

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELÉCTRONICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“SISTEMA WEB BASADO EN LA NORMA ISO/IEC 25010:2010
(SQuaRE) PARA LA GESTIÓN DE INFORMACIÓN EN EL
COLEGIO DE INGENIEROS DEL PERÚ CD PUNO - 2013”**

TESIS

PRESENTADA POR:

**JACKELINE BENITEZ LLANQUE
ALODIA FLORES ARNAO**

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

PUNO – PERÚ

2013

UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELÉCTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

“SISTEMA WEB BASADO EN LA NORMA ISO/IEC 25010:2010
(SQuaRE) PARA LA GESTIÓN DE INFORMACIÓN EN EL COLEGIO
DE INGENIEROS DEL PERÚ CD PUNO - 2013”

TESIS

PRESENTADA POR:

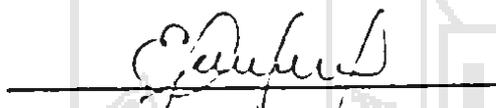
JACKELINE BENITEZ LLANQUE
ALODIA FLORES ARNAO

PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO DE SISTEMAS

APROBADA POR EL JURADO REVISOR CONFORMADO POR:

PRESIDENTE

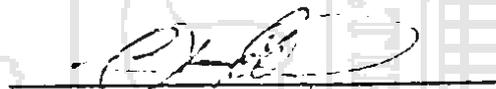
:



M.Sc. Edelfré Flores Velásquez

PRIMER MIEMBRO

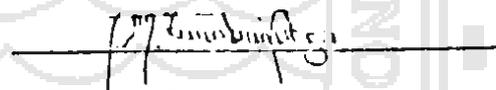
:



Mg. Oliver Amadeo Vilca Huayta

SEGUNDO MIEMBRO

:



Ing. Milder Zanabria Ortega

DIRECTOR DE TESIS

:



Mg. Carlos Borly Soya Maydana

ÁREA: Informática

TEMA: Sistemas de información tradicionales y expertos

DEDICATORIA

*A mis queridos padres Juan y Soledad,
por su cariño, sus enseñanzas y consejos
por guiarme por el buen camino, por
todo el esfuerzo que hicieron para
darme lo mejor y hacer de mí una
persona de bien.*

*A mi adorado hermano Jose Luís,
porque su bienestar y su felicidad me
impulsan a seguir adelante y ser mejor
cada día.*

Jackeline

*A mis amados padres José Fernando y
Haydeé, por su lucha constante en la
vida, por creer en mí y por su eterno
cariño, y a mis hermanos Marilia, Jhon y
Naina, por brindarme su apoyo, por ser
parte del motor que me impulsa a seguir
adelante siempre. Por ustedes, por el
inmenso amor que les tengo, por esa
fortaleza que poseen y por lo que han
hecho y hacen por mí.*

Alodia

AGRADECIMIENTOS

A Dios por mostrarnos día a día que con humildad, paciencia y esfuerzo todo es posible.

A nuestros padres y hermanos, por su apoyo incondicional, su comprensión y su fe en nosotras, a ellos nuestra gratitud eterna por estar siempre a nuestro lado.

A nuestros grandes amigos, por siempre exhortarnos a culminar cada tarea iniciada en la vida, como lo es esta investigación, en especial a Anita, Crist, Larisa, Mónica, Edgar, Chris, Edu, Cristhian, Jonathan y al Sr. Víctor.

A los ingenieros miembros del comité de jurados, M.Sc. Edelfré Flores Velásquez, Mg. Oliver A. Vilca Huayta e Ing. Milder Zanabria Ortega, por sus observaciones, sugerencias, consejos y su guía para la realización de la presente investigación.

Al director de tesis, Mg. Carlos B. Sosa Maydana por su dirección en el proyecto de investigación.

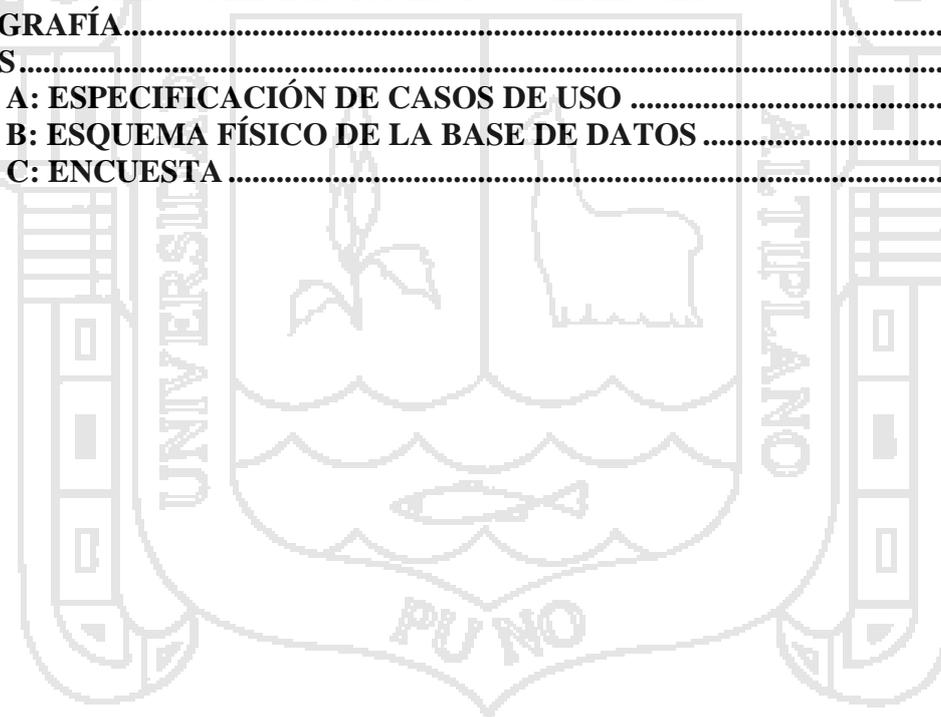
A toda la familia del Colegio de Ingenieros del Perú CD Puno de la gestión 2010 – 2012, dirigida por el Ing. Nicolás Luza Flores, a él un agradecimiento especial por darnos la oportunidad de ser parte de esa familia. Y al Dr. Norman Jesús Beltrán Castañón, miembro del cuerpo directivo, por su apoyo y colaboración en la ejecución de la presente investigación.

A todos, muchas gracias...

ÍNDICE

ÍNDICE DE FIGURAS	7
ÍNDICE DE TABLAS	11
RESUMEN.....	14
INTRODUCCIÓN	18
CAPÍTULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN	21
1.1. PLANTEAMIENTO DEL PROBLEMA	22
1.2. DEFINICIÓN DEL PROBLEMA	24
1.3. JUSTIFICACIÓN.....	24
1.4. OBJETIVOS DE LA INVESTIGACIÓN	26
1.4.1. OBJETIVO GENERAL	26
1.4.2. OBJETIVOS ESPECÍFICOS.....	26
1.5. HIPÓTESIS.....	26
1.6. LIMITACIONES	27
CAPÍTULO II. MARCO TEÓRICO	28
2.1. ANTECEDENTES.....	29
2.2. MARCO CONCEPTUAL.....	32
2.2.1. SISTEMA WEB	32
2.2.2. ORGANIZACIÓN DE LA ISO E IEC	40
2.2.3. CALIDAD DE SOFTWARE	41
2.2.4. MODELOS DE CALIDAD	44
2.2.5. NORMATIVAS DE CALIDAD.....	53
2.2.6. NORMA INTERNACIONAL ISO/IEC 25000 (SQuaRE)	56
2.2.6.1. Divisiones de la Norma ISO/IEC 25000 (SQuaRE)	56
2.2.6.2. Modelo de la Medida de Calidad del Software	60
2.2.6.3. Modelo de Calidad ISO/IEC 25010 (SQuaRE)	61
2.2.7. LA INFORMACIÓN Y LA GESTIÓN DE INFORMACIÓN	69
2.3. DEFINICIÓN DE TÉRMINOS BÁSICOS.....	77
CAPÍTULO III. MATERIALES Y MÉTODOS	80
3.1. TRABAJO EXPERIMENTAL	81
3.1.1. METODOLOGÍA DE LA INVESTIGACIÓN	81
3.1.2. DISEÑO DE LA INVESTIGACIÓN.....	81
3.1.3. POBLACIÓN Y MUESTRA	82
3.2. SISTEMA DE VARIABLES	82
3.2.1. DEFINICIÓN DE VARIABLES	82
3.2.2. OPERACIONALIZACIÓN DE VARIABLES	83
3.3. MATERIAL EXPERIMENTAL.....	85
3.3.1. MÉTODOS DE RECOPIACIÓN DE DATOS	85
3.3.2. MÉTODO DE TRATAMIENTO Y ANÁLISIS DE DATOS	86
3.4. MATERIAL APLICATIVO.....	87
3.4.1. METODOLOGÍA DE DESARROLLO DEL SISTEMA.....	87
3.4.2. HARDWARE Y SOFTWARE DE DESARROLLO	90
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN.....	91
4.1. ANÁLISIS Y DISEÑO DEL SISTEMA WEB	92
4.1.1. ANÁLISIS DEL SISTEMA WEB.....	92
4.1.1.1. Actores del Sistema Web.....	92
4.1.1.2. Casos de Uso del Sistema Web	92

4.1.1.3. Especificación de Casos de Uso	98
4.1.1.4. Diagramas de Secuencia	98
4.1.1.5. Diagramas de Actividades	103
4.1.1.6. Diagrama de Clases	106
4.1.2. DISEÑO DEL SISTEMA WEB	108
4.1.2.1. Modelo Conceptual E-R	108
4.1.2.2. Modelo Lógico	109
4.1.2.3. Modelo Físico	110
4.2. IMPLEMENTACIÓN DEL SISTEMA WEB	119
4.2.1. FRAMEWORK UTILIZADO	120
4.2.2. COMPONENTES UTILIZADOS	120
4.2.3. INTERFACES PRINCIPALES DEL SISTEMA WEB	121
4.3. PRUEBA Y VALIDACIÓN DEL SISTEMA WEB UTILIZANDO EL MODELO DE CALIDAD DE LA NORMA ISO/IEC 25010:2010 (SQuaRE)	129
4.3.1. PRUEBA DEL SISTEMA WEB	129
4.3.2. VALIDACIÓN DEL SISTEMA WEB UTILIZANDO EL MODELO DE CALIDAD DE LA NORMA ISO/IEC 25010:2010 (SQuaRE)	133
4.4. PRUEBA DE HIPÓTESIS	147
4.4.1. ESTADÍSTICO DE PRUEBA	148
4.4.2. OPERACIONALIZACIÓN DE LA HIPÓTESIS	150
4.4.3. RESULTADOS DE LA PRUEBA DE HIPÓTESIS	151
CAPÍTULO V. CONCLUSIONES	157
CAPÍTULO VI. RECOMENDACIONES Y SUGERENCIAS	159
BIBLIOGRAFÍA	161
ANEXOS	165
ANEXO A: ESPECIFICACIÓN DE CASOS DE USO	166
ANEXO B: ESQUEMA FÍSICO DE LA BASE DE DATOS	182
ANEXO C: ENCUESTA	197



ÍNDICE DE FIGURAS

Figura 1 Diagrama de Flujo de Aplicación de CodeIgniter	38
Figura 2 Modelo de calidad de McCall (1977).....	48
Figura 3 Diferentes visiones del modelo de calidad de McCall.	49
Figura 4 Modelo de Boehm para clasificar los criterios de calidad.....	50
Figura 5 Relación entre los estándares internacionales ISO/IEC 9126 e ISO/IEC 14598.	55
Figura 6 Ciclo de vida de la calidad en ISO.	55
Figura 7 Organización de la serie de estándares de SQuaRE.	56
Figura 8 Modelo de medición de la calidad del producto software según SquaRE.	61
Figura 9 Estructura del modelo calidad.	62
Figura 10 Características para la calidad del producto.	63
Figura 11 Características para la calidad en uso.	68
Figura 12 El modelo espiral de Boehm.....	89
Figura 13 CU para la gestión de usuarios.	93
Figura 14 CU para la gestión de registro de colegiaturas.	94
Figura 15 CU para la gestión de asignación de CIP.....	95
Figura 16 CU para la gestión de programación de colegiaturas.	95
Figura 17 CU para la gestión de colegiaturas.	96
Figura 18 CU para la gestión de pagos por juramentación.	96
Figura 19 CU para el pago de aportaciones.	97
Figura 20 CU para la consulta de aportaciones en línea.	97

Figura 21 Diagrama de secuencia para identificar al usuario.	98
Figura 22 Diagrama de secuencia para el registro de un ingeniero.	98
Figura 23 Diagrama de secuencia para registro de una especialidad a un ingeniero.	99
Figura 24 Diagrama de secuencia para registrar el pago de aportaciones de un ingeniero.	99
Figura 25 Diagrama de secuencia para verificar la existencia del CIP de un ingeniero.	100
Figura 26 Diagrama de secuencia para asignar un CIP a un ingeniero.	100
Figura 27 Diagrama de secuencia para el registro de una colegiatura.	101
Figura 28 Diagrama de secuencia para el registro de una fecha de colegiatura.	101
Figura 29 Diagrama de secuencia para verificar el concepto de pago.	102
Figura 30 Diagrama de secuencia para registrar el pago por juramentación.	102
Figura 31 Diagrama de secuencia para el registro de un pago.	103
Figura 32 Diagrama de actividad del proceso de registro de colegiatura.	103
Figura 33 Diagrama de de actividad del registro del colegiado.	104
Figura 34 Diagrama de actividad de la programación de colegiatura.	104
Figura 35 Diagramas de actividades del pago por juramentación y asignación de fecha de colegiatura.	105
Figura 36 Diagrama de actividad del cambio de estado a juramentado	105
Figura 37 Diagrama de actividad del proceso de pago de aportaciones u otro servicio.	106
Figura 38 Diagrama de clases del sistema Web.	107
Figura 39 Diagrama Entidad – Relación del sistema Web.	108

Figura 40 Modelo lógico de la base de datos del sistema Web (Parte I).	109
Figura 41 Modelo lógico de la base de datos del sistema Web (Parte II).	110
Figura 42 Interfaz de logeo.	121
Figura 43 Interfaz de inicio del sistema Web.	123
Figura 44 Interfaz con las opciones generales para el ingeniero.	124
Figura 45 Interfaz del detalle de pagos (Aportación).	125
Figura 46 Interfaz del detalle de pagos (Detalle de Aportación).	126
Figura 47 Interfaz del detalle de pagos (Servicio).	126
Figura 48 Interfaz del detalle de pagos (Colegiatura).	127
Figura 49 Impresión de voucher de pago.	128
Figura 50 Interfaz para la consulta de aportaciones en línea	129
Figura 51 Gráfico de comparación del pre-test y post-test.	132
Figura 52 Histograma de la frecuencia acumulada en la pregunta 1.	135
Figura 53 Histograma en la frecuencia acumulada en la pregunta 2.	136
Figura 54 Histograma de la frecuencia acumulada en la pregunta 3.	137
Figura 55 Histograma de la frecuencia acumulada en la pregunta 4.	138
Figura 56 Histograma de la frecuencia acumulada en la pregunta 5.	139
Figura 57 Histograma de la frecuencia acumulada en la pregunta 6.	140
Figura 58 Histograma de la frecuencia acumulada en la pregunta 7.	141
Figura 59 Histograma de la frecuencia acumulada en la pregunta 8.	142
Figura 60 Histograma de la frecuencia acumulada en la pregunta 9.	143
Figura 61 Histograma de la frecuencia acumulada en la pregunta 10.	144
Figura 62 Histograma de la frecuencia acumulada en la pregunta 11.	145
Figura 63 Histograma de la frecuencia acumulada en la pregunta 12.	146
Figura 64 Histograma de la frecuencia acumulada en la pregunta 13.	147

Figura 65 Histograma de la muestra pre-test.	152
Figura 66 Histograma de la muestra post-test.....	152
Figura 67 Histograma de la diferencia de las muestras (Pre-test – post-test).....	153
Figura 68 Prueba de hipótesis (Gráfica obtenida con MINITAB 16).....	155



ÍNDICE DE TABLAS

Tabla 1 Estado de la actualización de información de los miembros colegiados del CIP CD Puno.....	23
Tabla 2 Fases del ciclo de vida de la información.....	71
Tabla 3 Operacionalización de variables.....	83
Tabla 4 Límites de tamaño de ficheros del S.O.....	111
Tabla 5 Tabla ingeniero.....	112
Tabla 6 Tabla detalles.....	112
Tabla 7 Tabla cip.....	113
Tabla 8 Tabla familiar.....	113
Tabla 9 Tabla país.....	113
Tabla 10 Tabla departamento.....	114
Tabla 11 Tabla provincia.....	114
Tabla 12 Tabla distrito.....	114
Tabla 13 Tabla capítulo.....	115
Tabla 14 Tabla especialidad.....	115
Tabla 15 Tabla universidad.....	115
Tabla 16 Tabla ingeniero_has_especialidad.....	116
Tabla 17 Tabla colegiatura.....	116
Tabla 18 Tabla programación.....	117
Tabla 19 Tabla programación_has_colegiatura.....	117
Tabla 20 Tabla aportaciones.....	117
Tabla 21 Tabla pago.....	118

Tabla 22 Tabla pago_item.	118
Tabla 23 Tabla catalogo.	119
Tabla 24 Tabla catalogo_montos.....	119
Tabla 25 Presentación del calificativo para las alternativas de la encuesta.	130
Tabla 26 Intervalo de clase para los puntajes obtenidos de la encuesta.....	130
Tabla 27 Matriz de datos pre-test de la gestión de información en el CIP CD Puno.	130
Tabla 28 Matriz de datos post-test de la gestión de información en el CIP CD Puno.....	131
Tabla 29 Cuadro de preguntas planteado para la validación del sistema Web.....	134
Tabla 30 Frecuencia acumulada en la pregunta 1.	135
Tabla 31 Frecuencia acumulada en la pregunta 2.	136
Tabla 32 Frecuencia acumulada en la pregunta 3.	137
Tabla 33 Frecuencia acumulada en la pregunta 4.	138
Tabla 34 Frecuencia acumulada en la pregunta 5.	139
Tabla 35 Frecuencia acumulada en la pregunta 6.	140
Tabla 36 Frecuencia acumulada en la pregunta 7.	141
Tabla 37 Frecuencia acumulada en la pregunta 8.	142
Tabla 38 Frecuencia acumulada en la pregunta 9.	143
Tabla 39 Frecuencia acumulada en la pregunta 10.	144
Tabla 40 Frecuencia acumulada en la pregunta 11.	145
Tabla 41 Frecuencia acumulada en la pregunta 12.	146
Tabla 42 Frecuencia acumulada en la pregunta 13.	147
Tabla 43 Definición de variables para la contrastación entre el pre-test y post-test.....	148

Tabla 44 Contrastación entre el pre-test y post-test de las encuestas.	148
Tabla 45 Estadísticos de muestras relacionadas.....	153
Tabla 46 Correlación de muestras relacionadas.....	154
Tabla 47 Prueba de muestras relacionadas.....	154
Tabla 48 Prueba de muestras relacionadas.....	154



RESUMEN

El presente trabajo de investigación intitulado “Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE) para la Gestión de Información en el Colegio de Ingenieros del Perú CD Puno - 2013”, tiene como objetivo general desarrollar un sistema Web para mejorar la gestión de información en el Colegio de Ingenieros del Perú CD Puno. El sistema Web fue desarrollado para el Colegio de Ingenieros del Perú de la sede departamental Puno y sus comités provinciales, donde la gestión de información conlleva manejar una considerable cantidad de información por cada colegiado, sobre pagos por aportaciones, pagos por procesos de colegiaturas, pagos para la certificación de habilidades, y otros servicios. Esta información debe desempeñar la función para la que fue almacenada: servir al negocio y a la organización para prestar servicio de calidad a los colegiados, y ser compartida por toda la organización y los individuos que la componen.

La metodología de desarrollo de software utilizada para la presente investigación está basada en el modelo espiral de Boehm, el cual se adapta mejor a las posibles modificaciones que podría sufrir el sistema Web, y además, se enfoca en abordar los riesgos del producto software en forma incremental, de acuerdo a su prioridad. Así mismo, considera criterios de calidad relacionados con: los servicios que el sistema ofrece, la operación y mantenibilidad del sistema Web. Durante la ejecución de la investigación se cumplió objetivamente con cada actividad ligada a desarrollo; ya determinados los objetivos, se continuó con las tareas de análisis y diseño del sistema, donde se hizo el respectivo análisis de los requerimientos, para luego realizar el siguiente ciclo de desarrollo y pruebas. Esto se refleja las funcionalidades principales del sistema, aplicando así el modelo espiral de Boehm. Para la implementación del

sistema Web se consideraron las características de calidad interna y externa, y calidad en uso de la división del modelo de calidad de la Norma ISO/IEC 25010:2010 (SQuaRE). También se utilizaron herramientas de entorno Web, tales como CodeIgniter, framework seguro, ligero y que utiliza el patrón Modelo-Vista-Controlador (MVC), patrón de diseño orientado esencialmente a PHP5 (Hypertext Preprocessor), que permite crear aplicaciones rápidamente. El sistema de gestión de base de datos fue MySQL que trabaja directamente con CodeIgniter.

Luego de haber implantado el sistema, se realizaron las pruebas correspondientes, y teniendo ya los datos recopilados a través de las encuestas, se aplicaron una prueba de entrada (Pre-test) y una prueba de salida (Pos-test), para comprobar la hipótesis declarada; constituyendo un diseño de investigación pre-experimental. Los resultados obtenidos fueron claros al mostrar más de un 80% de aprobación del sistema.

Finalmente, una vez realizada la prueba de hipótesis, a través de los resultados de las encuestas, se demuestra empírica y estadísticamente, que el desarrollo de un “Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE) mejora la gestión de información en el Colegio de Ingenieros del Perú CD Puno”.

ABSTRACT

This paper titled "Web Application based on ISO/IEC Standard 25010:2010 (SQuaRE) for the Information Management in the Engineers Association of Peru CD Puno - 2013", has the overall objective to develop a Web application to improve the information management in the Engineers Association of Peru CD Puno. This Web application was developed for the Engineers Association of Peru CD Puno and its provincial committees where information management involves handling a considerable amount of information for each member as contributions payments, membership process fees, status certification of membership payments, and other services. All this information must perform the function for which it was stored: help the business and the organization to provide quality services to its members, and be shared throughout the organization and workers within it.

The software development methodology used for this research is based on the Boehm spiral model which best fits for this web application on possible modifications that could occur, and also focuses on addressing the risks of incremental methods of software products development according to its priority. Also, this methodology considers quality criteria relating to: services deployed by the application, operation of the system and maintainability of the Web application. The execution of the research allowed to carry out each activity linked to development itself. Once determined the objectives, the tasks of analysis and design of the system were completed, where there were possible to make the respective requirements analysis, and then perform the following cycle of development and tests. These cycles are reflected in the core functionality of the system, thus achieving to apply Boehm spiral model. For the implementation of the Web application it was considered the characteristics of internal

and external quality, and quality in use of the division of quality model of ISO/IEC 25010:2010 Standard (SQuaRE). Also tools of Web environment were used, such as CodeIgniter which is a safe, light and Model-View-Controller (MVC) framework, design pattern essentially oriented to PHP5 (Hypertext Preprocessor), which allows a rapid development of applications. The database management system used was MySQL that works directly with CodeIgniter.

After having implemented the application, the respective tests were performed, and having the data collected through the surveys, an entrance test (Pre-test) and an output test (Post-test) were applied to check hypothesis stated; constituting a pre-experimental design research. The results showed that more than 80% approval system.

Finally, once the hypothesis testing was made, through the survey results, it is shown empirically and statistically, that the development of a "Web Application based on ISO/IEC Standard 25010:2010 (SQuaRE) improve the management information on the Engineers Association of Peru CD Puno".

INTRODUCCIÓN

Cuando se aplican modelos de calidad, finalmente lo que se logra es que mejoren los procesos de software, así como la calidad del software que se desarrolla. Además, la gestión de calidad supone para la organización de hoy en día, un impacto estratégico y la oportunidad de tener una ventaja competitiva frente a otras organizaciones. Y teniendo en cuenta que el funcionamiento en el entorno Internet ayuda a producir un cambio realmente significativo: el centro neurálgico de los sistemas que estaba siendo ocupado por la red, al convertirse ésta en un medio, pasa a ser ocupado por la información almacenada. El crecimiento exponencial del volumen de información que se produce en todos los sectores económicos y el consiguiente crecimiento en la complejidad de la gestión de dicho volumen de información, explica por qué los sistemas y las soluciones de almacenamiento se basan en un modelo de calidad, y a su vez, se desplazan hacia el centro de la infraestructura tecnológica.

En estos momentos, y pensando en el futuro, no se concibe otra arquitectura que aquella en la que el sistema de almacenamiento de “datos / información / conocimiento” sea el centro neurálgico de la organización, sobre una solución perfectamente gestionada, segura, virtualizada y escalable (cuyo diseño, construcción y mantenimiento, requiere hardware, software y servicios). Este sistema de almacenamiento, a su vez, debe estar en red, de modo que se posibilite el acceso por cualquier servidor o cliente desde cualquier lugar y en cualquier momento, de modo que la información pueda desempeñar la función para la que fue almacenada: servir al negocio y a la organización para prestar servicio de calidad al usuario. Así mismo, con una buena gestión de la información, ésta podrá ser compartida por la organización y los individuos que la componen.

El Colegio de Ingenieros del Perú CD Puno, gestiona toda información relacionada al colegiado y sus aportaciones. Si se gestiona información de más de 5 mil profesionales agremiados a la sede, cifra que va en aumento mes tras mes, entonces la información ligada a los colegiados y los pagos de sus aportaciones u otros servicios, también tiene un crecimiento significativo. Las tareas de gestión de información, como el llevar el control de aportaciones por cada ingeniero colegiado, se realiza de manera aislada y poco coordinada con sus comités provinciales, dificultándose la gestión de información, ocasionando pérdida de tiempo durante el pago de aportaciones, insatisfacción en los agremiados, derivando éstas fallas en la pérdida de miembros y menos ingresos para la sede.

Esta tesis está constituida por seis capítulos:

El primer capítulo, denominado planteamiento de la investigación, se plantea el problema de investigación y los objetivos, se genera la hipótesis que se quiere demostrar y se establece el escenario de investigación.

El segundo capítulo, denominado marco teórico, se constituyen los antecedentes de la investigación, el marco teórico comprende todo lo relacionado con la norma ISO/IEC 25010:2010 (SQuaRE), la gestión de la información y la definición de términos básicos, describiéndose sintéticamente algunos de los principales conceptos.

El tercer capítulo, denominado materiales y métodos, está constituido por el trabajo experimental, sistema de variables, material experimental, métodos de recopilación de datos, método de tratamiento de datos y finalmente se describe la prueba de hipótesis.

El cuarto capítulo, denominado resultados y discusión, está constituido por el desarrollo de cada objetivo específico de la investigación.

El quinto capítulo, denominado conclusiones, detalla las conclusiones obtenidas del trabajo de investigación.

El sexto capítulo, denominado recomendaciones, menciona los aportes y las recomendaciones aplicables para trabajos de investigación futuros.

Finalmente las referencias bibliográficas y anexos.





**CAPÍTULO I. PLANTEAMIENTO DE LA
INVESTIGACIÓN**

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad las organizaciones son conscientes de la importancia de gestionar adecuadamente sus recursos económicos y sobre todo sus recursos de información. Así mismo, se preocupan por invertir en nuevas tecnologías, donde el software es uno de los componentes básicos, el mismo que juega un papel importante en el desarrollo de las organizaciones, no sólo como soporte a los procesos del negocio, productivos y administrativos, sino como parte de las estrategias corporativas para generar ventajas competitivas, realizar sus actividades de manera eficiente y contribuir al logro de sus objetivos. Entonces, es difícil imaginar alguna situación en la que el software no esté presente, ya que constantemente se pasan a producción distintos tipos de programas para diferentes clases de clientes, y para cada necesidad.

El Colegio de Ingenieros del Perú, una institución de gran prestigio a nivel nacional que agrupa a profesionales de las diferentes especialidades de la ingeniería, no es ajeno a los avances de la tecnología. Es así que en el Colegio de Ingenieros del Perú Consejo Departamental Puno (en adelante CIP CD Puno), se cuenta con un sistema de información para el pago de aportaciones, pago para la emisión de certificados de habilidad y trámites de colegiatura.

El sistema opera de la siguiente manera:

- El funcionamiento del sistema actual se restringe a tres computadoras a la vez, con la condición que éstas se encuentren en la misma red local.
- El acceso al sistema actual desde el Comité Provincial San Román Juliaca, se realiza mediante la configuración de una red privada virtual, con la cual muchas veces no se establece la conexión. Por esta razón, el control y registro de aportaciones realizados en el Comité Provincial San Román Juliaca es quincenal (dos veces al mes), tarea que realiza el personal asignado de manera manual. Y

considerando que los miembros que realizan sus aportaciones representan el 20% de colegiados hábiles y no hábiles del CIP CD Puno, el sistema debería estar disponible y ser de fácil acceso desde cualquier comité propio del CIP CD Puno.

- El manejo de información del sistema actual tiene como características la duplicidad, inconsistencia, no disponibilidad, no confiabilidad e inseguridad de la información, lo que hace difícil su uso y mantenibilidad (actualización, mantenimiento y administración).

A continuación se muestra el detalle de ingenieros que cuentan con una información actualizada y no actualizada:

Tabla 1

Estado de la actualización de información de los miembros colegiados del CIP CD Puno.

Estado de ingenieros	Estado de la Información		Total
	Actualizada	No Actualizada	
Hábiles	1349	278	1627
No Hábiles	2769	640	3409
	4118	918	5036

Fuente: Elaboración propia.

Entonces, un 81.7% de ingenieros tiene la información actualizada, mientras que el otro 18.3% representa a los ingenieros que carecen de ella. La información sólo actualizada no implica tener las demás características propias de una información de calidad.

- El sistema actual no cuenta con módulos para el registro de información adicional, ya sea sobre los familiares y/o especialidades que tenga el colegiado, pues un ingeniero puede contar con más de una especialidad.
- Así mismo, no se tiene un control sobre el proceso de colegiación de los ingenieros. Es decir, no se guarda un registro del proceso de colegiación en el sistema, desde la

etapa de inscripción (lo que incluye realizar el pago por derecho de colegiatura, por modalidad de obtención del título, por juramentación y por el FUT).

Teniendo en cuenta que el CIP CD Puno tiene más de 5 mil ingenieros agremiados (cifra que va en aumento mensualmente), los cuales radican en diferentes distritos de la región, y siendo su principal necesidad conocer el estado de sus aportaciones para poder realizar el pago respectivo y contar con la habilidad para su ejercicio profesional. Necesidad que no se satisface por las múltiples deficiencias que presenta el sistema actual, generando además, pérdida de tiempo y dinero.

1.2. DEFINICIÓN DEL PROBLEMA

¿En qué medida mejorará la gestión de información en el CIP CD Puno con el desarrollo de un Sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE)?

1.3. JUSTIFICACIÓN

Hoy en día la calidad ha dejado de ser un tópico y es necesario que forme parte de los productos o servicios que se ponen a disposición de los clientes y/o empresas.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto software y la calidad del proceso de desarrollo de éste (calidad de diseño y fabricación). No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos a establecer de calidad del proceso de desarrollo, ya que la calidad del primero va a depender, entre otros aspectos, de ésta. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

La importancia que las tecnologías Web han cobrado en los últimos años hace que se preste una atención especial a su calidad. Una forma de asegurar esta calidad es mediante la utilización de modelos que permitan controlar la calidad de los diseños realizados.

En general, un modelo de calidad descompone la calidad jerárquicamente en una serie de características y sub características que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad y usan generalmente como base la ISO 9126, y para el desarrollo de esta investigación, la extensión de esta norma, la ISO/IEC 25000 (SQuaRE), considerando solo la división del modelo de calidad (ISO/IEC 25010:2010). Aunque esas listas de propiedades proveen un punto de partida útil para entender y mejorar la calidad, la tónica general es la escasez de consenso en dichas propuestas, con una gran cantidad de esfuerzo dedicado a tareas ya desarrolladas por muchos autores. Entonces, el trabajo de investigación permitirá conocer los modelos de calidad existentes para promover su utilización en el desarrollo de productos software de calidad, que no solo sean un pilar fundamental para el logro de objetivos de las organizaciones, sino contribuyan con las tareas de gestión que se tengan establecidas. Además, desde que surge la teoría de la organización, el papel de la información y de la comunicación, la información empieza a considerarse como una reserva de conocimientos disponibles sin limitaciones en su uso, gracias al aporte de las nuevas tecnologías y estándares que hacen que cada día sea más accesible al usuario. Y contrastándolo con la actualidad, la información empieza a considerarse además, como un recurso económico y esto, porque, una vez procesada y utilizada racionalmente, adquiere ese valor. Extraer el valor de la información es un gran reto que empiezan a adoptar las organizaciones que buscan la competitividad y el liderazgo en su campo, pues para tal tarea es necesario contar con las mejores herramientas que ayuden a gestionar adecuadamente ese valor, herramientas como un producto software basado en un modelo de calidad.

Bajo estas consideraciones, el fin de esta investigación es desarrollar un sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE), tratando no de excluir el

planteamiento de otros modelos definidos, sino de aglutinar a los mismos adecuándolos para la consecución de un modelo fortalecido. Este modelo ayudará a evaluar la calidad del producto durante todas las etapas de desarrollo para asegurar que el producto final cumpla con los requerimientos y fines para los cuales fue creado, y así sea un producto software de calidad. Como resultado, representará un factor importante en los procesos de gestión de información del CIP CD Puno, cuyo núcleo fundamental como lo es la información, no es considerada o es dejada de lado en muchas ocasiones.

En tal sentido, la implementación de un sistema Web basado en la Norma ISO/IEC 5010:2010 (SQuaRE) en el CIP CD Puno, contribuirá a mejorar la gestión de información, permitiendo se brinde un mejor servicio a los agremiados.

1.4. OBJETIVOS DE LA INVESTIGACIÓN

1.4.1. OBJETIVO GENERAL

Desarrollar un Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE) para mejorar la gestión de información en el Colegio de Ingenieros del Perú CD Puno.

1.4.2. OBJETIVOS ESPECÍFICOS

- Realizar el análisis y diseño del Sistema Web Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE) para conocer los factores que determinan una buena gestión de información en el CIP CD Puno.
- Implementar el Sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE).
- Probar y validar el funcionamiento del Sistema Web utilizando el modelo de calidad basado en la Norma ISO/IEC 25010:2010 (SQuaRE).

1.5. HIPÓTESIS

Un Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE) mejora la gestión de información en el Colegio de Ingenieros del Perú CD Puno.

1.6. LIMITACIONES

Para la elaboración del presente trabajo de investigación, se tomaron en cuenta las siguientes limitaciones:

- La investigación se realizó en el Colegio de Ingenieros del Perú Consejo Departamental Puno, es decir, en la ciudad de Puno, provincia de Puno, departamento de Puno. De tal forma que el sistema Web muestra información real de ingenieros colegiados en la sede Puno del Colegio de Ingenieros del Perú.
- El sistema Web implementado, está operativo solo para la sede departamental Puno, no se configuraron accesos para los comités provinciales.
- El costo en dinero que significó trasladarse a otras localidades para la realización de pruebas, implementación y evaluación.
- Para la realización de las pruebas, se utilizó data correspondiente a la gestión 2010 – 2012 del CIP CD Puno, pues no se tuvo acceso a la base de datos de la actual gestión.
- Solamente se realizaron pruebas en dispositivos móviles con soporte 3G en adelante.



CAPÍTULO II. MARCO TEÓRICO

2.1. ANTECEDENTES

Algunas de las investigaciones realizadas de las cuales se tiene referencia por su cercanía con la presente investigación son las siguientes:

- ***Modelo de Calidad para Portales Bancarios:*** Medir la calidad de los portales que están siendo diseñados y utilizados hoy en día es de vital importancia para todas aquellas organizaciones bancarias que pretenden posicionarse en Internet, ya que datos empíricos muestran que este medio es el más utilizado de manera habitual (81.55% de usuarios lo utilizan diaria y/o semanalmente), por encima incluso del cajero automático (77.38%). Con este dato, la calidad del servicio y del portal al que acceden dichos usuarios es fundamental, ya que dicha calidad puede marcar la diferencia en su percepción de la calidad de servicio de la propia entidad bancaria (Córdoba, 2007).
- ***Sistema Web de Seguimiento de Procesos Judiciales para la Gestión de la Información en el estudio de abogados Trujillo y Asociados de la provincia de Azángaro:*** La gestión de información es una descripción de tareas muy completa, que refleja de manera pragmática todo lo que se espera que realice una persona encargada de coordinación de medios para conseguir el objetivo central de la información. El destino final de la misma se sobreentiende que está subordinado a los objetivos globales de la organización (Gómez, 2011).
- ***Modelos de Calidad Web. Clasificación de métricas:*** En el modelo WQM, se propone una estructura de cubo que muestra aquellos aspectos que deben ser tenidos en cuenta en la evaluación de la calidad de un sitio web: las características web, los procesos del ciclo de vida y las características de calidad. Con este modelo, se ha analizado un juego muy completo de métricas objetivas para cubrir cada aspecto de la calidad web, y teniendo el conjunto correcto de

métricas para cada aspecto de la calidad web (que incluye las tres dimensiones del modelo WQM) se podrá usar para construir la expresión de una calidad web total combinando estas métricas. Esta expresión podría ser usada para calcular la calidad de un sitio web dado (Esparza, 2004).

- ***Diseño de Prototipo de un Sistema de Gestión de la Información para el Grupo de Trabajo de Alimentos y Bebidas del Hotel “Brisas Santa Lucía”:*** La gestión de información surge como un nuevo concepto dentro del campo de la ciencia de la información, orientado al manejo de la inteligencia corporativa de una organización, que permite la estructuración interna a las organizaciones y les permite reaccionar ante los cambios de su entorno apoyándose en el uso de la información y de los recursos de información disponibles (Hernández, 2007).
- ***Propuesta de un Modelo de Calidad del software aplicado a la Web:*** La complejidad inherente a las aplicaciones Web se debe a sus características propias, más la integración con aplicaciones ya existentes en ambientes diversos. Mientras que en el desarrollo tradicional el objetivo principal es la construcción de productos software de calidad con el mínimo coste, en el desarrollo Web, el objetivo es la puesta de productos de calidad en el mercado lo más breve posible [...] (Parejas, 2004).
- ***Modelo de Calidad para un Producto de Software: Banca On-Line, basado en el estándar de calidad ISO/IEC 9126:*** El presente modelo de calidad se realizó considerando 6 características básicas del producto de software, desde el punto de vista de externo. Observamos al respecto que cuantas más características estén presentes en la evaluación del producto, más real será la evaluación del mismo. Sin embargo, usualmente existen características que entran en contraposición en la valoración final del producto, por ejemplo; cuanto más

contenido abarque el producto, menor será la usabilidad del mismo. La evaluación se desarrolló desde el punto de vista del cliente final con perfil de usuario con conocimientos básicos de banca y conocimientos básicos en informática. Para obtener una evaluación completa y real del producto, es necesario contar con todos los protagonistas (stakeholders) que influyen en el resultado final. No existe una metodología lo suficientemente clara para asignar pesos a los diferentes atributos del producto. Nuestra asignación de los pesos se basó principalmente en una decisión subjetiva en ciertos casos y experimental en otros de la probabilidad de ocurrencia de los atributos. Se requiere de una evaluación más exacta y práctica para obtener un patrón de medición reutilizable en posteriores evaluaciones (Parejas, 2004).

- ***Construcción de una herramienta para evaluar la calidad de un producto software:*** El mundo globalizado exige cada vez más la aplicación de estándares internacionales que garanticen la calidad de los productos. El uso adecuado de un modelo de calidad estándar permite que los productos de software puedan tener una calidad aceptable para el cliente. La utilización de las métricas dentro de esos modelos de calidad de evaluación de un producto de software juegan un papel determinante (Piedrahita, 2007).
- ***Gestión de la Información en las Organizaciones:*** Se concluye que, la gestión de la información que demandan hoy en día los individuos y las organizaciones, requiere de un reposicionamiento del rol del profesional en información. Es necesario dar la importancia y jerarquía necesaria en la organización, así como, fortalecer los vínculos necesarios con otras áreas del conocimiento para que a través de esa sinergia se enriquezca la capacidad y la gestión de la información que demandan los clientes (Rodríguez, 2002).

2.2. MARCO CONCEPTUAL

2.2.1. SISTEMA WEB

Un sistema o aplicación web es una aplicación informática distribuida, cuya interfaz de usuario es accesible desde un cliente web (un navegador web).

Los sistemas Web son populares debido a lo práctico del navegador Web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software a miles de usuarios potenciales.

Un sistema web tiene las siguientes características:

- La comunicación se realiza mediante HTTP sobre TCP/IP.
- Procesamiento en servidor.
- Acceso a base de datos.
- Arquitectura por capas.
- Distintos tipos de usuarios.

2.2.1.1. Ventajas de un Sistema Web

- *Ahorra tiempo:* Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- *No hay problemas de compatibilidad:* No hace falta crear diferentes clientes en función de cada sistema operativo. Basta tener un navegador actualizado para poder utilizarlas.
- *No ocupan espacio en el disco duro.*
- *Actualizaciones inmediatas:* Como el software lo gestiona el propio desarrollador, cuando se conecta se está usando siempre la última versión que se haya lanzado.

- *Consumo de recursos de hardware bajo:* Dado que toda (o gran parte) de la aplicación no se encuentra en el ordenador, muchas de las tareas que realiza el software no consumen recursos del propio ordenador porque se realizan desde otro ordenador.
- *Multiplataforma:* Se pueden usar desde cualquier sistema operativo porque solo es necesario tener un navegador.
- *Portabilidad:* Es independiente del ordenador donde se utilice (un PC de escritorio, un portátil) porque se accede a través de un navegador Web (solo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones Web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- *Disponibilidad:* Alto grado de disponibilidad porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
- *Colaboración:* Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.

2.2.1.2. Desventajas de un Sistema Web

- Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo. Pero cada vez los navegadores están más preparados para mejorar en este aspecto.

- La disponibilidad depende de un tercero, el proveedor de la conexión a Internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.

Los lenguajes de programación empleados para el desarrollo de aplicaciones Web son diversos, pero destacan, entre otros, los siguientes: PHP, JavaScript, Perl, Ruby, Python, XML o ASP.NET. Aunque este último no es un lenguaje de programación como tal, sino una arquitectura de desarrollo Web en la que se pueden usar por debajo diferentes lenguajes.

2.2.1.3. Arquitectura de un Sistema Web

En la arquitectura de un sistema Web se define como se organizan los distintos módulos que lo componen; se suelen distinguir tres niveles:

- **Interfaz de usuario:** Está compuesto por las páginas HTML que el usuario solicita a un servidor Web y que visualiza en un cliente Web.
- **Nivel de lógica de negocios:** Está compuesto por los módulos que implementan la lógica de la aplicación y que se ejecutan en un servidor de aplicaciones.
- **Nivel de datos:** Está compuesto por los datos gestionados por un sistema de gestor de base de datos, que maneja la aplicación Web.

Los dos primeros y una parte del tercero (el código encargado de las actualizaciones y consultas), están en el servidor mientras que la parte restante del tercer nivel se sitúa en la base de datos (debido al uso de procedimientos almacenados en la base de datos, una parte del segundo nivel también puede encontrarse en la misma). Teniendo en cuenta estas características en la arquitectura de los sistemas Web, se utilizan diversos patrones de diseño, entre los cuales destaca el patrón Modelo-Vista-Controlador (MVC), el mismo que se utilizó en el desarrollo de la investigación.

