

Universidad de las Ciencias Informáticas

Facultad 5



**Título: Análisis de Tendencia en el Sistema de
Supervisión Energética.**

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

Autor(es): Dayamí Lias García

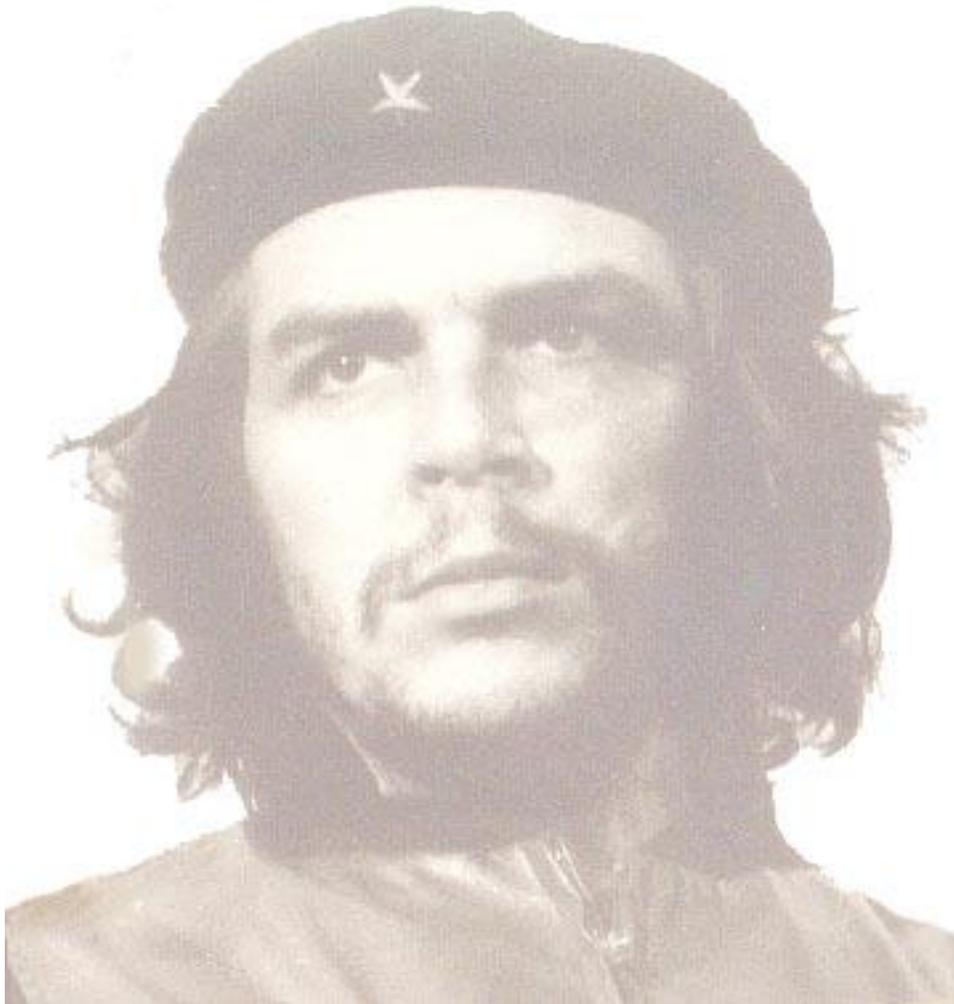
Tutor(es): Ing. Alexander Moreno Limonte
Ing. Leonel Salazar Videaux.

Ciudad de la Habana, junio del 2009

Año del 50 Aniversario del Triunfo de la Revolución

...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos...

Ernesto Che Guevara.



Datos de Contacto

Ing. Leonel Salazar Videaux.

Graduado en la UCI.

Años de Graduado: 2.

Centro de Trabajo: Unidad de Compatibilización, Integración y Desarrollo de productos informáticos para la Defensa (UCID).

Jefe de Proyectos de Automática en la UCID.

AGRADECIMIENTOS

A mis padres por todo el amor, el apoyo y la confianza que me brindaron durante el desarrollo de esta tesis.

A mis tutores Alexander y Leonel, por la ayuda y orientación prestada.

A mi novio por su amor, su comprensión, por saber darme siempre una alegría.

A Ángel y a Yudy por estar presentes y ayudarme en cualquier momento que los necesitara.

A Iliana y Yailin, a ustedes por estar en las buenas y en las malas, por la amistad que desarrollamos y que siempre va a perdurar, a ustedes por su confianza y por su apoyo incondicional.

A todos los que de una forma u otra me brindaron su apoyo y me dieron ánimos para seguir adelante.

DEDICATORIA

A mis padres: por su amor, sus enseñanzas, sus años de sacrificio, su confianza, sus consejos, y apoyo.

A la memoria de mi abuela Isabel, por todo lo que has representado para mí aunque ya no estés presente

A mi novio: por estar siempre a mi lado y apoyarme en todo momento.

A mi grandes amigas Iliana y Yailin.

A todos mis amigos de la UCI

RESUMEN

En la actualidad la energía eléctrica se ha convertido en el motor impulsor de la economía de todos los países del mundo y el pilar que sustenta el constante desarrollo científico tecnológico que experimenta la humanidad. Estos avances han traído consigo un mayor consumo de la energía eléctrica provocando un incremento de la utilización de los recursos naturales destinados a producir dicha energía y una mayor contaminación ambiental.

En Cuba, se han venido tomado un conjunto de medidas a raíz de la revolución energética, en aras de reducir el consumo de energía del país. Como parte de estas acciones se está desarrollando en la Facultad 5 de la Universidad de las Ciencias Informáticas un Sistema Supervisor Energético que posibilite controlar todo el proceso de consumo de la energía eléctrica. Este trabajo se enmarca en el desarrollo de una aplicación Web que permita integrar al Sistema de Supervisión Energético la funcionalidad de Analizar las Tendencias de los parámetros eléctricos en diferentes intervalos de tiempo y monitorizar el comportamiento en Tiempo Real Suave de dichos parámetros.

Para el desarrollo de la aplicación se utilizaron herramientas de software libre. Se usó como metodología, RUP (Rational Unified Process), como lenguaje de modelado, UML (Lenguaje Unificado de Modelado) y Visual Paradigm como herramienta Case, además de los framework Symfony para la programación del lado del servidor mediante el lenguaje PHP5, y ExtJS realizado con JavaScript para la creación de interfaces de usuario avanzadas y como servidor de base de datos PostgreSQL.

Palabras Claves: energía, análisis tendencia, parámetros eléctricos, supervisión energética, tiempo real suave.

Índice General

AGRADECIMIENTOS	II
DEDICATORIA	III
RESUMEN	IV
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	1
1.1- INTRODUCCIÓN.....	1
1.2 – SUPERVISIÓN.....	1
1.3 - SISTEMA DE SUPERVISIÓN ENERGÉTICO.....	2
1.4 - TENDENCIAS.....	7
1.4.1 - ANÁLISIS DE TENDENCIAS.....	8
1.5 - SUPERVISORES ENERGÉTICOS EXISTENTES.....	8
1.6-TENDENCIAS Y TECNOLOGIAS ACTUALES.....	12
1.6.1- APLICACIONES WEB.....	12
1.6.2 – ARQUITECTURA DE SOFTWARE.....	15
1.6.3 –MODELO VISTA CONTROLADOR.....	15
1.6.4- LENGUAJES DE PROGRAMACIÓN WEB UTILIZADOS.....	17
1.6.5 – SISTEMA DE GESTIÓN DE BASE DE DATOS.....	21
1.6.6 –SERVIDOR WEB APACHE.....	23
1.6.7 – FRAMEWORK.....	24
1.6.8- METODOLOGÍA DE DESARROLLO.....	26
1.6.9 – LENGUAJE DE MODELADO.....	28
1.6.10 - HERRAMIENTAS UTILIZADAS.....	29
1.7 – CONSIDERACIONES FINALES DEL CAPÍTULO.....	31
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	31
2.1- INTRODUCCIÓN:.....	31
2.2 – OBJETO DE ESTUDIO.....	31
2.2.1 - SITUACIÓN PROBLÉMICA:.....	31
2.2.2 - OBJETO DE AUTOMATIZACIÓN.....	32
2.2.3 - INFORMACIÓN QUE SE MANEJA.....	32
2.2.4 - PROPUESTA DE SISTEMA:.....	32
2.3 - MODELO DEL NEGOCIO:.....	33
2.3.1- REGLAS DEL NEGOCIO:.....	33
2.3.2- DEFINICIÓN DE ACTOR DE NEGOCIO:.....	33
2.3.3 - TRABAJADOR DEL NEGOCIO.....	34
2.3.4 - DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	34
2.3.5 - ESPECIFICACIÓN DE LOS CASOS DE USO DEL NEGOCIO:.....	35
2.3.6- MODELO DE OBJETO DEL NEGOCIO.....	36
2.4 - REQUISITOS.....	37
2.4.1 – REQUISITOS FUNCIONALES.....	37
2.4.2 – REQUISITOS NO FUNCIONALES.....	38
2.4.3 – ACTORES DEL SISTEMA:.....	41

2.4.4 – DIAGRAMA DE CASOS DE USO DEL SISTEMAS:	41
2.4.5 – DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA:	42
2.5 - CONSIDERACIONES FINALES DEL CAPÍTULO	52
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	53
3.1- INTRODUCCIÓN	53
3.1- ARQUITECTURA DE SYMFONY	53
3.1.1- LA IMPLEMENTACIÓN DEL MVC QUE REALIZA SYMFONY	54
3.2- ANÁLISIS	55
3.2.1- DEFINICIÓN DEL MODELO DE ANÁLISIS	56
3.2.2- DIAGRAMAS DE CLASES DEL ANÁLISIS	56
3.3 - DISEÑO DEL SISTEMA.....	59
3.3.1 -MODELO DEL DISEÑO.....	59
3.3.2 -DIAGRAMA DE CLASES DEL DISEÑO.....	59
3.3.3 -DIAGRAMA DE INTERACCIÓN.	64
3.4 -DESCRIPCIÓN DE LAS CLASES.....	65
3.5 -DISEÑO DE LA BASE DE DATOS.....	65
3.5.1-DIAGRAMA ENTIDAD RELACIÓN DE LA BD.	66
3.5.2-DESCRIPCIÓN DE LAS TABLAS.	67
3.6-PATRONES DE DISEÑO UTILIZADOS POR SYMFONY.....	70
3.7-TRATAMIENTO DE ERRORES.....	72
3.8 -SEGURIDAD.....	72
3.9 - INTERFAZ	73
3.10 - CONSIDERACIONES FINALES DEL CAPÍTULO	73
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	75
4.1-INTRODUCCIÓN.	75
4.2-IMPLEMENTACIÓN.....	75
4.2.1- DIAGRAMA DE DESPLIEGUE.	75
4.3 - DIAGRAMA DE COMPONENTES.....	76
4.4 - PRUEBA.....	83
4.4.1-PRUEBAS DE LA CAJA NEGRA.....	84
4. 5 - CONSIDERACIONES FINALES DEL CAPÍTULO.	86
CONCLUSIONES GENERALES	87
RECOMENDACIONES.....	88
BIBLIOGRAFÍA.....	89
ANEXO.....	93
GLOSARIO	118

INTRODUCCIÓN

En la actualidad es evidente la importancia que ha cobrado el uso de la energía eléctrica para el desarrollo de la sociedad, ya que representa el pilar fundamental para el avance de la tecnología y constituye la fuente de alimentación de equipos, cada vez más sofisticados, que brindan recreación, entretenimiento y comodidades, tanto para facilitar las labores en el hogar como en el trabajo. Los avances que han surgido en diferentes ramas como la industria metalúrgica, la salud, la ciencia, han traído consigo un mayor consumo de la electricidad en los últimos años. Esto constituye un factor preocupante hoy en día, puesto que las fuentes de generación de energía eléctrica, son en su mayoría combustibles fósiles, que además de ser recursos no renovables y que están llamados a desaparecer, traen consigo un problema tan complicado como la contaminación ambiental. Hoy se hace necesario el reflexionar y pensar en no malgastar irracionalmente este tipo de recurso planteándose siempre la siguiente interrogante: ¿Qué pasará con el mundo cuando sus fuentes de energía se agoten?

Cuba no está exenta de este problema. A pesar de que ya están cubiertas las necesidades de generación, producto de las nuevas tecnologías instaladas en el país después de comenzar la revolución energética, en el horario pico se hace necesario, a veces, la entrada de plantas generadoras que no son eficientes, lo que provoca un malgasto de recursos y de divisas al país. Por sus características naturales, el empleo del petróleo crudo nacional para alimentar estas plantas exige acortar los plazos de mantenimiento de las unidades generadoras, y además, en ocasiones, ocurren roturas imprevistas debido al alto contenido de azufre que posee el crudo cubano, lo que hace que algunas plantas salgan de servicio, provocando afectaciones. También se produce una mayor contaminación ambiental y, por otro lado, todo el proceso tiene un alto costo económico para el país.

En noviembre de 1997 comenzó el Programa de Ahorro de Electricidad en Cuba (PAEC), en momentos en que la economía cubana se encontraba en franco proceso de recuperación y en consecuencia, el crecimiento en la demanda y el consumo de electricidad de este propio año cerraba con tasas de 4,9% en la demanda máxima y de 7,8% en la generación de electricidad con relación al año anterior. El objetivo fundamental de este programa era transformar radicalmente el proceso de generación y ahorro de electricidad, a partir de las siguientes ideas expuestas por nuestro Comandante Fidel Castro:

1. "(...) garantizar el desarrollo sostenible y la invulnerabilidad económica de nuestra patria (...)"

