

# Trabajo de Diploma en opción al título de Ingeniero Informático

**"Sistema para la extracción automática y gestión  
de parámetros eléctricos en el Banco de Crédito  
y Comercio."**

**Autor:** Isidro Alberto Domínguez Anido

**Tutores:** Lic. Lorenzo Luis Peña Marrero

Ing. Leandro Rosales Rodés

## DECLARACIÓN DE AUTORÍA.

Declaro que soy el único autor de este trabajo y autorizo a la Facultad de Informática y Matemática de la Universidad de Holguín "Oscar Lucero Moya" para que hagan el uso que estimen pertinente con el mismo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ el 2014.

\_\_\_\_\_  
Isidro Alberto Domínguez Anido  
Autor

\_\_\_\_\_  
Lic. Lorenzo Luis Peña Marrero  
Tutor

\_\_\_\_\_  
Ing. Leandro Rosales Rodés  
Tutor

## OPINIÓN DE LOS TUTORES

Título: Sistema informático para la supervisión automática y gestión de parámetros eléctricos en el Banco de Crédito y Comercio.

Autor: Isidro Alberto Domínguez Anido

---

---

---

---

---

---

---

---

Por todo lo anteriormente expuesto se considera que el estudiante está apto para ejercer como Ingeniero Informático, y se propone que se le otorgue al Trabajo de Diploma la calificación de \_\_\_ puntos.

Para que así conste firman la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ de 2014.

\_\_\_\_\_  
Lic. Lorenzo Luis Peña Marrero  
Tutor

\_\_\_\_\_  
Ing. Leandro Rosales Rodés  
Tutor

## RESUMEN

La eficiencia energética es un criterio clave para el desarrollo de la sociedad cubana. Tal es el caso del Banco de Crédito y Comercio de Holguín, entidad donde la estabilidad eléctrica y el control estricto de los planes de consumo son un aspecto esencial, dado el creciente índice de consumo eléctrico que la misma ha experimentado en los últimos años. Por tal motivo se decide desarrollar un sistema para la extracción automática y gestión de parámetros eléctricos, a partir de una adecuada explotación del hardware de análisis de la red eléctrica que se encuentra disponible. Para ello se utilizaron diversas tecnologías, como Object Pascal, PHP, JavaScript, Symfony, ExtJS, SQL Server y Apache Web Server, siguiendo la metodología de desarrollo ágil AUP. El producto informático desarrollado demuestra su sostenibilidad y correctitud.

## ABSTRACT

Energetic efficiency is a key concept in the development of the Cuban society. Such is the case of BANDEC in Holguín, banking institution where the electric stability and a strict control of consumption plans are essential elements, give then growing use of electricity registered in the latest years. For such reason, we decided to develop a system to automatically extract and manage electric parameters, taking advantage of the existing hardware for electric networks supervising. To that end we employed several technologies, such as Object Pascal, PHP, JavaScript, Symfony, ExtJS, SQL Server and Apache Web Server. We followed the agile software development methodology AUP. The resulting product demonstrated its sustainability and correctness.

# ÍNDICE DE CONTENIDO

INTRODUCCIÓN .....	1
CAPÍTULO 1 FUNDAMENTOS TEÓRICOS .....	6
1.1 Descripción de la entidad objeto de estudio .....	6
1.2 Descripción del proceso actual .....	7
1.3 Soluciones integrales existentes .....	8
1.4 Conceptos asociados al dominio del problema .....	9
1.4.1 Definición de los parámetros supervisados .....	9
1.4.2 Lectura de medición automática .....	11
1.4.3 Analizador de Red .....	11
1.4.4 Sistemas de Supervisión y Control por Computadoras (SCADA) .....	12
1.4.5 Protocolos de comunicaciones .....	12
1.4.6 Protocolo de comunicación MODBUS .....	13
1.4.7 NPort 5230 .....	15
1.4.8 Servicio de Windows .....	16
1.5 Tendencias y tecnologías a usar para el desarrollo del sistema informático .....	17
1.5.1 Metodología de desarrollo de software .....	17
1.5.2 Lenguaje de modelado .....	20
1.5.3 Herramienta CASE .....	20
1.5.4 Sistema gestor de base de datos .....	22
1.5.5 Entorno de desarrollo integrado .....	23
1.5.6 Lenguaje de Programación .....	24
1.5.7 Servidor Web .....	25
1.5.8 Framework .....	26
1.6 Conclusiones parciales .....	28
CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA .....	29
2.1 Modelo .....	29
2.1.1 Requerimientos funcionales .....	29
2.1.2 Requerimientos no funcionales .....	30
2.1.3 Casos de uso del sistema .....	31
2.1.4 Arquitectura .....	33
2.1.5 Diagrama de despliegue .....	35
2.1.6 Diseño de la base de datos .....	36
2.2 Valoración de sostenibilidad .....	38

2.2.1	Dimensión Administrativa .....	38
2.2.2	Dimensión Socio – Humanista .....	38
2.2.3	Dimensión Ambiental .....	38
2.2.4	Dimensión Tecnológica.....	39
2.2.5	Conclusiones de la valoración de sostenibilidad .....	39
2.3	Implementación .....	39
2.3.1	Servicio que captura datos automáticamente del analizador de red.....	39
2.3.2	Aplicación de configuración del servicio.....	41
2.3.3	Aplicación web.....	42
2.4	Pruebas al sistema .....	43
2.4.1	Pruebas al servicio (lectura).....	43
2.4.2	Pruebas al servicio (almacenamiento de datos leídos) .....	45
2.4.3	Pruebas a gráficas y reportes de variables .....	45
2.4.4	Pruebas a la aplicación web .....	46
2.4.5	Prueba general del flujo de datos .....	46
2.5	Conclusiones parciales .....	46
	CONCLUSIONES GENERALES .....	48
	RECOMENDACIONES .....	49
	REFERENCIAS BIBLIOGRÁFICAS .....	50
	ANEXO 1. Descripción de los casos de uso en formato de alto nivel.....	53
	ANEXO 2. Modelos de pruebas de aceptación .....	55
	ANEXO 3. Diagrama Entidad Relación .....	58

## Índice de figuras

Figura 1.1 Organigrama de la Dirección Provincial de BANDEC .....	6
Figura 1.2 Diagrama de proceso .....	8
Figura 1.3 Analizador de red PowerLogic PM710 [4] .....	12
Figura 1.4 Detalle de una trama [12] .....	13
Figura 1.5 Comunicación Maestro - Esclavo .....	14
Figura 1.6 Trama RTU de solicitud [17] .....	14
Figura 1.7 Trama RTU de respuesta [17] .....	15
Figura 1.8 Serial to Ethernet Device Server NPort 5230 [19] .....	16
Figura 1.9 Modo de conexión COM Real [18] .....	16
Figura 2.1 Diagrama de Casos de Usos .....	33
Figura 2.2 Patrón de diseño Modelo-Vista-Controlador .....	34
Figura 2.3 Modelo Arquitectónico del sistema .....	35
Figura 2.4 Diagrama de Despliegue .....	36
Figura 2.5 Diagrama de Clases Persistentes .....	37
Figura 2.6 Diagrama de proceso de captura automática de datos .....	40
Figura 2.7 Interfaz principal de la aplicación de configuración .....	41
Figura 2.8 Interfaz de configuración de dispositivos .....	41
Figura 2.9 Interfaz de la aplicación web con usuario autenticado .....	43

## INTRODUCCIÓN

La eficiencia energética es un criterio que debe estar siempre presente en cada organización empresarial, institución pública o privada, y en cada ciudadano, a fin de aumentar la sostenibilidad del planeta, ya que dicho criterio está orientado a reducir la energía consumida para producir bienes y servicios, sin que la calidad de ambos se vea afectada. [1].

“La electricidad es la forma de energía más controlable. Por lo tanto, hacer un buen uso de ella puede ser la forma más inteligente de ahorrar energía y, por esta razón, es importante afrontar el reto de mejorar la eficiencia energética de las instalaciones eléctricas.” [2]

El uso de analizadores de red posibilita la medición de variables eléctricas como potencias, energías, corrientes de fuga, etc., en el sector de servicios e industrial. Con esto se obtiene la información necesaria para la realización de estudios de consumo de energía eléctrica. El hecho de que el usuario pueda visualizar permanentemente cuánto consume contribuye a que se haga un uso más eficiente de la energía [1].

En nuestro país, luego de la caída del campo socialista en la década del 90, se produjo una brusca reducción de los suministros de petróleo que se recibían de manera estable y a precios preferenciales. Al mismo tiempo se recrudeció el bloqueo económico, presentándose un intenso déficit de energía que impactó fuertemente en la economía nacional. Por esta causa se redujeron los niveles de actividad económica del país, incluida la generación de electricidad y el transporte, tanto de cargas como de pasajeros, quedando, además muy restringidos los combustibles domésticos empleados en la cocción de alimentos [3].

La nueva situación llevó a un reordenamiento de las concepciones de la política energética nacional, y a partir del año 2000 se llevó a cabo un programa de profundas transformaciones que tuvieron su punto culminante en el año 2005 con la Revolución Energética, que estableció un conjunto de estrategias encaminadas a transformar los esquemas de generación de electricidad y consumo de portadores energéticos, tanto en el sector estatal como en el residencial [3].

Las principales directivas de la Revolución Energética están orientadas, entre otras, a:

- La implementación del Programa de Ahorro de Electricidad (PAEC) para reducir las tasas de crecimiento de consumo y la demanda máxima.

- Las medidas de ahorro en los sectores residencial y estatal que permitan el uso racional de la energía.
- La rehabilitación de las redes eléctricas.

En sus inicios, el Banco de Crédito y Comercio (BANDEC) no contaba con el avance tecnológico que posee en la actualidad. Antes de su estructuración, para garantizar todos los procesos que tributan a su objeto social, se utilizaban tecnologías cuyo consumo eléctrico era notablemente bajo, como máquinas de escribir, cajas XX, cocinas de gas, etc. Con el cambio de equipamiento realizado a fin de satisfacer el aumento en la demanda de los servicios bancarios, el gasto de energía aumentó considerablemente, debido a que se adquirieron computadoras para todos los puestos de trabajo, se instalaron cajas y máquinas electrónicas para contar dinero, se instalaron además aires acondicionados para garantizar la climatización necesaria, etc. De igual forma, se aumentó en un número considerable la cantidad de refrigeradores y hornos micro ondas, como solución a la eliminación del almuerzo a los trabajadores.

Por el drástico aumento en el consumo eléctrico que experimentó BANDEC, el Departamento de Servicios Administrativos de la entidad asumió, entre otras tareas, la misión de controlar y dar seguimiento al consumo eléctrico, con el objetivo de garantizar el cumplimiento del plan de energía y evitar un consumo excesivo de electricidad, provocando gastos innecesarios a la empresa.

Para este fin se designó una plaza de “energético”, cuya responsabilidad principal fuera la lectura diaria del metro contador y el procesamiento de los datos recopilados durante la lectura. Hasta la fecha, los trabajadores que han ocupado dicha plaza han realizado todo este proceso de forma manual, llevando el control de los planes y el consumo en hojas de cálculo de Excel.

Por la forma en que se lleva a cabo este proceso se incurre en diversas deficiencias que pueden provocar errores o inconsistencias en la información que se procesa, tales como:

- Impedimento en la realización de la lectura diariamente y/o en el mismo horario.
- Errores humanos en la lectura del metro contador, ya sea en el momento de tomar los datos o en el momento de registrar la información.

- El consumo durante el horario pico y la madrugada no se controla ni analiza actualmente, ya que estos horarios están fuera del horario laboral.
- Los reportes son generados manualmente.
- No se aprovecha la bondad que brinda el analizador de red de realizar un monitoreo de la red eléctrica.

Todo esto complejiza la realización de análisis estadísticos y de tendencia, y además dificulta la opción de tener reportes actualizados con la situación energética en un momento dado.

A partir de esta situación se define el **problema científico**: ¿Cómo favorecer la extracción y gestión de los parámetros eléctricos en el Banco de Crédito y Comercio? Queda delimitado, por tanto, el **objeto de estudio**, como: el proceso de extracción automática y gestión de parámetros eléctricos.

Para dar solución al problema planteado, se propone como **objetivo**: desarrollar un sistema informático que permita la extracción automática y gestión de parámetros eléctricos en el Banco de Crédito y Comercio.

Queda delimitado, por tanto, el **campo de acción** en: el proceso de extracción automática y gestión de parámetros eléctricos en el Banco de Crédito y Comercio.

A fin de dar cumplimiento al objetivo se formularon las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos para conectar una central de medida a un dispositivo maestro a través de un servidor de dispositivos Serie - Ethernet?
- ¿Cómo desarrollar un servicio de Windows capaz de obtener automáticamente los parámetros eléctricos de un analizador de red?
- ¿Cómo implementar una base de datos para almacenar parámetros eléctricos obtenidos de un analizador de red?
- ¿Cómo desarrollar una aplicación web para la gestión de los datos obtenidos de un analizador de red?
- ¿Cumple el sistema los requerimientos para los cuales fue diseñado?

Para guiar la investigación se propusieron las siguientes **tareas científicas**:

1. Estudiar el marco teórico referencial para conectar una central de medida a un dispositivo maestro a través de un servidor de dispositivos Serie – Ethernet.
2. Implementar una base de datos para almacenar parámetros eléctricos obtenidos de una central de medida.
3. Desarrollar un servicio de Windows para obtener los parámetros eléctricos del analizador de red.
4. Desarrollar una aplicación web para la gestión de los datos obtenidos de un analizador de red.
5. Valorar la propuesta de solución mediante pruebas unitarias y pruebas de aceptación.

En el cumplimiento de estas tareas fueron empleados diversos **métodos de investigación científica** teóricos y empíricos.

#### **Métodos Teóricos:**

**Histórico – Lógico:** Se llevó a cabo un análisis del proceso de obtención de variables eléctricas de una central de medida. Esto permitió organizar la información de forma lógica para así lograr un mejor entendimiento de esta investigación.

**Enfoque sistémico:** Se analizó y determinó las relaciones funcionales, dependencias y la modelación del sistema.

**Modelación:** Se empleó durante la fase de diseño del sistema, guiado por la metodología de ingeniería de software AUP con el objetivo de lograr un mayor entendimiento de todos los procesos.

#### **Métodos Empíricos:**

**Observación científica:** Permite realizar un análisis detallado de las condiciones de los medios informáticos con que cuenta la entidad, para valorar el nivel de eficiencia y calidad que deberá tener el sistema acorde a las condiciones reales.

**Revisión de la documentación:** Se utilizó para comprender y determinar los requerimientos funciones del sistema.

**Entrevista:** Permite la obtención de información sobre el proceso actual y búsqueda de opiniones, facilitando la recopilación de elementos para el análisis del sistema.