2.2.1.4. Patrón de Diseño Modelo-Vista-Controlador (MVC)

Este patrón propone la separación en distintos componentes de la interfaz de usuario (vistas), el modelo de negocio y la lógica de control. Una vista es una “fotografía” del modelo (o una parte del mismo) en un determinado momento. Un control recibe un evento disparado por el usuario a través de la interfaz, accede al modelo de manera adecuada a la acción realizada, y presenta en una nueva vista el resultado de dicha acción. Por su parte, el modelo consiste en el conjunto de objetos que modelan los procesos de negocio que se realizan a través del sistema.

El modelo no tiene que acceder ni a la vista ni al controlador. La vista tiene que poder acceder al modelo (obviamente para representarlo) y al controlador (para enviar las entradas que hace el usuario). Y el controlador ha de acceder al modelo (para conocer los datos y eventualmente pedir el cambio de estos) y a la vista para indicar los cambios en los datos.

En una aplicación Web, las vistas serían las páginas HTML que el usuario visualiza en el navegador. A través de estas páginas el usuario interactúa con la aplicación, enviando eventos al servidor a través de peticiones HTTP. En el servidor se encuentra el código de control para estos eventos, que en función del evento concreto actúa sobre el modelo convenientemente. Los resultados de la acción se devuelven al usuario en forma de página HTML mediante la respuesta HTTP.

La clave está en la separación entre vista y modelo. El modelo suele ser más estable a lo largo del tiempo y menos sujeto a variaciones mientras que las vistas puede cambiar con frecuencia, ya sea por cambio del medio de presentación (por ejemplo HTML a WAP o a PDF) o por necesidades de usabilidad de la interfaz o simple renovación de la estética de la aplicación. Con esta clara separación las vistas pueden cambiar sin afectar

al modelo y viceversa. Los controladores son los encargados de hacer de puente entre ambos, determinando el flujo de salida de la aplicación (qué se ve en cada momento).

2.2.1.4.1. Ventajas y Desventajas de MVC

Las principales ventajas de hacer uso del patrón MVC son:

- La separación del Modelo de la Vista, es decir, separar los datos de la representación visual de los mismos.
- Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.
- Permite el escalamiento de la aplicación en caso de ser requerido.

Las desventajas de seguir el planteamiento de MVC son:

- La separación de conceptos en capas agrega complejidad al sistema.
- La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- La curva de aprendizaje del patrón de diseño es más alta que usando otros modelos más sencillos.

Cabe mencionar que la comparación de ventajas y desventajas de MVC puede ser un tema muy subjetivo y se puede prestar como tema de debate, sin embargo se tomó la decisión usando principalmente los puntos mencionados anteriormente ya que en términos generales la balanza se inclina a favor del MVC en vez de en su contra.

Actualmente existen varios frameworks de desarrollo de aplicaciones Web basados en el patrón MVC como son Kohana, Symfony, Zend Framework, etc.

Para el desarrollo de la investigación se utilizó el framework CodeIgniter, este framework (al igual que muchos otros) utiliza la estrategia descrita del MVC.

2.2.1.4.2. Características del Framework CodeIgniter

Algunos de los puntos más interesantes sobre este framework, sobre todo en comparación con otros productos similares, son los siguientes:

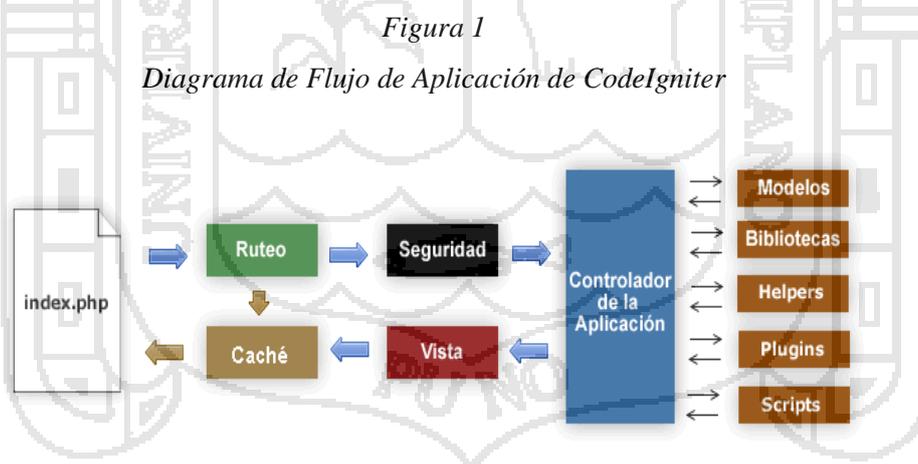
- **Versatilidad:** Quizás la característica principal de CodeIgniter, en comparación con otros frameworks PHP. CodeIgniter es capaz de trabajar en la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde solo se tiene un acceso por FTP para enviar los archivos al servidor y donde no se tiene acceso a su configuración.
- **Compatibilidad:** CodeIgniter es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Por supuesto, funciona correctamente también en PHP 5.
- **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde se debe escribir cosas como el acceso a la base de datos. Durante la configuración no es necesario el acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- **Flexibilidad:** CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos se puede seguirla o no y sus reglas de codificación muchas veces se pueden saltar para trabajar como más a gusto se encuentre. Algunos módulos como el uso de

plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.

- **Ligereza:** El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

Sin duda, lo más destacable de CodeIgniter es su accesibilidad, ya que se puede utilizar en la mayor gama de entornos. Esta es la principal razón de su elección como entorno de trabajo en el desarrollo de la investigación.

Además, es importante señalar que en CodeIgniter existe un procedimiento para atender una solicitud de página del cliente. Este proceso se realiza internamente por el propio CodeIgniter y de manera transparente para su comprensión. Durante el proceso participan varios módulos como el enrutamiento de la solicitud, la caché interna, etc., todo ello puede ser observado en la siguiente figura:



Fuente: *CodeIgniter - Guía del Usuario en Español v. 2.0.2*

En resumen, para que se pueda entender el flujo de aplicación que implementa CodeIgniter, se puede seguir los siguientes puntos:

1. Toda solicitud de una página a partir de CodeIgniter comienza en un `index.php` que se encuentra en la raíz del framework.

2. Luego se realiza un filtrado de la URL para saber cuál es el elemento que tiene que procesar esta página.
3. Si la página se había generado antes y está en la caché de CodeIgniter, se devuelve el archivo de la caché ya generado, con lo que se ahorra procesamientos repetidos. La caché se puede configurar y si se desea, incluso deshabilitar.
4. Antes de continuar con el proceso se realiza un tratamiento de seguridad sobre la entrada que se tenga, tanto de la información que haya en la URL como de la información que haya en un posible POST, si se ha configurado de esa manera.
5. El controlador adecuado realiza el procesamiento de la solicitud. CodeIgniter decide el controlador que debe procesar la solicitud en función de la URL solicitada.
6. El controlador se comunica con una serie de módulos, los que necesite, para producir la página.
7. A través de las vistas adecuadas, el controlador genera la página, tal cual se tiene que enviar al navegador.
8. Si la página no estaba en la caché, se introduce, para que las futuras solicitudes de ésta página sean más rápidas.

Algunos de estos módulos, como la caché o el enrutamiento, funcionan de manera transparente. Algunos otros, como los controladores, modelos y vistas, se tienen que programar por nuestra cuenta y localizar cada una de las partes del programa que, al estar separadas, ayudan a organizar también el código. También se tienen diversas librerías, ayudantes (Helpers) y plugins ya escritos en CodeIgniter con numerosas clases y funciones muy útiles para el desarrollo de aplicaciones web.

Uno de los puntos más básicos de CodeIgniter, pero no por ello menos útiles, es el módulo de enrutamiento (Routing) que permite que cualquier URL que se solicite al servidor se ejecute en el controlador adecuado. La URL se analiza y los datos se

procesan y aseguran antes de enviarlos al controlador adecuado, en el que simplemente se tiene que codificar sus diversos métodos.

2.2.2. ORGANIZACIÓN DE LA ISO E IEC

La ISO (International Standards Organization) es una organización de estandarización en cuyo funcionamiento intervienen organismos de todo el mundo interesados en regular y armonizar diversas áreas de la industria. Por su propia naturaleza, la ISO emite normas conocidas como “de facto”, es decir, normativas cuya adopción no es obligatoria legalmente. Sin embargo, los beneficios de la estandarización hacen que los documentos emitidos por la ISO sean adoptados rápidamente o al menos tenidos en cuenta a la hora de diseñar, implantar o mejorar un producto o servicio.

La ISO es una organización no gubernamental que actúa como puente entre la empresa pública y la privada y que según sus propias palabras “permite ofrecer al público productos con características atractivas o interesantes como el respeto al medio ambiente, eficiencia o productos con piezas intercambiables entre distintos fabricantes”. La IEC (International Electrotechnical Commission) es una organización que también trabaja a nivel mundial en el desarrollo de estándares sobre electricidad y electrónica. Según sus propios estatutos, la IEC “tiene como objetivo promover la normalización en todas las cuestiones relacionadas con la electrotécnica”.

Tanto en la ISO como en la IEC hay diversos comités técnicos (Technical Committees) en los que intervienen los organismos de cada nación y que participan en el desarrollo de estándares en un determinado ámbito. En ocasiones, la ISO y la IEC colaboran en los llamados Comités Técnicos Conjuntos (Joint Technical Committees). Uno de estos comités técnicos es el llamado ISO/IEC JTC 1, especializado en las tecnologías de la información. A menudo, un JTC puede estar dividido en varios subcomités (Sub

Committees) que son los que a menudo elaboran el grueso de un estándar y que dan nombre a los estándares. En este JTC 1 de ISO/IEC hay un SC con el número 7, responsable del área de Ingeniería del Software y de Sistemas.

Esta investigación está basada en las normas ISO/IEC JTC 1/SC 7 de la serie 25000 denominadas “Software Product Quality Requirements and Evaluation” y conocidas más comúnmente con el nombre “SQuaRE”.

2.2.3. CALIDAD DE SOFTWARE

La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. (IEEE, Std. 610-1990).

En la industria del software, ha sido común encontrar los fracasos continuos de las diferentes metodologías para dominar la complejidad del software, lo que implicaba el retraso de los proyectos de software, las desviaciones por exceso de los presupuestos fijados y la existencia de deficiencias respecto a los requisitos del cliente (Minguet, 2003).

Las empresas desarrolladoras de software liberaban productos al mercado con un alto porcentaje de defectos en el producto, y gran parte de ellas gastaban su tiempo y dinero en correcciones y ajustes a los productos, dedicando los esfuerzos de hoy a arreglar lo que se hizo mal ayer. Muchos de los proyectos de software eran cancelados, consecuencia directa de la desorganización y la falta de planificación.

El coste de obtener y mantener el software en los 80's fue el doble de lo que costó su desarrollo. Durante los 90's el coste de licencias y mantenimiento se incrementó en un 30% más que en los 80's. La mitad de los proyectos de software se pasaban del cronograma definido. Las tres cuartas partes de todo el software liberado para uso por el cliente tienen fallos.

Alrededor de 1968, en la segunda era de la evolución de los sistemas informáticos, se hizo presente la crisis del software. Etapa en que las actividades de mantenimiento del software (corrección de fallos, modificación por cambios de requerimientos de usuarios, y adaptación a nuevos dispositivos) y el esfuerzo empleado en dicho mantenimiento comenzaron a consumir una impresionante cantidad de recursos.

En este tipo de organización, los procesos software normalmente son improvisados, y si se han especificado, no se siguen rigurosamente, además se sacrifican funcionalidad y calidad del producto para satisfacer el plan. De esta forma, y sin bases objetivas para juzgar la calidad del producto, un gran porcentaje de los proyectos no alcanzan los objetivos, y el resto fracasan por completo o no llegan a entregarse.

Frente a todos estos problemas, las empresas productoras de software han tenido que establecer unos adecuados sistemas de calidad que proporcionen a los clientes la suficiente confianza para adquirir o contratar el desarrollo de un producto, y que éste pueda ser utilizado por mucho tiempo satisfaciendo las necesidades y/o requerimientos del cliente.

El desarrollo de software es más que solo el empleo de un lenguaje de programación, ya que durante todo el proceso de producción se adhiere la calidad. Muchas empresas productoras de software utilizan metodologías de calidad en los procesos de desarrollo de software, adquiriendo un compromiso adoptando estándares internacionales de calidad que proporcionen la confianza requerida por los clientes, y esto se verá reflejado en su competitividad con otras empresas a nivel internacional.

Los conceptos de la gestión de calidad se han adecuado a la industria del software, permitiendo la creación de estándares y modelos de calidad internacionales que pretenden dar apoyo a las empresas productoras de software en la incorporación de modelos de calidad para sus procesos de desarrollo.

Los productores de software han visto la necesidad de contar con herramientas y metodologías que apoyen la mejora de su proceso de desarrollo, ante la cada vez mayor exigencia de sus clientes. Esto unido a la falta de un consenso generalizado sobre calidad, ha llevado a la creación y evolución de modelos y estándares tales como CMM, SPICE, BOOTSTRAP, ISO/IEC 9126, y el modelo ISO 9000, entre otros (Minguet, 2003).

Puesto que la industria informática crece aceleradamente, el software se convierte en un producto industrial clave y de creciente valor estratégico para las empresas. Éstas tendrán que desarrollar las estrategias que les permitan un posicionamiento y un reconocimiento con productos de software competitivos, lo que requerirá entre otras cosas, de la implementación de modelos de aseguramiento de la calidad, dejando de lado la informalidad que caracterizaba a la industria tradicional de software.

Las propuestas de acción para el fortalecimiento de la industria del software han permitido que las empresas productoras de software identifiquen, como algo imprescindible para tener éxito, alcanzar los niveles de competitividad de las empresas extranjeras, para lo cual deben incorporar estándares internacionales de clase mundial. Esta búsqueda de un reconocimiento internacional de calidad que se ha iniciado en algunas empresas del sector, ofrece en los mercados mayores posibilidades de éxito y abre las puertas para que otras empresas se animen en estos procesos y se desate un alto interés y compromiso hacia la incorporación de dichos estándares.

En definitiva, y en un mundo cada vez más globalizado, donde las tecnologías de la información cobran un papel protagonista y la información pasa a ser un elemento muy valioso, la calidad del software se convierte en un eje fundamental para las organizaciones y su evaluación se hace pertinente para que se cumplan los propósitos que se quieren lograr con la ayuda de esos productos software.

La evaluación de la calidad del software se hace a través de modelos y estándares de calidad del software, los cuales reúnen todas las actividades y funciones de forma tal que cada una se planee, se controle y se ejecute de un modo formal y sistemático (Escalone, 2006).

Cuando se aplican modelos de calidad, finalmente lo que se logra es que mejoren los procesos de software, así como la calidad del software que se desarrolla.

Además, la gestión de calidad supone para la organización de hoy en día, un impacto estratégico y la oportunidad de tener una ventaja competitiva frente a otras organizaciones. Afecta a la sociedad en general, directivos, empleados y clientes, y se ha convertido en una filosofía estratégica, que como organización, lleva a cabo la planificación, fijación de objetivos, coordinación, formación y adaptación de toda la organización.

Existen varias razones que justifican el por qué la calidad es crítica para la supervivencia de las empresas, estas son:

- La calidad es un factor competitivo.
- La calidad es esencial para el comercio internacional.
- La calidad reduce las pérdidas producidas por la no calidad.
- La calidad mantiene a los clientes e incrementa los beneficios.
- La calidad es el sello distintivo de los negocios a nivel mundial.

2.2.4. MODELOS DE CALIDAD

Los modelos de calidad de software sirven como una guía para el desarrollo de un proyecto, pues permiten saber qué es lo que se debe hacer para llevar a cabo el desarrollo, más no dicen cómo hacerlo. Esto es porque dependiendo de la metodología establecida, se obtiene un resultado más apropiado para el proceso que se use en el desarrollo, lo cual va enfocado a los objetivos que se quieran perseguir en la

organización. Al implementar un modelo de calidad se logra tener una mejor productividad y calidad en el producto final, satisfaciendo las necesidades y deseos de los clientes.

Las empresas pueden implementar la calidad a nivel del proceso y en consecuencia obtener calidad a nivel producto.

“La calidad a nivel del proceso puede ser evaluada de manera genérica o específica, según el modelo o estándar seleccionado. Todo modelo o estándar a este nivel tiene un ámbito de aplicación específico y tiene como finalidad el mejoramiento continuo, luego de realizada la implantación del mismo” (Scalone, 2006). Entre los modelos y/o estándares de calidad de software a nivel proceso se pueden mencionar las normas: ISO 9001:2000, CMMi, ISO 90003:2004, ISO 20000, y otros.

La calidad a nivel producto plantea distintos modelos y estándares que poseen un conjunto de características, las cuales tienen asociadas sub-características. Todo equipo de desarrollo deberá evaluar la calidad de software durante las diferentes etapas de desarrollo del software y ambientes de trabajo respectivos (desarrollo, prueba y producción). Esto evita futuros problemas y contribuye a una posible disminución en los tiempos y costos. Entre los modelos y/o estándares de calidad de software a nivel producto se pueden mencionar: modelo de Boehm, modelo de Gilb, modelo de Dromey, ISO/IEC 9126, modelo de McCall, WebQEM, ISO/IEC 25000, Portal Quality Model (PQM) y otros. Los modelos y estándares a nivel producto surgen o se actualizan de acuerdo con la evolución tecnológica ocurrida.

Para la realización de esta investigación es de gran importancia, conocer los diferentes modelos de calidad que se utilizan para analizar un producto de software y que son los que permiten conocer los criterios de evaluación del producto. En este punto se

considera el modelo de calidad de la norma ISO/IEC 25000 (SQuaRE) ya que cuenta con las características de evaluación.

En cada etapa del desarrollo del sistema Web, es necesario tener en cuenta cada una de las características que considera el modelo de calidad de la norma ISO/IEC 25000 (SQuaRE), tarea que se torna compleja, ya que estos modelos suelen descomponer la calidad del producto software jerárquicamente en una serie de características y sub-características que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad. La mayor parte de ellas están basadas en la norma ISO/IEC 9126. Esta norma define un conjunto de características de calidad que son después refinadas en sub-características, las cuales están descompuestas en atributos. Los valores de estos atributos se calculan mediante la utilización de métricas o listas de chequeo (Muñoz, 2005).

Existen diversos modelos de calidad que permiten evaluar un producto de software, como se mencionó anteriormente. A continuación se ampliará la información sobre algunos de ellos.

A. Modelo de McCall. Este modelo establece tres áreas principales que intervienen en la calidad del software:

- Revisión de la calidad del producto de software. Tiene como objetivo realizar revisiones durante el proceso de desarrollo para detectar los errores que afecten a la operación del producto.
- Transición del producto. Requiere de la definición de estándares y procedimientos que sirvan como base para el desarrollo del software.
- Calidad en la operación del producto. Requiere que el software pueda ser entendido fácilmente, que opere eficientemente y que los resultados obtenidos sean los requeridos inicialmente por el usuario.

Con relación a la operación del producto, se sugieren las siguientes características, con sus correspondientes interrogantes:

- Corrección – ¿El producto hace lo que el usuario quiere?
- Confiabilidad – ¿El producto es seguro?
- Integridad – Lo que hace el producto, ¿lo hace en forma fiable a lo largo del tiempo?
- Usabilidad – ¿Se puede ejecutar?
- Eficiencia – ¿El comportamiento del producto será adecuado en el ambiente con el que cuenta el usuario para su ejecución?

Dentro de la revisión del producto se estudian las siguientes características, para las cuales se plantean diferentes interrogantes:

- Mantenibilidad - ¿El producto se puede arreglar?
- Flexibilidad - ¿El producto se puede cambiar?
- Facilidad de Prueba - ¿El producto se puede probar?

A la transición del producto se asocian las siguientes características:

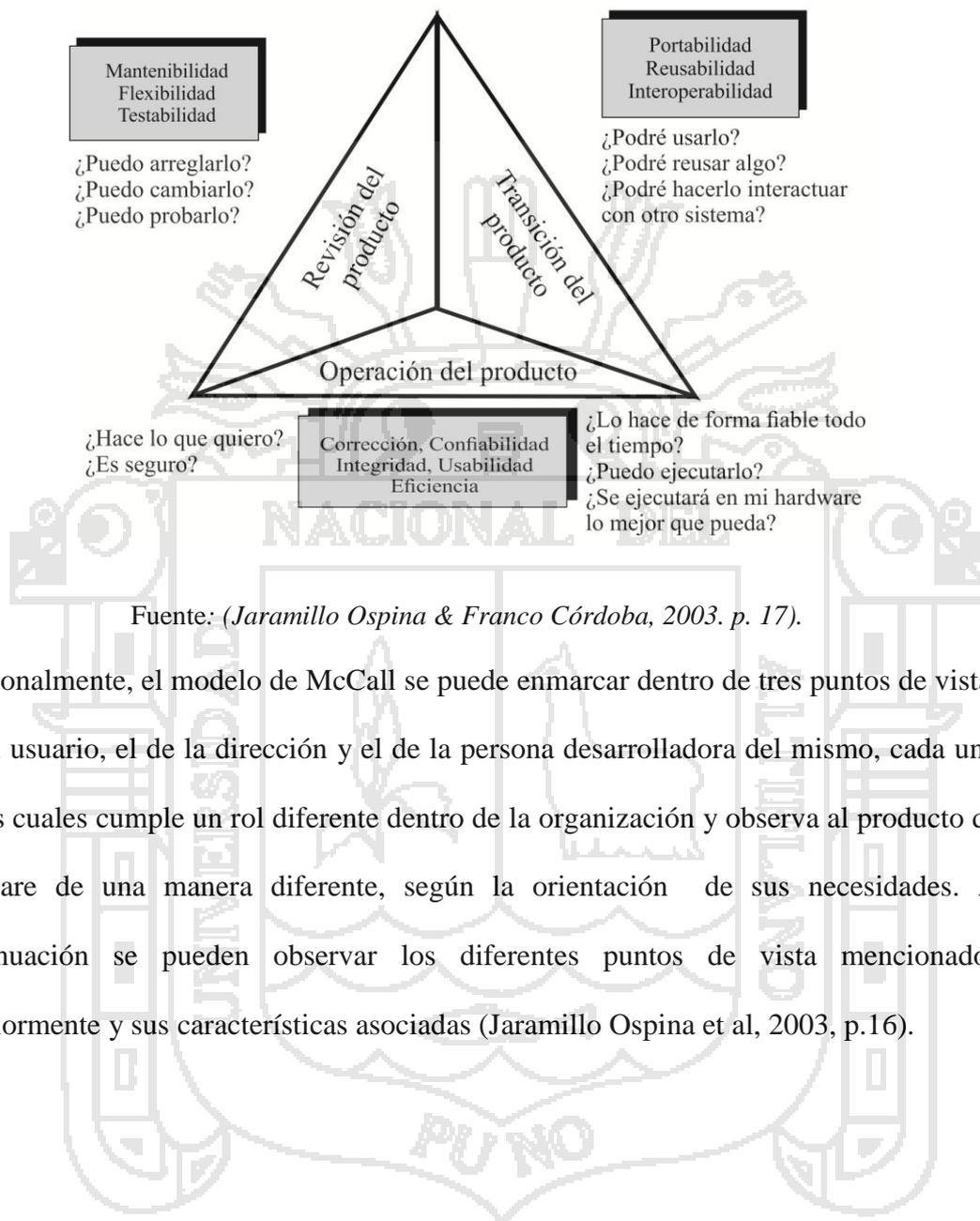
- Portabilidad – ¿El producto puede ser usado en diferentes ambientes?
- Reusabilidad – ¿Se puede reutilizar alguna parte del producto para satisfacer una necesidad?
- Interoperabilidad – ¿El producto puede interactuar con otros sistemas?

En la figura 2 puede observarse, en resumen, la propuesta del modelo McCall.

Figura 2

Modelo de calidad de McCall (1977).

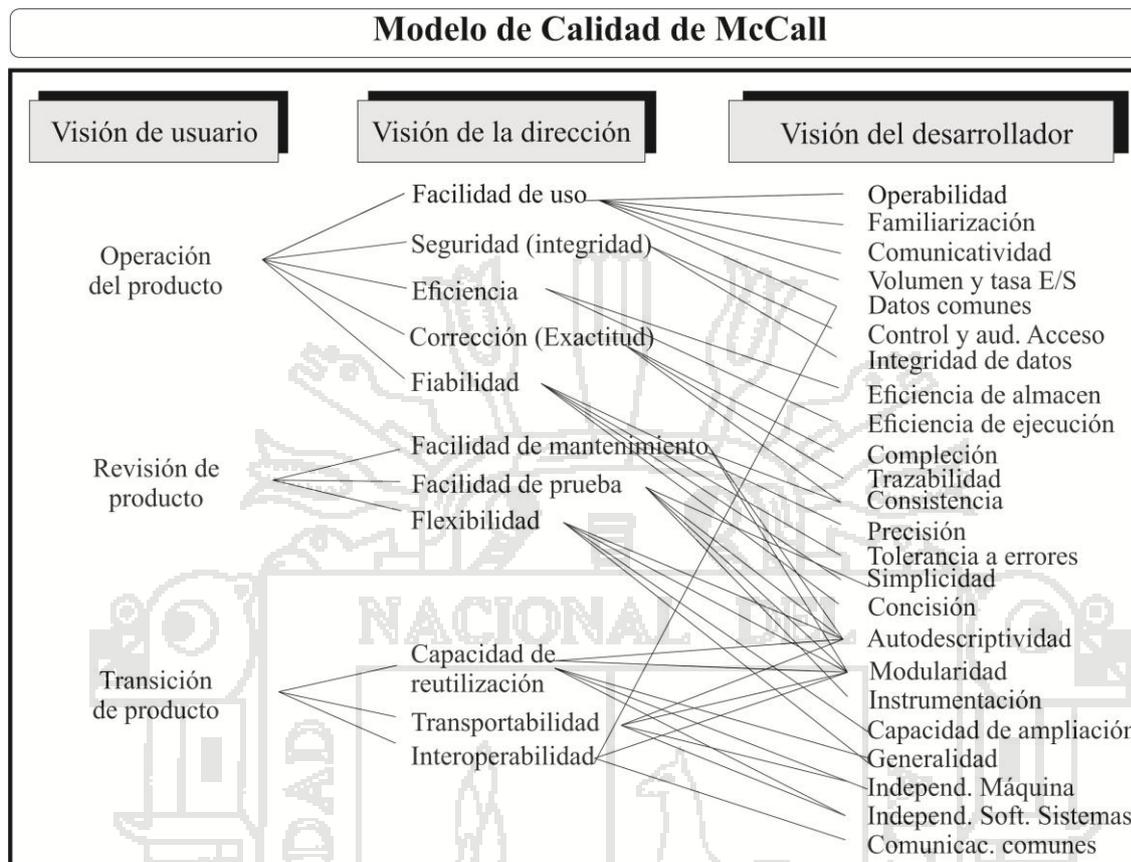
Modelo de Calidad de McCall



Fuente: (Jaramillo Ospina & Franco Córdoba, 2003. p. 17).

Adicionalmente, el modelo de McCall se puede enmarcar dentro de tres puntos de vista: el del usuario, el de la dirección y el de la persona desarrolladora del mismo, cada uno de los cuales cumple un rol diferente dentro de la organización y observa al producto de software de una manera diferente, según la orientación de sus necesidades. A continuación se pueden observar los diferentes puntos de vista mencionados anteriormente y sus características asociadas (Jaramillo Ospina et al, 2003, p.16).

Figura 3
Diferentes visiones del modelo de calidad de McCall.



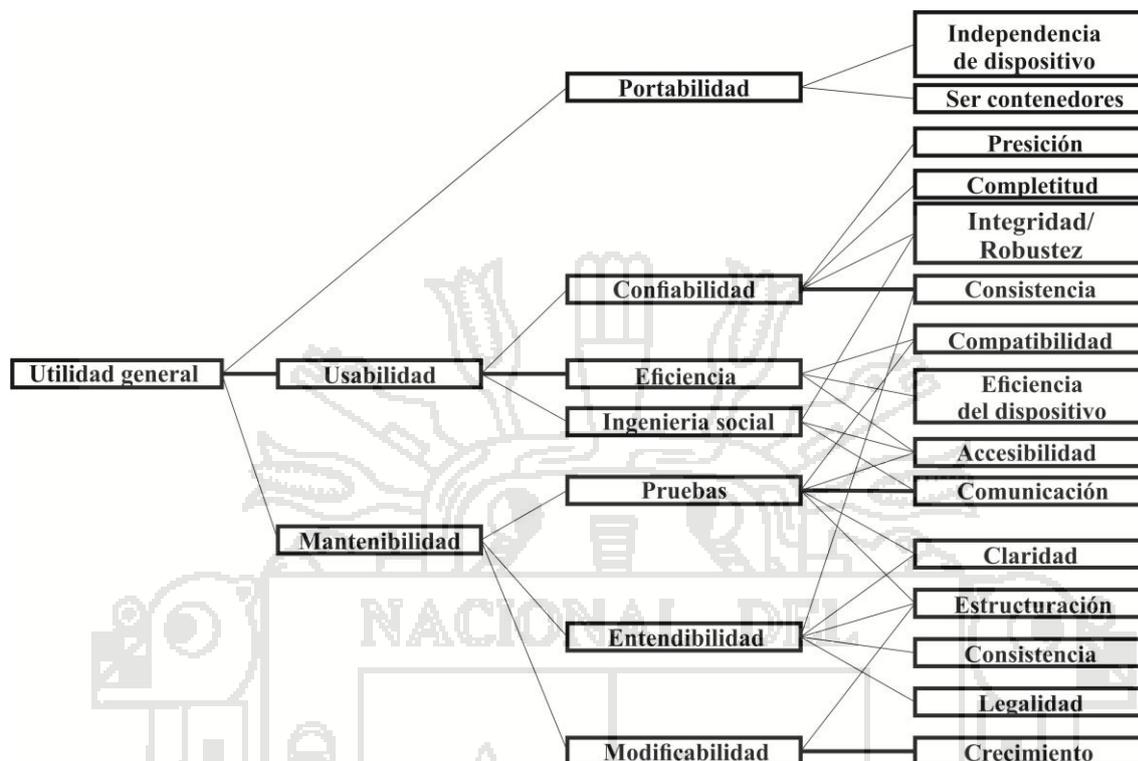
Fuente: (Jaramillo Ospina et al, 2003, p. 18).

B. Modelo de Boehm (1978). Este modelo es de naturaleza jerárquica y los criterios de calidad se presentan en tres grandes divisiones. La primera división es hecha acorde con los servicios que el sistema ofrece (portabilidad). La segunda se hace de acuerdo con la operación del producto (usabilidad) y la tercera división se hace de acuerdo con la mantenibilidad del producto de software (Dávila Nicanor & Mejía Álvarez, 2003).

En la figura 4 se presenta el modelo de Boehm, en donde se puede observar cada una de las divisiones mencionadas.

Figura 4

Modelo de Boehm para clasificar los criterios de calidad.



Fuente: (Dávila Nicanor & Mejía Álvarez, 2003).

C. Modelo de Gilb (1988). El modelo de Gilb presenta como aspecto fundamental la definición de los atributos de calidad que realmente interesan al usuario y el nivel de calidad que debe tener cada uno de ellos para satisfacerlo, ya que no tiene sentido exigir calidad en un producto, si no se cuenta con esta base.

Mediante el método de Gilb es posible especificar los atributos de calidad de software en forma cuantitativa, incluyendo tanto tiempos de respuesta como conceptos conocidos de usabilidad y portabilidad, entre otros.

Gilb propone características como la corrección, la integridad, la facilidad de mantenimiento y la facilidad de uso, como base para proporcionar indicadores útiles para los equipos de trabajo y sugiere las definiciones, puntos de vista y medidas para cada uno de las siguientes características:

- **Corrección:** Grado en el que el software lleva a cabo su función requerida. Si un programa no opera correctamente, no dará valor agregado a sus usuarios.
 - **Facilidad de mantenimiento:** Posibilidad de corregir un programa si se encuentra un error, adaptarlo si cambia su entorno, o mejorarlo si el cliente desea un cambio.
 - **Integridad:** Habilidad de un sistema para resistir ataques, tanto accidentales como intencionados, contra su seguridad, a nivel de cualquiera de los tres principales componentes del software: programas, datos y documentos. Para medir la integridad, Gilb sugiere la utilización de otros dos atributos como base: amenaza y seguridad.
 - **Amenaza:** Es la probabilidad (que se puede estimar o deducir de la evidencia empírica) de que un ataque de cualquier tipo ocurra en un tiempo determinado.
 - **Seguridad:** Es la probabilidad de que se pueda repeler un determinado ataque.
 - **Facilidad de uso:** Es un intento por cuantificar lo amigable que puede ser el producto con el usuario.
- D. Modelo propuesto Bertoa y Vallecillo (2002).** Para componentes software en el que los autores adaptan la norma ISO/IEC 9126 a los componentes COTS (Commercial off-the-shelf).
- E. Modelo de Simão y Belchior (2003).** En el que los autores han ampliado las sub-características y atributos propuestos por la norma ISO/IEC 9126, llegando a identificar 124 atributos de calidad para los componentes software.
- F. Modelo de Ortega (2000).** Se basa en el modelo de calidad sistémica de Callaos y Callaos.

- G. Modelo de calidad QUINT2 (Niessink, 2002).** Presenta una ampliación de la norma ISO/IEC 9126, pensada para valorar la calidad de arquitecturas software.
- H. Modelo de calidad propuesto por Franch and Carvalho (2003).** Presenta una adaptación de la ISO/IEC 9126 para servidores de correo electrónico.
- I. Modelo Botella (2003).** Proponen un modelo para la selección de ERP y también escogen como marco de trabajo el estándar de calidad ISO/IEC 9126-1.
- J. Modelo Cai (2000).** Proponen un modelo de calidad para componentes y sistemas basados en componentes.
- K. Modelo Fernández and Rossi (2000).** Definen un modelo de calidad para software distribuido.
- L. Modelo en Zo and Ramamurthy (2002).** Los autores presentan un modelo para valorar y seleccionar los sitios Web de comercio electrónico en un entorno B2C (Business-to-consumer).
- M. Modelo en Webb and Webb (2002).** Presentan los factores de calidad del sitio Web que son importantes para los consumidores.
- N. Modelo Parasuraman (1998).** Se describe el modelo SERVQUAL el cual contiene cinco dimensiones y 22 ítems para medir los diferentes elementos de la calidad de un servicio en general. La idea de este modelo es que puede ser adaptado a diferentes entornos en función de los servicios ofrecidos por cada uno de ellos, adaptando las dimensiones descritas en el modelo original.
- O. Modelo WQM.** Pretende ser un modelo global de calidad de la web. Está caracterizado por tres elementos:
- La característica de calidad (basada en Quint2 y en la ISO/IEC 9126)
 - El proceso del ciclo de vida (basado en la ISO/IEC 12207)

- Características (contenido, presentación y navegación). Modelo de Calidad para Portales (PQM). Utilizan Fase 1 y 2 de GQM y el modelo SERVQUAL (Muñoz, p.4)

P. Modelo GQM (Goal-Question-Metric): enfoque de medición para evaluar la calidad del software basado en la identificación de objetivos a lograr. A continuación se presenta la estructura del modelo:

- Nivel Conceptual (Goal-Meta). Se define un objetivo (meta) para un objeto (ente), con respecto a determinado “modelo de calidad”, para un punto de vista, relativo a un contexto en particular.
- Nivel Operativo (Question-Pregunta). Se refina un conjunto de preguntas a partir de una meta, identificando el objeto de medición con respecto a características de calidad seleccionadas para un punto de vista.
- Nivel Cuantitativo (Metric-Métrica). Se asocia un conjunto de métricas para cada pregunta, de modo de responder a cada una de ellas de un modo cuantitativo (Olsina, 2007).

Más adelante se explicará el modelo de calidad en el que se basó esta investigación, es decir la norma ISO/IEC 25010:2010 (SQuaRE). Además se describirán las características y sus respectivas sub características, las cuales se tendrán en consideración a la hora de evaluar el sistema Web.

2.2.5. NORMATIVAS DE CALIDAD

La especificación de requisitos de calidad y la evaluación de productos no son algo nuevo en el campo de la normalización. De hecho, las normas ISO/IEC 9126 (ISO/IEC 9126-1, 2001) e ISO/IEC 14598 (ISO/IEC 14598-1, 1999), respectivamente, han regulado hasta ahora los procedimientos necesarios para estos procesos.

Dentro de la serie ISO/IEC 9126 se incluyen los siguientes estándares:

- ISO/IEC 9126 - 1: Presenta las propiedades que deberían tener las medidas a utilizar para comparaciones.
- ISO/IEC 9126 - 2: Presenta las medidas de calidad externas.
- ISO/IEC 9126 – 3: Referido a las medidas de calidad internas.
- ISO/IEC 9126 - 4: Explica las medidas de calidad en el uso.

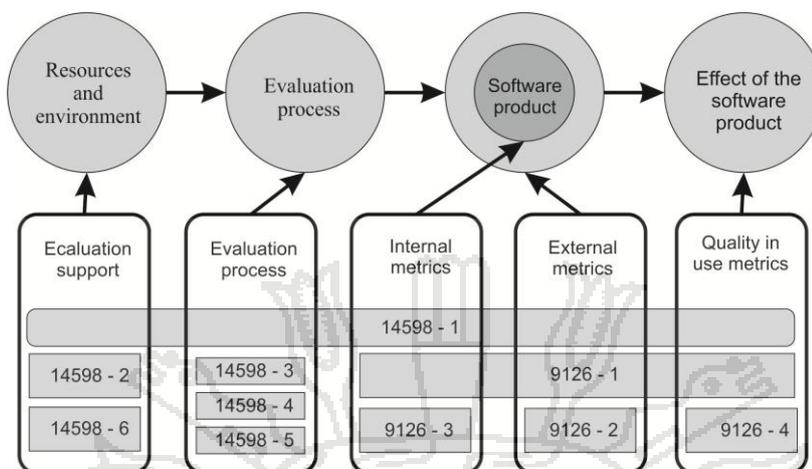
Dentro de la serie ISO/IEC 14598 se pueden destacar las siguientes normas:

- IO/IEC 14598 – 1: Enumera las medidas aplicables en la evaluación de software y los requisitos que aquellas deberían cumplir.
- IO/IEC 14598 – 2: Explica cómo realizar la planificación y gestión de la evaluación.
- IO/IEC 14598 – 3: Explica cómo elegir atributos que representen los requisitos de calidad.
- IO/IEC 14598 – 4: Regula el proceso y entre otras muchas cosas, proporciona ejemplos de métodos de evaluación, como listas, historiales o tablas y recomendaciones en el uso de los mismos.
- IO/IEC 14598 – 5: Especifica cómo adaptar el proceso general de evaluación a un entorno más específico.
- IO/IEC 14598 – 6: Trata sobre la documentación del proceso.

Como se aprecia en la figura 5 y tal como se ha comentado, el espectro de ambos grupos de normas se solapa en bastantes ámbitos, lo que ha hecho necesaria la redefinición de un estándar unificado.

Figura 5

Relación entre los estándares internacionales ISO/IEC 9126 e ISO/IEC 14598.

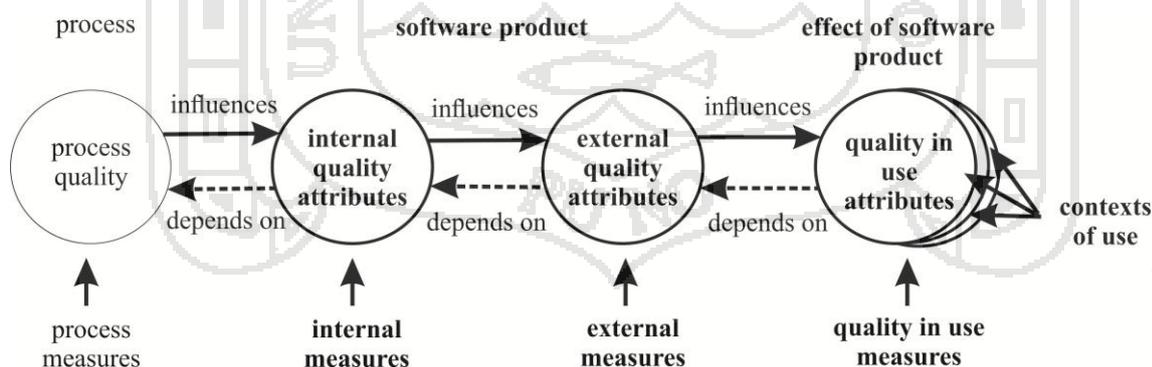


Fuente: ISO/IEC 25000 (SQuaRE), 2005.

Además se aboga por un modelo unificado para el ciclo de vida relacionado con la calidad. Este ciclo de vida, que puede verse en la figura 6, parte de un proceso que influye en un conjunto de características internas que a su vez, modifican ciertas características externas que en última instancia modifican los atributos relacionados con la calidad en uso.

Figura 6

Ciclo de vida de la calidad en ISO.



Fuente: ISO/IEC 25000 (SQuaRE), 2005.

2.2.6. NORMA INTERNACIONAL ISO/IEC 25000 (SQuaRE)

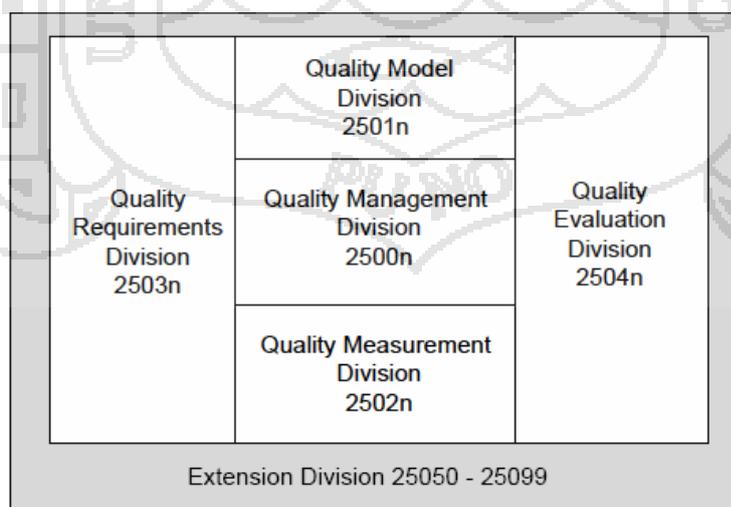
Aspectos como el hecho de tener dos normas complementarias: ISO/IEC9126 (Calidad del producto software) e ISO/IEC 14598 (Evaluación del producto software), han propiciado el impulso necesario para la elaboración de la norma ISO/IEC 25000 (2010) conocida como SQuaRE (Software Quality Requirements and Evaluation). La meta perseguida en la creación de esta norma es dar un paso hacia un conjunto de estándares organizados de manera más lógica, enriquecidos con nuevas aportaciones y unificados con respecto a las normas anteriores para ser capaces de cubrir los dos principales procesos: especificación de requisitos de calidad del software y evaluación de la calidad del software, soportados por un proceso de medición. SQuaRE se centra exclusivamente en el producto software estableciendo criterios para su especificación, su medición y su evaluación. Es decir, básicamente se trata de una unificación y revisión de los estándares ISO/IEC 9126 e ISO/IEC 14598.

2.2.6.1. Divisiones de la Norma ISO/IEC 25000 (SQuaRE)

La norma está compuesta de cinco divisiones bien diferenciadas:

Figura 7

Organización de la serie de estándares de SQuaRE.



Fuente: ISO/IEC FDIS 25010 (SQuaRe), 2010.