2. "(...) cambiar a la población los equipos electrodomésticos altos consumidores de energía y la distribución de otros ahorradores, que contribuyan a la calidad de vida de nuestros compatriotas (...)"
3. "(...) realizar inversiones a todas las redes eléctricas, postes, transformadores, etc. (...)".
4. "(...) desarrollar investigaciones de explotación de fuentes de energía renovables, como son la eólica y la solar, a partir de la identificación de las zonas geográficas factibles (...)"

A raíz de esta Revolución Energética surge en la Universidad de las Ciencias Informáticas (UCI) el proyecto de Supervisión Energética, con el objetivo de planificar, controlar y supervisar el plan de consumo de electricidad de la universidad y de los grandes consumidores del país. Los Supervisores Energéticos están diseñados para monitorear el comportamiento de los principales parámetros energéticos de un consumidor, optimizando así el control de la calidad y el consumo de la energía eléctrica. Estos sistemas permiten la visualización de parámetros en *tiempo real*¹ y la notificación de irregularidades en el comportamiento de los parámetros eléctricos. Dicha funcionalidad, también conocida como análisis de tendencia, permite detectar a tiempo que ocurran fallas en el sistema que conlleven a tener anomalías en el sistema. Además cuentan con aplicaciones que permiten realizar un *análisis histórico*² del comportamiento de las principales variables eléctricas, lo que posibilita una información valiosa a la hora de tomar políticas de consumo y en general hacer un análisis estadístico de los parámetros energéticos.

En nuestro caso específico se identificó como **Situación Problemática** dentro del proyecto de Supervisión Energética, que el sistema que se está desarrollando no cuenta con una funcionalidad que permita realizar un Análisis de Tendencia de parámetros eléctricos, función importante para la toma de decisiones, que puedan en definitiva, planificar políticas de ahorro que beneficien al país.

¹ Un sistema de tiempo real es aquel en el que para que las operaciones computacionales sean correctas no solo es necesario que la lógica e implementación de los programas computacionales sea correcto, sino también el tiempo en el que dicha operación entregó su resultado. Si las restricciones de tiempo no son respetadas el sistema se dice que ha fallado.

² En este caso en específico se refiere a un análisis del comportamiento de los parámetros eléctricos en un tiempo pasado, ya sean días, semanas o meses anteriores.

De aquí es que surge el siguiente **Problema Científico a Resolver**: ¿Cómo implementar una funcionalidad que permita realizar el análisis tanto en *tiempo real suave*³ así como histórico del comportamiento de los parámetros eléctricos?

Teniendo el trabajo como **Objetivo General** implementar una aplicación que permita realizar el análisis de tendencias de los parámetros eléctricos del sistema, que posibilite la toma de decisiones de los operarios, en pos de mejorar la calidad y disminuir el consumo de energía.

Se define como **objeto de estudio**, el análisis de tendencias de parámetros industriales. Siendo el **campo de acción** el análisis de tendencias de parámetros eléctricos en el proyecto de Supervisión Energética.

Para dar solución al objetivo general, se plantean las siguientes **Tareas a desarrollar**:

- Realizar una revisión del estado del arte en cuanto a las técnicas aplicadas al análisis de tendencias.
- Analizar las herramientas seleccionadas para el desarrollo de la aplicación.
- Realizar el proceso de captura de requisitos de la aplicación a desarrollar.
- Realizar el proceso de análisis y diseño de la aplicación a desarrollar.
- Implementar las clases y funcionalidades que den solución al problema.
- Diseñar las pruebas de unidad.
- Integrar la solución desarrollada al sistema de Supervisión Energética.

Para el estudio de la situación problemática antes planteada y conocer las características del análisis de tendencias de parámetros eléctricos se hace uso de los siguientes **Métodos Científicos de Investigación**.

Métodos Teóricos:

- **Analítico – Sintético:**

³ Es un sistema en el que los plazos de tiempo asignados a las tareas no son críticos. Esto significa que el no cumplimiento de una tarea en su tiempo establecido no conlleva a fallas catastróficas o pérdidas de información en el sistema.

Permite obtener a través del estudio y la investigación, las características principales del estado del arte de análisis de tendencias de parámetros, así como las herramientas más factibles para el desarrollo de la aplicación.

- **Modelación:**

Permite definir las clases y funcionalidades de acuerdo con los requisitos definidos anteriormente y así se obtiene un modelo de lo que se quiere desarrollar.

Métodos Empíricos:

- **Entrevista:**

Método por el cual se obtiene información acerca del análisis de tendencias de parámetros eléctricos a través de una conversación planificada.

El **documento se estructuró** en cuatro capítulos en los cuales se trataron los temas que a continuación se describen.

➤ **Capítulo No 1 Fundamentación Teórica:** Este capítulo constituye la base teórica del presente trabajo. En él se describen las características actuales del análisis de tendencias de variables energéticas, además se describen y justifican las tecnologías, herramientas, metodología y lenguajes a utilizar para la implementación del sistema.

➤ **Capítulo No 2** Contiene una propuesta de solución al problema existente. Se describen los artefactos generados en la etapa de modelo de Negocio como diagrama de caso de uso del negocio, diagrama de actividades, el modelo de datos. Se realiza el levantamiento de requisitos funcionales y no funcionales que deberá cumplir el sistema, la construcción del diagrama de casos de usos del sistema, y se realiza la descripción textual de cada caso de uso.

- Capítulo No 3 Se realiza el análisis y diseño del sistema que se propone, mostrándose los diagramas de clases del análisis para cada caso de uso del sistema, la realización de los diagramas de interacción del sistema (en este caso los diagramas de secuencia) y los diagramas de clases del diseño donde se presentan los estereotipos Web y el patrón de diseño a usar debido al tipo de aplicación que se va a construir, así como la descripción de la arquitectura usada.
- Capítulo No 4 Contiene los artefactos generados en los flujos de trabajo de Implementación y Prueba. En este capítulo se especifican las representaciones gráficas del diagrama de componentes y el de despliegue, así como la descripción de los casos de prueba de unidad para cada caso de uso.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1- Introducción.

La electricidad es una de las formas de energía de mayor consumo, de tal forma que su costo actúa sobre todos y cada uno de los sectores de la industria, servicios y la propia economía doméstica. A partir de este hecho, es fácil deducir que cualquier acción que tienda a un uso más racional de dicha energía, tendrá una repercusión sobre la economía de todos y cada uno de los sectores implicados. Una de las formas para reducir el consumo de energía es la utilización de sistemas de supervisión energéticos, los cuales realizan como una de sus funciones el análisis de las tendencias de los parámetros eléctricos. En este capítulo se realizará un estudio del estado del arte del análisis de tendencias de parámetros eléctricos, además de estudiar las posibles herramientas y lenguajes a utilizar para el desarrollo de una aplicación que facilite dicho análisis.

1.2 – Supervisión.

Se define como supervisión al acto de observar el trabajo o tareas de otro (individuo o máquina) que puede no conocer el tema en profundidad. Es un intento de garantizar la calidad de la respuesta del sistema frente a cualquier situación que pueda presentarse. Un sistema de supervisión debe poder disponer de información sobre la operación del objeto supervisado en la forma de representación adecuada. La supervisión ha de facilitar que el operador pueda detectar la ocurrencia de posibles anomalías conocidas y efectuar las acciones adecuadas para corregirlas. [1]

Se puede definir la palabra **supervisar** como ejercer la inspección superior en determinados casos, ver con atención o cuidado y someter una cosa a un nuevo examen para corregirla o repararla permitiendo una acción sobre el elemento supervisado. La labor del supervisor representa una tarea delicada y esencial desde el punto de vista normativo y operativo; de ésta acción depende en gran medida garantizar la calidad y eficiencia del proceso que se desarrolla. En el supervisor descansa la responsabilidad de orientar o corregir las acciones que se desarrollan.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.2.1 - Capacidades de la Supervisión.

- Registrar la evolución del proceso y detectar desviaciones indeseadas en las variables.
- Analizar estas desviaciones y deducir el motivo. Elaborar un diagnóstico de la situación.
- Resolver situaciones conflictivas en línea, en caso de ser posible.

1.2.2 - Etapas de la Supervisión.

- **Detección de fallos:** Obtener indicios de situaciones anómalas que puedan llevar al proceso a una situación de fallo y clasificarlas como tales.
- **Diagnóstico de fallos:** Averiguar las causas principales de ésta situación anómala.
- **Reconfiguración del sistema:** Acciones a realizar para mantener el proceso operativo.
- **Tomar las medidas adecuadas para que no vuelva a suceder.**

1.3 - Sistema de Supervisión Energético.

Un sistema de Supervisión energético realiza optimizaciones a tiempo real del estado estacionario del sistema eléctrico permitiendo automatizar el sistema y reducir las pérdidas y el pico de consumo. Permiten

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

garantizar la continuidad del suministro, maximizar el tiempo operativo de su proceso productivo y alcanzar los requisitos de calidad y tiempos de respuesta.

1.3.1- Principales beneficios de un sistema de supervisión energético.

- **La seguridad:**

- a. Supervisar los eventos de la instalación que pueden representar daños a bienes y/o a personas.
- b. Controlar el correcto funcionamiento de las protecciones.

- **El ahorro de costos energéticos:**

- a. Ahorro de costos energéticos directos, por pérdidas de continuidad de servicio y por reparaciones en equipamientos caros.

- **El confort:**

- a. Recogida centralizada de información.
- b. Tratamiento de la información para un análisis más sencillo.
- c. Control y mando centralizado [2].

1.3.2- Funciones de un sistema de supervisión energético.

- **Medida de parámetros eléctricos básicos**

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1. **Tensiones (simples/compuestas):** La supervisión de estas variables es de vital importancia para la protección del equipamiento instalado tanto en el sector empresarial como en el doméstico puesto que la tensión alta produce fundamentalmente un efecto de calentamiento de los receptores. En determinadas circunstancias, este calentamiento puede ocasionar la avería de equipos no aptos a este tipo de régimen.

La tensión baja produce que la mayor parte de los receptores pasen de un estado de funcionamiento normal a uno anómalo o a uno de no funcionamiento cuando se ven sometidos a una tensión baja, recuperando el estado normal cuando el valor de la tensión vuelve a situarse dentro de los márgenes de tolerancia.

2. **Intensidades:** El análisis del estado de las corrientes trifásicas nos brinda la información referente al balanceo de las cargas en un sistema trifásico. Este parámetro es de gran importancia pues es una medida del buen funcionamiento de un sistema trifásico.

3. **Potencias:** Estas variables nos dan una medida de cuan eficiente es el sistema que se analiza. En sistemas donde el consumo de potencia reactiva sea elevado se deben tomar medidas puesto que esto implica un alto consumo de energía.

4. **Factor de Potencia:** Es un indicador del correcto aprovechamiento de la energía eléctrica. El Factor de Potencia puede tomar valores entre 0 y 1, lo que significa que si toma valores inferiores a 0,95, implica que los artefactos tienen elevados consumos de energía reactiva respecto a la energía activa, produciéndose una circulación excesiva de corriente eléctrica en las instalaciones.

Además:

- ✓ Provoca daños por efecto de sobrecargas.
- ✓ Aumentan las pérdidas por recalentamiento.
- ✓ Aumenta la potencia aparente entregada por el transformador para igual potencia activa utilizada.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

5. **Frecuencia:** Al aumentar la frecuencia también aumenta su reactancia inductiva $X_L = \omega * L$ y por tanto aumenta el consumo, bajando el rendimiento. El rendimiento óptimo se obtiene alrededor de los 50 Hz, siendo a esta frecuencia donde las máquinas trabajan en condiciones económicas, frecuencias mayores se aplican cuando con poco peso se deben conseguir potencias elevadas, sin importar mucho el consumo, un ejemplo lo constituyen los aparatos destinados al transporte aéreo, donde prima el peso sobre el consumo[3].
6. **Energía:** Este parámetro brinda información de suma importancia dado que es la cantidad de potencia real que el usuario emplea en un determinado intervalo de tiempo, es decir, el producto de la potencia de la demanda por el tiempo empleado. Este parámetro nos da la medida de cuánto consumimos realmente y de cuán eficientes somos a la hora de emplear dicha energía.

- **Medida de parámetros no básicos.**

1. THD's (Total Harmonic Distortion, en español: Distorsión Total de Armónicos)
2. Potencias cuarto-horarias.
3. Mínimos, máximos, medias en un intervalo.
4. Armónicos individuales.

Los parámetros no básicos aquí mencionados permiten realizar un análisis profundo sobre el tema de los armónicos en las redes trifásicas. Este tipo de anomalía se debe en gran medida a la conexión de equipos electrónicos a la red y de hornos de arcos eléctricos que provocan que:

1. Se disparen las protecciones eléctricas.
2. Un aumento de la circulación de corriente por el sistema de suministro eléctrico.
3. Pueden ocurrir efectos de resonancia.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

4. Fallas en los circuitos de control.
5. Operación inadecuada de las protecciones.
6. Interferencia en las comunicaciones telefónicas.

- **Señalización de alarmas/eventos**

1. Notificación de parámetros fuera de condiciones de normalidad.
2. Señalización de disparo de los interruptores e información del tipo de defecto.

- **Visualizar el estado de los interruptores y sus regulaciones**

1. Abierto/cerrado
2. Disparado
3. Muelles cargados
4. Regulación instantánea
5. Regulación térmica

- **Control / Mando de interruptores y contactores a distancia.**

1. Para pilotar la instalación actuando a distancia sobre los interruptores de potencia.

- **Gestión / Supervisión de la instalación.**

1. Supervisión / optimización de consumos energéticos.
2. Optimización del mantenimiento, eficiencia de la instalación y vida útil de los equipos.
3. Aumento de la fiabilidad de la instalación [2]

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4 - Tendencias.

