



***Instituto Superior Minero Metalúrgico***  
***“Dr. Antonio Núñez Jiménez”.***  
***Facultad de Metalurgia - Electromecánica***  
***Moa, Holguín***

***Sistema Automatizado para La Gestión Energética en el  
ISMM.***

***Trabajo de diploma para optar por el título de Ingeniería en Informática***

***Autor (es): Osmani Méndez Rodríguez***

***Tutor (es): Ing. Yadira Romero Rodríguez***  
***Dr. Ing. Secundino Marrero Ramírez***  
***Ing. Roilber Lambert Sánchez***  
***Consultante (es): Irina Laborde Preval***

***Moa, Cuba***  
***Junio, 2008***

## **DECLARACIÓN DE AUTORÍA**

El autor de este trabajo de diplomado certifica su propiedad a favor del Instituto Superior Minero Metalúrgico de Moa, el cual podrá hacer uso del mismo con la finalidad que estime conveniente.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del 2008.

Autor.

Firma: \_\_\_\_\_

Osmani Méndez Rodríguez.

Tutor.

Firma: \_\_\_\_\_

Ing. Yadira Romero Rodríguez

Tutor.

Firma: \_\_\_\_\_

Dr. Secundino Marrero Ramírez

## **Agradecimientos**

*A Mi familia, mis padres; Victor y Emilia y hermanos; chichí, tatita y luís por tanto amor y apoyo en todo momento.*

*A mis tutores por la dedicación y por guiarme durante el desarrollo de este trabajo.*

*A Mis amigos; en especial, Alexander Noa, Yahir Suárez, Paucides Guevara, Jose A. Machado, Nerius Perez, Luis y Yodelqui Adames, por comprenderme y por la preocupación que siempre han tenido por mí.*

*A todos aquellos que de una forma u otra han contribuido con mi formación como futuro profesional y con la realización de este trabajo.*

*A mis compañeros de grupo en especial a Vladimir “El Don” por darme aliento y profesores del departamento en especial a Yadira Romero, Juana Marcia y Virgen Cuza Noa por su apoyo incondicional, y a todos aquellos que me impartieron clases. También a los profesores Roilber Lambert e Irina Laborde por su ayuda incondicional.*

*...y finalmente a todas las personas que se cruzaron en este camino y que me dieron palabras de aliento y apoyo.*

## **Dedicatoria**

*No existen palabras para describir el amor, el cariño, el apoyo y la confianza que he recibido en este largo camino de parte de mis padres, mis hermanos, mi familia en general, de todas aquellas personas que saben apreciar y mi linda sobrina por traer tanta felicidad y amor a la familia, a ellos dedico este trabajo.*

## **Resumen**

El incremento de la demanda, el aumento de los precios de la energía, las restricciones financieras para ampliar la oferta energética, la necesidad de lograr mayor competitividad y prestar mejores servicios con el menor gasto de energía, así como la imperiosa necesidad de protección del medio ambiente, son factores que impulsan actualmente la crecida de la eficiencia energética.

Es por ellos que se pretende diseñar e implementar un sistema informático, cuyo objetivo sea favorecer la eficiencia energética en el ISMM. Dicho sistema permitirá el manejo y la administración de la información de los portadores energéticos, establecerá una comparación del consumo de los recursos energéticos para evaluar dicho insumo. Compartirá una serie de informaciones afiliadas al plan de ahorro, elevando la cultura energética de los usuarios.

Para el desarrollo la investigación se realizó una revisión bibliográfica sobre las aplicaciones Web y las herramientas para la construcción de las mismas. En el presente documento se recoge un resumen del estudio bibliográfico realizado, se presenta la metodología de Ingeniería de Software que se siguió para la construcción del software la cuál se propone como solución de la problemática encontrada. En el trabajo se elabora además un estudio de factibilidad del producto que se obtiene como resultado de la indagación.

## **Summary**

The increment of the demand, the increase of the prices of the energy, the financial restrictions to enlarge the energy offer, the necessity to achieve bigger competitiveness and to lend better services with the smallest energy expense, as well as the imperious necessity of protection of the environment, are factors that speed up the increase of the energy efficiency nowadays.

It is for them that it is sought to design and to implement a computer system whose objective is to favor the energy efficiency in the ISMM. This system will allow the handling and the administration of the information of the energy payees, it will establish a comparison of the consumption of the energy resources to evaluate this consumption. It will share a serie of affiliated informations to the saving plan, elevating the energy culture of the users.

For the realization of the investigation was carried out a bibliographical revision on the applications Web and the tools for the construction of the same ones. Inside this document is picked up a summary of the realized bibliographical study, is presented the Software Engineering methodology that was followed for the product construction which intends as solution of the encountered problem. In the work is also carried out a feasibility study of the computer system that is obtained as research's result.

# Índice

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN .....</b>	<b>5</b>
1.1 INTRODUCCIÓN.....	5
1.2 ESTADO DEL ARTE. ....	5
1.2.1 <i>Conceptos Fundamentales</i> .....	5
1.2.2 <i>La gestión energética en las universidades</i> .....	7
1.2.3 <i>Antecedentes</i> .....	8
1.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES .....	12
1.3.1 <i>Protocolo TCP/IP</i> .....	12
1.3.2 <i>Protocolo HTTP</i> .....	13
1.3.3 <i>La Internet</i> .....	15
1.3.4 <i>Las Aplicaciones Web</i> .....	16
1.3.5 <i>El software libre</i> .....	18
1.3.6 <i>Lenguajes de programación para la Web</i> .....	19
1.3.7 <i>SGBD</i> .....	23
1.3.8 <i>PostgreSQL</i> .....	24
1.3.9 <i>Microsoft SQL Server</i> .....	24
1.3.10 <i>MySQL</i> .....	25
1.3.11 <i>¿Porque usar MySQL?</i> .....	27
1.3.12 <i>Herramientas seleccionadas</i> .....	27
1.3.13 <i>Metodología del software a usar</i> .....	27
1.3.14 <i>El Proceso Unificado de Modelado. RUP</i> .....	28
1.3.15 <i>UML</i> .....	29
1.4 CONCLUSIONES .....	30
<b>CAPÍTULO 2 MODELO DEL DOMINIO .....</b>	<b>31</b>
2.1 INTRODUCCIÓN.....	31
2.2 MODELO DE DOMINIO .....	31
2.2.1 <i>Definición de los conceptos principales del entorno en el que trabajará el sistema</i> .....	32
2.2.2 <i>Representación esquemática del Modelo de Dominio</i> .....	33
2.3 PAQUETES. ....	34
2.4 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA.....	35
2.4.1 <i>Requisitos Funcionales</i> .....	35
2.4.2 <i>Requisitos no Funcionales</i> .....	37
2.5 CONCLUSIONES. ....	38

<b>CAPÍTULO 3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA.</b>	<b>39</b>
3.1    INTRODUCCIÓN.....	39
3.2    ACTORES DEL SISTEMA A AUTOMATIZAR.....	39
3.3    DIAGRAMA DE CASOS DE USO DEL SISTEMA A AUTOMATIZAR. ....	40
3.4    DIAGRAMA DE CLASES DEL DISEÑO .....	47
3.5    PRINCIPIOS DE DISEÑO .....	49
3.6    DISEÑO DE LA BASE DE DATOS.....	50
3.6.1 <i>Modelo lógico de datos</i> .....	50
3.6.2 <i>Modelo físico de datos</i> .....	51
3.7    DIAGRAMA DE SECUENCIA .....	53
3.8    DIAGRAMA DE DESPLIEGUE. ....	54
3.9    DIAGRAMA DE COMPONENTES .....	55
3.10   CONCLUSIONES. ....	55
<b>CAPÍTULO 4 ESTUDIO DE FACTIBILIDAD.....</b>	<b>56</b>
4.1    INTRODUCCIÓN.....	56
4.2    PLANIFICACIÓN. ....	56
4.3    BENEFICIOS TANGIBLES E INTANGIBLES. ....	65
4.4    ANÁLISIS DE COSTOS Y BENEFICIOS.....	65
4.5    CONCLUSIONES. ....	65
<b>CONCLUSIONES GENERALES.....</b>	<b>66</b>
<b>RECOMENDACIONES. ....</b>	<b>67</b>
<b>REFERENCIAS BIBLIOGRÁFICAS Y BIBLIOGRAFÍAS. ....</b>	<b>68</b>
<b>ANEXO 1.....</b>	<b>I</b>
<b>ANEXO 2.....</b>	<b>VI</b>
<b>ANEXO 3.....</b>	<b>VIII</b>
<b>ANEXO 4.....</b>	<b>X</b>
<b>ANEXO 5.....</b>	<b>XXI</b>
<b>ANEXO 6.....</b>	<b>XXIX</b>
<b>ANEXO 7 .....</b>	<b>XXXIV</b>
<b>ANEXO 8.....</b>	<b>XL</b>



## **Introducción**

En el mundo, la generación de energía se ha sumado al deterioro del medio ambiente, fundamentalmente por la quema de combustibles fósiles. La mayoría de las acciones que realiza actualmente la humanidad producen un daño o deterioro a las condiciones iniciales de la superficie terrestre, la suma de estas acciones negativas acarrearán consecuencias generalizadas sobre todo el planeta.

La energía posibilita y facilita toda la actividad humana. Las diferentes fuentes y sistemas de producción y uso de la energía utilizadas por el hombre han marcado las grandes etapas en el desarrollo de la sociedad humana, dependiendo el curso de éste de las elecciones energéticas realizadas en cada momento. En el transcurso del tiempo el hombre pasó del empleo de su fuerza muscular al uso de diversas fuentes para satisfacer sus necesidades, el empleo del fuego, la utilización de la tracción animal, y finalmente, en rápida sucesión, el dominio de las tecnologías del carbón, del petróleo y el gas natural, y la producción y uso del vapor y la electricidad. Desde esta perspectiva, la historia de la humanidad no ha sido más que la historia del control de ésta sobre las fuentes y tecnologías energéticas, llegando al esquema energético global actual, el que descansa en la utilización de los combustibles fósiles; combustibles que son no renovables, contaminantes en alto grado, que están concentrados en pocas regiones de la tierra, en manos de grandes compañías transnacionales y que son utilizados de forma muy ineficiente. El inicio del tercer milenio representa para la humanidad la encrucijada de una nueva elección energética, frente al agotamiento de los combustibles fósiles por una parte, pero sobre todo, por la amenaza de una catástrofe ecológica, al rebasarse los límites de la capacidad del planeta para asimilar su impacto. Los procesos de producción y uso de la energía constituyen la causa fundamental del deterioro ambiental. El previsible agotamiento de los combustibles fósiles y el daño irreversible que se ocasiona al medio ambiente, exige la adopción de nuevas estrategias en materia de energía, como base de un modelo de desarrollo sostenible, que permita satisfacer las necesidades energéticas de la generación actual y preservar las posibilidades para

que las futuras generaciones puedan también encontrar soluciones para satisfacer las suyas.

La conservación de la energía puede considerarse como una fuente de energía alternativa. Tras la revolución energética se ha tomado aun más serio el control y el uso equitativo de los recursos energéticos para buscar una eficiencia energética, que significa la eficacia en la producción, distribución y servicios con el menor gasto de energía posible y costo.

El ISMM no alcanza la máxima eficiencia y eficacia de la administración de los portadores energéticos, al lograr esto de forma sistemática disminuye sustancialmente la contaminación ambiental y garantiza la utilización racional de los recursos energéticos no renovables contribuyendo también a la economía. Por lo que se necesita de un sistema informático que favorezca la gestión de la información de los portadores y contribuya a la eficiencia energética.

Por ende se formuló el siguiente **problema**: En la gestión y control de la información de los portadores energéticos en el ISMM es deficiente el manejo de dicha información.

Para darle solución a la problemática se planteó el siguiente **objetivo general**: Diseñar un sistema informático para favorecer la gestión de la información de los recursos energéticos permitiendo así el uso racional y el control estricto de los mismos en el ISMM.

Por tanto el **objeto de estudio** del trabajo, es el proceso de gestión universitaria asociada al uso y control de los portadores energéticos.

De aquí se deriva que **el campo de acción** se encierra en el uso eficiente, control y destino de los portadores energéticos en el ISMM.

Este problema se enmarca en los siguientes **objetivos específicos**:

1. Caracterizar el manejo actual de la información en el proceso de gestión energética.
2. Analizar los principales fundamentos teóricos y metodológicos de la investigación.
3. Diseñar Base de Datos para el manejo de la gestión de los portadores energéticos.
4. Crear un espacio de información especializado para la gestión de la información asociada al tema del ahorro de la energía.

Como **hipótesis** de la investigación se define la siguiente: Si se diseña un sistema informático para la automatización de la gestión de la información y administración de los portadores energéticos proporcionará y facilitará la eficiencia energética en el ISMM y permitirá la obtención de informaciones referente al tema del ahorro energético.

Para resolver la problemática planteada, se proponen las siguientes tareas:

1. Revisión bibliográfica y búsqueda de información.
2. Análisis de Factibilidad.
3. Diseño de la arquitectura del sistema.
4. Programación de la interfaz

La significación práctica del presente trabajo radica en establecer un control más seguro y eficiente de los portadores energéticos, así como la planificación y reducción del consumo, obtener y comparar el estado actual de los indicadores de consumo tan sensibles, además de contar con un grupo de información referente al plan de ahorro del país y del ISMM.

El presente trabajo consta de introducción, cuatro capítulos, conclusiones, recomendaciones, bibliografía y anexos:

El **Capítulo 1** trata sobre la fundamentación teórica del trabajo. Aquí se hace una descripción de los principales conceptos asociados al dominio del problema; se analizan otras soluciones posibles al problema que nos ocupa, también se tratará sobre los sistemas automatizados existentes que tienen relación con nuestro trabajo. Se abordan las tendencias y tecnologías actuales a considerar. Se fundamenta la selección de las herramientas utilizadas para el desarrollo del trabajo.

En el **Capítulo 2**, Análisis, se realiza un análisis del entorno en que trabajará el sistema explicando el flujo de proceso y representándolo en un modelo de dominio haciendo uso de la metodología expuesta en el capítulo inicial.

En el **Capítulo 3**, diseño e implementación, también utilizando la metodología mencionada anteriormente se diseña la arquitectura del sistema a desarrollar y se define las funcionalidades de dicho sistema.

En el **Capítulo 4**, estudio de factibilidad, se siguen una serie de pasos encaminados a conformar un estudio de factibilidad del sistema. Se reflejan los cálculos intermedios así como las tablas con datos de la aplicación y resultados finales. Se analizan los beneficios tangibles e intangibles y los costos del desarrollo de la propuesta.

Para concluir se muestran las conclusiones a las que se arribaron, las recomendaciones que se proponen, la bibliografía utilizada y anexos con información necesaria sobre el trabajo.

## **Capítulo 1 Fundamentación**

### **1.1 Introducción**

En este capítulo se definen los conceptos fundamentales para el desarrollo de aplicaciones Web, así como sus principales ventajas, entre otros. Se exponen las potencialidades del uso del software libre, las diferencias entre aplicación Web y sitios Web, las principales características de los lenguajes de programación Web y los sistemas de gestión de base de datos. Además se abordará brevemente sobre la estructura energética del ISMM. Se mencionan algunos sistemas relacionados con la cuestión energética.

Todos estos elementos constituyen el soporte teórico de la labor, que contribuye al logro de un mejor entendimiento de la situación problemática y de una valoración adecuada para su solución.

### **1.2 Estado del Arte.**

#### **1.2.1 Conceptos Fundamentales.**

##### **Gestión energética.**

La gestión energética puede definirse como el análisis, planificación y toma de decisiones con el fin de obtener el mayor rendimiento posible de la energía que se necesita; esto es, lograr un uso más racional de la energía, que permita reducir el consumo de la misma sin disminuir la calidad de los servicios y la producción.

Frecuentemente se considera el ahorro o conservación de energía como una cuestión a corto plazo. Es también opinión muy generalizada que con aplicar una buena administración del uso de la energía y otras técnicas básicas, se ha hecho todo lo que cabía y esperar hasta que nuevas tecnologías aporten nuevas soluciones. No obstante, tales convicciones son erróneas ya que la gestión energética es una tarea a mediano y largo plazo que debe significar; implantar y controlar la forma en que cualquier empresa use o planifique de forma más racional sus recursos energéticos.

La **gestión energética** debe verse ligada a otros aspectos muy importantes que dan lugar a una comprensión más clara del entorno o del término de gestión energética como son la eficiencia, la eficacia, la efectividad y la eficiencia energética.

### **El TCC.**

Cada portador energético tiene un factor de conversión para tener una equivalencia. El TCC, toneladas de combustible de convencional o equivalente, se determina multiplicando el gasto de los portadores por sus respectivos factores de conversión, el sistema también debe ser capaz de realizar este cálculo.

<u>Factores de conversión</u>	
<u>Portador</u>	<u>Factor</u>
Diesel	1.0534
Gasolina	1.0971
Fuel-oil	0.9903
Gas licuado	1.163
Electricidad	0.3817
Lubricantes	-

**Tabla 1.1** .Factores de conversión tcc por portador.

### **Principio de Pareto:**

El Principio de Pareto [3] afirma que en todo grupo de elementos o factores que contribuyen a un mismo efecto, unos pocos son responsables de la mayor parte de dicho efecto.

### **Análisis de Pareto:**

El Análisis de Pareto es una comparación cuantitativa y ordenada de elementos o factores según su contribución a un determinado efecto.

El objetivo de esta comparación es clasificar dichos elementos o factores en dos categorías: Las "Pocas Vitales" (los elementos muy importantes en su contribución) y los "Muchos Triviales" (los elementos poco importantes en ella).

Las Tablas y Diagramas de Pareto son herramientas de representación utilizadas para visualizar el Análisis de Pareto. El Diagrama de Pareto es la representación gráfica de la Tabla de Pareto correspondiente.

### 1.2.2 La gestión energética en las universidades.

Actualmente todo el proceso de la gestión energética en las instituciones universitarias se realiza manualmente y si existe algún sistema es ineficiente al hablar de una gestión mas detallada o del soporte del sistema, de este aspecto se comentará en próximos epígrafes, dicho proceso se recrudesció a raíz de la revolución energética en cuba con el fin de conseguir una mejor eficiencia energética en nuestro país. Este asunto es algo complejo y amplio, puesto que es una gran ventaja la automatización de proceso energético de las universidades.

#### Estructura de la gestión energética en el ISMM.

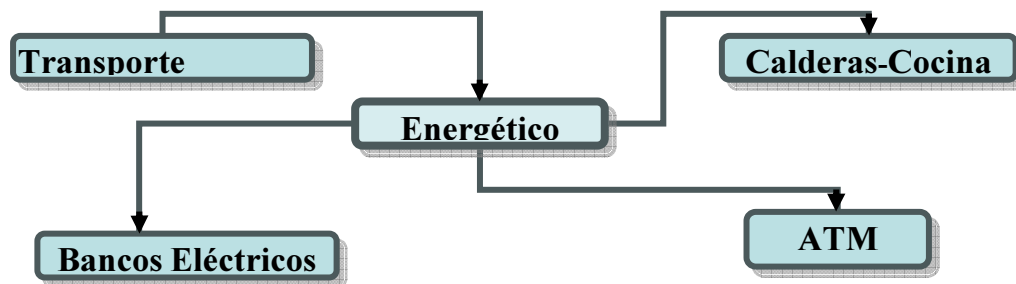


Fig. #1.1 Estructura de la gestión energética del ISMM.

De los bancos eléctricos, docente, residencia y transporte, se toma el valor de consumo de la electricidad mostrado por los contadores. Del almacén ATM se toman los valores de entrada, salida y existencia del gas y los lubricantes. De las calderas, donde se encuentra el depósito del fuel-oil, se toman los valores de consumo y la

existencia del producto. Desde transporte se recibe al final de cada mes los valores de consumo de los combustibles.

Para el agua, aunque no es un portador energético se contabiliza similar a los portadores por su importancia y costo para el ISMM. Se sabe la cantidad de litros suministrados al tanque diario y la cantidad de horas de bombeo al día, por tanto se conoce el consumo del líquido así como el índice de consumo durante todo el mes.

### **1.2.3 Antecedentes.**

En el mundo existen diversos sistemas relacionados con la gestión energética, mucho de estos sistemas son de carácter informativos:

CALFE: es una plataforma Web para gestionar el Cálculo de factura eléctrica.

SNIE: Sistema Nacional de Información de la Energía, es un sitio Web que muestra información sobre proveedores de productos, tecnologías y servicios para el sector energético en Cuba.

También tenemos otros sistemas implementados en Cuba. El Paquete de Soluciones Informáticas para La Gestión Gubernamental: Revolución Energética en Cuba [1], desarrollado en La Universidad de las Ciencias Informáticas, se enfoca en la construcción de un robusto sistema de gestión de información y recursos, que empleando las Tecnologías de la Informática y las Comunicaciones, garantiza la informatización de registros y procedimientos que hoy se llevan a cabo en su mayoría de forma manual respondiendo a su vez a los esfuerzos en la informatización de La Sociedad Cubana.



Los objetivos de este sistema se describen a través de sus cinco partes o módulos fundamentales:

1. La primera es la **informativa** donde está previsto a través de un conjunto de mecanismos y empleando las potencialidades de la Red de Redes (Internet) dar a conocer la verdad, los principales objetivos trazados así como las diferentes líneas de trabajo de La Revolución Energética en Cuba.
2. La segunda es la de **gestión**, la cual cuenta con un grupo de módulos que se encargan de manipular toda la información que fluye en la tareas llevadas a cabo por la dirección del país, lo cual se logra a través de un conjunto de registros de información que una vez estando en bases de datos relacionales pues pueden ser sin lugar a dudas procesadas para obtener de forma tabulada y en tiempo record gran cantidad de información importante para la continuidad y control de las principales acciones llevadas a cabo, que hoy en la mayoría de los casos se realiza manualmente. La incorporación de un **Sistema para la Gestión de Rutas Mínimas para el ahorro de combustible (GESRUM)** a la solución eleva considerablemente el aporte económico de este trabajo a la sociedad cubana y la inclusión de **GESCON (Gestión de Conocimientos)** que plantea una arquitectura enfocada a la gestión de conocimiento, basada en la nueva concepción de la Ingeniería del Conocimiento, sirve de apoyo directo la concepción de reutilización de experiencias y la sociedad del conocimiento enfocado en el desarrollo sostenido de la Revolución Cubana. Y además la inclusión de un **Censor de Videos Automáticos (CAVIDIG)** contribuye a través del reconocimiento de patrones identificar - entre otras - las intervenciones del comandante en jefe entre las mesas redondas informativas relacionadas con la Revolución Energética en Cuba.
3. La componente **atención al ciudadano** constituye otra de las unidades fundamentales de este trabajo, es un espacio que permite que todo aquel ciudadano que necesite realizar una consulta a un funcionario del gobierno u otro personal ya no tenga que trasladarse desde un lugar a otro -recorriendo grandes distancias- sino, que empleando uno de los servicios que implementa

la aplicación de la Revolución Energética en Cuba denominado “*Tele-Inquietudes*” podrá hacer sus preguntas y obtener las respuestas.

