

UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



TESIS

**SISTEMA DE CONSULTA DE LA UNIDAD DE PENSIONES Y
LIQUIDACIONES DE LA UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO**

PRESENTADA POR:

Bach. NELIDA SONIA JIHUALLANCA CCOA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO

PUNO – PERÚ

2016

UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

**SISTEMA DE CONSULTA DE LA UNIDAD DE PENSIONES Y LIQUIDACIONES DE LA
UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO**

TESIS

PRESENTADA POR:

Bach. NELIDA SONIA JIHUALLANCA CCOA

Para optar el Título Profesional de:

INGENIERO ESTADÍSTICO E INFORMÁTICO

APROBADO POR:

PRESIDENTE DEL JURADO


: _____
Dr. Juan Reynaldo Paredes Quispe

PRIMER MIEMBRO


: _____
M.Sc. Leonel Coyla Idme

SEGUNDO MIEMBRO


: _____
Dr. Reynaldo Sucari Leon

DIRECTOR DE TESIS


: _____
D.Sc. Percy Huata Panca

AREA: Informática

TEMA: Base de datos y sistemas de información

PUNO – PERÚ

2016

DEDICATORIA

Mis agradecimientos a Dios por darme la vida y salud para vivir este momento Gracias Señor por llenar mi vida de tantas bendiciones! y hacer posible la realización de mi más grande anhelo. Caminé con paso firme y constante, con profunda fe, y absoluta convicción de que alcanzaría mi meta más preciada.

A mis queridos padres Julia y Jerónimo, quienes me ofrecieron su constante apoyo y aliento para seguir adelante en todas las metas que me trace.

A mis hermanos, por el respaldo moral que siempre me brinda con inmenso cariño incondicional y por la esperanza que me da al saber que superare mis logros.

Nélida

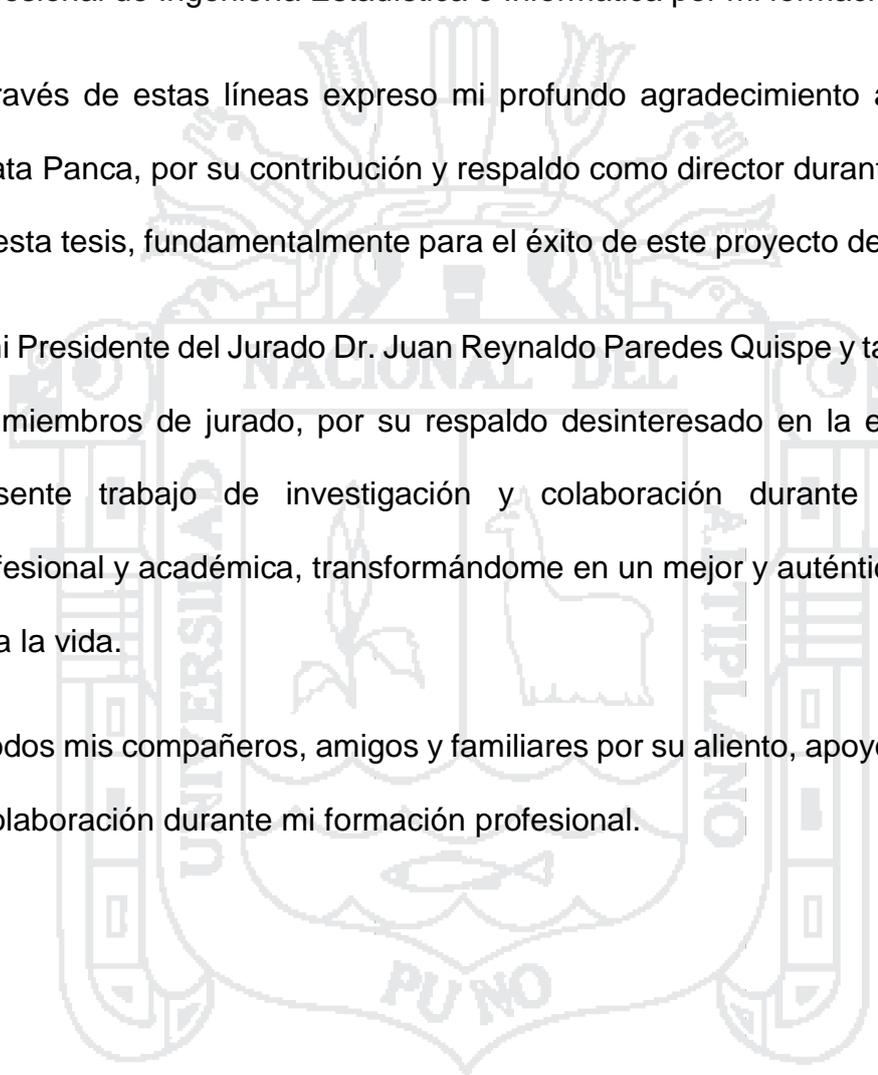
AGRADECIMIENTO

A la Universidad Nacional del Altiplano – Puno y a la Facultad de Ingeniería Estadística e Informática y en especial a todos los docentes de la Escuela profesional de Ingeniería Estadística e Informática por mi formación profesional.

A través de estas líneas expreso mi profundo agradecimiento al D.Sc. Percy Huata Panca, por su contribución y respaldo como director durante el desarrollo de esta tesis, fundamentalmente para el éxito de este proyecto de investigación.

A mi Presidente del Jurado Dr. Juan Reynaldo Paredes Quispe y también a todos los miembros de jurado, por su respaldo desinteresado en la elaboración del presente trabajo de investigación y colaboración durante mi formación profesional y académica, transformándome en un mejor y auténtico ser humano para la vida.

A todos mis compañeros, amigos y familiares por su aliento, apoyo incondicional y colaboración durante mi formación profesional.



INDICE DE CONTENIDO

RESUMEN	14
ABSTRACT	15
CAPÍTULO I EL PROBLEMA	18
1.1. Descripción del problema	18
1.2. Formulación del problema	19
1.3. Objetivos de la investigación	19
1.3.1. Objetivo general	19
1.3.2. Objetivos específicos	19
1.4. Hipótesis.....	19
1.4.1. Hipótesis general.....	19
1.4.2. Hipótesis específico	20
1.5. Justificación de la investigación	20
1.6. Delimitación de la investigación	20
CAPÍTULO II REVISIÓN DE LITERATURA	21
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	21
2.2. BASE TEÓRICA	22
2.2.1. Sistema de información.....	22

2.2.2.	Sistema de consulta.....	24
2.2.3.	Ingeniería de software.....	24
2.2.4.	Software.....	25
2.2.5.	Programación orientada a objetos.....	25
2.2.6.	Análisis y Diseño de Sistemas Orientado a Objetos.....	26
2.2.8.	El Lenguaje de Modelado Unificado UML.....	27
2.2.9.	Diagramas UML.....	28
2.2.9.1.	Diagrama de Caso de Uso.....	28
2.2.9.2.	Diagrama de Clases.....	29
2.2.9.3.	Diagrama de actividades.....	31
2.2.9.4.	Diagrama de interacción.....	32
2.2.9.5.	Diagrama de secuencia.....	32
2.2.9.6.	Diagramas de Colaboración.....	34
2.2.9.7.	Diagrama de Estados.....	34
2.2.9.8.	Diagramas de Implantación.....	34
2.2.9.9.	Diagrama de componentes.....	35
2.2.9.10.	Diagramas de despliegue.....	35
2.2.10.	Software Libre.....	36

2.2.10.1. Desarrollo de Software Libre.....	37
2.2.11. Lenguajes de Programación	37
2.2.12. Pruebas convencionales	43
2.2.13. Base de datos	44
2.2.14. Metodología XP (Extreme Programming)	44
Fases de la metodología XP	45
Fase I: ANÁLISIS.....	45
Fase II: DISEÑO	47
Fase III DESARROLLO	49
Fase IV PRUEBAS	50
2.2.15. Estándar ISO/IEC 9126.....	51
2.3. DEFINICIÓN DE TÉRMINOS BÁSICOS.....	56
2.4. OPERACIONALIZACION DE VARIABLES	58
CAPÍTULO III MATERIALES Y MÉTODOS	59
3.1. Materiales	59
3.1.1. Población	59
3.1.2. Diseño de la muestra	59
3.2. Métodos	59

3.2.1. Método de recolección de datos	59
3.2.2. Método de análisis de datos	60
CAPÍTULO IV RESULTADOS Y DISCUSIÓN	62
4.1. ANÁLISIS	62
4.1.1. Análisis de requisitos del sistema	62
4.1.2. Funcionalidades, requisito de tipo de usuario	63
4.1.2.1. En el sistema, se tiene un solo administrador y/o usuario	63
4.1.3. Casos de uso para el administrador y/o usuario	63
4.2. DISEÑO	64
4.2.1. Actor	64
4.2.2. Casos de uso	65
4.2.2.1. Caso de Uso Para el Administrador y/o usuario	65
4.2.3. Diagrama de clases	67
4.2.3.1. Diagrama de Clases Para el Administrador y/o usuario	68
4.2.4. Diagrama de Actividades	71
4.2.4.1. Diagrama de Actividades Para el Administrador y/o usuario	71
4.2.5. Diagrama de integraciones administrador	74
4.2.5.1. Diagrama de interacciones de Registro de perfiles	74

4.2.6.	Diagrama de secuencias de una consulta de perfil	75
4.2.6.1.	Diagrama de secuencia para el sistema de envi3 de archivo	76
4.2.7.	Diagrama de colaboraci3n de una consulta de perfil	76
4.2.8.	Diagrama de componentes	76
4.2.8.1.	Sistema	77
4.2.9.	Base de Datos	77
4.2.10.	Arquitectura de datos del sistema	78
4.2.11.	modelamiento y funciones del sistema	79
4.2.12.	Tablas de Archivos	80
4.2.12.1.	Tabla Para el Administrador y/o Usuario	80
4.2.12.2.	Tabla Para Personales	80
4.2.12.3.	Tabla Para Perfiles y Constancias	81
4.2.13.	Arquitectura de Datos funcionales del Sistema	81
4.2.13.1.	Tabla Administrador	81
4.2.13.2.	Tabla Personales	82
4.2.13.3.	Tabla Perfiles y Constancias	84
4.2.14.	Modelo Entidad Relaci3n	84
4.3.	DESARROLLO	85

4.3.1. Generación y proceso de formularios en CakePHP	86
4.3.2. Pantalla principal o Índice	86
4.3.3. Ventana de acceso.....	87
4.3.4. Ventana de inscripción de nuevos usuarios	87
4.3.5. Ventana de registro de nuevos perfiles	88
4.3.6. Ventana para subir archivos de perfiles o constancias	88
4.3.7. Reporte final del sistema.....	89
4.3.8. Base de datos	89
4.3.9. Implementación.....	91
4.1. PRUEBA.....	92
CONCLUSIONES	95
RECOMENDACIONES	96
REFERENCIAS	97
ANEXOS	99

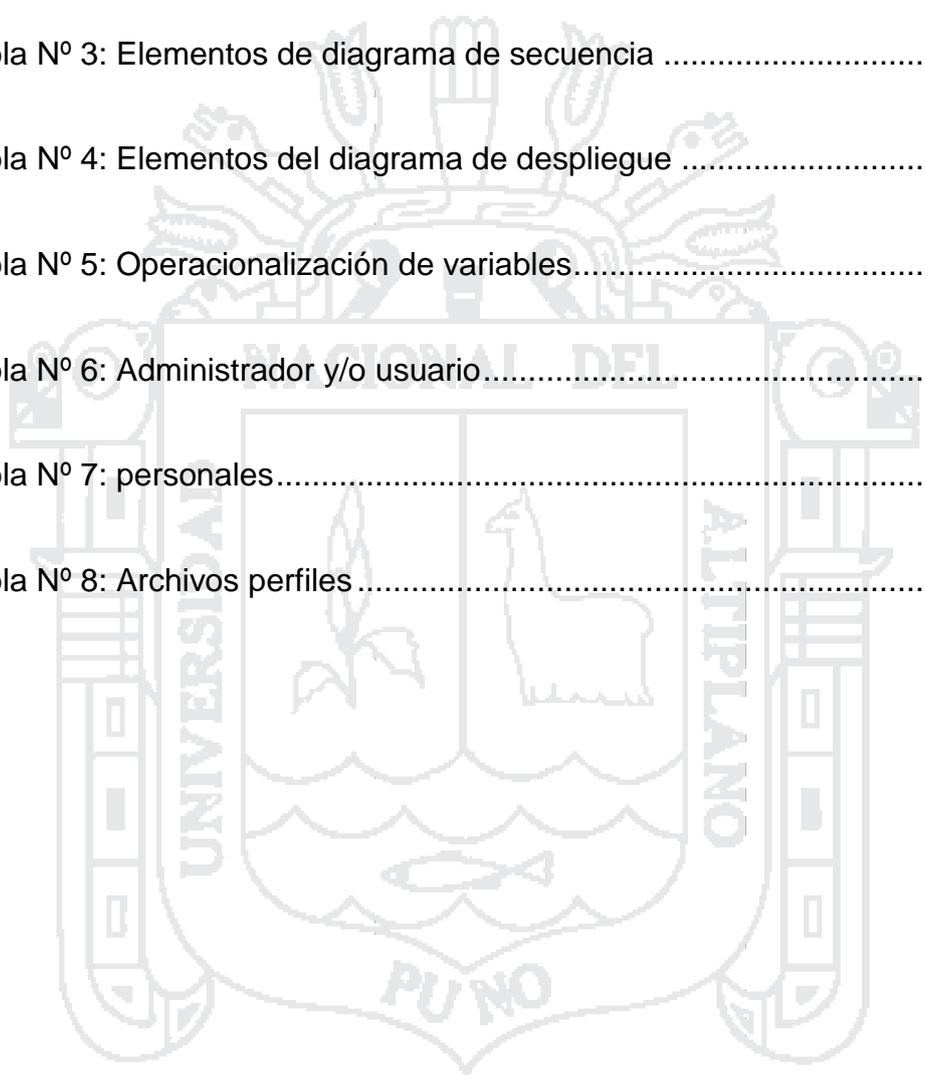
ÍNDICE DE FIGURAS

Figura N° 1: Una petición Modelo Vista Controlador básica.....	42
Figura N° 2: Fases de la Metodología Extrema.....	45
Figura N°3 Norma de evaluación ISO/IEC 9126	52
Figura N° 4: Diseño de investigación	60
Figura N° 5: Diagrama de caso de uso para los usuarios.	64
Figura N° 6: Diagrama de casos del sistema de consulta	64
Figura N° 7: diagrama de casos de uso para el administrador y/o usuario	66
Figura N° 8: Diagrama de clases para el administrador y/o usuario	67
Figura N° 9: Diagrama de clases login de administrador y/o usuario	68
Figura N° 10: Diagrama de clases para registrar nuevo perfil.....	69
Figura N° 11: Diagrama de clases para modificar y actualizar perfiles	69
Figura N° 12: Diagrama de clases para los mantenimientos.....	70
Figura N° 13: Diagrama de clases para el reporte final de constancias	71
Figura N°14: Diagrama de actividades para el administrador y/o usuario.....	72
Figura N°15: Diagrama de actividades, Actualizar Registros administrador y/o Usuario.....	72
Figura N° 16: Diagrama de actividades para actividades de mantenimiento ...	73
Figura N° 17: Diagrama de actividades, para reporte final de constancias	73
Figura N° 18: Diagrama de interacciones (Administrador)	74
Figura N° 19: Interacción en el registro de Datos de Perfiles.....	75
Figura N° 20: Diagrama de secuencias para una consulta Web	75

Figura N° 21: Diagrama de Secuencias para el sistema de Envío de archivos	76
Figura N° 22: Diagrama de Colaboración consulta de perfil de un trabajador..	76
Figura N° 23: Componentes del Sistema de Información	77
Figura N°24: Módulos del sistema	77
Figura N° 25: Arquitectura del sistema.....	79
Figura N°26: Modelamiento y funciones del sistema	79
Figura N° 27: Modelo Entidad Relación (E-R).....	85
Figura N° 28: ventana principal.....	86
Figura N° 29: Ventana de Acceso	87
Figura N° 30: Sub-sistema de Inscripción de nuevos usuarios.....	87
Figura N° 31: Ventana de registro de nuevos perfiles.....	88
Figura N° 32: Ventana para adjuntar archivos	88
Figura N° 33: Base de datos 1FN	90
Figura N° 34: Base de datos 2FN	90
Figura N° 35: Base de Datos 3FN.....	91

ÍNDICE DE TABLAS

Tabla N° 1: Relación entre clases	31
Tabla N° 2: Elementos de diagramas de despliegue	32
Tabla N° 3: Elementos de diagrama de secuencia	33
Tabla N° 4: Elementos del diagrama de despliegue	36
Tabla N° 5: Operacionalización de variables.....	58
Tabla N° 6: Administrador y/o usuario.....	80
Tabla N° 7: personales.....	80
Tabla N° 8: Archivos perfiles.....	81



ÍNDICE DE ACRÓNIMOS

SI	Sistema de Información
N°	Número.
UPL	Unidad de Pensiones y Liquidaciones
ISO	Organización de Estándares Internacionales
MVC	Modelo Vista Controlador
SICOPE	Sistema Consulta de Perfiles
HTML	Lenguaje de Marcado de Hiper Texto
PHP	Pre-procesador de hiper-texto
RDBMS	sistema de gestión de bases de datos relacionales
MySQL	Sistema de gestión de bases de datos relacional desarrollado
GNU	No es Unix
GPL	Licencia Pública General
XAMPP	Servidor web multi plataforma constituido por un servidor http apache
WORKBENCH	Herramienta visual de diseño de bases de datos que integra desarrollo de software
CakePHP	Marco de desarrollo [framework] rápido para PHP

RESUMEN

La presente investigación titulada “SISTEMA DE CONSULTA DE LA UNIDAD DE PENSIONES Y LIQUIDACIONES DE LA UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO”, se realizó en el distrito de Puno Provincia de Puno departamento Puno, durante el periodo del año 2016. El objetivo principal de la presente investigación es evaluar la calidad de servicio en los procesos de búsqueda y consulta de perfiles a través de un sistema de información de la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno usando tecnología Web, todo esto para el beneficio de los administrativos de dicha Unidad con el fin de reducir las tareas y llevar un mejor control de los años de servicio de los trabajadores de la Universidad Nacional del Altiplano. Por las características del proyecto se utilizó la metodología de desarrollo XP (Programación Extrema); metodología ágil para el desarrollo de software, tanto para sistemas tradicionales como para sistemas web, en esta metodología se abordaron los diagramas de UML (lenguaje de modelado unificado) y otros para el desarrollo del sistema. Con la ejecución y el procesamiento de datos del nuevo sistema se logró mejorar la administración que hacen manualmente, reduciendo las tareas a menor tiempo y evitando la reincidencia de información. El objetivo que se alcanzó es brindar una calidad de servicio a los trabajadores de la Universidad Nacional del Altiplano Puno.

Palabras claves: Sistema, Analizar, Diseñar, Modelar, Implementar, reincidencia y calidad de servicio.

ABSTRACT

This research entitled "SYSTEM CONSULTATION OF PENSION AND SETTLEMENT OF NATIONAL UNIVERSITY HIGHLANDS UNIT PUNO" was held in the district of Puno Province of Puno Puno department, during the period of 2016. The main objective of this research is to evaluate the quality of service processes search and query profiles through an information system Unit Pensions and Liquidations of the Universidad Nacional del Altiplano Puno using Web technology, all to the benefit of the administration of that office in order to reduce tasks and keep better control of the years of service of employees of the National University of the Altiplano. The characteristics of the project development methodology XP (Extreme Programming) was used; agile software development, both traditional systems and for web systems, this methodology diagrams UML (Unified Modeling Language) and other system development methodology addressed. With the execution and data processing of the new system was improved administration done manually, reducing tasks less time and preventing recidivism information. The objective was achieved is to provide quality service to workers Universidad Nacional del Altiplano Puno.

Keywords: System, Analyze, design, model, implement, recidivism and service quality.

INTRODUCCIÓN

Con el transcurso del tiempo la tecnología avanza, las instituciones se sienten en la necesidad de adquirir tecnología para el mejoramiento de sus sistemas y a la vez sus procedimientos, con el fin de garantizar un buen manejo de información en la institución.

Es un hecho que las computadoras liberan al hombre de las abrumadoras tareas de efectuar rutinas masivas y les permite emplear su inteligencia en tareas más estimulantes e interesantes, cualquier organización o institución que se encuentre en el campo competitivo del mundo, tiene que tratar de elevar la calidad de servicios que origina por consiguiente un mejor servicio a los clientes, reducir costos, entre otros, en tal sentido, toda institución deberían establecer un buen control en todos sus departamentos u oficinas para obtener una mayor efectividad y buen funcionamiento de la misma.

