



Universidad de las Ciencias Informáticas  
Centro de Informatización Universitaria (CENIA)  
Facultad 1

# Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

**Título: Procesos Planificación, Gestión del consumo y  
Configuración para el Sistema de Portadores  
Energéticos.**

**Autores: Enma Lidia Muñoz García  
Noél Hernández Bautista**

**Tutores: Ing. Adilén Sánchez Ramírez  
Ing. Enmanuel Azahares Reyes**



*“La grandeza de un hombre no se mide por el terreno que ocupan sus pies, sino por el horizonte que descubren sus ojos”*

*José Martí*

## *Dedicatoria*

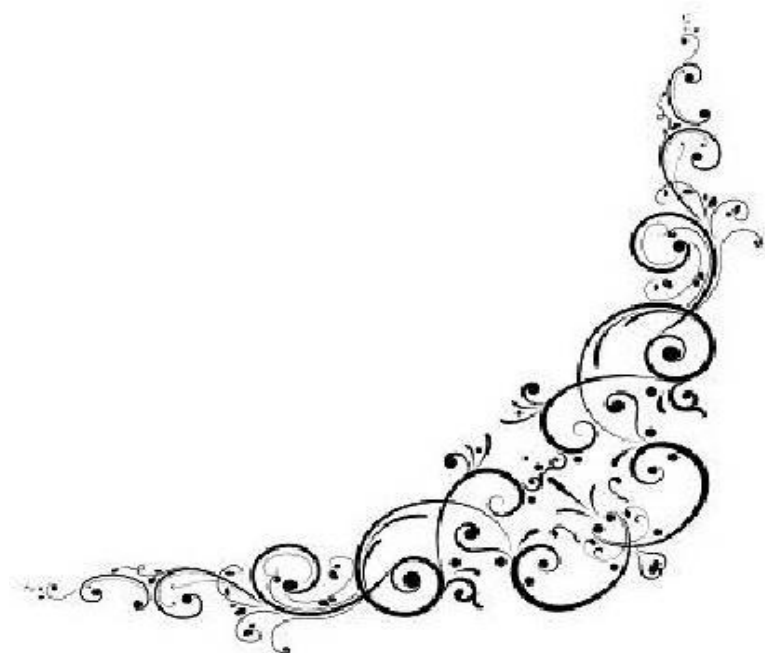


A mis padres Gladys y Noel por depositar toda su confianza en mí, por ser los primeros y mejores maestros de mi vida, por su amor y apoyo incondicional, por ser los mejores padres del mundo. A mi hermanita Nailyn por su amor y cariño. A Daímy por ser una parte de mí, confiar en mí, ayudarme en todo, por su amor, cariño y sinceridad. A toda mi familia por ayudarme siempre en especial a todos mis tíos, tías y abuelos.

Noel

A mi familia, por su apoyo incondicional durante todos estos años.

Enma



# *Agradecimientos*

## **Noel**

*A mis padres: por depositar toda su confianza en mí, por ser los primeros y mejores maestros de mi vida, por su amor y apoyo incondicional, por ser los mejores padres del mundo y guiarme por el buen camino.*

*A mi hermana: por ser tan cariñosa y quererme tanto.*

*A Daimy: por ser una parte de mí, por su confianza, ayudarme en todo, por su amor, cariño y sinceridad.*

*A mi compañera de tesis: sin su apoyo no fuera posible la realización del presente trabajo.*

*A mi familia: por ayudarme y apoyarme siempre en todo.*

*A mis tutores: por ser mi guía en el desarrollo del presente trabajo.*

*A todos mis amigos: por estar siempre ahí para mí.*

*A todos aquellos que de una forma u otra hay formado parte de éste gran triunfo,  
Muchas gracias.*

## **Enma:**

*Sobre todas las cosas muchas gracias a Dios por ser mi guía.*

*A mi mamá y mis dos papás muchas gracias por dar su vida por mí.*

*A mi tía, mi abuela, Ele, “las fieras”, Chagui, el Nene, mis abuelos de Bayamo, Manu y el resto de mi familia, los quiero mucho y les agradezco que hayan estado ahí para mí.*

*A Guillermo le agradezco por haberme brindado su amistad e incondicional compañía.*

*Noe, mi compañero de tesis, muchas gracias por aguantarme durante todos estos años y ser tan buen equipo.*

*Mis amigos y compañeros de aventuras: Milena, Celi, Nesty, Pasty, Jose, Tata, el piquete del 2 y todas las muchachitas del grupo 3, Yuyu, Carlos; los voy a llevar siempre en mi corazón.*

*Un abrazo especial al piquete de los “casi graduados” por haber compartido juntos momentos tan difíciles y haber permanecido como familia: Chacón, Alexander, Gleydis, Yisel, Mily, Yuro, Pedro, Gretel.*

*En fin, muchas gracias para todos mis compañeros, profesores, compañeros de misión, mis tutores y todo aquel que de una manera u otra me ha tendido la mano alguna vez.*

# *Declaración de autoría*

Enma Lidia Muñoz García y Noél Hernández Bautista declaran que son los únicos autores del trabajo titulado: “Procesos Planificación, Gestión del consumo y Configuración para el Sistema de Portadores Energéticos” y autorizan al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas a hacer el uso que estimen pertinente con este trabajo.

Para que así conste se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Enma Lidia Muñoz García**

**Noél Hernández Bautista**

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Autor

**Adilén Sánchez Ramírez**

**Enmanuel Azahares Reyes**

\_\_\_\_\_

\_\_\_\_\_

Firma del Tutor

Firma del Tutor

En uso de las atribuciones que me están conferidas emito el siguiente:

**CERTIFICO**

**PRIMERO:** Que el producto de Software titulado: **Sistema de Portadores Energéticos v 1.0** fue objeto de **Registro** en el Centro Nacional de Derecho de Autor de Cuba (CENDA) con **Número de Registro: 297- 2012** con los **autores** siguientes:

No	Nombres y Apellidos	Carné de Identidad	Entidad
1.	Enmanuel Azahares Reyes	85052923863	UCI
2.	José Alejandro García Calderón	85112011627	UCI
3.	Adilen Sánchez Ramírez	84082512253	UCI
4.	David Guilarte Labañino	88011500302	UCI
5.	Ivis Cañizares Riveras Hidalgo	87092425617	UCI
6.	Dairo Roberto Gil Martín	88010618607	UCI
7.	Noel Hernández Bautista	87090432684	UCI
8.	Enma Lidia Muñoz García	88051129439	UCI
9.	Reinaldo Companioni Sosa	88102218703	UCI
10.	Anisley Martínez Romero	90021947950	UCI

No	Nombres y Apellidos	Carné de Identidad	Entidad
11.	Dayami Alvarez Cobiella	88072049377	UCI
12.	Odrey Esteban Hernández	88090823703	UCI
13.	Geny Pérez Giro	88040414677	UCI
14.	Wendy Gutiérrez Molina	88031543532	UCI

Y como constancia se emite el presente documento, en dos ejemplares, en La Habana, a los 13 días del mes de febrero de 2012.

Lic. Dianela Zayas González.  
Especialista General de la Propiedad Intelectual  
Servicios Legales de la IP  
Universidad de las Ciencias Informáticas




Validez desconocida

Digitally signed by Tayché Capote García  
Date: 2011.11.23 19:26:18 CST  
Reason: Documento oficial  
Location: Cuba



Validez desconocida

Digitally signed by Enmanuel Azhares Reyes  
Date: 2011.11.24 17:16:14 EST  
Reason: Documento Oficial  
Location: Cuba

**Fecha de emisión del acta:** 23/11/2011

**Emitida a favor de:** Portadores Energéticos

**Datos del producto**

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
Sistema de Gestión de Portadores Energéticos	1.0	1	3+PF	Pruebas funcionales	23/11/2011

**Anexo:**

A continuación se listan las No Conformidades (NC) que quedaron pendientes a resolver por el equipo de proyecto o que fueron declaradas como No Procede, sin el acuerdo de Calisoft.

NC Pendientes / No Procede	Clasificación de la NC	Respuesta del Equipo de desarrollo
<b>Pendiente</b> En muchas ocasiones, cuando se desea registrar un auditor del dominio uci no permite hacerlo. Esto no ocurre para todos los usuarios, pero sí para un gran número. (Fig. 1)	Significativa	Problema de un servicio web de la UCI, hay que revisar con Soporte para que lo arreglen pues los que no incluye son los Recién Graduados en Adiestramiento (Nivel medio superior), eso lleva un proceso de actualización que es poco a poco a nivel de soporte de universidad, la aplicación solo utiliza esto, dependemos de ellos.



Fig. 1:



**Adicionar Auditor**

Tipo de auditor:*	Dominio UCI
Usuario (UCI)*:	yallinr

No existen resultados para este envío.

Esto ocurre para varios usuarios del dominio uci.

**Daniuska  
Fresneda  
Cruzata**

Firmado digitalmente por  
 Daniuska Fresneda Cruzata  
 Nombre de  
 asociacion/IDN:  
 0.9.2342.15000000.100.1.1  
 :daniuska, cn=Daniuska  
 Fresneda Cruzata,  
 serialNumber=38900,  
 givenName=Daniuska,  
 uri=Daniuska Cruzata, c=CI  
 Fecha: 2011.11.23 15:09:25  
 -05'00'

Daniuska Fresneda Cruzata  
Responsable Calisoft

Enmanuel Azhares Reyes  
Responsable Proyecto

Tayché Capote García  
Jefe del Departamento de Pruebas

## ***Resumen***

Actualmente con la revolución de la web han crecido asombrosamente los servicios que se prestan a través de las redes. La Universidad de las Ciencias Informáticas en busca de las más novedosas soluciones se ha convertido en una ciudad informatizada donde cada vez se incrementan más los servicios que se le brindan a la comunidad universitaria a través de la red. La Dirección de Energía surge por la necesidad de hacer un uso racional de los recursos energéticos que posee la universidad, aplicando un registro y control de los diferentes portadores energéticos; de ahí la necesidad de realizar una correcta y eficiente administración, dadas las actuales necesidades de la universidad y el país. El Sistema de Portadores Energéticos, propone una solución real que mejora la vieja aplicación con que cuenta este departamento. Específicamente la actual solución persigue el objetivo de informatizar los procesos: Configuración, Gestión del consumo y Planificación. Dichos procesos no forman parte del anterior sistema en explotación, lo cual crea deficiencias para la gestión de la energía de manera automática.

**Palabras clave:** configuración, gestión, planificación, portadores, consumo, informatizar.

<b>RESUMEN</b> .....	<b>X</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
<b>1.1 Introducción</b> .....	<b>6</b>
<b>1.2 Conceptos asociados al dominio del problema</b> .....	<b>6</b>
1.2.1 Planificación.....	6
1.2.2 Planificación energética.....	6
1.2.3 Gestión del consumo.....	7
1.2.4 Configuración.....	7
1.2.5 Plan anual.....	7
1.2.6 Plan mensual.....	7
1.2.7 Plan diario.....	7
1.2.8 Estimación.....	8
1.2.9 Desagregación de los planes energéticos.....	8
1.2.10 Olvido matemático u olvido exponencial.....	8
<b>1.3 Análisis de otras soluciones existentes en Cuba y en el mundo</b> .....	<b>8</b>
1.3.1 Sistema de Control de la Energía Eléctrica (SCEE VC).....	9
1.3.2 Sistema Portadores Energéticos.....	10
1.3.3 PowerStudio.....	11
1.3.4 PowerStudio SCADA.....	12
1.3.5 PowerStudio SCADA DELUXE.....	14
1.3.6 PowerVision.....	14
1.3.7 PowerSoft Energy Manager.....	15
1.3.8 ForecastPro.....	16
<b>1.4 Metodologías de Desarrollo de Software. Metodologías Ágiles</b> .....	<b>19</b>
1.4.1 SXP como metodología ágil.....	19
<b>1.5 Tendencias, tecnologías, lenguajes y herramientas</b> .....	<b>21</b>
1.5.1 Software libre.....	21
1.5.2 Aplicaciones web.....	22
1.5.3 Modelo Vista Controlador.....	23
1.5.4 Notación para el modelado de procesos.....	23
1.5.5 Lenguaje de modelado.....	24

1.5.6 Lenguaje de programación.....	24
1.5.6 Herramienta de modelado .....	27
1.5.7 Herramienta para el prototipado .....	27
1.5.8 Marco de trabajo (Framework).....	28
1.5.9 Sistema de gestión de base de datos.....	29
1.5.10 Entorno de desarrollo.....	29
1.5.11 Plataforma para el desarrollo .....	30
<b>1.6 Conclusiones.....</b>	<b>30</b>
<b>CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.....</b>	<b>32</b>
<b>2.1 Introducción.....</b>	<b>32</b>
<b>2.2 Especificación de requisitos de software.....</b>	<b>32</b>
2.2.1 Requerimientos Funcionales .....	32
2.2.2 Requerimientos no funcionales.....	33
<b>2.3 Definición de procesos .....</b>	<b>34</b>
2.3.1 Planificación.....	35
2.3.2 Gestión del consumo.....	36
2.3.3 Configuración.....	36
<b>2.4 Personas relacionadas con el sistema.....</b>	<b>37</b>
<b>2.5 Historias de usuario .....</b>	<b>37</b>
<b>2.6 Conclusiones.....</b>	<b>39</b>
<b>CAPÍTULO 3: CONSTRUCCIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN .....</b>	<b>40</b>
<b>3.1 Introducción.....</b>	<b>40</b>
<b>3.2 Estándares de codificación .....</b>	<b>40</b>
<b>3.3 Análisis de posibles implementaciones (reutilización de código) .....</b>	<b>42</b>
<b>3.4 Vista de datos .....</b>	<b>42</b>
3.4.1 Descripción de las entidades fundamentales.....	43
<b>3.5 Arquitectura .....</b>	<b>44</b>
<b>3.6 Patrones de diseño.....</b>	<b>45</b>
<b>3.7 Vista de despliegue .....</b>	<b>47</b>
<b>3.8 Tareas de Ingeniería.....</b>	<b>48</b>
<b>3.9 Principales funcionalidades del sistema .....</b>	<b>49</b>
<b>3.10 Pruebas de software.....</b>	<b>53</b>

3.11 Resultados esperados .....	56
3.12 Conclusiones .....	57
<b>CONCLUSIONES GENERALES .....</b>	<b>58</b>
<b>RECOMENDACIONES .....</b>	<b>59</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>60</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>64</b>
<b>BIBLIOGRAFÍA CONSULTADA .....</b>	<b>67</b>
<b>ANEXOS .....</b>	<b>68</b>

## *Índice de tablas*

Tabla 1. Funcionalidades de los sistemas encontrados. ....	18
Tabla 2. HU Estimar Plan Anual.....	39
Tabla 3. Descripción de la entidad Plan .....	44
Tabla 4. Asignación de responsabilidades .....	46
Tabla 5. Tarea de ingeniería 2 .....	49
Tabla 6. Caso de prueba HU Plan anual.....	55
Tabla 7.Tarea de ingeniería 1 .....	68
Tabla 8. Tarea de ingeniería 4 .....	68
Tabla 9.Tarea de ingeniería 5 .....	69
Tabla 10. Tarea de ingeniería 6 .....	69
Tabla 11. Tarea de ingeniería 7 .....	70
Tabla 12. Tarea de ingeniería 8 .....	70
Tabla 13. Tarea de ingeniería 11 .....	70
Tabla 14. Tarea de ingeniería 12 .....	71
Tabla 15. Tarea de ingeniería 13 .....	71
Tabla 16. Tarea de ingeniería 14 .....	72
Tabla 17. Tarea de ingeniería 15 .....	72
Tabla 18.Tarea de ingeniería 16 .....	73
Tabla 19.Tarea de ingeniería 17 .....	73
Tabla 20. Tarea de ingeniería 18 .....	73
Tabla 21. Tarea de ingeniería 19 .....	74
Tabla 22. Tarea de ingeniería 20 .....	74
Tabla 23. Tarea de ingeniería 21 .....	75
Tabla 24. Tarea de ingeniería 3 .....	75
Tabla 25. Tarea de ingeniería 10 .....	76
Tabla 26. Entidad Configuración.....	76
Tabla 27. Entidad Consumidor.....	77
Tabla 28. Entidad Consumo.....	77
Tabla 29. Entidad Correos .....	78
Tabla 30. Entidad Escada_01 .....	78
Tabla 31. Calcular consumo de los consumidores mensuales. ....	79
Tabla 32. Calcular consumo de los consumidores normales.....	80
Tabla 33. Calcular consumo de los consumidores automáticos. ....	81

## *Índice de tablas*

Tabla 34. Calcular consumo de los consumidores no medrados.....	82
Tabla 35. Servicio web para mostrar los datos en la Intranet. ....	82
Tabla 36. Plan Mensual. ....	83
Tabla 37. Plan de los servicios UCI.....	85
Tabla 38. Plan Diario de los consumidores internos.....	86
Tabla 39. Enviar correo a interesados.....	87
Tabla 40. Proceso Gestión de consumo. ....	90
Tabla 41. Proceso Configuración. ....	93
Tabla 42. CP Plan Anual.....	94
Tabla 43. CP Configuración .....	95
Tabla 44. Adicionar correo .....	96
Tabla 45. Eliminar correo .....	97
Tabla 46. Modificar correo .....	98
Tabla 47. Mostrar correos .....	98
Tabla 48. Plan de los servicios.....	99
Tabla 49. Desagregación para un mes .....	100
Tabla 50. Modificar plan diario .....	101
Tabla 51. CP Plan mensual .....	102
Tabla 52. Plan de los servicios.....	103
Tabla 53. Requerimientos funcionales y no funcionales.....	109

## *Índice de figuras*

Figura 1. SCEE .....	10
Figura 2. PowerStudio.....	12
Figura 3. Power Studio SCADA.....	13
Figura 4. PowerStudio DELUXE .....	14
Figura 5. PowerVision .....	15
Figura 6. ForecastPro .....	17
Figura 7. Requisitos identificados de acuerdo a su prioridad .....	34
Figura 8. Proceso Planificación .....	36
Figura 9. Modelo de datos.....	43
Figura 10. Modelo Vista Controlador .....	45
Figura 11. Modelo de Despliegue Nodo Central.....	48
Figura 12. Módulo Energía Eléctrica. ....	50
Figura 13. Plan Anual.....	51
Figura 14. Desagregación del plan anual por servicios.....	51
Figura 15. Plan mensual por servicios .....	52
Figura 16. Plan mensual de los servicios UCI por día .....	53
Figura 17. Configuración .....	53
Figura 18. Resultados de las pruebas.....	56



## ***Introducción***

La recopilación, estructuración y análisis de datos estadísticos vinculados con la economía energética no es reciente. A partir de la reducción de los suministros de petróleo y la duplicación del precio de los crudos, adquiere un nuevo interés que se pone de manifiesto en el desarrollo de lo que ha venido en llamarse el “análisis energético”. Desde entonces, el análisis energético ha prestado su mayor atención en la evaluación de las posibilidades futuras de suministro y en la utilización de todos los tipos de energía en su conjunto. Más recientemente, el desarrollo sostenible, como nuevo concepto del desarrollo económico propone además un punto de vista social y ecológico. Hoy en día, el disponer y manejar de manera adecuada las principales variables vinculadas a la economía energética resulta decisivo para enfrentar los retos del presente y el futuro, de ahí la importancia del uso racional de los diferentes portadores energéticos que posee el país. Los portadores energéticos se clasifican en dos grupos: naturales y secundarios. Los naturales son aquellos “provistos por la naturaleza” y los secundarios los productos resultantes de las transformaciones o elaboración a partir de portadores energéticos naturales. Son portadores energéticos elaborados la electricidad, toda la amplia gama de derivados del petróleo, el carbón vegetal, el alcohol desnaturalizado y el gas manufacturado (o gas de ciudad).

Hasta la actualidad el consumo de energía ha sido valorado como un índice del progreso económico y social de los países. Durante mucho tiempo las consecuencias ambientales de los patrones de consumo de energía fueron dejadas a un lado. Ya en los albores del siglo XXI, el problema energético adquiere una importancia crucial, no solo desde el punto de vista de la satisfacción de la creciente demanda social de energía, sino también en lo que se refiere al impacto ambiental del sistema energético contemporáneo, basado fundamentalmente en el uso de los combustibles fósiles y en menor medida la energía nuclear.

La energía eléctrica ha hecho posible que las computadoras estén a la cabeza del adelanto tecnológico en la actualidad, convirtiéndose en el dispositivo más utilizado alrededor del mundo, según un artículo escrito por Christopher Nickson, el cual ubica a la computadora de escritorio en el número uno entre los dispositivos electrónicos, pues cuando se trata de popularidad, adopción, revolución y evolución; sin duda no tienen rival [2]. Con la revolución de la web cada vez crecen más los servicios que se realizan a través de las redes, los cuales resultan en muchos casos más factibles para los cibernautas.

Desde la creación de la Universidad de las Ciencias Informáticas (en adelante UCI) en el 2002, por una idea del Comandante Fidel Castro, basado en el sueño de introducir al país en el avance de las tecnologías de la información; se empezaron a materializar procesos de desarrollo de sistemas

necesarios para brindar servicios a la comunidad universitaria. Progresivamente se han ido optimizando y perfeccionando los mismos con vistas a hacer realidad el sueño del comandante de convertir a la UCI en una universidad de excelencia.

Vinculados al impacto energético y a las más modernas tecnologías para brindar servicios, surge en la UCI la Dirección de Energía, la cual tiene como objetivo el registro y control de todos los recursos energéticos asociados a la universidad. Dicho departamento de energía cuenta con un sistema en explotación que presenta diversos problemas con el registro y control del consumo eléctrico, entre ellos se encuentran:

1. Carece de funcionalidades necesarias para la planificación, control y registro del consumo eléctrico.
2. Incorrecto diseño de la base de datos.
3. El sistema no posibilita definir una lista de correos electrónicos para el envío de los modelos oficiales de los planes a los responsables de los módulos.

Después de un estudio realizado en la propia Dirección de Energía el equipo de desarrollo del proyecto de Portadores Energéticos encontró que además de la incorporación de la gestión del consumo eléctrico y la configuración al módulo Energía Eléctrica, era necesaria la gestión de la planificación correspondiente al plan energético; dado que este proceso se realiza manualmente y que la universidad todos los años necesita demandar un plan al Ministerio de Economía y Planificación.

El especialista que realiza este trabajo actualmente, debe estimar dicho plan basándose en consumos históricos reales, para ello debe tener en cuenta además un conjunto de indicadores como: entrada en funcionamiento de inversiones, aumento de la jornada de trabajo de algún área, aumento en las capacidades instaladas por la incorporación de nuevos medios, entre otros que influyen en el comportamiento de la energía eléctrica en la universidad. Para realizar este proceso tiene que manipular mucha documentación y realizar numerosas tablas y diagramas. Además, después de realizar la demanda anual, con el plan que se le aprobó a la universidad, debe realizar la distribución del plan de forma mensual y diaria entre todos los consumidores. Todo este proceso se hace engorroso para una persona debido a que se basa en operaciones matemáticas que pueden ser en ocasiones inexactas.

Por todas las insuficiencias planteadas se formula el siguiente **problema a resolver**: ¿Cómo lograr que los procesos de Planificación, Gestión del consumo eléctrico y Configuración del Sistema de Portadores Energéticos controlen y planifiquen de manera informatizada el consumo energético en la Universidad de las Ciencias Informáticas?

El esbozo de todo el sistema está centrado en el **objeto de estudio**: Procesos de Planificación, Configuración y Gestión del consumo eléctrico del Sistema de Portadores Energéticos.

**Objetivo general:** Informatizar los procesos de Planificación, Gestión del consumo eléctrico y Configuración que permitan el control y registro de la energía para el Sistema de Portadores Energéticos.

Todo está basado en la siguiente **idea a defender**: El desarrollo de los procesos de Planificación, Gestión del consumo eléctrico y Configuración del módulo Energía Eléctrica en la UCI, incrementará la eficiencia en la estimación y desagregación de los planes energéticos, así como en la configuración y gestión del consumo eléctrico.

Para solucionar la problemática planteada se proponen las siguientes **tareas de la investigación científica**:

#### **Tareas comunes:**

- Identificar y analizar las tecnologías, herramientas, lenguajes y metodología de desarrollo de *software* a utilizar para el análisis, diseño y construcción del producto.
- Modelar la base de datos del sistema y describir cada uno de sus componentes.
- Realizar un análisis del estado del arte sobre soluciones similares al sistema propuesto.
- Identificar y describir las capacidades o condiciones que el sistema debe cumplir.

#### **Tareas individuales:**

##### Enma Lidia Muñoz García

- Identificar los procesos relacionados con la gestión de la planificación, el consumo y la configuración en la dirección de Energía de la Universidad de las Ciencias Informáticas.
- Identificar y describir los procesos del negocio.
- Identificar y describir las historias de usuario.
- Realizar las pruebas de validación al sistema.

##### Noél Hernández Bautista

- Describir la arquitectura del sistema.
- Implementar cada una de las historias de usuario definidas y descritas por el analista, utilizando para ello un sistema de clases que garantice el correcto funcionamiento del sistema.

#### **Posibles resultados:**

- Proceso Configuración que permita funcionalidades administrativas tales como activación y desactivación de módulos, así como la introducción de los correos electrónicos para el envío de los reportes a los responsables de los servicios.
- Proceso Gestión del consumo que garantice el cálculo del consumo a partir de las lecturas eléctricas recogidas, teniendo en cuenta cada tipo de consumidor. Además, el uso de un servicio web que propicie la conexión con otros sitios; de manera que los mismos muestren el plan y el consumo eléctrico general de la UCI, así como el porcentaje que representa dicho consumo del plan teniendo en cuenta una fecha.
- Proceso Planificación que permita la estimación, lo más acertada posible, de los planes anuales, mensuales y diarios a partir de los consumos históricos guardados, además se debe contemplar la correspondiente distribución de cada tipo de plan entre los consumidores de acuerdo con sus características.

Para garantizar un desarrollo óptimo de la investigación es necesario el empleo de métodos científicos y técnicas:

### Métodos empíricos:

Entrevista: se realizó a directivos de la Dirección de Energía para obtener información relacionada con el módulo en cuestión, lo que contribuyó al perfeccionamiento de la propuesta realizada.