4. A través de un **entorno virtual de aprendizaje** para la Educación a Distancia, **CEDIST (Centro de Educación a Distancia)** se ofrece un aspecto integrante en toda la obra de nuestra Revolución: la formación del pueblo. CEDIST que sienta sus bases en un flexible modelo pedagógico auditado por un sistema de calidad propio empleando un conjunto de registros y procedimientos de acorde a como se establece en los conocidos modelos de calidad (CMM, CMMI, etc.) posibilita la interacción de los interesados con especialistas en temas energéticos los cuales a través de dicha plataforma de aprendizaje impartirán cursos al pueblo, fomentando la cultura energética factor imprescindible para el éxito.
5. Toda la actividad en la **Revolución Energética en Cuba** involucra recursos humanos y recursos materiales que deben ser cuidadosamente gerenciados en las diferentes instancias de las instituciones involucradas y en franca lucha contra los hechos delictivos y manifestaciones de corrupción, incrementando la eficacia de los controles y auditorías a todos los niveles institucionales, por lo que se incluye una componente de **control interno empresarial** y que por tanto trasciende este proyecto de Revolución Energética al resto de la actividad empresarial cubana de estos tiempos. **AUDINTEC (Auditoría y Control Interno en al Empresa)** se inscribe dentro de los proyectos de tipo interno, toda vez que penetra en la relación del control económico en su más amplia concepción dentro de la empresa, desde la empresa y hacia la empresa. El proyecto informático integrado **AUDINTEC** aporta al pensamiento y a la gestión económica de la empresa cubana y de las entidades gubernamentales asociadas herramientas para elevar el control, la eficiencia y el desempeño en su actividad económica. Se fundamenta en la arquitectura de un sistema de aplicaciones informáticas capaces de informatizar procesos de entrega y verificación de la información solicitada en los diversos procesos del control de la actividad económica.

**La arquitectura** física del paquete comprende un servidor o repositorio de datos donde se encuentran centralizados los datos obtenidos en cada uno de los procesos: repositorio “master” o repositorio “central”, además de un conjunto de servidores de datos locales que estarán ubicados por zonas de acuerdo a las necesidades. Estos servidores diarios harán replica hacia arriba al servidor master de igual forma funciona hacia abajo en caso de necesitar actualizar determinados registros de datos. Se necesitará un servidor Web que sirva de “host” a la aplicación del portal así como al Centro de Educación a Distancia. En el caso del resto de los componentes la instalación es en la estación de trabajo cliente, preferentemente sobre Windows 2000 o superior.

Teniendo en cuenta lo expuesto anteriormente podemos observar que este sistema es complicado. Requiere de hardware potente, un servidor master y una serie de servidores clientes acorde a los requerimientos, esto necesita una alta velocidad de la red para el tráfico de la información entre el servidor master y los servidores clientes. No tiene como objetivo específico la gestión y control estricto de los portadores energéticos para el uso eficiente y planificación para los mismos, sino que es mayoritariamente informativo.

### **Antecedentes en el ISMM.**

Por otra parte en el departamento de contabilidad del ISMM existe un pequeño sistema diseñado en el gestor de bases de datos Access. Este sistema almacena o realiza una determinada gestión de los portadores energéticos manipulados por el ISMM. Al estar diseñado en Access no permite acceder desde cualquier puesto de trabajo sin transportarlo, no es seguro, no cumple con la filosofía de software libre, no permite visualizar el comportamiento del consumo de un determinado portador, ni la comparación entre los diferentes cursos o años y no brinda informaciones con relación al asunto de la conservación de la energía.

Se determinó diseñar un sistema que se ajuste a las necesidades y requerimientos específicos del centro. Con el presente trabajo se pretende desarrollar una aplicación Web que permita el control y uso preciso de los portadores energéticos, así como la planificación para los mismos, con un entorno agradable, flexible,

adaptable y sencillo permitiendo un acceso rápido y seguro a la información desde cualquier puesto de trabajo, que cumpla con la ética de software libre, compartiendo información de interés asociado al tema energético, entre otros.

### **1.3 Tendencias y tecnologías actuales**

#### **1.3.1 Protocolo TCP/IP.**

En realidad no es un protocolo sino la familia de protocolos TCP/IP [15] (*Transmission Control Protocol (TCP)* y el *Internet Protocol (IP)*.) de Internet, que permiten la transmisión de información en redes de computadoras.

Algunos de los motivos de su popularidad son:

- Independencia del fabricante.
- Soporta múltiples tecnologías.
- Puede funcionar en maquinas de cualquier tamaño.
- Estándar de EEUU desde 1983.

La arquitectura de un sistema en TCP/IP tiene una serie de metas:

- La independencia de la tecnología usada en la conexión a bajo nivel y la arquitectura del ordenador.
- Conectividad Universal a través de la red.
- Reconocimientos de extremo a extremo.
- Protocolos estandarizados

Este protocolo constituye la base sobre al que se soporta Internet y es utilizado para unir computadoras que pueden tener diferentes sistemas operativos y que pueden ser tanto PC como minicomputadoras o computadoras centrales. Su característica principal es que se dividen las tareas para un mejor funcionamiento: La tarea de IP es llevar los datos por paquetes de un sitio a otro. Las computadoras que encuentran las vías para llevar los datos de una red a otra (denominadas enrutadores) utilizan IP para trasladar los datos. En resumen IP mueve los paquetes de datos a granel, mientras TCP se encarga del flujo y asegura que los datos estén correctos.

Como otros tantos protocolos, este posee ventajas y desventajas que lo caracterizan: está diseñado para enrutar y tiene un grado muy elevado de fiabilidad, es adecuado

para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores Web. Es compatible con las herramientas estándar para analizar el funcionamiento de la red.

Un inconveniente de TCP/IP es que es más difícil de configurar y de mantener que NetBEUI o IPX/SPX; además es algo más lento en redes con un volumen de tráfico medio bajo. Sin embargo, puede ser más rápida en redes con un volumen de tráfico grande donde haya que enrutar un gran número de marcos.

El protocolo TCP/IP se puede utilizar en grandes redes empresariales como por ejemplo en campus universitarios o en complejos empresariales en donde utilizan muchos enrutadores y conexiones a mainframe o a ordenadores UNIX. También se puede utilizar en redes pequeñas en donde 100 ó 200 estaciones de trabajo funcionando con Windows acceden a servicios de intranet o Internet mediante un servidor NT que ofrezca servicios Web a través del servidor de información de Internet (IIS) de Microsoft.

### **1.3.2 Protocolo HTTP.**

Este protocolo, el http [15], que es acrónimo de *HyperText Transfer Protocol* y que es el lenguaje que emplean los servidores de Web. Este protocolo es el que controla la transferencia de datos en la World Wide Web WWW y proporciona un vehículo de entrega para las imágenes, gráficos, video, hipertexto u otros datos en la Web. Está basado en el modelo cliente-servidor y la comunicación generalmente se lleva a cabo sobre una conexión TCP/IP.

Mediante HTTP los clientes y los servidores determinan de forma dinámica el formato de los documentos, lo que permiten que utilicen formato de datos no estándar para el intercambio de datos. Si el receptor no tiene un modo de ver o acceder a los datos, puede descargar un programa complemento que le permita recibir el contenido. HTTP se basa en sencillas operaciones de solicitud-respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un **objeto o recurso** sobre el que actúan.

Las principales características del protocolo HTTP son:

Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.

- Permite la transferencia de objetos multimedia.
- Existen tres verbos básicos (hay más, pero por lo general no se utilizan) que un cliente puede utilizar para dialogar con el servidor: **GET**, para recoger un objeto, **POST**, para enviar información al servidor y **HEAD**, para solicitar las características de un objeto (por ejemplo, la fecha de modificación de un documento HTML).
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto.
- No mantiene estado.

El hecho de que sea un protocolo sin estado significa que, en principio, un servidor de HTTP carece de medios para relacionar información concerniente a una petición con otra petición anterior o posterior. Los datos de la respuesta se basan exclusivamente en la información que el cliente envía en la petición. El protocolo HTTP no conoce a la persona a quien está enviando una página ni cuántas páginas le haya podido enviar, incluso aunque nos hayamos conectado hace escasamente algunos segundos, ya que cada petición de página se procesa independientemente.

La ventaja de los protocolos que no guardan el estado de la comunicación es que se requiere menor número de recursos por parte del servidor, mientras que se exigen en cambio en el lado del cliente. Sin embargo, su desventaja es que el cliente necesita comunicar al servidor información suficiente en cada petición para poder recibir la respuesta correcta en cada caso.

Este mecanismo tan sencillo es el que permite que exista una gran variedad de aplicaciones que sean escritas para entornos basados en la Web.

### **1.3.3 La Internet.**

Internet [15] es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP (Protocolo de Control de Transmisión(TCP) y Protocolo de Internet (IP)) y garantiza que redes físicas heterogéneas funcionen como una red lógica única, de alcance mundial. En otras palabras la Internet es un conjunto de redes: redes de ordenadores y equipos físicamente unidos mediante cables que conectan puntos de todo el mundo. Estos cables se presentan en muchas formas, desde cables de red local (varias máquinas conectadas en una oficina o campus) a cables telefónicos convencionales, digitales y canales de fibra óptica que forman las vías principales. Esta gigantesca Red se difumina en ocasiones porque los datos pueden transmitirse vía satélite, o a través de servicios como la telefonía celular, o porque a veces no se sabe muy bien a dónde está conectada.

#### ***Historia de la Internet.***

Internet se inició en los Estados Unidos. A finales de los años 60, la ARPA [15] (Agencia de Proyectos de Investigación Avanzados) del Departamento de Defensa definió el protocolo TCP/IP con intención de facilitar la conexión de diversos tipos de ordenadores, principalmente para temas de investigación. Ahora el TCP/IP sirve para garantizar la transmisión de los paquetes de información entre lugares remotos, siguiendo cualquier ruta disponible.

En 1975, ARPAnet [6] (Advanced Research Projects Agency Network) comenzó a funcionar como red, sirviendo como base para unir centros de investigación militares y universidades, y se trabajó en desarrollar protocolos más avanzados para diferentes tipos de ordenadores y cuestiones específicas. En 1983 se adoptó el TCP/IP como estándar principal para todas las comunicaciones, y en 1990 desapareció ARPAnet para dar paso junto a otras redes TCP/IP a Internet. Por aquel entonces también comenzaron a operar organizaciones privadas en la red. Poco a poco, todos los fabricantes de ordenadores personales y redes incorporaron el TCP/IP en sus sistemas operativos, de modo que en la actualidad cualquier equipo está listo para conectarse a Internet. Internet une muchas redes, incluyendo como

más importantes la que proporciona acceso a los grupos de noticias (Usenet), que data de 1979 y (conceptualmente) la todavía más conocida World Wide Web, desde principios de los 90.

#### **1.3.4 Las Aplicaciones Web.**

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

En contraste, las aplicaciones Web [7] [15] generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores Web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página Web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

En fin, una aplicación Web es un conjunto de páginas Web estáticas y dinámicas, sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet.

#### **Estructura:**

Aunque muchas variaciones son posibles, una aplicación Web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador Web es la primera capa, un motor usando alguna tecnología Web dinámica (ejemplo: PHP, HTML, Java Servlets o ASP) es la capa de en medio, y una base de datos como última capa. El navegador Web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

#### **Ventajas:**

A las aplicaciones Web también se les denominan aplicaciones de servidor (server-side applications) porque a diferencia de la mayoría de los programas que estamos



acostumbrados a utilizar, los cálculos y procesamiento de la información se realizan en otro computador (el servidor). Nuestro computador, denominado cliente, se comunica con el servidor enviando y recibiendo la información por medio de un navegador Web. Como podrá imaginar, un mismo servidor puede interactuar con gran cantidad de clientes al mismo tiempo, por lo cual, todos éstos podrán estar compartiendo los mismos datos y utilizando una misma aplicación desde distintos lugares y sin más requerimientos que un navegador Web y una conexión a Internet (u otra red compartida). A continuación se mencionan algunas ventajas.

- Una empresa puede migrar de sistema operativo o cambiar el Hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor. Correo electrónico interno.
- No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo un computador con un buen navegador web.
- Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión a Internet.
- Actualizar o hacer cambios en el Software es sencillo y sin riesgos de incompatibilidades. Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. El proceso es rápido y limpio.
- Al funcionar en un navegador, se requiere un conocimiento básico de informática para utilizar una aplicación Web.
- Desarrollo barato, sencillo y rápido.
- Acceso ubicuo, sin necesidad de distribución e, idealmente, con pocos requerimientos técnicos.
- Datos centralizados y fácil integración de datos de múltiples fuentes.
- Permiten el desarrollo de comunidades que dan valor a las aplicaciones (software social).

### **1.3.5 El software libre**

La implantación y adquisición en nuestro país del software libre es una de nuestras principales metas, se pretende a la comunidad informática de que con software libre se le pueden dar solución a distintos problemas que nos encontramos hoy en día sin necesidad de invertir en software y sistemas operativos propietarios.

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido que los vendedores de los grandes computadores de la época aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia. Por el 1971, cuando la informática todavía no había tomado gran auge, las personas que hacían uso de ella, en ámbitos universitarios y empresariales, creaban y compartían el software sin ningún tipo de restricciones. Con la llegada de la década del 80 la situación empezó a cambiar. Las computadoras más modernas comenzaban a utilizar sistemas operativos privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software.

En caso de que algún usuario o programador encontrase algún error en la aplicación, lo único que podía hacer era darlo a conocer a la empresa desarrolladora para que esta lo solucionara. Aunque el programador estuviese capacitado para solucionar el problema y lo deseara hacer sin pedir nada a cambio, el contrato le impedía que mejorase el software. En 1984 Richard Stallman comenzó a trabajar en el proyecto GNU [4], y un año más tarde fundó la Free Software Foundation (FSF) [2]. Stallman introdujo una definición para free software y el concepto de "copyleft", el cual desarrolló para dar a los usuarios libertad y para restringir las posibilidades de apropiación del software.

**Software libre** [15] (free software) es la denominación del software que respeta la libertad de los usuarios. Según La Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.

El software libre suele estar disponible gratuitamente, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo no hay que asociar software libre a "software gratuito", ya que, conservando su carácter de libre, puede ser distribuido comercialmente . Análogamente, el "software gratis" o "gratuito" incluye en algunas ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquél que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquél cuyo autor lo dona a la humanidad o cuyos derechos de autor han acabado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es dominio público.

### ***1.3.6 Lenguajes de programación para la Web.***

En la programación de para la Web se habla de dos tipos de lenguajes. Los lenguajes del lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales

son simplemente incluidos en el código HTML), son aquellos que pueden ser directamente "digeridos" por el navegador. Cada uno de estos lenguajes tiene sus ventajas y sus inconvenientes. Así, por ejemplo, un lenguaje de lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con scripts de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas. Inversamente, un lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

Por otra parte, los scripts son almacenados en el servidor quien los ejecuta y traduce a HTML por lo que permanecen ocultos para el cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página de servidor. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Este hecho puede resultar a todas luces una forma legítima de proteger el trabajo intelectual realizado.

Existe una multitud de lenguajes concebidos o no para Internet, cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para desarrollar distintas aplicaciones. La versatilidad de un lenguaje está íntimamente relacionada con su complejidad. Un lenguaje complicado en su aprendizaje permite en general el realizar un espectro de tareas más amplio y más profundamente. Es por ello que a la hora de elegir el lenguaje que queremos utilizar tenemos que saber claramente qué es lo que queremos hacer y si el lenguaje en cuestión nos lo permite o no. En el dominio de la red, los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP, entre otros.

## **ASP**

ASP [12] (Active Sever Page) es la Tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript [9] (Javascript de Microsoft). Este lenguaje es empleado fundamentalmente en servidores que funcionan con sistemas basados en Windows NT [12], aunque también se puede utilizar en un PC con Windows 98 [12] si se instala un servidor denominado Personal Web Server. En sistemas Linux se puede utilizar la Tecnología ASP instalando un componente denominado Chilisoft, aunque será mejor trabajar sobre el servidor Web para el que está pensado: Internet Infomation Server.

Con las páginas ASP se pueden realizar muchos tipos de aplicaciones distintas, permite acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor. También se tiene la posibilidad de comprar componentes ActiveX fabricados por distintas empresas de desarrollo de software que sirven para tareas múltiples, como el envío de correo, generar gráficas dinámicamente, etc.

Actualmente se ha presentado ya la segunda versión de ASP, el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona.

ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro.

## **PHP**

PHP [14] (*Personal Home Page*) es el acrónimo de Hipertext Preprocesor, su sintaxis está inspirada en el lenguaje C. Es un lenguaje de programación del lado del servidor, libre, gratuito, independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de la red al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con ASP, pero con las ventajas de ser más rápido, libre y seguro. Cualquiera puede descargar a través de la página principal de PHP y de manera gratuita, un módulo que hace que nuestro servidor Web comprenda los scripts realizados en este lenguaje. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor Web y su código fuente se puede compilar para la mayoría de los sistemas operativos existentes o descargarlo ya compilado. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

PHP es más rápido que ASP dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las Tecnologías implicadas en una página ASP.

Algunas de las más importantes capacidades de PHP son: compatibilidad con los sistemas de gestión de bases de datos más comunes, como MySQL[13], PostgreSQL[8], SQLServer [12], Oracle, Informix, conexiones ODBC. Incluye funciones para el envío de correo electrónico, upload de archivos, creación dinámicamente de imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales que se enriquece a diario con las contribuciones de la comunidad de desarrolladores del lenguaje.

Estos dos lenguajes resultan bastante útiles para la explotación de bases de datos y su aprendizaje resulta accesible para una persona que no sea especialista en programación. Cualquiera de ellos resultaría la opción ideal a la hora de hacer evolucionar un sitio Web realizado en HTM.

### **1.3.7 SGBD**

Los Sistemas de gestión de base de datos (SGBD) [15] son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla, ordenada y segura un conjunto de datos que posteriormente se convertirán en información. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

El funcionamiento del SGBD está muy interrelacionado con el del Sistema Operativo, especialmente con el sistema de comunicaciones. El SGBD utilizará las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (que puede estar utilizando un terminal físicamente remoto) y para devolverle los resultados. Las peticiones se realizan generalmente en forma de sentencias SQL (Structured Query Language), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado.

Anteriormente se hablaba de software libres, a continuación se relacionan algunos SGBD:

#### **SGBD libres:**

- PostgreSQL (<http://www.postgresql.org> Postgresql) Licencia BSD
- MySQL Licencia Dual, depende el uso.
- Firebird basada en la versión 6 de InterBase, Initial Developer's PUBLIC LICENSE Version 1.0.
- SQLite (<http://www.sqlite.org> SQLite) Licencia Dominio Público
- [DB2Express-C](http://www.ibm.com/ar/businesscenter/catalogo/db2_express-c.phtml) ([http://www.ibm.com/ar/businesscenter/catalogo/db2\\_express-c.phtml](http://www.ibm.com/ar/businesscenter/catalogo/db2_express-c.phtml))
- Apache Derby (<http://db.apache.org/derby/>).

### ***SGBD gratuitos:***

- Microsoft SQL Server Compact Edition.
- Sybase ASE Express Edition para Linux (Edición gratuita para Linux)

#### ***1.3.8 PostgreSQL***

PostgreSQL [8] es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS), liberado bajo la licencia BSD [15], que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation.

En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continuó el desarrollo del código de Ingres para crear un sistema de bases de datos objeto-relacionales llamado Postgres. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL, tras un breve periplo como Postgres95. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características, algunas son la alta concurrencia y Amplia variedad de tipos nativos, entre otras.

#### ***1.3.9 Microsoft SQL Server***

Microsoft SQL Server [12] es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

Entre sus características figuran:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.



- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

Además permite administrar información de otros servidores de datos

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en su versión 2005 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

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

Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

### **1.3.10 MySQL.**

MySQL [13] es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL (Licencia Pública General) de la GNU [4]. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL [4] (*Licencia Pública General de GNU*).

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

## **Historia de MySQL.**

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Se comenta en el manual [MySQL\_Manual] que no se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

## **Características de MySQL**

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, Delphi, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Fácil y rápido aprendizaje
- Definición de vínculos en la consulta, esto le da a SQL una gran flexibilidad.
- Arquitectura cliente-servidor.

### **1.3.11 ¿Porque usar MySQL?**

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos relativamente grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos en Internet.

### **1.3.12 Herramientas seleccionadas.**

Basado en los argumentos antes expuestos y dado que se hará uso de Tecnologías cliente/servidor sobre plataforma Web para implementar el trabajo propuesto, se ha escogido al PHP como lenguaje de programación producto de su portabilidad y eficiencia, y como SGBD al MySQL por su rapidez y robustez. Ambos, por sus características, son inseparables, uno esta hecho para el otro.

### **1.3.13 Metodología del software a usar.**

Una metodología para el desarrollo de un proceso de software, según Presuman [9], es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos. Por ello escoger la metodología que va a guiar el proceso de desarrollo del sistema es un paso tan importante.

Para controlar, y planificar la propuesta que presenta este trabajo, se decidió utilizar como metodología el Proceso Unificado de Modelado [10] (RUP), por sus características y las facilidades que aporta a todo el proceso. Y teniendo en cuenta de que viene acompañado de una herramienta muy buena que soporta cada uno de los procesos a utilizar: Rational Rose Enterprise Edition 2003.

### **1.3.14 El Proceso Unificado de Modelado. RUP.**

El Proceso Unificado de Rational (RUP) es el producto final de tres décadas de desarrollo y uso práctico. Su desarrollo sigue un camino desde 1967 con la Metodología Ericsson (Ericsson Approach), una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; pasando por el proceso Objectory de Rational (publicado en 1997) hasta el Proceso Unificado de Rational (RUP: Rational Unified Process, publicado en 1998).

El Proceso Unificado es una propuesta de proceso para el desarrollo de software orientado a objetos que utiliza UML (Unified Model Language) [11] para describir todo el proceso. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

Sus características principales son:

- Guiado/Manejado por casos de uso.
- Centrado en arquitectura.
- Iterativo e Incremental.
- Desarrollo basado en componentes.
- Utilización de un único lenguaje de modelación.
- Proceso Integrado

Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases: Inicio, Elaboración, Construcción y Transición las cuales se explican brevemente a continuación y se hace detalladamente en la bibliografía correspondiente [10].

En la fase Inicio se establece el alcance del proyecto y se identifican todas las entidades externas con las que se trata (actores) y se define la interacción a un alto nivel de abstracción:

- Se identifican todos los casos de uso.
- Se describen algunos en detalle.

En la fase Elaboración se analiza el dominio del problema, se establece una arquitectura base sólida, se desarrolla un plan de proyecto y se eliminan los elementos de mayor riesgo para el desarrollo exitoso del proyecto.

