

Tipo de artículo: Artículo original

Temática: seleccionar la temática a partir de las líneas editoriales de la revista

Recibido: 15/08/2018 | Aceptado: 20/11/2018 | Publicado: 22/11/2018

Analisis y diseño de un sistema para el control energético

Analysis and design of a system for energy control

Nohemy Cardentey Moreno¹

¹ Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas, ncardentey@uci.cu

* Autor para correspondencia: ncardentey@uci.cu

Resumen

Como parte del proceso de Informatización de la sociedad cubana, se crea la Universidad de las Ciencias Informáticas UCI, la cual cuenta con un gran número de tecnología, convirtiéndola en una alta consumidora de energía eléctrica. Para garantizar el control del consumo energético se utilizan los metros contadores, dispositivos que son leídos una vez al día por el personal encargado. Sin embargo no existe un mecanismo que genere informes sobre un posible exceso de consumo, que les permita a los principales directivos planificarse y tomar decisiones de acuerdo a los planes de consumo previstos. En el presente trabajo se describe la solución a la problemática planteada a partir de la informatización del control energético de la tecnología, para lo cual se implementó una aplicación Web en lenguaje de programación PHP, con framework de desarrollo CodeIgniter, gestor de base de datos PostgreSQL y herramienta de monitoreo Nmap. La aplicación permite crear informes que les facilitan el monitoreo y toma de decisiones a los directivos, sobre la aplicación de las regulaciones vigentes.

Palabras clave: Consumo de energía, control de energía, control energético, computadoras (PC), tecnología, monitoreo, monitoreo de redes.

Abstract

As part of the process of computerization of Cuban society, the UCI University of Computer Science is created, which has a large number of technology, making it a high consumer of electricity. To ensure control of energy consumption meters are used, devices that are read once a day by the personnel in charge. However, there is no mechanism that generates reports on possible excess consumption, which allows the main managers to plan and make decisions according to the planned consumption plans. This paper describes the solution to the problem posed from the computerization of the energy control of technology, for which a Web application was implemented in PHP programming language, with development framework CodeIgniter, PostgreSQL database manager and Nmap monitoring tool. The application allows the creation of reports that make it easier for managers to monitor and make decisions about the application of current regulations.

Keywords: *Energy consumption, energy control, energy control, computers (PC), technology, monitoring, network monitoring.*

Introducción

La energía como fuente vital es posible manifestarse como corriente eléctrica, debido a la facilidad atribuida a su uso cotidiano, representa la base de la sociedad moderna. De ella dependen la iluminación de interiores y exteriores, el calentamiento y refrigeración de nuestras casas, el transporte de personas y mercancías, la obtención de alimento y su preparación, así como el soporte de la infraestructura tecnológica de nuestras sociedades.

En ocasiones olvidamos el inmenso valor que tiene la disponibilidad de energía estrechamente vinculada al nivel de bienestar, a la salud, tributando a la calidad de vida del ser humano. Los especialistas consideran que vivimos en una sociedad que se podría denominar como "energívora", donde las personas pasan a ser altos derrochadores, manifestándose en patrones de comportamientos: los países más pobres muestran los consumos más bajos de energía y los países más ricos utilizan excesivamente grandes cantidades (MAR and CAEDENTEY 2016).

Con el desarrollo de las tecnologías ha sido posible la generación de electricidad por diversas vías como la transformación de la energía solar en eléctrica, la utilización de las fuentes hidráulicas, la generación eólica, la manipulación de la energía atómica, la utilización de la descomposición orgánica para fabricar material inflamable. El método tradicional implementado en casi todos los países del mundo y que representa un 90% de la generación en Cuba, se sustenta en las termoeléctricas, las cuales se nutren de combustibles fósiles no renovables para este proceso. Este origina grandes cantidades de gases de efecto invernadero y la generación de lluvia ácida a la atmósfera, junto a partículas volantes que pueden contener metales pesados altamente dañinos al medio ambiente.

Estudios realizados para entornos empresariales demuestran que, después de la climatización y la iluminación, los equipos de cómputos tienen el mayor consumo de energía en los entornos de oficina. Un elemento importante a tener en cuenta lo representa la administración de energía en las computadoras (PC) y monitores. Se puede reducir significativamente su consumo, y de esta forma contribuir al ahorro de importaciones representado por el costo sobre la generación eléctrica.

