

Otros Desc. 1: 0

Otros Desc. 2: 0

Imp. Renta: 0

Total a Recibir: 588.5750

Aportado por: 76- PROGRA

76- PROGRAMA SALUD MATERNO INFANTIL STO. DOMINGO DE LOS TSACHILAS

18- REFUGIO DE LOS SUEÑOS QUITO

18- CECAPI - CENTRO INFANTIL TDH QUITO

76- Proyecto Guardería Quito

Ilustración 67 ANEXO 3: CAMPOS PARA EL REGISTRO DE UN ROL DE PAGOS

MODULO: ROLES DE PAGO

Registro de Rol de Pagos al mes 6 del año 2016

Roles de pagos AYALA TORRES, CESAR EDMUNDO . Total del sueldo asignado 650.00 / 675.33

S. Unificado	Movilización	T. Ingresos	Aporte IESS	Anticipo	T. Recibir	Cuenta
650.00	0.00	588.58	61.43	0.00	588.58	REFQUI19982 - 18

Aporte Patronal IESS 78.98

Descuento 3 0.00

Descuento 1 0.00

Descuento 2 0.00

Días no laborados 0

Prestamo IESS 0.00

Imp. Renta 0.00

Ilustración 68 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN ROL DE PAGOS

5.2 IMPRESIÓN DE ROLES: CONSOLIDADO PERSONAL

MODULO: ROLES DE PAGO

Registro de Rol de Pagos al mes 6 del año

SALDO DISPONIBLE

MILES DE DOLARES

10000.0
6666.7
3333.3
0.0

NIVEL SALD

• MENU
• GESTION FINANCIERA
ROLES DE PAGO
GASTOS
INFORMES
• ADM. SISTEMA

Bienvenido (a) BMC

Logout

• Nuevo • Imprimir Consolidado

Roles de Pago

S. Unificado	Movilización
650.00	0.00
100.00	0.00

o asignado 750.00 / 675.33

T. Recibir	Cuenta
588.58	REFQUI19982 - 18
90.55	ProQui20165 - 76

RECIBI CONFORME

Print

Ilustración 69 ANEXO 3: IMPRESIÓN ROL CONSOLIDADO PERSONAL

5.3 IMPRESIÓN DE ROLES: PERSONAL

MODULO: ROLES DE PAGO

Registro de Rol de Pagos al mes 6 del año

SALDO DISPONIBLE

MILES DE DOLARES

10000.0
6666.7
3333.3
0.0

NIVEL SALD

• MENU
• GESTION FINANCIERA
ROLES DE PAGO
GASTOS
INFORMES
• ADM. SISTEMA

Bienvenido (a) BMC

Logout

• Nuevo • Imprimir Consolidado

Roles de Pago

S. Unificado	Movilización
650.00	0.00
100.00	0.00

o asignado 750.00 / 675.33

T. Recibir	Cuenta
588.58	REFQUI19982 - 18
90.55	ProQui20165 - 76

ROL INDIVIDUAL DE PAGOS

APLA TORRES CESAR EDUARDO 2016

IMPRESIONA

RECIBI CONFORME

Print

Ilustración 70 ANEXO 3: IMPRESIÓN ROL DE PAGO PERSONAL

5.4 IMPRESIÓN DE ROLES: CONSOLIDADO GLOBAL POR CUENTA



Ilustración 71 ANEXO 3: IMPRESIÓN ROL CONSOLIDADO GLOBAL



Ilustración 72 ANEXO 3: FORMATO DE IMPRESIÓN ROL CONSOLIDADO GLOBAL

6. INGRESO DE GASTOS

MODULO: GASTOS

+ Nuevo

GASTOS.

N	CODIGO	AÑO	MES	DESCRIPCIÓN	MONTO	
1	PROSTO20071 - 76			16 PROSTO20071 - 76	0.00	Eliminar
2	PROSTO20071 - 76			16 sfsfa	120.00	Eliminar
3	PROSTO20071 - 76			16 aaaaa	150.00	Eliminar
4	PROSTO20071 - 76			16 sdadasad	1250.00	Eliminar
5	PROSTO20071 - 76			16 adfasdf	250.00	Eliminar
6	PROSTO20071 - 76			08 gasgsa	2000.00	Eliminar
7	PROSTO20071 - 76					Eliminar

Creación Nuevo Gasto

Año: 2016
 Mes: 6
 Descripción: Gastos de oficina
 Monto: 2000
 Proyecto: 76- PROGRA

76- PROGRAMA SALUD MATERNO INFANTIL STO. DOMINGO DE LOS TSACHILAS
 18- REFUGIO DE LOS SUEÑOS QUITO
 18- CECAPI - CENTRO INFANTIL TDH QUITO
 76- Proyecto Guardería Quito

Ilustración 73 ANEXO 3: INGRESO DE GASTOS

7. INFORME DE EJECUCIÓN DEL PRESUPUESTO

MODULO: INFORME DE EJECUCIÓN DEL PRESUPUESTO

Ingrese el año: 2020 CARGAR

Ejecución del Presupuesto

Código Cuenta

Dólares

CÓDIGO	PRESUPUESTO ASIGNADO	MONTO EJECUTADO	% EJECUTADO
No records found.			