### Métodos teóricos:

- Analítico - sintético: se utilizó en el estudio de la literatura especializada relacionada con el tema y la exploración de resultados de investigaciones afines, lo que permitió adoptar posiciones teóricas relacionadas con el objeto de investigación.
- Histórico - lógico: se empleó para el estudio y profundización de la evolución de las aplicaciones y tendencias lo que facilitó el desglose de la propuesta de solución sobre el problema planteado.
- Modelación: se utilizó durante el desarrollo de la actual propuesta de solución pues se hace necesario explicarle al cliente mediante modelos (interfaz de usuario), cómo se tiene pensado que quede el sistema para saber si cumple con sus necesidades.
- Enfoque del sistema: se aplicó para el estudio y profundización constante del tema de investigación, lo que propició la introducción de proyecciones concebidas en las diferentes tareas integradoras realizadas en función del objetivo propuesto.

El trabajo de diploma está estructurado de la siguiente forma:

**Capítulo 1. Fundamentación teórica:** Se exponen los fundamentos generales que sirven de soporte teórico en la solución del problema, lo cual incluye un estudio valorativo de los principales conceptos envueltos en el negocio y un estado del arte de los sistemas similares existentes a nivel internacional, nacional y de la Universidad, de las técnicas de programación, herramientas tales como: sistemas gestores de bases de datos, marcos de trabajo, entornos de desarrollo y plataformas, tecnologías referentes al análisis, así como de las metodologías y *software* en las que se apoya para la propuesta de solución.

**Capítulo 2. Descripción y análisis de la solución propuesta:** Se realiza una descripción del flujo de los procesos involucrados mostrando la modelación de los mismos incluyendo las actividades que serán objeto de informatización. Se plantean los requisitos funcionales y no funcionales que se originan a consecuencia de la situación problemática, dando paso a la descripción detallada de las historias de usuario identificadas; lo cual permitirá hacer un desglose de la propuesta de solución.

**Capítulo 3. Construcción y validación de la propuesta de solución:** Se orienta a la implementación del sistema. Se establecen los estándares de codificación a utilizar y se realizan pruebas para la validación de la solución. Además se muestran las principales funcionalidades del producto, una vez terminado, dando una valoración de los beneficios derivados de su implementación.

# Capítulo 1: *Fundamentación teórica*

## **Capítulo 1: Fundamentación teórica**

### **1.1 Introducción**

Para el desarrollo de sistemas informáticos tanto web como de escritorio, se emplean en todo el mundo diferentes metodologías de desarrollo y técnicas de programación. Antes de crear una aplicación es necesario conocer bien los distintos conceptos envueltos en el negocio, cuales son las ventajas de las metodologías y herramientas a utilizar, así como, las aplicaciones homólogas que existen en Cuba y el mundo.

### **1.2 Conceptos asociados al dominio del problema**

#### **1.2.1 Planificación**

Acción y efecto de planificar. Plan general, metódicamente organizado y frecuentemente de gran amplitud, para obtener un objetivo determinado, tal como el desarrollo armónico de una ciudad, el desarrollo económico, la investigación científica y el funcionamiento de una industria [3].

#### **1.2.2 Planificación energética**

- Según Pablo Álvarez Watkins y Juan José Sánchez Inarejos la planificación energética es estudiar metódicamente de dónde se van a sacar los recursos energéticos que utilizaremos en el futuro. “Sabemos cuánta energía se ha consumido en el mundo entre 1950 y 2000, quién la ha consumido y de qué clases<sup>1</sup> era y en qué se ha invertido. También sabemos que se ha llevado a un determinado número de personas (todos los habitantes del planeta en ese tiempo) desde unos puntos de IDH (Índice de Desarrollo Humano) hasta otros. Tenemos esos datos anotados de 5 en 5 años, sabemos tanto los datos iniciales como los intermedios y los finales. ¿Este conocimiento nos puede ayudar a saber cuánta energía, y de qué tipo, necesitaremos para llevar a una determinada cantidad de personas desde el IDH de hoy (2000) hasta otro IDH mayor en 2050?” [4].
- (...) una planificación energética que sirva de plataforma a la doble estrategia que conduce al desarrollo de un país: proceso de industrialización+ fuerte inversión en ciencia y técnica. (...) la planificación energética es una función clave del Estado nacional. Sin planificación y estrategias nacionales (...), el colapso energético y la profundización del subdesarrollo insostenible resultan inevitables [4].

---

<sup>1</sup> Clases de energía: cinética, eólica, geotérmica, hidráulica, mareomotriz, nuclear, química, solar, eléctrica.

# Capítulo 1: *Fundamentación teórica*

La planificación energética en el país tiene características especiales con relación al propio desarrollo económico y social del mismo. Por ser Cuba un país subdesarrollado y además bloqueado hace más de 50 años, ha adoptado experiencia a través de los años y ha desarrollado técnicas para evolucionar. Es por esa razón que desde tiempos inmemoriales se ha racionalizado y desarrollado planes en los diferentes sectores económicos.

## **1.2.3 Gestión del consumo**

Del latín *gestiō*, el concepto de gestión hace referencia a la acción y al efecto de gestionar o de administrar. [5] Gestionar es realizar diligencias conducentes al logro de un negocio o de un deseo cualquiera.

El módulo Energía Eléctrica del Sistema de Portadores Energéticos centra su trabajo en la gestión del consumo eléctrico de toda la universidad y ello permite llevar a cabo la administración del consumo eléctrico de manera eficiente.

## **1.2.4 Configuración**

En informática la configuración es un conjunto de datos que determina el valor de algunas variables de un programa o sistema operativo, puede ser predeterminada o personalizada por el usuario. Especialmente en el módulo Energía Eléctrica se informatizará un espacio donde el usuario, con permisos especiales, pueda personalizar el sistema en determinadas situaciones. Por ejemplo, podría activar o desactivar algún módulo para realizar ajustes técnicos en el mismo.

## **1.2.5 Plan anual**

Plan energético que se estima para el próximo año. Los planes, no importa su tipo, forman parte de las inversiones de empresas y corporaciones, dado que los mismos rigen el control a mantenerse para el venidero año. En Cuba incluso se lleva a nivel mensual y diario, para lograr una mayor racionalización de los recursos. En este caso, mediante el Sistema de Portadores Energéticos se propone un plan anual estimado basándose en consumos históricos reales de años anteriores.

## **1.2.6 Plan mensual**

Plan energético que se estima para el próximo mes, con este tipo de plan se pretende lograr mayor rigor en el cumplimiento del plan anual previo. En este caso, mediante el Sistema de Portadores Energéticos se propone un plan mensual estimado basándose en consumos históricos reales de meses anteriores; para ello se evalúa de manera individual cada mes.

## **1.2.7 Plan diario**

Plan energético que se estima para cada día del mes. En este caso, mediante el Sistema de Portadores Energéticos se propone un plan diario estimado basándose en consumos históricos reales,

# Capítulo 1: *Fundamentación teórica*

para ello se evalúa de manera individual cada día y su influencia dentro de la semana. Este tipo de plan se acerca más a la realidad y tiene como objetivo asegurar el plan mensual y por ende el anual.

## **1.2.8 Estimación**

Evaluación de los consumos históricos de años anteriores para el análisis y cálculo numérico de futuros consumos.

## **1.2.9 Desagregación de los planes energéticos**

Correspondiente distribución que se realiza de acuerdo con el plan que se estimó. En este caso: para la desagregación del plan anual, se distribuye por meses y por servicios; para la desagregación del plan mensual, por días y por servicios; por último para la desagregación del plan diario, por consumidores internos.

## **1.2.10 Olvido matemático u olvido exponencial**

Para realizar estimaciones con frecuencia, es posible solo asumir que las variaciones de los parámetros ocurren con suficiente lentitud. En tal caso, una técnica heurística conocida en la literatura como el "olvido exponencial", puede aplicarse sin introducir complicaciones adicionales. La idea esencial de esta técnica es también muy simple y puede explicarse en la forma siguiente:

Sea  $t$  el tiempo actual mientras  $\tau$  denota el instante pasado en que los datos  $y(\tau)$  y  $z(\tau)$  fueron obtenidos. El estimador de los parámetros del modelo en el tiempo  $t$  se calcula a partir de un conjunto de ecuaciones del tipo:

$$y(\tau) = P^T z(\tau) + e(\tau) \text{ para } \tau = 1, 2, 3, \dots, t.$$

Si los parámetros han variado en el tiempo, resulta evidente que las ecuaciones más "viejas" en este conjunto resultan menos confiables que las más recientes. Es posible tener este hecho en cuenta considerando que la varianza del término estocástico  $e(\tau)$  no es constante, sino que varía de forma tal que a las ecuaciones más viejas corresponde una varianza mayor que la de las ecuaciones últimas, es decir, para  $\tau$  próximo a  $t$ .

La técnica del olvido exponencial se utiliza también en el caso de parámetros constantes para eliminar rápidamente la influencia de las estimaciones iniciales que generalmente son malas como consecuencia de la falta de información a priori de los valores reales de los parámetros [6].

La técnica del olvido matemático se utiliza en la estimación de los planes del módulo Energía Eléctrica para manejar los consumos históricos y realizar una estimación reduciendo el rango de error más efectivamente.

## **1.3 Análisis de otras soluciones existentes en Cuba y en el mundo.**



# *Capítulo 1: Fundamentación teórica*

Hoy en día empresas e industrias buscan, además de controlar energéticamente sus instalaciones, tener un control centralizado de toda la instalación para su gestión y mantenimiento. A continuación se listan una serie de soluciones para la gestión energética:

## **1.3.1 Sistema de Control de la Energía Eléctrica (SCEE VC)**

En Cuba existe una aplicación desarrollada en la Universidad de Villa Clara para el Ministerio de Ciencia, Tecnología y Medio Ambiente de Cuba (CITMA), llamado Sistema de Control de la Energía Eléctrica (SCEE). Este es un sistema integral desarrollado para la gestión de la energía eléctrica, su planeación y control en el entorno organizacional.

Está diseñado en forma de módulos independientes que pueden ser implantados de modo secuencial y que mantienen interrelaciones útiles que van elevando su grado de efectividad. Dispone de formularios sencillos para la entrada de datos y múltiples salidas de datos e información para el análisis y la correspondiente toma de decisión.

SCEE es un sistema sencillo, manejable y austero que puede ser instalado en una computadora con los mínimos requerimientos para Windows. Programado en Visual Basic, emplea como interfaces el Office, especialmente el Microsoft Office Excel 2003. Posee un tamaño aproximado de 2000 KB y puede ser instalado en una computadora con los mínimos requisitos [7].

# Capítulo 1: Fundamentación teórica

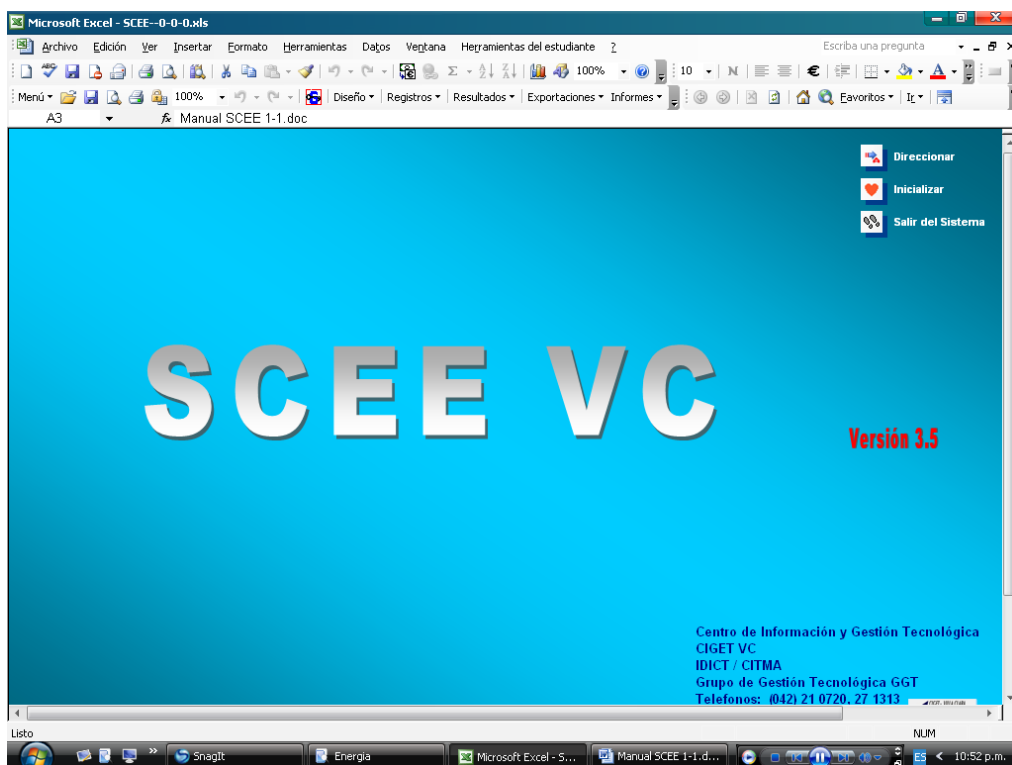


Figura 1. SCEE

Este sistema presenta una funcionalidad llamada Registro del Plan en la cual se introduce el plan del mes deseado y los días físicos que tendría dicho mes y el sistema automáticamente calcula el promedio diario de consumo para ese plan mensual. Además, le permite al usuario calcular el promedio real del consumo de los días de la semana de manera diferenciada.

### 1.3.2 Sistema Portadores Energéticos

La Universidad de las Ciencias Informáticas posee un sistema desarrollado para la Dirección de Energía. Este sistema brinda la posibilidad de controlar el consumo de energía, muestra además cuadros directivos de la UCI y a la comunidad universitaria en general los resultados de consumo.

#### Prestaciones

##### 1. Autenticación.

Es un sistema modular con acceso por niveles de autenticación, toda acción a realizar dentro del sistema requiere la previa autenticación al mismo.

##### 2. Registrar usuarios.

Permite registrar nuevos usuarios al sistema a través del módulo de Administración. Se muestran una serie de datos tales como: foto, nombre, apellidos y usuario, además brinda la posibilidad de asignarle en el proceso los roles que tendrá el usuario.

# Capítulo 1: Fundamentación teórica

## 3. Gestionar usuarios.

Muestra los usuarios identificados y registrados con los siguientes datos: nombre, apellidos y usuario del dominio UCI, además una opción que permite la activación o desactivación del mismo en la base de datos, por defecto todos los usuarios aparecerán en estado activo, permitiéndole realizar todas las acciones que le están permitidas según el rol o los roles que tenga, si se desactivara el usuario entonces no tendría acceso a ninguna acción dentro del sistema.

## 4. Gestionar consumidores.

Muestra el listado de todos los consumidores previamente registrados, así como la opción de seleccionar si su registro de consumo será automático o no.

## 5. Registrar plan de consumo diario.

Muestra un formulario en el cual se ingresan los siguientes datos: fecha de registro, consumidores y cantidad de kw/h a consumir, en caso de error a la hora de introducir la lectura, permite que este dato sea modificado aunque ya este insertado en la base de datos. El plan diario puede ser editable hasta la hora de introducir el consumo, para dar la posibilidad de ajustes al plan.

### 1.3.3 PowerStudio

Mediante el PowerStudio se puede leer en tiempo real los valores instantáneos de todos los equipos que posee este paquete de la compañía CIRCUTOR y generar un histórico de datos que se guarda en una PC<sup>2</sup> para su posterior estudio. Además mediante PowerVision se pueden descargar los datos almacenados en los equipos que disponen de memoria interna.

Este *software* permite al usuario tener un control absoluto de la instalación, conociendo en tiempo real y de primera mano, el estado de sus líneas de potencia e incluso de consumo general tanto en baja como en media tensión. Dicho control es importante ya que puede realizarse un excelente mantenimiento preventivo, desde el cual se puede controlar una gran cantidad de parámetros eléctricos [8].

---

<sup>2</sup> PC: término informático utilizado en el argot popular para llamar a una computadora.

# Capítulo 1: Fundamentación teórica

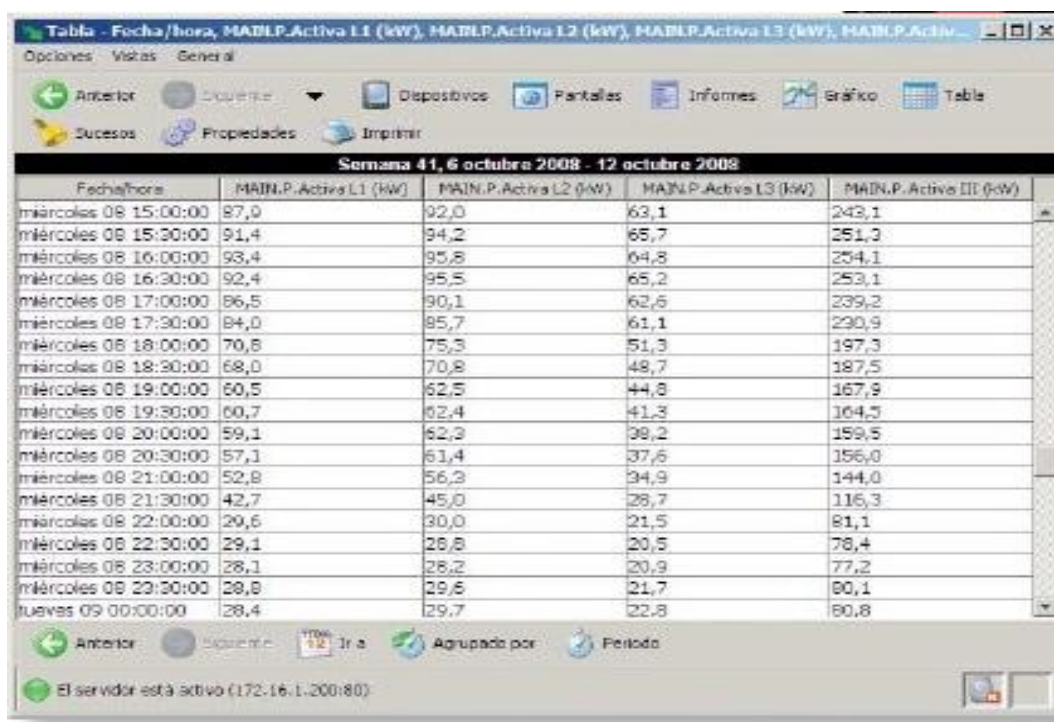


Figura 2. PowerStudio

## Aplicación:

- Parametrización remota de los equipos.
- Visualización de parámetros en tiempo real.
- Registro de históricos.
- Visualización de históricos mediante tablas y gráficos.
- Impresión de gráficos<sup>3</sup> y tablas.
- Gran versatilidad y muy fácil uso.
- Acceso a través de internet con contraseña y posibilidad de creación de perfiles de acceso.

### 1.3.4 PowerStudio SCADA

Debido al gran volumen de información que aporta cada central de medida, es necesario, disponer de un sistema centralizado de recogida de datos. El PowerStudio SCADA está diseñado para actuar como centralizador y gestor de información. Además, está pensado para que cualquier usuario pueda crear sus propias pantallas personalizadas.

<sup>3</sup> Entiéndase por gráfica como la representación de los datos mediante gráficos de evolución temporal que pueden ser de barras, de pastel, etc.

# Capítulo 1: Fundamentación teórica

La finalidad de este *software* es el procesamiento de los datos y elaboración de informes, con el objetivo de adoptar medidas preventivas o correctivas en la instalación. En definitiva, este *software* permite la integración de los equipos CIRCUTOR para su gestión [8].

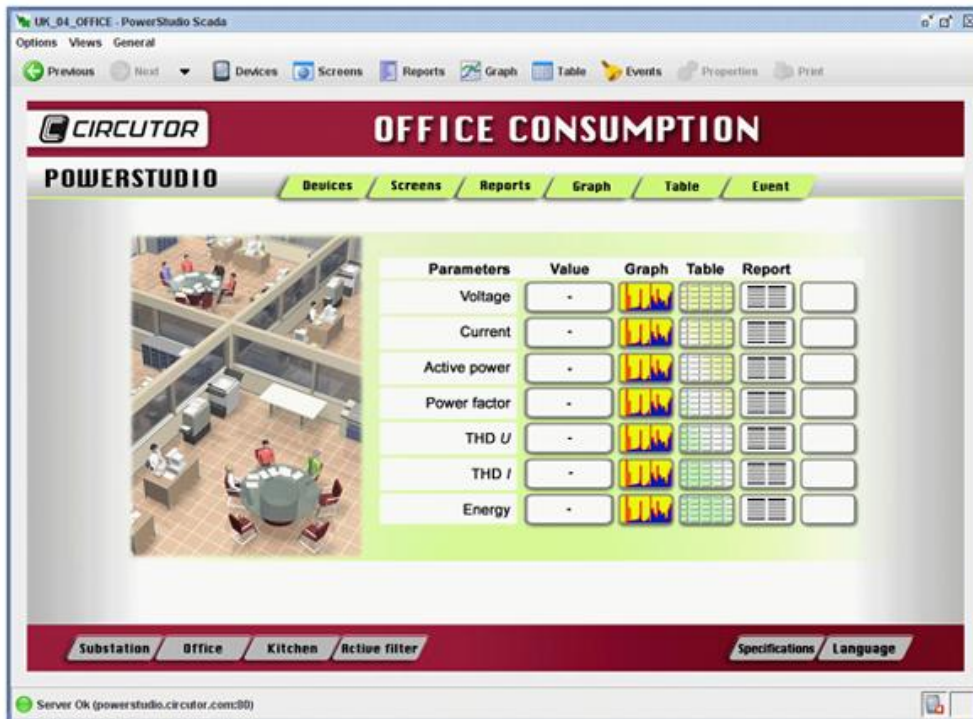


Figura 3. Power Studio SCADA

## Prestaciones

- Además de brindar todas las del PowerStudio se agregan:
- Módulo de alarmas.
- *Software* multipuesto (Servidor Web integrado)<sup>4</sup>.
- Construcción de pantallas personalizadas, es decir, permite fijar etiquetas de visualización de parámetros o estados que indiquen el estado de un punto concreto de la instalación o el estado de la línea.
- Generador de informes y simulador de recibos.

<sup>4</sup> El *software* usa un servidor web interno propio y por ello cualquier usuario puede acceder conectado desde la red corporativa de la empresa (LAN) o desde cualquier punto de internet. Así se pueden visualizar los datos históricos o en tiempo real, que se actualizaran permanentemente. Además, el número de usuarios web que pueden conectarse al servidor es ilimitado.

# Capítulo 1: Fundamentación teórica

PowerStudio SCADA, además de comportarse como un excelente gestor energético, tiene implementadas funciones de intercambio dinámico de datos, facilitando así la integración rápida y cómoda con otras aplicaciones de mercado.

## 1.3.5 PowerStudio SCADA DELUXE

Además de todas las prestaciones ofrecidas por el *software* PowerStudio SCADA, la versión DELUXE intenta dar cobertura a dispositivos con conexiones estándar, además incorpora todos los *drivers* (controladores) de la factoría CIRCUTOR [8].



Figura 4. PowerStudio DELUXE

## 1.3.6 PowerVision

*Software* para la telemedida y gestión de la información registrada por los analizadores de calidad de suministro eléctrico. Permite tanto a los usuarios expertos como a aquellas personas que lo utilizan por primera vez, sacar el máximo rendimiento a la información de forma rápida y sencilla.

### Gestión de información

- Realización de gráficas y listados de todas las variables.
- Exportación de la información a ficheros ".txt"<sup>5</sup> para luego poderla tratar desde cualquier hoja de cálculo.

<sup>5</sup> .txt: extensión para archivos de texto plano.

# Capítulo 1: Fundamentación teórica

- Presentación de gráficos con los resultados.
- Impresión de los resultados del análisis.
- Exportación de gráficas y tablas a “.bmp”<sup>6</sup>.
- Estudio gráfico mediante tablas de los datos obtenidos de un equipo.
- Posibilidad de realizar gráficas de evolución temporal.
- Gráficas configurables (color, tipo de gráfica).

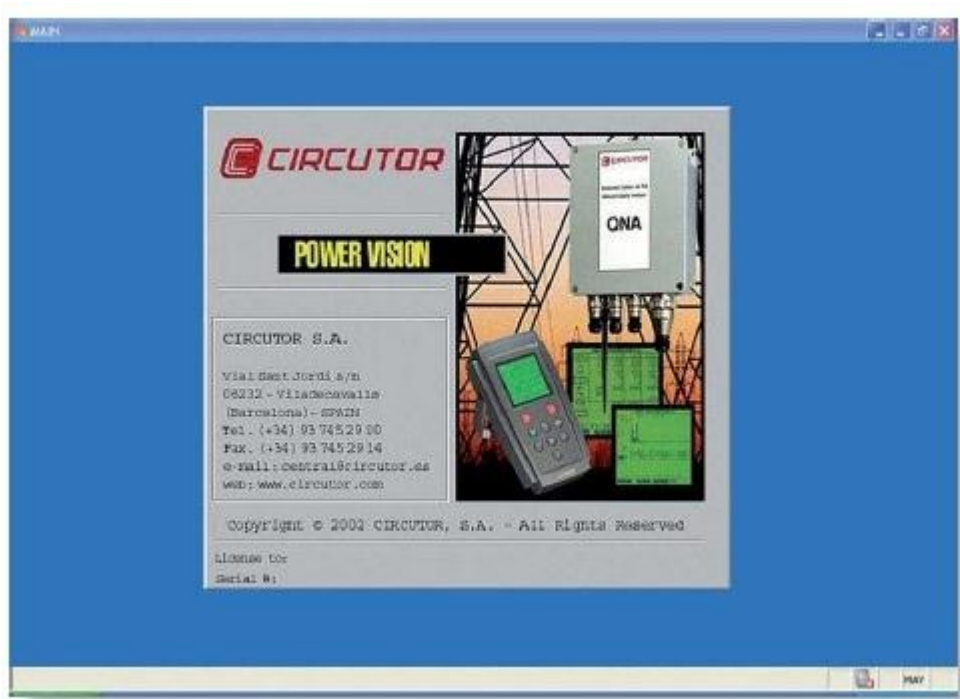


Figura 5. PowerVision

## Características y ventajas

- Unificación y centralización del control de múltiples SAI (Sistemas de Alimentación Ininterrumpida) y otros equipos de protección energética.
- Mantenimiento de un rendimiento energético excepcional a través del control a escala empresarial y la gestión de alarmas en tiempo real.
- Racionalización de las tareas de gestión de dispositivos de todos los SAI de una red [9].