El ahorro de energía eléctrica se ha convertido en una de las principales tareas del Gobierno del país, tal es así que se declaró el año 2006 como “Año de la Revolución Energética en Cuba”, reorganizando el modelo de gestión energética existente hasta la actualidad y replanteando estrategias para mitigar las principales deficiencias en el sector. El consumo anual de petróleo combustible en el país asciende a unos 6 millones de toneladas y de ellos el 55% se destina a la generación de electricidad en termoelectricas (MAR and CAEDENTY 2016), (MAR and CARDENTY 2012).

Según cifras oficiales, al país le cuesta unos 2 600 000 dólares diarios la generación de electricidad. El ahorro de energía eléctrica no impone limitar el crecimiento económico y social del país. Se debe usar la electricidad consciente y responsablemente, aplicando las medidas indicadas en dependencia de la necesidad del ahorro que exista.

La cantidad de energía eléctrica que consumen las computadoras y monitores depende de la potencia y la cantidad de horas que se utiliza. El consumo de energía se mide en kilowatt hora (kW.h). Muchas veces dejan el monitor encendido y este es uno de los elementos que más consume energía en una PC. Una computadora promedio consume entre 120 W y 300 W dependiendo de sus características, fuente de poder y antigüedad del equipo. Las PC más modernas consumen menos energía (MUTO *et al.* 2016), (KAHRAMAN *et al.* 2016), (WANG *et al.* 2015).

La Universidad de las Ciencias Informáticas cuenta con un gran número de computadoras, usadas diariamente por estudiantes y trabajadores, y distribuidas en apartamentos, oficinas, y laboratorios; muchas de ellas permanecen encendidas día y noche. El área de Laboratorios cuenta con 1 300 computadoras, lo que equivale a un consumo energético de aproximadamente 156 000 W a 390 000 W (USMAN *et al.* 2015), (MENDIS *et al.* 2015), (ARNOLD *et al.* 2015).

La siguiente investigación va dirigida al control energético de los laboratorios de la UCI, dada la importancia que representa el consumo de electricidad atribuido a la tecnología. Hasta el momento este control lo llevan a cabo los técnicos de laboratorios, haciendo recorridos periódicos, apoyados por las configuraciones que se predeterminan en las PC para activar las funciones de ahorro de energía. En la práctica esta labor no es efectiva.

Materiales y métodos o Metodología computacional

Descripción del sistema propuesto

El sistema o aplicación informática que se propone permitirá establecer un control sobre el consumo energético de la tecnología que existe en la Dirección de Laboratorios de la UCI. Esta se encuentra distribuida por laboratorios

destinados a la docencia y producción. La propuesta será una herramienta de control para futuras decisiones administrativas, que permitirán apoyar al llamado de ahorro de energía que se le ha hecho a la Universidad.

Será una aplicación de alto grado de fiabilidad por el uso de la herramienta de monitoreo de redes Nmap que no requiere previa configuración de las PC que serán escaneadas, privilegios administrativos, o clientes dentro de las PC examinadas(M *et al.* 2014).

Definir Modelo de Dominio

Debido a las características, grado de complejidad del problema que se plantea y el conocimiento que se posee sobre el mismo, se decide realizar Modelo de Dominio o Modelo Conceptual como también se le conoce. Se considera suficiente el empleo de esta variante de modelo de negocio, dado que se centra en una parte del negocio, la relacionada con el ámbito del proyecto. El modelo conceptual de la problemática a resolver se muestra en la Figura.1.

El Modelo de Dominio es una representación visual “estática” del entorno real de los objetos del proyecto. Así como ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación.

