

UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“SISTEMA BASADO EN LA PLATAFORMA JEE SOBRE UN MOTOR DE BASE
DE DATOS ORIENTADO A OBJETOS DB4O, PARA LA GESTIÓN DE
PROCESOS EN LA ADMINISTRACIÓN DE RECURSOS HUMANOS DE LA
MUNICIPALIDAD PROVINCIAL CHUCUITO JULI”**

TESIS

PRESENTADO POR:

BENIGNO CHAMBILLA ESCOBAR

WILLIAM REMIGIO MONTÚFAR CHATA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

PUNO - PERÚ

2014

UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

“SISTEMA BASADO EN LA PLATAFORMA JEE SOBRE UN MOTOR DE BASE
DE DATOS ORIENTADO A OBJETOS DB40, PARA LA GESTIÓN DE
PROCESOS EN LA ADMINISTRACIÓN DE RECURSOS HUMANOS DE LA
MUNICIPALIDAD PROVINCIAL CHUCUITO JULI”

TESIS PRESENTADA POR:

BENIGNO CHAMBILLA ESCOBAR

WILLIAM REMIGIO MONTÚFAR CHATA

PARA OPTAR EL TÍTULO PROFESIONAL DE: INGENIERO DE SISTEMAS
APROBADA POR EL JURADO REVISOR CONFORMADO POR:

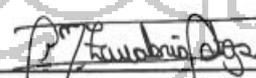
PRESIDENTE

:


 Mg. Ing. Elmer Coyla Idme

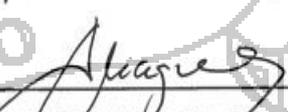
PRIMER MIEMBRO

:


 M.Sc. Ing. Milder Zanabria Ortega

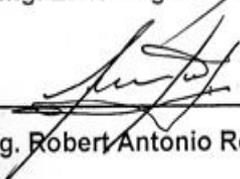
SEGUNDO MIEMBRO

:


 M.Sc. Ing. Elvis Augusto Aliaga Payehuanca

DIRECTOR

:


 Mg. Ing. Robert Antonio Romero Flores

PUNO - PERÚ

2014

ÁREA: Informática

TEMA: Sistemas de información tradicionales y expertos

DEDICATORIA

Dedico el presente trabajo a mis padres Albino Chambilla y Guillermina Escobar, por su paciencia y comprensión, por su bondad, sacrificio y ese apoyo incondicional por darme la mano cuando sentía que el camino se terminaba a ustedes por siempre mi corazón Papá y Mamá ya que sus consejos me dieron ese impulso para seguir adelante.

Benigno.

A mis queridos padres Clorinda y Juan Antonio.

William.



AGRADECIMIENTOS

Primeramente agradecer infinitamente a Dios por haberme dado la fortaleza, el coraje, la sabiduría y sobre todo la salud y no dejarme solo en los momentos más difíciles, por darme ese ánimo espiritual con ello guiarme y mostrarme el camino correcto.

A mis padres, por ese apoyo incondicional por ese sacrificio tremendo y a mis hermanos en especial a mi hermano Fredy por ser como es, por darme ese apoyo cuando había sufrido un accidente mil gracias por estar ahí a mi lado para ellos mi agradecimiento que siempre están en las buenas y en las malas, y a todas las personas que han formado parte de mi vida profesional a las que me encantaría agradecerles su amistad, en especial a mi compañero William por el gran aporte en este trabajo de investigación.

Benigno.

A Dios, por haberme puesto en el camino de la vida con sabiduría, fortaleza, salud, coraje y no dejarme solo en los momentos difíciles.

William.

Y de manera especial al Presidente y a los miembros del jurado, Mg. Ing. Elmer Coyla Idme, M.Sc. Ing. Milder Zanabria Ortega, M.Sc. Ing. Elvis Augusto Aliaga Payahuanca, respectivamente, agradecemos su paciencia y capacidad de guiarnos durante todo el proceso de control del trabajo de investigación.

INDICE

RESUMEN	12
ABSTRACT	13
INTRODUCCIÓN	14

CAPÍTULO I

PROBLEMÁTICA DE LA INVESTIGACIÓN	17
1.1. PLANTEAMIENTO DEL PROBLEMA	17
1.2. FORMULACIÓN DEL PROBLEMA	19
1.2.1. Problema General	19
1.2.2. Problemas Específicos	19
1.3. JUSTIFICACIÓN	20
1.4. OBJETIVOS	21
1.4.1. Objetivo General	21
1.4.2. Objetivos Específicos	21
1.5. HIPÓTESIS DE LA INVESTIGACIÓN	22
1.5.1. Hipótesis General	22
1.5.2. Hipótesis Específica	22
1.6. ALCANCES	23
1.7. LIMITACIONES	23

CAPÍTULO II

MARCO TEÓRICO	26
2.1. ANTECEDENTES	26
2.1.1. Antecedentes Nacionales	26
2.1.2. Antecedentes Internacionales	27
2.2. MARCO CONCEPTUAL	29
2.2.1. Sistema de Información	29
2.2.1.1. Componentes de un Sistema de Información	31
2.2.1.2. Sistemas de Información Web	32
2.2.2. Base de Datos Orientado a Objetos	33
2.2.3. DB4O	46
2.2.4. Plataforma JEE	52
2.2.5. Java Server Faces	55
2.2.6. Spring	57
2.2.7. JasperReport	61
2.2.8. Servidor Apache	63
2.2.9. Proceso Unificado de Desarrollo	64
2.2.10. UML	69
2.2.11. Reingeniería de Software	77
2.2.12. Norma ACID	79
2.2.13. Administración de Recursos Humanos	82
2.2.14. Municipalidad Provincial Chucuito Juli	86
2.2.14.1. Visión	86
2.2.14.2. Misión	87
2.2.14.3. Objetivos Generales	87
2.2.14.4. Objetivos Específicos	88

2.2.14.5.	Organigrama.....	89
2.2.14.6.	Unidad de Personal.....	90
2.3.	DEFINICIÓN DE TÉRMINOS BÁSICOS.....	93
2.3.1.	Sistema.....	93
2.3.2.	Administración de Recursos Humanos.....	93
2.3.3.	Optimización.....	93
2.3.4.	Gestión.....	94
2.3.5.	Procesos.....	94
2.3.6.	Base de Datos Orientado a Objetos.....	95
2.3.7.	DB4O.....	95
2.3.8.	Java.....	95
2.3.9.	JavaBeans.....	96
2.3.10.	JSP.....	96
2.3.11.	JSF.....	97
2.3.12.	Framework.....	97
2.3.13.	Hibernate.....	98
2.3.14.	IoC.....	98
2.3.15.	Servlet.....	98
2.3.16.	JAR.....	99
2.3.17.	Planilla de Remuneraciones.....	99
2.3.18.	Planilla Electrónica.....	100
2.3.19.	AFP.....	100
2.3.20.	T-registro.....	100
2.3.21.	PLAME.....	101
2.3.22.	PDT.....	101

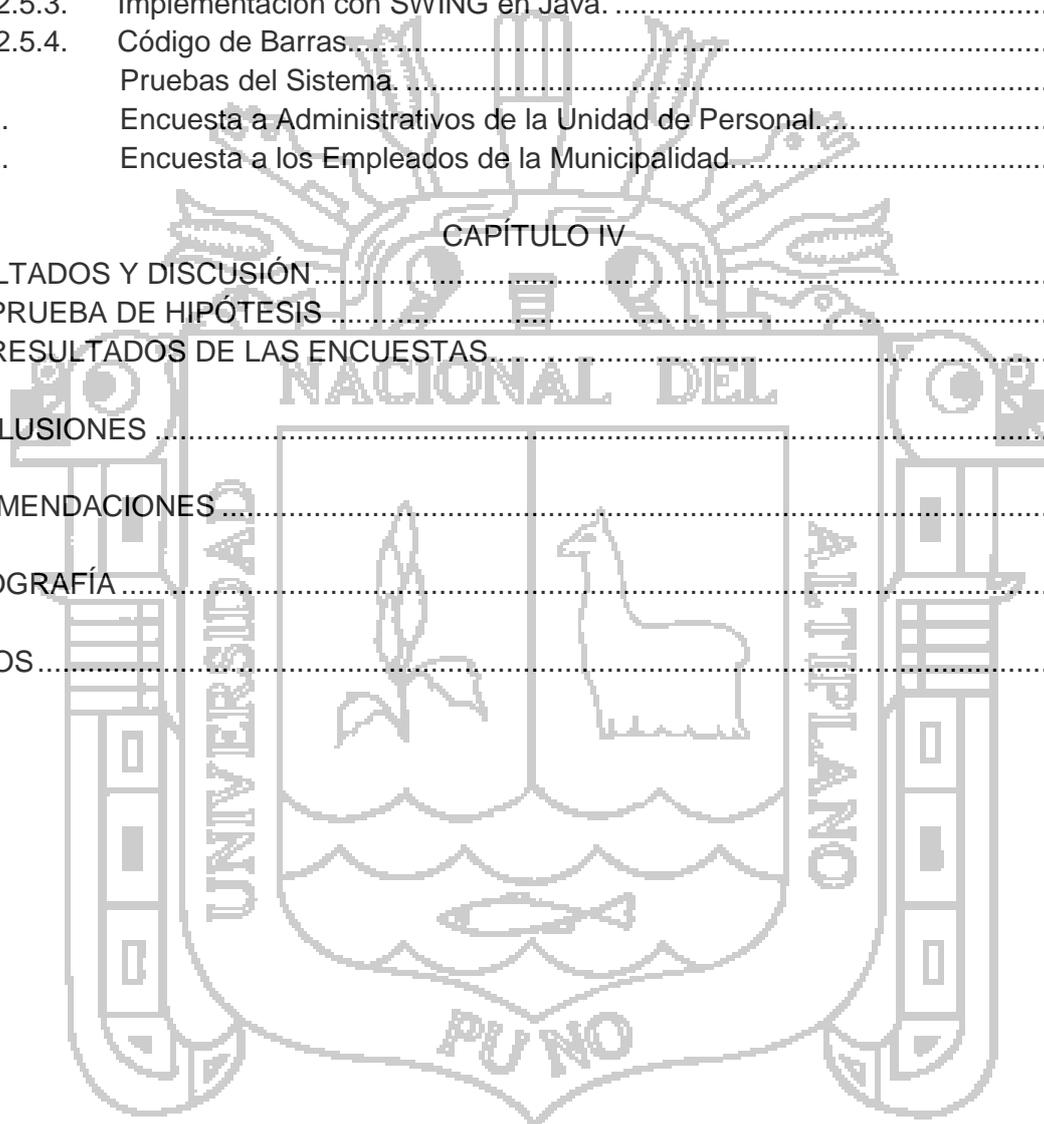
CAPÍTULO III

MATERIALES Y MÉTODOS.....	103
3.1. TRABAJO EXPERIMENTAL.....	103
3.1.1. Metodología de la Investigación.....	103
3.1.1.1. Criterios para la Selección de la Metodología.....	104
3.1.2. Diseño de la Investigación.....	105
3.1.3. Población y Muestra.....	107
3.1.4. Sistemas de Variables.....	109
3.1.5. Operacionalización de Variables.....	110
3.1.6. Métodos de Recopilación de Datos.....	111
3.1.7. Métodos de Tratamientos de Datos.....	111
3.2. MATERIAL EXPERIMENTAL.....	111
3.2.1. Metodología de Desarrollo del Sistema.....	111
3.2.2. Análisis de Requisitos.....	113
3.2.2.1. Requisitos Funcionales.....	114
3.2.2.2. Requisitos no Funcionales.....	116
3.2.3. Modelado de Análisis.....	117
3.2.3.1. Identificación de los Actores.....	117
3.2.3.2. Diagrama de Casos de Uso.....	118
3.2.3.3. Especificaciones de Casos de Uso.....	120
3.2.3.4. Modelado de Dominio.....	126

3.2.4.	Diseño del Sistema.	127
3.2.4.1.	Diseño de la Arquitectura del Sistema.	127
3.2.4.2.	Diagramas de Secuencias.	129
3.2.4.3.	Diagramas de Colaboraciones.	135
3.2.4.4.	Diagrama de Clases.	138
3.2.4.5.	Diseño de Interfaz.	139
3.2.5.	Implementación del Sistema.	143
3.2.5.1.	Control de Concurrencia en DB4O.	143
3.2.5.2.	Implementación con AJAX.	145
3.2.5.3.	Implementación con SWING en Java.	147
3.2.5.4.	Código de Barras.	150
3.2.6.	Pruebas del Sistema.	153
3.2.6.1.	Encuesta a Administrativos de la Unidad de Personal.	153
3.2.6.2.	Encuesta a los Empleados de la Municipalidad.	155

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN.	1588
4.1. PRUEBA DE HIPÓTESIS.	158
4.2. RESULTADOS DE LAS ENCUESTAS.	160
CONCLUSIONES.	172
RECOMENDACIONES.	173
BIBLIOGRAFÍA.	174
ANEXOS.	178



INDICE DE FIGURAS

Figura N° 01: Funciones de un sistema de información.....	30
Figura N° 02: Componentes de un sistema de información.....	32
Figura N° 03: Características de un sistema gestor de base de datos orientado a objetos.....	36
Figura N° 04: Base de datos orientado a objetos comparado con base de datos relacionales .	48
Figura N° 05: Esquema general del java EE	53
Figura N° 06: Componentes de la arquitectura java EE.....	55
Figura N° 07: Arquitectura spring	60
Figura N° 08: Proceso de desarrollo de software	64
Figura N° 09: Metodología RUP	65
Figura N° 10: Fases de la metodología RUP.....	67
Figura N° 11: Recursos utilizados en las fases del RUP	69
Figura N° 12: Bloques de construcción - clases	70
Figura N° 13: Bloques de construcción - colaboración	71
Figura N° 14: Bloques de construcción - casos de Uso	72
Figura N° 15: Bloques de construcción - clases activas	72
Figura N° 16: Bloques de construcción - componentes	73
Figura N° 17: Elementos de comportamiento - mensajes.....	73
Figura N° 18: Elementos de comportamiento - estados.....	73
Figura N° 19: Elemento de agrupación - paquetes	73
Figura N° 20: Elemento de anotación - notas	74
Figura N° 21: Elemento de anotación - dependencia.....	74
Figura N° 22: Representación de reingeniería de software	79
Figura N° 23: Objetivos de la administración de recursos humanos.....	85
Figura N° 24: Organigrama de la Municipalidad Provincial de Chucuito Juli.....	98
Figura N° 25: Número de trabajadores en planta	106
Figura N° 26: Número de trabajadores en obra	108
Figura N° 27: Número de trabajadores en la Municipalidad.....	108
Figura N° 28: Modelo de desarrollo de software.....	112
Figura N° 29: Diagrama de casos de uso.....	118
Figura N° 30: Modelado de dominio - diagrama conceptual	126
Figura N° 31: Modelado de dominio - clases conceptuales	127
Figura N° 32: Diseño de la arquitectura del sistema	128
Figura N° 33: Diagrama de secuencia del caso de uso configuración	129
Figura N° 34: Diagrama de secuencia del caso de uso ficha del empleado.....	130
Figura N° 35: Diagrama de secuencia del caso de uso registro del empleado	131

Figura N° 36: Diagrama de secuencia del caso de uso planilla	132
Figura N° 37: Diagrama de secuencia del caso de uso asistencia	133
Figura N° 38: Diagrama de secuencia del caso de uso declaraciones	134
Figura N° 39: Diagrama de colaboración del caso de uso configuración	135
Figura N° 40: Diagrama de colaboración del caso de uso ficha del empleado	135
Figura N° 41: Diagrama de colaboración del caso de uso registro del empleado	136
Figura N° 42: Diagrama de colaboración del caso de uso planilla	136
Figura N° 43: Diagrama de colaboración del caso de uso asistencia	137
Figura N° 44: Diagrama de colaboración del caso de uso declaraciones	137
Figura N° 45: Diagrama de clases - modelo base de datos orientado a objetos	138
Figura N° 46: Diseño de interfaz - esquema de diseño.....	140
Figura N° 47: Diseño de interfaz - registro de asistencia	140
Figura N° 48: Diseño de interfaz - interfaz acceso a usuario	141
Figura N° 49: Diseño de interfaz - crear usuario.....	141
Figura N° 50: Diseño de interfaz - interfaz planilla de remuneraciones.....	142
Figura N° 51: Diseño de interfaz - control de asistencia	142
Figura N° 52: Código con semáforos DB4O	144
Figura N° 53: Función que ocupa el sistema para recargar la Web.....	146
Figura N° 54: Función objetoAjax().....	146
Figura N° 55: Método reescrito en el modelo de jTable.....	148
Figura N° 56: Grafica de prueba de hipótesis.....	159
Figura N° 57: Resultado de encuesta - rendimiento del sistema	160
Figura N° 58: Resultado de encuesta - opinión con respecto a la interfaz del programa.	161
Figura N° 59: Resultado de encuesta -opinión con respecto a la usabilidad del sistema.....	162
Figura N° 60: Resultado de encuesta - opinión con respecto a la confiabilidad del sistema.....	163
Figura N° 61: Resultado de encuesta - opinión con respecto a la eficiencia del sistema	164
Figura N° 62: Resultado de encuesta - opinión con respecto al tiempo de respuesta	165
Figura N° 63: Resultado de encuesta - opinión con respecto al costo.....	166
Figura N° 64: Resultado de encuesta - opinión con respecto a la gestión	167
Figura N° 65: Resultado de encuesta - opinión con respecto al sistema	168
Figura N° 66: Resultado de encuesta - opinión con respecto al tiempo sin sistema	169
Figura N° 67: Resultado de encuesta - opinión con respecto al tiempo con el sistema.	170
Figura N° 68: Resultado de encuesta - opinión con respecto a la dificultad del sistema.....	171

INDICE DE TABLAS

Tabla N° 01: Esfuerzo - horario contra fases del RUP.....	68
Tabla N° 02: Comparación de metodologías de desarrollo de software	104
Tabla N° 03: Operacionalización de variables	110
Tabla N° 04: Requisito funcional RF001.....	114
Tabla N° 05: Requisito funcional RF002.....	114
Tabla N° 06: Requisito funcional RF003.....	115
Tabla N° 07: Requisito funcional RF004.....	115
Tabla N° 08: Requisito funcional RF005.....	115
Tabla N° 09: Requisito funcional RF006.....	115
Tabla N° 10: Identificación de actores - usuario	117
Tabla N° 11: Identificación de actores - empleado	117
Tabla N° 12: Descripción de caso de uso - configuración.....	119
Tabla N° 13: Descripción de caso de uso - registro de empleado	119
Tabla N° 14: Descripción de caso de uso - planilla.....	119
Tabla N° 15: Descripción de caso de uso - declaraciones.....	119
Tabla N° 16: Descripción de caso de uso - asistencia.....	119
Tabla N° 17: Especificaciones de caso de uso - configuración.....	120
Tabla N° 18: Especificaciones de caso de uso - ficha de empleado.....	121
Tabla N° 19: Especificaciones de caso de uso - registro de empleado.....	122
Tabla N° 20: Especificaciones de caso de uso - planilla.....	123
Tabla N° 21: Especificaciones de caso de uso - declaraciones a entidades.....	124
Tabla N° 22: Especificaciones de caso de uso - asistencia.....	125
Tabla N° 23: Resultado de encuesta - rendimiento del sistema.....	160
Tabla N° 24: Resultado de encuesta - opinión con respecto a la interfaz del programa	161
Tabla N° 25: Resultado de encuesta - opinión con respecto a la usabilidad del sistema	162
Tabla N° 26: Resultado de encuesta - opinión con respecto a la confiabilidad del sistema	163
Tabla N° 27: Resultado de encuesta - opinión con respecto a la eficiencia del sistema	164

Tabla N° 28: Resultado de encuesta - opinión con respecto al tiempo de respuesta..... 165

Tabla N° 29: Resultado de encuesta - opinión con respecto al costo 166

Tabla N° 30: Resultado de encuesta - opinión con respecto a la gestión 167

Tabla N° 31: Resultado de encuesta - opinión con respecto al sistema 168

Tabla N° 32: Resultado de encuesta - opinión con respecto al tiempo sin sistema 169

Tabla N° 33: Resultado de encuesta - opinión con respecto al tiempo con el sistema 170

Tabla N° 34: Resultado de encuesta - opinión con respecto a la dificultad del sistema 171



RESUMEN

El presente trabajo de investigación, tiene como objetivo general: Determinar qué el sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, optimiza la gestión de procesos en la administración de recursos humanos dentro de la Municipalidad Provincial de Chucuito Juli. Ya que al personal de trabajo de toda institución se le denomina como activos intangibles, y mucho depende del correcto funcionamiento de este capital humano y su desenvolvimiento en la entidad como tal. La municipalidad es una institución de gobierno local, que cumple con sus funciones como entidad del estado y se rige bajo las normativas de una institución pública. El sistema de administración de recursos humanos se tiene como ámbito de desarrollo en toda institución y más en una entidad de gobierno local. Todo se inicia por lo que la Unidad de Personal no cuenta y no maneja información adecuadamente automatizada. Para el desarrollo del sistema se utilizó la metodología RUP con todas sus especificaciones. El sistema cuenta con módulos estos son: identificación, planillas, control de asistencia, declaración a entidades y escalafón del personal; finalmente se concluye que el sistema mejora en un 85% la gestión de procesos en cuanto a la administración de recursos humanos en la institución, se tiene mayor eficiencia en cada proceso, menor tiempo de cada proceso, reduce los costos en la gestión de procesos y se tiene mejor control en la administración de cada proceso, dando resultados para el mes de Julio de 2014.

Palabras clave: Optimizar, gestión de procesos, administrar recursos, automatizar la información, patrones de diseño.

ABSTRACT

The present research has as general objective: Determine why the JEE platform based on a database engine object oriented DB4O system optimizes the management of processes in the management of human resources within the Provincial Municipality of Chucuito Juli. Since working staff of any institution is referred to as intangible assets, and much depends on the proper functioning of this human capital and its development in the state as such. The municipality is a local government institution that fulfills his duties as state entity and is governed under the regulations of a public institution. The management system has the human resources field development institution and a local government entity. Everything starts at the Personnel Unit does not have and does not handle properly automated information. To develop the system, the RUP methodology was used with all specifications. The system has modules which are: identification, payroll, time and attendance, reported entities and ranks of staff; finally concluded that the system improves by 85% process management in terms of human resource management in the institution, have greater efficiency in each process, the less time each process, reduces costs in managing processes and you have better control in the administration of each process, giving results for the month of July 2014.

Keywords: Optimization, process management, manage resources, an automated information design patterns.

INTRODUCCIÓN

Cuando se desarrolla un sistema, se tiene como objetivo mejorar una situación inicial, para luego hacer una evaluación y verificar cuanto se mejora a una situación final. Hoy en día el manejo adecuado de la información en una organización, es de un impacto estratégico y la oportunidad de tener una ventaja competitiva frente a otras organizaciones. Y teniendo en cuenta que el funcionamiento en el entorno ayuda a producir un cambio realmente significativo. El crecimiento exponencial del volumen de información que se produce en todos los sectores económicos y el consiguiente crecimiento en la complejidad de la gestión de dicho volumen de información, explica por qué los sistemas brindan mejores soluciones.

En la Municipalidad Provincial Chucuito Juli, se gestiona toda la información relacionada con los diferentes procesos de cada una de sus áreas, las tareas de gestión de información como llevar el control de la administración de recursos humanos, se realiza de manera aislada y poco coordinada, dificultándose la gestión de información, ocasionando pérdida de tiempo y dinero durante estos procesos, por dichas razones se decidió desarrollar un **Sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, para la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial Chucuito Juli.**

El presente trabajo de investigación contiene los siguientes aspectos:

En el primer capítulo se detalla el planteamiento del problema, justificación, los objetivos de la investigación y posteriormente se formula la hipótesis de la investigación que se quiere demostrar y se establece el escenario de investigación.

En el segundo capítulo se desarrolla el marco teórico, se constituyen los antecedentes considerados en el trabajo de investigación, el marco conceptual comprende todo lo relacionado con los términos utilizados en la investigación, describiéndose sintéticamente algunos de los principales conceptos.

En el tercer capítulo se detalla los métodos e instrumentos que se utilizó en la investigación; también se determina el tipo de investigación cuantitativa con diseño cuasi experimental; sistema de variables, material experimental, métodos de recopilación de datos, método de tratamiento de datos y metodología de desarrollo.

En el cuarto capítulo, denominado resultados y discusión, está constituido por la prueba de hipótesis.

Finalmente se tiene las conclusiones alcanzadas en la investigación, las recomendaciones respectivas y los anexos.





PROBLEMÁTICA DE LA INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

La Municipalidad Provincial Chucuito Juli, que cumple funciones de gobierno local, administración para promover la satisfacción de las necesidades básicas de los vecinos, su bienestar y desarrollo de la circunscripción, para lograr mejor estas funciones de desarrollo de la población se encuentra dentro de sus áreas de administración, la unidad de personal; es la encargada de regular los procesos de selección, progresión en la carrera y evaluación del personal que labora dentro de la institución. Así mismo regula el otorgamiento de las remuneraciones, bonificaciones, beneficios, compensaciones y pensiones del personal que tiene vínculo laboral con la entidad, hoy en día se toma al personal como activos intangibles ya que tienen su origen en los conocimientos, habilidades, valores y actitudes de las personas que forman parte del núcleo estable de la entidad.

A estos activos intangibles se les denomina Capital Humano y comprende todos aquellos conocimientos tácitos o explícitos que generan valor económico para la entidad, por cuanto se logró detectar que no cuenta con una óptima administración de recursos humanos. Lo cual ha causado fallas a nivel de la funcionabilidad del mismo.

La unidad de personal, administra sus procesos separando sus actividades en módulos, estos son los siguientes: modulo planillas (que elabora las planillas de remuneraciones del personal), modulo escalafón (que crea y mantiene actualizado los legajos personales), modulo asistencia (que controla al personal

en función a una directiva interna de asistencia y permanencia) y modulo evaluación (que elabora y ejecuta el plan de capacitación), Dentro de la unidad de personal se tiene diferentes problemas a continuación se detallan cada problemática en cada módulo, esto nos lleva a un estado situacional general pésimo:

- Dentro de módulo de planillas, la elaboración de estas planillas se hace a través de hojas de cálculo (Microsoft Excel), el cual no garantiza un buen trabajo por la seguridad y la demanda de tiempo para la elaboración esto como perdidas económicas, también ocasiona información aislada, deficiente uso de información, no es posible realizar consultas de datos históricos, todo esto ocasiona un mal manejo de datos.
- Dentro del módulo de escalafón, se realizan estos trabajos de forma manual y almacenando estos documentos de forma física el cual no brinda seguridad el material físico de la misma forma tampoco se puede realizar reportes, consultas por días, meses y años.
- Dentro del módulo de control de asistencia y permanencia, estos trabajos se realizan de manera manual, cuentan con un marcador biométrico sin funcionamiento y marcador de tarjetas esté en funcionamiento, el cual no garantiza la seguridad de control de asistencia, tampoco se puede realizar consultas de manera inmediata, se tiene información aislada todo esto conlleva el pésimo trabajo de control de asistencia.
- Dentro del módulo de evaluación, en la unidad de personal no se tiene implementado este módulo, por los diferentes problemas mencionados en los módulos anteriores, como toda información está aislada y sin consultas

de forma inmediata no se tiene este módulo con respecto al desempeño del personal.

Por tanto se observa que dentro de la unidad de personal no se tiene información automatizada, para la gestión de procesos y en lo cual no garantiza la correcta administración de los recursos humanos de la Municipalidad Provincial Chucuito Juli.

Toda esta información se obtuvo haciendo las entrevistas al personal que labora dentro de la unidad de personal de la Municipalidad.

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema General.

¿Cuáles son los efectos de la aplicación del sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, para la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial Chucuito Juli?

1.2.2. Problemas Específicos.

a) ¿En qué medida la determinación de los requerimientos iniciales de acuerdo a la lógica de cada proceso va mejorar la correcta administración de recursos humanos en la Municipalidad.

- b) ¿En qué medida el sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, automatizara la gestión de procesos en la administración de recursos humanos a la Municipalidad Provincial Chucuito Juli.
- c) ¿Cuánto se mejora con el sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, puesto la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial Chucuito Juli?

1.3. JUSTIFICACIÓN

El análisis y gestión de la información es considerada como uno de los procesos más valorados en cualquier organización. Nuestra sociedad ha creado tanta información que el volumen es de tal magnitud y la forma de acceder tan variada, que nos enfrentamos a nuevos problemas como el saber que la información se necesita, de qué forma obtenerla y como aprovechar la que se posee. La Municipalidad Provincial Chucuito Juli, no es ajena a este avance tecnológico.

Mejorar la gestión de procesos en la Municipalidad Provincial Chucuito Juli, sobre todo en la unidad de personal, porque aún no existe un sistema para la administración de recursos humanos, teniendo como beneficiarios al personal que labora en la Municipalidad y las distintas personas que realizan tramites dentro de la entidad, y así facilitaremos al personal que labora en esta unidad, porque el trabajo se viene haciendo manualmente, esto dificulta la rapidez de atención demanda más gasto de dinero y sobre todo el tiempo. Con el uso del

sistema de recursos humanos con uso tecnología de base de datos orientada a objetos nos permite mayor rendimiento y optimización de los procesos de recursos humanos en la institución.

Por las razones expuestas anteriormente es indispensable desarrollar un Sistema para la administración de recursos humanos basado en la plataforma JEE y sobre un motor de base de datos orientado a objetos DB4O para optimizar la gestión de procesos Municipalidad Provincial Chucuito Juli.

1.4. OBJETIVOS

1.4.1. Objetivo General.

Determinar qué el sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, optimiza la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial de Chucuito Juli.

1.4.2. Objetivos Específicos.

- a) Determinar los requerimientos iniciales de acuerdo a la lógica de cada proceso. Para mejorar la correcta administración de recursos humanos dentro de la Municipalidad.
- b) Proporcionar un sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, que le permita automatizar la gestión de procesos en la administración de recursos humanos a la Municipalidad Provincial Chucuito Juli.

- c) Evaluar las incidencias una vez puesto el sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, en la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial Chucuito Juli.

1.5. HIPÓTESIS DE LA INVESTIGACIÓN

1.5.1. Hipótesis General.

El sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, tiene un impacto a nivel de optimización de la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial Chucuito Juli.

1.5.2. Hipótesis Específica.

- a) Los requerimientos iniciales de acuerdo a la lógica de cada proceso, mejorar la correcta administración de recursos humanos dentro de la Municipalidad.
- b) El sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, automatiza la administración de procesos en la administración de recursos humanos de la Municipalidad Provincial de Chucuito Juli.
- c) Se evaluara el adecuado funcionamiento del sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, para la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial Chucuito Juli.

1.6. ALCANCES

El sistema es propuesto para ser utilizado en varias Municipalidades por que se adecuan al caso de estudio de la investigación, también la investigación trata de dar a conocer, de cierta manera que el uso de tecnologías de base de datos orientado a objetos, es adecuado para el desarrollo de aplicaciones empresariales por sus estándares y especificaciones.

Otro de los alcances más próximos de la implementación de este sistema, es que permitirá que se ahorre una gran cantidad de tiempo y esfuerzo a medida de que lo que actualmente se hace de manera mecánica pueda automatizarse, de forma que se obtenga un control mucho más extenso y eficiente para su administración.

1.7. LIMITACIONES

- Para el módulo de asistencia, la asistencia registrada anterior a la implantación del sistema están expresadas en tarjetas de asistencia (picadas por un reloj digital), dichas tarjetas de deben de expresar en información digital para su almacenamiento, sobre una base de datos segura y confiable.
- Para el módulo de planillas, para la elaboración de planillas se toma en cuenta las diferentes normas, decretos entre otros, establecidas por las entidades del gobierno, como: Essalud, AFPs, bonificación, descuentos entre otros, esto se sujeta al cambio constante de las fórmulas matemáticas a utilizar.

- Para el módulo de escalafón, toda la información del personal se tiene en documentos físicos, cargar dicha información será punto de limitación por su trabajo y tiempo hasta tener toda esa información en digital.
- Para la base de datos del sistema se debe de resaltar que dicha base de datos será dentro de un motor de base de datos orientado a objeto esto limita algunos parámetros al sistema.





MARCO TEÓRICO

2.1. ANTECEDENTES

2.1.1. Antecedentes Nacionales.

CALDERÓN, J. L.& RODRÍGUEZ, C.(2010). Realizaron un artículo de investigación científica titulado: “Proyectos de Desarrollo de Sistemas de Información - El Usuario Final como Factor Clave de Éxito”.El artículo científico que investiga al usuario final como factor clave de éxito y tiene estadísticas significativas sobre el fracaso, total o parcial, de los desarrollos de sistemas de información (SI) –basados en tecnologías de la información y las telecomunicaciones. Esta crítica problemática persiste en una gran cantidad de organizaciones, a pesar de la utilización de modernas metodologías y técnicas para este tipo de proyectos, así como de los avances en la literatura y la investigación sobre el área. La importancia de esta investigación en proceso radica en identificar factores críticos de éxito inherentes al usuario-final y mostrar la correlación existente entre los mismos. El objetivo principal es determinar cómo las actitudes y aptitudes del usuario-final impactan los desarrollos de SI, a fin de brindar información preventiva, al respecto, a las organizaciones que emprendan estos proyectos. El beneficio último pretendido es el de aportar fundamentos teóricos que promuevan la relevancia de conocer con antelación actitudes y aptitudes claves del usuario-final y la necesidad de darles la atención adecuada. Además, contribuir a fortalecer las estrategias de desarrollo de SI, en este sentido, para actuar en concordancia y con la debida anticipación.

2.1.2. Antecedentes Internacionales.

QUINTAN, Y., CAMEJO, L. & DÍAZ, A. (2008). Realizaron un artículo de investigación científica titulado: Diseño de la Base de Datos para Sistemas de Digitalización y Gestión de Medias. El Proyecto de Captura y Catalogación de Medias que se lleva a cabo en el Centro de Desarrollo de Software de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 de la Universidad de las Ciencias Informáticas (UCI), desarrolla un sistema que tiene como objetivo principal automatizar los procesos que se realizan durante el manejo y almacenamiento de materiales audiovisuales en empresas que cuenten con grandes volúmenes de este tipo de archivos, tal es el caso de televisoras como el Instituto Cubano de Radio y Televisión (ICRT) en Cuba y Venezolana de Televisión (VTV) en Venezuela. Algunos de estos procesos son: gestión de fallas en las medias, catalogación de medias, gestión de solicitudes y préstamos de materiales, entre otros. Para cumplir el objetivo que persigue este software, nombrado Sistema de Captura y Catalogación de Medias (SCCM), se necesita una base de datos, solución que se propone en esta investigación. A través de este trabajo se muestra dicha solución describiendo la base de datos desarrollada con su validación teórica y funcional. El informe incluye un estudio sobre las soluciones existentes, las herramientas que se van a utilizar y las técnicas empleadas para el diseño de la base de datos.

SILVA, D. A. & MERCERAT, B. (2012). Realizaron el artículo científico cuyo título: Construyendo aplicaciones web con una metodología de

diseño orientada a objetos. Ellos resumieron que el artículo tiene como principal objetivo mostrar las ventajas del uso de una metodología de diseño orientada a objetos para desarrollar aplicaciones web. Existen en la actualidad tecnologías que permiten un rápido desarrollo de aplicaciones poco reusables y difíciles de mantener. La metodología propuesta en este artículo, aplicada con las tecnologías brevemente descritas, permite obtener aplicaciones mediante un proceso de desarrollo en capas, aprovechando al máximo la potencia de la programación orientada a objetos.

GUERRERO, L. A., PORTUGAL, R. C. & FULLER, D. A. (2009). Realizaron el artículo con el título: Algunas Experiencias en el Desarrollo de Interfaces Web. Resumieron: Internet es una plataforma ideal para el desarrollo de aplicaciones distribuidas y el World Wide Web es un framework ideal para el desarrollo de interfaces. En el presente artículo se describen algunas experiencias sobre la implementación de interfaces Web para aplicaciones distribuidas, principalmente aplicaciones colaborativas. También se muestra un esquema de comunicación para aplicaciones cliente-servidor, un sistema alternativo para el almacenamiento temporal de datos y un sistema de notificación de eventos, todo bajo Web. Este esquema está basado en los patrones de diseño: MVC y broker. Se muestran también algunas aplicaciones desarrolladas sobre Web siguiendo el enfoque propuesto.

POSADA, N. & SOL, D. (2011). Realizaron el trabajo de investigación titulado: Modelado de Datos Orientado a Objetos para un Sistema de Información Geográfica. Los autores llegaron a la conclusión: Nuestro trabajo presenta el diseño y la implementación de Objetos Geográficos (OG) que permitan el manejo de datos en un Sistema de Información Geográfica (SIG). Utilizando una Base de Datos Objeto-Relacional (Informix Universal Server IUS), es posible manejar la persistencia de los OG. El modelo describe OG del mundo real, así como conceptos del enfoque orientado a objetos tales como la herencia, la asociación y la clasificación. Un visualizador de objetos geográficos apoya en la administración del almacenamiento y la recuperación de objetos geográficos, permitiendo así el manejo de la persistencia. De esta manera los objetos geográficos no necesitan ser armados y desarmados para su utilización.

2.2. MARCO CONCEPTUAL

2.2.1. Sistema de Información.

(LAUDON, 2002: 43). Señala lo siguiente: Un sistema de información se puede definir técnicamente como un conjunto de componentes interrelacionados que reúne (u obtiene), procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también ayudan a los administradores y trabajadores a analizar problemas, visualizar aspectos complejos y crear productos nuevos.

Los sistemas de información contienen información acerca de personas, lugares y cosas importantes dentro de la organización o en su entorno, se expresa gráficamente en la siguiente imagen.

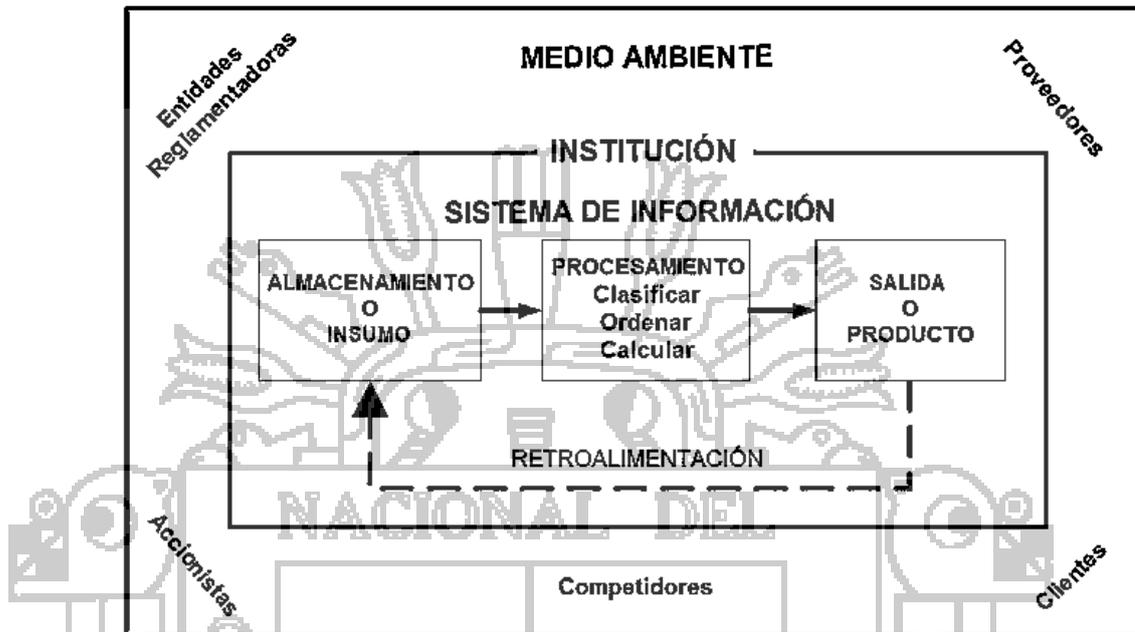


Figura N° 01: Funciones de un sistema de información.