Print

Ilustración 74 ANEXO 3: INFORME DE EJECUCIÓN DEL PRESUPUESTO

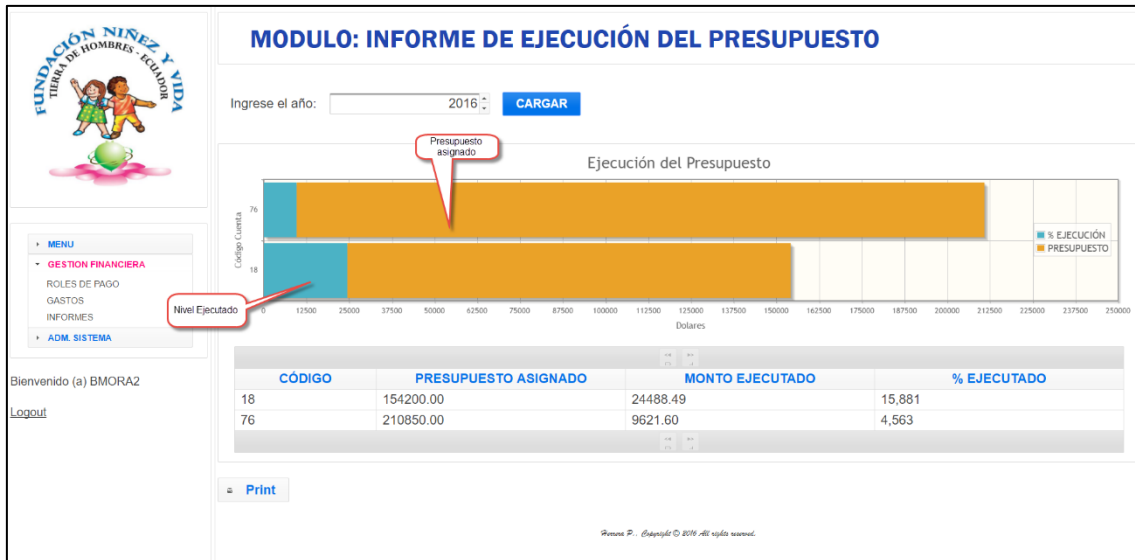


Ilustración 75 ANEXO 3: GRÁFICA DE EJECUCIÓN DE PRESUPUESTO

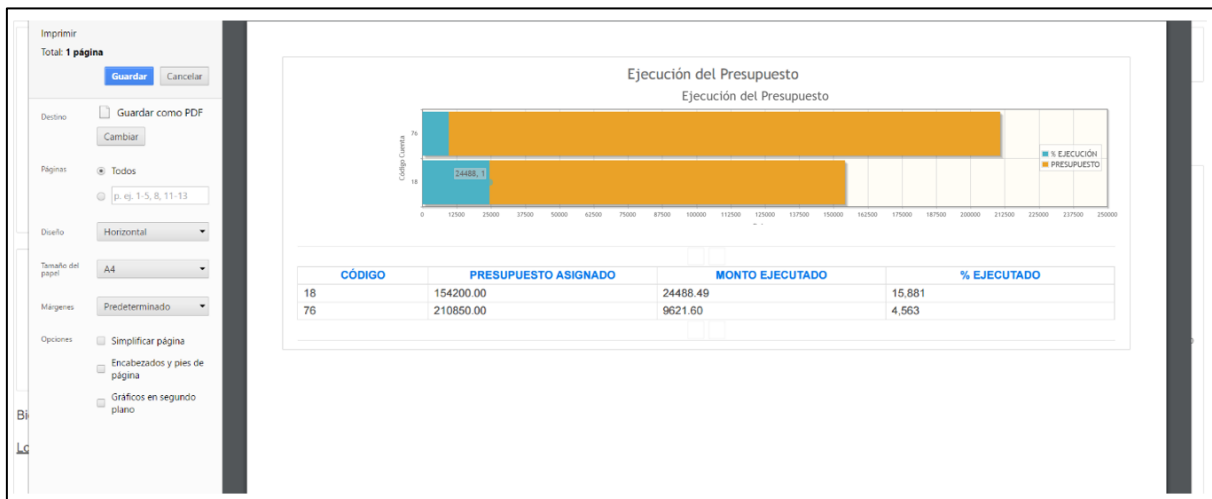


Ilustración 76 ANEXO 3: FORMATO DE IMPRESIÓN DEL INFORME DE EJECUCIÓN DEL PRESUPUESTO

8. USUARIO



Ilustración 77 ANEXO 3: MÓDULO DE USUARIOS

8.1 INGRESO

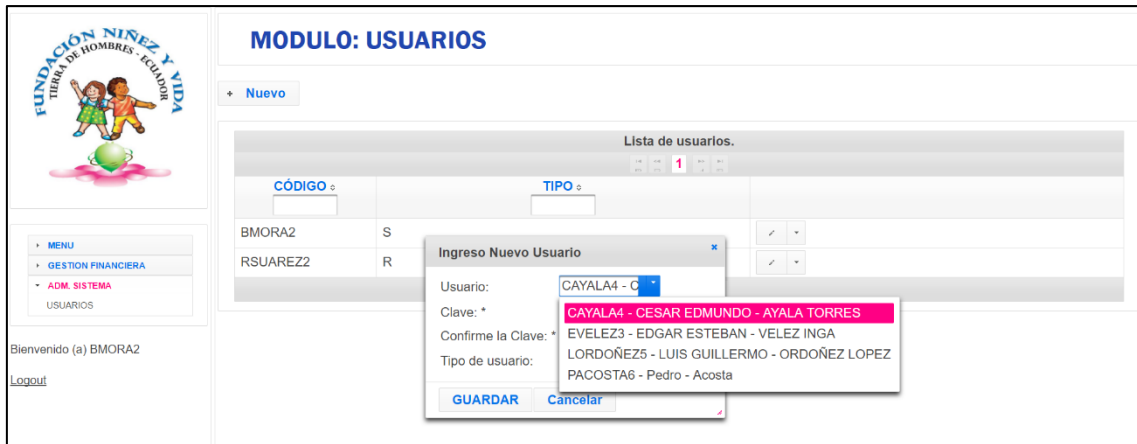


Ilustración 78 ANEXO 3: INGRESO DE NUEVO USUARIO



Ilustración 79 ANEXO 3: VALIDACIÓN DEL INGRESO DE UN NUEVO USUARIO

8.3 MODIFICACIÓN DE USUARIO



Ilustración 80 ANEXO 3: EDICIÓN DE USUARIOS

9. INGRESO Y SALIDA DEL APLICATIVO

9.2 INGRESAR AL SISTEMA



Ilustración 81 ANEXO 3: PÁGINA PRINCIPAL