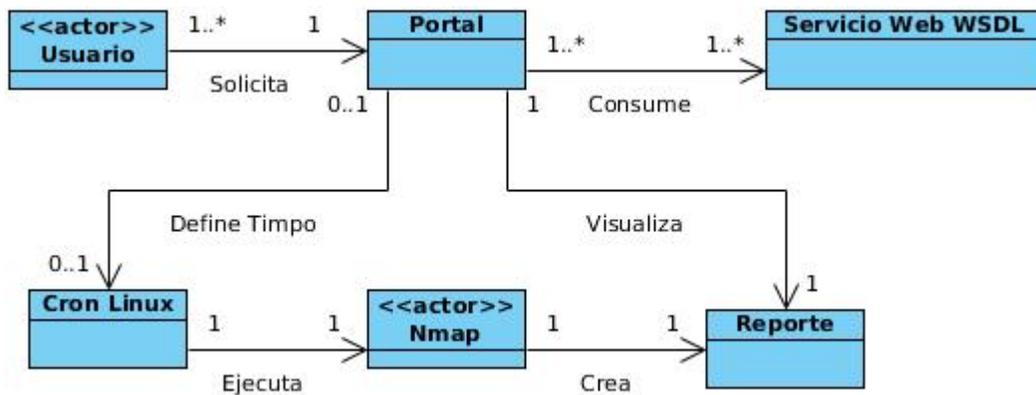


Figura.1: Diagrama del Modelo de Dominio creado a partir del problema planteado.

Descripción de los Objetos

La siguiente tabla muestra la definición de las entidades que interactúan en el negocio y los conceptos principales que se emplean en el problema que se analiza.

Tabla.1: Descripción de las entidades que interactúan en el negocio.

<i>Entidad</i>	<i>Descripción</i>
Cron Linux	Entidad encargada de iniciar la herramienta de escaneo de forma automática.
Nmap	Subsistema que ejecuta las peticiones previamente definidas en el <i>cron</i> o por solicitud del usuario.
Reporte	Contiene la información de los resultados obtenidos.
Portal	Es la entidad que recibe las peticiones del usuario y se encarga de procesar y devolver la solución.
WSDL	Servicios web que pueden ser consumidos.
Usuario	Representa la persona que interactúa con el portal solicitando una respuesta.

Especificación de los requisitos del sistema.

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional.

Los requerimientos para un sistema de software determinan lo que hará el sistema y definen las restricciones de su operación e implementación.

Requisitos funcionales.

Los requisitos funcionales establecen el comportamiento y denotan la funcionalidad del sistema. Son capacidades o condiciones que el sistema debe cumplir. El sistema en análisis debe efectuar y garantizar:

R.1 – Autenticar Usuario.

R.2 – Gestionar Usuario.

R.2.1 – Insertar un nuevo usuario.

R.2.2 – Buscar usuario partiendo de un criterio especificado.

R.2.3 – Modificar datos de un usuario existente.

R.2.4 -Eliminar un usuario del sistema.

R.3 – Gestionar Área.

R.3.1 – Insertar una nueva área.

R.3.2 – Buscar un área partiendo de un criterio especificado.

R.3.3 – Modificar datos de un área existente.

R.3.4 -Eliminar un área existente en el sistema.

R.4 – Gestionar Laboratorio.

R.4.1 – Insertar un nuevo laboratorio, teniendo en cuenta el área al que pertenece.

R.4.2 – Buscar laboratorio partiendo de un criterio especificado.

R.4.3 – Modificar los datos del laboratorio.

R.4.4 -Eliminar laboratorio existente.

R.5 – Administrar *cron* de Linux.

R.5.1 – Agregar un nuevo horario de ejecución del script.

R.5.2 – Modificar horario de ejecución del script.

R.5.3 – Eliminar horario de ejecución existente.

R.6 – Mostrar reporte correspondiente a los parámetros especificados por el usuario.

R.7 – Listar consumo de acuerdo a la selección que realice el usuario (Semanal, Mensual o Anual).

R.8 – Realizar escaneo por VLAN

Requisitos no funcionales

Llámesse requisito no funcional a todas las exigencias de cualidades que se le imponen al proyecto: exigencias de usar un cierto lenguaje de programación o plataforma tecnológica por ejemplo. Es una característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo.

Aseguran que se disponga de un sistema manejable y gestionable que ofrezca la funcionalidad requerida de manera fiable, ininterrumpida o con el tiempo mínimo de interrupción, incluso ante situaciones inusuales.