Las tendencias son información presentada en forma de gráficas, tablas, diagramas, etc. donde se muestran los valores de las señales (puntos) y su comportamiento en el tiempo. Sirven para revisar el comportamiento de una señal o grupo de señales analógicas y pronosticar su tendencia y así facilitar una correcta toma de decisiones por parte de los operarios.

La tendencia es un gráfico histórico de corta duración, generalmente de 1 a 24 horas como máximo que se va actualizando permanentemente o en tiempo real según cambie la variable graficada (Figura 1).

La visualización gráfica de la variación de los parámetros de una planta determinada en el tiempo es una herramienta muy útil en el control de procesos ya que permite anticiparse a futuros acontecimientos y tomar las decisiones correctas. De acuerdo a como se visualizan respecto al tiempo los mismos se denominan tendencias o históricos. Los gráficos históricos son similares a los de tendencias, solo que el periodo de tiempo en el que se muestra suele ser más grandes (días, meses, años) y no se actualizan en forma permanente sino que visualizan en un rango de tiempo pasado.

Tanto las tendencias como los gráficos históricos generalmente pueden visualizar una o más variables, llegando hasta 16 variables en un solo gráfico, esto permitiría analizar todas las variables en un solo gráfico y con una misma base de tiempo. [3]

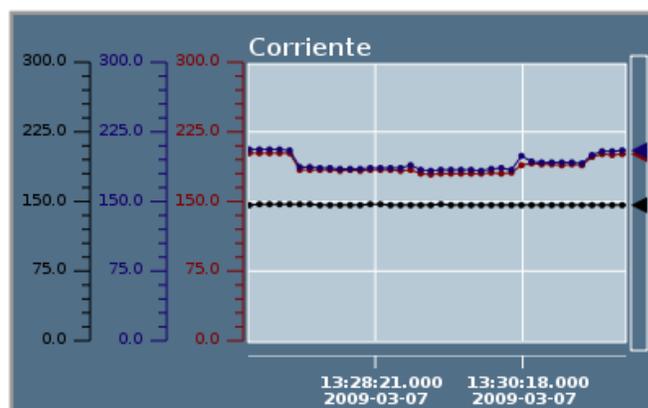
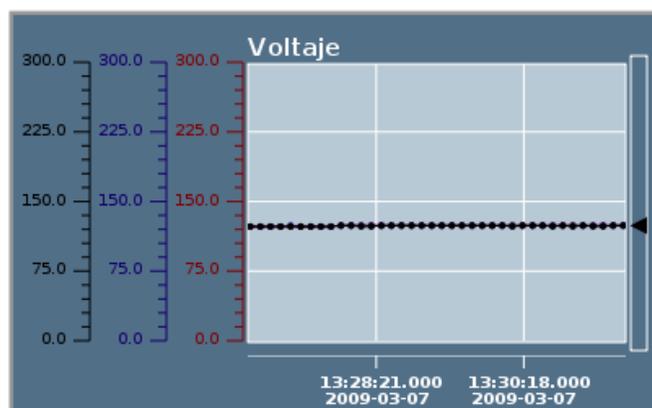


Figura 1: Gráfica de Tendencias.

1.4.1 - Análisis de tendencias.

En los procesos actuales la cantidad de variables a monitorear, y sobre las cuáles se pueden tomar decisiones de manera que permitan controlar o mejorar la producción, es tan grande que sería casi imposible estar siguiendo el comportamiento de todas en forma permanente. Conocer el estado y tendencia real de los diferentes parámetros en régimen estable o transitorio que determinan el perfecto desempeño de los sistemas eléctricos, permite prevenir y evitar sobre costos producidos por fallos inesperados en el sistema energético, bloqueo de los sistemas de procesamiento de datos y pérdida de información.

Beneficios obtenidos por el correcto empleo de técnicas de análisis de tendencias:

- Incremento de la seguridad y confiabilidad de los procesos productivos.
- Asegurar un mayor tiempo de vida útil para los dispositivos eléctricos y electrónicos.
- Conocimiento real y completo del comportamiento del sistema eléctrico.
- Mejoramiento de la compatibilidad electromagnética de los equipos más sensibles y críticos.
- Planteamiento de soluciones para mitigar el efecto de los armónicos que causan mal funcionamiento en los equipos sensibles y pérdidas de energía en el sistema eléctrico.

1.5 - Supervisores Energéticos existentes.

En el mundo existen una gran variedad de sistemas supervisores de energía capaces de controlar un número cada vez mayor de variables eléctricas, optimizando el control de la calidad y el consumo de energía. A continuación se tratarán algunos de estos sistemas.

1.5.1- Ámbito Internacional.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1- ETAP PSMS (Supervisión y Simulación de Sistemas de Potencia): PSMS (Power System Monitoring & Simulation) es una opción para sistemas eléctricos de pequeña y gran escala, centrales de generación, plantas industriales, fábricas y plataformas mar adentro. PSMS puede determinar la respuesta apropiada del sistema ante una variedad de cambios y perturbaciones por la aplicación de parámetros físicos y eléctricos, niveles de carga y generación, topología de las redes y las lógicas de control. Entre sus principales funciones se encuentran:

- **Supervisión avanzada:**

Verifica las condiciones de operación, la estimación de estados del sistema, la detección de anomalías en el sistema y alarmas anunciadoras basadas en las condiciones de operación y cambios en el estado de la red.

- **Simulación en tiempo real:**

Permite predecir el comportamiento del sistema en respuesta a acciones de operadores y eventos por el análisis de datos en tiempo real e históricos guardados.

- **Reproducción de eventos:**

El reproductor de eventos es especialmente útil para el análisis de causa-efecto, mejora de la operación del sistema y exploración de acciones alternativas. La capacidad del reproductor de eventos se traduce en la reducción de los costos de mantenimiento y la prevención de costosas pérdidas de suministro.

- **Generador de tendencias:**

Permite rastrear y grabar actividad inusual con registro de eventos y herramientas de alarmas. Ésta característica permite detectar y anunciar problemas en forma temprana antes que ocurran fallas críticas. Los cambios en la información del sistema se visualizan gráficamente y se registran.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Alarmas y advertencias:**

Proporciona alarmas para activar la respuesta del operador ante condiciones de emergencia. La anunciación de alarmas puede configurarse para servicios de mensajes, ayuda, remedios, colores y sonido.

- **Análisis del costo energético:**

Proporciona una comparación entre costos y consumos con respecto a los puntos de generación local, asignación de intercambios de potencia y reservas rodantes mientras se reportan los consumos y los perfiles de precios de energía. [4]

2- ETAP EMS (Energy Management System): Este sistema de gestión de energía está diseñado para reducir y optimizar el consumo disminuyendo los costos, mejorar la utilización del sistema, incrementar la confiabilidad y predecir el comportamiento del sistema. Algunas de sus funciones serán tratadas a continuación:

- **Control de Supervisión y Consultivo:**

Permite al operador aplicar los objetivos y restricciones para lograr una operación óptima aplicando algoritmos de flujo de cargas definidos por el usuario, para determinar los mejores ajustes operativos del sistema.

- **Gestión Inteligente de la Demanda:**

Proporciona soluciones en forma automática para minimizar los costos de operación y maximizar el valor de las inversiones en energía.

- **Simulador de Sistemas de Control:**

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El simulador es una herramienta automática de verificación de instrumentación y control. Esto se logra mediante el mapeo configurable de los ajustes de los dispositivos de control y una simulación continua de la respuesta del sistema. [4]

3- Circutor es una empresa fundada en 1973 cuyo objetivo principal es la fabricación de equipos, de protección eléctrica industrial así como la medida y control de la energía eléctrica. Cuenta con un grupo de Investigación y Desarrollo con más de 50 ingenieros y ha desarrollado sistemas para la gestión energética dentro de los cuales se destaca el PowerStudio. Su principal función es la comunicación con los equipos CIRCUTOR, y la elaboración a posteriori de tablas y gráficas de los históricos registrados. Algunas de las principales prestaciones de PowerStudio son:

- Visualización de parámetros en tiempo real.
- Módulo de alarmas.
- Software Multipuesto (Servidor Web) mediante pantallas estáticas.
- Gran versatilidad y muy fácil uso.
- Acceso a través de Internet con contraseña y posibilidad de creación de perfiles de acceso. [5]

1.5.2- Ámbito Nacional.

En la facultad 5 de la UCI en el proyecto SCADA Guardián del Alba se desarrolló, en el módulo Web, una aplicación que permite realizar un análisis de tendencia de los parámetros eléctricos. Esta aplicación funciona, implementándose paralelo al ambiente de ejecución del HMI, un servidor que se nutre de la implementación existente en ese módulo, y mediante una aplicación Desktop que corre bajo el módulo

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

HMI grafica los puntos enviados desde la Base de Datos, enviando esta petición al servidor web apache que mediante unas modificaciones realizadas al mismo, convierte esta petición a HTML.

De esta forma es que se logra obtener un análisis del comportamiento de los parámetros eléctricos teniéndose como inconveniente, que el uso de una aplicación Desktop corriendo bajo el módulo HMI torna ineficiente el proceso de peticiones cuando existen muchas conexiones realizadas a la aplicación.

1.6-TENDENCIAS Y TECNOLOGIAS ACTUALES.

Para desarrollar esta aplicación se realizó un estudio de las tendencias y tecnologías que fueron implantadas por el proyecto Supervisión Energética para el desarrollo de los módulos Web: lenguajes de programación, sistema gestor de bases de datos, frameworks, y servidor Web.

1.6.1- Aplicaciones Web.

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML (**HyperText Markup Language** (Lenguaje de Marcas de Hipertexto)), JavaScript, Java, etc.) en la que se confía la ejecución al navegador.

Las aplicaciones web pueden ser clasificadas en siete categorías:

- **Informacionales:** periódicos online, catálogos de productos, newsletters, manuales de servicios, clasificadores online, libros electrónicos.
- **Interactivos:** formularios de registros, presentación de información personalizada, juegos.
- **Transaccionales:** tiendas electrónicas, órdenes de bienes y servicios, bancos electrónicos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Flujo de Trabajo (Workflow):** sistemas de planificación y organización en línea, gestión de inventarios, monitorización de estado.
- **Ambientes de trabajo colaborativos:** sistemas de autoría distribuidos, herramientas de diseño colaborativo.
- **Comunidades en línea, mercados:** grupos de chat, ofertas y demandas de productos o servicios, mercados en línea, subastas en línea.
- **Portales Web:** búsquedas, tiendas, intermediarios, etc.

Ventajas:

-Ahorro en formación y capacitación de usuarios.

Una aplicación Web tiene la ventaja de ejecutarse en un navegador de Internet por lo que su manejo es muy intuitivo ya que los usuarios están acostumbrados al sistema de hipervínculos de Internet.

- Entorno cliente / servidor

La aplicación es servida simultáneamente a múltiples usuarios de una o varias redes sin que se den errores por escrituras al mismo tiempo en las bases de datos, por ejecución simultánea de determinados procesos evitándose, de esta forma, duplicidades de datos.

- Ahorro en costos de implantación

Las aplicaciones Web no requieren instalación en los puestos de cliente por lo que se evitan costos de intervención al ampliar el número de ordenadores o realizar cambios en los mismos. Así mismo, al funcionar en el navegador de Internet se minimizan los costos de intervención técnica derivados de fallos de ejecución ya que en el 90% de los casos si la navegación en Internet es operativa, también lo es la aplicación Web.

- Alta disponibilidad

Las aplicaciones Web pueden estar disponibles desde cualquier parte del mundo donde exista acceso a Internet por lo que facilita el trabajo fuera de las oficinas de la empresa así como el del personal en movilidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Centralización de datos y Backups.

La aplicación está ubicada, generalmente, en un único servidor por lo que los trabajos de copia de seguridad y mantenimiento son mucho más eficientes y con un costo sustancialmente menor que el de las aplicaciones en modo local o limitadas a una LAN (Local Area Network o Red de Área Local).

- Posibilidad de redundancia

En este tipo de aplicaciones existe la posibilidad de disponer de servidores de redundancia para casos de fallo en la red de datos principal reduciendo así las posibilidades de una parada en el servicio.

- Actualizaciones inmediatas

Las actualizaciones, mejoras o cambios en la aplicación son puestas en servicio de manera inmediata para todos los usuarios y para todos los ordenadores de la red sin intervención sobre los equipos ya que se realizan exclusivamente sobre el servidor.

- Seguridad

Las aplicaciones Web son completamente seguras a pesar de transmitir sus datos por la red pública de Internet ya que pueden dotarse de sistemas de encriptación que evitan que la información transmitida pueda ser visible a personas no autorizadas.

Desventajas:

- Interfaces de usuario atractivas, pero de menor usabilidad que las aplicaciones de escritorio tradicionales.
- Velocidad de ejecución mucho menor comparada con las aplicaciones de escritorio.
- Interacción con el ordenador del usuario limitada.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Al ser una aplicación pública a nivel de Internet, se requiere un nivel de mantenimiento de los servidores donde se encuentra alojada la aplicación mucho mayor que las aplicaciones de escritorio.

1.6.2 – Arquitectura de Software.

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficientemente dentro de la organización.

En una aplicación web la arquitectura es sencilla, esta es conocida como cliente servidor donde sus componentes principales son el Servidor Web, una Red y un Navegador o cliente además de la aplicación en el Servidor, que es la que permite al sistema manejar lógica de negocio y tener un estado. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo. Para la comunicación de los procesos con la red se emplea un tipo de equipo lógico denominado middleware que controla las conversaciones. Su función es independizar ambos procesos (cliente y servidor). La interfaz que presenta es la estándar de los servicios de red que hace que los procesos "piensen" en todo momento que se están comunicando con una red.