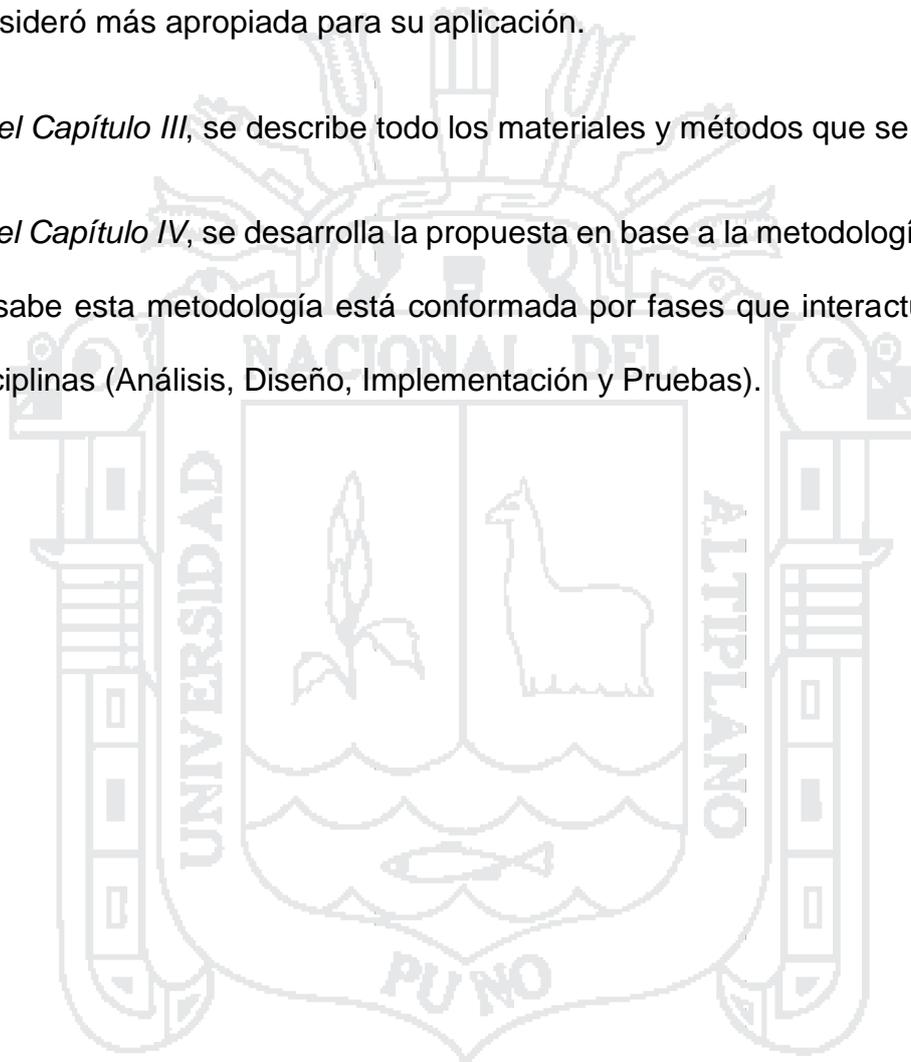
Al presentarse este panorama, la optimización en el uso de recursos económicos e intangibles, como el tiempo, se da en una escala considerable dentro de la institución. Esto no solo perjudica el logro de los objetivos y metas institucionales, sino también, a la calidad del servicio que se les ofrece a los docentes y administrativos de la Universidad Nacional del Altiplano Puno. Por eso se ve la necesidad de desarrollar el sistema de información basado en tecnología web para la Unidad de Pensiones y Liquidaciones de dicha institución. Este trabajo está estructurado de la siguiente manera.

En el Capítulo I, se describe el plan de investigación; los objetivos de la investigación; delimitaciones de la investigación.

En el Capítulo II, se establece la revisión de literatura, realizando una recopilación de antecedentes de estudio e investigación, así como la definición conceptual y la terminología empleada, correspondiente al marco metodológico, se analiza la metodología de desarrollo de software (XP Extrema) la que se consideró más apropiada para su aplicación.

En el Capítulo III, se describe todo los materiales y métodos que se utilizó.

En el Capítulo IV, se desarrolla la propuesta en base a la metodología XP. Como se sabe esta metodología está conformada por fases que interactúan con sus disciplinas (Análisis, Diseño, Implementación y Pruebas).



CAPÍTULO I

EL PROBLEMA

1.1. Descripción del problema

En la actualidad la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano – Puno (UNA - P), no cuenta con un sistema automático de consulta de perfiles (años de servicios como contratados y nombrados) de docentes y administrativos, que permita mejorar el tiempo de búsqueda y emisión de constancias. Sin este instrumento se ocasiona confusión de datos, reincidencia de información, pérdida de datos al momento de realizar el intercambio de información al utilizar dispositivos de almacenamiento, retención de constancias, no existe un registro exclusivo de control de perfiles de tiempo de años de servicio y falta de actualización automática en los equipos de la Unidad de Pensiones y Liquidaciones de la UNA – P.

1.2. Formulación del problema

¿Cómo es la calidad de servicio en los procesos de búsqueda y consulta de perfiles a través de un sistema de información en la Unidad de Pensiones y Liquidaciones de la UNA - Puno?

1.3. Objetivos de la investigación

1.3.1. Objetivo general

Evaluar la calidad de servicio en los procesos de búsqueda y consulta de perfiles a través de un sistema de información en la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno.

1.3.2. Objetivos específicos

- Analizar el tiempo de demora en la búsqueda de perfiles antes y después de la implementación de un sistema de información.
- Evaluar el tiempo de demora en la generación de constancias de años de servicio.

1.4. Hipótesis

1.4.1. Hipótesis general

La calidad de servicio en los procesos de búsqueda y consulta de perfiles a través del sistema de información en la Unidad de Pensiones y Liquidaciones es mejor.

1.4.2. Hipótesis específico

- El tiempo de demora en la búsqueda y consulta es menor después de usar un sistema de información.
- El tiempo de demora es menor en la generación de constancias.

1.5. Justificación de la investigación

El presente proyecto de investigación es de gran interés que responde a la necesidad de contar con un nuevo sistema de consulta de la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno. En cambio con la sistematización permite mejorar los problemas mencionados de la calidad de servicio al usuario cumpliendo con las normas establecidas para el manejo de los perfiles, se requiere empezar solucionando las actualizaciones de los perfiles de cada personal docente y administrativos ya que es de gran interés para el usuario porque se podrá realizar un buen reporte que será de mucha importancia para la unidad. Teniendo en cuenta las necesidades y expectativas

1.6. Delimitación de la investigación

El proyecto de sistema de consulta de la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno solo abarca la sede principal.

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Jiménez Dorila (2009), En su investigación su objetivo general señala: Desarrollar una aplicación vía intranet que permite el trámite de documentos de pago a proveedores. La metodología aplicada son las fases análisis y diseño tiene como uno de sus propósitos mejorar el proceso de trámite de los documentos de pago a proveedores y la implementación del sistema con el fin de apoyar las labores administrativas de una institución organizada en unidades como la PUCP.

Mendoza Lolimar (2011), En su investigación su objetivo general señala: Implementar un sistema automatizado que optimice la gestión de los procesos administrativos del área de servicios médicos. Concluye que la comunicación con el cliente represento una clave fundamental para poder validar los requisitos y cumplir con sus necesidades o requerimientos que se da a partir de cada una de las iteraciones a lo largo del proceso de

desarrollo, facilitando la labor de muchas tareas e impactando de manera positiva en el tiempo como análisis y diseño.

Galindo Raúl (2012), En su investigación su objetivo general señala: Implementar un sistema de información web orientado a la gestión educativa de un centro educativo especial que brinde soporte a las labores y actividades pedagógicas efectuadas por los especialistas de esta institución. Concluye que comprueba la capacidad de integración de aplicaciones construidas bajo la plataforma NET Framework con proyectos de código abierto logrando una significativa reducción de costos en la solución y cumpliendo los requerimientos no funcionales en cuanto la arquitectura.

2.2. BASE TEÓRICA

2.2.1. Sistema de información

Un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos un mejor apoyo en la toma de decisiones y desarrollo de acciones (Peña, 2006).

Un sistema de información realiza cuatro actividades básicas:

Entrada de información: Es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas.

Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras tanto que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

Procesamiento de información: Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente de información que pueden ser utilizadas para la forma de decisiones lo que hace posible, entre otras cosas que un tomador de decisiones genere una proyección financiera a partir de los datos que contienen un estado de resultados o un balance general de un año base.

Salida de información: Es la capacidad de un sistema de información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz y los plotters, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro

sistema de información o módulo. En este caso, también existe una interfaz automática de salida.

Otro autor define que "un sistema de información es el proceso útil de la manipulación de datos organizados que se emplean en las organizaciones para la toma de decisiones.

Según los autores Laudon, profesores de Administración de Empresas, un sistema de información es un organismo que recolecta, procesa, almacena y distribuye información. Son indispensables para ayudar a los gerentes a mantener ordenada su compañía, analizar nuevos productos que coloquen en un buen lugar a la organización.

2.2.2. Sistema de consulta

Las consultas son el mecanismo mediante el cual el usuario final (humano o cibernético) accede al conocimiento y a los recursos.

2.2.3. Ingeniería de software

La ingeniería de software es el establecimiento que comprende todos los aspectos de la traducción del software y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funciones eficientemente sobre máquinas reales.

Es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadoras y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software.

La ingeniería de software no solo comprende los procesos técnicos del desarrollo del software sino también con actividades tales como la gestión del proyecto y desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

2.2.4. Software

Se conoce como software al equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos del sistema, llamados hardware.

Tales componentes lógicos incluyen, entre muchos otros, aplicaciones informáticas como el procesador de textos, que permite al usuario realizar todas las tareas concernientes a la edición de textos o el software de sistema tal como el sistema (González, Marzo 2005) operativo, que, básicamente, permite al resto de los programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de las aplicaciones, proporcionando también una interfaz para el usuario.

2.2.5. Programación orientada a objetos

Es un paradigma más de programación en el que un sistema se expresa como un conjunto de objetos que interactúan entre ellos. Un paradigma de programación nos proporciona una abstracción del sistema real a algo que podemos programar y ejecutar, y puede decirse que el tipo de abstracción está directamente relacionada con los problemas que puede resolver o al menos con la facilidad con que podremos resolverlos. Mientras que el

lenguaje ensamblador es una abstracción del procesador, podríamos decir que otros lenguajes de programación como Basic o C son abstracciones del propio lenguaje ensamblador. Aunque han supuesto un importante avance sobre éste, aún obligan a los programadores a pensar en términos de la estructura del ordenador en lugar de la estructura del problema que están intentando solucionar. Según (Álvaro PEÑA, 2005).

2.2.6. Análisis y Diseño de Sistemas Orientado a Objetos

Es un enfoque cuyo propósito es facilitar el desarrollo de sistemas que deben cambiar con rapidez en respuesta a entornos de negocios dinámicos. Es difícil trabajar bien con técnicas orientadas a objetos en situaciones en las cuales los sistemas de información complicados requieren mantenimiento, adaptación y rediseño de manera continua. Los enfoques orientados a objetos utilizan el estándar de la industria para la modelación de sistemas orientados a objetos, el lenguaje unificado de modelación [UML, Unified Modeling Language], para analizar un sistema en forma de modelo de casos de uso.

La programación orientada a objetos difiere de la programación tradicional de procedimientos en que la primera examina los objetos que conforman un sistema. Cada objeto es una representación en computadora de alguna cosa o suceso real. Los objetos se representan y agrupan en clases, que son óptimas para su reutilización y mantenimiento. Una clase define el conjunto de atributos y comportamientos que comparten los objetos que ésta contiene. Según (Kendall & Kendall, 2005).

2.2.7. PhpMyAdmin

PhpMyAdmin es un programa de libre distribución en PHP, creado por una comunidad sin ánimo de lucro, que sólo trabaja en el proyecto por amor al arte. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz Web muy intuitiva. La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor Web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

2.2.8. El Lenguaje de Modelado Unificado UML

El UML (Unified Modeling Language) tiene sus orígenes en la necesidad que se habían generado en la industria para construir modelos orientados a objetos. Nace por la iniciativa de parte de Grady Booch, James Rumbaugh e Ivar Jacobson. UML es ante todo un lenguaje que se centra en representación gráfica de un sistema. Es un lenguaje visual estándar por medio de diversos elementos y procesos que interactúan de alguna forma con el software. Después de tantas discusiones fue presentado el proyecto de UML a OMG quienes adoptaron a UML como estándar en la industria del software. Según (Taboada Jiménez).

2.2.9. Diagramas UML

Un diagrama es la representación gráfica de una colección de elementos con sus relaciones, ofreciendo así una vista del sistema a modelar. Para poder representar de forma correcta un sistema, el lenguaje presenta una amplia variedad de diagramas para así visualizar el sistema desde diversas perspectivas.

2.2.9.1. Diagrama de Caso de Uso

Un diagrama de Casos de Uso muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuario u otras aplicaciones) Es una herramienta esencial para la captura de requerimientos y para la planificación y control de un proyecto interactivo. Los casos de Uso se representan en el diagrama por una elipse que denota un requerimiento solucionando por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema. Según (Rumbaugh, Jacobson, & Booch, 2004).

Elementos de Casos de Uso.

- ✓ Actor: Es un usuario del sistema, que necesita o usa alguno de los casos de uso. Un usuario puede jugar más de un rol. Un solo actor puede actuar en muchos casos de uso; recíprocamente, un caso de uso puede tener varios actores. Los actores no necesitan ser humanos pueden ser sistemas externos que necesitan alguna información del sistema actual. También se

puede encontrar tres tipos de relaciones, como son:

- ✓ **Comunica:** (comunicantes) Entre un actor y un caso de uso, denota la participación del actor en el caso de uso determinado.
- ✓ **Usa:** (use) Relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro. Se utiliza cuando se repite un caso de uso en dos o más casos de uso separados. Frecuentemente no hay actor asociado con el caso de uso común.
- ✓ **Extiende:** (extends) Relación entre dos casos, denota cuando un caso de uso es una especialización de otro. Se usa cuando se describe una variación sobre el normal comportamiento.

2.2.9.2. Diagrama de Clases

Un diagrama de clases o estructura estática de un modelo, las cosas que existen en términos de clase, su estructura interna y relaciones entre ellas, es decir las características de cada una de las clases, interfaces colaboraciones y relaciones de dependencia y generalización. Un diagrama de clases está compuesto por los siguientes elementos:

Clase: Una clase es un conjunto de objetos que comparten una estructura común y un comportamiento común.

Los atributos o características de las clases pueden ser de tres tipos, según el grado de comunicación y visibilidad de ellos con el entorno, estos son:

Públicos (+): indican que el atributo será visible tanto fuera como dentro de la clase, es decir, es accesible desde todos lados.

Privados (-): indican que el atributo solo será accesible desde dentro de la clase (solo sus métodos lo pueden acceder)

Protegidos (#) indica que el atributo no será accesible desde afuera de la clase, pero si podrá ser accesado por métodos de la clase.

Los métodos u operaciones de una clase son la forma en cómo esta interactúa con su entorno, estos pueden tener las características:

Publico (+): indican que el método será visible tanto fuera como dentro de la clase, es decir, es accesible desde todos lados.

Existen cinco tipos de relaciones diferentes entre clases: dependencia, generalización, asociación, agregación y composición.

- a. **Dependencia:** Es una relación de uso, es decir una clase usa a otra, que la necesita para su cometido. Se representa con una flecha discontinua que va desde la clase utilizadora a la clase utilizada. Con la dependencia se muestra que un cambio en la clase utilizada puede afectar el funcionamiento de la clase utilizadora, pero no al contrario.
- b. **Generalización:** Es un relación entre un elemento más general (el padre) y elemento más específico (el hijo). El elemento más específico es totalmente consistente con el elemento más general y contiene la información adicional, también se define como la herencia, donde tenemos una o varias clases padre o superclase o madre, y una clase hija o subclase. Por ejemplo, un animal es un concepto más general que un gato, un perro o un pájaro. Inversamente, un gato es un concepto más específico que un animal.

- c. **Agregación:** Es un tipo especial de asociación que representa una relación estructural entre las clases donde el llamado agregado indica el todo y el componente es una parte del mismo.
- d. **Asociación:** Relación estructural que describe un conjunto de conexiones entre objetos de forma bidireccional.
- e. **Composición:** Es un tipo de agregación donde la relación de posesión es tan fuerte como para marcar otro tipo de relación.

Relaciones entre clases.

Tabla Nº 1: Relación entre clases

Nombre	Símbolo
Asociación	
Agregación	
Composición	
Dependencia	
Generalización	

2.2.9.3. Diagrama de actividades

Permiten modelar el comportamiento de un sistema o alguno de sus elementos, mostrando la secuencia de actividades o pasos que tienen lugar para la obtención de un resultado o la consecución de un determinado objetivo. Opcionalmente, permite mostrar los flujos de información (objetos) producidos como resultado de una actividad y que serían utilizados posiblemente como entrada por la actividad siguiente:

Tabla Nº 2: Elementos de diagramas de despliegue

Nombre	Símbolo	Descripción
Acción		Nodo de actividad primitiva ejecutable de asignación o computación.
Nodo de inicio		Nodo de control que indica el inicio de un flujo de control cuando una actividad es invocada.
Nodo fin de actividad		Nodo de control que indica el fin de todos los flujos dentro de una muestra el fin de la actividad.
Flujo de control		Eje de actividades para flujo de control conecta dos acciones. Usado para indicar secuencia.
Nodo de sincronización (fork)		Nodo de control que divide un flujo en dos o más flujos concurrentes (paralelos).
Nodo de concurrencia (join)		Nodo de control que sincroniza múltiples flujos.
Nodo de decisión		Nodo de control que selecciona entre dos o más flujos de salida.

2.2.9.4. Diagrama de interacción

Estos son modelos que describen como los grupos de objetos que colaboran en algunos ambientes. Por lo general, un diagrama de interacción captura el comportamiento de un único caso de uso. Hay dos tipos de diagramas de interacción. “diagramas de secuencia y diagramas de colaboración”. Según (Rumbaugh, Jacobson, & Booch).

2.2.9.5. Diagrama de secuencia

Un diagrama de secuencia es un tipo de diagrama de interacción en el cual

se destaca el tiempo: los mensajes entre objetos vienen ordenados explícitamente por el instante en que se envían. Consta de dos ejes. Generalmente, el eje vertical es el eje del tiempo, transcurriendo éste de arriba a abajo. En el otro eje se muestran los objetos que participan en la interacción, siendo el primero de ellos el actor que inicia la ejecución de la secuencia modelada. De cada objeto parte una línea discontinua, llamada línea de la vida, que representa la vida del objeto durante la interacción. Si el objeto existe durante toda la interacción, éste aparecerá en el eje horizontal y su línea llegará hasta el final del diagrama de secuencia.

Tabla N° 3: Elementos de diagrama de secuencia

Nombre	Símbolo	Descripción
Línea de vida		Indica el periodo en que estuvo vivo el objeto durante la secuencia de actividad.
Activación		Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación.
Mensaje de un objeto a otro		El envío de mensajes entre objetos se denota mediante una línea solida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.
Mensaje a un mismo objeto		Como su nombre lo indica, es el mensaje que un objeto se envió a sí mismo.

2.2.9.6. Diagramas de Colaboración

Es una forma de representar interacción entre los objetos, es decir, las relaciones entre ellos y la secuencia de los mensajes de las iteraciones que están indicadas por un número, a diferencia de los diagramas de secuencia, pueden mostrar el contexto de la operación (cuáles objetos son atributos, cuáles temporales) y ciclos en la ejecución. Muestra como varios objetos colaboran en un solo caso de uso.

2.2.9.7. Diagrama de Estados

Muestra el conjunto de estado por los cuales pasa un objeto durante su vida en una aplicación junto con los cambios que permiten pasar de un estado a otro.

Está representado principalmente por los siguientes elementos. “Estado, elemento y transición”. Según (Rumbaugh, Jacobson, & Booch, 2004).

Estado: Identifica un período de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos.

2.2.9.8. Diagramas de Implantación

Muestran aspectos de la implementación del sistema, donde se incluyen la estructura del código fuente y su implementación en tiempo real con la estructura física del sistema. Hay dos tipos de diagramas de implementación:

2.2.9.9. Diagrama de componentes

Un diagrama de componentes muestra dependencias entre los componentes. Cada componente ofrece algunas interfaces y utiliza otras. Si las dependencias entre componentes se hacen a través de interfaces, los componentes se pueden sustituir por otros componentes que realicen las mismas interfaces. Según (Rumbaugh, Jacobson, & Booch, 2004).