### **1.3.7 PowerSoft Energy Manager**

Este *software* es parte del proyecto Innovation for Energy Management (Innovación para la Gestión Energética) que brinda además todo el paquete de metrocontadores y analizadores correspondientes.

<sup>6</sup> .bmp: archivo de mapa de bits, es decir, un archivo de imagen de gráficos.

# Capítulo 1: *Fundamentación teórica*

PowerSoft es una plataforma de análisis adecuado para Windows 98, 98SE, Windows XP y Vista. Consta de un núcleo principal y de una serie de módulos opcionales conectados, que permiten administrar correcta y eficientemente un sistema de distribución eléctrica, sobre todo teniendo en cuenta el punto de vista de la reducción de costes. Este objetivo se logra mediante el control de los consumos, el control de los picos de potencia demandada y la adopción de un análisis preciso y herramientas de procesamiento de datos.

## Visualización en tiempo real

Cada instrumento tiene una sección de 4 páginas donde sus datos en tiempo real pueden ser examinados de diferentes maneras: como indicadores analógicos; analíticamente en una tabla completa, incluyendo todas las variables y los medidores de servicios públicos; por medio de un diagrama de Fresnel<sup>7</sup>. Es posible comprobar el estado de las entradas digitales y comprobar o cambiar la salida digital para propósitos de prueba o para actuar de forma remota en un dispositivo externo. Una tabla que muestra al mismo tiempo todos los datos de hasta 5 instrumentos por página está disponible, lo que permite al usuario llevar a cabo una comparación de datos simple e inmediata.

## Estimación de costos

De acuerdo con los parámetros de utilidad de contrato, el *software* permite estimar los costes correspondientes a un período, a consecuencia de la energía, el agua y el consumo de gas. Esto es útil para realizar la asignación de costos entre las líneas de control, para mostrar la tendencia del consumo diario o para identificar las razones de cualquier sanción.

## Análisis estadístico

PowerSoft lleva a cabo análisis estadísticos sobre las tendencias de alimentación y consumo de energía mediante la extrapolación de la demanda de cada día de la semana, las tendencias de consumo basados en una semana y la estimación de la potencia instalada ideal para cada tarifa, calculada con un nivel de confianza seleccionable [10].

### **1.3.8 ForecastPro**

Internacionalmente muchas empresas realizan el proceso de planificación dirigido a distintos tipos de negocios, generando beneficios para sus usuarios. Específicamente este tipo de planificación basado en datos históricos se usa efectivamente para hacer previsiones de recursos. Varias empresas han implementado sus sistemas, ya sea de estimación de costos o proyectos. ForecastPro, compañía que posee un *software*, del mismo nombre, líder en planeación de demanda y pronósticos; presenta entre sus características las siguientes:

---

<sup>7</sup> Diagrama para medir la intensidad y el voltaje de un circuito.



# Capítulo 1: Fundamentación teórica

- ForecastPro permite definir libremente los parámetros (tales como ventas, el número de contactos con clientes, el tiempo medio de actuación, o el volumen de producción) que influyen en las previsiones futuras. No hay restricciones en el tipo de datos implicados. De hecho, incluso información poco usual como las estadísticas meteorológicas pueden ser introducidas y usadas.
- El asistente de previsión de ForecastPro combina todos los pasos necesarios para calcular las futuras cargas de trabajo, presentando, paso a paso, los campos para introducir los datos necesarios. El sistema reconoce automáticamente las tendencias estacionales e integra toda la información necesaria, como las curvas de distribución diarias y semanales, para el periodo específico [11].

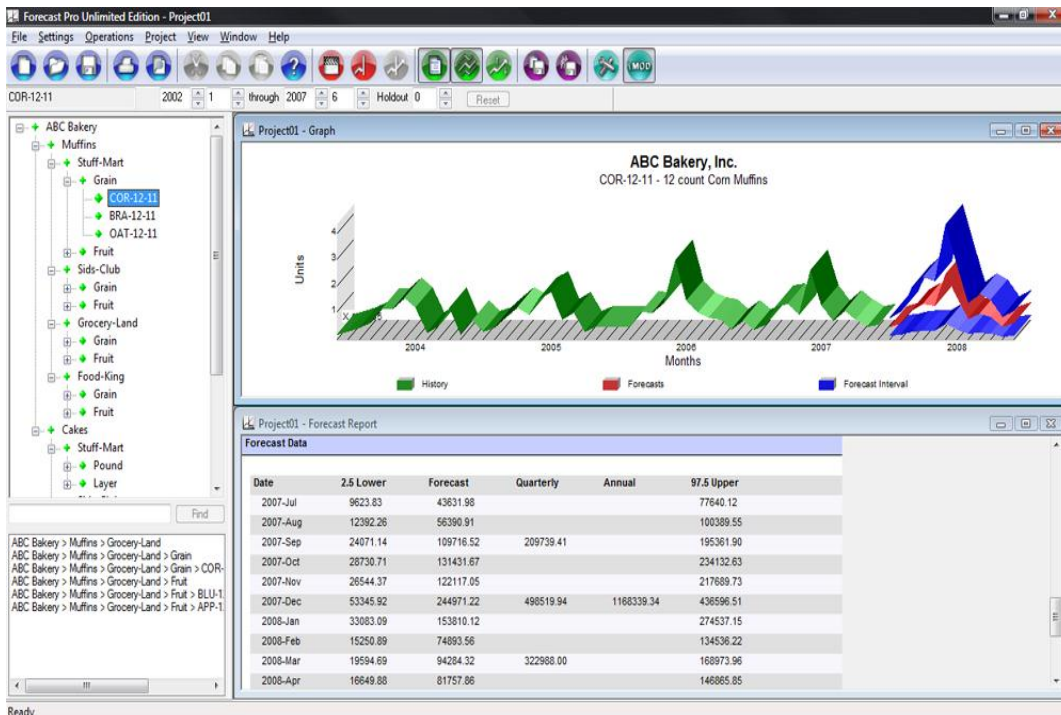


Figura 6. ForecastPro

En general la gestión energética en el mundo alcanza auge en la actualidad debido a las crisis económicas que existen y al deterioro del medio ambiente, es por ello que se plantean nuevas alternativas para el ahorro de recursos y la protección ambiental. De manera general en los sistemas estudiados se destacan las siguientes funcionalidades:

# Capítulo 1: Fundamentación teórica

Funcionalidades/Sistemas	SCEE VC	Power Vision	Power Studio	PowerSoft Energy Manager	ForecastPro	Sistema Portadores Energéticos
<u>Distribución de consumo</u>	x					
<u>Registro de consumos históricos</u>			x			
<u>Exportación de resultados</u>		x				
<u>Estimación</u>				x	x	

Tabla 1. Funcionalidades de los sistemas encontrados.

En cuanto a los sistemas encontrados internacionalmente vale destacar que aunque son muy actualizados y avanzados no es recomendable su adquisición dado que:

- En su mayoría son un paquete que integra los metrocontadores y analizadores, además del *software* para la telemedida, condición que no resulta factible dado que la mayoría de los consumidores universitarios se encuentran medrados y las lecturas son recogidas a través de un SCADA (Control de Supervisión y Adquisición de Datos), o sea que solo es necesario el *software* para la gestión energética.
- Son tecnologías propietarias que resultan muy costosas.

El Sistema de Control de la Energía Eléctrica del CITMA en Villa Clara no se recomienda su uso pues carece de las principales funcionalidades propuestas en el actual sistema, además fue desarrollado usando tecnologías propietarias y está diseñado para ser solo usado sobre Windows.

El Sistema Portadores Energéticos de la UCI, a pesar de que se utiliza en la actualidad para el control energético y que sirve como punto de partida para el desarrollo del sistema propuesto, carece de funcionalidades que permitan llevar un buen control de los recursos energéticos, razón por la que debe realizarse un nuevo diseño de la base de datos, además no posee documentación asociada que respalde su desarrollo.

Por ello es por lo que surge el interés de crear un sistema que se ajuste a las características organizacionales de la universidad y a los recursos instalados que ya posee. La UCI fue creada con la idea de mantener al país a la par del adelanto científico-técnico y lograr soberanía tecnológica utilizando a la juventud como cantera.

# Capítulo 1: Fundamentación teórica

## 1.4 Metodologías de Desarrollo de Software. Metodologías Ágiles

Las metodologías imponen un proceso disciplinado sobre el desarrollo de *software* con el fin de hacerlo más predecible y eficiente. Las metodologías ingenieriles han estado presentes durante mucho tiempo. No se han distinguido precisamente por ser muy exitosas. Aún menos por su popularidad. La crítica más frecuente a estas metodologías es que son burocráticas. Hay tanto que hacer para seguir la metodología que el ritmo entero del desarrollo se retarda.

Hoy en día existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Un ejemplo de ellas son las propuestas tradicionales centradas específicamente en el control del proceso. Estas han demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo aquellos proyectos de gran tamaño (respecto a tiempo y recursos).

Como respuesta a los problemas aplicando metodologías tradicionales surgieron otras metodologías que tratan de adaptarse a la realidad del desarrollo de *software*.

El encanto de estas metodologías ágiles es su reacción ante la burocracia de las metodologías monumentales. Los métodos ágiles cambian significativamente algunos de los énfasis de los métodos tradicionales. La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. De muchas maneras son más bien orientados al código, siguiendo un camino que dice que la parte importante de la documentación es el código fuente. A continuación se muestran algunas de las metodologías ágiles definidas hasta el momento [12]:

- Extreme Programming- XP
- Scrum
- Crystal Methodologies
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development (ASD)
- Feature -Driven Development (FDD)
- Lean Development (LD)

### 1.4.1 SXP como metodología ágil

En el año 2007 se hizo en la Facultad 7 de la Universidad de las Ciencias Informáticas un estudio basado en las principales características de los proyectos productivos, evaluando todos los problemas a los que estaban sometidos. Razón por la que se le otorgó como tema de tesis a la estudiante Malay Rodríguez Villar la “Introducción de procedimientos ágiles en la producción de *software* en la Facultad 7 de la Universidad de las Ciencias Informáticas”. Malay en su trabajo hizo una amplia y profunda

# Capítulo 1: *Fundamentación teórica*

investigación de las metodologías ágiles, exponiendo su definición y rasgos distintivos, lo que la llevó a analizar un importante documento que abarca todo sobre estos procedimientos: el Manifiesto Ágil, documento que expone los principios, características y valores de este tipo de metodologías. Para darle solución al problema planteado se propuso explicar, guiar y orientar cómo sería introducir procedimientos ágiles en el proceso productivo de la Facultad 7. Las metodologías ágiles usadas en la propuesta son las metodologías: SCRUM, para la planificación de los proyectos que usarán métodos ágiles como proceso de desarrollo y para llevar a cabo el proceso de desarrollo del proyecto se tomará en cuenta las mejores prácticas de XP, procurando que el proceso sea efectivo y eficiente [13].

A continuación, aparece el guion de la propuesta [13]:

## **Planificación ↔ Definición**

- Entrevista con el cliente (concepción inicial).
- Juego de la planificación.
- Captura de requisitos:
  - Creación de la Lista de Reserva del Producto (LRP).
  - Priorización de la LRP.
  - Definir las historias de usuario.
  - Asignar las tareas de las historias de usuario.
- Valoración del esfuerzo.
- Diseño con las metáforas.
- Creación de las tarjetas CRC.
- Refactorización (eliminar complejidad, diseñar lo más simple que se pueda).
- Reunión de la revisión del diseño.

## **Desarrollo**

- Junta de planificación.
  - Definir las historias de usuario a implementar.
  - Tareas para lograr dicha implementación.
- Fase de programación:
  - Programación en pareja.
  - Propiedad colectiva.
  - Integración continua.
  - Estándares de programación.
  - Cliente en sitio.

# Capítulo 1: Fundamentación teórica

- Junta de seguimiento.
- Junta de revisión.
- Pruebas:
  - Unitarias.
  - Autorizadas.

## Entrega

- Entrega de la documentación.
- Entrenamiento.
- *Marketing*.

## Mantenimiento

- Soporte.

En el año 2007-2008 la propuesta hecha por la ingeniera Malay Rodríguez Villar fue puesta en práctica en uno de los proyectos productivos de la Facultad 10 de la UCI; el proyecto UNICORNIOS (Servicios Especializados para la Migración a Software Libre). Gladys Marsi Peñalver Romero realizó un estudio de los resultados de la puesta en marcha de la propuesta, lo cual trajo como consecuencia la aplicación de la metodología SXP en la infraestructura productiva de la Universidad de las Ciencias Informáticas.

La introducción de las metodologías ágiles, especialmente de SXP, por parte del Centro de Informatización Universitaria propone una solución más viable a cada uno de los proyectos de la facultad 1. Especialmente en el caso del Sistema de Portadores Energéticos constituye la opción ideal dado que su enfoque dirigido a la interrelación con los clientes y al trabajo en equipo optimiza el desarrollo de dicha aplicación.

## 1.5 Tendencias, tecnologías, lenguajes y herramientas

### 1.5.1 Software libre

El *software* libre es un movimiento tecnológico que ha revolucionado la sociedad. Presenta características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos y nuevas normas legales. Es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en “libre” como en libertad de expresión [14].

Para que las libertades de hacer modificaciones y publicar versiones mejoradas tengan sentido, se debe tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el *software* libre. Esto da la medida de la viabilidad económica

# Capítulo 1: *Fundamentación teórica*

de este sistema libre, que por ser libre no es necesariamente gratuito, sino que da la posibilidad de comercializarlo, regalarlo, prestarlo con total libertad y protegerlo legalmente. Evidentemente, es la alternativa para los países subdesarrollados. El costo al usar *software* libre es menor, pues el precio total de propiedad del sistema operativo libre Linux, es menos de la mitad que el de Windows. Gran parte del ahorro proviene de no tener que pagar licencia y de sus menores costos de administración. Da pie a la innovación tecnológica, donde el desarrollo en comunidad de este sistema y el conocimiento del código fuente propician que a cada instante, un desarrollador necesite nuevas actualizaciones y las realice él mismo, proponiendo nuevas funcionalidades al programa.

Dentro de estas ventajas también se encuentra la independencia del proveedor gracias a la disponibilidad del código fuente. Se garantiza la privacidad y la seguridad, pues el *software* libre, por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el mundo del *software* libre, cualquier programador puede realizar una auditoría para comprobar que no se ha introducido ningún código malicioso, y a su vez, cualquier entidad puede añadir libremente encriptación adicional a la aplicación que utilice para proteger sus datos [15].

La Universidad de las Ciencias Informáticas teniendo en cuenta las últimas tendencias y tecnologías ha enfocado sus esfuerzos en crear un ambiente de desarrollo que beneficie la infraestructura productiva y que a la vez se ajuste a los recursos que posee la universidad. Es por ello que poco a poco se han ido integrando los proyectos al uso de la plataforma GNU/Linux. El Centro de Informatización Universitaria utiliza dicha plataforma, logrando una total incorporación a las tecnologías libres para el desarrollo de los proyectos de la facultad 1.

## **1.5.2 Aplicaciones web**

Son programas que se diseñan para funcionar a través de un navegador de Internet; es decir, son aplicaciones que se ejecutan de forma online [16].

Las aplicaciones web pueden tener numerosos usos tanto para los visitantes como para los ingenieros de desarrollo, entre otros:

- ✓ Permitir a los usuarios localizar información de forma rápida y sencilla en un sitio web en el que se almacena gran cantidad de contenido. Este tipo de aplicación web ofrece a los visitantes la posibilidad de buscar contenido, organizarlo y navegar por él de la manera que estimen oportuna.

# Capítulo 1: Fundamentación teórica

- ✓ Recoger, guardar y analizar datos suministrados por los visitantes de los sitios. Una aplicación web permite guardar datos de formularios directamente en una base de datos, además de extraer datos y crear informes basados en la web para su análisis.
- ✓ Actualizar sitios web cuyo contenido cambia constantemente. Una aplicación web evita al diseñador tener que actualizar continuamente el código HTML (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales) del sitio. Los proveedores de contenido, como los editores de noticias, proporcionan el contenido a la aplicación y esta actualiza el sitio automáticamente [17].

## 1.5.3 Modelo Vista Controlador

Es una aproximación al *software* que separa la lógica de la aplicación de la presentación. En la práctica, permite que sus páginas web contengan mínima codificación ya que la presentación es separada del código PHP (Hypertext Preprocessor). El **Modelo** representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos. La **Vista** es la información que es presentada al usuario. Normalmente es una página web, pero en CodeIgniter, se puede ver como un fragmento de una página, como un encabezado o un pie de página. También puede ser una página RSS (Really Simple Syndication) o cualquier otro tipo de "página". El **Controlador** sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP (HyperText Transfer Protocol o Protocolo de Transferencia de Hipertexto) y generar una página web. CodeIgniter tiene un enfoque bastante flexible del Modelo Vista Controlador (en adelante MVC), ya que los Modelos no son requeridos. CodeIgniter también permite incorporar sus códigos existentes, o incluso desarrollar librerías de núcleo para el sistema, habilitándolo a trabajar de forma que tenga más sentido para el desarrollador [18].

## 1.5.4 Notación para el modelado de procesos

### Business Process Modeling Notation (BPMN)

Business Process Modeling Notation (en adelante BPMN) es una notación gráfica que describe la lógica de pasos en un proceso de negocio. Esta notación se ha diseñado especialmente para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes en las diferentes actividades.

¿Por qué es importante modelar con BPMN?

- BPMN es un estándar de modelado de procesos internacionalmente aceptados.
- BPMN es independiente de cualquier metodología de modelado de procesos.

# Capítulo 1: *Fundamentación teórica*

- BPMN crea un puente estandarizado que reduce la brecha entre los procesos de negocio y su aplicación [19].

El principal objetivo de BPMN es proveer una notación que es fácilmente comprensible por todos los usuarios de negocios, de los analistas de negocio que crean los borradores iniciales de los procesos, a los desarrolladores técnicos responsables de la aplicación de la tecnología que llevará a cabo los procesos y, por último, a la gente de negocios que administran y supervisan los procesos. Por lo tanto, BPMN crea un puente estandarizado para la diferencia entre el diseño de procesos de negocio y proceso de implementación [20].

## **1.5.5 Lenguaje de modelado**

### **UML**

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de lo que estos diagramas y símbolos significan [21].

UML puede ser utilizado para modelar distintos tipos de sistemas: sistemas de software, hardware, y organizaciones del mundo real. Se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Las extensiones de UML para el modelado de negocio aportan elementos muy importantes ya que proporcionan otras vistas de la arquitectura de negocio. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Además UML tiene entre sus ventajas que es independiente de la arquitectura [22].

## **1.5.6 Lenguaje de programación**

Un lenguaje de programación es un conjunto de instrucciones, órdenes, comandos y reglas que permite la creación de programas. Los ordenadores, como todo equipo electrónico, funcionan por las señales eléctricas que reciben o dejan de recibir. Cuentan con pequeños interruptores por donde pasa o no la corriente, de ahí que cada interruptor pueda encontrarse sólo en dos estados: apagado o encendido, 0 ó 1. Esto es lo que da la posibilidad a la máquina que ejecute algún proceso según la combinación de estados de cada uno de los interruptores. Los lenguajes permiten al programador indicar lo que debe hacer un programa, sin tener que escribir largas cadenas de ceros y unos, sino palabras (instrucciones) más comprensibles por las personas. Un lenguaje de programación es una herramienta que permite crear programas y software. Como ejemplo de ello, tenemos a Delphi, Visual Basic, Pascal, Java, entre otros [23].

### **Lenguaje del lado del servidor**



# Capítulo 1: *Fundamentación teórica*

Un lenguaje del lado del 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. Los scripts en lenguaje del lado del servidor son almacenados, obviamente, en este, que es quien los ejecuta y traduce a HTML, por lo que su código queda inaccesible para el usuario, que solo recibe código HTML [24].

## **PHP**

Es un lenguaje de programación utilizado para la creación de sitios web. PHP es un acrónimo recursivo que significa “Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas.

### Ventajas:

- ✓ Muy fácil de aprender.
- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Orientado a objetos. Clases y herencia.
- ✓ Es un lenguaje multiplataforma: Linux, Windows, Mac OS.
- ✓ Capacidad de conexión con la mayoría de los gestores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server.
- ✓ Capacidad de expandir su potencial utilizando módulos.
- ✓ Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Incluye gran cantidad de funciones.
- ✓ No requiere definición de tipos de variables ni manejo detallado del bajo nivel [25].

## **Lenguaje del lado del cliente**

Los lenguajes de programación del lado cliente son totalmente independientes del servidor, lo cual permite que las páginas puedan ser albergadas en cualquier sitio, ya que únicamente dependen del navegador [24].

## **CSS**

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML (Siglas del inglés

# Capítulo 1: Fundamentación teórica

Extensible HyperText Markup Language o Lenguaje de Marcado de Hipertexto Extensible). Es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes [26].

## HTML

Desde el surgimiento de Internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). HTML es sencillo, el texto es presentado de forma estructurada y agradable, no necesita de grandes conocimientos cuando se cuenta con un editor de páginas web, sus archivos son pequeños, permite un despliegue rápido, es de fácil aprendizaje y lo admiten todos los exploradores. Es un lenguaje estático, que no permite la creación de páginas dinámicas. La interpretación de cada navegador puede ser diferente [24].

## XML

Extensible Markup Language (lenguaje extensible de etiquetado), es un método para introducir datos estructurados en un archivo de texto.

Por su sintaxis XML tiene cierta similitud con HTML, maneja etiquetas (tags) y atributos, pero a diferencia de HTML el lenguaje extensible de etiquetado utiliza las etiquetas para delimitar los datos, y la interpretación de los mismos dependen completamente de la aplicación que los esté leyendo.

### Ventajas:

- ✓ Una gran ventaja de XML es que es gratuito ya que no se requiere adquirir una licencia costosa como sucede con otros sistemas.
- ✓ Solo requiere un editor de textos ASCII para desarrollar aplicaciones en XML (por ejemplo: Bloc de notas), y un visualizador con un *parser* (analizador) adaptado para XML (por ejemplo: Microsoft Internet Explorer 5.0 o posterior).
- ✓ Las aplicaciones de XML pueden ser muchas, ya que no solamente se limita a ser un lenguaje que permita interactuar con sistemas diseñados para web, sino que está pensado como un lenguaje universal que facilite su uso con casi cualquier plataforma, incluso podemos interactuar con aplicaciones de oficina como Microsoft Excel [27].

## JavaScript

# Capítulo 1: *Fundamentación teórica*

Lenguaje interpretado que no requiere compilación y se utiliza principalmente para crear páginas web dinámicas. Fue creado por Brendan Eich en la empresa Netscape Communications. Es utilizado principalmente en páginas web, similar a Java, aunque no es un lenguaje orientado a objetos, pues no dispone de herencia. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript. El código JavaScript es ejecutado en el cliente, y es visible por cualquier usuario [24].

## 1.5.6 Herramienta de modelado

### Visual Paradigm

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste [28]. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta soporta UML y Notación para el Modelado de Procesos de Negocio (en adelante BPMN), permite realizar ingeniería tanto directa como inversa. BPMN define diagramas de procesos de negocios basados en la técnica de diagramas de flujo, adaptados para graficar las operaciones de los procesos de la organización.

Es posible generar código desde Visual Paradigm para plataformas como .NET, Java y PHP, así como obtener diagramas a partir del código, esto es de gran utilidad pues ahorra tiempo a los desarrolladores y reduce las posibilidades de cometer errores. Brinda la posibilidad de obtener una base de datos relacional y el código necesario para acceder a ésta a partir del diagrama entidad relación, además se conecta fácilmente a varios servidores de base de datos. Se integra con varios ambientes de desarrollo integrados (IDE) lo cual permite pasar del código al modelado y viceversa (Visual Paradigm International Ltd., 2007).

## 1.5.7 Herramienta para el prototipado

### Evolus Pencil

La única misión del proyecto Pencil es construir una herramienta libre y de código abierto para crear diagramas y prototipos de interfaz gráfica de usuario que todos puedan usar.

#### Características Principales:

- Construir plantillas de diagramación y prototipado.
- Conexión entre páginas.
- Exportación a formatos HTML, PNG<sup>8</sup>, Documento de Office, Documento de Word y PDF.

---

<sup>8</sup> PNG: formato gráfico basado en un algoritmo de compresión sin pérdida para mapa de bits.

# Capítulo 1: *Fundamentación teórica*

- Soporte de Hacer/Deshacer.
- Multiplataforma: Puede ser instalado tanto en Windows, como Linux, así como puede agregarse como complemento para el Firefox.
- Funcionamiento del tipo: arrastrar y soltar.

El funcionamiento es muy sencillo y el resultado exquisito. Se puede presentar rápidamente un esbozo de la aplicación a los clientes, con unos pocos minutos y con un resultado impresionante [29].

## **1.5.8 Marco de trabajo (Framework)**

Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de *software*, un *framework* es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de *software* puede ser organizado y desarrollado [30].