La presente investigación está estructurada de la siguiente manera:

**Capítulo I: Fundamentos teóricos.** Se exponen las bases teóricas que sustentan la investigación y los conceptos relacionados con el objeto de estudio. Luego se describen las principales herramientas que se utilizan para la elaboración de la propuesta y la metodología de desarrollo de software utilizada en la investigación.

**Capítulo II: Descripción de la solución propuesta.** Resume la elaboración de la solución propuesta, modelo e implementación, además de la valoración de sostenibilidad del sistema informático y las pruebas realizadas al mismo para demostrar su correctitud.

## CAPÍTULO 1 FUNDAMENTOS TEÓRICOS

En este capítulo se abarcan los conceptos esenciales relacionados con la supervisión automática de parámetros eléctricos de un analizador de red, se describen las tecnologías empleadas para la elaboración de la solución propuesta y se fundamenta la metodología de desarrollo utilizada.

### 1.1 Descripción de la entidad objeto de estudio

El Banco de Crédito y Comercio (BANDEC) es una institución financiera con funciones de banda universal, que cuenta con una Dirección Provincial en cada una de las provincias del país. Las Direcciones Provinciales están subordinadas a la Presidencia del Banco, y a ellas se le subordinan las Sucursales del BANDEC incluidas dentro de su territorio que no constituyan Sucursales Especiales adscriptas a la Oficina Central. A la dirección provincial también se subordinan diferentes departamentos, como se muestra en la Figura 1.1 Organigrama de la Dirección Provincial de BANDEC.

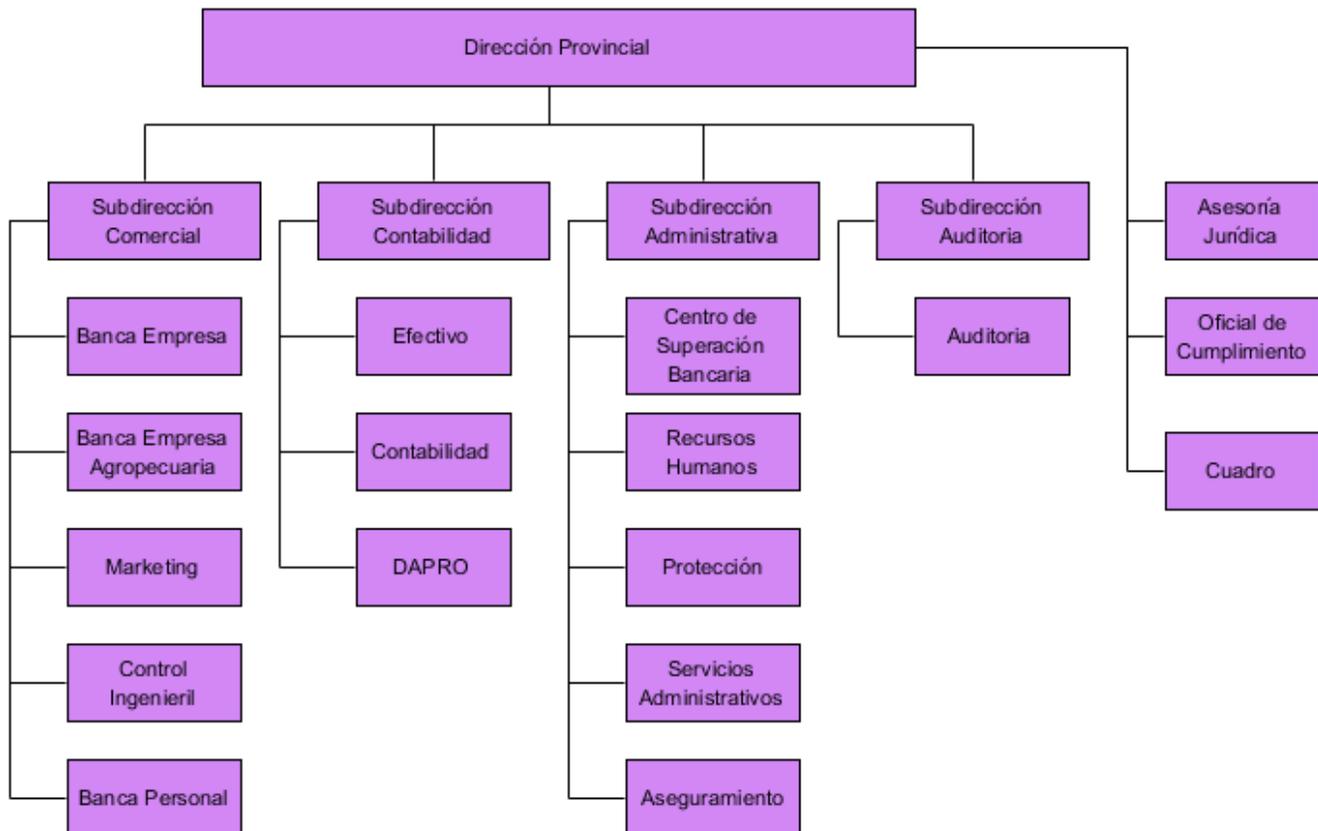


Figura 1.1 Organigrama de la Dirección Provincial de BANDEC

El Banco de Crédito y Comercio de Holguín es una entidad que se caracteriza por ser alta consumidora de energía eléctrica, lo cual está dado por el aumento en las tecnologías informáticas que ha experimentado en los últimos años con el objetivo de favorecer la prestación de servicio y el trabajo contable. A esto se le suma el aumento en un número considerable de refrigeradores y hornos microondas, como solución alternativa ante la eliminación del almuerzo a los trabajadores.

Este aumento de carga experimentado en los últimos años implicó remodelar completamente el sistema eléctrico, eliminando toda la cablearía obsoleta y problemas de conectividad existentes. Para lograr una mejor eficiencia eléctrica se instaló una Pizarra General de Distribución (PGD), en la cual se encuentra instalado un analizador de red que permite un mejor control de los parámetros eléctricos. Con esta medida, además, quedaron balanceadas las cargas de todos los circuitos instalados. Otra consecuencia directa del gran aumento de carga ha sido la creciente necesidad de un control minucioso del consumo diario de energía, a fin de cumplir con el plan energético.

## 1.2 Descripción del proceso actual

El energético pertenece al departamento de Servicios Administrativos y es el encargado de controlar el consumo eléctrico de la entidad. Para poder obtener la energía consumida durante el día el mismo debe dirigirse a la PGD, dónde se encuentra instalado el analizador de red, y realizar la lectura manualmente, ya que no se cuenta con una herramienta informática que permita el monitoreo de ese dispositivo desde una PC.

Este proceso resulta tedioso, ya que para obtener un valor confiable del consumo diario se debe garantizar la lectura de la energía todos los días a la misma hora, sin tener en cuenta que se pueden cometer errores tanto a la hora de capturar la energía o de calcular el consumo. Los datos adquiridos son procesados en una hoja de cálculo de Excel, los partes energético son confeccionados manualmente y enviados por correo electrónico a la Empresa Eléctrica, como se muestra en la Figura 1.2 Diagrama de proceso.

Esta problemática motivó la investigación actual, con una propuesta de sistema para mejorar las condiciones de trabajo del energético.

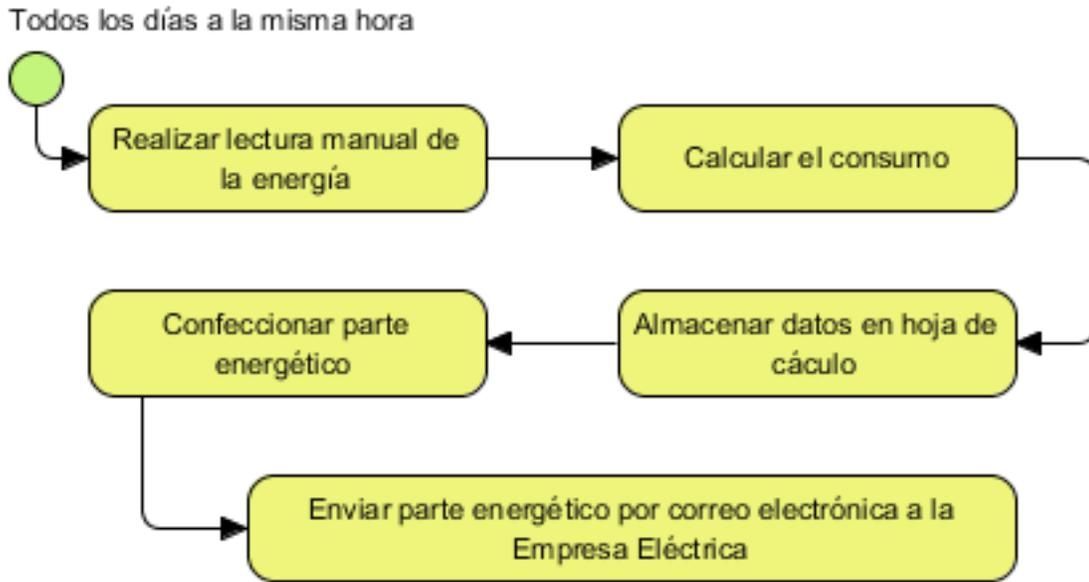


Figura 1.2 Diagrama de proceso

### 1.3 Soluciones integrales existentes

Como parte de esta investigación se procedió a realizar una búsqueda en internet de sistemas informáticos relacionados con la supervisión automática de redes eléctricas, obteniéndose los siguientes resultados:

**CIRCUTOR** dispone de diferentes tipos de software según la aplicación y la necesidad del cliente, desde el software de programación hasta el de tratamiento de datos, diseñados para la operación con los contadores de la serie CIRWATT.

- PowerWattes: Software para la tele medida y la gestión de los contadores CIRWATT.
- SoftWattes: Programa orientado a compañías eléctricas, distribuidoras, comerciantes de energía y usuarios con autorización para programar contadores para facturación de energía.
- PowerPLCes: Software de gestión y control para los contadores con comunicación PLC.
- PowerStudio: Software de control enfocado a la eficiencia energética para la gestión de los productos de CIRCUTOR.

**Schneider Electric** ha desarrollado una serie de productos PowerLogic destinados al análisis, gestión y supervisión de redes eléctricas.

- La familia System Manager™3000 Software (SMS) de PowerLogic constituye un sistema integrado de control de la energía eléctrica, que ofrece gran variedad de funciones para gestionar y analizar el sistema de control más fácilmente.

Los sistemas integrales relacionados anteriormente brindan numerosas prestaciones, por lo que pueden considerarse como opciones muy tentativas. Sin embargo, todos ellos son productos propietarios, que requieren la compra de licencias para su uso. Por ser de código cerrado, tampoco permiten la reutilización de determinados componentes para aumentar sus funcionalidades en correspondencia con las necesidades de una empresa cubana. Además, si bien ofrecen numerosas características, la mayoría de estas sobrepasan o no están relacionadas directamente con las necesidades del proceso que se quiere automatizar, en el contexto cubano, por lo que tampoco justifican los costos de adquisición de las licencias correspondientes.

Por los elementos antes expuestos quedó confirmada la necesidad de continuar con la investigación, para la creación de un sistema informático para la supervisión automática y gestión de parámetros eléctricos en BANDEC.

## 1.4 Conceptos asociados al dominio del problema

A continuación se presentan algunos conceptos asociados al dominio del problema, cuya comprensión resulta necesaria para el posterior entendimiento de la solución propuesta.

### 1.4.1 Definición de los parámetros supervisados

- Corriente por fase y trifásica: Medición en amperios de la intensidad  $rms^1$  de cada una de las tres fases del circuito [4].
- Tensión entre fases y tensión entre fase y neutro: Es la medición de las tensiones nominales de fase a fase del circuito y, de fase a neutro [4].
- Demanda: Valor medio de una variable, como potencia, a lo largo de un intervalo de tiempo determinado [4].
- Potencia activa: Cálculo de la potencia activa (calculados el total de las 3 fases y la potencia activa por fase) para obtener kilovatios (kW) [5].

---

<sup>1</sup> (rms) media cuadrática (valor eficaz). Los analizadores de red son dispositivos sensibles a la rms real.

- **Potencia reactiva:** Se entiende por energía reactiva aquella energía adicional a la activa, consumida por los elementos de una instalación eléctrica, que no se convierte en trabajo útil. Es producto del desfase existente entre la intensidad y la tensión y se cuantifica mediante el factor de potencia. Esta energía es especialmente significativa en instalaciones del sector terciario e industrial [1].
- **Potencia aparente:** Es la que resulta de considerar la tensión aplicada y la corriente consumida, esta potencia es lo que limita la utilización de transformadores, líneas de alimentación y demás elementos componentes de los circuitos eléctricos [6].
- **Factor de potencia:** Es la relación entre la potencia activa y la potencia aparente usando el contenido completo de armónicos de potencia activa y aparente. Se calcula dividiendo vatios entre voltamperios. El factor de potencia es la diferencia existente entre la potencia total que la compañía eléctrica proporciona y la parte de la potencia total que realiza un trabajo útil. Se trata del grado de desfase de la tensión y la intensidad de una carga [5, 7].
- **Energía activa:** Los receptores eléctricos alimentados por corriente eléctrica transforman la energía eléctrica en trabajo mecánico y en calor. A este efecto útil se le denomina “energía activa” y se mide en kW. Los receptores formados por resistencias puras (aparatos de calefacción, lámparas incandescentes, etc.) consumen, exclusivamente, este tipo de energía [7].
- **Energía reactiva:** existen numerosos receptores, tales como motores, transformadores, reactancias, etc., que para funcionar necesitan que se formen campos magnéticos. Estos equipos, en general inductivos, absorben energía de la red para crear los campos magnéticos y la devuelven mientras desaparecen. Con este intercambio de energía, se provoca un consumo suplementario que no es aprovechable por los receptores. A esta energía se le denomina “energía reactiva” y se mide en kVArh. La energía reactiva provoca una sobrecarga en líneas, transformadores y generadores, sin llegar a producir un rendimiento útil. Sin embargo, la factura de energía sí la contabiliza, por lo que puede llegar a incrementarla en cantidades importantes [7, 8].
- **Frecuencia:** Número de ciclos en un segundo [4], constituye un fenómeno físico que se repite cíclicamente un número determinado de veces durante un segundo de

tiempo y puede abarcar desde uno hasta millones de ciclos por segundo o Hertz (Hz) [9].

### **1.4.2 Lectura de medición automática**

La lectura de medición automática es la tecnología de medición de los diferentes parámetros eléctricos, junto con el diagnóstico y datos de estado del contador eléctrico, así como las transferencias de dichos valores a una base de datos alojada en un ordenador para su posterior análisis, todo esto de forma automatizada [5].