2.2.9.10. Diagramas de despliegue

Son aquellos que muestran las relaciones físicas entre los componentes de software y hardware en el sistema desarrollado, es decir cómo se encuentran y se mueven los componentes y los objetos. En otras palabras, los diagramas de despliegue muestran la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software, procesos y objetos que residen en ellos.

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema mostrando la configuración de los elementos de hardware y mostrando cómo los elementos y artefactos del software se trazan en esos nodos. Según (Alvaro PEÑA, 2005).

Tabla Nº 4: Elementos del diagrama de despliegue

Nombre	Símbolo	Descripción
Nodo		Es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Utiliza para identificar cualquier servidor.
Componente		Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas.
Interface		Las interfaces se utilizan como lazo de unión entre unos componentes y otros.

2.2.10. Software Libre

El Software Libre es definido por su tipo de licenciamiento, por lo que se puede llamar “software licenciado bajo condiciones libres”. Según Hernández, J., (2005):

Un software o programa de computación cuya licencia nos permite ejercer una serie de libertades:

- a. La libertad de ejecutar el programa con cualquier propósito.
- b. La libertad de estudiar cómo funciona el programa y adaptarlo a las necesidades propias (para lo cual es una precondition el acceso al código fuente).
- c. La libertad de redistribuir copias del programa y de ese modo ayudar a otros.

- d. La libertad de mejorar el programa y liberar esas mejoras al público beneficiando así a toda la comunidad.

El software libre se basa en la cooperación y la transparencia y garantiza una serie de libertades a los usuarios. Bajo esta perspectiva el Software Libre sólo exige una cosa, en el caso de la licencia GPL: y ellas es que si el programa resultante de la modificación es distribuido, debe hacerse bajo las mismas condiciones del programa original. Las licencias que contienen esta condición son llamadas “licencias Copyleft”, y su objetivo es evitar que se distribuyan obras derivadas bajo licencias privativas.

2.2.10.1. Desarrollo de Software Libre

Las condiciones de licenciamiento de los programas libres permiten la construcción comunitaria de software. Los desarrolladores de software pueden acudir a inmensas colecciones de programas y bibliotecas altamente funcionales e intensamente probadas. Esto reduce el esfuerzo y el riesgo de desarrollo, comparado con la alternativa de empezar de cero. Usando el modo cooperativo de construcción, tan esencial al método científico, y no se limitan las posibilidades del programa a lo que pueda ocurrírsele a un grupo pequeño de usuarios.

2.2.11. Lenguajes de Programación

Un lenguaje de programación se refiere a cualquier lenguaje artificial que pueda ser empleado para definir una secuencia de instrucciones para su procesamiento por una computadora u ordenador. Por lo general, se encuentra formado por un conjunto de símbolos y reglas de tipo semánticas

y sintácticas, que permiten a los programadores definir de manera precisa acerca de qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar ante diferentes eventos.

➤ HTML

HTML significa HyperText Markup Lenguaje que en español se introduce a lenguaje de marcas de hipertexto. Es el lenguaje que más predomina en la actualidad para construir páginas web. Los documentos HTML son ficheros de texto plano que se usan la extensión htm o html

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básicos admita texto sin formato como por ejemplo el bloc de notas de Windows o Gedit de Linux. Los procesadores de texto se utilizan para escribir documentos en lenguaje HTML que posteriormente será interpretado por el programa navegador correspondiente.

➤ PHP

PHP (Hypertext Pre-processor), es un lenguaje de alto nivel ejecutado por diferentes tipos de servidores, que toman el código PHP como entrada, y crean páginas web como salida Posee variables, sentencias, condiciones, bucles y funciones. Es publicado bajo la PHP license, y la Free Software Foundation considera este tipo de licencia como software libre. El lenguaje PHP posee la característica de poder mezclarse con código HTML, es multiplataforma, tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se emplean actualmente, posee una

gran documentación en su página oficial, destacando que todas sus funciones están explicadas y ejemplificadas y permite las técnicas de la programación orientada a objetos.

➤ **MySQL**

Es un sistema para la administración de datos relacionados (RDBMS) rápido y sólido. Las bases de datos permiten almacenar, buscar, ordenar y recuperar datos de forma eficiente. El servidor de MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios, para pronosticar acceso a dichos datos y para asegurarse de sólo obtienen acceso a ellos los usuarios con autorización. Por lo tanto, MySQL es un servidor multiusuario y de subprocesamiento múltiple. Utiliza SQL (del inglés Structured Query Language de consulta de bases de datos utilizados en todo el mundo).

MySQL se distribuye bajo un sistema de licencias dual puede utilizarlo bajo una licencia de código abierto (GNU GPL), que es gratuita mientras cumpla las condiciones de la misma. Si desea distribuir una aplicación que no sea GPL y que incluya MySQL, puede adquirir una licencia comercial. Según (welling & Thomson, 2005).

➤ **XAMPP**

Es un servidor independiente de plataforma de código libre, te permite instalar de forma sencilla Apache en tu propio ordenador sin importar tu sistema operativo y lo mejor es de uso gratuito XAMPP incluye servidores de bases de datos como MySQL con sus respectivos gestores

phpMyAdmin y phpSQLiteAdmin, etc, entre muchas cosas más. También es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programa por ejemplo) en tu propio ordenador sin necesidad de tener que acceder a internet. (Ben, Laurie., 2005)

➤ **MySQL WORKBENCH**

Es una herramienta visual de diseño de bases de datos que integra y desarrollo el software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL

➤ **Sublime text**

Es un editor de código multiplataforma, ligero y con pocas concesiones a las florituras. Es una herramienta concebida para programar sin distracciones. Su interfaz de color oscuro y la riqueza de coloreado de la sintaxis, centra nuestra atención completamente. Permite tener varios documentos abiertos mediante pestañas, e incluso emplear varios paneles para aquellos que utilicen más de un monitor. Dispone de modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla.

➤ **CakePHP**

Es un marco de desarrollo (framework) rápido para PHP, libre de código abierto trata de una estructura que sirve de base a los programadores para que estos puedan crear aplicaciones web, nuestro principal objetivo es que puedas trabajar de forma estructurada y rápida, sin perder la flexibilidad,

utilizando el patrón de diseño MVC (Modelo Vista Controlador). La aplicación está en tres partes principales:

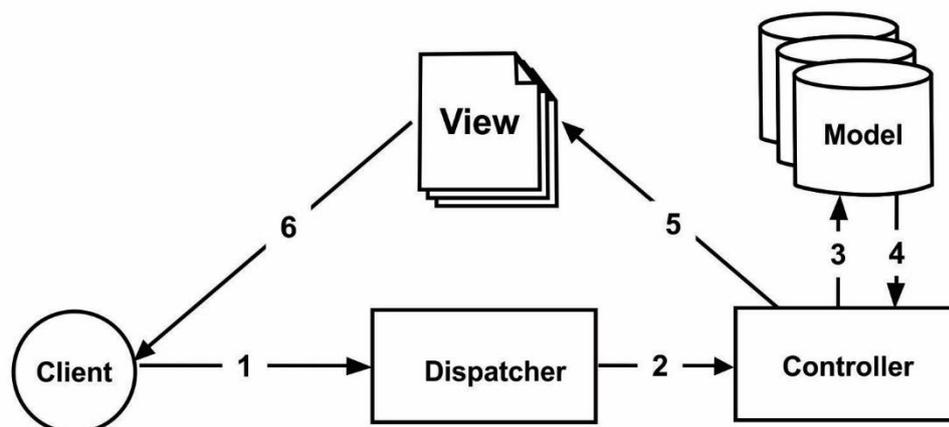
- ✚ La capa MODELO
- ✚ La capa VISTA
- ✚ La capa CONTROLADOR

MVC: Porque es un patrón de diseño de software probado y se sabe que funciona. Con MVC la aplicación se puede desarrollar rápidamente, de forma modular y mantenible. Separar las funciones de la aplicación en modelos, vistas y controladores hace que la aplicación sea muy ligera. Estas características nuevas se añaden fácilmente y las antiguas toman automáticamente una forma nueva

El diseño modular permite a los diseñadores y a los desarrolladores trabajar conjuntamente, así como realizar rápidamente el prototipo. Esta separación también permite hacer cambios en una parte de la aplicación sin que las demás se vean afectadas.

Entendiendo Modelo-Vista-Controlador: Las aplicaciones CakePHP bien escritas siguen el patrón de diseño de software MVC (Modelo-Vista-Controlador). Programar utilizando MVC consiste en separar la aplicación en tres partes principales. El modelo representa los datos de la aplicación, la vista hace una presentación del modelo de datos, y el controlador maneja y enrruta las peticiones [requests] hechas por los usuarios.

Figura N° 1: Una petición Modelo Vista Controlador básica



Estructura de archivos de CakePHP

Tras descargar y extraer CakePHP, estos serán los ficheros y carpetas que deberías ver:

- App
- Cake
- Vendors
- Htaccess
- index.php
- README

Observarás 3 carpetas principales:

- **La carpeta app:** Será donde haremos nuestra magia: es donde se ubicarán los ficheros de tu aplicación.
- **La carpeta cake:** Es donde nosotros hemos hecho nuestra magia.

Comprométete a no modificar los ficheros de esta carpeta. No podremos ayudarte si has modificado el núcleo.

- **La carpeta vendors:** Es donde ubicarás las librerías PHP de terceros que necesites

2.2.12. Pruebas convencionales

Son pruebas que se pueden ejecutarse o probarse en un sistema y se realiza en la fase de la implementación del sistema, los cuales son:

CAJA BLANCA.- La prueba de Caja Blanca. Denominada a veces Prueba de Caja de Cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante el método de prueba de Caja Blanca, se puede obtener casos de prueba que:

1. Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
2. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites y con sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.

CAJA NEGRA.- La prueba de Caja Negra se centra en los requisitos funcionales del software; es decir esta prueba permite obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de Caja Negra no es una alternativa

a la técnica de prueba de la Caja Blanca, sino por el contrario, se trata de un enfoque complemento que intenta descubrir diferentes tipos de errores.

La prueba de Caja Negra intenta encontrar errores como:

1. Funciones incorrectas o ausentes.
2. Errores de Interfaz.
3. Errores de estructura de datos o en acceso a bases de datos externas.
4. Errores de rendimiento y
5. Errores de inicialización y determinación.

2.2.13. Base de datos

Las bases de datos no son tan sólo una colección de archivos. Más bien, una base de datos es una fuente central de datos destinados a compartirse entre muchos usuarios para una diversidad de aplicaciones. El corazón de una base de datos lo constituye el sistema de administración de base de datos (dbms, datábase management system), el cual permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de informes y pantallas. La persona encargada de garantizar que la base de datos cumpla sus objetivos se conoce como administrador de base de datos. Según (Kendall & Kendall, 2005).

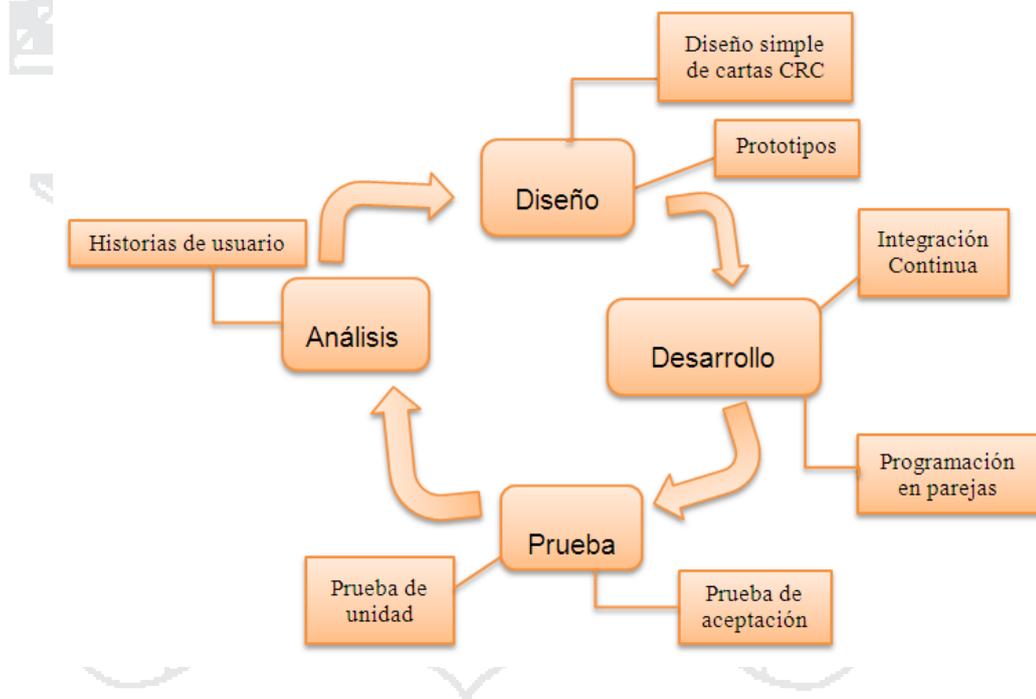
2.2.14. Metodología XP (Extreme Programming)

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en valores, principios y practicas esenciales son la

simplicidad, la comunicación, la realimentación y la valentía (Kendall & Kendall, 2005)

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Letelier p. & Penádes M., 2003)

Figura N° 2: Fases de la Metodología Extrema



Fases de la metodología XP

Fase I: ANÁLISIS

La planeación es la etapa inicial de todo proyecto en XP. En este punto se comienza a interactuar con el cliente y el resto del grupo de desarrollo para descubrir los requerimientos del sistema.

En este apartado se tendrán en cuenta ocho elementos, los cuales son los siguientes:

- ✓ **Historias De Usuario:** Las historias de usuario son utilizadas como herramienta para dar a conocer los requerimientos del sistema al equipo de desarrollo.
- ✓ **Velocidad Del Proyecto:** Es una medida de la capacidad que tiene el equipo de desarrollo para evacuar las historias de usuario en una determinada iteración.
- ✓ **Iteraciones:** Por lo general, los proyectos constan de más de tres etapas, las cuales toman el nombre de iteraciones, de allí se obtiene el concepto de metodología iterativa.

Para cada iteración se define un módulo o conjunto de historias que se van a implementar.

- ✓ **Entregas Pequeñas:** La duración de una iteración varía entre una y tres semanas, al final de la cual habrá una entrega de los avances del producto, los cuales deberán ser completamente funcionales.
- ✓ **Reuniones:** El planeamiento es esencial para cualquier tipo de metodología, es por ello que XP requiere de una revisión continua del plan de trabajo.
- ✓ **Roles En XP:**

El jefe de proyecto tiene como responsabilidad la dirección y organización de las reuniones que se realizan durante el proyecto.

El usuario o cliente determina qué se va a construir en el sistema, además de decidir el orden en que se entregarán cada segmento del proyecto.

En el grupo de los programadores se encuentran además los diseñadores y los analistas. Los programadores son quienes construyen el sistema y realizan las pruebas correspondientes a cada módulo o unidad de código.

El tester o quien realiza las pruebas, colabora en la realización de las pruebas de aceptación y es quien muestra los resultados de las mismas.

El rastreador (tracker) tiene como tarea observar la realización del sistema. Varias veces por semana cuestiona a los integrantes del equipo para anotar sus logros y avances.

- ✓ **Traslado Del Personal:** Se evitan problemas relacionados con la pérdida de conocimiento. En la medida que todos los programadores entienden todas las partes del programa.
- ✓ **Ajuste A XP:** Todos los proyectos tienen características específicas por lo cual XP puede ser modificado para ajustarse bien al proyecto en cuestión. Al iniciar el proyecto se debe aplicar XP tal como es, sin embargo no se debe dudar en modificar aquellos aspectos en que no funcione.

Fase II: DISEÑO

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado se considera que no es posible tener un diseño completo del sistema y sin

errores desde el principio. El segundo motivo es que dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales el proyecto para luego modificarlo, se considera un desperdicio de tiempo.

- ✓ **Simplicidad En El Diseño** Una de las partes más importantes de la filosofía XP es la simplicidad en todos los aspectos. Se considera que un diseño sencillo se logra más rápido y se implementa en menos tiempo.
- ✓ **Metáfora Del Sistema** Es muy importante dentro del desarrollo de la metáfora darle nombres adecuados a todos los elementos del sistema constantemente, y que estos correspondan a un sistema de nombres consistente.
- ✓ **Tarjetas de clase, responsabilidad, colaboración (CRC cards)** La principal funcionalidad que tienen estas, es ayudar a dejar el pensamiento procedimental para incorporarse al enfoque orientado a objetos.

En el proceso de diseñar el sistema por medio de las tarjetas CRC como máximo dos personas se ponen de pie adicionando o modificando las tarjetas, prestando atención a los mensajes que éstas se transmiten mientras lo demás miembros del grupo que permanecen sentados, participan en la discusión obtenida así lo que puede considerarse un diagrama de clases preliminar.

- ✓ **Soluciones puntuales (Spike Solution):** Se trata de una pequeña aplicación completamente desconectada del proyecto con la cual se intenta explorar el problema y propone una solución potencial.

- ✓ **No Solucionar Antes De Tiempo:** Los desarrolladores tienden a predecir las necesidades futuras e implementarlas antes. Según mediciones, esta es una práctica ineficiente.
- ✓ **Refactorización (Refactoring):** La refactorización en el código pretende conservarlo tan sencillo y fácil de mantener como sea posible. En cada inspección que se encuentre alguna redundancia, funcionalidad no necesaria o aspecto en general por corregir.

Fase III DESARROLLO

El desarrollo es un proceso que se realiza en forma paralela con el diseño y la cual está sujeta a varias observaciones por parte de XP consideradas controversiales por algunos expertos tales como la rotación de los programadores o la programación en parejas.

- ✓ **Cliente Siempre Presente:** Uno de los requerimientos de XP es que el cliente esté siempre disponible. No solamente para solucionar las dudas del grupo de desarrollo, debería ser parte de éste. En este sentido se convierte en gran ayuda al solucionar todas las dudas que puedan surgir, especialmente para garantizar que lo implementado.
- ✓ **Codificar Primero La Prueba:** Una de las ventajas de crear una prueba antes que el código es que permite identificar los requerimientos de dicho código. se encuentran de una forma más sencilla y con mayor claridad todos los casos especiales que debe considerar el código a implementar.
- ✓ **Programación En Parejas:** Cuando se trabaja en parejas se obtiene un diseño de mejor calidad y un código más organizado y con menores errores

que si se trabajase solo, además de la ventaja que representa contar con un compañero que ayude a solucionar

- ✓ **Integración Secuencial:** Uno de los mayores inconvenientes presentados en proyectos de software tiene que ver con la integración, sobre todo si todos los programadores son dueños de todo el código los cuales son los responsables de mantenerlas actualizadas y consistentes.
- ✓ **Integraciones Frecuentes:** Se deben hacer integraciones cada pocas horas y siempre que sea posible no debe transcurrir más un día entre una integración y otra. De esta forma se garantiza que surjan problemas.

Fase IV PRUEBAS

Del buen uso de las pruebas depende el éxito de otras prácticas, tales como la propiedad colectiva del código y la refactorización. Cuando se tienen bien implementadas las pruebas no habrá temor de modificar el código del otro programador en el sentido que si se daña alguna sección, las pruebas mostrarán el error y permitirán encontrarlo. Uno de los elementos que podría obstaculizar que un programador cambie una sección de código funcional es precisamente hacer que esta deje de funcionar. Si se tiene un grupo de pruebas que garantice su buen funcionamiento, este temor se mitiga en gran medida.

- ✓ **Pruebas Unitarias:** Estas pruebas se aplican a todos los métodos no triviales de todas las clases del proyecto con la condición que no se liberará ninguna clase que no tenga asociada su correspondiente paquete de pruebas.

- ✓ **Pruebas de Aceptación:** Las pruebas de aceptación, también llamadas pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario de las cuales deberán determinar los casos de prueba e identificar los errores que serán corregidos.
- ✓ **Cuando se Encuentra un Error:** Al momento de encontrar un error debe escribirse una prueba antes de intentar corregirlo. De esta forma tanto el cliente logrará tener completamente claro cuál fue y dónde se encontraba el mismo como el equipo de desarrollo podrá enfocar mejor sus esfuerzos para solucionarlo. Por otro lado se logrará evitar volver a cometerlo.

2.2.15. Estándar ISO/IEC 9126

Esta norma internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software.

El objetivo de esta norma ISO/IEC 9126 es proporcionar un modelo de calidad que sirve como elemento central en un proceso de evaluación. El modelo de calidad que propone la norma puede aplicarse a cualquier tipo de software incluido el desarrollo para el ámbito educativo.

Figura N°3 Norma de evaluación ISO/IEC 9126



Fuente: Estándar ISO/IEC 9126

Características generales:**1. Funcionalidad**

Es la capacidad de un producto software para satisfacer los requisitos funcionales y las necesidades de los usuarios. Se divide en 5 criterios:

Adecuación: La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

Exactitud: Evalúa el resultado final que obtiene el software y si tiene consistencia a lo que espera de él.