1.6.3 –Modelo Vista Controlador.

La arquitectura MVC (Model-View-Controller o Modelo Vista Controlador) fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, como se ha citado, o en sistemas CAD (Computer Aided Design o diseño asistido por computadora), en donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas.

En la (Figura 2), vemos la arquitectura MVC en su forma más general. Hay un Modelo, múltiples Controladores que manipulan ese Modelo y varias Vistas de los datos del Modelo, que cambian cuando cambia el estado de ese Modelo.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa, lo cual nos permite implementarlos por separado
- Hay una API (Application Programming Interface o interfaz de programación de aplicaciones) muy bien definida; cualquiera que use la API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unir las en tiempo de ejecución. Si posteriormente, se observa que uno de los componentes funciona mal, este puede ser reemplazado sin que las otras piezas se vean afectadas.

Definición de las partes:

El modelo es la representación física de la información, incluye la lógica de datos que asegura la integridad de la información y la derivación de nuevos datos.

La vista es la representación específica de la información contenida en el modelo en un formato visual generalmente, preparado para interactuar con los usuarios mediante una GUI (Graphical User Interface,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Interfaz Gráfica de Usuario). Escucha los cambios en el modelo y se comunica con el modelo usando el controlador.

El controlador es el encargado de responder a eventos, dígame acciones del usuario, invocar cambios en el modelo y en la vista al permitir la comunicación entre éstos al agregar una instancia de la vista y otra del modelo y servirle a la vista para interactuar con el modelo. [6]

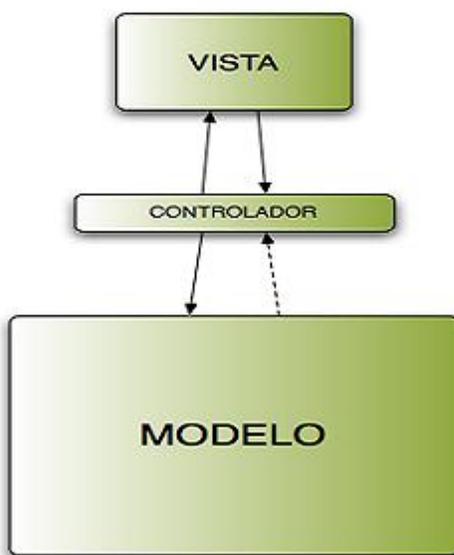


Figura 2: Modelo Vista Controlador

1.6.4- Lenguajes de Programación Web Utilizados.

Existe una multitud de lenguajes concebidos para la Web. Cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útil para desarrollar distintas aplicaciones. Se pueden clasificar de dos formas diferentes; de acuerdo a la arquitectura Cliente/Servidor, los lenguajes del lado del servidor son aquellos reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

un formato comprensible para él, por otro lado, los lenguajes del lado del cliente son aquellos que pueden ser directamente comprendidos por el navegador y no necesitan un pre-tratamiento.

Cada uno de estos tipos tiene por supuesto sus ventajas y sus inconvenientes, por ejemplo: un lenguaje del lado del cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Inversamente, un lenguaje del lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo; por otra parte, los scripts son almacenados en el servidor quien los ejecuta y traduce a HTML por lo que permanecen ocultos para el cliente. Este hecho puede resultar a todas luces una forma legítima de proteger el trabajo intelectual realizado.

1.6.4.1- Lenguajes del lado del cliente utilizados.

JavaScript:

Es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con JavaScript podemos crear efectos especiales en las páginas y definir interacciones con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas, de modo que el mayor recurso, y tal vez el único, con que cuenta éste lenguaje es el propio navegador. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender éste lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica. Permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, JavaScript pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente. [7]

Ajax:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML (Extensible Markup Language (lenguaje de marcas extensible))), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. AJAX es una combinación de cuatro tecnologías ya existentes:

- XHTML (o HTML (HyperText Markup Language)) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asíncronicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano y JSON (JavaScript Object Notation).

Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente. [8]

1.6.4.2 – Lenguaje del lado del servidor utilizado:

PHP:

PHP (acrónimo de PHP: Hypertext Preprocessor) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (Server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt (Quasar Technologies) o GTK+ (The GIMP Toolkit).

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Ventajas:

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cuál se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Desventajas:

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización, Unicode, etc.
- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.
- Por sus características promueve la creación de código desordenado y complejo de mantener.
- Está diseñado especialmente para un modo de hacer aplicaciones web que es ampliamente considerado problemático y obsoleto (mezclar el código con la creación de la página web).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño **Modelo Vista Controlador** (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes. [9]

1.6.5 – Sistema de Gestión de Base de Datos.

Los Sistemas de Gestión de Base de Datos (en inglés Data Base Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Ventajas:

- Proveen facilidades para la manipulación de grandes volúmenes de datos. Entre éstas:
 - Simplifican la programación de chequeos de consistencia.
 - Manejando las políticas de respaldo adecuadas garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores en el disco, o hay muchos usuarios accediendo simultáneamente a los mismos datos, o se ejecutaron programas que no terminaron su trabajo correctamente, etc.
 - Permiten realizar modificaciones en la organización de los datos con un impacto mínimo en el código de los programas.
 - Permiten implementar un manejo centralizado de la seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- Las facilidades anteriores bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Desventajas:

1. Típicamente, es necesario disponer de una o más personas que administren la base de datos, en la misma forma en que suele ser necesario en instalaciones de cierto porte disponer de una o más personas que administren los sistemas operativos.
2. Complejidad: los SGBD (Sistema de Gestión de Base de Datos) son software muy complejos y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.
3. Tamaño: la complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder ejecutarse.
4. Coste del hardware adicional: los requisitos de hardware para correr un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero.

1.6.5. 1- PostgreSQL.

PostgreSQL: Es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD (*Berkeley Software Distribution*).

Características:

- Alta concurrencia:

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

- Disparadores (*triggers*) :

Un *disparador* o *trigger* se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Todos los disparadores se definen por seis características:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- El nombre del trigger o disparador
- El momento en que el disparador debe arrancar
- El evento del disparador deberá activarse sobre...
- La tabla donde el disparador se activará
- La frecuencia de la ejecución
- La función que podría ser llamada

Unas de sus principales características es el soporte de distintos tipos de datos, además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, etc. y permite la creación de tipos propios.

Otro punto importante sobre PostgreSQL es la poca necesidad de recursos de hardware que requiere y la simplificación del proceso de administración de licencias de software, que no es necesario cuando se usa software libre. Una de sus características más importantes es poder lidiar con gran volumen de datos. Es capaz de ajustarse al número de CPUs (Central Processing Unit o Unidad Central de Procesamiento) y a la cantidad de memoria que posee el sistema de forma óptima, permitiéndole soportar una mayor cantidad de peticiones simultáneas de manera correcta.

1.6.6 –Servidor Web Apache.

Apache es el servidor Web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en éste programa. El servidor Apache es un software (libre) servidor HTTP (Protocolo de Transferencia de Hipertexto) de código abierto para plataformas Unix, Linux, Windows y otras. Apache no necesita de licencias, y no recibe soporte técnico. Eso sí, está demostrado que ante los problemas de seguridad, actúan más rápido que Microsoft, poniendo a disposición de los usuarios hotfixes (es el término que se utiliza para identificar todos los diferentes tipos de paquetes que pueden hacer que un sistema esté al día) que solucionan el problema en menos tiempo.

- Apache supone una clara alternativa para el que pretenda trastear con el mundo de los servidores Web, con la tranquilidad de no estar quebrantando ningún tipo de ley.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Está disponible para sistemas Windows.
- Gran facilidad de manejo.
- Amplias librerías disponibles, especialmente en Perl (Practical Extraction and Report Language - Lenguaje Práctico para la Extracción e Informe) y PHP.

1.6.7 – Framework.

Un **framework**, es una estructura de soporte definida mediante la cuál un proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cuál extiende o utiliza las aplicaciones del dominio.

1.6.7.1 - Symfony Framework.

Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Características:

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales adaptándose a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

1.6.7.2 - EXTJS.

ExtJS es un framework JavaScript para interfaces avanzadas, amparado bajo GPLv3, nace como solución a tareas comunes pero complejas. Este framework es completamente basado en programación orientada a objeto, cada objeto contiene lo típico: propiedades, métodos, eventos etc.

ExtJS basa toda su funcionalidad en JavaScript a través de librerías: YUI (Librería de Interfaces de Usuario), jQuery y Prototype/Script.aculo.us y un poderoso núcleo interno. Así, en tiempo de ejecución carga y crea todos los objetos HTML a través del uso intenso del Document Object Model o modelo de objetos del documento (DOM), ventanas, mensajes emergentes, grillas, date pickers etc. Los datos son obtenidos mediante AJAX a través de XML y/o JSON

Ventajas:

- La orientación a objetos intensa hará modular todos los scripts.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- El diseño está completamente separado de la funcionalidad.
- Funciones comunes como validación, combobox editables, ventanas arrastrables (con minimizar y maximizar), grillas editables, que son muy fáciles de implementar.
- Buena y amplia documentación, así como también su comunidad.

Desventajas:

- Crear un sistema serio con esta herramienta requiere un previo uso prolongado, ya que se puede perder con muchos nuevos objetos en su extensa y bien documentada API
- El tiempo de aprendizaje puede llegar a compararse con aprender a programar en un lenguaje nuevo.
- Al estar todo el sitio en JavaScript, no podrá ser accesible para los buscadores, limitando su uso a sistemas y no sitios web.
- Si existiese algún objeto que se necesite y no existiera, se necesitaría crear un nuevo objeto (sólo apto para programadores JavaScript avanzados).

1.6.8- Metodología de Desarrollo.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. La tendencia actual de la producción de software requiere cada día la construcción de sistemas más grandes, complejos y rápidos, que se ajusten a las necesidades de los usuarios.

Para esto, se hizo necesario realizar un estudio de las metodologías de ingeniería del software que guiara el proceso de automatización de la aplicación que se propone en este trabajo.

1.6.8.1 – RUP.

El **Proceso Unificado de Rational** (Rational Unified Process en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Es un marco de desarrollo de software (conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software), está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

Ésta metodología de desarrollo se basa en casos de uso para describir lo que se espera del software y para guiar el proceso además de prestar atención especial al establecimiento temprano de una buena arquitectura la cuál se basa en componentes, se caracteriza por su forma disciplinada de asignar tareas y responsabilidades (quién, qué, cuándo y cómo).

Principales Características

- Forma disciplinada de asignar tareas y responsabilidades (quién, qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al concluir cada ciclo; estos se dividen en 4 fases:

- **Inicio:** El objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener una versión del proyecto.

Cada una de éstas etapas es desarrollada mediante el ciclo de iteraciones, la cuál consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El ciclo de vida se desarrolla a través de nueve flujos de trabajo:

- **Modelo del Negocio:** Entendiendo las necesidades del negocio.
- **Requerimientos:** Trasladando las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Trasladando los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- **Configuración y administración del cambio:** Guardando todas las versiones del proyecto.
- **Administrando el proyecto:** Administrando horarios y recursos.
- **Ambiente:** Administrando el ambiente de desarrollo.
- **Distribución:** Hacer todo lo necesario para la salida del proyecto

1.6.9 – Lenguaje de Modelado.

Es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar (parte de) un objeto. Algunas organizaciones los usan extensivamente en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores. A continuación se hará referencia al que será usado para el desarrollo de dicho sistema.

1.6.9.1 – UML.

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad siendo empleado diariamente por grandes organizaciones como: Microsoft, Oracle y Rational.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

No es un lenguaje de programación pero las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. [10]

1.6.10 - Herramientas Utilizadas.

1.6.10.1 - Visual Paradigm.

Se escoge esta herramienta de modelado porque constituye una de las herramientas líderes en el mundo de la modelación visual. Es una herramienta Case que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta un entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de usos, además de una integración con Visio para el dibujo de diagramas UML.

1.6.10.2 - Eclipse.

Eclipse es un Entorno Integrado de Desarrollo, IDE (Integrated Development Environment), inicialmente desarrollado por IBM (International Business Machines), y actualmente destinado por la Fundación Eclipse, de código abierto, multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Es una herramienta para el programador, desarrollada principalmente para la creación de aplicaciones Java, facilitando al máximo la gestión de proyectos colaborativos mediante el control de versiones cvs, es posible también con subversion, exportar e importar proyectos. Es posible añadir nuevas funcionalidades al editor, a través de nuevos módulos (*plugins*), para programar en otros lenguajes de programación además de Java como C/C++, PHP, Python, Ruby, Cobol, etc.

Características:

1. Multiplataforma (GNU/Linux, Solaris, Mac OSX, Windows).
2. Soportado para distintas arquitecturas (x86, x64, etc.).
3. Estructura de plugins que hace sencillo añadir nuevas características y funcionalidades.
4. Control de versiones con CVS (Concurrent Versions System) o con subversión.
5. Resaltado de sintaxis, autocompletado, tabulador de un bloque de código seleccionado, compilación en tiempo real, refactorización, pruebas unitarias con Junit, entre otras.
6. Asistentes (wizards): para la creación, exportación e importación de proyectos; para generar esqueletos de códigos (*templates*), etc.

1.6.10.2.1 Eclipse PHP Development Tools (PDT).

Se trata de una extensión de Eclipse echa específicamente para programar en PHP. Ofrece resaltado de sintaxis, autocompletado, asistente para el código, plantillas. Es un conjunto de herramientas y frameworks que incrementan el rendimiento de los desarrolladores usando PHP.