Fuente: (LAUDON. 2002: 43)

El término información se refiere a datos a los que se les ha dado una forma que tiene sentido y es útil para los humanos. Los datos, en cambio, son flujos de hechos en bruto que representan sucesos ocurridos en las organizaciones o en el entorno físico, antes de ser organizados y acomodados de tal forma que las personas puedan entenderlos y usarlos.

Tres actividades de un sistema de información producen la información que las organizaciones necesitan para tomar decisiones, controlar operaciones, analizar problemas y crear productos o servicios

nuevos. Estas actividades son: entrada, procesamiento y salida. La entrada captura o reúne datos en bruto del interior de la organización o de su entorno externo. El procesamiento convierte esas entradas brutas en una forma que tiene más significado. La salida trasfiere la información procesada a las personas que la usarán o a las actividades en las que será usada. Los sistemas de información también requieren retroalimentación, que consiste en salidas que se devuelven a los miembros apropiados de la organización para ayudarles a evaluar o corregir la etapa de entrada. (LAUDON, 2002: 44)

2.2.1.1. Componentes de un Sistema de Información.

Sus componentes son los siguientes:

- a. **DATOS**; En general se consideran datos estructurados como los no estructurados, las imágenes, sonidos, etc.
- b. **APLICACIONES**; Se incluyen las aplicaciones manuales y automáticas.
- c. **TECNOLOGÍA**; El software y el hardware; los sistemas operativos; los sistemas de gestión de base de dato; los sistemas de redes.
- d. **INSTALACIONES**; En ellas se ubican y se mantienen los sistemas de información.
- e. **PERSONAL**; Los conocimientos específicos que ha de tener el personal de los sistemas de información para planificarlos, organizarlos, y administrarlos y gestionarlos

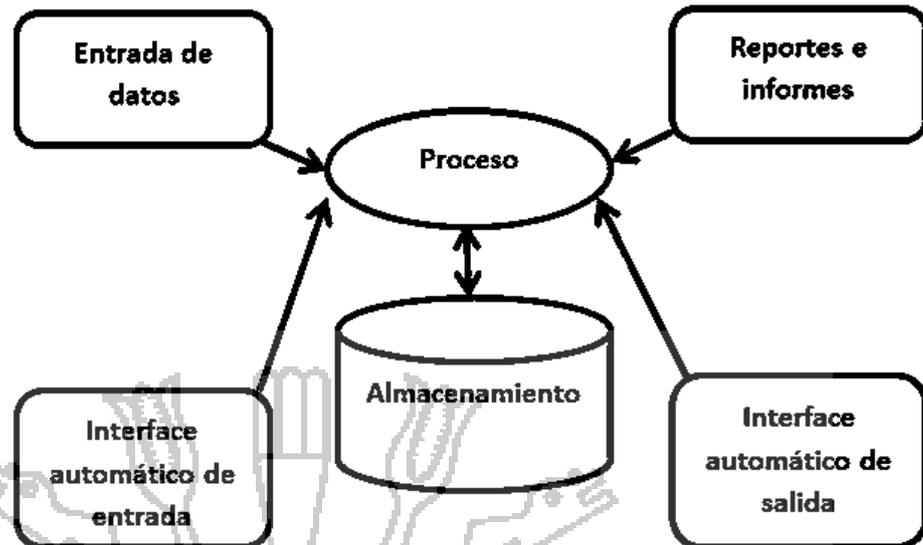


Figura N° 02: Componentes de un sistema de información.

Fuente: (LAUDON, 2002)

2.2.1.2. Sistemas de Información Web.

La evolución de internet y el posicionamiento global para la comunicación global y el surgimiento y desarrollo de Web como servicio indispensable para la distribución de información. Los sitios Web, como expresión de sistemas de información. Entonces se define a los sistemas de información al conjunto de elementos relacionados ordenados según ciertas reglas que aporta el sistema objeto es decir, la organización a la que sirve y que marca sus directrices de funcionamiento la información necesaria para el cumplimiento de sus fines; para ello, debe recoger, procesar, y almacenar datos, procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación. (ROLDÁN, 2010).

2.2.2. Base de Datos Orientado a Objetos.

Una base de datos orientada a objetos es una base de datos inteligente soporta el paradigma orientado a objetos almacenando métodos y datos, y no solamente datos. Está diseñada para ser eficaz, desde el punto de vista físico, para almacenar objetos complejos. Evite el acceso a los datos; esto gracias a los métodos almacenados en ella. Es más segura, ya que no permite tener acceso a los datos (objetos); esto debido a que para poder entrar se tiene que hacer por los métodos que haya utilizado el programador.(SILBERSCHATZ, 2002).

Un SGBDOO¹ es un SGBD que almacena objetos y por tanto posee todas las ventajas de la orientación a objetos. Las bases de datos orientadas a objetos, se diseñan para trabajar bien en conjunción con los lenguajes de programación orientados a objetos como:

- JAVA
- C++
- Visual Basic.Net
- C#

Características de un SGBDOO

- Persistencia
- Concurrencia
- Recuperación
- Gran almacén secundario
- Consultas

¹ SGBDOO, por sus siglas: Sistema Gestor de Base de Datos Orientado a Objetos.

- Abstracción
- Encapsulación
- Modularidad
- Jerarquía
- Tipos
- Generosidad

En total deben de ser 13 características obligatorias para los SGBDOO, basado en dos criterios:

1. Debe ser un sistema orientado a objetos.
2. Debe de ser un sistema gestor de base de datos.

Esto según ATKINSON. (1989).

Y las características son:

- 1) Debe de soportar objetos complejos. Debe de ser posible construir objetos complejos aplicando constructores a objetos básicos.
- 2) Identidad del objeto. Todos los objetos deben tener un identificador, el cual es independiente de los valores de sus atributos.
- 3) Encapsulamiento. Los programadores solo tienen acceso a la especificación de interfaz de los métodos, y los datos e implementación de estos métodos están ocultos en los objetos.
- 4) Tipos o clases. El esquema de una base de datos orientada a objetos contiene un conjunto de clases o tipos.

- 5) Tipos o clases deben de ser capaces de heredar de sus súper tipos o súper clases los atributos y los métodos.
- 6) La sobrecarga debe ser soportada. Los métodos deben de poder aplicarse a diferentes tipos.
- 7) El DML debe ser completo. El DML en los sistemas gestores de base de datos orientados a objetos debe ser un lenguaje de programación de propósito general.
- 8) El conjunto de tipos de datos debe ser extensible. No habrá distinción entre los tipos definidos por el usuario y los tipos definidos por el sistema.
- 9) Persistencia de datos. Los datos deben mantenerse después de que la aplicación que los creó haya finalizado, el usuario no tiene que hacer copia explícitamente.
- 10) El SGBD debe de ser capaz de manejar bases de datos grandes.
- 11) El SGBD debe soportar la concurrencia. Debe de disponer del mecanismo para el control de la concurrencia.
- 12) Recuperación. El sistema gestor debe de proveer mecanismos de recuperación de la información en caso de fallo del sistema.
- 13) El SGBD debe de proveer de una manera fácil de hacer consultas.

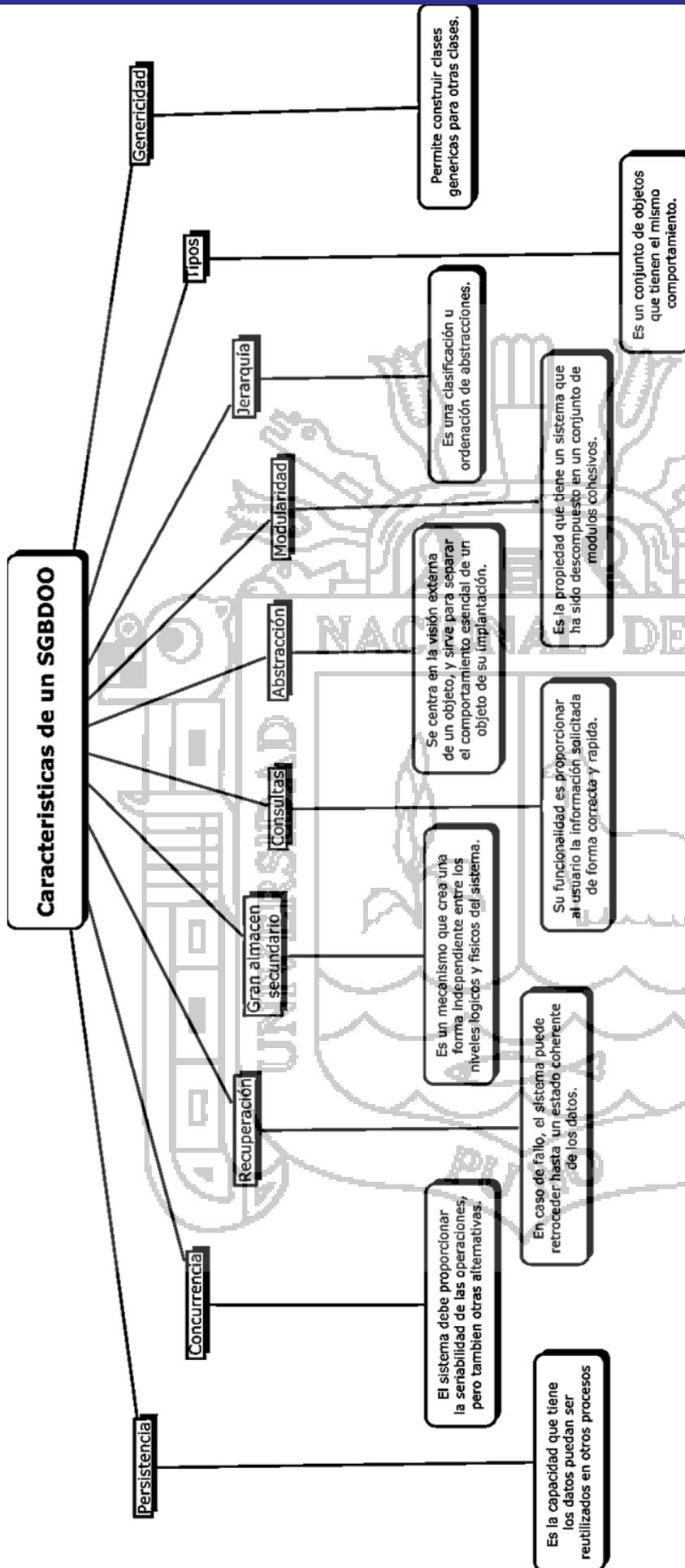


Figura N° 03: Características de un sistema gestor de base de datos orientado a objetos.

Fuente: (ATKINSON, 1989).

El paradigma orientado a objetos se basa en el encapsulamiento de datos y del código relacionado con cada objeto en una sola unidad. Conceptualmente, todas las interacciones entre cada objeto y el resto del sistema se realizan mediante mensajes. Por lo tanto, la interfaz entre cada objeto y el resto del sistema se define mediante un conjunto de mensajes permitidos. En general, cada objeto está asociado con:

- Un conjunto de variables que contiene los datos del objeto; las variables corresponden con los atributos del modelo E-R.
- Un conjunto de mensajes a los que responde; cada mensaje puede o no tener parámetros o tener uno o varios.
- Un conjunto de métodos, cada uno de los cuales es el código que implementa un mensaje; el método devuelve un valor como respuesta al mensaje.

Según Atkinson, (1989). En entorno OO² no implica uso de mensajes físicos en redes informáticas. Por el contrario, hace referencia al intercambio de solicitudes entre los objetos, independientemente de los detalles correctos de su implementación. Se utiliza a veces la expresión invocar un método para detonar al hecho de enviar un mensaje a un objeto y la ejecución del método correspondiente.

² OO por sus siglas: Orientado a Objetos

EJEMPLOS: CLASES DE OBJETOS

```
class empleado {  
    // Variables  
    string nombre;  
    string dirección;  
    date fecha de alta;  
    int sueldo;  
    // Mensajes  
    int sueldo-anual ();  
    string obtenerNombre ();  
    string obtenerDireccion ();  
    int definirDireccion (string nueva-dir);  
    int antigüedad();  
};
```

Generalmente en una base de datos hay muchos objetos similares (se entiende que responden a los mismos mensajes, utilizan los mismos métodos y tienen variables del mismo nombre y tipo). Por tanto sería un derroche definir por separado cada uno de estos objetos. Por tanto, los objetos se agrupan para formar clases. Todos los objetos de una clase comparten una definición común, pese a que se diferencien en los valores asignados a las variables. El concepto de clase del modelo orientado a objetos se corresponde con el concepto de entidad del modelo E-R³. (DBSHARDS, 2009).

El ejemplo Empleado muestra las variables y los mensajes que responden a los objetos de la clase; no se muestran aquí los métodos para el tratamiento de los mensajes.

Los principales conceptos que se utilizan en las BDOO son:

³ E-R abreviatura de: Entidad - Relación

A: IDENTIDAD DE OBJETOS.

Un sistema de BDOO provee una identidad única a cada objeto independiente almacenado en la base de datos. Esta identidad única suele implementarse con un identificador de objeto único, generado por el sistema, u OID. El valor de un OID no es visible para el usuario externo, sino que el sistema lo utiliza a nivel interno para identificar cada objeto de manera única y para crear y manejar las referencias entre objetos. La principal propiedad que debe tener un OID es la de ser inmutable; es decir, el valor del OID para un objeto en particular nunca debe cambiar. Esto preserva la identidad del objeto del mundo real que se está presentando. También es preferible que cada OID se utilice sólo una vez; esto es aunque un objeto se elimine de la Base de datos, su OID no se deberá asignar a otro objeto. Estas dos propiedades implican que el OID no debe depender del valor de ningún atributo del objeto, pues estos valores pueden cambiar. También suele considerarse inapropiado basar el OID en la dirección física del objeto en el almacenamiento, ya que una reorganización de los objetos de la base de datos podría cambiar los OID. Sin embargo, algunos sistemas sí usan la dirección física como OID para aumentar la eficiencia de la obtención de los objetos. Si la dirección física cambia, puede colocarse un apuntador indirecto en la dirección anterior, dando la nueva ubicación física del objeto. Un sistema de BDOO debe contar con algún mecanismo para generar los OID con la propiedad de inmutabilidad.

Algunos modelos de datos OO requieren que todo se represente como un objeto, ya sea un valor simple o un objeto complejo; así, todo valor básico, como un entero, una cadena o un valor boleano, tiene un OID. Con ello dos valores básicos pueden tener diferentes OID, lo cual es muy útil en algunos casos. Por ejemplo, en algunas ocasiones se podría usar el valor entero 50 para representar un peso en Kilogramos, y en otras para referirse a la edad de una persona. Así podrían crearse dos objetos básicos con diferentes OID, y ambos tendrían el mismo valor básico de 50. Aunque resulta útil como modelo teórico, esto no es muy práctico porque puede obligar a generar demasiados OID. Por ello también, la mayor parte de los sistemas de BDOO⁴ permiten representar tanto objetos como valores. Todo objeto debe tener un OID inmutable, pero los valores no tienen OID y se representan así mismo. (DBSHARDS, 2009).

- Los objetos tienen identidades únicas, independientes de los valores de sus atributos.
- La estructura orientada a objetos automáticamente impone las restricciones relacionales, generalmente más aplicables: dominio, llave integridad de entidad e integridad referencial.

B: CONSTRUCTORES DE TIPOS

En las BDOO, los valores (o estados) de los objetos complejos se pueden construir a partir de otros objetos mediante ciertos constructores de tipos. Una forma de representar tales objetos es

⁴ BDOO, por sus siglas: Base de Datos Orientado a Objetos

considerar a cada objeto como tripleta (i, c, v) , donde i es un identificador de objeto único (el OID), c es un constructor (esto es, una indicación de cómo se construye el valor del objeto) y v es el valor (o estado) del objeto. Puede haber varios constructores, según el modelo de datos y el sistema OO.

Los tres constructores básicos son:

- Constructores de átomos.
- Constructores de tuplas.
- Constructores de conjuntos.

Otros constructores de uso más común son los de listas y de arreglos.

También existe un dominio D que contiene todos los valores atómicos básicos que están disponibles directamente en el sistema. Por lo regular estos incluyen los enteros, los números reales, las cadenas de caracteres, los tipos boléanos, las fechas y cualesquiera otros tipos de datos que el sistema maneje directamente.

C: ENCAPSULAMIENTO:

(CASILLAS, 2008), señala que la estructura de los objetos como las operaciones que se pueden aplicar a ellos se incluyen en las definiciones de clases de los objetos.

D: COMPATIBILIDAD CON LENGUAJES DE PROGRAMACION

Si se sigue el enfoque cuando se utilizan los diagramas de Entidad-Relación para modelar los datos y luego se convierten de manera manual en un conjunto de relaciones; por lo tanto los conceptos de la Programación Orientada a Objetos se utilizan simplemente como herramientas de diseño y se codifican, utilizándose para trabajar con una base de datos. (CASILLAS, 2008)

Hay varios lenguajes posibles en los que se pueden integrar estos conceptos:

- Una opción es extender un lenguaje para el tratamiento de datos como el SQL añadiendo tipos complejos y la programación orientada a objetos. Los sistemas proporcionan extensiones orientadas a objetos a los sistemas relacionales se denominan sistemas relacionales orientados a objetos.
- Otra opción es tomar un lenguaje de programación orientado a objetos ya existente y extenderlo para que trabaje con las bases de datos. Estos lenguajes se denominan lenguajes de programación persistentes. Estos lenguajes permiten a los programadores trabajar directamente con los datos, desde el lenguaje de programación; sin tener que pasar por un lenguaje para el tratamiento de datos como SQL. Se denominan persistentes porque los datos siguen existiendo una vez que el programa que los creó ha concluido.

- A la hora de decidir que opción utilizar se debe tener en cuenta que los Lenguajes Persistentes suelen ser potentes y resulta relativamente sencillo cometer errores de programación que dañen las bases de datos.

E: JERARQUIA DE TIPOS Y HERENCIA.

Los esquemas de BDOO suelen necesitar un gran número de clases. Sin embargo, varias clases son parecidas entre sí. Para permitir la representación directa de parecidos entre las clases, hay que ubicarlas en una jerarquía de especializaciones. Las especializaciones de las clases son denominadas subclases; lo cual especifica atributos y métodos adicionales para una clase existente. Los objetos creados por medio de una sub clases heredan todos los atributos y métodos de la clase padre. Algunas de estas características heredadas pueden ellas mismas haber sido heredadas de clases más altas en la jerarquía. (CASILLAS, 2008).

EJEMPLO: (Grafico)

```
Class persona {  
  string nombre;  
  string dirección;  
};  
Class cliente isa persona {  
  int interés-prestamo;  
};  
Class empleado isa persona {  
  date fecha de alta;  
  int sueldo;  
};  
Class secretaria isa empleado {  
  int velocidad;  
  int horas-trabajadas  
};
```

F: MANEJO DE OBJETOS COMPLEJOS.

Los objetos se consideran complejos porque requieren un área de almacenamiento sustancial y no forman parte de los tipos de datos estándar que suelen ofrecer los SGBD. Puesto que el tamaño de los objetos es considerable, un SGBD podría obtener una porción del objeto y proporcionarla al programa de aplicación antes de obtener todo el objeto. El SGBD podría también usar técnicas de almacenamiento intermedio y caché para obtener por anticipado porciones del objeto, antes de que el programa de aplicación necesite tener acceso a ellas.

En un SGBDOO, esto puede lograrse definiendo un nuevo tipo de datos abstracto para los objetos no interpretados y suministrados los métodos para seleccionar, comprar y exhibir tales objetos. Como un SGBDOO permite a los usuarios crear nuevos tipos, y como un tipo incluye tanto estructura como operaciones, podemos considerar que un SGBDOO tiene un sistema de tipos extensibles. Podemos crear bibliotecas de nuevos tipos definiendo su estructura y operaciones, incluso con tipos complejos. Muchos SGBDOO pueden almacenar y obtener objetos no estructurados extensos en forma de cadenas y caracteres o de bits, que se pueden pasar "tal cual" al programa de aplicación para que las interprete. Es posible almacenar y manipular objetos complejos tanto estructurados como no estructurados. (CASILLAS, 2008).

G: POLIMORFISMO.

El polimorfismo se refiere al uso de la misma firma de mensaje para dirigir diferentes métodos en diferentes clases. Cuando el diseñador envía una señal a un objeto, el método de la clase de objeto, posiblemente heredado, procesa la señal. Un método puede tener acceso directamente a atributos de un objeto destino por no nombre, al incluir cualesquiera atributos heredados de clases padres, pero debe tener acceso a atributos de otros objetos con señales secundarias.

(CASILLAS, 2008) señala lo siguiente “En síntesis este concepto permite enlazar el mismo nombre o símbolo de operador a dos o más implementaciones diferentes del operador, dependiendo del tipo de objetos a los que éste se aplique”

EJEMPLO:

OBJETO_GEOMETRICO: Forma, Área, PuntoCentral
RECTANGULO subtype_of OBJETO_GEOMETRICO
(Forma= 'rectángulo'): Ancho, Altura
TRIANGULO subtype_of OBJETO_GEOMETRICO
(Forma= 'triángulo'): Lado1, Lado2, Angulo
CIRCULO subtype_of OBJETO_GEOMETRICO(Forma= 'círculo'):
Radio

H: CREACION DE VERSIONES

Muchas aplicaciones de bases de datos que usan sistemas OO requieren la existencia de varias versiones del mismo objeto. Por lo regular, se aplican actividades de mantenimiento a un sistema de software conforme sus requerimientos evolucionan. Por lo regular, el mantenimiento implica modificar algunos de los módulos de diseño y de

implementación. Si el sistema ya está en operación, y si es preciso modificar uno o más módulos, el diseñador deberá crear una nueva versión de cada uno de ellos para efectuar cambios. Cabe señalar que puede haber más de dos versiones de un objeto. En caso que se requieran dos versiones, además del módulo original. Se puede actualizar concurrentemente las propias versiones del mismo módulo del software. Esto se llama ingeniería concurrente. Sin embargo, siempre llega el momento en que es preciso combinar (fusionar) estas dos versiones para que la versión híbrida incluya los cambios realizados. Es necesario de que sus cambios sean compatibles. Un objeto complejo, como un sistema de software, puede constar de muchos módulos. Cuando se permite la creación de múltiples versiones, es posible que cada una de esos módulos tenga varias versiones distintas y un grafo de versiones. Como se deduce del análisis anterior, un SGBDOO debe ser capaz de almacenar y controlar múltiples versiones del mismo objeto. (CASILLAS, 2008).

2.2.3. DB4O.

DB4O⁵ es un novedoso motor de base de datos orientada a objetos. Sus siglas se corresponden con la expresión "DataBase 4 (for) Objects", que a su vez es el nombre de la compañía que lo desarrolla: db4objects, Inc. Las claves innovadoras de este producto es su alto rendimiento (sobre todo en modo embebido) y el modelo de desarrollo que proporciona a las aplicaciones para su capa de acceso a datos, el

⁵ DB4O, por sus siglas en inglés Data Base 4 Objects.

cual propugna un abandono completo del paradigma relacional de las bases de datos tradicionales. (KAPUR, 2010, P. 147).

De este modo, tenemos las siguientes consecuencias directas resultantes de este nuevo paradigma:

- Deja de existir un lenguaje SQL de consultas/modificaciones para pasar a crearse sistemas de consulta por métodos delegados y actualización/creación/borrado automático de entidades mediante código compilable.
- Se elimina la necesidad de representar el modelo de datos de la aplicación en dos tipos de esquemas: modelo de objetos y modelo relacional. Ahora el esquema de datos del dominio viene representado por la implementación que se realice del diagrama de clases.
- Se consigue evitar el problema del Object-RelationallmpedanceMismatch sin sacrificar el rendimiento que los mapeadores objeto-relacionales sufren actualmente para llevar a cabo el mismo objetivo.

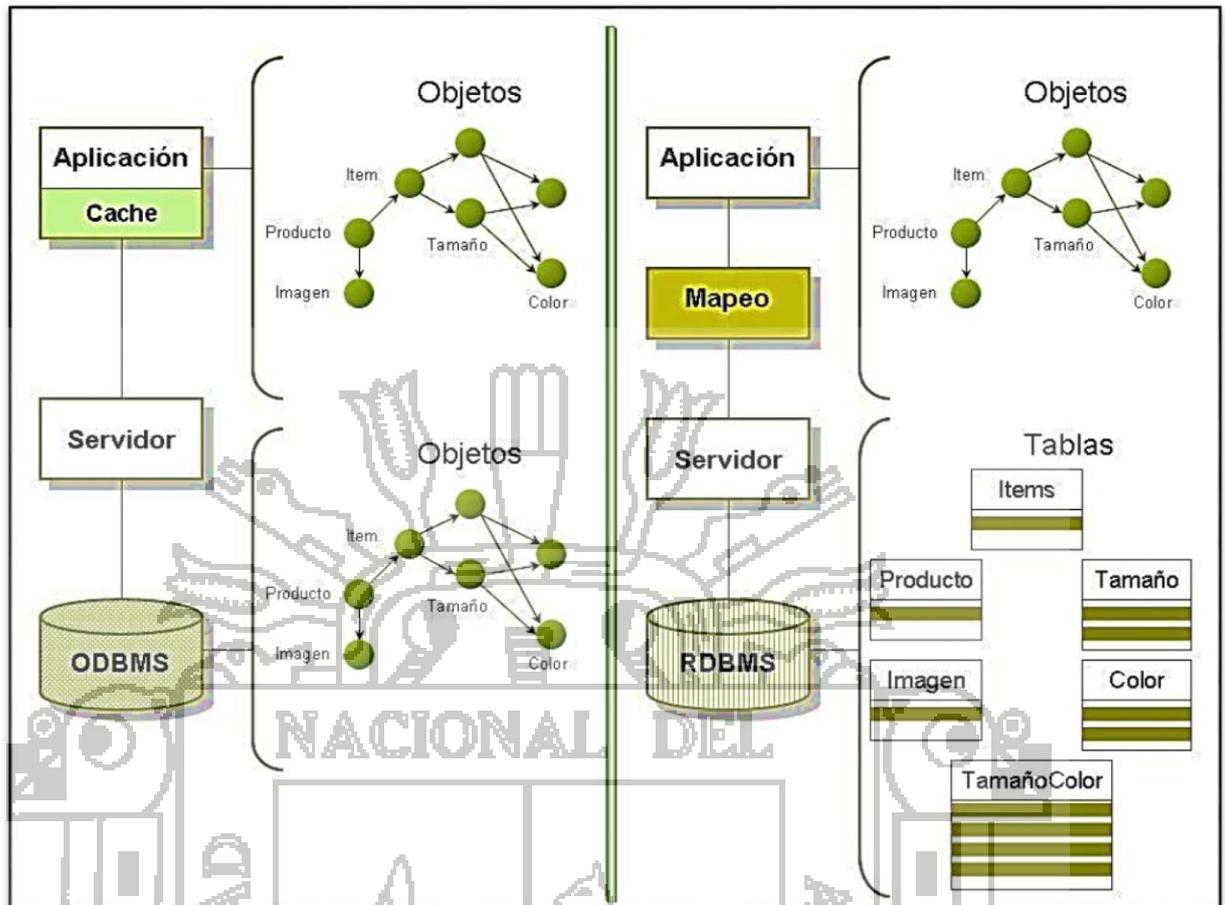


Figura N° 04: Base de datos orientado a objetos comparado con base de datos relacionales.

Fuente: (KAPUR, 2010, P. 147).

A. CARACTERISTICAS

La base de datos DB4O llegó al mercado en 2004 con el objetivo de convertirse en la corriente principal de la arquitectura de persistencia para aplicaciones integradas (en los que la base de datos es invisible para el usuario final) en general, y para dispositivos móviles e integrados que se ejecutan en Java o .NET. La base de datos db4o está diseñada para ser una plataforma universal, producto asequible que es fácil de aprender y usar.

Los entornos de destino para db4o son arquitecturas de persistencia donde no hay presente ningún administrador de base de datos y RDBMS (sistema gestor de base de datos relacional) legado es decir, principalmente en equipos, clientes móviles y de escritorio, y en el middleware. Industrias típicas de los clientes DB4O incluyen el transporte, la comunicación, la automatización, las ciencias médicas, industriales, de consumo y aplicaciones financieras, entre muchos otros. Open Source es uno de los factores clave de éxito DB4O. Éxito es la tecnología utilizada como base de datos objeto de nueva generación, nativa de Java y NET., DB4O elimina el tradicional trade-off entre el desempeño y orientación a objetos. Recientes resultados de referencia poleposition muestran que DB4O supera objeto-relacionales mappers en órdenes de magnitud, hasta 44x en los casos de uso con modelos de objetos complejos. (KAPUR, 2010, P. 147).

Los clientes, analistas y expertos coinciden en que la base de datos de objetos DB4O es una de las opciones mejores y más populares del mundo, ya que almacena y recupera objetos de forma nativa y no sólo elimina el consumo de gastos generales y de recursos de un ORM, pero también reduce en gran medida el desarrollo de productos y los costes de mantenimiento, lo que resulta en una inclinación, rápido y fácilmente integrable en una solución de desarrollo OO persistencia ambiente, muy superior en muchos casos a la de cualquier RDBMS.

DB4O único que ofrece la persistencia de objetos con cero administraciones, orientado a objetos de replicación consulta y capacidades de navegación y una pequeña huella. Su única biblioteca (JAR / DLL) es fácil de implementar y se ejecuta en el proceso de la memoria junto con la solicitud.

DB4O se destina para el uso encajado con bases de datos más pequeñas alrededor de 2-16 GByte y como máximo 256 GBytes, satisfacer las necesidades cambiantes, o para fines de depuración o refactorización. DB4O permite a los desarrolladores trabajar con estructuras de objetos como si fueran "en memoria" 'estructuras. Poca codificación adicional es necesaria para gestionar la persistencia de objetos. (KAPUR, 2010, P. 148).

B. VENTAJAS

- Mayor velocidad de desarrollo y transparencia
 - No hay mapeos entre objetos y tablas
 - No hay que crear componentes que accedan a las bases de datos
 - El código de acceso a la base es muy sencillo y entendible (métodos get, set y delete) en el caso de db4o.
- Mejor performance con objetos de negocio complejos (árboles, estructuras anidadas, relaciones N a N, relaciones recursivas)
- Fácil Backup, es decir (la base completa está en un solo archivo)
- No necesita administración

- Tiene un recolector de basura – garbagecollector – que borra los objetos que no son referenciados
- Al cambiar algo en las clases no se necesita modificar nada en la base de objetos.
- Las búsquedas se hacen directamente usando objetos.
- Los cambios en los objetos (agregar o quitar atributos a una clase) se aplican directamente en la base, sin tener que migrar datos ni reconfigurar nada.

C. DESVENTAJAS

De este modo, tenemos las siguientes consecuencias directas resultantes de este nuevo paradigma:

- Deja de existir un lenguaje SQL de consultas/modificaciones para pasar a crearse sistemas de consulta por métodos delegados y actualización/creación/borrado automático de entidades mediante código compilable.
- Se elimina la necesidad de representar el modelo de datos de la aplicación en dos tipos de esquemas: modelo de objetos y modelo relacional. Ahora el esquema de datos del dominio viene representado por la implementación que se realice del diagrama de clases.
- Se consigue evitar el problema del Object-RelationallmpedanceMismatch sin sacrificar el rendimiento que los mapeadores objeto-relacionales sufren actualmente para llevar a cabo el mismo objetivo.

La mayor clave del éxito que está teniendo este motor de base de datos frente a otros competidores que han desarrollado tecnologías similares, es que se ha optado por un modelo de licenciamiento idéntico al utilizado por empresas como MySQL: licencia dualGPL/comercial. Es decir, si se quiere desarrollar software libre con esta biblioteca, su uso no conlleva ningún coste por licencia; sin embargo si se desea aplicar a un software privativo, se aplica otro modelo de licenciamiento concreto. (KAPUR, 2010, P. 147).

2.2.4. Plataforma JEE.

JEE⁶ o Java EE es una plataforma de programación parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java⁶ con arquitectura de N capas distribuidas y que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del JCP⁷, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por JCP. Java EE es una plataforma para el desarrollo de aplicaciones empresariales. Ofrece un framework para el desarrollo de aplicaciones distribuidas multicapa basadas en Web, define una infraestructura común básica para el acceso a bases de datos, gestión de la

⁶Por sus siglas en Inglés de *Java Platform Enterprise Edition*

⁷Por sus siglas en Inglés de *Java Community Process*

persistencia, control de seguridad, gestión de transacciones; separación clara entre presentación (interfaz), modelo (lógica) de negocio y datos; es una plataforma basada en componentes: entes (objetos) intercambiables que residen en un servidor de aplicaciones y son gestionados por él. (CASILLAS, 2008).

A: ESQUEMA GENERAL DE JAVA EE

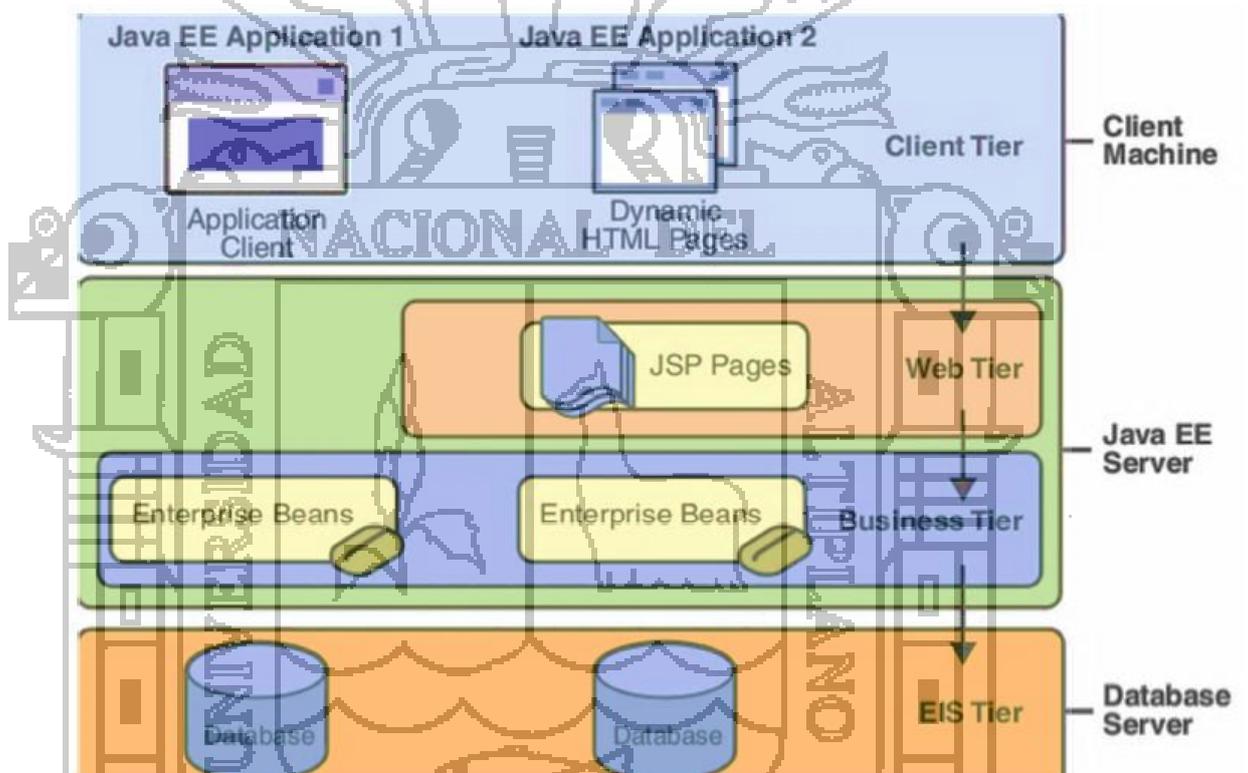


Figura N° 05: Esquema general del java EE.

Fuente: (CASILLAS, 2008).

B: COMPONENTES DE LA ARQUITECTURA JAVA EE

Tipos de componentes:

- Clientes de la aplicación:
 - clientes web [navegador]
 - clientes de escritorio Java [swing, awt]
 - clientes de escritorio no Java
- Componentes Web: servlets, paginas JSP⁸, JSF⁹:
 - Se ejecutan en el contenedor Web
 - Responsables de componer la presentación de datos en formatoHTML.
 - Suelen apoyarse en el uso de componentes JavaBeans¹⁰objetos Java que verifican 2 requisitos:
 1. tienen un constructor sin argumentos
 2. todos sus atributos son accesibles mediante pares demétodos get() y set()
- Componentes de negocio: EJB¹¹ y entidades JPA¹².
 - Se ejecutan en el contenedor de EJBs
 - Componentes (objetos Java) responsables de implementar lalógica de la aplicación
 - EJB gestionan interacciones con los clientes e implementanreglas de negocio

⁸Por sus Siglas en Inglés de Java Server Pages

⁹Por sus Siglas en Inglés de Java Server Faces

¹⁰ Modelo de componente para construir aplicaciones en JAVA

¹¹Por sus Siglas en Inglés de Enterprise JavaBeans

¹²Por sus Siglas en Inglés de Java Persistence API

- Entidades Java: objetos persistentes que representan los datos de la capa EIS.
- Capa EIS¹³.
 - Capa de datos) gestiona la información permanente del sistema.
 - Bases de datos o aplicaciones empresariales "heredadas" (legacysystems) que actúan como almacenes de datos.

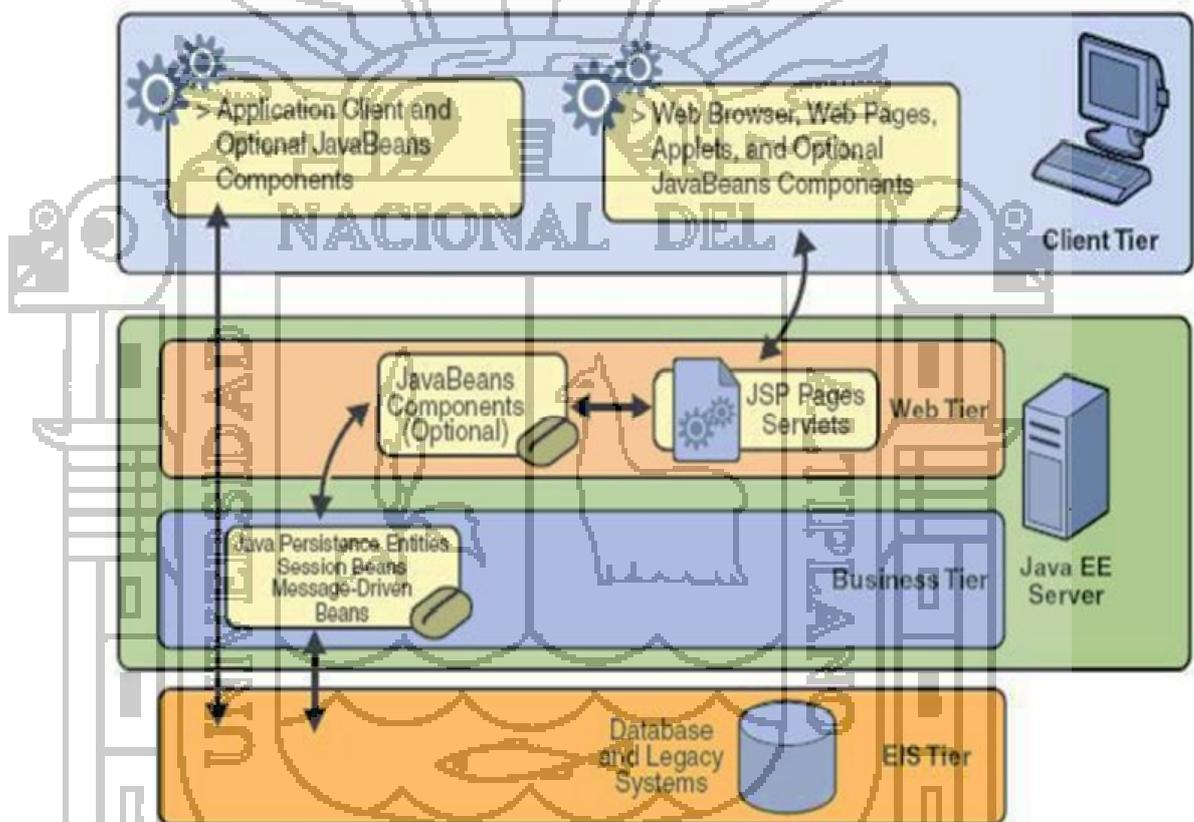


Figura N° 06: Componentes de la arquitectura java EE.