### **CodeIgniter**

CodeIgniter es un conjunto de herramientas para personas que construyen aplicaciones web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido que si lo escribiese desde cero, proporcionándole un rico juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y una estructura lógica para acceder a esas librerías. CodeIgniter permite creativamente enfocarse en el proyecto minimizando la cantidad de código necesario para una tarea dada. CodeIgniter se encuentra bajo una licencia OpenSource Apache/BSD-style, lo cual facilita el éxito de las tendencias de migración a software libre del país y de la UCI, pudiéndose utilizar en cualquier proyecto que se desee. CodeIgniter está escrito para ser compatible con PHP 4; aunque corre en PHP 5, simplemente no toma ventaja de cualquiera de las características nativas que sólo están disponibles en esa versión. Es verdaderamente liviano, ya que el núcleo del sistema sólo requiere unas pocas librerías que permiten realizar las tareas de desarrollo web más comunes como: acceder a una base de datos, mandar un correo electrónico, validar datos de un formulario, mantener sesiones, manipular imágenes. Aunque se le pueden agregar librerías cargándolas dinámicamente a pedido, basado en las necesidades de un proceso dado, así que el sistema base es delgado y bastante rápido. Usa la arquitectura Modelo Vista Controlador (MVC), que permite una buena separación entre lógica y presentación. Esto es particularmente bueno para proyectos en los cuales los diseñadores están trabajando con sus archivos de plantilla, ya que el código en esos archivos será mínimo. CodeIgniter cuenta con clases de base de datos llenas de características con soporte para varias plataformas, una clase de FTP (File Transfer Protocol o Protocolo de Transferencia de Archivos), localización,

# Capítulo 1: *Fundamentación teórica*

paginación, encriptación de datos, puntos de referencia, cacheo de páginas completas, historial de errores, clase de calendario, clase de codificación Zip y clase de prueba de unidad [31].

## **1.5.9 Sistema de gestión de base de datos**

Los sistemas gestores de bases de datos surgen por la necesidad que tienen las empresas de manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes de una manera segura. Ante este enfoque, un sistema gestor de bases de datos (en adelante SGBD) deberá ofrecer soluciones de forma fiable, rentable y de alto rendimiento. Los SGBD proporcionan herramientas de apoyo a la toma de decisiones ("datawarehouse") proporcionando una plataforma de transacciones "en línea" que hacen que la información esté siempre actualizada y consistente. Ofrecen además, las herramientas de administración completas que simplifican la tarea de la configuración, seguridad, creación y gestión de bases de datos; y facilitan los mecanismos de integración con otros sistemas, políticas de copias de seguridad y herramientas que permitan su programación tanto a nivel de diseño como a nivel de reglas o procedimientos que encapsulen la arquitectura de la base de datos [32].

### **PostgreSQL**

Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales. Está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo y se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL (Structured Query Language). Entre las características de PostgreSQL están: alta concurrencia, que evita tener que bloquear una tabla cuando se está escribiendo en ella; copias de seguridad en línea, replicación asíncrona, transacciones anidadas y optimizador de consultas [33].

## **1.5.10 Entorno de desarrollo**

Un entorno de desarrollo integrado o IDE (Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Constituye un editor de código para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación que pueda funcionar con diferentes lenguajes de programación. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic [34].

### **NetBeans**

# Capítulo 1: *Fundamentación teórica*

NetBeans IDE es un entorno de desarrollo. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación como por ejemplo PHP, HTML y C++. Existe además un número importante de módulos para extender el NetBeans IDE. Es un producto libre y gratuito sin restricciones de uso. También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio [35].

## **1.5.11 Plataforma para el desarrollo**

En informática, una plataforma es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones). Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles [36].

### **GNU/Linux**

GNU/Linux es un sistema operativo de *software* libre que cumple las normas POSIX (Portable Operating System Interface based on UNIX), su base es un núcleo o Kernel monolítico llamado Linux combinado con un grupo de librerías y herramientas. Su estructura general es la típica de cualquier sistema UNIX (núcleo, "intérprete de comandos", aplicaciones). GNU/Linux tiene todas las características que se pueden esperar de un moderno y flexible sistema operativo. Incluye multitarea real, memoria virtual, librerías compartidas, dirección y manejo propio de memoria. Es sin lugar a dudas uno de los ejemplos más prominentes del *software* libre y del desarrollo del código abierto. Varias son las distribuciones de GNU/Linux que se han creado a nivel mundial gracias al trabajo constante de los desarrolladores y promotores del *software* libre en el mundo entero, algunas de ellas son: Debian, Mandriva, Ubuntu, Novel/Suse, Red Hat y Gentoo [37].

## **1.6 Conclusiones**

El desglose de los conceptos fundamentales del negocio condujo a un mayor entendimiento del objeto de estudio en cuestión. Luego de un estudio del estado del arte de otras soluciones existentes en Cuba y el mundo se pudo establecer una comparación entre los mismos y determinar las distintas funcionalidades que sirven como base de conocimiento para el desarrollo del Sistema de Portadores Energéticos.

Movidos por las actuales tendencias de desarrollo de *software* en la facultad 1 se ha orientado el ambiente productivo de manera que el mismo se enfoque en la vinculación de técnicas más modernas a la producción. Luego de haber estudiado las distintas técnicas de programación, lenguajes, plataformas, herramientas y metodologías de desarrollo de *software*, se especifica la selección por

## *Capítulo 1: Fundamentación teórica*

parte del Centro de Informatización Universitaria (CENIA) de la UCI el uso de la programación orientada a objetos, lenguaje de programación PHP 5.0, sobre la plataforma GNU/Linux, distribución Ubuntu 10.04 o superior, sistema gestor de base de datos PostgreSQL, en su versión 8.4, marco de trabajo CodeIgniter, entorno de desarrollo NetBeans 6.8 o superior y metodología ágil SXP para el proceso de desarrollo.

En general se estima que el sistema propuesto brinde un espacio dinámico y eficiente; además contribuya a la informatización de servicios para la comunidad universitaria, arduo trabajo que en el futuro hará a la Universidad de las Ciencias Informáticas única de su tipo en Cuba. La desventaja principal que presenta, es que se ajusta a las características de la UCI y para utilizarla en otra entidad tendría que reajustarse.

# *Capítulo 3. Construcción y validación de la propuesta de solución*

## **Capítulo 2: Descripción y análisis de la solución propuesta.**

### **2.1 Introducción**

La creciente demanda de energía a nivel mundial ha originado la racionalización de la misma, es por ello que resulta de vital importancia la planificación eficiente de la energía. Cuba y la UCI no están exentas del ahorro, razón por lo que la Dirección de Energía dentro de la universidad necesita tener un mayor control de los recursos energéticos con que cuenta la misma. A continuación se muestran los artefactos asociados al desarrollo de la solución en cuestión y que documentan el ciclo de desarrollo del Sistema de Portadores Energéticos.

### **2.2 Especificación de requisitos de software**

Roger S. Pressman expresa que para que un esfuerzo de desarrollo de software tenga éxito, es esencial comprender perfectamente los requisitos del software. Independientemente de lo bien diseñado o codificado que esté un programa, si se ha analizado y especificado pobremente, decepcionará al usuario y desprestigiará al que lo ha desarrollado. La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir (Pressman, 2002).

La calidad con que se realice la captura de los requisitos va a influenciar todo el proceso de desarrollo del software repercutiendo en el resto de las fases de desarrollo del mismo, es por ello que se utilizaron como técnicas para la captura de requerimientos:

Reuniones: Desde que se concibió el Sistema de Portadores Energéticos se mantuvo un intercambio con el personal de la Dirección de Energía, realizando reuniones programadas por el grupo de trabajo del proyecto para explorar el campo de acción e identificar los requisitos del negocio.

Prototipos: Se confeccionaron prototipos de interfaz de usuario que se le fueron presentando progresivamente al personal de la Dirección de Energía a medida que se iba describiendo los requisitos, de manera que el cliente se familiarizara con la propuesta y quedara satisfecho.

Entrevistas: Se realizaron encuentros dirigidos a requisitos específicos detectados, de manera que el equipo del proyecto aclarara todas las dudas al respecto.

#### **2.2.1 Requerimientos Funcionales**

A continuación se listan algunos de los requerimientos funcionales de mayor criticidad dentro de la actual solución:

##### Planificación

- ✓ Calcular Plan anual a partir de consumos históricos.
- ✓ Calcular peso que representa cada mes y cada servicio del total (año).



# Capítulo 3. Construcción y validación de la propuesta de solución

- ✓ Desagregar el plan anual por meses y por servicios teniendo en cuenta los pesos<sup>9</sup>.
- ✓ Mostrar plan anual.
- ✓ Modificar plan anual.
- ✓ Calcular demanda mensual.
- ✓ Calcular peso que representa cada servicio del total mensual.
- ✓ Desagregar plan mensual por servicios teniendo en cuenta los pesos.
- ✓ Mostrar plan mensual por servicios.
- ✓ Modificar plan mensual por servicios.
- ✓ Calcular peso que representa cada día del total mensual para cada servicio.
- ✓ Desagregar plan mensual de los servicios por días del mes.
- ✓ Mostrar plan mensual de los servicios por día
- ✓ Modificar plan mensual de los servicios por día
- ✓ Calcular el peso que representa cada consumidor interno del total diario.
- ✓ Desagregar el plan diario entre los consumidores internos teniendo en cuenta los pesos.
- ✓ Mostrar plan diario de los consumidores internos.
- ✓ Leer y guardar lecturas para consumidores automáticos.

## 2.2.2 Requerimientos no funcionales

Los requisitos no funcionales imponen restricciones en el diseño o la implementación. El Sistema de Portadores Energéticos tiene entre sus requerimientos no funcionales los siguientes:

### Usabilidad

Facilidad de uso: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo por parte de los usuarios los cuales deben poder acceder de manera rápida y efectiva a la información solicitada. Debe además ser una interfaz de manejo cómodo donde la curva de aprendizaje para los usuarios sea lo menos inclinada posible y que posibilite en estos una rápida adaptación.

Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por parte de los mismos sobre la herramienta de trabajo.

Emplear perfiles de usuario: diferenciar las interfaces y opciones para los usuarios que accedan al sistema según los diferentes roles que estos tengan dentro del mismo.

---

<sup>9</sup> Pesos: porciento que representa del total (se especifica en cada tipo de desagregación).

# Capítulo 3. Construcción y validación de la propuesta de solución

Menús: el sistema debe presentar una serie de menús tanto laterales como en barras horizontales de iconos que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.

## Fiabilidad

Disponibilidad: El sistema estará disponible las 24 horas del día los siete días de la semana.

## Eficiencia

El sistema debe soportar un tiempo de respuesta menor o igual a 5 segundos.

El sistema debe soportar una conexión simultanea de más de 10 000 usuarios.

## Soporte

Grupo de soporte y asesoría: el sistema contará con un grupo de soporte y asesoría al cliente del producto destinado a brindar asesoría y soporte técnico al mismo.

## Interfaz

Interfaz Web: la interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.

Interfaz interna: la interfaz interna estará determinada por los desarrolladores, construyendo así una vista escalable de las clases o agrupaciones de clases que permitirán un mejor encapsulamiento de las funcionalidades y una mayor abstracción modular del sistema.

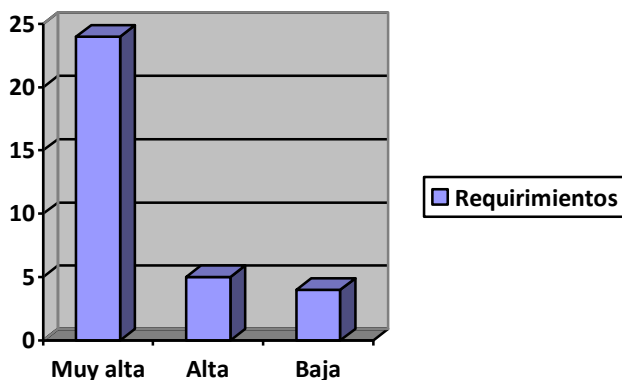


Figura 7. Requisitos identificados de acuerdo a su prioridad

Véanse los requerimientos definidos en el artefacto Lista de Reserva del Producto en el Anexo 6. Lista de Reserva del Producto.

## 2.3 Definición de procesos

## *Capítulo 3. Construcción y validación de la propuesta de solución*

Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente o mercado particular. Implica un fuerte énfasis en 'cómo' se hace el trabajo en una organización, en contraposición al enfoque en 'qué' de producto. Así, un proceso es un ordenamiento específico de actividades de trabajo a través del tiempo y del espacio, con un comienzo, un fin, entradas y salidas claramente identificados: una estructura para la acción [38]. La notación para el modelado de procesos de negocio BPMN propone una descripción detallada de cada uno de los procesos envueltos en el Sistema de Portadores Energéticos, ofreciendo como mejora la informatización de los mismos y la incorporación de nuevas funcionalidades.

BPMN ha tratado de dar soporte a tres categorías principales de procesos: Orquestación, Coreografía y Colaboración. En este caso, los procesos han sido descritos como Orquestación pues se muestra una única perspectiva de coordinación, es decir, cómo una única entidad del negocio lleva a cabo las cosas (Stephen A. White, 2009).

A continuación se describe el flujo actual de los procesos y en el caso de planificación la modelación del proceso con las mejoras incorporadas para la actual solución:

### **2.3.1 Planificación**

Actualmente el proceso planificación en la Dirección de Energía lo lleva a cabo un especialista manualmente. Para ello primero debe estimar el plan anual basándose en consumos históricos de años anteriores; luego, realizar la desagregación anual por meses y por servicios. Igualmente todos los meses debe demandar el plan estimado y redistribuir entre los servicios. Luego debe realizar la desagregación diaria de los servicios UCI para los consumidores internos. Además de la informatización de este proceso el Sistema de Portadores energéticos propone la exportación de los planes estimados a formato excel y pdf, así como el envío de los mismos por correo a los responsables de los servicios.

# Capítulo 3. Construcción y validación de la propuesta de solución

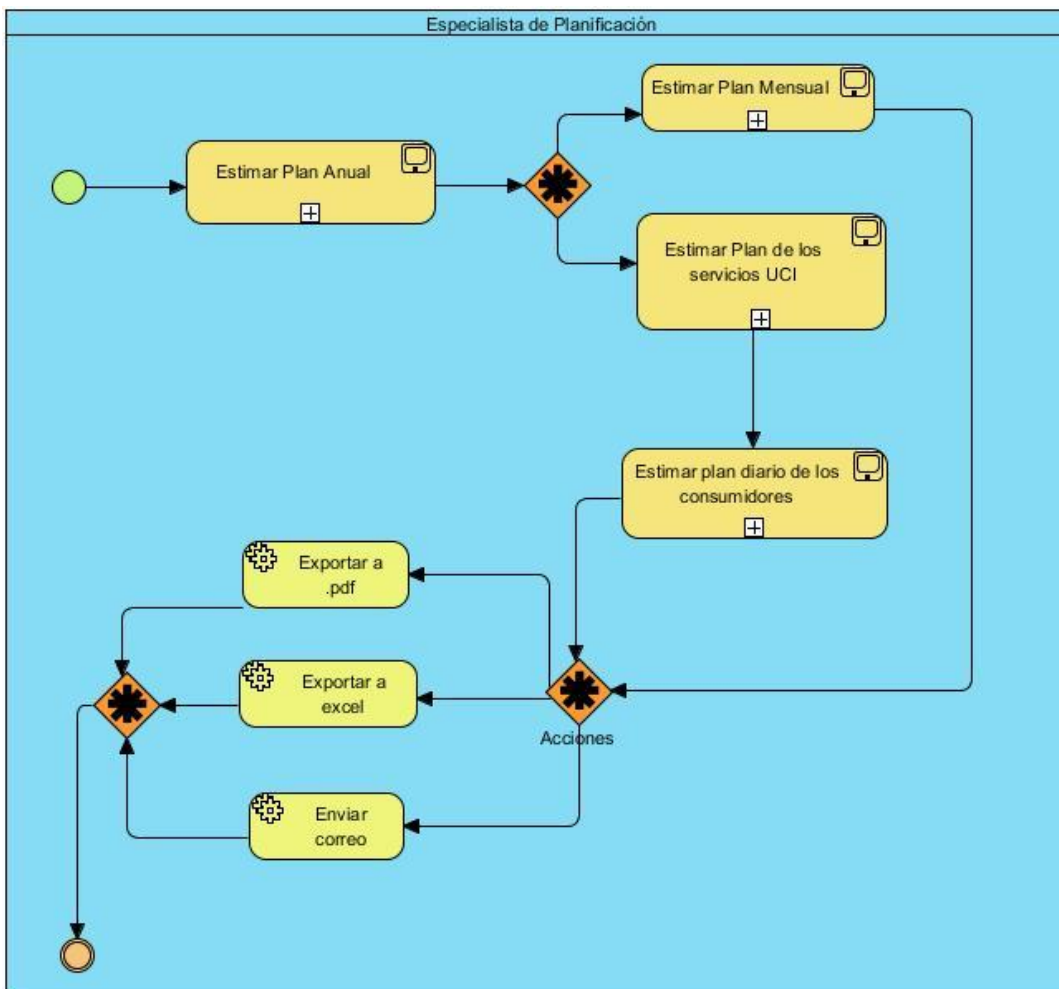


Figura 8. Proceso Planificación

## 2.3.2 Gestión del consumo

La gestión del consumo en la Dirección de Energía se realiza de manera manual, es decir, tomando las lecturas se realiza el cálculo del consumo; por lo tanto este nuevo sistema propone como vía de solución el cálculo automático del consumo.

## 2.3.3 Configuración

El antiguo sistema con que cuenta la Dirección de Energía no presenta esta funcionalidad, es por ello que en aras de desarrollar un sistema más usable y eficiente se decide incorporar la misma.

Véase el artefacto *Definición de procesos con la descripción del anterior modelo* y el resto de los procesos en el Anexo 4. Anexo 4. Definición de procesos

# Capítulo 3. Construcción y validación de la propuesta de solución

## 2.4 Personas relacionadas con el sistema

La Dirección de Energía cuenta con distintos especialistas que se desempeñan en distintas áreas relacionadas con los distintos portadores energéticos que posee la universidad. En la rama de la gestión energética se especializan:

**Especialista general:** Actúa directamente en la gestión de cada uno de los procesos de los portadores energéticos dentro de la Dirección de Energía.

**Técnico general:** Se relaciona con los portadores facilitando el funcionamiento de los mismos para su correcta gestión, realiza funciones como la recogida manual de las lecturas y la entrada manual de datos en la aplicación.

## 2.5 Historias de usuario

Las historias de usuario son la técnica utilizada en XP para describir las funcionalidades especificadas por los requisitos del software. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla [39].

A continuación se muestra la descripción de una de las historias de usuario de mayor criticidad dentro de la actual propuesta de solución:

Historia de Usuario	
<b>Código:</b> HU_PE_EE 11	<b>Nombre Historia de Usuario:</b> Plan Anual
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 8. Calcular plan anual a partir de consumos históricos. 9. Calcular peso que representa cada mes y cada servicio del total (año). 10. Desagregar el plan anual por meses y por servicios teniendo en cuenta los pesos. 11. Mostrar plan anual. 12. Modificar plan anual.	
<b>Programador:</b> Noél Hernández Bautista	<b>Iteración Asignada:</b> 1
<b>Prioridad :</b> Muy alta	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 2

## Capítulo 3. Construcción y validación de la propuesta de solución

### **Descripción:** (Pseudocódigo)

(Mostrar Planificación del año siguiente al actual. Encabezado)

-Sumar los consumos de todos los servicios por meses de años anteriores. (Calcular consumo de años anteriores). *Tomar datos de tabla Consumo.*

-Promediar los consumos de los años anteriores, utilizar además Teorema del olvido matemático para minimizar el margen de error.

*Olvido matemático: Al promediar se toma el último dato doble pues es el que más se acerca a la realidad.*

-Guardar en la base de datos los resultados del plan anual y desagregación correspondientes como plan estimado.

-Mostrar al usuario el resultado obtenido para el plan anual en un campo de texto editable.

-Si se realiza alguna modificación en el plan anual este nuevo valor se guardaría en la base de datos como plan real. *Guardar datos en tabla Plan.*

-Al dar clic en la opción Guardar se sale de la vista y se muestra mensaje "Los datos han sido guardados satisfactoriamente".

-Promediar los consumos mensuales existentes de todos los servicios aplicando olvido matemático.

-Calcular el peso que representa cada mes del total anual.

-Promediar los consumos mensuales existentes de cada servicio aplicando olvido matemático. *Tomar datos de tabla Consumo.*

-Calcular el peso que representa cada servicio del total mensual de todos los servicios.

-Si el usuario da clic en la acción Desagregar se muestra una tabla con la desagregación anual por meses y por servicios.

-Si se realiza alguna modificación en el consumo de algún servicio este nuevo valor se guardaría en la base de datos como plan real al dar clic en el botón Modificar.

-Al dar clic en el botón Modificar no solo se sobrescribiría en la base de datos con los datos nuevos sino que se realizaría la redistribución teniendo en cuenta la modificación que se realizó.

*Redistribución: División equitativa para el resto de los servicios o división equitativa para los meses siguientes partiendo del que se modificó.*

-Si el usuario da clic en la opción Exportar debe darse la opción de que lo haga en excel y en pdf.

- Si el usuario da clic en la opción Cancelar se cierra la vista y se muestra la vista principal.

**Observaciones:** Interactúa con esta acción el especialista de planificación.

# Capítulo 3. Construcción y validación de la propuesta de solución

Prototipo de interfaz:

Planificación Año [año siguiente]

Plan anual:  Desagregar

C EE EP

Meses	Servicios...n
Enero	
Febrero..	
Total	

Slider:

Modificar Guardar Cancelar

IU. 1 Plan Anual

Tabla 2. HU Estimar Plan Anual

Véanse el resto de las descripciones para todas las funcionalidades en el Anexo 3. Historias de usuario.

## 2.6 Conclusiones

Debido a que el sistema que presentaba la Dirección de Energía no permitía el correcto control de la energía, surge la necesidad de implementar un sistema que incorpore de manera informatizada todos los procesos que realiza dicho departamento en la UCI para la gestión energética. Durante el desarrollo de este capítulo se hizo referencia al trabajo realizado como parte la fase Planificación-Definición de SXP para la elaboración de la propuesta de solución, haciendo una descripción de los artefactos que genera dicha fase, de manera que tanto los requisitos identificados como las historias de usuarios sirvan de guía para la posterior implementación de las funcionalidades.

# Capítulo 3. Construcción y validación de la propuesta de solución

## Capítulo 3: Construcción y validación de la propuesta de solución

### 3.1 Introducción

La etapa de implementación de un sistema es el resultado del análisis previo y la identificación de los requerimientos del cliente. La metodología SXP plantea que la implementación de un *software* debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado. Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas. Esta metodología se enfoca en la implementación, motivo por lo que se hace necesario de un buen uso de las técnicas de programación y validación para el desarrollo de un buen producto final.

### 3.2 Estándares de codificación

Tomando como referencia los estándares de codificación definidos por el Centro de Informatización Universitaria se realizó la implementación del sistema logrando la mayor organización posible en el código. A continuación se muestran ejemplos de la utilización de dichos estándares:

#### 1. Identación, llaves de apertura y cierre, y tamaño de las líneas

Usar una indentación sin tabulaciones, con un equivalente a 4 espacios, para mantener integridad en las revisiones svn. El uso de las llaves “{}” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres. Para mantener la legibilidad del código [40].

Ejemplo:

```
public function responsablesAsociadosConsInternos()
{
    return $this->_ci->gestionar_plan_mng->responsablesAsociadosConsInternos();
}
```

#### 2. Conversión de nomenclatura

Variables: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Ejemplo:

\$consumo

\$planesParaEIMes

Funciones: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa [40].



# Capítulo 3. Construcción y validación de la propuesta de solución

Ejemplo:

```
public function desagregar2($id)
{
    $this->db->select('*');
    $this->db->from('public.tb_dconsumo');
    $this->db->where('public.tb_dconsumo.id_consumidor', $id);
    $this->db->order_by('public.tb_dconsumo.fecha_consumo');
    $resultado = $this->db->get()->result();
    return $resultado;
}

public function listarDatosReales($fecha, $id)
{
    $this->db->select('*');
    $this->db->from('public.tb_dplan');
    $this->db->join('public.tb_dconsumidor ', 'public.tb_dplan.id_consumidor =
public.tb_dconsumidor.id_consumidor');
    $this->db->where('public.tb_dplan.estado_plan', 'Estimado');
    $this->db->where('public.tb_dplan.fecha_plan', $fecha);
    $this->db->where('public.tb_dplan.id_consumidor', $id);
    $aux = $this->db->get()->result();
    return $aux[0];
}
```

### 3. Estructuras de control

Se incluye if, for, foreach, while, switch, entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales, esto aumenta la legibilidad y disminuye la probabilidad de errores lógicos [40].

Ejemplos:

```
if ($acceso['funcionalidad'] == 1 && $acceso['energia'] == 1)
{
    $anual = $this->gestionar_planificacion_lib->buscarPlanExistente();
    $this->template->set_data('datos', $anual);
}
```

## Capítulo 3. Construcción y validación de la propuesta de solución

```
$this->template->render('planificacion/plan_anual_view');  
$this->work->traza(45);  
}  
else  
{  
    $this->Denegar();  
}
```

Véase la descripción detallada de los estándares de codificación en la planilla Arquitectura de Software metodología SXP definida por el CENIA [40].

### 3.3 Análisis de posibles implementaciones (reutilización de código)

La reutilización de código es un aspecto de suma importancia cuando se trata de reducir el tiempo de desarrollo. Consiste en la utilización de programas o fragmentos de códigos ya existentes, que pueden fortalecer o desempeñar una funcionalidad en un producto determinado. Para el caso de la actual propuesta de solución se han utilizado fragmentos de código para el desarrollo de funcionalidades como exportar a excel y pdf. Además, el equipo del proyecto como parte del entorno productivo de la universidad no está ajeno a la reutilización de código sino que fomenta el intercambio entre desarrolladores para lograr mayor rapidez en el desarrollo y lograr una estandarización en determinadas funcionalidades que presenten aspectos comunes.