- **División de gestión de calidad (ISO/IEC 2500n):** En esta división se definen estándares para los modelos, términos y definiciones comunes utilizados por el resto de estándares SQuaRE, así como sugerencias para la aplicación del estándar. Dentro de esta división están los siguientes estándares:
 - *ISO/IEC 25000 - Guía de SQuaRE:* Describe la arquitectura de SQuaRE, la terminología y los modelos de referencia.
 - *ISO/IEC 25001 - Planificación y gestión:* Describe el soporte necesario para la gestión en la evaluación y especificación de requisitos de calidad.
- **División del modelo de calidad (ISO/IEC 2501n):** Las normas de esta división presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Esta división se encuentra formada por:
 - *ISO/IEC 25010 – Modelo de calidad:* Describe el modelo de calidad para el producto software y para la calidad en uso. Esta Norma presenta las características y subcaracterísticas de calidad frente a las cuales evaluar el producto software.
 - *ISO/IEC 25012 – Modelo de calidad de datos:* Define un modelo general para la calidad de los datos, aplicable a aquellos datos que se encuentran almacenados de manera estructurada y forman parte de un Sistema de Información.
- **División de mediciones de calidad (ISO/IEC 2502n):** Los estándares de esta división incluyen un modelo de referencia de la medida de calidad del producto software (Figura 8), definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.

Contiene los siguientes estándares:

- *ISO/IEC 25020 - Modelo de referencia para la medida con guía*: Presenta una introducción y un modelo de referencia común para las primitivas, medidas de calidad interna, externa y en el uso. En la guía se proporciona ayuda para seleccionar un conjunto adecuado de medidas de los estándares internacionales ISO/IEC 9126-1 a 9126-4 e ISO/IEC 14598-1.
- *ISO/IEC 25021 - Primitivas*: Define y especifica medidas base y derivadas a utilizar durante el desarrollo del software. Este conjunto de primitivas se utilizará como entrada en el proceso de medida de la calidad interna, externa y en el uso.
- *ISO/IEC 25022 - Medidas de calidad interna*: Define las medidas de calidad interna en función de características y subcaracterísticas.
- *ISO/IEC 25023 - Medidas de calidad externa*: Define las medidas de calidad externa en función de características y subcaracterísticas.
- *ISO/IEC 25024 - Medidas de calidad en uso*: Describe un conjunto de medidas para la calidad en el uso y una guía para la utilización de estas medidas.
- **División de requisitos de calidad (ISO/IEC 2503n)**: Los estándares de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado o como entrada para un proceso de evaluación.

El único estándar de esta división es:

- *ISO/IEC 25030 - Requisitos de calidad:* Provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.
- **División de evaluación de la calidad (ISO/IEC 2504n):** Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto software, tanto si la llevan a cabo evaluadores, como cliente o desarrolladores. Basándose en la ISO/IEC 14598.
Esta división se encuentra formada por:
 - *ISO/IEC 25040 – Modelo de referencia para la evaluación:* Propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas.
 - *ISO/IEC 25041 – Guía de evaluación para desarrolladores, compradores y evaluadores:* Describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes.
 - *ISO/IEC 25042 – Módulos de evaluación:* Define lo que la norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos.
 - *ISO/IEC 25045 – Módulo de evaluación para recuperabilidad:* Define un módulo para la evaluación de la subcaracterística recuperabilidad.
- **División de extensión de SQuaRE (ISO/IEC 25050 a ISO/IEC 25099):** Se reserva para normas o informes técnicos que aborden dominios de aplicación

específicos y que puedan ser utilizados para complementar las cinco divisiones anteriores.

2.2.6.2. Modelo de la Medida de Calidad del Software

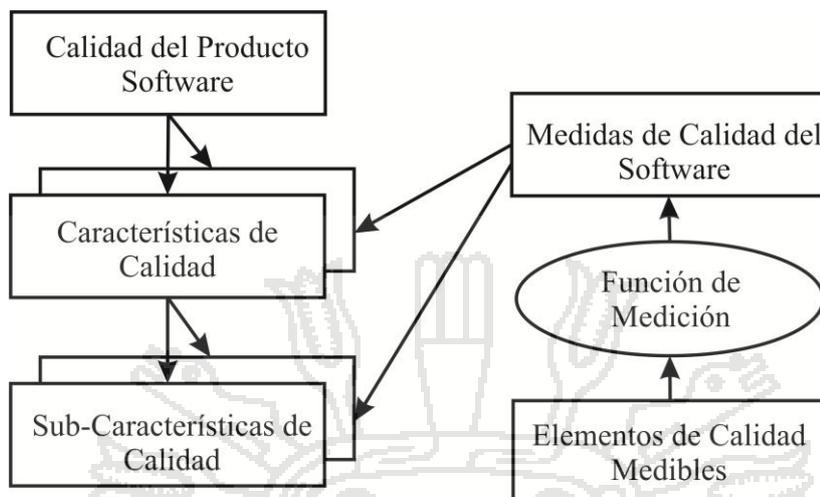
Las propiedades inherentes al software, que pueden ser distinguidas cuantitativamente o cualitativamente, son llamadas atributos. Los atributos de calidad son inherentes a las propiedades del software que contribuyen a la calidad. Los atributos de calidad son categorizados entre una o más subcaracterísticas.

Los atributos de calidad son medidos aplicando un método de la medida. Un método de la medida es una secuencia lógica de operaciones usadas para cuantificar un atributo con respecto a una escala específica. El resultado de aplicar un método de la medida es llamado una medida base. Las características y subcaracterísticas de calidad pueden ser cuantificadas para aplicar funciones de la medida. Una función de la medida es un algoritmo usado para combinar elementos de la medida de calidad. El resultado de aplicar una función de la medida es llamada una medida de la calidad del software. De esta manera las medidas de la calidad del software llegan a ser cuantificaciones de las características de la calidad.

Más de una medida de la calidad del software puede ser usada para medir una característica o subcaracterística de calidad.

Figura 8

Modelo de medición de la calidad del producto software según SquaRE.



Fuente: ISO/IEC 25000 (SquaRE), 2005.

2.2.6.3. Modelo de Calidad ISO/IEC 25010 (SQuaRE)

El modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto software determinado.

La calidad del producto software se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios aportando de esta manera un valor.

El modelo define tres diferentes vistas de calidad:

- **Calidad interna del software:** La calidad interna del software proporciona una vista de “caja blanca” y trata las características del producto software que están típicamente disponibles durante el desarrollo. La calidad interna del software está principalmente relacionada con las propiedades estáticas del software.
- **Calidad externa del software:** La calidad externa del software proporciona una vista de “caja negra” del software y trata las características relacionadas con la ejecución del software en el hardware y el sistema operativo.

- **Calidad en uso:** Es una medida de la calidad del sistema en su ambiente operacional para usuarios, para las tareas específicas que realizan.

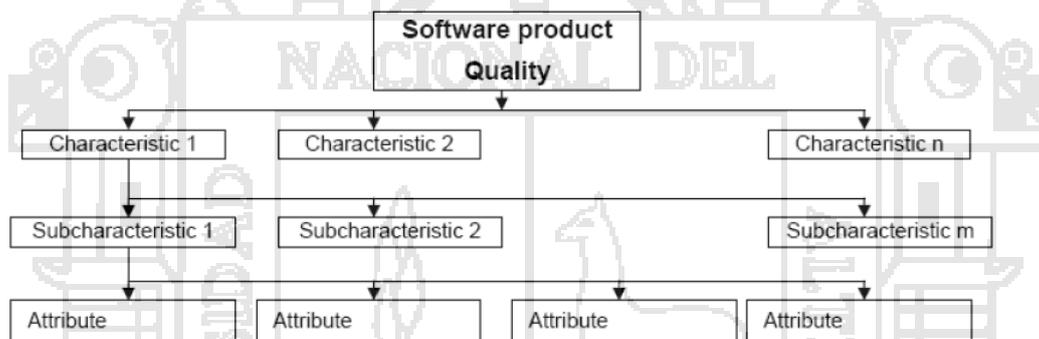
La calidad interna del software tiene un impacto en la calidad externa del software, que tiene a su vez un impacto en la calidad en uso.

– Estructura del modelo de calidad

El modelo de calidad SQuARE categoriza la calidad del software entre características, las cuales son subdivididas más a fondo en sub características y atributos de calidad.

Figura 9

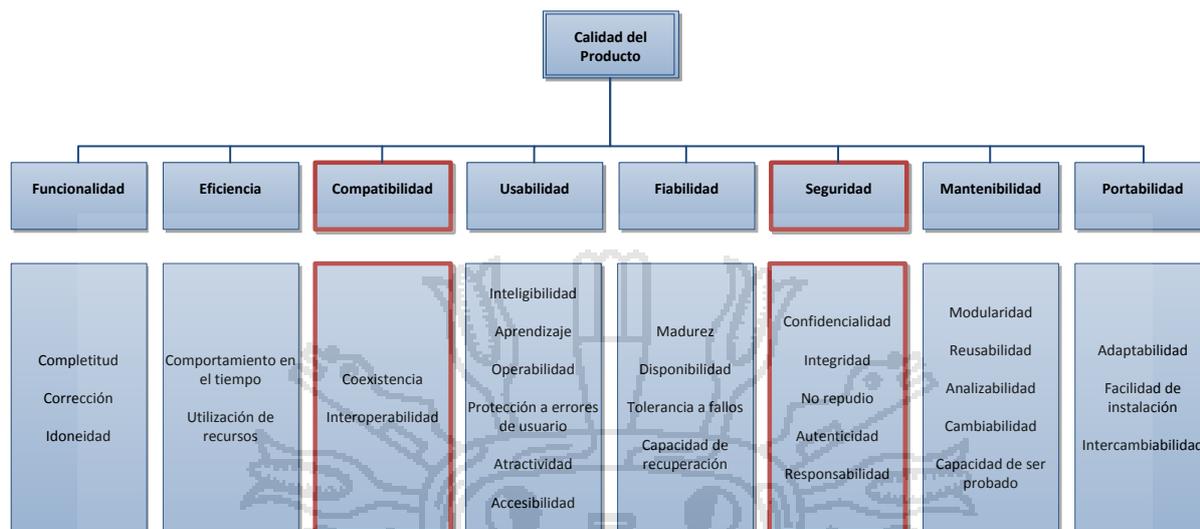
Estructura del modelo calidad.



Fuente: ISO/IEC 25000 (SquaRE), 2005.

El modelo de calidad del producto (calidad interna y externa) definido por la ISO/IEC 25010 se encuentra compuesto por ocho características: funcionalidad, seguridad, interoperabilidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, y éstas están divididas en sub características, las cuales se muestran en la siguiente figura:

Figura 10
Características para la calidad del producto.



Fuente: Calidad de Procesos y Productos Software (Rodríguez Monje, 2010).

• **Funcionalidad**

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Complejitud: Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.

Corrección: Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.

Adecuación: Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

• **Eficiencia**

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Comportamiento en el tiempo: Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas.

Utilización de recursos: Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.

- **Compatibilidad**

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Coexistencia: Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.

Interoperabilidad: Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

- **Usabilidad**

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Capacidad para reconocer su adecuación: Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.

Aprendizaje: Capacidad del producto que permite al usuario aprender su aplicación.

Operabilidad: Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

Protección contra errores de usuario: Capacidad del sistema para proteger a los usuarios de hacer errores.

Atractividad: Capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario.

Accesibilidad: Capacidad del producto que permite que sea utilizado por usuarios con determinadas discapacidades.

- **Fiabilidad**

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Madurez: Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.

Disponibilidad: Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.

Tolerancia a fallos: Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.

Capacidad de recuperación: Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

- **Seguridad**

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Confidencialidad: Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.

Integridad: Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.

No repudio: Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.

Responsabilidad. Capacidad de rastrear de forma inequívoca las acciones de una entidad.

Autenticidad. Capacidad de demostrar la identidad de un sujeto o un recurso.

- **Mantenibilidad**

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Modularidad: Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.

Reusabilidad: Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.

Analizabilidad: Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.

Capacidad para ser modificado. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.

Capacidad para ser probado: Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

- **Portabilidad**

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

Adaptabilidad: Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.

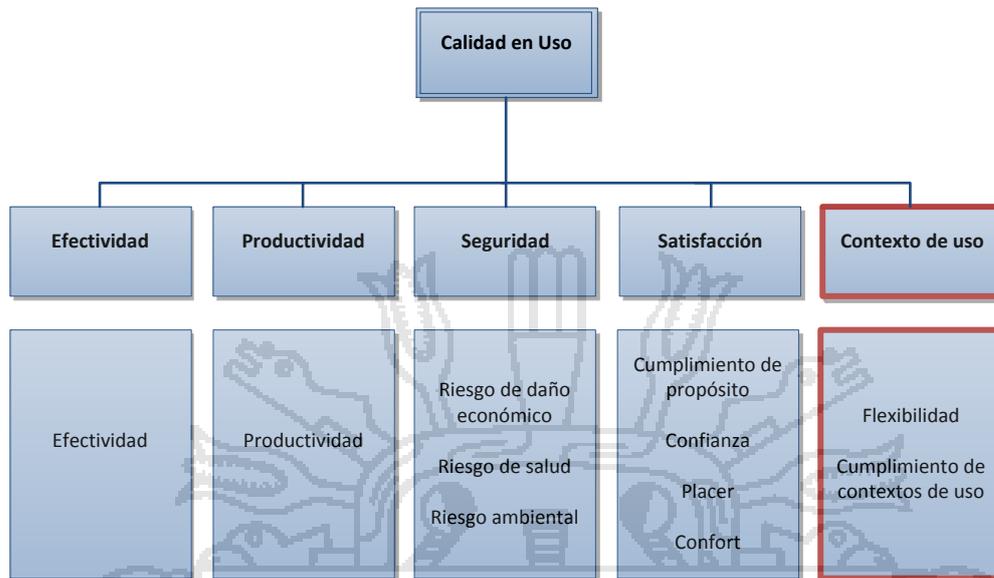
Capacidad para ser instalado: Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.

Capacidad para ser reemplazado. Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

El *modelo de calidad en uso* define cinco características: efectividad, productividad, seguridad, satisfacción y contexto de uso.

Figura 11

Características para la calidad en uso.



Fuente: *Calidad de Procesos y Productos Software* (Rodríguez Monje, 2010).

- **Efectividad**

Capacidad del software de facilitar al usuario alcanzar objetivos con precisión y completitud.

- **Productividad**

Capacidad del software de permitir a los usuarios gastar la cantidad apropiada de recursos en relación a la efectividad obtenida.

- **Seguridad**

Capacidad del software para cumplir con los niveles de riesgo permitidos tanto para posibles daños físicos como para posibles riesgos de datos.

- **Satisfacción**

Capacidad del software de cumplir con las expectativas de los usuarios en un contexto determinado.

- **Contexto de uso**

Capacidad del software de tolerar cambios.

2.2.7. LA INFORMACIÓN Y LA GESTIÓN DE INFORMACIÓN

Para poder comprender la definición de Gestión de Información y lo que ésta conlleva, es importante conceptualizar su centro de importancia: la información.

El término información es usado con diferentes significados por diferentes grupos y en diferentes contextos y además, existe la ambigüedad de términos para ‘datos’ e ‘información’. Una definición muy simple y común afirma que la información es la interpretación de datos. Como resultado, los datos son signos utilizados para representar la información.

El Harrod's Librarian's Glossary, en su novena edición, define información como un conjunto de datos organizados de forma comprensible para comunicar un mensaje que incluye desde el contenido presentado en los más diversos formatos hasta el conocimiento personal de los miembros de una organización (Prytherch, 2000).

En otra definición, son los símbolos, números, letras y signos, que representa los acontecimientos que se producen en el universo y la agrupación y organización de éstos, a través de determinadas reglas y convenciones formales y compartidas, lo que genera datos. Y la información es la contextualización de esos datos (Debons, 1988).

2.2.7.1. Ciclo de Vida de la Información

Las personas, los ecosistemas, las tecnologías e incluso negocios se mueven por ciclos de vida, distintas fases de desarrollo desde el nacimiento pasando por la adolescencia, madurez y declive. Es ahora cuando se comprende que la información también tiene un ciclo de vida durante el cual su valor evoluciona, desde su nacimiento hasta el término de su vida útil. En los negocios, el valor de la información fluctúa ligado normalmente a los ciclos operativos de la organización, como por ejemplo:

- Horas o días en las transacciones que generan las ventas,
- Horas mensuales en el tratamiento de nóminas,

- Horas anuales en los cierres de ejercicio, impuestos, o
- Eventos inesperados como una auditoría o un desastre.

Las empresas cada vez utilizan y generan mayores volúmenes de información, que han de mantener y/o modificar (a menudo de forma impredecible) y que juegan un papel clave en el mantenimiento y crecimiento del propio negocio. Por otra parte, están apareciendo numerosas normativas que establecen el plazo que hay que mantener salvaguardados los correos y otras informaciones en formato electrónico, así como el tiempo máximo para su recuperación y puesta a disposición del organismo solicitante. A medida que el valor de la información varía en el tiempo, parece lógico moverla a sistemas de almacenamiento que ofrezcan prestaciones acordes con su criticidad, con los niveles de protección, replicación y recuperación necesarios, y, sobretodo, al coste más bajo posible.

La automatización de la gestión de esta categorización y traspaso de la información en función de su valía para el negocio es lo que se denomina “Gestión del Ciclo de Vida de la Información” y busca gestionar la información de manera que soporte completamente las necesidades de la empresa, los objetivos de negocio y los niveles de servicio. Esto se traduce en estratificar la información y las aplicaciones que le afectan conforme a su valor para el negocio, al tiempo que se definen políticas que ayuden a determinar este valor y minimizar costes.

El acrónimo en inglés usado para indicar las seis fases en el ciclo de vida de la información es POSMAD. En la siguiente tabla se describen las fases y se dan ejemplos de actividades para cada fase del ciclo de vida y como se aplica a la información.

Tabla 2
Fases del ciclo de vida de la información.

<i>Fases del Ciclo de Vida de la Información (POSMAD)</i>	<i>Definición</i>	<i>Ejemplos de actividades para la información</i>
<i>Planificar</i>	Prepararse para el recurso.	Identificar los objetivos, planificar la arquitectura de información, desarrollar estándares y definiciones. Cuando se modelan, diseñan y desarrollan aplicaciones, bases de datos, procesos, organizaciones, etc., muchas actividades podrían ser consideradas parte de la fase de planificación de la información.
<i>Obtener</i>	Adquirir el recurso.	Crear grabaciones, adquirir data, cargar archivos externos, etc.
<i>Guardar y compartir</i>	Conservar el recurso de la información en un medio electrónico o impreso, y hacer posible su uso a través de un medio de distribución.	Guardar data en bases de datos o algún tipo de archivo, o guardarla como en un medio impreso como un paper.
<i>Mantener</i>	Garantizar que el recurso continúe trabajando apropiadamente.	Actualizar, cambiar, manipular, analizar, estandarizar, validar, o verificar la data; realzar o incrementar la data; enlazar, o relacionar las grabaciones; mezclar o consolidar las grabaciones, etc.
<i>Aplicar</i>	Usar el recurso para lograr las metas.	Revisar la data, usar la información. Esto incluye todos los usos de la información: completar una transacción, escribir un reporte, tomar una decisión administrativa de la información en esos reportes, ejecutar los procesos automáticos, etc.
<i>Disponer</i>	Desechar el recurso cuando ya no se utilice.	Archivar la información; eliminar la data o las grabaciones.

Fuente: *Ten Steps to Quality Data and Trusted Information™* (McGilvray, 2008).

2.2.7.2. Gestión de Información

La Gestión de Información se define como un término impreciso que sirve para designar un conjunto de actividades orientadas a la generación, coordinación, almacenamiento o conservación, búsqueda y recuperación de la información tanto interna como externa contenida en cualquier soporte (Prytherch, 2000).

La Gestión de Información tiene como objetivo optimizar la utilidad y contribución de los recursos de información con el fin de alcanzar los objetivos de la organización. En este sentido, la práctica de la gestión de información se traduce en la creación de

canales y medios para transmitir y acceder a la información, así como, en añadirle valores a ésta (Choo, 2002).

La Gestión de Información surge como un nuevo concepto dentro del campo de la ciencia de la información, orientado al manejo de la inteligencia corporativa de una organización, que permite la estructuración interna a las organizaciones y les permite reaccionar ante los cambios de su entorno apoyándose en el uso de la información y de los recursos de información disponibles. Lo que lleva a concepciones más recientes defendidas por la Gestión de Información, en el sentido de que las organizaciones deben ser consideradas fundamentalmente como sistemas de información.

Algunos autores coinciden en que la Gestión de Información, plantea que "gerencia es todo lo que se refiere a la obtención de la información adecuada, para la persona adecuada, a su precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada". Entonces, la gestión de información debe garantizar que la información esté disponible para cada persona de la organización en el momento requerido (Hill, 2000).

Su objetivo es el de incrementar los niveles de eficiencia y efectividad dentro de una organización. Este proceso se conduce a través de la integración adecuada de los recursos humanos, las políticas, las actividades y procedimientos, el hardware, el software y los datos.

Owen (1989), denomina la Gestión de Información como una triple hélice. En primer lugar lo importante de la información es su contenido y no tanto su soporte, en segundo lugar considera que los gastos para sistemas y tecnologías de la información son un gasto para recursos y no deben ser considerados como gastos generales de funcionamiento, y la tercera parte de la filosofía de la Gestión de Información es la exacta coordinación del recurso dentro de la propia organización, ya que en la actualidad este recurso está muy disperso en diversas empresas.

Los elementos involucrados con la gestión de información se pueden resumir en tres:

- Los que competen a la información como fuente/recurso (procesos productivos al interior de las organizaciones).
- Los relacionados con el usuario de productos y servicios de información.
- Los que conforman el canal de comunicación entre el usuario y la fuente.

El concepto de Gestión de Recursos de Información, se define como “el manejo de la inteligencia corporativa de una organización a objeto de incrementar sus niveles de eficacia, eficiencia y efectividad en el cumplimiento de sus metas” (Paez Urdaneta, 1992).

Otra definición se refiere a que: "La Gestión de Recursos de Información, es la gerencia (planeamiento, organización, operaciones y control) de los recursos (humanos y físicos) que tienen que ver con el apoyo a sistemas (desarrollo, mejoría, mantenimiento) y servicios (procesamiento, transformación, distribución, almacenamiento y recuperación) de la información (datos, textos, imagen) para una organización”.

En cuanto al Gestor de Recursos de Información, parte del supuesto del reconocimiento de que la información es un recurso, un valor que debe ser gerenciado como cualquier otro, pues entre las características similares tenemos:

- Adquirida a un costo, a un precio.
- Tiene características específicas y medibles.
- El valor que tiene la información es difícil de definir ya que en algunos casos la información tiene extrema importancia y en otros esta misma información no “informa” de nada.
- Control del costo de la información.
- Posee un ciclo de vida.
- Capacidad de ser procesada y refinada, entre otras.

"La Gestión de Recursos de Información es una función de alta dirección, para desarrollar una serie de políticas, programas y procedimientos para planificar, gestionar y controlar eficaz y efectivamente las necesidades de información y los recursos de soporte del manejo de la información" (Adams, 2002). En esta tarea directiva, los componentes que participan y las áreas en que debe operar la persona responsable de la misma son:

- Recursos informativos: es decir, identificación, valoración y uso de los mismos, no sólo internos sino también externos.
- Tecnología adecuada: para recoger, almacenar, recuperar y/o difundir la información.
- Gestión: planificación general, recursos humanos, comunicación, contabilidad y marketing.

Como se puede observar, tanto las definiciones de Gestión de Información y de Gestión de Recursos de Información, manifiestan el reconocimiento de la información como el punto clave, decisivo para el mejor desempeño de cualquier organización con o sin fines de lucro.

De lo anterior se puede inferir que la Gestión de Información es más general, y está vinculada con la generación y aplicación de estrategias, políticas y teniendo en cuenta, la cultura organizacional, para lograr un uso efectivo, racional y eficiente de la información. En cuanto a la Gestión de Recursos de Información, ésta se lleva a cabo mediante la planificación, organización, ejecución y control de los recursos humanos y físicos, responsables de crear, explotar y desarrollar el sistema de información integral, que permita el manejo de la información organizacional en función de la consecución de los objetivos de la organización.

Obviamente, es fundamental entender que lo que se debe gestionar, gerenciar es la información y no la tecnología. El conocimiento y la inteligencia potenciados por la información son lo importante, el soporte que contenga, es menos importante.

El obtener un resultado de las tecnologías de la información dependerá de cuán inteligentemente se gestionen. Y parte de esa inteligencia consiste en pasar a entender que la función de las tecnologías de información es gestionar mejor la información, para convertirla en conocimiento, personal u organizacional. Pero, para conseguirlo, tenemos primero que entender que transferir información es muy poco útil, y que la clave está en que los sistemas de información nos permitan intercambiar información o sea transaccionarla.

Desde que surge la Gestión de Información, se ha pasado por diferentes fases que coinciden con diferentes momentos en el desarrollo de las organizaciones. Estas son:

- Gerencia del trabajo impreso: hacia fines del siglo pasado y hasta la década del 50, donde cumple con una función supervisora, de apoyo, una no priorizada.
- Gerencia de la tecnología automatizada: desde los años 60 hasta la década del 70, donde su objetivo era lograr una mayor eficiencia técnica y control físico de las nuevas tecnologías, especialmente en el nivel medio de la organización. Todavía la GI es fragmentada y no coordinada.
- Gerencia de los recursos de información corporativos: mediados de los 70 y hasta la década del 80; constituye una función de apoyo en el nivel máximo de la organización. En esta etapa, los objetivos estratégicos se centran en el tratamiento de la información como recurso clave de la organización, recurso que se puede tratar como otro recurso estratégico. Desde esta óptica, eleva la posición de los gerentes de recursos de información dentro de la misma organización.

- Gerencia del conocimiento y análisis comercial-competitivo: fines de los 80's y años 90's. En este punto, el objetivo es lograr ventajas competitivas en unidades de negocios y estrategia competitiva. Se centra en el contenido de la información como tal y cómo es utilizada y valorada en la organización y cómo ésta colabora en el crecimiento de la organización.
- Gerencia estratégica de la información: década del 90. Su objetivo se ubica en la dirección y la estrategia corporativa, poniendo énfasis en la calidad de la toma de decisiones y en el uso de la información necesaria para perfeccionar el funcionamiento general de la organización.

Más allá de lo anteriormente dicho, no existen definiciones sistematizadas de gerencia de información y de gerencia de recursos de información. Pero si podemos decir que la Gestión de Recursos de Información es un tema emergente, cuya metodología y estrategias están en un cambio constante. Su desarrollo y comprensión es vital para lograr alcanzar una visión estratégica del impacto de los recursos de información y la tecnología utilizada, para un eficaz desempeño de las organizaciones.

Cada vez son más las distintas organizaciones que al ver aumentar la competitividad, sienten la necesidad de incorporar innovaciones para sobrevivir. Las más "rápidas" comienzan a comprender el valor de la gestión de la información para orientar globalmente sus negocios. Esto significa un cambio estructural de todo el sistema informativo de la institución, una reestructuración que permita percibir mejor las variaciones del medio y dar, de este modo, respuestas acordes con el momento. Esto significa que organizar cualquier tipo de institución como un sistema de información, no es sólo incluir una innovación, sino además, prepararla para que pueda estar siempre en "estado de innovación". La Gestión de Información debe colaborar con todas las

innovaciones que la organización pretenda incorporar en forma rápida y oportuna, y ella misma debe ser una innovación.

Por lo tanto, la gerencia de información en las organizaciones conforma un factor que no sólo puede influir en el desarrollo de la actividad sino también en la posición que ocupe en la misma institución. En síntesis, el consumo de información contribuye a aumentar la productividad, así como la competitividad, la eficiencia y efectividad.

2.3. DEFINICIÓN DE TÉRMINOS BÁSICOS

- **BOOTSTRAP:** Es otra de las iniciativas para resolver la crisis del desarrollo de software. Esta metodología mediante prácticas y herramientas y estándares de calidad internacional; mide, evalúa y propone mejoras al proceso de desarrollo de software.
- **Browser o Navegador:** Programa de aplicación que actúa como entorno cliente para un sitio o aplicación Web. El navegador presenta el contenido y actúa como entorno de ejecución para el código móvil enviado por el servidor Web al computador del usuario. No todos los navegadores implementan los estándares de la misma manera, por tanto para asegurar que los usuarios tengan una experiencia homogénea y de la calidad esperada se requiere tomar acuerdos respecto al navegador seleccionado o trabajar con el mínimo común denominador entre los navegadores a utilizar y programar pruebas extensivas.
- **Calidad:** Conjunto de propiedades y características (implícitas o establecidas) de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades implícitas o establecidas.
- **Calidad interna:** Capacidad de un conjunto estático de atributos para satisfacer las necesidades declaradas e implícitas de un producto software bajo ciertas condiciones especificadas.

- **Calidad externa:** Capacidad de un producto software para desarrollar el comportamiento de un sistema de forma que satisfaga las necesidades declaradas e implícitas de un sistema utilizado bajo ciertas condiciones especificadas.
- **Calidad en uso:** Grado en que un producto satisface objetivos con efectividad, seguridad, satisfacción y productividad.
- **CMM:** (Capability Maturity Model), es un modelo de evaluación de los procesos de una organización.
- **DataTables:** Es un plugin para la librería jQuery. Es una herramienta altamente flexible que añade controles de interacción avanzados a cualquier tabla HTML. Entre sus características, se puede encontrar:
 - Longitud de paginación variable.
 - Filtrado en tiempo de ejecución.
 - Control eficiente de ancho de columnas.
 - Mostrar información de casi cualquier tipo de fuente DOM, JavaScript array, AJAX y server-side processing (PHP, C#, Perl, Ruby, AIR, Gears, etc.).
 - Completamente internacionalizable.
 - Sistema robusto y muy testado.
 - Gran variedad de plugins incluidos.
- **Gestión de Información:** Conjunto de actividades orientadas a la generación, coordinación, almacenamiento o conservación, búsqueda y recuperación de la información tanto interna como externa contenida en cualquier soporte.

- **ISO:** (International Standards Organization) es una organización de estandarización en cuyo funcionamiento intervienen organismos de todo el mundo interesados en regular y armonizar diversas áreas de la industria.
- **IEC:** (International Electrotechnical Commission) es una organización que también trabaja a nivel mundial en el desarrollo de estándares sobre electricidad y electrónica.
- **PHP:** (Hypertext Preprocessor) Lenguaje de programación especializado para crear “scripts” del lado del servidor Web. Estos “scripts” permiten crear plantillas paramétricas, las cuales al ser interpretadas generan “páginas dinámicas” que varían en función de los datos del sistema.
- **SPICE:** (Software Process Improvement and Capability Determination), es un modelo de valoración de la arquitectura que define los procesos y prácticas aconsejables.
- **SQuaRE:** (Software Product Quality Requirements and Evaluation).
- **WQM:** (Web Quality Model) Modelo de Calidad Web.



CAPÍTULO III. MATERIALES Y MÉTODOS

3.1. TRABAJO EXPERIMENTAL

3.1.1. METODOLOGÍA DE LA INVESTIGACIÓN

Esta tesis, de acuerdo con las características de estudio, la hipótesis, los objetivos y la pregunta de investigación, y al referirnos a la gestión de información en relación directa a los resultados se enmarcó dentro del enfoque de investigación de tipo cuantitativa.

3.1.2. DISEÑO DE LA INVESTIGACIÓN

El diseño de la investigación es cuasi experimental. Los diseños cuasi-experimentales manipulan por lo menos una variable independiente para observar su efecto y relación con una o más variables dependientes.

El grupo experimental estuvo conformado por los usuarios del sistema Web del Colegio de Ingenieros del Perú CD Puno.

La representación gráfica es la siguiente:

$$G_1 : O_1 X O_2$$

Donde:

G_1 : Grupo experimental.

X: Tratamiento con la implementación del Sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE).

O_1 : Test antes del experimento.

O_2 : Test después del experimento.

Este diseño con grupo experimental permitió la comparación de resultados Pre-test y Post-test, con un alto grado de probabilidad, que el Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE) (variable independiente), ha sido factor determinante

en la apropiada Gestión de Información en el Colegio de Ingenieros del Perú CD Puno (variable dependiente).

3.1.3. POBLACIÓN Y MUESTRA

3.1.3.1. Población

La población objeto de estudio está delimitada y constituida por el promedio mensual de ingenieros miembros del Colegio de Ingenieros del Perú CD Puno que realiza sus aportaciones, los mismos que están agrupados en 16 capítulos (Agrícolas, Agrónomos, Agroindustriales, Civiles, Economistas, Electricistas, Electrónicos, Estadísticos, Geólogos, Mecánicos electricistas, Metalurgistas, Minas, Pesqueros, Químicos, Sistemas, Topógrafos y Agrimensores), haciendo un total de 440 miembros entre hábiles y no hábiles.

3.1.3.2. Muestra

Para la prueba de hipótesis, la muestra se determinó de forma aleatoria, siendo 29 los sujetos de estudio.

3.2. SISTEMA DE VARIABLES

3.2.1. DEFINICIÓN DE VARIABLES

Independiente: Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE).

Dependiente: Gestión de información.

3.2.2. OPERACIONALIZACIÓN DE VARIABLES

Tabla 3

Operacionalización de variables.

VARIABLES	DIMENSIONES	INDICADORES	ESCALA
Variable Independiente: Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE).	Funcionalidad: Capacidad del software de proveer los servicios necesarios para cumplir con los requisitos funcionales.	<ul style="list-style-type: none"> - Adecuación. - Exactitud. - Interoperabilidad. 	<ul style="list-style-type: none"> - Sí. - No.
	Eficiencia: relación entre las prestaciones del software y los requisitos necesarios para su utilización.	<ul style="list-style-type: none"> - Cantidad de recursos hardware y software. - Tiempo necesario para completar exitosamente una transacción. - Número de transacciones completadas con éxito en un período de tiempo dado. 	<ul style="list-style-type: none"> - Muy bueno - Bueno - Regular - Malo - Muy malo
	Compatibilidad: Capacidad del software para comunicarse con otras aplicaciones o sistemas.	<ul style="list-style-type: none"> - Coexistencia. - Interoperabilidad. 	<ul style="list-style-type: none"> - Sí. - No.
	Usabilidad: esfuerzo requerido por el usuario para utilizar el producto satisfactoriamente.	<ul style="list-style-type: none"> - Facilidad de aprendizaje. - Operabilidad. - Atractividad. 	<ul style="list-style-type: none"> - Muy bueno. - Bueno. - Regular. - Malo. - Muy malo.
	Fiabilidad: capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.	<ul style="list-style-type: none"> - Tolerancia a fallos. - Capacidad de recuperación. 	<ul style="list-style-type: none"> - Muy bueno. - Bueno. - Regular. - Malo. - Muy malo.



<p>Seguridad: capacidad del software para cumplir con los niveles de riesgo permitidos tanto para posibles daños físicos como para posibles riesgos de datos.</p>	<ul style="list-style-type: none"> - Confidencialidad. - Integridad. - Autenticidad. - No repudio. - Responsabilidad. - Riesgo de daño económico. 	<ul style="list-style-type: none"> - Sí. - No.
<p>Mantenibilidad: esfuerzo necesario para adaptarse a las nuevas especificaciones y requisitos del software.</p>	<ul style="list-style-type: none"> - Modularidad. - Reusabilidad. - Analizabilidad. - Cambiabilidad. - Capacidad de ser probado. 	<ul style="list-style-type: none"> - Sí. - No.
<p>Portabilidad: capacidad del software ser transferido de un entorno a otro.</p>	<ul style="list-style-type: none"> - Adaptabilidad. - Facilidad de instalación. - Coexistencia. - Intercambiabilidad. 	<ul style="list-style-type: none"> - Sí. - No.
<p>Efectividad: capacidad del software de facilitar al usuario alcanzar objetivos con precisión y completitud.</p>	<ul style="list-style-type: none"> - Precisión de uso. - Tiempo de operación. - Número de transacciones diarias. - Número de sesiones de trabajo diarias. 	<ul style="list-style-type: none"> - Muy bueno. - Bueno. - Regular. - Malo. - Muy malo.
<p>Productividad: capacidad del software de permitir a los usuarios gastar la cantidad apropiada de recursos en relación a la efectividad obtenida.</p>	<ul style="list-style-type: none"> - Número de transacciones diarias (aportaciones y otros trámites). - Número de sesiones de trabajo diarias. 	<ul style="list-style-type: none"> - Muy bueno. - Bueno. - Regular. - Malo. - Muy malo.
<p>Satisfacción: capacidad del software de cumplir con las expectativas de los usuarios en un</p>	<ul style="list-style-type: none"> - Cumplimiento del propósito. - Confianza. - Placer. 	<ul style="list-style-type: none"> - Sí. - No.



	contexto determinado.	- Confort.	
	Contexto de uso: capacidad del software de tolerar cambios.	- Flexibilidad.	- Sí. - No.
Variable dependiente: Gestión de Información	Productividad: Consumo adecuado de recursos humanos y físicos para el manejo de la información y el logro de objetivos.	- Nivel recursos utilizados.	- Muy bueno. - Bueno. - Regular. - Malo. - Muy malo.
	Eficiencia: capacidad de gestionar la información en el tiempo adecuado.	- Tiempo necesario para completar una tarea.	- Muy bueno. - Bueno. - Regular. - Malo. - Muy malo.
	Efectividad: capacidad de gestionar la información para alcanzar los objetivos con precisión y completitud.	- Precisión de la información.	- Muy bueno. - Bueno. - Regular. - Malo. - Muy malo.

Fuente: Elaboración propia.

3.3. MATERIAL EXPERIMENTAL

3.3.1. MÉTODOS DE RECOPIACIÓN DE DATOS

Para la recopilación de datos se tomó las siguientes consideraciones específicas, según la metodología de la investigación científica:

- Identificar la información que se requiere para hacer la investigación.
- Señalar la clase de información que se requiere.
- Especificar los requerimientos que se tiene que emplear para conseguir la información.

- Señalar las fuentes de información (dónde se obtendrá la información).

Estos procedimientos sistemáticos y estandarizados usados en la investigación ayudaron a obtener medidas variables y de esta manera se pudo proporcionar evidencias empíricas a la investigación.

Luego se procedió a obtener las fuentes de información primarias mediante:

- ***Encuestas:***

Se ha utilizado una encuesta dirigida a los usuarios del sistema en la que se ha planteado 13 preguntas (considerando los atributos del modelo de calidad SQuaRE), que han posibilitado respuestas cualitativas y valorativas referentes al proyecto, dichas respuestas han sido sistematizadas con el fin de establecer cuáles son las necesidades y/o requerimientos de los usuarios en relación al nuevo sistema y sus funcionalidades.

- ***Registro de Observación:***

Se ha utilizado la observación para obtener las características sobre el desempeño del sistema. Esta guía de observación se realizó en tiempo real con el propósito de determinar las cualidades y beneficios del sistema basados en la norma ISO/IEC 25010:2010 (SQuaRE).

3.3.2. MÉTODO DE TRATAMIENTO Y ANÁLISIS DE DATOS

Para el tratamiento de los datos de esta investigación se siguió los siguientes procesos:

- ***Recopilación y tabulación de datos:*** Se usaron las encuestas como instrumentos de recolección de datos, luego se registraron los datos y posteriormente se realizó su tabulación haciendo uso de un software estadístico.
- ***Análisis y consistencia de Datos:*** Se realizó la consistencia de datos recopilados (Control y Verificación) para realizar el análisis estadístico,

usando los softwares estadísticos IBM SPSS Statistics v.21, MINITAB 16 y Microsoft Excel 2010.

- *Interpretación de datos y validación de la Hipótesis mediante la prueba de*

Hipótesis: Se realizó la prueba de hipótesis para las variables de respuestas usando el estadístico t-Student.

$$t = \frac{\bar{D} - 0}{S_{\bar{D}}}$$

Donde:

\bar{D} : Es la media de la diferencia de medias de la muestra.

$S_{\bar{D}}$: Es el error típico de la diferencia de medias.

3.4. MATERIAL APLICATIVO

3.4.1. METODOLOGÍA DE DESARROLLO DEL SISTEMA

La metodología de desarrollo está basada en el modelo espiral de Boehm, el cual se enfoca en abordar los riesgos en forma incremental, de acuerdo a su prioridad. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por la de riesgos más asumibles y se hace un ciclo de la espiral. Si el cliente quiere seguir haciendo mejoras en el software, se vuelven a evaluar las nuevas alternativas y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto software desarrollado sea aceptado y no necesite seguir mejorándose con otro nuevo ciclo. Cada ronda está compuesta por cuatro fases. Durante la primera fase los desarrolladores exploran alternativas, definen restricciones e identifican los objetivos. Durante la segunda fase los desarrolladores manejan los riesgos asociados con las soluciones definidas durante la primera fase. Durante la tercera fase los desarrolladores realizan y validan un prototipo o la parte del sistema asociada con los riesgos que se tratan en esta ronda. La cuarta fase se enfoca en la planeación de la siguiente ronda con base en los resultados producidos en la ronda actual. La última fase de la ronda se

realiza por lo general, como una revisión que involucra a los participantes en el proyecto, incluyendo a los desarrolladores, clientes y usuarios. Esta revisión abarca los productos desarrollados durante las rondas anteriores y los planes para la siguiente ronda.

El modelo espiral de Boehm distingue entre las siguientes rondas: conceptos de operación, requerimientos de software, diseño de productos de software, diseño detallado, código, prueba unitaria, integración y pruebas, prueba de aceptación, implementación.

Este modelo fue propuesto por Boehm en 1988 en su artículo “A Spiral Model of Software Development and Enhancement”. Cada ronda sigue el modelo de cascada e incluye las siguientes actividades:

- Determinar objetivos.
- Especificar restricciones.
- Generar alternativas.
- Identificar los riesgos.
- Resolver los riesgos.
- Desarrollar y verificar el producto del siguiente nivel.
- Planear.

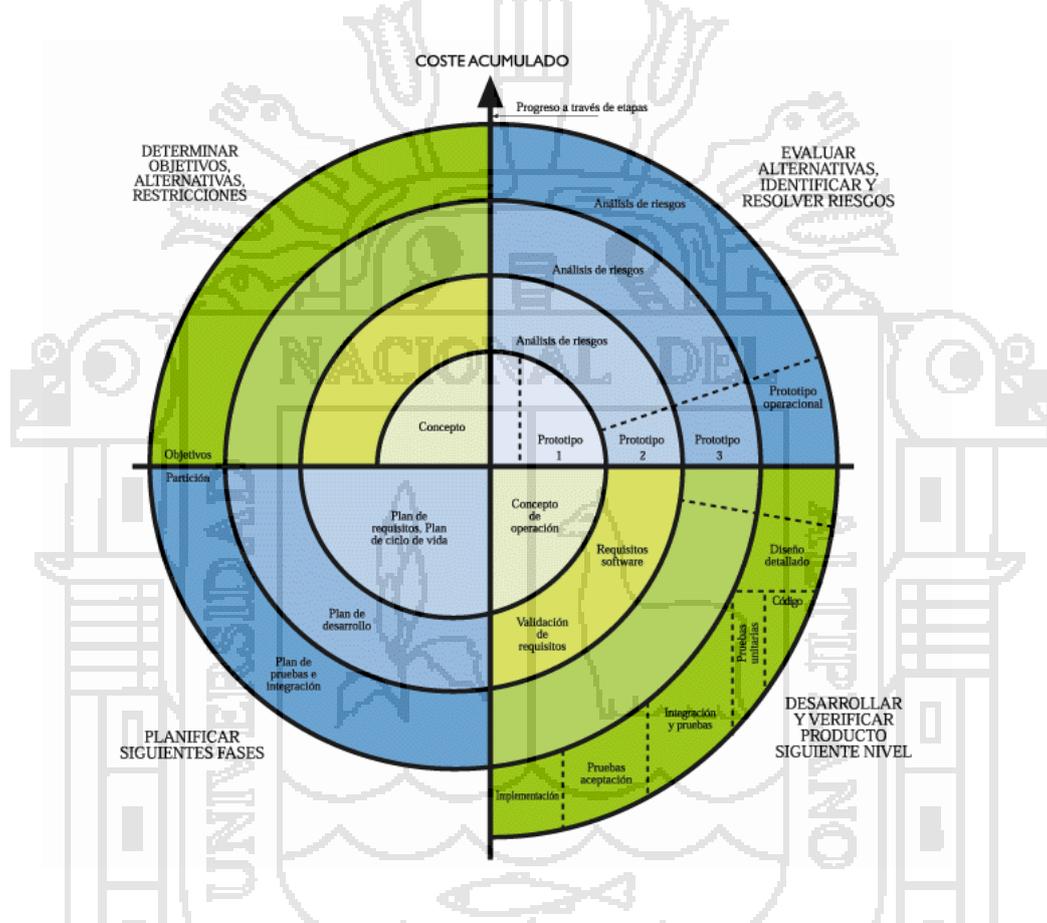
Las dos primeras actividades definen el problema que se trata en el ciclo actual. La tercera actividad, generar alternativas, define el espacio de solución. La cuarta y quinta sirven para identificar problemas futuros que pueden causar costos elevados o la cancelación del proyecto. La penúltima actividad define la realización del ciclo. La última actividad es una tarea administrativa que prepara el siguiente ciclo.