En la fase Construcción se desarrollan las componentes restantes y se incorporan al producto. Se realizan pruebas al sistema.

Por último, en la fase Transición se traspasa el software desarrollado a los usuarios. Aquí se realizan:

- Pruebas para validar el producto con las expectativas del cliente.
- Entrenamiento de usuarios.
- Distribución del producto.

### **1.3.15 UML**

El desarrollo del Unified Model Language (UML) [10] empezó en octubre de 1994, cuando Grady Booch y Jim Rumbaugh en la Rational Software Corp. empezaron a trabajar para unificar el Booch (Metodología de Grady Booch) y la OMT (Objects Modelling Techniques). Un proyecto versión 0.8 del Método Unificado (UML), como se llamó desde un comienzo, salió al público en octubre de 1995. En el otoño de 1995, Ivar Jacobson se unió a la compañía y unió su esfuerzo al nuevo modelo, uniendo el OOSE (Object Oriented Software Engineering) al UML.

En resumen, UML es el resultado de la experiencia sumada, anotaciones, y conceptos. Ya que todas las metodologías bases han tenido una aplicación extensa en el campo del la Programación Orientada a Objeto (POO) se han desarrollado en la práctica, tienen su historia, y han sido aplicados en una gran variedad de industrias y problemas por lo que pueden ser clasificadas por como muy maduras. UML no es una salida revolucionaria de Booch, OMT, y OOSE, sino una evolución y síntesis de estos tres.

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las

cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.

#### **1.4 Conclusiones**

Al finalizar este capítulo se ha mostrado lo referente a la base teórica que fundamenta el presente trabajo de investigación, donde se exponen los conceptos fundamentales, las ventajas de las aplicaciones Web, los lenguajes de programación para la Web, el software libre, los sistemas de gestión de bases de datos, la gestión y estructura energética del ISMM y la metodología de ingeniería del software a utilizar para el desarrollo del sistema.

## **Capítulo 2 Modelo del dominio**

### **2.1 Introducción.**

En este capítulo se detallarán los distintos pasos de la metodología propuesta en el capítulo anterior para el desarrollo de la aplicación. Esto incluirá la elaboración del modelo del dominio, la exposición de los requerimientos de la plataforma, definición de actores y relaciones entre ellos, los diagramas de caso de uso y las descripciones textuales de los mismos, los diagramas de clases del diseño, diagrama de clases persistentes, modelo de datos, diagrama de despliegue y diagrama de componentes, entre otros.

### **2.2 Modelo de dominio**

En el capítulo anterior en el epígrafe estructura de la gestión energética en el ISMM se expuso una breve descripción del proceso de gestión energética donde se observó que no es posible identificar un negocio en el entorno en que trabajará el sistema, puesto que no existe algún flujo de procesos o intercambio entre actores y/o entre entidades o conceptos, es decir, el negocio tiene un bajo nivel de estructuración.

Por lo expuesto anteriormente, RUP [16] formula realizar un modelo de dominio. El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software.

Una clase conceptual puede ser una idea o un objeto físico (símbolo, definición y extensión).

El modelo de dominio se representa en UML con un Diagrama de Clases en los que se muestra:

- conceptos u objetos del dominio del problema: clases conceptuales
- asociaciones entre las clases conceptuales
- atributos de la clase conceptuales

En el Modelo de Dominio no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos.

Características:

- No hay cronología
- No se diferencia entre dentro y fuera del sistema
- Es global, no por caso de uso
- No es completo: esquemático, las asociaciones están resumidas.

### ***2.2.1 Definición de los conceptos principales del entorno en el que trabajará el sistema.***

A continuación se exponen algunos conceptos y entidades con sus respectivas descripciones asociadas al entorno del sistemas.

**Portador:** define como tal al producto tangible, producto o recurso energético.

**Tipo de Portador:** es la especificación del portador energético que describe a dicho portador.

**Modelo:** es el contenedor de la información de un portador.

**Energético:** pertenece al Dpto de Contabilidad y es el encargado de manipular o gestionar la información referente a los portadores energéticos.

**Áreas:** es donde se manipula tangiblemente el portador y es consumido posteriormente.

**Dpto Cont:** es el departamento de contabilidad donde se tienen los registros de los consumos y otros parámetros de los portadores.

En el capítulo anterior se describió brevemente la estructura del proceso de la gestión energética asociada a estos términos descritos ya y representados por la siguiente figura.



### 2.2.2 Representación esquemática del Modelo de Dominio.

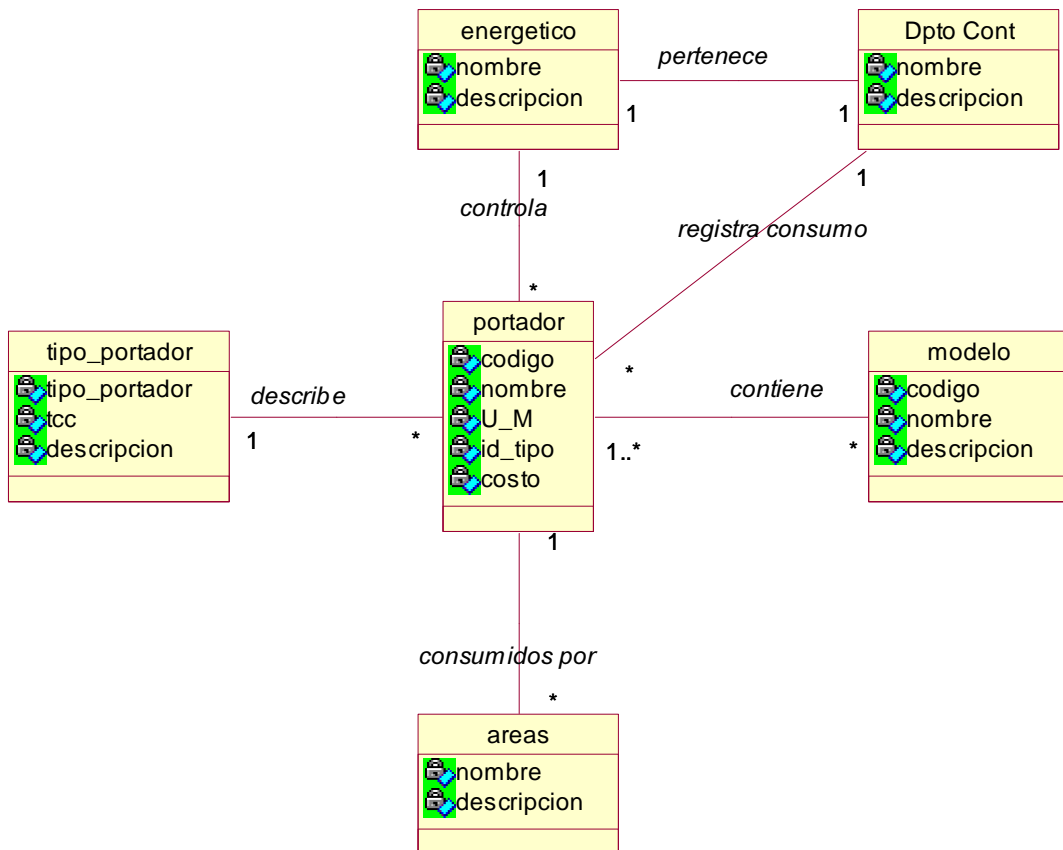


Fig. #2.1 Modelo de dominio.

### 2.3 Paquetes.

Cuando un sistema contiene muchos casos de uso, es necesario un mecanismo para agruparlos y así facilitar el uso, mantenimiento y es reutilizable. Esto se logra a través de los paquetes, que son utilizados para organizar los elementos de modelado en partes mayores que se pueden manipular como un grupo [10]. Los paquetes también ayudan a organizar los elementos en los modelos con el fin de comprenderlos más fácilmente.

UML representa los paquetes como carpetas. En el sistema se definieron los siguientes paquetes:

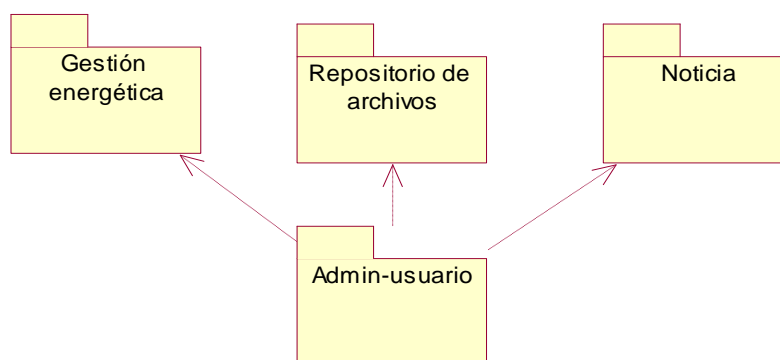
**Admin-user:** se manipulan algunas funcionalidades administrativas y se realiza la gestión de usuarios.

**Gestión energética:** donde se manipula o gestiona la información energética.

**Repositorio de archivos:** se realiza la gestión de los archivos subidos al servidor.

**Noticia:** se realiza la gestión de las noticias publicadas relacionadas con el tema energético.

Los paquetes están relacionados de la siguiente forma:



**Fig. #2.3** Diagrama de Paquetes

## **2.4 Requerimientos Funcionales y no Funcionales del Sistema.**

Los requerimientos del sistema son las características que debe cumplir o tener el sistema para su correcto funcionamiento, su apariencia y poder responder a las necesidades del cliente. A continuación se muestran los principales requerimientos funcionales y no funcionales respectivamente.

### **2.4.1 Requisitos Funcionales.**

Los requerimientos funcionales definen las responsabilidades del sistema, o sea, las funciones que éste será capaz de realizar.

#### **Paquete Admin-usuario**

1. Gestionar cuentas de usuarios.
  - Crear cuenta de usuarios.
  - Eliminar cuenta de usuario.
2. Buscar usuario.
3. Actualizar información de usuario.
4. Actualizar texto de bienvenida en la página de inicio.
5. Cambiar contraseña.
6. Validar usuario.
7. Gestionar proyecto.
  - Agregar proyecto.
  - Actualizar proyecto.
  - Eliminar proyecto.
8. Enviar petición.

#### **Paquete Gestión energética.**

1. Gestionar portador.
  - Agregar portador.
  - Actualizar portador.
  - Eliminar portador.

2. Gestionar tipo de portador.
  - Agregar tipo de portador.
  - Actualizar tipo de portador.
  - Eliminar tipo de portador.
3. Gestionar índice de consumo.
  - Agregar parámetros para el índice de consumo.
  - Actualizar parámetros índices de consumo.
  - Eliminar parámetros índices de consumo.
4. Gestionar indicador
  - Agregar indicador.
  - Actualizar indicador.
  - Eliminar indicador.
5. Gestionar tipo indicador
  - Agregar tipo indicador.
  - Actualizar tipo indicador.
  - Eliminar tipo indicador.
6. Gestionar información general de portador
  - Agregar información general de portador.
  - Actualizar información general de portador.
  - Eliminar información general de portador.
  - Mostrar información general.
7. Buscar datos de un portador.
8. Generar gráficos de conductas de los portadores.
9. Visualizar historial de las acciones realizadas en la base de datos.
10. Mostrar proyectos.

### ***Paquete Repositorio de archivos:***

1. Subir archivo al Repositorio.
2. Renombrar archivo.
3. Borrar archivo.
4. Buscar archivo.
5. Descargar archivo.

### ***Paquete Noticia***

1. Gestionar noticias
  - Agregar noticia.
  - Eliminar noticia
  - Editar noticia.
2. Gestionar comentario
  - Agregar comentario.
  - Eliminar comentario.

### ***2.4.2 Requisitos no Funcionales***

Los requerimientos no funcionales son las propiedades o cualidades que el producto o sistema debe tener. Incumbe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

Ellos son los siguientes.

#### ***Apariencia:***

La interfaz debe ser agradable, sencilla y atractiva para una mejor relación con los usuarios. Se debe tener en cuenta algunos elementos de diseño como gráficos de encabezamiento, estilos, formatos de texto y colores.

#### ***Rendimiento:***

Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a sus páginas. La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el usuario requiere la respuesta a su acción.

Las herramientas utilizadas para la implementación propician el resultado de un sistema con rapidez y multiplataforma.

***Seguridad:***

El sistema debe identificar con seguridad a los diversos usuarios que interactúan con él. El sistema tiene que garantizar que la información sea vista únicamente por quien tiene derecho a verla.

***Documentación para los usuarios y ayudas del sistema:***

Crear un sistema de ayuda de forma tal que en todo momento le permita al usuario orientarse respecto a las opciones que le brinda el sistema, utilizando textos explicativos que describan la acción que se está realizando.

***2.5 Conclusiones.***

En este capítulo se muestran algunos conceptos del entorno en que trabajará el sistema, así como la representación del modelo de dominio, la definición de las principales funcionalidades descritas por los requerimientos y la organización de estos por paquetes.

## Capítulo 3 Diseño e Implementación del Sistema.

### 3.1 Introducción.

En este nuevo capítulo se continuará aplicando la metodología RUP para modelar las funcionalidades, expuestas en el capítulo anterior, mediante los diversos diagramas propuestos por dicha metodología.

### 3.2 Actores del sistema a automatizar.

Actores	Justificación
Admin	Administrador del sistema, configura y actualiza el sistema, puede ejercer rol de trabajador e invitado.
Trabajador	Una vez que se le asigna privilegios como trabajador puede manipular la gestión energética.
Invitado	Ve información que los trabajadores y administrador permitan.
UsuarioP	Generalización de los <b>Actores</b> Admin y trabajadores.
Usuario	Generalización de los <b>Actores</b> trabajadores e invitado.

**Tabla 3.1** Justificación de Actores

### **3.3 Diagrama de casos de uso del sistema a automatizar.**

Utilizando las facilidades que brinda el UML, se representarán los requisitos funcionales del sistema mediante un diagrama de casos de uso. Para ello se definirán primeramente cuáles son los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

Un caso de uso es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor [10]. Un actor no es parte del sistema, representa un rol que es jugado por una persona, un dispositivo hardware o incluso otro sistema al interactuar con nuestro sistema. Puede intercambiar información o puede ser un recipiente pasivo de información.

Un diagrama de casos de uso es un diagrama que muestra un conjunto de casos de uso, actores y sus relaciones. [10] Se utiliza para el modelado del comportamiento de un sistema, un subsistema o una clase.

Se representarán los diagramas teniendo en cuenta los paquetes definidos con anterioridad. Además se abordan las descripciones textuales de los casos de uso más significativos.



**Paquete admin-usuario**

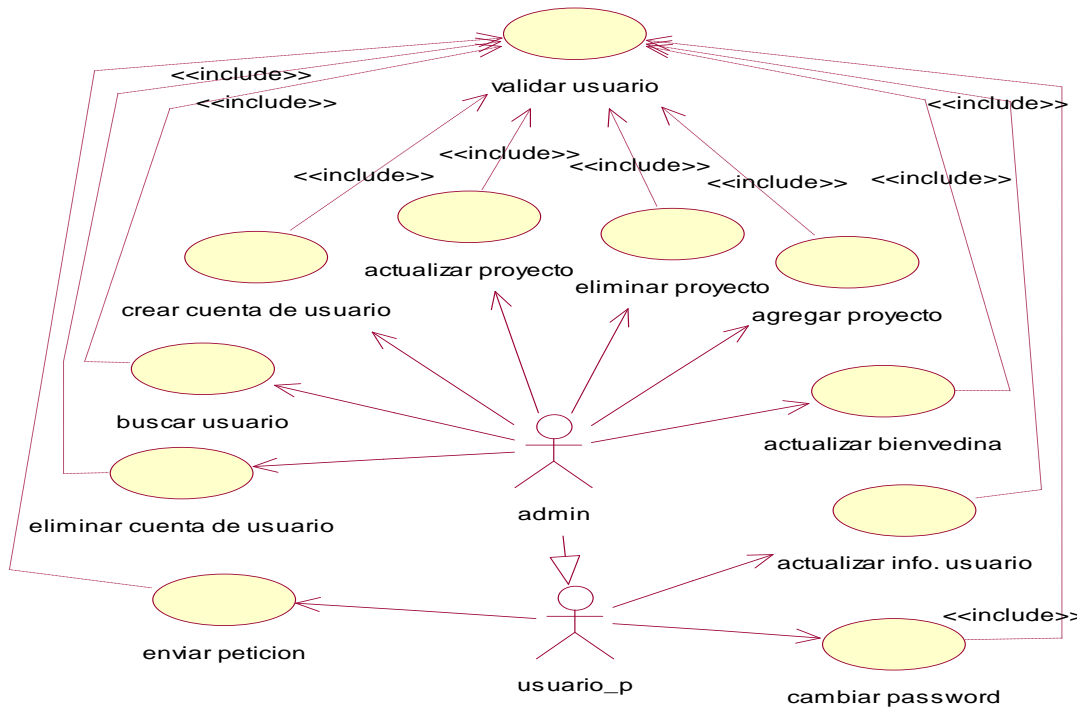


Fig. # 3.1 Diagrama de Casos de Uso del paquete Admin-usuario

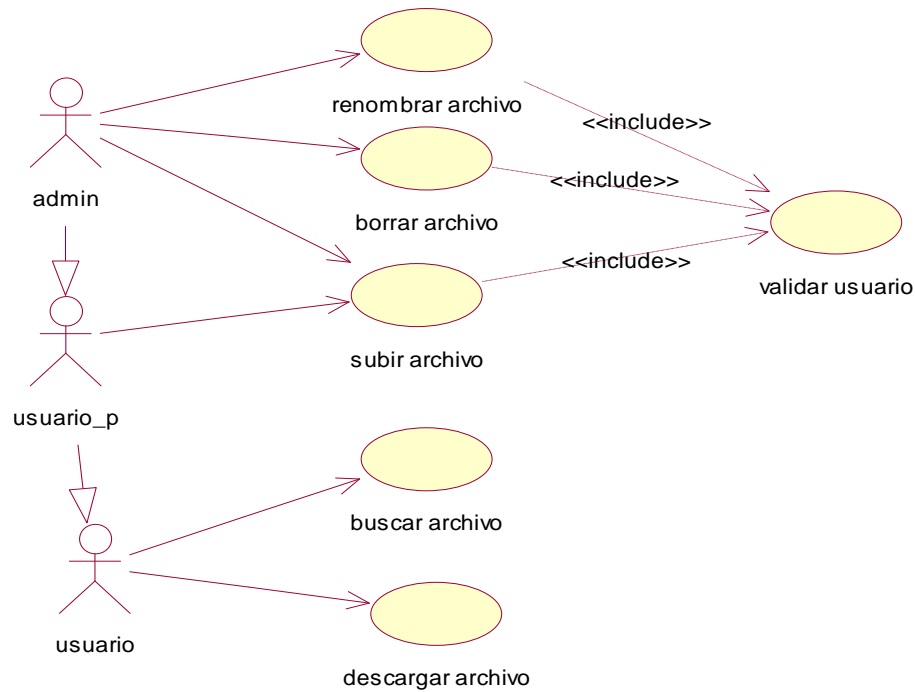
**Descripciones textuales de los casos de uso del**

**Paquete Admin-usuario:**

<b>Nombre del caso de uso</b>	Crear cuenta de usuario
<b>Actores</b>	administrador (inicia)
<b>Propósito</b>	Crema una cuenta para un nuevo usuario.
<b>Resumen</b>	El caso de uso inicia cuando el administrador decide hacer una cuenta para un nuevo usuario del sistema. Si el usuario ya ha sido matriculado anteriormente como visitante, el administrador solamente le asignará el nivel de acceso deseado, de lo contrario el administrador agrega todos los datos del usuario incluyendo el nivel de acceso.
<b>Referencias</b>	R1
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se creara una nueva cuenta de usuario.

Para ver el resto de las descripciones ir al anexo 1

**Paquete Repositorio de archivos.**



**Fig. # 3.2** Diagrama de casos de uso del paquete Repositorio de archivos.

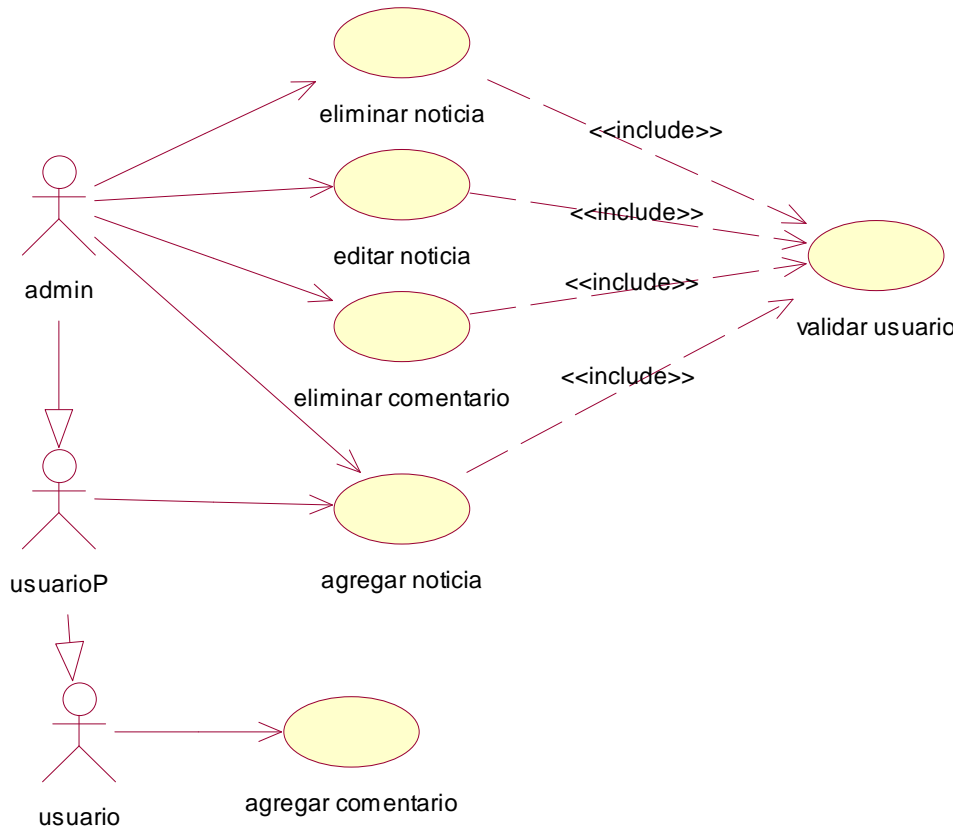
**Descripciones textuales de los casos de uso del paquete**

**Repositorio de archivos:**

<b>Nombre del caso de uso</b>	Subir archivo
<b>Actores</b>	UsuarioP (inicia)
<b>Propósito</b>	Guardar un archivo en el servidor.
<b>Resumen.</b>	El caso de uso se inicia cuando algún administrador o trabajador decide subir un archivo al Repositorio con fin de una posible reutilización por otros usuarios.
<b>Referencias</b>	R1
<b>Precondiciones</b>	Debe existir el paquete con el formato definido. El usuario debe ser válido.
<b>Poscondiciones</b>	Se ha exportado al Repositorio un nuevo archivo.