### 3.4 Vista de datos

Una vez que se han definido y modelado las funcionalidades que debe realizar el módulo a implementar, se deben definir las entidades del negocio con sus atributos, que no son más que elementos gestionados y utilizados para realizar las operaciones. Estas entidades son tratadas en la declaración de los servicios e implementación de los métodos. A continuación se muestra el modelo de datos de la aplicación:

# Capítulo 3. Construcción y validación de la propuesta de solución

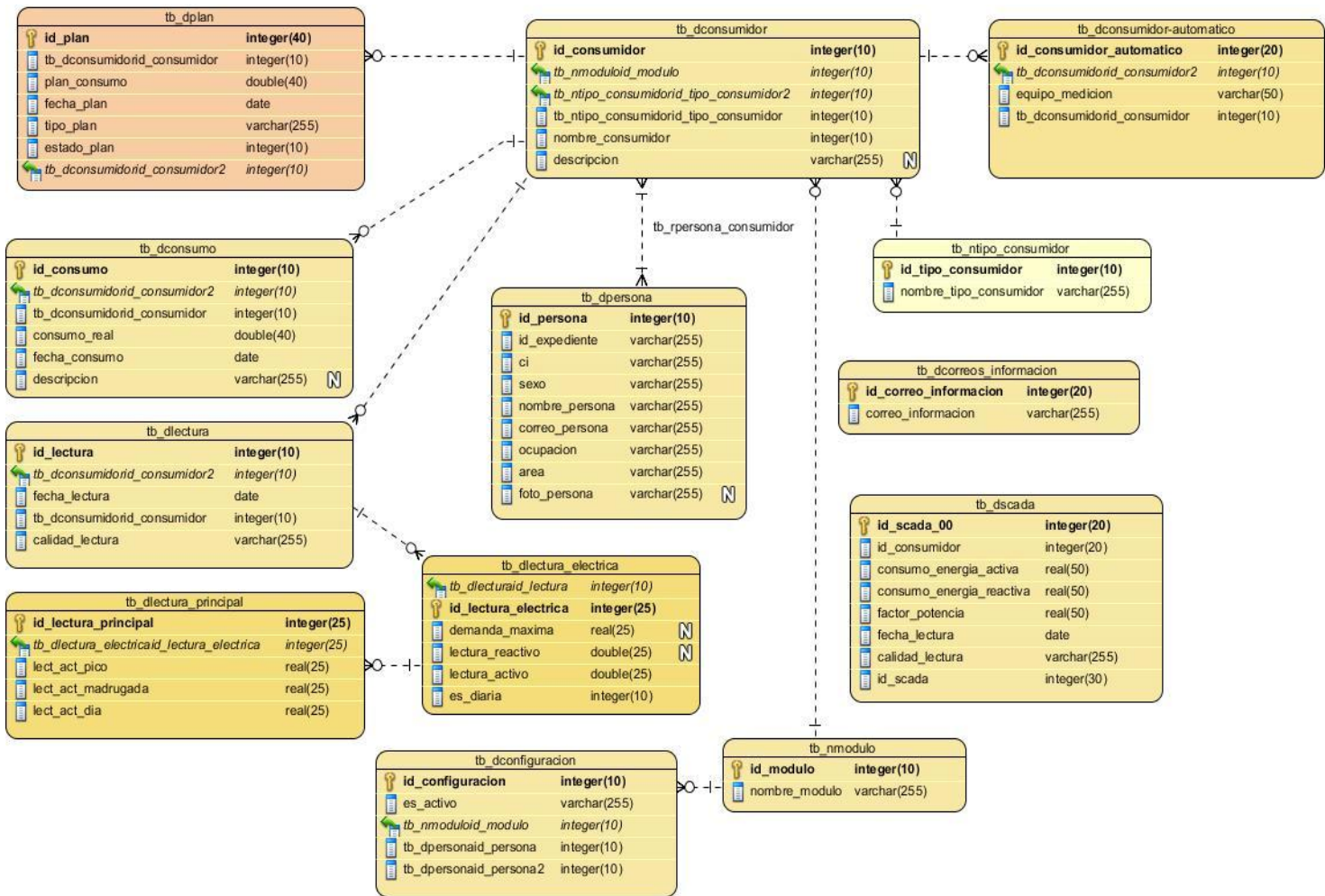


Figura 9. Modelo de datos

### 3.4.1 Descripción de las entidades fundamentales

A continuación se muestran las características lógicas de los datos que se van a utilizar en de una de las entidades del Sistema de Portadores Energéticos descrita en el artefacto Diccionario de Datos:

<b>Nombre de la entidad</b>	Plan (tb_dplan)		
<b>Descripción de la entidad</b>	Plan del consumo eléctrico que se propone para los distintos consumidores.		
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Puede ser Nulo</b>
id_plan	Identifica el plan	Entero	No

## Capítulo 3. Construcción y validación de la propuesta de solución

id_consumidor	Identifica el consumidor	Entero	No
plan_consumo	Plan de consumo eléctrico determinado	Real	No
fecha_plan	Fecha en que se crea el plan	Date	No
tipo_plan	Tipo de plan. Puede ser: anual, mensual o diario.	Cadena de caracteres (varchar)	No
estado	Estado del Plan. Puede ser: propuesto y aprobado.	Cadena de caracteres (varchar)	No

Tabla 3. Descripción de la entidad Plan

Véanse el resto de las entidades en el Anexo2. Descripción de las entidades de la base de datos.

### 3.5 Arquitectura

El marco de trabajo CodeIgniter, debido a que utiliza como patrón de arquitectura Modelo Vista Controlador, propone un modelo de diseño a través del cual deben regirse las aplicaciones que se desarrollen en el mismo.

La capa Vista contiene el paquete **views**: el cual incluye a su vez las páginas HTML con las que interactúa el cliente, llamadas interfaces.

La capa Controlador contiene los paquetes **controllers**: el cual incluye las clases que contienen las llamadas a los métodos que se encuentra en el paquete **libraries**, este último incluye los métodos relacionados con la lógica del negocio.

La capa Modelo contiene los paquetes **manager**: donde se incluyen los modelos que serán utilizados en el módulo que se está desarrollando, sirve de intermediario entre el paquete **libraries** y **models**, este último contiene las consultas de acceso a datos, además de un paquete denominado **base**, que encierra una serie de clases con métodos básicos predefinidos para el acceso a datos.

# Capítulo 3. Construcción y validación de la propuesta de solución

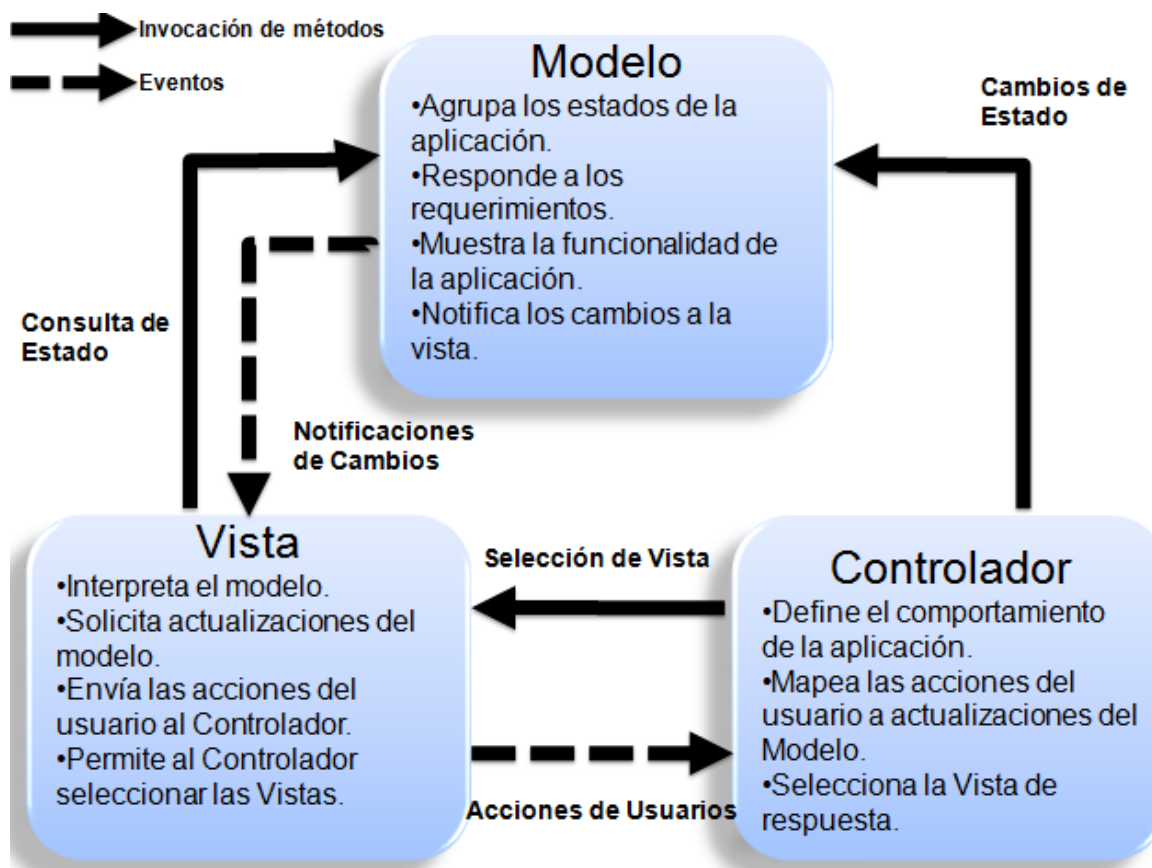


Figura 10. Modelo Vista Controlador

## 3.6 Patrones de diseño

El objetivo de los patrones es guardar la experiencia en diseños de programas orientados a objetos. Los patrones hacen más fácil reutilizar con éxito los diseños y arquitecturas, ayudan a elegir entre diseños alternativos, hacen a un sistema reutilizable y evitan alternativas que comprometen dicha reutilización.

Existen tres tipos de patrones de diseño:

- ✓ **Patrones de creación:** Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos un objeto delegará responsabilidades.
- ✓ **Patrones estructurales:** Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.

# Capítulo 3. Construcción y validación de la propuesta de solución

- ✓ **Patrones de comportamiento:** Se utilizan para organizar, manejar y combinar comportamientos [41].

## Patrones GOF utilizados:

A continuación se describen los patrones que más se ponen de manifiesto en el marco de trabajo CodeIgniter:

### Como parte de los patrones creacionales se encuentra:

- Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. El uso de la función **get\_instance()** dentro de la librería de CodeIgniter retorna el objeto súper de CodeIgniter (instancia única) permitiendo utilizar los recursos nativos del mismo.

## Patrones para asignar responsabilidades (GRASP) utilizados:

**Creador:** Asignarle a la clase B la responsabilidad de crear una instancia de clase A [40]. Ejemplo:

**Consumidor** tiene los datos de inicialización que serán transmitidos a **Consumidor Automático** cuando este objeto sea creado (así que **Consumidor** es un Experto respecto a la creación de **Consumidor Automático**).

**Experto:** Es el principio básico de asignación de responsabilidades. Indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtiene un diseño con mayor cohesión y así la información se mantiene encapsulada [40]. Ejemplo: para cumplir con la responsabilidad de conocer el plan y consumo de los consumidores internos de la universidad se asignaron tres responsabilidades:

CLASE	RESPONSABILIDAD
Plan	Conoce el plan de consumo de los consumidores
Consumidor	Conoce los consumidores internos de la universidad
Consumo	Conoce el consumo de los consumidores

Tabla 4. Asignación de responsabilidades

**Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado [40]. Ejemplo: Para realizar la funcionalidad “Desagregar plan anual”, al dar clic en el botón Desagregar la vista transmite el mensaje “desagregar” a la controladora **planificación\_cc**. Esta controladora hace de intermediario para enviar

## Capítulo 3. Construcción y validación de la propuesta de solución

los datos obtenidos a la librería `gestionar_planificacion_lib` que se encargará de realizar la desagregación.

**Alta Cohesión:** La información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase [40]. Ejemplo: Para registrar una lectura teniendo en cuenta el patrón *Creador*, **Lectura** es la encargada de realizar la funcionalidad, pero para garantizar una alta cohesión, la misma hace referencia a **Lectura Eléctrica** y esta a su vez delega a **Lectura Principal** la responsabilidad de efectuar el registro.

**Bajo Acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases [40]. Ejemplo: Para registrar una lectura teniendo en cuenta el patrón *Experto*, **Lectura** es la encargada de realizar la funcionalidad, pero para reducir su dependencia a otras clases, hace referencia a **Lectura Eléctrica** y esta a su vez delega a **Lectura Principal** la responsabilidad de efectuar el registro.

### 3.7 Vista de despliegue

"Los diagramas de despliegue muestran la configuración física de un sistema, revelando qué piezas de software se ejecutan sobre qué piezas de hardware." [42]

#### Componentes del diagrama de despliegue.

**Nodos:** Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos (impresoras, lectoras de códigos de barras o dispositivos de comunicación).

**Dispositivos:** Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela. Generalmente se usan para identificar el tipo de dispositivo. Los nodos se conectan mediante asociaciones de comunicación que indican:

- Algún tipo de ruta de comunicación entre los nodos.
- Los nodos intercambian objetos o envían mensajes a través de esta ruta.

**Conectores:** Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

# Capítulo 3. Construcción y validación de la propuesta de solución

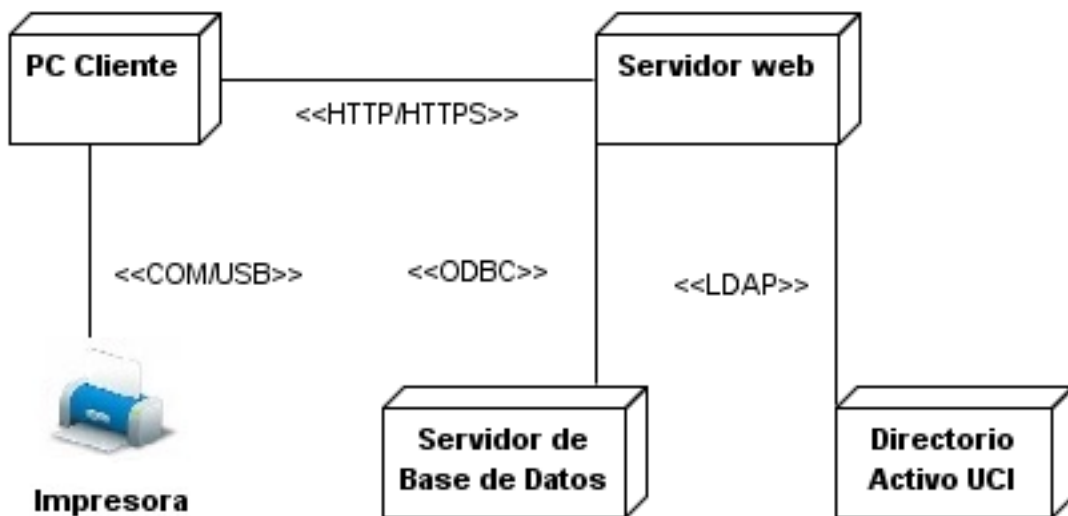


Figura 11. Modelo de Despliegue

**PC Cliente:** Computadora que cuente con un navegador actualizado y que siga los estándares web. Se recomiendan estaciones de trabajo con sistema operativo GNU/Linux.

**Servidor web:** Representa una estación donde estará montado el servidor Apache sobre el cual correrá la aplicación.

**Servidor de base de datos:** Representa el servidor donde estará el SGBD Postgres que dará respuesta a las peticiones hechas por la aplicación.

**Directorio activo:** directorio de información de los usuarios del dominio uci.cu.

### 3.8 Tareas de Ingeniería

Las tareas de ingeniería son las tareas asignadas al programador para implementar cada historia de usuario. Estas tareas pueden ser desarrollo, corrección, mejora u otra; en dependencia de la situación que se pueda presentar durante el transcurso de las iteraciones. A continuación se muestra un ejemplo de una tarea de ingeniería descrita para la solución del presente trabajo de diploma:

Tarea de ingeniería	
Número tarea: 2	
Nombre tarea: Estudiar la librería de JavaScript jQuery.	
Tipo de tarea: Estudio	Puntos estimados: 1



## Capítulo 3. Construcción y validación de la propuesta de solución

<b>Fecha inicio:</b> 10/09/2010	<b>Fecha fin:</b> 17/09/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Estudiar el funcionamiento de la librería de java Script jQuery y su integración con el marco de trabajo.	

Tabla 5. Tarea de ingeniería 2

Véanse el resto de las tareas en el Anexo 1. Tareas de ingeniería

### 3.9 Principales funcionalidades del sistema

#### ❖ Vista principal del Módulo Energía Eléctrica:

Para tener acceso a este módulo el usuario debe estar autenticado, una vez que tenga acceso a la página principal puede seleccionar en la parte derecha el módulo al cual desea entrar. En la figura se muestran las diferentes funcionalidades que nos brinda este módulo.

# Capítulo 3. Construcción y validación de la propuesta de solución



Figura 12. Módulo Energía Eléctrica.

## ❖ Plan anual:

Esta es la principal funcionalidad del proceso planificación, dado que a través de ella el sistema realiza la estimación para el plan anual de los servicios de la universidad basándose en consumos históricos de la misma. Para realizar dicha estimación se emplea el método del olvido matemático aplicado en la fórmula implementada. Este método se enfoca en darle más peso al consumo de los años más cercanos a la actualidad de manera que el promedio se acerque más a la realidad.

Ejemplo:  $\text{Consumo (2009)} + \text{Consumo (2010)} + \text{Consumo (2011)} + \text{Consumo (2011)} / 4$

Nótese que en este caso se selecciona el consumo del año 2011 como el más crítico y por ello se duplica su valor en la fórmula para el cálculo.

# Capítulo 3. Construcción y validación de la propuesta de solución

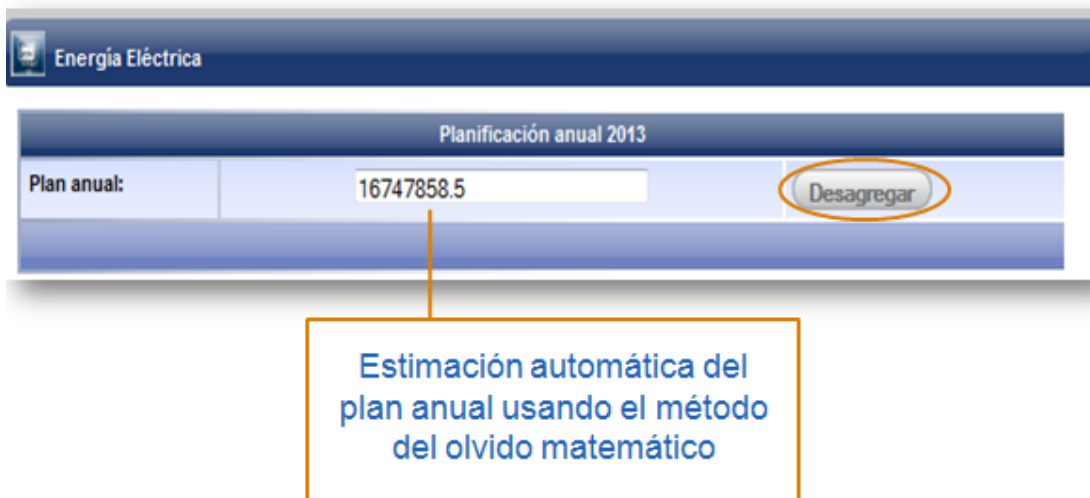


Figura 13. Plan Anual

❖ **Desagregación del plan anual por servicios (plan mensual de los servicios UCI):**

Luego de obtener el plan anual el usuario puede presionar el botón Desagregar (figura 13) y el sistema muestra, en la misma vista, una tabla con la distribución del plan anual entre todos los servicios que posee la universidad.

Serv.	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct
Mini UCI Artemisa	37815	37815	41017	41890	59836	55229	50589	0	57301	55800
Bombeo UJC	13944	10201	15221	12000	17060	14971	14222	0	12687	3559
Mercado concentrador	6696	6696	7684	7852	13559	13428	12756	0	11988	11545
Mini UCI Ciego de Avila	33950	33950	41578	49546	51275	52234	47675	0	54348	54630
Ecomarco UCI	6080	4507	14718	17000	14150	13428	6737	0	3195	3319
Circuito 1600(2680)	801889	881171	991970	974310	109234	103326	871349	397569	159496	119775
Circuito 1590(6795)	334194	333150	363664	413155	448432	487193	402375	225268	458198	433004
<b>Total</b>	<b>1234568</b>	<b>1307490</b>	<b>1475852</b>	<b>1515753</b>	<b>1696656</b>	<b>1669744.5</b>	<b>1405703</b>	<b>622637.5</b>	<b>2192663</b>	<b>1759606.5</b>

Desagregación del plan anual

Figura 14. Desagregación del plan anual por servicios

❖ **Plan mensual real por servicios UCI:**

## Capítulo 3. Construcción y validación de la propuesta de solución

El usuario puede acceder a esta funcionalidad a través del menú principal del módulo. El sistema muestra una tabla con la distribución mensual del plan para todos los consumidores, esta tabla muestra el plan real luego que el plan anual le ha sido aprobado a la universidad.



Serv.	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Se
Mini UCI Artemisa	14533.53	14533.53	14533.53	14533.53	14533.53	14533.53	14533.53	14533.53	14533.53
Mercado concentrador	3160.02	3160.02	3160.02	3160.02	3160.02	3160.02	3160.02	3160.02	3160.02
Mini UCI Ciego de Avila	13390.58	13390.58	13390.58	13390.58	13390.58	13390.58	13390.58	13390.58	13390.58
Bombeo UJC	3253.78	3253.78	3253.78	3253.78	3253.78	3253.78	3253.78	3253.78	3253.78
Ecomarco UCI	2205.71	2205.71	2205.71	2205.71	2205.71	2205.71	2205.71	2205.71	2205.71
Circuito 1600(2680)	3294528.9	3294528.9	3294528.9	3294528.9	3294528.9	3294528.9	3294528.9	3294528.9	3294528.9
Circuito 1590(6795)	1395654.88	1395654.88	1395654.88	1395654.88	1395654.88	1395654.88	1395654.88	1395654.88	1395654.88
Acciones									

### Plan mensual real por servicios UCI

Figura 15. Plan mensual por servicios

#### ❖ Plan mensual de los servicios UCI por día:

Esta funcionalidad muestra una tabla con el plan mensual para las dos líneas principales de la universidad, dando la opción de desagregar un mes específico para obtener la distribución de dicho mes de manera diaria.

## Capítulo 3. Construcción y validación de la propuesta de solución

Meses	Circuito 1600(2680)	Circuito 1600(2680)	Total	Acción
Enero	801889	334194	1136083	<input type="checkbox"/>
Febrero	881171	333150	1214321	<input type="checkbox"/>
Marzo	991970	363664	1355634	<input type="checkbox"/>
Abril	974310	413155	1387465	<input type="checkbox"/>
Mayo	1092344	448432	1540776	<input type="checkbox"/>
Junio	1033261.5	487193	1520454.5	<input type="checkbox"/>
Julio	871349	402375	1273724	<input type="checkbox"/>
Agosto	397569.5	225268	622837.5	<input type="checkbox"/>
Septiembre	1594965.5	458198.5	2053164	<input type="checkbox"/>
Octubre	1197751.5	433004	1630755.5	<input type="checkbox"/>
Noviembre	661952.5	396459	1058411.5	<input type="checkbox"/>
Diciembre	315614.5	286084	601698.5	<input type="checkbox"/>

### Plan mensual para los servicios principales de la UCI

Figura 16. Plan mensual de los servicios UCI por día

#### ❖ Configuración:

Con un clic se activan o desactivan los módulos influyendo también en la activación o desactivación del módulo en el menú de acceso rápido.

#### Listado de Módulos

Módulo	Activado
Agua	<input checked="" type="checkbox"/>
Incidencias	<input checked="" type="checkbox"/>
Energía Eléctrica	<input checked="" type="checkbox"/>

Figura 17. Configuración

### 3.10 Pruebas de software

Las pruebas de *software* son los procesos que permiten verificar y revelar la calidad de un producto. Estas se integran dentro de las diferentes fases del ciclo del *software*. Para determinar el nivel de calidad del producto se deben efectuar pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema [43].

## Capítulo 3. Construcción y validación de la propuesta de solución

Existen dos métodos de pruebas:

Mediante la prueba de **caja blanca**, el ingeniero del *software* puede obtener casos de prueba que: garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo; ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los bucles en sus límites y con sus límites operacionales; y ejerciten las estructuras internas de datos para asegurar su validez.

La prueba de **caja negra** se centra en los requisitos funcionales del *software*, o sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

Para probar el Sistema de Portadores Energéticos después de implementado se utilizó el método de **caja negra** y en base a ello se diseñaron **casos de prueba** que se enfocan en la funcionalidad del sistema (Pressman, 2002). A continuación se muestra uno de los casos de pruebas descrito en el artefacto Caso de prueba HU:

**Caso de prueba:** HU\_PE\_EE 11 Plan Anual.

### Escenario 1: Plan Anual

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
Si se selecciona a la opción Plan Anual del menú principal del módulo Energía Eléctrica.  Plan anual: 24959.4666667	Muestra al usuario el resultado obtenido de la estimación del Plan anual en un campo de texto:	Satisfactorio	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Campo de texto	Se hace necesario para el	Satisfactorio	

## *Capítulo 3. Construcción y validación de la propuesta de solución*

para mostrar el Plan Anual en blanco.	usuario que la tabla de consumo no este vacío y los datos de consumo este correctos, puesto que a partir de los datos existentes en ella se realiza la estimación para el plan anual.		
<b>Evaluación de la Prueba:</b>	<i>Satisfactoria</i>		

Tabla 6. Caso de prueba HU Plan anual.

Véanse el resto de los casos de prueba en el [Anexo 5. Casos de prueba.](#)

Una vez inmerso en la etapa de pruebas el Sistema de Portadores Energéticos es sometido a una primera iteración por parte del departamento de calidad del Centro de Informatización Universitaria donde se obtuvo como resultado que de treinta y tres casos de pruebas revisados se obtuvieron veintidós no conformidades siendo de estas, diez de implementación y doce de documentación. Una vez concluida esta iteración, se le realiza una segunda donde se revisaron un igual número de casos de pruebas y no se detectaron no conformidades. De esta forma el proceso de pruebas por parte de calidad de CENIA se da por concluido y el sistema es propuesto a ser pasado por el departamento de calidad a nivel de universidad (CALISOFT).