### **1.4.3 Analizador de Red**

Un analizador de red es un dispositivo multifuncional de instrumentación digital, adquisición de datos y control. Puede sustituir distintos medidores, relés, transductores y otros componentes. Está equipado con comunicaciones RS485 para su integración en cualquier sistema de control y supervisión de potencia y se puede instalar en varios lugares de un edificio. Son capaces de medir con una precisión muy elevada altas cargas no lineales, brindando información sobre el consumo de electricidad en subsistemas y el factor de potencia en tiempo real [5, 10].

En el mercado existen diferentes tipos de analizadores, como: PowerLogic, Power View, Masterpact, ION/PMs, SEPAM 20/40.

El analizador de red instalado en BANDEC pertenece a la familia de los PowerLogic de Schneider Electric, más específicamente, PM710.

#### ***1.4.3.1 Características del analizador de red PowerLogic PM710***

El analizador de red PowerLogic PM710 de Schneider Electric mide intensidades y tensiones, y registra en tiempo real los valores de las intensidades de las tres fases y de neutro. De igual forma calcula el factor de potencia, la potencia activa, la potencia reactiva, entre otros muchos parámetros. [4].

Este equipo usa el puerto de comunicación RS-485 de 2 hilos, implementando el protocolo de comunicación MODBUS-RTU, con dirección de esclavo de 1 a 247 y triple estado de paridad (ninguna, par, impar).



Figura 1.3 Analizador de red PowerLogic PM710 [4]

#### 1.4.4 Sistemas de Supervisión y Control por Computadoras (SCADA<sup>2</sup>)

Un sistema de supervisión y control por computadores constituye un software especialmente diseñado para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controlando el proceso de forma automática desde la pantalla del ordenador.

Los programas necesarios junto con el hardware adicional que se necesite se denominan de manera general sistema SCADA [11]. Estos sistemas son muy amplios y están destinados al control del proceso industrial. Por su complejidad y costo de producción, la mayoría son propietarios.

#### 1.4.5 Protocolos de comunicaciones

Un protocolo de comunicación es el lenguaje con el cual se comunican diversos instrumentos, ya sea entre sí o con el controlador. Un protocolo, por lo general, se divide en tramas que organizan de forma lógica la información transferida entre los instrumentos [12].

Las tramas también se dividen, como mínimo, en tres subdivisiones: cabecera, cuerpo y cola, cada una con información básica requerida para una transmisión correcta de los datos, y pueden incluir chequeo de redundancia para la detección y posible corrección de errores (Figura 1.4 Detalle de una trama [12]).

De manera general la cabecera contiene información de identificación del protocolo, el cuerpo contiene los datos principales a transmitir (en el ejemplo: identificación del esclavo, operación a realizar, datos adicionales, y código de error) y la cola contiene información complementaria [12].

<sup>2</sup> SCADA: Viene de las siglas de "Supervisory Control And Data Acquisition", es decir: Adquisición de Datos y Control de Supervisión o Control de los Procesos Industriales.

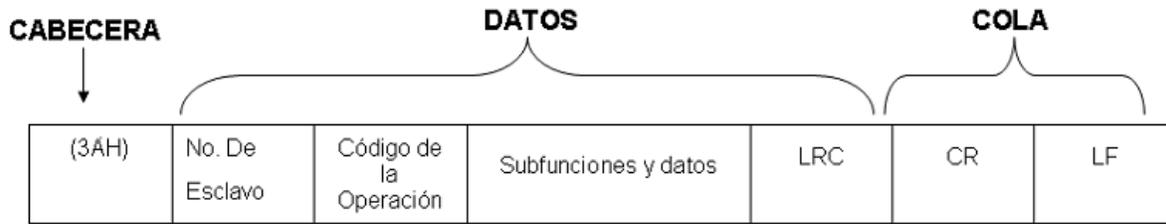


Figura 1.4 Detalle de una trama [12]

### 1.4.6 Protocolo de comunicación MODBUS

MODBUS es el protocolo de comunicación más extendido en entornos industriales y sistemas de telecontrol y monitorización. Tiene como objetivo la transmisión de información entre distintos equipos electrónicos conectados a un mismo bus<sup>3</sup>, existiendo en dicho bus un solo dispositivo maestro (MASTER) y varios equipos esclavos (SLAVES) conectados.

En su origen la conectividad era realizada a través de líneas series como pueden ser RS-232<sup>4</sup> o RS-485<sup>5</sup>, aunque en la actualidad han aparecido variantes como el MODBUS TCP, que permite el encapsulamiento del MODBUS serie en tramas Ethernet TCP/IP. Esto es posible porque desde un punto de vista del modelo OSI<sup>6</sup>, MODBUS puede ubicarse en la capa de aplicación [13] [14] [15].

#### 1.4.6.1 Comunicación Maestro – Esclavo

MODBUS siempre funciona con un maestro y uno o varios esclavos, siendo el maestro quien controla el inicio de la comunicación enviando un paquete de información a todos los esclavos con su dirección. El esclavo elegido responde la petición enviando también un paquete. Cada esclavo debe tener una dirección única en el intervalo de 1 hasta 247 dentro de una misma red. Esta limitación está dada porque en una trama MODBUS la dirección del esclavo se representa con un solo byte, existiendo algunas direcciones reservadas para propósitos específicos como *broadcast*<sup>7</sup>. [13, 16].

<sup>3</sup> BUS: La función del bus es la de permitir la conexión lógica entre distintos subsistemas de un sistema digital, enviando datos a entre dispositivos.

<sup>4</sup> RS-232: **Recommended Standard 232**: Es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre un equipo terminal de datos y un equipo de comunicación de datos.

<sup>5</sup> RS-485: Es un estándar de comunicaciones en bus de la capa física del Modelo OSI

<sup>6</sup> OSI: **Open System Interconnection**: Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones

<sup>7</sup> Broadcast: Es una forma de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

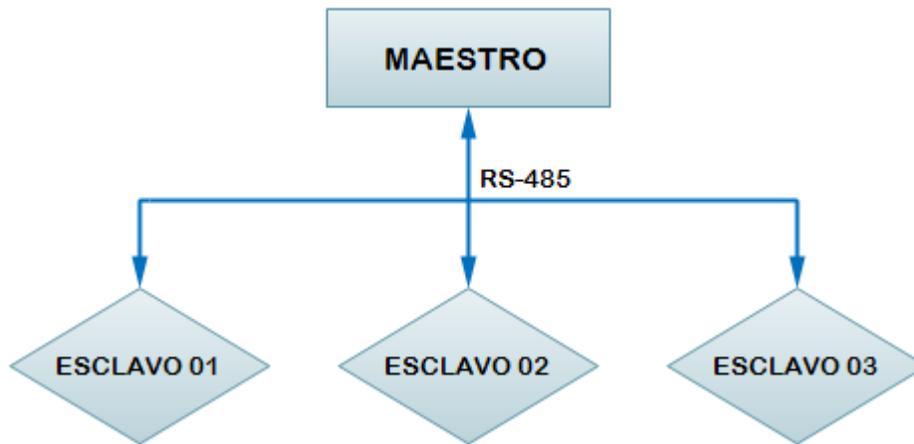


Figura 1.5 Comunicación Maestro - Esclavo

#### 1.4.6.2 Modos de transmisión MODBUS

En el protocolo MODBUS están definidos dos modos de transmisión: ASCII<sup>8</sup> y RTU<sup>9</sup>. Los modos definen cómo son transmitidos los bytes del mensaje entre maestros y esclavos. En una misma red no es posible utilizar los dos modos de transmisión [14, 16].

- MODBUS ASCII: La comunicación entre dispositivos se hace mediante caracteres ASCII.
- MODBUS RTU: La comunicación entre dispositivos se realiza mediante datos binarios sin ningún tipo de conversión, esta es la opción más usada del protocolo y la que se implementó en la solución propuesta.

Existe una versión MODBUS TCP/IP que es muy semejante al formato RTU, con la diferencia de que encapsula las transmisiones en paquetes TCP/IP.

#### 1.4.6.3 Estructura de los mensajes en el MODBUS RTU

Toda la comunicación se inicia cuando el maestro hace una petición a un esclavo y este contesta lo que fue solicitado, en ambas direcciones las tramas tienen la misma estructura [17]. (Figura 1.6 Trama RTU de solicitud [17] y Figura 1.7 Trama RTU de respuesta [17])

Dirección (1 byte)	Función (1 byte)	Datos de la solicitud (n bytes)	CRC (2 bytes)
-----------------------	---------------------	------------------------------------	------------------

Figura 1.6 Trama RTU de solicitud [17]

<sup>8</sup> (ASCII) **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange: Es un código de caracteres basado en el alfabeto latino con el objetivo de representarlos numéricamente.

<sup>9</sup> (RTU) **R**emote **T**erminal **U**nit: Define a un dispositivo basado en microprocesadores, el cual permite obtener señales independientes de los procesos y enviar la información a un sitio remoto donde se procese.

Dirección (1 byte)	Función (1 byte)	Datos de la respuesta (n bytes)	CRC (2 bytes)
-----------------------	---------------------	------------------------------------	------------------

Figura 1.7 Trama RTU de respuesta [17]

- Dirección: El maestro inicia la comunicación enviando un byte con la dirección del esclavo al cual se destina el mensaje; el esclavo también inicia la trama con su propia dirección.
- Función: El maestro especifica en un byte el tipo de servicio o función solicitada al esclavo (lectura, escritura, etc.). Cada función se utiliza para acceder a un tipo específico de dato.
- Campo de Datos: Campo con tamaño variable, cuyo formato y contenido depende de la función.
- CRC<sup>10</sup>: La última parte de la trama es el campo para el chequeo de errores de transmisión. Este campo está formado por 2 bytes, y se organiza primero el byte menos significativo (CRC-) y después el más significativo (CRC+).

#### 1.4.6.4 Funciones disponibles en MODBUS RTU

En la especificación del protocolo MODBUS RTU son definidas funciones para acceder a diferentes tipos de datos, algunas de estas funciones son las siguientes. [14, 17].

- Read Holding Registers: Lectura de bloque de registradores de retención. Código 03.
- Read Input Registers: Lectura de bloque de registradores de entrada. Código 04
- Write Single Registers: Prestablecer registros individuales.

En la solución propuesta la función utilizada para leer las variables eléctricas es la que permite leer los registros de retención.

#### 1.4.7 NPort 5230

El NPort 5230 es un conversor que permite conectar dispositivos serial como PLC, analizadores de red o sensores, a una red LAN a través del protocolo TCP/IP, haciendo posible acceder a ellos desde cualquier punto de la red LAN o internet [18].

<sup>10</sup> **Cycling Redundancy Check**: Es un código de detección de errores usado frecuentemente en redes digitales y en dispositivos de almacenamiento para detectar cambios accidentales en los datos.

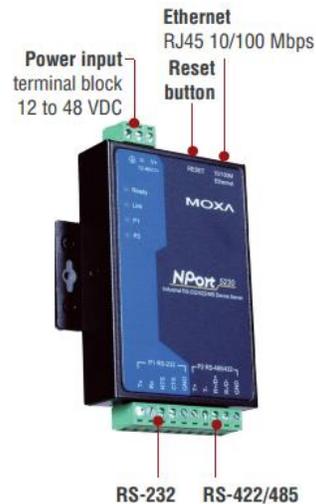


Figura 1.8 Serial to Ethernet Device Server NPort 5230 [19]

#### 1.4.7.1 Modo de conexión COM Real

Para completar el esquema de conexión se instala un driver que establece una conexión transparente entre el host y el dispositivo serie (en este caso el analizador de red) mapeando la dirección IP del NPort con el puerto serie local de la PC. Este modo permite hasta 4 conexiones simultaneas, permitiendo que varios host puedan recibir información del mismo dispositivo serie simultáneamente [18].

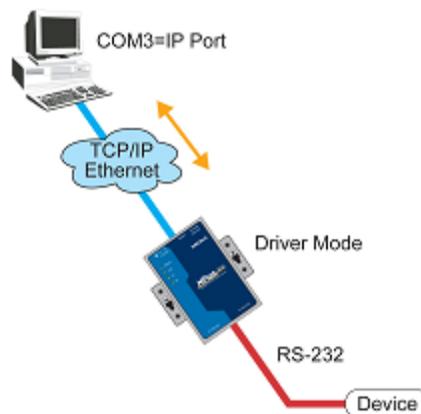


Figura 1.9 Modo de conexión COM Real [18]

#### 1.4.8 Servicio de Windows

Un servicio de Windows consiste en uno o varios procesos iniciados por el sistema operativo en su arranque (si así ha sido configurado) o de forma manual. El usuario, en una situación normal, no los inicia ni los detiene, delegando al sistema operativo la realización de estas tareas de forma automática. Los procesos asociados se ejecutan en segundo plano sin

interacción directa con el usuario. No están asociados a ninguna interfaz gráfica y suelen ser aplicaciones que realizan tareas que no requieren de la intervención del usuario [20].

## 1.5 Tendencias y tecnologías a usar para el desarrollo del sistema informático

A continuación se describen las principales tendencias y tecnologías utilizadas durante la investigación, partiendo de la metodología de desarrollo de software empleada hasta llegar a las tecnologías implicadas directamente en la implementación del sistema.

### 1.5.1 Metodología de desarrollo de software

Una metodología de desarrollo se define como un conjunto de pasos y procedimientos que permiten el desarrollo de un software. Esta idea se traduce en las fases y etapas en las que se van obteniendo los componentes de software que dan lugar a un producto final, económicamente factible. Cada una de las tareas a desarrollar requiere de un grupo de técnicas y herramientas que garanticen un adecuado análisis, diseño e implementación del software, para lo cual es necesario saber en qué momento se debe realizar cada una de ellas, qué persona es la responsable de su cumplimiento, y las vías que se necesitan para su realización. Por esta razón se puede decir que en un proyecto de desarrollo de software la metodología define “quién está haciendo qué, cuándo y cómo debe hacerlo” [21].

El desarrollo de software es un proceso que no puede ser conducido de la misma manera en todo tipo de proyectos, por lo que debe emplearse una metodología apropiada de acuerdo con las características de cada proyecto en particular. Por esta razón, entre las distintas metodologías de desarrollo de la actualidad, como Rational Unified Process (RUP), Extreme Programming (XP), Agile Unified Process (AUP) e ICONIX, se selecciona AUP como la más conveniente para la realización del sistema informático para la extracción automática y gestión de parámetros eléctricos.

#### 1.5.1.1 Agile Unified Process (AUP)

Proceso Unificado Ágil (AUP por sus siglas en inglés) se describe como una metodología fácil de entender para el desarrollo de software de negocios, utilizando técnicas sencillas y conceptos fieles a RUP, por lo que constituye una versión simplificada de este [22-24].

El proceso AUP establece un modelo más simple que el que aparece en RUP, por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y

Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP [25].