Estas rondas pueden verse en un sistema de coordenadas polares, como se muestra en la figura 12. La primera ronda, conceptos de operación, se inicia en el cuadrante

superior izquierdo. Las rondas subsiguientes se representan como capas adicionales de la espiral. La notación permite que se determine con facilidad el estado del proyecto a lo largo del tiempo. La distancia con respecto al origen es el costo acumulado del proyecto. La coordenada angular indica el avance logrado en cada fase.

Figura 12

El modelo espiral de Boehm.



Fuente: (Dávila Nicanor & Mejía Álvarez, 2003).

Además, este modelo posee las siguientes ventajas frente a otros modelos:

- El análisis de riesgos se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos. Entre ellos:
 - Reduce riesgos del proyecto.
 - Incorpora objetivos de calidad.
 - Integra el desarrollo con el mantenimiento.

- También es posible tener en cuenta mejoras y nuevos requerimientos sin romper con el modelo, ya que el ciclo de vida no es rígido ni estático.
- Mediante este modelo se produce software en etapas tempranas del ciclo de vida y suele ser adecuado para proyectos largos de misión crítica.

3.4.2. HARDWARE Y SOFTWARE DE DESARROLLO

3.4.2.1. Hardware

El hardware utilizado en el desarrollo de la investigación, es el siguiente:

- 01 Portátil Sony VAIO Intel (R) Core (TM) i5-2410M CPU @ 2.30 GHz 4GB RAM,
- 01 Portátil Compaq Presario 700 Intel (R) Pentium (R) Dual Core 2 GB RAM.
- 01 Computadora de escritorio: Intel Core 2 Duo 2 GB RAM.
- 01 Impresora: HP LaserJet 1020.

3.4.2.2. Software

El software utilizado en el desarrollo de la investigación, es el siguiente:

- Sistema operativo: Windows 7 Professional y Ultimate.
- Lenguaje de programación Web: PHP v5.3.8.
- Gestor de Base de Datos: MySQL v5.5.16.
- Modelador de Base de Datos: MySQL Workbench 5.2 CE.
- Framework: CodeIgniter v2.1.3.
- Diseño de interfaces: Bootstrap v2.0.4
- Entorno de desarrollo Web: WampServer 2.2
- Entorno de programación: NetBeans IDE 7.3.
- Paquetes estadísticos: IBM SPSS Statistics 21.0, Minitab 16.2.3.
- Herramienta CASE: Visual Paradigm for UML 8.0 Enterprise Edition.



CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. ANÁLISIS Y DISEÑO DEL SISTEMA WEB

4.1.1. ANÁLISIS DEL SISTEMA WEB

Esta parte describe lo que el sistema Web debe hacer según la especificación de requerimientos, este análisis consiste en describir lo que el sistema Web, proporcionará en términos de funcionalidad, el cual define requisitos funcionales y operativos del sistema.

Se han utilizado los diagramas de Casos de Uso (CU) para describir la relación entre los actores y actividades, los cuales nos permite definir los requerimientos del sistema Web.

4.1.1.1. Actores del Sistema Web

Los actores del sistema Web fueron identificados como:

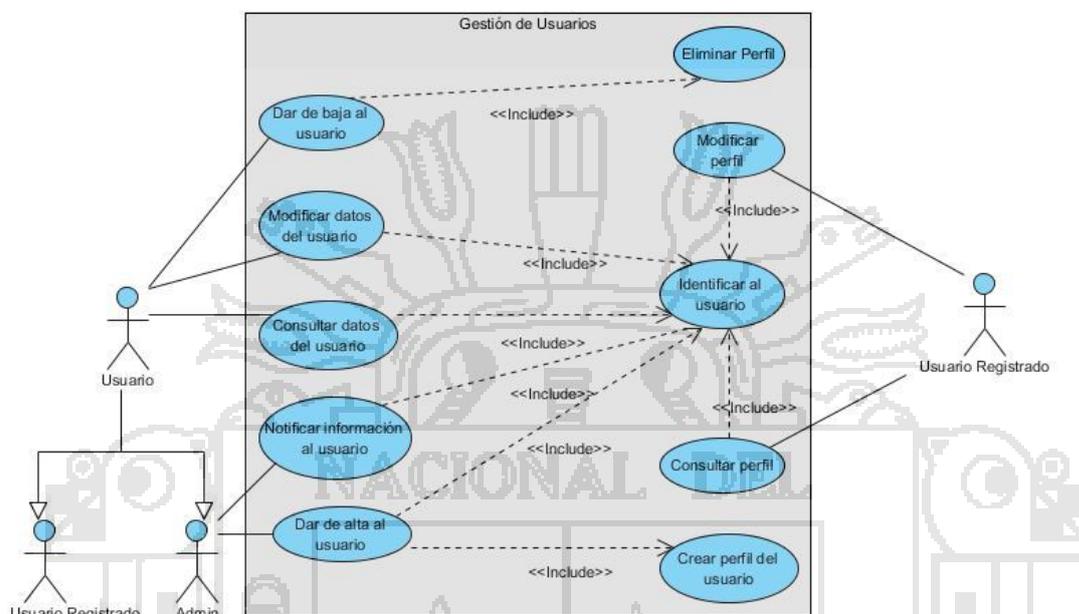
- *Actor Usuario:* Este actor representa al grupo de usuarios (registrados, ingenieros y administradores) que interactúan con el sistema Web.
- *Actor Usuario Registrado:* Este actor representa los usuarios conectados a Internet y debidamente registrados en el sistema, que está encargado del manejo del sistema Web.
- *Actor Admin:* Este actor representa los usuarios con capacidad total para gestionar el sistema Web, es decir, realiza las tareas de gestión y control de todos los servicios del sistema.
- *Actor Ingeniero:* Ingeniero colegiado que realiza el pago de sus aportaciones, pago de servicios de colegiatura, pago de otros servicios, y así mismo, realiza la consulta de sus aportaciones vía Web.

4.1.1.2. Casos de Uso del Sistema Web

Los casos de uso obtenidos como resultado en la etapa de análisis se dividen en: gestión de usuarios, gestión de registro de colegiaturas, gestión de asignación de CIP,

programación de colegiaturas, gestión de colegiaturas, gestión de pagos por juramentación, pago de aportaciones y gestión de consulta de aportaciones.

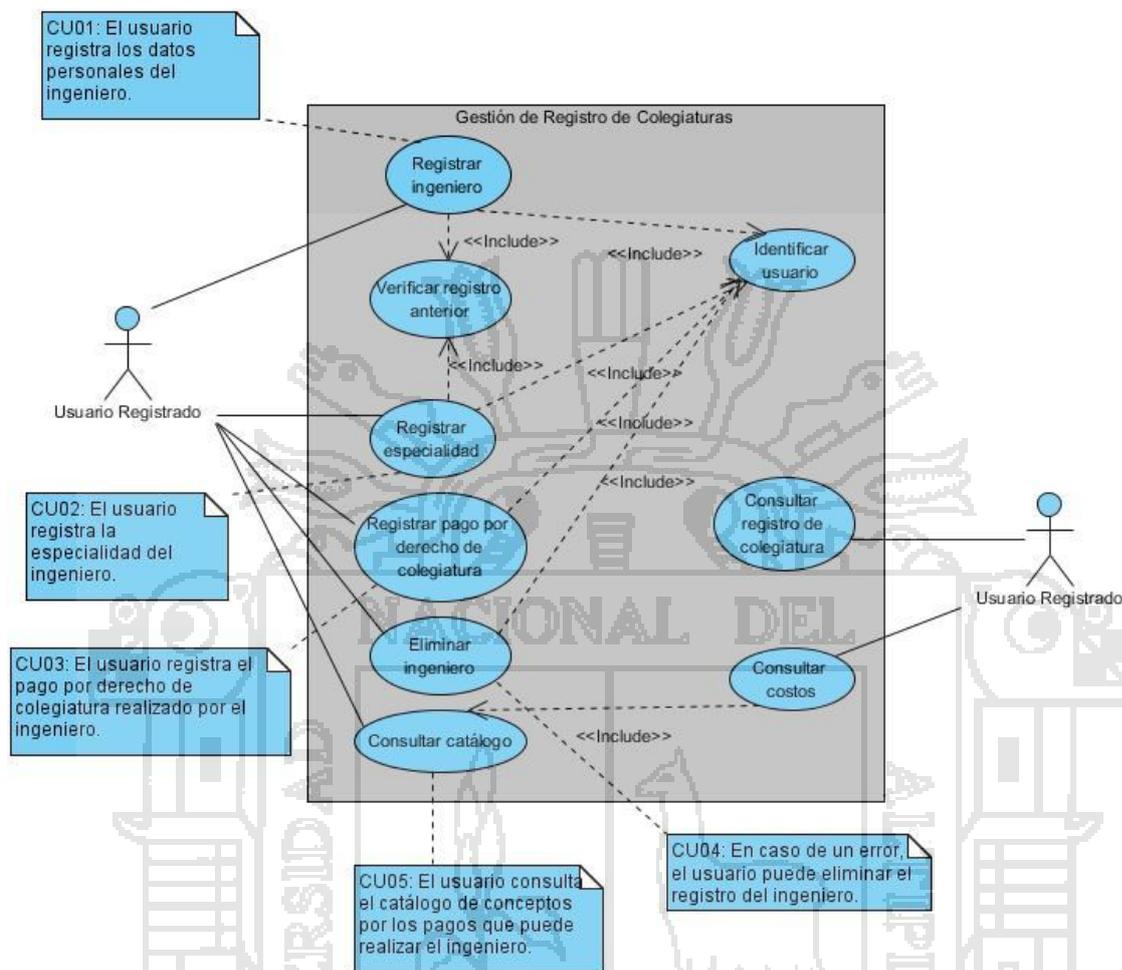
Figura 13
CU para la gestión de usuarios.



Fuente: Elaboración propia.

Figura 14

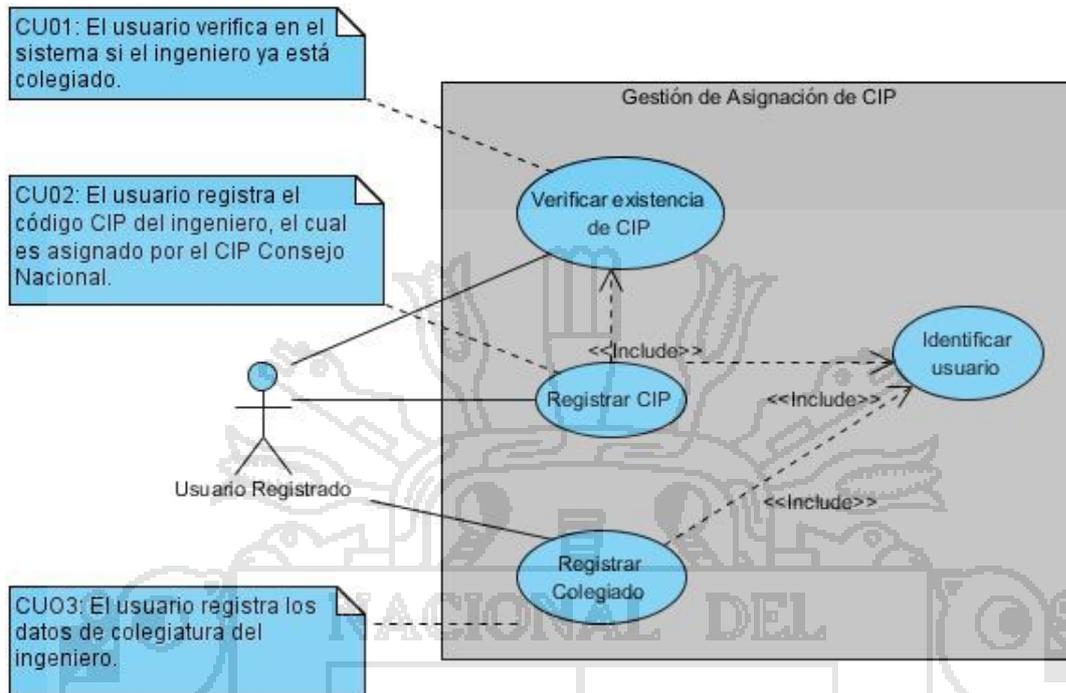
CU para la gestión de registro de colegiaturas.



Fuente: Elaboración propia.

Figura 15

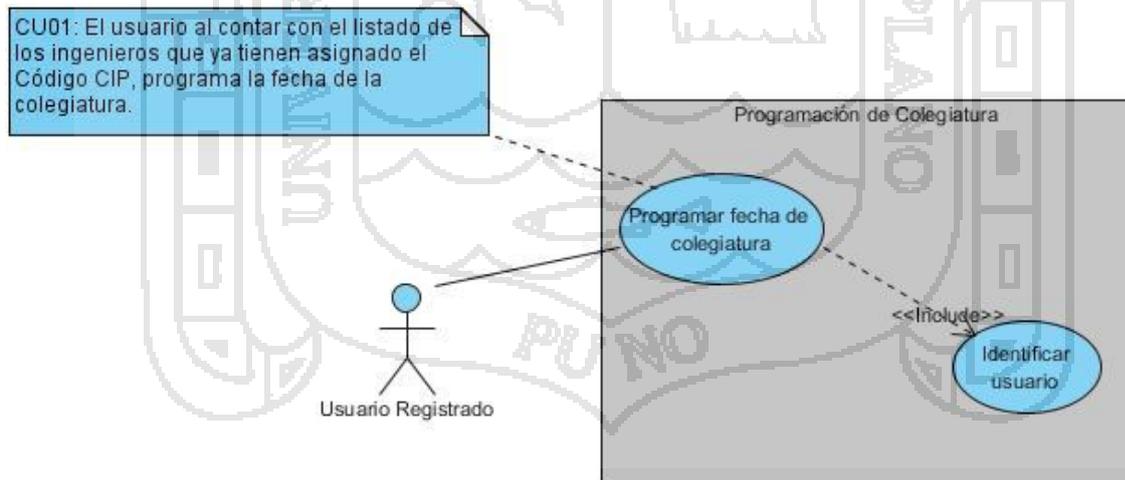
CU para la gestión de asignación de CIP.



Fuente: Elaboración propia.

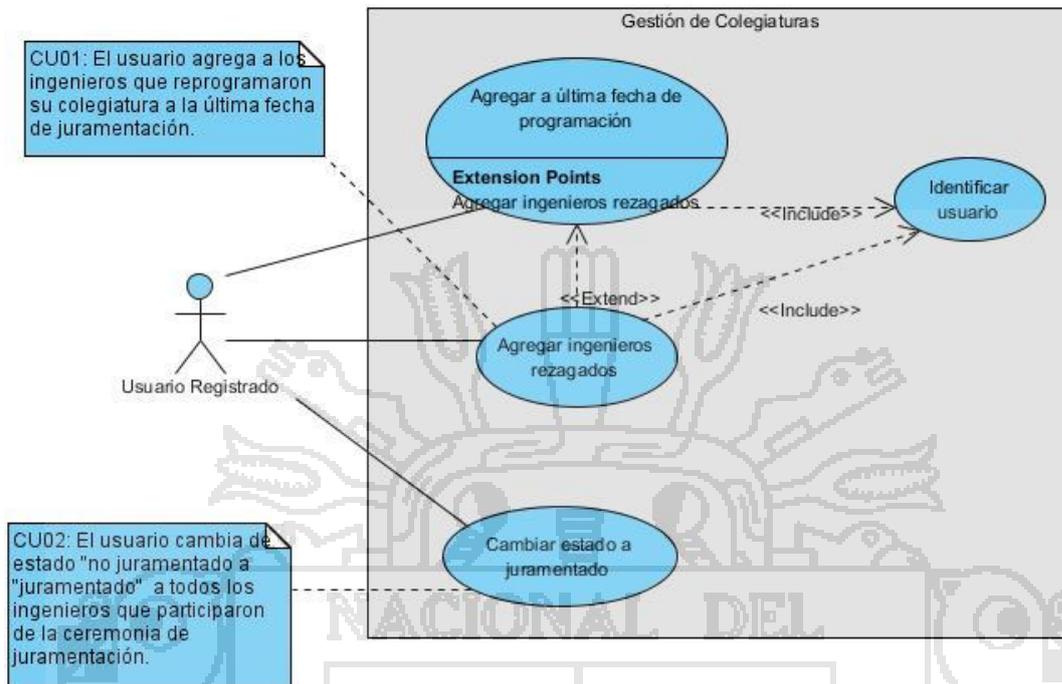
Figura 16

CU para la gestión de programación de colegiaturas.



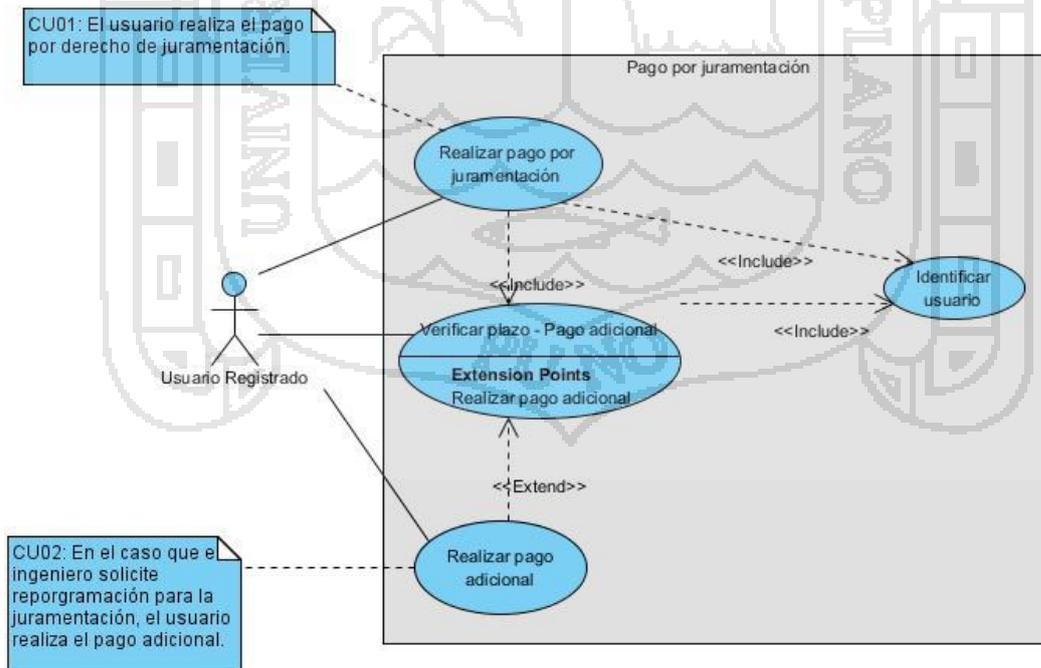
Fuente: Elaboración propia.

Figura 17
CU para la gestión de colegiaturas.



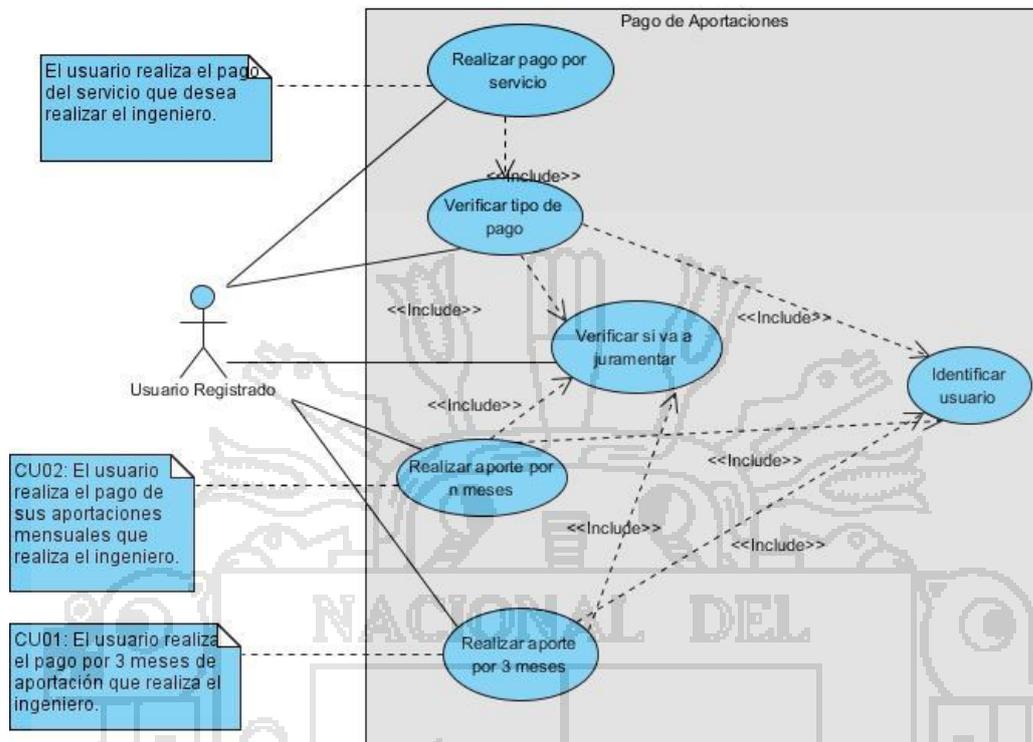
Fuente: Elaboración propia.

Figura 18
CU para la gestión de pagos por juramentación.



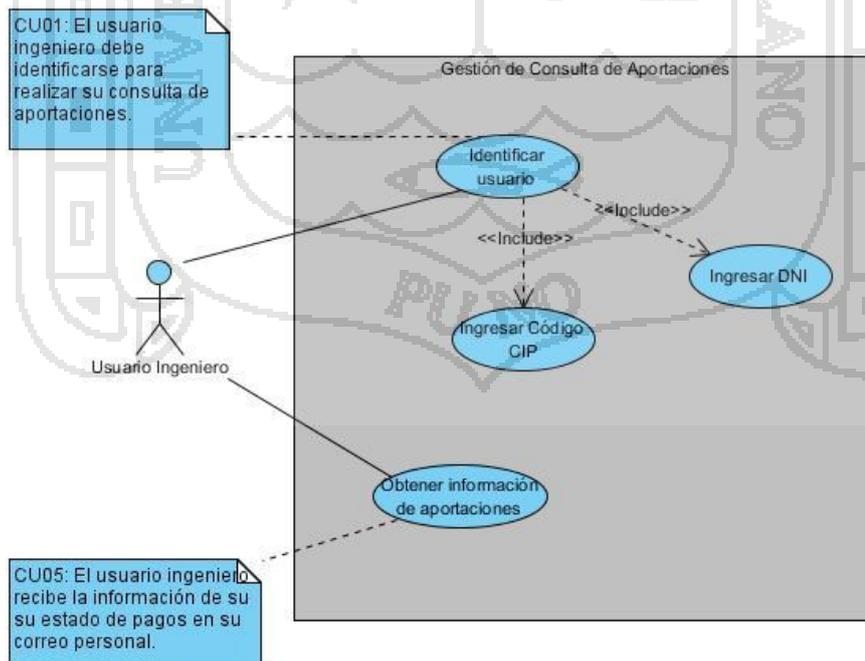
Fuente: Elaboración propia.

Figura 19
CU para el pago de aportaciones.



Fuente: Elaboración propia.

Figura 20
CU para la consulta de aportaciones en línea.



Fuente: Elaboración propia.

4.1.1.3. Especificación de Casos de Uso

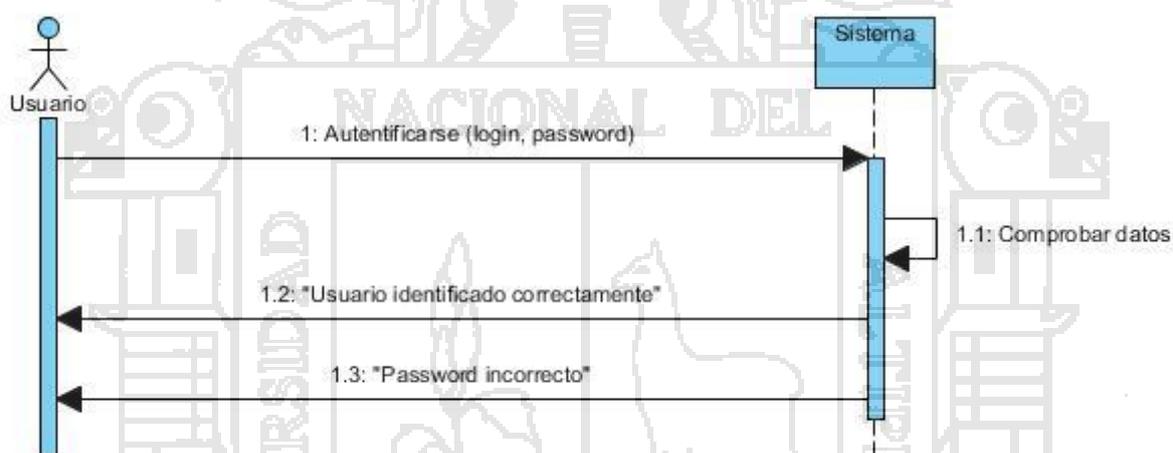
Los casos de uso fueron basados en actividades que realizan los actores. En la especificación de casos de uso se describen esas actividades (Anexo A).

4.1.1.4. Diagramas de Secuencia

Los siguientes diagramas de secuencia permiten visualizar como los objetos identificados del sistema Web se comunican entre sí para generar un evento en el flujo de control a lo largo del tiempo.

Figura 21

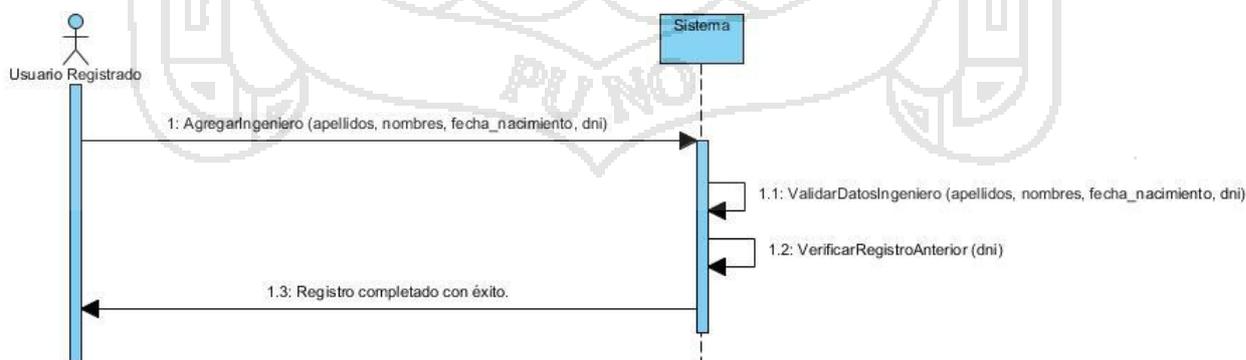
Diagrama de secuencia para identificar al usuario.



Fuente: Elaboración propia.

Figura 22

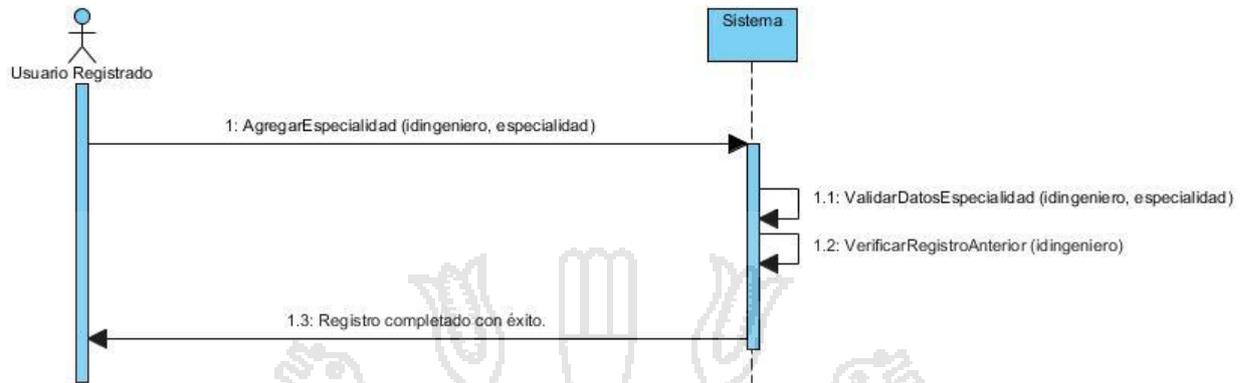
Diagrama de secuencia para el registro de un ingeniero.



Fuente: Elaboración propia.

Figura 23

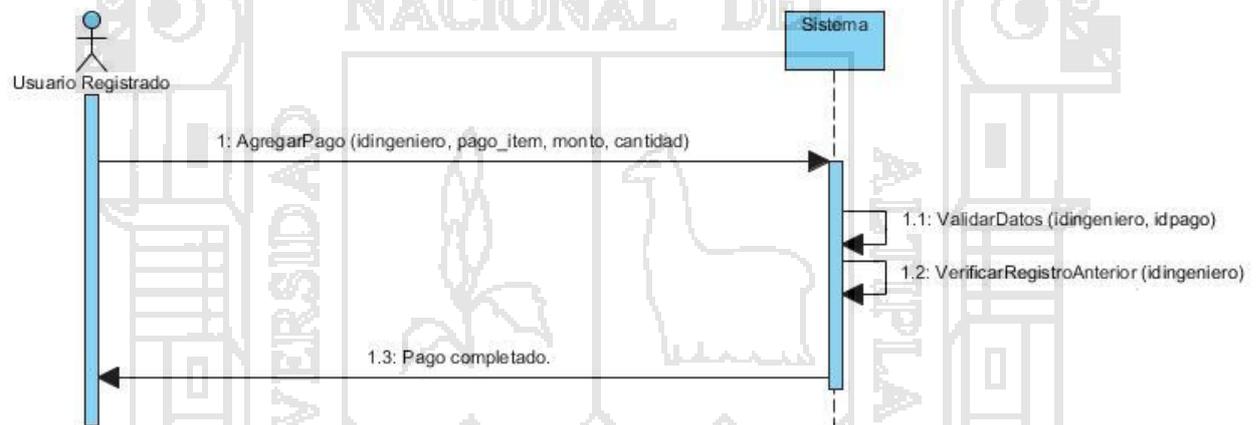
Diagrama de secuencia para registro de una especialidad a un ingeniero.



Fuente: Elaboración propia.

Figura 24

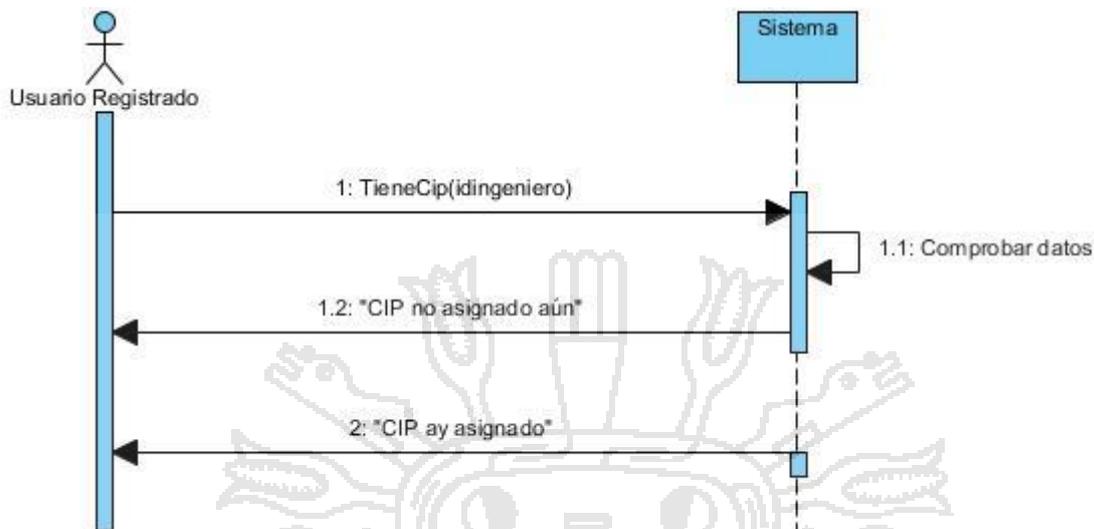
Diagrama de secuencia para registrar el pago de aportaciones de un ingeniero.



Fuente: Elaboración propia.

Figura 25

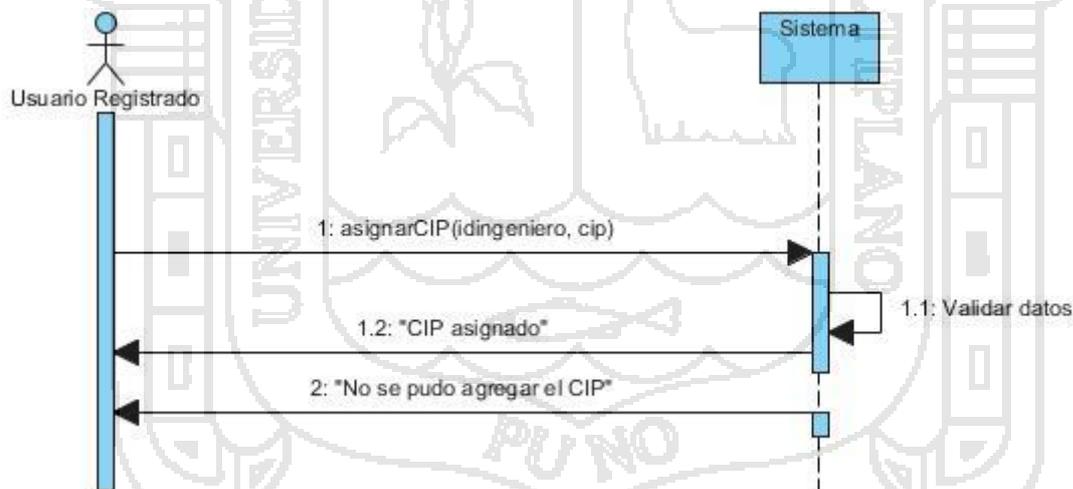
Diagrama de secuencia para verificar la existencia del CIP de un ingeniero.



Fuente: Elaboración propia.

Figura 26

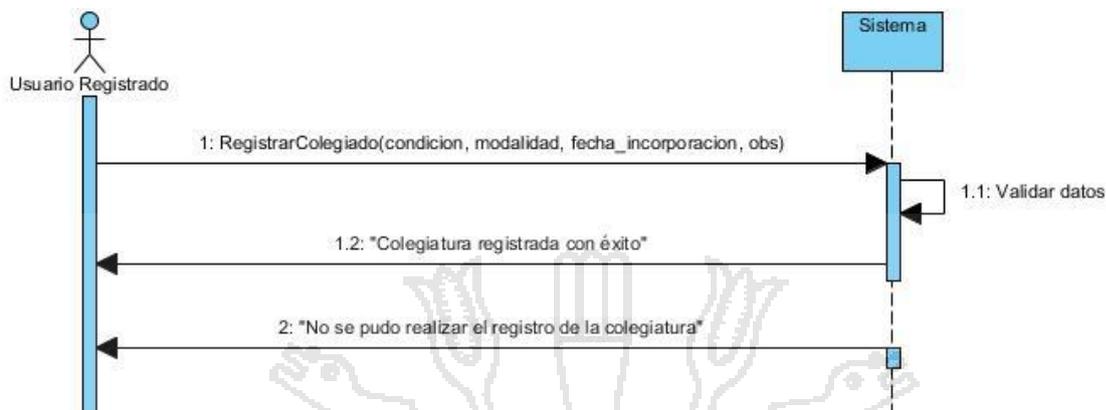
Diagrama de secuencia para asignar un CIP a un ingeniero.



Fuente: Elaboración propia.

Figura 27

Diagrama de secuencia para el registro de una colegiatura.



Fuente: Elaboración propia.

Figura 28

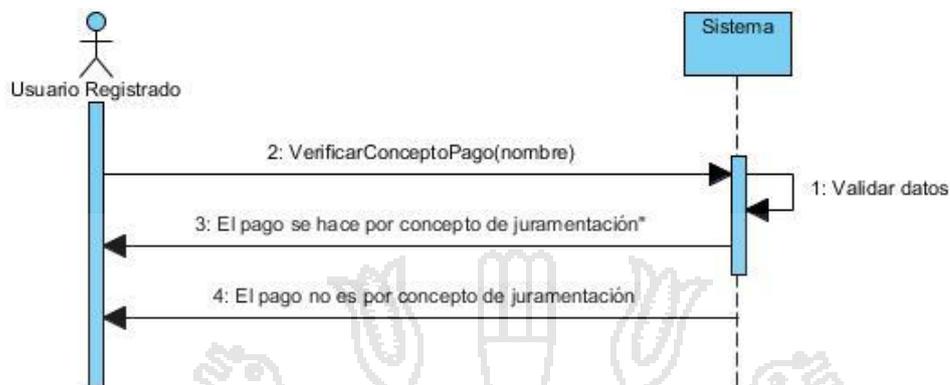
Diagrama de secuencia para el registro de una fecha de colegiatura.



Fuente: Elaboración propia.

Figura 29

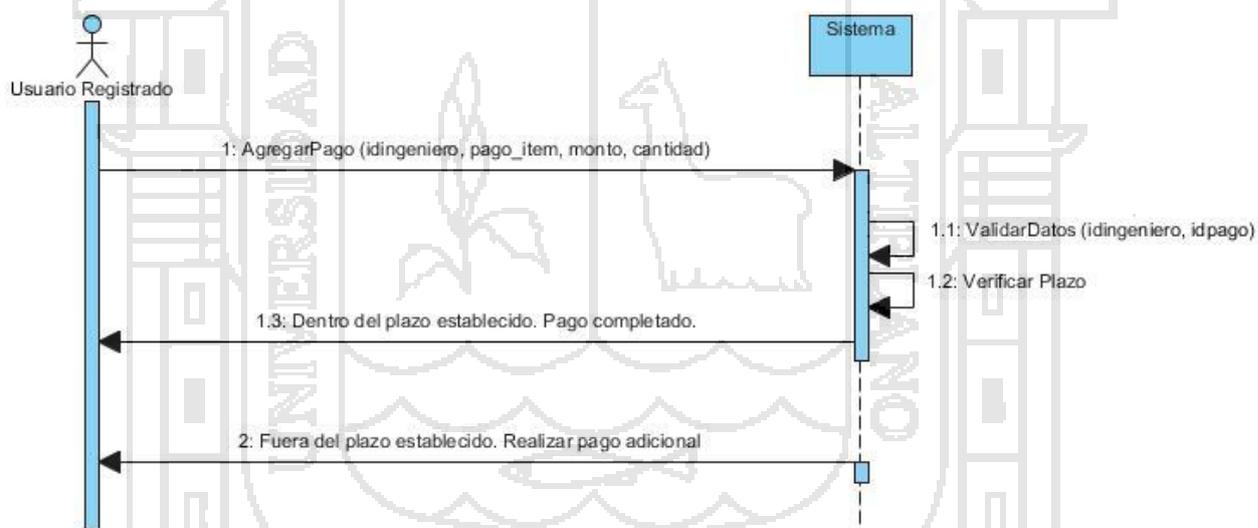
Diagrama de secuencia para verificar el concepto de pago.



Fuente: Elaboración propia.

Figura 30

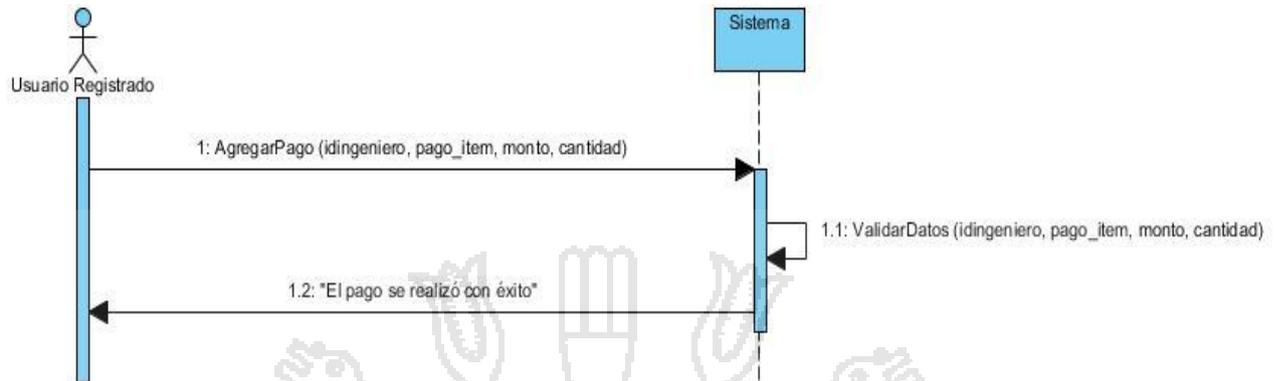
Diagrama de secuencia para registrar el pago por juramentación.



Fuente: Elaboración propia.

Figura 31

Diagrama de secuencia para el registro de un pago.



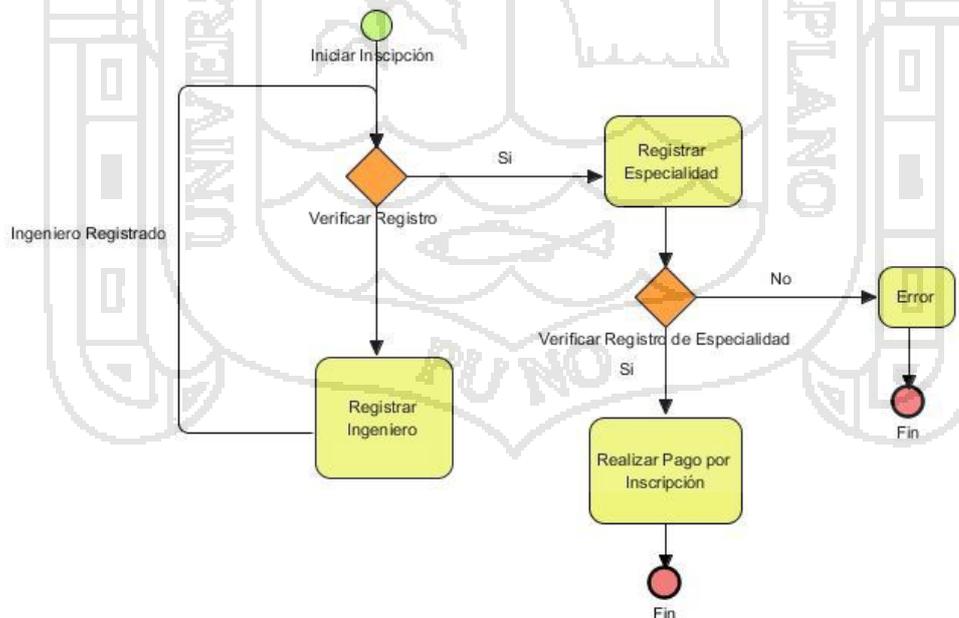
Fuente: Elaboración propia.