## Capítulo 3. Construcción y validación de la propuesta de solución

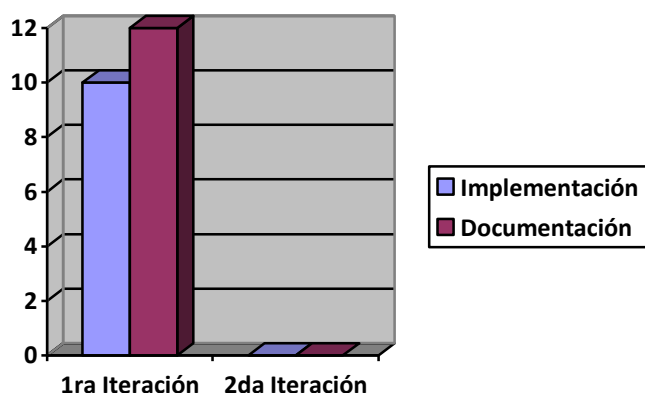


Figura 18. Resultados de las pruebas

Luego de haber pasado por calidad del CENIA el Sistema de Portadores Energéticos fue probado por CALISOFT. Dicho departamento realizó pruebas funcionales al sistema para evaluar las distintas funcionalidades del mismo. Para la primera iteración realizada: de treinta y nueve casos de prueba analizados se detectaron cuarenta y seis no conformidades, de ellas ocho correspondientes a la documentación y treinta y ocho a la implementación. Una vez concluida esta iteración, el sistema es sometido a una segunda en la cual solo fue detectada una no conformidad la cual fue definida como no procede dado que depende de eventos externos a la aplicación. De modo que el Sistema de Portadores Energéticos fue liberado por calidad y así consta en el acta emitida el 23 de noviembre de 2011 por el Centro de Calidad.

*Véase el acta de liberación por el Centro de Calidad (Calisoft) en la página VI.*

Además el Sistema de Portadores Energéticos fue registrado por el Centro Nacional de derecho de autor de Cuba (CENDA) con el número de registro: 297-2012.

*Véase el acta de registro en la página IV.*

### 3.11 Resultados esperados

- Se logró la incorporación de nuevas funcionalidades como el uso de un servicio web que permite acceder a los datos de consumo y plan de la universidad; la activación o desactivación de módulos como parte de las opciones administrativas del sistema, así como la exportación de los planes a archivos excel y pdf.
- Se elaboró un adecuado diseño de la base de datos que permite la incorporación de las funcionalidades, lo cual posibilita un correcto registro de la energía.



## *Capítulo 3. Construcción y validación de la propuesta de solución*

- Se implementó una gestión automática del consumo a través de scripts, así como la estimación de los planes anual, mensual y diario usando el método del olvido matemático para reducir el rango de error.
- Se incorporó el envío de los planes por correo, incluyendo no solo los correos de los responsables de los consumidores internos de la universidad, sino de todos los servicios que posee la universidad de manera general.
- Se implementó un sistema utilizando una versión beta de la arquitectura del marco de trabajo CodeIgniter en su versión 1.7.3. Esta arquitectura fue propuesta por el centro CENIA e incorpora nuevas funcionalidades que facilitan el desarrollo de aplicaciones web usando CodeIgniter.

### **3.12 Conclusiones**

Queda plasmado el desarrollo del módulo Energía Eléctrica, obteniéndose resultados favorables en las funcionalidades implementadas. Además, se especifica el uso de los estándares de codificación definidos por el Centro de Informatización Universitaria. La verificación del correcto funcionamiento de los procesos Planificación, Configuración y Gestión del consumo a través de los casos de pruebas diseñados como parte del proceso de pruebas realizado a la aplicación, permitió la detección de no conformidades y la corrección de las mismas.

## Conclusiones Generales

Con la culminación de este trabajo de diploma se arribó a las siguientes conclusiones:

- ❖ Luego de identificar las tecnologías, lenguajes y metodología de desarrollo de software a utilizar para el análisis, diseño y construcción del producto, se comprobó en la práctica la efectividad de las seleccionadas por el Centro de Informatización Universitaria.
- ❖ El estudio del estado del arte de otras soluciones similares en Cuba y el mundo, arrojó nuevas perspectivas y la medida del avance que existe en la actualidad con respecto al tema en cuestión. Esto permitió comprender que aunque las soluciones encontradas son muy avanzadas, la implantación del Sistema de Portadores Energéticos propone funcionalidades adecuadas a las necesidades de la universidad.
- ❖ La identificación de los procesos Planificación, Configuración y Gestión del consumo permitió conocer cómo se realizan los mismos en la actualidad y en base a ello proponer una mejora de los mismos en la actual solución.
- ❖ La identificación de los requisitos y la descripción de las historias de usuarios propuso un análisis detallado de las funcionalidades, de manera que el programador se familiarice con cada uno de los procesos involucrados.
- ❖ La implementación de cada una de las clases del sistema se realizó cumpliendo con los estándares de codificación definidos por el Centro de Informatización Universitaria, lo cual permitió mayor organización del código. Además, los algoritmos implementados resuelven los requerimientos identificados.
- ❖ Las pruebas de validación realizadas al sistema comprobaron la efectividad del mismo, en este caso, las funcionalidades asociadas a los procesos Planificación, Gestión del consumo y Configuración muestran al usuario cálculos y estimaciones que deben funcionar de manera correcta por su significación dentro del sistema.

El presente trabajo de diploma representa una primera versión de la herramienta **Sistema de Portadores Energéticos**, aplicación que gestiona de manera eficiente la energía en la universidad.

Se cubrieron las principales expectativas del cliente y dicha herramienta se encuentra actualmente en explotación por parte del personal de la Dirección de Energía.

## ***Recomendaciones***

Para el desarrollo de una nueva versión del Sistema de Portadores Energéticos, se recomienda:

- ❖ La integración del sistema con la arquitectura del nuevo marco de trabajo GUUD definido en el centro, que provee de una serie de funcionalidades y componentes que apoyan la organización de la implementación de las vistas del sistema.

## Glosario de términos

### A \_\_\_\_\_

**ASCII:** (acrónimo inglés de American Standard Code for Information Interchange — Código Estándar Estadounidense para el Intercambio de Información), es un código de caracteres basado en el alfabeto latino que utiliza 7 bits para representar los caracteres.

### C \_\_\_\_\_

**CIRCUTOR:** CIRCUTOR SA. Es una de las principales empresas europeas centradas en el diseño, fabricación y comercialización de equipos de Eficiencia Energética Eléctrica. CIRCUTOR ofrece toda una gama de software para esta problemática: PowerVision y PowerStudio.

**Código fuente:** El código fuente de un programa informático (o software) es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento.

**Comandos:** Es una instrucción u orden que el usuario proporciona a un sistema informático, desde la línea de comandos (como una *shell*) o desde una llamada de programación.

**Compilar:** Traducir un lenguaje de alto nivel a código absoluto o lenguaje binario.

**C#:** Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

**C++:** lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

### D \_\_\_\_\_

**Datawarehouse:** es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. Se trata, sobre todo, de un expediente completo de una organización almacenado en una base de datos diseñada para favorecer el análisis y la divulgación eficiente de datos.

**Debian:** es un sistema operativo libre, desarrollado por más de mil voluntarios alrededor del mundo, que colaboran a través de Internet.

**Delphi:** Delphi es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal: lenguaje de programación imperativo, diseñado entre 1967 y 1971 por Niklaus Wirth. Se trata de un lenguaje compilado y estructurado, que

simplifica su sintaxis a la vez que incluye nuevos tipos de datos y estructuras, como subrangos, tipos de datos enumerados, archivos, registros y conjuntos.

**Driver:** Un driver o controlador posibilita que el sistema operativo de una computadora pueda entenderse con un equipamiento periférico, como es el caso de una impresora, una placa de video, un mouse o un módem. Para poder interactuar con el periférico, el sistema operativo debe realizar una abstracción del hardware brindando una forma de manipularlo mediante una interfaz, esto es, algún mecanismo que permita controlar su funcionamiento, un conjunto de instrucciones que indican de qué manera debería comunicarse con tal o cual dispositivo.

## **F** \_\_\_\_\_

**Firefox:** es un navegador web libre y de código abierto descendiente de Mozilla Application Suite y desarrollado por la Fundación Mozilla.

**Formulario:** Un formulario HTML es una sección de un documento que contiene contenido normal, código, elementos especiales llamados controles (casillas de verificación (checkboxes), radiobotones (radio buttons), menús, etc.), y rótulos (labels) en esos controles. Los usuarios normalmente "completan" un formulario modificando sus controles (introduciendo texto, seleccionando objetos de un menú, etc.).

**FTP:** File Transfer Protocol usado en internet. Permite transferir archivos locales hacia un servidor web.

## **J** \_\_\_\_\_

**Java:** es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

## **L** \_\_\_\_\_

**Librería:** conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos.

## **M** \_\_\_\_\_

**MacOS:** del inglés Macintosh Operating System, en español Sistema Operativo de Macintosh, es el nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh. Es conocido por haber sido el primer sistema dirigido al gran público en contar con una interfaz gráfica compuesta por la interacción del mouse con ventanas, Icono y menús.

**Mandriva:** Mandrake o Mandriva Linux es una distribución Linux publicada por la compañía francesa Mandriva.

**Microsoft SQL Server:** es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

**MySQL:** es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

## N \_\_\_\_\_

**.NET:** Framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

## O \_\_\_\_\_

**Open Source (Código abierto):** es el término con el que se conoce al *software* distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

**C++:** es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

**Oracle:** es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

## P \_\_\_\_\_

**Parametrización:** En relación al uso de los recursos informáticos actuales surge el término parametrización como la organización de elementos (grupos y bloques) que permiten la modificación y la bifurcación del mismo en familias agrupando las similitudes y concentrando la información de un grupo en sus parámetros generando sistemas de información con conocimiento para elementos tridimensionales.

**Parser (analizador sintáctico):** Un analizador sintáctico es una de las partes de un compilador que transforma su entrada en un árbol de derivación.

**Plataforma:** una plataforma es el hardware sobre el cual puede ejecutarse o desarrollarse un software

**Python:** es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

## R \_\_\_\_\_

**Red Hat:** es una distribución Linux creada por Red Hat, que fue una de las más populares en los entornos de usuarios domésticos.

**RSS:** formato que permite emitir contenidos desde un sitio para que sean agregados fácilmente en aplicaciones o sitios web.

### **S**\_\_\_\_\_

**SCADA:** es un sistema basado en computadores que permite supervisar y controlar variables de proceso a distancia, proporcionando comunicación con los dispositivos de campo (controladores autónomos) y controlando el proceso de forma automática por medio de un software especializado. También provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros usuarios supervisores dentro de la empresa (supervisión, control calidad, control de producción, almacenamiento de datos, etc.).

**Script:** programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

### **U**\_\_\_\_\_

**UNIX:** Sistema operativo portable, multitarea y multiusuario.

### **V**\_\_\_\_\_

**Visual Basic:** Visual Basic es uno de los tantos lenguajes de programación que podemos encontrar hoy en día. Dicho lenguaje nace del BASIC (Beginner's All-purpose Symbolic Instruction Code) que fue creado en su versión original en el Dartmouth College, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación.

# Referencias Bibliográficas

## Referencias Bibliográficas

1. ELECTRICIDAD, MUSEO DE LA. *Museo de la Electricidad*. [En línea] <http://museoelectri.perucultural.org.pe/fuentes.htm>.
2. NICKSON, Christopher. *Digital Trends*. [En línea] 2 de June de 2009. <http://www.digitaltrends.com/features/ten-most-influential-tech-products/>.
3. REAL ACADEMIA ESPAÑOLA. Vigésima segunda edición. *Diccionario de la Lengua Española*. [En línea] [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=planificacion](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=planificacion).
4. INAREJOS, Pablo Álvarez Watkins; SÁNCHEZ, Juan José. *Planificación Energética y Desarrollo Sostenible*. [En línea] <http://www.encuentros-multidisciplinares.org/Revistan%C2%BA19/Pablo%20Alvarez%20Watkins%20y%20Juan%20Jos%C3%A9%20S%C3%A1nchez%20Inarejos.pdf>
5. DEFINICIÓN DE. [En línea] 2008. <http://definicion.de/gestion/>.
6. AGUADO BEHAR, Alberto. *Temas de Identificación y Control Adaptable*. 1ra edición. 2000. ISBN: 8420535702.
7. HERNÁNDEZ, Israel Gallardo. *SCEE VC v2.0 Manual de usuario*.
8. CIRCUTOR. *Software de gestión energética M.9*.
9. WORLDWIDE. EATON. *Powering Business*. [En línea] <http://powerquality.eaton.com/Spain/Products-services/Power-Management/Software-Drivers/ES-PowerVision.asp>
10. GAVAZZI, Carlo. *Carlo Gavazzi. Automation Components*. [En línea] <http://www.gavazzi-automation.com/es/group.asp>.
11. AG, INVISION SOFTWARE. *Forecast Pro. Predicciones en que su negocio puede confiar*. [En línea] 2005. [http://www.invisionwfm.com/es/productos/prevision\\_de\\_la\\_necesidad\\_de\\_forecastpro](http://www.invisionwfm.com/es/productos/prevision_de_la_necesidad_de_forecastpro).
12. CANÓS, José H.; LETELIER, Patricio; PENADÉS, Ma. Carmen. *Metodologías Ágiles en el Desarrollo de Software*. DSIC-Universidad Politécnica de Valencia: s.n.
13. PEÑALVER ROMERO, Gladys Marsi. *MA-GMPR-UR2. Metodología ágil para proyectos de software libre*. Ciudad de La Habana. Cuba: s.n., Junio de 2008.
14. FERRIOL ORTIZ, Acralys; AZAHARES REYES, Enmanuel. "Tesis de Análisis y Diseño del módulo Registro y Control Docente para Akademos v2.0". Universidad de las Ciencias Informáticas. La Habana. Junio 2009.
15. ARRAMBIDE, Paola Bruccoleri. *Software libre*. [En línea] [http://www.reducativa.com/webquests/licencias/software\\_libre.html](http://www.reducativa.com/webquests/licencias/software_libre.html)



# Referencias Bibliográficas

16. JMPEREDA'SWEBLOG. *¿Qué es una Aplicación Web?* [En línea]  
<http://jimpereda.wordpress.com/2007/08/24/definiendo-la-plantilla>.
17. MACROMEDIA, INC. *Primeros pasos con Dreamweaver*. [En línea]  
2003.<http://es.scribd.com/doc/54276618/55/Funcionamiento-de-una-aplicacion-Web>
18. ALEXANDER, Christopher; ISHIKAWA, Sara; SILVERSTEIN, Murray; JACOBSON, Max; FIKSDAHL-KING, Ingrid; ANGEL, Shlomo. *Towns/Building/Construction*. Oxford University Press :s.n. 1977.
19. BIZAGI. *BPMN Business Process Modeling Notation*. [En línea] <http://www.bizagi.com>
20. WHITE, Stephen A. *Business Process Modeling Notation (BPMN)*. IBM Corporation. May 3, 2004. Version 1.0.
21. POPKIN SOFTWARE AND SYSTEMS. *Modelado de Sistemas con UML*. [En línea]  
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>
22. BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. *El lenguaje unificado de modelado*. Madrid: Addison Wesley, 2000. ISBN: 84-7829-036-2
23. HISPANETWORK PUBLICIDAD Y SERVICIOS, S.L. *Glosario.net*. [En línea]  
<http://tecnologia.glosario.net/terminos-unicos/lenguaje-de-programaci%F3n-9768.html>.
24. MAESTROS DEL WEB. *Los diferentes lenguajes de programación para la web*. Pérez Valdés, Damián. [En línea] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>
25. PROYECTO EATS. *Programación en PHP y Bases de Datos*. [En línea] 2007.  
[http://www.iescamp.es/fileadmin/informatica/php\\_y\\_mysql.pdf](http://www.iescamp.es/fileadmin/informatica/php_y_mysql.pdf)
26. LIBROS WEB. *Capítulo 1. Introducción ¿Qué es CSS?* [En línea]  
<http://www.librosweb.es/css/capitulo1.html>
27. SOLÍS SILVA, Hugo Javier . *Tecnología e internet*. [En línea] 2008.  
<http://www.gestiopolis.com/administracion-estrategia/xml-alternativa-para-el-manejo-de-datos-moviles.htm>
28. SITIO DE DESCARGAS DE SOFTWARE. *Paradigma visual para UML (Plataforma Java) (Visual Paradigm for UML [Java Platform]) 6.0*. [En línea] Marzo, 2007.  
[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%5Bcuenta\\_de\\_Plataforma\\_de\\_Java\\_14715\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/)
29. AYASCA ESQUIVES, Alexander. *Evolus Pencil*. De todo. [En línea] 2010.  
<http://sudokill.blogspot.com/2010/10/evolus-pencil.html>.

## Referencias Bibliográficas

30. M3RM4. *Desarrollo de tecnologías para la web*. [En línea] <http://m3rm4.wordpress.com/>
31. ALEXANDER, Christopher; ISHIKAWA, Sara; SILVERSTEIN, Murray; JACOBSON, Max; FIKSDAHL-KING, Ingrid; ANGEL, Shlomo. *Towns/Building/Construction*. Oxford University Press : s.n. 1977.
32. ECURED. *Servidores de Bases de Datos*. [En línea] 2012. [http://www.ecured.cu/index.php/Servidores\\_de\\_Base\\_de\\_Datos](http://www.ecured.cu/index.php/Servidores_de_Base_de_Datos)
33. UPTODOWN. *PostgreSQL*. [En línea] 2009. <http://postgresql.uptodown.com/>.
34. BUENAS TAREAS. *Qué es un IDE de Programación*. [En línea] 2010. <http://www.buenastareas.com/ensayos/Que-Es-Un-Ide-De-Programacion/163537.html>
35. NETBEANS [En línea] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html)
36. RED DE APRENDIZAJE. *Plataforma informática*. [En línea] 2012. <http://www.reddeaprendizaje.com/inicio/item/47-plataforma-informatica>
37. ZABALLA COCA, Roilán. *“Personalización de distribuciones basadas en la familia SUSE Linux.”*. 2009.
38. SPARX SYSTEMS. *Modelando procesos de negocios*. [En línea] [http://www.sparxsystems.com.ar/resources/tutorial/business\\_process\\_model.html](http://www.sparxsystems.com.ar/resources/tutorial/business_process_model.html)
39. LETELIER, Patricio; PENADÉS, Ma. Carmen. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia. Camino de Vera s/n, 46022.
40. GARCÍA VIDAL, Yanio. *Arquitectura de Software metodología SXP*. Centro de Informatización Universitaria. 2011.
41. *Diseño y Programación Orientado a Objetos Ingeniería Informática*. Ingeniería Técnica de Informática de Sistemas y Gestión Optativa (6 créditos). Capítulo 4 Patrones y Patrones de Diseño. Curso 2003-2004. [En línea] <http://www.info-ab.uclm.es/asignaturas/42579>
42. FOWLER, Martin, "UML Distilled", Tercera Edición, EUA, Editorial Addison-Wesley, 2003, Cap. 8.
43. PRUEBAS DE SOFTWARE. *La prueba de aceptación es la prueba más importante para los productos software*. [En línea] 2010. <http://pruebasdesoftware.com/pruebadeaceptacion.htm>

## ***Bibliografía Consultada***

- EXPÓSITO, Ery Delgado. *Metodologías de desarrollo de software. ¿Cuál es el camino?* Monografías.com. [En línea] <http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml>
- ROSSEL DR., Gerardo. *Programación lógica*. [En línea] [http://www.amzi.com/articles/code07\\_whitepaper.pdf](http://www.amzi.com/articles/code07_whitepaper.pdf)
- QUINTERO, Antonia M<sup>a</sup> Reina. [En línea] Diciembre, 2000. <http://www.willydev.net/descargas/abcpoa.pdf>
- GONZÁLEZ BRAMBILA DR., Silvia Beatriz. *Introducción al diseño y a la programación orientada a objetos*. [En línea] <http://delfosis.uam.mx/~sgb/docs/INTRO.pdf>. .
- LENGUAJES DE PROGRAMACIÓN. *Programación orientada a objetos* [En línea] 2009. <http://www.lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>.
- ELLISLAB INC. *Manual de CodeIgniter*. Medellín, Colombia. : s.n.
- MAESTROS DEL WEB. *Los frameworks de php agilizan tu trabajo* [En línea] <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
- Ma. GONZALEZ CASTELLANOS, Argeni; ROJAS PABON, Wilson. “Trabajo De Grado: Comparación entre sistemas de gestión de bases de datos (SGBD)”. Bogotá: Universidad Católica de Colombia, Facultad de Ingeniería de Sistemas, 2005: s.n.
- MAKAROV, Alexander . *The Big PHP IDE Test: Why Use One And Which To Choose*. Smashing Magazine [En línea] <http://www.smashingmagazine.com/2009/02/11/the-big-php-ides-test-why-use-oneand-which-to-choose>
- SOTOMAYOR BASILIO, Borja. *La plataforma .NET: ¿El futuro de la Web?* [En línea] <http://people.cs.uchicago.edu/~borja/pubs/revistaeside2002.pdf>
- SITIO DE DESCARGAS DE SOFTWARE. *Paradigma visual para UML (Plataforma Java) (Visual Paradigm for UML [Java Platform]) 6.0* [En línea] Marzo, 2007. [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%5Bcuenta\\_de\\_Plataforma\\_de\\_Java\\_14715\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/)
- WHITE, Stephen A.; MIERS, Derek. *Guía de Referencia y Modelado BPMN*. Marzo, USA: Future Strategies Inc., 2009. ISBN: 978-0-9819870-3-3.
- PRESSMAN, Roger S. *Ingeniería del Software*. Quinta edición. Madrid: Mc Graw Hill. 2002.

**Anexos**

**Anexo 1. Tareas de ingeniería**

Tabla 7. Tarea de ingeniería 1

<b>Tarea de ingeniería</b>	
<b>Número tarea: 1</b>	
<b>Nombre tarea:</b> Estudiar el marco de trabajo CodeIgniter	
<b>Tipo de tarea:</b> Estudio	<b>Puntos estimados: 1</b>
<b>Fecha inicio:</b> 3/09/2010	<b>Fecha fin:</b> 10/09/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Estudiar el funcionamiento del marco de trabajo CodeIgniter, en especial la aplicación del patrón de arquitectura Modelo-Vista-Controlador.	
<b>Tarea de ingeniería</b>	
<b>Número tarea: 4</b>	<b>Número de la Historia de Usuario: 7</b>
<b>Nombre tarea:</b> Implementar script para consumidores normales.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados: 0.5</b>
<b>Fecha inicio:</b> 19/01/2011	<b>Fecha fin:</b> 22/01/2011
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar un script que tome las lecturas de cada uno de los consumidores normales y calcule el consumo para mostrarlo en reportes posteriores.	

Tabla 8. Tarea de ingeniería 4

<b>Tarea de ingeniería</b>	
<b>Número tarea: 5</b>	<b>Número de la Historia de Usuario: 8</b>
<b>Nombre tarea:</b> Implementar script para consumidores automáticos.	

<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 23/01/2011	<b>Fecha fin:</b> 26/01/2011
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar un script que tome las lecturas de cada uno de los consumidores automáticos y calcule el consumo para mostrarlo en reportes posteriores.	

Tabla 9. Tarea de ingeniería 5

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 6	<b>Número de la Historia de Usuario:</b> 9
<b>Nombre tarea:</b> Implementar script para consumidores no medrados.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 27/01/2011	<b>Fecha fin:</b> 30/01/2011
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar un script que tome las lecturas de cada uno de los consumidores no medrados y calcule el consumo para mostrarlo en reportes posteriores.	

Tabla 10. Tarea de ingeniería 6

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 7	<b>Número de la Historia de Usuario:</b> 10
<b>Nombre tarea:</b> Implementar servicio web	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 02/03/2011	<b>Fecha fin:</b> 09/03/2011
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar un servicio web que muestre los datos de consumo ofrecidos por la	

aplicación.

Tabla 11. Tarea de ingeniería 7

Tarea de ingeniería	
<b>Número tarea: 8</b>	<b>Número de la Historia de Usuario: 11</b>
<b>Nombre tarea:</b> Implementar conexión con el SCADA	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 10/03/2011	<b>Fecha fin:</b> 17/03/2011
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar los métodos para la conexión a la base de datos del sistema SCADA para la toma de las lecturas.	

Tabla 12. Tarea de ingeniería 8

Tarea de ingeniería	
<b>Número tarea: 11</b>	<b>Número de la Historia de Usuario: 12</b>
<b>Nombre tarea:</b> Desagregar plan anual por meses y por servicios.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 06/11/2010	<b>Fecha fin:</b> 13/11/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Calcular el peso que representa cada mes del total anual y el peso que representa cada servicio del total mensual; para luego guardar esos datos y mostrarlos en una tabla, además la vista debe permitir modificar los consumos.	

Tabla 13. Tarea de ingeniería 11

Tarea de ingeniería
---------------------

<b>Número tarea: 12</b>	<b>Número de la Historia de Usuario: 13</b>
<b>Nombre tarea:</b> Implementar Plan mensual de los servicios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 14/11/2010	<b>Fecha fin:</b> 17/11/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Buscar en la base de datos los datos de consumo mensuales para los servicios y mostrar en una tabla los mismos.	

Tabla 14. Tarea de ingeniería 12

<b>Tarea de ingeniería</b>	
<b>Número tarea: 13</b>	<b>Número de la Historia de Usuario: 14</b>
<b>Nombre tarea:</b> Implementar Plan mensual de los servicios UCI	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 18/11/2010	<b>Fecha fin:</b> 25/11/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar una vista que muestre en una tabla el consumo mensual de los servicios UCI y perita desagregar el mismo de manera diaria entre los consumidores internos.	

Tabla 15. Tarea de ingeniería 13

<b>Tarea de ingeniería</b>	
<b>Número tarea: 14</b>	<b>Número de la Historia de Usuario: 15</b>
<b>Nombre tarea:</b> Implementar Plan diario de los consumidores internos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1

<b>Fecha inicio:</b> 26/11/2010	<b>Fecha fin:</b> 03/12/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar una vista que muestre en una tabla el consumo diario de los consumidores internos, para ello se debe calcular el peso que representa cada día del total mensual y cada consumidor del total diario, además debe permitir modificar los consumos de los consumidores.	