Fuente: (CASILLAS, 2005, P. 256).

2.2.5. Java Server Faces.

(MANN, 2005). Señala que la tecnología JSF, constituye un marco de trabajo (framework) de interfaces de usuario del lado de servidor para

¹³Por sus Siglas en Inglés de Enterprise InformationSystems

aplicaciones web basadas en tecnología Java y en el patrón MVC (Modelo Vista Controlador).

A: COMPONENTES:

- Una API y una implementación de referencia
- Representar componentes de interfaz de usuario (UI-User Interface) y manejar su estado
- Manejar eventos, validar en el lado del servidor y convertir datos
- Definir la navegación entre páginas
- Soportar internacionalización y accesibilidad
- Proporcionar extensibilidad para todas estas características.
- Una librería de etiquetas JSP personalizadas para dibujar Tutorial de JSF componentes UI dentro de una página JSP.

B: BENEFICIOS DE LA TECNOLOGÍA JSF

Una de las grandes ventajas de la tecnología JSF es que ofrece una clara separación entre el comportamiento y la presentación. Las aplicaciones Web construidas con tecnología JSP conseguían parcialmente esta separación. Sin embargo, una aplicación JSP no puede mapear peticiones HTTP al manejo de eventos específicos de los componentes o manejar elementos UI como objetos con estado en el servidor. La tecnología JSF nos permite construir aplicaciones Web que implementan una separación entre el comportamiento y la presentación tradicionalmente ofrecida por arquitectura UI del lado del cliente. La separación de la lógica de la presentación también le permite

a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte del proceso de desarrollo, y proporciona un sencillo modelo de programación para enlazar todas las piezas. Por ejemplo, los Autores de páginas sin experiencia en programación pueden usar las etiquetas de componentes UI de la tecnología JSF para enlazar código de la aplicación desde dentro de la página Web sin escribir ningún script. Pero lo más importante, la tecnología JSF proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos. (MANN, 2005).

2.2.6. Spring.

Spring es un framework de aplicaciones Java/JEE desarrollado usando licencia de OpenSource. Se basa en una configuración a base de javabeans bastante simple. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable. Es interesante el uso de programación orientada a aspectos (IoC). Tiene plantillas que permiten un más fácil uso de Hibernate, iBatis, JDBC¹⁴, se integra "de fábrica" con Quartz, Velocity, Freemarker, Struts, Webwork2 y tienen un plugin para eclipse. Ofrece un ligero contenedor de bean para los objetos de la capa de negocio, DAOs y repositorio de DataSources JDBC y sesiones Hibernate. Mediante un XML definimos el contexto de la aplicación siendo una potente herramienta para manejar objetos

¹⁴ Por sus siglas en inglés, Java DatabaseConnectivity

Singleton o “factorias” que necesitan su propia configuración. (TAPIA, 2011).

El objetivo de Spring es no ser intrusivo, aquellas aplicaciones configuradas para usar beans mediante Spring no necesitan depender de interfaces o clases de Spring, pero obtienen su configuración a través de las propiedades de sus beans. Este concepto puede ser aplicado a cualquier entorno, desde una aplicación JEE a un applet.

Spring proporciona:

- Una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control (IoC). Esto hace que la configuración de aplicaciones sea rápida y sencilla. Ya no es necesario tener singletons ni ficheros de configuración, una aproximación consistente y elegante. Estas definiciones de beans se realizan en lo que se llama el contexto de aplicación.
- Una capa genérica de abstracción para la gestión de transacciones, permitiendo gestores de transacción añadibles (pluggables), y haciendo sencilla la demarcación de transacciones sin tratarlas a bajo nivel. Se incluyen estrategias genéricas para JTA y un único JDBC DataSource. En contraste con el JTA simple o EJB CMT, el soporte de transacciones de Spring no está atado a entornos JEE.
- Una capa de abstracción JDBC que ofrece una significativa jerarquía de excepciones (evitando la necesidad de obtener de SQLException los códigos que cada gestor de base de datos

asigna a los errores), simplifica el manejo de errores, y reduce considerablemente la cantidad de código necesario.

- Integración con Hibernate, JDO e iBatis SQL Maps en términos de soporte a implementaciones DAO y estrategias con transacciones. Especial soporte a Hibernate añadiendo convenientes características de IoC, y solucionando muchos de los comunes problemas de integración de Hibernate. Todo ello cumpliendo con las transacciones genéricas de Spring y la jerarquía de excepciones DAO.
- Funcionalidad AOP, totalmente integrada en la gestión de configuración de Spring. Se puede aplicar AOP a cualquier objeto gestionado por Spring, añadiendo aspectos como gestión de transacciones declarativa. Con Spring se puede tener gestión de transacciones declarativa sin EJB, incluso sin JTA, si se utiliza una única base de datos en un contenedor Web sin soporte JTA.
- Un framework MVC (Model-View-Controller), construido sobre el núcleo de Spring. Este framework es altamente configurable vía interfaces y permite el uso de múltiples tecnologías para la capa vista como pueden ser JSP, Velocity, Tiles, iText o POI. De cualquier manera una capa modelo realizada con Spring puede ser fácilmente utilizada con una capa web basada en cualquier otro framework MVC, como Struts, WebWork o Tapestry. (TAPIA, 2011)

Toda esta funcionalidad puede usarse en cualquier servidor JEE, y la mayoría de ella ni siquiera requiere su uso. El objetivo central de Spring

es permitir que objetos de negocio y de acceso a datos sean reutilizables, no atados a servicios JEE específicos. Estos objetos pueden ser reutilizados tanto en entornos JEE (Web o EJB), aplicaciones “standalone”, entornos de pruebas, etc., sin ningún problema.

A: ARQUITECTURA DE SPRING

“La arquitectura en capas de Spring ofrece mucha flexibilidad. Toda la funcionalidad está construida sobre los niveles inferiores. Por ejemplo se puede utilizar la gestión de configuración basada en JavaBeans sin utilizar el framework MVC o el soporte AOP” (WALLS, 2011)

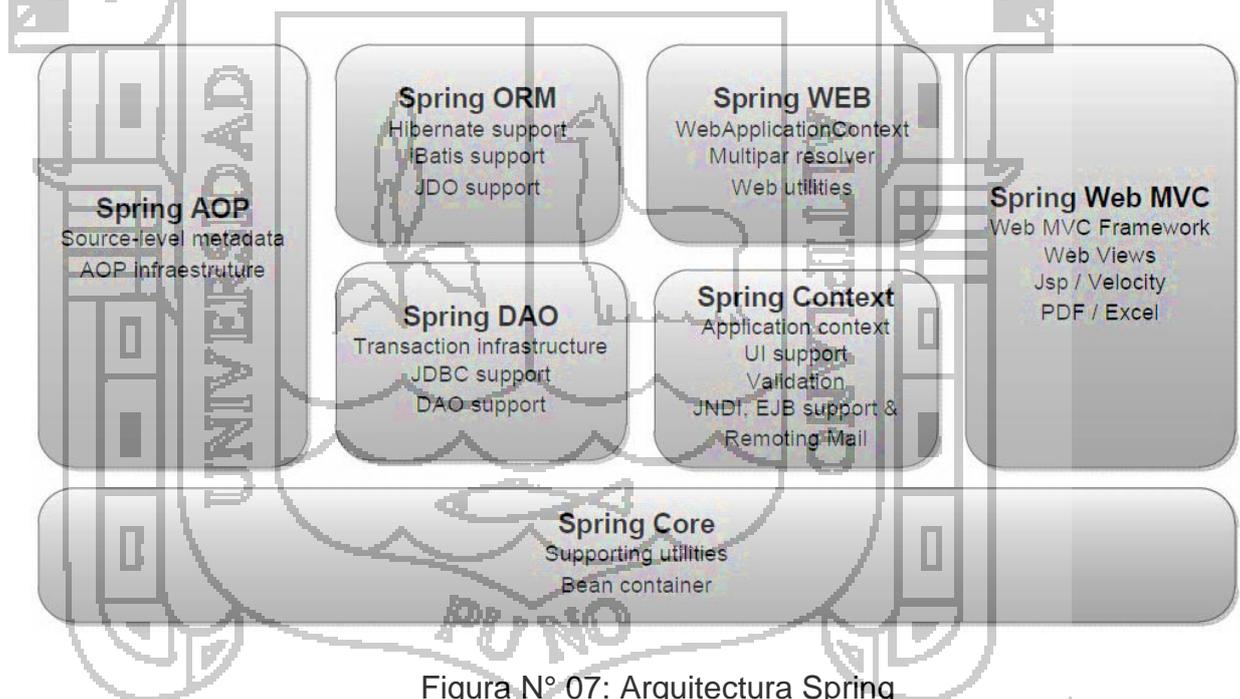


Figura N° 07: Arquitectura Spring

Fuente: (WALLS,2011)

B: SPRING SECURITY

“Spring Security es un subproyecto del framework Spring, que permite gestionar completamente la seguridad de nuestras aplicaciones Java”(FIELDS, 2000),y cuyas ventajas principales son las siguientes:

- Es capaz de gestionar seguridad en varios niveles: URLs que se solicitan al servidor, acceso a métodos y clases Java, y acceso a instancias concretas de las clases.
- Permite separar la lógica de nuestras aplicaciones del control de la seguridad, utilizando filtros para las peticiones al servidor de aplicaciones o aspectos para la seguridad en clases y métodos.
- La configuración de la seguridad es portable de un servidor a otro, ya que se encuentra dentro del WAR25 o el EAR26 de nuestras aplicaciones.
- Soporta muchos modelos de identificación de los usuarios (HTTP BASIC, HTTP Digest, basada en formulario, LDAP, OpenID, JAAS y muchos más). Además podemos ampliar estos mecanismos implementando nuestras propias clases que extiendan el modelo de Spring Security.

2.2.7. JasperReport.

JasperReports es la mejor herramienta de código libre en Java para generar reportes. Puede entregar ricas presentaciones o diseños en la pantalla, para la impresora o para archivos en formato PDF, HTML, RTF, XLS, CSV y XML. Está completamente escrita en Java y se puede

utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico.

A: FUNCIONAMIENTO DE JASPERREPORTS

JasperReports trabaja en forma similar a un compilador y a un intérprete, el usuario diseña el reporte codificándolo en XML de acuerdo a las etiquetas y atributos definidos en un archivo llamado `jasperreports.dtd` (parte de JasperReports). Usando XML el usuario define completamente el reporte, describiendo donde colocar texto, imágenes, líneas, rectángulos, cómo adquirir los datos, como realizar ciertos cálculos para mostrar totales, etc.

El archivo fuente XML debe ser compilado para obtener un reporte real. La versión compilada del fuente es nombrada "archivo jasper" (este termina con `.jasper`). Un Archivo jasper es el compilado de un código fuente. Cuando tenemos un archivo jasper, necesitamos otra cosa para producir un reporte: necesitamos datos. Esto no siempre es cierto. En algunos casos querríamos generar un reporte que no mostrara datos dinámicos, solo texto estático por ejemplo, pero esto puede simplificarse a un reporte que tiene solamente un registro vacío. Para proporcionar estos registros al "jasperengine" necesitamos presentarlos usando una interfaz especial específica llamada `JRDataSource`. Una fuente de datos más un Archivo jasper = un "archivo print". Un "archivo print" puede exportarse en muchos formatos como PDF, HTML, RTF,

XML, XLS, CVS, etc. La exportación se puede realizar utilizando clases especiales para implementar exportadores específicos. (WALLS, 2011)

2.2.8. Servidor Apache.

Existen una serie de características que convierten a Apache en uno de los servidores Web más utilizados, como son el tener el código fuente abierto, mantener una evolución rápida y continuada de versiones, poder ser utilizado por desarrolladores de cualquier plataforma, y además, es gratuito. Apache es un servidor Web multiplataforma, que permite indexación de directorios, uso de sobrenombres con las carpetas, informes configurables sobre errores http, ejecución de programas CGI¹⁵ y que además admite la última versión del protocolo http/1.1.

Una característica importante a señalar es que Apache permite trabajar con servidores virtuales tanto con direcciones IP así como con nombres virtuales. También se podría convertir nuestro servidor en un servidor Proxy. En todo momento, a través de un explorador Web, se podría conocer el estado de nuestro servidor, pues tiene registros configurables para guardar dicho estado, así como poder registrar las acciones de los usuarios. Además de CGI, Apache puede trabajar con otros lenguajes de respuesta del servidor como Perl y Java (servlets) siempre y cuando se añadan los módulos necesarios en el fichero de configuración. (TAPIA, 2011).

¹⁵ Por sus siglas en inglés Common Gateway Interface.

2.2.9. Proceso Unificado de Desarrollo.

El Proceso Unificado Racional (RationalUnifiedProcess en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM¹⁶. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Es por ello que se define claramente: Quién debe hacer, Qué hacer, Cuándo hacer y Cómo debe hacerlo. (QUATRANI, 2000).



Figura N° 08: Proceso de desarrollo de software.

Fuente: (QUATRANI, 2000).

Un proceso de software universal no existe. Las características de cada trabajo de investigación (recursos, equipo de desarrollo) exigen que todo proceso sea configurable. El Proceso Unificado para preparar todos los esquemas de un sistema de Software utiliza el Lenguaje Unificado de Modelado (UML), que además es parte esencial del proceso Unificado.

¹⁶ Por sus siglas en inglés: International Business Machines

A. DIMENSIÓN DEL RUP:

El proceso puede describirse en dos dimensiones, o a lo largo de dos ejes:

- a. *EL EJE HORIZONTAL.* Representa el tiempo y muestra el aspecto dinámico del proceso cuando se procesa, y es expresado en términos de ciclos, fases, iteraciones, e hitos.
- b. *EL EJE VERTICAL.* Representa el aspecto estático del proceso: cómo se describe en términos de actividades, artefactos..

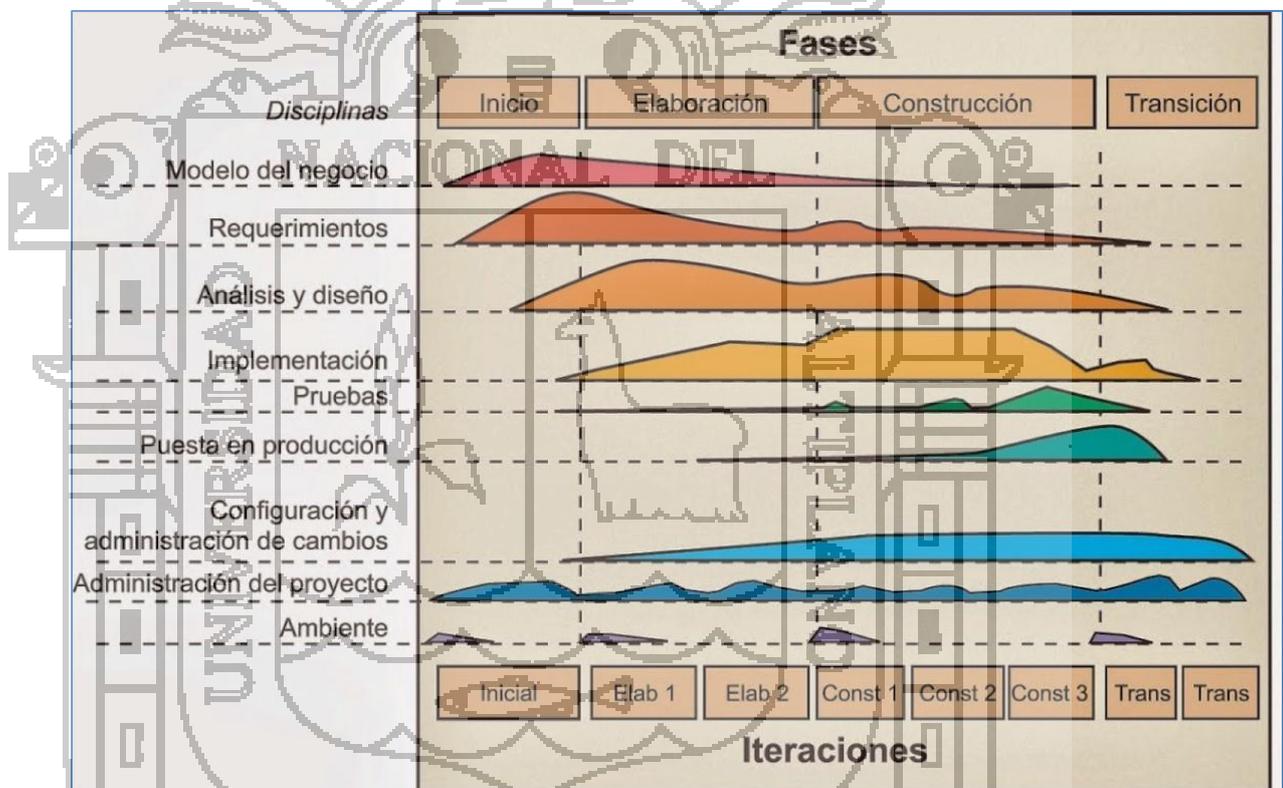


Figura N° 09: Metodología RUP.

Fuente: (QUATRANI, 2000).

Cada fase se concluye con un producto definido, en cada culminación de las fases se deben tomar ciertas decisiones críticas a tiempo, y por consiguiente se deben de lograr metas importantes.

B. PROCESOS DIRIGIDO POR CASOS DE USO

Toda esta afirmación se refiere a la utilización de los casos de uso para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. un caso de uso es la secuencia de pasos que conlleva la realización e implementación de un requerimiento planteado por el cliente. (QUATRANI, 2000).

C. PROCESOS ITERATIVO E INCREMENTAL

Es el modelo utilizado por RUP, Plantea la implementación del proyecto a realizar en iteraciones, se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración, y de esta manera se tendrían varias ventajas, entre ellas se puede mencionar la de tener pequeños avances del proyecto que son entregables al cliente y este puede probar mientras se está desarrollando otra iteración del proyecto.

D. PROCESO CENTRADO EN LA ARQUITECTURA

En esta parte se define la Arquitectura de un sistema, y una arquitectura ejecutable construida como un prototipo evolutivo. Una arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades. RUP establece refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo que va evolucionando. (QUATRANI, 2000).

E. FASES DE LA METODOLOGÍA RUP

Se divide en 4 fases secuenciales como lo indica la siguiente figura, y en cada extremo de una fase se realiza una evaluación para determinar si se cumplió con los objetivos de la fase.

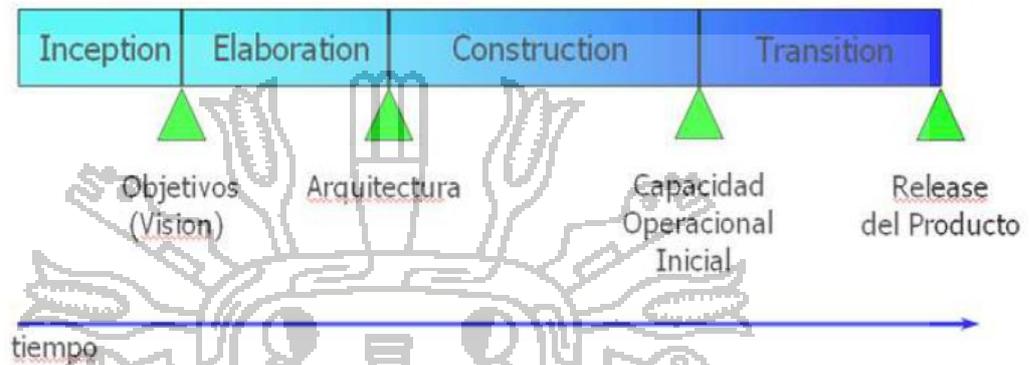


Figura N° 10: Fases de la metodología RUP.

Fuente: (QUATRANI, 2000).

Dentro de las fases de la metodología RUP, se debe aclarar que cada fase produce una nueva versión del producto.

- a. Inicio: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.
- b. Elaboración: En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

- c. **Construcción:** El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
- d. **Transición:** El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

Se describe los manuales de usuario, se completan y refinan con la información anterior, estas tareas se realizan también en iteraciones. Las fases mencionadas anteriormente son idénticas en términos de tiempo y esfuerzo aunque esto depende mucho del tipo de proyecto. (QUATRANI, 2000).

	Inicio	Elaboración	Construcción	Transmisión
Esfuerzo	~5%	20%	65%	10%
Horario	10%	30%	50%	10%

Tabla N° 01: Esfuerzo - Horario contra fases del RUP.

Fuente: (QUATRANI, 2000).

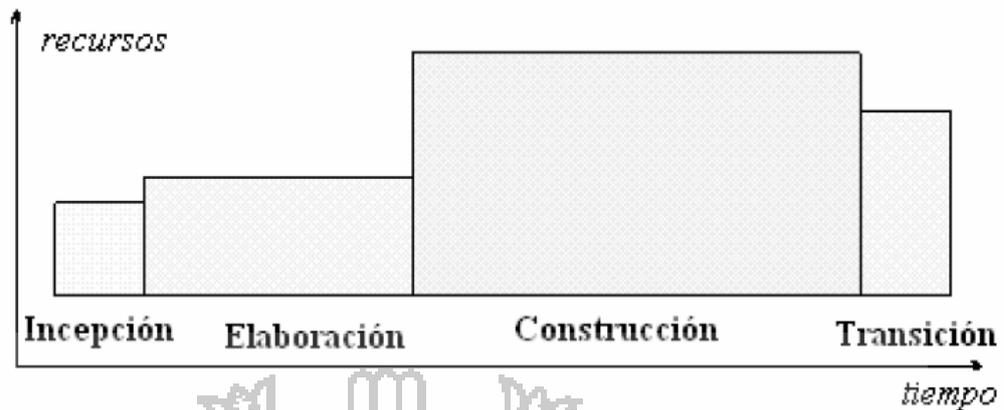


Figura N° 11: Recursos utilizados en las fases del RUP.

Fuente: (QUATRANI, 2000).

Como se muestra en la figura en todo un ciclo evolutivo las fases de incepción y elaboración serían las más pequeñas. El resultado de las cuatro fases es un software.

2.2.10. UML.

Es un lenguaje estándar que sirve para escribir los planos del software, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empotrados de tiempo real. UML es solamente un lenguaje por lo que es sólo una parte de un método de desarrollo software, es independiente del proceso aunque para que sea óptimo debe usarse en un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. (SOMMERVILLE, 2011).

A. BLOQUES DE CONSTRUCCIÓN

- a. *CLASES*: Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces.

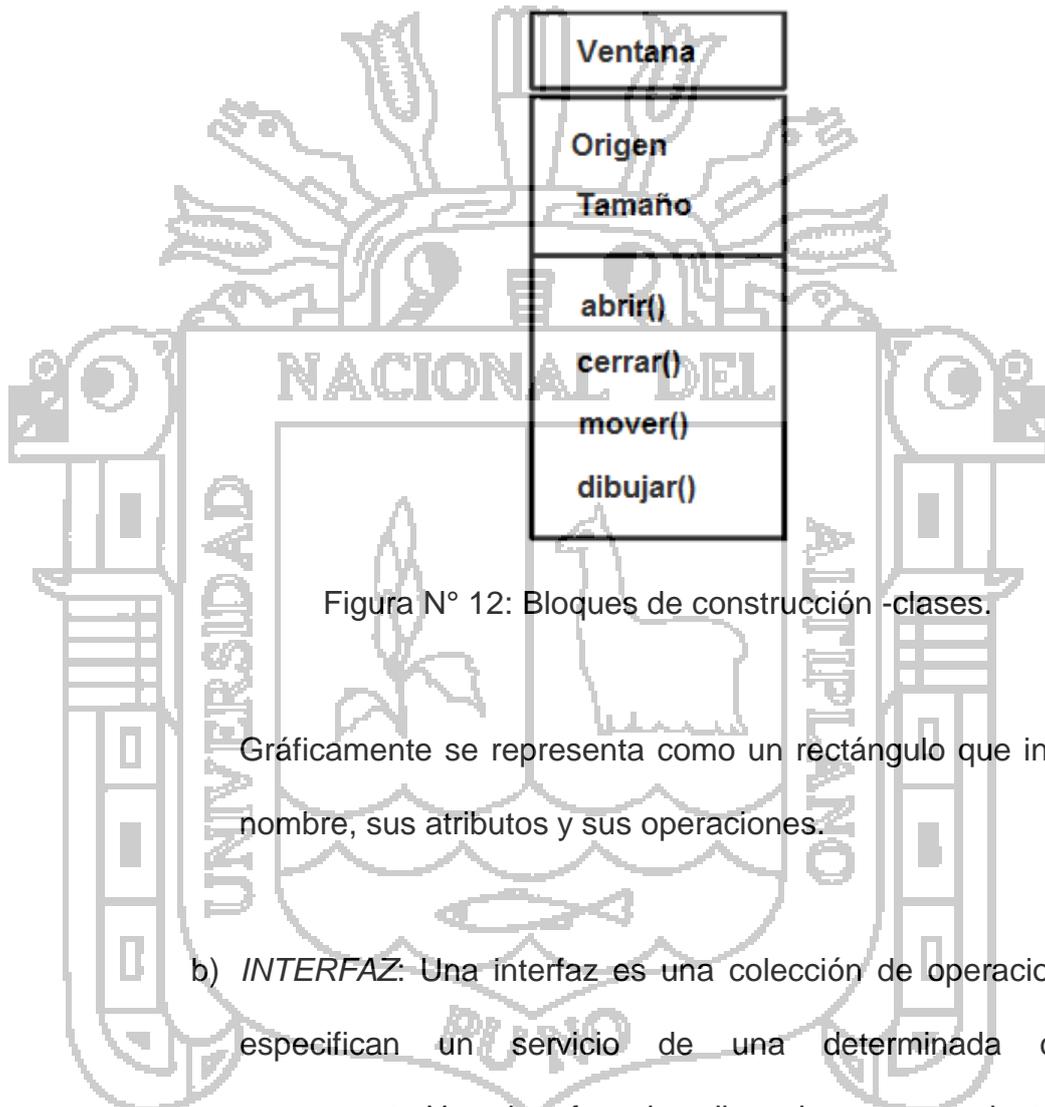


Figura N° 12: Bloques de construcción -clases.

Gráficamente se representa como un rectángulo que incluye su nombre, sus atributos y sus operaciones.

- b) *INTERFAZ*: Una interfaz es una colección de operaciones que especifican un servicio de una determinada clase o componente. Una interfaz describe el comportamiento visible externamente de ese elemento, puede mostrar el comportamiento completo o sólo una parte del mismo. Una interfaz describe un conjunto de especificaciones de operaciones (o sea su *signatura*) pero nunca su implementación. Se

representa con un círculo, y rara vez se encuentra aislada sino que más bien conectada a la clase o componente que realiza.(SOMMERVILLE, 2011).

- c) **COLABORACIÓN**: Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Las colaboraciones tienen una dimensión tanto estructural como de comportamiento. Una misma clase puede participar en diferentes colaboraciones. Las colaboraciones representan la implementación de patrones que forman un sistema. Se representa mediante una elipse con borde discontinuo.

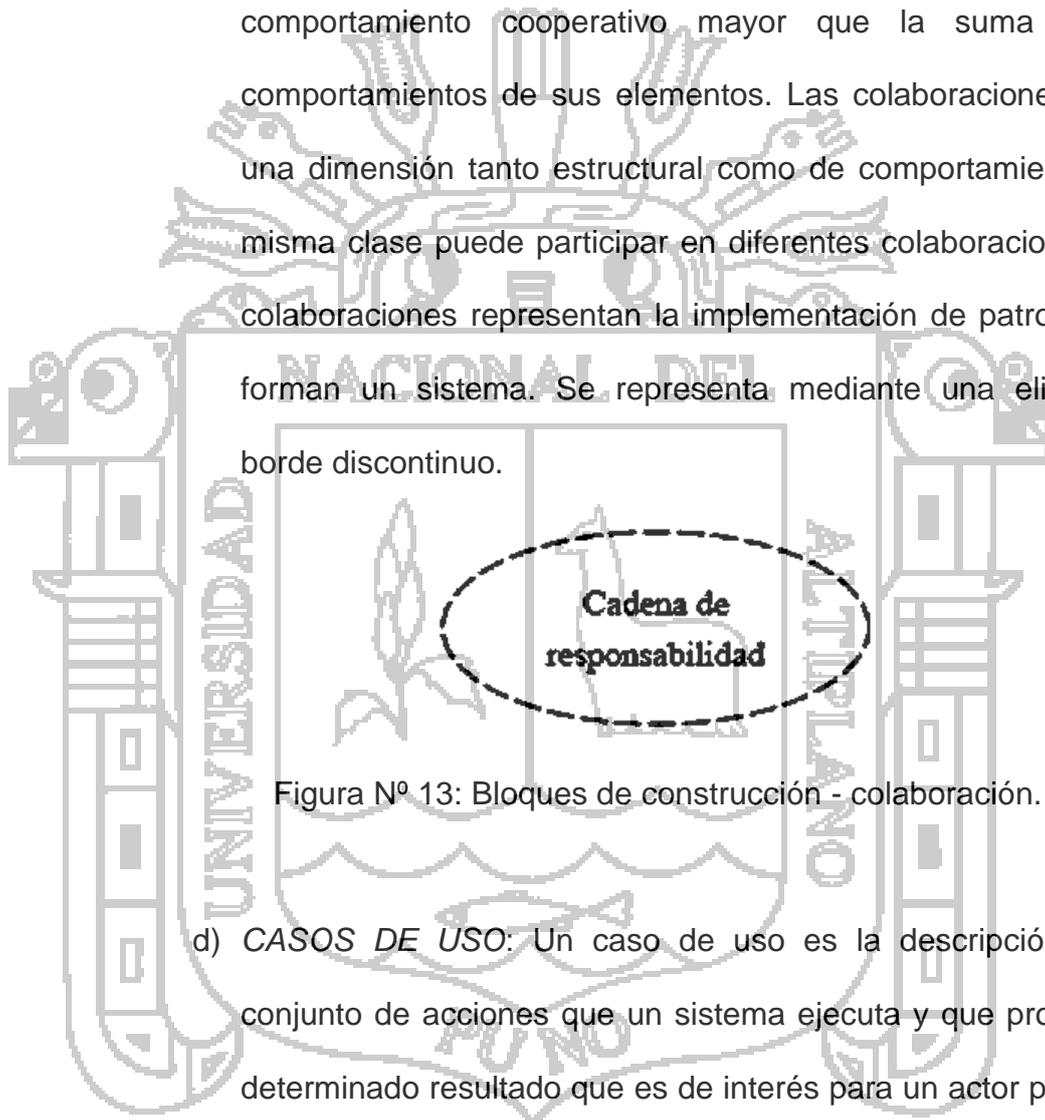


Figura N° 13: Bloques de construcción - colaboración.

- d) **CASOS DE USO**: Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular. Un caso de uso se utiliza para organizar los aspectos del comportamiento en un modelo. Un caso de uso es realizado por una colaboración. Una elipse con borde continuo.



Figura N° 14: Bloques de construcción - casos de Uso.

- e) *CLASE ACTIVA*: Es una clase cuyos objetos tienen uno o más procesos o hilos de ejecución por lo tanto pueden dar lugar a actividades de control. Una clase activa es igual que una clase, excepto que sus objetos representan elementos cuyo comportamiento es concurrente con otros elementos. Se representa igual que una clase, pero con líneas más gruesas.

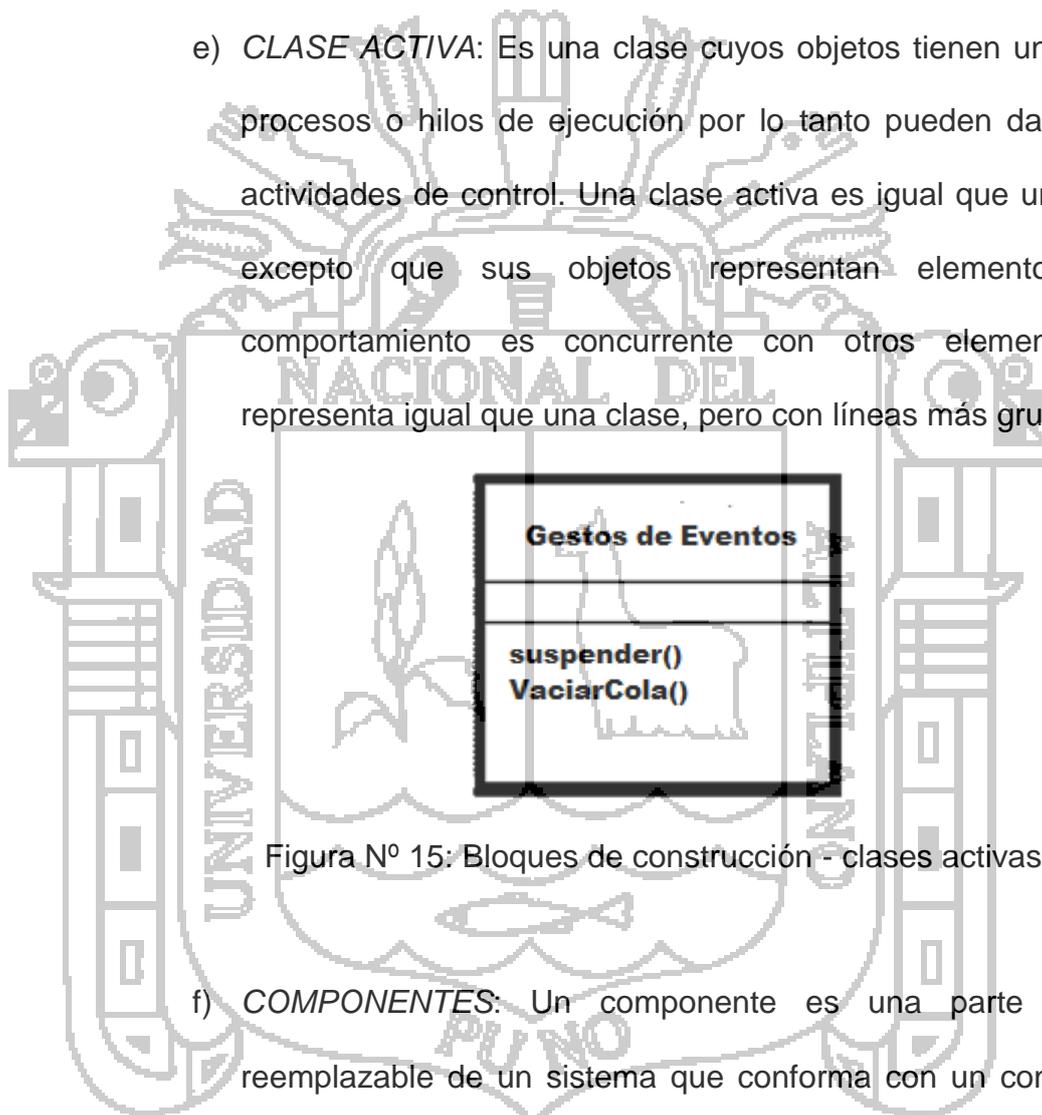


Figura N° 15: Bloques de construcción - clases activas.

- f) *COMPONENTES*: Un componente es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto. Un componente representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases, interfaces y colaboraciones. (SOMMERVILLE, 2011).

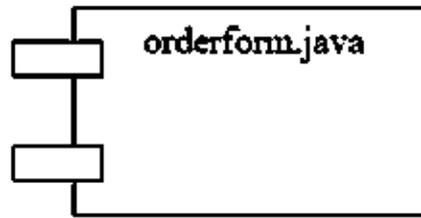


Figura N° 16: Bloques de construcción - componentes.

B. ELEMENTOS DE COMPORTAMIENTO.

a) **INTERACCIÓN:** Es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular para conseguir un propósito específico. Una interacción involucra otros muchos elementos, incluyendo mensajes, secuencias de acción (comportamiento invocado por un objeto) y enlaces (conexiones entre objetos). La representación de un mensaje es una flecha dirigida que normalmente con el nombre de la operación.

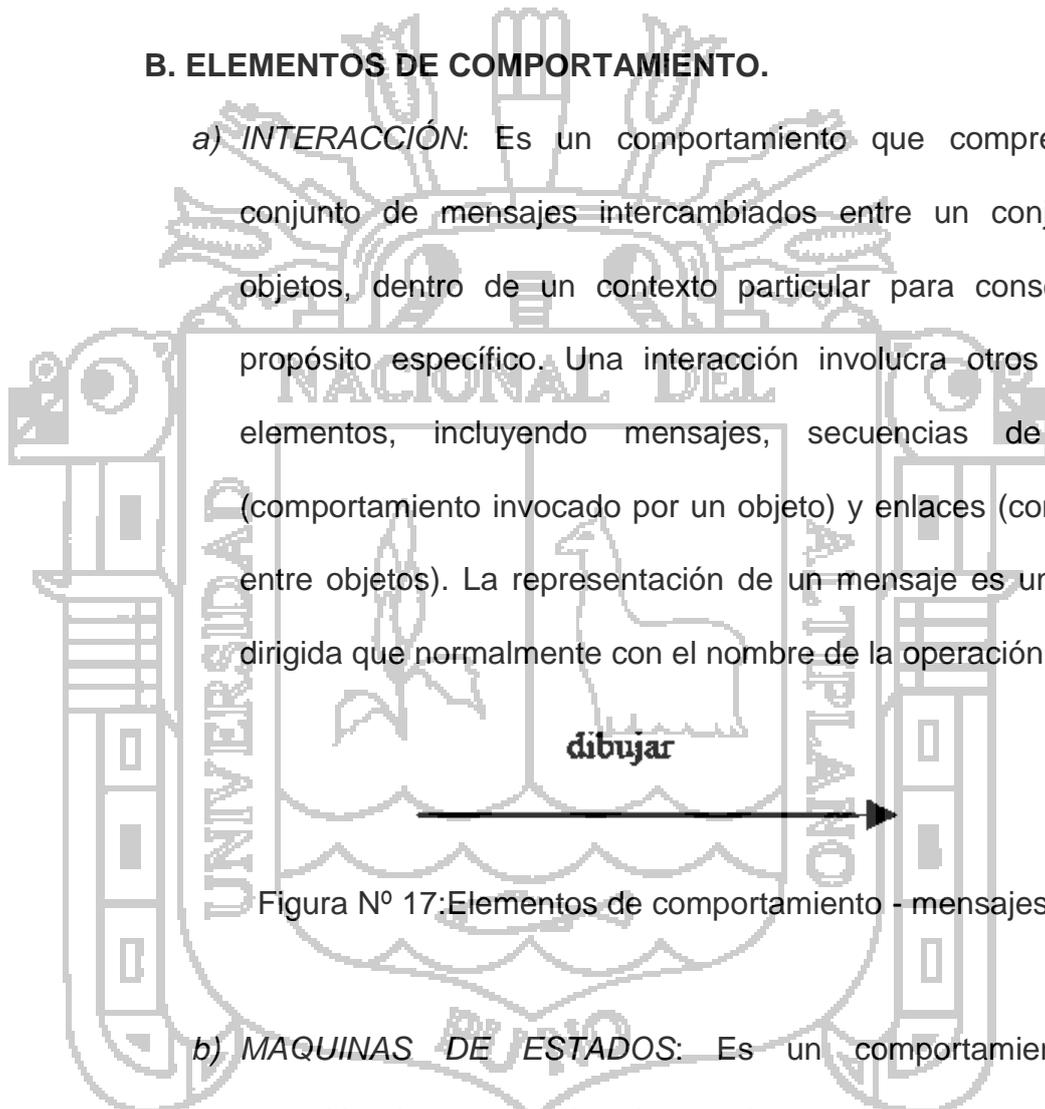


Figura N° 17: Elementos de comportamiento - mensajes.

b) **MAQUINAS DE ESTADOS:** Es un comportamiento que especifica las secuencias de estados por las que van pasando los objetos o las interacciones durante su vida en respuesta a eventos, junto con las respuestas a esos eventos. Una máquina de estados involucra otros elementos como son estados,

transiciones (flujo de un estado a otro), eventos (que disparan una transición) y actividades (respuesta de una transición).