Interoperabilidad: consiste en revisar si el sistema puede interactuar con otro sistema independiente.

Seguridad: verifica si el sistema puede impedir el acceso a personal no autorizado.

Conformidad de la funcionalidad: La capacidad del software de cumplir los estándares referentes a la funcionalidad.

2. Confiabilidad

Es la capacidad de un producto software para mantener su nivel de desempeño bajo condiciones establecidas, por un periodo de tiempo. Se divide en 4 criterios:

Madurez: Se debe verificar las fallas del sistema y si muchas de estas han sido eliminadas durante el tiempo de pruebas o uso del sistema.

Tolerancia a errores: Evalúa si la aplicación desarrollada es capaz de manejar errores.

Recuperabilidad: Verificar si el software puede reasumir el funcionamiento y restaurar datos perdidos después de un fallo ocasional.

Conformidad de la fiabilidad: La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

3. Usabilidad

Es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido. se divide en 5 criterios:

Entendimiento: La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de manera fácil como ser utilizado para las

tareas y las condiciones particulares de la aplicación.

Aprendizaje: Determina que tan fácil es para el usuario aprender a utilizar el sistema.

Operabilidad: La manera como el software permite al usuario operarlo y controlarlo.

Atracción: Verifica que tan atractiva se ve la interfaz de la aplicación.

Conformidad de uso: La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

4. Eficiencia

Es la capacidad de un producto software de proporcionar un rendimiento apropiado, de acuerdo a la cantidad de recursos usados bajo condiciones establecidas. se divide en 3 criterios:

Comportamiento de tiempos: Verifica la rapidez en que responde el sistema.

Comportamiento de recursos: Determina si el sistema utiliza los recursos de manera eficiente.

Conformidad de eficiencia: La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

5. Mantenimiento

Es la capacidad de un producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del

software a cambios en el entorno, en los requisitos o en las especificaciones funcionales se divide en 5 criterios:

Estabilidad: Verifica si el sistema puede mantener su funcionamiento a pesar de realizar cambios.

Facilidad de analizado: Determina si la estructura de desarrollo es funcional con el objetivo de diagnosticar fácilmente las fallas.

Facilidad de Cambios: Verifica si el sistema fácilmente modificado.

Facilidad de prueba: Evalúa si el sistema puede ser probado fácilmente.

Conformidad de facilidad de mantenimiento: La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

6. Portabilidad

La capacidad de un producto software de ser transferido de un ambiente a otro. Nota: El ambiente puede ser organizacional, de software o de hardware. Se divide en 5 criterios:

Adaptabilidad: Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, transacciones, formatos de reporte, etc.).

Facilidad de instalación: Verifica si el software se puede instalar fácilmente

Coexistencia: La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.

Capacidad de reemplazamiento: Determina la facilidad con la que el software puede reemplazar otro software similar.

Conformidad de portabilidad: La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

2.3. DEFINICIÓN DE TÉRMINOS BÁSICOS

Arquitectura: Una arquitectura es un entramado de componentes funcionales, aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que puedan ser utilizadas eficazmente dentro de la organización.

Automatización: Es la realización de una tarea del hombre con la ayuda de una computadora diseñada y acondicionada para una tarea especificada

Proceso: un conjunto de actividades, material y/o flujo de información que transforma un conjunto de entradas en resultados definidos.

Registro: Es una colección de campos (items) o agregado de datos.

Archivo: Es un conjunto de Registros homogéneos (de un mismo tipo).

Sistema: Es el conjunto de elementos relacionados entre sí con un propósito determinado (único). Es un conjunto u ordenación de elementos

organizados para llevar a cabo algún método, procedimiento o control mediante el procesamiento de información. En el sentido más amplio, un sistema es simplemente un conjunto de componentes que interactúan para alcanzar algún objetivo.

Sistema de información: Es un conjunto de procedimientos organizados que, cuando se ejecutan, proporcionan información para la toma de decisiones y/o el control de la organización. Es un conjunto de personas, datos y procedimientos que funcionan en conjunto; el énfasis en sistemas significa que los variados componentes buscan un objetivo común para apoyar las actividades de la organización.

Lenguaje de programación: Es cualquier lenguaje según el cual se expresan o se escriben las instrucciones (programas) de procesamiento de la computadora. Lenguajes en el cual se escriben las instrucciones que controlan el movimiento y procesado de datos.

Programación: Conjunto de instrucciones escritas en un lenguaje particular, que representa la resolución del problema. En otros términos se puede decir que un “programa” es la elaboración de un algoritmo efectuado en lenguaje para ordenador.

Prototipo: Un Prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

Enrutador: Un enrutador es un dispositivo que regula el tráfico en la

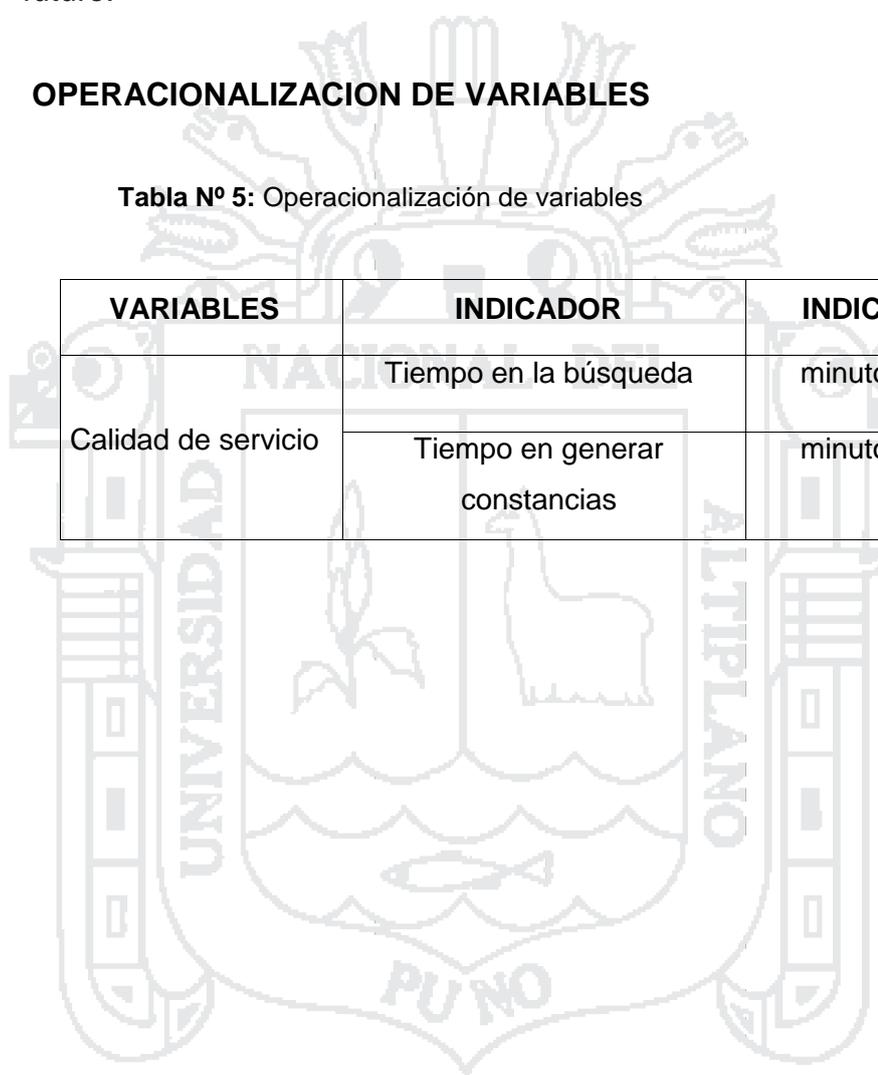
Internet haciéndolo más eficiente para cada paquete. Un paquete puede pasar a través de muchos enrutadores antes de llegar a su destino.

Tiempo: Sucesión de fenómenos que va irreversiblemente del pasado al futuro.

2.4. OPERACIONALIZACION DE VARIABLES

Tabla Nº 5: Operacionalización de variables

VARIABLES	INDICADOR	INDICE
Calidad de servicio	Tiempo en la búsqueda	minutos
	Tiempo en generar constancias	minutos



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Materiales

3.1.1. Población

La población para efectos de la investigación del proyecto presentado lo constituyen 2108 perfiles de la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno.

3.1.2. Diseño de la muestra

Se tomó el método muestral no probabilístico a criterio del investigador en este sentido la muestra está conformado por 20 registros de perfiles de la Unidad de Pensiones y Liquidaciones de la UNA – Puno

3.2. Métodos

3.2.1. Método de recolección de datos

La recolección de datos para el presente trabajo de investigación se obtuvo de los perfiles de la Unidad de Pensiones y Liquidaciones (UPL) del personal administrativo de la Universidad Nacional del Altiplano Puno.

Figura N° 4: Diseño de investigación



Fuente: Elaboración propia

3.2.2. Método de análisis de datos

Para el sistema de consulta de la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno. Se utilizó la prueba t –student para datos relacionados. Siendo la prueba de hipótesis estadística.

H_0 : No hay diferencia significativa en el tiempo promedio de búsqueda de perfiles antes y después de implementar el sistema de información.

H_1 : Si hay diferencia significativa en el tiempo promedio de búsqueda de perfiles antes y después de implementar el sistema de información.

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 > \mu_2$$

$$\alpha = 0.05 \quad (\text{Nivel significancia } (\eta = 20))$$

Los datos sobre la calificación del software serán contrastados mediante la Prueba t - student para datos relacionados.

$$t = \frac{\bar{d}}{\sqrt{\frac{\sigma_d^2}{N}}}$$

Dónde:

\bar{d} : Promedio de las diferencias.

σ_d^2 : Varianza de las diferencias.

$\bar{d} = 1.5195$

$\sigma_d^2 = 0.4957$

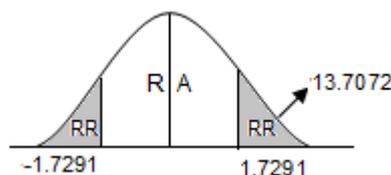
$$\Rightarrow T = \frac{\bar{d}}{\frac{\sigma_d}{\sqrt{N}}} = 13.7072$$

Hallamos t tabulado:

$\tau_{(N-1, 0.05)}$

$\tau_{(19,0.05)} = 1.7291$

Decisión: $\tau(13.7072) > \tau_{(19,0.05)} (1.7291)$



Si /t calculado / > /t tabulado/ \Rightarrow se rechaza H_0

$$13.7072 \geq 1.7291$$

Interpretación: Lo que nos indica que el tiempo promedio de búsqueda de perfiles antes de implementar el sistema de información es mayor al tiempo promedio medido después, es decir hay una diferencia significativa.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. ANÁLISIS

Para el desarrollo del sistema de consulta SICOPE, el primer paso es analizar la especificación de requisitos de software que tiene como objetivo implementar un sistema de información de consulta para la optimización de la Unidad de Pensiones y Liquidaciones la cual permitió registrar, actualizar, verificar y reportar los perfiles de los docentes y administrativos de la Universidad Nacional del Altiplano Puno.

4.1.1. Análisis de requisitos del sistema

El desarrollo del software no consiste únicamente en la codificación, aunque sea una de las fases más importantes, sino que existen múltiples tareas que deben ser desarrolladas si se desea un sistema robusto y de fácil mantenimiento y con garantías de finalización en el tiempo y coste estimados.

Para determinar las tareas es necesario asegurar un proceso de desarrollo software, adaptándolo según se estime conveniente al proyecto en el cual se aplique.

Una de estas tareas es permitir conocer en todo momento el estado en el que se encuentra el sistema durante el desarrollo.

4.1.2. Funcionalidades, requisito de tipo de usuario

4.1.2.1. En el sistema, se tiene un solo administrador y/o usuario

Administrador: toda persona con una cuenta y accesos autorizados al sistema realiza funciones tales como cuentas, perfiles de usuario y monitorear del funcionamiento del sistema de la notificación de errores a presentarse con la competencia exclusiva de este actor.

4.1.3. Casos de uso para el administrador y/o usuario

La propuesta de una interfaz está dirigida a desarrollar una herramienta tecnológica que facilite la integración de los miembros de la Unidad de pensiones y Liquidaciones (UPL) de la UNA – Puno. Utilizando las tecnologías que ofrece.

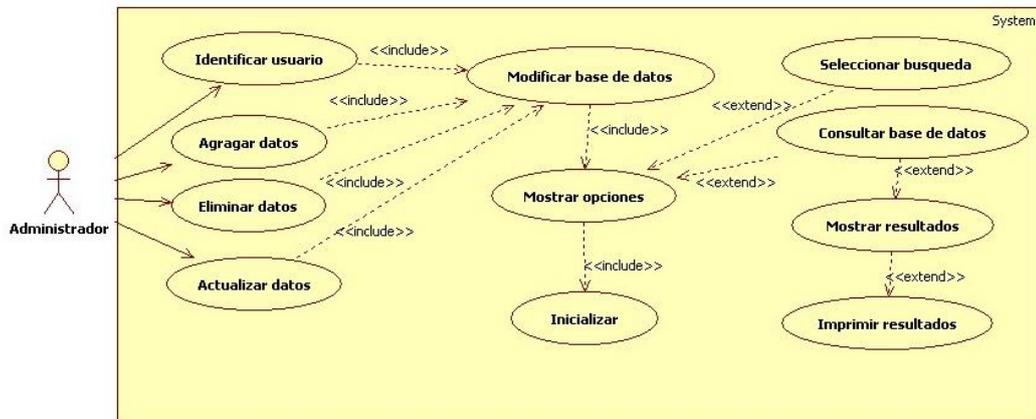
Para ello se realizó un modelado de las situaciones y vivencias que tienen los actores de la UPL, a continuación, se muestra un diagrama de Casos de Uso para mostrar el modelo de negocio de los actores de dicha Unidad.

Figura N° 5: Diagrama de caso de uso para los usuarios.



Fuente: Elaboración Propia

Figura N° 6: Diagrama de casos del sistema de consulta



Fuente: Elaboración propia

4.2. DISEÑO

El propósito del diseño es de crear la solución propuesta. La primera parte comprende el diseño en alto nivel de la arquitectura justificando la elección de un patrón arquitectónico. Respecto a la interfaz gráfica, se mencionan los patrones y estándares adoptados para uniformizar el aspecto visual y la interacción a partir de los requerimientos del sistema los cuales se diseñaran a continuación.

4.2.1. Actor

Se indica un actor según el rol que desempeña al interactuar con el sistema de consulta es la siguiente:

- **Administrador** (persona que pertenece a la institución y/o Unidad de Pensiones y Liquidaciones y además tiene atributos para modificar y actualizar la información del sistema)

4.2.2. Casos de uso

4.2.2.1. Caso de Uso Para el Administrador y/o usuario

El administrador y/o usuario se encarga del proceso de registro de los usuarios para el manejo y control del sistema y asignación de distintas cuentas y contraseñas desarrolladas es decir, son los protagonistas principales con el manejo de perfiles y constancias sus actividades generan la información que permite hacer el seguimiento de años de servicios de cada docente y administrativo. Que está estrechamente relacionada con la actualización de los perfiles debido a su relación diaria que genera información relacionada con el comportamiento, con la asignación de reportes de constancias desarrollada dentro de la Oficina. Estas situaciones se presentan en el siguiente Modelo de Casos de Uso.

Actor: Administrador y/o usuario.

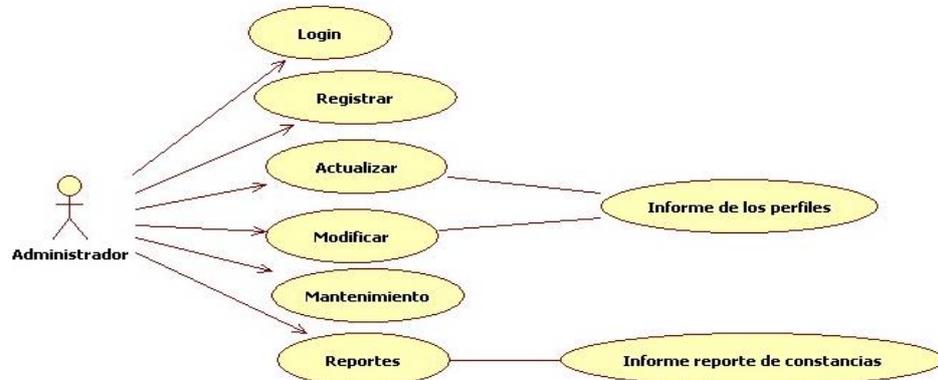
Comunica (comunicantes): El administrador y/o usuario está relacionado con los casos de uso registrar, actualizar modificar y reportar el sistema de información de manejo de perfiles de años de servicios y constancias.

Usa (use): El administrador y/o usuario planifica las actividades de mantenimiento del sistema.

Extiende (extends): Elabora el informe de funcionamiento del sistema y el

reporte de los perfiles actualizados. Estos Casos de Uso se muestran en la siguiente figura.

Figura Nº 7: diagrama de casos de uso para el administrador y/o usuario



Fuente: Elaboración propia

Descripción de los casos de uso

Login: a través del cual se activa el acceso a la interfaz para el control de la seguridad.

Registrar: mediante el cual se hace el registro de usuarios para tener control y acceso al sistema.

Actualizaciones: Este medio consiste en las actualizaciones y registro único de perfiles ya actualizados del registro de datos emitidos de parte de los usuarios de la UPL, en este proceso está asociado a la elaboración del Informe de evaluaciones de perfiles desarrollados del registro de los datos personales de cada docente, administrativo y controlar los años de servicio.

Modificaciones: Dentro de este aspecto se considera la modificación del registro de sistema de información para el desarrollo del usuario dichas modificaciones son coordinadas con los usuarios registrados conjuntamente con la jefa de la Unidad de Pensiones y Liquidaciones.

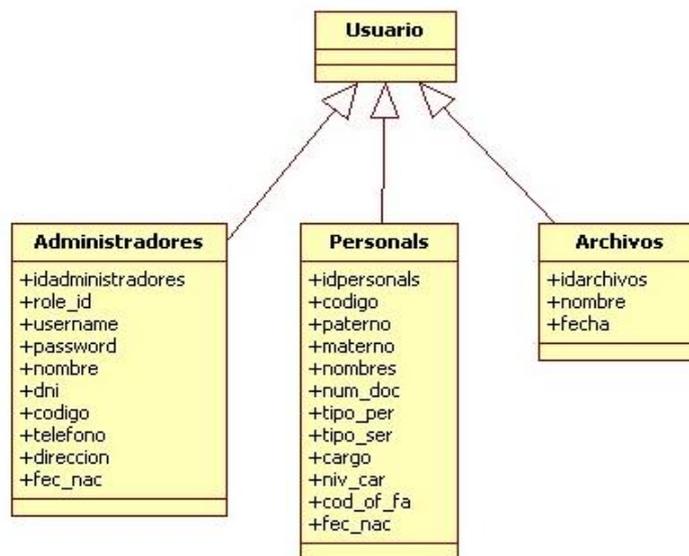
Mantenimiento: De acuerdo con la información obtenida en el proceso de registro de información se realiza el cronograma de mantenimiento estos aspectos integran la planificación del sistema de la institución.

Reporte: En este proceso se realiza un reporte final una vez obtenidos los resultados diarios de los perfiles y emitir constancias de años de servicios desarrollados durante las horas de trabajo.

4.2.3. Diagrama de clases

Todos los casos de uso anteriores se representan en diagramas de clases, comenzando por el modelado del administrador y/o usuario que representan a las entidades de negocio identificadas en la etapa de análisis con sus atributos y tipos de datos utilizados la cual se muestra en la siguiente figura.

Figura Nº 8: Diagrama de clases para el administrador y/o usuario

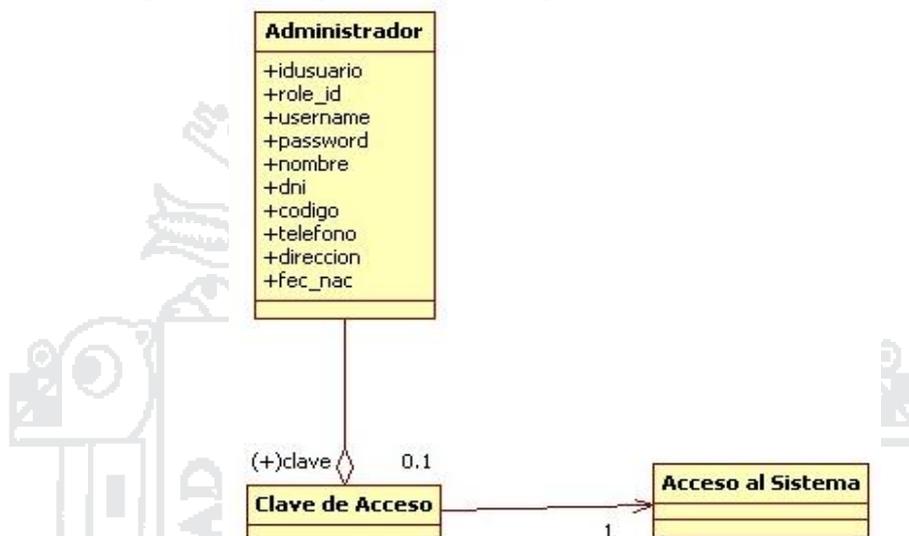


Fuente: Elaboración propia

4.2.3.1. Diagrama de Clases Para el Administrador y/o usuario

Empezaremos por la entidad Administrador y/o usuario el cual está asociado a los casos de uso: LOGIN.