4.1.1.5. Diagramas de Actividades

El diagrama de actividades proporcionó una vista del comportamiento centrada en la tarea, ya que su propósito es capturar acciones (trabajo y actividades que serán realizadas) y sus resultados en términos de cambios de estado.

Figura 32

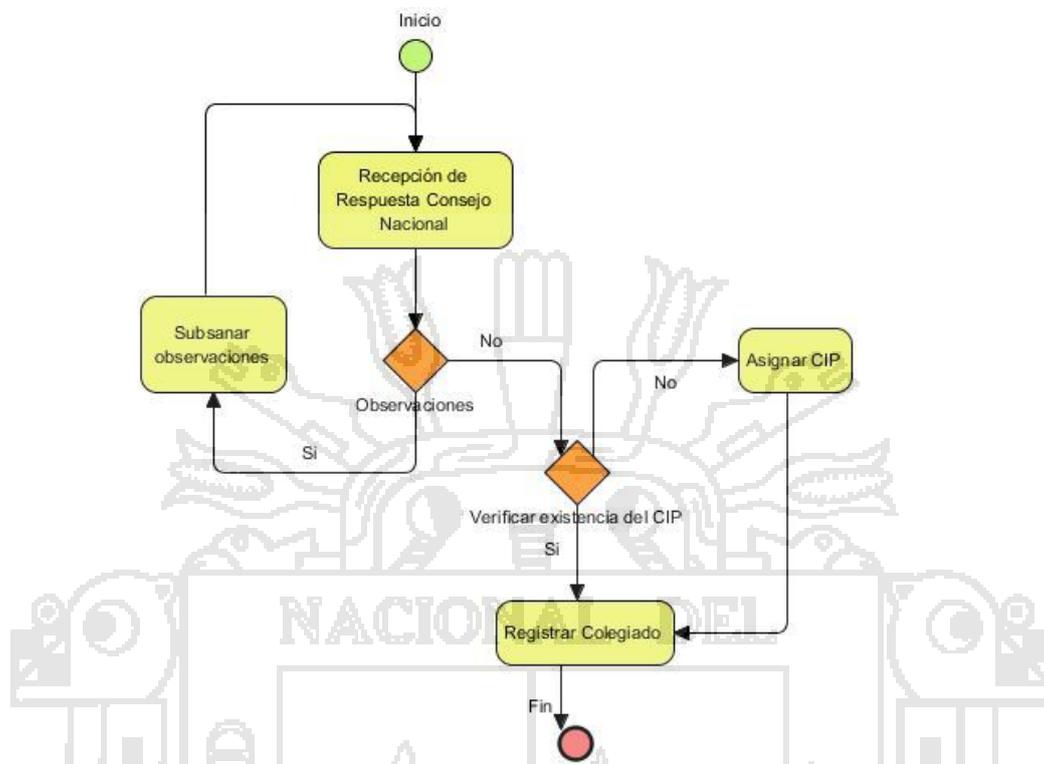
Diagrama de actividad del proceso de registro de colegiatura.



Fuente: Elaboración propia.

Figura 33

Diagrama de de actividad del registro del colegiado.



Fuente: Elaboración propia.

Figura 34

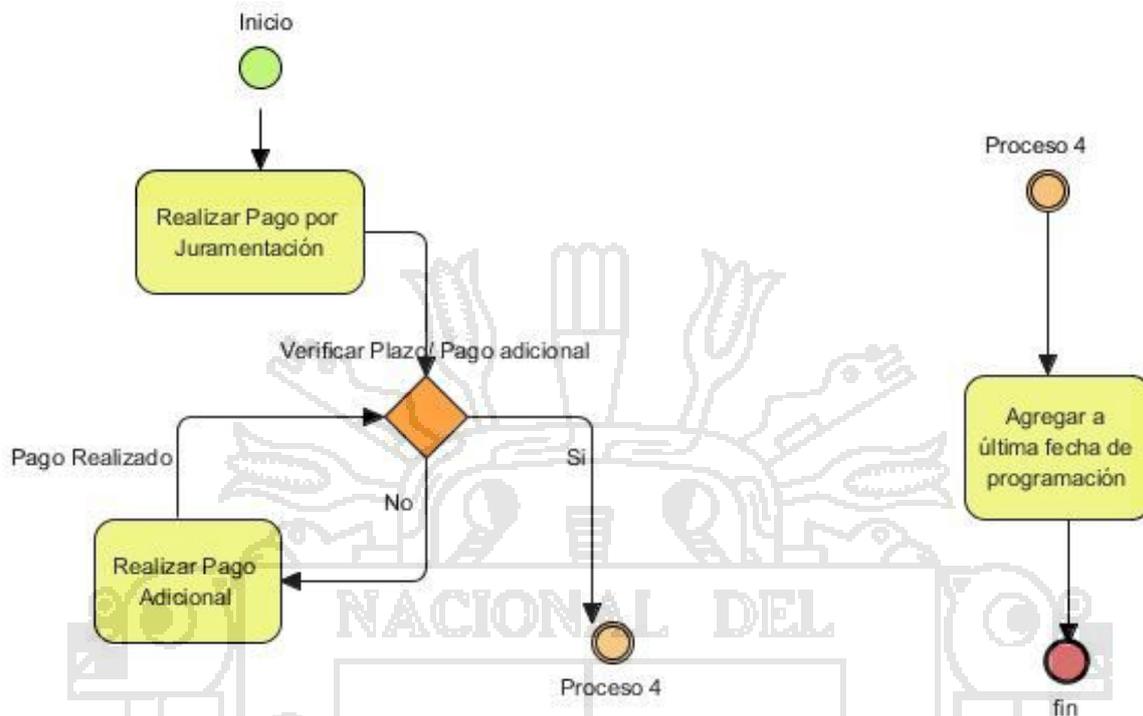
Diagrama de actividad de la programación de colegiatura.



Fuente: Elaboración propia.

Figura 35

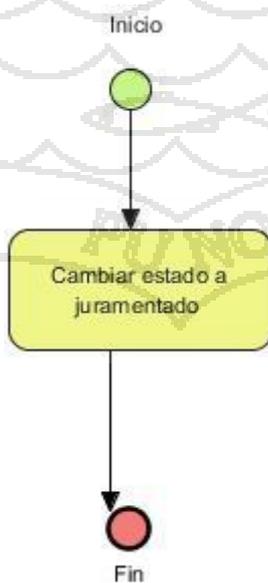
Diagramas de actividades del pago por juramentación y asignación de fecha de colegiatura.



Fuente: Elaboración propia.

Figura 36

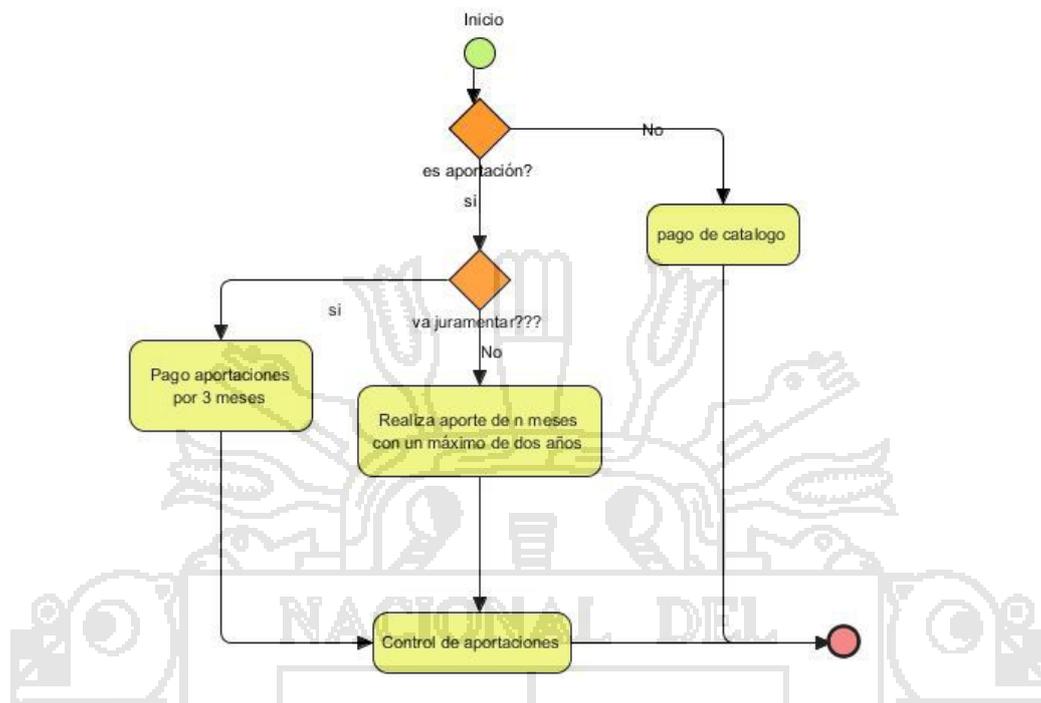
Diagrama de actividad del cambio de estado a juramentado
(Después de la ceremonia de colegiatura).



Fuente: Elaboración propia.

Figura 37

Diagrama de actividad del proceso de pago de aportaciones u otro servicio.



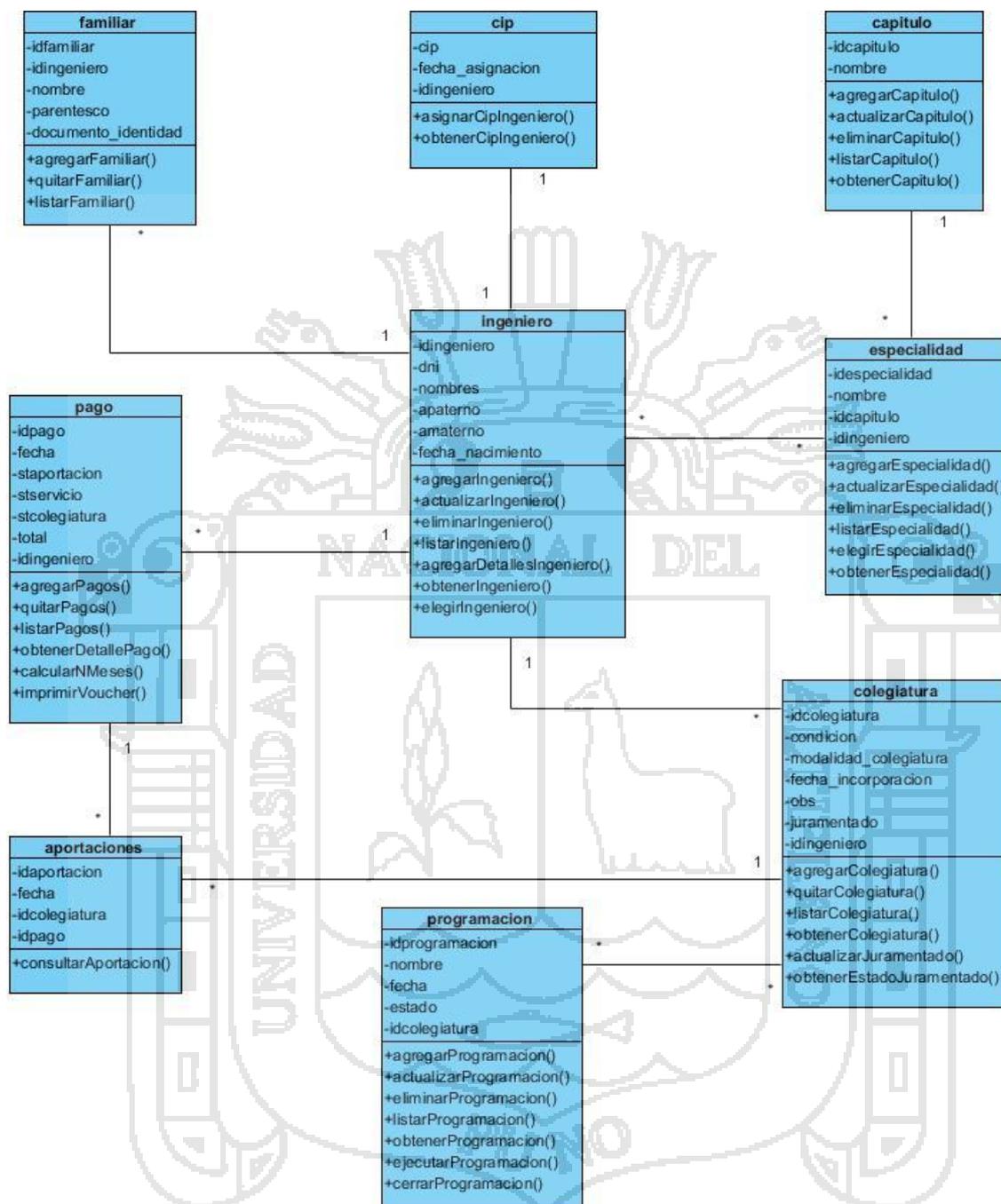
Fuente: Elaboración propia.

4.1.1.6. Diagrama de Clases

El siguiente diagrama de clases analizado describe la estructura del sistema desde el punto de vista de clases y objetos. A continuación se muestra las clases que se han identificado y las abstracciones genéricas de sus atributos y comportamiento del conjunto de objetos.

Figura 38

Diagrama de clases del sistema Web.



Fuente: Elaboración propia.

4.1.2. DISEÑO DEL SISTEMA WEB

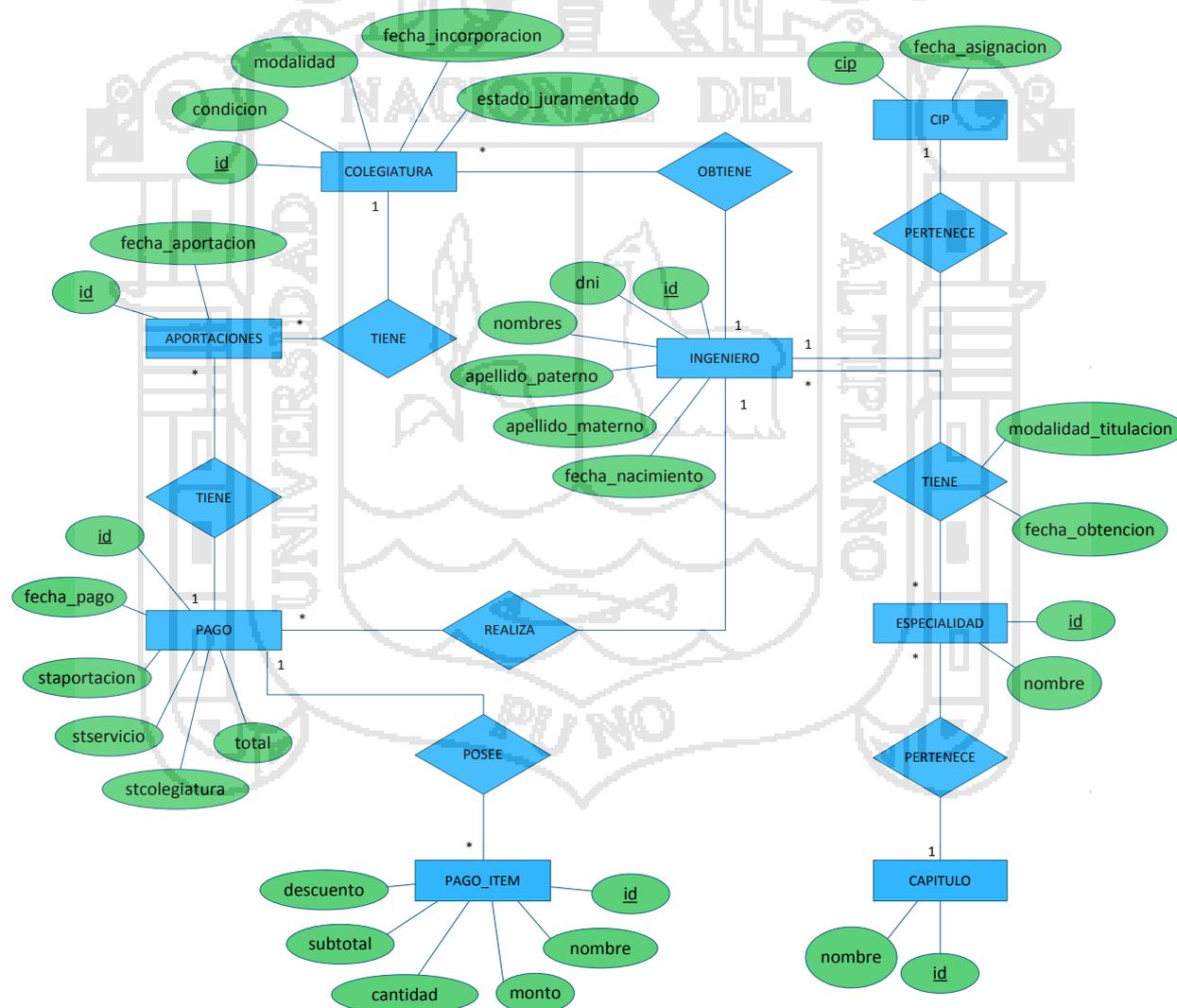
En este espacio se muestra el modelo conceptual (Esquema Entidad-Relación), lógico (Esquema Relacional) y físico de la base de datos.

4.1.2.1. Modelo Conceptual E-R

El modelo conceptual fue el punto de partida para el diseño de la base de datos, donde la primera representación está basada en el modelo Entidad-Relación (Atributos, relaciones e interrelaciones entre objetos).

Figura 39

Diagrama Entidad – Relación del sistema Web.



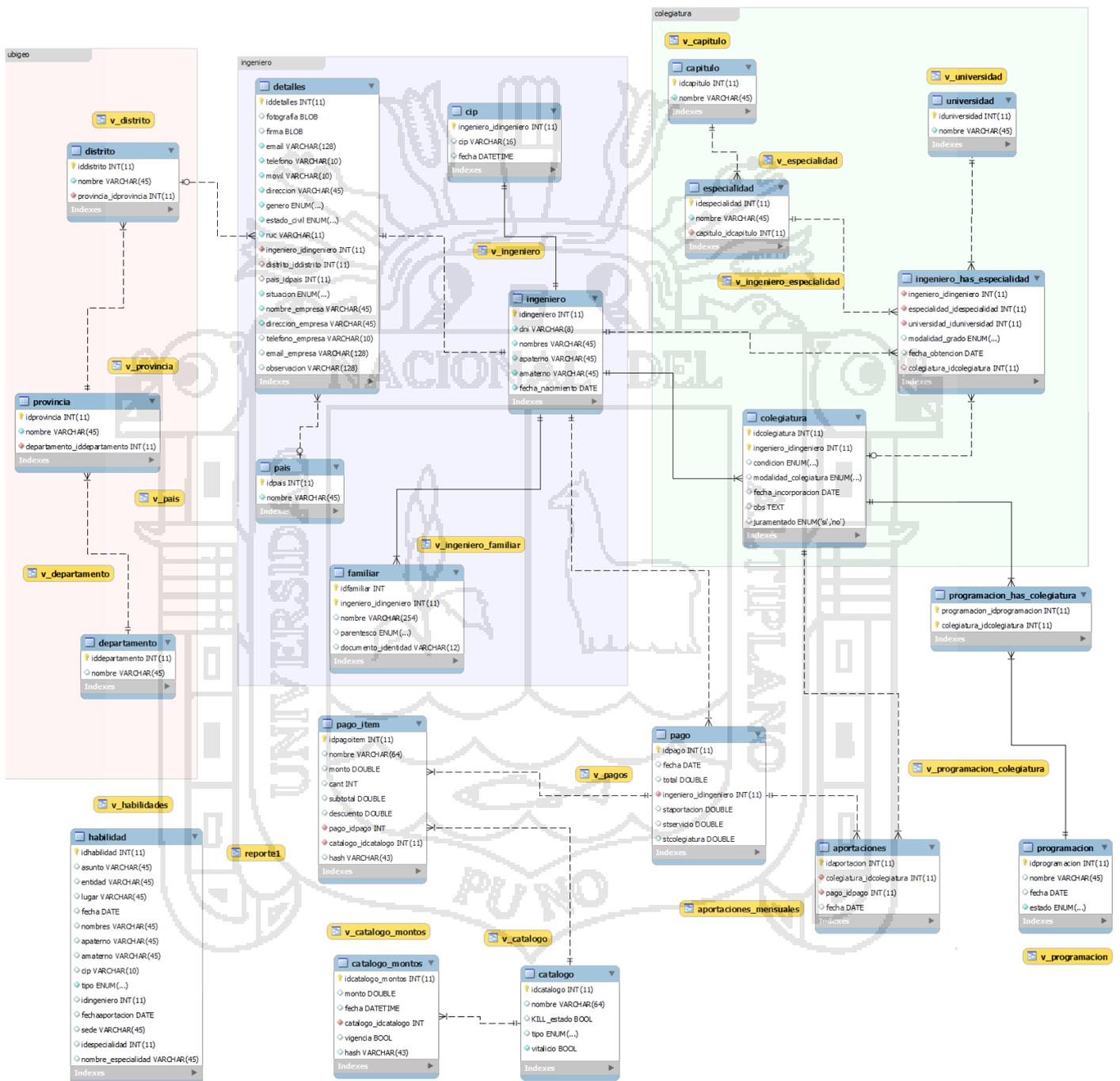
Fuente: Elaboración propia.

4.1.2.2. Modelo Lógico

El modelo lógico de la base de datos es el siguiente:

Figura 40

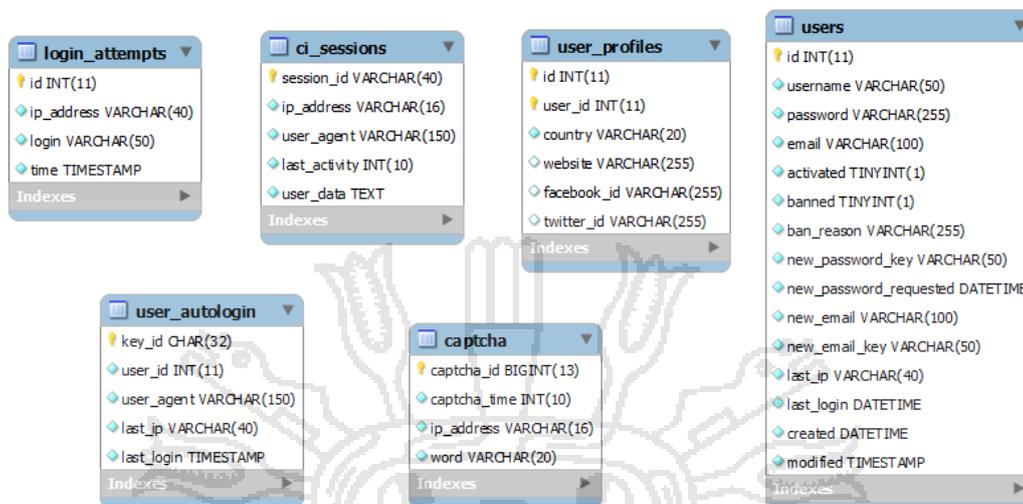
Modelo lógico de la base de datos del sistema Web (Parte I).



Fuente: Elaboración propia.

Figura 41

Modelo lógico de la base de datos del sistema Web (Parte II).



Fuente: Elaboración propia.

4.1.2.3. Modelo Físico

Para esta investigación se ha modelado una base de datos relacional la cual está compuesta por 27 tablas (Figura 40 y Figura 41) que almacenan información de los datos de los ingenieros, de las aportaciones que éstos realizan, y datos de los usuarios que manipularán el sistema.

En el anexo B se puede ver el esquema físico de la base de datos (Código fuente).

- **Tamaño de la BD:** El tamaño de la BD no depende de un dimensionamiento fijo que se realice al momento del diseño, sino del motor de BD que se utilice y de la capacidad de almacenamiento en disco y de procesamiento que se tenga en el servidor de producción.
- **Los límites en el número de bases de datos y tablas:** MySQL no tiene límite en el número de bases de datos. Sin embargo, el sistema de archivos puede tener un límite en el número de directorios. Motores de almacenamiento individuales pueden imponer limitaciones específicas del motor. InnoDB (que es el motor que utilizamos) permite hasta 4 mil millones de tablas.

- **Límites de tamaño de tabla:** El máximo tamaño efectivo de la tabla de bases de datos MySQL suele estar determinado por las limitaciones del sistema operativo sobre el tamaño de los archivos, y no por los límites internos de MySQL. En la siguiente tabla se muestran algunos ejemplos de límites de tamaño de ficheros del sistema operativo.

Tabla 4
Límites de tamaño de ficheros del S.O.

Sistema Operativo	Límite del tamaño del archivo
Win32 w / FAT/FAT32	2GB/4GB
Win32 w / NTFS	2TB (posiblemente más grande)
Linux 2.2 Intel de 32 bits	2GB (LFS: 4GB)
Linux 2.4 +	(Con sistema de archivos ext3) 4 TB
Solaris 9/10	16 TB
MacOS X w / HFS +	2TB
NetWare w / sistema de archivos NSS	8 TB

Fuente: Elaboración propia.

Para entornos Windows se debe tener en cuenta que FAT y VFAT (FAT32) no son aptos para uso en producción de MySQL. Se debe usar NTFS en su lugar.

En Linux 2.2, usando MyISAM se puede obtener tablas de más de 2 GB de tamaño usando el Soporte de archivos grandes (LFS) para el sistema de archivos ext2. En Linux 2.4, también existen parches para RaiserFS y tener soporte para archivos más grandes (hasta 2TB). Con JFS y XFS, petabyte y archivos aún más grandes son posibles en Linux.

La razón por la que se determinó el uso de InnoDB en lugar de MyISAM se debe al control de consistencia e integridad referencial que InnoDB ofrece.

- **Diccionario de Datos:** El diccionario de datos especifica con mayor detalle las tablas correspondientes a la base de datos, y se observan en las siguientes tablas:

Tabla 5

Tabla ingeniero.

Nombre	ingeniero	
Descripción	Contiene la información de los principales datos personales del ingeniero.	
Clave Primaria	Idingeniero	
Campo	Tipo	Descripción
Idingeniero	int(11)	Id del ingeniero
Dni	VARCHAR(8)	DNI del ingeniero
Nombres	VARCHAR(45)	Nombres del ingeniero
Apaterno	VARCHAR(45)	Apellido paterno del ingeniero
Amaterno	VARCHAR(45)	Apellido materno del ingeniero
fecha_nacimiento	DATE	Fecha de nacimiento del ingeniero

Fuente: Elaboración propia.

Tabla 6

Tabla detalles.

Nombre	Detalles	
Descripción	Contiene la información de datos secundarios del ingeniero.	
Clave Primaria	Iddetalles	
Clave Foránea	Idingeniero, iddistrito, idpais	
Campo	Tipo	Descripción
Iddetalles	int(11)	Id de detalles
Fotografía	BLOB	Imagen de la foto del ingeniero
Firma	BLOB	Firma del ingeniero
Email	VARCHAR(128)	Email personal del ingeniero
Teléfono	VARCHAR(10)	Número de teléfono del ingeniero
Móvil	VARCHAR(10)	Número de celular del ingeniero
Dirección	VARCHAR(45)	Dirección actual del ingeniero
Genero	ENUM('m','f')	Género del ingeniero
estado_civil	ENUM('s','c','d','v')	Estado civil del ingeniero
Ruc	VARCHAR(11)	Número RUC del ingeniero
Situación	ENUM('v','f')	Situación de vivo o fallecido del ingeniero
nombre_empresa	VARCHAR(45)	Nombre del centro de trabajo del ingeniero
direccion_empresa	VARCHAR(45)	Dirección del centro de trabajo del ingeniero
telefono_empresa	VARCHAR(10)	Número de teléfono del centro de trabajo del ingeniero
email_empresa	VARCHAR(128)	Email del centro de trabajo del ingeniero
Observación	VARCHAR(128)	Observaciones adicionales
Idingeniero	int(11)	Referencia tabla: ingeniero
Iddistrito	int(11)	Referencia tabla: distrito
Idpais	int(11)	Referencia tabla: país

Fuente: Elaboración propia.

Tabla 7
Tabla cip.

Nombre	Cip	
Descripción	Contiene la información del registro del Código CIP del ingeniero.	
Clave Primaria	Idingeniero	
Clave Foránea	Idingeniero	
Campo	Tipo	Descripción
Cip	VARCHAR(16)	Código CIP del ingeniero
Fecha	DATETIME	Fecha de asignación del CIP
Idingeniero	int(11)	Referencia tabla: ingeniero

Fuente: Elaboración propia.

Tabla 8
Tabla familiar.

Nombre	Familiar	
Descripción	Contiene la información de los datos de los familiares asignados por parte del ingeniero.	
Clave Primaria	Idfamiliar	
Clave Foránea	Idingeniero	
Campo	Tipo	Descripción
Idfamiliar	int(11)	Id del familiar
Nombre	VARCHAR(254)	Nombre del familiar del ingeniero
Parentesco	ENUM('p','h','e')	Parentesco del familiar con el ingeniero
documento_identidad	VARCHAR(12)	DNI del familiar del ingeniero
Idingeniero	int(11)	Referencia tabla: ingeniero

Fuente: Elaboración propia.

Tabla 9
Tabla país.

Nombre	país	
Descripción	Contiene la información de los países.	
Clave Primaria	Idpais	
Clave Foránea	Idingeniero	
Campo	Tipo	Descripción
Idpais	int(11)	Id del país
Nombre	VARCHAR(45)	Nombre del país
Idingeniero	int(11)	Referencia tabla: ingeniero

Fuente: Elaboración propia.

Tabla 10

Tabla departamento.

Nombre	departamento	
Descripción	Contiene la información de los departamentos del país.	
Clave Primaria	Iddepartamento	
Campo	Tipo	Descripción
iddepartamento	int(11)	Id del departamento
Nombre	VARCHAR(45)	Nombre del departamento

Fuente: Elaboración propia.

Tabla 11

Tabla provincia.

Nombre	provincia	
Descripción	Contiene la información de las provincias del país.	
Clave Primaria	Idprovincia	
Clave Foránea	Iddepartamento	
Campo	Tipo	Descripción
Idprovincia	int(11)	Id de la provincia
Nombre	VARCHAR(45)	Nombre de la provincia
iddepartamento	int(11)	Referencia tabla: departamento

Fuente: Elaboración propia.

Tabla 12

Tabla distrito.

Nombre	distrito	
Descripción	Contiene la información de los distritos del país.	
Clave Primaria	Iddistrito	
Clave Foránea	Idprovincia	
Campo	Tipo	Descripción
Iddistrito	int(11)	Id del distrito
Nombre	VARCHAR(45)	Nombre del distrito
idprovincia	int(11)	Referencia tabla: provincia

Fuente: Elaboración propia.

Tabla 13

Tabla capítulo.

Nombre	capítulo	
Descripción	Contiene la información de los diversos capítulos.	
Clave Primaria	Idcapitulo	
Campo	Tipo	Descripción
Idcapítulo	int(11)	Id del capítulo
Nombre	VARCHAR(45)	Nombre del capítulo

Fuente: Elaboración propia.

Tabla 14

Tabla especialidad.

Nombre	especialidad	
Descripción	Contiene la información de las diversas especialidades.	
Clave Primaria	Idespecialidad	
Clave Foránea	Idcapitulo	
Campo	Tipo	Descripción
idespecialidad	int(11)	Id de la especialidad
Nombre	VARCHAR(45)	Nombre de la especialidad
Idcapítulo	int(11)	Referencia tabla: capitulo

Fuente: Elaboración propia.

Tabla 15

Tabla universidad.

Nombre	universidad	
Descripción	Contiene la información de los diferentes universidades del país.	
Clave Primaria	Iduniversidad	
Campo	Tipo	Descripción
Iduniversidad	int(11)	Id de la universidad
Nombre	VARCHAR(45)	Nombre de la universidad

Fuente: Elaboración propia.

Tabla 16

Tabla ingeniero_has_especialidad.

Nombre	ingeniero_has_especialidad	
Descripción	Contiene la información de las especialidades que poseen los ingenieros.	
Clave Foránea	Idingeniero Idespecialidad iduniversidad idcolegiatura	
Campo	Tipo	Descripción
modalidad_grado	int(11)	Modalidad de la obtención del título del ingeniero
fecha_obtencion	VARCHAR(45)	Fecha de la obtención del título del ingeniero
Idingeniero	int(11)	Referencia tabla: ingeniero
Idespecialidad	int(11)	Referencia tabla: especialidad
Iduniversidad	int(11)	Referencia tabla: universidad
Idcolegiatura	int(11)	Referencia tabla: provincia

Fuente: Elaboración propia.

Tabla 17

Tabla colegiatura.

Nombre	Colegiatura	
Descripción	Contiene la información de los datos de la colegiatura del ingeniero.	
Clave Primaria	Idcolegiatura	
Clave Foránea	Idingeniero	
Campo	Tipo	Descripción
Idcolegiatura	int(11)	Id del distrito
Condición	ENUM('t', 'o', 'v', 't')	Condición del ingeniero colegiado
modalidad_colegiatura	ENUM('e', 't')	Modalidad de colegiatura
fecha_incorporacion	DATE	Fecha de incorporación del ingeniero
Obs	TEXT	Observaciones adicionales
Juramentado	ENUM('si', 'no')	Estado de juramentación del ingeniero
Idingeniero	int(11)	Referencia tabla: ingeniero

Fuente: Elaboración propia.

Tabla 18

Tabla programación.

Nombre	Programación	
Descripción	Contiene la información de las programaciones de acto de juramentación de los ingenieros.	
Clave Primaria	Idprogramacion	
Campo	Tipo	Descripción
idprogramacion	int(11)	Id de programación
Nombre	VARCHAR(45)	Nombre de la programación
Fecha	DATE	Fecha de la juramentación
Estado	ENUM('c','e','c')	Estado de la programación

Fuente: Elaboración propia.

Tabla 19

Tabla programación_has_colegiatura.

Nombre	programacion_has_colegiatura	
Descripción	Contiene la información de los ingenieros que juramentarán en cada programación de colegiatura.	
Clave Foránea	Idprogramacion Idcolegiatura	
Campo	Tipo	Descripción
idprogramacion	int(11)	Referencia tabla: programación
idcolegiatura	int(11)	Referencia tabla: colegiatura

Fuente: Elaboración propia.

Tabla 20

Tabla aportaciones.

Nombre	Aportaciones	
Descripción	Contiene la información de las aportaciones que realizan los ingenieros.	
Clave Primaria	Idaportacion	
Clave Foránea	Idcolegiatura Idpago	
Campo	Tipo	Descripción
Idaportacion	int(11)	Id de la aportación
Fecha	DATE	Fecha de la aportación
Idcolegiatura	int(11)	Referencia tabla: colegiatura
Idpago	int(11)	Referencia tabla: pago

Fuente: Elaboración propia.

Tabla 21

Tabla pago.

Nombre	pago	
Descripción	Contiene la información de los pagos que realizan los ingenieros.	
Clave Primaria	Idpago	
Clave Foránea	Idingeniero	
Campo	Tipo	Descripción
Idpago	int(11)	Id del pago
Fecha	DATE	Fecha del pago
Total	DOUBLE	Total del pago realizado
Staportacion	DOUBLE	Subtotal del pago de aportaciones
Stservicio	DOUBLE	Subtotal del pago de servicios
Stcolegiatura	DOUBLE	Subtotal del pago de colegiatura
Idingeniero	int(11)	Referencia tabla: ingeniero

Fuente: Elaboración propia.

Tabla 22

Tabla pago_item.

Nombre	pago_item	
Descripción	Contiene la información de los conceptos de pago.	
Clave Primaria	Idpagoitem	
Clave Foránea	Idpago Idcatalogo	
Campo	Tipo	Descripción
idpagoitem	int(11)	Id del pago ítem
Nombre	VARCHAR(64)	Nombre del concepto de pago
Monto	DOUBLE	Monto de cada concepto de pago
Cant	int(3)	Cantidad de cada concepto de pago
subtotal	DOUBLE	Subtotal de cada ítem
descuento	DOUBLE	Descuento de un concepto de pago
Idpago	int(11)	Referencia tabla: pago
idcatalogo	int(11)	Referencia tabla: catalogo

Fuente: Elaboración propia.

Tabla 23

Tabla catalogo.

Nombre	Catalogo	
Descripción	Contiene la información de los servicios por los que hace pago el ingeniero.	
Clave Primaria	Idcatalogo	
Campo	Tipo	Descripción
Idcatalogo	int(11)	Id del catálogo
Nombre	VARCHAR(64)	Nombre del servicio
Tipo	ENUM('a','c','s')	Tipo del servicio
Vitalicio	BOOL	Flag para el descuento de vitalicio

Fuente: Elaboración propia.

Tabla 24

Tabla catalogo_montos.

Nombre	catalogo_montos	
Descripción	Contiene la información de los montos de los servicios.	
Clave Primaria	idcatalogo_montos	
Clave Foránea	Idcatalogo	
Campo	Tipo	Descripción
idcatalogo_montos	int(11)	Id de los montos del catálogo
Monto	DOUBLE	Monto de los servicios
Fecha	DATETIME	Fecha de creación del monto del catálogo
Vigencia	BOOL	Vigencia del monto del servicio
Idcatalogo	int(11)	Referencia tabla: catalogo

Fuente: Elaboración propia.

4.2. IMPLEMENTACIÓN DEL SISTEMA WEB

El desarrollo de la presente aplicación se basó en un esquema modular. El componente principal es el framework de desarrollo CodeIgniter, el cual permitió implementar el sistema bajo el patrón MVC descrito en el Capítulo II. Se utilizaron también módulos y componentes para la autenticación, diseño de interfaz del lado cliente así como herramientas ágiles de desarrollo.

4.2.1. FRAMEWORK UTILIZADO

El framework de desarrollo fue el CodeIgniter en su versión 2.1.3, así que se utilizó a éste como el core del sistema desarrollado. Esta herramienta permitió diseñar la aplicación bajo un esquema MVC (Modelo – Vista – Controlador) muy utilizado para la producción de sistemas.

Las clases utilizadas para cada uno de los elementos del MVC fueron:

CI_Controller: Un controlador es simplemente un archivo de clase que se denomina de manera que se puede asociar con un URI. Las clases heredadas de CI_Controller incluyen funciones que son utilizadas como rutas de ejecución desde el lado cliente que invocan un determinado módulo. Estas funciones son los encargados de coordinar el acceso a datos a través de los modelos y la consecuente generación de vistas de acuerdo a su ejecución.

CI_Model: La clase CI_Model es la encargada de realizar la conexión con la Base de Datos y gestionar la información para su procesamiento. Los modelos son clases PHP que están diseñados para trabajar con la información en la base de datos.

4.2.2. COMPONENTES UTILIZADOS

- *Autenticación Tank Auth*

Tank Auth es una librería de autenticación para el framework CodeIgniter de PHP. Está basada en DX Auth.

Su arquitectura tiene opciones para logeo, cierre de sesión, registro y desregistro. De ellos únicamente el logeo ha sido utilizado en el presente desarrollo.

La base de datos utiliza 4 tablas que han sido incluidas en el esquema de base de datos general. Las contraseñas son almacenadas de manera cifrada (MD5). Asimismo se tiene un mecanismo de conteo de intentos de logeo en base a IP y nombre de usuario.

- **Charisma**

Charisma es una plantilla con características que le permiten adaptarse a cualquier dispositivo (está optimizada para tablets y dispositivos celulares) que se utiliza para paneles de administración

Está basado en bootstrap y JQuery UI que incluye elementos válidos de HTML y CSS3.

- **Bootstrap y otros componentes cliente**

Para el lado cliente, se utilizaron varios componentes de desarrollo ágil: Bootstrap 2.0, JQuery y JQuery UI.

4.2.3. INTERFACES PRINCIPALES DEL SISTEMA WEB

Para el diseño de interfaces se utilizó Bootstrap 2.0 y una plantilla de distribución libre denominada Charisma, la misma que cuenta con diseño adaptativo (responsive design), es decir se adapta a cualquier dispositivo, también basado en Bootstrap. Las principales interfaces del sistema Web son:

Interfaz de logeo:

El acceso al sistema Web se realiza a través de un explorador web, donde la página inicial el usuario registrado, tiene que autenticarse para iniciar sesión y así poder ingresar a la interfaz principal.

Figura 42

Interfaz de logeo.



Bienvenido a SISCIP

[Olvidé mi contraseña](#) / [Registrarme](#)

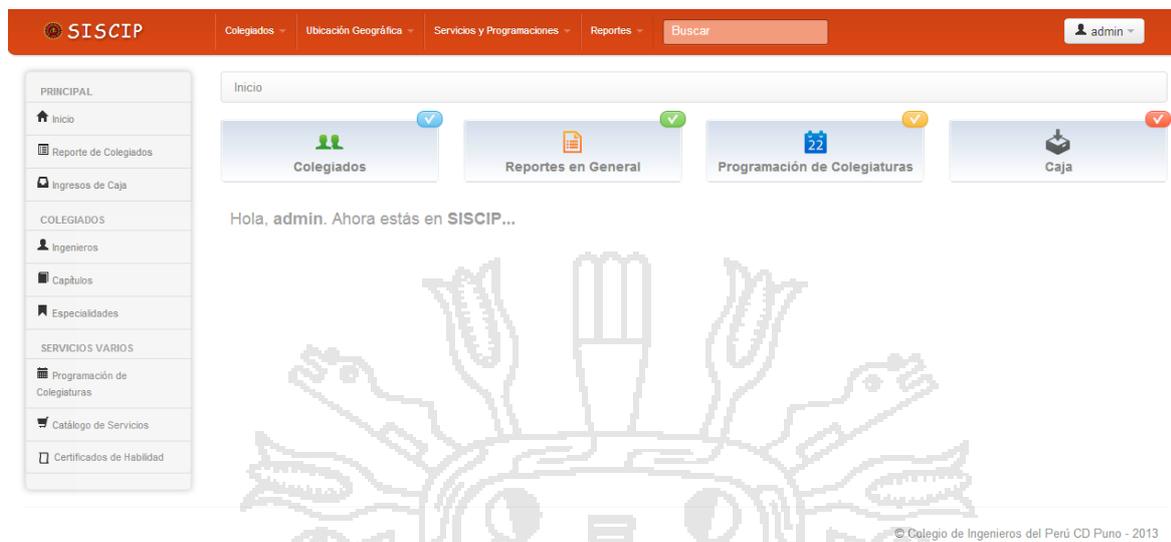
Fuente: Elaboración propia.

Interfaz de inicio del sistema Web:

Una vez realizada la identificación se accede a la interfaz principal del sistema Web, la cual muestra las principales funciones que realiza el usuario (Figura 42), y se presentan los accesos directos a las secciones principales del sistema Web (Colegiados, Reportes en General, Programación de Colegiaturas y Caja). También, se puede observar el menú superior, donde se distinguen las siguientes secciones:

- Sección **COLEGIADOS**, donde se registran los datos personales de los ingenieros, especialidades, capítulos y centros de estudios.
- Sección **UBICACIÓN GEOGRÁFICA**, donde se registran el país, departamento, provincia y distritos nuevos o desconocidos, según sea el caso que se requiera.
- Sección de **SERVICIOS VARIOS**, donde se registran los servicios del catálogo, la programación de colegiaturas, y también, se realiza la emisión de certificados de habilidad solicitados por los ingenieros.
- Sección de **REPORTES DE COLEGIADOS**, donde se obtienen el reporte de colegiados (Ingenieros hábiles a la fecha solicitada) y el reporte del pago por concepto de ISS y CN (Detalle de aportaciones realizado por los ingenieros), este último es enviado al CIP Consejo Nacional.
- Finalmente la sección de **INGRESOS DE CAJA**, donde se obtienen los ingresos diarios y/o mensuales, del pago de aportaciones y/o pago por los diferentes servicios solicitados por los colegiados.