### 1.5.1.2 Fases de AUP

Al igual que RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que terminan con hitos claros alcanzados:

- Inicio (Concepción): El objetivo de esta fase es obtener una comprensión común cliente - equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en la validación de la arquitectura.
- Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- Transición: el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación, y finalmente se despliega en los sistemas de producción.

### 1.5.1.3 Disciplinas de AUP

Las disciplinas se llevan a cabo de manera sistemática a la definición de las actividades que realizan los miembros del equipo de desarrollo a fin de desarrollar, validar, y entregar el software de trabajo que responda a las necesidades de sus interlocutores. Las disciplinas son [25]:

1. Modelo. El objetivo de esta disciplina es entender el negocio de la organización y el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio.
2. Implementación. El objetivo de esta disciplina es transformar el o los modelos en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.
3. Prueba. El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.

4. Despliegue. El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo esté a disposición de los usuarios finales.
5. Gestión de configuración. El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.
6. Gestión de proyectos. El objetivo de esta disciplina es dirigir las actividades que se llevan a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc.), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.
7. Entorno. El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (hardware, software, etc.) están disponibles para el equipo según sea necesario.

#### **1.5.1.4 Principios de AUP**

- El personal sabe lo que está haciendo. La gente no va a leer un detallado proceso de documentación, pero algunos quieren una orientación de alto nivel y/o formación de vez en cuando. AUP proporciona enlaces a muchos de los detalles para aquellos que estén interesados, pero no obliga a aquellos que no lo deseen.
- Simplicidad. Todo se describe concisamente utilizando pocas páginas, no miles de ellas.
- Agilidad. Ajuste a los valores y principios de la Alianza Ágil.
- Centrarse en actividades de alto valor. La atención se centra en las actividades que se identifican como esenciales para el de desarrollo, y no en todas las actividades que forman parte del proyecto.
- Independencia de herramientas. El conjunto de herramientas a utilizar es de libre decisión. Es aconsejable utilizar las herramientas que son más adecuadas para el trabajo, que a menudo son las más simples o incluso de código abierto. [26]

## 1.5.2 Lenguaje de modelado

Un lenguaje de modelado es una serie de símbolos y modos de utilizarlos para modelar un diseño de software orientado a objetos. Algunas entidades los usan combinándolos con una metodología de desarrollo de software para lograr de una especificación inicial, un plan de implementación, y que el mismo sirva de puente de comunicación con un equipo de desarrollo.

### 1.5.2.1 Lenguaje de Modelado Unificado

El Lenguaje Unificado de Modelado (UML) surge con el objetivo de comunicar ideas a otros desarrolladores, y para servir de apoyo en los procesos de análisis de un problema. UML se ha convertido en un estándar popular para representar y modelar la información con la que se trabaja en las fases de análisis y diseño [27, 28].

Este lenguaje mejora los tiempos totales de desarrollo, facilitando el modelado de sistemas mediante conceptos orientados a objetos. Posibilita establecer conceptos y artefactos ejecutables y encamina el desarrollo de forma escalable en sistemas complejos. También brinda un mejor soporte a la planeación y al control de proyectos, permitiendo una alta reutilización y minimización de costos [29].

## 1.5.3 Herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering o Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras [30].

Desde que se crearon estas herramientas (1984) hasta la actualidad, las CASE cuentan con una amplia credibilidad y exactitud, y por ende con un gran reconocimiento universal, siendo usadas por cualquier desarrollador o programador que busca un resultado óptimo y eficiente. Estas aplicaciones informáticas permiten representar unidades y procesos que estarán presentes en toda la realización del software [30].

Las herramientas CASE son muy utilizadas por los ingenieros y programadores de software, entre ellas ArgoUML, BoUML, Umbrello, Rational Rose Enterprise, Enterprise Architect y Visual Paradigm for UML.

### 1.5.3.1 *Visual Paradigm for UML*

Con el objetivo de mejorar los procesos de modelado, disminuir el tiempo de producción, facilitar la generación automática de la documentación y la reutilización de componentes, se ha determinado que la herramienta CASE a utilizar es Visual Paradigm for UML, decisión influenciada por la posibilidad que ofrece la versión Community con su licencia gratuita.

Otras características que ratifican el uso de la herramienta son su capacidad de trabajo multiplataforma y colaborativo, diseñado tanto para el sistema operativo Windows como para las diferentes distribuciones de Linux, facilitando la importación y exportación de componentes, ya sea como imagen o como archivos XML. Además, proporciona la integración con diversos entornos de desarrollo integrados, como Eclipse.

Visual Paradigm es una herramienta de modelado que utiliza UML, permite la generación de códigos e ingeniería inversa. Con un diseño de clases bien especificado, Visual Paradigm puede generar código hasta en quince lenguajes de programación, entre ellos PHP, Java, C++, XML Schema, .NET exe/dll y CORBA IDL [31].

Visual Paradigm actualmente cumple con las políticas de migración a Software Libre en Cuba, ya que es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows. Su interfaz es muy intuitiva, de fácil aprendizaje para los desarrolladores. Permite la generación automática de diagramas a partir de descripciones de casos de usos, por ejemplo, diagramas de secuencia, permitiendo la agilidad en el trabajo del analista.

A manera de resumen, Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.

- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDE.
- Disponibilidad en múltiples plataformas.

#### 1.5.4 Sistema gestor de base de datos

Un sistema gestor de bases de datos (SGBD) es una colección de programas cuyo objetivo es permitir la creación y mantenimiento de bases de datos. Si bien no es imprescindible contar con un SGBD para implementar una base de datos, este software de uso general facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones [32-34].

Los SGBD tienen como objetivo eliminar la inconsistencia de los datos, compartir dichos datos entre distintas aplicaciones con diferentes necesidades, ahorrar espacio al reducir considerablemente la redundancia, mejorar la seguridad de los datos, crear entornos de alta disponibilidad y mejorar servicios de copia de seguridad y recuperación de fallos [30, 35].

##### 1.5.4.1 SQL Server 2005

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft, basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft ante potentes sistemas gestores de bases de datos como Oracle, Sybase ASE, PostgreSQL [36, 37].

SQL Server 2005, en comparación con sus predecesores, realiza mejoras importantes en el modelo de seguridad de la plataforma de base de datos, con la intención de ofrecer un control más preciso y flexible que permite una seguridad mayor de los datos.

Además, puede desarrollar servicios Web XML en el nivel de la base de datos, convirtiendo SQL Server en un detector HTTP. De este modo, se proporciona la capacidad de acceso a datos para aplicaciones centralizadas en los servicios Web. En SQL Server 2005 se puede utilizar HTTP para tener acceso directamente a SQL Server, sin utilizar un detector de nivel medio como los servicios de Internet Information Server (IIS) de Microsoft [38, 39].

A continuación se muestran algunas características de Microsoft SQL Server 2005 [36, 40]:

- Soporte de transacciones.

- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos.
- Las bases de datos no poseen restricción de tamaño.
- Soporta múltiples usuarios simultáneamente.
- Posee un buen grado de desempeño.
- Soporta un elevado tráfico de usuarios.
- Soporta la configuración de políticas de seguridad efectivas

### 1.5.5 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE por sus siglas en inglés) se define como el conjunto de herramientas creadas como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Un IDE provee un marco de trabajo amigable para la mayoría de los lenguajes de programación y en algunos casos puede funcionar como un sistema en tiempo de ejecución, donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

#### 1.5.5.1 Delphi

Delphi XE5 es un potente compilador que soporta el diseño estructurado orientado a objetos. Está basado en el lenguaje de programación Object Pascal, e incluye como beneficios la forma fácil de leer código, la compilación rápida y la utilización de múltiples archivos unit<sup>11</sup> para la programación modular. Delphi tiene características especiales que soportan el *framework* y ambiente de RAD Studio [41, 42].

---

<sup>11</sup> Este archivo tiene extensión .pas y contiene la mayor parte del código de una aplicación. Una aplicación puede tener varios archivos unit.pas

### 1.5.5.2 *NetBeans*

NetBeans es un IDE de desarrollo que se adapta a un gran número de lenguajes de programación. Funciona en los sistemas operativos compatibles con la máquina virtual de Java. Contiene un grupo de características importantes como es el caso de la integración con PHP.

NetBeans 7.2 provee de una estructura para los proyectos. El editor integra lenguajes como JavaScript, HTML y CSS. Posee un sistema de escaneo de todos los proyectos que se crean o se cargan, para así reconocer funcionalidades, clases, y proporcionar un completamiento de código eficiente, que aumenta la velocidad de elaboración de aplicaciones.

Presenta una sencilla integración con Symfony y Zen Framework. Además elimina la cantidad de comandos de consola necesarios para instalar y configurar Symfony. Posee un robusto y ágil editor de código fuente, sobre todo el de PHP, ya que provee todas las ventajas que presenta PHP 5.3.

Además, utiliza Xdebug, posibilitando examinar cada variable local, establecer puntos de interrupción y evaluar el código. Ofrece líneas de comandos para la depuración posibilitando inspeccionar el código HTML sin tener que cambiar a un navegador.

NetBeans es un IDE gratuito y de código abierto. Presenta todas las herramientas para elaborar no solo aplicaciones en un entorno web sino también aplicaciones de escritorio, móviles y empresariales. Incorpora diferentes lenguajes como C/C++, Java, Ruby y sus variantes tales como JRuby y Ruby on Rails [43-45].

### 1.5.6 **Lenguaje de Programación**

Los lenguajes de programación se crearon con el objetivo de permitir a las personas darles instrucciones a los equipos, describiendo un conjunto de acciones que estos deben cumplir. Dichas instrucciones están compuestas por un grupo de reglas gramaticales, términos y símbolos que le permiten al desarrollador comunicarse con el equipo y crear programas.

#### 1.5.6.1 *Object Pascal*

Object Pascal es una evolución del lenguaje de programación Pascal que incluye los elementos inherentes al paradigmático modelo de programación orientada a objetos de forma asequible, gracias a una sintaxis intuitiva y consolidada a la par de los lenguajes

interpretados como Python o Basic. Brinda además la posibilidad de integración del lenguaje ensamblador y notaciones en el código Pascal, el soporte para múltiples arquitecturas.

La potencia de otros lenguajes compilados de similar alcance a C++ y brinda la gran ventaja de que las aplicaciones generadas no sólo son multiplataforma, sino que también pueden ser compiladas al código nativo de las diferentes plataformas soportadas, lo que se traduce en una velocidad de ejecución muy elevada y siempre superior a la de los lenguajes interpretados anteriormente mencionados [46-48].

### **1.5.6.2 PHP**

Personal Home Page (PHP) es un lenguaje de programación del lado del servidor utilizado en aplicaciones web. Está ampliamente difundido gracias a su código abierto y no causa costos extras en cuanto al software del servidor. Se utiliza principalmente para la elaboración de sitios web grandes que tengan una fuerte administración y consultas de datos. Esto es posible debido a que es un lenguaje fácil y eficiente.

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles y funciones. No es un lenguaje de marcas como HTML [49].

A diferencia de JavaScript que se ejecuta en el navegador, PHP se ejecuta en el servidor, por lo que permite acceder a los recursos que tenga el servidor, como una base de datos. Un programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML [49].

La mayoría de los Sistemas de Administración de Contenidos (CMS) de código abierto están desarrollados con este lenguaje. Presenta una sintaxis flexible y es sencillo de utilizar, además de brindar la posibilidad de conectarse a la mayoría de los gestores de base de datos. Incluye la implementación de muchas funciones que agilizan su interpretación y no presenta tipos de datos. Es un lenguaje libre.

### **1.5.7 Servidor Web**

Un servidor web es un programa que se ejecuta continuamente, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o alguna información de acuerdo con los comandos solicitados [50].

Básicamente un servidor web consta de un intérprete HTTP el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla [51].

### 1.5.7.1 Apache

Apache es el servidor web por excelencia cuya facilidad de configuración, robustez y estabilidad hacen que millones de servicios en internet reiteren su confianza en él [52, 53]. A continuación se detallan algunas de sus principales características.

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierto.
- Es altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache. Actualmente existen muchos módulos instalables.
- Soporta lenguajes de *scripting* como Perl y PHP. También trabaja con Java y páginas JSP.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado *script* cuando ocurra un error en concreto.
- Tiene una alta capacidad de configuración en la creación y gestión de *logs*. Permite la creación de ficheros de *log* a la medida del administrador, permitiendo un mayor control sobre lo que sucede en el servidor.

### 1.5.8 Framework

Un framework es una estructura de software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que pueden añadirse las últimas piezas para construir una aplicación concreta [54].

#### 1.5.8.1 Symfony

Symfony es un framework PHP que facilita el desarrollo de aplicaciones web. Se encarga de todos los aspectos comunes y rutinarios de las aplicaciones web, dejando que el

programador se dedique a aportar el valor agregado, desarrollando las características únicas de cada proyecto [55].

Es un framework muy documentado, productivo, elegante y estable. Es fácil de instalar en diferentes sistemas operativos, funciona con todos los gestores de base de datos comunes, es ampliamente flexible y extensible mediante un sistema de *plugins*, traducido a diversos idiomas, publicado bajo licencia de software libre GNU Lesser Public General License (LGPL) y compatible con PHP5 [56].

Symfony sigue las mejores prácticas y patrones de diseño para la web. Contiene un código legible, fácil de entender, que incluye comentarios y permite un mantenimiento sencillo. Permite el uso de plantillas en la capa de presentación, validación de datos y generación dinámica de formularios, es muy potente contra las inyecciones SQL, contiene un eficiente sistema de caché, paginación automática, facilita la integración con Ajax mediante *helpers*, los cuales encapsulan código JavaScript en una sola línea de código [57].

Symfony emplea el estilo arquitectónico modelo-vista-controlador (MVC), con el objetivo de organizar las distintas partes que componen una aplicación web, separándola en tres capas principales: el modelo, la vista y el controlador. El modelo es el encargado del dominio de toda la información del sitio, darle significación a dicha información, acceder a los datos de la aplicación, además de la comunicación con la base de datos y la ejecución de consultas. La vista es la que se encarga de la interfaz que se le muestra al usuario, presentando los datos del modelo en un formato adecuado para la interacción, además decide cómo mostrar los datos suministrados por el controlador. El controlador es el que decide qué se va a mostrar y en qué momento, además es el que dirige todo el proceso de gestión y responde a las acciones de los usuarios [58].

#### 1.5.8.2 Ext JS 4

ExtJS es un conjunto de librerías JavaScript para el desarrollo de aplicaciones web iterativas, usa tecnología AJAX<sup>12</sup>, DHTML<sup>13</sup> y manipulación del DOM<sup>14</sup>. Permite realizar complejas

---

<sup>12</sup> JavaScript asíncrono y XML: Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas.