Figura Nº 9: Diagrama de clases login de administrador y/o usuario

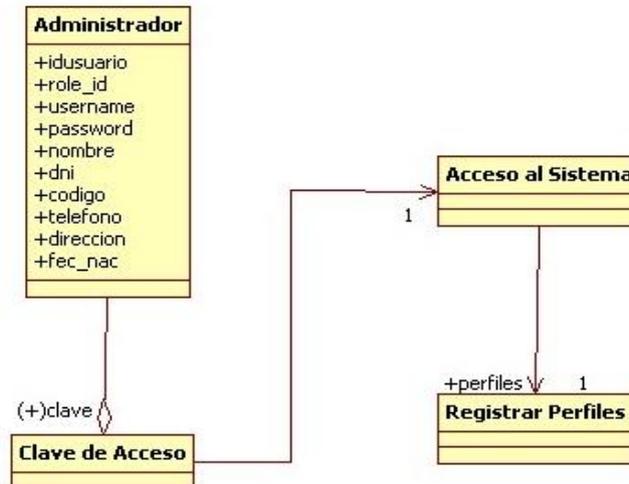


Fuente: Elaboración propia

Según se observa en la Figura, el administrador y/o usuario a través de una clave de acceso puede ingresar al sistema, esta clave de acceso tiene dos únicas posibilidades (cierto = 1 o falso = 0) de lo cual se define el acceso a la interfaz la cual solamente se hace por una sola entidad con esta descripción.

Similarmente se presentan los Diagramas de Clases para el caso de uso **REGISTRAR** en la cual el administrador y/o usuario incluye en la Interfaz relacionada al realizar registros nuevos usuarios. Esta clase se representa en el siguiente diagrama.

Figura N° 10: Diagrama de clases para registrar nuevo perfil

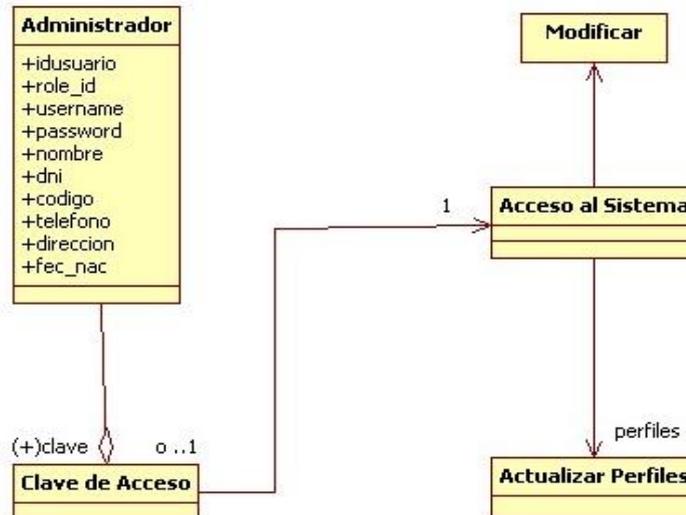


Fuente: Elaboración propia

De acuerdo con la Figura anterior, el administrador y/o usuario una vez ingresado al sistema esta entidad realiza individualmente el registro de nuevos usuarios en el sistema.

Los Diagramas de Clases para los casos de uso **ACTUALIZAR** y **MODIFICAR** en los cuales el administrador y/o usuario registra en la Interfaz información relacionada con las modificaciones realizadas en el sistema. Estas clases se representan en la siguiente figura.

Figura N° 11: Diagrama de clases para modificar y actualizar perfiles

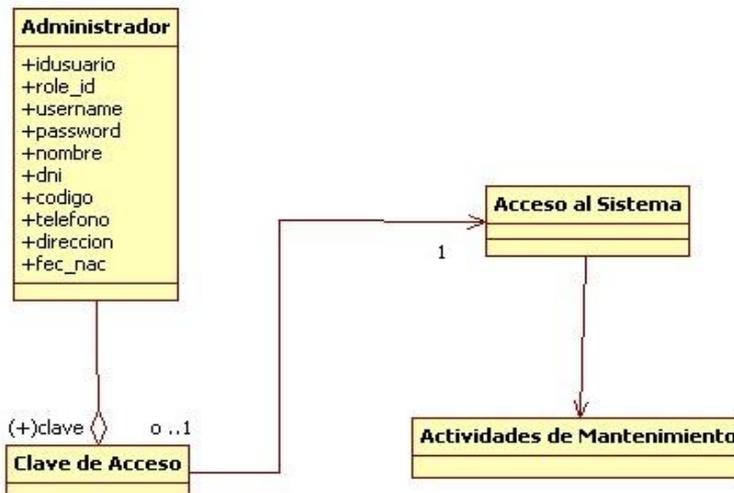


Fuente: Elaboración propia

De acuerdo con la figura anterior, el administrador y/o usuario una vez ingresado al sistema esta entidad realiza cualquiera de las actividades especificadas en el diagrama, Modificar o actualizar los perfiles. Todas estas clases están relacionadas con las actividades del actor Administrador.

El Diagrama de Clases para los casos de uso **MANTENIMIENTOS** en las cuales el administrador y/o usuario registra en la Interfaz información relacionada con las actividades de mantenimientos realizando las verificaciones respectivas. Estas clases se representan en la siguiente figura.

Figura N° 12: Diagrama de clases para los mantenimientos



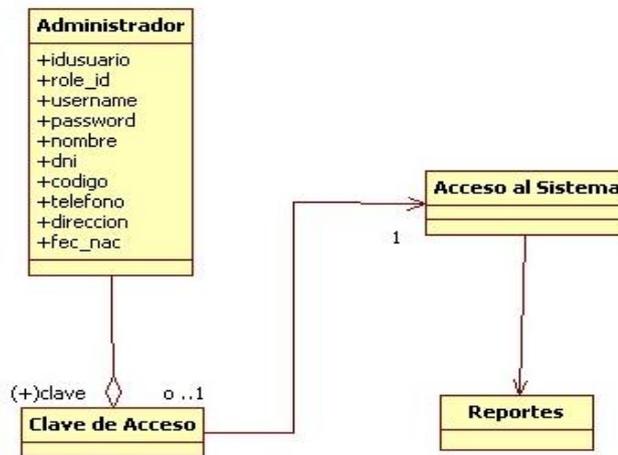
Fuente: Elaboración propia

En la Figura anterior, se observa que el administrador a través de una clave de acceso puede ingresar al sistema, una vez ingresado al sistema esta entidad registra la información correspondiente a los mantenimientos del sistema.

Finalmente, como consecuencia del registro constante de perfiles, el usuario actualiza a diario el perfil presentado durante el tiempo solicitado,

en el Caso de Uso: **REPORTE DE CONSTANCIAS**, en este caso, el usuario puede registrar en la interfaz las distintas observaciones que haya hecho, relacionados con la actualización realizada que estará disponible en la interfaz para el reporte final de constancias de años de servicio de cada uno de los Docentes y Administrativos. Esta clase se representa en la siguiente figura:

Figura Nº 13: Diagrama de clases para el reporte final de constancias



Fuente: Elaboración propia

4.2.4. Diagrama de Actividades

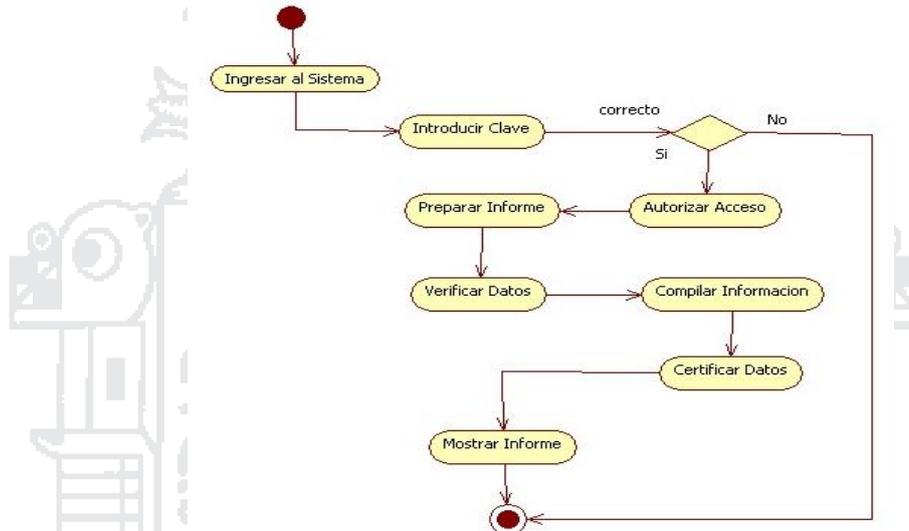
La siguiente etapa en el modelo de análisis basado en UML comprende el modelado de las acciones a ejecutarse en el sistema, para ello se hace la representación de las acciones a desarrollar por cada uno de los actores.

4.2.4.1. Diagrama de Actividades Para el Administrador y/o usuario

Comenzando con el actor Administrador que es el encargado de actualizar lo concerniente a la información se puede resumir las actividades a desarrollar en tres actividades generales que comprenden: Realizar informe de Registros, Actualizaciones, Actividades de Mantenimientos y Reportes. En el primer caso, para la elaboración del Informe de registros el

Administrador debe verificar previamente la ejecución de las ACTIVIDADES DE REGISTROS por parte del personal usuario, posteriormente, con la compilación de esta información se realiza el reporte de constancias que sirve de guía para el seguimiento a los perfiles. Esto se presenta en la siguiente figura.

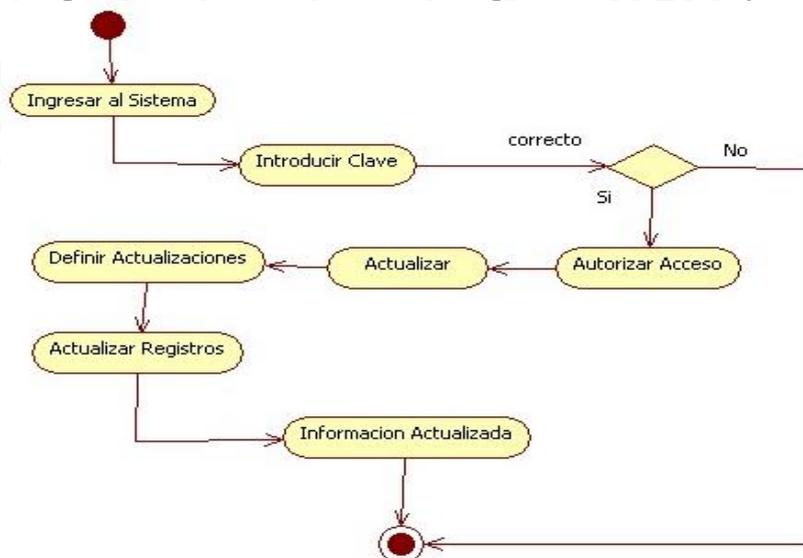
Figura N°14: Diagrama de actividades para el administrador y/o usuario



Fuente: Elaboración propia

Por su parte, el Actor Administrador y/o Usuario cumple sus funciones dentro del Sistema en este proceso se muestran en el diagrama siguiente.

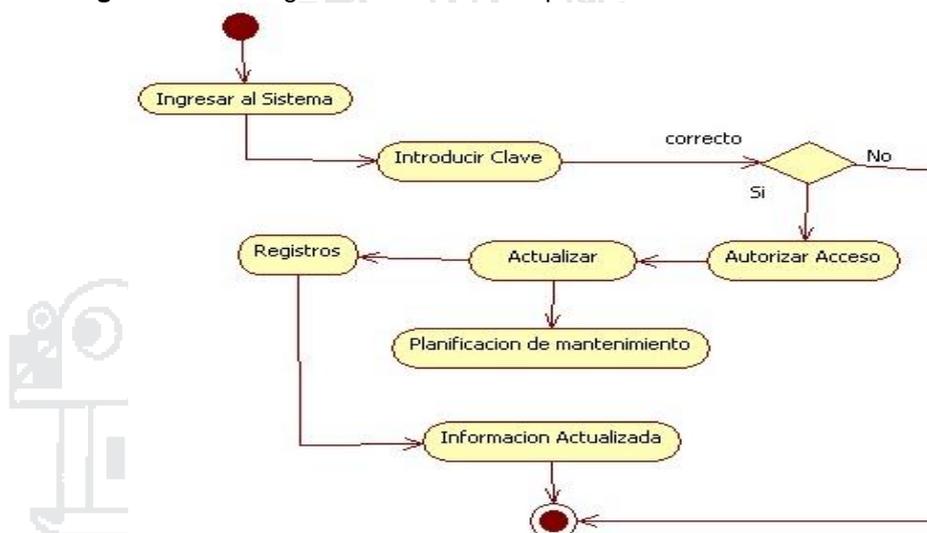
Figura N°15: Diagrama de actividades, Actualizar Registros administrador y/o Usuario



Fuente: Elaboración propia

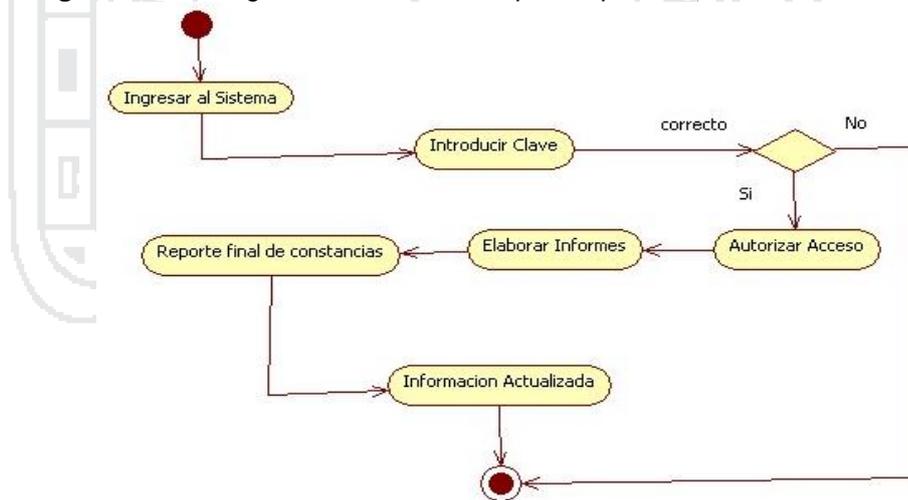
Por otra parte, el Administrador y/o Usuario como responsable de la publicación de los REGISTROS realizados durante el desarrollo ACTIVIDADES DE MANTENIMIENTO programadas cumple con las actividades señaladas en la figura que se presentan a continuación.

Figura N° 16: Diagrama de actividades para actividades de mantenimiento



Fuente: Elaboración propia

Figura N° 17: Diagrama de actividades, para reporte final de constancias

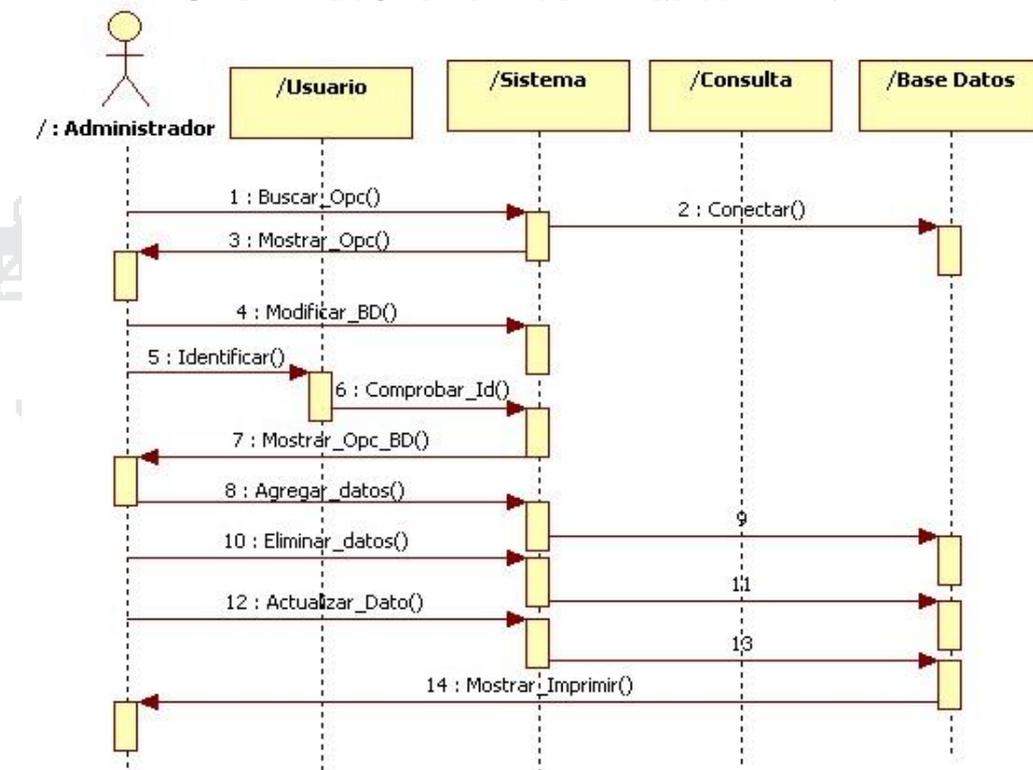


Fuente: Elaboración propia

4.2.5. Diagrama de integraciones administrador

En el diagrama de interacciones se representa la forma como los actores (Administrador) y los (Objetos) se comunican entre sí según los eventos del sistema. Para desarrollar el siguiente diagrama de integración se utiliza la información obtenida de los Casos de Uso:

Figura N° 18: Diagrama de interacciones (Administrador)



Fuente: Elaboración propia

4.2.5.1. Diagrama de interacciones de Registro de perfiles

En la figura muestra la interacción de objetos en el proceso de registro de datos de perfiles. Donde administrador habilita un nuevo registro e ingresa los datos personales del docente, Administrativo (Contratado, Nombrado) y finalmente el administrador valida y guarda los datos en la base de datos.