Requerimientos de Usabilidad

El sistema está destinado principalmente a los directivos de la Dirección de Laboratorios, de igual forma puede ser utilizado por cualquier otro individuo. Los usuarios finales deberán poseer conocimientos básicos de computación, navegación y exploración de los sitios web en sentido general para lograr un mejor uso de las funcionalidades que brinda la aplicación.

Requerimientos de Software

Para obtener un óptimo funcionamiento del sistema se requiere la existencia de los siguientes requisitos en el servidor y las máquinas clientes que harán uso de la aplicación.

Cliente

- Cualquier sistema operativo con interfaz gráfica
- Red activa.
- Cualquier navegador web

Servidor (Requerimientos mínimos)

- Sistema Operativo Linux.
- Servidor de Base de Datos PostgreSQL 9.0.1
- Servidor Web Apache 2.26
- Nmap 5.0
- Lenguaje PHP 5.3.3

Requerimientos de Hardware

El buen estado de las conexiones de red de la Universidad es imprescindible y de suma importancia para la recogida de la información que será almacenada en el servidor y posteriormente solicitada por los usuarios.

Cliente

- Tarjeta de red de 10 Megabytes (Mb) o superior
- 64 Mb de RAM

Servidor (Requerimientos mínimos)

- Microprocesador Pentium IV
- Memoria RAM 1.0 Gigabytes (Gb)
- Capacidad de disco duro 80 Gb

Requerimientos de apariencia o interfaz externa

Este tipo de requisito describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto, necesidades que debe cumplir con normas estándares, o con los estándares de la empresa para la cual se esté desarrollando el software.

- La interfaz del sistema se realizará a través de una página Web.
- Personalizada de acuerdo al tipo de usuario que acceda, logrando así que los mismos se sientan confiados.
- Sencilla y de fácil uso, siguiendo un orden lógico de los eventos y permitiendo una navegación eficiente.
- Con reconocimiento visual a través de elementos que identifican las acciones
- Gama cromática respetando pautas y en concordancia con el estilo de las aplicaciones de la UCI

Requerimientos de Diseño e Implementación

Los requerimientos de diseño e implementación especifican o restringen la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

- *Legales*

Aplicación web destinada al control del consumo energético de la tecnología de la Dirección de Laboratorios de la UCI. Se rige por las normas y leyes establecidas por el país y la Universidad y es desarrollada utilizando herramientas libres bajo la licencia GNU/GPL.

- **Confiabilidad**

La aplicación está concebida para lograr un funcionamiento óptimo las 24 horas del día; garantizando un alto porcentaje de fiabilidad en la información que se recoge. Posibilitando además que los usuarios autorizados tengan acceso a la misma en cualquier momento y desde cualquier lugar.

En caso de presentarse alguna anomalía en el funcionamiento del sistema, se estableció un tiempo de reparación menor de 24 horas.

Requerimientos de Seguridad

Fueron definidas un grupo de políticas de seguridad que prevén el uso inadecuado del sistema. Los usuarios deberán identificarse antes de acceder a cualquier acción del sistema, así se garantiza que la aplicación sea utilizada solo por aquellos usuarios que tienen permisos. El administrador es el máximo responsable de la aplicación y el encargado de asignar estos permisos.

La contraseña de los usuarios del dominio es manipulada con minucioso cuidado por su alto grado de confiabilidad que representa dentro de la UCI, utilizando para ello el servicio WSDL de autenticación que brinda la Universidad, evitando que dicha información sea almacenada en el Servidor Web que alberga la aplicación. Para el resto de las contraseñas se utiliza BD con restricciones de seguridad previamente configuradas y mecanismos de encriptación (MD5) evitando que la información viaje en texto plano (claro).

Resultados y discusión

El diagrama de casos de uso (DCU) de la Figura.2 muestra los actores y casos de uso definidos para el sistema propuesto, así como las diferentes relaciones que existen: generalización y asociación.