<sup>13</sup> El HTML Dinámico no es más que una forma de aportar interactividad a las páginas web, tiene la ventaja de poder crear efectos que requieren poco ancho de banda a la hora de ejecutarse y son estos efectos los que aumentan la funcionalidad a la página.

interfaces de usuario, fáciles de usar, muy parecidas a las aplicaciones de escritorio. Esto permite a los desarrolladores web concentrarse en la funcionalidad de las aplicaciones [59].

El paquete ExtJS contiene un conjunto de archivos que comprenden los recursos que necesita para correr adecuadamente, los cuales se debe copiar dentro de la carpeta principal del servidor web y referenciarse debidamente por las respuestas HTML.

## 1.6 Conclusiones parciales

En este capítulo se ha recopilado el fundamento teórico que sustenta la presente investigación, acercando al lector a la comunicación con los analizadores de red eléctrica y las tendencias y tecnologías escogidas para la implementación del producto informático, que se adaptan correctamente al marco de la entidad y a las necesidades requeridas.

El estudio realizado demostró la necesidad de realizar un sistema para la extracción automática y gestión de los parámetros eléctricos en BANDEC. Se seleccionó AUP como metodología de desarrollo, PHP, JavaScript y Object Pascal como lenguajes de programación, Symfony y ExtJS como tecnologías de desarrollo web, y Apache como servidor web junto con Microsoft SQL Server como sistema gestor de bases de datos.

---

<sup>14</sup> Document Object Model: Es una plataforma neutral que permite a los programadores scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de los documentos.

## CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se describe la solución propuesta, siguiendo las etapas que comprende la metodología AUP. Se define un modelo compuesto por los requerimientos funcionales y no funcionales que delimitan las propiedades que el sistema requiere, así como el modelo de casos de usos como guía para describir los pasos hacia la implementación del sistema. Se describe la arquitectura, se presenta el diagrama de despliegue con el objetivo de mostrar las relaciones que existen entre los componentes de hardware y de software y se realiza el diseño de la base de datos. Por último se detalla el análisis de sostenibilidad realizado a la aplicación, junto con las pruebas efectuadas al sistema para demostrar su correctitud y fiabilidad.

### 2.1 Modelo

En esta disciplina se realizan una serie de acciones encaminadas a lograr una comprensión clara del negocio, para determinar una solución viable al problema planteado.

#### 2.1.1 Requerimientos funcionales

A partir del entendimiento del negocio se extrajeron los siguientes requisitos funcionales:

- RF-1 Crear, modificar y eliminar tipo de dispositivo (nombre, desplazamiento y parámetro de consumo).
- RF-2 Crear, modificar y eliminar tipo de dato (nombre y tamaño).
- RF-3 Crear, modificar y eliminar grupo de parámetros (nombre).
- RF-4 Crear, modificar y eliminar parámetro (dirección, tipo de dispositivo, descripción, tipo de dato, unidad, escala, activo y grupo de parámetros).
- RF-5 Crear, modificar y eliminar puerto (nombre, tasa de baudios, bit de datos, bits de parada y paridad).
- RF-6 Crear, modificar y eliminar dispositivo (dirección, tipo de dispositivo, puerto, descripción del área e intervalo de lectura).
- RF-7 Crear, modificar y eliminar horario (nombre, hora de inicio y hora de finalización).
- RF-8 Crear y modificar plan mensual (año, mes, dispositivo y plan).

- RF-9 Registrar consumo energético (fecha, horario, dispositivo y consumo).
- RF-10 Registrar valor histórico de parámetro (hora, parámetro, valor).
- RF-11 Eliminar valores históricos de parámetros envejecidos.
- RF-12 Configurar parámetros globales del sistema (contraseña de acceso único, visibilidad del sitio, frecuencia de limpieza de datos históricos)
- RF-13 Capturar de forma automatizada las variables provenientes de analizadores de redes eléctricas (descritas en la sección 1.4.1).
- RF-14 Iniciar, detener o reiniciar servicio de captura automatizada de variables.
- RF-15 Graficar el comportamiento de las variables eléctricas en tiempo real.
- RF-16 Graficar el comportamiento de las variables eléctricas en un intervalo de tiempo dado.
- RF-17 Graficar el comportamiento del consumo energético en un intervalo de tiempo dado.
- RF-18 Calcular el plan de consumo diario ajustado de acuerdo con el consumo del mes hasta la fecha.
- RF-19 Generar reporte sobre la situación energética en un intervalo de tiempo dado.
- RF-20 Generar parte diario de consumo energético.
- RF-21 Exportar reportes (formato CSV, PDF).
- RF-22 Imprimir reportes.

### 2.1.2 Requerimientos no funcionales

Además de los requerimientos funcionales también se definieron los requerimientos no funcionales siguientes:

- RNF-1 El diseño debe ser sencillo e intuitivo.
- RNF-2 La interfaz debe ser amigable con colores tenues agradable a la vista.
- RNF-3 El sistema debe ejecutar sus funcionalidades con tiempos de respuestas aceptables.
- RNF-4 El sistema debe ser preciso en la captura de las variables provenientes de los analizadores de redes eléctricas.

- RNF-5 Los reportes y las funcionalidades gráficas del sistema deberán ser accesibles desde todos los departamentos de la entidad que cuenten con conectividad.
- RNF-6 El sistema deberá estar protegido contra el acceso de personal no autorizado.
- RNF-7 El sistema deberá validar la información que se procesa, de modo que se minimice la probabilidad de tener almacenados datos inconsistentes.

### 2.1.3 Casos de uso del sistema

El diseño de los casos de uso del sistema (Figura 2.1 Diagrama de Casos de Usos) partió de los requisitos funcionales extraídos del negocio. Los actores identificados fueron el Energético, que es el responsable de toda la configuración del sistema y la gestión de los planes mensuales; el Usuario, que es quien consulta las gráficas y reportes; y el Reloj, que es una entidad abstracta encargada de desencadenar todas las acciones automatizadas del sistema.

- CU-1 Gestionar tipo de dispositivo: El energético crea, modifica o elimina un tipo de dispositivo.
- CU-2 Gestionar tipo de dato: El energético crea, modifica o elimina un tipo de dato.
- CU-3 Gestionar grupo de parámetros: El energético crea, modifica o elimina un grupo de parámetros.
- CU-4 Gestionar parámetro: El energético crea, modifica o elimina un parámetro.
- CU-5 Gestionar puerto: El energético crea, modifica o elimina un puerto.
- CU-6 Gestionar dispositivo: El energético crea, modifica o elimina un dispositivo.
- CU-7 Gestionar horario: El energético crea, modifica o elimina un horario.
- CU-8 Gestionar plan mensual: El energético crea o modifica un plan mensual.
- CU-9 Capturar las variables de los analizadores de redes eléctricas: El reloj desencadena una acción de consultar los analizadores de redes eléctricas y almacenar los valores de las variables consultadas en memoria, de forma temporal.
- CU-10 Registrar datos energéticos: El reloj desencadena una acción de registrar el valor de un parámetro o el consumo energético que reporta un dispositivo.

- CU-11 Configurar parámetros globales del sistema: El energético modifica los parámetros globales del sistema (contraseña de acceso único, visibilidad del sitio para la red, etc.)
- CU-12 Controlar servicio de captura automatizada de variables: El energético inicia, detiene o reinicia el servicio automatizado de captura de variables.
- CU-13 Consultar gráficos: El usuario consulta los gráficos del comportamiento energético en tiempo real o en un intervalo de tiempo dado.
- CU-14 Generar reportes: El usuario genera reportes sobre el comportamiento energético en un intervalo de tiempo dado. Los reportes se pueden imprimir o exportar a PDF.



Figura 2.1 Diagrama de Casos de Usos

### 2.1.4 Arquitectura

Todos los componentes fueron desarrollados bajo las restricciones, suposiciones y recomendaciones del patrón arquitectónico Modelo-Vista-Controlador (MVC).

MVC sugiere la separación del software en tres componentes: Modelo, Vista y controlador, los cuales serán explicados brevemente.

**Modelo:** Es la representación de la información que maneja la aplicación. El modelo en sí lo constituyen los datos puros y la lógica de los propios datos, que puestos en el contexto del sistema proveen de información al usuario, y en algunos casos a la propia aplicación.

**Vista:** Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación web, la “Vista” sería una página HTML con contenido dinámico sobre la cual el usuario puede realizar sus operaciones. En el caso de una aplicación *desktop* son los formularios y componentes gráficos con los que interactúa el usuario.

**Controlador:** Es la parte encargada de manejar y responder las solicitudes del usuario, procesando toda la información necesaria y modificando el Modelo en caso de ser necesario.

El ciclo de vida del MVC es normalmente representado por los tres componentes presentados anteriormente y el cliente (también conocido como usuario o actor). El siguiente diagrama representa el ciclo de vida de manera sencilla:

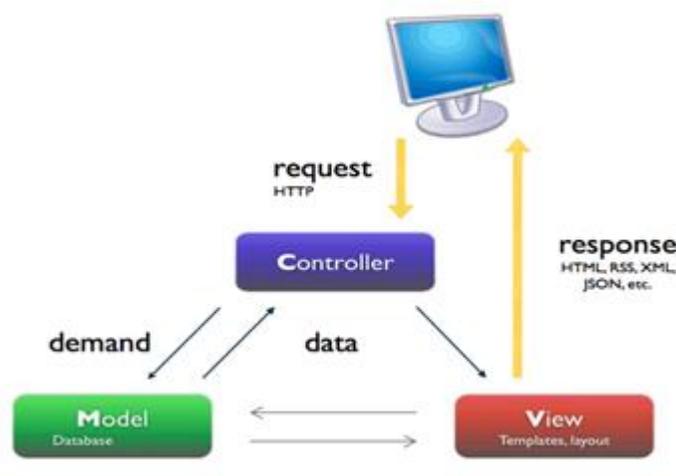


Figura 2.2 Patrón de diseño Modelo-Vista-Controlador

Este patrón se utilizó en todos los subsistemas propuestos, como se describe en el modelo arquitectónico del sistema (Figura 2.3):

- Vista (VIEW): Se encarga de gestionar todas las interfaces visuales del sistema.
- Controlador (CONTROLLER): De manera general gestiona el manejo de las entidades del sistema. En el servicio controla la extracción automatizada de parámetros eléctricos. En la aplicación de configuración del servicio y la aplicación web controla el flujo de datos entre el modelo y la vista.

- Modelo (DAO): Se encarga de leer y escribir los datos de las entidades del sistema.
- Servicio de lectura (SERVICE): Subsistema para la extracción automática de parámetros eléctricos y su almacenamiento en la base de datos.
- Aplicación de configuración del servicio (SERVICE CONFIG APP): Subsistema para la configuración del servicio de lectura.
- Aplicación web (WEB APP): Subsistema para la gestión de planes y generación de reportes.
- Symfony: Framework para optimizar el desarrollo de aplicaciones web, basado en el patrón Modelo-Vista-Controlador.
- ExtJS: Biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas, mediante peticiones web asíncronas y manipulación del DOM.

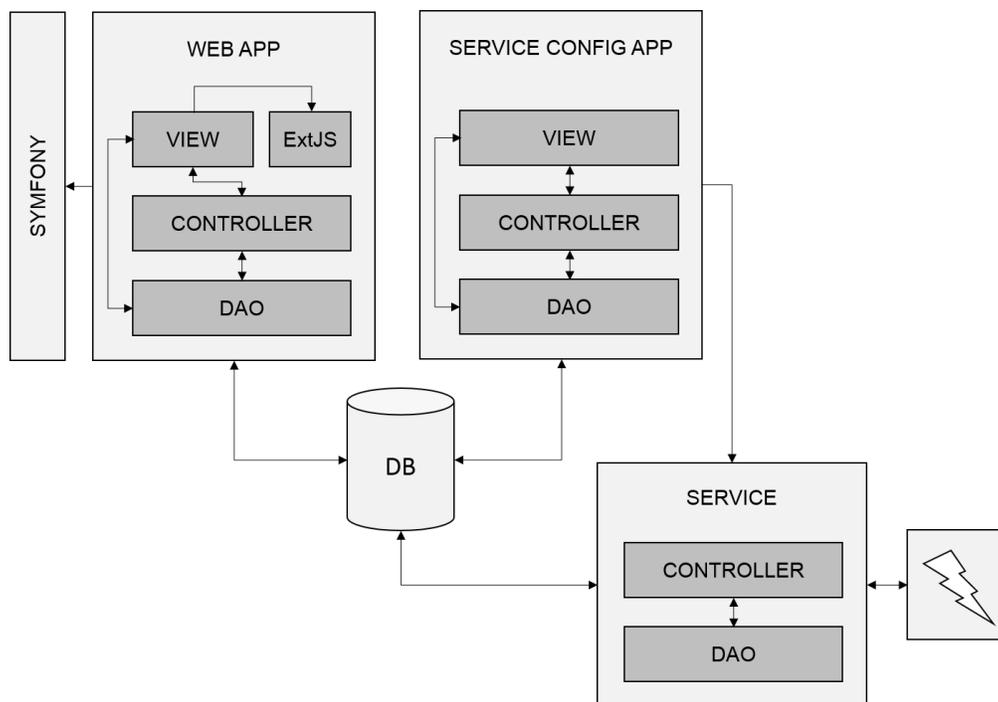


Figura 2.3 Modelo Arquitectónico del sistema

### 2.1.5 Diagrama de despliegue

El diagrama de despliegue se realiza con el objetivo de mostrar las relaciones que existen entre los componentes hardware y software de la aplicación. Está compuesto por un conjunto de nodos unidos por conexiones de comunicación (Figura 2.4 Diagrama de Despliegue).