Ilustración 82 ANEXO 3: MENSAJE DE VALIDACIÓN DE CLAVE Y USUARIO



Ilustración 83 ANEXO 3: INGRESO EXITOSO AL SISTEMA

9.1 SALIR DEL SISTEMA

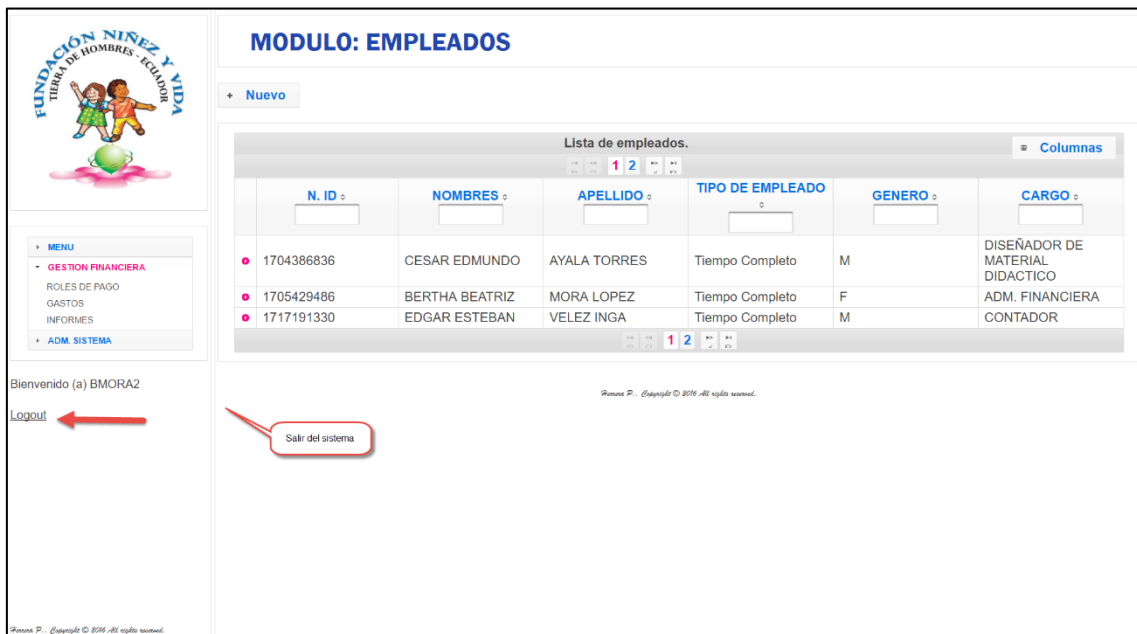


Ilustración 84 ANEXO 3: SALIR DEL SISTEMA



Ilustración 85 ANEXO 3: SALIDA EXITOSA DEL SISTEMA

ANEXO 4: MANUAL DE INSTALACIÓN

CONTENIDO

1. ARCHIVOS REQUERIDOS	118
2. INSTALAR JDK	118
2.1 VARIABLES DE ENTORNO JAVA-HOME	121
3. INSTALAR MOTOR DE BASE DE DATOS POSTGRESQL 9.5	122
3.1 CARGAR LA BASE DE DATOS RESPALDO XXXXXX.BACKUP	125
4. INSTALAR EL APLICATIVO	126

Autor: Pablo Herrera

Quito, 04 de julio del 2016

OBJETIVO

El presente manual tiene la finalidad de ubicar al instalador en los pasos seguidos para que el aplicativo puede funcionar.