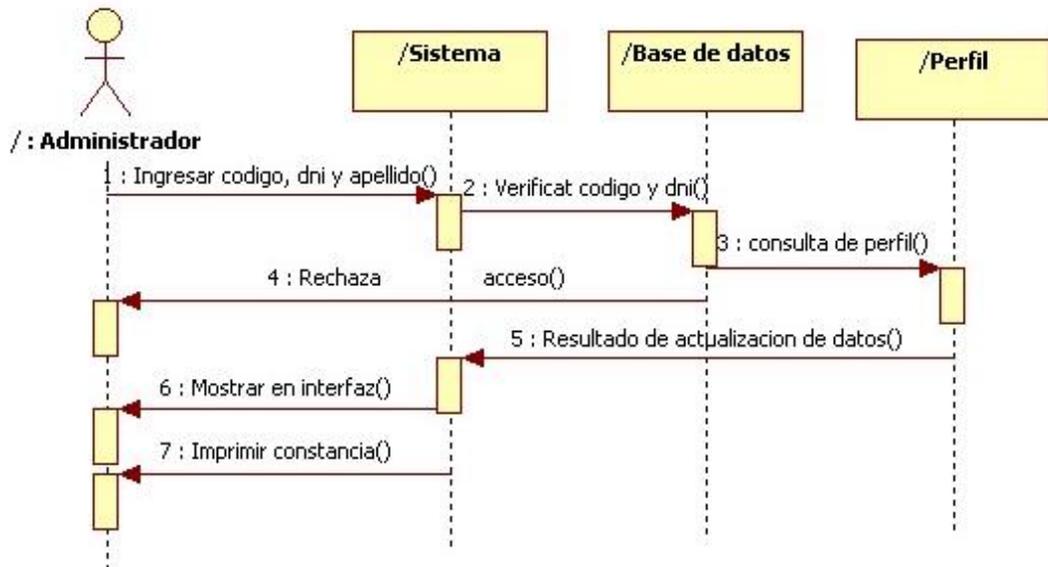
Figura N° 19: Interacción en el registro de Datos de Perfiles



Fuente: Elaboración propia

4.2.6. Diagrama de secuencias de una consulta de perfil

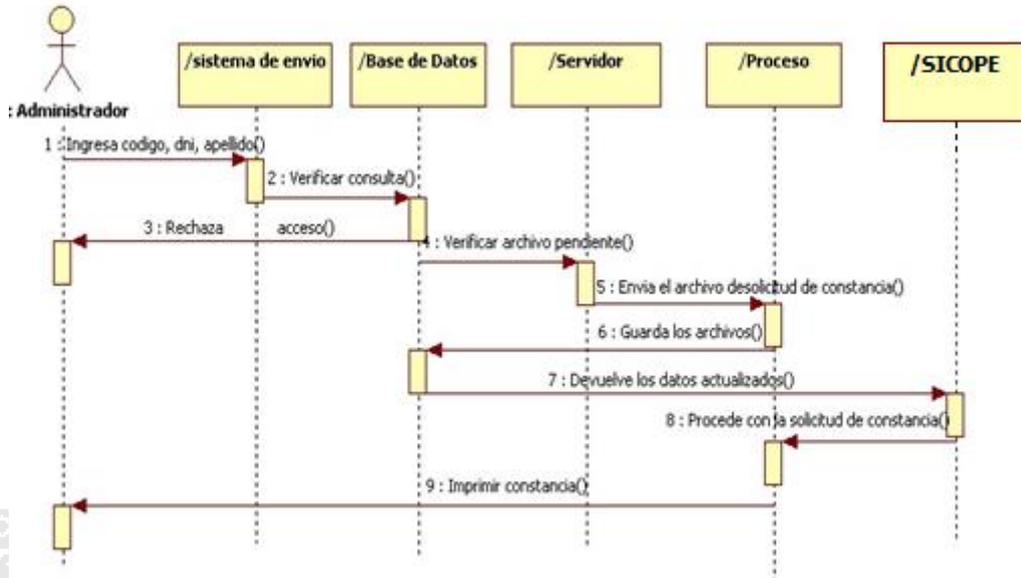
Figura N° 20: Diagrama de secuencias para una consulta Web



Fuente: Elaboración propia

4.2.6.1. Diagrama de secuencia para el sistema de envío de archivo

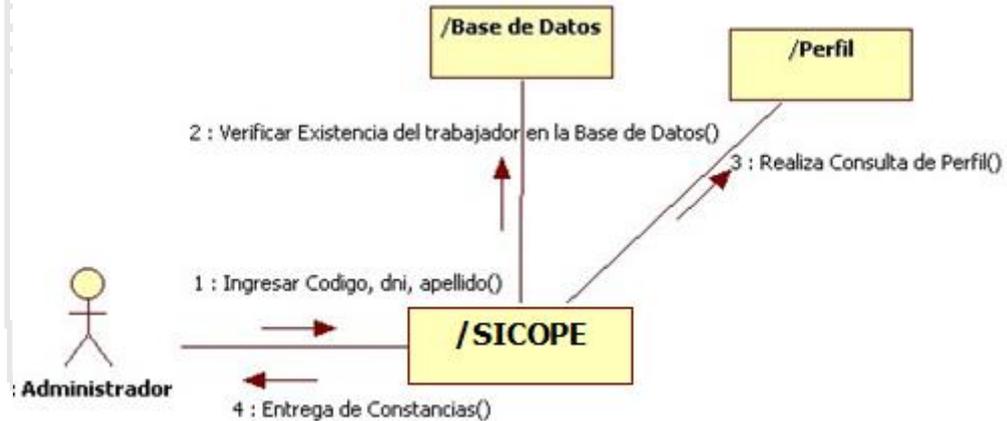
Figura N° 21: Diagrama de Secuencias para el sistema de Envío de archivos



Fuente: Elaboración propia

4.2.7. Diagrama de colaboración de una consulta de perfil

Figura N° 22: Diagrama de Colaboración consulta de perfil de un trabajador

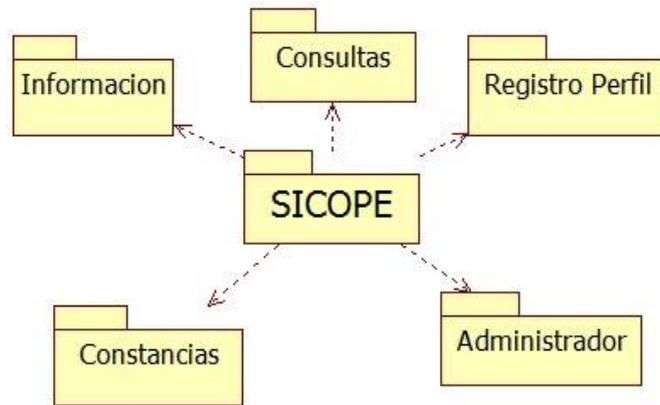


Fuente: Elaboración propia

4.2.8. Diagrama de componentes

Diagrama de componentes de sistemas de información que se muestra en la siguiente figura.

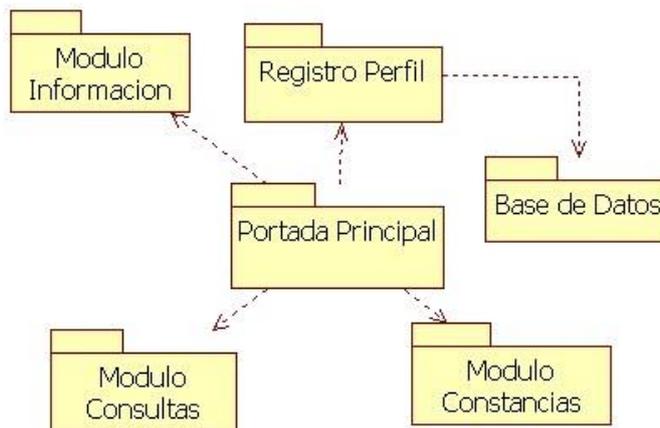
Figura N° 23: Componentes del Sistema de Información



Fuente: Elaboración propia

4.2.8.1. Sistema

Figura N°24: Módulos del sistema



Fuente: Elaboración propia

4.2.9. Base de Datos

El modelado de las Bases de Datos constituye uno de los puntos débiles en UML por cuanto los diagramas de clase presentan un mecanismo de implementación neutral para modelar los aspectos de almacenado de datos del sistema, pero no cubre toda la semántica involucrada en el modelo relacional. Para capturar esta relación se utiliza un diagrama de Relación de Entidad como una extensión a UML. Para ello se efectúa un análisis sobre las necesidades de información de la institución, así como de los

requerimientos de operación y manejo de datos durante cada uno de los procesos que hacen en la oficina.

Entidad: Administrador

Atributos: idusuario, username, password, nombres, ape_paterno, ape_materno, dirección, teléfono, código.

Entidad: Personales

Atributos: idpersonals, código, ape_paterno, ape_materno, nombres, num_doc, tipo_per, tipo_ser, cargo, niv_car, cod_of_fa, fec_nac.

Entidad: Archivos

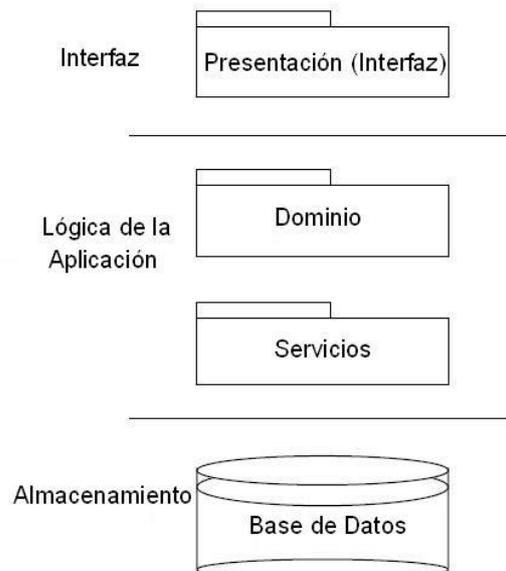
Atributos: idarchivos, nombre, fecha.

4.2.10. Arquitectura de datos del sistema

Se utiliza la arquitectura clásica para sistemas de información, la cual consiste en tres niveles y son los siguientes:

- **Nivel 1:** presentación (interfaz), donde se encuentra las pantallas e informes web.
- **Nivel 2:** lógica de la aplicación, se refiere a las tareas y reglas de los procesos.
- **Nivel 3:** Almacenamiento, se encuentra la base de datos del sistema.

Figura N° 25: Arquitectura del sistema

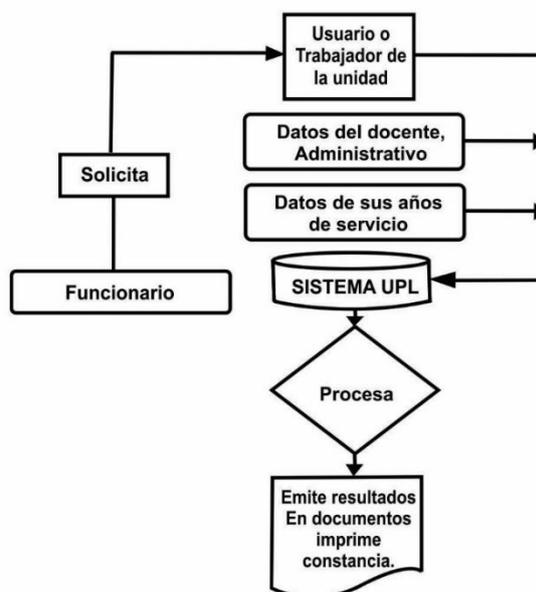


Fuente: Universidad Politécnica de Madrid

4.2.11. modelamiento y funciones del sistema

Diagrama de casos de uso, que describe el tipo de dato como entrada al proceso de análisis del software. Un caso de uso es, en esencia, una interacción típica entre un usuario y un sistema de cómputo; y emplearemos el termino actor para llamar al usuario, cuando desempeña ese papel con respecto al sistema.

Figura N°26: Modelamiento y funciones del sistema



Fuente: Elaboración propia

4.2.12. Tablas de Archivos

Seguidamente se procede a establecer la arquitectura de datos del sistema, la cual está compuesta de tres (03) tablas.

4.2.12.1. Tabla Para el Administrador y/o Usuario

Tabla Nº 6: Administrador y/o usuario

Campo	Nombre	Tipo	Tamaño	Decimales
1	Idusuario	Numérico	11	0
2	role_id	Texto	10	
3	username	Texto	15	
4	password	Texto	8	
5	nombres	Texto	35	
6	ape_paterno	Texto	30	
7	ape_materno	Texto	30	
8	Dirección	Texto	35	
9	Teléfono	Texto	9	
10	Código	Texto	10	

Fuente: tabla Base de Datos xampp

4.2.12.2. Tabla Para Personales

Tabla Nº 7: personales

Campo	Nombre	Tipo	Tamaño	Decimales
1	Id	Numérico	11	0
2	código	Texto	50	
3	paterno	Texto	50	
4	materno	Texto	50	
5	nombres	Texto	30	
6	num_doc	LOGICO	10	
7	tipo_per	Texto	50	
8	tipo_ser	Texto	50	
9	cargo	Texto	50	
10	niv_car	Texto	50	
11	cod_of_fa	FECHA	40	
12	fec_nac	Datetime		

Fuente: tabla Base de Datos xampp

4.2.12.3. Tabla Para Perfiles y Constancias

Tabla N° 8: Archivos perfiles

Campo	Nombre	Tipo	Tamaño	Decimales
1	Id	Numérico	11	0
2	Personal_id	Texto	10	
3	tipo	texto	200	
4	nombre	Texto	100	
5	fecha	Date time		

Fuente: tabla Base de Datos xampp

4.2.13. Arquitectura de Datos funcionales del Sistema

Una vez diseñada la arquitectura de datos del sistema es necesario especificar la arquitectura de los datos funcionales, en este sentido se tiene:

4.2.13.1. Tabla Administrador

Corresponde a la Tabla para almacenar los usuarios autorizados por la Dirección de Estudios para modificar una o todas las tablas que comprenden la Base de Datos. Está compuesta por los siguientes campos:

IDUSUARIO: Campo destinado a almacenar el nombre de usuario de identidad autorizado conjuntamente con el password conforma la clave de acceso al sistema, es de tipo texto y puede contener hasta 15 caracteres.

PASSWORD: Campo destinado a almacenar la clave de acceso de cada usuario, en principio será asignada por el administrador de la interfaz pero que el usuario puede modificar posteriormente, comprende datos de tipo texto y puede utilizar hasta 50 dígitos.

NOMBRES: Campo reservado para almacenar el nombre del usuario autorizado, comprende datos de tipo texto y puede alcanzar hasta 35 caracteres.

APE_PATERNO: Campo reservado para almacenar el apellido paterno del usuario autorizado, comprende datos de tipo texto y puede alcanzar hasta 30 caracteres.

APE_MATERNO: Campo reservado para almacenar el apellido materno del usuario, comprende datos de tipo texto y puede alcanzar hasta 30 caracteres.

NUMERO DOCUMENTO: Campo reservado para almacenar el N° documento del usuario, comprende datos de tipo numérico y puede alcanzar hasta 8 caracteres.

DIRECCION: Campo reservado para almacenar la dirección del usuario autorizado, comprende datos de tipo texto y puede alcanzar hasta 35 caracteres.

TELEFONO: Campo reservado para almacenar el número de celular o teléfono fijo del usuario autorizado, comprende datos de tipo numérico y puede contener hasta 9 caracteres.

4.2.13.2. Tabla Personales

Corresponde a la Tabla para almacenar los datos de los usuarios que prestan servicio a la Unidad que pueden actualizar una o más tablas que comprenden la Base de Datos. Está compuesta por los siguientes campos:

IDCODIGO: Almacena el código del usuario respectivo, es de tipo numérico y puede tener hasta 11 dígitos.

APE_PATERNO: Campo reservado para almacenar el apellido paterno del

usuario autorizado, comprende datos de tipo texto y puede alcanzar hasta 50 caracteres.

APE_MATERNO: Campo reservado para almacenar el apellido materno del usuario autorizado, comprende datos de tipo texto y puede alcanzar hasta 50 caracteres.

NOMBRES: Campo reservado para almacenar el nombre del usuario autorizado, comprende datos de tipo texto y puede alcanzar hasta 30 caracteres.

NUMERO DOCUMENTO: Campo reservado para almacenar el N^o documento del usuario, comprende datos de tipo numérico y puede alcanzar hasta 8 caracteres.

TIPO DE PERSONAL: Campo reservado para almacenar el tipo de personal (Administrativo, Docente), comprende datos de tipo texto y puede alcanzar hasta 50 caracteres.

TIPO DE SERVICIO: Campo reservado para almacenar el tipo de servicio (Nombrado, Contratado), comprende datos de tipo texto y puede alcanzar hasta 50 caracteres.

CARGO: Campo reservado para almacenar el cargo que ocupa, comprende datos de tipo texto y puede alcanzar hasta 50 caracteres.

NIVEL DE CARGO: Campo reservado para almacenar el nivel de cargo (categoría) que ocupa el personal, comprende datos de tipo texto y puede alcanzar hasta 100 caracteres.

ESCUELA O FACULTAD: Campo reservado para almacenar la escuela o facultad que se desempeñan, comprende datos de tipo texto y puede alcanzar hasta 100 caracteres

FEC_NAC: Campo reservado para almacenar la fecha de nacimiento del usuario autorizado, comprende datos de tipo texto y puede alcanzar hasta 30 caracteres.

4.2.13.3. Tabla Perfiles y Constancias

Contiene información correspondiente a los perfiles que consta en planillas, esta tabla permite actualizar la información sin necesidad de modificar la programación de la interfaz. Comprende los siguientes campos:

TIPO: campo que indica la selección del campo que se desea realizar (constancia o perfil) que se quiere actualizar, este dato es de tipo texto y puede alcanzar hasta 200 caracteres.

NOMBRE: Comprende el campo donde se registra los perfiles actualizados, es de tipo texto y puede alcanzar hasta de 100 caracteres

FECHA: Campo destinado a almacenar la fecha de emisión del registro realizado, es un dato tipo date time.

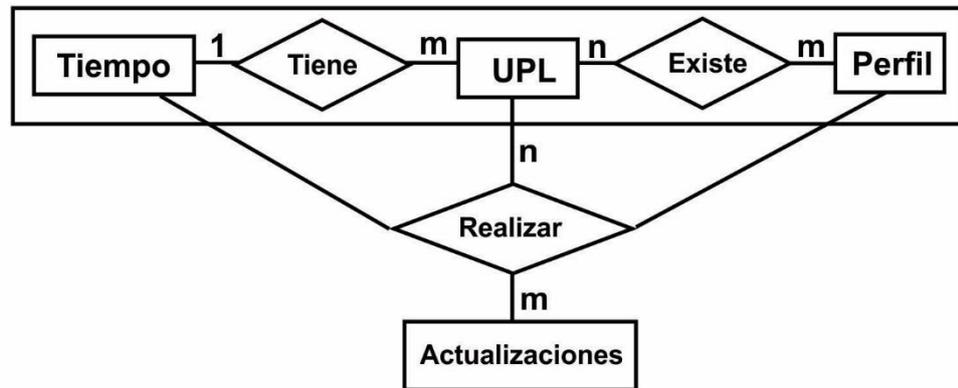
4.2.14. Modelo Entidad Relación

En este apartado, se analizará el diagrama entidad / relación, los datos se han modelado según el diagrama E/R mostrando en la figura.

La Unidad de Pensiones y Liquidaciones (UPL) tienen perfil en un

determinado tiempo y realiza actualizaciones que a su vez están relacionados con el mismo tiempo y perfil.

Figura N° 27: Modelo Entidad Relación (E-R)



Fuente: Elaboración propia

4.3. DESARROLLO

En el presente capítulo describiremos la metodología para la implementación del sistema de consulta, patrones de implementación y la arquitectura utilizada, también se tiene los prototipos de las pantallas a implementar.

La planificación de análisis de los requisitos, es necesario plasmar la solución en un lenguaje de programación. De forma análoga al diseño, en el que se abstraen patrones para aplicarlos, durante la implementación se pueden abstraer detalles.

Durante el desarrollo se ha generado una serie de ficheros de código, documentación que han precisado de cierto tiempo para completarse. Un análisis del tiempo invertido en cada una de las fases, junto al tamaño del programa final desarrollado.

4.3.1. Generación y proceso de formularios en CakePHP

El poder de CakePHP reside en su simplicidad y velocidad. Una de las aplicaciones más comunes que se utilizan en este lenguaje son los formularios de CakePHP, por su parte CakePHP posee conocimientos generales de PHP y conocimientos básicos de programación orientada a objetos para el desarrollo de los formularios

4.3.2. Pantalla principal o Índice

Al acceder al dominio reservado por la institución en este caso el servidor local (<http://localhost/perfil/>) se inicia una pantalla. En ella se muestra el título se va a mantener constante durante toda la navegación que haga el usuario.

Figura N° 28: ventana principal

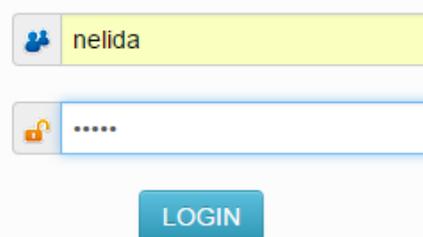


Fuente: Elaboración propia

4.3.3. Ventana de acceso

Con esta pantalla se inicia la ejecución del SISTEMA DE CONSULTA, para lo cual el administrador encargado deberá ingresar el nombre de usuario y la clave de acceso, para acceder al de inscripción. Cabe recalcar que si no se conoce la clave administrador, negará el acceso al sub-sistema de administración.

Figura N° 29: Ventana de Acceso



Fuente: Elaboración propia

4.3.4. Ventana de inscripción de nuevos usuarios

La figura, muestra la subventana de inscripción de registro de nuevos usuarios. Esta ventana se ocupa de almacenar la información de nuevos usuarios; y además nos muestra los elementos que lo componen.

Figura N° 30: Sub-sistema de Inscripción de nuevos usuarios

Registrar Usuario



Fuente: Elaboración propia

4.3.5. Ventana de registro de nuevos perfiles

La figura N° 31, muestra la ventana de Ingreso, el registro de nuevos perfiles, la misma que permite realizar ingresar datos de los perfiles anteriores y actuales que servirán para emitir los formatos y otros.