Para ver el resto de las descripciones ir al anexo 2

**Paquete Noticia.**



**Fig. # 3.3** Diagrama de casos de uso del paquete noticia.

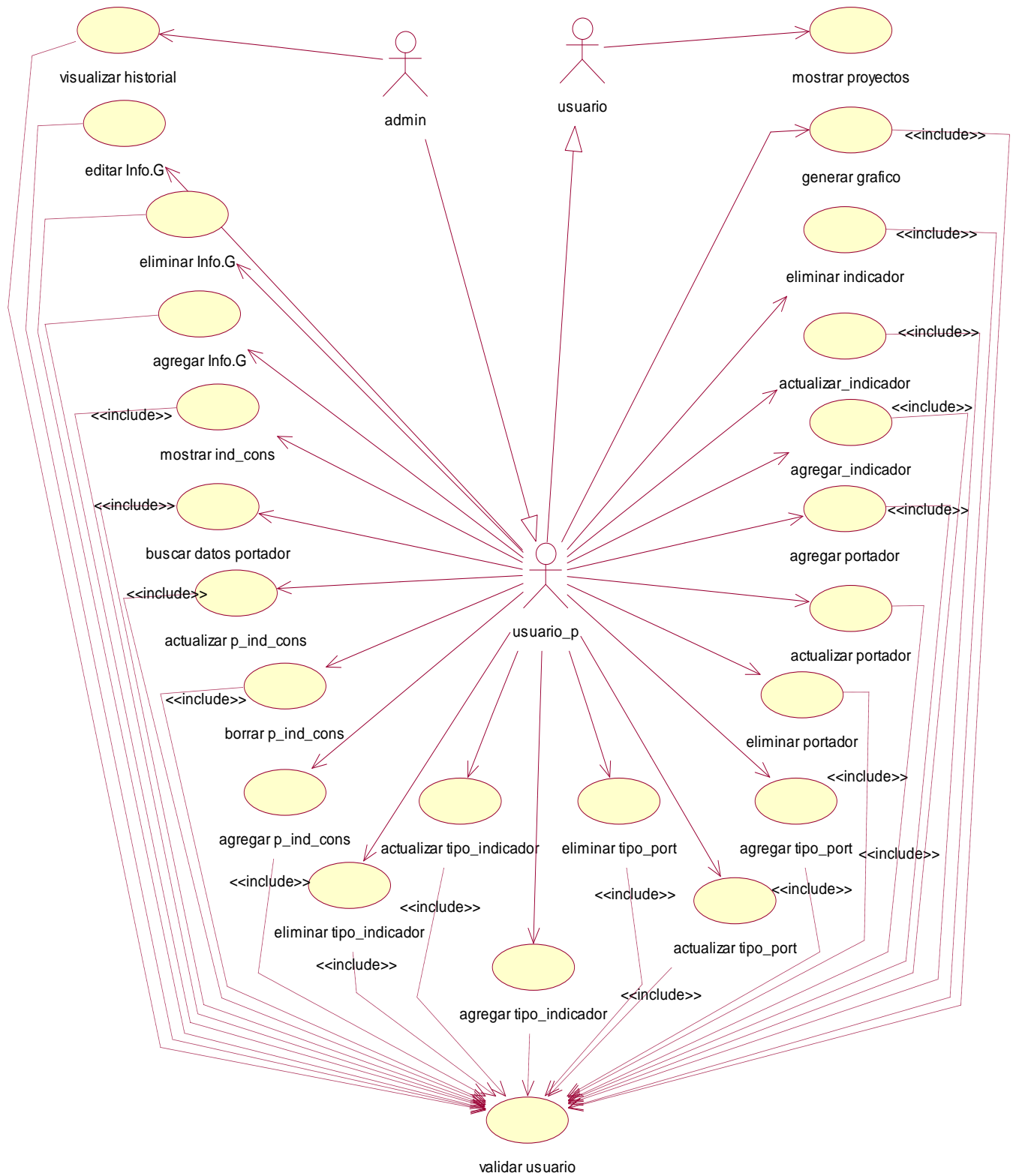
**Descripciones textuales de los casos de uso del paquete noticia:**

<b>Nombre del caso de uso</b>	Agregar noticia.
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Publicar una noticia.
<b>Resumen.</b> El caso de uso inicia cuando el Administrador decide publicar una noticia accesible a los visitantes.	
<b>Referencias</b>	R1
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Queda publicada una noticia.

<b>Nombre del caso de uso</b>	Editar noticia
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Edita el contenido de una noticia.
<b>Resumen.</b> El caso de uso inicia cuando el Administrador decide actualizar el contenido de una noticia publicada.	
<b>Referencias</b>	R1
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se actualiza la noticia publicada.

Para ver el resto de las descripciones ir al [anexo 3](#)

**Paquete Gestión energética.**



**Fig. # 3.4** Diagrama de casos de uso del paquete gestión energética.

**Descripciones textuales de los casos de uso del paquete Gestión energética:**

<b>Nombre del caso de uso</b>	agregar info.G
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Agregar información general de un portador
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide agregar la información general del portador energético. Esta información son los datos generales como resultado del consumo del portador durante todo un mes.	
<b>Referencias</b>	6
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se adiciona un nuevo registro a la BD.

<b>Nombre del caso de uso</b>	actualizar info.G
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	actualizar información general de un portador
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide modificar la información general del portador energético. Esta información son los datos generales como resultado del consumo del portador durante todo un mes.	
<b>Referencias</b>	6
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se modifica el registro seleccionado.

Para ver el resto de las descripciones ir al [anexo 4](#)

### 3.4 Diagrama de clases del diseño

En el diagrama de clases para las Aplicaciones Web, se modelan las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de las mismas, una vez que estén en el navegador del cliente. A continuación se representan los diagramas de las clases Web por paquetes y agrupados algunos por casos de usos:

#### Diagramas clase Web del paquete Admin.-user:

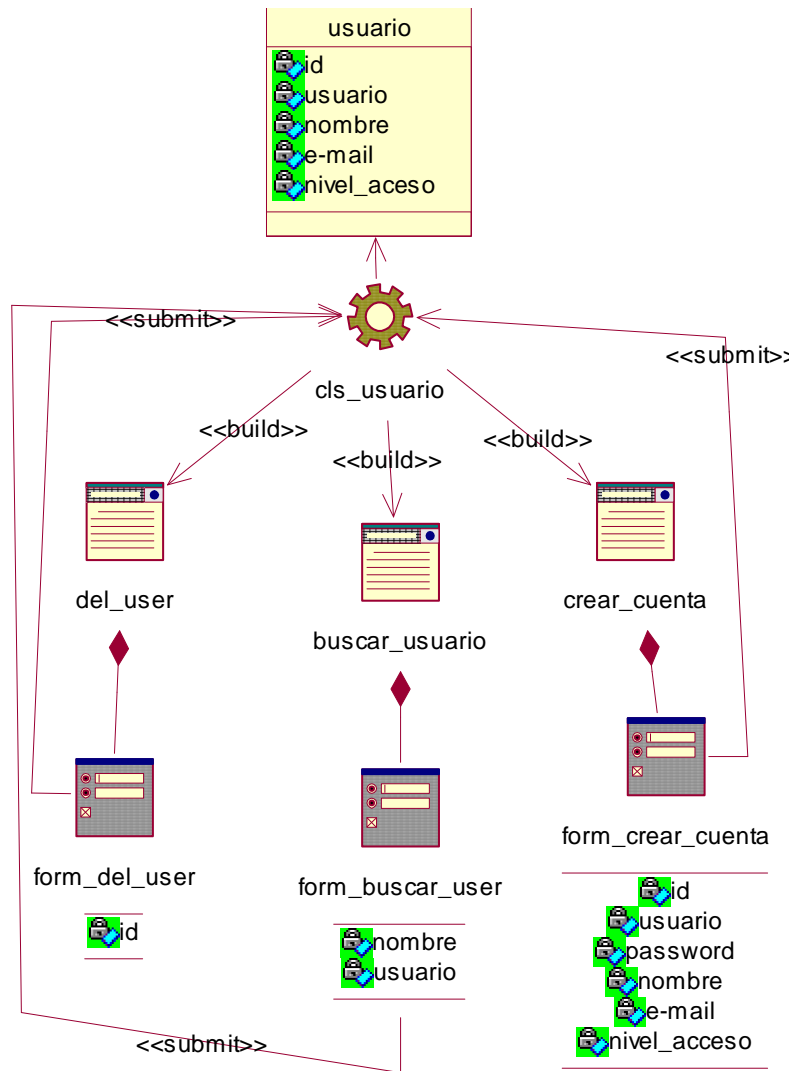
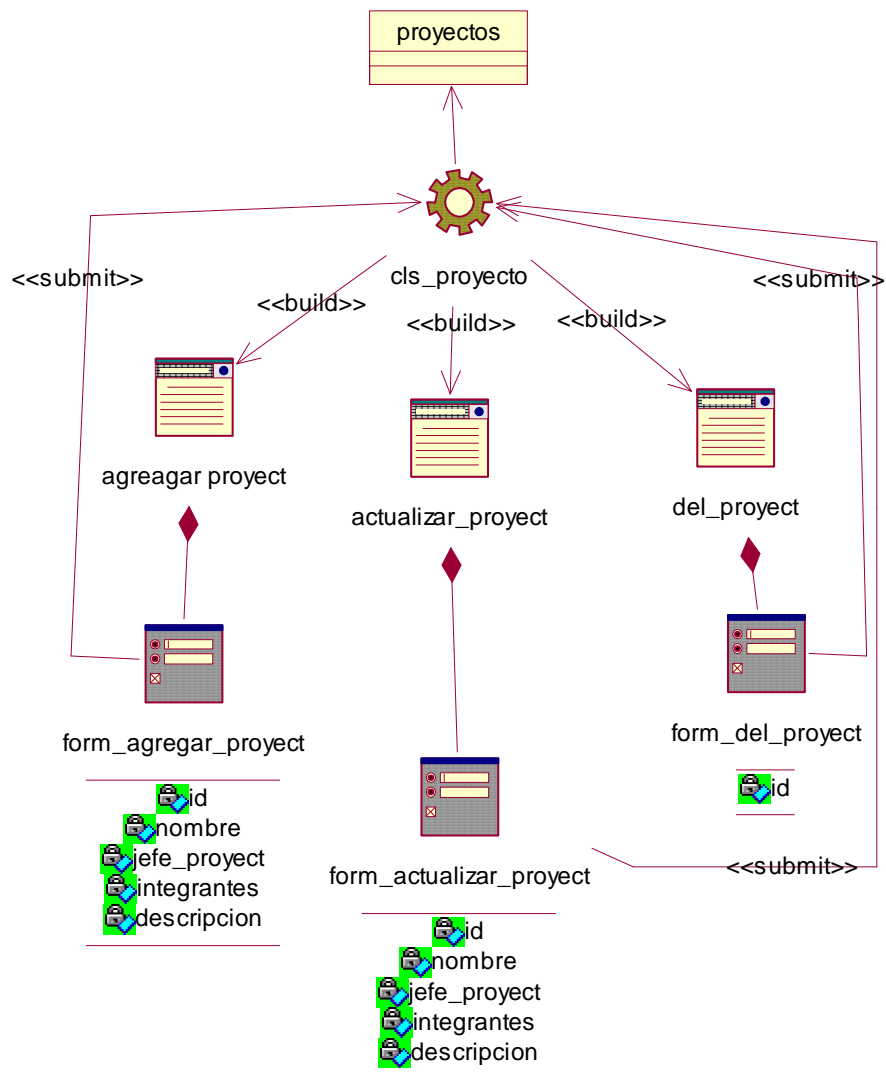


Fig. # 3.5 Diagrama clase Web crear cuenta, eliminar cuenta y buscar usuario.



**Fig. # 3.6** Diagrama clase Web agregar proyectos, actualizar proyectos y eliminar proyectos.

Para ver el resto de los diagramas ir al [anexo 5](#)



### **3.5 Principios de diseño**

La página principal de la aplicación, contiene una imagen asociada al entorno del ISMM. Existe un menú visible desde casi todas las páginas del sitio, para guiar al usuario en su navegación por el sitio. Siempre existe la forma de llegar directamente a la página principal del sitio, de igual forma, en cualquier momento el usuario puede finalizar la sesión.

El diseño de la interfaz es totalmente independiente de la capa lógica facilitando un cómodo trabajo a la hora de modificar su apariencia, donde se ha usado algunos estilos, fuentes estándares y colores suaves.

### 3.6 Diseño de la base de datos

#### 3.6.1 Modelo lógico de datos

Las clases persistentes son las clases que necesitan ser capaces de guardar su estado en un medio permanente, la necesidad de guardar su estación esta dado por el almacenamiento físico permanente de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información.

A continuación se representan las clases persistentes:

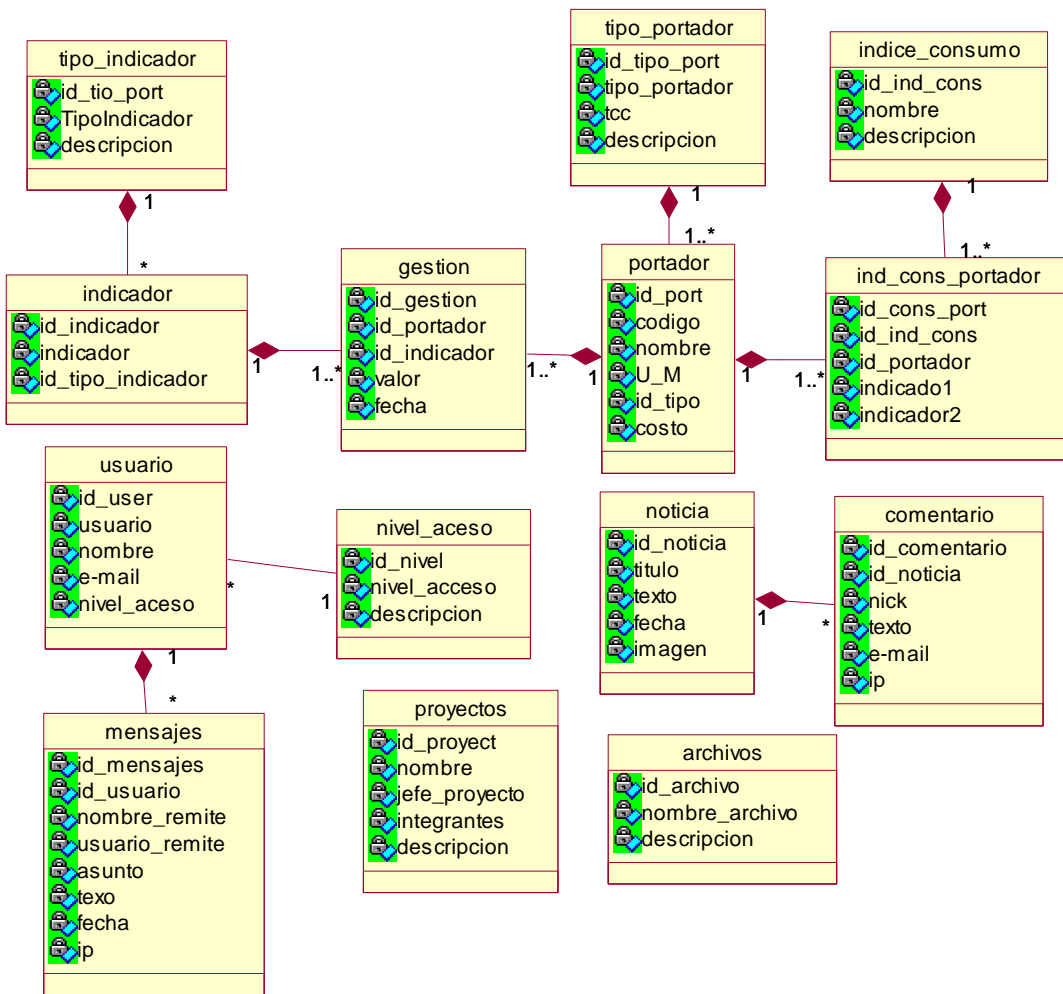


Fig. 3.7 Diagrama de clases persistentes.

**Descripción textual de las clases persistentes:**

<b>Nombre Clase</b>	gestion
<b>Propósito</b>	Almacenar los datos de la información general referente a la gestión energética.
<b>atributo</b>	<b>Descripción</b>
id	Es el identificador o llave de la clase.
Id_portador	Es el identificador del portador.
Id_indicador	Identificador del indicador.
valor	Valor numérico que posee en indicador correspondiente al portador.
fecha	Fecha a la que corresponde la información registrada.

<b>Nombre Clase</b>	portador
<b>Propósito</b>	Almacenar los datos del portador.
<b>atributo</b>	<b>Descripción</b>
id	Es el identificador del portador.
codigo	Es un código perteneciente a cada portador, donde dicho código es único.
nombre	Nombre del portador.
U_M	Es la unidad de medida del portador energético
tipo_portador	Es el tipo de portador a que pertenece este.
costo	Valor monetario del recurso energético.

Para ver el resto de las descripciones ir al [anexo 7](#)

### 3.6.2 Modelo físico de datos

El modelo de datos se utilizó para describir la estructura lógica de la información persistente almacenada por el sistema. El modelo de datos fue obtenido del diagrama de clases persistentes, seguido se muestra el modelo de datos.

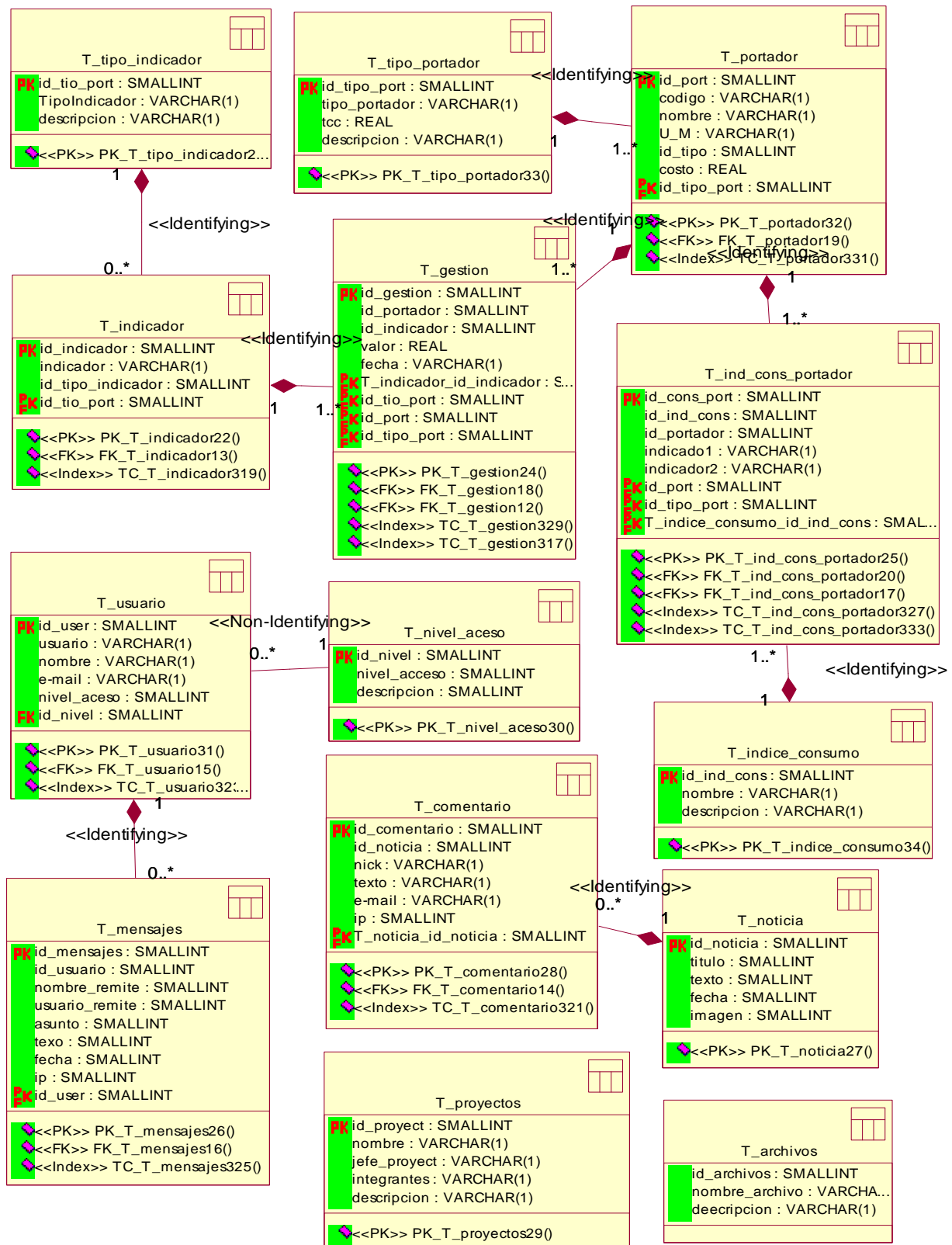


Fig. 3.8 Diagrama del modelo de datos.

### 3.7 Diagrama de Secuencia

El diagrama de secuencia muestra como los objetos se comunican unos con otros para llenar los requerimientos del sistema. Da una visión gráfica de las interacciones de los actores y las operaciones del sistema a que dan origen. El diagrama de secuencia da una vista del comportamiento del sistema mostrando qué hace ante el medio y sin explicar cómo lo hace.

A continuación tenemos un diagrama de secuencia, en el anexo correspondiente se muestra los más importantes.