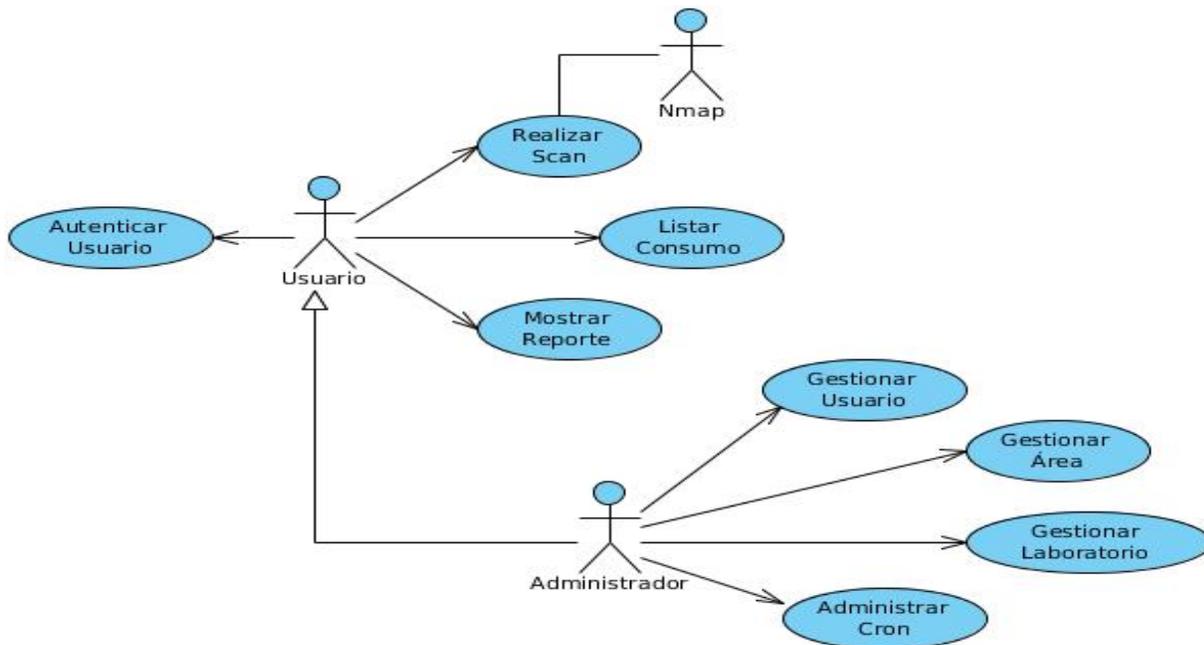


Figura.2: DCU del sistema definido a partir del problema planteado.

Patrón de Casos de Uso (CRUD)

Un patrón de casos de uso define los comportamientos que deben existir en el sistema, ayuda a describir qué es lo que el sistema debe hacer, es decir, describe el uso del sistema y cómo este interactúa con los usuarios. Son utilizados generalmente como plantillas que especifican como deberían ser estructurados y organizados los casos de uso y capturan mejores prácticas para modelar casos de uso.

Los patrones de casos de uso brindan beneficios tales como:

- Aumentar la productividad.
- Reutilizar elementos existentes.
- No invertir tiempo en resolver problemas ya resueltos.
- Aplicar la teoría al trabajo práctico.
- Habilitar las herramientas de soporte para modelar el desarrollo.

Durante el diseño de los casos de uso del sistema se utilizó el patrón CRUD (acrónimo de Crear, Obtener, Actualizar y Borrar del original en inglés: Create, Read, Update and Delete). El mismo permite modelar todas las operaciones

que se pueden realizar sobre una misma entidad, tales como crearla, leerla, actualizarla y eliminarla. Definiéndose específicamente en los casos de uso: Gestionar Usuario, Gestionar Área, Gestionar Laboratorio y Administrar Cron.

Los patrones y estilos son comportamientos que deben existir en el sistema, ayudan a describir qué es lo que debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. Estos patrones son utilizados generalmente como plantillas que describen como debería ser estructurados y organizados los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso.

Cliente-Servidor

En el mundo de Protocolo de Control de Transmisión y Protocolo de Internet (*Transmission Control Protocol / Internet Protocol*) TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama modelo Cliente-Servidor. Este término fue usado por primera vez en 1980 para referirse a las PC en red. Esta arquitectura se divide en dos partes claramente diferenciadas: servidor y clientes. Normalmente el servidor es una PC bastante potente que actúa de depósito de datos y funciona como un SGBD. Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red.