Figura N° 18: Elementos de comportamiento - estados.

C. ELEMENTOS DE AGRUPACIÓN

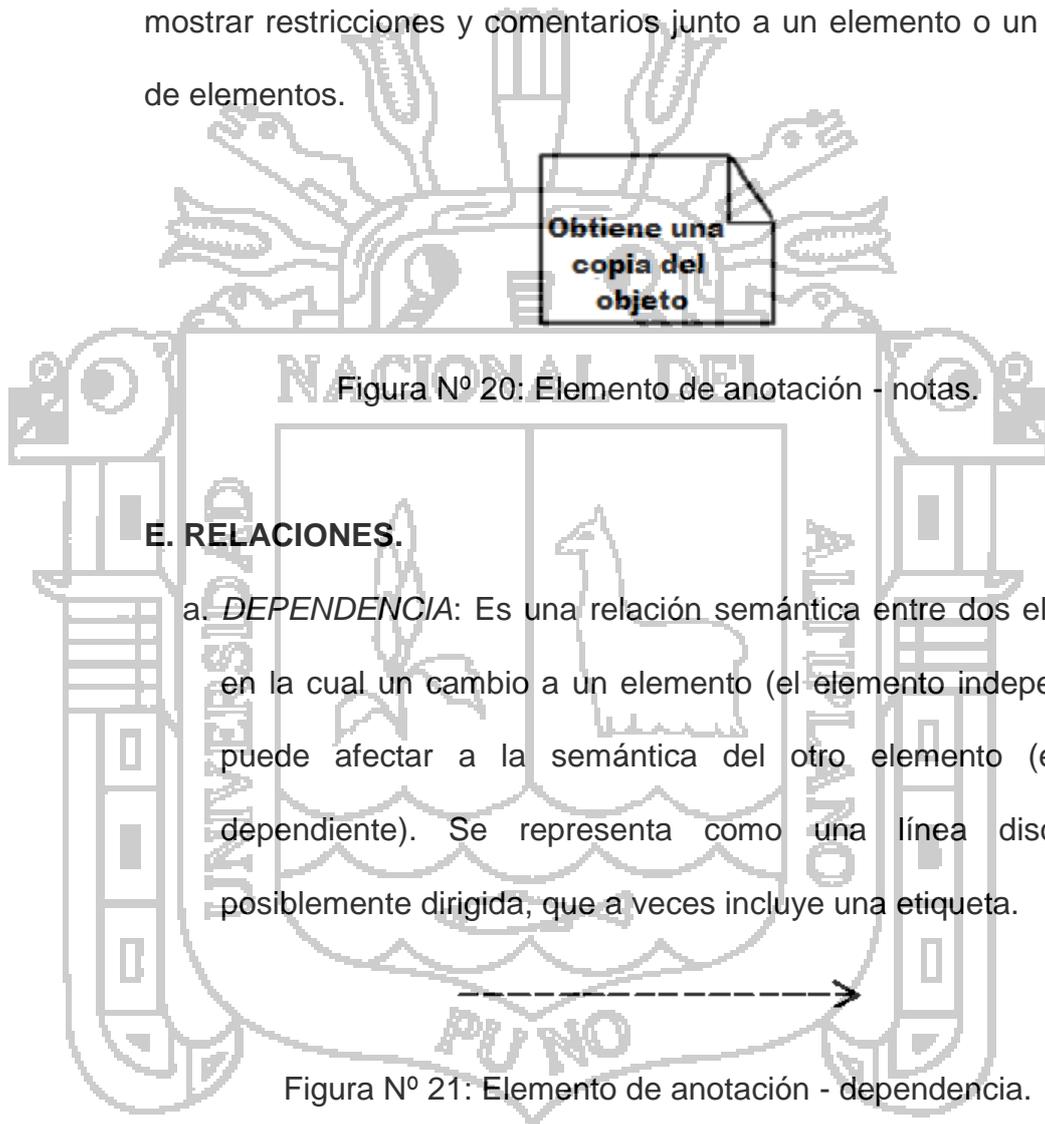
Forman la parte organizativa de los modelos UML. El principal elemento de agrupación es el paquete, que es un mecanismo de propósito general para organizar elementos en grupos. Los elementos estructurales, los elementos de comportamiento, incluso los propios elementos de agrupación se pueden incluir en un paquete. Un paquete es puramente conceptual (sólo existe en tiempo de desarrollo). Gráficamente se representa como una carpeta conteniendo normalmente su nombre y, a veces, su contenido. (SOMMERVILLE, 2011).



Figura N° 19: Elemento de agrupación - paquetes.

D. ELEMENTOS DE ANOTACIÓN.

Los elementos de anotación son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo. El tipo principal de anotación es la nota que simplemente es un símbolo para mostrar restricciones y comentarios junto a un elemento o un conjunto de elementos.



E. RELACIONES.

- a. *DEPENDENCIA*: Es una relación semántica entre dos elementos en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente). Se representa como una línea discontinua, posiblemente dirigida, que a veces incluye una etiqueta.

Figura N° 21: Elemento de anotación - dependencia.

F. DIAGRAMAS.

Los diagramas se utilizan para representar diferentes perspectivas de un sistema de forma que un diagrama es una proyección del mismo. UML proporciona un amplio conjunto de diagramas que normalmente

se usan en pequeños subconjuntos para poder representar las cinco vistas principales de la arquitectura de un sistema.(SOMMERVILLE, 2011).

a) *DIAGRAMAS DE CLASES*: Muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos y cubren la vista de diseño estática o la vista de procesos estática (sí incluyen clases activas).

b) *DIAGRAMAS DE OBJETOS*: Muestran un conjunto de objetos y sus relaciones, son como fotos instantáneas de los diagramas de clases y cubren la vista de diseño estática o la vista de procesos estática desde la perspectiva de casos reales o prototípicos.

c) *DIAGRAMAS DE CASOS DE USOS*: Muestran un conjunto de casos de uso y actores (tipo especial de clases) y sus relaciones. Cubren la vista estática de los casos de uso y son especialmente importantes para el modelado y organización del comportamiento.

d) *DIAGRAMAS DE SECUENCIA Y DE COLABORACIÓN*: Tanto los diagramas de secuencia como los diagramas de colaboración son un tipo de diagramas de interacción. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar unos objetos a otros. Cubren la vista dinámica del sistema.

Los diagramas de secuencia enfatizan el ordenamiento temporal de los mensajes mientras que los diagramas de colaboración muestran la organización estructural de los objetos que envían y reciben mensajes. Los diagramas de secuencia se pueden

convertir en diagramas de colaboración sin pérdida de información, lo mismo ocurren en sentido opuesto.

- e) *DIAGRAMAS DE ESTADOS*: Muestran una máquina de estados compuesta por estados, transiciones, eventos y actividades. Estos diagramas cubren la vista dinámica de un sistema y son muy importantes a la hora de modelar el comportamiento de una interfaz, clase o colaboración.
- f) *DIAGRAMAS DE ACTIVIDADES*: Son un tipo especial de diagramas de estados que se centra en mostrar el flujo de actividades dentro de un sistema. Los diagramas de actividades cubren la parte dinámica de un sistema y se utilizan para modelar el funcionamiento de un sistema resaltando el flujo de control entre objetos. (SOMMERVILLE, 2011).

2.2.11. Reingeniería de Software.

(KENDALL, 2005). Reingeniería del software se puede definir como: “modificación de un producto software, o de ciertos componentes, usando para el análisis del sistema existente técnicas de Ingeniería Inversa y, para la etapa de reconstrucción, herramientas de Ingeniería Directa, de tal manera que se oriente este cambio hacia mayores niveles de facilidad en cuanto a mantenimiento, reutilización, comprensión o evaluación.”

Cuando una aplicación lleva siendo usada años, es fácil que esta aplicación se vuelva inestable como fruto de las múltiples correcciones,

adaptaciones o mejoras que han podido surgir a lo largo del tiempo. Esto deriva en que cada vez que se pretende realizar un cambio se producen efectos colaterales inesperados y hasta de gravedad, por lo que se hace necesario, si se prevé que la aplicación seguirá siendo de utilidad, aplicar reingeniería a la misma.

Entre los beneficios de aplicar reingeniería a un producto existente se puede incluir:

- Pueden reducir los riesgos evolutivos de una organización.
- Puede ayudar a las organizaciones a recuperar sus inversiones en software.
- Puede hacer el software más fácilmente modificable
- Amplía las capacidades de las herramientas CASE
- Es un catalizador para la automatización del mantenimiento del software
- Puede actuar como catalizador para la aplicación de técnicas de inteligencia artificial para resolver problemas de reingeniería

La reingeniería del software involucra diferentes actividades como son:

- análisis de inventarios
- reestructuración de documentos
- ingeniería inversa
- reestructuración de programas y datos
- ingeniería directa

Con la finalidad de crear versiones de programas ya existentes que sean de mejor calidad y los mismos tengan una mayor facilidad de mantenimiento.

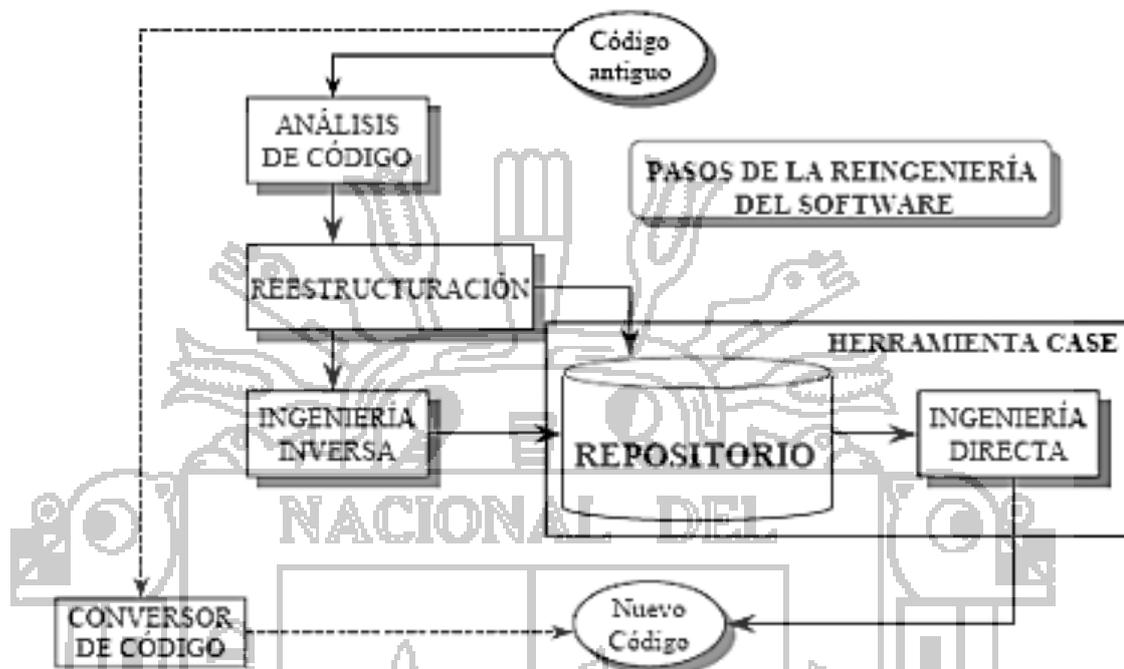


Figura N° 22: Representación de reingeniería de software

Fuente: (SOMMERVILLE. 2011, P. 158).

2.2.12. Norma ACID.

Acrónimo en inglés: Atomicity, Consistency, Isolation, and Durability.

Según Casillas (2005). Define que se utiliza en el ámbito de las bases de datos. A menudo se utilizan expresiones como "esta base de datos o este sistema cumple las propiedades ACID".

- Atomicity (atomicidad) significa que el sistema permite operaciones atómicas. Una operación atómica es aquella que si está formada

por operaciones más pequeñas, se consideran como un paquete indivisible. Deben ejecutarse todas correctamente, o en el caso de que alguna de ellas no pueda hacerlo, el efecto de las que ya se han ejecutado no debe hacerse notar, debe deshacerse, como si el conjunto de las operaciones no se hubieran realizado. La atomicidad está íntimamente ligada al concepto de transacción de los sistemas gestores de bases de datos. En un SGBD, cuando se indica que un conjunto de operaciones forman una transacción, o se ejecutan todas correctamente, o el SGBD deshacerá los cambios, como si la transacción nunca se hubiera iniciado. No obstante, atomicidad y transacción no son sinónimos. Mientras atomicidad es una propiedad, la transacción es el mecanismo que utilizan los SGBD para lograr la atomicidad.

- Consistency (consistencia), significa que se permite la definición de reglas de integridad de los datos, y que el sistema garantiza que ninguna operación puede dejar los datos en algún estado que deje de cumplir esas reglas. Por ejemplo, los SGBD actuales permiten definir claves ajenas, checks, asserts, etc... Son reglas lógicas que definen los diseñadores de bases de datos sobre un conjunto de datos, y que todo dato perteneciente al conjunto en todo momento debe cumplir. Si el sistema permite consistencia, significa que esas reglas se vigilan constantemente para todos los datos.
- Isolation (aislamiento). Esta propiedad indica que las operaciones que se realicen sobre un conjunto de datos serán independientes de cualquiera otras operaciones que se realicen sobre el mismo

conjunto de datos. En otras palabras, si una operación atómica intenta actuar sobre un conjunto de datos alterandolos de alguna manera mientras simultáneamente otra operación atómica intenta actuar sobre el mismo conjunto de datos, el sistema nos garantiza que no va a dejar que se interfieran entre ellas. El sistema escogerá el orden correcto de ejecución para que ambas se ejecuten como si la otra no existiera. Además, el sistema no dejará que durante una operación atómica se pueda acceder a resultados intermedios de la operación; los datos no reflejarán su nuevo estado hasta estar finalizada correctamente la operación atómica.

- Durability (durabilidad). Significa que en el mismo momento en que una operación ha terminado satisfactoriamente y el sistema informa de ello, sus efectos quedan ya registrados permanentemente. El sistema garantiza que incluso ante un fallo del sistema, los efectos de la operación quedarán ya registrados. Esto, que puede parecer obvio, no siempre ocurre. Hay muchos sistemas que informan del éxito de una operación, pero los resultados no han quedado aún registrados correctamente. (Por ejemplo, si copias un fichero mp3 a tu reproductor USB, y justo en el instante en el que el sistema te dice que ya lo ha copiado retiras el reproductor del puerto USB tienes muchas probabilidades de que el fichero no se haya grabado correctamente, debido a la utilización de una caché de escritura intermedia que utiliza el sistema operativo. Ese sistema no cumple la durabilidad).

2.2.13. Administración de Recursos Humanos.

La administración de los recursos humanos (ARH) es el conjunto de políticas y prácticas necesarias para dirigir los aspectos de los cargos gerenciales relacionados con las personas o recursos humanos, incluidos reclutamiento, selección, capacitación, recompensas y evaluación del desempeño. La ARH es una función administrativa dedicada a la adquisición, entrenamiento, evaluación y remuneración de los empleados. En cierto sentido todos los gerentes son gerentes de personas porque están involucrados en actividades como reclutamiento, entrevistas, selección y entrenamiento. La ARH es un conjunto de decisiones integradas sobre las relaciones de empleo que influyen en la eficacia de los empleados y las organizaciones. (ARIAS, LOPEZ, & HEREDIA, 1999)

La ARH es una función de la organización relacionada con la provisión, el entrenamiento, el desarrollo, la motivación y el mantenimiento de los empleados” Durante muchos años existió la creencia generalizada de que el obstáculo para el desarrollo de la industria era el capital. Sin embargo, la incapacidad de una empresa para reclutar y mantener una buena fuerza laboral es el principal obstáculo para la producción.

La ARH debe contribuir a la eficacia de la empresa a través de los siguientes medios:

- a. Ayudar a la organización a alcanzar sus objetivos y realizar su misión.

- b. Proporcionar competitividad a la organización.
- c. Suministrar a la organización empleados bien entrenados y motivados.
- d. Permitir la autorrealización y satisfacción de los empleados en el trabajo.
- e. Desarrollar y mantener la calidad de vida en el trabajo.
- f. Administrar el cambio.
- g. Establecer políticas éticas y desarrollar comportamientos socialmente responsables.

El aporte de ARIAS GALICIA, L.& HEREDIA ESPINOSA V., es: “La Administración de Recursos Humanos consiste en la planeación, organización, desarrollo y coordinación, así como también control de técnicas, capaces de promover el desempeño eficiente del personal, a la vez que la organización representa el medio que permite a las personas que colaboran en ella alcanzar los objetivos individuales relacionados directa o indirectamente con el trabajo”.

Al respecto WILLIAM, B. & WERTHER, KEITH, D., aporta “Significa conquistar y mantener las personas en la organización, trabajando y dando el máximo de sí, con una actitud positiva y favorable. Representa todas aquellas cosas que hacen que el personal permanezca en la organización”

La interpretación de FRENCH, WEDELL, L., es la siguiente “En la actualidad las técnicas de selección del personal tiene que ser más subjetivas y más afinadas, determinando los requerimientos de los recursos humanos, acrecentando las fuentes más efectivas que permitan allegarse a los candidatos idóneos, evaluando la potencialidad física y mental d los solicitantes, así como su aptitud para el trabajo, utilizando para ello una serie de técnicas, como la entrevista, las pruebas psicosométricas y los exámenes médicos”

Autores como: FRENCH, W. L. & WERTHER W. B. y KEITH DAVIS y ARIAS GALICIAL. FERNANDO y HEREDIA ESPINOSA VÍCTOR, recomiendan a que el proceso de selección del personal debe ser más objetiva, considerando los requerimientos de la organización para luego incrementar el conocimiento, capacidad del personal contratado y mencionan varias técnicas que son factibles en el desenvolvimiento del personal.

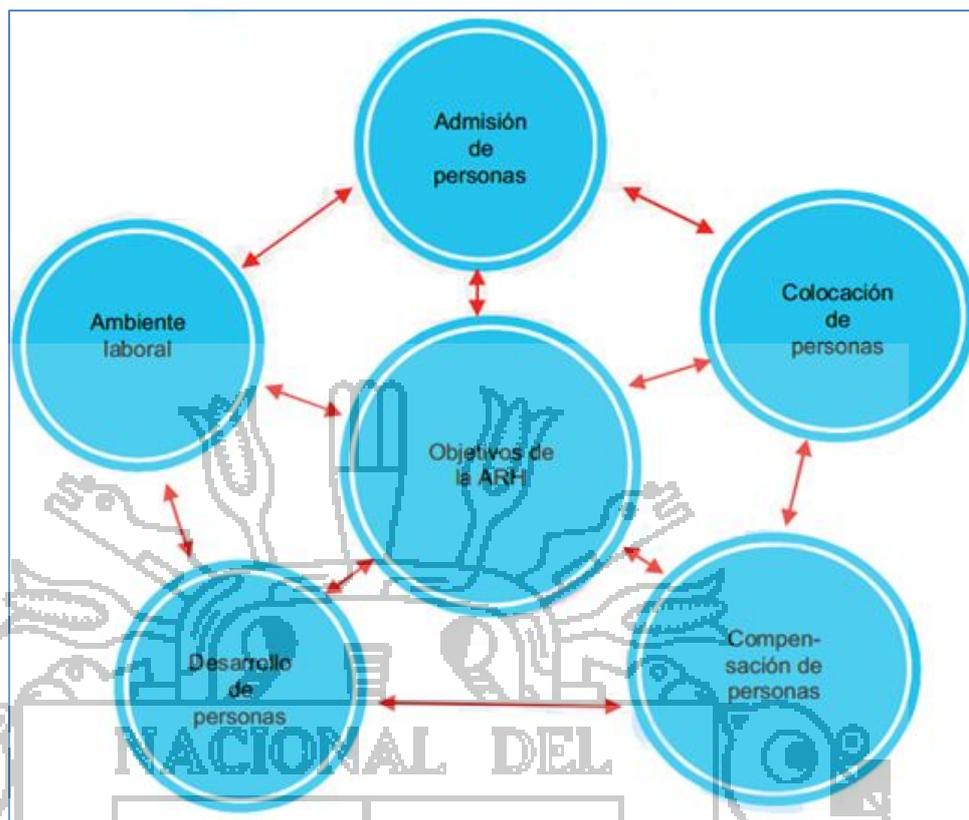


Figura N° 23: Objetivos de la administración de recursos humanos.

Fuente: (FRENCH & WERTHER, 2004)

La planeación estratégica de RH debe ser parte integral de la planeación estratégica de la organización. Ackoff, (2005) señala tres tipos de filosofía de planeación estratégica:

- a. Planeación conservadora. Orientada a hacia la estabilidad y mantenimiento de la situación existente.
- b. Planeación optimizadora, orientada hacia la adaptabilidad e innovación de la organización.
- c. Planeación prospectiva. Orientada hacia la contingencia y el futuro de la organización.

En la actualidad la administración de recursos humanos cuenta con módulos, para su correcto funcionamiento, estos son.

- 1) Modulo Planilla (planilla, boleta de pago ambos según formato del ministerio de trabajo)
- 2) Modulo Asistencia (asistencia gestión de contratos, prestamos, vacaciones, permisos, descansos médicos)
- 3) Modulo Escalafón (legajo de persona, ficha única de personal)
- 4) Modulo Interfaces (planilla electrónica, PDT, AFP Net, T-registro, CTS y bancos)
- 5) Modulo Reportes (empleados, planillas, asistencia, impuestos, presupuestos)

2.2.14. Municipalidad Provincial Chucuito Juli.

La Municipalidad Provincial de Chucuito Juli, es un ente de gobierno local, emana la voluntad popular. Es una persona Jurídica de derecho público con autonomía política, económica y administrativa en los asuntos de su competencia.

2.2.14.1. Visión.

En el año 2015, la Municipalidad Provincial de Chucuito Juli, es una institución líder en gestión democrática participativa, planificada y transparente; con capacidad de organización para la promoción del desarrollo local sostenible y la prestación de servicios de calidad, a través de su recurso humano innovador y con identidad institucional.(MPCHJ, 2014).

2.2.14.2. Misión.

La Municipalidad de la Provincia de Chucuito Juli, es un órgano de Gobierno Local y sus acciones se orientan por el Plan de Desarrollo Concertado, articulando esfuerzos con las instituciones del sector público y privado para la promoción del desarrollo local y la prestación de servicios de calidad, con la que se busca mejorar el nivel de vida de la población con: equidad, inclusión, manejo responsable y transparente de los recursos públicos.

2.2.14.3. Objetivos Generales.

- Desarrollar capacidades orientadas al fortalecimiento institucional para estar en condiciones de elaborar, ejecutar y evaluar el plan de desarrollo y presupuesto participativo basados en la activa participación de la población organizada del ámbito urbano y rural, en concertación y en alianza estratégica con otras instituciones públicas y privadas, comprometidas con el desarrollo local.
- Potenciar las capacidades internas e invertir con mayor eficiencia los recursos disponibles del sector público mejorando sus procesos, sistemas administrativos sistema financiero y el desempeño laboral de sus trabajadores y funcionarios.

2.2.14.4. Objetivos Específicos.

- Promover la gobernabilidad y la participación ciudadana para la gestión integral de desarrollo.
- Promover la gestión eficiente y eficaz de la Municipalidad.
- Reducción de la violencia familiar.
- Mejorar y mantener la ciudad limpia y saludable.
- Promover un programa integral de Vialidad y Organización del tránsito vehicular.
- Actualizar el Plan de Desarrollo Concertado Provincial, teniendo en cuenta los Planes de Desarrollo de los Centros Poblados.
- Promover la Demarcación Territorial Provincial, Distrital y Centro Poblado.
- Elevar el nivel de compromiso de los servidores y funcionarios de la Municipalidad Provincial respecto de las políticas institucionales, planes, proceso de presupuesto participativo, simplificación administrativa, innovación en la prestación de servicios y proyectos de la Municipalidad.
- Regular y mejorar el crecimiento urbano de la ciudad y el ordenamiento del transporte, mediante acciones de acondicionamiento territorial, así como la ampliación y mejoramiento de la infraestructura vial existente.
- Mejorar la percepción de seguridad de los vecinos.

2.2.14.5. Organigrama.

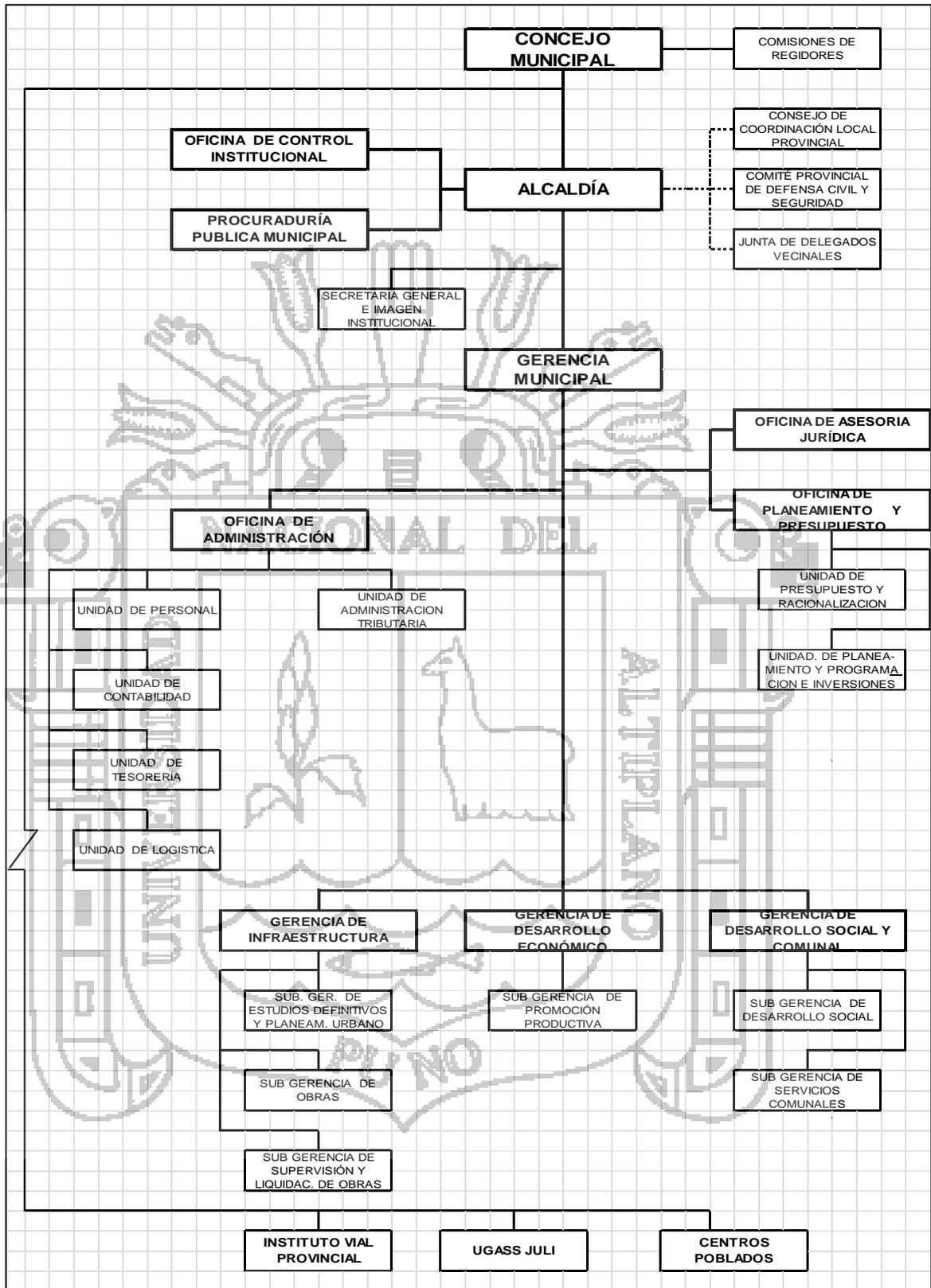


Figura N° 24: Organigrama de la Municipalidad Provincial de Chucuito Juli.

Fuente: MPCHJ. (2013).

2.2.14.6. Unidad de Personal.

Dentro de las funciones de la que cumple la unidad de personal se tiene.

- a. Realizar los procedimientos de reclutamiento, selección, calificación, evaluación, control, promoción y rotación del personal de acuerdo lo dispuesto por la Alta Dirección en concordancia con las disposiciones legales vigentes.
- b. Coordinar, promover y ejecutar programas de capacitación de personal en forma selectiva, mejorando el nivel de formación y prestación de servicios dentro de la Municipalidad.
- c. Organizar y actualizar la información de los legajos y escalafón del personal.
- d. Ejecutar la política de remuneraciones e incentivos de acuerdo a los lineamientos aprobados por la autoridad.
- e. Conducir y supervisar las actividades relacionadas con la emisión de la planilla de remuneraciones y liquidación de los beneficios sociales en los términos que señala la Ley.
- f. Velar por la correcta aplicación y cumplimiento de las disposiciones legales vigentes sobre derechos y beneficios del personal.
- g. Normar, coordinar, controlar y supervisar el cumplimiento del Reglamento de Control de Asistencia y Permanencia de los trabajadores municipales.

- h. Participar como miembro de la comisión de procesos administrativos de la Municipalidad.
- i. Diseñar y elaborar proyectos de directivas y normas técnicas que mejoren y optimicen el proceso técnico de personal.
- j. Desarrollar los procesos técnicos de pensiones conforme a Ley.
- k. Evaluar y ejecutar las normas y procedimientos establecidos para el sistema de personal de la Municipalidad.
- l. Preparar informes técnicos sobre las actividades a su cargo.
- m. Mantener estrecha comunicación con las entidades recaudadoras, como AFPs, SNP, ESSALUD, entre otras, para coordinaciones varias sobre el personal municipal.
- n. Coordinar con Tesorería el pago de obligaciones sociales a las entidades correspondientes.
- o. Estudiar y proponer el mejoramiento de las normas y procedimientos relativos al control de asistencia y permanencia del personal.
- p. Visar el informe diario de faltas y tardanzas del personal de la Municipalidad.
- q. Programar y tramitar la aprobación del Rol Anual de Vacaciones del personal.

- r. Velar por el cumplimiento de las normas técnicas de control y las disposiciones legales vigentes que regulan las actividades del Personal.
- s. Proponer al Jefe de la Oficina de Administración las directivas que sean necesarias para su aprobación en el cumplimiento de sus funciones.
- t. Requerir oportunamente la presentación de la Declaración Jurada de Bienes y Rentas y documentos personales de los funcionarios de la Municipalidad.
- u. Participar en la elaboración del CAP, PAP, ROF y MOF de la Municipalidad, en coordinación con el Jefe de la Unidad de Presupuesto y Racionalización.
- v. Conocer todas las disposiciones legales inherentes a la gestión municipal, como: Ley Orgánica de Municipalidades, ley de procedimientos administrativos, ley de la carrera administrativa, Ordenanzas, Decretos y todos los documentos de gestión.
- w. Exigir el cumplimiento del MOF a todo el personal de la Municipalidad, bajo responsabilidad
- x. Otras que le asigne el Jefe de la Oficina de Administración y; las que le corresponda de acuerdo al nivel y naturaleza del cargo.

2.3. DEFINICIÓN DE TÉRMINOS BÁSICOS

2.3.1. Sistema.

Se trata de la coordinación de diversos subconjuntos con un objetivo común. Estos subconjuntos son los recursos con los que se cuenta: personal, presupuesto, tecnología, locales, información. De la dotación adecuada de estos recursos, su organización y su relación interdependiente, a través de los procesos y tareas efectuados, resultan los servicios y productos que la biblioteca pone a disposición de sus usuarios o clientes.

2.3.2. Administración de Recursos Humanos.

La administración de recursos humanos tiene como objetivo el mejoramiento continuo del desempeño y las aportaciones del personal a la organización, en el marco de una actividad ética y socialmente responsable.

2.3.3. Optimización.

Optimización es la acción y efecto de optimizar. Este verbo hace referencia a buscar la mejor manera de realizar una actividad. El término se utiliza mucho en el ámbito de la informática.

La optimización de software busca adaptar los programas informáticos para que realicen sus tareas de la forma más eficiente posible. Virtualmente, existen infinitas maneras de desarrollar una misma aplicación, y uno de los factores más influyentes a la hora de crear el diseño es la arquitectura de hardware con la cual se desea trabajar. En

pocas palabras, conseguir el mejor rendimiento en una plataforma enfocada en el tipo y la cantidad memoria es muy diferente a hacerlo en una cuyo fuerte es la velocidad de los procesadores.

2.3.4. Gestión.

Gestión hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación o de un anhelo cualquiera. La noción de gestión, por lo tanto, se extiende hacia el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto.

2.3.5. Procesos.

Proceso es el conjunto de actividades o tareas, mutuamente relacionadas entre sí que admite elementos de entrada durante su desarrollo ya sea al inicio o a lo largo del mismo, los cuales se administran, regulan o autorregulan bajo modelos de gestión particulares para obtener elementos de salida o resultados esperados. Las entradas al proceso pueden ser iniciales o intermedias. Asimismo, los resultados o salidas a lo largo del proceso pueden ser intermedios o finales. La presencia e interacción de los elementos que lo componen conforman un sistema de trabajo, al cual puede denominarse “Sistema de gestión del proceso”.

2.3.6. Base de Datos Orientado a Objetos.

Una base de datos orientada a objetos es una base de datos inteligente soporta el paradigma orientado a objetos almacenando métodos y datos, y no solamente datos. Está diseñada para ser eficaz, desde el punto de vista físico, para almacenar objetos complejos. Evite el acceso a los datos; esto gracias a los métodos almacenados en ella. Es más segura, ya que no permite tener acceso a los datos (objetos); esto debido a que para poder entrar se tiene que hacer por los métodos que haya utilizado el programador. Estructura de una BD OO

2.3.7. DB4O.

B4O es un novedoso motor de base de datos orientada a objetos. Sus siglas se corresponden con la expresión "DataBase 4 (for) Objects", que a su vez es el nombre de la compañía que lo desarrolla: db4objects, Inc. Es una base de datos orientada a objetos completamente nativa de alto rendimiento. Esta base de datos fue desarrollada en el corazón de Silicon Valley. Esta base de datos es compacta y válida como Base de datos embebida en la aplicación. El sistema gestor de base de datos orientado a objetos soporta aplicaciones Standalone así como Cliente/Servidor (Aplicaciones distribuidas). Este sistema está disponible para entornos Java o .Net.

2.3.8. Java.

Java es un lenguaje de programación por objetos creado por Sun Microsystems, Inc. que permite crear programas que funcionan en

cualquier tipo de ordenador y sistema operativo. Se usa el Java para crear programas especiales denominados applets, que pueden ser incorporados en páginas web para hacerlas interactivas. Los apliques Java requieren que el navegador utilizado sea compatible con Java.

2.3.9. JavaBeans.

Los JavaBeans definen un modelo de componentes potente y simple para Java. El objetivo de los JavaBeans es proporcionar unas unidades reutilizables y autosuficientes que los desarrolladores puedan manipular mediante programación o que las herramientas de construcción puedan manipular de forma visual.

Los JavaBeans pueden ser controles de interfaz gráfica de usuario o pueden no tener representación visual alguna. Los controles de interfaz gráfica de usuario en Java habitualmente son JavaBeans para que las herramientas de construcción puedan manipularlos. En J2EE, los JavaBeans sencillos se utilizan habitualmente desde JSP, donde proporcionan una separación de la presentación en HTML y el código Java, que lo contienen los JavaBeans.

2.3.10. JSP.

JavaServerPages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML entre otros tipos de documentos. JSP es similar a PHP pero usa el lenguaje de programación Java.

2.3.11. JSF.

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServerPages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario).

2.3.12. Framework.

Un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

2.3.13. Hibernate.

Es una herramienta ORM (ObjectRelationalMapping), sirve de puente entre un mundo orientado a objetos y registros de una base de datos.

2.3.14. IoC.

Spring se basa en IoC. IoC es lo que nosotros conocemos como El Principio de Inversión de Dependencia, "Inversion of Control" (IoC) consiste en:

- Un Contenedor que maneja objetos por ti.
- El contenedor generalmente controla la creación de estos objetos. Por decirlo de alguna manera, el contenedor hace los "new" de las clases java para que no los realices tú.
- El contenedor resuelve dependencias entre los objetos que contiene.

2.3.15. Servlet.

El servlet es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, éstos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web. Este tipo de servlets son la contraparte Java de otras tecnologías de contenido dinámico Web, como PHP y ASP.NET.

La palabra servlet deriva de otra anterior, applet, que se refiere a pequeños programas que se ejecutan en el contexto de un navegador web. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

2.3.16. JAR.

Un archivo JAR (Java Archives) es simplemente un archivo comprimido que incluye una estructura de directorios con Clases, lo anterior permite:

- Distribuir/Utilizar Clases de una manera eficiente a través de un solo archivo.
- Declarar dichas Clases de una manera más eficiente en la variable CLASSPATH.

2.3.17. Planilla de Remuneraciones.

Es el documento llevado a través de medios electrónicos, presentado mensualmente a través del medio informático desarrollado por la SUNAT, en el que se encuentra registrada la información de los trabajadores, pensionistas, prestadores de servicios, prestador de servicios -modalidad formativa, personal de terceros y derechohabientes.

Las planillas de pago son un registro contable. Brindan elementos que permiten demostrar, de manera transparente, ante la autoridad

competente, la relación laboral del trabajador con mi empresa, su remuneración y los demás beneficios que se le pagan. Estos registros pueden ser llevados en libros, hojas sueltas o micro formas.

2.3.18. Planilla Electrónica.

Es el documento llevado a través de los medios informáticos desarrollados por la SUNAT, en el que se encuentra la información de los empleadores, trabajadores, pensionistas, prestadores de servicios, personal en formación -modalidad formativa laboral y otros, personal de terceros y derechohabientes.