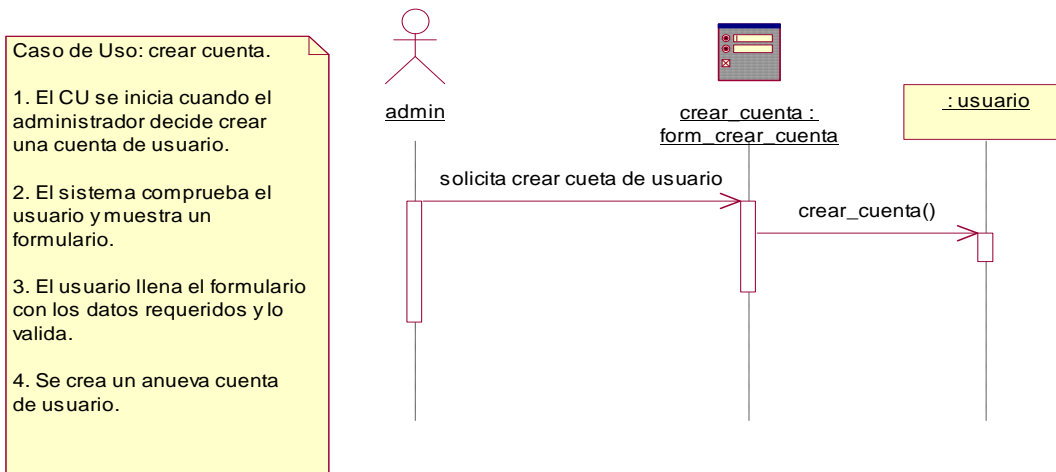
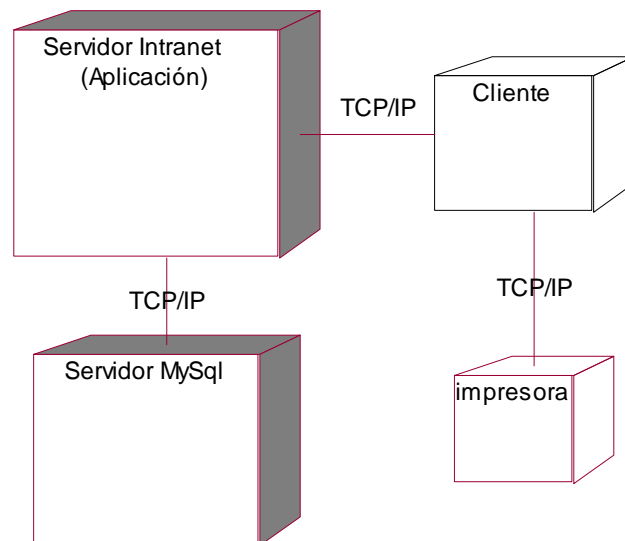


Fig. 3.9 Diagrama de Secuencia crear cuenta.

Para ver el resto de los diagramas ir al [anexo 6](#)

### 3.8 Diagrama de despliegue.

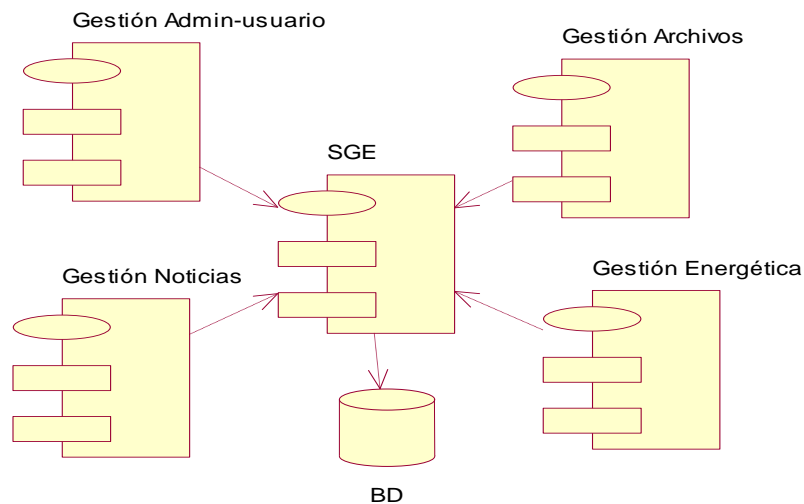
Los diagramas de despliegue se utilizan para visualizar la topología de procesadores y dispositivos sobre los que se ejecuta la aplicación. Se define un diagrama de despliegue como un esquema que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Como se puede observar en la siguiente figura el sistema de gestión energética está formado por una arquitectura Web clásica, o sea, un cliente y un servidor donde se hospedarán los Servicios Web y el gestor de base de datos.



**Fig. 3.10** Diagrama despliegue.

### 3.9 Diagrama de Componentes

El Diagrama de Componentes se usa para modelar la estructura de la aplicación, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables. El esquema de componentes modela componentes del sistema, a veces agrupados por paquetes, y las dependencias que existen entre componentes. A continuación se muestra el diagrama de componente por paquetes y los diagramas de componentes de cada paquete se pueden consultar en el [anexo #8](#).



**Fig. 3.11** Esquema de componentes por paquetes.

### 3.10 Conclusiones.

Este capítulo se ha mostrado los aspectos que, dentro de la metodología de ingeniería del software empleada, brindan informaciones específicas para el entendimiento de la solución propuesta mediante la modelación del problema y la solución propiamente dicha. Se ha expuesto el camino seguido para la elaboración de la aplicación en su conjunto, así como de cada una de sus partes o componentes.

## **Capítulo 4 Estudio de factibilidad**

### **4.1 Introducción.**

Cualquier proyecto de software puede realizarse si el equipo desarrollador cuenta con recursos y tiempo infinito; pero tristemente no es así. La mayor parte de los proyectos informáticos presentan carencias de recursos, o no existe entendimiento entre los desarrolladores y el cliente y las fechas de entrega no se corresponden con la realidad. Es por ello que cada día se hacen más necesarias la realización de estimaciones al inicio y a lo largo del ciclo de vida de los proyectos, aun cuando esta mirada al futuro tenga cierto grado de inseguridad. La estimación es la base de todas las demás actividades de planificación del proyecto y sirve como guía para una buena Ingeniería de Software.

En este capítulo se expone el estudio y factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costo, realizadas con el método de puntos de función del modelo de COCOMO II en la etapa de diseño temprano. Se estiman los beneficios tangibles e intangibles que representan para el sistema propuesto, un análisis de costos y beneficios.

### **4.2 Planificación.**

La estimación del proyecto se realizó mediante los puntos de función desajustados, los cuales se utilizan para el cálculo de las instrucciones fuentes. De esta forma se estima la magnitud del sistema y se obtienen además indicadores como la cantidad de hombre, el esfuerzo, el tiempo de duración y el costo del mismo.

Barry Boehm, en su libro clásico sobre economía de la Ingeniería del Software, introduce una jerarquía de modelos de estimación de Software con el nombre de COCOMO, por su nombre en Inglés (Constructive, Cost, Model) modelo constructivo de costos. Basados en estos modelos se estimarán el esfuerzo, tiempo de desarrollo, cantidad de hombres y costo de componentes representativos del software.



**Salidas externas:**

**Tabla 4.1** Salidas externas.

Nombre de la entrada externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y alto)
Generar grafico	1	12	Bajo
Visualizar historial	1	4	Bajo
Mostrar proyectos	1	4	Bajo
Mostrar info. G	1	5	Bajo

**Ficheros lógicos internos:**

Nombre del ILF	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y alto)
Usuario	1	4	Bajo
Noticia	1	4	Bajo
Comentario	1	5	Bajo
Portador	1	5	Bajo
Tipo_portador	1	3	Bajo
Indicador	1	2	Bajo
Tipo_indicador	1	2	Bajo
Ind_cons_portador	1	4	Bajo
Indice_consumo	1	2	Bajo
Archivo	1	2	bajo
Gestión	1	5	Bajo
Proyectos	1	4	Bajo

**Tabla 4.2** Ficheros lógicos internos.

**Entradas externas:**

<b>Nombre de la entrada externa</b>	<b>Cantidad de Ficheros</b>	<b>Cantidad de Elementos de datos</b>	<b>Clasificación (Bajo, Medio y alto)</b>
Agregar portador	1	4	Bajo
Actualizar portador	1	4	Bajo
Agregar tipo portador	1	3	Bajo
Actualizar tipo portador	1	3	Bajo
Agregar indicador	1	2	Bajo
Actualizar indicador	1	2	Bajo
Agregar tipo indicador	1	2	Bajo
Actualizar tipo indicador	1	2	Bajo
Agregar parámetros índice consumo	1	4	Bajo
Actualizar parámetros índice consumo	1	4	Bajo
Agregar info. G.	1	5	Bajo
Actualizar info. G.	1	5	Bajo
Agregar proyectos	1	4	Bajo
Actualizar proyectos	1	4	Bajo
Crear cuenta de usuario	1	4	Bajo
Actualizar info. De usuario	1	4	Bajo
Agregar noticia	1	4	Bajo
Agregar comentario	1	5	Bajo
Subir archivo	1	2	Bajo
Renombrar archivo	1	2	Bajo

**Tabla 4.3** Entradas externas.

### Consultas Externas:

Nombre de la consulta externa	Cantidad de Ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y alto)
Eliminar portador	1	4	Bajo
Eliminar tipo portador	1	3	Bajo
Eliminar indicador	1	2	Bajo
Eliminar tipo indicador	1	2	Bajo
Eliminar parámetro índice consumo	1	4	Bajo
Eliminar info. G.	1	5	Bajo
Eliminar proyectos	1	4	Bajo
Eliminar cuenta usuario	1	4	Bajo
Eliminar noticia	1	4	Bajo
Eliminar comentario	1	5	Bajo
Cambiar password	1	3	Bajo
Validar usuario	1	4	Bajo
Buscar usuario	1	4	Bajo
Buscar datos portador	1	5	Bajo
borrar archivo	1	2	Bajo
Enviar petición	1	7	Bajo

**Tabla 4.4** Consultas externas.

### Ponderación aplicando pesos.

Elementos	Bajos		Medios		Altos		Subtotal de puntos de función
	No	X Peso	No	X Peso	No	X Peso	
Ficheros lógicos internos	12	7	0	10	0	15	84
Entradas externas	20	3	0	3	0	6	60
Salidas externas	4	4	0	4	0	7	16
Consultas Externas	16	3	0	3	0	6	48
Total	52	17	0	20	0	34	208

**Tabla 4.5** Ficheros lógicos internos.

### Estimación de la cantidad de instrucciones fuentes. (SLOC).

Para el cálculo de las instrucciones fuentes (SLOC) se utilizó la fórmula siguiente:

$$\text{SLOC} = \text{UFP} * \text{Ratio}$$

$$\text{SLOC} = 208 * 90$$

$$\text{SLOC} = 18720$$

$$\text{KSLOC} = 18.720 \text{ (Miles de líneas de código)}$$

Donde UFP es el total de puntos de función desajustados, y ratio es una constante para las SLOC de cada lenguaje de programación en este caso tiene un valor para PHP de 90.

### Aplicación de las fórmulas de Bohem.

Luego de calculada la cantidad de instrucciones fuentes, se utilizó este valor en el cálculo del esfuerzo dado por la fórmula de **Bohem**:

Obtener esfuerzo (PM) y tiempo de desarrollo (TDEV).

### Calculo Del esfuerzo.

El esfuerzo es la cantidad de tiempo que una persona invierte trabajando en el desarrollo de un proyecto durante un mes. La sigla que lo representa es PM.

$$PM_{NS} = A \times \text{Size}^E \times \prod_{i=1}^n EM_i$$

Donde

$$E = B + 0.01 \times \sum_{j=1}^s SF_j$$

Se tiene además los valores constante: A=2.94 y B=0.91.

$$E = B + 0.01 \times \sum_{j=1}^s SF_j$$

$$E = 0.91 + 0.01 * 17.64$$

$$E = 0.91 + 0.1553$$

E= 1.0864 entonces

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$PM = 2.94 * 18.720^{1.0864} * 1.0402938$$

$$PM = 2.94 * 24,111990805227602183827319588094 * 1.0402938$$

$$PM = 73.74 \approx 73$$

Se necesitan 73 personas para un mes de trabajo.

Para el cálculo del tiempo se empleó la fórmula:

$$TDEV_{NS} = C \times (PM_{NS})^F \quad \text{Donde:}$$

$$F = D + 0.2 \times 0.01 \times \sum_{j=1}^s SF_j \quad \text{ó} \quad F = D + 0.2 \times (E - B)$$

Se tiene también los valores de C y D donde C= 3.67 y D= 0.28

$$F = 0.28 + 0.2 * 0.01 * 17.64$$

$$F = 0.28 + 0.2 (1.0864 - 0.91)$$

$$F = 0.31528$$

$$F = 0.31528$$

$$TDEV_{NS} = C \times (PM_{NS})^F = 3.67 * (73.74)^{0.31528}$$

$$TDEV_{NS} = 3.67 * 3,8801582987268279340084739737728$$

$$TDEV_{NS} = 14.24 \Rightarrow TDEV_{NS} = 15 \text{ meses}$$

Se puede desarrollar el proyecto en 15 meses.

Para obtener los resultados de las fórmulas anteriormente expuestas, se calcularon los valores de cada factor de escala (SF<sub>j</sub>) y de cada multiplicador de esfuerzo (EM<sub>i</sub>).

**Factores de escala:**

Factor de escala	Valor
PREC	3.72
FLEX	2.03
RESL	4.24
TEAM	1.10
PMAT	4.68

**Tabla 4.6** Factores de escala.**Multiplicadores de escala:**

Multiplicador	4.2.1.1 Valor
RCPX	1.00
RUSE	1.07
PDIF	1.00
PERS	0.63
PREX	1.00
FCIL	0,73
SCED	1,00

**Tabla 4.7** Multiplicadores de escala.**Valores calculados.**

Características	Valor
Puntos de función desajustados	208
Lenguaje	php
Instrucciones fuentes por puntos de función	90
Instrucciones fuentes	17.720

**Producto de los multiplicadores de esfuerzo:**

$$\square EM = PERS *RCPX*RUSE*PDIF*PREX* SCED*FCIL$$

$$\square EM = 1.26*1.30*1.00*0.87*1.00*1.00*0.73$$

$$\square EM = 1.0402938$$

**Sumatoria de los factores de escala.**

$$\Sigma SF = PREC + FLEX + RESL + TEAM+PMAT$$

$$\Sigma SF = 3.72 + 2.03 + 2.83 +4.38+ 4.68$$

$$\Sigma SF = 17.64$$

**Hallar la cantidad de hombre.**

Siglas	Indicador	Valor o Formula
CH	Costo del proyecto	PMTDES
PM	Costo de hombres por mes	73.74 hombre- mes
TDES	Tiempo de desarrollo	14.24 meses

Sabemos que  $CH = PM/TDES$  entonces

$$CH = 73.74/14.24 \approx 5 \text{ persona}$$

El software, en 16.64 meses pudiera ser realizado con 8 persona.

**Hallar el costo del software.**

Para el cálculo del costo del software se tiene en cuenta el salario promedio de las personas que lo desarrollan y del esfuerzo que ellas realizan para la ejecución del mismo. Usando los valores de la siguiente tabla se halla el costo.

Siglas	Indicador	Valor o Formula
C	Costo del proyecto	CHM*PM
CHM	Costo de hombres por mes	CH*SP
SP	Salario básico de un ingeniero	\$ 225.00
PM	Esfuerzo	73.74

El salario medio es de \$225.00

$$C = CHM * PM \quad \text{y} \quad CHM = CH * SP = 5 * 225 = 1125$$

$$C = 1125 * 73.74$$

$$C = 82957,5$$

El software vale \$82957,5

### Resumen de los cálculos.

Calculo de:	Valor	Justificación
Esfuerzo	73.74 hombre-mes	Cantidad de tiempo invertida por una persona desarrollando el proyecto
Tiempo de desarrollo	15 meses	Cantidad de meses para finalizar el software
Cantidad de personas	5	Cantidad de personas necesarias para terminar el proyecto en 2 meses
Costo	\$82957,5	Lo que vale el software



### **4.3 Beneficios tangibles e intangibles.**

Estos beneficios pueden contribuir a largo plazo sustancialmente a la economía y al cuidado del medio ambiente.

#### **Beneficios tangibles:**

El resultado del presente trabajo traería consigo beneficios en cuanto al ahorro energético, posibilitaría el estudio del consumo de los recursos energéticos con respecto al plan señalado y establecería una planificación del insumo de los portadores.

#### **Beneficios intangibles:**

El producto permitirá publicar y compartir informaciones asociadas al término energético enriqueciendo los conocimientos y la cultura energética de los visitantes, entre otros.

### **4.4 Análisis de costos y beneficios.**

El desarrollo de un producto informático siempre tiene un costo. Este puede estar justificado por los beneficios el mismo.

El costo que significa la implementación en este caso no es elevado si se tiene en cuenta las contribuciones que trae consigo el ahorro energético.

Además, la tecnología o herramientas utilizadas para el desarrollo del sistema es totalmente libre y código abierto, por tanto no es necesario incurrir en gastos en el pago de licencias de uso.

Por ello se plantea que es factible el desarrollo del mismo.

### **4.5 Conclusiones.**

El estudio de factibilidad llevado a cabo en este capítulo ha permitido calcular matemáticamente el comportamiento del desarrollo del sistema, por lo que es de suma importancia en la planificación y toma de decisiones referentes al mismo.

### ***Conclusiones generales.***

Con el desarrollo de la aplicación Web para la gestión energética del ISMM se darán cumplimiento a los objetivos propuestos en este trabajo, pues se obtendrá como resultado un producto informático en el que se aplican los resultados de la investigación realizada arribándose a las siguientes conclusiones:

- Se elaboró el marco teórico metodológico que fundamenta la investigación, permitiendo analizar el proceso y la estructura de la gestión energética en el ISMM.
- Se realizó el análisis para determinar las principales herramientas usadas en el desarrollo de la aplicación.
- Se diseñó un sistema informático que permite el control de la información de los portadores energéticos. También brinda o comparte otro tipo de informaciones asociadas al ahorro de la energía.
- Se realizó un estudio de factibilidad donde se obtuvo como resultado el tiempo de estimación para el desarrollo del sistema, el esfuerzo y la cantidad de personas necesarias para la realización del mismo, así como el costo.

### ***Recomendaciones.***

Por la importancia que tiene el presente trabajo para la eficiencia energética en el ISMM se recomienda lo siguiente:

- Realizar un estudio más profundo respaldado por una mejor coordinación con el propósito de mejorar aun más el presente sistema.
- Implementar un modulo especializado para la gestión llevada a cabo por transporte.
- Implementar un modulo de reportes estadístico energético para la obtención de una mas amplia información.
- Establecer funciones evaluadoras para flexibilizar el cálculo complejo de índice de consumos deseados, facilitando al usuario establecer de forma dócil el cálculo de los mismos.

## Referencias bibliográficas y bibliografías.

Aníbal E. Borroto Nordelo, *Gestión Energética Empresarial*, Editorial Universidad de Cienfuegos, Cuba. 2002.

- [1] Carlos del Porto Blanco, *Ministerio de la Informática y las Comunicaciones de Cuba*, [http://www.informaticahabana.com/evento\\_virtual/?q=uno](http://www.informaticahabana.com/evento_virtual/?q=uno) (1/17/2008).
- [2] Free Software Foundation. <http://www.fsf.org>. (Consulta: 09/abr/2007).
- [3] Fundación Iberoamericana de para la gestión de la calidad, [www.fundibeq.org/metodologias/herramientas/diagrama\\_de\\_pareto.pdf](http://www.fundibeq.org/metodologias/herramientas/diagrama_de_pareto.pdf)
- [4] GNU Project Web Site. <http://www.gnu.org> (Consulta: 07/abr/2007).
- [5] MIC (2002). *Estrategia para el uso del Software Libre en Cuba*.
- [6] Millán, José Antonio. Breve Historia de la Internet. Alonso, F. B. (2007) *Construyendo una Intranet*.
- [7] Monmany, Jordi. Aplicaciones Web: Rentabilizar Internet, <http://www.webvillage.info> .(Consulta 24/marzo/2007)
- [8] PostgreSQL Database Server. <http://www.postgresql.org>. (Consulta: 05/feb/2007).
- [9] Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico*. McGraw-Hill/Interamericana de España.
- [10] Rumbaugh, J. Ivar Jacobson, I. Grady, B (2002). *El Lenguaje Unificado de Modelado*. Manual de Referencia. (Consulta: marzo-abril/2007).
- [11] Rumbaugh, J. Ivar Jacobson, I. Grady, B (2000). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.
- [12] Sitio Oficial de Microsoft. <http://www.microsoft.com>. (Consulta: 06/feb/2007).
- [13] Sitio Oficial de MySQL. <http://www.mysql.org>. (Consulta: 07/feb/2007).
- [14] Web Oficial de PHP. <http://www.php.net>. (Consulta: 05/may/2007).
- [15] Wikipedia, la enciclopedia libre. <http://www.eswikipedia.org> (Consulta: Febrero- Mayo, 2008).

## Anexo 1

<b>Nombre del caso de uso</b>	Eliminar cuenta de usuario
<b>Actores</b>	administrador (inicia)
<b>Propósito</b>	Darle baja a un usuario.
<b>Resumen</b>	El caso de uso inicia cuando el administrador decide darle baja del sistema a algún usuario, se requerirá confirmación y se borrarán los registros del usuario de la base de datos.
<b>Referencias</b>	R1
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se eliminará un registro de usuario

<b>Nombre del caso de uso</b>	buscar usuario
<b>Actores</b>	administrador (inicia)
<b>Propósito</b>	Buscar a un usuario matriculado en el sistema.
<b>Resumen</b>	El caso de uso inicia cuando el administrador decide buscar a un usuario determinado del sistema dando como referencia el nombre de usuario.
<b>Referencias</b>	R2
<b>Precondiciones</b>	El usuario debe ser válido. El usuario a buscar debe existir.
<b>Poscondiciones</b>	Se mostrará el usuario buscado.

<b>Nombre del caso de uso</b>	Actualizar Información de usuario
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Actualizar Información del perfil de usuario.
<b>Resumen</b>	
El caso de uso inicia cuando alguno de los usuarios del sistema decide actualizar su información personal de usuario, y termina cuando el sistema ha actualizado estos datos.	
<b>Referencias</b>	R3
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Quedan actualizados los datos del usuario.

<b>Nombre del caso de uso</b>	Validar Usuario
<b>Actores</b>	usuarioP y Administrador (inician)
<b>Propósito</b>	Validar Usuario
<b>Resumen</b>	
El caso de uso inicia cuando algún usuario o el administrador deciden ejecutar una acción y se requiere verificar que este tenga permisos para ello.	
<b>Referencias</b>	R6
<b>Poscondiciones</b>	Se validado o autentifica el usuario.
<b>Nombre del caso de uso</b>	Validar Usuario

<b>Nombre del caso de uso</b>	Actualizar bienvenida
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Cambiar el contenido de la página principal, o sea, el texto de bienvenida.
<b>Resumen</b>	
El Administrador puede definir o actualizar el texto o el contenido de la página principal propiciando flexibilidad.	
<b>Referencias</b>	R4
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Queda definido el aspecto de la página principal

<b>Nombre del caso de uso</b>	Cambiar contraseña
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Cambiar contraseña de usuario
<b>Resumen</b>	
El caso de uso inicia cuando algún usuario del sistema decide cambiar su contraseña, para esto debe escribir contraseña anterior, su nueva contraseña y confirmar la nueva contraseña.	
<b>Referencias</b>	R5
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se actualiza la contraseña de usuario en la base de datos

<b>Nombre del caso de uso</b>	Actualizar proyecto
<b>Actores</b>	administrador (inicia)
<b>Propósito</b>	Actualiza la información de un proyecto en base de datos..
<b>Resumen</b> El caso de uso inicia cuando el administrador decide actualizar la información de un proyecto determinado en la base de datos.	
<b>Referencias</b>	R7
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se actualizará el registro de un proyecto.

<b>Nombre del caso de uso</b>	eliminar proyecto
<b>Actores</b>	administrador (inicia)
<b>Propósito</b>	Borra la información de un proyecto de la base de datos.
<b>Resumen</b> El caso de uso inicia cuando el administrador decide darle baja de la base de datos a la información de un proyecto determinado.	
<b>Referencias</b>	R7
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se borrará el registro correspondiente al proyecto seleccionado.