1. ARCHIVOS REQUERIDOS

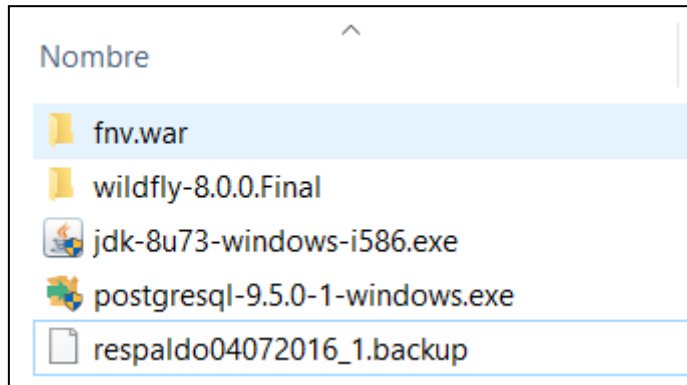


Ilustración 86 ANEXO 4: ARCHIVOS REQUERIDOS PARA LA INSTALACIÓN

2. INSTALAR JDK

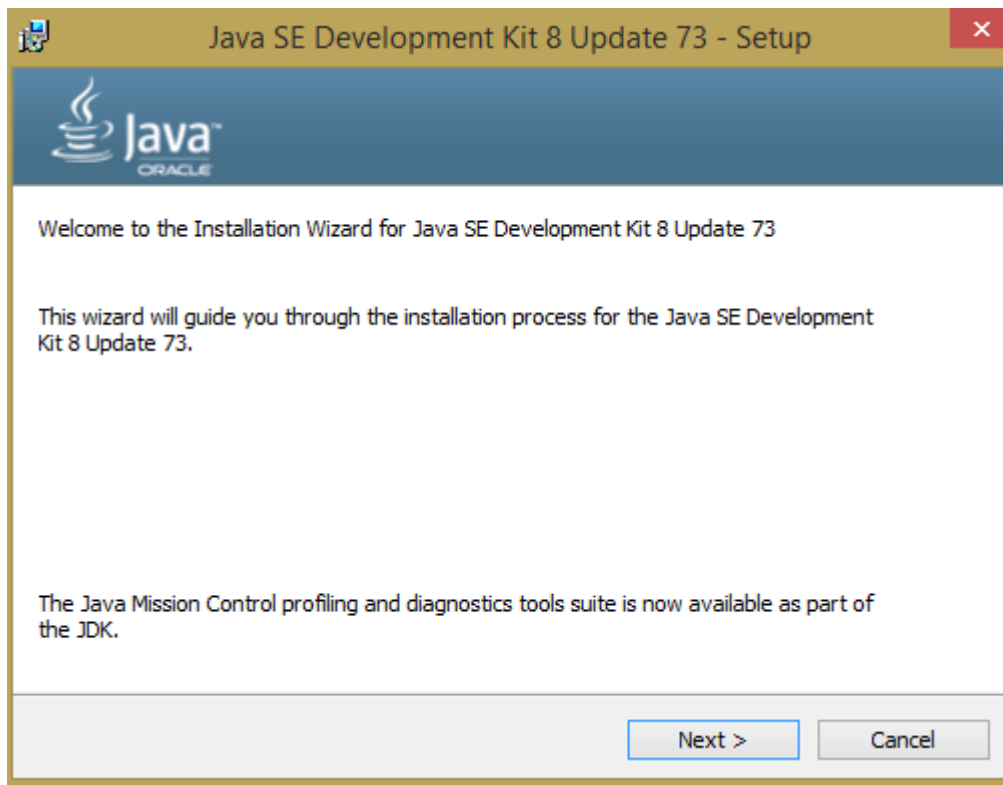


Ilustración 87 ANEXO 4: PANTALLA DE INSTALACIÓN JDK 1

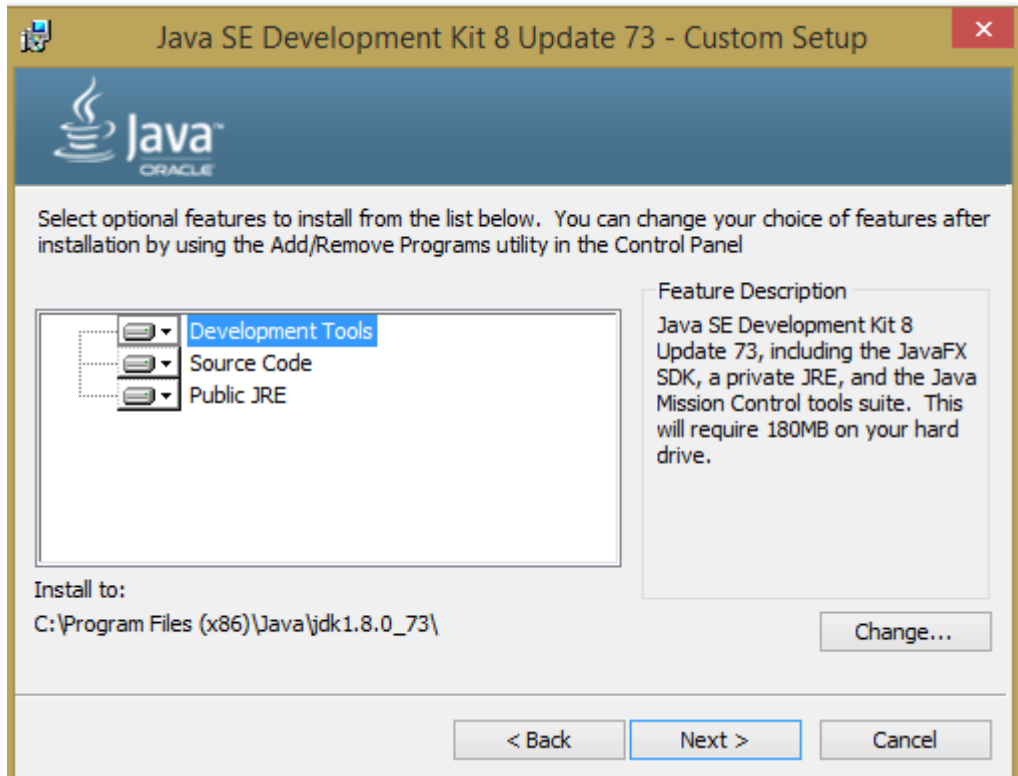


Ilustración 88 ANEXO 4: PANTALLA DE INSTALACIÓN JDK 2