2.3.19. AFP.

AFP es la sigla de Administradora Privada de Fondos de Pensiones. Es en principio una sociedad anónima, de duración indefinida y con el único objeto de administrar los fondos de pensión de un trabajador.

2.3.20. T-registro.

Es el Registro de Información Laboral de los empleadores, trabajadores, pensionistas, prestadores de servicios, personal en formación, personal de terceros y derechohabientes. Comprende información laboral, de seguridad social y otros datos sobre el tipo de ingresos de los sujetos registrados.

2.3.21. PLAME.

Se denomina PLAME a la Planilla Mensual de Pagos, segundo componente de la Planilla Electrónica, que comprende información mensual de los ingresos de los sujetos inscritos en el Registro de Información Laboral (T-REGISTRO), así como de los Prestadores de Servicios que obtengan rentas de 4ta Categoría; los descuentos, los días laborados y no laborados, horas ordinarias y en sobretiempo del trabajador; así como información correspondiente a la base de cálculo y la determinación de los conceptos tributarios y no tributarios cuya recaudación le haya sido encargada a la SUNAT. La PLAME se elabora obligatoriamente a partir de la información consignada en el T-REGISTRO.

2.3.22. PDT.

El Programa de Declaración Telemática-PDT, es un sistema informático desarrollado por la Superintendencia Nacional de Administración Tributaria (SUNAT) con la finalidad de facilitar la elaboración de las declaraciones juradas bajo condiciones de seguridad del registro de la información.



MATERIALES Y MÉTODOS

3.1. TRABAJO EXPERIMENTAL

3.1.1. Metodología de la Investigación.

Se presentan una definición de las metodologías de desarrollo muy clara, destacando sus principales componentes, fases, herramientas y técnicas.

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.

Esta tesis, de acuerdo con la pregunta de investigación, planteamiento del problema se resuelve con la hipótesis, los objetivos y la pregunta de investigación, se enmarca dentro del enfoque cuantitativo correlacional, es decir, una investigación que pretende analizar la relación entre las variables: sistema de recursos humanos utilizando tecnologías de base de datos BD4O. y la gestión de procesos. El lugar donde se llevará la investigación será en la Municipalidad Provincial Chucuito Juli, específicamente en la Unidad de Personal de dicha institución.

3.1.1.1. Criterios para la Selección de la Metodología.

En función del grupo de trabajo o de diseño, se consideran los siguientes criterios en función de los conocimientos que el equipo de desarrollo tenga de las metodologías a evaluar.

Estos criterios son:

- Grado de conocimiento
- Soporte orientado a objetos
- Adaptable a cambios
- Basado en casos de uso
- Posee documentación adecuada
- Facilita la integración entre las etapas de desarrollo
- Relación con UML
- Permite desarrollo software sobre cualquier tecnología

Criterio	%	RUP	MSF	RAD	XP	Total
Grado de conocimiento	20	15	10	10	10	45
Soporte orientado a objetos	10	10	10	10	10	40
Adaptable a cambios	15	10	15	10	15	50
Basado en casos de uso	10	10	5	10	5	30
Posee documentación adecuada	15	15	15	15	10	55
Facilita la integración entre las etapas de desarrollo	10	10	10	10	10	50
Relación con UML	10	10	8	8	8	34
Permite desarrollo software sobre cualquier tecnología	10	10	10	10	10	40
Total	100	90	83	83	78	

Tabla N° 02: Comparación de metodologías de desarrollo de software.

Dónde:

- RUP, Rational Unified Process
- MSF, Microsoft Solution Framework
- RAD, Rapid Application Development

- XP, Extreme Programming

En función de los conocimientos que el equipo tenga, se establecen los pesos para cada criterio. Por ejemplo:

- 20% para el Grado de conocimiento
- 15% para Adaptable a cambios y Posee documentación adecuada
- 10% para el resto de criterios

INTERPRETACIÓN: si se va a realizar un trabajo de tesis en base a conocimientos se debe tener en cuenta los diferentes criterios mencionados anteriormente, al observar la calificación del cuadro anterior se concluye que se debe tomar la metodología RUP, por tener la mayor puntuación con 90 puntos, esto implica que se debe aplicar la metodología RUP en este trabajo.

Por lo tanto tomaremos a este proceso de desarrollo de software para la el desarrollo de nuestro sistema basado en la plataforma JEE y sobre un motor de base de datos orientado a objetos, por ser el mas adecuado según la información en la Tabla 02.

3.1.2. Diseño de la Investigación.

El diseño de la investigación es cuasi experimental. Los diseños cuasi experimentales manipulan por lo menos una variable independiente

para observar su efecto y relación con una o más variables dependientes.

El grupo experimental estuvo conformado por los encargados del personal que laboran en la Municipalidad Provincial Chucuito Juli.

La representación es la siguiente:

$$G_1: A_1 X A_2$$

Donde:

G_1 : Grupo experimental

X : Tratamiento de con el sistema de recursos humanos

A_1 : Test antes del experimento

A_2 : Test despues del experimentos

Este diseño con grupo experimental permitió la comparación de resultados pretest y posttest, con un alto nivel de probabilidad, que el sistema de recursos humanos (variable independiente) ha sido factor determinante en el control, seguimiento y consecuentemente en la mejora de gestión de procesos en el la Municipalidad Provincial Chucuito Juli, específicamente en la Unidad de Personal.

3.1.3. Población y Muestra.

A. Población

El personal de planta que labora dentro de la institución es total de 144 empleados, bajo las distintas modalidades de contrato como: personal nombrado, permanente, Contrato fijo y Contrato Administrativo de Servicios (CAS), para todo este personal se debe realizar; control de asistencia, control de permisos, control de vacaciones, planilla de remuneraciones, boleta de pago, control de AFP, control de escalafón, bonificaciones. También se debe considerar al personal que cuenta con un vínculo laboral con la entidad como el personal de las diferentes obras ya sean residentes, supervisores, asistentes, maestros de obra, operarios, almaceneros, oficiales y peones. A este personal se tiene que elabora lo siguiente; planilla de remuneraciones, boleta de pago, control de AFP, bonificaciones.

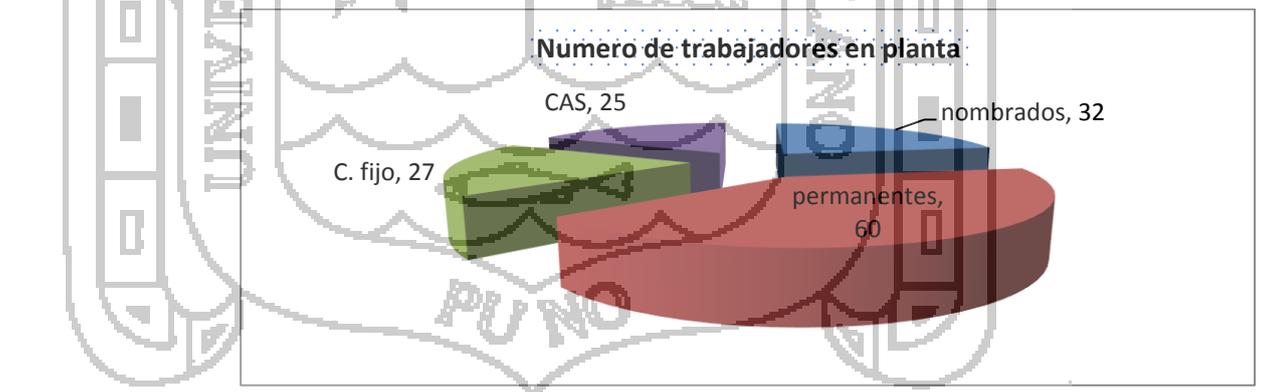


Figura N° 25: Número de trabajadores en planta.

Fuente: Unidad de Personal - MPCHJ.

El personal de obra asciende en un promedio de 350 trabajadores mensual (como máximo 500 y mínimo 100), dependiendo de cada mes y de las aperturas de nuevas obras.

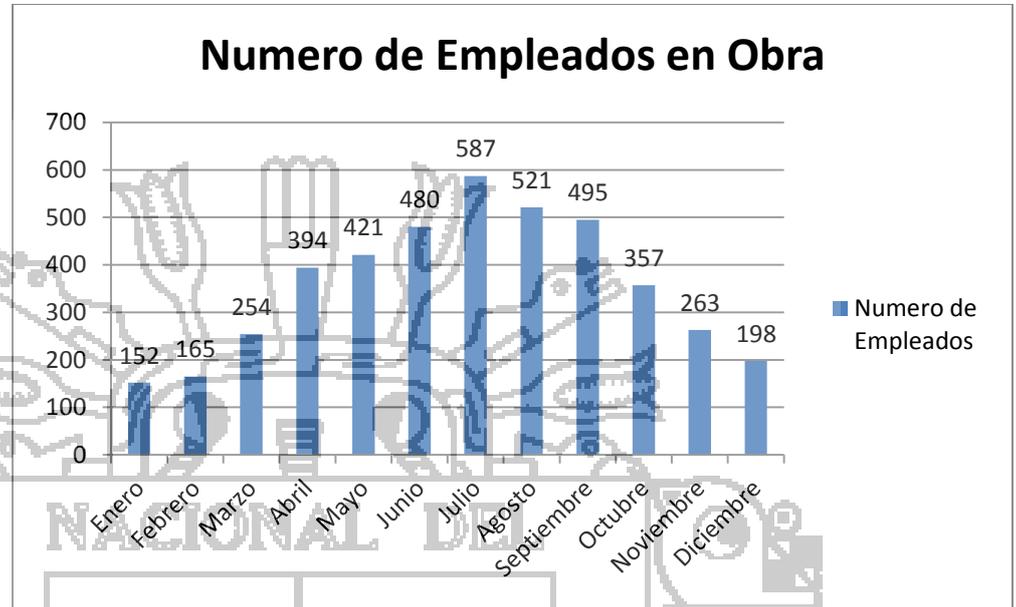


Figura N° 26: Número de trabajadores en obra

Fuente: Unidad de Personal - MPCHJ.

Se tendría un total de 494 empleados promedio que están vinculadas laboralmente a la Municipalidad, para todos estos de debe elaborar sus planillas de remuneraciones, boletas de pago entre otros.

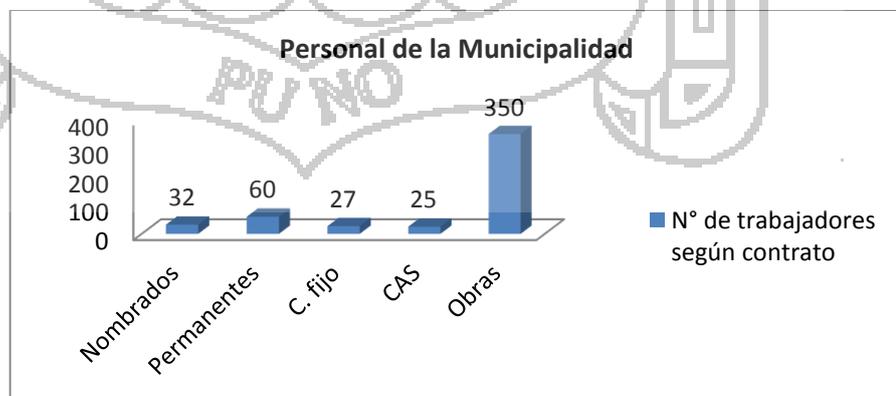


Figura: N° 27: Número de trabajadores en la Municipalidad.

Fuente: Unidad de Personal - MPCHJ.

B. Muestra

La selección de la muestra no probabilística o dirigida se obtuvo por la técnica de muestreo por conveniencia para el personal administrativo que serán los usuarios directos, siendo el tamaño de muestra (MORALES, 2012):

$$n = N-1,$$

Dónde:

N= 494; Tamaño de la Población.

n= 493; Tamaño de Muestra.

e=0.05 (error de tolerado del 5% y 95% de Nivel de Confianza)

3.1.4. Sistemas de Variables.

A. Variables Dependientes

Gestión de procesos de la Municipalidad Provincial Chucuito Juli.

B. Variables Independientes

Sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB40.

3.1.5. Operacionalización de Variables.

Variables	Dimensión	Indicadores	Índice	Escala	Instrumentos
Dependiente		Eficiencia	Se tiene el buen desempeño de sus procesos	Muy de acuerdo Algo de acuerdo Ni de acuerdo ni en desacuerdo Algo en desacuerdo Muy en desacuerdo	Encuestas
		Tiempo	Se realizan en tiempos reducidos	Segundos Minutos Horas Días	
	Gestión de Procesos	Costo	Se reducen los costos en los procesos	Muy de acuerdo Algo de acuerdo Ni de acuerdo ni en desacuerdo Algo en desacuerdo Muy en desacuerdo	
		Control	Se tiene el mejor control de los procesos	Muy de acuerdo Algo de acuerdo Ni de acuerdo ni en desacuerdo Algo en desacuerdo Muy en desacuerdo	
		Escalabilidad	Capacidad de adaptarse a las circunstancias cambiantes. Cantidad de registros. Cantidad de usuarios. Cantidad de módulos.	Muy de acuerdo Algo de acuerdo Ni de acuerdo ni en desacuerdo Algo en desacuerdo Muy en desacuerdo	
Independiente	Sistema basado en la plataforma JEE sobre un motor de base de datos orientado a objetos DB4O.	Usabilidad	Facilidad de utilizar la interfaz	Muy de acuerdo Algo de acuerdo Ni de acuerdo ni en desacuerdo Algo en desacuerdo Muy en desacuerdo	Test y/o pruebas
		Confiabilidad	Precisión con la que el sistema proporciona confianza, tolerancia a fallas	Muy de acuerdo Algo de acuerdo Ni de acuerdo ni en desacuerdo Algo en desacuerdo Muy en desacuerdo	

Tabla N° 03: Operacionalización de variables.

3.1.6. Métodos de Recopilación de Datos.

Para el desarrollo de la investigación se basa en análisis comparativos que permitan recolectar información cuantitativa y cualitativa. Mediante la evaluación de las herramientas y/o aplicaciones más reconocidas en la actualidad, identificando los servicios y los recursos tecnológicos requeridos para la implementación de Sistemas de Información. Para poder proponer un modelo como resultado de esta investigación.

Se utilizó tanto entrevistas y encuestas, realizadas al personal que labora en la unidad de personal y en la gerencia de infraestructura, las cuales estas involucradas en la elaboración de planillas, asistencia, permisos entre otros, para el personal que labora en la Municipalidad Provincial Chucuito Juli.

3.1.7. Métodos de Tratamientos de Datos.

Para el tratamiento de datos se realizó las siguientes tareas.

- Recopilación y tabulación de datos.
- Análisis y consistencia de datos.
- La interpretación de los datos y la validación de la hipótesis mediante la prueba de hipótesis.

3.2. MATERIAL EXPERIMENTAL

3.2.1. Metodología de Desarrollo del Sistema.

Las investigaciones vistas en el marco teórico permitieron establecer un marco de trabajo a seguir para el proceso de software y así cumplir con el

objetivo de la investigación. Para el modelo de análisis se usó el modelo del dominio del Proceso Unificado, modelo de datos; para el modelado del diseño se usó los artefactos del UML y el diseño de interfaz se desarrolló con la metodología de diseño orientados a objetos hipermmedia.

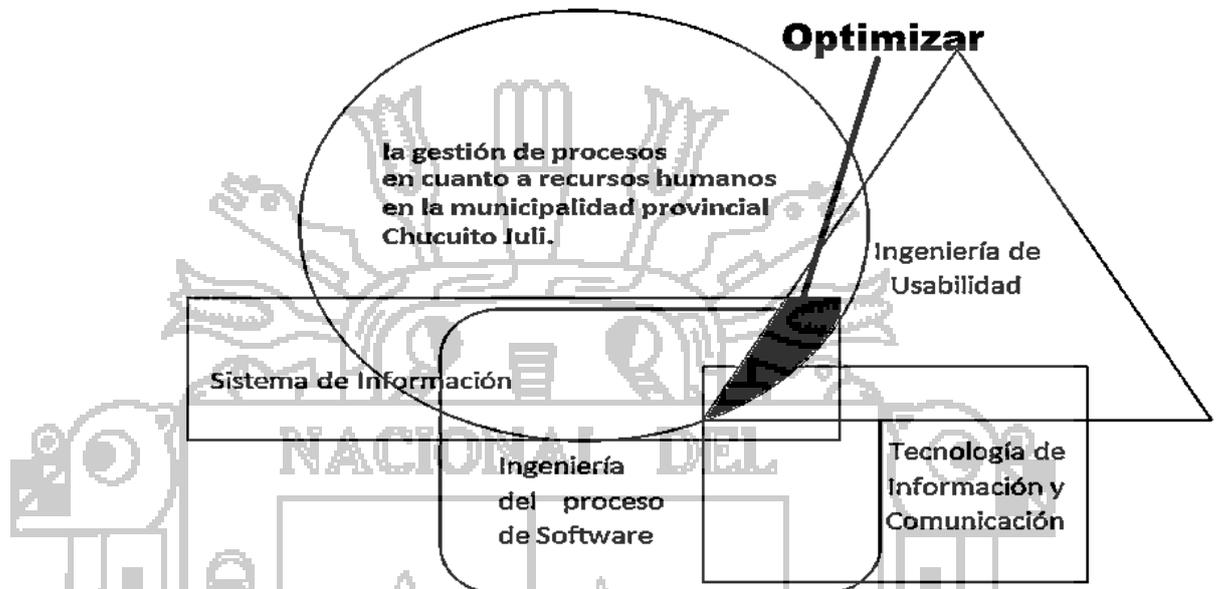


Figura N° 28: Modelo de desarrollo de software.

Fuente: Elaboración propia.

El material experimental estuvo constituido por: “Sistema de recursos humanos utilizando tecnologías de base de datos orientado a objetos BD4O, para la gestión de procesos en la Municipalidad Provincial Chucuito Juli”.

Para el desarrollo del Sistema se utilizó como

Servidor Web:

- Apache-Tomcat Web server 7.0.50

Base de datos:

- Db4o 8.0 (versión java)

Lenguaje de programación:

- Java EE
- JSP
- HTML

Análisis y Diseño:

- Rational Rose.
- MS Excel

Entorno de programación

- NetBeans 7.3.1

3.2.2. Análisis de Requisitos.

El análisis de requisitos permitió determinar a los objetos de dominio, las relaciones entre objetos que intervienen en el problema, los actores, los objetivos y responsabilidades principales que realizan. Para la siguiente etapa de análisis se utilizó el Modelo de Dominio del Proceso Unificado que viene a ser un subconjunto del proceso de negocios.(LARMAN, 2005).

Para el análisis de requisitos de la unidad de personal, se especifican dos tipos: requisitos funcionales y los requisitos no funcionales.

3.2.2.1. Requisitos Funcionales.

Identificador	RF001
Nombre	Configuraciones
Prioridad	ALTA
Descripción	Permite la configuración de fórmulas, parámetros y tipo de cambio que se realiza en el cálculo de planillas.
Observaciones	Los valores son: tipo de cambio, afp (según el monto de cada AFP), EsSalud (según la modalidad de contrato)

Tabla N° 04: Requisito funcional RF001.

Identificador	RF002
Nombre	Ficha del empleado
Prioridad	ALTA
Descripción	Permite agregar, modificar y eliminar la ficha del empleado este consta de datos personales, datos laborales.
Observaciones	<p>Los datos necesarios son:</p> <p>Datos del empleado</p> <ul style="list-style-type: none"> - Nombre(s). - Apellidos. - DNI - Dirección. - E-mail. - Teléfono. - Fecha de nacimiento - RUC. - AFP - CUSPP - Estado civil - Tipo de empleado. - Profesión. - Especialidad. - Otros. <p>Para el control de los datos laborales se debe implementar el almacenamiento de los archivos originales en formato digital (PDF),</p>

Tabla N° 05: Requisito funcional RF002.

Identificador	RF003
Nombre	Control del empleado
Prioridad	ALTA
Descripción	Permite el control de cada empleado en cuanto a sus contratos, permisos, descanso médico, papeletas de salida, vacaciones, sanciones entre otros para tener un historial del empleado.
Observaciones	Para el control de contratos de debe almacenar los archivos originales en formato digital (PDF), para la programación de vacaciones, permisos, descanso medico se debe almacenar las solicitudes en formato digital (PDF)

Tabla N° 06: Requisito funcional RF003.

Identificador	RF004
Nombre	Asistencia del personal
Prioridad	ALTA
Descripción	Registra la asistencia del empleado.
Observaciones	Al registrar la asistencia se debe tomar una imagen del momento en el que empleado marca su asistencia

Tabla N° 07: Requisito funcional RF004.

Identificador	RF005
Nombre	Generaciones de planillas
Prioridad	ALTA
Descripción	Permite la generación de planillas, boletas de pago, CTS, impuesto a la renta, gratificaciones.
Observaciones	La planilla de remuneraciones debe ser según formato establecida por el ministerio de trabajo.

Tabla N° 08: Requisito funcional RF005.

Identificador	RF006
Nombre	Declaraciones a entidades
Prioridad	ALTA
Descripción	Permite la declaraciones a entidades como: AFP NET, planilla electrónica, T-registro, PLAME, pensionistas y bancos
Observaciones	Estas declaraciones deben ser de acuerdo a los formatos establecidos por el ministerio de trabajo, entre otras entidades.

Tabla N° 09: Requisito funcional RF006.

3.2.2.2. Requisitos no Funcionales.

A. Requisitos de Rendimiento

El tiempo de respuesta de la aplicación a cada función solicitada por el usuario no debe ser superior a los 5 segundos. El tiempo de respuesta a los listados dependerá de la tecnología de impresión.

B. Seguridad

- El Sistema tendrá un acceso restringido.
- La seguridad de los datos será establecida por el sistema gestor de Base de Datos que se emplee.
- Para brindar seguridad al sistema, las técnicas utilizadas serán distintas por ejemplo la encriptación para el caso de las contraseñas de los usuarios
- También para evitar la pérdida de datos de la base de datos por diferentes circunstancias, se realizará semanalmente una copia de respaldo, esta funcionalidad será añadida en el sistema.

C. Disponibilidad

- Estará disponible el 60% del día de domingo a jueves.

D. Mantenibilidad

- El Grupo de desarrollo solamente podrá realizar la mantenibilidad del Sistema.

E. Portabilidad

- Está sujeta al traslado de otras plataformas o entornos.

OBSERVACIÓN:

Se debe registrar la fecha y hora de cualquier acción¹⁷:

- Registrar
- Actualizar y/o modificar
- Bloquear
- Consultar

3.2.3. Modelado de Análisis.

3.2.3.1. Identificación de los Actores.

Se pudo distinguir dos tipos de actores, participando cada uno en los diferentes casos de uso y esperando algo distinto del sistema. A continuación se definirá las funcionalidades que ejecutara en el sistema.

Actor	USUARIO
Tipo	Primario
Papel	Se encarga de realizar las operaciones CRUD (Create, Retrieve, Update, Delete) con datos del empleado, realiza configuraciones, gestionar actividades, gestionar historial e imprimir reportes.

Tabla N° 10: Identificación de actores - usuario.

Actor	EMPLEADO
Tipo	Primario
Papel	Se encarga de marcar su asistencia personal al sistema a través de un lector de tarjetas de barras electrónico.

Tabla N° 11: Identificación de actores - empleado.

¹⁷ Para los campos de auditoria

3.2.3.2. Diagrama de Casos de Uso.

A continuación se definen los casos de uso del sistema el cual permitirá identificar las diferentes funcionalidades del sistema.

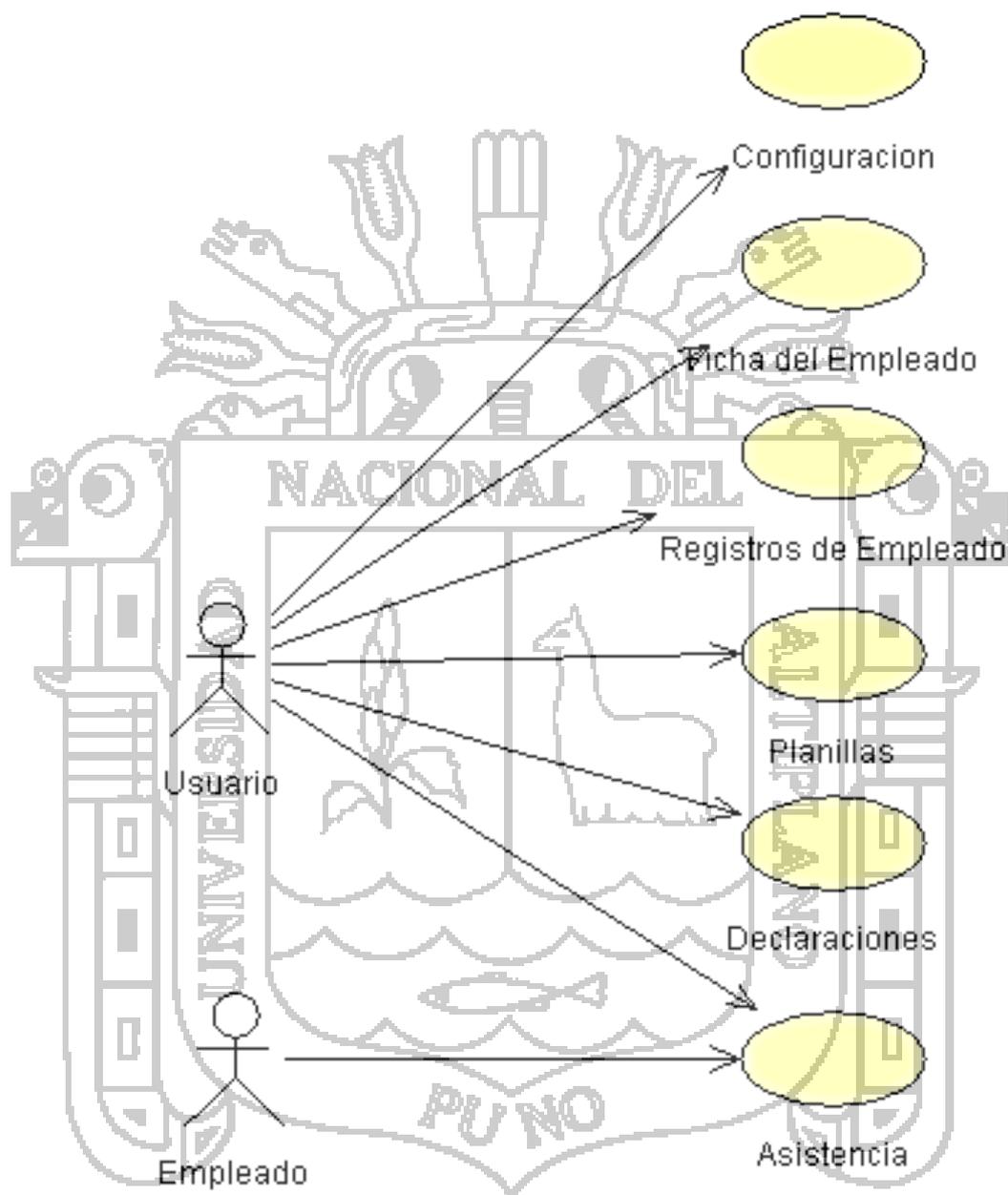


Figura N° 29: Diagrama de casos de uso.

Fuente: Elaboración propia.

A. DESCRIPCION DE CASOS DE USO

Nombre	Configuración
Descripción	El caso de uso, permite la configuración de fórmulas, parámetros y tipo de cambio que se realiza en el cálculo de planillas.
Nombre	Ficha del Empleado
Descripción	El caso de uso permite agregar, modificar y eliminar la ficha del empleado este consta de datos personales y datos laborales. (Modulo Escalafón). También permite a búsqueda de empleados para reportes, según datos ingresados.

Tabla N° 12: Descripción de caso de uso - configuración.

Nombre	Registro de Empleado
Descripción	El caso de uso permite el control de cada empleado en cuanto a sus contratos, permisos, descanso médico, papeletas de salida, vacaciones, sanciones entre otros para tener un historial del empleado. (Módulo Escalafón y Modulo Evaluación)

Tabla N° 13: Descripción de caso de uso - registro de empleado.

Nombre	Planilla
Descripción	El caso de uso permite la generación de planillas, boletas de pago, CTS, impuesto a la renta, gratificaciones. (Modulo Planilla). También se realiza los reportes de planillas según número de planilla, tipos de empleado, entre otros. Se toma los diferentes tipos de planilla.

Tabla N° 14: Descripción de caso de uso - planilla.

Nombre	Declaraciones
Descripción	El caso de uso permite la declaraciones a entidades como: AFP NET, planilla electrónica, T-registro, PLAME y bancos. (Modulo Planillas)

Tabla N° 15: Descripción de caso de uso - declaraciones.

Nombre	Asistencia
Descripción	El caso de uso registra la asistencia del empleado. Ingresada a través de un lector de tarjetas de barras, generada por el sistema y usada por el empleado directamente. (Modulo Asistencia y Modulo Evaluación)

Tabla N° 16: Descripción de caso de uso - asistencia.

3.2.3.3. Especificaciones de Casos de Uso.

Caso de Uso	Configuración	
Actor (es)	Usuario	
Referencia	RF001	
Precondición	Usuario identificado	
Descripción	Permite la configuración de fórmulas, parámetros y tipo de cambio que se realiza en el cálculo de planillas, declaraciones a entidades, boletas de pago.	
Flujo normal de eventos		ACTOR
	1	Solicita módulo de configuración del sistema
	2	
	3	Cambia el valor asignado del parámetro seleccionado
	4	
	5	Determina el cambio de los valores ingresados
	6	Almacena los nuevos valores registrados
Pos condición	Nuevos valores de parámetros cambiados	

Tabla N° 17: Especificaciones de caso de uso - configuración.



Caso de Uso	Ficha del Empleado		
Actor (es)	Usuario		
Referencia	RF002		
Precondición	Usuario identificado		
Descripción	Permite el control de empleados. Nuevo, edita o elimina empleado		
Flujo normal de eventos		ACTOR	SISTEMA
	1	Solicita acceso al control de empleados registrados.	
	2		Muestra la lista de empleados registrados.
	3	Solicita la búsqueda de un empleado	
	4		Solicita el ingreso de datos personales para su búsqueda.
	5	Ingresar datos personales para su búsqueda.	
	6		Muestra a empleados relacionados a los datos ingresados.
	7	Selecciona empleado para su control o actualización de datos	
	8		Muestra opciones de editar o eliminar empleado
	9	Actualiza datos de empleado existente.	
	10		Verifica la validez de los datos ingresados
	11	Valida los datos ingresados	
	12		Almacena datos de empleado actualizados.
Flujo normal de eventos	13		Muestra los datos actualizados
	8a		Datos no encontrados, carga configuración para la creación de nuevo empleado.
	9a	Ingresar datos de nuevo empleado.	
	10a		Verifica la validez de los datos ingresados
	11a	Valida los datos ingresados	
	12a		Almacena los datos de nuevo empleados.
13a		Muestra los datos actualizados	
Pos condición	Actualización de empleados		

Tabla N° 18: Especificaciones de caso de uso - ficha de empleado.

Caso de Uso	Registro del Empleado		
Actor (es)	Usuario		
Referencia	RF003		
Precondición	Empleado existente		
Descripción	Permite el control del historial de los empleados. Programa eventos de empleado		
Flujo normal de eventos		ACTOR	SISTEMA
	1	Solicita acceso al control de empleados registrados.	
	2		Muestra la lista de empleados registrados.
	3	Solicita la búsqueda de un empleado	
	4		Solicita el ingreso de datos personales para su búsqueda.
	5	Ingresar datos personales para su búsqueda.	
	6		Muestra a empleados relacionados a los datos ingresados.
	7	Selecciona empleado para su control.	
	8		Muestra opciones de editar o eliminar empleado
	9	Actualiza datos de empleado existente.	
	10		Verifica y valida de los datos ingresados
	11		Almacena datos de empleado actualizados.
Flujo normal de eventos	12		Muestra los datos actualizados
	8a		Datos no encontrados, carga configuración para la creación de nuevo empleado.
	9a	Ingresar datos de nuevo empleado.	
	10a		Verifica y valida los datos ingresados
	11a		Almacena los datos de nuevo empleados.
	12a		Muestra los datos actualizados
Pos condición	Actualización de empleados		

Tabla N° 19: Especificaciones de caso de uso - registro de empleado.



Caso de Uso	Planilla		
Actor (es)	Usuario		
Referencia	RF004		
Precondición	Empleado existente, asistencia de personal aprobado		
Descripción	Permite la formulación de la planilla de remuneraciones del empleado.		
Flujo normal de eventos		ACTOR	SISTEMA
	1	Solicita acceso al módulo planilla.	
	2		Muestra planillas existentes
	3	Solicita la búsqueda de planillas	
	4		Solicita el ingreso del número o tipo de planilla.
	5	Ingresar datos de planilla para búsqueda	
	6		Muestra planillas relacionados a los datos ingresados.
	7	Selecciona planilla para su control de planilla	
	8		Muestra opciones de editar o eliminar planilla
	9	Actualiza datos de planilla existente.	
	10		Verifica y valida de los datos ingresados
	11		Almacena datos de empleado actualizados.
12		Muestra los datos actualizados	
Flujo normal de eventos	8a		Datos no encontrados, carga configuración para la creación de nueva planilla
	9a	Ingresar datos de nueva planilla.	
	10a		Verifica y valida los datos ingresados
	11a		Almacena los datos de nueva planilla.
	12a		Muestra los datos actualizados
Pos condición	Planillas aprobadas para declaraciones a entidades		

Tabla N° 20: Especificaciones de caso de uso - planilla.



Caso de Uso	Declaración a entidades		
Actor (es)	Usuario		
Referencia	RF005		
Precondición	Planilla existente		
Descripción	Permite al formulación de las declaraciones a entidades		
Flujo normal de eventos		ACTOR	SISTEMA
	1	Solicita acceso a módulo de declaraciones a entidades	
	2		Muestra la lista de diferentes declaraciones a formular
	3	Determina la declaración a formular	
	4		Solicita el ingreso del número de planilla
	5	Ingresa datos de planilla	
	6		Muestra las planillas relacionados a los datos ingresados.
	7	Selecciona las planillas por realizar declaración	
	8		Genera planilla de declaraciones
	9	Actualiza datos de declaración.	
	10		Verifica y valida de los datos ingresados
	11		Almacena datos de declaración actualizados.
	12		Muestra los datos actualizados
Pos condición	Documentos de declaraciones generadas.		

Tabla N° 21: Especificaciones de caso de uso - declaraciones a entidades.

Caso de Uso	Asistencia		
Actor (es)	Usuario		
Referencia	RF006		
Precondición	Empleado activado, programación de registro del empleado		
Descripción	Permite en control de la asistencia de cada empleado		
Flujo normal de eventos		ACTOR	SISTEMA
	1	Solicita acceso a módulo de asistencia.	
	2		Muestra la lista de empleados activos.
	3	Solicita la búsqueda de un empleado	
	4		Solicita el ingreso de datos personales de empleado para su búsqueda.
	5	Ingresa datos personales para su búsqueda.	
	6		Muestra a empleados relacionados a los datos ingresados.
	7	Selecciona empleado para su verificación de control de asistencia.	
	8		Muestra opciones de editar asistencia.
	9	Actualiza datos de la asistencia del empleado.	
	10		Verifica y valida de los datos ingresados
	11		Almacena datos de la asistencia del empleado.
	12	Aprueba la asistencia del empleado.	
13		Muestra los datos actualizados	
Pos condición	Aprobación de asistencia del empleado.		

Tabla N° 22: Especificaciones de caso de uso - asistencia.

3.2.3.4. Modelado de Dominio.

Diagrama conceptual: Antes de definir el modelo estático o de clases, es necesario definir el modelo conceptual, el cual nos muestra los conceptos presentes en el dominio del problema. Un concepto para este caso, en términos de la programación orientada a objetos, es un objeto del mundo real; es decir, es la representación de cosas del mundo real y no de componentes de software. En él no se definen operaciones (o métodos); en este modelo se pueden mostrar los conceptos, los atributos de los conceptos (opcionalmente) y la relación o asociación entre ellos.

Informalmente podríamos decir que un concepto es una idea, cosa u objeto. Para descubrirlos debemos analizar los sustantivos en las descripciones textuales del dominio del problema, es decir, de la descripción del sistema, de los requerimientos y de los Casos de Uso:

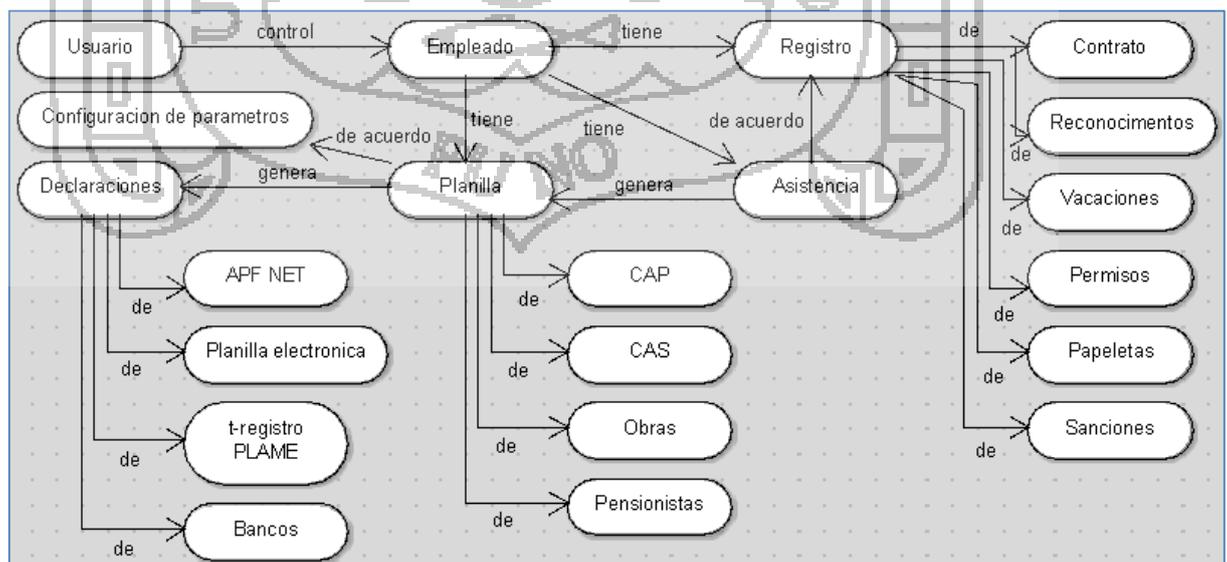


Figura N° 30: Modelado de dominio - diagrama conceptual.

En esta etapa corresponde al análisis orientado a objetos, aquí se detalla una vista parcial de los objetos que intervienen y la relación de asociación entre ellos la siguiente figura muestra la representación visual de las clases conceptuales que se van a utilizar.