Esta tecnología proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso, en múltiples plataformas.

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). Es el proceso encargado de atender a múltiples clientes que hacen peticiones. Figura 3 Cliente-Servidor



Figura.3: Representación gráfica del flujo Cliente-Servidor

Considerando la lógica de la aplicación que se desea desarrollar y analizando que el modelo cliente- servidor provee usabilidad, flexibilidad, interoperabilidad, escalabilidad en las comunicaciones, uso de entornos multiplataforma y recursos heterogéneos, se decidió utilizar este estilo arquitectónico para el desarrollo de la aplicación.

Modelo Vista Controlador

Modelo Vista Controlador (*Model - View - Controller*) MVC, es un patrón de desarrollo o un estilo de arquitectura de software que separa el código fuente de las aplicaciones en tres grupos: Modelo, Vista y Controlador.

Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el SGBD y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. Figura 4.

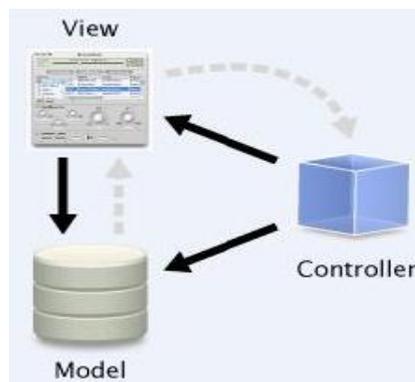


Figura.4 Representación gráfica del flujo MVC

El framework CodeIgniter fue seleccionado por estar basado en este patrón de desarrollo (Figura 5), que proporciona enormes ventajas como son:

- Separación de capas como presentación, lógica de negocio, acceso a datos
- Desarrollo de arquitecturas consistentes, reutilizables y fáciles de mantener, lo que resulta en un ahorro de tiempo en el desarrollo de posteriores proyectos.

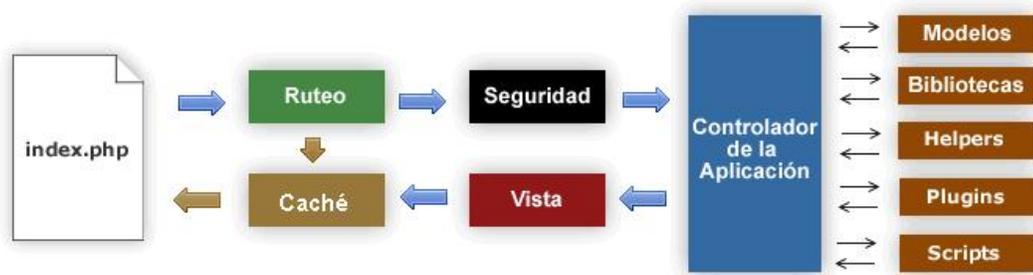


Figura.5: Patrón MVC en el Framework CodeIgniter

Patrones de diseño (GRASP)

Los patrones de diseño son una manera práctica de describir ciertos aspectos de la organización de un programa, la conexión (relación) entre sus componentes, así como crea esquemas para definir estructuras que permiten construir sistemas de software.

Durante el desarrollo de la aplicación utilizarán patrones básicos de asignación de responsabilidades como Patrones Generales de Software para Asignar Responsabilidades (*General Responsibility Assignment Software Patterns*)

GRASP

GRASP: El nombre se elige para indicar la importancia de captar (*grasping*) estos principios, si se quiere diseñar eficazmente el software orientado a objetos.

- **Experto:** Se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad.
- **Creador:** Este patrón es el responsable de crear una nueva instancia de alguna clase. Asignarle a la clase B la responsabilidad de crear una instancia de clase A.
- **Alta Cohesión:** Asigna una responsabilidad de forma tal que la cohesión siga siendo alta.
- **Bajo Acoplamiento:** Este patrón es el encargado de asignar una responsabilidad para conservar bajo acoplamiento. Una clase con bajo acoplamiento no depende de muchas otras clases. Las clases con alto acoplamiento recurren a muchas clases y no es conveniente. Son más difíciles de mantener, entender y reutilizar

- **Controlador:** Este patrón tiene la responsabilidad del manejo de mensajes de los eventos del sistema a una clase.