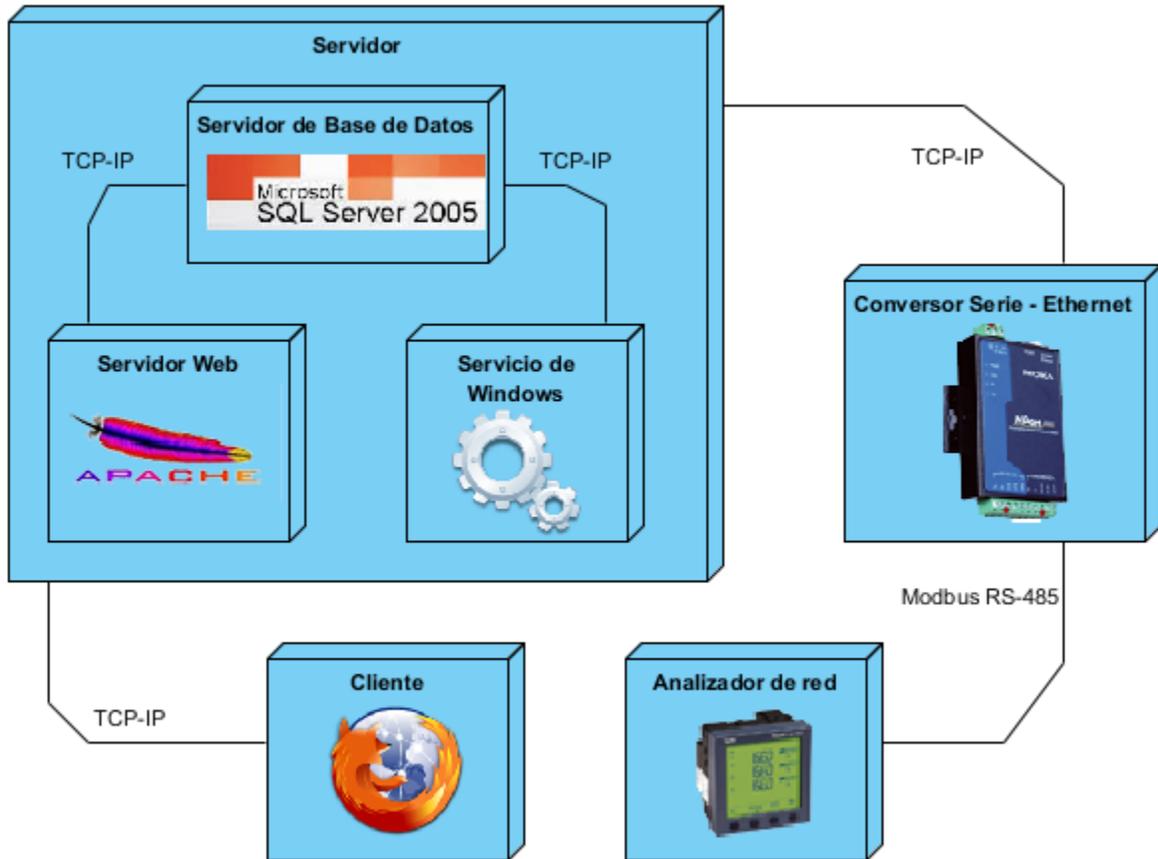


Figura 2.4 Diagrama de Despliegue

### 2.1.6 Diseño de la base de datos

El correcto funcionamiento de una aplicación también depende de un correcto diseño e implementación de la base de datos. El diagrama Entidad-Relación representa la realidad de la problemática identificada a través de las entidades y de los enlaces que rigen la unión de las mismas, y que constituyen la relación del modelo (ANEXO 3). El diagrama de clases persistentes muestra las relaciones existentes entre aquellas clases que implementan las entidades del problema de negocio, manteniendo su valor en un espacio y tiempo determinado (Figura 2.5 Diagrama de Clases Persistentes).

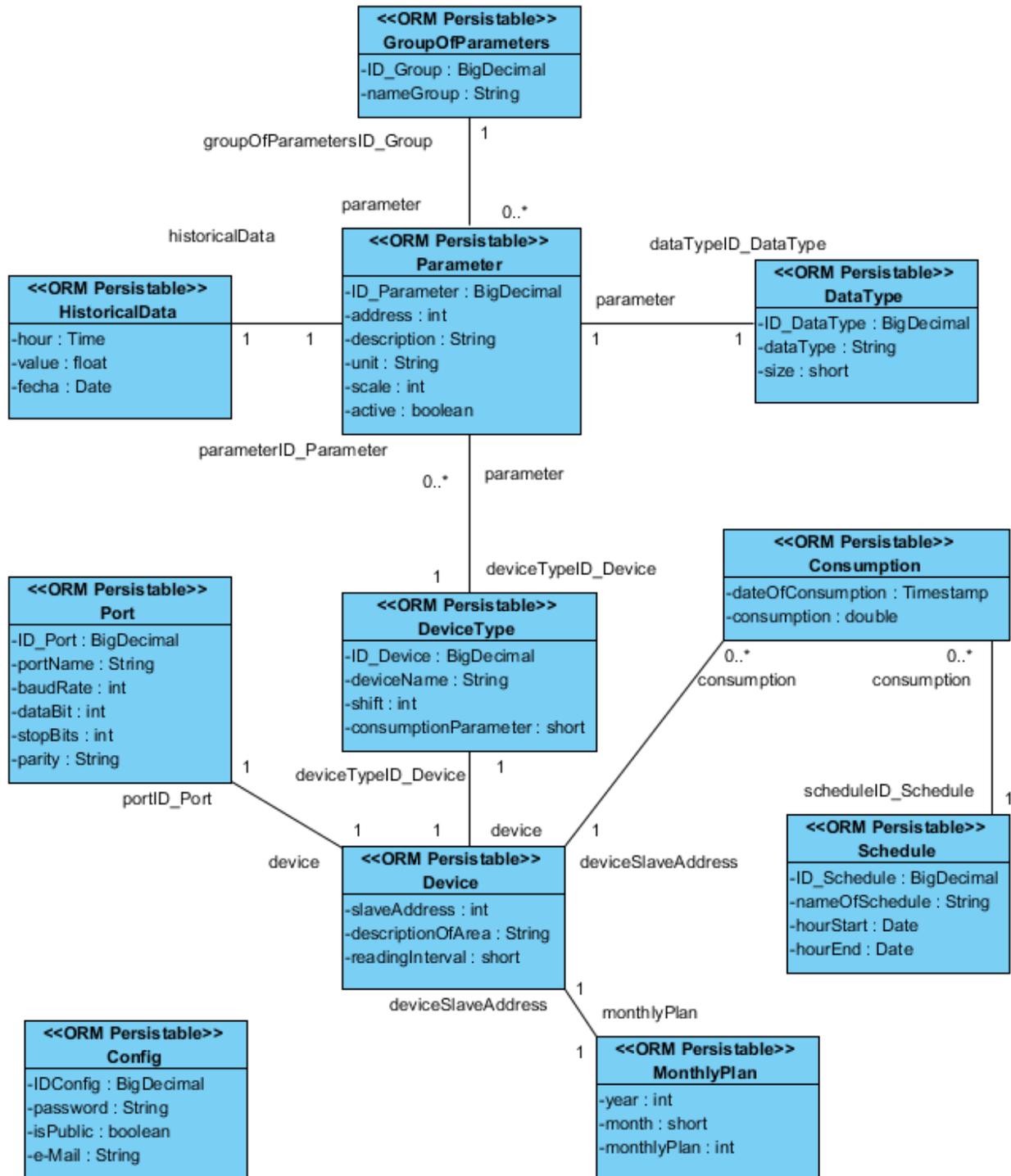


Figura 2.5 Diagrama de Clases Persistentes

## **2.2 Valoración de sostenibilidad**

Con el propósito de favorecer la extracción y gestión de los parámetros eléctricos que brinda el analizador de red en el Banco de Crédito y Comercio de Holguín, se tuvo en cuenta que el sistema informático propuesto fuera sostenible desde las dimensiones administrativa, socio – humanista, ambiental y tecnológica.

### **2.2.1 Dimensión Administrativa**

El producto informático no incurre en gastos técnicos ya que se puede implantar con el equipamiento computacional actual y emplea la red corporativa LAN como soporte de transmisión, además el analizador de red ya se encuentra instalado en la pizarra general de distribución. El uso de esta herramienta informática favorece el ahorro energético ya que se tiene un mejor control de la energía consumida y apoya la toma de decisiones por parte de la administración ya que permite identificar pérdidas de energía, desbalances y detectar factores de potencia costosamente bajos.

Por estas razones se afirma que este software es sostenible desde el punto de vista de la dimensión administrativa.

### **2.2.2 Dimensión Socio – Humanista**

El sistema informático no genera ni disminuye empleos pero si elimina el tiempo y esfuerzo físico del encargado en realizar la lectura manualmente, así como la posibilidad de cometer errores al manipular la información. El sistema puede ser desplegado a otras delegaciones del banco siempre y cuando tengan instalado un analizador de red. En el caso del energético de la provincia le facilita el trabajo ya que se elimina todo el proceso de gestión de la información.

El sistema es sostenible desde la dimensión socio – humanista.

### **2.2.3 Dimensión Ambiental**

El sistema informático propuesto contiene una interfaz de agrado para el usuario empleado colores claros y el uso de gráficas para mostrar la información evitando así interactuar con las incómodas tablas numéricas. Al atenuar el trabajo que se realizaba manual como era el acomodo y calculo en una tabla de Excel se reduce considerablemente el tiempo de operación en la computadora evitando así efectos dañinos para la salud.

Teniendo en cuenta estas razones se puede afirmar que el producto informático es sostenible en la dimensión ambiental.

#### **2.2.4 Dimensión Tecnológica**

Los usuarios que manipularán el sistema no necesitan ser capacitados informáticamente ya que estos tienen experiencia con el uso de las microcomputadoras. La entidad cuenta con la infraestructura tecnológica necesaria para la implementación y aplicación del producto informático. El código de la aplicación está debidamente comentado facilitando así un futuro mantenimiento.

Debido a lo antes expuesto es sistema es sostenible en la dimensión tecnológica.

#### **2.2.5 Conclusiones de la valoración de sostenibilidad**

Analizado el software en las dimensiones administrativas, socio-humanista, ambiental y tecnológica se llega a la conclusión de que el mismo es perdurable en el tiempo porque soluciona la necesidad social que lo origina. Esta aplicación es fácil de usar y como la empresa cuenta con todos los recursos necesarios para su desarrollo, no se originaron gastos adicionales en su creación.

### **2.3 Implementación**

Por la complejidad del sistema propuesto se decidió descomponerlo en subsistemas para lograr un mejor funcionamiento y autonomía de las partes involucradas. Estos subsistemas son, como se explicó en el modelo arquitectónico: servicio de lectura, aplicación para la configuración del servicio, y aplicación web para la consulta de datos.

#### **2.3.1 Servicio que captura datos automáticamente del analizador de red**

Para poder obtener automáticamente los parámetros del analizador de red fue necesario implementar un servicio en constante ejecución, encargado de hacer las peticiones al dispositivo y guardar los datos leídos en la base de datos. Antes de iniciarlo debe ser previamente configurado mediante la aplicación de configuración. Estos parámetros de configuración se almacenan en una base de datos.

Cuando se inicia el servicio se comprueba si ya está configurado, de no ser así el usuario debe configurarlo para que pueda continuar, luego se carga la configuración y se comunica

con el analizador de red para capturar los datos y almacenarlos en la base de datos (Figura 2.6 Diagrama de proceso de captura automática de datos)

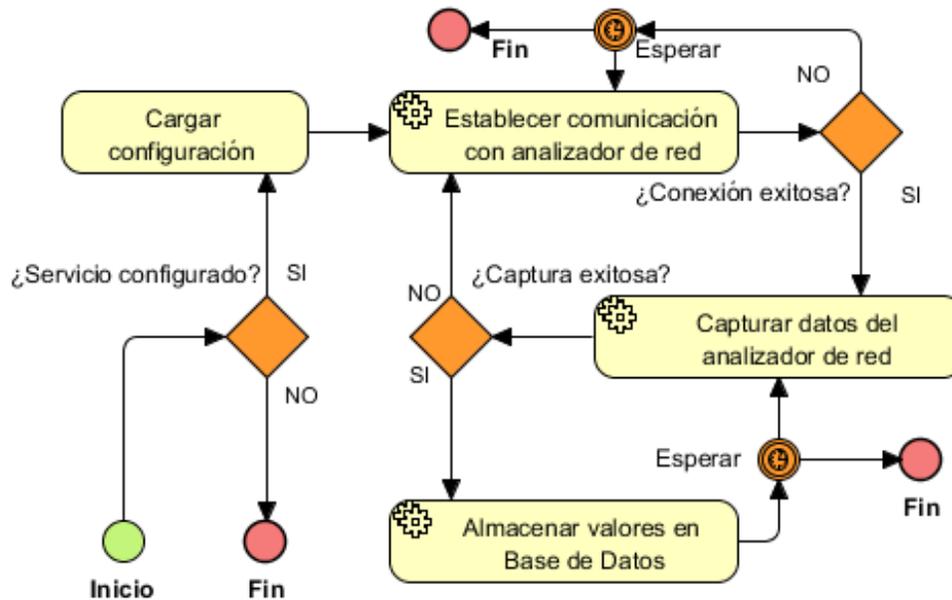


Figura 2.6 Diagrama de proceso de captura automática de datos

Este servicio tiene tres hilos de ejecución:

ReadDeviceThread: Hilo principal que se encarga de conectarse al dispositivo, hacer las peticiones y guardar en una lista de registros (temporal) los valores leídos. Este hilo se ejecuta constantemente sin limitación de tiempo manteniendo siempre actualizada la lista de registros.

ConsumptionThread: Adquiere el valor de la energía en la lista de registros según los horarios que han sido configurado, calcula el consumo por cada horario y los guarda en la base de datos.

HistorialDataThread: Adquiere el valor de todos los parámetros de la lista de registros según el intervalo de lectura configurado y los almacena en la base de datos.

Para la implementación de este servicio se utilizó el entorno de desarrollo integrado Embarcadero Delphi XE5 y el lenguaje de programación Object Pascal.

### 2.3.2 Aplicación de configuración del servicio

Esta aplicación se ejecuta en el mismo nodo donde se encuentra el servicio y es la encargada de gestionar toda la configuración que necesita el servicio para funcionar. Los principales parámetros de configuración son: tipo de dispositivo con sus respectivos parámetros, dirección esclavo del dispositivo, puerto de comunicación, horarios de consumo, intervalo de lectura, etc.