Ilustración 89 ANEXO 4: PANTALLA DE INSTALACIÓN JDK 3

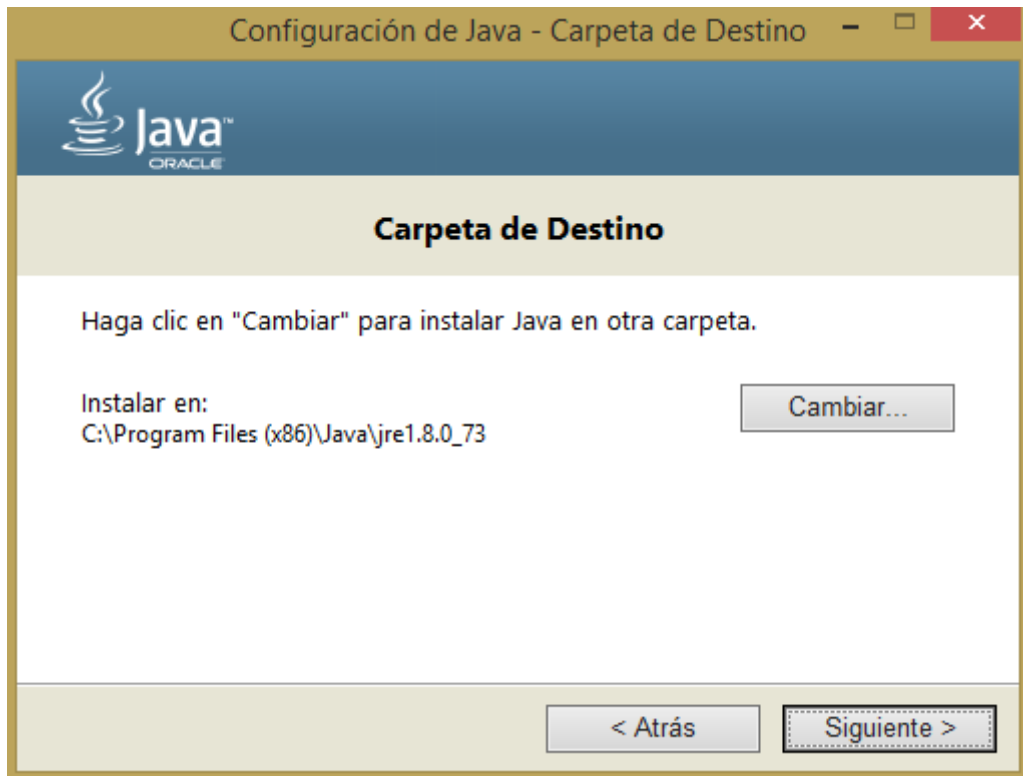


Ilustración 90 ANEXO 4: SELECCIÓN CARPETA DESTINO JDK



Ilustración 91 ANEXO 4: FIN DE INSTALACIÓN JDK

2.1 VARIABLES DE ENTORNO JAVA-HOME

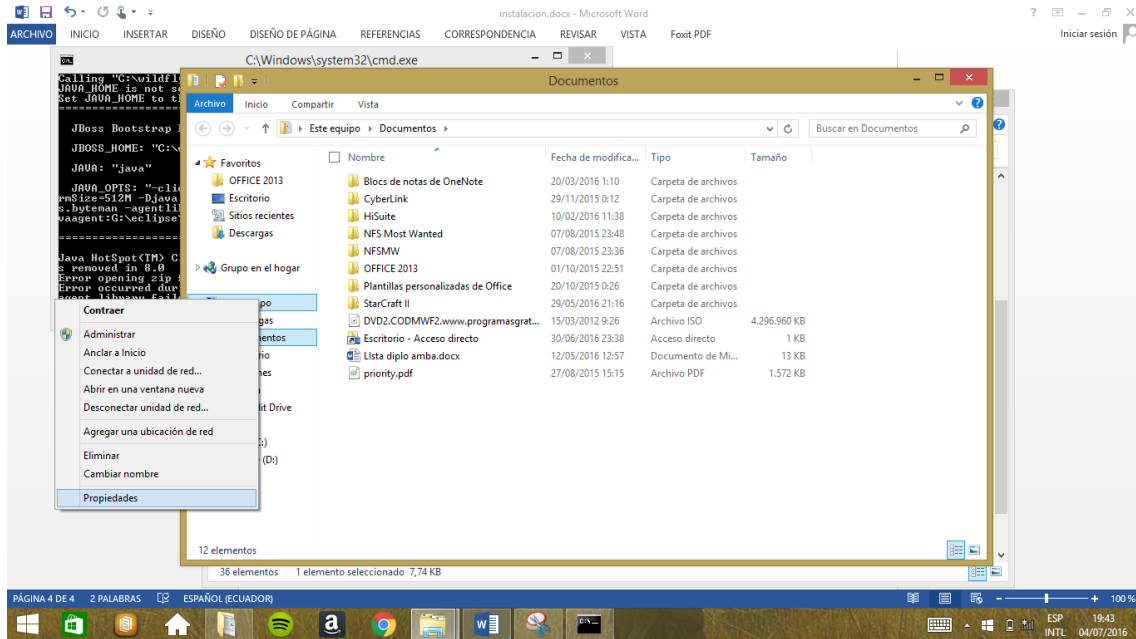
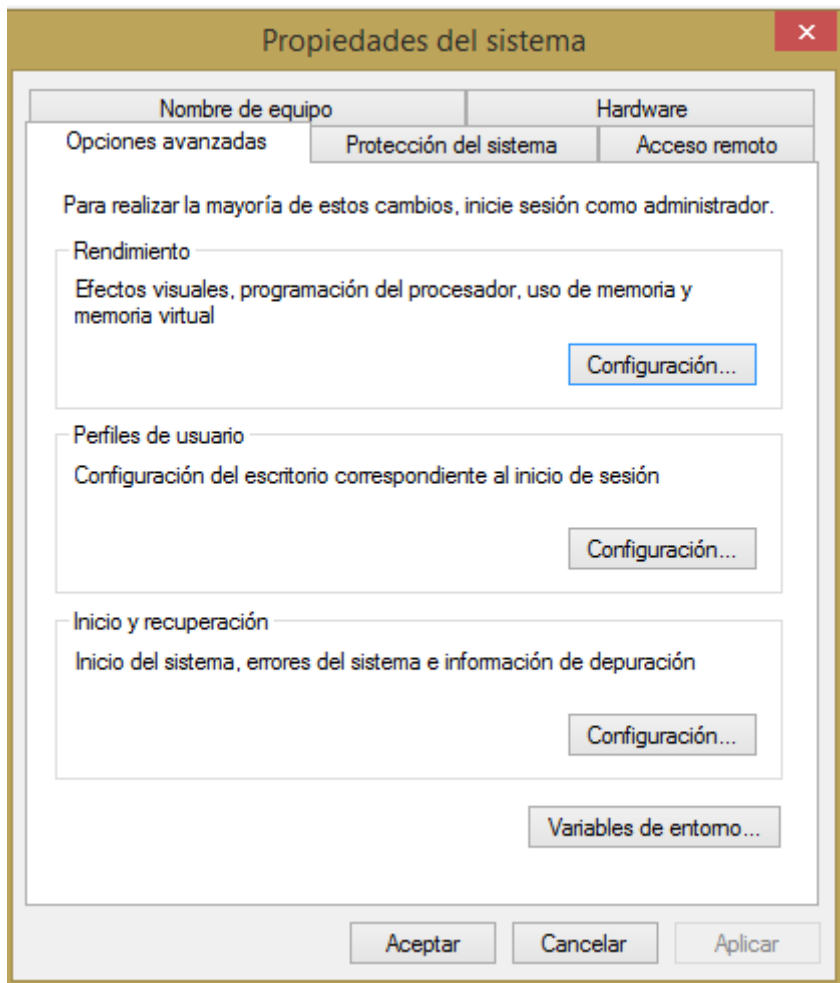