<b>Nombre del caso de uso</b>	Agregar proyecto
<b>Actores</b>	administrador (inicia)
<b>Propósito</b>	Guarda la información de un proyecto en la base de datos (BD).
<b>Resumen</b> El caso de uso inicia cuando el administrador inserta en la BD la información de un determinado proyecto.	
<b>Referencias</b>	R7
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se tendrá un nuevo registro de un proyecto en la BD.

<b>Nombre del caso de uso</b>	Enviar petición
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Establecer cierta comunicación entre el usuarioP y el administrador.
<b>Resumen</b> El caso de uso inicia cuando el usuarioP decide enviar alguna petición o mensaje al administrador.	
<b>Referencias</b>	R8
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se le enviará un mensaje al administrador.

## Anexo 2

<b>Nombre del caso de uso</b>	Renombrar archivo.
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	
<b>Resumen.</b>	El caso de uso se inicia cuando algún administrado o trabajador decide renombrar un paquete del Repositorio
<b>Referencias</b>	R2
<b>Precondiciones</b>	Debe introducir un nombre válido. El usuario debe ser válido.
<b>Poscondiciones</b>	Se renombra un paquete del repositorio

<b>Nombre del caso de uso</b>	Borrar archivo.
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Borrar un paquete
<b>Resumen.</b>	El caso de uso se inicia cuando algún administrador o trabajador decide borrar un paquete del Repositorio
<b>Referencias</b>	R3
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se ha borrado un paquete del repositorio

<b>Nombre del caso de uso</b>	Buscar archivo
<b>Actores</b>	Usuario (inicia)
<b>Propósito</b>	Buscar paquete en el repositorio.
<b>Resumen.</b> El caso de uso se inicia cuando algún usuario decide buscar un archivo, dando como referencia el nombre del mismo.	
<b>Referencias</b>	R4
<b>Precondiciones</b>	Debe existir archivo en el repositorio. El usuario debe ser válido. La entrada para la búsqueda debe ser válida.
<b>Poscondiciones</b>	Se muestra un listado de archivo.

<b>Nombre del caso de uso</b>	Descargar archivo
<b>Actores</b>	Usuario (inicia)
<b>Propósito</b>	Guardar un archivo en la PC
<b>Resumen.</b> El caso de uso se inicia cuando algún usuario quiere descargar un archivo del Repositorio a su PC.	
<b>Referencias</b>	R5
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	Se guarda un archivo en la PC

### Anexo 3

<b>Nombre del caso de uso</b>	Eliminar comentario.
<b>Actores</b>	administrador (inicia)
<b>Propósito</b>	Elimina un comentario referente a una noticia.
<b>Resumen</b>	El caso de uso inicia cuando el administrador decide borrar un comentario referente a alguna noticia publicada en el sistema.
<b>Referencias</b>	R2
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se elimina un comentario asociado a una noticia.

<b>Nombre del caso de uso</b>	Agregar comentario.
<b>Actores</b>	usuario (inicia)
<b>Propósito</b>	Agregar un comentario referente a una noticia.
<b>Resumen</b>	El caso de uso inicia cuando el usuario decide agregar un comentario referente a alguna noticia publicada.
<b>Referencias</b>	R2
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se agrega un comentario asociado a una noticia.

<b>Nombre del caso de uso</b>	Eliminar noticia
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Dejar de publicar una noticia.
<b>Resumen</b>	
Inicia el caso de uso cuando el Administrador del sistema decide que una noticia no debe ser mostrada más a los visitantes y la elimina del sistema.	
<b>Referencias</b>	R1
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Deja de publicarse la noticia.

#### Anexo 4

<b>Nombre del caso de uso</b>	Agregar Portador.
<b>Actores</b>	usuarioP
<b>Propósito</b>	Guarda la información de un portador en la BD.
<b>Resumen</b>	El caso de uso inicia cuando el usuarioP decide guardar en la BD un nuevo registro contenedor de la información un portador.
<b>Referencias</b>	R1
<b>Precondiciones</b>	Usuario validado.
<b>Poscondiciones</b>	La existencia de un nuevo registro de un portador.

<b>Nombre del caso de uso</b>	Actualizar Portador.
<b>Actores</b>	usuarioP
<b>Propósito</b>	Actualiza la información de un portador en la BD.
<b>Resumen</b>	El caso de uso inicia cuando el usuarioP decide modificar en la BD un registro correspondiente a un portador.
<b>Referencias</b>	R1
<b>Precondiciones</b>	Usuario validado.
<b>Poscondiciones</b>	Se actualizara el registro seleccionado adecuado a un portador.

<b>Nombre del caso de uso</b>	Eliminar Portador.
<b>Actores</b>	usuarioP
<b>Propósito</b>	Dar baja a un portador en la BD.
<b>Resumen</b>	
En caso de inicia cuando el usuarioP resuelve darle baja de la BD a un registro contenedor de la información de un portador.	
<b>Referencias</b>	R1
<b>Precondiciones</b>	Usuario validado.
<b>Poscondiciones</b>	Se borra de la BD la información de un portador.

<b>Nombre del caso de uso</b>	Agregar tipo portador.
<b>Actores</b>	usuarioP
<b>Propósito</b>	Guarda la información de un tipo de portador en la BD.
<b>Resumen</b>	
El caso de uso inicia cuando el usuarioP decide guardar en la BD un nuevo registro contenedor de la información un tipo de portador.	
<b>Referencias</b>	R2
<b>Precondiciones</b>	Usuario validado.
<b>Poscondiciones</b>	La existencia de un nuevo registro de un tipo de portador.

<b>Nombre del caso de uso</b>	Actualizar tipo de portador.
<b>Actores</b>	usuarioP
<b>Propósito</b>	Cambia la información de un tipo de portador en la BD.
<b>Resumen</b>	
El caso de uso inicia cuando el usuarioP quiere cambiar en la BD la información de algún tipo de portador.	
<b>Referencias</b>	R2
<b>Precondiciones</b>	Usuario validado.
<b>Poscondiciones</b>	Se actualizara el registro seleccionado correspondiente al tipo de portador.

<b>Nombre del caso de uso</b>	Eliminar tipo de portador.
<b>Actores</b>	usuarioP
<b>Propósito</b>	Dar baja a un tipo de portador en la BD.
<b>Resumen</b>	
El caso de uso inicia cuando el usuarioP resuelve darle baja de la BD a un registro contenedor de la información de un tipo de portador.	
<b>Referencias</b>	R2
<b>Precondiciones</b>	Usuario validado.
<b>Poscondiciones</b>	Se borra de la BD la información de un tipo de portador.



<b>Nombre del caso de uso</b>	Actualizar p_ind_cons.
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Modificar parámetros de índice de consumo para el cálculo del mismo.
<b>Resumen.</b> El caso de uso se inicia cuando el usuarioP decide modificar los datos de un registro referente a un índice de consumo de la base de datos. Estos datos hacen referencia a los parámetros para el cálculo posteriormente del índice de consumo de un portador.	
<b>Referencias</b>	R3
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	El registro de índice de consumo quedará actualizado.

<b>Nombre del caso de uso</b>	Agregar p_ind_cons.
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Agregar parámetros de índice de consumo para el cálculo del mismo.
<b>Resumen.</b> El caso de uso se inicia cuando el usuario decide guardar nuevos parámetros de índice de consumo para realizar el cálculo del nuevo índice de consumo de algún portador.	
<b>Referencias</b>	R3
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Existencia de un nuevo registro de un índice de consumo en la BD.

<b>Nombre del caso de uso</b>	Eliminar p_ind_cons.
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Borrar parámetros de un índice de consumo
<b>Resumen.</b> El caso de uso se inicia cuando el usuarioP decide darle baja a un registro con los parámetros de un índice de consumo.	
<b>Referencias</b>	R3
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se borrara de la base de datos el registro.

<b>Nombre del caso de uso</b>	Mostrar índice de consumo.
<b>Actores</b>	usuarioP
<b>Propósito</b>	Mostrar los índices de consumo de un portador energético.
<b>Resumen</b> El caso de uso lo inicia el usuarioP cuando desea ver los índices de consumo de un determinado portador y el sistema efectúa el cálculo de los mismos correspondientes al portador y lo muestra.	
<b>Referencias</b>	3
<b>Precondiciones</b>	El usuario debe ser valido.
<b>Poscondiciones</b>	Se muestran los índices de consumos.

<b>Nombre del caso de uso</b>	agregar indicador
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Agregar un nuevo indicador.
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide agregar un nuevo indicador a la BD.	
<b>Referencias</b>	4
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se adiciona un nuevo registro a la BD.

<b>Nombre del caso de uso</b>	actualizar indicador
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Editar la información referente a un indicador.
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide modificar la información de un indicador.	
<b>Referencias</b>	4
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se actualiza la información en la BD de registro seleccionado.

<b>Nombre del caso de uso</b>	eliminar indicador
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Borrar la información referente a un indicador.
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide dar baja a la información de un indicador.	
<b>Referencias</b>	4
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se borraría de la información de la BD de registro seleccionado.

<b>Nombre del caso de uso</b>	agregar tipo_indicador
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Agregar un nuevo tipo de indicador.
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide agregar un nuevo tipo de indicador a la BD.	
<b>Referencias</b>	5
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se adiciona un nuevo registro a la BD.

<b>Nombre del caso de uso</b>	Actualizar tipo_indicador
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Editar la información referente a un indicador.
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide modificar la información de un indicador.	
<b>Referencias</b>	5
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se actualiza la información en la BD de registro seleccionado.

<b>Nombre del caso de uso</b>	Eliminar tipo_ indicador
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Borrar la información referente a un tipo de indicador.
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide dar baja a la información de un tipo de indicador.	
<b>Referencias</b>	5
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se borrar de la información de la BD de registro seleccionado.

<b>Nombre del caso de uso</b>	eliminar info.G
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	eliminar información general de un portador
<b>Resumen.</b> El caso de uso inicia cuando el usuarioP decide dar baja a la información general del portador energético. Esta información son los datos generales como resultado del consumo del portador durante todo un mes.	
<b>Referencias</b>	6
<b>Precondiciones</b>	El usuario debe ser válido
<b>Poscondiciones</b>	Se modifica el registro seleccionado.

<b>Nombre del caso de uso</b>	Buscar datos de portador
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Buscar los datos de un portador
<b>Resumen.</b> El caso de uso se inicia cuando el usuarioP decide buscar los datos asociados a un portador, dando como referencia el nombre del portador.	
<b>Referencias</b>	7
<b>Precondiciones</b>	El usuario debe ser válido. La entrada para la búsqueda debe ser válida.
<b>Poscondiciones</b>	Se muestra un listado de registros.

<b>Nombre del caso de uso</b>	Generar gráficos de conductas.
<b>Actores</b>	usuarioP (inicia)
<b>Propósito</b>	Generar gráficos de barras del comportamiento de los portadores utilizando algunos datos numéricos de un portador.
<b>Resumen</b> El caso de uso inicia cuando el usuarioP decide ver conductas de un portador energético en cualquier período de tiempo y se genera el gráfico correspondiente.	
<b>Referencias</b>	R8
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	Se visualiza el comportamiento de portador deseado en un grafico de paretto.

<b>Nombre del caso de uso</b>	Visualizar historial.
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Ver los accesos de usuarioP a la base de datos.
<b>Resumen.</b> El caso de uso se inicia cuando el administrador decide ver los registros de accesos y acciones realizadas en la BD.	
<b>Referencias</b>	R9
<b>Precondiciones</b>	El usuario debe ser válido.
<b>Poscondiciones</b>	Se listan los registros de las acciones realizadas y accesos a la BD.

<b>Nombre del caso de uso</b>	Mostrar proyectos.
<b>Actores</b>	usuario
<b>Propósito</b>	Muestra la información de los proyectos en ejecución registrados en la BD.
<b>Resumen</b>	
El caso de uso lo inicia el usuario cuando desea ver los proyectos que están en curso, mostrándole así sus características.	
<b>Referencias</b>	R10
<b>Precondiciones</b>	-
<b>Poscondiciones</b>	Se visualizan los proyectos existentes.



## Anexo 5

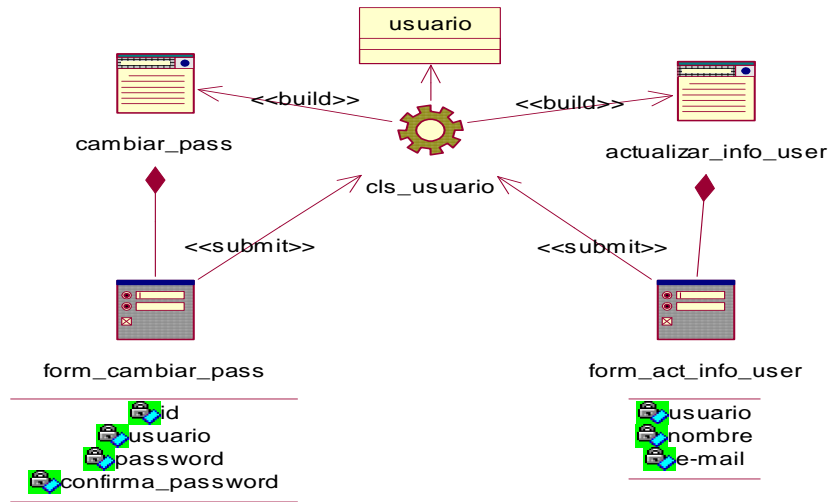


Fig. #1 Diagrama clase Web actualizar datos de usuario y cambiar password.

### Diagramas clase Web del paquete Gestión energética:

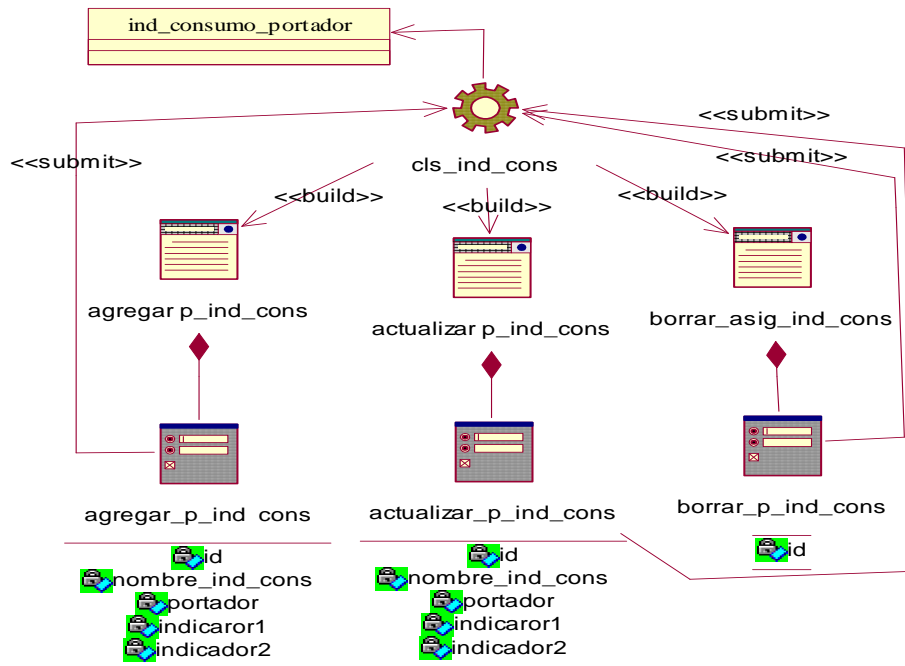
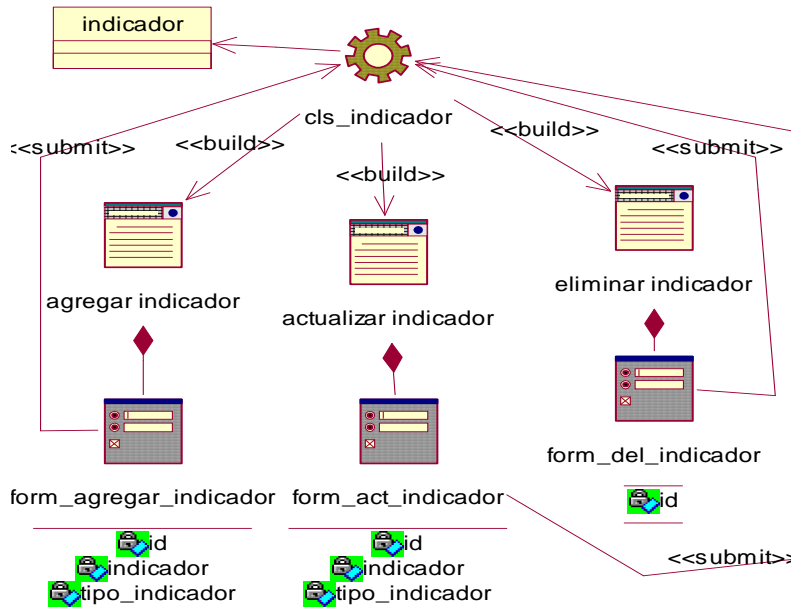
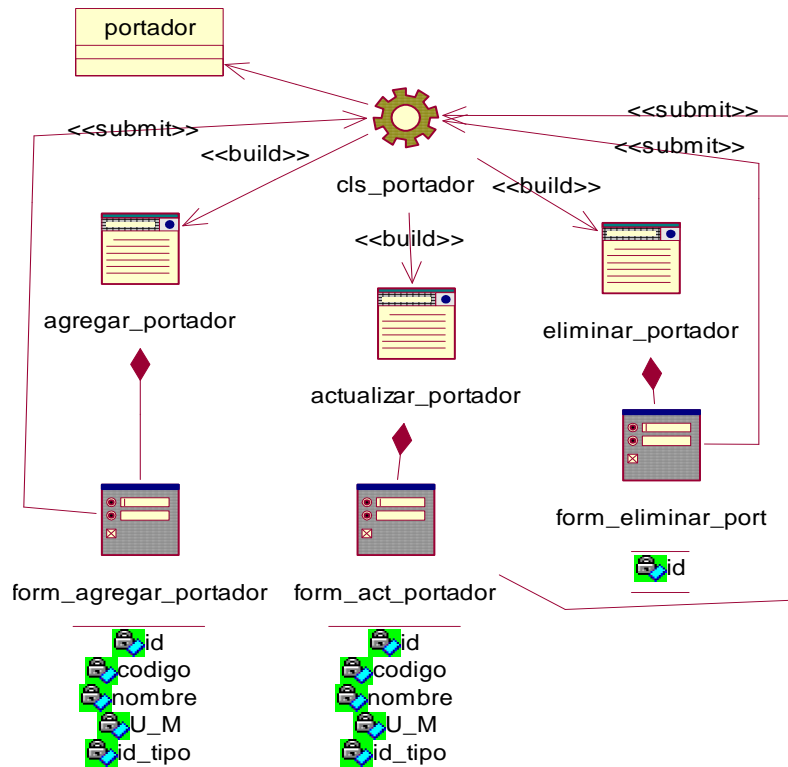


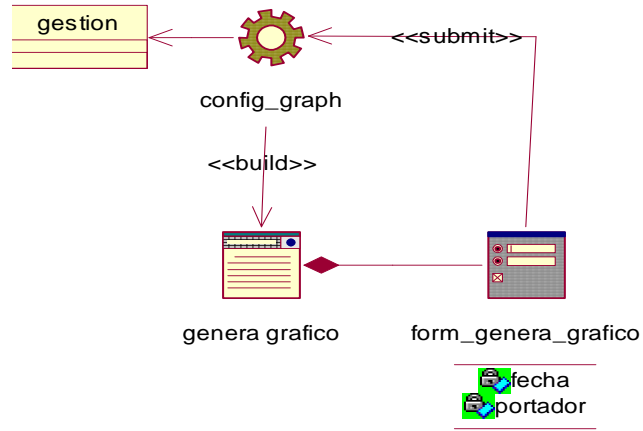
Fig. #2 Diagrama clase web agregar, eliminar y actualizar parámetros índice de consumo.



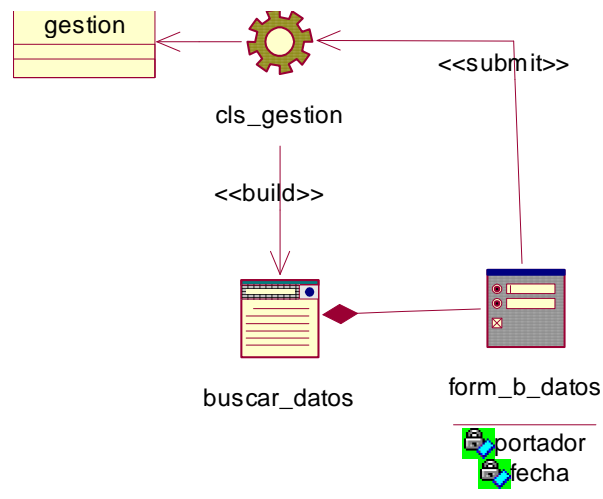
**Fig. # 3** Diagrama clase Web agregar, eliminar y actualizar indicador.



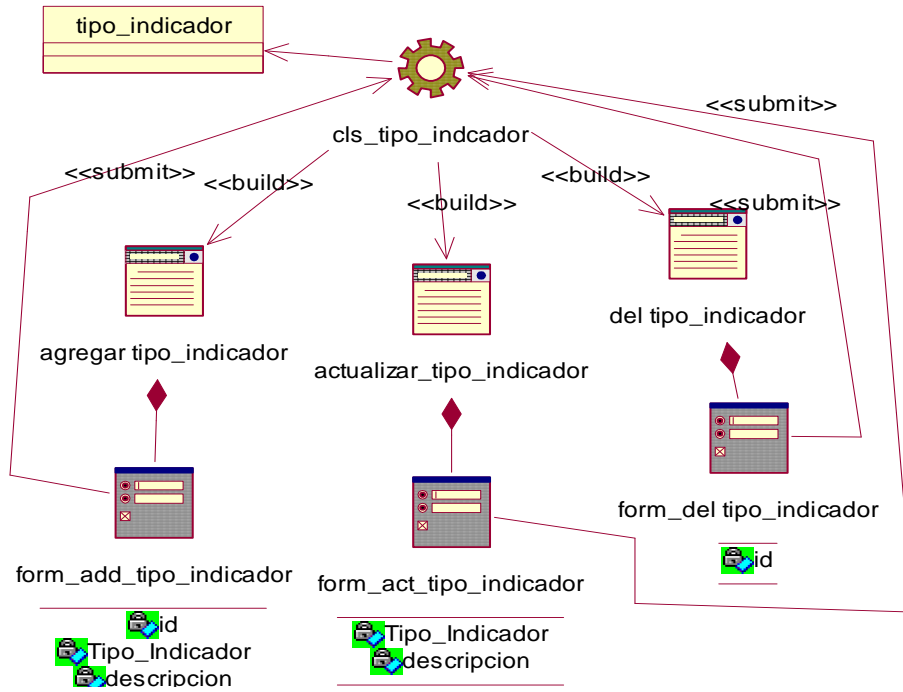
**Fig. # 4** Diagrama clase web agregar portador, actualizar portador y eliminar portador.