Tabla 16. Tarea de ingeniería 14

Tarea de ingeniería	
<b>Número tarea:</b> 15	<b>Número de la Historia de Usuario:</b> 16
<b>Nombre tarea:</b> Implementar Enviar correos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 01/04/2011	<b>Fecha fin:</b> 08/04/2011
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar el envío de los planes, adicionar además en las vistas los botones para ejecutar esta acción.	

Tabla 17. Tarea de ingeniería 15

Tarea de ingeniería	
<b>Número tarea:</b> 16	<b>Número de la Historia de Usuario:</b> 1
<b>Nombre tarea:</b> Implementar Mostrar correos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 7/10/2010	<b>Fecha fin:</b> 10/10/2010
<b>Programador responsable:</b> Noél Hernández Bautista	



<b>Descripción:</b> Buscar en la base de datos los correos que guardados y mostrarlos en una vista en forma de listado
------------------------------------------------------------------------------------------------------------------------

Tabla 18.Tarea de ingeniería 16

Tarea de ingeniería	
<b>Número tarea: 17</b>	<b>Número de la Historia de Usuario: 2</b>
<b>Nombre tarea:</b> Implementar Adicionar correo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 11/10/2010	<b>Fecha fin:</b> 14/10/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Crear un formulario para la entrada de los datos del correo a adicionar, guardar los datos introducidos.	

Tabla 19.Tarea de ingeniería 17

Tarea de ingeniería	
<b>Número tarea: 18</b>	<b>Número de la Historia de Usuario: 4</b>
<b>Nombre tarea:</b> Implementar Eliminar correo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 21/10/2010	<b>Fecha fin:</b> 23/10/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar la acción de un botón que elimine en el listado de los correos alguno en específico a través de una consulta a la base de datos.	

Tabla 20. Tarea de ingeniería 18

Tarea de ingeniería	
<b>Número tarea: 19</b>	<b>Número de la Historia de Usuario: 5</b>
<b>Nombre tarea:</b> Implementar Habilitar/Deshabilitar módulos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 20/09/2010	<b>Fecha fin:</b> 27/09/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar una vista en la que a través de la acción de un select se active o desactive un módulo en específico.	

Tabla 21. Tarea de ingeniería 19

Tarea de ingeniería	
<b>Número tarea: 20</b>	<b>Número de la Historia de Usuario: 5</b>
<b>Nombre tarea:</b> Implementar Definir correo del responsable del módulo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 21/09/2010	<b>Fecha fin:</b> 23/09/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Crear en la misma vista del Habilitar/Deshabilitar módulos los campos de texto necesarios para la entradas de los correos de los responsables de los módulos, crear las consultas a la base de datos para la inserción de los mismos.	

Tabla 22. Tarea de ingeniería 20

Tarea de ingeniería	
<b>Número tarea: 21</b>	<b>Número de la Historia de Usuario: 5</b>

<b>Nombre tarea:</b> Implementar definir lista de correos a enviar información.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 21/09/2010	<b>Fecha fin:</b> 23/09/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Crear los campos de texto necesarios para la entradas de los correos de los responsables de los módulos, crear las consultas a la base de datos para la inserción de los mismos.	

Tabla 23. Tarea de ingeniería 21

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 3	<b>Número de la Historia de Usuario:</b> 6
<b>Nombre tarea:</b> Implementar script para consumidores mensuales.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 15/01/2011	<b>Fecha fin:</b> 18/01/2011
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar un script que tome las lecturas de cada uno de los consumidores mensuales y calcule el consumo para mostrarlo en reportes posteriores.	

Tabla 24. Tarea de ingeniería 3

<b>Tarea de ingeniería</b>	
<b>Número tarea:</b> 10	<b>Número de la Historia de Usuario:</b> 12
<b>Nombre tarea:</b> Estimar plan anual	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5

<b>Fecha inicio:</b> 01/11/2010	<b>Fecha fin:</b> 05/11/2010
<b>Programador responsable:</b> Noél Hernández Bautista	
<b>Descripción:</b> Implementar la estimación del plan anual tomando como referencia los consumos históricos guardados.	

Tabla 25. Tarea de ingeniería 10

**Anexo2. Descripción de las entidades de la base de datos.**

<b>Nombre de la entidad</b>	Configuración (tb_dconfiguracion)		
<b>Descripción de la entidad</b>	Se guarda la configuración de los módulos con su responsable y si es activo o no.		
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Puede ser Nulo</b>
id_configuracion	Identifica la configuración	Entero	No
correo_responsable	Identifica un correo para el responsable del modulo	Cadena de caracteres (varchar)	Si
es_activo	Identifica si el modulo está activo o no.	Cadena de caracteres (varchar)	No
id_modulo (sin referencia)	Identifica el modulo	Entero	No

Tabla 26. Entidad Configuración

<b>Nombre de la entidad</b>	Consumidor (tb_dconsumidor)		
<b>Descripción de la entidad</b>	Consumidores. Entidades que consumen energía ya sea eléctrica, agua u otra.		
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Puede ser Nulo</b>

id_consumidor	Identifica al consumidor	Entero	No
tipo_consumidor	Identifica al tipo consumidor	Entero	No
nombre_consumidor	Describe el nombre del consumidor	Cadena de caracteres (varchar)	No
descripcion	Describe otros detalles sobre el consumidor	Cadena de caracteres (varchar)	Si
id_modulo (sin referencia)	Identifica al modulo	Entero	No

Tabla 27. Entidad Consumidor

<b>Nombre de la entidad</b>	Consumo (tb_dconsumo)		
<b>Descripción de la entidad</b>			
<b>Nombre del atributo</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Puede ser Nulo</b>
id_consumo	Identifica el consumo	Entero	No
id_consumidor	Identifica al consumidor	Entero	No
consumo_real	Consumo real	Número real	No
fecha_consumo	Identifica al auditor	Entero	No
descripcion	Describe otros detalles del consumo guardado en esa fecha.	Cadena de caracteres (varchar)	No

Tabla 28. Entidad Consumo

<b>Nombre de la entidad</b>	Correos información (tb_dcorreos_informacion)
<b>Descripción de la entidad</b>	Describe un listado de correos a quien enviarle ciertas informaciones que se definen en el negocio del sistema.

Nombre del atributo	Descripción	Tipo	Puede ser Nulo
id_correos_informacion	Identifica al correo	Entero	No
correos_informacion	correo	Cadena de caracteres (varchar)	No

Tabla 29. Entidad Correos

<b>Nombre de la entidad</b>	Scada 01 (scada_01)		
<b>Descripción de la entidad</b>			
Nombre del atributo	Descripción	Tipo	Puede ser Nulo
id_scada	Identifica el rol	Entero	No
id_consumidor	Identifica al usuario	Entero	No
lectura_activa	Lectura activa determinada en la lectura	Real	Si
lectura_reactiva	Lectura reactiva determinada en la lectura	Real	No
factor_potencia	Factor de potencia determinado en la lectura	Real	Si
fecha_lectura	Fecha de la lectura	Time	No
calidad_lectura	Calidad de la lectura	Entero	No

Tabla 30. Entidad Escada\_01

### **Anexo 3. Historias de usuario**

<b>Historia de Usuario</b>
----------------------------

<b>Código:</b> HU_PE_EE_39	<b>Nombre Historia de Usuario:</b> Calcular consumo de los consumidores mensuales
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 26. Insertar lectura mensual para consumidores mensuales.	
<b>Programador:</b> Noel Hernández Bautista	<b>Iteración Asignada:</b> 2
<b>Prioridad :</b> Muy Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Esta historia de usuario realiza el cálculo del consumo de los consumidores mensuales a partir de las lecturas diarias. Para realizar dicho cálculo se tomarían las lecturas del primer y último día del mes y se realizaría la resta.	
<b>Observaciones:</b> Consumidor mensual: consumidor cuya lectura se realiza una vez al mes.	
<b>Prototipo de interfaces:</b> Esta funcionalidad es invisible para el usuario.	

Tabla 31. Calcular consumo de los consumidores mensuales.

Historia de Usuario	
<b>Código:</b> HU_PE_EE_40	<b>Nombre Historia de Usuario:</b> Calcular consumo de los consumidores normales
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 31. Insertar lectura diaria para consumidores normales.	
<b>Programador:</b> Noel Hernández Bautista	<b>Iteración Asignada:</b> 2

<b>Prioridad :</b> Muy Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1
<p><b>Descripción:</b> Esta historia de usuario realiza el cálculo del consumo de los consumidores normales (diarios) a partir de las lecturas diarias.</p> <p>Esta funcionalidad para el caso de los consumidores normales, debe permitir tomar de la base de datos las lecturas de los últimos dos días y establecer una validación entre las fechas de las mismas:</p> <ul style="list-style-type: none"> <li>• En el caso que exista diferencia de un día entre ambas lecturas se restan.</li> <li>• En caso que exista más de un día se resta el último con el primero y se divide entre la cantidad de días de diferencia, asignándole en la base de datos un consumo aproximado a cada uno de los días correspondientes entre lectura y lectura.</li> </ul>	
<p><b>Observaciones:</b> Consumidor normal (diario): consumidor cuya lectura se realiza manual.</p>	
<p><b>Prototipo de interfaces:</b> Esta funcionalidad es invisible para el usuario.</p>	

Tabla 32. Calcular consumo de los consumidores normales.

<b>Historia de Usuario</b>	
<b>Código:</b> HU_PE_EE_41	<b>Nombre Historia de Usuario:</b> Calcular consumo de los consumidores automáticos.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 38. Leer y guardar lecturas para consumidores automáticos.	
<b>Programador:</b> Noel Hernández Bautista	<b>Iteración Asignada:</b> 2
<b>Prioridad :</b> Muy Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1



<p><b>Descripción:</b> Esta historia de usuario realiza el cálculo del consumo de los consumidores automáticos a partir de las lecturas diarias.</p> <p>Esta funcionalidad, debe permitir tomar de la base de datos las lecturas de los últimos dos días y establecer una validación entre las fechas de las mismas:</p> <ul style="list-style-type: none"> <li>• En el caso que exista diferencia de un día entre ambas lecturas se restan.</li> <li>• En caso que exista más de un día se resta el último con el primero y se divide entre la cantidad de días de diferencia, asignándole en la base de datos un consumo aproximado a cada uno de los días correspondientes entre lectura y lectura.</li> </ul>
<p><b>Observaciones:</b> Consumidor automático: consumidor cuya lectura es recogida por el SCADA.</p>
<p><b>Prototipo de interfaces:</b> Esta funcionalidad es invisible para el usuario.</p>

Tabla 33. Calcular consumo de los consumidores automáticos.

<b>Historia de Usuario</b>	
<b>Código:</b> HU_PE_EE_42	<b>Nombre Historia de Usuario:</b> Calcular consumo de los consumidores no metrados.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 35. Registrar lecturas generales	
<b>Programador:</b> Noel Hernández Bautista	<b>Iteración Asignada:</b> 2
<b>Prioridad :</b> Muy Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1
<p><b>Descripción:</b> Esta historia de usuario realiza el cálculo del consumo de los consumidores no metrados, para ello se hallaría la diferencia entre el consumo total mensual de las dos líneas principales y el consumo total de los consumidores diarios.</p>	
<p><b>Observaciones:</b> Consumidor no metrados: consumidor que no posee metro-contador.</p>	

<i>Esta funcionalidad es invisible para el usuario.</i>
<b>Prototipo de interfaces:</b> Invisible para el usuario

Tabla 34. Calcular consumo de los consumidores no medrados.

Historia de Usuario	
<b>Código:</b> HU_PE_EE 43	<b>Nombre Historia de Usuario:</b> Servicio web para mostrar los datos en la Intranet.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b>	
<b>Programador:</b> Noél Hernández Bautista	<b>Iteración Asignada:</b> 2
<b>Prioridad :</b> Muy alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Esta historia de usuario permite mostrar a través de la Intranet de la universidad los datos que ofrece la aplicación. En este caso el servicio web brinda datos del sistema como pueden ser el plan diario de consumo eléctrico para la universidad y el consumo eléctrico real de la misma.	
<b>Observaciones:</b> Cualquier usuario que interactúe con la Intranet de la universidad podrá acceder a este servicio.	
<b>Prototipo de interfaz:</b> Esta funcionalidad es invisible para el usuario	

Tabla 35. Servicio web para mostrar los datos en la Intranet.

Historia de Usuario	
<b>Código:</b> HU_PE_EE 16	<b>Nombre Historia de Usuario:</b> Plan Mensual
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 13, 14,1 5,16 ,17.	
<b>Programador:</b> Noél Hernández Bautista	<b>Iteración Asignada:</b>


<b>Prioridad</b> : Muy Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b>
<p><b>Descripción:</b> (Pseudocódigo)</p> <ul style="list-style-type: none"> <li>-Promediar los consumos históricos de los meses anteriores, utilizar además Teorema del olvido matemático para minimizar el margen de error. <i>Tomar datos de tabla Consumo.</i></li> <li>-Calcular el peso que representa cada servicio del total mensual para todos los servicios.</li> <li>-Al realizar la desagregación se guardan los datos en la base de datos como plan real. <i>Tabla Plan.</i></li> <li>- Mostrar una tabla con la Desagregación de todos los meses por servicios. La tabla seria la misma que la de la desagregación del plan anual pero como en la vista del plan anual se modifica la demanda estimada teniendo en cuenta el plan asignado, en esta tabla se mostraría los datos reales asignados ya desagregados.</li> <li>-Si el usuario da clic en la opción Exportar de un mes específico debe darse la opción de que lo haga en excel y en pdf. (<i>Se exportará la dupla correspondiente al mes que está seleccionando</i>).</li> <li>- Si el usuario da clic en la opción Cancelar se cierra la vista y se muestra la vista principal.</li> <li>-Al dar clic en la opción Aceptar se sale de la vista y se muestra mensaje “Los datos han sido guardados satisfactoriamente”. <i>Guardar datos en la tabla Plan como mensual.</i></li> </ul>	
<p><b>Observaciones:</b> Interactúa con esta acción el especialista de planificación.</p>	
<p>Prototipo de interfaz:</p> <div style="text-align: center;">  </div>	

Tabla 36. Plan Mensual.

Historia de Usuario	
<b>Código:</b> HU_PE_EE 20	<b>Nombre Historia de Usuario:</b> Plan de los servicios UCI.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 16, 17, 18, 19, 20, 23, 24.	
<b>Programador:</b> Noél Hernández Bautista	<b>Iteración Asignada:</b>
<b>Prioridad :</b> Muy Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b>
<p><b>Descripción:</b> <i>(Pseudocódigo)</i></p> <ul style="list-style-type: none"> <li>- El sistema muestra la tabla de la desagregación por meses de los servicios UCI. <i>Tomar datos de la tabla Plan de todos los meses, pero solo de los servicios UCI.</i></li> <li>-Promediar los consumos históricos de meses en años anteriores, tomando como referencia el día de la semana. <i>Los días de la semana de cada mes agrupados reducirían el rango de error dado que los mismos tienen características comunes.</i></li> <li>- Calcular el peso que representa cada servicio UCI del total mensual de los servicios.</li> <li>-Si el usuario da clic en la opción Desagregar de un mes específico se le mostrará una tabla con la desagregación diaria de dicho mes para los consumidores internos teniendo en cuenta el total entre los dos servicios UCI (15900 y 1600).</li> <li>-Si se modifica el valor de un día específico, la diferencia se debe redistribuir equitativamente a partir del día en que se modificó para ese consumidor.</li> <li>-Si el usuario desea exportar la desagregación por días del mes y por consumidores, debe dar clic en el botón Exportar.</li> <li>- Si el usuario da clic en la opción Cancelar se cierra la vista y se muestra la vista principal.</li> <li>-Al dar clic en la opción Aceptar se sale de la vista y se muestra mensaje "Los datos han sido guardados satisfactoriamente". <i>Guardar datos en la tabla Plan como diario.</i></li> </ul>	
<b>Observaciones:</b> Interactúa con esta acción el especialista de planificación.	
<b>Prototipo de interfaz:</b>	

Plan de los servicios UCI				
Meses	Circuito_1590	Circuito_1600	Total	Acciones
Enero...				D
Febrero...				D
Diciembre				D

EE EP

Aceptar Cancelar

Tabla 37. Plan de los servicios UCI.

Historia de Usuario	
<b>Código:</b> HU_PE_EE 24	<b>Nombre Historia de Usuario:</b> Plan Diario de los consumidores internos.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 16, 17, 22, 23, 24.	
<b>Programador:</b> Noél Hernández Bautista	<b>Iteración Asignada:</b>
<b>Prioridad :</b> Muy Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b>
<p><b>Descripción:</b> <i>(Pseudocódigo)</i></p> <ul style="list-style-type: none"> <li>-El sistema muestra la tabla de la desagregación diaria de los servicios UCI, o sea, el plan diario de los consumidores internos. <i>Tomar datos de la tabla Plan de todos los meses, pero solo de los servicios UCI.</i></li> <li>-Promediar el plan teniendo en cuenta los consumos históricos de meses en años anteriores, tomando como referencia el día de la semana.</li> <li>-Calcular el peso que representa cada servicio UCI del total mensual de los servicios.</li> <li>-Si se modifica el valor de un día específico, la diferencia se debe redistribuir equitativamente a partir del día en que se modificó para ese consumidor.</li> <li>- Si el usuario da clic en la opción Cancelar se cierra la vista y se muestra la vista principal del módulo.</li> <li>-Al dar clic en la opción Aceptar se sale de la vista y se muestra mensaje “Los datos han sido</li> </ul>	

guardados satisfactoriamente”. *Guardar datos en la tabla Plan como diario.*  
 -Si el usuario desea exportar la desagregación por días del mes y por consumidores, debe dar clic en el botón Exportar.

**Observaciones:** Interactúa con esta acción el especialista de planificación.

**Prototipo de interfaz:**

**Plan diario de los consumidores internos**

C EE EP

Días	Circuito_1590	Circuito_1600	Total	Cosumidores...
Dom				
Lun...				
Sab				

Tabla 38. Plan Diario de los consumidores internos.

Historia de Usuario	
<b>Código:</b> HU_PE_EE 44	<b>Nombre Historia de Usuario:</b> Enviar correo a interesados
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Referencia:</b> RF PE 64.	
<b>Programador:</b> Noél Hernández Bautista	<b>Iteración Asignada:</b>
<b>Prioridad :</b> Muy Alta	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b>
<p><b>Descripción:</b> <i>(Pseudocódigo)</i></p> <p>-En el caso del plan anual el sistema debe enviar automáticamente un correo al responsable de planificación y a un grupo de entidades involucradas con la desagregación por meses y por servicios. <i>Tomar datos de tabla Responsable (Estos servicios deben figurar juntos como un responsable, no son</i></p>	

<p>consumidores).</p> <p>Asunto: Desagregación del plan operativo anual.</p> <p>-En el caso del plan mensual se le envía un correo todos los responsables de las entidades con la desagregación del mes por servicios. <i>Tomar datos de tabla Responsable. Sacar tupla de la desagregación anual por servicios del mes correspondiente.</i></p> <p>Asunto: Desagregación del plan mensual operativo.</p> <p>-En el caso del plan diario de los consumidores se le envía un correo todos los responsables de las entidades con la desagregación del mes por día para todos los consumidores. <i>Tomar datos de tabla Responsable</i></p> <p>Asunto: Desagregación del plan diario operativo.</p>
<b>Observaciones:</b>
<b>Prototipo de interfaz:</b>

Tabla 39. Enviar correo a interesados.

**Anexo 4. Definición de procesos**

<b>Ficha de Proceso</b>	
Proceso:	Gestión del consumo
Entradas:	XML brindado por el SCADA. Lecturas registradas.
Salidas:	Consumo energético diario para cada tipo de consumidor.
Involucrados:	Administrador del sistema.
<b>Descripción</b>	
<i>Subproceso Leer y guardar lecturas para consumidores automáticos</i>	
Parsear xml del SCADA.	El sistema hace corresponder el id del consumo recogido en el xml con su correspondiente consumidor registrado en la base de datos.
Guardar datos en la	Después del parseo el sistema guarda los datos de consumo registrados.

base de datos del sistema.	
<i>Subproceso Calcular consumo</i>	
El sistema en un momento determinado puede ejecutar cualquiera de los siguientes subprocesos:	
<i>Script para consumidores automáticos y diarios.</i>	
Tomar lecturas de los últimos 2 días.	El sistema toma de la base de datos las lecturas de los últimos dos días y establece una validación entre las fechas de las mismas
Si existe diferencia de 1 día o no, entre ambas fechas se realizan las siguientes tareas:	
Restar lecturas	El sistema resta la lectura del último día y la del anterior, determinando el consumo.
Se restan ambas lecturas y se dividen entre la cantidad de días de diferencia.	El sistema realiza el cálculo del consumo en este caso asignando consumo de manera equitativa a cada uno de los días, para ello resta primero las dos lecturas registradas y ese total lo divide entre la cantidad de días de diferencia.
<i>Script para consumidores mensuales.</i>	
Restar lecturas del primer y último día del mes.	El sistema realiza el cálculo de consumo anteriormente descrito para consumidores diarios y automáticos.
Dividir entre la cantidad de días.	El valor hallado anteriormente es dividido automáticamente entre la cantidad de días del mes.
<i>Script para consumidores no medrados.</i>	
Calcular diferencia entre el consumo total diario de los consumidores UCI (líneas principales) y el consumo total de los consumidores internos.	El sistema determina la diferencia entre el total diario calculado para los servicios UCI y el total diario para todos los consumidores.
Después de realizado cualquiera de estos sub-procesos termina el sub-proceso Calcular consumo guardándose los datos:	



Guardar datos de consumo.	El sistema guarda los datos de consumo calculados mediante los scripts.
<i>Grupo Servicio web</i>	
Mostrar datos de consumo.	El sistema usa la conexión entre plataformas a través de este servicio web para mostrar datos como el consumo energético y plan energético.

**Flujo-grama**

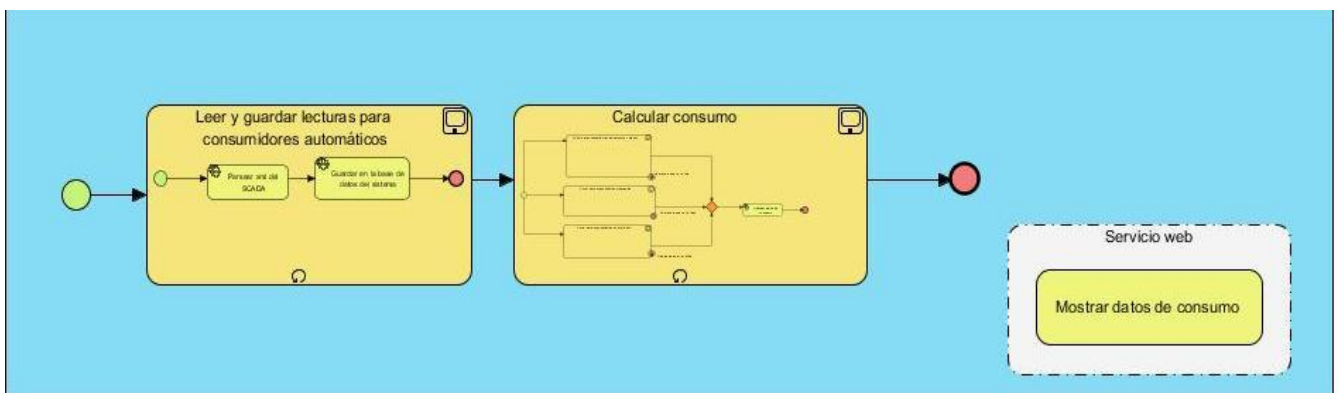


Figura 6. Gestión del consumo.



Figura 7. Sub-proceso Leer y guardar lecturas para consumidores automáticos.

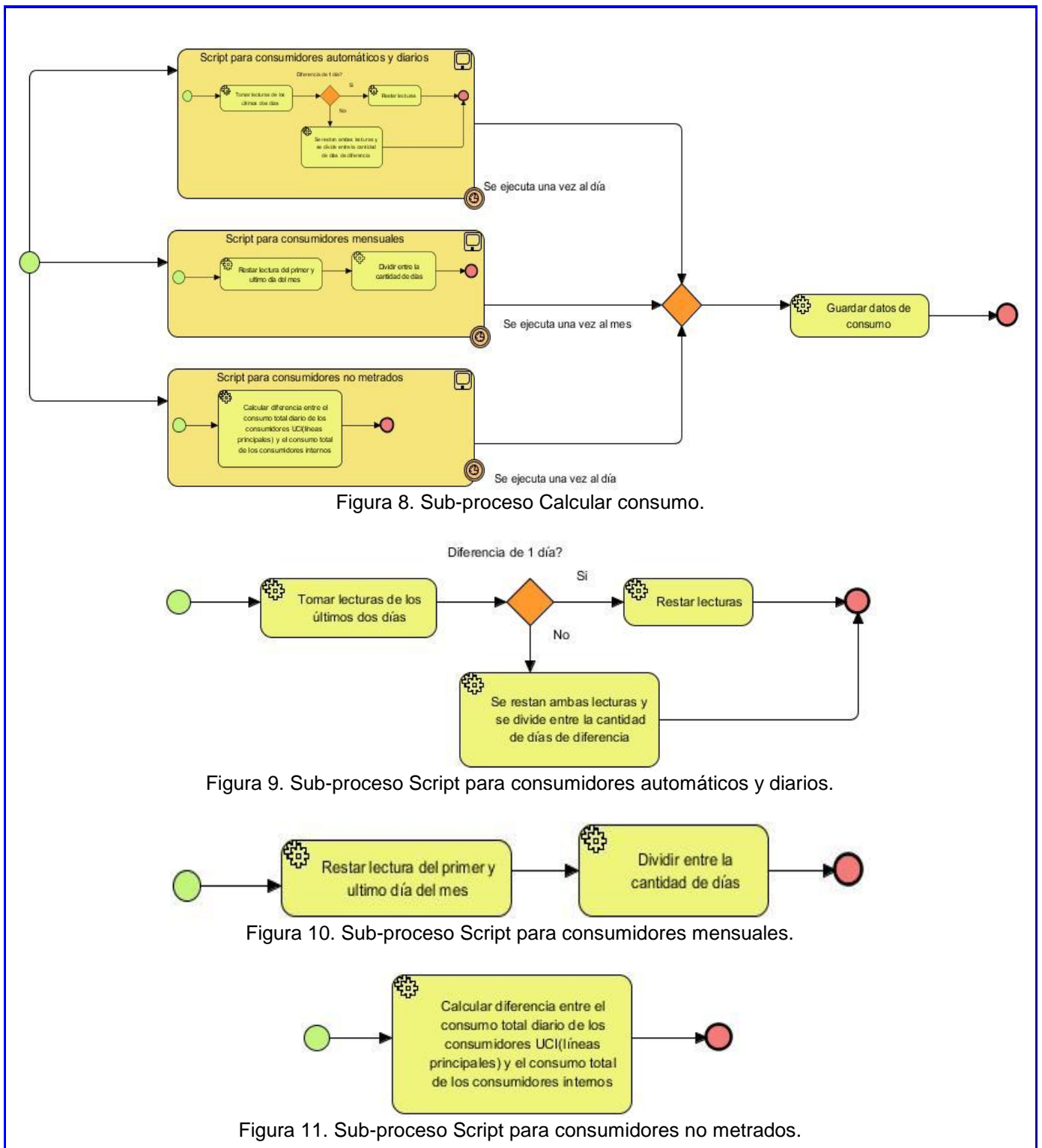


Figura 8. Sub-proceso Calcular consumo.