Características:

- Editores de texto sensibles al contexto, que proporcionan funcionalidades tales como el coloreado de la sintaxis, asistente de código y plegado de código.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Integración con el modelo de proyecto Eclipse que permite la inspección de código a través de las vistas outline del proyecto de fichero y la nueva vista de Explorador de PHP.
- Ayuda para eliminar errores de código de PHP adicionales.
- Amplios frameworks y API's que permiten a los desarrolladores e ISVs (Independent software vendor) extender fácilmente PDT para crear nuevas herramientas de desarrollo orientadas a PHP.

1.7 – Consideraciones finales del capítulo.

En este capítulo se hizo referencia a conceptos relacionados con los temas de supervisión energética así como de las características que poseen algunas de las aplicaciones de este tipo existentes a nivel mundial. El software a desarrollar implementará algunas de las funcionalidades que brindan los Supervisores Energéticos revisados en este trabajo como la visualización de parámetros eléctricos en tiempo real, la generación de graficas con valores históricos de los parámetros eléctricos y la implementación de un pequeño sistema de alarmas para anunciar al operador condiciones de emergencia. En la segunda parte, se desarrolló el estudio de algunas de las tecnologías actuales que serán empleadas durante el desarrollo del software, seleccionando como lenguaje de programación a utilizar el PHP, por la serie de ventajas descritas anteriormente y se utilizara además los lenguaje del lado del cliente JavaScript y Ajax. Como sistema gestor de base de datos se escogió PostgreSQL. Se tiene también como propuesta a utilizar la metodología de desarrollo RUP con notación UML debido a sus características y organización del trabajo. Herramienta CASE de Modelado UML: Visual Paradigm y Servidor Web: Apache.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1- Introducción:

Para desarrollar un sistema, es necesario comprender primeramente el entorno de trabajo donde se va a aplicar la herramienta informática, bajo el estudio de los procesos que en ella se desarrollan. Es por ésta razón que en este capítulo se realizará una breve descripción de los procesos de negocio relacionados con el objeto de estudio. Se especifican los actores y trabajadores del negocio, los casos de uso, los diagramas de actividad y el modelo de objetos del negocio. Además se especifican los requisitos funcionales y no funcionales que debe cumplir el sistema, lo cuál permite definir los actores del sistema, diagramas de casos de uso así como las especificaciones de cada uno de estos casos de uso del sistema.

2.2 – Objeto de Estudio

2.2.1 - Situación problemática:

Como parte del amplio programa de ahorro de electricidad que desarrolla la revolución cubana en busca de una economía energética sostenible se desean instalar en las industrias elementos de medición y análisis de red eléctrica como son los analizadores y metrocontadores digitales. Estos dispositivos poseen una extensa capacidad de medición, procesamiento y almacenamiento de todo tipo de parámetros energéticos entre otras. Poder monitorear en cualquier momento como se están comportando las variables energéticas leídas contribuye a una mejor toma de decisiones por parte de los operarios, ante cambios bruscos del comportamiento de dichas variables. En nuestro país éste proceso se realiza generalmente mediante el uso de analizadores de red trifásicos, los cuáles son equipos que se emplean para realizar mediciones en un plazo de tiempo determinado. Estas mediciones son usadas, posteriormente, en el estudio del comportamiento de los parámetros energéticos mediante el uso de herramientas propietarias en una PC. Éste tipo de servicio es brindado por la UNE (Unión Eléctrica) pero cuenta con la dificultad de que el análisis que se realiza está sujeto a datos históricos implicando esto que se detecte el problema luego de que haya ocurrido. A ésto debemos de agregar que tanto el analizador trifásico como el software son caros y cómo se dijo anteriormente no permiten realizar un análisis en

tiempo real de los parámetros eléctricos imposibilitando así una toma correcta de decisiones ante anomalías.

2.2.2 - Objeto de automatización.

Los procesos a automatizar son:

- Comportamiento de las variables en un intervalo de tiempo.
- Configuración de los límites (máximos y mínimos) de las variables.
- Generación de alarmas por límites.
- Configuración de alarmas por tasas de cambio.

2.2.3 - Información que se maneja.

Básicamente la información que se maneja es:

- Información del comportamiento de las variables energéticas.

Cada usuario, de acuerdo con los privilegios que les fueron otorgados tienen acceso a la información referente al comportamiento de las variables energéticas en diferentes intervalos de tiempo y modificar los valores máximos y mínimos de dichas variables así como su tasa de cambio y el plan anual o mensual establecido por la UNE.

2.2.4 - Propuesta de Sistema:

Para dar solución a los problemas existentes con respecto al proceso de análisis de tendencia de parámetros eléctricos vistos con anterioridad y en vista de que no existe otro sistema que pueda dar solución a dichos problemas se ha decidido realizar una aplicación Web que solucione satisfactoria y eficientemente las necesidades planteadas. El sistema estará confeccionado de forma tal que solo le brinde al usuario las funcionalidades que le está permitido realizar. El sistema debe de ser fácil de usar y seguro para todo aquel que posea un conocimiento básico acerca de la electricidad y el análisis de

tendencia de parámetros energéticos.

2.3 - Modelo del Negocio:

La modelación del negocio describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización. En él, se modelan los clientes y trabajadores del negocio y su interacción con los procesos y objetos que lo componen, además de especificar los requisitos que debe cumplir el sistema. Está formado por el modelo de casos de uso del negocio y el modelo de objetos del negocio.

2.3.1- Reglas del Negocio:

1. Solo el Especialista Energético está autorizado a configurar los máximos y los mínimos para las variables energéticas, así como el plan de consumo y las tasas de cambio.
2. Para poder realizar cualquiera de los dos tipos de análisis se debe seleccionar por lo menos un parámetro.
3. Es necesario para los usuarios autenticarse antes de entrar al sistema.
4. Las gráficas del análisis de tendencia deben durar 24 horas como máximo.

2.3.2- Definición de Actor de Negocio:

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término *actor* significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados.

Actor	Justificación
Operador	Es la persona que solicita monitorear el comportamiento de las variables energéticas en un intervalo de tiempo.

Tabla 1: Actores del Negocio.

2.3.3 - Trabajador del Negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.

Trabajador	Justificación
Software del Analizador Trifásico	Es el encargado de tomar los valores de las variables y mostrar al usuario en forma de gráfica el comportamiento de dichas variables a partir de los valores obtenidos.

Tabla 2: Trabajador del Negocio

2.3.4 - Diagrama de Casos de Uso del Negocio

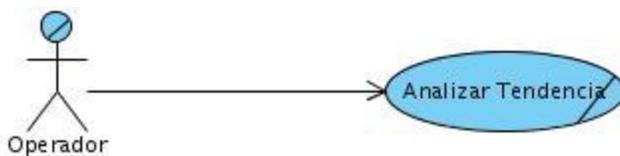


Figura 3: Diagrama Casos de Uso del Negocio.

2.3.5 - Especificación de los Casos de Uso del Negocio:

La descripción textual de los casos de uso del negocio, ayuda a comprender la lógica del mismo, mostrando además cómo ocurre la relación actor-negocio.

Tabla 3: CUN Analizar Tendencia

Caso de Uso:	Analizar Tendencia	
Actores:	Operador	
Trabajadores:	Software del Analizador Trifásico	
Resumen:	El caso de uso comienza cuando el Operador solicita realizar un análisis de tendencia. El software muestra una grafica que refleja el comportamiento de los parámetros a partir de las lecturas realizadas.	
Precondiciones:	Los valores de los parámetros deben de haber sido descargados del Analizador Trifásico.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1-Solicita realizar Análisis de Tendencia de las variables energéticas.	1.1-Busca valores de los parámetros energéticos descargados por el Analizador Trifásico. 1.2-Muestra en una gráfica el comportamiento de los parámetros energéticos a partir de los valores leídos.	
Flujos Alternos		

Acción del Actor	Respuesta del Negocio
Pos-condiciones	El Operador puede realizar un análisis a simple vista del comportamiento de los parámetros a partir de los valores graficados.

Ver Diagrama de Actividad del caso de uso Analizar Tendencia. [Anexo](#)

2.3.6- Modelo de Objeto del Negocio

El Modelo de Objetos del Negocio no es más que la representación, mediante un diagrama, de los trabajadores, las entidades y las unidades de trabajo, así como las interacciones entre ellos, que intervienen en el proceso definido como negocio.

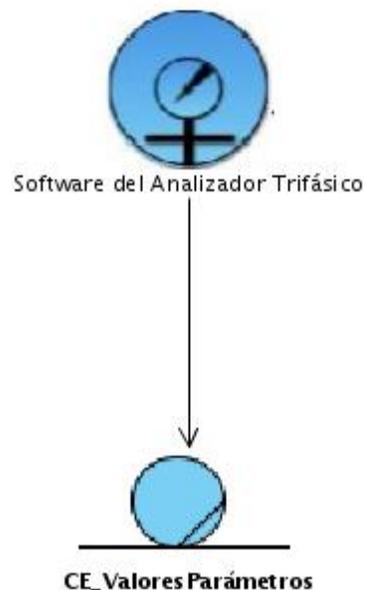


Figura 4: Modelo de Objeto del Negocio.

2.4 - Requisitos

Los requisitos son condiciones o capacidades que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Los mismos pueden ser clasificados en Funcionales y No funcionales.

2.4.1 – Requisitos Funcionales

Los Requisitos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por lo general se describen mejor a través del modelo de Casos de uso. Por lo tanto los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. [11]

R1- Realizar Análisis de Tendencia Histórico.

- 1.1- El sistema de permitir seleccionar el parámetro a graficar.
- 1.2- El sistema de permitir seleccionar un intervalo de tiempo.

R2- Realizar Análisis de Tendencia.

2.1- El sistema de permitir seleccionar el parámetro a graficar.

R3- Configurar valores

3.1- El sistema debe permitir establecer límites (máximo y mínimo) a un parámetro.

3.2- El sistema debe permitir establecer planes de consumo por períodos de tiempo: Anual, Mensual y Diario.

R4- Graficar límites de la tendencia.

4.1 – El sistema debe dibujar una línea azul para representar el máximo.

4.2 – El sistema debe dibujar una línea roja representando el mínimo.

R5- Establecer tasa de cambio.

5.1 – El sistema debe permitir seleccionar un parámetro.

5.2 – El sistema debe permitir adicionar la tasa de cambio y el tiempo de chequeo.

2.4.2 – Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, como las características que hacen al producto atractivo, usable, rápido o confiable.

A continuación se exponen los requisitos no funcionales definido para la propuesta de solución a implementar:

- **Apariencia o Interfaz Externa**

La herramienta propuesta debe tener una interfaz amigable y fácil de usar, de manera que no sea una dificultad para el usuario el uso de ella. La aplicación usa colores aceptables a la vista, preferentemente azul, blanco y gris.

- **Rendimiento:**

Se debe garantizar que la respuesta a las solicitudes del usuario se produzca en tiempos breves. Debe estar disponible las 24 horas del día.

- **Seguridad:**

Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema. Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo. Garantizar a los usuarios del sistema el acceso a la información solicitada en todo momento. El sistema debe comunicarse usando un protocolo http.

- **Confiabilidad:**

El sistema garantiza todas las validaciones de los datos de entrada. Deberá tener disponibilidad de un 100%, pues debe estar accesible durante todo el día y toda la semana. La información manejada por el sistema debe de estar protegida de acceso no autorizado.

- **Software:**

Para el funcionamiento del sistema es necesaria una PC (Personal Computer) con sistemas operativos Linux o Windows. Se requiere el uso del Apache como servidor Web. Gestor de Base de Datos PostgreSQL. Navegador Internet Explorer 5 superior o navegador Mozilla Firefox versión 1.5 o superior. El sistema se desarrollará con tecnología PHP versión 5.0.

- **Hardware:**

Para las computadoras del cliente: Se requiere que tengan tarjeta de red, al menos 512 MB de memoria RAM, 1 GB de disco duro y procesador 512 MHz como mínimo. Para los servidores: Se requiere tarjeta de red, al menos 512 MB de RAM, 40 GB de disco duro y procesador 2.0 GHz como mínimo.

- **Restricciones en el diseño y la implementación:**

Para el análisis y diseño se utiliza la metodología RUP que emplea lenguaje UML, apoyadas en el Visual Paradigm como herramienta case. Para la Implementación, como se necesita que el sistema sea multiplataforma se utiliza PHP como lenguaje de programación, Apache Web Server como servidor Web y como SGBD (Sistema Gestor de Base de Datos) PostgreSQL.

- **Requerimientos de Usabilidad:**

La aplicación será usada por aquellas personas que tiene relación con la información que se maneja y por aquellas que cuenten con conocimientos básicos de computación y entornos web.

- **Soporte:**

Se requiere de cualquier servidor con las siguientes características:

1. Apache (Servidor Web) Versión 2.2 o superior.
2. PHP (Lenguaje de programación) Versión 5 o superior.
3. PostgreSQL (Gestor de Bases de Datos) 8.3 Versión o superior.

2.4.3 – Actores del Sistema:

Un actor del sistema no es más que personas o sistemas externos al sistema que interactúan con él para lograr la realización de sus funcionalidades.