Figura N° 31: Ventana de registro de nuevos perfiles

Registrar de nuevo personal

Fuente: Elaboración propia

4.3.6. Ventana para subir archivos de perfiles o constancias

Para subir un perfil encontrara un botón en la lista de cada personal, dándole click en el botón <<“ADJUNTAR” >>

Figura N° 32: Ventana para adjuntar archivos

Datos Adjuntos

Tipo de Archivo

PERFIL

Adjuntar Archivo

PERFIL

CONSTANCIA

Subir archivo

Fuente: Elaboración propia

4.3.7. Reporte final del sistema

REPORTE DE CONSTANCIAS DE AÑOS DE SERVICIO

De la impresión de las constancias de los datos actualizados, se encargan los responsables de la Unidad de Pensiones y Liquidaciones.

4.3.8. Base de datos

Construcción de la base de datos

Es el Componente principal de un sistema de información, para la construcción de la base de datos se utilizó el Programa MySQL.

La Base de Datos del SISTEMA DE CONSULTA está construida por 03 tablas básicas; las mismas que interactúan en el sistema a través de relaciones establecidas entre ellas. Las relaciones de la Base de Datos, se establecen a través de codificación.

La figura, muestra las tablas que componen la Base de Datos del SISTEMA DE CONSULTA.

Normalización de la base de datos

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica.

Primera forma normal (1FN)

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Figura N° 33: Base de datos 1FN

users	personal	tipo_personal	tipo_servicio	cargo	of_esc_fac	archivos
id: INTEGER	id: INTEGER	id: INTEGER	id: INTEGER	id: INTEGER	id: INTEGER	id: INTEGER
username: VARCHAR(200)	codigo: VARCHAR(100)	docentes: VARCHAR(20)	nombrado: VARCHAR(45)	DE: VARCHAR(20)	name: VARCHAR(45)	nombre: VARCHAR(200)
password: VARCHAR(200)	paterno: VARCHAR(200)	administrativos: VARCHAR(20)	contratado: VARCHAR(45)	TC: VARCHAR(45)	descripcion: TEXT	tipo: VARCHAR(200)
name: VARCHAR(100)	materno: VARCHAR(200)			TP: VARCHAR(45)		fecha: DATETIME
dni: VARCHAR(8)	nombres: VARCHAR(200)			STA: VARCHAR(45)		
	num_doc: VARCHAR(100)					

Fuente: DBDesigner 4

Segunda forma normal (2FN)

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.

Figura N° 34: Base de datos 2FN

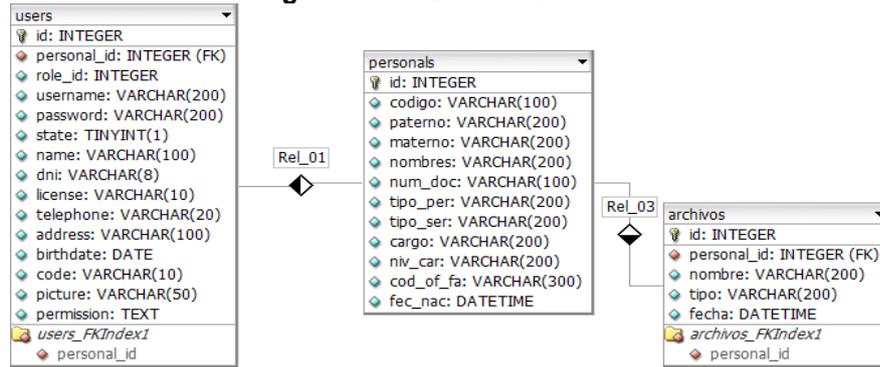
users	tipo_personal	servicio	archivos
id: INTEGER	id: INTEGER	id: INTEGER	id: INTEGER
role_id: INTEGER	codigo: VARCHAR(100)	tipo_serv: VARCHAR(255)	nombre: VARCHAR(200)
username: VARCHAR(200)	paterno: VARCHAR(200)	cargo: VARCHAR(45)	tipo: VARCHAR(200)
password: VARCHAR(200)	materno: VARCHAR(200)		doc_of_fa: VARCHAR(45)
state: TINYINT(1)	nombres: VARCHAR(200)		fecha: DATETIME
name: VARCHAR(100)	num_doc: VARCHAR(100)		
dni: VARCHAR(8)	tipo_per: VARCHAR(200)		
license: VARCHAR(10)			
telephone: VARCHAR(20)			
address: VARCHAR(100)			
birthdate: DATE			
code: VARCHAR(10)			
picture: VARCHAR(50)			
permission: TEXT			

Fuente: DBDesigner 4

Tercera forma normal (3FN)

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas.

Figura N° 35: Base de Datos 3FN



Fuente: DBDesigner 4

4.3.9. Implementación

a. **CakePHP:** CakePHP es un marco de desarrollo (framework) rápido para PHP, libre, de código totalmente abierto, se trata de una estructura que sirve de base a los programadores para que se puedan crear aplicaciones. El objetivo primordial es trabajar de forma estructurada y rápida, sin pérdida de flexibilidad.

b. **PHP:** Como se mencionó, PHP permite generar páginas Web con contenido dinámico, por lo cual es necesario recordar cómo funcionan estas páginas.

Las páginas Web son documentos que se pueden visualizar a través de Internet, que pueden incluir texto, imágenes, ligas de hipertexto y otros medios.

c. **PHPMYADMIN:** Es un programa de libre distribución en PHP, creado por una comunidad sin ánimo de lucro, que sólo trabaja en el proyecto por amor al arte. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz Web muy intuitiva. La aplicación en si no es más que un conjunto

de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor Web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas

4.1. PRUEBA

A. Pruebas no convencionales

Son pruebas que consisten en las revisiones técnicas formales que se realizaron en las etapas de análisis y diseño del sistema, que corrigen errores básicamente de:

- Omisiones y ambigüedad en las definiciones de clases y jerarquías, así como en las relaciones.
- Inconsistencias en la elaboración de Diagrama de Casos de Uso, Interacción, Clases y Actividades.

B. Pruebas convencionales

Son pruebas que se pueden ejecutarse o probarse, los cuales se realizan en la etapa de la implementación del sistema como son las pruebas de caja negra y caja blanca.

Prueba de caja blanca: Esta prueba de software, fue desarrollada durante la construcción de cada módulo. Mediante esta prueba se garantiza que el prototipo del sistema de control, cumplió con:

- Ejecutar todos los caminos independientes de cada módulo.
- La estructura de los datos es compatible.
- Ninguno de los bucles es infinito o su ejecución es por demás.
- Cuenta con todas las decisiones lógicas necesarias.
- Dar mayor prioridad a las consultas de búsqueda

Prueba de caja negra: Esta prueba de software se aplicó en el desarrollo de los módulos así como también estuvieron terminados y enlazados entre ellos para su funcionamiento como sistema. Mediante esta prueba aseguramos que el Prototipo de Sistema SIRECI, no tiene errores de:

- Procedimientos o funciones incorrectas.
- Los pie de reporte muestran información de continuación
- Errores de entrada y salida.
- Errores de rendimiento.
- Errores de inicialización y finalización.
- Los resultados de la consulta muestra lo requerido

Prueba de validación del software

Esta fase de prueba del sistema se realiza mediante el método Prueba Basado en Escenarios, con el fin de descubrir errores de interfaz y errores del procesamiento de datos al nivel de los resultados esperados. La prueba se concentra en lo que el usuario hace, interacción del usuario con el sistema. La validación de la funcionalidad integra del sistema se prueba con los datos de la población de personal docente, administrativo, esto

permite verificar la certidumbre de los resultados proporcionados en el proceso de análisis del sistema.

Prueba del sistema

Antes de implementar el sistema	Después de implementar el sistema
Emisión de reportes días o semanas	Emisión de reportes inmediato
Elaboración de búsqueda promedio de 1 hora a más.	Elaboración de búsqueda inmediata
Se produce constantes errores en el llenado de las constancias.	No se produce errores al momento de generar las constancias.
No hay seguridad	Tiene seguridad
Solo el personal autorizado puede actualizar la información de las planillas	Los usuarios registrados pueden actualizar su información
La información de la oficina es vulnerable.	Los trabajadores de la unidad son los únicos concedores de su clave de acceso al sistema
La información se guarda en planillas físicas las cuales son realizadas con el apoyo del Excel.	La Base de Datos de los perfiles están guardadas en un servidor.

Fuente: Elaboración propia

Las investigaciones de Jiménez Doria (2009) y Mendoza Lolimar (2011) concluyen que la metodología XP se adapta al tema de investigación que realizaron. En mi investigación también se utilizó la misma metodología porque es la más usada en el desarrollo de sistemas de información.

Por lo tanto concluimos que el sistema implementado en la Unidad de Pensiones y Liquidaciones de la UNA – P. mejoró notablemente para los encargados del sistema, en beneficio de los usuarios.

CONCLUSIONES

En la presente investigación se ha llegado a las siguientes conclusiones.

1. Se ha logrado implementar un sistema para los procesos de búsqueda y consulta de perfiles usando la tecnología web, el desarrollo de dicho sistema mejora la calidad de servicio en la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno.
2. Hay Diferencia significativa en el análisis del tiempo de demora entre el antes y el después de la implementación del sistema en la búsqueda de información.
3. El tiempo de demora redujo notablemente con la implementación del sistema de consulta de perfiles de la Unidad de Pensiones y Liquidaciones de la Universidad Nacional del Altiplano Puno. Se realizó un análisis estadístico utilizando la prueba t-student para datos relacionados:

$$\tau(13.7072) > \tau_{(19,0.05)}(1.7291)$$

Lo que nos indica que el tiempo promedio de búsqueda de perfiles es mucho mejor.

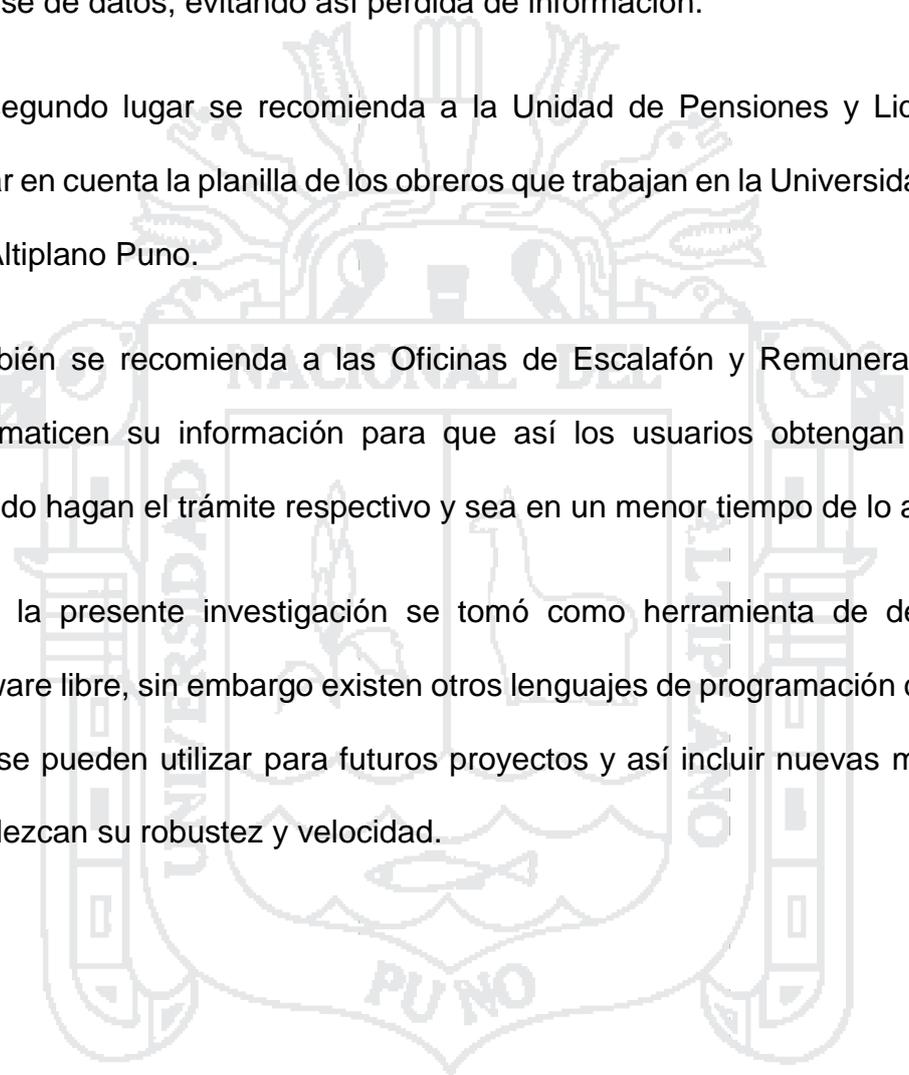
RECOMENDACIONES

En primer lugar se recomienda a la Unidad de Pensiones y Liquidaciones crear respaldos periódicamente para mantener en resguardo una copia actualizada de la base de datos, evitando así pérdida de información.

En segundo lugar se recomienda a la Unidad de Pensiones y Liquidaciones tomar en cuenta la planilla de los obreros que trabajan en la Universidad Nacional del Altiplano Puno.

También se recomienda a las Oficinas de Escalafón y Remuneraciones que sistematicen su información para que así los usuarios obtengan resultados cuando hagan el trámite respectivo y sea en un menor tiempo de lo actual.

Para la presente investigación se tomó como herramienta de desarrollo el software libre, sin embargo existen otros lenguajes de programación con licencia que se pueden utilizar para futuros proyectos y así incluir nuevas mejoras que fortalezcan su robustez y velocidad.



REFERENCIAS

- Ardila, N. I. (13 de Marzo de 2013). *blogspot*. Obtenido de <http://actividadreconocimiento-301569-8.blogspot.pe/2013/03/norma-de-evaluacion-isoiec-9126.html>
- Ayala, A. P. (2006). *Ingeniería de Software - una guía para crear sistemas de información* (1era ed.). Mexico.
- Ben, L. (2005). *Software libre, php y mysql. Tecnologías para el desarrollo de aplicación web*. España: Diaz de Santos.
- Block. (s.f.). *basededatos.umh.es*. Obtenido de http://basededatos.umh.es/e_r.htm#entidades_relaciones
- E.K., J. (2005). *Análisis y diseño de sistemas*. Mexico: pearson; educacion de mexico S.A.
- Eugenia, B. (s.f.). *Poo y mvc en php, CreativeCommons 3.0*.
- Foundation, C. S. (2016). *CakePhp Cookbook Documentation Release 3.X*.
- Galindo, R. M. (2012). *Análisis, diseño e implementación de un sistema de información aplicado a la gestión educativa en centros de educación especial*. Lima.
- García, C. A., & Marín Mazo, E. (2005). *Guía Técnica para Evaluación de Software* (1era ed.).
- Grady, B. (1996). *análisis y diseño orientado a objetos*. wesley iberoamericana: Edit Addison.
- Jimenez, D. S. (2009). *Análisis y diseño de un sistema de trámite de documentos de pago a proveedores via intranet*. Lima: Creative Commons.
- kendall, & Kendall. (2005). *Análisis y diseños de sistemas* (6ta ed.). Mexico: pearson educacion S.A.

- Letelier, P., & Penades, M. c. (2003). *Metodologías ágiles para el desarrollo del software extreme programming (XP)*.
- Mendoza, L. d. (2011). *Implementación de un sistema automatizado que optimice la gestión de los procesos administrativos del área servicios médicos*. Maturin, Venezuela .
- Peña, A. A. (2006). *Ingeniería de software - guía para crear sistemas de información* (1era ed.). México.
- Raymundo. (s.f.). *RD ray-design*. Obtenido de http://www.ray-design.com.mx/psicoparaest/index.php?option=com_content&view=article&id=232:t-student-dr&catid=52:pruebaspara&Itemid=61
- Rumbaugh, J., Jacobson, I., & Booch, G. (s.f.). *Lenguaje unificado de modelado, manual de referencia* (2da ed.). Madrid: pearson educacion S. A.
- smartsys. (s.f.). Obtenido de <http://www.bemuserp.com/smartsys/?p=391>
- Sommerville, I. (2005). *Ingeniería del Software* (7mo ed.). (M. M. Romo, Ed.) Madrid , España: Pearson Educacion S. A. .
- Taboada, J. (2011). *Fundamentos UML* (1ra ed.). Madrid: Anaya S. A.
- Wikipedia. (16 de Octubre de 2016). Obtenido de https://es.wikipedia.org/wiki/ISO/IEC_9126



ANEXOS

Anexo N° 01**ENCUESTA ANTES DE LA IMPLEMENTACIÓN DEL SISTEMA (UPL)**

Instrucciones: lea detenidamente cada uno de los ítems. Encierre en un círculo y/o marque con una (x) en cada ítem, solo una de las alternativas que mejor sugiera su opinión.

1. ¿Demora de emisión de reportes?

- a) 1 hora a 2 horas
- b) 2 hora a 3 horas
- c) 3 horas a más

2. A tenido alguna vez confusión de datos en sus reportes

- a) Siempre
- b) De vez en cuando
- c) Nunca

3. Cree Ud. Que la reincidencia de información de constancias emitidas se deben a la falta de:

- a) Tecnología
- b) Personal
- c) Tiempo

Anexo N° 02**ENCUESTA DESPUÉS DE LA IMPLEMENTACIÓN DEL SISTEMA (UPL)**

Instrucciones: lea detenidamente cada uno de los ítems. Encierre en un círculo y/o marque con una (x) en cada ítem, solo una de las alternativas que mejor sugiera su opinión.

4. ¿Demora de emisión de reportes?

- a) 0.10 horas a 0.40 horas
- b) 0.40 horas a 1.10 horas
- c) 1.10 horas a más

5. A tenido alguna vez confusión de datos en sus reportes

- a) Siempre
- b) De vez en cuando
- c) Nunca

d) ¿Cuál es su apreciación Hacia el sistema de consulta de perfiles (UPL)?

- a) Buena
- b) Regular
- c) Mala

ANEXO N° 03

Resultados de las encuestas realizadas a los usuarios y sus respectivos cálculos.

TIEMPO	ANTES	DESPUES	d	$d - \bar{d}$	$(d - \bar{d})^2$
1	3.40	1.00	2.4	0.8805	0.77528025
2	2.50	1.12	1.38	-0.1395	0.01946025
3	2.15	0.55	1.6	0.0805	0.00648025
4	1.58	1.00	0.58	-0.9395	0.88266025
5	2.18	0.40	1.78	0.2605	0.06786025
6	2.30	1.09	1.21	-0.3095	0.09579025
7	2.20	1.08	1.12	-0.3995	0.15960025
8	1.55	1.00	0.55	-0.9695	0.93993025
9	2.15	1.10	1.05	-0.4695	0.22043025
10	2.25	1.14	1.11	-0.4095	0.16769025
11	2.10	0.45	1.65	0.1305	0.01703025
12	2.33	0.50	1.83	0.3105	0.09641025
13	2.45	0.30	2.15	0.6305	0.39753025
14	2.25	1.05	1.2	-0.3195	0.10208025
15	2.30	0.30	2	0.4805	0.23088025
16	2.30	0.42	1.88	0.3605	0.12996025
17	2.04	0.20	1.84	0.3205	0.10272025
18	1.45	0.10	1.35	-0.1695	0.02873025
19	2.40	0.59	1.81	0.2905	0.08439025
20	2.20	0.30	1.9	0.3805	0.14478025
Suma	44.08	13.69	30.39		4.669695
Promedio	2.204	0.6845	1.5195		

Fuente: Elaboración propia

Anexo N° 04

**UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERIA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**



MANUAL DE USUARIO



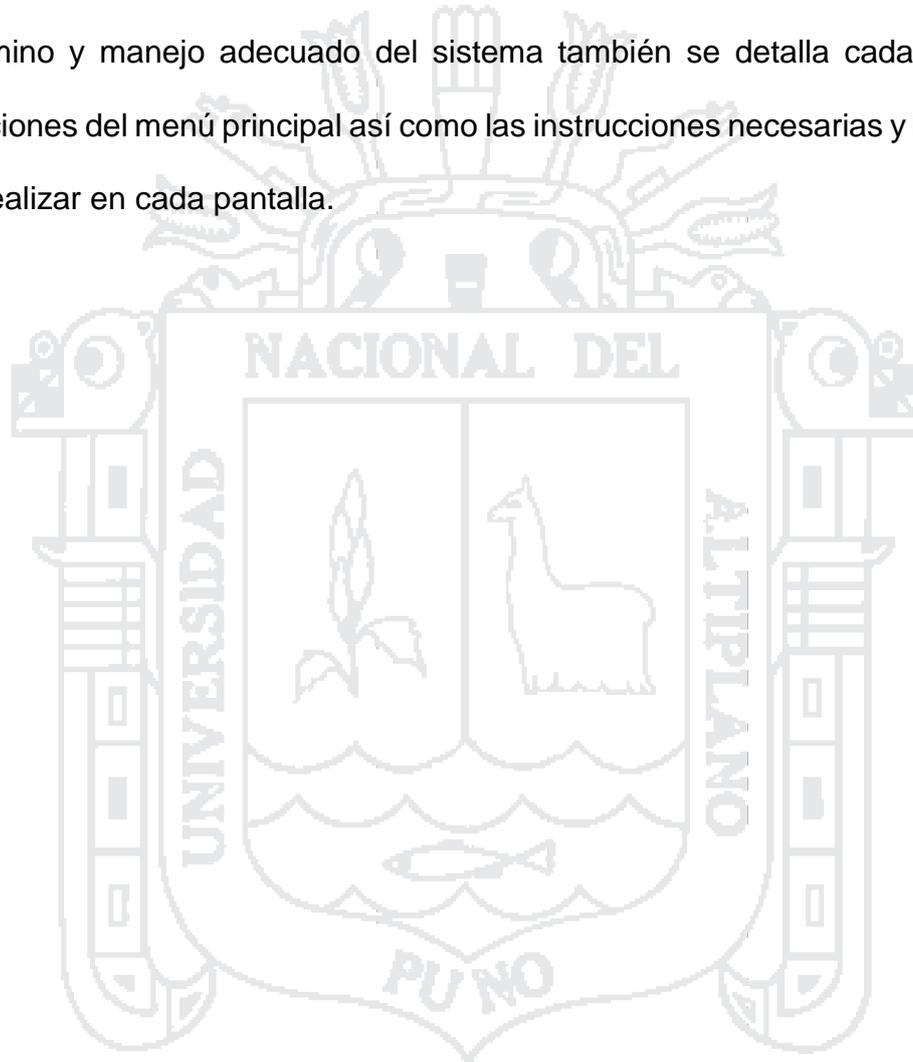
**“SISTEMA DE CONSULTA DE
PERFILES DE LA UNIDAD DE
PENSIONES Y LIQUIDACIONES
DE LA UNIVERSIDAD
NACIONAL DEL ALTIPLANO
PUNO”**

Elaborado por Bach: Nélide Sonia Jihuallanca Ccoa

Puno octubre 2016

PRESENTACION

En este documento se presenta el manual de usuario del sistema de consulta de perfiles para la UNIDAD DE PENSIONES Y LIQUIDACIONES que tiene como objetivo proporcionar una guía práctica a los encargados de dicha área, para el dominio y manejo adecuado del sistema también se detalla cada una de las opciones del menú principal así como las instrucciones necesarias y las acciones a realizar en cada pantalla.