Diseño de la Base de Datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos correctamente diseñada permite obtener acceso a la información exacta y actualizada.

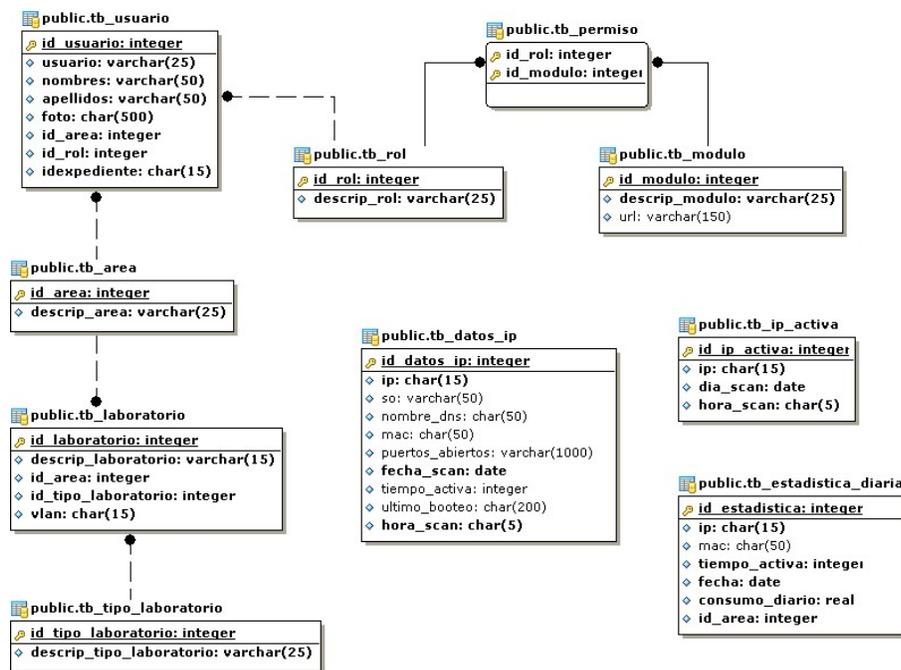


Figura.6: Representación gráfica de la base de datos.

Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos. Describen la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos, componentes de software. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

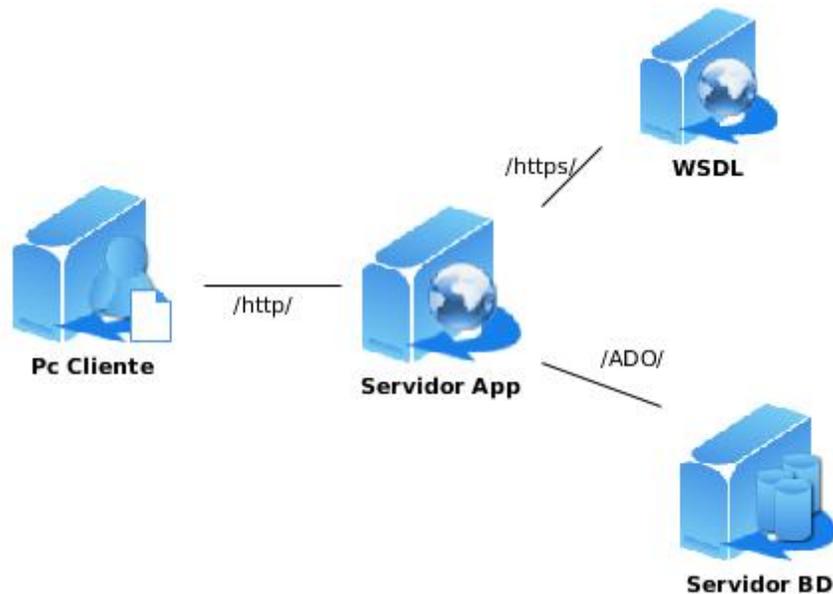


Figura.7: Diagrama de Despliegue de la Aplicación.