Figura 43
Interfaz de inicio del sistema Web.



Fuente: Elaboración propia.

Interfaz con las opciones generales para el ingeniero:

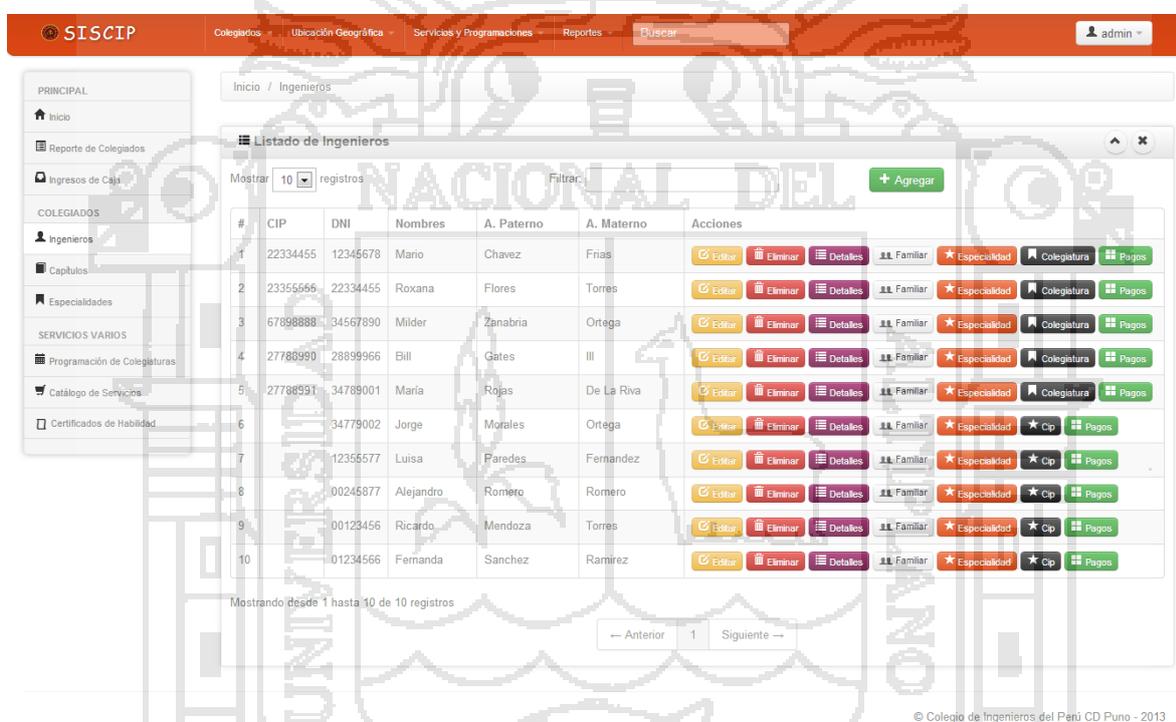
En esta interfaz se observa un DataTable con las diferentes opciones de registro que puede realizarse respecto de un ingeniero, entre ellas se encuentran:

- **DETALLES**, para el registro de otros datos del ingeniero, como situación civil, e-mail, residencia, centro laboral, entre otros.
- **FAMILIAR**, para el registro de los datos de los familiares de cada ingeniero (Herederos que reciben el beneficio correspondiente en caso de fallecimiento del colegiado).
- **ESPECIALIDAD**, para el registro de especialidades que tenga un ingeniero (Puede registrarse más de una especialidad).
- **CIP**, para el registro de código CIP asignado al ingeniero (El CIP es único y solo se asigna una vez. No se asigna el CIP por especialidad, sino por ingeniero).

- **COLEGIATURA**, para el registro de los datos de la colegiatura del ingeniero, tales como: Fecha de incorporación, modalidad de colegiatura, condición y observaciones (Si existiera).
- **PAGOS**, en esta opción se registran los pagos que debe y/o puede realizar el ingeniero, los mismos que se encuentran distribuidos en tres grupos: Aportación, Servicio y Colegiatura.

Figura 44

Interfaz con las opciones generales para el ingeniero.



#	CIP	DNI	Nombres	A. Paterno	A. Materno	Acciones
1	22334455	12345678	Mario Chavez	Frias		Editar Eliminar Detalles Familiar Especialidad Colegiatura Pagos
2	23355666	22334455	Roxana Flores	Torres		Editar Eliminar Detalles Familiar Especialidad Colegiatura Pagos
3	67898888	34567890	Milder Zenabria	Ortega		Editar Eliminar Detalles Familiar Especialidad Colegiatura Pagos
4	27788990	28899966	Bill Gates	Ill		Editar Eliminar Detalles Familiar Especialidad Colegiatura Pagos
5	27788991	34789001	Maria Rojas	De La Riva		Editar Eliminar Detalles Familiar Especialidad Colegiatura Pagos
6		34779002	Jorge Morales	Ortega		Editar Eliminar Detalles Familiar Especialidad Cip Pagos
7		12355577	Luisa Paredes	Fernandez		Editar Eliminar Detalles Familiar Especialidad Cip Pagos
8		00245877	Alejandro Romero	Romero		Editar Eliminar Detalles Familiar Especialidad Cip Pagos
9		00123456	Ricardo Mendoza	Torres		Editar Eliminar Detalles Familiar Especialidad Cip Pagos
10		01234566	Fernanda Sanchez	Ramirez		Editar Eliminar Detalles Familiar Especialidad Cip Pagos

Mostrando desde 1 hasta 10 de 10 registros

← Anterior 1 Siguiente →

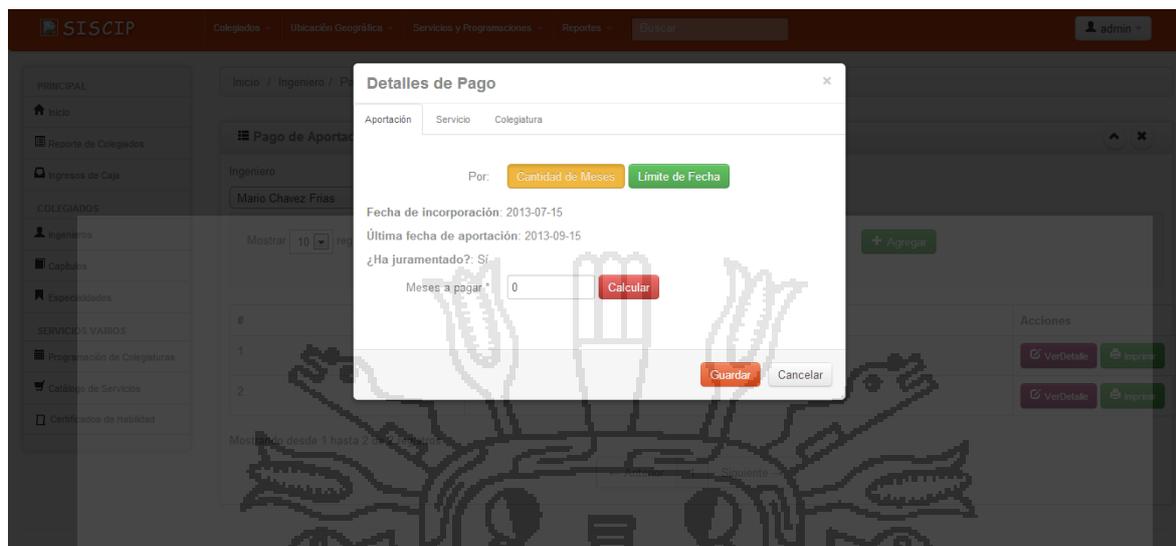
© Colegio de Ingenieros del Perú CD Puno - 2013

Fuente: Elaboración propia.

Interfaz del detalle de pagos (Aportación):

En esta interfaz (Figura 45) se observa el detalle de pago de aportaciones que realiza el ingeniero. Es importante señalar que éste pago solo se realiza cuando el ingeniero ha sido programado para la ceremonia de juramentación y ya ha pasado del estado de trámite a ordinario.

Figura 45
Interfaz del detalle de pagos (Aportación).



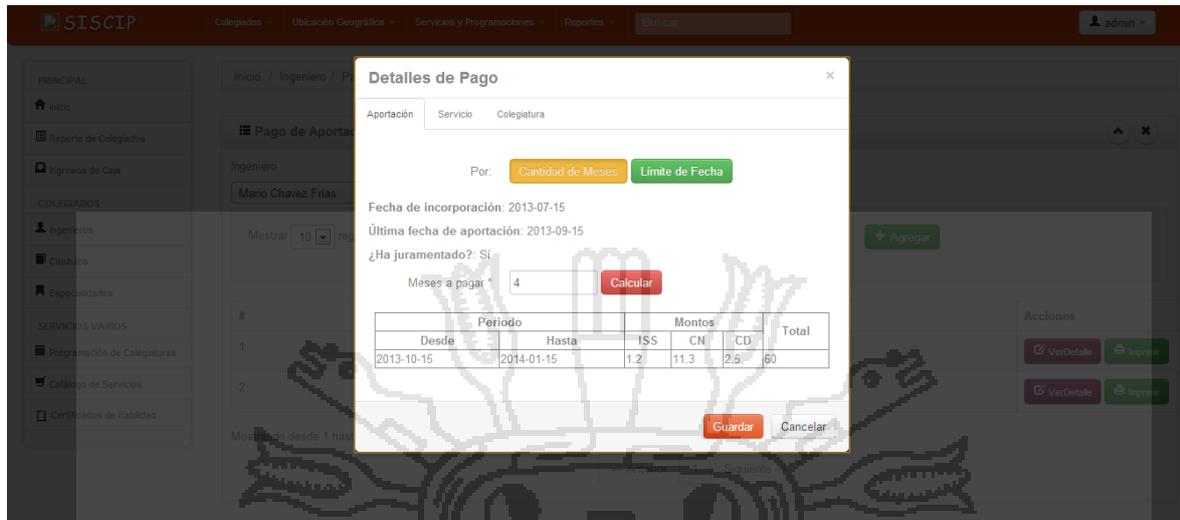
Fuente: Elaboración propia.

Interfaz del detalle de pagos (Detalle de Aportación):

Esta interfaz (Figura 46) pertenece al detalle de pago de *aportación*, pero una vez hecho el cálculo de la aportación (Según el número de meses a aportar o hasta la fecha de aporte deseada). Aquí se observan los detalles del pago por ISS (Seguro Social), CN (Consejo Nacional) y CD (Consejo Departamental), el intervalo de fecha del pago respectivo y el total del pago de aportaciones.

Figura 46

Interfaz del detalle de pagos (Detalle de Aportación).



Fuente: Elaboración propia.

Interfaz del detalle de pagos (Servicio):

Esta interfaz (Figura 47) pertenece al detalle de pago de *servicio*, en la cual se detalla el pago por los conceptos de pago y sus montos respectivos, tales como: Certificados de habilidad, duplicado de carné, constancia de no adeudo, entre otros.

Figura 47

Interfaz del detalle de pagos (Servicio).



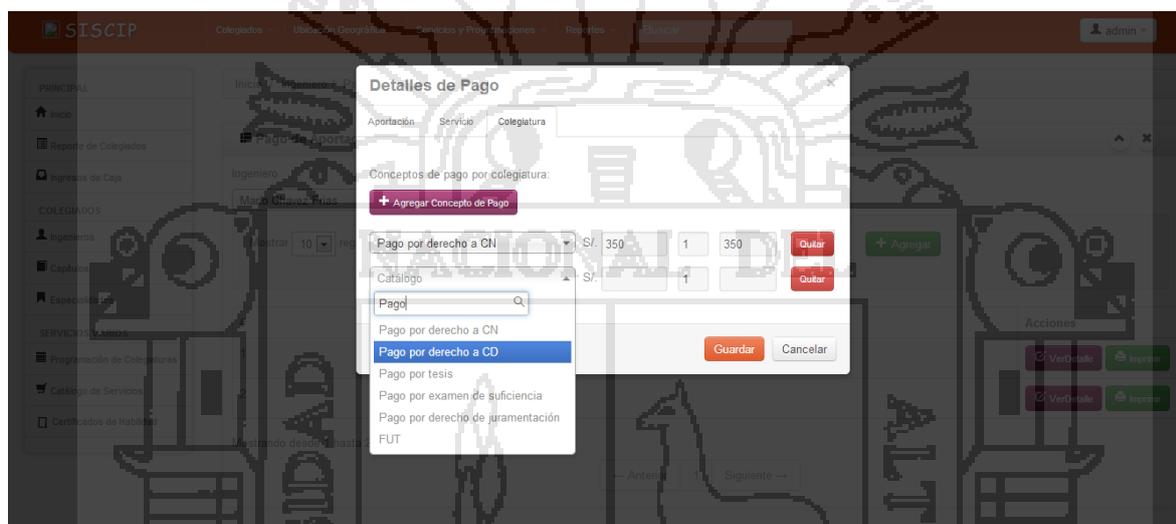
Fuente: Elaboración propia.

Interfaz del detalle de pagos (Colegiatura):

Esta interfaz (Figura 48) pertenece al detalle de pago de *colegiatura*, en la cual se contemplan los pagos por: derecho a CN, derecho a CD, modalidad de obtención de colegiatura (por tesis o por examen de suficiencia) y el pago por FUT (Formulario único de trámite de colegiatura).

Figura 48

Interfaz del detalle de pagos (Colegiatura).



Fuente: Elaboración propia.

Interfaz de impresión de voucher de pago:

Esta interfaz (Figura 49) pertenece a la impresión del voucher de pago, el cual contiene los datos del ingeniero, como: nombres y apellidos, código CIP, dirección, el detalle del pago realizado (Descripción detallada de los conceptos y tipos de pago, cantidades y subtotales), y el monto total del pago realizado.

Figura 49
Impresión de voucher de pago.


COLEGIO DE INGENIEROS DEL PERÚ
 Consejo Departamental Puno
J. Mariano H. Cornejo N° 123 - Teléfono: (051) 3200050

Recibo Nro.
 1 - 0000003

Señor (es) : Mario Chavez Frías CIP Nro. 22334455
 Dirección :

Aportaciones:

Periodo		Montos			SubTotal
Desde	Hasta	ISS	CN	CD	
2013-11-15	2014-02-15	1.2	11.3	2.5	60

Servicios y Pagos por colegiatura:

CANT.	DESCRIPCIÓN	P. UNIT.	SUBTOTAL
1	Certificado de Habilidad Especifico	10	10
1	Pago por derecho a CN	350	350
Total :			420



Fuente: Elaboración propia.

Interfaz para la consulta de aportaciones en línea:

Esta interfaz (Figura 50) pertenece a la consulta de aportaciones en línea que pueden realizar los ingenieros colegiados. Para conocer su estado de aportaciones, el colegiado debe ingresar su código CIP, su DNI y un identificador CAPTCHA en el formulario de consulta, e inmediatamente después, se enviará la información del estado de sus aportaciones a su correo electrónico (Registrado en la base de datos del sistema Web).

Figura 50

Interfaz para la consulta de aportaciones en línea



Colegio de Ingenieros - CD Puno

Consulta en línea

CIP

DNI

Captcha

Enviar

Los datos de su estado serán enviados a su dirección de correo electrónico.

Colegio de Ingenieros del Perú
Consejo Departamental Puno

Fuente: Elaboración propia.

4.3. PRUEBA Y VALIDACIÓN DEL SISTEMA WEB UTILIZANDO EL MODELO DE CALIDAD DE LA NORMA ISO/IEC 25010:2010 (SQuaRE)

4.3.1. PRUEBA DEL SISTEMA WEB

Para conocer la situación actual sobre la Gestión de Información en el Colegio de Ingenieros del Perú CD Puno se realizó un par de encuestas uno antes de la demostración del modelo (Pre-test) y otro después de la demostración de modelo (Post-test).

Los datos se recopilaron con el fin de obtener un patrón de mejora, a través de un proceso de selección, comparación y normalización de datos (Anexo C, Encuesta).

“Los datos deben resumirse, codificarse y repararse para el análisis” según (Hernández Sampieri & Batista, 2006, pág. 278).

Para el análisis de la encuesta más allá de un conteo de casos, para lo cual es necesario transformar las respuestas en símbolos o valores numéricos. En la tabla 25 se muestra la codificación de las alternativas de la encuesta (Ver anexo C).

Tabla 25

Presentación del calificativo para las alternativas de la encuesta.

CALIFICATIVO	VALOR
Muy Bueno	5
Bueno	4
Regular	3
Mala	2
Muy Mala	1

Fuente: Elaboración propia.

En la tabla 26 se muestra el intervalo de clase agrupando posibles puntajes de los resultados de la encuesta realizada a los usuarios.

Tabla 26

Intervalo de clase para los puntajes obtenidos de la encuesta.

Intervalo	Calificativo
[0-9]	Muy mala
[10-19]	Mala
[20-29]	Regular
[30-38]	Bueno
[39-48]	Muy bueno

Fuente: Elaboración propia.

Tabla 27

Matriz de datos pre-test de la gestión de información en el CIP CD Puno.

	PREGUNTAS DE LA ENCUESTA													PUNTAJE TOTAL
	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	1	2	2	2	2	1	1	1	2	2	1	1	1	19
2	1	2	2	2	2	1	1	1	2	2	1	1	1	19
3	1	2	2	2	2	1	1	1	2	2	1	1	1	19
4	1	2	2	2	2	1	1	1	2	2	1	1	1	19
5	1	2	2	2	2	1	1	1	2	1	1	1	1	18
6	1	2	2	2	2	1	1	1	2	2	1	1	1	19
7	1	2	2	2	2	1	1	1	2	2	1	1	1	19
8	1	2	2	2	2	1	1	1	2	1	1	1	1	18
9	1	1	1	2	2	1	1	1	2	2	1	1	1	17



10	2	3	2	3	2	2	1	1	3	3	2	2	1	27
11	1	1	1	2	2	1	1	1	2	2	1	1	1	17
12	1	1	1	2	2	1	1	1	2	2	1	1	1	17
13	1	2	2	2	2	1	1	1	2	2	1	1	1	19
14	1	2	2	2	2	1	1	1	2	2	1	1	1	19
15	1	2	2	2	2	1	1	1	2	2	1	1	1	19
16	1	2	2	2	2	1	1	1	2	2	1	1	1	19
17	1	2	2	2	2	1	1	1	2	2	1	1	1	19
18	1	2	2	2	2	1	1	1	2	2	1	1	1	19
19	2	4	2	3	2	2	1	1	3	3	2	1	1	27
20	1	2	2	2	2	1	1	1	2	2	1	1	1	19
21	1	2	2	2	2	1	1	1	2	1	1	1	1	18
22	1	2	2	2	2	1	1	1	2	2	1	1	1	19
23	1	2	2	2	2	1	1	1	2	2	1	1	1	19
24	1	2	2	2	2	1	1	1	2	2	1	1	1	19
25	1	2	2	2	2	1	1	1	2	2	1	1	1	19
26	1	2	2	2	2	1	1	1	2	2	1	1	1	19
27	2	3	2	3	2	1	1	1	3	3	2	2	1	26
28	1	2	2	2	2	1	1	1	2	2	1	1	1	19
29	1	2	2	2	2	1	1	1	2	2	1	1	1	19
TOTAL														565

Fuente: Elaboración propia.

Tabla 28

Matriz de datos post-test de la gestión de información en el CIP CD Puno.

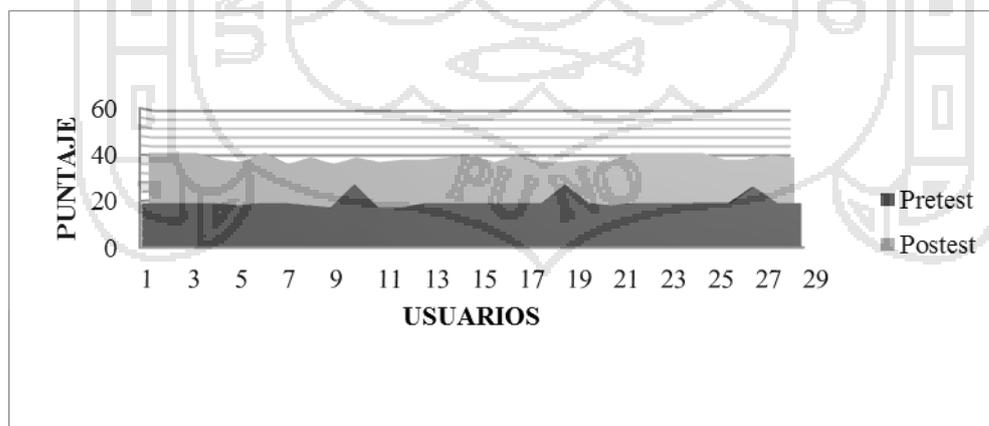
	PREGUNTAS DE ENCUESTA													PUNTAJE	
	1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL	
NUMERO DE PERSONAS OBSERVADAS	1	2	5	2	5	5	2	2	2	5	5	2	2	2	41
	2	2	5	2	5	5	2	2	2	5	5	2	2	2	41
	3	2	5	2	5	5	2	2	2	5	5	2	2	2	41
	4	2	4	2	5	4	2	2	2	5	4	2	2	2	38
	5	2	4	2	4	4	2	2	2	5	4	2	2	2	37
	6	2	5	2	5	5	2	2	2	5	5	2	2	2	41
	7	2	4	2	4	4	2	2	2	4	4	2	2	2	36
	8	2	5	2	4	4	2	2	2	5	5	2	2	2	39
	9	2	4	2	4	4	2	2	2	4	4	2	2	2	36
	10	2	5	2	5	5	2	2	2	5	4	2	1	2	39
	11	2	4	2	4	4	2	2	2	4	5	2	2	2	37

12	2	4	2	5	4	2	2	2	4	5	2	2	2	38
13	2	4	2	5	4	2	2	2	4	5	2	2	2	38
14	2	5	2	5	4	2	2	2	5	4	2	2	2	39
15	2	5	2	5	4	2	2	2	5	5	2	2	2	40
16	2	4	2	4	4	2	2	2	4	5	2	2	2	37
17	2	5	2	5	4	2	2	2	5	5	2	2	2	40
18	2	4	2	4	4	2	2	2	4	5	2	2	2	37
19	2	4	2	4	4	2	2	2	5	5	2	1	2	37
20	2	4	2	4	4	2	2	2	5	5	2	2	2	38
21	2	4	2	4	4	2	2	2	5	4	2	2	2	37
22	2	5	2	5	5	2	2	2	5	5	2	2	2	41
23	2	5	2	5	5	2	2	2	5	5	2	2	2	41
24	2	5	2	5	5	2	2	2	5	5	2	2	2	41
25	2	5	2	5	5	2	2	2	5	5	2	2	2	41
26	2	5	2	4	4	2	2	2	4	5	2	2	2	38
27	2	5	2	4	4	2	2	2	5	5	2	1	2	38
28	2	5	2	5	4	2	2	2	5	5	2	2	2	40
29	2	5	2	4	4	2	2	2	5	5	2	2	2	39
TOTAL														1126

Fuente: Elaboración propia.

Obteniéndose el siguiente gráfico de resultados.

Figura 51
Gráfico de comparación del pre-test y post-test.



Fuente: Elaboración propia.

Interpretación

La parte oscura representa la prueba *pre-test*, donde se aprecia que de los 29 usuarios, la mayoría desapueba la gestión de información existente, con un promedio de 19 estando en el calificativo de “*Mala*”.

Luego de la implementación del Sistema Web basado en la Norma ISO/IEC 25010:2010 SQuaRE para mejorar la gestión de información en el Colegio de Ingenieros del Perú CD Puno, se realizó un *post-test*, de lo cual se obtuvo un promedio de 39 estando en el calificativo de “*Muy bueno*”, observándose que la gestión de la información mejoró por la mayor fluidez de la información, de este modo se agilizan los procesos.

4.3.2. VALIDACIÓN DEL SISTEMA WEB UTILIZANDO EL MODELO DE CALIDAD DE LA NORMA ISO/IEC 25010:2010 (SQuaRE)

Para la validación del sistema se elaboró una encuesta (Tabla 29) en la que se plantean una serie de preguntas basadas y consideradas dentro del modelo de calidad de la norma ISO/IEC 25000 (SQuaRE). Es decir, se consideraron las características de calidad interna y externa, y calidad en uso con las que debe contar un producto software. La encuesta fue aplicada en el grupo experimental antes y después de la implementación del sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE), de esta manera se valida el sistema Web.

Tabla 29

Cuadro de preguntas planteado para la validación del sistema Web.

N°	DESCRIPCIÓN	ESCALA DE MEDICIÓN						
		Muy bueno	Bueno	Regular	Mala	Muy mala	SI	NO
		5	4	3	2	1	2	1
1	Desde su punto de vista, ¿el sistema Web en funcionamiento hace lo que le pide?							
2	Los tiempos de respuesta para cada solicitud o petición al sistema Web, según Ud., ¿cómo lo califica?							
3	Al utilizar el sistema Web, ¿puede Ud. acceder o hacer uso de otras aplicaciones y/o programas en su ordenador?							
4	Desde su punto de vista, ¿cómo califica el aprendizaje del sistema Web en funcionamiento? ¿Es fácil de aprender a usar?							
5	Cuando hay un corte de fluido eléctrico o pérdida de comunicación con el servidor, el sistema Web se restablece, ¿cómo considera Ud. el tiempo de recuperación del mismo?							
6	¿Considera Ud. que cualquier persona puede acceder al sistema Web y manipularlo?							
7	Desde su punto de vista, ¿cree Ud. que el sistema Web podrá adaptarse a nuevos requerimientos? ¿Podrán añadirse nuevas opciones?							
8	Desde su punto de vista, ¿cree Ud. que el sistema Web podrá entrar en funcionamiento en otro sistema operativo u otro ordenador?							
9	Desde su punto de vista, ¿cómo califica el grado de facilidad que le ofrece el sistema Web en funcionamiento para realizar sus tareas?							
10	Si compara el sistema Web en funcionamiento con el sistema anterior, ¿cómo califica la productividad (número de aportaciones realizadas y número de registros de colegiaturas realizadas por hora) que le ofrece el nuevo sistema?							
11	¿Se siente satisfecho con las funcionalidades que le brinda el sistema en funcionamiento? ¿Funciona en la forma esperada y correcta?							
12	¿Considera Ud. que el sistema en funcionamiento es usable sólo por un usuario experto?							
13	¿La información que muestra el sistema en funcionamiento es completa y precisa?							

Fuente: Elaboración propia

A continuación, se muestra a detalle el análisis de los resultados del proceso de evaluación post-test (Después de la prueba del sistema Web).

Pregunta 1: Desde su punto de vista, ¿el sistema Web en funcionamiento hace lo que le pide?

Figura 52

Histograma de la frecuencia acumulada en la pregunta 1.

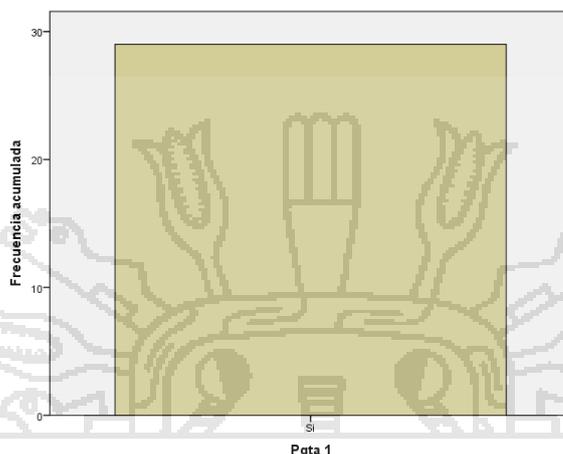


Tabla 30

Frecuencia acumulada en la pregunta 1.

Pregunta 1

	Frecuencia	Porcentaje
Válidos Si	29	100,0

Fuente: Elaboración propia.

De los resultados, el 100% opina que sí, entonces, se concluye que el sistema Web en funcionamiento hace lo que se le pide.

Pregunta 2: Los tiempos de respuesta para cada solicitud o petición al sistema Web, según Ud., ¿cómo lo califica?

Figura 53

Histograma en la frecuencia acumulada en la pregunta 2.

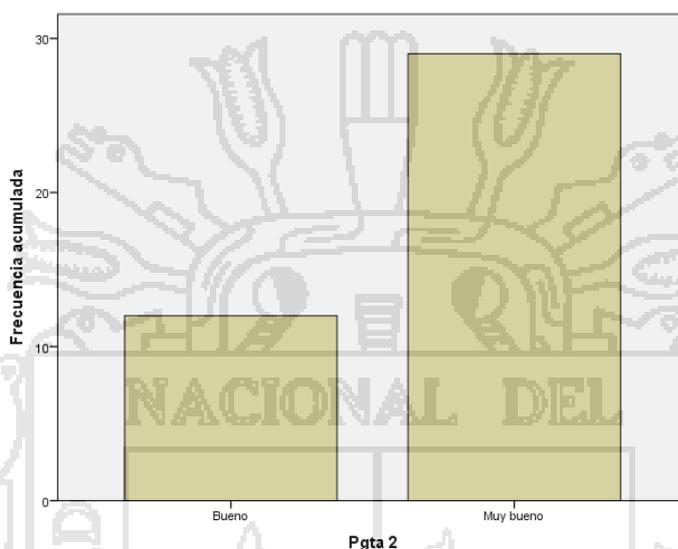


Tabla 31

Frecuencia acumulada en la pregunta 2.

Pregunta 2

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Buena	12	41,4	41,4	41,4
Muy buena	17	58,6	58,6	100,0
Total	29	100,0	100,0	

Fuente: Elaboración propia.

El 41,4% opina que es buena, y el 58,6% opina que es muy buena, entonces, se concluye que los tiempos de respuesta para cada solicitud o petición al sistema Web es bueno o muy bueno.

Pregunta 3: Al utilizar el sistema Web, ¿puede Ud. acceder o hacer uso de otras aplicaciones y/o programas en su ordenador?

Figura 54

Histograma de la frecuencia acumulada en la pregunta 3.



Tabla 32

Frecuencia acumulada en la pregunta 3.

Pregunta 3

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Si	29	100,0	100,0	100,0

Fuente: Elaboración propia.

De los resultados, el 100% opina que sí, entonces, se concluye los usuarios pueden acceder o hacer uso de otras aplicaciones y/o programas mientras utilizan el sistema Web.

Pregunta 4: Desde su punto de vista, ¿cómo califica el aprendizaje del sistema Web en funcionamiento? ¿Es fácil de aprender a usar?

Figura 55

Histograma de la frecuencia acumulada en la pregunta 4.

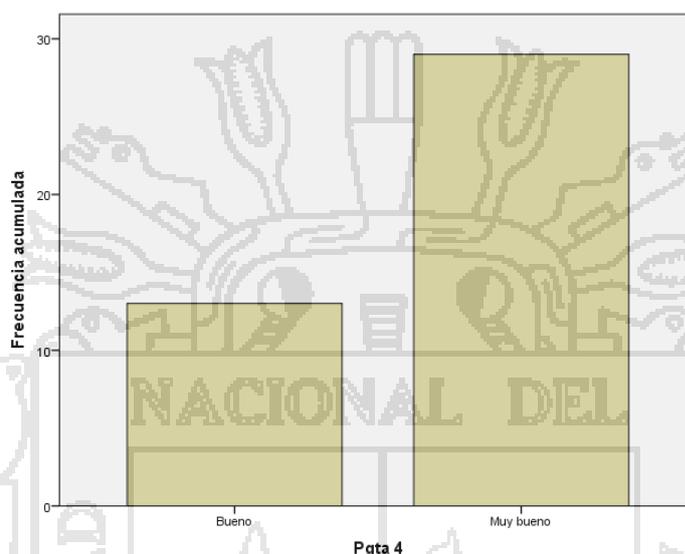


Tabla 33

Frecuencia acumulada en la pregunta 4.

Pregunta 4

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Buena	13	44,8	44,8	44,8
	Muy buena	16	55,2	55,2	100,0
	Total	29	100,0	100,0	

Fuente: Elaboración propia.

De los resultados, el 44,8% opina que es buena, y el 55,2% opina que es muy buena, entonces, se concluye que el sistema Web en tiene un sencillo aprendizaje en su uso.

Pregunta 5: Cuando hay un corte de fluido eléctrico o pérdida de comunicación con el servidor, el sistema Web se restablece, ¿cómo considera Ud. el tiempo de recuperación del mismo?

Figura 56

Histograma de la frecuencia acumulada en la pregunta 5.

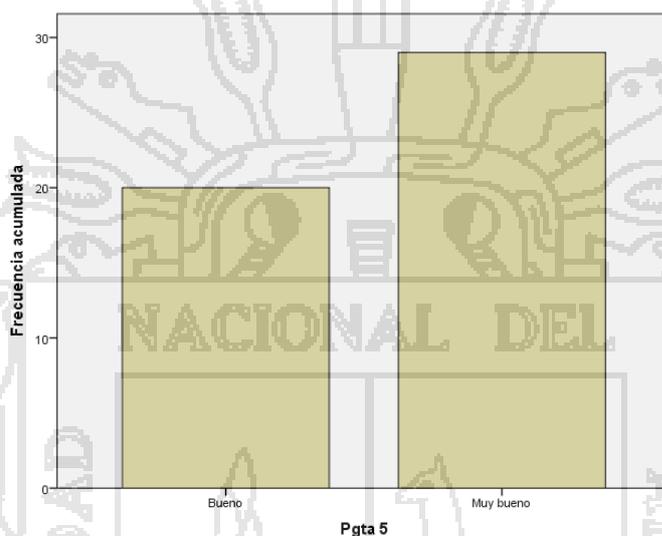


Tabla 34

Frecuencia acumulada en la pregunta 5.

Pregunta 5

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Buena	20	69,0	69,0	69,0
Muy buena	9	31,0	31,0	100,0
Total	29	100,0	100,0	

Fuente: Elaboración propia.

De los resultados, el 69% opina que es bueno, y el 31% opina que es muy bueno, entonces, se concluye que el sistema Web tiene un tiempo de recuperación bueno.

Pregunta 6: ¿Considera Ud. que el sistema Web es seguro? (cualquier usuario no puede acceder a él).

Figura 57

Histograma de la frecuencia acumulada en la pregunta 6.



Tabla 35

Frecuencia acumulada en la pregunta 6.

Pregunta 6

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Si	29	100,0	100,0	100,0

Fuente: Elaboración propia.

De los resultados, el 100% opina que sí, entonces, se concluye que el sistema Web es seguro y no cualquiera puede acceder a él.

Pregunta 7: Desde su punto de vista, ¿cree Ud. que el sistema Web podrá adaptarse a nuevos requerimientos? ¿Podrán añadirse nuevas opciones?

Figura 58

Histograma de la frecuencia acumulada en la pregunta 7.

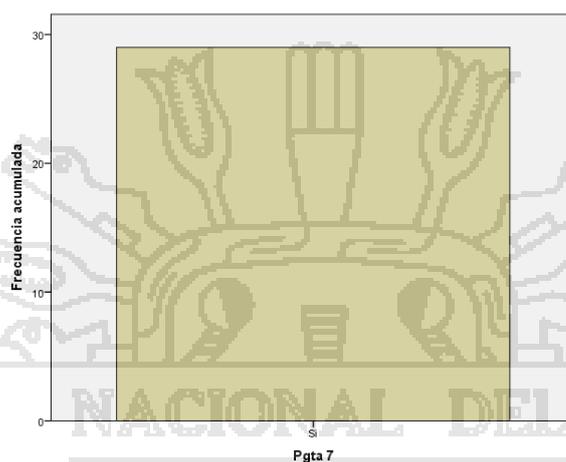


Tabla 36

Frecuencia acumulada en la pregunta 7.

Pregunta 7

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Si	29	100,0	100,0	100,0

Fuente: Elaboración propia.

De los resultados, el 100% opina que sí, entonces, se concluye que el sistema Web podrá adaptarse a nuevos requerimientos o podrá incorporar nuevas opciones.

Pregunta 8: Desde su punto de vista, ¿cree Ud. que el sistema Web podrá entrar en funcionamiento en otro sistema operativo u otro ordenador?

Figura 59

Histograma de la frecuencia acumulada en la pregunta 8.



Tabla 37

Frecuencia acumulada en la pregunta 8.

Pregunta 8

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Si	29	100,0	100,0	100,0

Fuente: Elaboración propia.

De los resultados, el 100% opina que sí, entonces, se concluye que el sistema Web puede entrar en funcionamiento en otro sistema operativo u ordenador.

Pregunta 9: Desde su punto de vista, ¿cómo califica el grado de facilidad que le ofrece el sistema Web en funcionamiento para realizar sus tareas?

Figura 60

Histograma de la frecuencia acumulada en la pregunta 9.

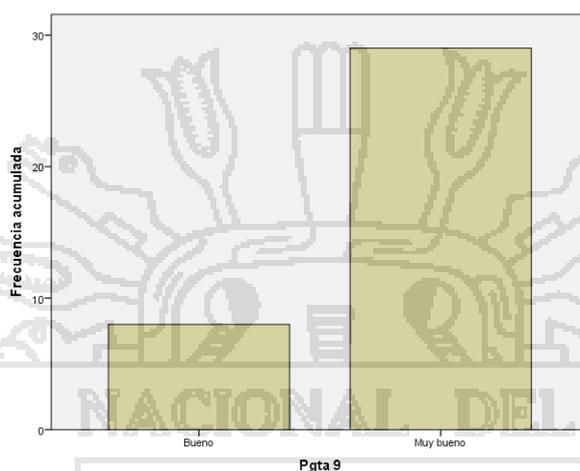


Tabla 38

Frecuencia acumulada en la pregunta 9.

Pregunta 9

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Buena	8	27,6	27,6	27,6
	Muy buena	21	72,4	72,4	100,0
	Total	29	100,0	100,0	

Fuente: Elaboración propia.

De los resultados, el 27,6% opina que es buena, y el 72,4% que es muy buena, entonces, se concluye que el sistema Web permite fácilmente realizar las tareas de los usuarios.

Pregunta 10: Si compara el sistema Web en funcionamiento con el sistema anterior, ¿cómo califica la productividad (número de aportaciones realizadas y número de registros de colegiaturas realizadas por hora) que le ofrece el nuevo sistema?

Figura 61

Histograma de la frecuencia acumulada en la pregunta 10.

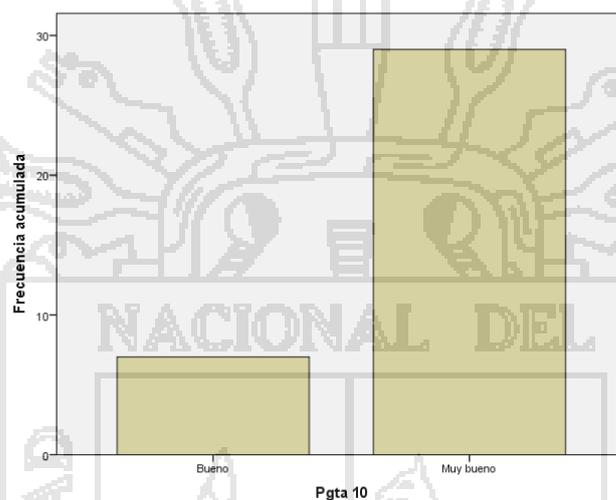


Tabla 39

Frecuencia acumulada en la pregunta 10.

Pregunta 10

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Buena	7	24,1	24,1	24,1
	Muy buena	22	75,9	75,9	100,0
	Total	29	100,0	100,0	

Fuente: elaboración propia.

De los resultados, el 24,1% opina que es buena, y el 75,9% opina que es muy buena, entonces, se concluye que el sistema Web en funcionamiento cuenta con una muy buena productividad.

Pregunta 11: ¿Se siente satisfecho con las funcionalidades que le brinda el sistema en funcionamiento? ¿Funciona en la forma esperada y correcta?

Figura 62

Histograma de la frecuencia acumulada en la pregunta 11.

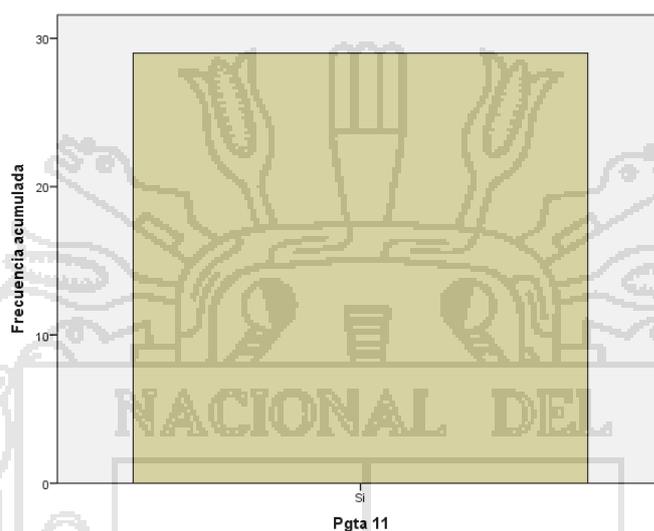


Tabla 40

Frecuencia acumulada en la pregunta 11.

Pregunta 11

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos - Si	29	100,0	100,0	100,0

Fuente: Elaboración propia.

De los resultados, el 100% opina que sí, entonces, se concluye que el sistema Web en funcionamiento cuenta con las funcionalidades necesarias para los usuarios.

Pregunta 12: ¿Considera Ud. que el sistema Web en funcionamiento puede ser utilizado por usuarios con conocimientos básicos en computación y no sólo por un usuarios expertos?

Figura 63

Histograma de la frecuencia acumulada en la pregunta 12.

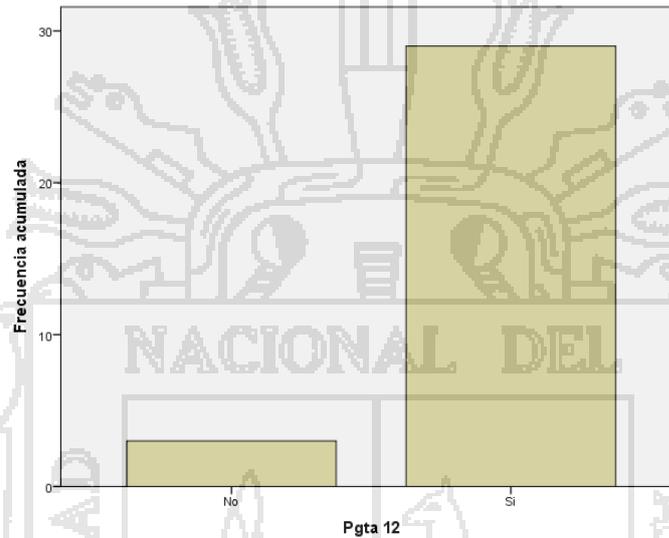


Tabla 41

Frecuencia acumulada en la pregunta 12.

Pregunta 12

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos No	3	10,3	10,3	10,3
Válidos Si	26	89,7	89,7	100,0
Total	29	100,0	100,0	

Fuente: Elaboración propia.

De los resultados, el 10,3% opina que no, y el 89,7% opina que sí, entonces, se concluye que el sistema Web en funcionamiento no requiere de usuarios expertos para su uso.

Pregunta 13: ¿La información que muestra el sistema Web en funcionamiento es completa y precisa?