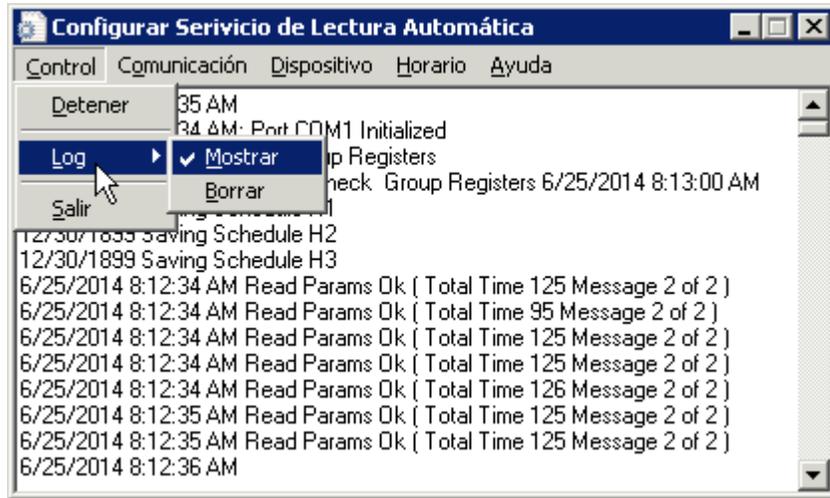


Figura 2.7 Interfaz principal de la aplicación de configuración

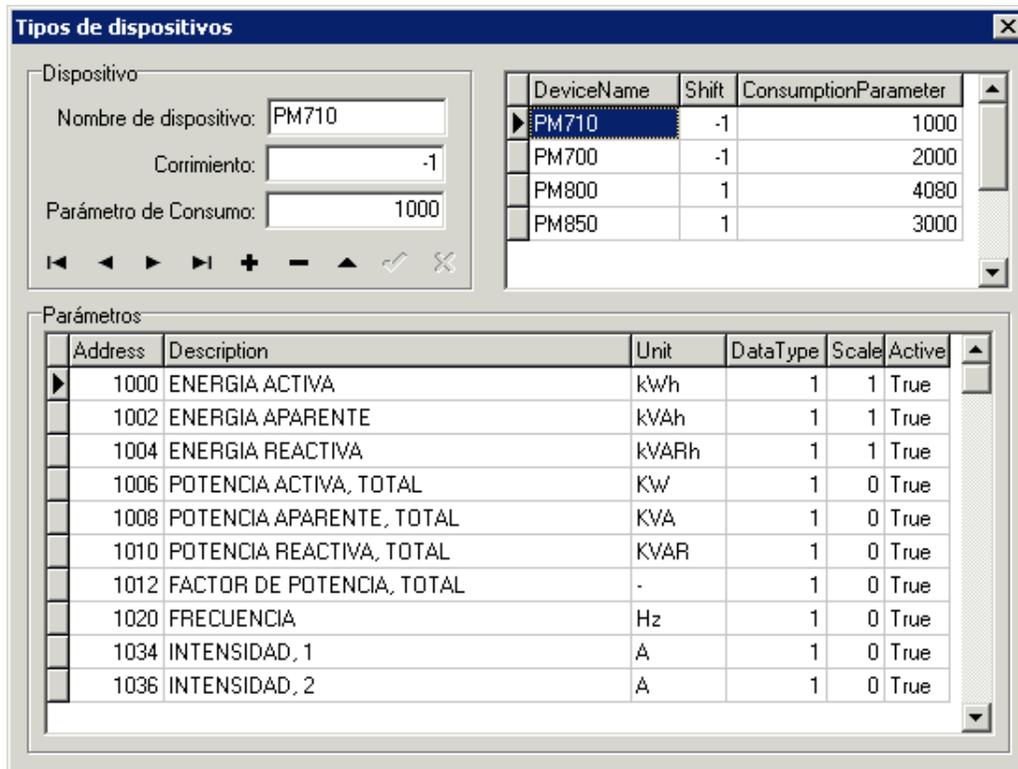


Figura 2.8 Interfaz de configuración de dispositivos

Para la implementación de esta aplicación se utilizó el entorno de desarrollo integrado Embarcadero Delphi XE5 y el lenguaje de programación Object Pascal.

### 2.3.3 Aplicación web

En esta aplicación se muestra, en forma de gráficos, toda la información adquirida del analizador de red. Permite también la generación de reportes y la gestión del plan mensual de consumo.

Inicialmente esta aplicación web es privada y solamente se puede ejecutar desde *localhost*, a partir de ese momento se configura la contraseña de acceso único que utiliza el energético para acceder al sistema desde cualquier *host* y solo en ese momento la aplicación pasa a ser pública. La visibilidad de la aplicación también es configurada mediante un parámetro, por lo que puede ser ocultada nuevamente.

Si la aplicación está en modo público, cualquier usuario de la red puede acceder a ella y utilizar todas sus opciones, excepto de la gestión del plan, que solo quedará habilitada cuando al iniciar sesión.

Para la implementación de esta aplicación se usó el entorno de desarrollo NetBeans, el lenguaje de programación PHP, el framework Symfony y la librería JavaScript ExtJS.

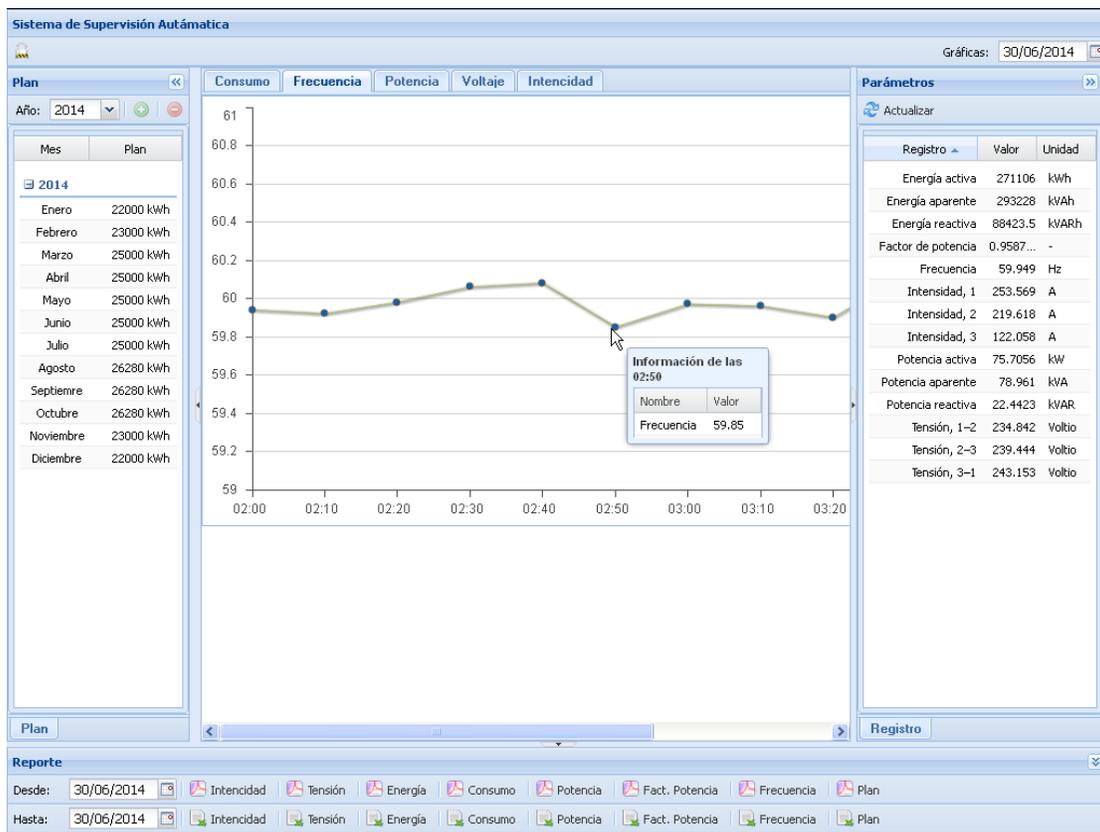


Figura 2.9 Interfaz de la aplicación web con usuario autenticado

## 2.4 Pruebas al sistema

Existen diferentes técnicas para corroborar la calidad de una solución informática durante su proceso de construcción e implantación, por lo general el objetivo principal que persiguen es demostrar que el sistema desarrollado cumple con los requisitos funcionales y no funcionales extraídos durante el flujo de trabajo modelo y que la arquitectura propuesta fue la correcta evaluando la estabilidad, usabilidad, fiabilidad y precisión del software.

Las características e impacto del sistema desarrollado hacen que resulte imprescindible la realización de pruebas que garanticen una correcta elicitación de la información a través de los distintos subsistemas. Por tanto, se diseñaron y realizaron las pruebas de modo que pudiera comprobarse la correctitud de cada etapa del flujo de datos, a partir de diferentes técnicas, las cuales se explican a continuación.

### 2.4.1 Pruebas al servicio (lectura)

El servicio de lectura de parámetros eléctricos, por ser un servicio de Windows, no dispone de interfaz gráfica ni se puede asociar directamente a ningún entorno o aplicación que presuponga la utilización de una capa de presentación. Los servicios de Windows se inician y

ejecutan incluso sin que haya un usuario autenticado en la máquina donde se encuentran instalados. Por esta razón se hace necesaria la utilización de ficheros de texto plano (*logs*) a fin de mostrar al usuario cualquier información de salida que se desee.

El objetivo del servicio de lectura es capturar las variables del analizador de red y posteriormente almacenarlas en la base de datos. Por tanto, toda la lógica de lectura de datos y su subsiguiente almacenamiento en la base de datos son el blanco de las pruebas realizadas.

Para comprobar la correctitud de la extracción de datos se necesita verificar que los procedimientos realizados, basados en la implementación del protocolo MODBUS, funcionan correctamente. No existe otra manera de comprobar esta correctitud salvo comparar, de forma manual, la correspondencia entre los valores extraídos y los que pueden apreciarse directamente, de forma visual, en el analizador de la red.

Estas pruebas fueron diseñadas como pruebas de aceptación:

**Código:** PSL\_1

**Nombre:** Prueba al servicio de lectura

**Descripción:** Prueba para garantizar que los datos extraídos del analizador de red se corresponden con los datos que muestra el analizador de red.

**Condiciones de ejecución:** Servicio de lectura debe tener acceso de escritura al fichero de *log* que se configuró, el fichero de log debe ser actualizado de manera visual a intervalos de 1 segundo.

**Entrada/Pasos de ejecución:** Iniciar el servicio.

**Resultado esperado:** Los valores de las variables extraídas deben corresponderse, de manera general, con los valores que muestra el analizador de red. Es necesario considerar errores producidos por factores como: a) el intervalo de actualización del fichero de salida no se corresponde con el intervalo de actualización de los datos del analizador de red, b) el tiempo requerido por el usuario para alternar su foco de visión entre analizador de red y el fichero de salida puede ser determinante en que no se aprecie el mismo valor en ambas

fuentes de datos.

### **Evaluación de la prueba:** Satisfactoria

Es válido destacar que estas pruebas se realizaron por intervalos de tiempo de 1 a 3 minutos, en diversas iteraciones. Durante estas iteraciones se procedió a conectar o desconectar equipos de alto impacto en la red eléctrica, para verificar que los valores en las variables extraídas mostraran una variación considerable.

#### **2.4.2 Pruebas al servicio (almacenamiento de datos leídos)**

En una segunda fase se procedió a verificar la correctitud de los procedimientos relacionados con el almacenamiento de los datos leídos, para lo cual se realizaron pruebas unitarias. Estas pruebas fueron diseñadas de la siguiente manera:

1. Seleccionar valores arbitrarios para las variables eléctricas a almacenar.
2. Llamar al procedimiento encargado de almacenar las variables eléctricas.
3. Realizar una consulta directa a la base de datos y extraer los últimos valores almacenados.
4. Procesar los valores (donde proceda).
5. Verificar la igualdad de los valores almacenados y extraídos.

Se diseñaron dos pruebas de este tipo, destinadas a verificar la correctitud de los dos procedimientos empleados para el almacenamiento de datos. Uno de estos procedimientos se encarga de almacenar los valores históricos (valores que se muestran en tiempo real). El otro procedimiento se encarga de almacenar el consumo eléctrico. Para verificar este segundo procedimiento fue necesario el procesamiento de algunos datos para su posterior comparación, debido a que en este caso la variable capturada se corresponde con la lectura del consumo acumulativo del analizador de red, mientras que los datos almacenados representan un consumo en un intervalo de tiempo dado.

#### **2.4.3 Pruebas a gráficas y reportes de variables**

Después de verificar la capacidad del servicio para extraer y almacenar de forma correcta las variables del analizador de red, se procedió a comprobar la correctitud de las funcionalidades

de la aplicación web relacionadas con el graficado y reporte en tiempo real de datos. Para ello se realizaron pruebas en formato de pruebas de aceptación.

En estas pruebas el objetivo consiste en verificar que los datos almacenados son los que se están graficando, y que el panel de datos en tiempo real reacciona debidamente cuando se producen cambios en los datos históricos en la base de datos.

Para estas pruebas se emplearon dos usuarios: uno realizando cambios arbitrarios en la base de datos, y otro reportando al primero las modificaciones que veía en los gráficos o panel de datos en tiempo real. Los modelos de estas pruebas aparecen en el ANEXO 2.

#### **2.4.4 Pruebas a la aplicación web**

Una vez comprobada la correcta elicitación de los datos a través de los distintos subsistemas, se procedió a verificar la correctitud de las funcionalidades de la aplicación web. Para ello se diseñaron y realizaron diversas pruebas de aceptación, las cuales aparecen en el ANEXO 2.

#### **2.4.5 Prueba general del flujo de datos**

A manera de verificar la integración de todos los subsistemas implicados, se decidió realizar una prueba general del flujo de datos. Esta prueba consistió en comparar los valores mostrados en el analizador de red, con los valores mostrados en el panel de datos en tiempo real de la aplicación web. El formato de esta prueba es muy similar al descrito en la sección 2.4.1, con la notable diferencia de que en vez de un log, en esta ocasión fue empleado el ambiente gráfico brindado por la aplicación web.

Todas las pruebas realizadas permitieron corregir errores durante la implementación, y mostraron resultados satisfactorios al término de la investigación.

### **2.5 Conclusiones parciales**

En este capítulo se ha detallado a través de los flujos más importantes de la metodología AUP el modelo y la implementación del sistema propuesto, logrando un producto que cumple con los requerimientos funcionales planteados.

Mediante la valoración sostenibilidad se pudo determinar que el producto informático propuesto puede ser explotado sin ningún problema de tipo técnico u operacional y que el

mismo es sostenible desde las dimensiones administrativa, socio-humanista, ambiental y tecnológica.

Las pruebas realizadas a los distintos subsistemas demuestran la correctitud y fiabilidad del sistema desarrollado.

## CONCLUSIONES GENERALES

Durante el desarrollo de esta investigación se realizó un estudio de las principales tendencias y tecnologías aplicables al diseño, análisis e implementación del producto informático propuesto. Con este estudio se adquirieron los conocimientos necesarios sobre las variables eléctricas más importantes y el protocolo MODBUS, que posibilitaron una clara comprensión del proceso de extracción de datos de analizadores de redes eléctricas. Se demostró además la necesidad de desarrollar un sistema informático que solucionara la problemática planteada.

El desarrollo de la solución informática estuvo guiado por la metodología AUP, que permitió un entendimiento claro del negocio, actividad imprescindible para lograr una correcta elicitación de los requisitos y un diseño adecuado del sistema. La arquitectura propuesta facilitó la construcción del software, posibilitando que se elaborara por partes, de forma organizada y rápida, logrando obtener un sistema con alta cohesión y bajo acoplamiento entre los subsistemas diseñados.

Para validar la solución informática se aplicaron un conjunto de pruebas al servicio, pruebas unitarias y pruebas de aceptación, todas arrojando resultados satisfactorios y permitiendo afirmar que los métodos y herramientas seleccionadas para el desarrollo del producto fueron acertados. También es posible afirmar que el sistema desarrollado es sencillo, organizado, fiable y preciso.