Ilustración 92 ANEXO 4: SETEO DE LAS VARIABLES DE ENTORNO



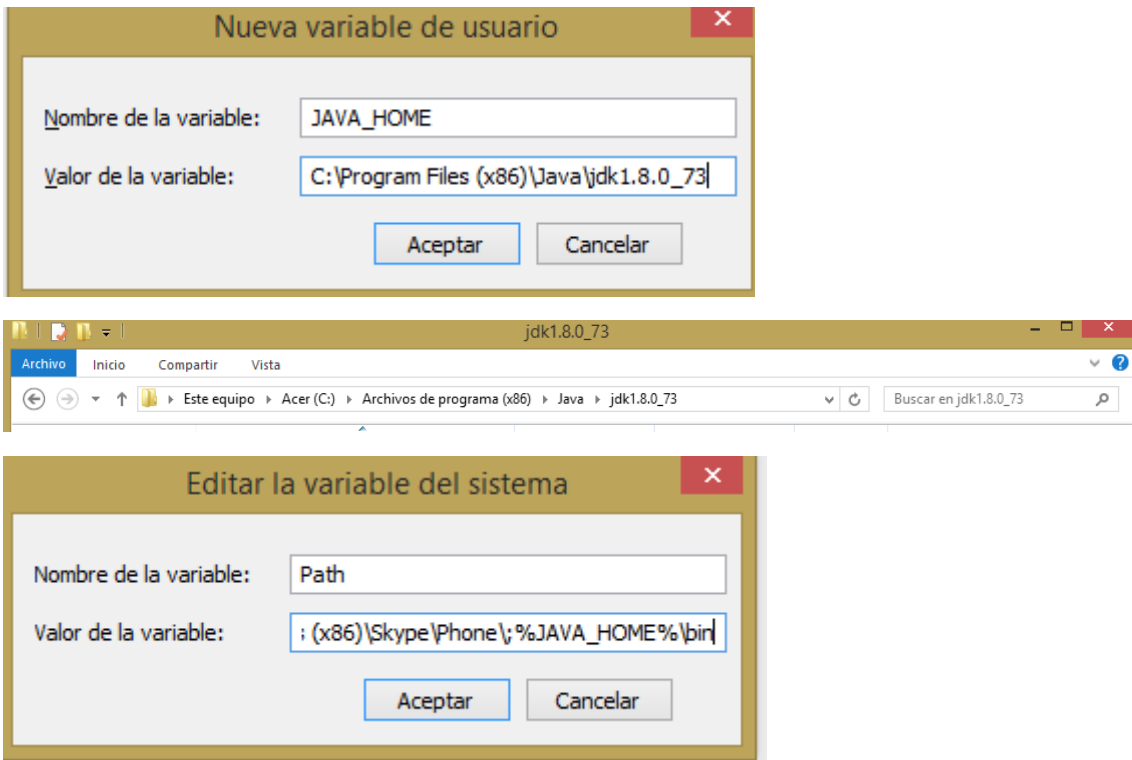


Ilustración 93 ANEXO 4: INGRESO DEL REGISTRO

3. INSTALAR MOTOR DE BASE DE DATOS POSTGRESQL 9.5



Ilustración 94 ANEXO 4: INSTALACIÓN POSTGRESQL 9.5

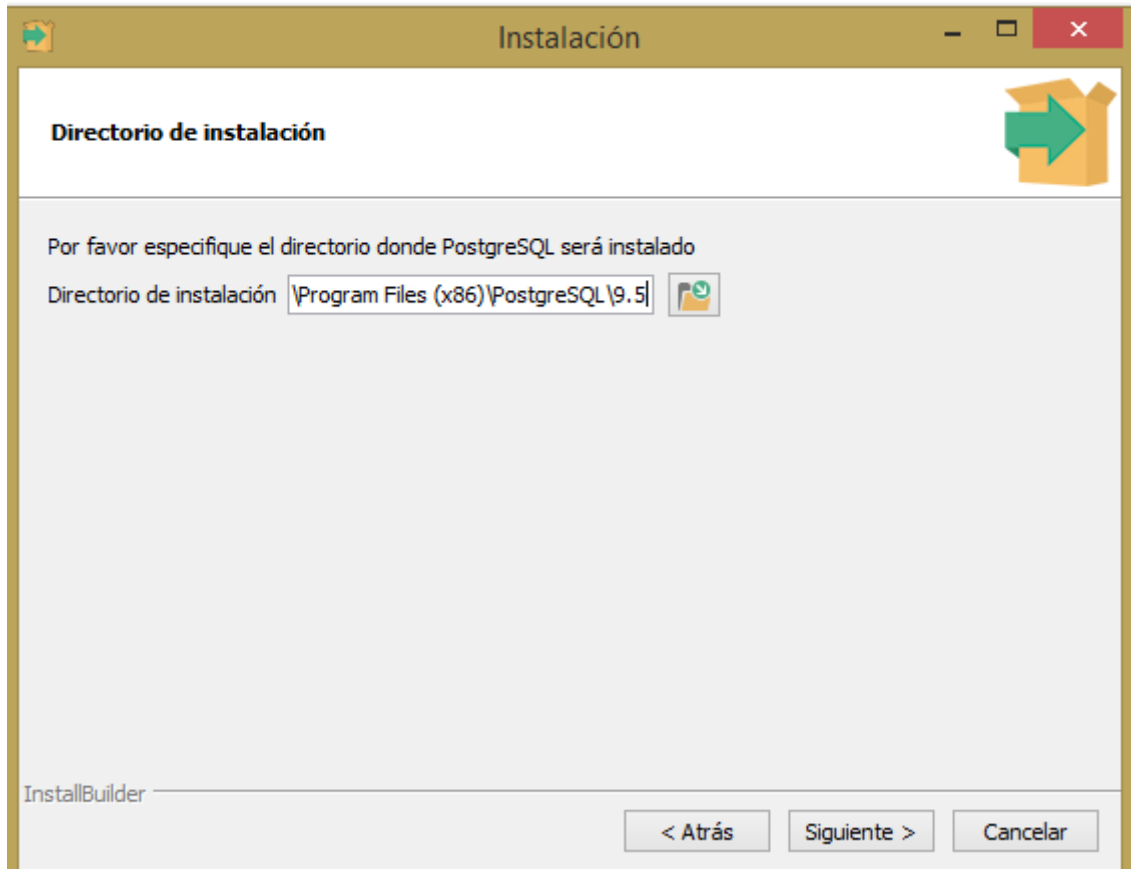


Ilustración 95 ANEXO 4: SELECCIÓN DEL DIRECTORIO POSTGRESQL 9.5

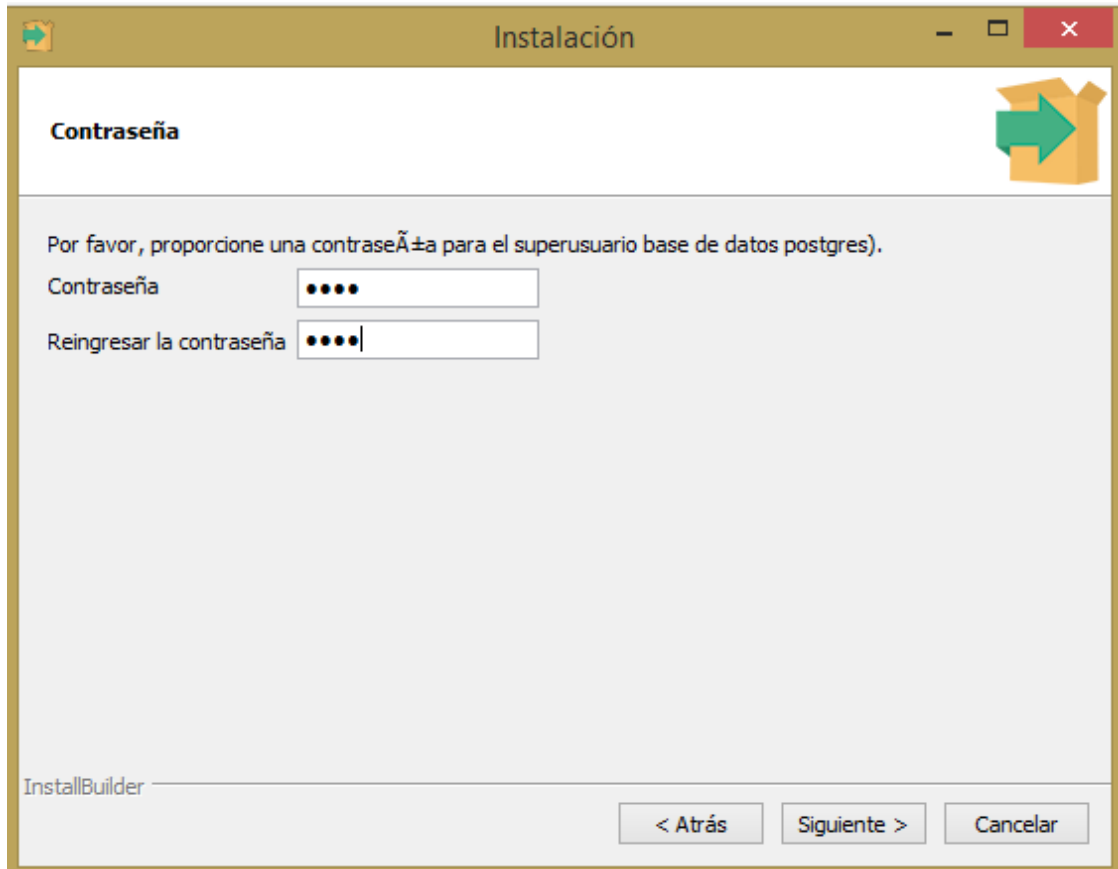


Ilustración 96 ANEXO 4: INGRESO DE CONTRASEÑA POSTGRESQL 9.5



Ilustración 97 ANEXO 4: FIN DE INSTALACIÓN POSTGRESQL 9.5

3.1 CARGAR LA BASE DE DATOS RESPALDO XXXXXX.BACKUP

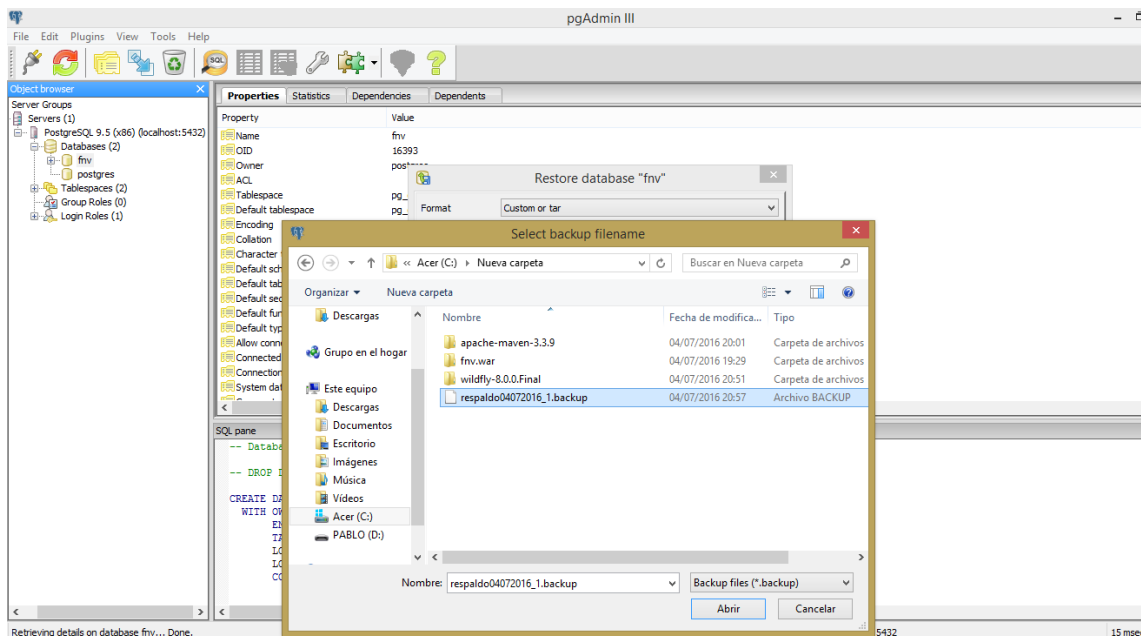


Ilustración 98 ANEXO 4: CARGA DE LA BASE DE DATOS EN EL MOTOR POSTGRESQL 9.5

5. INSTALAR EL APLICATIVO

Pegar el archivo FNV.war en la ruta wildfly-8.0.0.Final\standalone\deployments

BIBLIOGRAFÍA

- Almeida Moyano, W. R., & Imbacuán López, D. F. (Mayo de 2009). Benchmark para el uso de tecnologías relacionadas al servidor de aplicaciones de Glassfish y Jboss. *1, 1*, 214. (M. F. Otañez, Ed.) Quito, Ecuador: Escuela Politécnica Nacional. Recuperado el 28 de Enero de 2016, de <http://bibdigital.epn.edu.ec/bitstream/15000/1513/1/CD-2217.pdf>
- Beck, K., & Andres, C. (2012). *Extreme Programming Explained 2nd ed.* Boston: Addison-Wesley.
- Campoverde Luna, B. C. (2004). *Crecer con Respeto.* Quito: Editextos.