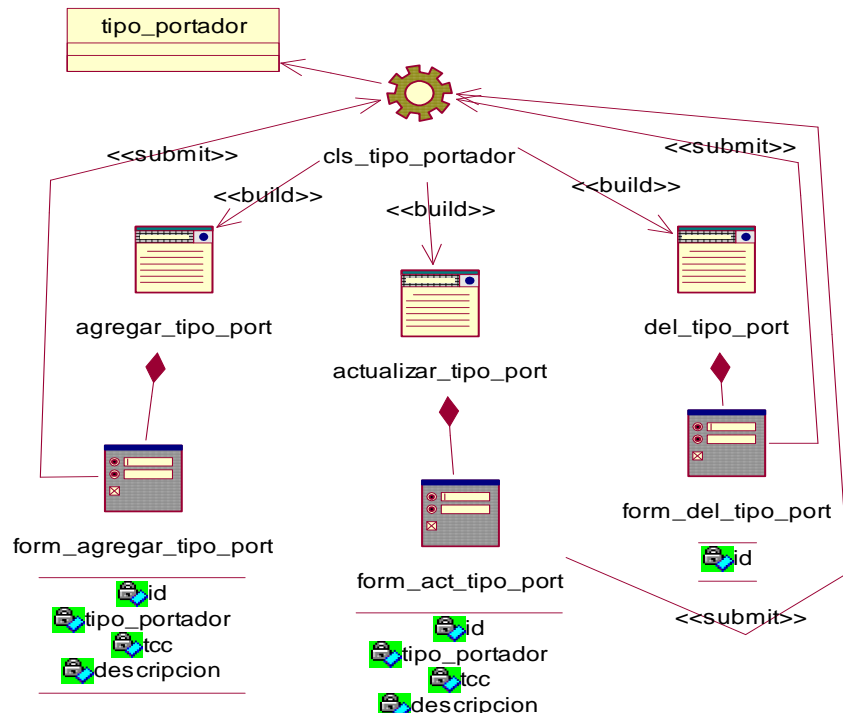
**Fig. # 5** Diagrama clase Web generar gráfico.



**Fig. # 5** Diagrama clase Web buscar datos portador.



**Fig. # 6** Diagrama clase Web agregar, actualizar y eliminar tipo indicador.



**Fig. # 7** Diagrama clase Web agregar, actualizar y eliminar tipo portador.

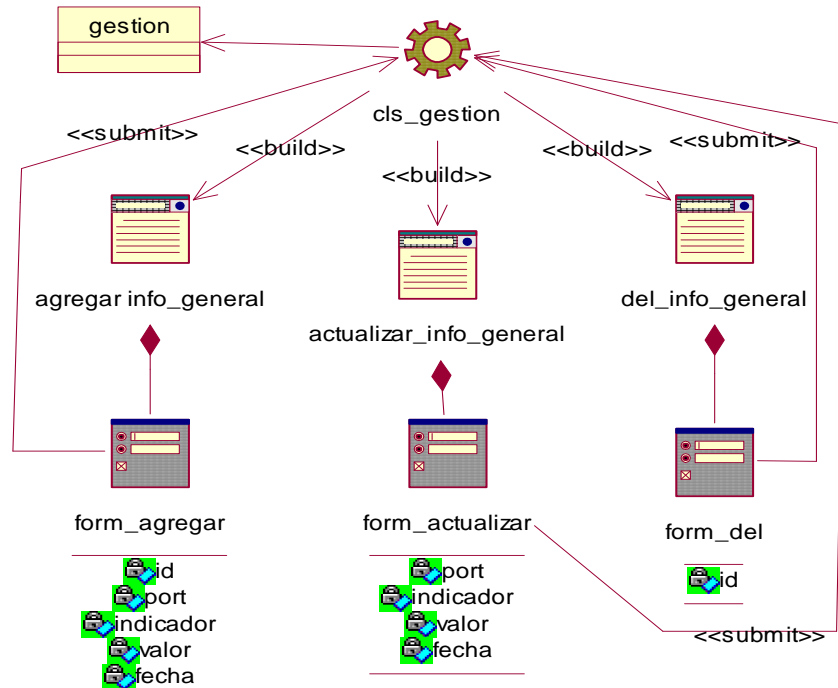


Fig. # 8 Diagrama clase Web agregar, actualizar y eliminar información general.

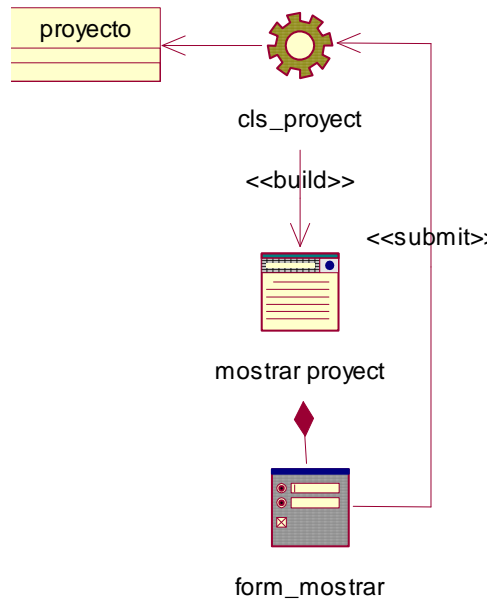


Fig. # 9 Diagrama clase Web mostrar proyecto,

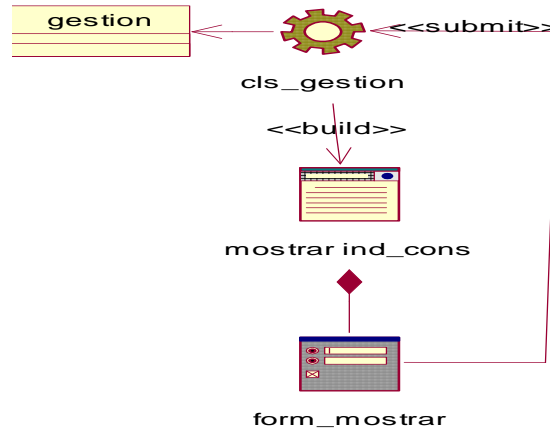


Fig. # 10 Diagrama clase Web mostrar índice de consumo.

**Diagramas clase Web del paquete Repositorio de archivos:**

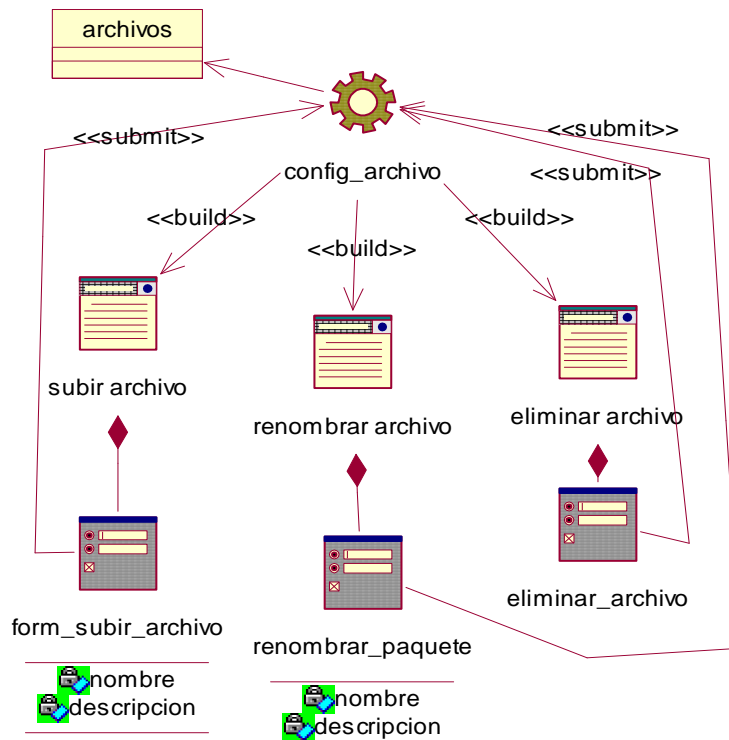


Fig. # 11 Diagrama clase Web subir, renombrar y eliminar archivos del repositorio.

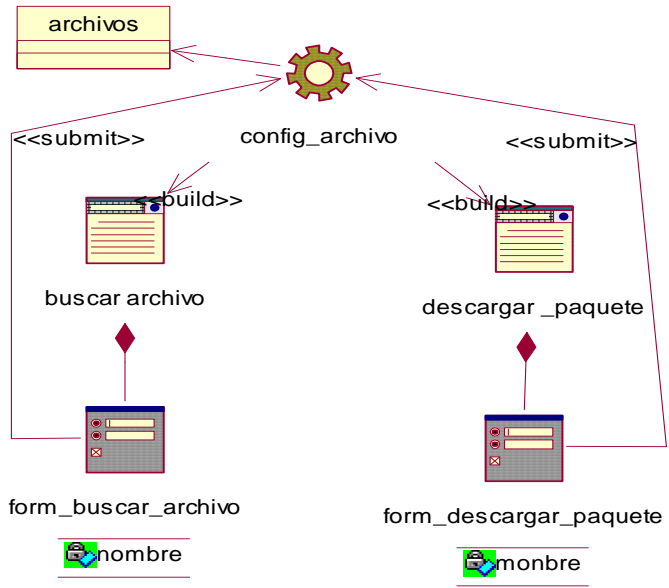


Fig. # 12 Diagrama clase Web buscar y descargar archivos.

**Diagramas clase Web del paquete Noticia:**

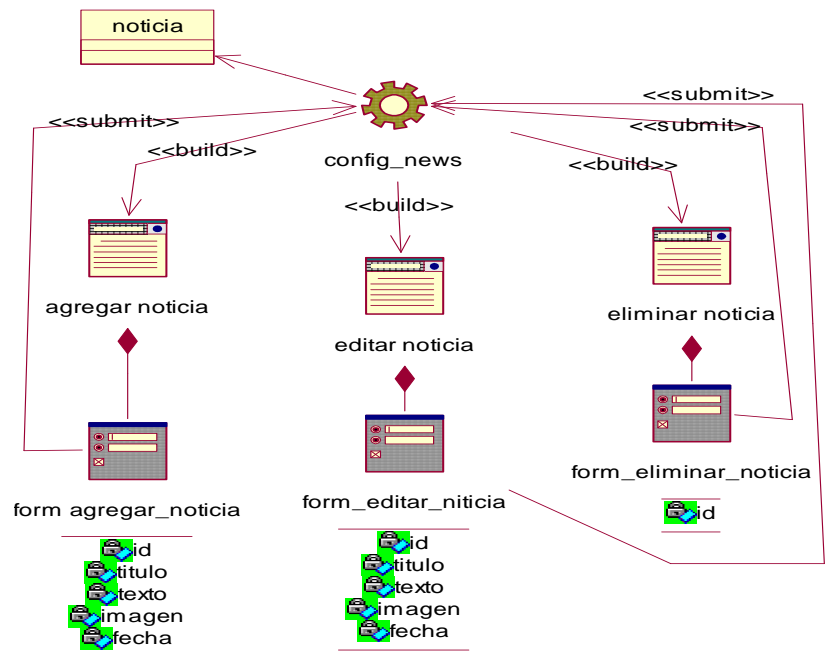
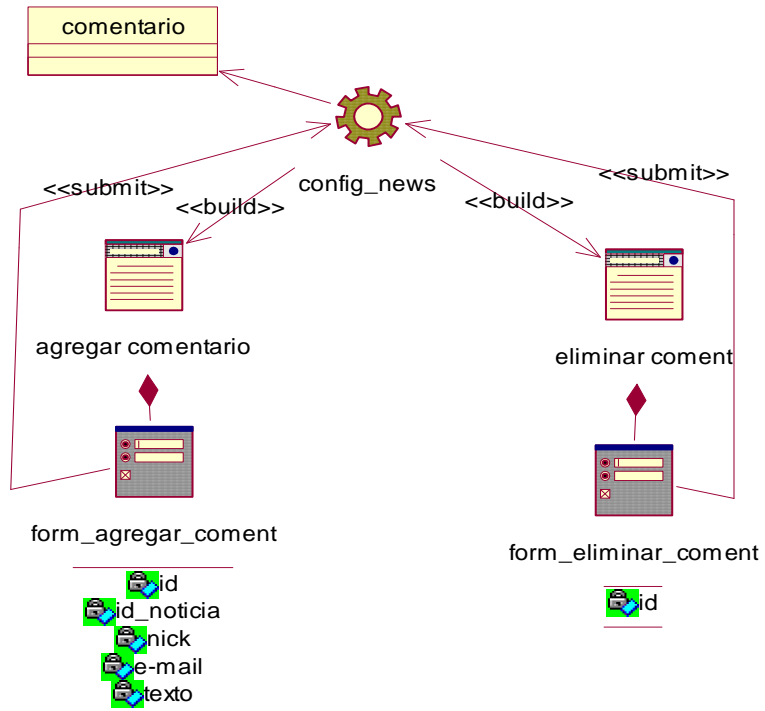


Fig. # 13 Diagrama clase web agregar, actualizar, eliminar noticia.



**Fig. # 14** Diagrama clase web agregar, eliminar comentario de noticia.



## Anexo 6

### Diagramas de secuencia:

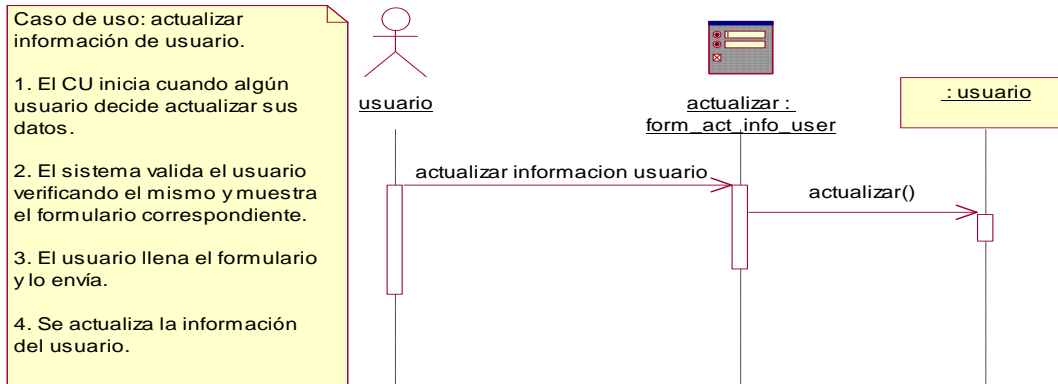


Fig. # 1 Diagrama secuencia actualizar información de usuario.

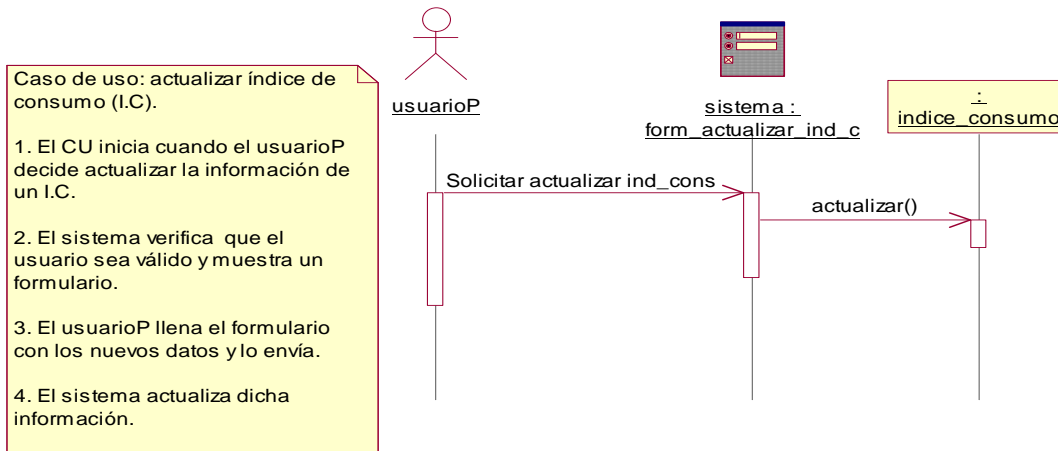


Fig. # 2 Diagrama secuencia actualizar información índice de consumo.

Caso de uso: agregar indicador.

1. El CU inicia cuando el usuarioP decide agregar un nuevo indicador al sistema.
2. El sistema verifica que el usuario sea válido y muestra un formulario.
3. El usuarioP llena el formulario y lo envía.
4. El sistema añade un nuevo indicador.

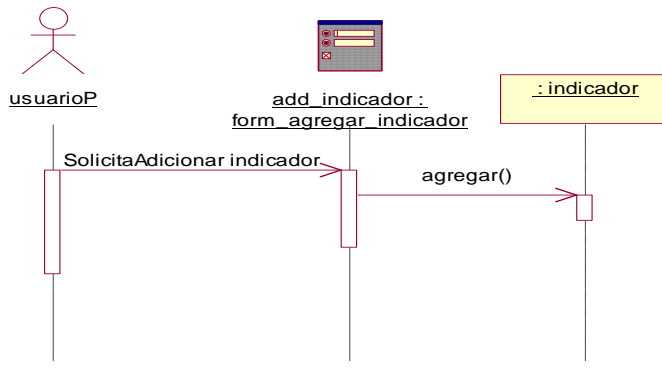


Fig. # 3 Diagrama secuencia agregar indicador.

Caso de uso: agregar portador.

1. El CU inicia cuando el usuarioP decide agregar un nuevo portador al sistema.
2. El sistema verifica que el usuario sea válido y muestra un formulario.
3. El usuarioP llena el formulario con los datos correspondientes y lo envía.
4. El sistema añade un nuevo portador.

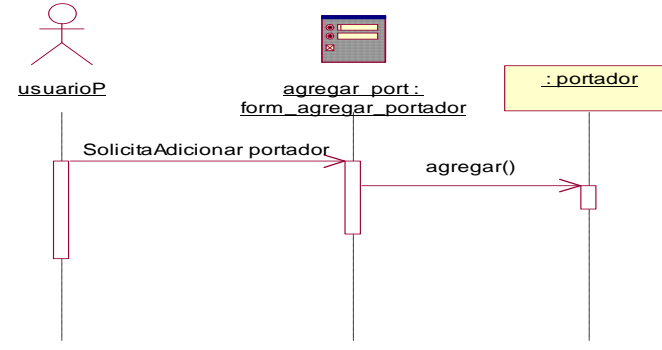


Fig. # 4 Diagrama secuencia agregar portador.

Caso de uso: generar grafico.

1. El CU inicia cuando el usuarioP decide ver el comportamiento del consumo de un portador.
2. El sistema muestra un formulario con los parametros requeridos a introducir.
3. El usuario llena el formulario y genera el grafico.

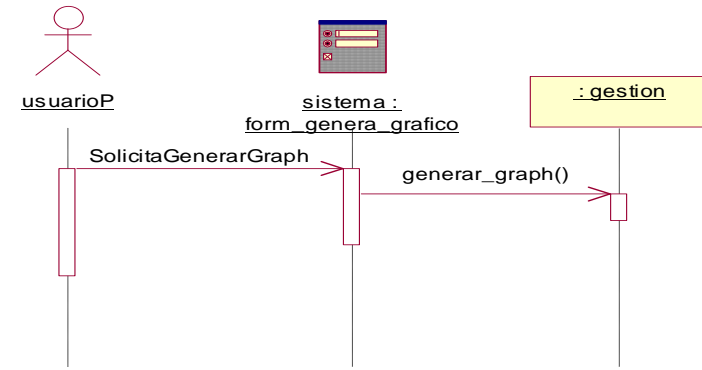
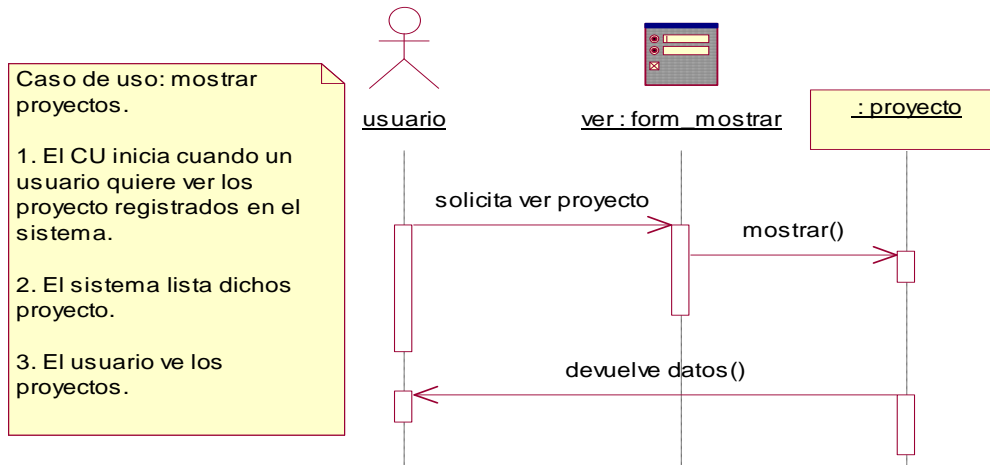
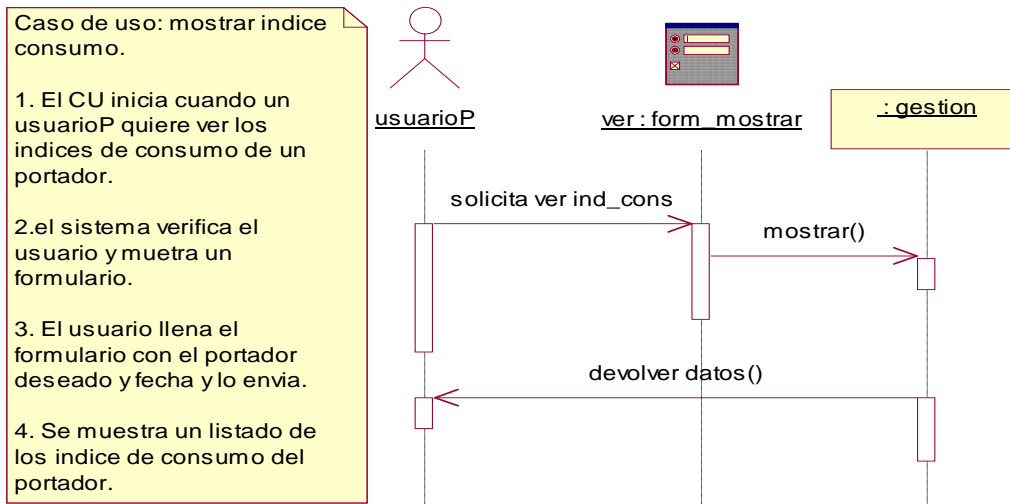


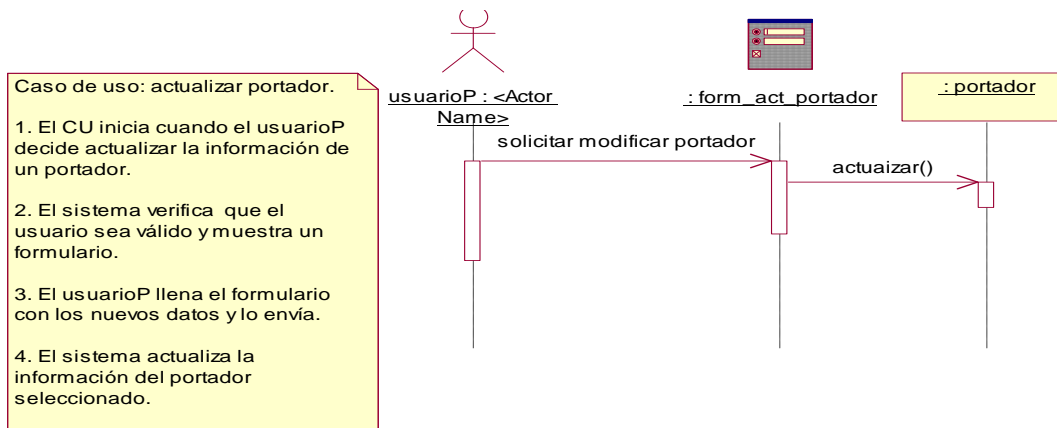
Fig. # 5 Diagrama secuencia generar gráfico.