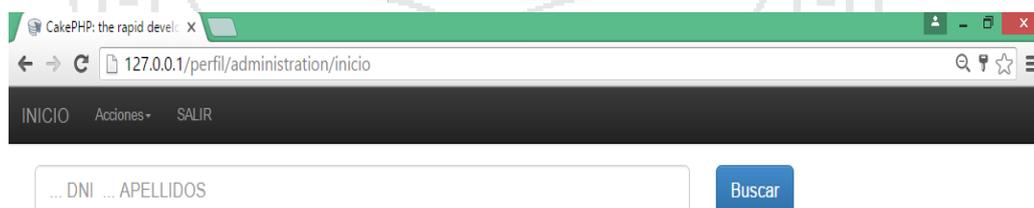
1. INGRESO AL SISTEMA

1.1. Ingreso al Sistema condición Administrador

Aquí se muestra la primera pantalla para el ingreso del administrador al sistema.



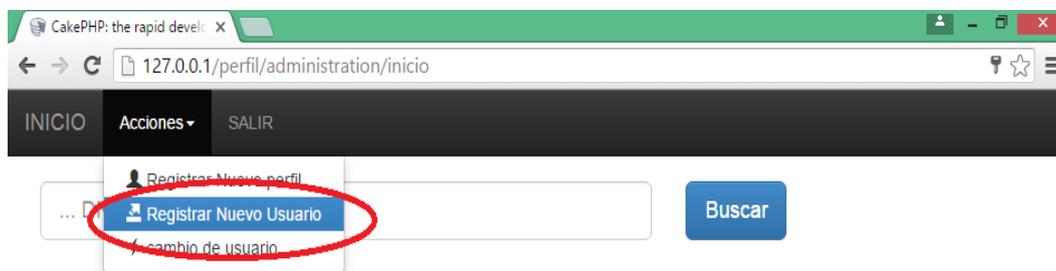
Nos mostrara lo siguiente:



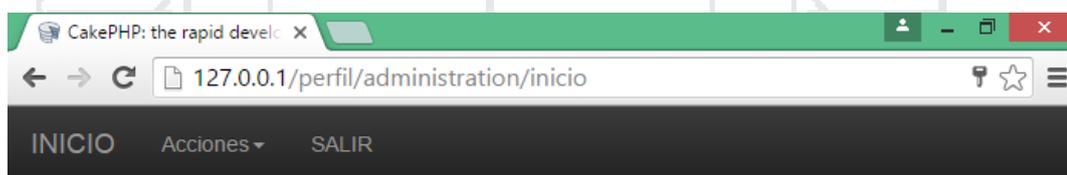
Ahora si tenemos acceso a todo el personal que nos permite visualizar, subir archivos de cada personal

1.2. Registrar nuevo usuario

Haga clic en la opción desplegable “Acciones” y Registrar nuevo usuario.



Una vez ingresado se deberá de llenar todos los parámetros:



Registrar Usuario

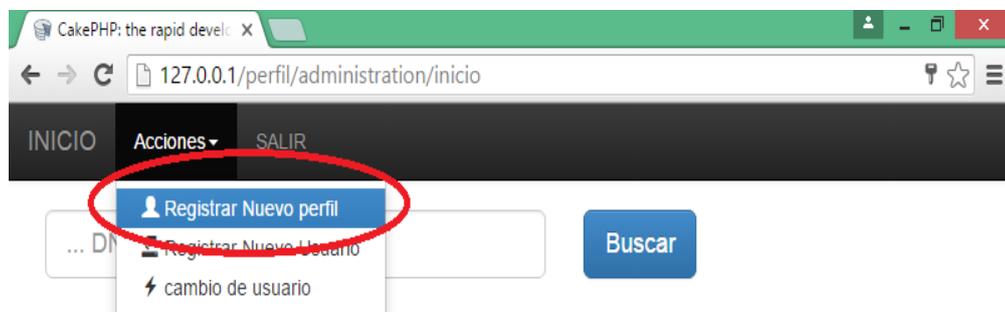
Usuario
password
Nombre completo
Nro documento
Seleccione NIVEL

Registrar

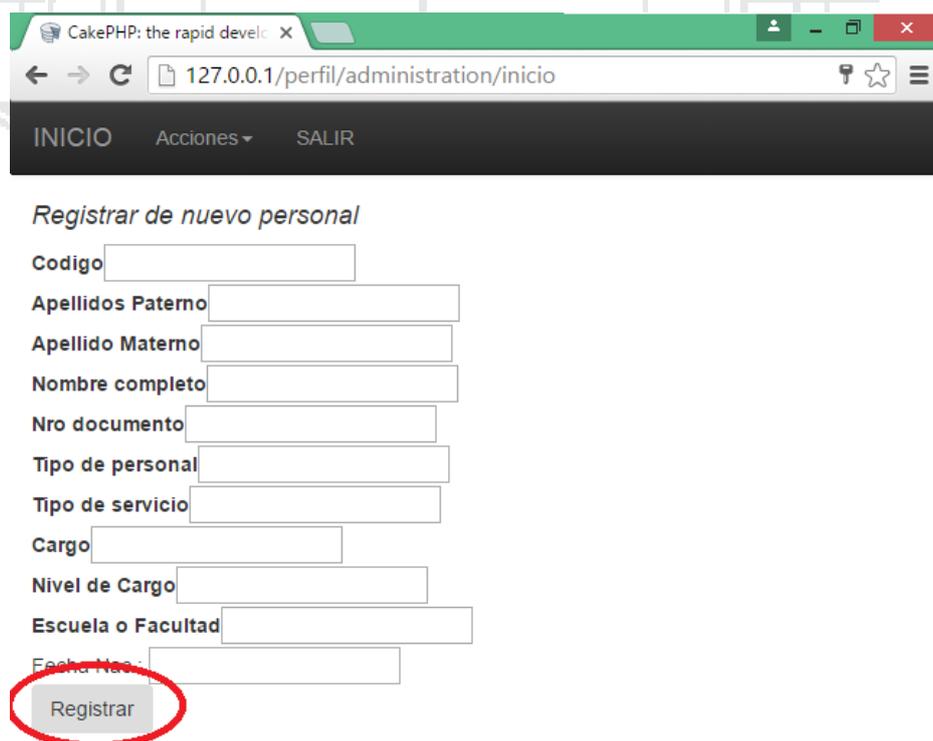
Después de haber llenado los campos del formulario haga clic en registrar. Recordar en usuario y contraseña que escribió.

1.3. Registrar nuevo perfil

Haga clic en la opción desplegable “Acciones” y Registrar nuevo perfil.



Una vez ingresado se deberá de llenar todos los parámetros:


 A screenshot of a web browser window showing a registration form titled 'Registrar de nuevo personal'. The form contains several input fields: 'Codigo', 'Apellidos Paterno', 'Apellido Materno', 'Nombre completo', 'Nro documento', 'Tipo de personal', 'Tipo de servicio', 'Cargo', 'Nivel de Cargo', 'Escuela o Facultad', and 'Fecha Nac:'. A 'Registrar' button is located at the bottom of the form and is highlighted with a red circle.

Después de haber llenado los campos del formulario haga clic en registrar y se guardara un nuevo perfil en la base de datos.

- **Personal docentes administrativos y contratados:** se llevara el registro y control de cada personal teniendo en cuenta los años de servicios.

- **Personal administrativo:** registro y actualización permanente de cada administrativo de la universidad nacional del altiplano puno.

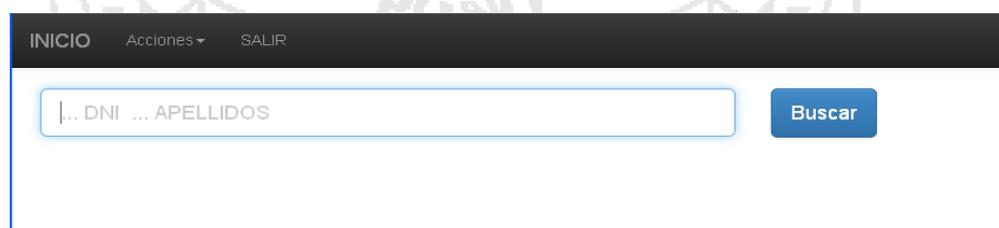
Una vez llenado todos los datos se dará un click en aceptar para confirmar el nuevo registro.



1.4. Para el administrador y usuarios

Buscador:

El sistema cuenta con un buscador de beneficiarios que está ubicado en el la parte superior de la pantalla principal del sistema, la cual nos ayudara a obtener la información más relevante del personal (docente y administrativo), haga click en el icono que se encuentra en el lado derecho de la pantalla del entorno del sistema.



La búsqueda de los beneficiarios puede ser por su DNI, APELLIDOS o NOMBRES

INICIO Acciones SALIR

Busqueda de Personal ... (Para ver datos adjuntos ir a Adjuntar)

Mostrar 10 registros. Buscar

id	Nro Doc	Ape. y Nomb.	Tipo de personal	Tipo de servicio	Cargo	Editar	Adjuntar
227	01202467	CHAHUARES FLORES, FREDY	ADMINISTRATIVO	NOMBRADO	ANALISTA SISTEMAS PAD II	EDITAR	Adjuntar
237	01300516	FLORES MENDOZA, FREDY LUIS	ADMINISTRATIVO	NOMBRADO	TECNICO ADMINISTRATIVO III	EDITAR	Adjuntar
286	01292554	REYES APAZA, FREDY RUBEN	ADMINISTRATIVO	NOMBRADO	ESPECIALISTA ADMINISTRATIVO I	EDITAR	Adjuntar
347	01322191	PARI ARCAYA, JOEL FREDY	ADMINISTRATIVO	NOMBRADO	DIRECTOR SIST.ADM.II-ABOGADO	EDITAR	Adjuntar
375	01319519	ARUHUANCA MAQUERA, EDWIN FREDY	ADMINISTRATIVO	NOMBRADO	ESPECIALIS.ADMINISTR.IV-RESPONS	EDITAR	Adjuntar
397	02447196	COLLANGUI MARTINEZ, FREDY	ADMINISTRATIVO	NOMBRADO	ANALISTA SIST. PAD IV-JEF UNID	EDITAR	Adjuntar
482	01200802	QUISPE QUISPE, FREDY	ADMINISTRATIVO	NOMBRADO	DIRECTOR SIST.ADM.II-JEFE UNID	EDITAR	Adjuntar
489	01234752	APAZA ATENCIO, JULIO FREDY	ADMINISTRATIVO	NOMBRADO	TECNICO ADMINISTRATIVO III	EDITAR	Adjuntar
591	02064462	BENDITA CAMASITA, FREDY	ADMINISTRATIVO	CONTRATADO	TECNICO ADMINISTRATIVO I	EDITAR	Adjuntar
627	01332920	VELASQUEZ JALLO, VICTOR FREDY	ADMINISTRATIVO	CONTRATADO	DIRECTOR SISTEMA ADMINISTR.II	EDITAR	Adjuntar

Mostrando 1 al 10 de 21 registros

← Previous 1 2 3 Next →

Para listar el personal específico haga click en el icono que se encuentra en la parte superior derecha de la pantalla del entorno del sistema.



Aparecerá el siguiente cuadro donde se muestra el registro específico que desea buscar.



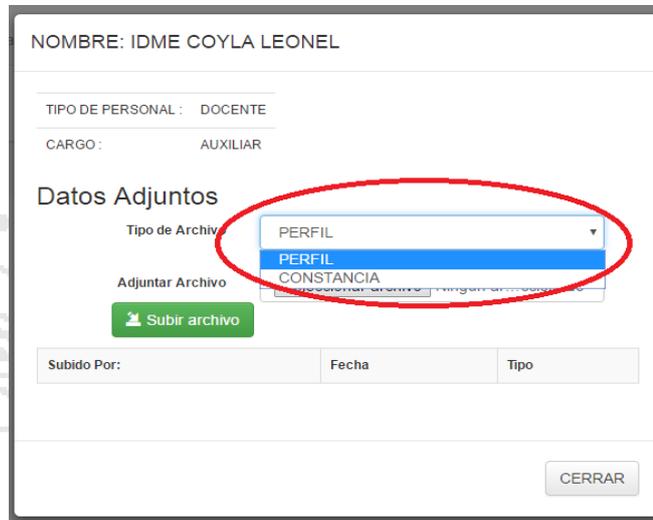
Para poder modificar los datos de cada personal haremos un click en editar

1.5. Subir perfiles

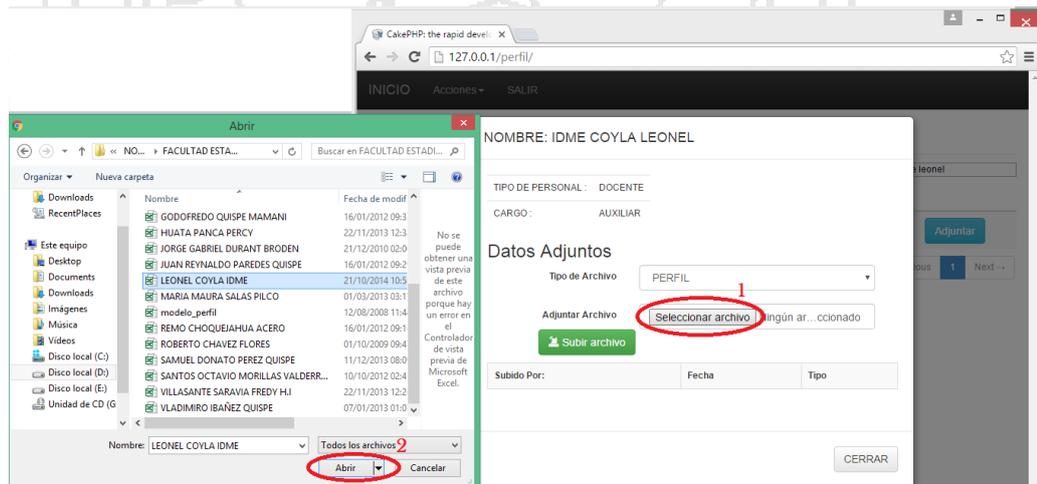
Para subir un perfil encontrara un botón en la lista de cada personal, dándole click en el botón <<“ADJUNTAR” >>



Después de adjuntar un nuevo perfil se podrá observar la siguiente pantalla:



Seguidamente se hace click en el botón desplegable nos da la opción de seleccionar “PERFIL o CONSTANCIA” como se observa en la imagen, seleccionando lo deseado a continuación seleccionamos el archivo requerido haciendo clic en <<seleccionar archivo>> después de haber seleccionado el perfil hace clic en <<abrir>>



Seguidamente se hace clic en el botón <<Subir archivo>>.

Adjuntar Archivo



Una vez ya subido los archivos nos mostrara una ventana de confirmación del perfil y constancia de cada personal (docentes y administrativos) a la vez mostrará el nombre del administrador que realizó la operación.

NOMBRE: IDME COYLA LEONEL

TIPO DE PERSONAL : DOCENTE

CARGO : AUXILIAR

Datos Adjuntos

Tipo de Archivo: PERFIL

Adjuntar Archivo: Seleccionar archivo Ningún ar...ccionado

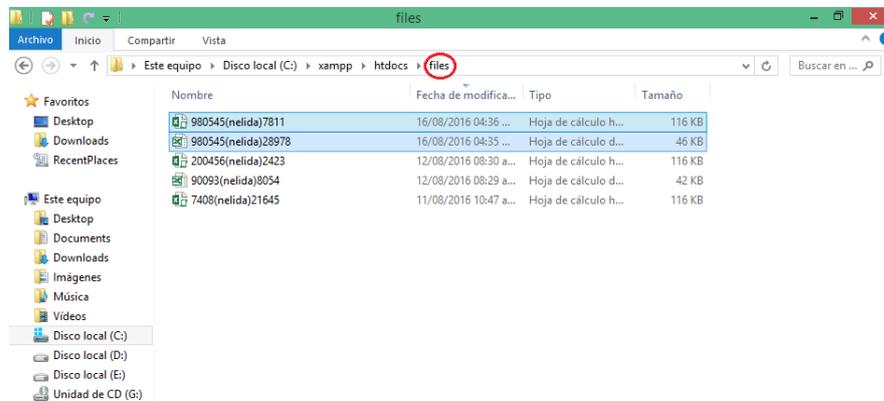
Subir archivo

Subido Por:	Fecha	Tipo	
940651(nelida)23133.xlsm	2016-08-18 03:47:38	CONSTANCIA	Eliminar
940651(nelida)11316.xls	2016-08-18 03:47:11	PERFIL	Eliminar

CERRAR

NOTA: Los archivos que son permitidos son los formatos “xls (Perfil) y xlsm (Constancia)”.

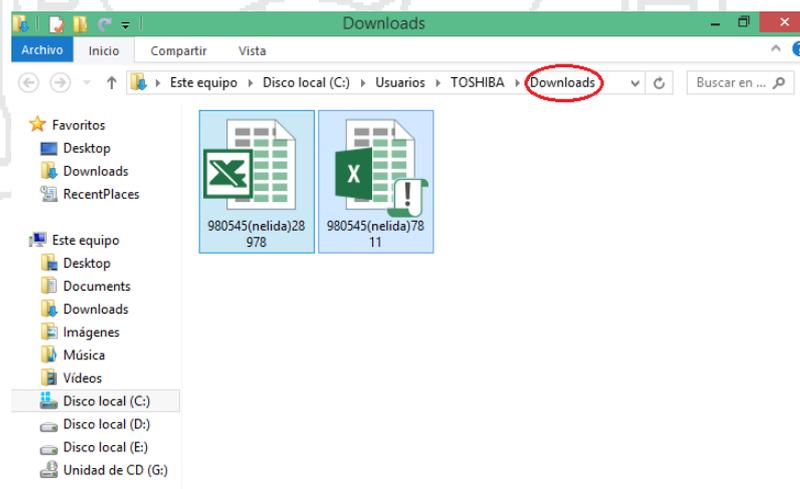
Después de haber realizado las operaciones pertinentes hacemos clic en el botón <<CERRAR>>. La información se almacenara en la carpeta “files”



Quando se desea actualizar la información de un trabajador (Docente o administrativo) se descarga el archivo.

Subido Por:	Fecha	Tipo	
980545(nelida)7811.xlsm	2016-08-16 23:36:19	CONSTANCIA	Eliminar
980545(nelida)28978.xls	2016-08-16 23:35:18	PERFIL	Eliminar

La descarga se guardara en la carpeta Downloads



Finalizando ya el uso del sistema daremos un click en el botón << salir>>.