Actores del Sistema	Justificación
Analizador de Tendencia	Es la persona que a través de la aplicación solicita realizar un análisis de tendencia, beneficiándose de la automatización de este proceso que se realizará en el sistema.
Especialista Energético	Es la persona encargada de configurar los valores máximos y mínimos de las variables energéticas así como establecer y modificar el plan de consumo por períodos de tiempo y establecer las tasas de cambio para cada variable energética. También realiza el análisis de tendencia de los parámetros energéticos

Tabla 4: Actores del Sistema

2.4.4 – Diagrama de Casos de Uso del Sistemas:

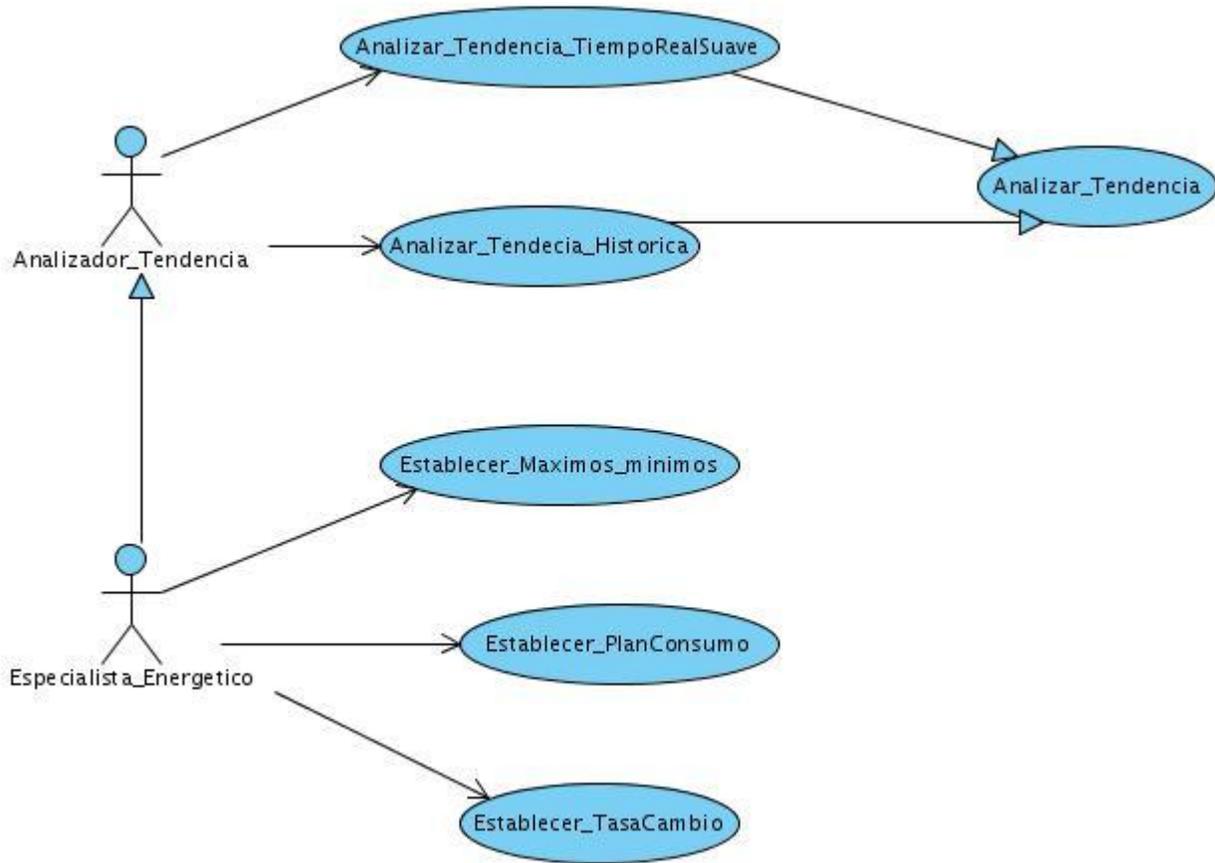


Figura 5: Diagrama Casos de uso del Sistema

2.4.5 – Descripción de los Casos de Uso del Sistema:

Mediante la descripción de los casos de uso del sistema se detalla paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema. Éste sería el último paso para pasar a la construcción de la solución propuesta.

A continuación se muestran las descripciones de los casos de uso a automatizar en el sistema, donde se

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

describe de forma especificada la interacción de los actores con el sistema.

Caso de Uso:	Analizar_Tendencia (Caso de uso base generalizado)	
Actor:		
Propósito	Este caso de uso contiene los procesos comunes a los casos de uso Analizar_Tendencia_Histórica y Analizar_Tendencia_TiempoRealSuave.	
Resumen:	El caso de uso inicia cuando el Caso de Uso “Analizar_Tendencia_Histórica” o “Analizar_Tendencia_TiempoRealSuave” invoca el comportamiento del caso de uso en cuestión, que generaliza acciones comunes de ambos casos.	
Referencias:	R2.1 R2.2	
Precondiciones:	Tiene que ser inicializado el caso de uso Analizar_Tendencia_TiempoRealSuave o Analizar_Tendencia_Histórica.	
Pos-condiciones:		
Prioridad:	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

<p>2 - Selecciona el parámetro a analizar.</p>	<p>1.1-Muestra un formulario en el que pide seleccionar el parámetro a analizar.</p> <p>3.1-Busca valores máximo y mínimo del parámetro seleccionado.</p> <p>3.2-- Muestra en una gráfica el comportamiento del parámetro seleccionado según los valores encontrados y los máximos y mínimos establecidos para dicho parámetro.</p>
<p>Flujos Alternos</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
	<p>3.1.1- Si no se encuentran máximos y mínimos definidos para el parámetro se muestra un mensaje informando dicha situación.</p> <p>4- No encuentra valores para el parámetro seleccionado.</p> <p>5 -Muestra un mensaje informando que no se han encontrado valores para el parámetro seleccionado.</p>

Tabla 5: Caso de Uso Base Generalizado Analizar Tendencia.

<p>Caso de Uso:</p>	<p>Analizar_Tendencia_Histórica (Especialización)</p>
<p>Actor:</p>	<p>Analizador de tendencias</p>

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Propósito	Realizar un análisis histórico del comportamiento de las variables energéticas en un intervalo de tiempo especificado.
Resumen:	El caso de uso inicia cuando el analizador de tendencia solicita realizar un análisis histórico de los parámetros energéticos. El sistema muestra una interfaz con un formulario donde pide seleccionar el parámetro y un intervalo de tiempo. El caso de uso termina cuando el sistema muestra en una gráfica el comportamiento según los valores del parámetro seleccionado en el intervalo de tiempo entrado.
Referencias:	R3.1
Precondiciones:	Seleccionar un parámetro a analizar.
Pos-condiciones:	Quedan reflejados en una gráfica los valores encontrados del parámetro seleccionado.
Prioridad:	Critico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1-Se dirige al menú principal y selecciona el submenú Análisis.</p> <p>2-Selecciona la opción Análisis Tendencia Histórica.</p> <p>3-Se hace referencia al flujo normal de eventos 2 del caso de uso base generalizado Analizar_Tendencia.</p> <p>4- Entra rango de fecha.</p>	<p>1.1-Muestra los dos tipos de análisis que se pueden realizar.</p> <p>2.1 – Se hace referencia al flujo normal de eventos 1.1 del caso de uso base generalizado Analizar_Tendencia.</p> <p>3.1- El sistema pide entrar rango de fecha en el que se analizara el parámetro seleccionado.</p> <p>4.1- Valida que el rango de fecha entrado este correcto.</p> <p>4.2- Busca valores del parámetro seleccionado en el rango de fecha especificado.</p> <p>4.3-Se hace referencia al flujo normal de eventos 3.1</p>

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<p>del caso de uso base generalizado Analizar_Tendencia.</p> <p>4.3- Se hace referencia al flujo normal de eventos 3.2 del caso de uso base generalizado Analizar_Tendencia.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>4.1.1- Si el rango de fecha es incorrecto muestra un mensaje informando dicha situación.</p> <p>4.2.1- Se hace referencia al flujo alternativo 4 del caso de uso base generalizado Analizar_Tendencia.</p> <p>4.2.2 -Se hace referencia al flujo alternativo 5 del caso de uso base generalizado Analizar_Tendencia.</p> <p>4.3.1 -Se hace referencia al flujo alternativo 3.1.1 del caso de uso base generalizado Analizar_Tendencia.</p>

Tabla 6: Caso de Uso Especializado Analizar Tendencia Histórica.

Caso de Uso:	Analizar_Tendencia_TiempoRealSuave(Especialización)
Actor:	Analizador de tendencias
Propósito	Realizar un análisis en tiempo real suave del comportamiento de las variables energéticas.
Resumen:	El caso de uso inicia cuando el analizador de tendencia solicita realizar un análisis de tendencia en tiempo real de los parámetros energéticos. El sistema muestra una interfaz con un formulario donde pide seleccionar el parámetro que se desea

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	analizar. El caso de uso termina cuando el sistema muestra en una gráfica el comportamiento según los valores encontrados del parámetro seleccionado.
Referencias:	R3.1
Precondiciones:	Seleccionar un parámetro.
Poscondiciones:	Se puede monitorear en tiempo real como se está comportando el parámetro seleccionado.
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1-Se dirige al menú principal y selecciona el submenú Análisis.</p> <p>2-Selecciona la opción Análisis Tendencia Tiempo Real.</p> <p>3-Se hace referencia al flujo normal de eventos 2 del caso de uso base generalizado Analizar_Tendencia.</p>	<p>1.1-Muestra los dos tipos de análisis que se pueden realizar.</p> <p>2.1 – Se hace referencia al flujo normal de eventos 1.1 del caso de uso base generalizado Analizar_Tendencia.</p> <p>3.1 - Busca valores del parámetro seleccionado a partir de la hora 00:00:00 del día actual hasta el último valor que esté entrando a la base de datos.</p> <p>3.2 -Se hace referencia al flujo normal de eventos 3.1 del caso de uso base generalizado Analizar_Tendencia.</p> <p>3.3- Se hace referencia al flujo normal de eventos 3.2 del caso de uso base generalizado Analizar_Tendencia.</p> <p>3.4 - Muestra un mensaje si se sobrepasa el máximo o mínimo establecido para el parámetro seleccionado.</p>

	<p>3.5 – Muestra un mensaje si se sobrepasa la tasa de cambio definida para el parámetro seleccionado.</p> <p>3.6 – Muestra un mensaje si se sobrepasa el plan de consumo anual, mensual o diario convenido con la UNE.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>3.1.1- Se hace referencia al flujo alternativo 4 del caso de uso base generalizado Analizar_Tendencia.</p> <p>3.1.2 -Se hace referencia al flujo alternativo 5 del caso de uso base generalizado Analizar_Tendencia.</p> <p>3.2.1 -Se hace referencia al flujo alternativo 3.1.1 del caso de uso base generalizado Analizar_Tendencia.</p>

Tabla 7: Caso de Uso Especializado Analizar Tendencia Tiempo Real Suave.

Caso de Uso:	Establecer Máximos y Mínimos
Actor:	Especialista Energético.
Propósito	Establecer los valores máximos y mínimos para las variables energéticas existentes.
Resumen:	El caso de uso inicia cuando el Especialista solicita configurar los valores máximos y mínimos de las variables energéticas. El sistema muestra una interfaz con las variables energéticas existentes. El especialista selecciona una variable y establece los valores para la misma. El sistema captura éstos datos y así termina

	el caso de uso.
Referencias:	R3.1
Precondiciones:	Seleccionar al menos un parámetro.
Pos-condiciones:	Quedan establecidos en el sistema los valores máximos y mínimos para los parámetros eléctricos.
Prioridad:	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1 -Solicita configurar máximos y mínimos.</p> <p>2-Selecciona el parámetro a configurar.</p> <p>3-Introduce valores máximos y mínimos para el parámetro seleccionado y acepta.</p>	<p>1.1- Muestra en un formulario los parámetros a los cuales se les configurara sus valores máximos y mínimos.</p> <p>3.1 - El sistema guarda estos valores.</p> <p>3.2 – Muestra un mensaje informando que los valores han sido guardados.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<p>3.1. 1– No introduce valor máximo o mínimo para el parámetro seleccionado y acepta.</p>	<p>3.1.1.1- Muestra un mensaje informando que deben ser llenados todos los campos del formulario.</p>

Tabla 8: Caso de Uso Establecer Máximos y Mínimos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Caso de Uso:	Establecer Plan Consumo	
Actores:	Especialista Energético	
Propósito	Este caso de uso se encarga de establecer el Plan de Consumo por periodos de tiempo.	
Resumen:	El caso de uso inicia cuando el Especialista solicita configurar el Plan de Consumo. El sistema muestra una interfaz con los periodos de tiempo en que puede ser configurado el Plan de Consumo. El Especialista selecciona el o los periodos a configurar e introduce los valores. El caso de uso termina cuando el sistema guarda estos valores.	
Referencias:	R3.2	
Precondiciones:	Seleccionar al menos un período.	
Pos-condiciones:	Quedan establecidos en el sistema los valores en diferentes periodos del plan de consumo.	
Prioridad:	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1 -Solicita configurar Plan de Consumo. 2-Selecciona el período a configurar. 3-Introduce valores para este período y acepta	1.1- Muestra los períodos de tiempo en que puede ser configurado el Plan de Consumo. -Anual -Mensual -Día 3.1 - El sistema guarda estos valores. 3.2 – Muestra un mensaje informando que los valores han sido guardados.	
Flujos Alternos		

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Caso de Uso:	Establecer Tasa de Cambio	
Actores:	Especialista Energético	
Propósito	Este caso de uso se encarga de establecer la tasa de cambio para cada variable energética.	
Resumen:	El caso de uso inicia cuando el Especialista solicita establecer la tasa de cambio a las variables energéticas. El sistema muestra una interfaz los parámetros a los que se les especificará su tasa de cambio. El Especialista selecciona él o los parámetros. El caso de uso termina cuando el sistema guarda éstos valores que el especialista especifica para cada variable.	
Referencias:	R5	
Precondiciones:	Seleccionar al menos un parámetro	
Pos-condiciones:	Quedan establecidas en el sistema las tasas de cambio para cada variable energética.	
Prioridad:	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1 -Solicita configurar Tasas de cambio.</p> <p>2-Selecciona el parámetro al que se le establecerá la tasa de cambio.</p> <p>3-Introduce valores de la tasa de cambio para el parámetro seleccionados.</p> <p>4- Introduce tiempo de chequeo para calcular la pendiente en la cuál se comprobará la tasa de cambio y acepta.</p>	<p>1.1- Muestra los parámetros a los que se les especificara su tasa de cambio.</p> <p>4.1 - El sistema guarda los valores introducidos.</p> <p>4.2 – Muestra un mensaje informando que los valores han sido guardados.</p>	
Flujos Alternos		

Acción del Actor	Respuesta del Sistema
3.1 – No introduce valores para la tasa de cambio y el tiempo de chequeo y acepta.	3.1.1 - Muestra un mensaje informando que debe de ser llenado al menos un campo.