Descripción de los Nodos:

- **Pc Cliente:** El nodo correspondiente a la computadora cliente representa todas aquellas PC que pueden ser utilizadas por los usuarios para acceder a la aplicación.
- **Servidor App:** El nodo Servidor App representa el servidor donde está alojada la aplicación destinada al control energético y su correspondiente base de datos.
- **WSDL:** Este nodo representa gráficamente el servicio web que brinda la UCI a la comunidad universitaria.

Descripción del tipo de Comunicación:

La comunicación entre los nodos que representan las Pc Clientes y el Servidor de aplicaciones (Servidor App) se realiza mediante el protocolo HTTP. Protocolo a nivel de aplicación con la agilidad y velocidad necesaria para sistemas de información distribuidos, colaborativos y de hipermedia. Es un protocolo orientado a objetos, genérico, que puede usarse para muchas tareas extendiendo sus métodos. Una característica de HTTP es que permite que los sistemas se construyan independientemente de la información que se transfiere.

Los nodos correspondientes al Servidor App y WSDL utilizan el protocolo de comunicación seguro HTTPS, como protocolo de seguridad dado el alto grado de confidencialidad que requieren las contraseñas de los usuarios que

acceden a la aplicación. HTTPS utiliza un cifrado basado en la Capa de Conector Seguro (*Secure Socket Layer*) y *Capa de Transporte Seguro (Transport Layer Security)* SSL/TLS para crear un canal cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente.

El nodo Servidor App establece comunicación con el Servidor de BD mediante el protocolo de Acceso a Objetos de Datos (*ActiveX Data Objects*) ADO. Protocolo basado en el modelo de objetos, el cual define una jerarquía de objetos programables que pueden ser usados por desarrolladores de páginas Web para acceder a la información almacenada en una base de datos.

Conclusiones

Se elaboraron los diagramas de clases de diseño e interacción de los principales casos de uso, así como el diseño de la BD y el diagrama de despliegue.

Una vez modelado los artefactos que genera la metodología OpenUp durante la etapa de desarrollo Análisis y Diseño que dan solución a la situación problemática existente es posible pasar a las siguientes fases: Implantación y Análisis de los Resultados.

Referencias

- ARNOLD, T.; A. C. ADEWOLE, *et al.* *Performance testing and assessment of multi-vendor protection schemes using proprietary protocols and the IEC 61850 standard.* 2015 International Conference on the Industrial and Commercial Use of Energy (ICUE), 2015. 284-290 p. 2166-0581
- KAHRAMAN, C.; B. ÖZTAYŞI, *et al.* A Comprehensive Literature Review of 50 Years of Fuzzy Set Theory *International Journal of Computational Intelligence Systems*, 2016, 9(sup1): 3-24.
- M, J.; X00D, *et al.* Compact Airborne C-Band Radar Sounder *IEEE Transactions on Geoscience and Remote Sensing*, 2014, 52(10): 6326-6332.
- MAR, O. and N. CAEDENTY Monitoreo energético en los laboratorios de la Universidad de las Ciencias Informáticas *Ingeniería Industria*, 2016, Vol. XXXVI(No.2): 190-199.
- MAR, O. and N. CARDENTY Aplicación informática para el control energético de la tecnología utilizando herramienta de monitoreo de red Nmap *Revista Cubana de Ciencias Informáticas*, 2012, Vol.6(No.2).

- MENDIS, N.; K. M. MUTTAQI, *et al.* Remote Area Power Supply System: An Integrated Control Approach Based on Active Power Balance *Industry Applications Magazine, IEEE*, 2015, 21(2): 63-76.
- MUTO, D.; E. GONZÁLEZ, *et al.* Estrategia colaborativa para asimilar tecnologías energéticas alternativas y co-productos de biomasa forestal *Ingeniería Industrial*, 2016, Vol. XXXVII(No.2): p. 218-231.
- USMAN, A.; P. PAVAN KUMAR, *et al.* Battery charging and discharging kit with DAQ to aid SOC estimation. 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), 2015. 0013-0020 p.
- WANG, T.; T. BI, *et al.* Decision tree based online stability assessment scheme for power systems with renewable generations *CSEE Journal of Power and Energy Systems*, 2015, 1(2): 53-61.