Figura 64

Histograma de la frecuencia acumulada en la pregunta 13.

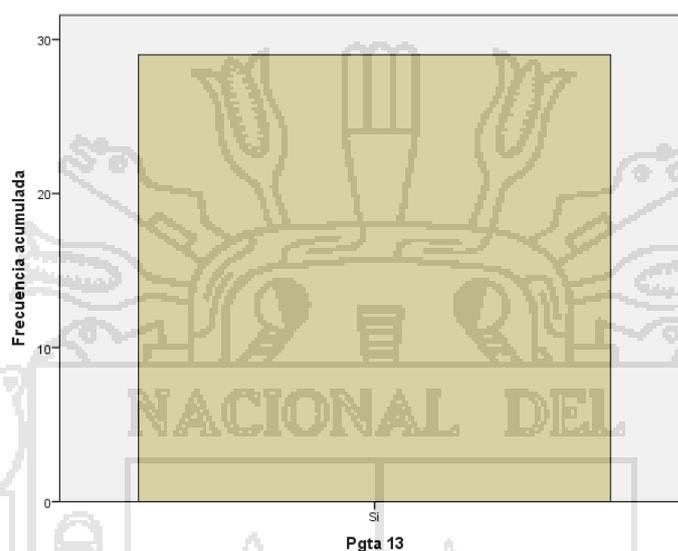


Tabla 42

Frecuencia acumulada en la pregunta 13.

Pregunta 13

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Si	29	100,0	100,0	100,0

Fuente: Elaboración propia.

De los resultados, el 100% opina que sí, entonces, se concluye que el sistema Web en funcionamiento presenta la información solicitada en forma completa y precisa.

4.4. PRUEBA DE HIPÓTESIS

La prueba de hipótesis se realizó con el método Pretest – Postest (Mencionados en el Capítulo III), que permite aceptar o rechazar la hipótesis. Para esto se realizó un cálculo manual con el estadístico t-Student, así como también se utilizaron los paquetes estadísticos del SPSS Statistics 21 y el MINITAB 16 para comparar los resultados.

4.4.1. ESTADÍSTICO DE PRUEBA

Para muestras que son menores a 30 es conveniente aplicar la prueba de t-Student para validar la hipótesis, en este caso basándose en los resultados obtenidos de las dos encuestas.

Los resultados de la tabla 43 se procesaron devolviéndose como resultado el valor de t-Student.

Tabla 43

Definición de variables para la contrastación entre el pre-test y post-test.

Abreviatura	Descripción
$GA_i (\bar{Y}_1)$	Gestión Actual.
$GP_i (\bar{Y}_2)$	Gestión Propuesta.
D_i	$(GA_i - GP_i)$
$D_j - \bar{D}$	Diferencia D_j – Promedio de diferencias
$(D_j - \bar{D})^2$	(Diferencia D_j – Promedio de diferencias) ²

Fuente: Elaboración propia.

Tabla 44

Contrastación entre el pre-test y post-test de las encuestas.

Nº	PRE-TEST	POST-TEST	D_j	$D_j - \bar{D}$	$(D_j - \bar{D})^2$	
	$(GA_i \sigma_1^2)$	$(GP_i \sigma_2^2)$				
Número de Personas Observadas	1	19	41	-22	-2.65517241	7.04994055
	2	19	41	-22	-2.65517241	7.04994055
	3	19	41	-22	-2.65517241	7.04994055
	4	19	38	-19	0.344827586	0.11890606
	5	18	37	-19	0.344827586	0.11890606
	6	19	41	-22	-2.65517241	7.04994055
	7	19	36	-17	2.344827586	5.49821641
	8	18	39	-21	-1.66517241	2.73959572
	9	17	36	-19	0.344827586	0.11890606
	10	27	39	-12	7.344827586	53.9464923
	11	17	37	-20	-0.65517241	0.42925089



12	17	38	-21	-1.65517241	2.73959572
13	19	38	-19	0.344827586	0.11890606
14	19	39	-20	-0.65517241	0.42925089
15	19	40	-21	-1.66517241	2.73959572
16	19	37	-18	1.344827586	1.80856124
17	19	40	-21	-1.66517241	2.73959572
18	19	37	-18	1.344827586	1.80856124
19	27	37	-10	9.344827586	87.3258026
20	19	38	-19	0.344827586	0.11890606
21	18	37	-19	0.344827586	0.11890606
22	19	41	-22	-2.65517241	7.04994055
23	19	41	-22	-2.65517241	7.04994055
24	19	41	-22	-2.65517241	7.04994055
25	19	41	-22	-2.65517241	7.04994055
26	19	38	-19	0.344827586	0.11890606
27	26	38	-12	7.344827586	53.9464923
28	19	40	-21	-1.66517241	2.73959572
29	19	39	-20	-0.65517241	0.42925089
SUMATORIAS			-561	7.10543E-15	276.551724

Fuente: Elaboración propia.

Dado que el estudio se realizó sobre un grupo experimental medido antes y después para verificar un cambio (muestras relacionadas), el estadístico t-Student de contraste a utilizar será el siguiente:

$$t = \frac{\bar{D} - 0}{S_{\bar{D}}}$$

Donde:

\bar{D} : Es la media de la diferencia de medias de la muestra.

$$\bar{D} = \bar{Y}_1 - \bar{Y}_2 = \sum \frac{D_j}{n} \dots \dots \dots (1.2)$$

$S_{\bar{D}}$: Es el error típico de la diferencia de medias.

$$S_{\bar{D}} = \frac{S_D}{\sqrt{n}} \dots \dots \dots (1.3)$$

S_D : Es la desviación típica de la diferencia menos la diferencia de medias.

$$S_D = \sqrt{\frac{\sum(D_j - \bar{D})^2}{n - 1}} \dots\dots\dots (1.4)$$

El tamaño de la muestra: $n = 29$

Para un t-Student que se distribuye con $k = n - 1$ grados de libertad.

4.4.2. OPERACIONALIZACIÓN DE LA HIPÓTESIS

HIPÓTESIS NULA H_0 = Un Sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE) no mejora la gestión de información del Colegio de Ingenieros del Perú CD Puno.

HIPÓTESIS ALTERNA H_1 = Un Sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE) mejora la gestión de información del Colegio de Ingenieros del Perú CD Puno.

$$H_0 = GA \geq GP$$

$$H_1 = GA < GP$$

NIVEL DE SIGNIFICANCIA:

El nivel de significancia (α) escogido para la prueba de la hipótesis será del 5%. Siendo $\alpha = 0.05$ (Nivel de Significancia) y $k = n - 1$ grados de libertad.

Para $k = n - 1 = 29 - 1 = 28$, se tiene el valor crítico de t-Student:

$$t_{\alpha-0.05} = -1.7011$$

La región de rechazo consiste en aquellos valores de t menores que $t_{\alpha-0.05}$.

4.4.3. RESULTADOS DE LA PRUEBA DE HIPÓTESIS

Diferencia Promedio:

Reemplazando en la fórmula 1.2, se tiene:

$$\bar{D} = \frac{-561}{29}$$

$$\bar{D} = -19,3448276$$

Desviación típica:

Reemplazando en la fórmula 1.4, se tiene:

$$S_D = \sqrt{\frac{276.551724}{28}}$$

$$S_D = 3.142745184$$

Error típico:

Reemplazando en la fórmula 1.3, se tiene:

$$S_{\bar{D}} = \frac{3.142745184}{\sqrt{29}}$$

$$S_{\bar{D}} = 0.58359313$$

Cálculo de T:

Reemplazando en la fórmula 1.1, se tiene:

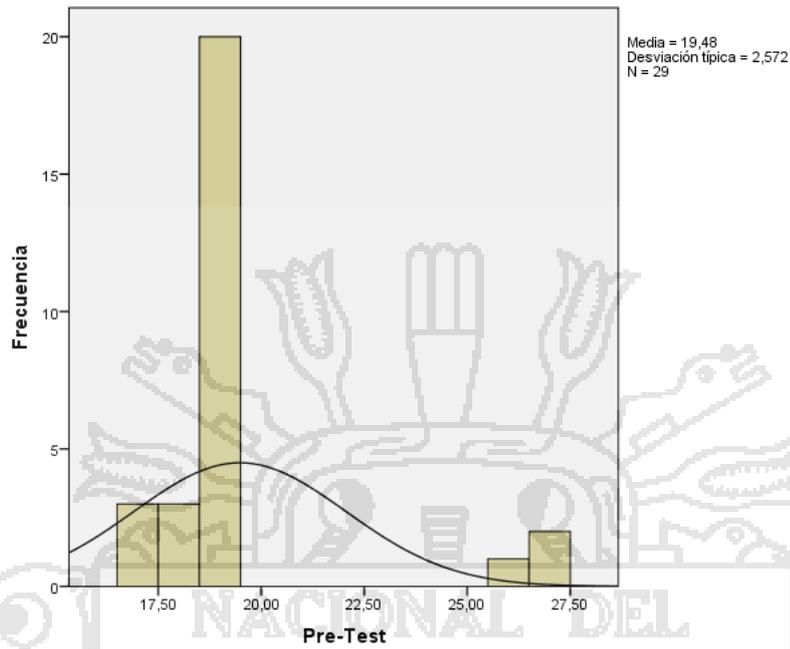
$$t = \frac{-19,3448276 - 0}{0.58359313}$$

$$t = -33.1477987$$

Resultado obtenido con el paquete estadístico IBM SPSS Statistics 21:

Figura 65

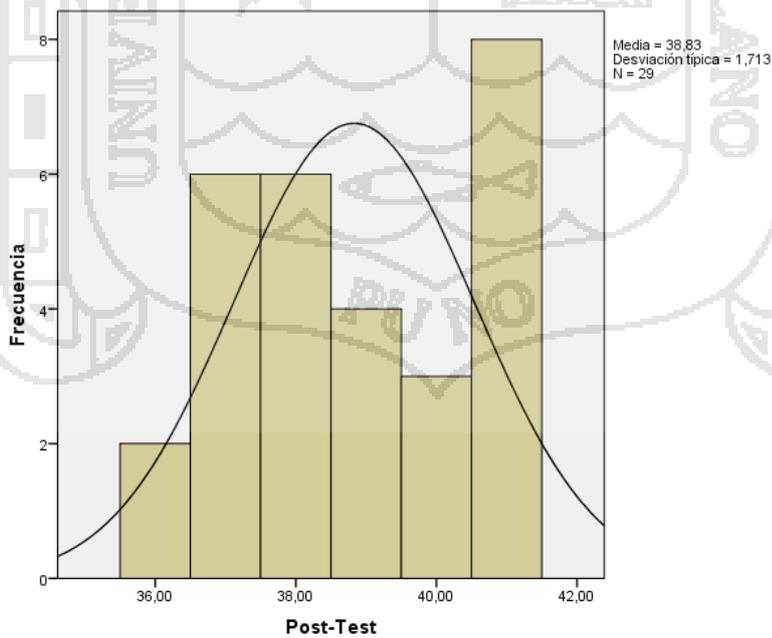
Histograma de la muestra pre-test.



Fuente: Elaboración propia.

Figura 66

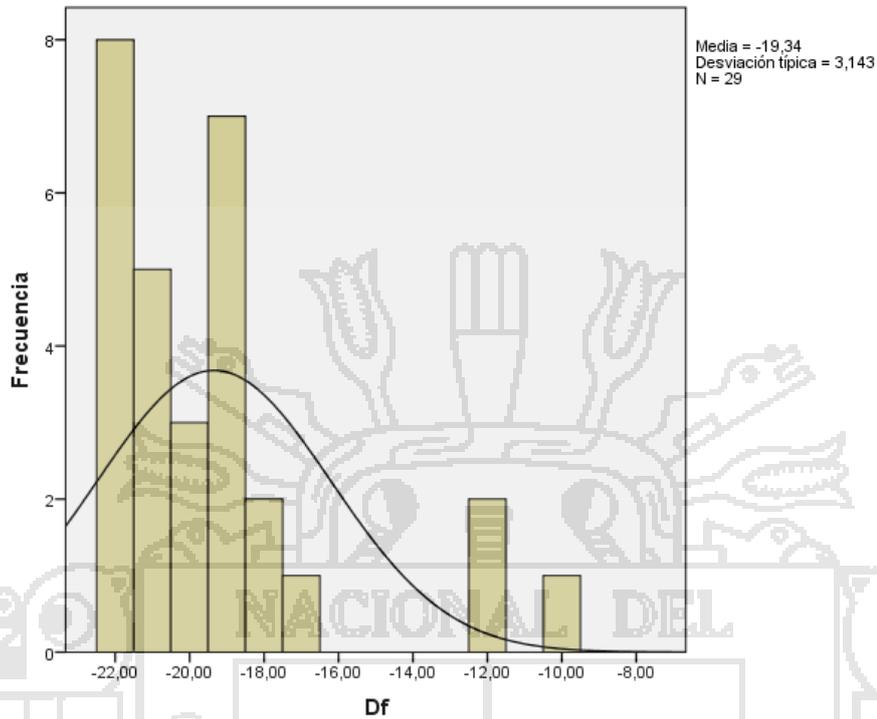
Histograma de la muestra post-test.



Fuente: Elaboración propia.

Figura 67

Histograma de la diferencia de las muestras (Pre-test – post-test).



Fuente: Elaboración propia.

Puede observarse la asimetría que existe en las muestras del *pre-test* y *post-test*. En la gráfica de las diferencias muestra un claro apuntamiento o curtosis positiva, además de un claro desplazamiento de los datos hacia el lado negativo de la gráfica (por debajo de la media) que indica una alta asimetría positiva.

Prueba T

Tabla 45

Estadísticos de muestras relacionadas.

		Media	N	Desviación típ.	Error típ. de la media
Par 1	PreTest	19,4828	29	2,57211	,47763
	PostTest	38,8276	29	1,71275	,31805

Fuente: Elaboración propia.

Tabla 46

Correlación de muestras relacionadas.

	N	Correlación	Sig.
Par 1 PreTest y PostTest	29	-,037	,848

Fuente: Elaboración Propia.

Tabla 47

Prueba de muestras relacionadas.

	Diferencias relacionadas			
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia
				Inferior
Par 1 PreTest - PostTest	-19,34483	3,14275	,58359	-20,54026

Fuente: Elaboración propia.

Tabla 48

Prueba de muestras relacionadas.

	Diferencias relacionadas	T	Gl	Sig. (bilateral)
	95% Intervalo de confianza para la diferencia			
	Superior			
Par 1 PreTest – PostTest	-18,14939	-33,148	28	,000

Fuente: Elaboración propia.

La primera de las tablas muestra los estadísticos descriptivos más comúnmente usados.

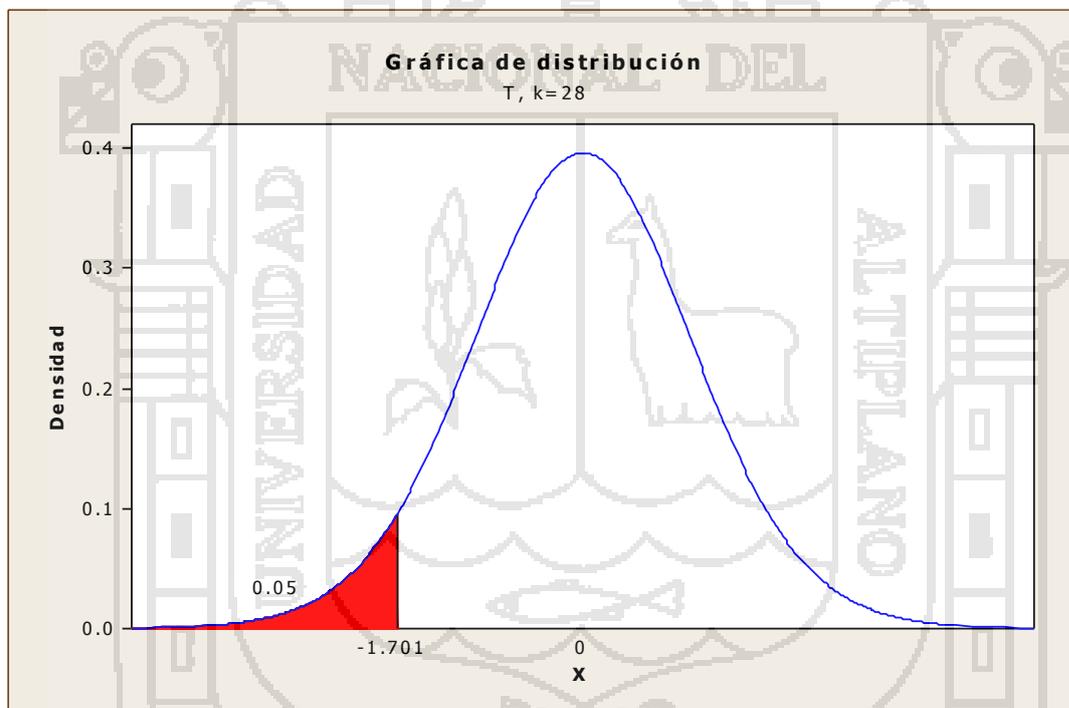
La segunda muestra la correlación y la significación de la prueba *t*. Si el valor de la significación estadística es menor que α , aceptamos que la correlación es significativamente diferente de 0, es decir, existe relación entre las variables.

Las últimas tablas permiten conocer el valor de t con un $\alpha = 0.05$ (nivel de significancia del 95%), donde la diferencia entre las medias es de -19.34483 y el límite aceptable está comprendido entre -20.54026 y $-18,14939$. Entonces, la diferencia se encuentra dentro de ese intervalo, y se asume que las medias son diferentes. También se observa el estadístico t que vale $-33,148$ y junto a él su significancia o valor p que es $0,00$. Dado que este valor es menor que $0,05$, se rechaza la hipótesis nula H_0 .

Conclusión de la prueba de hipótesis

Figura 68

Prueba de hipótesis (Gráfica obtenida con MINITAB 16).



Fuente: Elaboración propia.

Resultado:

El valor calculado u obtenido de $t = -33.1477$ se compara con los valores críticos de la distribución de la tabla t -Student y se observa que a una probabilidad de 0.05 y con $k = 28$ grados de libertad le corresponde $t_{\alpha=0.05} = -1.7011$.

Decisión:

Puesto que: $t = -33.1477$ (Calculado) $< t_{\alpha-0.05} = -1.7011$ (Tabular), estando este

valor dentro de la región de rechazo, se concluye que:

$H_0: GA \geq GP$ se rechaza y $H_1: GA < GP$ se acepta.

Por lo tanto se prueba la validez de la hipótesis con un nivel de error de 5% ($\alpha = 0.05$),

siendo la implementación Sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE) una alternativa de solución para el problema de la gestión de información en el CIP CD Puno.

Interpretación:

La gestión de información en el CIP CD Puno mejora después de la implementación del Sistema Web basado en la Norma ISO/IEC 25010:2010 (SQuaRE), permitiendo realizar el proceso de colegiación y pago de aportaciones con mayor eficiencia, efectividad y con un alto grado de productividad.



PRIMERO: Desarrollando el Sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE), se mejoró la gestión de información en el Colegio de Ingenieros del Perú CD Puno, permitiendo realizar el proceso de colegiación y pago de aportaciones con mayor eficiencia, efectividad y con un alto grado de productividad.

SEGUNDO: Con el análisis y diseño del sistema Web se determinaron los factores que intervienen en una buena gestión de información del CIP CD Puno. Es decir, se logró conocer todas las necesidades de los usuarios y establecer las funcionalidades del nuevo sistema para el logro de objetivos del CIP CD Puno.

TERCERO: Con la implementación del sistema Web basado en la norma ISO/IEC 25010:2010 (SQuaRE), se logró construir un producto Web de calidad, el mismo que permitirá al CIP CD Puno realizar sus tareas para beneficio de la organización y de los afiliados.

CUARTO: Con la encuesta elaborada en base al modelo de calidad de la norma ISO/IEC 25000 (SQuaRE), que considera las características de calidad interna, externa y en uso, y realizada en el CIP CD Puno, se obtuvo como resultado que el 83.3% de la muestra aprueba el sistema Web como “Muy Bueno”, lo que significa que tiene un alto grado de aceptación, por consiguiente en la realización de la prueba estadística de nivel de significancia de 5% se obtuvo como $p\text{-value} = 0,00 \leq 0,05$, lo que indica la validación de la hipótesis planteada en esta investigación.



**CAPÍTULO VI. RECOMENDACIONES Y
SUGERENCIAS**

PRIMERO: Se recomienda la adquisición de servidores para la base de datos que se administra, así como para sus aplicaciones Web. El uso de servidores permitirá personalizar y configurar los niveles de seguridad.

SEGUNDO: Para brindar un mejor servicio y proporcionar información completa y segura a los agremiados, se recomienda realizar la actualización de los datos de cada colegiado periódicamente.

TERCERO: Ya que no es común el uso de estándares o normas en el desarrollo de productos software, y menos aún, se tiene un estándar para modelos de calidad en entornos Web, se recomienda la creación de una base de conocimiento que recoja tanto factores de calidad como modelos, y así mismo, hacer uso de esos modelos para el desarrollo de productos software en las diferentes etapas de su ciclo de vida, pues ello contribuirá a tener mejores prácticas en el desarrollo de productos software.

CUARTO: Se sugiere investigar sobre otras tecnologías, metodologías, normas y buenas prácticas que pudieran mejorar la gestión de información, no solo en la sede departamental del CIP CD Puno y sus comités provinciales, sino también en otras organizaciones y/o entidades similares o distintas a ésta, tales como: MPLS, ITIL, COBIT, normas ISO, etc.

BIBLIOGRAFÍA

- ADAMS, E. (2002). *La gestión de recursos de información: importancia, desafíos y responsabilidades*. Obtenido de <http://abgra.sisbi.uba.ar/t14.htm>
- CALERO, C. R. (Mayo de 2005). "Classifying web metrics using the web quality model". *Online Information Review*. OIR - 29, 3. United Kingdom: Emerald Press.
- CHOO, C. W. (2002). *Information Management for the Intelligent Organization: The Art of Scanning the Environment*. 3rd ed. Medford: American Society for Information Science.
- CORDOBA, J. (2007). *Modelo de Calidad para Portales Bancarios*.
- DEBONS, A. H. (1988). *Information Science: An Integrated View*. Boston: G. K. Hall.
- ESPARZA SALANOVA, P. (2004). *Modelos de Calidad Web. Clasificación de Métricas*. Madrid.
- GÓMEZ CRUZ, R. D. (2011). *Sistema Web de Seguimiento de Procesos Judiciales para la Gestión de la Información en el estudio de abogados Trujillo y Asociados de la provincia de Azángaro*. Puno.
- HERNÁNDEZ SAMPIERI, R. B. (2006). *Metodología de la Investigación Científica (4ta ed.)*. México: McGraw-Hill.
- HERNÁNDEZ VEGA, B. (2007). *Diseño de Prototipo de un Sistema de Gestión de la Información para el Grupo de Trabajo de Alimentos y Bebidas del Hotel "Brisas Santa Lucia"*.
- HILL, M. W. (2000). *The impact of Information on Society: An examination of its nature, value and usage*. London, Bowker-Saur.

- ISO/IEC. (1999). *Iso/iec 14598-1:1999, Information technology - Software product evaluation - part 1: General overview. Technical report, International Organization for Standardization.* Geneva, Switzerland.
- ISO/IEC. (2001). *Iso/iec 9126-1:2001, Software engineering - Product quality - Part 1: Quality model. Technical report, International Organization for Standardization.* Geneva, Switzerland.
- ISO/IEC. (2005). *Iso/iec 25000:2005 Software engineering - Software product QUALity Requirements and Evaluation (SQuaRE) - Guide to SQuaRE. Technical report, International Organization for Standardization.* Geneva, Switzerland.
- ISO/IEC. (2010). *Iso/iec fdis 25010, Systems and software engineering -Systems and software Quality Requirements and Evaluation (SQuaRE)- System and software quality models.*
- L.BUGLIONE, F. E. (23-25 de September de 2003). "A Quality Model for Web-based Environments". IWSM2003. 13th International Workshop on Software Measurement Montréal. GUFPI ISMA - Software Measurement Committee (SMC).
- LILBURNE, B. D. (2004). "Measuring Quality Metrics for Web Applications". Technical Report n°: CIT/08/2004. Irma International Conference. New Orleans, USA.
- McGILVRAY, D. G. (2008). *Excerpted from Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information™.* Morgan Kaufmann Publishers. Copyright © 2008 Elsevier Inc. All rights reserved.
- MINGUET, J. H. (2003). *La calidad del software y su medida.*

- OLSINA, L. (2007). *Ingeniería Web: Marco de medición y evaluación de calidad*. La Rioja - Catamarca: Departamento de informática de la Universidad nacional de San Luis.
- OWEN, D. (1989). *IRM Concepts: Building blocks for the 1990's. Information Management Review, Vol.5, no.2*.
- PAEZ URDANETA, I. (1992). *Gestión de la inteligencia, aprendizaje tecnológico y modernización del trabajo intelectual: retos y oportunidades*. Caracas: Universidad Simón Bolívar.
- PAREJAS, C. (2004). *Modelo de Calidad para un Producto de Software: Banca On-Line, basado en el estándar de calidad ISO/IEC 9126*. Madrid.
- PAREJAS, C. (2004). *Propuesta de un Modelo de Calidad del Software aplicado a la Web*. Madrid.
- PIEDRAHITA, S. (2007). *Construcción de una Herramienta para evaluar la Calidad de un Producto Software*. Medellín.
- PRESSMAN, R. S. (2001). *Ingeniería del Software: Un Enfoque Práctico*. Madrid: Mc Graw Hill.
- PRYTHERCH, R. (2000). *Harrod's Librarian's Glossary and Reference Book*. Aldershot, Gower.
- RODRÍGUEZ ROVIRA, J. (1999). *Gestión del conocimiento. En: El Profesional de la Información, Vol. 8, no. 3,*
- RODRÍGUEZ SALAS, K. (2002). *Gestión de la Información en las Organizaciones*.
- RUIZ, J. C. (2003). *"A Three Dimensional Web Quality Model", Proceedings of the International Conference on Web Engineering (ICWE'03)*.
- SCALONE, F. (2006). *Software quality management. Overview sobre modelos/estándares de calidad del SW*.

SOMMERVILLE, I. (2005). *Ingeniería de Software*. Madrid: Pearson Addison Wesley.





ANEXO A: ESPECIFICACIÓN DE CASOS DE USO

A continuación se muestra cada una de las especificaciones de los casos de uso que se han identificado en la etapa del análisis.

- Gestión de Usuarios:

CU-01		Dar de alta al usuario	
Versión	1.0		
Actores	Usuario (Admin)		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee registrarse en el sistema, permitiendo el acceso a servicios restringidos.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El Usuario solicita al sistema comenzar el proceso de registrar un nuevo usuario.	
	2	Si el usuario es un administrador, el sistema solicita al sistema realizar su alta como usuario registrado.	
	3	El sistema solicita los datos necesarios para registrar el nuevo usuario.	
	4	El Usuario proporciona al sistema los datos del nuevo usuario.	
	5	El sistema comprueba la validez de los datos introducidos.	
	6	El sistema almacena la información del nuevo usuario.	
	7	El sistema indica al usuario que la operación se ha realizado correctamente.	
Postcondición	El usuario se ha registrado en el sistema y tiene disponible su perfil de usuario.		
Excepciones	Paso	Acción	
	4	Si el usuario es administrador, el actor Administrador proporciona el tipo del usuario, a continuación ese caso de uso continúa.	
	5	Si los datos proporcionados no son correctos o ya existen, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.	
	6	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.	
Comentarios	Ninguno		

CU-02		Dar de baja al usuario	
Versión	1.0		
Actores	Usuario (Usuario Inscrito)		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiere dar de baja a otro usuario.		
Precondición	El usuario está registrado en el sistema		
Secuencia normal	Paso	Acción	
	1	Se realiza el caso de uso Identificar al usuario(CU-06)	
	2	El actor Usuario solicita al sistema comenzar el proceso de eliminar un usuario.	
	3	El sistema solicita el usuario a eliminar.	



	4	El actor Usuario proporciona al sistema el usuario a eliminar.
	5	El sistema elimina los datos correspondientes al usuario.
	6	El sistema indica que el proceso de eliminación ha terminado correctamente.
Postcondición	El sistema ha eliminado al usuario indicado.	
Excepciones	Paso	Acción
	4	Si el usuario es administrador, el actor Administrador proporciona el usuario a eliminar, a continuación ese caso de uso continúa.
	4	Si el usuario no es administrador, el sistema selecciona automáticamente al propio usuario, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-03	Modificar datos del usuario	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera modificar sus datos almacenados en el sistema.	
Precondición	El usuario está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificar al usuario (CU-06).
	2	El actor Usuario solicita al sistema comenzar el proceso de modificación de los datos del usuario.
	3	El sistema muestra los datos del usuario.
	4	El actor Usuario modifica los datos mostrados por el sistema.
	5	El sistema comprueba la validez de los datos introducidos.
	6	El sistema almacena la información actualizada.
	7	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El sistema ha modificado los datos de usuario.	
Excepciones	Paso	Acción
	5	Si los datos proporcionados no son correctos o ya existen, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.
	6	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-04	Consultar datos del usuario	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desea consultar sus datos almacenados en el sistema.	
Precondición	El usuario está registrado en el sistema	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificar al usuario(CU-06)
	2	El actor Usuario solicita al sistema comenzar el proceso de



		consulta de datos del usuario.
	3	El sistema muestra los datos del usuario.
	4	El actor Usuario consulta los datos mostrados por el sistema.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	3	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-05	Notificar información al usuario	
Versión	1.0	
Actores	Usuario (Admin)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador quiere enviarle información a un usuario.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificar al usuario (CU-06).
	2	El actor Administrador solicita al sistema comenzar el proceso de notificación al usuario.
	3	El sistema muestra la lista de usuarios registrados en el sistema y solicita al administrador el texto de la notificación.
	4	El actor Administrador proporciona al sistema el texto de la notificación.
	5	El sistema envía la notificación al usuario e indica al administrador que la operación se ha realizado correctamente.
Postcondición	El sistema envía un mensaje al usuario.	
Excepciones	Paso	Acción
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-06	Identificar al usuario	
Versión	1.0	
Actores	Usuario (Usuario Registrado, Admin)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso abstracto durante la realización de los demás casos de uso.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El sistema solicita al usuario que se identifique.
	2	El actor Usuario proporciona los datos de identificación.
	3	El sistema valida la información de identificación proporcionada por el usuario.
	4	El sistema confirma la identidad del usuario.
Postcondición	El sistema identifica al usuario y permite el acceso a los módulos del sistema de acuerdo a los privilegios que tiene.	
Excepciones	Paso	Acción
	3	Si la información no es correcta, el sistema muestra el correspondiente mensaje de error y solicita al usuario de nuevo la información de identificación, a continuación este caso de uso queda sin efecto.

Comentarios	Ninguno
-------------	---------

CU-07	Crear perfil del usuario	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso abstracto durante la realización del siguiente caso de uso: CU-01	
Precondición	El usuario está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema identifica al usuario al que se le creará el perfil.
	2	El sistema solicita al usuario que indique inicialmente las preferencias en su perfil.
	3	El actor Usuario proporciona los datos de su perfil.
	4	El sistema crea un perfil al usuario identificado.
Postcondición	El usuario está registrado en el sistema y tiene asignado un perfil.	
Excepciones	Paso	Acción
	1	Si se produce algún tipo de error o fallo durante la identificación del usuario, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
	4	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	El proceso de alta consiste tanto en el almacenamiento de su información como de su perfil con la información que desea recibir. El proceso de alta termina con la creación de su perfil.	

CU-08	Eliminar perfil	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso abstracto cuando el usuario desee eliminar su perfil.	
Precondición	El usuario está registrado en el sistema, junto con su perfil.	
Secuencia normal	Paso	Acción
	1	El sistema identifica al usuario al que se le eliminará el perfil.
	2	El sistema elimina el perfil del usuario identificado.
Postcondición	El usuario no está registrado en el sistema.	
Excepciones	Paso	Acción
	1	Si se produce algún tipo de error durante la identificación del usuario, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
	2	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	El proceso de baja consiste en dos pasos: eliminación de la información del usuario y la eliminación de su perfil.	

CU-09	Modificar perfil	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee cambiar sus preferencias de información a recibir.	
Precondición	El usuario está registrado en el sistema y tiene un perfil creado.	



Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificar al usuario (CU-06).
	2	El actor Usuario Registrado solicita al sistema comenzar el proceso de modificación de su perfil.
	3	El sistema muestra las preferencias actuales del perfil.
	4	El actor Usuario Registrado modifica su perfil, añadiendo y/o eliminando informaciones que desea o no recibir.
	5	El sistema actualiza el perfil del usuario con la nueva información proporcionada.
6	El sistema indica al usuario que la operación se ha realizado correctamente.	
Postcondición	El sistema modificó las preferencias del perfil del usuario.	
Excepciones	Paso	Acción
	5	Si se produce algún tipo de error o fallo en el proceso de modificación del perfil, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-10	Consultar perfil	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee consultar su perfil con la información a recibir y/o consultar.	
Precondición	El usuario está registrado en el sistema y tiene un perfil creado.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificar al usuario (CU-06).
	2	El actor Usuario Registrado solicita al sistema el proceso de consulta del perfil del usuario.
	3	El sistema muestra el perfil del usuario registrado.
	4	El actor Usuario consulta los datos reflejados en su perfil.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	3	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

- Gestión de Registro de Colegiaturas:

CU-11	Verificar registro anterior	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe verificar si ya existe un registro anterior de un ingeniero, antes de proceder con el registro de información del mismo.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de búsqueda de registro de ingenieros.
	2	El sistema solicita los datos necesarios para realizar la búsqueda del registro del ingeniero.
	3	El Usuario proporciona al sistema los datos del ingeniero.
	4	El sistema realiza la búsqueda del registro.
	5	El sistema devuelve la información de registro del



		ingeniero.
Postcondición	El usuario ha realizado la verificación de la existencia del registro de un ingeniero.	
Excepciones	Paso	Acción
	4	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
	5	En el caso que no exista el registro del ingeniero solicitado, el sistema muestra el mensaje correspondiente.
Comentarios	Ninguno	

CU-12	Registrar ingeniero	
Versión	1.0	
Actores	Usuario(Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario debe registrar los datos personales y detalles como residencia y centro laboral de un ingeniero en el sistema.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Verificar registro anterior (CU-11).
	2	El Usuario solicita al sistema comenzar el proceso de registrar los datos personales del ingeniero.
	3	El sistema solicita los datos necesarios para registrar los datos del ingeniero.
	4	El Usuario proporciona al sistema los datos personales y/o detalles del ingeniero.
	5	El sistema comprueba la validez de los datos introducidos.
	6	El sistema almacena la información del ingeniero.
	7	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado los datos personales y/o detalles de un nuevo ingeniero, en el sistema.	
Excepciones	Paso	Acción
	1	Si ya se encuentra un registro del ingeniero, el caso de uso queda sin efecto.
	5	Si los datos proporcionados no son correctos, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.
	6	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-13	Registrar especialidad	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario debe registrar en el sistema la(s) especialidad(es) del ingeniero previamente registrado.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar la(s) especialidad(es) del ingeniero.
	2	El sistema solicita los datos necesarios para registrar la(s) especialidad(es) del ingeniero.



	3	El Usuario proporciona al sistema la(s) especialidad(es) del ingeniero.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado la(s) especialidad(es) del ingeniero, en el sistema.	
Excepciones	Paso	Acción
	4	Si los datos proporcionados no son correctos, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-14	Registrar pago por derecho de colegiatura	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe registrar el pago por derecho de colegiatura en el sistema de un ingeniero.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar el pago por derecho de colegiatura de un ingeniero.
	2	El sistema proporciona la información de los conceptos de pago por derecho de colegiatura del ingeniero.
	3	El Usuario elige los conceptos de pago correspondientes al pago por derecho de colegiatura que debe realizar el ingeniero.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado el pago por derecho de colegiatura del ingeniero, en el sistema.	
Excepciones	Paso	Acción
	4	Si los datos proporcionados no son correctos, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-15	Eliminar ingeniero	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee eliminar a un ingeniero.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema identifica al ingeniero al que se le eliminará su



		registro.
	2	El sistema elimina el registro del ingeniero identificado.
Postcondición	El ingeniero no está registrado en el sistema.	
Excepciones	Paso	Acción
	1	Si se produce algún tipo de error durante la identificación del ingeniero, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
	2	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-16	Consultar registro de inscripción	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee consultar el registro de inscripción de un ingeniero.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso (CU-11).
	2	El actor Usuario solicita al sistema el proceso de consulta del registro de inscripción.
	3	El sistema muestra el registro del ingeniero registrado.
	4	El actor Usuario consulta los datos reflejados en su registro.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	3	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-17	Consultar costos	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee consultar los costos de los conceptos de los diversos pagos que realizan los ingenieros.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor Usuario solicita al sistema el proceso de consulta de costos de los conceptos de pago.
	2	El sistema muestra los registros de los conceptos de pago con sus respectivos montos.
	3	El actor Usuario consulta los datos reflejados en los registros solicitados.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	2	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-18	Consultar catálogo	
Versión	1.0	



Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee consultar el catálogo de servicios y/o conceptos de pago que puede realizar el ingeniero.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El actor Usuario solicita al sistema el proceso de consulta del catálogo de servicios y/o conceptos de pago que puede realizar el ingeniero.
	2	El sistema muestra los datos almacenados en el catálogo de servicios y/o conceptos de pago que puede realizar el ingeniero.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	2	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

- Programación de Colegiatura:

CU-19	Programar fecha de colegiatura	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee registrar la programación de una fecha para colegiatura.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso (CU-11).
	2	El actor Usuario solicita al sistema el proceso de programar una fecha de colegiatura.
	3	El sistema solicita los datos necesarios para registrar la fecha de colegiatura.
	4	El Usuario proporciona al sistema los datos de la fecha de colegiatura.
	5	El sistema comprueba la validez de los datos introducidos.
	6	El sistema almacena la información de la colegiatura.
	7	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	5	Si los datos proporcionados no son correctos, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.
	6	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-20	Agregar ingenieros a una fecha de programación	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee agregar ingenieros a una fecha de programación de colegiatura.	



Precondición	El ingeniero ya tiene asignado un Código CIP y ha realizado el pago correspondiente.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso (CU-11).
	2	El actor Usuario solicita al sistema el proceso de agregar ingenieros a una fecha de programación.
	3	El sistema muestra los ingenieros que ya cuentan con un código CIP.
	4	El Usuario selecciona a los ingenieros a quienes se les asignó recientemente el código CIP y previamente realizaron el pago respectivo.
5	El sistema indica al usuario que la operación se ha realizado correctamente.	
Postcondición	Listado de ingenieros que realizarán su juramentación.	
Excepciones	Paso	Acción
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-21	Agregar ingenieros rezagados	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee agregar ingenieros rezagados que no asistieron a su fecha de colegiatura programada.	
Precondición	El ingeniero ya tiene asignado un Código CIP y ha realizado el pago correspondiente, incluido el pago adicional.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso (CU-11).
	2	El actor Usuario solicita al sistema el proceso de agregar ingenieros rezagados a la última fecha de programación.
	3	El sistema muestra los ingenieros que ya cuentan con un código CIP.
	4	El Usuario selecciona a los ingenieros a quienes se les asignó recientemente el código CIP y previamente realizaron el pago respectivo.
5	El sistema indica al usuario que la operación se ha realizado correctamente.	
Postcondición	Listado de ingenieros que realizarán su juramentación.	
Excepciones	Paso	Acción
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-22	Cambiar estado a juramentado	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee cambiar el estado de no juramentado a juramentado a los ingenieros que asistieron a la ceremonia de colegiatura programada.	
Precondición	El ingeniero asistió a la ceremonia de juramentación programada.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso (CU-11).
	2	El actor Usuario solicita al sistema el proceso de cambio de

		estado a juramentado de los ingenieros que fueron programados.
	3	El sistema muestra los ingenieros que fueron programados para la última colegiatura.
	4	El usuario cambia el estado a los ingenieros que hayan asistido a su colegiatura programada.
	5	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	Ninguna	
Excepciones	Paso	Acción
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

- Gestión de asignación de CIP:

CU-23	Verificar existencia de CIP	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe verificar si ya existe un registro anterior de un ingeniero con código CIP asignado.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de búsqueda de registro de ingenieros.
	2	El sistema solicita los datos necesarios para realizar la búsqueda del registro del ingeniero.
	3	El Usuario proporciona al sistema los datos del ingeniero.
	4	El sistema realiza la búsqueda del registro.
	5	El sistema devuelve la información solicitada.
Postcondición	El usuario ha realizado la verificación de la existencia del registro de un ingeniero.	
Excepciones	Paso	Acción
	4	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
	5	En el caso que no exista un registro del código CIP solicitado, el sistema muestra el mensaje correspondiente.
Comentarios	Ninguno	

CU-24	Registrar CIP	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario debe registrar el código CIP del ingeniero en el sistema.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Verificar registro anterior (CU-11).
	2	El Usuario solicita al sistema comenzar el proceso de registrar el código CIP del ingeniero.
	3	El sistema solicita los datos necesarios para registrar el código CIP del ingeniero.
	4	El Usuario proporciona al sistema los datos necesarios para el registro de código CIP del ingeniero.

	5	El sistema comprueba la validez de los datos introducidos.
	6	El sistema almacena la información del código CIP del ingeniero.
	7	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado el código CIP de un ingeniero, en el sistema.	
Excepciones	Paso	Acción
	1	Si ya se encuentra un registro de código CIP del ingeniero, el caso de uso queda sin efecto.
	5	Si los datos proporcionados no son correctos, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.
	6	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-25	Registrar colegiado	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario debe registrar los datos de colegiatura como condición, fecha de incorporación y modalidad de colegiatura de un ingeniero, en el sistema.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar los datos de colegiatura del ingeniero.
	2	El sistema solicita los datos necesarios para registrar los datos de colegiatura del ingeniero.
	3	El Usuario proporciona al sistema los datos de colegiatura del ingeniero.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información de la colegiatura del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado los datos de colegiatura de un ingeniero, en el sistema.	
Excepciones	Paso	Acción
	4	Si los datos proporcionados no son correctos, el sistema muestra el correspondiente mensaje de error y permite al usuario modificar los datos proporcionados, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

- Pago por juramentación

CU-26	Realizar pago por juramentación	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe registrar el pago por concepto de	



	juramentación de un ingeniero en el sistema.	
Precondición	El ingeniero se encuentra en el listado de ingenieros programados para la próxima juramentación.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar el pago por derecho de juramentación del ingeniero.
	2	El sistema muestra los conceptos de pago (rubro de colegiatura).
	3	El Usuario selecciona el concepto de pago por derecho de juramentación.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información del pago del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado el pago por derecho de juramentación de un ingeniero, en el sistema.	
Excepciones	Paso	Acción
	4	Si no se ha seleccionado ningún concepto de pago, el sistema muestra el correspondiente mensaje de error y permite al usuario seleccionar el pago correspondiente, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-27	Realizar pago adicional	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe registrar el pago adicional por concepto de reprogramación de fecha para juramentación de un ingeniero, en el sistema.	
Precondición	El ingeniero ya no se encuentra en el listado de ingenieros programados para la próxima juramentación.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar el pago adicional por derecho de juramentación del ingeniero.
	2	El sistema muestra los conceptos de pago adicional (rubro de Colegiatura).
	3	El Usuario selecciona el concepto de pago adicional por derecho de juramentación.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información del pago adicional del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado el pago adicional por derecho de juramentación de un ingeniero, en el sistema.	
Excepciones	Paso	Acción
	4	Si no se ha seleccionado ningún concepto de pago, el sistema muestra el correspondiente mensaje de error y permite al usuario seleccionar el pago correspondiente, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación

	este caso de uso queda sin efecto.
Comentarios	Ninguno

- Pago de Aportaciones

CU-28	Realizar pago por servicio	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe registrar el pago por servicios, como certificado de habilidad, constancia de no adeudo, duplicado de carné, entre otros, que son solicitados por los ingenieros u otras personas, en el sistema.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar el pago por servicios.
	2	El sistema muestra los conceptos de pago por servicios (rubro de Servicios).
	3	El Usuario selecciona el concepto de pago por servicios solicitados por el ingeniero u otra persona.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información del pago del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado el pago por servicios solicitado por el ingeniero u otra persona, en el sistema.	
Excepciones	Paso	Acción
	4	Si no se ha seleccionado ningún concepto de pago, el sistema muestra el correspondiente mensaje de error y permite al usuario seleccionar el pago correspondiente, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-29	Realizar aporte por 3 meses	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe registrar el pago de aportes por 3 meses (pago realizado junto con el pago por juramentación) del ingeniero, en el sistema.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar el pago de aportes por 3 meses del ingeniero.
	2	El sistema muestra los conceptos de pago (rubro de Aportación).
	3	El Usuario selecciona el concepto de pago de aportes de 3 meses del ingeniero.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información del pago de aportes de 3 meses del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.