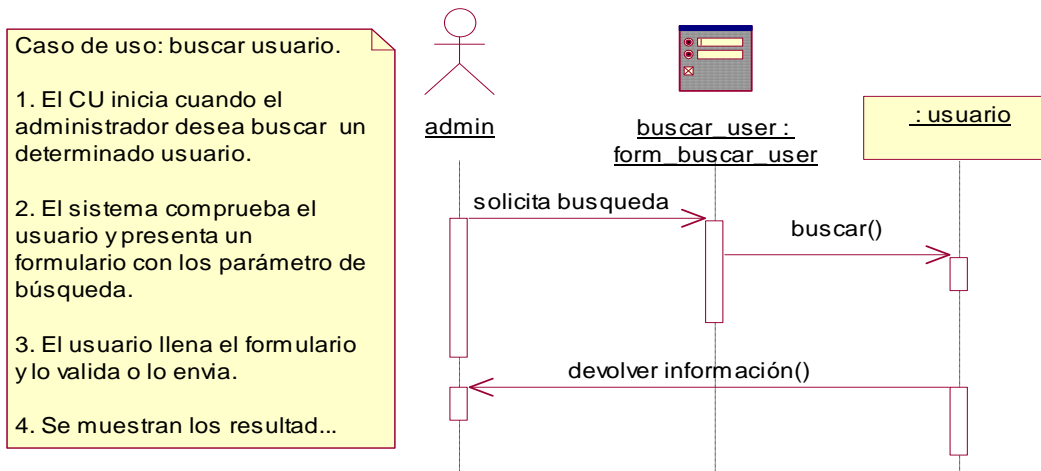
**Fig. # 5** Diagrama secuencia mostrar proyecto.



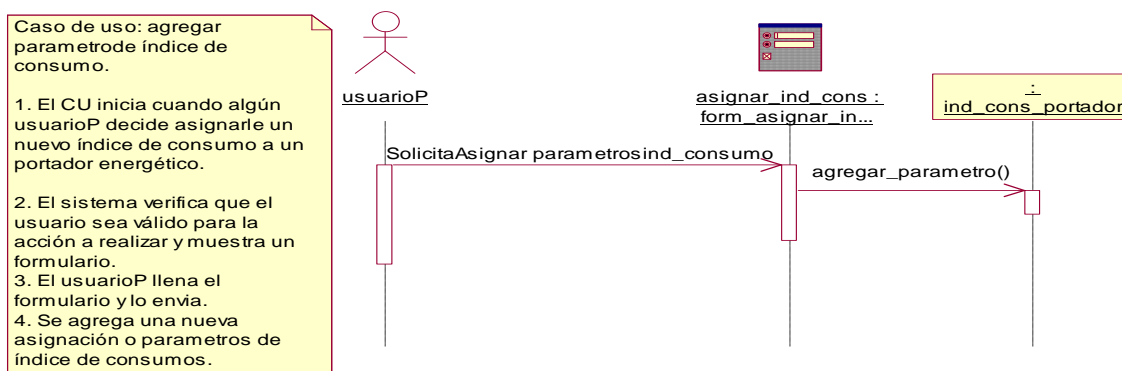
**Fig. # 6** Diagrama secuencia mostrar índice de consumo.



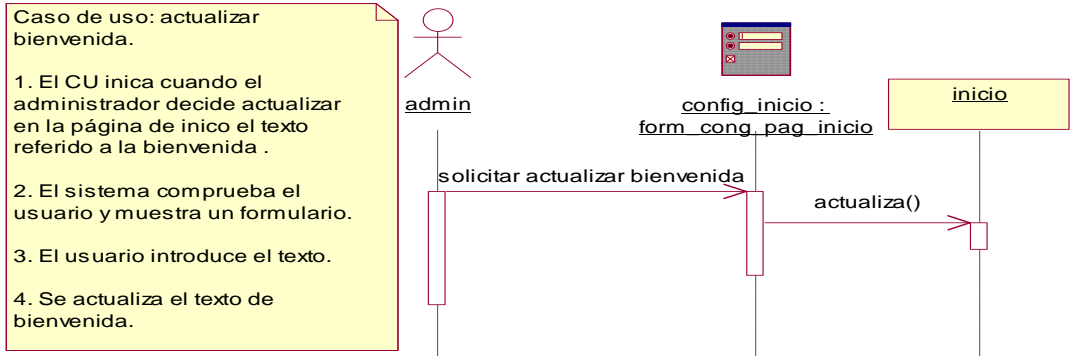
**Fig. # 7** Diagrama secuencia actualizar portador.



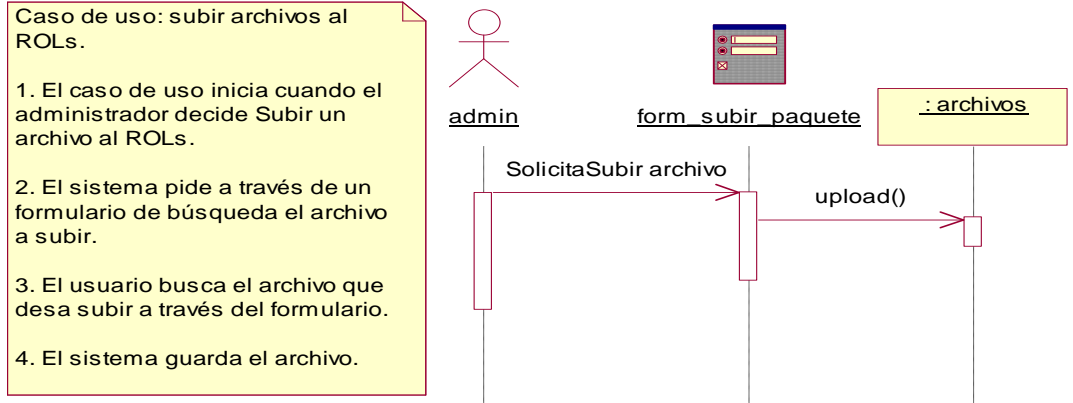
**Fig. # 8** Diagrama secuencia buscar usuario



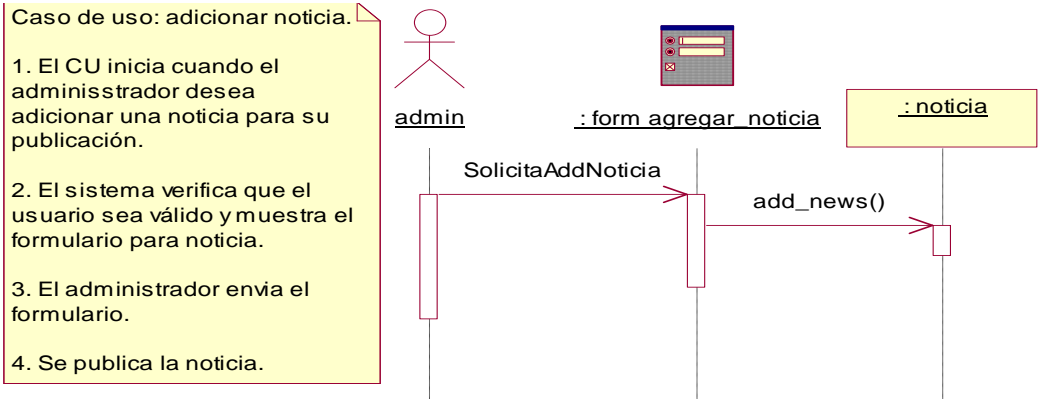
**Fig. # 9** Diagrama secuencia agregar parámetros de índice de consumo.



**Fig. # 10** Diagrama secuencia actualizar bienvenida.



**Fig. # 11** Diagrama secuencia subir archivo al repositorio.



**Fig. # 12** Diagrama secuencia subir paquete al repositorio.

## **Anexo 7**

<b>Nombre Clase</b>	tipo_portador
<b>Propósito</b>	Almacenar los datos del tipo de portador.
<b>atributo</b>	<b>Descripción</b>
Id_tipo_port	Es el identificador del tipo portador.
tipo_portador	Es el nombre del tipo de portador.
tcc	Valor numérico del factor de conversión de los portadores.
descripcion	Breve descripción del tipo.

<b>Nombre Clase</b>	indicador
<b>Propósito</b>	Almacenar los datos del indicador.
<b>atributo</b>	<b>Descripción</b>
Id_indicador	Es el identificador del indicador.
indicador	Es el nombre del tipo del indicador.
id_tipo_indicador	Identificador del tipo indicador.

<b>Nombre Clase</b>	tipo_indicador
<b>Propósito</b>	Almacenar los datos del tipo indicador.
<b>atributo</b>	<b>Descripción</b>
Id_tipo_ind	Es el identificador del tipo indicador.
Tipoindicador	Es el nombre del tipo del tipo indicador.
descripción	Breve descripción o comentario del tipo.

<b>Nombre Clase</b>	ind_cons_portador
<b>Propósito</b>	Almacenar los datos de la asignación de un índice de consumo a un portador.
<b>atributo</b>	<b>Descripción</b>
Id_ind_port	Es el identificador de la asignación.
id_ind_cons	Identificador del índice de consumo.
id_portador	Identificador del portador.
indicador1	Indicador que actuara como numerador en la razón matemática para el cálculo del índice de consumo.
indicador2	Indicador que actuara como denominador en la razón matemática.

<b>Nombre Clase</b>	indice_consumo
<b>Propósito</b>	Almacenar los datos del tipo indicador.
<b>atributo</b>	<b>Descripción</b>
Id_ind_cons	Es el identificador del índice de consumo.
nombre	Es el nombre del índice de consumo.
descripción	Breve descripción o comentario del índice de consumo.

<b>Nombre Clase</b>	usuario
<b>Propósito</b>	Almacenar los datos de los usuarios.
<b>atributo</b>	<b>Descripción</b>
Id_usuario	Es el identificador del usuario.
usuario	Es el nombre usuario.
pass	Contraseña de acceso del usuario.
nombre	Nombre y apellidos del usuario.
e-mail	Dirección de correo electrónico del usuario.
nivel_acceso	Identificador del nivel de acceso a que pertenece.



<b>Nombre Clase</b>	nivel_acceso.
<b>Propósito</b>	Almacenar los datos de los usuarios.
<b>atributo</b>	<b>Descripción</b>
Id_nivel	Es el identificador del nivel de acceso.
nivel_acceso	Es el nombre del nivel de acceso.
Descripción	Descripción del nivel de acceso.

<b>Nombre Clase</b>	mensajes
<b>Propósito</b>	Almacenar los mensajes o los datos de contactos emitido por los usuarios.
<b>atributo</b>	<b>Descripción</b>
Id_mensaje	Es el identificador del mensaje.
id_usuario	Identificador de usuario.
nombre_remitente	Nombre del usuario remitente.
nombre_destino	Nombre del usuario destinatario.
asunto	Asunto del mensaje.
texto	Contenido del mensaje.
fecha	Fecha en la que fue emitido el mensaje.
ip	Numero IP de la PC desde donde se emite o se redacta el mensaje.

<b>Nombre Clase</b>	noticia
<b>Propósito</b>	Almacenar las noticias registradas o publicadas.
<b>atributo</b>	<b>Descripción</b>
Id_noticia	Es el identificador de la noticia.
titulo	Titulo de la noticia.
texto	Contenido de la noticia.
fecha	Fecha en la que fue publicada la noticia.
imagen	Pequeña imagen que acompaña la noticia en caso de que la tenga.

<b>Nombre Clase</b>	comentario
<b>Propósito</b>	Almacenar los comentarios realizado por los usuario de las noticias.
<b>atributo</b>	<b>Descripción</b>
Id_coment	Es el identificador del comentario.
id_noticia	Identificador de la noticia.
nick	Usuario que realizó el comentario.
texto	Contenido del comentario.
e-mail	Dirección de correo electrónico del usuario.
ip	Dirección IP desde donde se realizo el comentario.

<b>Nombre Clase</b>	archivos
<b>Propósito</b>	Almacenar ciertas características de los archivos subidos al repositorio.
<b>atributo</b>	<b>Descripción</b>
Id_archivo	Es el identificador del archivo o paquete.
nombre_archivo	Identificador de la noticia.
descripcion	Breve descripción o comentario del paquete.

## Anexo 8

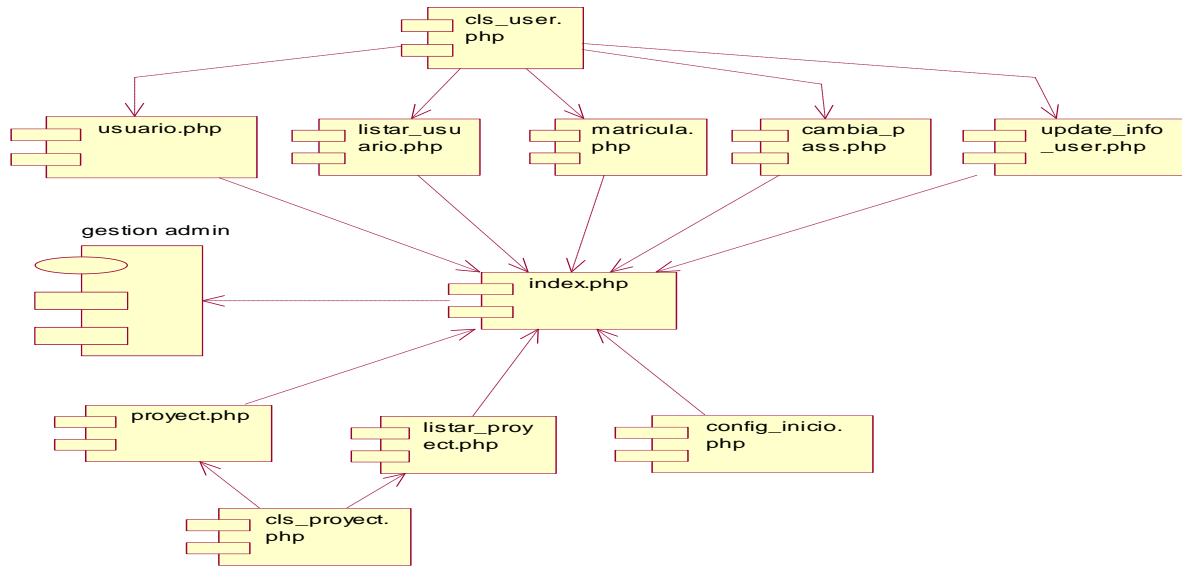


Fig. # 1 Diagrama de componente gestión admin-usuario.

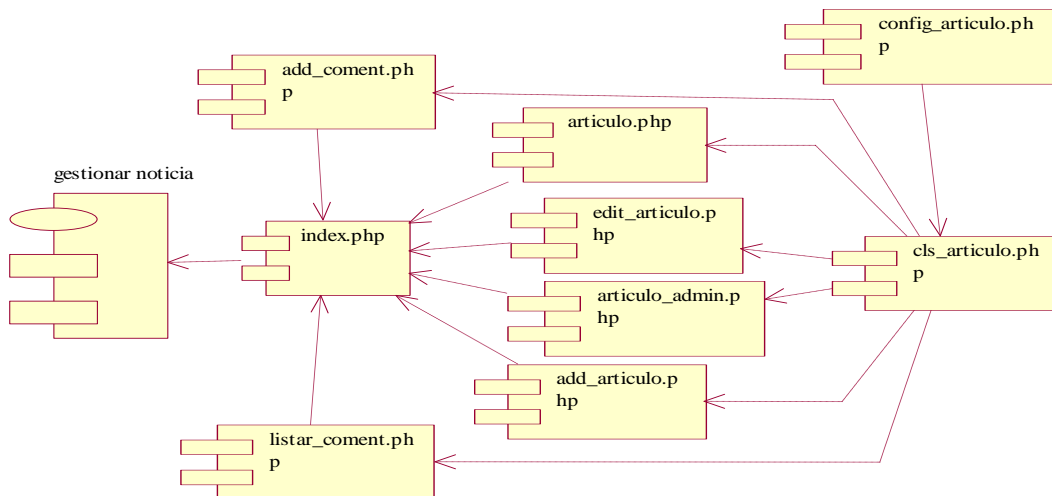
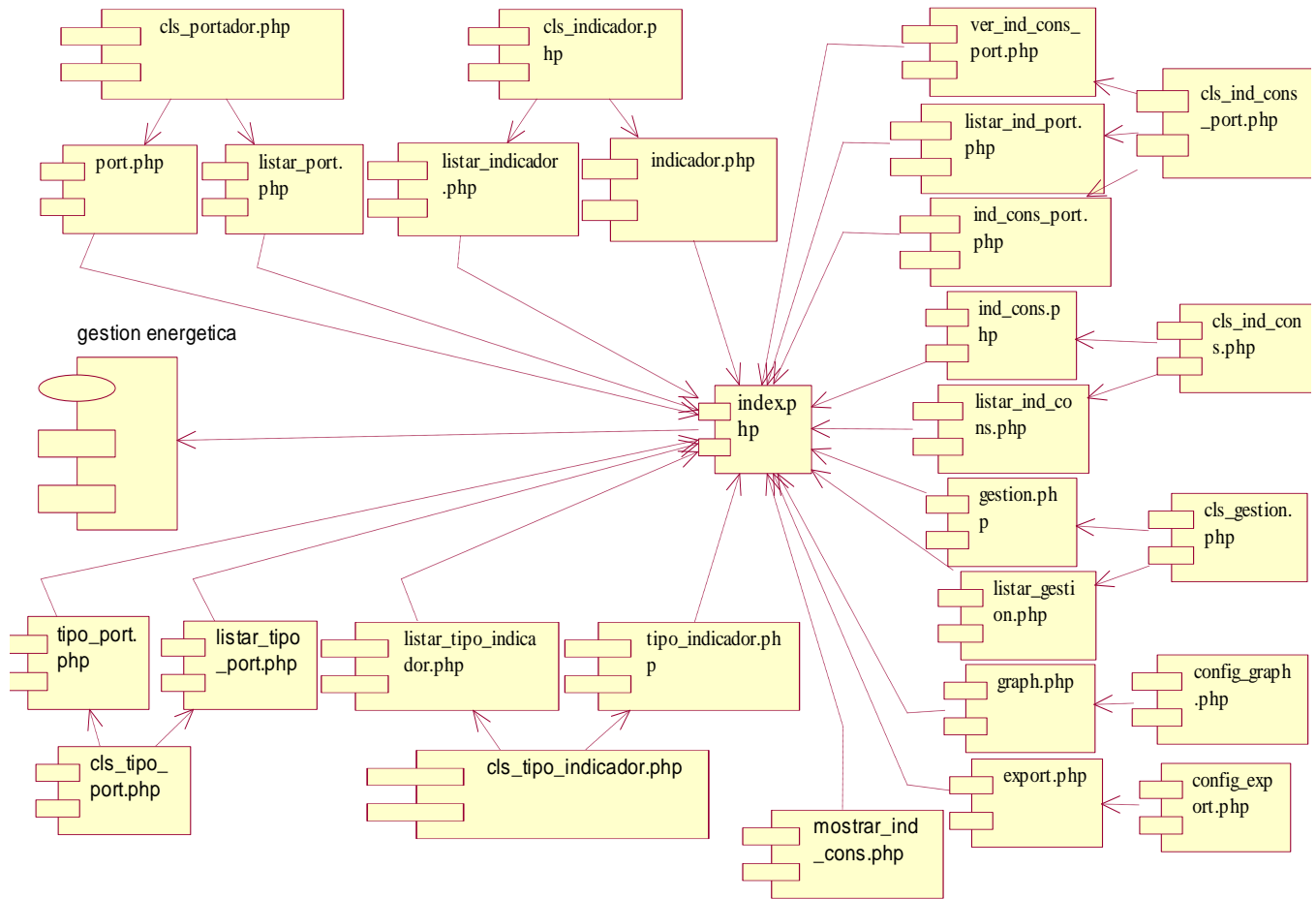
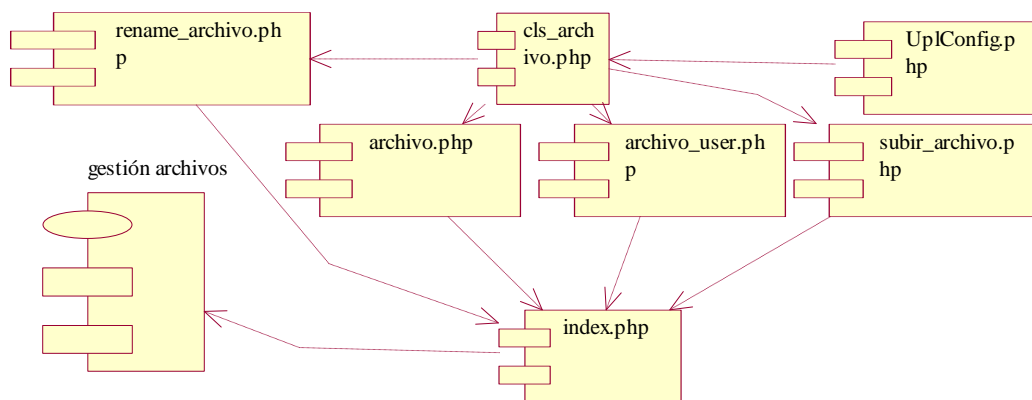


Fig. # 2 Diagrama de componente gestión noticias.



**Fig. # 3** Diagrama de componente gestión energética.



**Fig. # 4** Diagrama de componente gestión repositorios de archivos (ROLs).