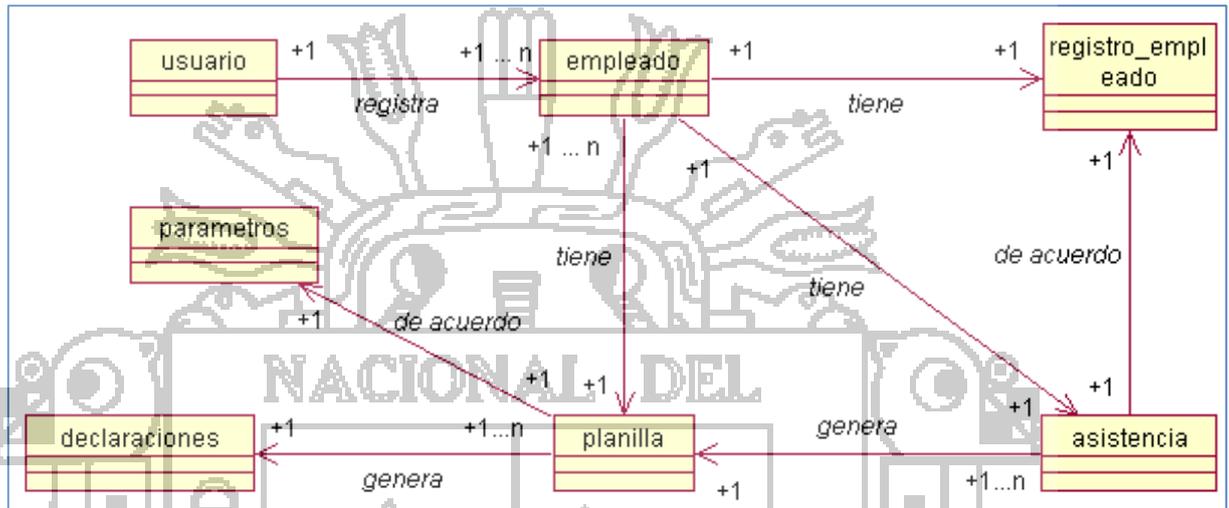


Figura N° 31: Modelado de dominio - clases conceptuales.

3.2.4. Diseño del Sistema.

3.2.4.1. Diseño de la Arquitectura del Sistema.

El diseño arquitectónico, define la relación entre los elementos estructurales del software. A continuación se presenta diseño arquitectónico del sistema.

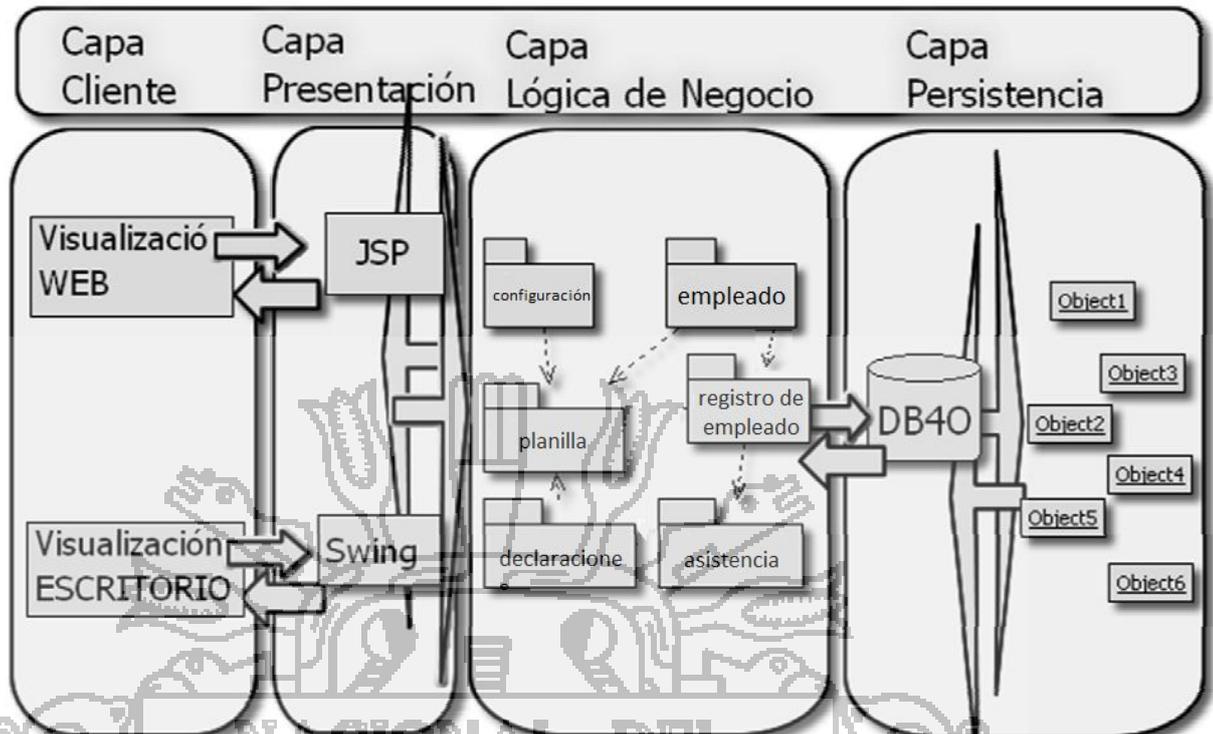


Figura N° 32: Diseño de la arquitectura del sistema.

Fuente: Elaboración propia.

- La capa cliente corresponde a las interfaces con las que interactúa el cliente, en este caso una interfaz Web.
- La capa de presentación prepara datos para su envío a la capa de cliente y procesa solicitudes desde la capa de cliente para su envío a la capa lógica de negocios, como lo hace JSP y los componentes Swing.
- En capa lógica de negocio se realizan las funciones principales de la aplicación: procesamiento de objetos y la implementación de los métodos. Es aquí donde se sitúan los paquetes asociados al sistema.
- La capa de persistencia permite que los objetos perduren en el tiempo, a cargo de DB4o.

3.2.4.2. Diagramas de Secuencias.

Los siguientes gráficos muestran los eventos que el actor realiza para los escenarios principales de éxito. Cada uno de los diagramas representa a los formularios que presenta el prototipo según actor.

A: D.S. para el C.U. Configuración.

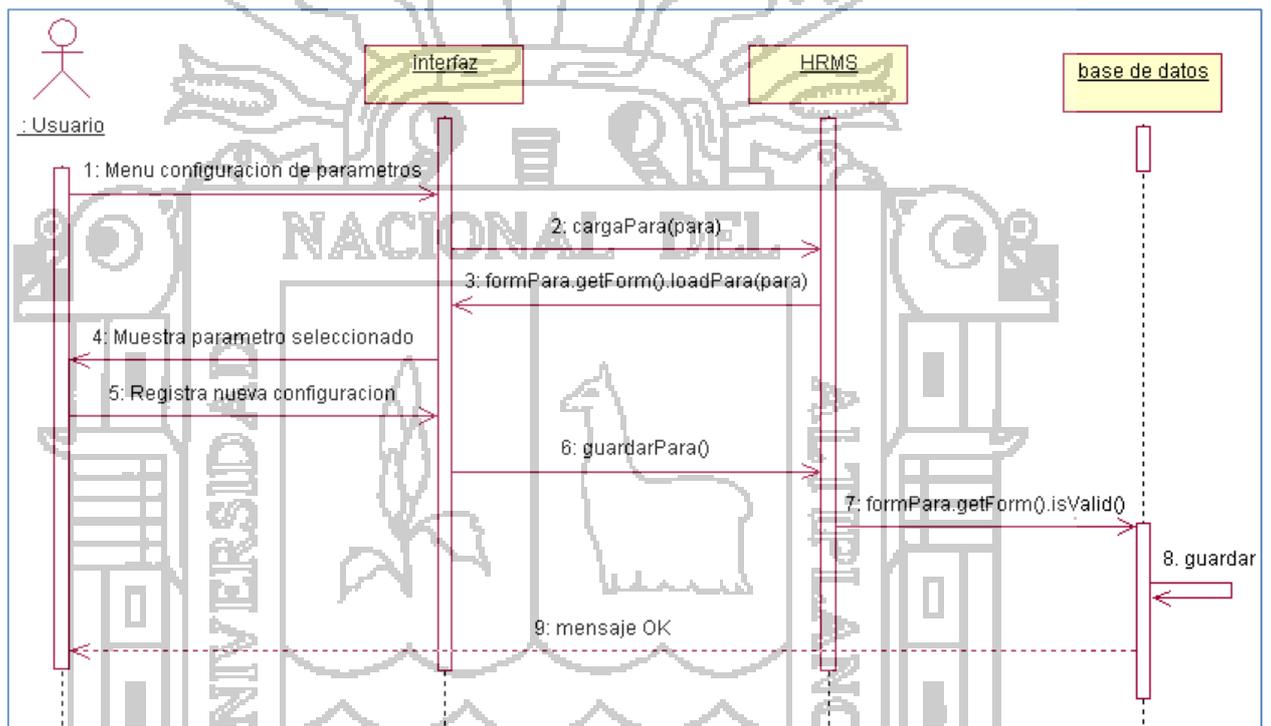


Figura N° 33: Diagrama de secuencia del caso de uso configuración.

B: D.S. para el C.U. Ficha del Empleado.

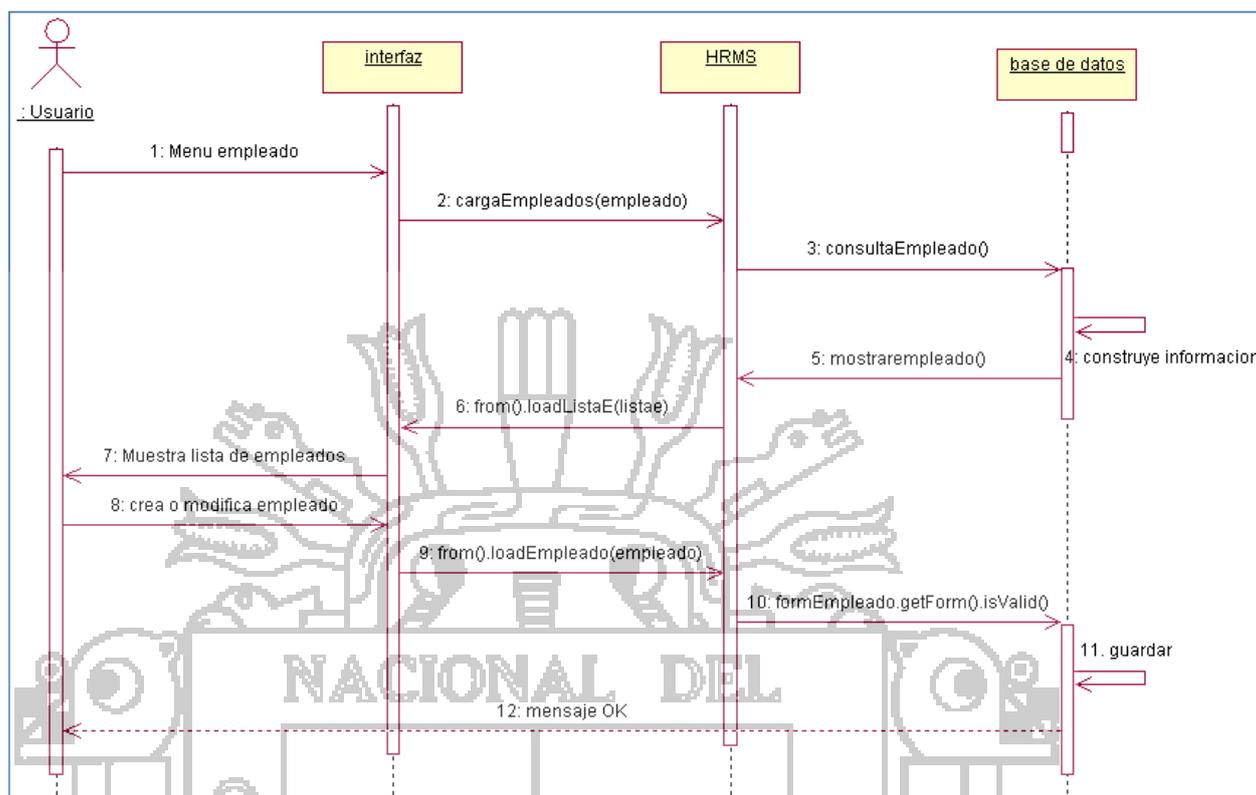


Figura N° 34: Diagrama de secuencia del caso de uso ficha de empleado.

C: D.S. para el C.U. Registro de Empleado.

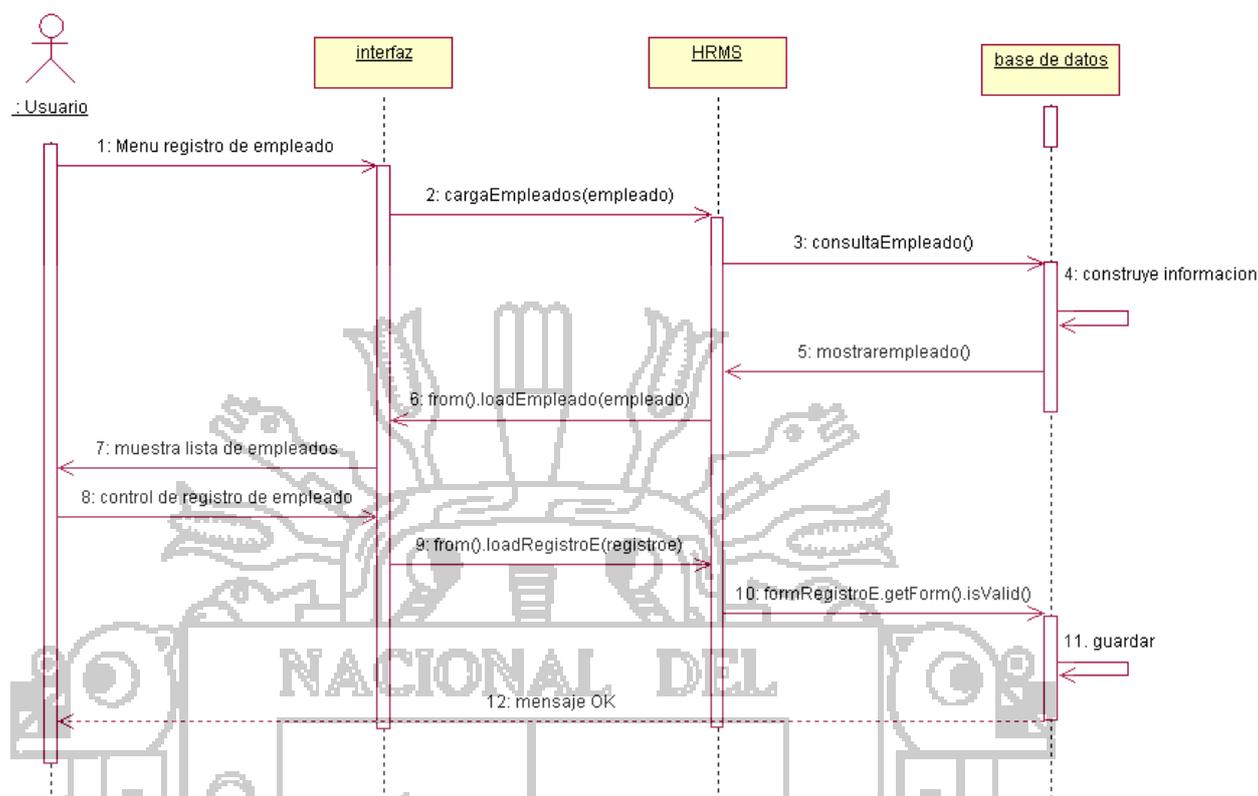


Figura N° 35: Diagrama de secuencia del caso de uso registro del empleado.

D: D.S. para el C.U. Planilla.

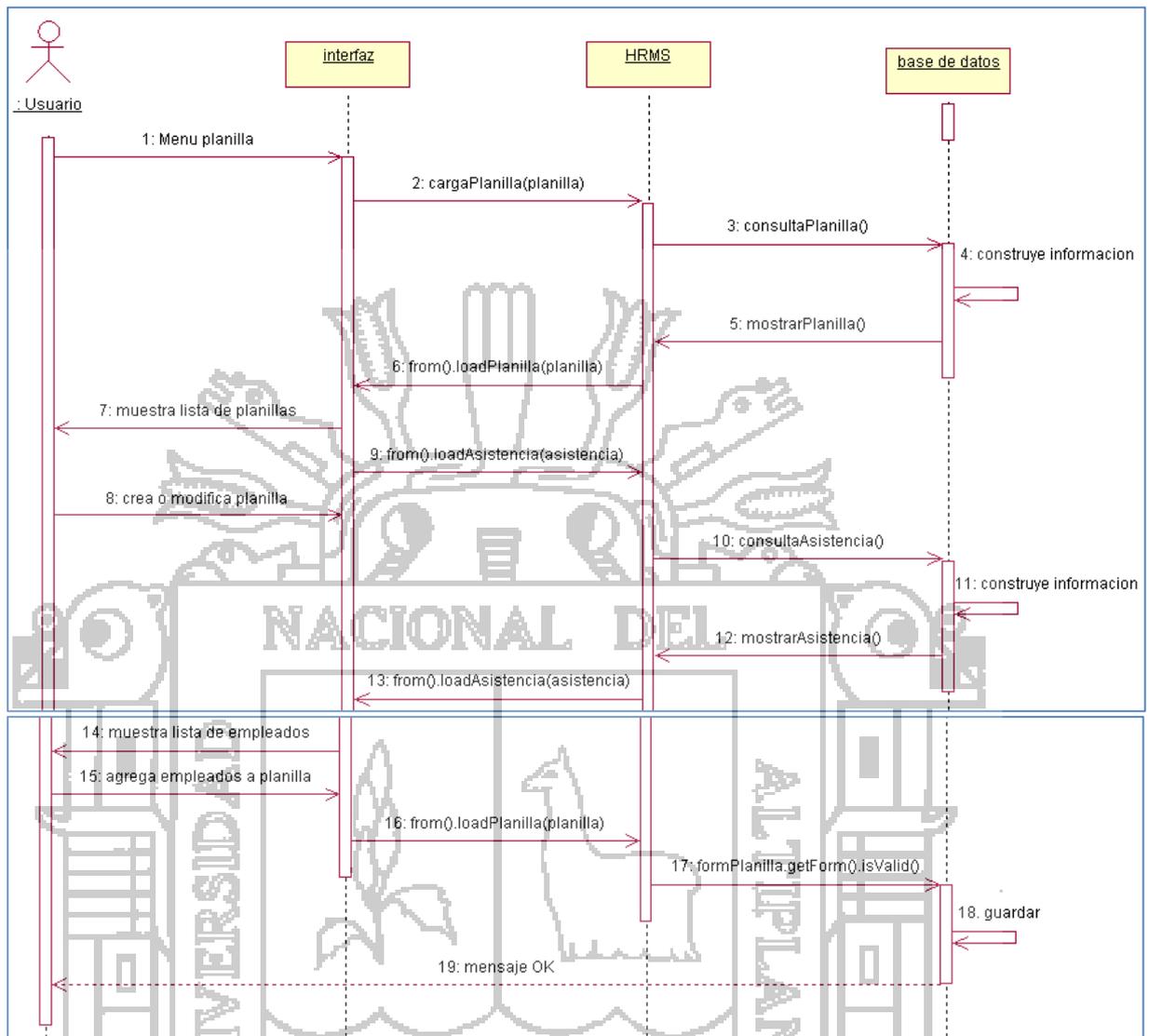


Figura N° 36: Diagrama de secuencia del caso de uso planilla.

E: D.S. para el C.U. Asistencia.

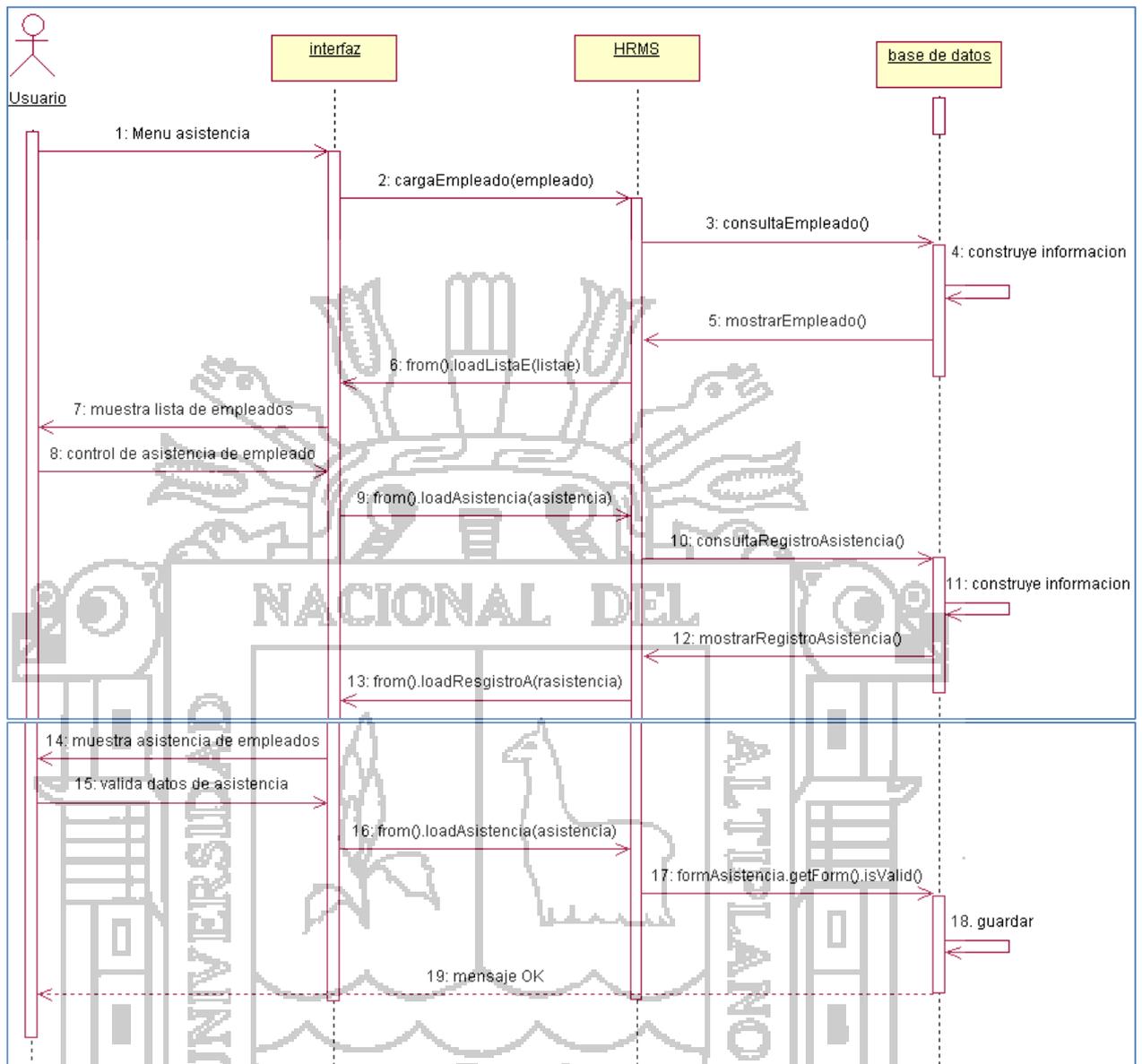


Figura N° 37: Diagrama de secuencia del caso de uso asistencia.

F: D.S. para el C.U. Declaraciones.

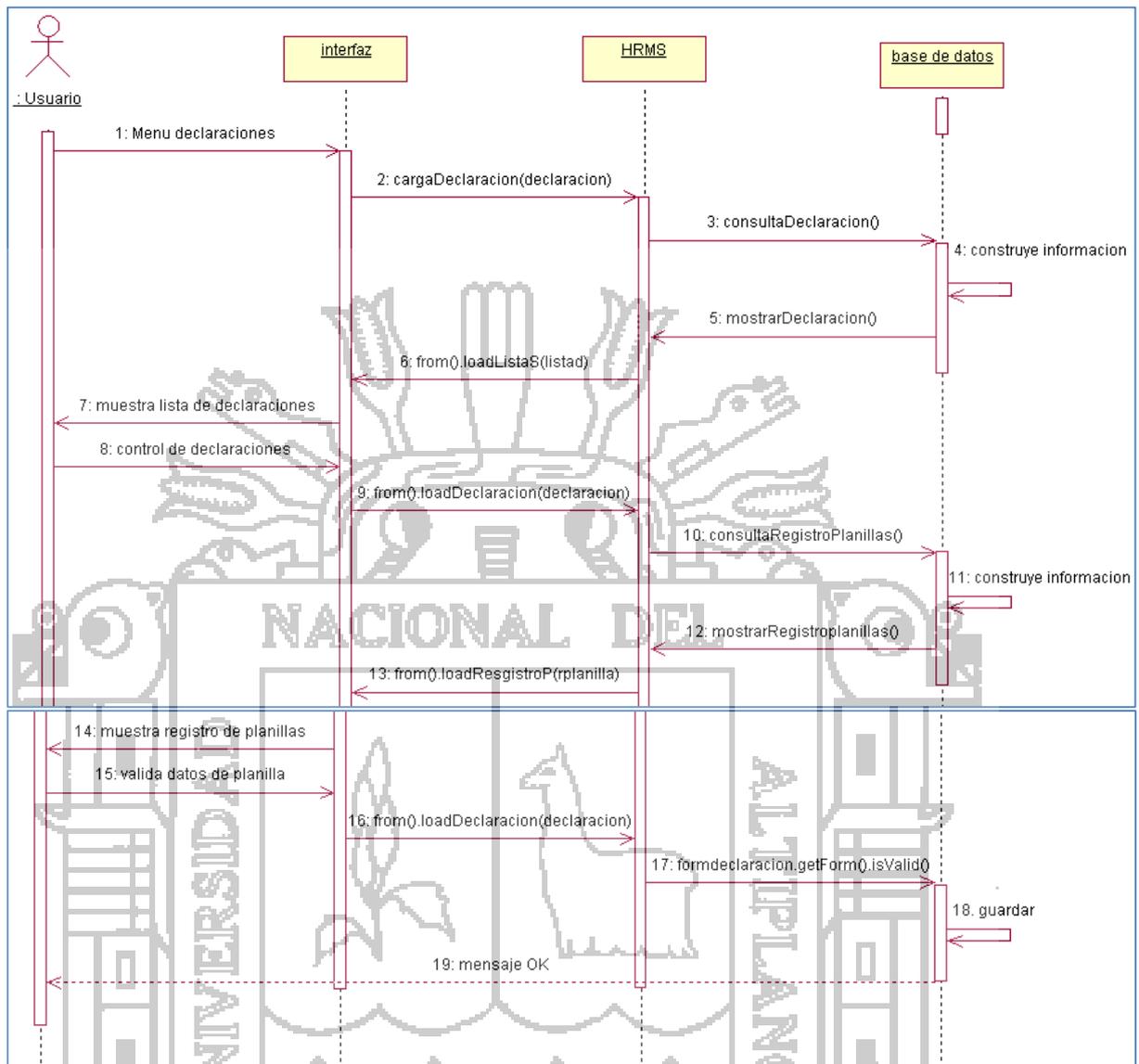


Figura N° 38: Diagrama de secuencia del caso de uso declaraciones.

3.2.4.3. Diagramas de Colaboraciones.

A: D.C. para el C.U. Configuración.

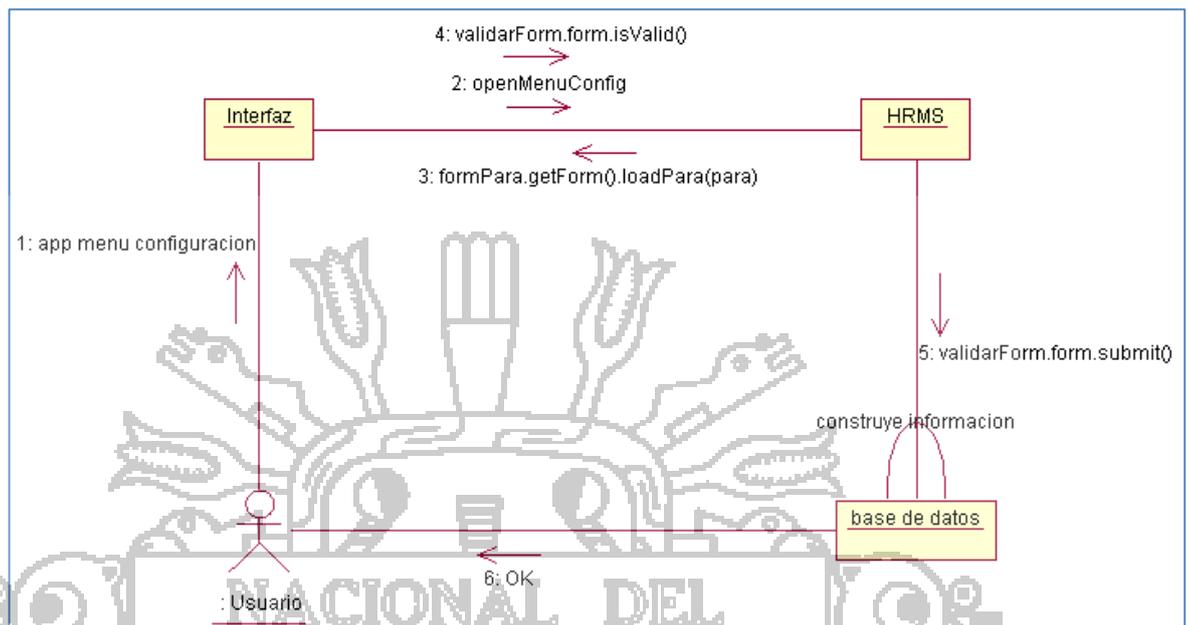


Figura N° 39: Diagrama de colaboración del caso de uso configuración.

B: D.C. para el C.U. Ficha del Empleado.

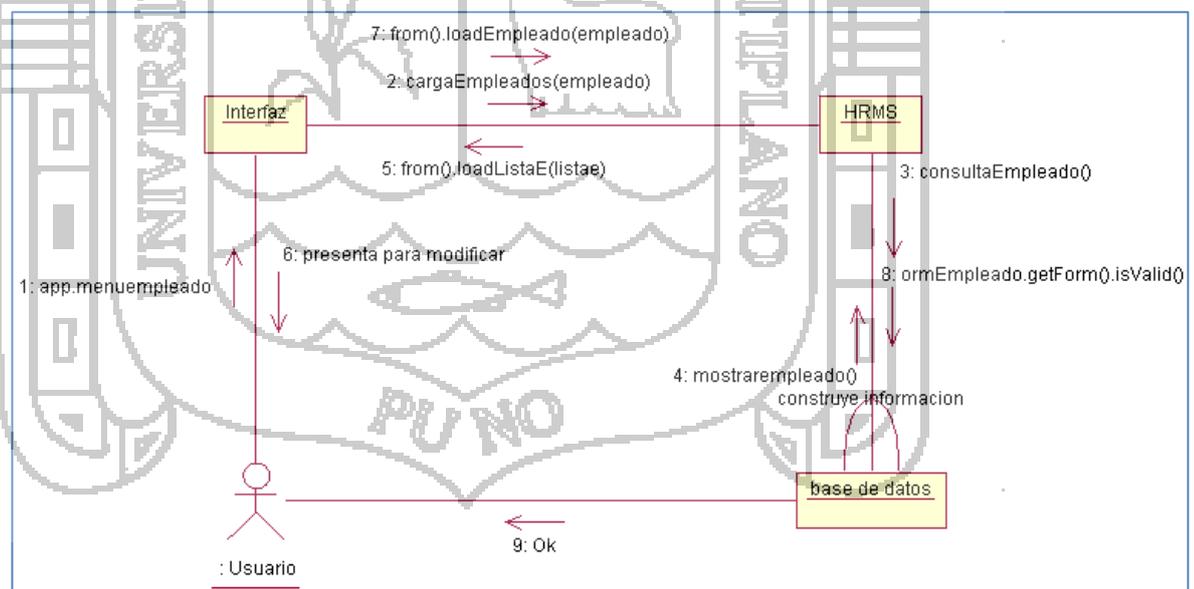


Figura N° 40: Diagrama de colaboración del caso de uso ficha del empleado.

C: D.C. para el C.U. Registro de Empleado.

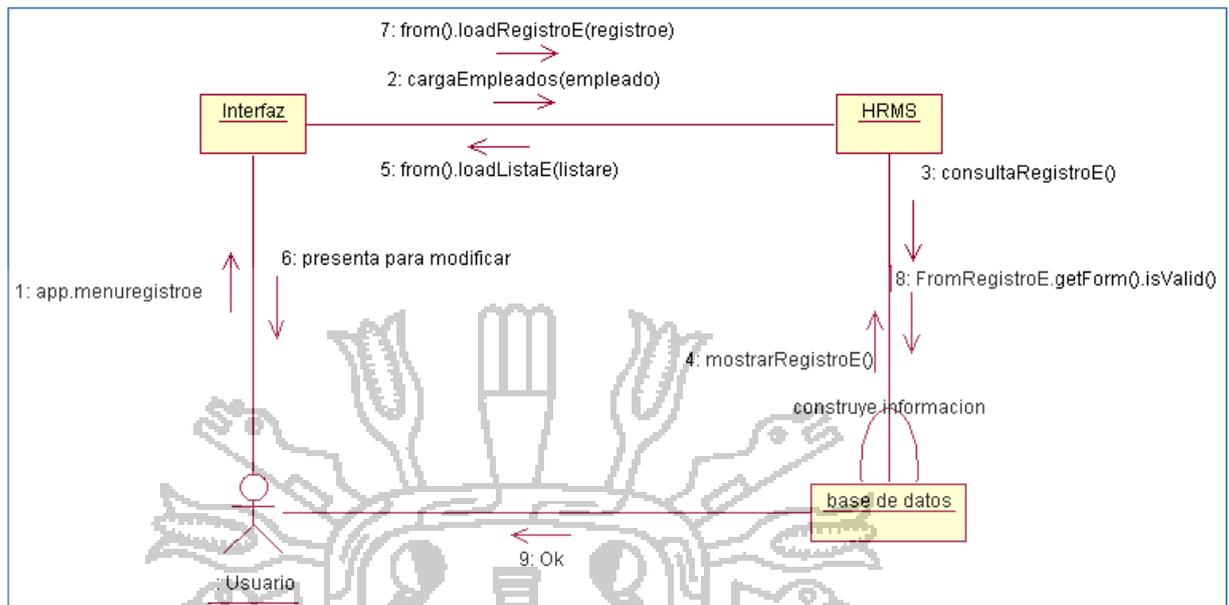


Figura N° 41: Diagrama de colaboración del caso de uso registro de empleado.

D: D.C. para el C.U. Planilla.

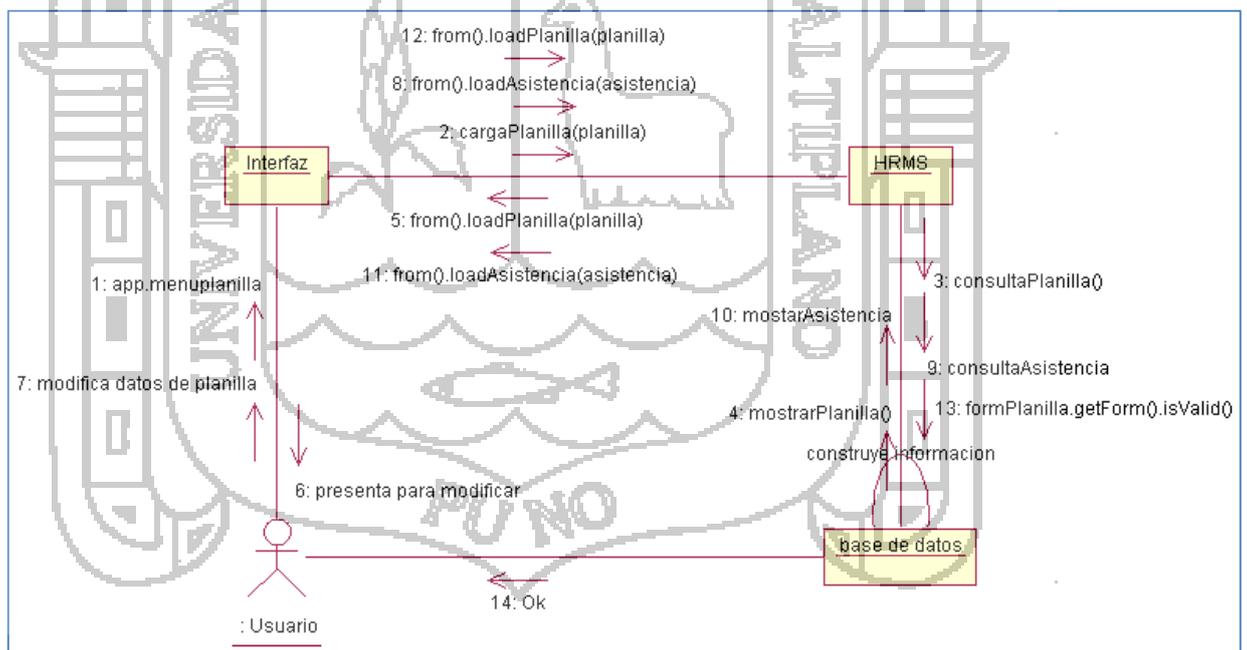


Figura N° 42: Diagrama de colaboración del caso de uso planilla.

E: D.C. para el C.U. Asistencia.

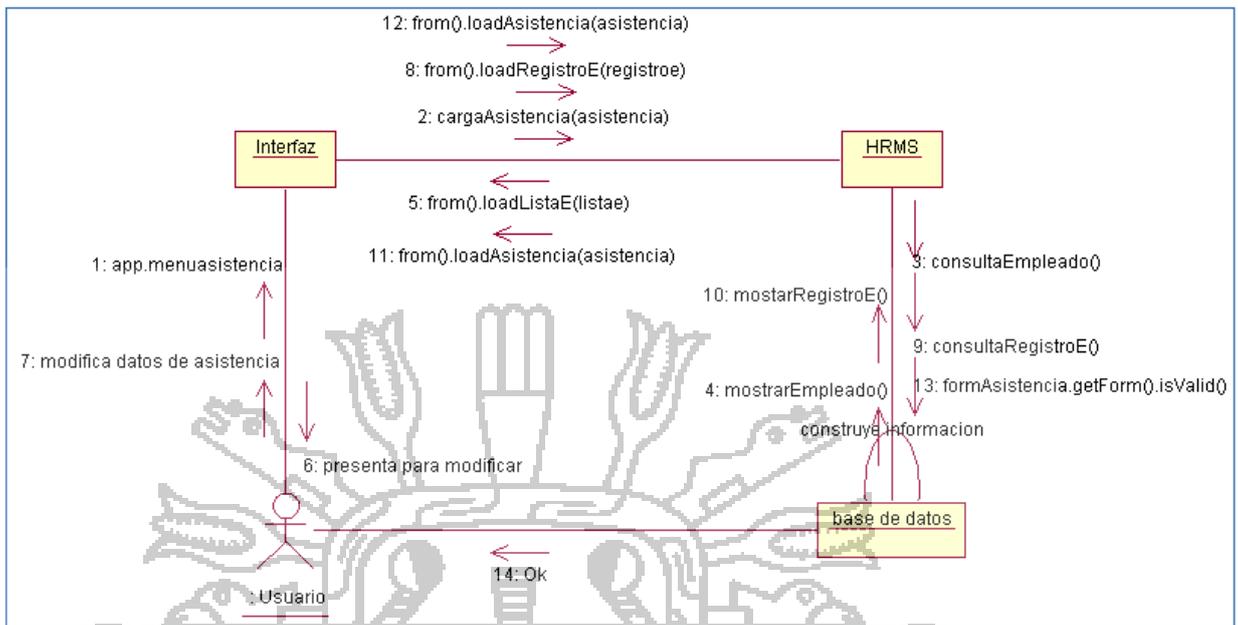


Figura N° 43: Diagrama de colaboración del caso de uso asistencia.

F: D.C. para el C.U. Declaraciones.