Tabla 9: Caso de Uso Establecer Tasa de Cambio.

2.5 - Consideraciones finales del capítulo.

Éste capítulo ha brindado una clara definición de los requisitos que deberá cumplir la aplicación, además se hizo una representación de las funcionalidades a construir mediante el diagrama de casos de usos del sistema, donde finalmente se describieron paso a paso todas las acciones del actor del sistema con los casos de uso que interactúa. El empleo de los modelos de casos de usos para describir la solución propuesta permitió una adecuada captación y modelación de requerimientos, demostrando su importancia en ésta etapa. Se ganó claridad en cuánto a la concepción del sistema a construir y se sentaron las bases para las restantes fases del proceso de análisis, diseño e implementación. Los resultados serán utilizados como entrada en la siguiente etapa de trabajo.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1- Introducción.

En éste capítulo, se aborda todo lo referido al flujo de trabajo análisis y diseño para el desarrollo de la aplicación. Teniendo en cuenta lo que se obtuvo en las etapas anteriores a estas, se representan los artefactos fundamentales generados: en el análisis se define el diagrama de clases del análisis, donde se especifican las clases y las relaciones existentes entre ellas. Del mismo modo se definen los diagramas de interacción que por su parte muestran la interacción entre los actores y el sistema, quedando expuestos los mensajes que se transmiten entre los objetos. En el diseño se especifican los diagramas de clases y el diseño de la base de datos.

3.1- Arquitectura de Symfony.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

1. El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
2. La vista transforma el modelo en una página web, que permite al usuario interactuar con ella.
3. El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista), por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo, una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

3.1.1- La implementación del MVC que realiza Symfony.

Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.

En primer lugar, el controlador frontal y el layout son comunes para todas las acciones de la aplicación. Se pueden tener varios controladores y varios layouts, pero solamente es obligatorio tener uno de cada acción. El controlador frontal es un componente que sólo tiene código relativo al MVC, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática.

Las clases de la capa del modelo también se generan automáticamente, en función de la estructura de datos de la aplicación. La librería Propel se encarga de esta generación automática, ya que crea el esqueleto o estructura básica de las clases, y genera automáticamente el código necesario. Cuando Propel encuentra restricciones de claves foráneas (o externas) o cuando encuentra datos de tipo fecha, crea métodos especiales para acceder y modificar esos datos. La abstracción de la base de datos es completamente invisible al programador, ya que la realiza otro componente específico llamado Creole. Así, si se cambia el sistema gestor de bases de datos en cualquier momento, no se debe reescribir ni una línea de código, ya que tan sólo es necesario modificar un parámetro en un archivo de configuración. Por último, la lógica de la vista se puede transformar en un archivo de configuración sencillo, sin necesidad de programarla. [12]

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

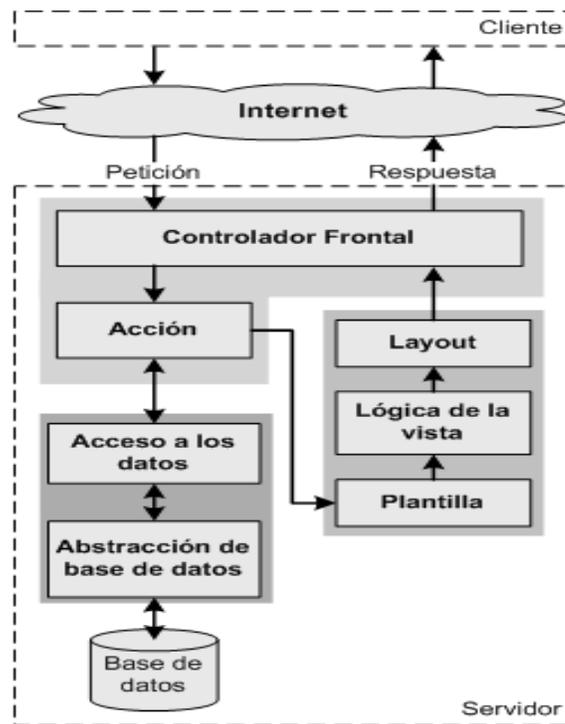


Figura 6: Modelo Vista Controlador implementado por Symfony.

3.2- Análisis

Con el flujo de trabajo análisis, se obtiene una visión general del futuro sistema en función de los requisitos funcionales anteriormente definidos. En dicha etapa se analizan cada caso de uso, siendo transformados a un lenguaje técnico para obtener la primera aproximación al diseño e incluso la arquitectura. De forma general, en esta etapa se han definido por caso de uso todas las clases de la interfaz externa, expresando la interacción del usuario con el sistema, una clase de control encapsula funciones complejas para el manejo de la información almacenada, mostrando el comportamiento de los casos de usos y se definen también, todas las clases entidades que modelan la información utilizada por el sistema.

3.2.1- Definición del Modelo de Análisis.

El modelo de análisis es fundamental en el proceso de estructuración y refinamiento de los requisitos, nos permite razonar sobre los aspectos internos del sistema, incluyendo los recursos internos compartidos, proporciona un mayor poder expresivo y una mayor formalización, así como una arquitectura centrada en el mantenimiento de aspectos tales como la flexibilidad ante los cambios y la reutilización.

3.2.2- Diagramas de Clases del Análisis.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

A continuación se muestran los diagramas de clases del análisis correspondiente a los diferentes CU (Caso de Uso) del sistema.

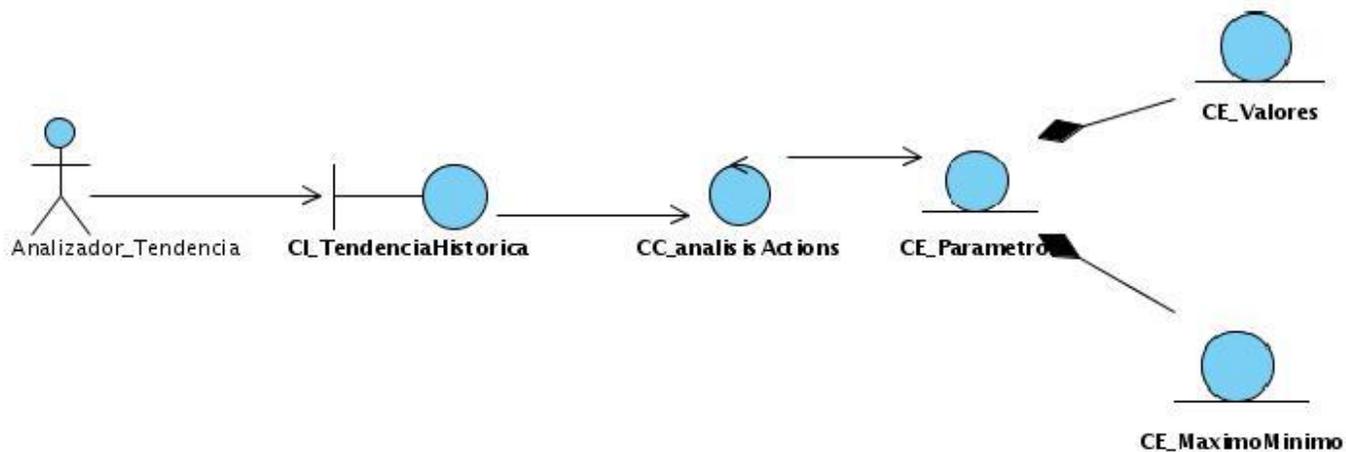


Figura 7: Diagrama de Clases del Análisis Caso de Uso “Analizar Tendencia Histórica”.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

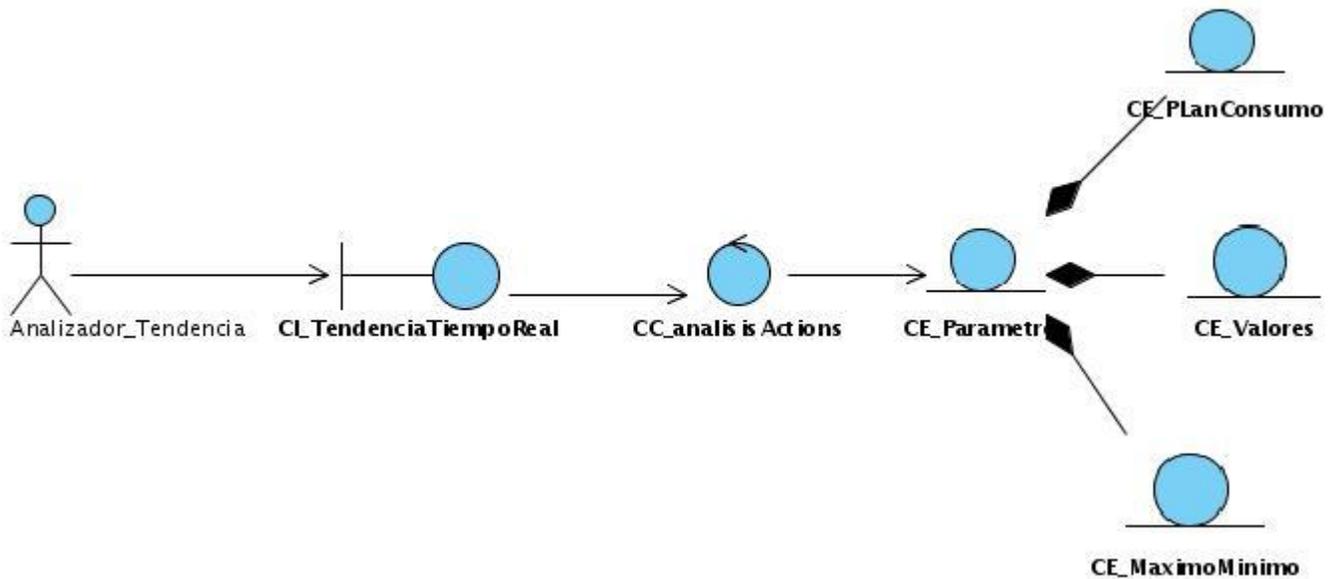


Figura 8: Diagrama de Clases del Análisis Caso de Uso “Analizar Tendencia Tiempo Real”.

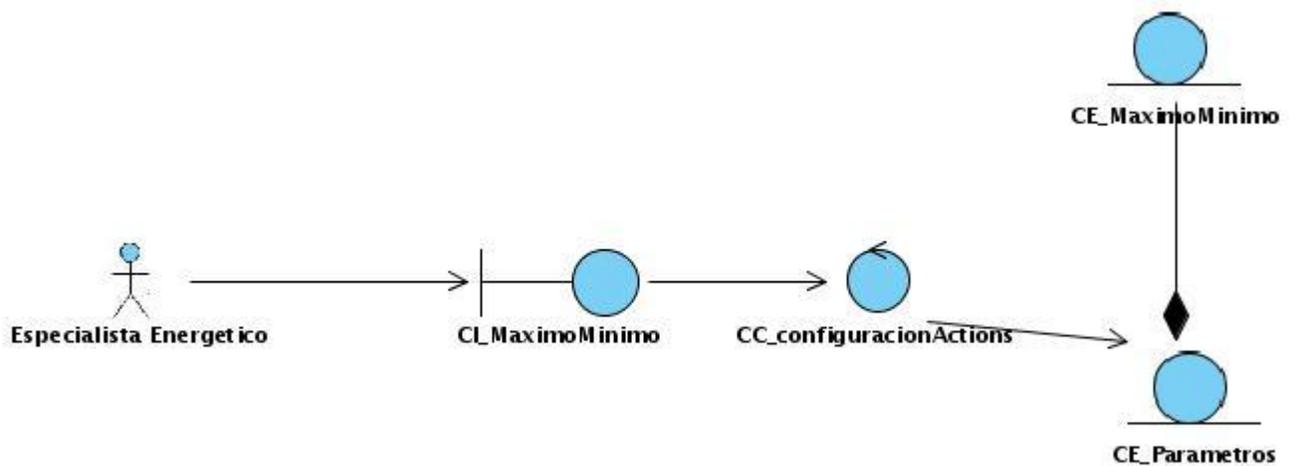


Figura 9: Diagrama de Clases del Análisis Caso de Uso “Establecer Máximo Y Mínimo”.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

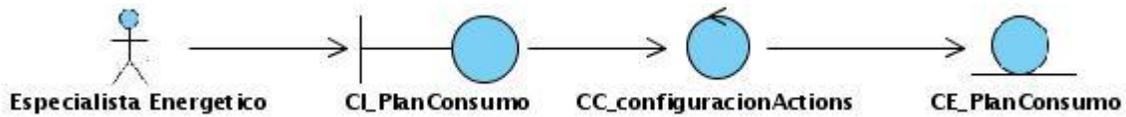


Figura 10: Diagrama de Clases del Análisis Caso de Uso “Establecer Plan de Consumo”.