- Cendejas Veldéz, J. L. (8 de 5 de 2014). *IMPLEMENTACIÓN DEL MODELO INTEGRAL COLABORATIVO (MDSIC) COMO FUENTE DE INNOVACIÓN PARA EL DESARROLLO ÁGIL DE SOFTWARE EN LAS EMPRESAS DE LA ZONA CENTRO - OCCIDENTE EN MÉXICO.* (D. C. Lebrún, Ed.) Puebla, Mexico: Universidad Popular Autónoma del Estado de Puebla. Recuperado el 22 de 05 de 2016, de <http://www.eumed.net/tesis-doctorales/2014/jlcv/ficha.htm>
- FUNDACION NIÑEZ Y VIDA, T. (14 de ENERO de 2016). FOLLETO INFORMATIVO. *FOLLETO INFORMATIVO.*
- Gallardo, D. (26 de 11 de 2012). *Iniciandose en la plataforma eclipse.* Obtenido de IBM website: <http://www.ibm.com/developerworks/ssa/library/os-ecov/>
- Groussard, T. (Marzo de 2012). *JAVA 7 Los fundamentos del lenguaje Java.* Barcelona, España: ENI.
- Instituto Nacional de Tecnologías de las Comunicaciones. (03 de 2009). *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA.* España. Recuperado el 22 de 05 de 2016, de file:///D:/Mis%20documentos/Descargas/guia_de_ingenieria_del_software.pdf
- Martinez Ladrón de Guevara, J. (s.f.). *Fundamentos de programacion en Java. Fundamentos de programacion en Java, 1, 1.* Madris, España: EME. Recuperado el 21 de Febrero de 2016, de <http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/fjava.pdf>
- Martinez, R. (02 de 10 de 2010). *Sobre PostgreSQL.* Obtenido de PostgreSQL en español: http://www.postgresql.org.es/sobre_postgresql
- Morales, R. (1 de Septiembre de 2014). *Colombia Digital.* Recuperado el 21 de Febrero de 2016, de *Lenguajes de programación: ¿qué son y para qué sirven?*: <https://colombiadigital.net/actualidad/articulos-informativos/item/7669-lenguajes-de-programacion-que-son-y-para-que-sirven.html>