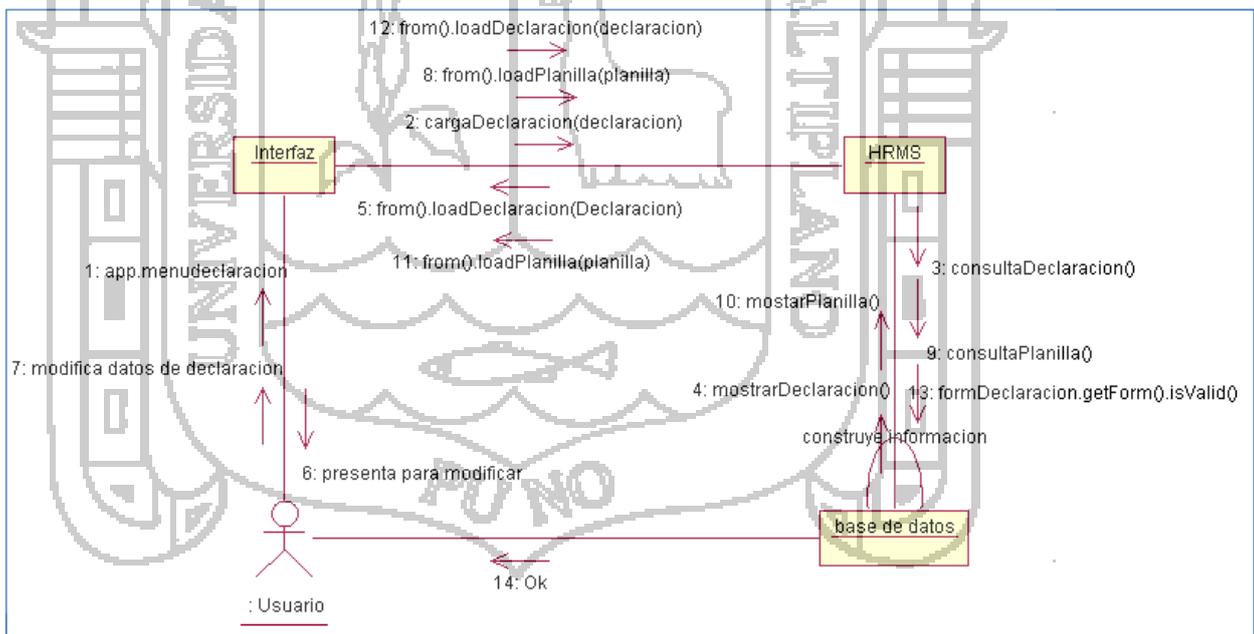


Figura N° 44: Diagrama de colaboración del caso de uso declaraciones.

3.2.4.4. Diagrama de Clases.

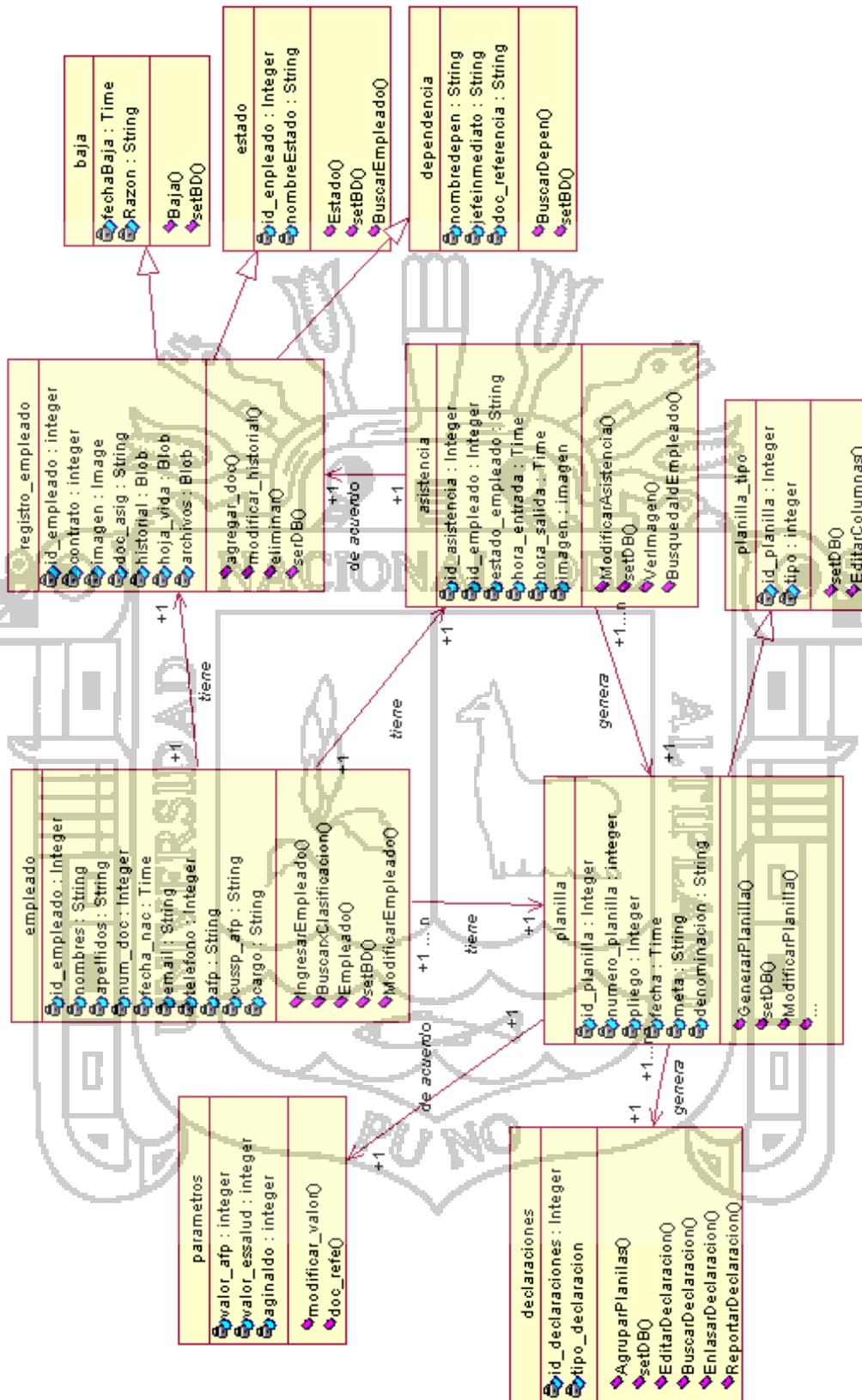


Figura N° 45: Diagrama de clases - modelo base de datos orientado a objetos.

Fuente: Elaboración propia

3.2.4.5. Diseño de Interfaz.

Estándares HCI: La Interacción Persona-Ordenador del inglés Human Computer Interaction (HCI) “es una disciplina relacionada con el diseño, evaluación, implementación y el estudio de los fenómenos que rodean los sistemas para uso humano”.

HCI se ocupa del diseño de sistemas informáticos que coincidan con las necesidades de los usuarios, usando conocimientos, métodos y disciplinas muy diferentes. Los estándares HCI están relacionados con la usabilidad y se pueden categorizar en:

- El uso del producto.
- La interfaz de usuario y la interacción.
- El proceso utilizado para desarrollar un producto.
- La capacidad de una organización de aplicar diseño centrado en el usuario

El objetivo del HCI es crear software usable, seguro y funcional. La usabilidad tiene como objetivo crear sistemas:

- Eficientes.
- Efectivos.
- Seguros.
- Útiles.
- Fáciles de aprender.

Cumpliendo los requerimientos de usabilidad, se realizó los siguientes diseños de interfaz de usuario. HRMS: Para el diseño de la interfaz el usuario tiene la opción de elegir la base de colores según su preferencia y gustos; para los gráficos se usaron imágenes que ayuden a la interacción de las opciones del sistema (ayuda visual e intuitiva).



Figura N° 46: Diseño de interfaz - esquema de diseño.
Fuente: Elaboración propia.



Figura N° 47: Diseño de interfaz -registro de asistencia.
Fuente: Elaboración propia

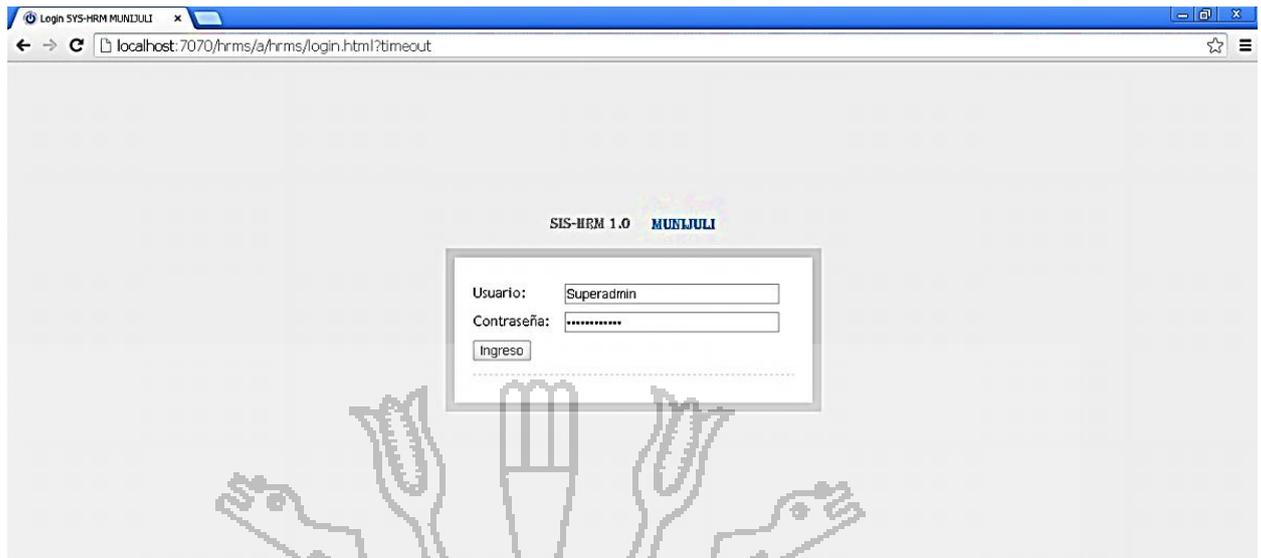


Figura N° 48: Diseño de interfaz - interfaz acceso a usuario.
Fuente: Elaboración propia.



Figura N° 49: Diseño de interfaz - crear usuario.
Fuente: Elaboración propia.

Empleados

Cod. empleado: buscar insertar

Nombre:

Tipo empleado:

Tipo planilla: nombrados mayo insertar

Modelo

RMV: 750,00

A. Familiar: 10%

APORTE OBLIGATORIO COMISIÓN % SOBRE R.A.: 13%

PRIMA SEGURO: 3,00%

SNP/ ONP: 13%

PRIMA: 15,00%

HORIZONTE INTEGRAL: 16,00%

PROFUTURO: 17,00%

18,00%

ESSALUD: 9%

SCTR: 1,25%

PLANILLA DE REMUNERACIONES

PERIODO:

RUC:

DENOMINACIÓN O RAZÓN SOCIAL:

INGRESOS DEL TRABAJADOR

RETENCIONES A CARGO DEL TRABAJADOR

ORDEN CÓDIGO APELLIDOS Y NOMBRES CARGO U. OCUPACIÓN ASIGNACIÓN FAMILIAR SUeldo ASIGNACIÓN TOTAL REMUNERACIÓN SISTEMA PRIVADO DE PENSIONES - AFP TOTAL

Figura N° 50: Diseño de interfaz - interfaz planilla de remuneraciones.
Fuente: Elaboración propia.

Asistencia

Desde: 2014-06-01 Hasta: 2014-06-07

Modificar Buscar Imprimir Guardar Eliminar

Empleado	E/S	Lun 01 Jun 2014	Mar 02 Jun 2014	Mir 03 Jun 2014	Jue 04 Jun 2014	Vie 05 Jun 2014	Sab 06 Jun 2014	Dom 07 Jun 2014	Total
iperez	E	00:00 hrs	00:00 hrs						
iperez	S	00:00 hrs	00:00 hrs						
Rcardenas	E	00:00 hrs	00:00 hrs						
Rcardenas	S	00:00 hrs	00:00 hrs						
Jvaldez	E	00:00 hrs	00:00 hrs						
Jvaldez	S	00:00 hrs	00:00 hrs						
Aarcaya	E	00:00 hrs	00:00 hrs						
Aarcaya	S	00:00 hrs	00:00 hrs						
Tardanza acumulada		00:00 hrs	00:00 hrs						

Figura N° 51: Diseño de interfaz - control de asistencia.
Fuente: Elaboración propia.

3.2.5. Implementación del Sistema.

La implementación del Sistema, se hizo utilizando toda la potencialidad de plataforma JAVA EE y el gestor de Base de datos orientado a objetos DB4O. Se utilizó el patrón MVC para la implementación teniéndose en la vista al framework Java Server Faces por su alta usabilidad en las interfaces para usuario; en el controlador se utilizó el framework Spring por su alta interacción con diferentes frameworks; el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Para la generación de reportes se utilizó JaspersReports conjuntamente con su herramienta de diseño IReport, dándonos la capacidad de tener un diseño exacto según los requerimientos de la unidad de personal de la Municipalidad Provincial de Chucuito Juli.

A continuación se detallan las secuencias más importantes para la implementación.

3.2.5.1. Control de Concurrencia en DB4O.

Db4o es una base de datos orientada a objetos, por lo cual obtiene todas las características del paradigma, pero también limita al programador de algunas características que provee una base de datos relacional. En primera instancia, en un objeto no se puede declarar un atributo como Primary Key, por lo que si existe algún atributo de un objeto que deba ser único,

debe ser tratado mediante métodos, en otras palabras, se debe implementar la funcionalidad de impedir el ingreso de un atributo que ya está en la base de datos.

¿Cómo se hace con una base de datos en modo cliente-servidor?

La pregunta anterior hace mención a la posibilidad, que dos clientes ingresen al mismo tiempo el mismo valor para un atributo “identificador”, por ejemplo el DNI, o en el caso de los empleados, el código de registro. Para aquello, DB4O provee semáforos, los que permiten bloquear las operaciones riesgosas, es decir, si dos usuario solicitan un semáforo con un nombre determinado, el segundo al momento de solicitarlo mediante el método para aquello, le retornará **false**, indicando que otro usuario pidió un semáforo antes y no lo ha devuelto.

A continuación se describe el código utilizado para aquello.

```
28 try{
29     ObjectContainer cliente =
30         Db4o.openClient(Util.HOST, Util.PUERTO, Util.USER, Util.PASS);
31     while(!cliente.ext().setSemaphore("semaforo", 2));
32     //Hacer verificación
33     //Guardar el dato si se desea
34     cliente.ext().releaseSemaphore("semaforo");
35     cliente.close();
36 }catch (IOException msg){
37     System.out.println(msg.getMessage());
38 }
```

Figura N° 52: Código con semáforos DB4O.

En la Figura 52: Se puede observar que si dos usuarios piden el semáforo de nombre “semaforo”,

(cliente.ext().setSemaphore("semáforo",2)), al segundo usuario le retornará falso y como en el ciclo while existe una negación, esperará 2 milisegundos antes de preguntar nuevamente por el semáforo, es decir, el usuario podrá pasar el ciclo **while** una vez se devuelva el semáforo (cliente.ext().releaseSemaphore("semaforo").

De ésta forma, se controlan los ingresos de atributos que son identificadores dentro del objeto.

3.2.5.2. Implementación con AJAX.

El sistema Web dedicado, principalmente, a la búsqueda de material, posee varias características importantes, ya que posee las siguientes opciones:

- Búsqueda de empleados.
- Consultar de asistencia por empleado.
- Sistema de almacenamiento para archivos digitales.
- Recomendación de búsqueda.
- Configuración de cuenta.

Todas las opciones mencionadas anteriormente han sido implementadas mediante la tecnología Ajax, esto permite que la interfaz principal se mantenga intacta mientras se realiza la recarga solo del contenido que solicita. A continuación, se dará a conocer un ejemplo de cómo se realiza la recarga (funciones en JavaScript).

```

30 function cargarPagina(div,pagina){
31     var ajax=objetoAjax();
32     var divMensaje = document.getElementById(div);
33     ajax.open("GET", pagina);
34     ajax.onreadystatechange=function()
35     {
36         if (ajax.readyState==4)
37         {
38             divMensaje.innerHTML = ajax.responseText;
39         }
40     }
41     ajax.send(null)
42 }

```

Figura N° 53: Función que ocupa el sistema para recargar la Web.

En la Figura 53: se muestra que la función (de javascript) recargarPagina(div,pagina), se puede apreciar que se le entregan 2 parámetros, el primero (div) es el nombre o identificador del contenedor en la que realizará la recarga, y el segundo parámetro (pagina) es la dirección de la pagina que estamos solicitando para que sea recargada, ésta puede venir con datos incluidos, por ejemplo, pagina.jsp?idObjeto=123.

A continuación se especifica la función objetoAjax() que se nombra en la imagen anterior en la línea 31.

```

4 function objetoAjax(){
5     var xmlhttp=false;
6     try {
7         xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
8     } catch (e) {
9         try {
10            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
11        } catch (E) {
12            xmlhttp = false;
13        }
14    }
15
16    if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
17        xmlhttp = new XMLHttpRequest();
18    }
19    return xmlhttp;
20 }

```

Figura N° 54: Función objetoAjax().

En la imagen anterior se aprecia la función encargada de crear el objeto Ajax, que nos permitirá mantener la comunicación asíncrona con el servidor.

3.2.5.3. Implementación con SWING en Java.

La aplicación de escritorio que utilizará el bibliotecario está realizada en java con componentes Swing (Biblioteca gráfica), los componentes que más se utilizaron son:

- JLabel : Utilizado para texto en la aplicación.
- JButton : Utilizado para los botones.
- JList : Utilizado para las listas.
- JTextArea : Utilizado para las cajas de texto de varias líneas.
- JTextField : Utilizado para cajas de texto.
- JTable : Utilizado para la creación de tablas.
- JOptionPane : Utilizado para mensajes al usuario.
- JDialog : Utilizado para contener otros componentes swing, (ventana flotante).
- JFrame : Utilizado para la contener otros componentes swing (ventana).

A: JTABLE

Es relevante hacer mención que el componente JTable no fue utilizado por defecto, es decir no tiene las propiedades definidas que la biblioteca proporciona, ya que la tabla por

defecto puede ser editada (modificar el texto que aparece en las celdas), y en el sistema aquello no tiene utilidad, por lo cual fue reescrito un método. El método que fue reescrito fue:

```
23  [Borrar metodos isCellEditable de la clase javax.swing.table.DefaultTableModel]
    ↑
    public boolean isCellEditable (int row, int column)
25  {
26      return false;
27  }
28
```

Figura N° 55: Método reescrito en el modelo de JTable.

En la Figura 55: se aprecia el método que fue reescrito en una clase que hereda las propiedades de DefaultTableModel (Clase que se asocia al modelo de tabla). Este método se encarga de retornar si las celdas son editables o no. Para que el componente JTable se encargue de aquello, esa es la razón de los parámetros que se le entregan, row y column, pero en este caso puntual ninguna fila y columna se podrá editar, es decir, no importa la celda que se le entregue por parámetro, el método siempre retornará false.

Al ser reescrita ésta propiedad, la herramienta CASE también nos hace saber que el método está sobrescribiendo al método del padre. Luego, ésta nueva clase será asociada como modelo al componente JTable.

B: JDIALOG y JFRAME

Al comenzar la programación de la interfaz gráfica, lo primero que se hace es la creación de ventanas, para este caso JFrame, la herramienta CASE proporciona una serie de utilidades para que la creación de ventanas se pueda realizar con un entorno gráfico.

¿Hay alguna restricción al momento de crear ventanas?

En el transcurso del proyecto, esta pregunta fue tomando el peso que en un comienzo se debió tener, el sistema estaba repleto de ventanas tipo JFrame, además cada vez que se llamaba a una ventana se construía nuevamente, es decir, si una ventana se abre 20 veces, 20 veces el recolector de basura de java tendrá la misión de liberar memoria (ya que se pierde referencia al objeto). Entonces, antes de implementar cualquier parte del software es necesario recurrir a la literatura y a la experiencia que han tenido otros programadores que ya han pasado por aquello. Luego de esto, se debió reconstruir prácticamente todo, pero ésta vez con nuevos criterios que han hecho que la aplicación pueda ser utilizada con mayor calidad.

Las conclusiones que se tomaron son las siguientes:

- En toda la aplicación debe existir solo un contenedor JFrame como ventana principal. Como ventanas secundarias se utiliza el componente JDialog.

- Las ventanas secundarias al momento de cerrarlas solo quedarán “ocultas” `<setVisible(false)>`, es decir, no destruirlas, y al momento de llamarlas nuevamente hacerlas visible `<setVisible(true)>`, esto hace que las ventanas aparezcan mucho más rápida que estarlas construyendo nuevamente. También en cada `JDialog` del sistema se implementó un método que es fundamental para esto, es el método `reset()`. Este método, deja la ventana nuevamente por defecto, es decir elimina todos los contenidos en las cajas de texto, deja la ventana en la posición inicial, etc.

- Al iniciar el programa, cargar en memoria todas las ventanas que el sistema necesita, lo que proporciona mayor rapidez al momento de su ejecución (basta con crear las ventanas al inicio del programa).

3.2.5.4. Código de Barras.

Una de las funciones principales del sistema, es que los empleados de la Municipalidad, registren su asistencia a través de un lector de código de barras, cada empleado debe portar su tarjeta de empleado en ella está impresa los datos del empleado y su código de empleado, este código es expresado en código de barras, al momento de deslizar tu tarjeta por el lector de código de barras se registra la asistencia del empleado.

A: IMPRESORA DE CÓDIGOS DE BARRA.

Para la generación de códigos de barra, existen variadas posibilidades, pero no todas ofrecen soluciones eficientes. La solución más óptima, es la adquisición de una impresora de etiquetas. Una impresora de este tipo no posee el mismo funcionamiento de una impresora tradicional, una de las propiedades más importantes es la impresión térmica sobre etiquetas autoadhesivas, la que ofrece una buena calidad de impresión. Una vez obtenida la impresora, existe la interrogante, ¿cómo imprimir los códigos de barra?, nuevamente existen varias opciones, entre ellas están:

- Obtener un software para el diseño e impresión de códigos de barra.
- Utilizar bibliotecas para JAVA como "barbecue-1.5-beta1", las que producen imágenes de códigos de barra.
- Impresión directa mediante el lenguaje de programación que soporte la impresora, en el caso puntual de Zebra LP2824, el lenguaje es EPL2 (EltronProgrammingLanguage).

La última opción es la más eficiente y compleja, pero para este proyecto es indispensable que el mismo software genere las etiquetas de clasificación y las etiquetas de códigos de barra, por lo cual, adquirir otra aplicación es descartada en primer

instancia, luego se ha decidido la impresión a través del lenguaje que provee la impresora.

Para imprimir una etiqueta, se crea un archivo temporal, por ejemplo archivo.txt en él se encuentran los comandos que son enviados al puerto de la impresora (LPT1) con los comandos de sistema, por ejemplo, copy código.txt LPT1

Ejemplo de un código de barras sencillo:

```
N
A50,5,0,1,2,2,N,"MUNICIPALIDADPROVINCIAL CHUCUITO JULI"
A50,30,0,4,1,2,N,"NOMBRE EMPLEADO"
B250,100,0,E30,3,20,160,B,"8424070022915"
P1
```

Comando N:

Descripción: Indica el comienzo de una nueva etiqueta.

Comando A:

Descripción: Indica texto, formato =
Ap1,p2,p3,p4,p5,p6,p7,"TEXTO"

Dónde: p1= posición del texto horizontal.

p2= posición vertical.

p3= rotación. 0=sin rotación; 1=90 grados; 2= 180 grados; 3= 270 grados.

p4= fuente. No se muestra la tabla debido a lo extenso de ésta.

p5= amplía el texto horizontalmente (lo multiplica por el valor) de 1 a 9.

p6= amplía el texto verticalmente (lo multiplica por el valor) de 1 a 9.

p7= indica si es normal o al revés. Valores N y R.

Comando B:

Descripción: similar al comando "A" pero en TEXTO se indica el código de barras.

Comando P:

Descripción: Indica la cantidad de copias que se deben imprimir.

B: LECTOR DE CÓDIGOS DE BARRA.

Un lector de códigos de barra decodifica la información a través de la digitalización proveniente de una fuente de luz reflejada en el código y luego se envía la información a una computadora como si la información hubiese sido ingresada por teclado. Lo bueno de los lectores de códigos de barra es que finalizan el ingreso de la lectura con [Enter], lo que permite manejar los eventos.

3.2.6. Pruebas del Sistema.

Una vez terminada la implementación del Sistema, se realizó la prueba con 3 trabajadores Administrativos de la unidad de personal de la Municipalidad Provincial de Chucuito Julo, y con 50 empleados. Después de esto, se realizó una encuesta para ver cuál es la opinión que tienen los Trabajadores Administrativos y empelados de la Municipalidad con respecto al sistema.

3.2.6.1. Encuesta a Administrativos de la Unidad de Personal.

En la encuesta se realizó a los 3 trabajadores Administrativos de la unidad de personal de la Municipalidad con las siguientes preguntas:

1. ¿Con el Sistema obtuvo buenos y esperados resultados con poco trabajo?

- a) Definitivamente si
- b) Probablemente si
- c) Indeciso
- d) Probablemente no
- e) Definitivamente no

2. ¿Encuentra amigable la interfaz del Sistema?

- a) Muy de acuerdo
- b) Algo de acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Algo en desacuerdo
- e) Muy en desacuerdo Excelente

3. En cuanto a la Facilidad de Uso del Sistema ¿Encuentra de fácil uso el sistema?

- a) Definitivamente si
- b) Probablemente si
- c) Indeciso
- d) Probablemente no
- e) Definitivamente no

4. ¿El Sistema tiene la capacidad de realizar sus funciones sin incidentes?

- a) Muy de acuerdo
- b) Algo de acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Algo en desacuerdo
- e) Muy en desacuerdo Excelente

5. ¿Se concluyeron con satisfacción todas las operaciones que realizó?

- a) Muy de acuerdo
- b) Algo de acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Algo en desacuerdo
- e) Muy en desacuerdo

6. ¿Cuánto tiempo te demoraste en realizar tus procesos?

- a) Menos de 1 minuto
- b) De 2 a 3 minutos
- c) De 4 a 5 minutos
- d) De 6 a 10 minutos
- e) Más de 10 minutos

7. ¿Los gastos de operación son menores con el Sistema que sin Él?

- a) Muy de acuerdo
- b) Algo de acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Algo en desacuerdo
- e) Muy en desacuerdo Excelente

8. ¿Con la Implementación del Sistema se tiene un mejor control?

- a) Muy de acuerdo
- b) Algo de acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Algo en desacuerdo
- e) Muy en desacuerdo Excelente

3.2.6.2. Encuesta a los Empleados de la Municipalidad.

1. ¿Sabía Ud. que la Municipalidad cuenta con un sistema en la unidad de personal?

- a) Definitivamente si
- b) Probablemente si
- c) Indeciso
- d) Probablemente no
- e) Definitivamente no

2. ¿Cuánto tiempo le tomaba realizar sus transacciones antes de que hubiera el sistema?

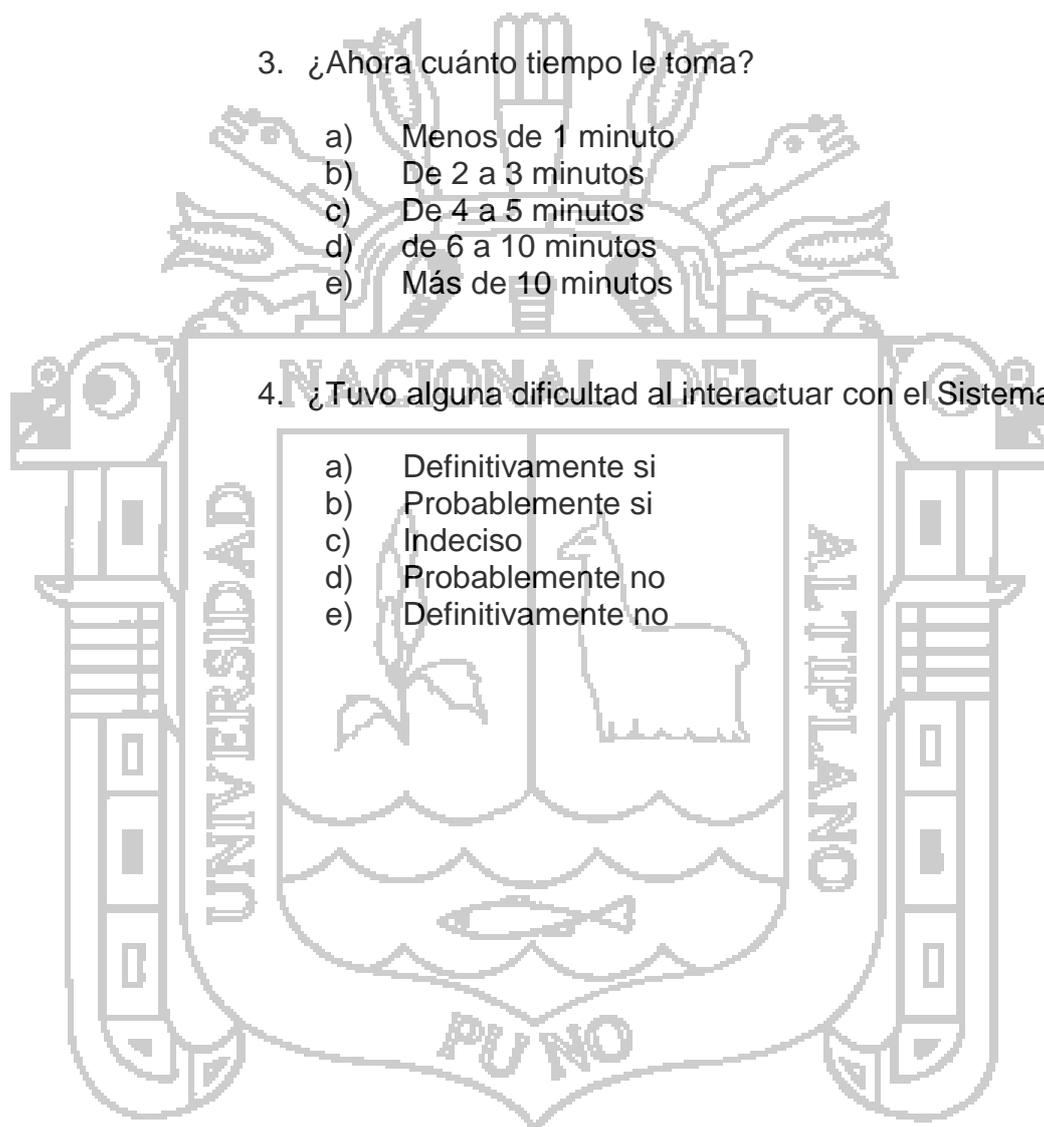
- a) Menos de 1 minuto
- b) De 2 a 3 minutos
- c) De 4 a 5 minutos
- d) De 6 a 10 minutos
- e) Más de 10 minutos

3. ¿Ahora cuánto tiempo le toma?

- a) Menos de 1 minuto
- b) De 2 a 3 minutos
- c) De 4 a 5 minutos
- d) de 6 a 10 minutos
- e) Más de 10 minutos

4. ¿Tuvo alguna dificultad al interactuar con el Sistema?

- a) Definitivamente si
- b) Probablemente si
- c) Indeciso
- d) Probablemente no
- e) Definitivamente no





RESULTADOS Y DISCUSIÓN

4.1. PRUEBA DE HIPÓTESIS

Hipótesis Nula: H_0 : El sistema de recursos humanos utilizando tecnologías de base de datos orientado a objetos BD4O, no mejora la gestión de procesos en la Municipalidad Provincial Chucuito Juli.

Hipótesis General: H_1 El sistema de recursos humanos utilizando tecnologías de base de datos orientado a objetos BD4O, mejora la gestión de procesos en la Municipalidad Provincial Chucuito Juli.

Se ha observado que el tiempo promedio en la gestión de procesos de la unidad de personal con el proceso manual es de: 10 minutos y con desviación estándar ($S= 3.07$) y con ayuda del Sistema es de 2 minutos y con desviación estándar ($S= 1.12$) siendo la diferencia notoria.

Planteamiento de hipótesis

$$H_0: u \leq 10$$

$$H_1: u > 10$$

El nivel de significancia (α) escogido para la prueba de la hipótesis será del 5%. Siendo $\alpha = 0.05$ (Nivel de Significancia) y $n-1 = 5$ grados de libertad ($n =$ número de administrativos), se tiene el valor crítico de T student:

$$\alpha = \text{significancia} = 5$$

$$\text{Ahora si tenemos valor critico } t_{\alpha=0.05} = 2.015$$

$$T = \frac{\bar{d}}{S_{\bar{d}}}$$

Dónde: d : diferencia (antes - después)

$S_{\bar{d}}$: Desviación estándar de la diferencias

$$S_{\bar{a}} = \sqrt{\frac{\sum(a-\bar{d})}{n-1}}$$

Se tiene que $T = 12.67$

Ahora comparamos $T > t_{\alpha=0.05}$, tiene un valor verdadero se rechaza H_0 y se acepta H_1 , es decir. El sistema de recursos humanos utilizando tecnologías de base de datos orientado a objetos DB4O, **optimiza** la gestión de procesos de la Municipalidad Provincial Chucuito Juli. En cuanto a tener mayor eficiencia en los procesos, mejora el tiempo de administración de cada proceso, reduce los costos en la gestión de procesos y se tiene mayor control de cada proceso que administra la unidad de personal.

Gráficamente se expresa en el siguiente cuadro:

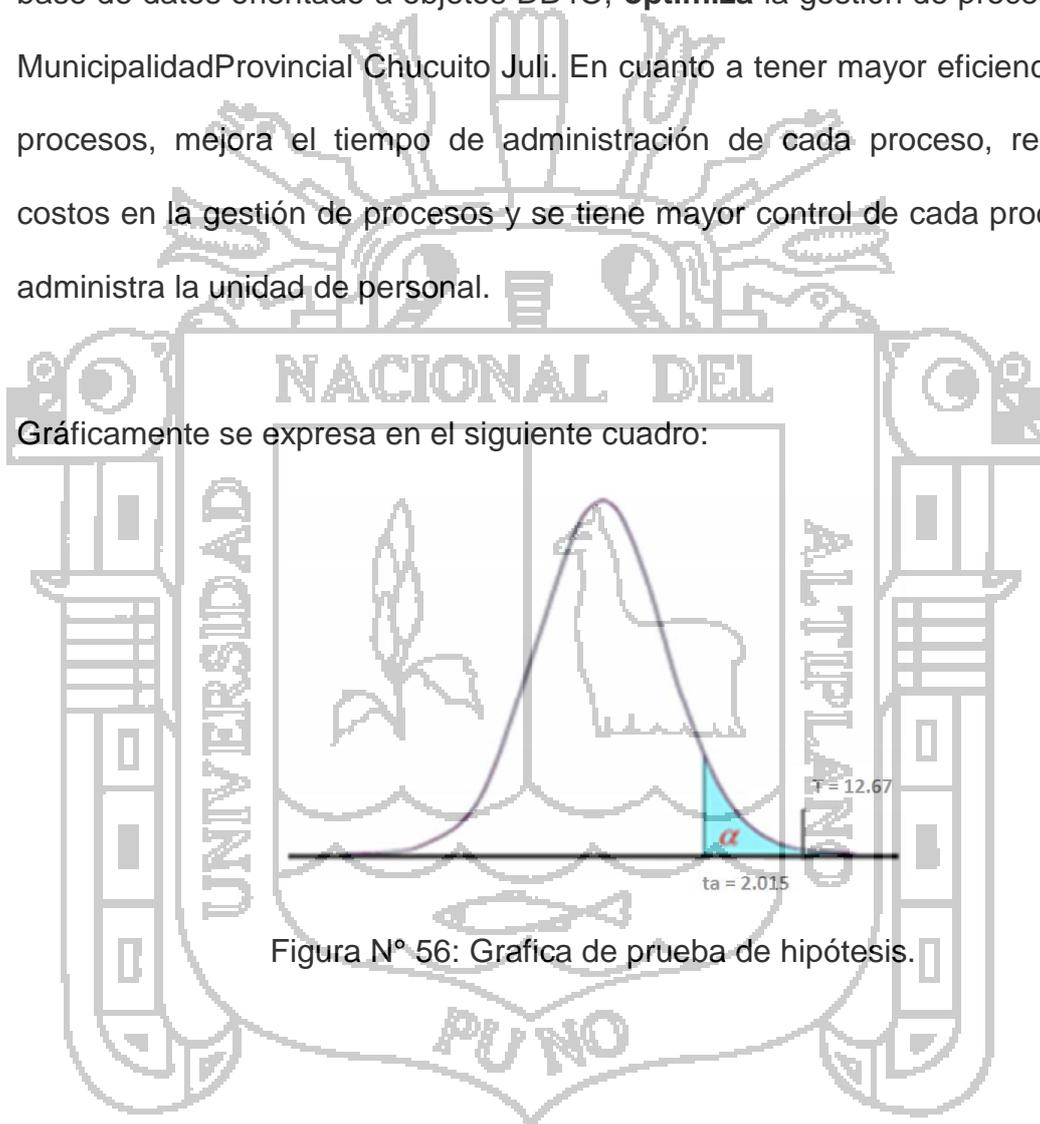


Figura N° 56: Grafica de prueba de hipótesis.

4.2. RESULTADOS DE LAS ENCUESTAS

A: Trabajadores Administrativos de la Unidad de Personal

1. ¿Con el Sistema obtuvo buenos y esperados resultados con poco trabajo?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Definitivamente si	2	2	0.7	67
Probablemente si	1	3	0.3	33
Indeciso	0	3	0	0
Probablemente no	0	3	0	0
Definitivamente no	0	3	0	0
TOTAL	3		1	100

Tabla N° 23: Resultado de encuesta - rendimiento del sistema.

Fuente: Elaboración propia.

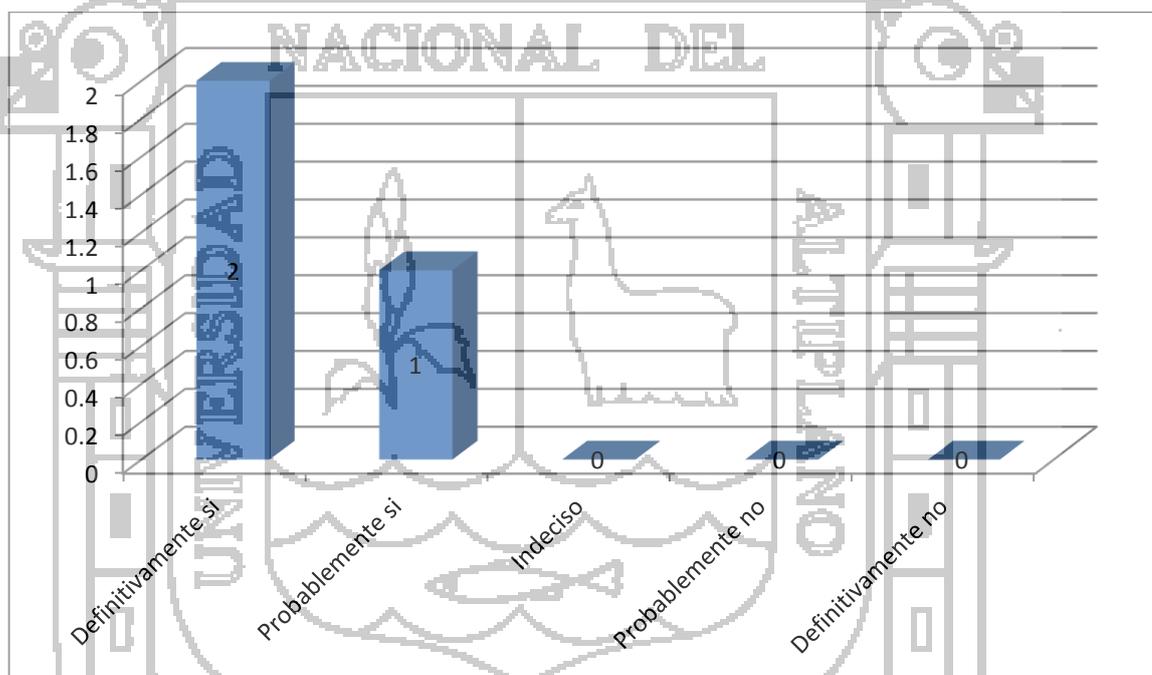


Figura N° 57: Resultado de encuesta - rendimiento del sistema.