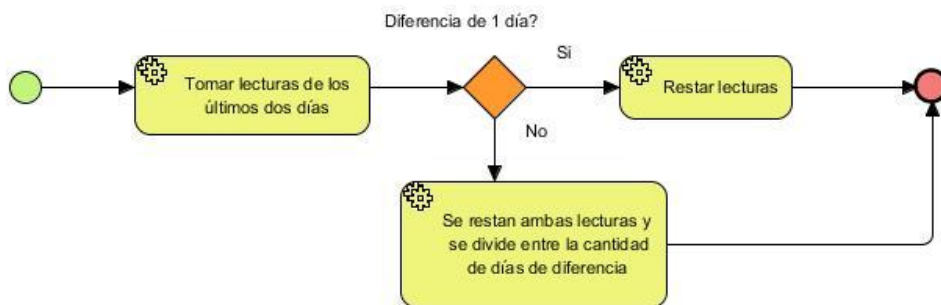


Figura 9. Sub-proceso Script para consumidores automáticos y diarios.

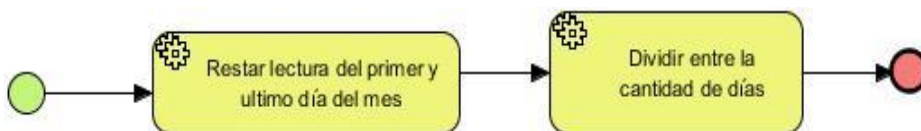


Figura 10. Sub-proceso Script para consumidores mensuales.

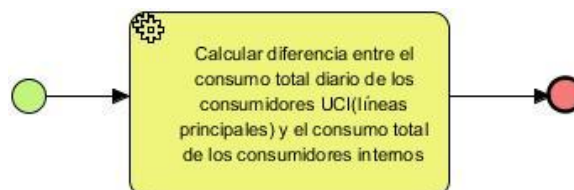


Figura 11. Sub-proceso Script para consumidores no medrados.

Tabla 40. Proceso Gestión de consumo.

<b>Ficha de Proceso</b>	
Proceso:	Configuración
Entradas:	Correos registrados en la base de datos.  Tablas de la base de datos Módulo-Configuración asociadas.
Salidas:	Correos registrados en la base de datos.  Módulos activados o desactivados.
Involucrados:	Administrador del sistema
<b>Descripción</b>	
<i>Subproceso Habilitar/Deshabilitar módulos</i>	
El usuario tiene la opción de realizar una o ambas actividades. Si escoge Marcar/Desmarcar módulo:	
Marcar/Desmarcar módulo.	Si el usuario selecciona o no algún módulo estará activando la desactivación/activación de módulos.
Habilitar/Deshabilitar módulo.	El sistema automáticamente responde a la acción del usuario cual sea el caso para el módulo correspondiente.
Si desea entrar un nuevo correo:	
Entrar nuevo correo de responsable.	El usuario entra un nuevo correo para cualquier responsable de un módulo registrado en la aplicación.
Guardar nuevo correo.	El sistema guarda los datos nuevos entrados sobrescribiendo los antiguos, luego del usuario acepte la acción realizada.
<i>Subproceso Gestionar correos</i>	
Mostrar tabla con correos registrados en la base de datos.	El sistema muestra una tabla con los correos registrados en la base de datos.

A continuación el usuario puede accionar cualquiera de las siguientes actividades:	
Adicionar correo.	El usuario activará la adición de un nuevo correo que no esté registrado.
Modificar correo.	El usuario activará la modificación de alguno de los correos registrados.
Eliminar correo.	El usuario activará la eliminación de algún correo de acuerdo a su consideración.
Si el usuario desea adicionar o modificar algún correo, a continuación se ejecutará la siguiente actividad:	
Mostrar formulario para datos.	El sistema mostrará un formulario que recoja los nuevos datos a ser introducidos por el usuario.
Para finalizar el sub-proceso Gestionar correos:	
Guardar datos en la base de datos.	El sistema guardará los datos, luego del usuario acepte la acción realizada.

**Flujo-grama**

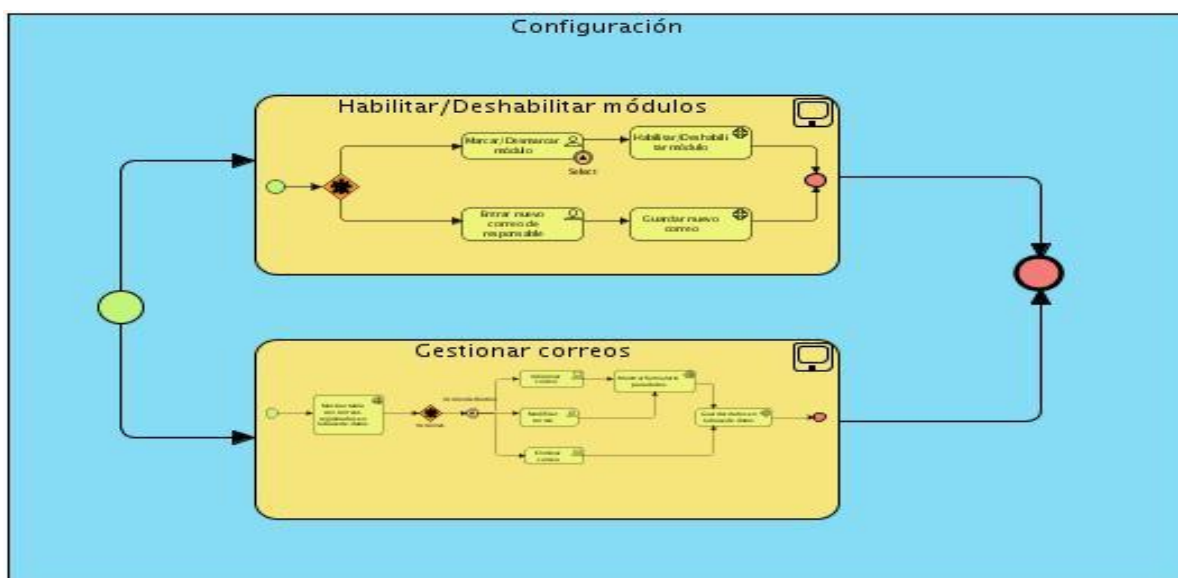


Figura 12. Configuración

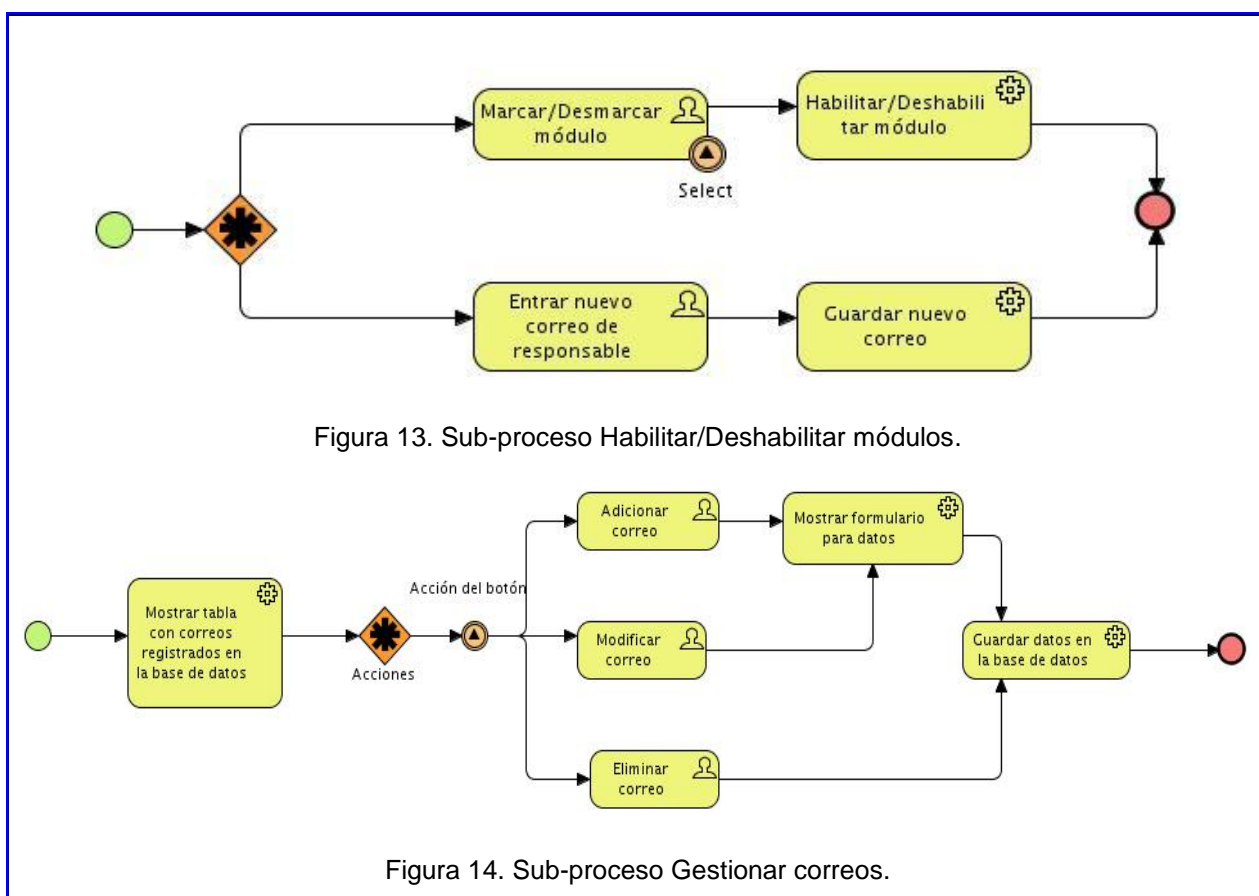


Figura 13. Sub-proceso Habilitar/Deshabilitar módulos.

Figura 14. Sub-proceso Gestionar correos.

Tabla 41. Proceso Configuración.

**Anexo 5. Casos de pruebas**

HU\_PE\_EE 11 Plan Anual

**Escenario 2: Desagregación del Plan Anual**

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
Si se da clic en el botón <b>Desagregar</b> .	Muestra al usuario una tabla con la desagregación del plan anual por meses para todos los servicios. <i>Automáticamente se guarda el plan anual estimado en la base de datos, se guarda además el plan estimado por</i>	Satisfactorio	

	<i>meses para cada uno de los consumidores.</i>		
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se da clic en el botón <b>Modificar</b> .	Se habilita para modificar cada uno de los campos de la tabla mostrada.	Satisfactorio	
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se da clic en el botón <b>Guardar</b> .	<i>Se guarda en la base de datos el plan introducido por el usuario como real.</i>	Satisfactorio	

Tabla 42. CP Plan Anual

HU\_PE\_A 71 Configuración

Escenario 1.Habilitar/Deshabilitar Módulos

<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
-----------------------	---------------------------	-------------------------------	----------------------

Al dar clic en la opción Configuración del menú del módulo. El sistema muestra una tabla donde el usuario puede habilitar o deshabilitar módulos: Módulo Activado Agua Sí. EE No. Inci. Sí.	El sistema habilita o deshabilita el modulo seleccionado luego que el usuario del clic en Aceptar.	Satisfactorio	
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
<b>Evaluación de la Prueba:</b>	<i>Satisfactoria</i>		

Tabla 43. CP Configuración

HU Gestionar correos

Escenario 1. Adicionar nuevo correo

<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Se inserta la dirección de correo correctamente: Correo: rosa@uci.cu	Se adiciona la dirección de correo.	Satisfactoria	
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de</b>	<b>Observaciones</b>

		la prueba	
Se deja el campo en blanco : Correo:	Muestra un mensaje de error "La dirección de correo no es válida"	Satisfactoria	
Se inserta la dirección de correo incorrectamente :  Correo: @uci.cu	Muestra un mensaje de error avisando al usuario de que la dirección de correo no es válida.	Satisfactoria	
Se inserta una dirección de correo que ya existe: Correo: icanizares@uci.cu	Muestra un mensaje de error advirtiendo al usuario que la dirección de correo ya existe en el sistema	Satisfactoria	
<b>Evaluación de la Prueba:</b>	<i>Satisfactoria</i>		

Tabla 44. Adicionar correo

### Escenario 2. Eliminar correo

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
Si se presiona el botón eliminar, el sistema muestra un mensaje de confirmación, el cual permite las siguientes opciones: Aceptar:	Se eliminará el correo del sistema.  No se eliminará el correo	Satisfactoria	



Cancelar:	del sistema. Y le muestra el listado de las direcciones de correos existentes.		
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
<b>Evaluación de la Prueba:</b>	<i>Satisfactoria</i>		

Tabla 45. Eliminar correo

**Escenario 3. Modificar correo**

<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se presiona la opción Modificar Correo: rosa@uci.cu	El sistema guarda el nuevo correo introducido.	Satisfactoria	
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Se modifica la dirección de correo dejando el campo en blanco: Correo:	El sistema muestra un mensaje de error "Error en la entrada de datos"  Al dar clic en Aceptar el sistema da la posibilidad de entrar los valores correctamente.	Satisfactoria	

Evaluación de la Prueba:	Satisfactoria		

Tabla 46. Modificar correo

**Escenario 4: Mostrar correos**

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
Se muestran las direcciones de correo existentes.	Se muestra una ventana con un listado de todas las direcciones de correo que están registrada en el sistema	Satisfactoria	
Clases inválidas	Resultado esperado	Resultado de la prueba	Observaciones
Evaluación de la Prueba:	Satisfactoria		

Tabla 47. Mostrar correos

HU\_PE\_EE 24 Plan Diario de los consumidores internos.

**Escenario 1: Plan de los servicios UCI.**

Clases válidas	Resultado esperado	Resultado de la prueba	Observaciones
Si se escoge la opción Plan Servicios (UCI) X Días del menú principal.	Muestra una tabla con los datos de la desagregación por meses de los consumidores principales de la UCI.	Satisfactoria	

<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Región de la vista donde se muestran las tablas en blanco.	<i>Se hace necesario que previamente se haya realizado la estimación del plan anual y luego haberse desagregado de manera que existan en la base de datos planes mensuales para todos los servicios.</i>	Satisfactoria	

Tabla 48. Plan de los servicios

**Escenario 2: Desagregación para un mes específico**

<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se presiona el botón Desagregar para algún mes específico.	Se muestra una tabla con la desagregación por días de los consumidores internos para el mes seleccionado.	Satisfactoria	
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se presiona el botón Guardar.	<i>El sistema guarda los planes para los consumidores y muestra la vista de Plan de los servicios UCI.</i>	Satisfactoria	
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>

		<b>prueba</b>	
Si se presiona el botón Exportar a excel o pdf(un botón para cada opción)	<i>Muestra el cuadro de diálogo para el abrir o guardar, de acuerdo a la exportación seleccionada.</i>	Satisfactoria	
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>

Tabla 49. Desagregación para un mes

**Escenario 3: Modificar plan diario de los consumidores.**

<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se selecciona en la lista desplegable --Seleccione-- algún día para modificar	<i>Muestra una tabla con el plan para cada consumidor del día seleccionado permitiéndole al usuario editar el plan según estime conveniente.</i>	Satisfactoria	
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se da clic en el botón Modificar luego de haber cambiado el valor de algún plan.	<i>Si el usuario incrementó el plan para un consumidor, el sistema equitativamente resta del resto de los consumidores la diferencia. Si el usuario disminuye el plan para un consumidor el sistema equitativamente asigna la diferencia al resto de los</i>	Satisfactoria	

	consumidores.		
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se da clic en el botón Aceptar.	<i>El sistema guarda los nuevos datos en la base de datos como plan real.</i>	Satisfactoria	
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Al seleccionar un día para modificar no se realiza la redistribución luego de introducir nuevos valores y dar clic en el botón Modificar.	El sistema debe guardar en la base de datos el valor introducido por el usuario y realizar una redistribución con la diferencia entre todos los consumidores.	Satisfactoria	
<b>Evaluación de la Prueba:</b>	<b>Satisfactoria</b>		

Tabla 50. Modificar plan diario

HU\_PE\_EE 16 Plan Mensual.

**Escenario 1: Plan Mensual**

<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se selecciona la opción <b>Plan Mensual</b> el menú principal del módulo Energía Eléctrica. <b>Meses</b>	Muestra una tabla con los datos del plan mensual que son estimados, en datos reales.	Satisfactoria	

Enero Circuito_1590 1350.2 Circuito_1590 1155 <b>Total</b> 2505.2 <b>Acción</b>			
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se selecciona la opción <b>Exportar a pdf o excel</b> (un botón para cada opción)	Muestra el cuadro de diálogo para el abrir o guardar, de acuerdo a la exportación seleccionada.	Satisfactoria	
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Se muestra la tabla con los campos vacíos o datos negativos.	Se hace necesario para el usuario que la tabla de los planes no este vacía y que los datos de consumo estén correctos, puesto que a partir de los datos guardados ahí se muestra el plan mensual.	Satisfactoria	
<b>Evaluación de la Prueba:</b>	<b>Satisfactoria</b>		

Tabla 51. CP Plan mensual

HU\_PE\_EE 20 Plan de los servicios UCI.

**Escenario 1: Plan de los servicios UCI.**

<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se escoge la opción Plan Servicios (UCI) X Días del menú principal.	Muestra una tabla con los datos de la desagregación por meses de los consumidores principales de la UCI.	Satisfactoria	
<b>Clases válidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Si se presiona el botón Desagregar.	Muestra una tabla con la desagregación por días de los consumidores internos para el mes seleccionado.	Satisfactoria	
<b>Clases inválidas</b>	<b>Resultado esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
Región de la vista donde se muestran las tablas en blanco.	<i>Se hace necesario que previamente se haya realizado la estimación del plan anual y luego haberse desagregado de manera que existan en la base de datos planes mensuales para todos los servicios.</i>	Satisfactoria	
<b>Evaluación de la Prueba:</b>	<b>Satisfactoria</b>		

Tabla 52. Plan de los servicios

**Anexo 6. Lista de Reserva del Producto**

RF (Requisitos funcionales)			
Código		Descripción	Prioridad
		<b><u>MÓDULO ENERGÍA ELÉCTRICA</u></b>	
		<b><u>Planificación</u></b>	
RF PE	1.	Calcular Plan anual a partir de consumos históricos.	Muy alta
RF PE	2.	Calcular peso que representa cada mes y cada servicio del total (año).	Muy alta
RF PE	3.	Desagregar el plan anual por meses y por servicios teniendo en cuenta los pesos.	Muy alta
RF PE	4.	Mostrar plan anual.	Muy alta
RF PE	5.	Modificar plan anual.	Muy alta
RF PE	6.	Calcular demanda mensual.	Muy alta
RF PE	7.	Calcular peso que representa cada servicio del total mensual.	Muy alta
RF PE	8.	Desagregar plan mensual por servicios teniendo en cuenta los pesos.	Muy alta
RF PE	9.	Mostrar plan mensual por servicios.	Muy alta
RF PE	10.	Modificar plan mensual por servicios.	Muy alta
RF PE	11.	Calcular peso que representa cada día del total mensual para cada servicio.	Muy alta
RF PE	12.	Desagregar plan mensual de los servicios por días del mes.	Muy alta



RF PE	13.	Mostrar plan mensual de los servicios por día	Muy alta
RF PE	14.	Modificar plan mensual de los servicios por día	Muy alta
RF PE	15.	Calcular el peso que representa cada consumidor interno del total diario.	Muy alta
RF PE	16.	Desagregar el plan diario entre los consumidores internos teniendo en cuenta los pesos.	Muy alta
RF PE	17.	Mostrar plan diario de los consumidores internos.	Muy alta
RF PE	18.	Leer y guardar lecturas para consumidores automáticos.	Muy alta
		<b><u>Gestionar consumo.</u></b>	
RF PE	19.	Calcular consumo de consumidores mensuales.	Muy alta
RF PE	20.	Calcular consumo de consumidores normales y generales.	Muy alta
RF PE	21.	Calcular consumo de consumidores automáticos.	Muy alta
RF PE	22.	Calcular consumo de consumidores no metrados.	Muy alta
RF PE	23.	Servicio web para mostrar los datos en la Intranet	Muy alta
RF PE	24.	Enviar correo a responsable en caso de que el consumidor no cumpla con el plan de consumo propuesto.	Muy alto
RF PE	25.	Exportar reportes a formatos excel y pdf.	Alta
RF PE	26.	Enviar los reportes por correos.	Alta
		<b><u>MÓDULO ADMINISTRACIÓN</u></b>	
		<b><u>Gestionar Correo</u></b>	

RF PE	27.	Adicionar un nuevo correo	Alta
RF PE	28.	Eliminar un correo	Alta
RF PE	29.	Actualizar correo	Alta
RF PE	30.	Mostrar correo	Baja
		<b><u>Gestionar configuraciones</u></b>	
RF PE	31.	Definir correo del responsable del módulo	Baja
RF PE	32.	Activar y/o desactivar módulos.	Baja
RF PE	33.	Definir lista de contactos para enviar información	Baja
<b>RNF (Requisitos no funcionales)</b>			
		<b>Usabilidad</b>	
RFN	1.	Facilidad de uso: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo por parte de los usuarios los cuales deben poder acceder de manera rápida y efectiva a la información solicitada. Debe además ser una interfaz de manejo cómodo donde la curva de aprendizaje para los usuarios sea lo menos inclinada posible y que posibilite en estos una rápida adaptación.	
RFN	2.	Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por parte del ese sobre la herramienta de trabajo.	
RFN	3.	Emplear perfiles de usuario: diferenciar las interfaces y opciones para los usuarios que accedan al sistema según los diferentes roles que estos	

		tengan dentro del mismo.
RFN	4.	Menús: el sistema debe presentar una serie de menús tanto laterales como en barras horizontales de iconos que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.
		<b>Fiabilidad</b>
RFN	5.	Disponibilidad: El sistema estará disponible las 24 horas del día los siete días de la semana.
RFN	6.	
		<b>Eficiencia</b>
RFN	7.	El sistema debe soportar un tiempo de respuesta menor o igual a 5 segundos.
RFN	8.	El sistema debe soportar una conexión simultanea de más de 10 000 usuarios.
		<b>Soporte</b>
RFN	9.	Grupo de soporte y asesoría: el sistema contará con un grupo de soporte y asesoría al cliente del producto destinado a brindar asesoría y soporte técnico al mismo.
		<b>Restricciones de diseño</b>
RFN	10.	Lenguaje de programación: PHP 5.1 o superior
RFN	11.	El marco de trabajo base de desarrollo que se utilizará es: CodeIgniter

		1.7.2
RFN	12.	Como IDE se empleará NetBeans 6.7. 1
RFN	13.	Como servidor Web se explotará Apache 2.2.2.
RFN	14.	El sistema gestor de bases de datos deberá ser PostgreSQL 8.4.1.
RFN	15.	El diseño de la base de datos se realizará con Visual Paradigm 3.0.
RFN	16.	El sistema operativo a utilizar en el entorno de desarrollo deberá ser: GNU Linux.
RFN	17.	El repositorio principal, el entorno de prueba y el servidor de base de datos estarán montados sobre Ubuntu Server 9.04 o superior.
		<b>Requisitos para la documentación de usuarios en línea y ayuda del sistema</b>
RFN	18.	Manual de usuario: el sistema deberá presentar un manual de usuario, permitiendo con ello un correcto uso de sus funcionalidades y brindarle al usuario una mayor experiencia del trabajo con el mismo.
RFN	19.	Documentación actualizada del grupo de desarrollo: se precisa que la documentación del sistema esté actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.
RFN		<b>Interfaz</b>
RFN	20.	Interfaz Web: la interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.
RFN	21.	Interfaz interna: la interfaz interna estará determinada por los

		desarrolladores, construyendo así una vista escalable de las clases o agrupaciones de clases que permitirán un mejor encapsulamiento de las funcionalidades y una mayor abstracción modular del sistema.
		<b>Interfaces Hardware</b>
RFN	22.	Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHZ o superior, 512 MB RAM o superior, 160 GB HDD o superior.
RFN	23.	Para explotación del cliente: PC Pentium 3 o superior, CPU 133 MHZ o superior, 256 RAM mínimo 512 RAM recomendada o superior.
RFN	24.	Para explotación del servidor: CPU Dual Core 2.0 GHZ o superior, memoria RAM de 4 GB (recomendado 6 GB), 250 GB HDD.
		<b>Requisitos Legales, de Derecho de Autor y otros</b>
RFN	25.	El sistema debe ser sometido a un análisis legal por parte de los abogados y personal autorizado con vistas a declarar su autenticidad y evitar restricciones legales para su uso y comercialización; así mismo se debe proceder a una evaluación y certificación por parte del cliente del producto.

Tabla 53. Requerimientos funcionales y no funcionales