- Navarrete Jara, J. A. (2013). Desarrollo de un sistema informatico que permita gestionar la entrega aleatoria de premios de campañas publicitarias a través de una aplicación en la red social Facebook para la empresa "Esquema Publicidad" . 1, 136. (I. F. Cruz, Ed.) Quito, Ecuador: Pontificia Universidad Católica del Ecuador. Recuperado el Enero de 2016
- Nevado Cabello, V. (2010). *Introduccion a las Bases de Datos Relacionales*. Madrid: Vision Libros.
- Object Management Group . (24 de Junio de 2015). *Introduction To OMG's Unified Modeling Language® (UML®)*. Obtenido de OMG PAGINA WEB: http://www.omg.org/gettingstarted/what_is_uml.htm
- PostgreSQL Global Development Group. (s.f.). *About: PostgreSQL*. Obtenido de Sitio web PostgreSQL en Inglés: <http://www.postgresql.org/about/>
- Red Hat. Inc. (s.f.). *Red Hat Jboss Web Server*. Obtenido de Red Hat web site.
- SG Software Guru. (Junio de 2006). RedHat Adquiere JBoss . *Software Guru*(03), 12. Recuperado el 2016 de Enero de 2016, de http://desquer.ens.uabc.mx/afi/articulos/SG06_3_may_jun.pdf
- Umbrello UML Modeller Autores. (01 de 06 de 2013). *Introducción a UML*. Obtenido de Manual Umbrello UML Modeller Autores: <https://docs.kde.org/stable4/es/kdesdk/umbrello/index.html>
- Wells, D. (2009). *Extreme Programming: A gentle introduction*. Obtenido de Extreme Programming org: <http://www.extremeprogramming.org/>