Postcondición	El usuario ha registrado el pago de aportes por 3 meses del ingeniero, en el sistema.	
Excepciones	Paso	Acción
	4	Si no se ha seleccionado ningún concepto de pago, el sistema muestra el correspondiente mensaje de error y permite al usuario seleccionar el pago correspondiente, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-30	Realizar aporte por n meses	
Versión	1.0	
Actores	Usuario (Usuario Registrado)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario debe registrar el pago de aportes por n meses (posteriores a los tres primeros) del ingeniero, en el sistema.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El Usuario solicita al sistema comenzar el proceso de registrar el pago de aportes de n meses del ingeniero.
	2	El sistema muestra los conceptos de pago (rubro de Aportación).
	3	El Usuario selecciona el concepto de pago de aportes de n meses del ingeniero.
	4	El sistema comprueba la validez de los datos introducidos.
	5	El sistema almacena la información del pago de aportes de n meses del ingeniero.
	6	El sistema indica al usuario que la operación se ha realizado correctamente.
Postcondición	El usuario ha registrado el pago de aportes de n meses del ingeniero, en el sistema.	
Excepciones	Paso	Acción
	4	Si no se ha seleccionado ningún concepto de pago, el sistema muestra el correspondiente mensaje de error y permite al usuario seleccionar el pago correspondiente, a continuación este caso de uso continúa.
	5	Si se produce algún tipo de error o fallo durante el proceso, el sistema informa al usuario de lo ocurrido, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

- Gestión de Consulta de Aportaciones

CU-31	Identificar al usuario	
Versión	1.0	
Actores	Usuario Ingeniero	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el ingeniero desee consultar el estado de sus aportaciones vía Web.	
Precondición	El ingeniero está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema solicita al usuario ingeniero que se identifique.
	2	El actor Usuario proporciona los datos de identificación (código CIP y número de DNI).
	3	El sistema valida la información de identificación



		proporcionada por el usuario ingeniero.
	4	El sistema confirma la identidad del usuario ingeniero y envía la información de aportaciones a su correo electrónico personal.
Postcondición	El sistema identifica al usuario ingeniero y envía la información de su estado de aportaciones.	
Excepciones	Paso	Acción
	3	Si la información no es correcta, el sistema muestra el correspondiente mensaje de error y solicita al usuario de nuevo la información de identificación.
	4	Si el correo electrónico personal del ingeniero identificado no se encuentra almacenado en la base de datos, el sistema muestra el correspondiente mensaje de error, a continuación este caso de uso queda sin efecto.
Comentarios	Ninguno	

CU-32	Obtener información de aportaciones	
Versión	1.0	
Actores	Usuario Ingeniero	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario ingeniero recibe la información de su estado de aportaciones mediante la consulta Web que realizó previamente, en su correo electrónico personal.	
Precondición	El ingeniero se identificó en el módulo de Consulta de Aportaciones del sistema Web.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso (CU-31).
	2	El actor Usuario recibe en su correo electrónico personal la información de su estado de aportaciones.
Postcondición	El ingeniero tiene un mensaje en su correo electrónico de la información de su estado de aportaciones, enviada automáticamente al identificarse en el módulo de Consulta de aportaciones del sistema Web.	
Excepciones	Paso	Acción
	2	Si el correo electrónico que se encuentra registrado en la base de datos es incorrecto, el usuario ingeniero no recibirá la información de su estado de aportaciones, a continuación este caso de uso queda sin efecto.
Comentarios	El ingeniero recibe la información de su estado de aportaciones en el correo electrónico que registró a momento de colegiarse.	

ANEXO B: ESQUEMA FÍSICO DE LA BASE DE DATOS

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

DROP SCHEMA IF EXISTS `cipsisdb` ;
CREATE SCHEMA IF NOT EXISTS `cipsisdb` DEFAULT CHARACTER SET utf8 COLLATE utf8_spanish_ci ;
USE `cipsisdb` ;

-----
-- Table `cipsisdb`.`ingeniero`
-----
DROP TABLE IF EXISTS `cipsisdb`.`ingeniero` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`ingeniero` (
  `idingeniero` INT(11) NOT NULL AUTO_INCREMENT,
  `dni` VARCHAR(8) NOT NULL,
  `nombres` VARCHAR(45) NOT NULL,
  `apaterno` VARCHAR(45) NOT NULL,
  `amaterno` VARCHAR(45) NOT NULL,
  `fecha_nacimiento` DATE NOT NULL,
  PRIMARY KEY (`idingeniero`),
  UNIQUE INDEX `dni` (`dni` ASC)
) ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`capitulo`
-----
DROP TABLE IF EXISTS `cipsisdb`.`capitulo` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`capitulo` (
  `idcapitulo` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idcapitulo`)
) ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`especialidad`
-----
DROP TABLE IF EXISTS `cipsisdb`.`especialidad` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`especialidad` (
  `idespecialidad` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NOT NULL,
  `capitulo_idcapitulo` INT(11) NOT NULL,
  PRIMARY KEY (`idespecialidad`),
  INDEX `fk_especialidad_capitulo1_idx` (`capitulo_idcapitulo` ASC),
  CONSTRAINT `fk_especialidad_capitulo1`
    FOREIGN KEY (`capitulo_idcapitulo`)
      REFERENCES `cipsisdb`.`capitulo` (`idcapitulo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
) ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`departamento`
-----
DROP TABLE IF EXISTS `cipsisdb`.`departamento` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`departamento` (
  `iddepartamento` INT(11) NOT NULL AUTO_INCREMENT,

```

```

`nombre` VARCHAR(45) NULL DEFAULT NULL ,
PRIMARY KEY (`iddepartamento`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`provincia`
-----
DROP TABLE IF EXISTS `cipsisdb`.`provincia` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`provincia` (
  `idprovincia` INT(11) NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(45) NOT NULL ,
  `departamento_iddepartamento` INT(11) NOT NULL ,
  PRIMARY KEY (`idprovincia`),
  INDEX `fk_provincia_departamento_idx` (`departamento_iddepartamento` ASC),
  CONSTRAINT `fk_provincia_departamento`
  FOREIGN KEY (`departamento_iddepartamento`)
  REFERENCES `cipsisdb`.`departamento` (`iddepartamento`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`distrito`
-----
DROP TABLE IF EXISTS `cipsisdb`.`distrito` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`distrito` (
  `iddistrito` INT(11) NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(45) NOT NULL ,
  `provincia_idprovincia` INT(11) NOT NULL ,
  PRIMARY KEY (`iddistrito`),
  INDEX `fk_distrito_provincia1_idx` (`provincia_idprovincia` ASC),
  CONSTRAINT `fk_distrito_provincia1`
  FOREIGN KEY (`provincia_idprovincia`)
  REFERENCES `cipsisdb`.`provincia` (`idprovincia`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`pais`
-----
DROP TABLE IF EXISTS `cipsisdb`.`pais` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`pais` (
  `idpais` INT(11) NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`idpais`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`detalles`
-----
DROP TABLE IF EXISTS `cipsisdb`.`detalles` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`detalles` (
  `iddetalles` INT(11) NOT NULL AUTO_INCREMENT ,
  `fotografia` BLOB NULL ,
  `firma` BLOB NULL ,
  `email` VARCHAR(128) NOT NULL ,
  `telefono` VARCHAR(10) NOT NULL ,
  `movil` VARCHAR(10) NOT NULL ,

```

```

`direccion` VARCHAR(45) NOT NULL ,
`genero` ENUM('masculino','femenino') NOT NULL ,
`estado_civil` ENUM('soltero','casado','divorciado','viudo') NOT NULL ,
`ruc` VARCHAR(11) NOT NULL ,
`ingeniero_idingeniero` INT(11) NOT NULL ,
`distrito_iddistrito` INT(11) NULL ,
`pais_idpais` INT(11) NULL ,
`situacion` ENUM('Vivo','Fallecido') NOT NULL ,
`nombre_empresa` VARCHAR(45) NOT NULL ,
`direccion_empresa` VARCHAR(45) NOT NULL ,
`telefono_empresa` VARCHAR(10) NULL ,
`email_empresa` VARCHAR(128) NULL ,
`observacion` VARCHAR(128) NULL ,
PRIMARY KEY (`iddetalles`),
INDEX `fk_detalle_ingeniero1_idx` (`ingeniero_idingeniero` ASC),
INDEX `fk_detalle_distrito1_idx` (`distrito_iddistrito` ASC),
INDEX `fk_detalle_pais1_idx` (`pais_idpais` ASC),
CONSTRAINT `fk_detalle_ingeniero1`
  FOREIGN KEY (`ingeniero_idingeniero`)
  REFERENCES `cipsisdb`.`ingeniero` (`idingeniero`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_detalle_distrito1`
  FOREIGN KEY (`distrito_iddistrito`)
  REFERENCES `cipsisdb`.`distrito` (`iddistrito`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_detalle_pais1`
  FOREIGN KEY (`pais_idpais`)
  REFERENCES `cipsisdb`.`pais` (`idpais`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

```

-- Table `cipsisdb`.`universidad`

DROP TABLE IF EXISTS `cipsisdb`.`universidad` ;

```

CREATE TABLE IF NOT EXISTS `cipsisdb`.`universidad` (
  `iduniversidad` INT(11) NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`iduniversidad`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

```

-- Table `cipsisdb`.`colegiatura`

DROP TABLE IF EXISTS `cipsisdb`.`colegiatura` ;

```

CREATE TABLE IF NOT EXISTS `cipsisdb`.`colegiatura` (
  `idcolegiatura` INT(11) NOT NULL AUTO_INCREMENT ,
  `ingeniero_idingeniero` INT(11) NOT NULL ,
  `condicion` ENUM('tramite','ordinario','vitalicio','temporal') NULL ,
  `modalidad_colegiatura` ENUM('examen','tesis') NULL ,
  `fecha_incorporacion` DATE NULL ,
  `obs` TEXT NULL ,
  `juramentado` ENUM('si','no') NULL ,
  PRIMARY KEY (`idcolegiatura`, `ingeniero_idingeniero`),
INDEX `fk_colegiatura_ingeniero1_idx` (`ingeniero_idingeniero` ASC),
CONSTRAINT `fk_colegiatura_ingeniero1`
  FOREIGN KEY (`ingeniero_idingeniero`)
  REFERENCES `cipsisdb`.`ingeniero` (`idingeniero`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

```

```

-----
-- Table `cipsisdb`.`ingeniero_has_especialidad`
-----
DROP TABLE IF EXISTS `cipsisdb`.`ingeniero_has_especialidad` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`ingeniero_has_especialidad` (
  `ingeniero_idingeniero` INT(11) NOT NULL ,
  `especialidad_idespecialidad` INT(11) NOT NULL ,
  `universidad_iduniversidad` INT(11) NOT NULL ,
  `modalidad_grado` ENUM('tesis', 'actualizacion', 'experiencia') NULL ,
  `fecha_obtencion` DATE NULL ,
  `colegiatura_idcolegiatura` INT(11) NULL ,
  INDEX `fk_ingeniero_has_especialidad_ingeniero1_idx` (`ingeniero_idingeniero` ASC) ,
  INDEX `fk_ingeniero_has_especialidad_especialidad1_idx` (`especialidad_idespecialidad` ASC) ,
  INDEX `fk_ingeniero_has_especialidad_universidad1_idx` (`universidad_iduniversidad` ASC) ,
  INDEX `fk_ingeniero_has_especialidad_colegiatura1_idx` (`colegiatura_idcolegiatura` ASC) ,
  UNIQUE INDEX `ing_esp_unique` (`ingeniero_idingeniero` ASC, `especialidad_idespecialidad` ASC) ,
  UNIQUE INDEX `in_esp_col_unique` (`ingeniero_idingeniero` ASC, `especialidad_idespecialidad` ASC, `colegiatura_idcolegiatura` ASC) ,
  CONSTRAINT `fk_ingeniero_has_especialidad_ingeniero1`
    FOREIGN KEY (`ingeniero_idingeniero` )
    REFERENCES `cipsisdb`.`ingeniero` (`idingeniero` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ingeniero_has_especialidad_especialidad1`
    FOREIGN KEY (`especialidad_idespecialidad` )
    REFERENCES `cipsisdb`.`especialidad` (`idespecialidad` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ingeniero_has_especialidad_universidad1`
    FOREIGN KEY (`universidad_iduniversidad` )
    REFERENCES `cipsisdb`.`universidad` (`iduniversidad` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_ingeniero_has_especialidad_colegiatura1`
    FOREIGN KEY (`colegiatura_idcolegiatura` )
    REFERENCES `cipsisdb`.`colegiatura` (`idcolegiatura` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`ci_sessions`
-----
DROP TABLE IF EXISTS `cipsisdb`.`ci_sessions` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`ci_sessions` (
  `session_id` VARCHAR(40) NOT NULL DEFAULT 0 ,
  `ip_address` VARCHAR(16) NOT NULL DEFAULT 0 ,
  `user_agent` VARCHAR(150) NOT NULL ,
  `last_activity` INT(10) NOT NULL DEFAULT 0 ,
  `user_data` TEXT NOT NULL ,
  PRIMARY KEY (`session_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`login_attempts`
-----
DROP TABLE IF EXISTS `cipsisdb`.`login_attempts` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`login_attempts` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `ip_address` VARCHAR(40) NOT NULL ,
  `login` VARCHAR(50) NOT NULL ,
  `time` TIMESTAMP NOT NULL ,
  PRIMARY KEY (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8

```

```

COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`user_autologin`
-----
DROP TABLE IF EXISTS `cipsisdb`.`user_autologin` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`user_autologin` (
  `key_id` CHAR(32) NOT NULL ,
  `user_id` INT(11) NOT NULL DEFAULT 0 ,
  `user_agent` VARCHAR(150) NOT NULL ,
  `last_ip` VARCHAR(40) NOT NULL ,
  `last_login` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,
  PRIMARY KEY (`key_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`user_profiles`
-----
DROP TABLE IF EXISTS `cipsisdb`.`user_profiles` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`user_profiles` (
  `id` INT(11) NOT NULL ,
  `user_id` INT(11) NOT NULL AUTO_INCREMENT ,
  `country` VARCHAR(20) NOT NULL ,
  `website` VARCHAR(255) NULL DEFAULT NULL ,
  `facebook_id` VARCHAR(255) NULL DEFAULT NULL ,
  `twitter_id` VARCHAR(255) NULL DEFAULT NULL ,
  PRIMARY KEY (`user_id`, `id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`users`
-----
DROP TABLE IF EXISTS `cipsisdb`.`users` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`users` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `username` VARCHAR(50) NOT NULL ,
  `password` VARCHAR(255) NOT NULL ,
  `email` VARCHAR(100) NOT NULL ,
  `activated` TINYINT(1) NOT NULL DEFAULT 1 ,
  `banned` TINYINT(1) NOT NULL DEFAULT 0 ,
  `ban_reason` VARCHAR(255) NOT NULL ,
  `new_password_key` VARCHAR(50) NOT NULL ,
  `new_password_requested` DATETIME NOT NULL ,
  `new_email` VARCHAR(100) NOT NULL ,
  `new_email_key` VARCHAR(50) NOT NULL ,
  `last_ip` VARCHAR(40) NOT NULL ,
  `last_login` DATETIME NOT NULL ,
  `created` DATETIME NOT NULL ,
  `modified` TIMESTAMP NOT NULL ,
  PRIMARY KEY (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`catalogo`
-----
DROP TABLE IF EXISTS `cipsisdb`.`catalogo` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`catalogo` (
  `idcatalogo` INT(11) NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(64) NULL ,
  `KILL_estado` TINYINT(1) NULL ,
  `tipo` ENUM('aportacion','colegiatura','servicio') NULL ,
  `vitalicio` TINYINT(1) NOT NULL DEFAULT 0 ,

```

```

PRIMARY KEY (`idcatalogo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`pago`
-----
DROP TABLE IF EXISTS `cipsisdb`.`pago` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`pago` (
  `idpago` INT(11) NOT NULL AUTO_INCREMENT,
  `fecha` DATE NULL,
  `total` DOUBLE NULL DEFAULT 0,
  `ingeniero_idingeniero` INT(11) NOT NULL,
  `staportacion` DOUBLE NULL,
  `stservicio` DOUBLE NULL,
  `stcolegiatura` DOUBLE NULL,
  PRIMARY KEY (`idpago`),
  INDEX `fk_pago_ingeniero1_idx` (`ingeniero_idingeniero` ASC),
  CONSTRAINT `fk_pago_ingeniero1`
  FOREIGN KEY (`ingeniero_idingeniero`)
  REFERENCES `cipsisdb`.`ingeniero` (`idingeniero`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`pago_item`
-----
DROP TABLE IF EXISTS `cipsisdb`.`pago_item` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`pago_item` (
  `idpagoitem` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(64) NULL,
  `monto` DOUBLE NULL,
  `cant` INT NULL,
  `subtotal` DOUBLE NULL,
  `descuento` DOUBLE NULL,
  `pago_idpago` INT NOT NULL,
  `catalogo_idcatalogo` INT(11) NOT NULL,
  `hash` VARCHAR(43) NULL,
  PRIMARY KEY (`idpagoitem`),
  INDEX `fk_pago_item_pago1_idx` (`pago_idpago` ASC),
  INDEX `fk_pago_item_catalogo1_idx` (`catalogo_idcatalogo` ASC),
  CONSTRAINT `fk_pago_item_pago1`
  FOREIGN KEY (`pago_idpago`)
  REFERENCES `cipsisdb`.`pago` (`idpago`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_pago_item_catalogo1`
  FOREIGN KEY (`catalogo_idcatalogo`)
  REFERENCES `cipsisdb`.`catalogo` (`idcatalogo`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`catalogo_montos`
-----
DROP TABLE IF EXISTS `cipsisdb`.`catalogo_montos` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`catalogo_montos` (
  `idcatalogo_montos` INT(11) NOT NULL AUTO_INCREMENT,
  `monto` DOUBLE NULL,
  `fecha` DATETIME NULL,
  `catalogo_idcatalogo` INT NOT NULL,
  `vigencia` TINYINT(1) NULL,

```

```

`hash` VARCHAR(43) NULL ,
PRIMARY KEY (`idcatalogo_montos` ) ,
INDEX `fk_catalogo_montos_catalogo1_idx` (`catalogo_idcatalogo` ASC) ,
CONSTRAINT `fk_catalogo_montos_catalogo1`
  FOREIGN KEY (`catalogo_idcatalogo` )
  REFERENCES `cipsisdb`.`catalogo` (`idcatalogo` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`cip`
-----
DROP TABLE IF EXISTS `cipsisdb`.`cip` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`cip` (
  `ingeniero_idingeniero` INT(11) NOT NULL ,
  `cip` VARCHAR(16) NULL ,
  `fecha` DATETIME NULL ,
  PRIMARY KEY (`ingeniero_idingeniero` ) ,
  INDEX `fk_cip_ingeniero1_idx` (`ingeniero_idingeniero` ASC) ,
  UNIQUE INDEX `cip` (`cip` ASC) ,
  CONSTRAINT `fk_cip_ingeniero1`
    FOREIGN KEY (`ingeniero_idingeniero` )
    REFERENCES `cipsisdb`.`ingeniero` (`idingeniero` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `cipsisdb`.`familiar`
-----
DROP TABLE IF EXISTS `cipsisdb`.`familiar` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`familiar` (
  `idfamiliar` INT NOT NULL AUTO_INCREMENT ,
  `ingeniero_idingeniero` INT(11) NOT NULL ,
  `nombre` VARCHAR(254) NULL ,
  `parentesco` ENUM('padres','hijos','esposos') NULL ,
  `documento_identidad` VARCHAR(12) NULL ,
  PRIMARY KEY (`idfamiliar` , `ingeniero_idingeniero` ) ,
  INDEX `fk_familiar_ingeniero1_idx` (`ingeniero_idingeniero` ASC) ,
  CONSTRAINT `fk_familiar_ingeniero1`
    FOREIGN KEY (`ingeniero_idingeniero` )
    REFERENCES `cipsisdb`.`ingeniero` (`idingeniero` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `cipsisdb`.`aportaciones`
-----
DROP TABLE IF EXISTS `cipsisdb`.`aportaciones` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`aportaciones` (
  `idaportacion` INT(11) NOT NULL AUTO_INCREMENT ,
  `colegiatura_idcolegiatura` INT(11) NOT NULL ,
  `pago_idpago` INT(11) NOT NULL ,
  `fecha` DATE NULL ,
  PRIMARY KEY (`idaportacion` ) ,
  INDEX `fk_colegiatura_has_pago_pago1_idx` (`pago_idpago` ASC) ,
  INDEX `fk_colegiatura_has_pago_colegiatura1_idx` (`colegiatura_idcolegiatura` ASC) ,
  CONSTRAINT `fk_colegiatura_has_pago_colegiatura1`
    FOREIGN KEY (`colegiatura_idcolegiatura` )
    REFERENCES `cipsisdb`.`colegiatura` (`idcolegiatura` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION ,
  CONSTRAINT `fk_colegiatura_has_pago_pago1`
    FOREIGN KEY (`pago_idpago` )
    REFERENCES `cipsisdb`.`pago` (`idpago` )
    ON DELETE NO ACTION

```

```

ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_spanish_ci;

-----
-- Table `cipsisdb`.`programacion`
-----
DROP TABLE IF EXISTS `cipsisdb`.`programacion` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`programacion` (
  `idprogramacion` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(45) NULL,
  `fecha` DATE NULL,
  `estado` ENUM('creado','ejecutado','cerrado') NOT NULL DEFAULT 'creado',
  PRIMARY KEY (`idprogramacion`))
ENGINE = InnoDB;

-----
-- Table `cipsisdb`.`programacion_has_colegiatura`
-----
DROP TABLE IF EXISTS `cipsisdb`.`programacion_has_colegiatura` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`programacion_has_colegiatura` (
  `programacion_idprogramacion` INT(11) NOT NULL,
  `colegiatura_idcolegiatura` INT(11) NOT NULL,
  PRIMARY KEY (`programacion_idprogramacion`, `colegiatura_idcolegiatura`),
  INDEX `fk_programacion_has_colegiatura_colegiatura1_idx` (`colegiatura_idcolegiatura` ASC),
  INDEX `fk_programacion_has_colegiatura_programacion1_idx` (`programacion_idprogramacion` ASC),
  CONSTRAINT `fk_programacion_has_colegiatura_programacion1`
    FOREIGN KEY (`programacion_idprogramacion`)
    REFERENCES `cipsisdb`.`programacion` (`idprogramacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_programacion_has_colegiatura_colegiatura1`
    FOREIGN KEY (`colegiatura_idcolegiatura`)
    REFERENCES `cipsisdb`.`colegiatura` (`idcolegiatura`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `cipsisdb`.`habilidad`
-----
DROP TABLE IF EXISTS `cipsisdb`.`habilidad` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`habilidad` (
  `idhabilidad` INT(11) NOT NULL AUTO_INCREMENT,
  `asunto` VARCHAR(45) NULL,
  `entidad` VARCHAR(45) NULL,
  `lugar` VARCHAR(45) NULL,
  `fecha` DATE NULL,
  `nombres` VARCHAR(45) NULL,
  `apaterno` VARCHAR(45) NULL,
  `amaterno` VARCHAR(45) NULL,
  `cip` VARCHAR(10) NULL,
  `tipo` ENUM('interno','externo') NOT NULL,
  `idingeniero` INT(11) NULL,
  `fechaaportacion` DATE NULL,
  `sede` VARCHAR(45) NULL,
  `idespecialidad` INT(11) NULL,
  `nombre_especialidad` VARCHAR(45) NULL,
  PRIMARY KEY (`idhabilidad`))
ENGINE = InnoDB;

-----
-- Table `cipsisdb`.`captcha`
-----
DROP TABLE IF EXISTS `cipsisdb`.`captcha` ;

CREATE TABLE IF NOT EXISTS `cipsisdb`.`captcha` (

```

```

`captcha_id` BIGINT(13) UNSIGNED NOT NULL AUTO_INCREMENT ,
`captcha_time` INT(10) UNSIGNED NOT NULL ,
`ip_address` VARCHAR(16) CHARACTER SET 'utf8' COLLATE 'utf8_spanish_ci' NOT NULL DEFAULT '0' ,
`word` VARCHAR(20) CHARACTER SET 'utf8' COLLATE 'utf8_spanish_ci' NOT NULL ,
PRIMARY KEY (`captcha_id`) ,
INDEX `word` (`word` ASC);

USE `cipsisdb` ;

-----
-- Placeholder table for view `cipsisdb`.`v_ingeniero`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_ingeniero` (`idingeniero` INT, `cip` INT, `apaterno` INT, `amaterno` INT,
`nombres` INT, `dni` INT, `fecha_nacimiento` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_especialidad`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_especialidad` (`idespecialidad` INT, `nombre` INT, `capitulo_nombre` INT,
`acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_capitulo`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_capitulo` (`idcapitulo` INT, `nombre` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_pais`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_pais` (`idpais` INT, `nombre` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_departamento`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_departamento` (`iddepartamento` INT, `nombre` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_provincia`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_provincia` (`idprovincia` INT, `nombre` INT, `departamento_nombre` INT,
`acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_distrito`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_distrito` (`iddistrito` INT, `nombre` INT, `provincia_nombre` INT,
`departamento_nombre` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_universidad`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_universidad` (`iduniversidad` INT, `nombre` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_catalogo`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_catalogo` (`idcatalogo` INT, `nombre` INT, `tipo_catalogo` INT, `acciones`
INT);

-----
-- Placeholder table for view `cipsisdb`.`v_catalogo_montos`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_catalogo_montos` (`idcatalogo_montos` INT, `tipo` INT, `monto` INT, `fecha`
INT, `h_idcatalogo` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_ingeniero_especialidad`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_ingeniero_especialidad` (`especialidad_idespecialidad` INT, `ingeniero` INT,
`nombre_especialidad` INT, `nombre_universidad` INT, `h_idingeniero` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_ingeniero_familiar`
-----

```

```

CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_ingeniero_familiar` (`idfamiliar` INT, `nombre` INT, `parentesco` INT,
`documento_identidad` INT, `ingeniero` INT, `h_idingeniero` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_pagos`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_pagos` (`idpago` INT, `fecha` INT, `h_idingeniero` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_programacion`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_programacion` (`idprogramacion` INT, `nombre` INT, `fecha` INT, `estado` INT,
`acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_programacion_colegiatura`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_programacion_colegiatura` (`colegiatura_idcolegiatura` INT, `ingeniero` INT,
`h_idprogramacion` INT, `h_estado` INT, `acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`v_habilidades`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`v_habilidades` (`idhabilidad` INT, `ingeniero` INT, `asunto` INT, `fecha` INT,
`acciones` INT);

-----
-- Placeholder table for view `cipsisdb`.`aportaciones_mensuales`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`aportaciones_mensuales` (`nombre` INT, `capitulo` INT, `cip` INT, `idpago` INT,
`fecha` INT, `mesaportado` INT, `pagoitem` INT, `monto` INT, `catalogo_idcatalogo` INT);

-----
-- Placeholder table for view `cipsisdb`.`reporte1`
-----
CREATE TABLE IF NOT EXISTS `cipsisdb`.`reporte1` (`apaterno` INT, `amaterno` INT, `nombres` INT, `fecha_nacimiento` INT,
`capitulo` INT, `especialidad` INT, `genero` INT, `direccion` INT, `telefono` INT, `email` INT, `provincia` INT, `distrito` INT,
`fecha_incorporacion` INT, `cip` INT);

-----
-- View `cipsisdb`.`v_ingeniero`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_ingeniero` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_ingeniero`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_ingeniero` AS

select
`a`.`idingeniero`,
`b`.`cip`,
`a`.`apaterno`,
`a`.`amaterno`,
`a`.`nombres`,
`a`.`dni`,
`a`.`fecha_nacimiento`,
" as acciones
from ingeniero a left JOIN cip b ON (a.idingeniero = b.ingeniero_idingeniero);

-----
-- View `cipsisdb`.`v_especialidad`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_especialidad` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_especialidad`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_especialidad` AS
select `especialidad`.`idespecialidad` AS `idespecialidad`,
`especialidad`.`nombre` AS `nombre`,
`capitulo`.`nombre` AS `capitulo_nombre`,
"AS `acciones` from (`especialidad` join `capitulo` on((`especialidad`.`capitulo_idcapitulo` =
`capitulo`.`idcapitulo`)));

-----
-- View `cipsisdb`.`v_capitulo`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_capitulo` ;

```

```

DROP TABLE IF EXISTS `cipsisdb`.`v_capitulo`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_capitulo` AS

select idcapitulo, nombre, " as acciones from capitulo;
;

-----
-- View `cipsisdb`.`v_pais`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_pais` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_pais`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_pais` AS

select idpais, nombre, " as acciones from pais;
;

-----
-- View `cipsisdb`.`v_departamento`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_departamento` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_departamento`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_departamento` AS

select iddepartamento, nombre, " as acciones from departamento;
;

-----
-- View `cipsisdb`.`v_provincia`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_provincia` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_provincia`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_provincia` AS
SELECT `provincia`.`idprovincia` AS `idprovincia`,
       `provincia`.`nombre` AS `nombre`,
       `departamento`.`nombre` AS `departamento_nombre`,
       "AS `acciones` FROM (`provincia` JOIN `departamento` ON((`provincia`.`departamento_iddepartamento` =
`departamento`.`iddepartamento`)));
;

-----
-- View `cipsisdb`.`v_distrito`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_distrito` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_distrito`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_distrito`
AS
SELECT
       `a`.`iddistrito` AS `iddistrito`,
       `a`.`nombre` AS `nombre`,
       `b`.`nombre` AS `provincia_nombre`,
       `c`.`nombre` AS `departamento_nombre`,
       "AS `acciones`

FROM distrito a
JOIN provincia b
  ON a.provincia_idprovincia = b.idprovincia
JOIN departamento c
  ON b.departamento_iddepartamento = c.iddepartamento;

-----
-- View `cipsisdb`.`v_universidad`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_universidad` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_universidad`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_universidad` AS

select iduniversidad, nombre, " as acciones from universidad;

```

```

;
-----
-- View `cipsisdb`.`v_catalogo`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_catalogo` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_catalogo`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_catalogo` AS
select `catalogo`.`idcatalogo` AS `idcatalogo`,
       `catalogo`.`nombre` AS `nombre`,
       `catalogo`.`tipo` AS `tipo_catalogo`,
       " AS `acciones`
from `catalogo`;

-----
-- View `cipsisdb`.`v_catalogo_montos`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_catalogo_montos` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_catalogo_montos`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_catalogo_montos` AS
select `catalogo_montos`.`idcatalogo_montos` AS `idcatalogo_montos`,
       `catalogo`.`tipo` AS `tipo`,
       `catalogo_montos`.`monto` AS `monto`,
       `catalogo_montos`.`fecha` AS `fecha`,
       `catalogo`.`idcatalogo` AS `h_idcatalogo`,
       " AS `acciones` from (`catalogo_montos` join `catalogo` on((`catalogo_montos`.`catalogo_idcatalogo` =
`catalogo`.`idcatalogo`)));

-----
-- View `cipsisdb`.`v_ingeniero_especialidad`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_ingeniero_especialidad` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_ingeniero_especialidad`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_ingeniero_especialidad` AS
select
`ingeniero_has_especialidad`.`especialidad_idespecialidad` AS `especialidad_idespecialidad`,
CONCAT_WS(" ", `ingeniero`.`nombres`, `ingeniero`.`apaterno`, `ingeniero`.`amaterno`) AS `ingeniero`,
`especialidad`.`nombre` AS `nombre_especialidad`,
`universidad`.`nombre` AS `nombre_universidad`,
`ingeniero`.`idingeniero` AS `h_idingeniero`,
" AS `acciones`
from (((`ingeniero_has_especialidad` join `especialidad` on((`ingeniero_has_especialidad`.`especialidad_idespecialidad` =
`especialidad`.`idespecialidad`)))
join `ingeniero` on((`ingeniero_has_especialidad`.`ingeniero_idingeniero` = `ingeniero`.`idingeniero`)))
join `universidad` on((`ingeniero_has_especialidad`.`universidad_iduniversidad` = `universidad`.`iduniversidad`)));

-----
-- View `cipsisdb`.`v_ingeniero_familiar`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_ingeniero_familiar` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_ingeniero_familiar`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_ingeniero_familiar` AS
select
`familiar`.`idfamiliar` AS `idfamiliar`,
`familiar`.`nombre` AS `nombre`,
`familiar`.`parentesco` AS `parentesco`,
`familiar`.`documento_identidad` AS `documento_identidad`,
CONCAT_WS(" ", `ingeniero`.`nombres`, `ingeniero`.`apaterno`, `ingeniero`.`amaterno`) AS `ingeniero`,
`ingeniero`.`idingeniero` AS `h_idingeniero`,
" AS `acciones`

from (`familiar` join `ingeniero` on((`familiar`.`ingeniero_idingeniero` = `ingeniero`.`idingeniero`)));

-----
-- View `cipsisdb`.`v_pagos`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_pagos` ;
DROP TABLE IF EXISTS `cipsisdb`.`v_pagos`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_pagos` AS
select

```

```

`pago`.`idpago` AS `idpago`,
`pago`.`fecha` AS `fecha`,
`ingeniero`.`idingeniero` AS `h_idingeniero`,
" AS `acciones`
from (`pago` join `ingeniero` on(`pago`.`ingeniero_idingeniero` = `ingeniero`.`idingeniero`));

-----
-- View `cipsisdb`.`v_programacion`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_programacion`;
DROP TABLE IF EXISTS `cipsisdb`.`v_programacion`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_programacion` AS

select
idprogramacion,
nombre,
fecha,
estado,
" as acciones
from programacion;

-----
-- View `cipsisdb`.`v_programacion_colegiatura`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_programacion_colegiatura`;
DROP TABLE IF EXISTS `cipsisdb`.`v_programacion_colegiatura`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_programacion_colegiatura` AS
select
`programacion_has_colegiatura`.`colegiatura_idcolegiatura` AS `colegiatura_idcolegiatura`,
CONCAT_WS(" ", `ingeniero`.`nombres`, `ingeniero`.`apaterno`, `ingeniero`.`amaterno`) AS `ingeniero`,
`programacion`.`idprogramacion` AS `h_idprogramacion`,
`programacion`.`estado` AS `h_estado`,
" AS `acciones`
from
(
(
(
`programacion_has_colegiatura` join `colegiatura` on
((`programacion_has_colegiatura`.`colegiatura_idcolegiatura`
`colegiatura`.`idcolegiatura`))
)
)
)
join `programacion` on
((`programacion_has_colegiatura`.`programacion_idprogramacion`
`programacion`.`idprogramacion`))
)
join `ingeniero` on
((`colegiatura`.`ingeniero_idingeniero` = `ingeniero`.`idingeniero`));

-----
-- View `cipsisdb`.`v_habilidades`
-----
DROP VIEW IF EXISTS `cipsisdb`.`v_habilidades`;
DROP TABLE IF EXISTS `cipsisdb`.`v_habilidades`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`v_habilidades` AS
select
`habilidad`.`idhabilidad` AS `idhabilidad`,
CONCAT_WS(" ", `habilidad`.`nombres`, `habilidad`.`apaterno`, `habilidad`.`amaterno`) AS `ingeniero`,
`habilidad`.`asunto` AS `asunto`,
`habilidad`.`fecha` AS `fecha`,
" AS `acciones`
from `habilidad`;

```

```

;

-----
-- View `cipsisdb`.`aportaciones_mensuales`
-----
DROP VIEW IF EXISTS `cipsisdb`.`aportaciones_mensuales` ;
DROP TABLE IF EXISTS `cipsisdb`.`aportaciones_mensuales`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`aportaciones_mensuales` AS
SELECT
    CONCAT(`ingeniero`.`nombres`, " ", `ingeniero`.`apaterno`, " ", `ingeniero`.`amaterno`) as nombre,
    `capitulo`.`nombre` as `capitulo`,
    `cip`.`cip` as `cip`,
    `pago`.`idpago` as `idpago`,
    `pago`.`fecha` as `fecha`,
    `aportaciones`.`fecha` as `mesaportado`,
    `pago_item`.`nombre` as `pagoitem`,
    `pago_item`.`monto` as `monto`,
    `pago_item`.`catalogo_idcatalogo` as `catalogo_idcatalogo`

FROM `ingeniero`

LEFT JOIN `pago`
    ON `ingeniero`.`idingeniero` = `pago`.`ingeniero_idingeniero`

INNER JOIN `aportaciones`
    ON `pago`.`idpago` = `aportaciones`.`pago_idpago`

INNER JOIN `pago_item`
    ON `aportaciones`.`pago_idpago` = `pago_item`.`pago_idpago`

INNER JOIN `cip`
    ON `ingeniero`.`idingeniero` = `cip`.`ingeniero_idingeniero`

INNER JOIN `ingeniero_has_especialidad`
    ON `ingeniero`.`idingeniero` = `ingeniero_has_especialidad`.`ingeniero_idingeniero`

INNER JOIN `especialidad`
    ON `especialidad`.`idespecialidad` = `ingeniero_has_especialidad`.`especialidad_idespecialidad`

INNER JOIN `capitulo`
    ON `capitulo`.`idcapitulo` = `especialidad`.`idespecialidad`

WHERE `pago_item`.`catalogo_idcatalogo` IN (10,11)
;

-----
-- View `cipsisdb`.`reporte1`
-----
DROP VIEW IF EXISTS `cipsisdb`.`reporte1` ;
DROP TABLE IF EXISTS `cipsisdb`.`reporte1`;
USE `cipsisdb`;
CREATE OR REPLACE VIEW `cipsisdb`.`reporte1` AS
SELECT
    `ingeniero`.`apaterno` as `apaterno`,
    `ingeniero`.`amaterno` as `amaterno`,
    `ingeniero`.`nombres` as `nombres`,
    `ingeniero`.`fecha_nacimiento` as `fecha_nacimiento`,
    `capitulo`.`nombre` as `capitulo`,
    `especialidad`.`nombre` as `especialidad`,
    `detalles`.`genero` as `genero`,
    `detalles`.`direccion` as `direccion`,
    `detalles`.`telefono` as `telefono`,
    `detalles`.`email` as `email`,
    `provincia`.`nombre` as `provincia`,
    `distrito`.`nombre` as `distrito`,
    `colegiatura`.`fecha_incorporacion` as `fecha_incorporacion`,
    `cip`.`cip` as `cip`

FROM `ingeniero`

LEFT JOIN `cip`
    ON `ingeniero`.`idingeniero` = `cip`.`ingeniero_idingeniero`

```

```
LEFT JOIN `colegiatura`  
  ON `ingeniero`.idingeniero = `colegiatura`.ingeniero_idingeniero  
  
LEFT JOIN `ingeniero_has_especialidad`  
  ON `ingeniero`.idingeniero = `ingeniero_has_especialidad`.ingeniero_idingeniero  
  
LEFT JOIN `especialidad`  
  ON `ingeniero_has_especialidad`.especialidad_idespecialidad = `especialidad`.idespecialidad  
  
LEFT JOIN `capitulo`  
  ON `especialidad`.capitulo_idcapitulo = `capitulo`.idcapitulo  
  
LEFT JOIN `detalles`  
  ON `ingeniero`.idingeniero = `detalles`.ingeniero_idingeniero  
  
LEFT JOIN `distrito`  
  ON `detalles`.distrito_iddistrito = `distrito`.iddistrito  
  
LEFT JOIN `provincia`  
  ON `distrito`.provincia_idprovincia = `provincia`.idprovincia  
;  
  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```



ANEXO C: ENCUESTA

La presente encuesta, que se le presenta estimado usuario es con la finalidad de verificar la satisfacción que Ud. ha alcanzado tras la puesta en funcionamiento de un nuevo sistema, además esto corroborará a que el ejecutor de este proyecto realice el sustento adecuado de la aceptación del nuevo sistema.

Nota: Lea cuidadosamente y marque con la X la alternativa que Ud. considere.

Apellidos y Nombres:.....

1.- Desde su punto de vista, ¿el sistema Web en funcionamiento hace lo que le pide?

- a) Sí
- b) No

2.- Los tiempos de respuesta para cada solicitud o petición al sistema Web, según Ud., ¿cómo lo califica?

- a) Muy bueno
- b) Bueno
- c) Regular
- d) Mala
- e) Muy mala

3.- Al utilizar el sistema Web, ¿puede Ud. acceder o hacer uso de otras aplicaciones y/o programas en su ordenador?

- a) Sí
- b) No

4.- Desde su punto de vista, ¿cómo califica el aprendizaje del sistema Web en funcionamiento? ¿Es fácil de aprender a usar?

- a) Muy bueno
- b) Bueno
- c) Regular
- d) Mala
- e) Muy mala

5.- Cuando hay un corte de fluido eléctrico o pérdida de comunicación con el servidor, el sistema Web se restablece, ¿cómo considera Ud. el tiempo de recuperación del mismo?

- a) Muy bueno
- b) Bueno
- c) Regular
- d) Mala
- e) Muy mala

6.- ¿Considera Ud. que el sistema Web es seguro? (cualquier usuario no puede acceder a él).

- a) Sí
- b) No

7.- Desde su punto de vista, ¿cree Ud. que el sistema Web podrá adaptarse a nuevos requerimientos? ¿Podrán añadirse nuevas opciones?

- a) Sí
- b) No

8.- Desde su punto de vista, ¿cree Ud. que el sistema Web podrá entrar en funcionamiento en otro sistema operativo u otro ordenador?

- a) Sí
- b) No

9.- Desde su punto de vista, ¿cómo califica el grado de facilidad que le ofrece el sistema Web en funcionamiento para realizar sus tareas?

- a) Muy bueno
- b) Bueno
- c) Regular
- d) Mala
- e) Muy mala

10.- Si compara el sistema Web en funcionamiento con el sistema anterior, ¿cómo califica la productividad (número de aportaciones realizadas y número de registros de colegiaturas realizadas por hora) que le ofrece el nuevo sistema?

- a) Muy bueno
- b) Bueno
- c) Regular
- d) Mala
- e) Muy mala

11.- ¿Se siente satisfecho con las funcionalidades que le brinda el sistema en funcionamiento? ¿Funciona en la forma esperada y correcta?

- a) Sí
- b) No

12.- ¿Considera Ud. que el sistema Web en funcionamiento puede ser utilizado por usuarios con conocimientos básicos en computación y no sólo por un usuarios expertos?

- a) Sí
- b) No

13.- ¿La información que muestra el sistema en funcionamiento es completa y precisa?

- a) Sí
- b) No