Interpretación: De los resultados se observa que el 67% de los encuestados dicen que el sistema definitivamente si tiene un buen rendimiento y el 33% respondió que probablemente si tenga un buen rendimiento. Con esto se concluye que el Sistema es bueno en cuanto a rendimiento.

2. En cuanto a Interfaz Amigable ¿Cuál es su opinión de la interfaz del Sistema?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Muy de acuerdo	2	2	0.7	67
Algo de acuerdo	1	3	0.3	33
Ni de acuerdo ni en desacuerdo	0	3	0	0
Algo en desacuerdo	0	3	0	0
Muy en desacuerdo	0	3	0	0
Excelente				
TOTAL	3		1	100

Tabla N° 24: Resultado de encuesta - opinión con respecto a la interfaz del programa.

Fuente: Elaboración propia.

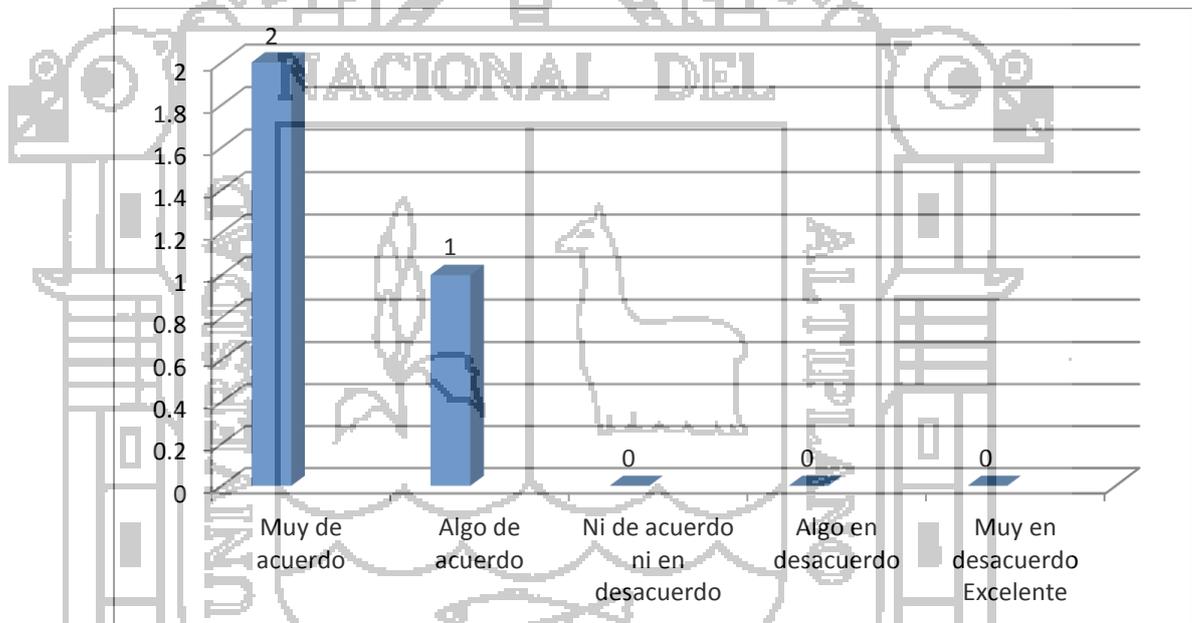


Figura N° 58: Resultado de encuesta - opinión con respecto a la interfaz del programa.

Interpretación: De los resultados se observa que el 67% de los encuestados están muy de acuerdo que el sistema tiene una interfaz amigable y el 33% les pareció algo de acuerdo con la interfaz amigable. Con esto se concluye que la interfaz del Sistema es agradable para el Usuario en cuanto a su interfaz.

3. En cuanto a la Facilidad de Uso del Sistema ¿Encuentra de fácil uso el sistema?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Definitivamente si	3	3	1	100
Probablemente si	0	3	0	0
Indeciso	0	3	0	0
Probablemente no	0	3	0	0
Definitivamente no	0	3	0	0
TOTAL	3		1	100

Tabla N° 23: Resultado de encuesta - opinión con respecto a la usabilidad del sistema.

Fuente: Elaboración propia.



Figura N° 59: Resultado de encuesta - opinión con respecto a la usabilidad del sistema.

Interpretación: De los resultados se observa que el 100% de los encuestados les pareció que definitivamente si es fácil usar el Sistema. Con esto se concluye que el Sistema es muy fácil de usar.

4. ¿El sistema tiene sus capacidades de realizar sus funciones sin incidentes?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Muy de acuerdo	3	3	1	100
Algo de acuerdo	0	3	0	0
Ni de acuerdo ni en desacuerdo	0	3	0	0
Algo en desacuerdo	0	3	0	0
Muy en desacuerdo	0	3	0	0
Excelente	0	3	0	0
TOTAL	3		1	100

Tabla: N° 26: Resultado de encuesta - opinión con respecto a la confiabilidad del sistema.

Fuente: Elaboración propia.



Figura: N° 60: Resultado de encuesta - opinión con respecto a la confiabilidad del sistema.

Interpretación: De los resultados se observa que el 100% de los encuestados están muy de acuerdo con la confianza al manipularlo. Con esto se concluye que el Sistema es confiable.

5. ¿Se concluyeron con satisfacción todas las operaciones que realizó?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Muy de acuerdo	3	3	1	100
Algo de acuerdo	0	3	0	0
Ni de acuerdo ni en desacuerdo	0	3	0	0
Algo en desacuerdo	0	3	0	0
Muy en desacuerdo	0	3	0	0
Excelente	0	3	0	0
TOTAL	3		1	100

Tabla N° 27: Resultado de encuesta - opinión con respecto a la eficiencia del sistema.

Fuente Elaboración propia.



Figura N° 61: Resultado de encuesta - opinión con respecto a la eficiencia del sistema.

Interpretación: De los resultados se observa que el 100% de los encuestados están muy de acuerdo con satisfacción sus operaciones en el Sistema. Con esto se concluye que el Sistema es eficaz.

6. ¿Cuánto tiempo te demoraste en realizar tus procesos?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Menos de 1 minuto	2	2	0.7	67
De 2 a 3 minutos	1	3	0.3	33
De 4 a 6 minutos	0	3	0	0
De 6 a 10 minutos	0	3	0	0
De 10 a más minutos	0	3	0	0
TOTAL	3		1	100

Tabla N° 28: Resultado de encuesta - opinión con respecto al tiempo de respuesta.

Fuente Elaboración propia.

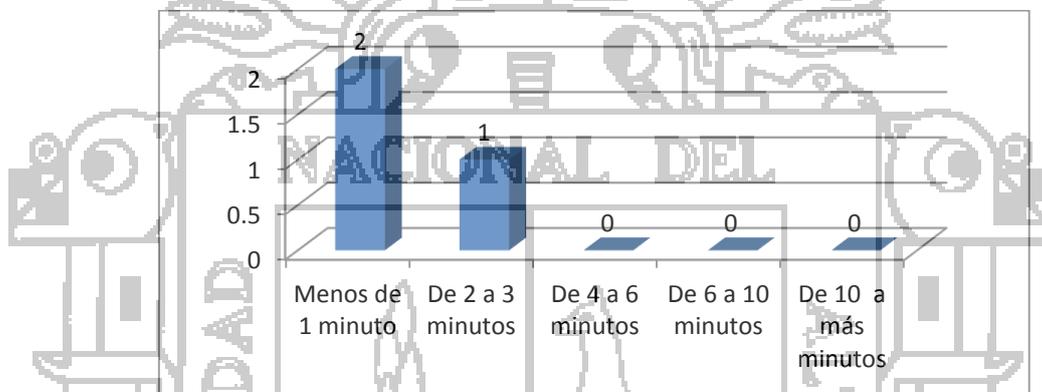


Figura N° 62: Resultado de encuesta - opinión con respecto al tiempo de respuesta.

Interpretación: De los resultados se observa que el 67% de los encuestados concluyeron sus operaciones en el Sistema en menos de 1 minuto, el 33% de 2 a 3 minutos. Con esto se concluye que el Sistema realiza sus procesos en menos tiempo.

7. ¿Los gastos de operación son menores con el sistema que sin él?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Muy de acuerdo	3	3	1	100
Algo de acuerdo	0	3	0	0
Ni de acuerdo ni en desacuerdo	0	3	0	0
Algo en desacuerdo	0	3	0	0
Muy en desacuerdo	0	3	0	0
Excelente				
TOTAL	3		1	100

Tabla N° 29: Resultado de encuesta - opinión con respecto al costo.

Fuente Elaboración propia.

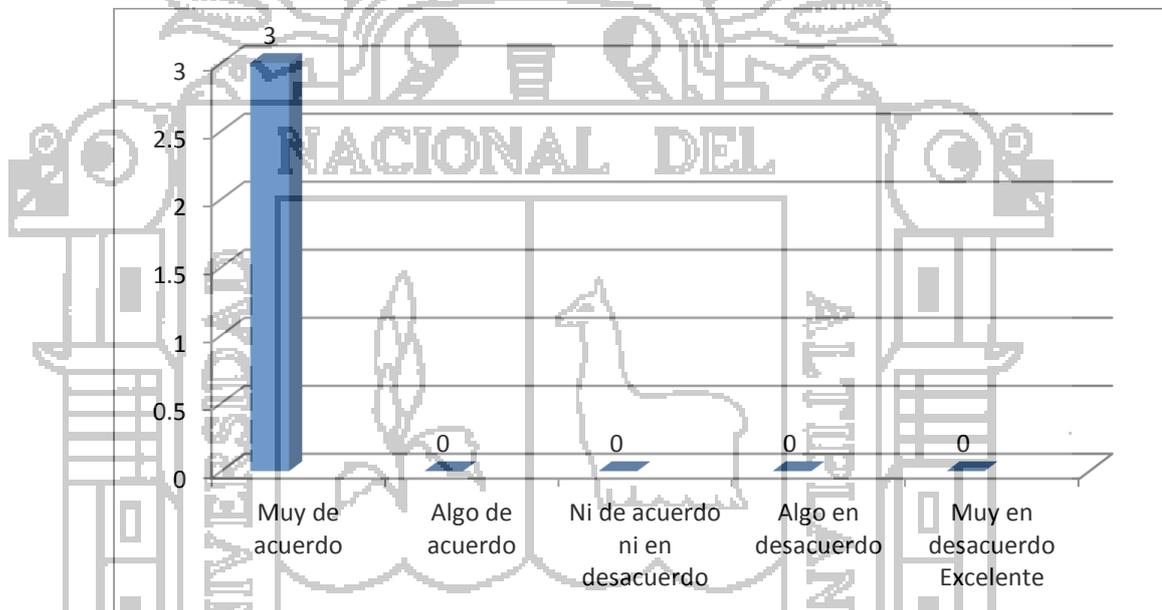


Figura N° 63: Resultado de encuesta - opinión con respecto al costo.

Interpretación: De los resultados se observa que el 100% de los encuestados dijeron que se redujo los costos en las operaciones con el Sistema. Con esto se concluye que el Sistema redujo Costos.

8. ¿Con la implementación del sistema se tiene más control?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Muy de acuerdo	3	3	1	100
Algo de acuerdo	0	3	0	0
Ni de acuerdo ni en desacuerdo	0	3	0	0
Algo en desacuerdo	0	3	0	0
Muy en desacuerdo	0	3	0	0
Excelente				
TOTAL	3		1	100

Tabla N° 30: Resultado de encuesta - opinión con respecto a la gestión.

Fuente: Elaboración propia.

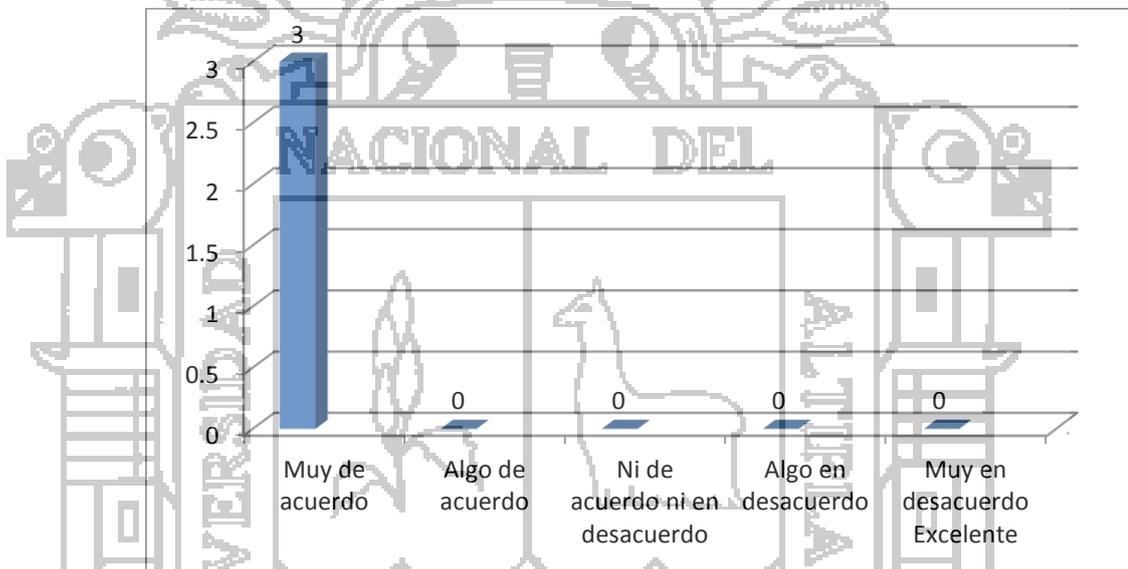


Figura N° 64: Resultado de encuesta - opinión con respecto a la gestión.

Interpretación: De los resultados se observa que el 100% de los encuestados dijeron que se tiene mejor control con el Sistema. Con esto se concluye que el Sistema realiza una buena Gestión.

B: Empleados:

1. ¿Sabía Ud. que la Municipalidad cuenta con un Sistema en la unidad de personal?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Definitivamente si	4	4	0.08	8
Probablemente si	16	20	0.32	32
Indeciso	20	40	0.4	40
Probablemente no	8	48	0.16	16
Definitivamente no	2	50	0.04	4
TOTAL	50		1	100

Tabla N° 31: Resultado de encuesta - opinión con respecto al sistema.

Fuente: Elaboración propia.

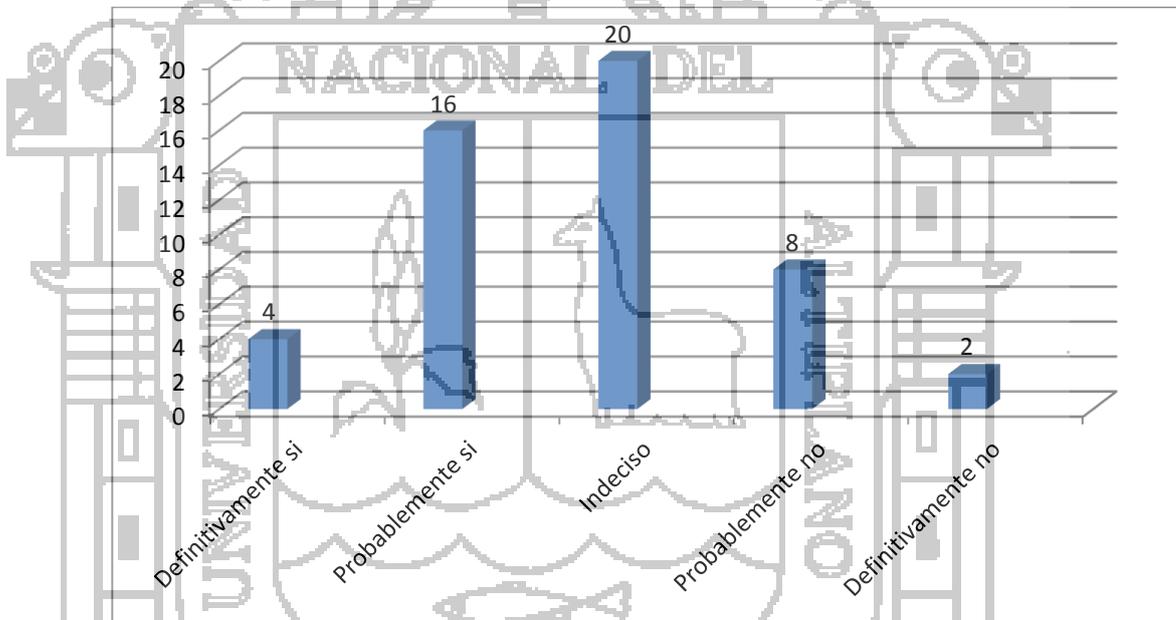


Figura N° 65: Resultado de encuesta - opinión con respecto al sistema.

Interpretación: De los resultados se observa que el 8% de los encuestados definitivamente si sabía de la existencia de un sistema dentro de la unidad de personal, el 32% respondió probablemente si, el 40% contestó indeciso, el 16% respondió probablemente no y 4% definitivamente no sabía de la existencia de un sistema en la unidad de personal.

2. ¿Cuánto tiempo le tomaba realizar sus transacciones antes de que hubiera el sistema?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Menos de 1 minuto	2	2	0.04	4
De 2 a 3 minutos	5	7	0.1	10
De 4 a 6 minutos	8	15	0.16	16
De 6 a 10 minutos	25	40	0.5	50
De 10 a más minutos	10	50	0.2	20
TOTAL	50		1	100

Tabla N° 32: Resultado de encuesta - opinión con respecto al tiempo sin sistema.

Fuente: Elaboración propia.

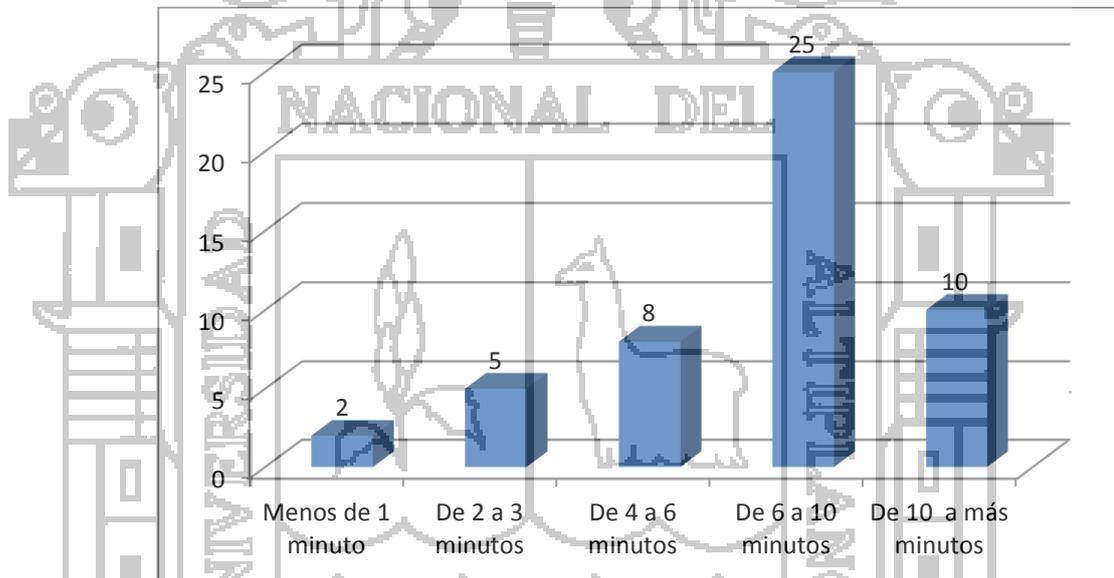


Figura: N° 66: Resultado de encuesta - opinión con respecto al tiempo sin sistema.

Interpretación: De los resultados se observa que el 4% de los encuestados dijeron menos de 1 minuto, el 10% de 2 a 3 minutos, el 16% de 4 a 5 minutos, el 50 % de 6 a 10 minutos y el 20% más de 10 minutos.

3. ¿Ahora cuánto tiempo le toma?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Menos de 1 minuto	9	9	0.18	18
De 2 a 3 minutos	35	44	0.7	70
De 4 a 6 minutos	4	48	0.08	8
De 6 a 10 minutos	1	49	0.02	2
De 10 a más minutos	1	50	0.02	2
TOTAL	50		1	100

Tabla N° 33: Resultado de encuesta - opinión con respecto al tiempo con el sistema.

Fuente: Elaboración propia.

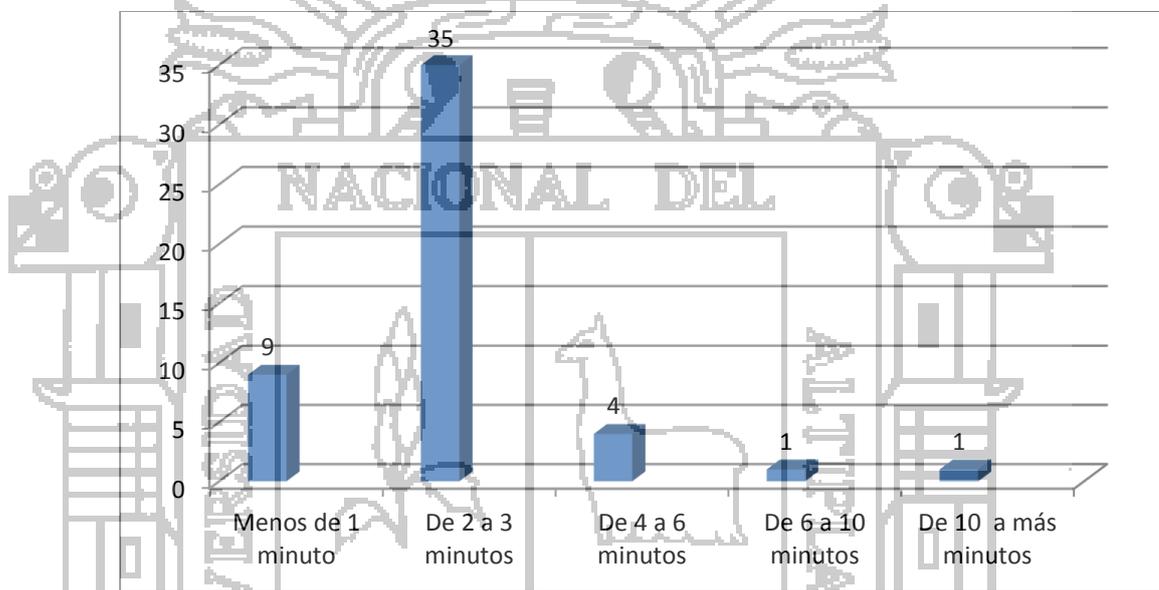


Figura N° 67: Resultado de encuesta - opinión con respecto al tiempo con el sistema.

Interpretación: De los resultados se observa que el 18% de los encuestados dijeron menos de 1 minuto, el 70 % de 2 a 3 minutos, el 8% de 4 a 5 minutos, el 2 % de 6 a 10 minutos y el 2% más de 10 minutos.

Con lo que se Concluye con comparación con la anterior pregunta que el sistema mejoro en el tiempo en realizar sus operaciones de los contribuyentes.

4. ¿Tuvo alguna dificultad al interactuar con el Sistema?

Descripción	Frecuencia absoluta	Frecuencia acumulada	Frecuencia relativa	Frecuencia relativa porcentual (%)
Definitivamente si	1	1	0.02	2
Probablemente si	1	2	0.02	2
Indeciso	2	4	0.04	4
Probablemente no	3	7	0.06	6
Definitivamente no	43	50	0.86	86
TOTAL	50		1	100

Tabla N° 34: Resultado de encuesta - opinión con respecto a la dificultad del sistema.

Fuente: Elaboración propia.



Figura N° 68: Resultado de encuesta - opinión con respecto a la dificultad del sistema.

Interpretación: De los resultados se observa que el 2% de los encuestados dijeron que definitivamente si tenían problemas al interactuar con el sistema, un 2% respondió probablemente sí, el 4% respondió indeciso, el 6% respondió que probablemente no y el 86% definitivamente no tuvo problemas al interactuar con el sistema, por tanto el sistema es de fácil usar tanto para el personal administrativo como para el empleado de la Municipalidad.

CONCLUSIONES

PRIMERA:El correcto requerimiento de los requisitos iniciales nos especifica el correcto funcionamiento lógico de cada proceso, así se tendrá una mejor y mayor gestión de procesos en la administración de recursos humanos en la Unidad de Personal dentro de la Municipalidad.

SEGUNDA:El Sistema basado en la Plataforma JEE sobre un motor de base de datos orientado a objetos DB4O, automatiza en un 85% en la gestión de procesos de la unidad de personal de la Municipalidad Provincial Chucuito Juli, en relación a la administración de procesos de una situación inicial sin el sistema.

TERCERO:El sistema **mejora** la gestión de procesos en la administración de recursos humanos de la Municipalidad Provincial Chucuito Juli. En cuanto a tener mayor eficiencia en los procesos, mejora el tiempo de administración de cada proceso, reduce los costos en la gestión de procesos y se tiene mayor control de cada proceso que administra la unidad de personal.

RECOMENDACIONES

PRIMERA: Se recomienda realizar las entrevistas, encuestas entre otros datos para la recolección de datos, así se puedan establecer los requerimientos iniciales, a fin de plantear el correcto funcionamiento de los procesos en la administración de recursos humanos de la Municipalidad.

SEGUNDA: Se recomienda el uso de entornos de desarrollo integrados (IDEs) ya que facilitan la implementación de sistemas, proveen un marco de trabajo amigable además de la interacción con el lenguaje.

TERCERO: Se recomienda que al desarrollar un Sistema es necesario seguir un proceso de planificación para que estos contribuyan a actividades ordenadas y con un propósito para luego realizar las pruebas a través de la opinión de los usuarios potenciales del sistema.



BIBLIOGRAFÍA

- AGÜERO, M. (2007). *Introducción a Spring*. Recuperado el 07 de mayo de 2014 de:http://palermo.edu/ingenieria/downloads/introduccion_framework_v1.0.pdf
- ARI JAAKSI, (2008). Extraído del libro 'A MethodFoz tour Object-Oriented Project' (Un método para tu propio Proyecto Orientado a Objetos), Volumen 9, Nro 10.
- ARIAS, G., LOPEZ, F.& HEREDIA, V. (1999) Administración de Recursos Humanos para el Alto desempeño. 5ª. Ed. México, Trillas.
- AVILA, R. (2001). Metodología de la investigación. Lima: Estudios y ediciones R.A.
- BAUER, C. &. (2004). *Hibernate in Action* (2da ed.). United States of America: Manning Publications Co.
- BISQUERRA, R. (2004). Metodología de la Investigación (3ra ed.). Madrid: La muralla.
- CALDERÓN, J. L. & RODRÍGUEZ, C. (2010). Proyectos de Desarrollo de Sistemas de Información - El Usuario Final como Factor Clave de Éxito. Eighth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2010) "Innovation and Development for the Americas", June 1-4, 2010, Arequipa, Perú. Recuperado el 07 de mayo de 2014 de:http://www.laccei.org/LACCEI2010-Peru/published/IT027_Calderon.pdf
- CARRILLO, I. P. (2008). Metodología de desarrollo de software. Buenos Aires: Ciencia que ladra.
- CASILLAS, R. (2004). Desarrollo de aplicaciones web. Barcelona: Eureka.
- CASILLAS, R. (2005). Base de Datos. Barcelona: Eureka.
- CAZAU, P. (2006). Introducción a la investigación. Buenos Aires: Ciencia que ladra.
- DBSHARDS. (2009). Data Access Object. Recuperado el 10 de mayo de 2014, de:<http://www.codefutures.com/data-access-object/>

- FIELDS, D. &. (2000). Web Development with JavaServer Pages (2da ed.). Estados Unidos: ManningPublications Co.
- FRENCH, WENDELL L., (1994). Administración de Recursos Humanos Tomo IV Administración de Personal de Recursos Humanos. 5ª. Ed.Mexico, Editorial Limusa, S.A. de C.V. 1994.
- GUERRERO, L. A. & PORTUGAL, R. C. & FULLER, D. A. (2009). Dpto. de Ciencias de la Computación, Universidad de Chile. Recuperado el 18 de mayo de 2014 de: <http://users.dcc.uchile.cl/~luguerre/papers/ECC-99.pdf>
- HERNANDEZ, S. F. (2006). Metodología de la Investigación (4ta ed.). México: McGrawHill.
- JACOBSON, I. B. (2000). El Proceso Unificado de Desarrollo de Software. Madrid: Pearson Education.
- KAPUR, R. B. (2010). Open Source Development (1ra ed.). Canada: IBM Canada.
- KENDALL, E. &. (2005). Análisis y Diseño de Sistemas. México: Pearson Education.
- KING, G. B. (2010). Hibernate Reference Documentation.
- LARMAN, C. (2003). UML y Patrones. Madrid: Prentice Hall.
- LAUDON, K. &. (2002). Sistemas de Información Gerencial. México: Alhambra.
- LINWOOD, J. &. (2010). Beginning Hibernate (Expert's Voice in Java (2da ed.). Estados Unidos: Apress.
- MANN, K. D. (2005). JavaServer Faces in Action.Estados Unidos: ManningPublications Co.
- MORALES, P. (13 de Diciembre de 2012). Estadística Aplicada a las Ciencias Sociales: Tamaño necesario de la muestra. Recuperado el 10 de Agosto En:<http://www.upcomillas.es/personal/peter/investigacion/Tama%F1omuestra.pdf>

MPCHJ. (2013). Portal de Transparencia de la Municipalidad Provincial Chucuito Juli.

Recuperado el 05 de mayo de 2014, En:<http://www.munijuli.gob.pe>

POSADA, N. & SOL, D. (2011). CENTIA, Universidad de las Américas-Puebla, 72820 Sta. Catarina Mártir, Puebla, México, {posada, sol} @mail.udlap.mx. recuperado en 13 de diciembre de 2013.

En:http://www.academia.edu/449893/Modelado_De_Datos_Orientado_a_Objetos_Par_a_Un_Sistema_De_Informacion_Geografica.

PRESMAN, R. (2009). Ingeniería del software: un enfoque práctico (6th ed.). Madrid: McGraw-Hill.

QUATRANI, T. (2000). Visual modeling with Rational Rose and UML. Estados Unidos: Technology Series.

QUINTAN, Y., CAMEJO, L. & DÍAZ, A. (2008). Diseño de la Base de Datos para Sistemas de Digitalización y Gestión de Medias. Universidad de las Ciencias Informáticas. Carretera San Antonio Km 1 1/2 Boyeros, Ciudad la Habana, Autor para la correspondencia: yqrondon@uci.cu. Recuperado De: <http://laboratorios.fi.uba.ar/lie/Revista/Articulos/080815/A3mar2011.pdf>.

ROLDÁN, D. V. (2010). Aplicaciones Web: Un Enfoque Práctico. México: ALFAOMEGA RA-MA.

SÁNCHEZ, M. & BARBERAN, A. (2005). El lenguaje de programación java, edición N° 1, Editorial: Universidad Politécnica de Valencia, ISBN: 9788477215264

SILVA, D. A. & MERCERAT, B. (2012). LIFIA, Laboratorio de Investigación y Formación en Informática Avanzada. Facultad de Informática, Universidad Nacional de La Plata. Calle 50 esq. 115 (1900), La Plata, Provincia de Buenos Aires, República Argentina. {dsilva,bmercerat}@lifia.info.unlp.edu.ar.

Recuperado el 27 de mayo de

2014 de: http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf

SOMMERVILLE, Ian. (2011). "Ingeniería del Software", Madrid. ISBN: 84-7829-074-5.

TAPIA, S. (2011). Integración JSF - Spring - Hibernate. Quito Ecuador.

WALLS, C. (2011). Spring (3ra ed.). Estados Unidos: Anaya Multimedia.

WELLS, D. (2009). Extreme Programming XP. Recuperado el 01 de 08 de 2013, de

<http://www.extremeprogramming.org/> ZAMBON, G. &. (2007). Beginning JSP™,

JSF™, and Tomcat Web Development: From Novice to Professional (4ta ed.).

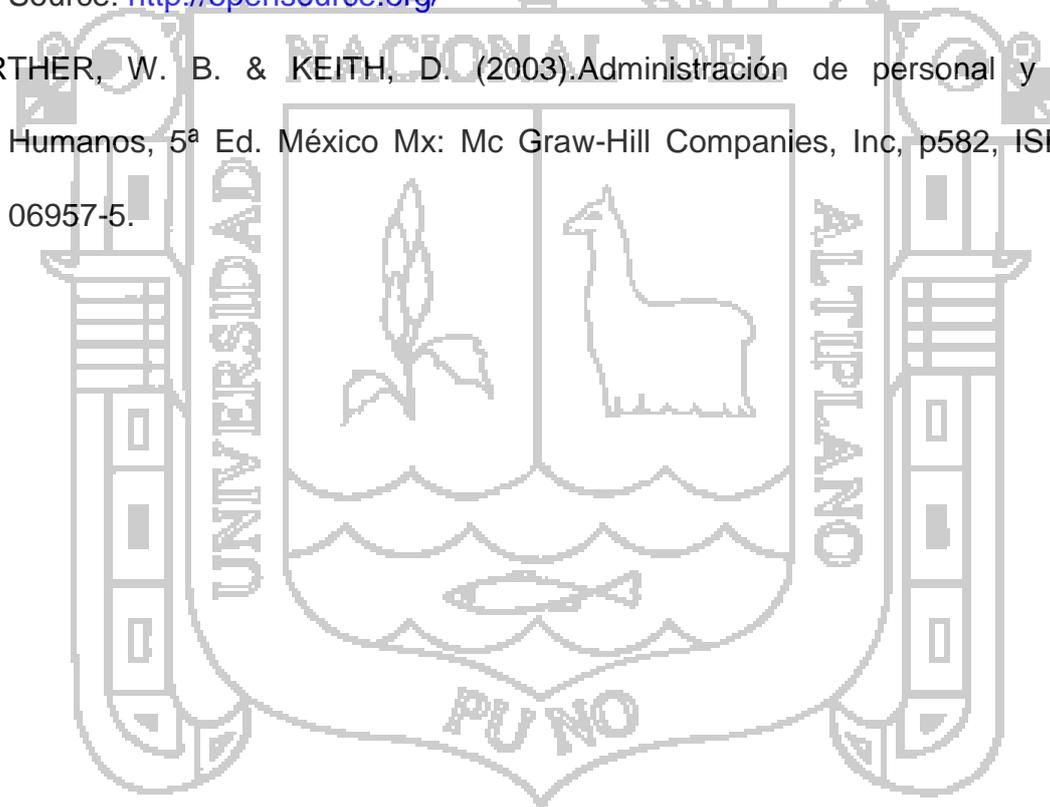
Estados Unidos: Apress. (2000). Recuperado el 01 de 08 de 2013, de Open

Source: <http://opensource.org/>

WERTHER, W. B. & KEITH, D. (2003). Administración de personal y Recurso

Humanos, 5ª Ed. México Mx: Mc Graw-Hill Companies, Inc, p582, ISBN 0-07-

06957-5.



ANEXOS

ANEXO 1: TARJETA DE EMPLEADO



ANEXO 2: PLANILLA DE REMUNERACIONES

PLANILLA DE REMUNERACIONES, REGIMEN ESPECIAL DE CONTRATO ADMINISTRATIVO DE SERVICIOS (D. Leg.Nº1057) DECRETO LEGISLATIVO 1057 PLANILLA Nº 405 MES DE ABRIL DEL 2014															
FUNCION : 17 MEDIO AMBIENTE : 039 MEDIO PROGRAMA : 0086 LIMPIEZA PUBLICA SUB PROGRAMA : 1000584 MANEJO DE RESIDUOS SOLIDOS MUNICIPALES ACTIVIDAD : martes, 22 de julio de 2014 FECHA DE ELABORACION															
N°	APELLIDOS Y NOMBRES	RUC	DIAS LABORADOS	REM. SEG. CONTRATO	REMNER. MENSUAL	A.P. ESSALUD 9%	SNP 13% PRIMA	AFP INTEGRAL	AFP HABITAT	AFP PROFUTURO	DSCTO.SINDICATO	DEC. JUDICIAL	APORT. VOL.	TOTAL DESCUENTO	TOTAL LIQUIDO
01	WILSON CARDENAS PACCO Trabajador de Servicios	10123456789	30	750.00	750.00	67.50	0.00	0.00	0.00	95.40	0.00	0.00	0.00	95.40	654.60
	SUB TOTAL			750.00	750.00	67.50	0.00	0.00	0.00	95.40	0.00	0.00	0.00	95.40	654.60
232811			750.00	ESSALUD	67.50	SNP									
232811			0.00				AFP PRIMA								
232812			67.50				AFP INTEGRAL								
							AFP HABITAT								
							AFP PROFUTURO								
							DESCTO. SINDICATO								
							FALTAS Y TARDANZA								
	TOTAL		817.50		67.50					750.00					
															0.00

ANEXO 3: BOLETA DE PAGO

MUNICIPALIDAD PROVINCIAL CHUCUITO JULI											
BOLETAS DE PAGO: CONTRATOS ADMINISTRATIVOS DE SERVICIOS (CAS) DECRETO LEGISLATIVO 1057											
UNIVERSIDAD NACIONAL DEL ALTIPLANO											
FUNCIÓN : 17 MEDIO AMBIENTE											
PROGRAMA : 039 MEDIO AMBIENTE											
SUB PROGRAMA : 0086 LIMPIEZA PÚBLICA											
ACTIVIDAD : 1000584 MANEJO DE RESIDUOS SÓLIDOS MUNICIPALES											
MES DE ABRIL DEL 2014											
N°	APELLIDOS Y NOMBRES	DÍAS LABORADOS	MONTO MENSUAL	A.P. ESSALUD 9%	SNP 13%	AFP PRIMA	AFP HORIZONTE	AFP INTEGRAL	AFP PROFUTURO	TOTAL DESCUENTO	TOTAL LIQUIDO
1	WILSON CARDENAS PACCO Trabajador de Servicios	30	750.00	67.50	0.00	0.00	0.00	0.00	95.40	0.00	654.60

ANEXO 4: RESUMEN DE EMPLEADOS PARA DECLARACIÓN AFP NET.

Nro	CUSSP	TIPO DOC	NRO. DOC	APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE	MONTO
1	45SDF578RGT5	0	44562367	CARDENAS	PACCO	WILSON	750.00
2	43GHF578RJU5	0	01734567	QUISPE	YANA	EDGAR	1000.00
3	35SG578HGU72	0	44226874	MAMANI	NINA	JAIME	900.00
4	36ER45578GL8	0	72348960	VALDES	CACEREZ	ESTHER	1500.00