Se puede concluir que el objetivo planteado en esta investigación fue cumplido en su totalidad, al obtenerse un sistema informático que permite la extracción automática y gestión de parámetros eléctricos en el Banco de Crédito y Comercio.

.

## RECOMENDACIONES

Una vez concluida la investigación, se proponen las siguientes recomendaciones:

1. Generalizar el uso del producto informático a todas las sucursales de BANDEC en la provincial Holguín.
2. Migrar los servicios implementados sobre tecnologías privativas a tecnologías libres.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] A. d. F. d. M. Eléctrico, "Contribución del material eléctrico a la eficiencia energética de las instalaciones," ed, 2010.
- [2] IEC. (2010, Coping with the Energy Challenge.
- [3] Tomas, "Energía. Indicadores seleccionados," 2010.
- [4] B. Wilkins, "PM710 User Guide," S. Electric, Ed., ed, 2008.
- [5] B. Wilkins. (2011, PM800 User Guide June\_2011.book.
- [6] (2006, Parámetros eléctricos fundamentales. Available: [http://www.aero.ing.unlp.edu.ar/catedras/archivos/electrotecnia\\_Apunte.pdf](http://www.aero.ing.unlp.edu.ar/catedras/archivos/electrotecnia_Apunte.pdf)
- [7] I. S.A. (2014). *¿Qué es la energía activa, la energía reactiva y el factor de potencia?*
- [8] F. R. Quintela, R. C. Redondo, J. M. G. Arévalo, N. R. Melchor, and M. M. Redondo, "Uso de la energía reactiva para evaluar las pérdidas en el sistema eléctrico," *Técnica Industrial. Octubre de, 2006* 2006.
- [9] J. A. E. G. Álvarez. *QUÉ ES LA FRECUENCIA DE LA CORRIENTE ALTERNA (C.A.)*.
- [10] S. Electric, "Eficiencia Energética," S. Electric, Ed., ed, 2010.
- [11] Franco-Galeano-Mendoza, "Sistemas de Control," Facultad De Electrotecnia y Computación, Recinto Universitario Simón Bolívar, 2004.
- [12] C. P. A. Ramón, "DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE SCADA PARA EL MÓDULO DE TEMPERATURA DEL CEMA," Instituto de Investigación de la Facultad de Ingeniería Industrial, 2011.
- [13] X.-M. C. Mohedano. *ModBUS, el protocolo Bueno, Bonito, Barato*. Available: <http://www.xmcarne.com>
- [14] F. A. C. Herías, "Comunicación con RS-485 y MODBUS," Grupo de Innovación Educativa en Automática, Universidad de Alicante, 2011.
- [15] L. Zhen, "APPLICATION OF MODBUS PROTOCOL COMMUNICATION BETWEEN DCS AND PLC IN PETROCHEMICAL PLANTS [J]," *Computer Applications and Software*, vol. 4, p. 062, 2009 2009.
- [16] V. Guerrero. (2013). *MODBUS parte III: ¿Qué es el MODBUS?* Available: <http://www.tecdigitaldelbajo.com>
- [17] michelez. (2012, Manual del Usuario Modbus RTU.
- [18] MOXA. (2013, NPort 5200 Series User's Manual.
- [19] MOXA, "NPort 5200 Series," MOXA, Ed., ed, 2012.
- [20] Alonsojpd. (2011, Cómo desarrollar un servicio de Windows con Delphi.
- [21] A. V. Ménde, "Title," unpublished].
- [22] S. Ambler, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*: Robert Ipsen, 2002.

- [23] A. S. W. (2006). *The Agile Unified Process (AUP)* Available: <http://www.ambyssoft.com/unifiedprocess/agileUP.html>
- [24] S. W. Ambler, "Unified and Agile," 2006.
- [25] J. L. C. L. Ervin Flores "METODOLOGÍAS ÁGILES, PROCESO UNIFICADO ÁGIL (AUP)," ed, 2005.
- [26] (2014). *PROCESO UNIFICADO AGIL (AUP)*. Available: <http://ingenieriadesoftware.mex.tl>
- [27] E. H. Orallo. (2010). *El Lenguaje Unificado de Modelado (UML)*. Available: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
- [28] T. Weilkiens, *Systems engineering with SysML/UML: modeling, analysis, design*: Morgan Kaufmann, 2011.
- [29] J. B. Quintero, R. A. de Páez, J. C. Marín, and A. B. López, "Un estudio comparativo de herramientas para el modelado con UML," *revista universidad eafit*, vol. 41, pp. 60-76, 2012 2012.
- [30] A. Alarcon and E. Sandoval, "Herramientas CASE para ingeniería de Requisitos," *Cultura Científica*, pp. 70-74, 2011 2011.
- [31] R. S. Pressman, *Ingeniería del Software un enfoque práctico*.
- [32] B. Pereyra. (2005, Cátedra de Introducción a la Computación. BASES DE DATOS.
- [33] A. RSV, M. K. Kavitha, T. R. P. Monisha, and T. S. B. Pavithra, "Database Management Systems," *DATABASE*, vol. 1, 2010 2010.
- [34] "¿Qué es un Sistema Gestor de Bases de Datos o SGBD?," ed, 2014.
- [35] P. M. Madruga, "Sistemas Gestores de Bases de Datos," ed, 2006.
- [36] C. Gómez, "MySQL vs. SQL Server," 2010.
- [37] D. M. R. Ortega, "Microsoft SQL Server," 2009.
- [38] P. Nielsen, *SQL Server 2005 Bible*, 1 edition ed. Indianapolis, IN: Wiley, 2006.
- [39] K. England and G. J. Powell, *Microsoft SQL Server 2005 Performance Optimization and Tuning Handbook*: Digital Press, 2011.
- [40] G. E. G. Hernández, "Ventajas de SQL Server 2005," 2001.
- [41] "Delphi XE5 y RAD Studio XE5, 5 razones más para dar el salto | Danysoft," ed, 2014.
- [42] "Más ventajas al dar el salto a Delphi XE5, o Rad studio XE5 | Danysoft," ed, 2014.
- [43] S. Microsystem. *Netbeans.org*. Available: [www.netbeans.org](http://www.netbeans.org)
- [44] H. Böck, J. Tulach, and G. Wielenga, *The Definitive Guide to NetBeans Platform*: Springer, 2009.
- [45] R. Dantas, *NetBeans IDE 7 Cookbook*: Packt Publishing Ltd, 2011.
- [46] P. Fernández. (2014). *Programación multiplataforma asequible en Object Pascal*. Available: <http://www.kriptopolis.com/programacion-object-pascal>
- [47] M. A. Azeem, *Start Programming using Object Pascal*, 2013.

- 
- [48] W. Rachele, *Learn Object Pascal with Delphi*: Wordware Publishing Inc., 2000.
- [49] J. G. Murugarren, "Web Estilo," ed, 2004.
- [50] (2014). *Qué es un servidor web ?*
- [51] P. A. A. Morales. (2014). *Servidores Web - Monografias.com*.
- [52] I. V. Trance. (2014). *¿Que es un servidor apache y para que sirve?*
- [53] G. González. (2014). *Una Introducción a Apache*.
- [54] J. J. Gutiérrez, "¿Qué es un framework web?," 2008.
- [55] J. Eguiluz. (2011). *symfony.es*. Available: <http://www.symfony.es/que-es-symfony/>
- [56] C. Tupe and J. Cisneros, "Evaluación y Selección de Framework de Desarrollo PHP: Symfony, Kumbia, CakePHP y Zend," 2008 2008.
- [57] M. E. Valdés, Y. G. Avilés, O. M. Díaz, and Y. G. Martínez, "Sistema para el Manejo Integral de Perforación de Pozos de Petróleo," *Serie Científica*, vol. 6, 2013 2013.
- [58] F. Potencier, Zaninotto, François, *symfony*, 2009.
- [59] L. Groner. (2011, Ext JS 4 First Look. Available: <http://PacktLib.PacktPub.com>

## ANEXO 1. Descripción de los casos de uso en formato de alto nivel

**Caso de Uso** CU1 Gestionar tipo de dispositivo

**Actores** Energético

**Importancia** Alta

**Descripción** El energético crea, modifica o elimina un tipo de dispositivo para configurar el servicio de lectura.

**Referencia** RF-1

**Caso de Uso** CU-2 Gestionar tipo de dato

**Actores** Energético

**Importancia** Alta

**Descripción** El energético crea, modifica o elimina un tipo de dato para configurar el servicio de lectura.

**Referencia** RF-2

**Caso de Uso** CU-3 Gestionar grupo de parámetros

**Actores** Energético

**Importancia** Baja

**Descripción** El energético crea, modifica o elimina un grupo de parámetros para organizar los parámetros.

**Referencia** RF-3

**Caso de Uso** CU-4 Gestionar parámetro

**Actores** Energético

**Importancia** Alta

**Descripción** El energético crea, modifica o elimina un parámetro para configurar el servicio de lectura.

**Referencia** RF-4

**Caso de Uso** CU-5 Gestionar puerto

**Actores** Energético

**Importancia** Alta

**Descripción** El energético crea, modifica o elimina un puerto para configurar el servicio de lectura.

**Referencia** RF-5

**Caso de Uso** CU-6 Gestionar dispositivo

---

<b>Actores</b>	Energético
<b>Importancia</b>	Alta
<b>Descripción</b>	El energético crea, modifica o elimina un dispositivo para configurar el servicio de lectura.
<b>Referencia</b>	RF-6
<b>Caso de Uso</b>	CU-7 Gestionar horario
<b>Actores</b>	Energético
<b>Importancia</b>	Media
<b>Descripción</b>	El energético crea, modifica o elimina un horario para controlar el consumo de electricidad.
<b>Referencia</b>	RF-7
<b>Caso de Uso</b>	CU-8 Gestionar plan mensual
<b>Actores</b>	Energético
<b>Importancia</b>	Media
<b>Descripción</b>	El energético crea o modifica un plan mensual para controlar el consumo de electricidad.
<b>Referencia</b>	RF-8
<b>Caso de Uso</b>	CU-9 Capturar las variables de los analizadores de redes eléctricas
<b>Actores</b>	Reloj
<b>Importancia</b>	Alta
<b>Descripción</b>	El reloj desencadena una acción de consultar los analizadores de redes eléctricas y almacenar los valores de las variables consultadas en memoria, de forma temporal, para su posterior almacenamiento en la base de datos.
<b>Referencia</b>	RF-9, RF-10, RF-13
<b>Caso de Uso</b>	CU-10 Registrar datos energéticos
<b>Actores</b>	Reloj
<b>Importancia</b>	Alta
<b>Descripción</b>	El reloj desencadena una acción de registrar el valor de un parámetro o el consumo energético que reporta un dispositivo en la base de datos.
<b>Referencia</b>	RF-9, RF-10, RF-11
<b>Caso de Uso</b>	CU-11 Configurar parámetros globales del sistema
<b>Actores</b>	Energético
<b>Importancia</b>	Media
<b>Descripción</b>	El energético modifica los parámetros globales del sistema (contraseña de acceso único, visibilidad del sitio para la red, etc.)
<b>Referencia</b>	RF-12

---

---

<b>Caso de Uso</b>	CU-12 Controlar servicio de captura automatizada de variables
<b>Actores</b>	Energético
<b>Importancia</b>	Baja
<b>Descripción</b>	El energético inicia, detiene o reinicia el servicio automatizado de captura de variables.
<b>Referencia</b>	RF-14

<b>Caso de Uso</b>	CU-13 Consultar gráficos
<b>Actores</b>	Usuario
<b>Importancia</b>	Media
<b>Descripción</b>	El usuario consulta los gráficos del comportamiento energético en tiempo real o en un intervalo de tiempo dado.
<b>Referencia</b>	RF-15, RF-16, RF-17, RF-18

<b>Caso de Uso</b>	CU-14 Generar reportes
<b>Actores</b>	Usuario
<b>Importancia</b>	Baja
<b>Descripción</b>	El usuario genera reportes sobre el comportamiento energético en un intervalo de tiempo dado. Los reportes se pueden imprimir o exportar a CSV, PDF.
<b>Referencia</b>	RF-19, RF-20, RF-21, RF-22

## ANEXO 2. Modelos de pruebas de aceptación

**Código:** PAW\_1

**Nombre:** Prueba de parámetros históricos

**Descripción:** Prueba para garantizar que los parámetros históricos almacenados en la base de datos se muestren correctamente en el panel de datos en tiempo real.

**Condiciones de ejecución:** Tester 1: Aplicación web cargada. Tester 2: Aplicación de administración de la base de datos en ejecución.

**Entrada/Pasos de ejecución:** Tester 2: Modificar el valor de un parámetro histórico arbitrario en la base de datos. Tester 1: Esperar intervalo de actualización automática del

---

---

panel de datos en tiempo real, reportar al Tester 2 el valor que cambió.

**Resultado esperado:** El parámetro cuyo valor cambió debe corresponderse con el parámetro modificado por el Tester 2. La magnitud del parámetro también debe corresponderse con el valor introducido.

**Evaluación de la prueba:** Satisfactoria

**Código:** PAW\_3

**Nombre:** Prueba de gestión de planes

**Descripción:** Prueba para garantizar que los planes de consumo son gestionados de manera correcta.

**Condiciones de ejecución:** Aplicación web cargada con usuario autenticado.

**Entrada/Pasos de ejecución:** Cambiar el valor de un plan mensual, hacer clic en el botón de guardar, abrir aplicación web en nueva pestaña.

**Resultado esperado:** El plan mensual modificado debe aparecer con el último valor guardado.

**Evaluación de la prueba:** Satisfactoria

**Código:** PAW\_5

**Nombre:** Prueba de reporte de consumo

**Descripción:** Prueba para garantizar que el reporte de consumo se genera de forma correcta.

**Condiciones de ejecución:** Aplicación web cargada.

**Entrada/Pasos de ejecución:** Clic en el botón de reporte de consumo.

---

---

**Resultado esperado:** El reporte de consumo debe mostrarse con datos en el intervalo de tiempo seleccionado.

**Evaluación de la prueba:** Satisfactoria

**Código:** PAWS\_1

**Nombre:** Prueba de acceso

**Descripción:** Prueba para garantizar que un usuario anónimo no puede gestionar planes.

**Condiciones de ejecución:** Aplicación web cargada, ningún usuario autenticado.

**Entrada/Pasos de ejecución:** Actualizar página web, crear petición POST con datos de modificación de un plan de consumo, enviar la petición. Autenticarse en la página, verificar datos de planes.

**Resultado esperado:** El plan que se intentó modificar debe tener el valor original, no el modificado.

**Evaluación de la prueba:** Satisfactoria

---

## ANEXO 3. Diagrama Entidad Relación