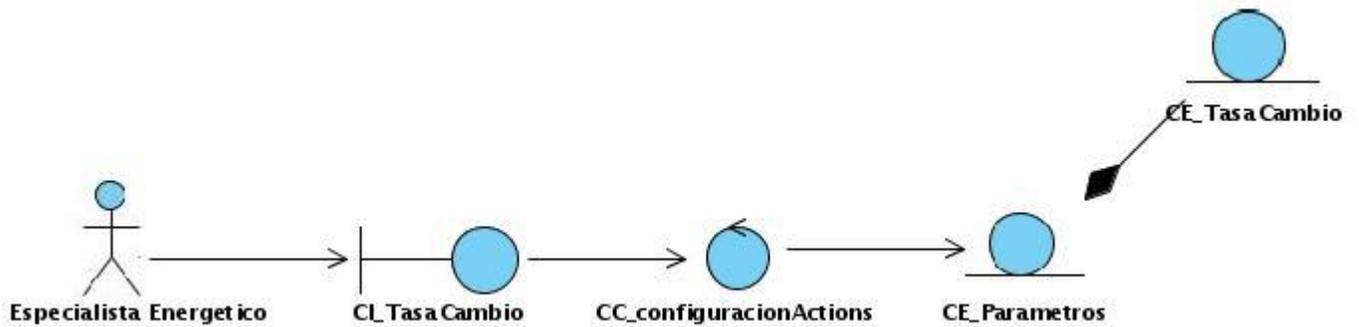


Figura 11: Diagrama de Clases del Análisis Caso de Uso “Establecer Tasa de Cambio”.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.3 - Diseño del sistema.

Con el flujo de trabajo diseño se modela el futuro sistema que da soporte a los requisitos funcionales planteados en la etapa de la captura de requisitos. También se aterrizan todos los aspectos con las restricciones y características del sistema como lo son: el lenguaje de programación a utilizar, el sistema operativo donde se podrá ejecutar la aplicación, las tecnologías de interfaz de usuario, en fin, agrupar en el diseño los requisitos no funcionales definidos.

3.3.1 -Modelo del diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación.

3.3.2 -Diagrama de clases del diseño.

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia, todo el código que irá creando las páginas, así como el contenido dinámico de éstas una vez que estén en el navegador del cliente. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. Al tratar de utilizar el diagrama de clases tradicional para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación de una extensión al modelo de análisis y diseño, que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML.

Este sistema tiene concebido por cada caso de uso lo siguiente:

- ✓ Un paquete **Interfaces JS**, en el cual van a estar todos los ficheros JavaScript del framework

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

ExtJS, y los restantes ficheros para la creación de las páginas clientes (PC_) y los formularios de la aplicación.

- ✓ La clase interfaz **PC_Portada** será común para todos los casos de usos, ésta es la portada que aparece con un menú de las funcionalidades de un usuario después que se autentica en el sistema. A partir de esta página se inician todos los casos de uso.
- ✓ La clase controladora **Index** es el controlador frontal de la aplicación, la misma la provee el framework Symfony. Esta será la encargada de la seguridad de cada módulo y el direccionamiento de las peticiones del cliente al servidor.
- ✓ El paquete **Symfony Core** contiene las clases del núcleo de Symfony que son usadas por la aplicación. Entre estas se encuentran sfController y sfAction.
- ✓ El paquete **Abstracción de la BD**, contiene todas las clases que genera Symfony para la abstracción de la base de datos mediante Creole, posibilitando que la aplicación sea independiente del gestor de base datos.

El diagrama de clases Web, fue definido, a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para Web, a continuación se muestran los diagramas de clases para los distintos Casos de Uso.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

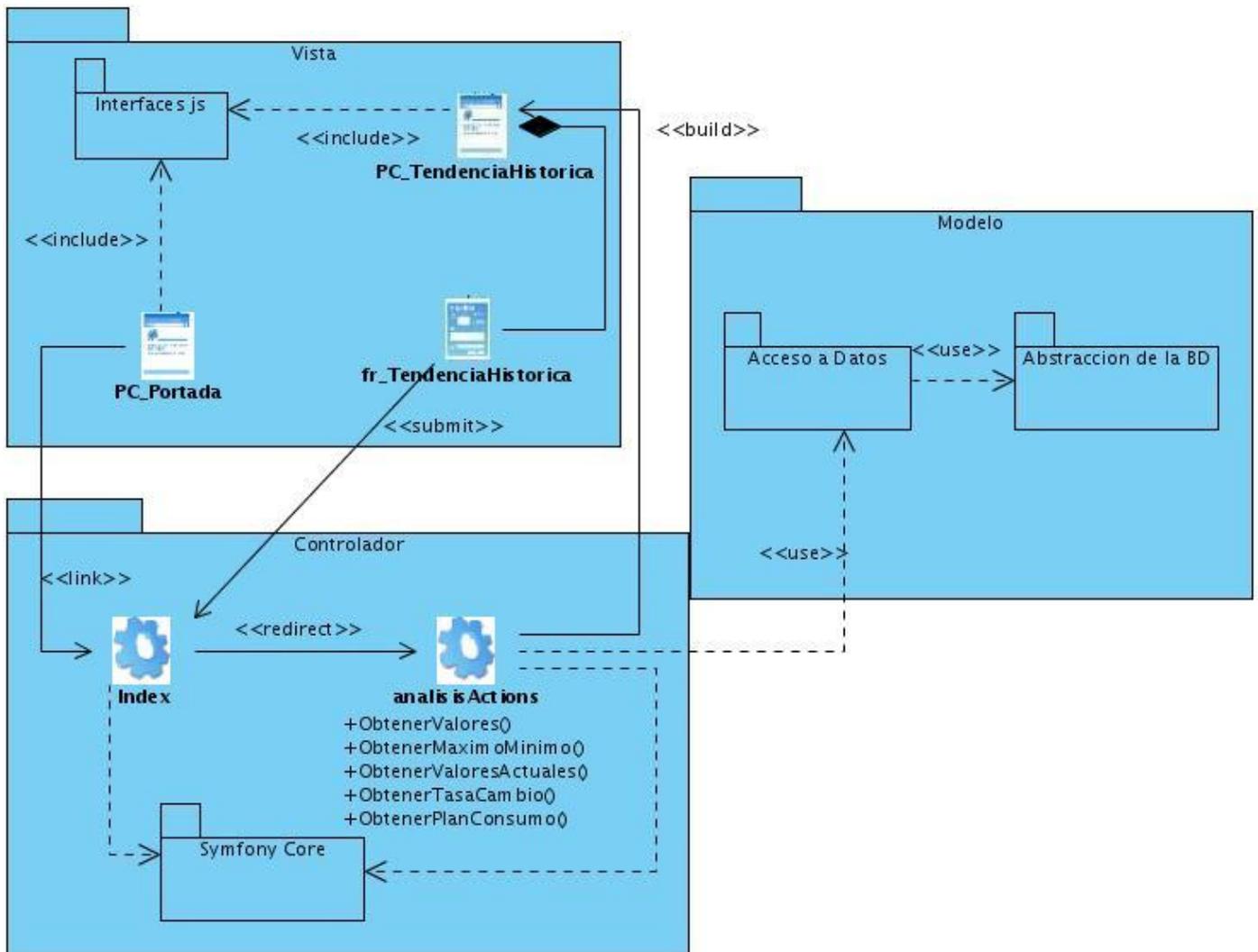


Figura 12: Diagrama de Clases del Diseño Caso de Uso “Analizar Tendencia Histórica”.

Para el caso de uso Analizar Tendencia Histórica la especificación del paquete de Acceso a datos es el que se muestra en la Figura 13.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

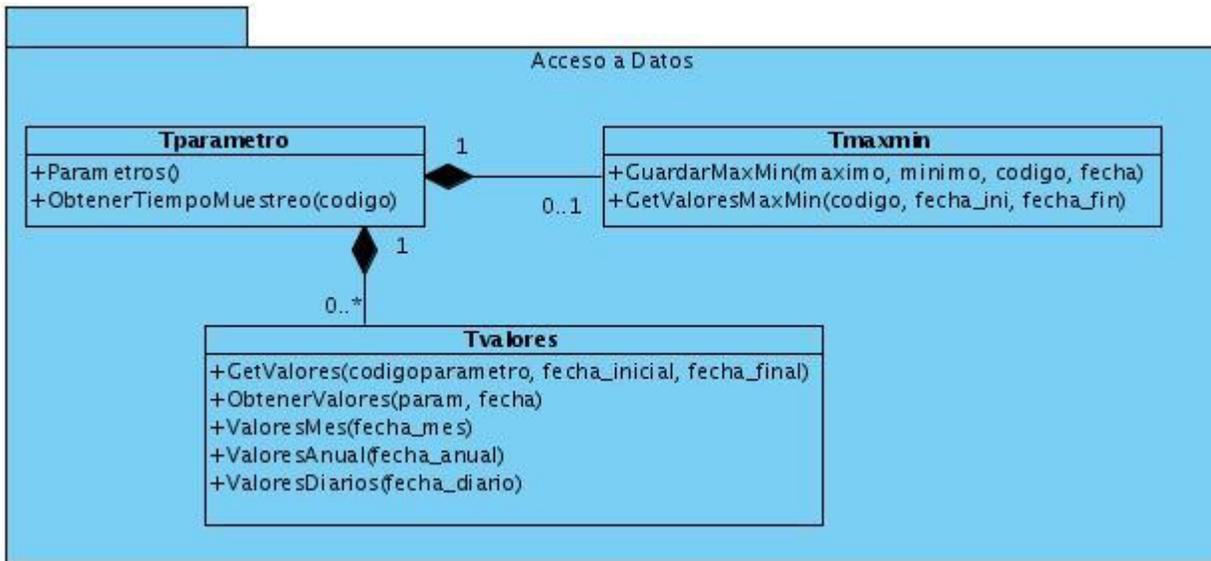


Figura 13: Paquete de Acceso a Datos Caso de Uso “Analizar Tendencia Histórica”.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

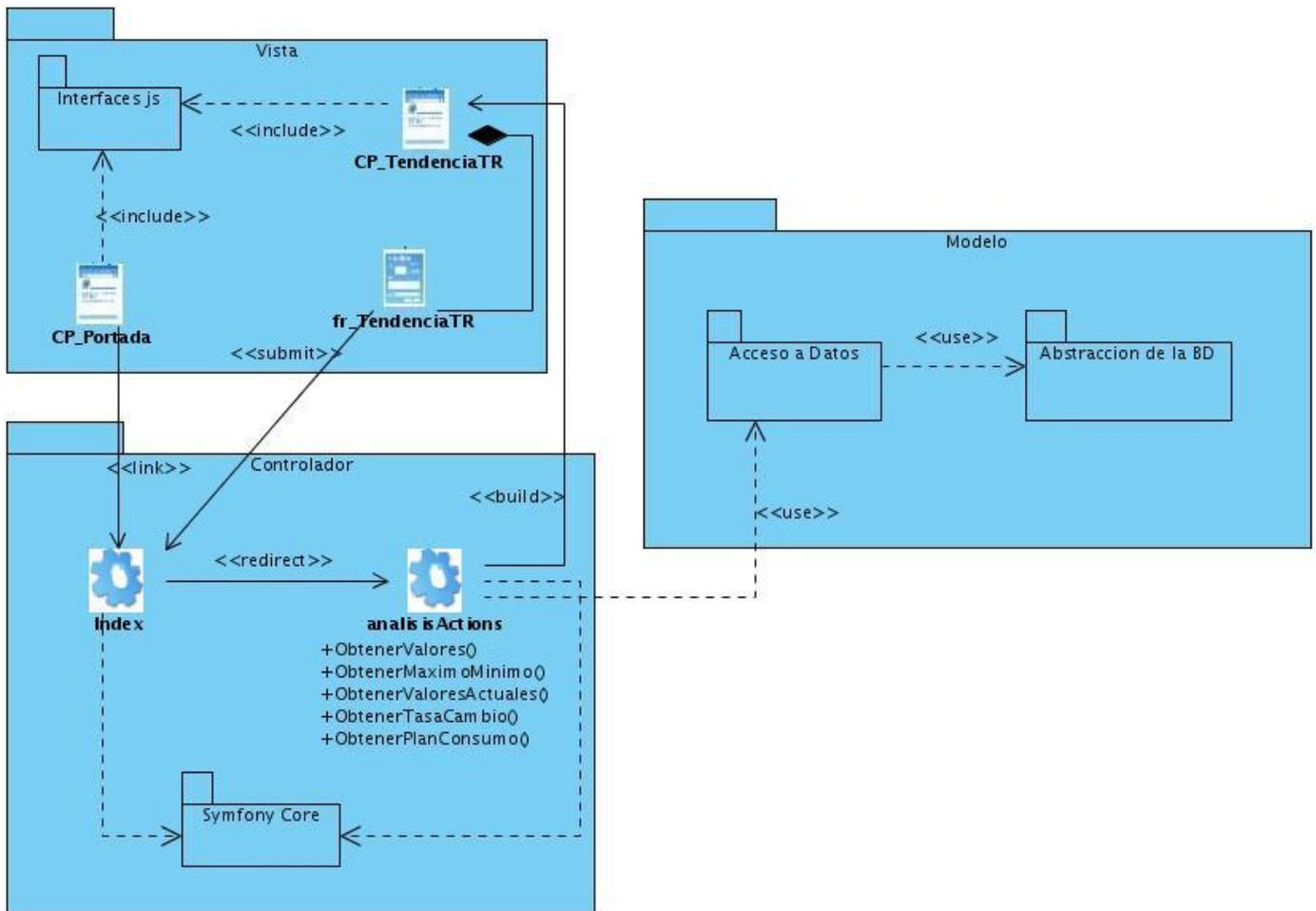


Figura 14: Diagrama de Clases del Diseño Caso de Uso “Analizar Tendencia Tiempo Real”.

Para el caso de uso Analizar Tendencia Tiempo Real la especificación del paquete de Acceso a datos es el que se muestra en la Figura 15.

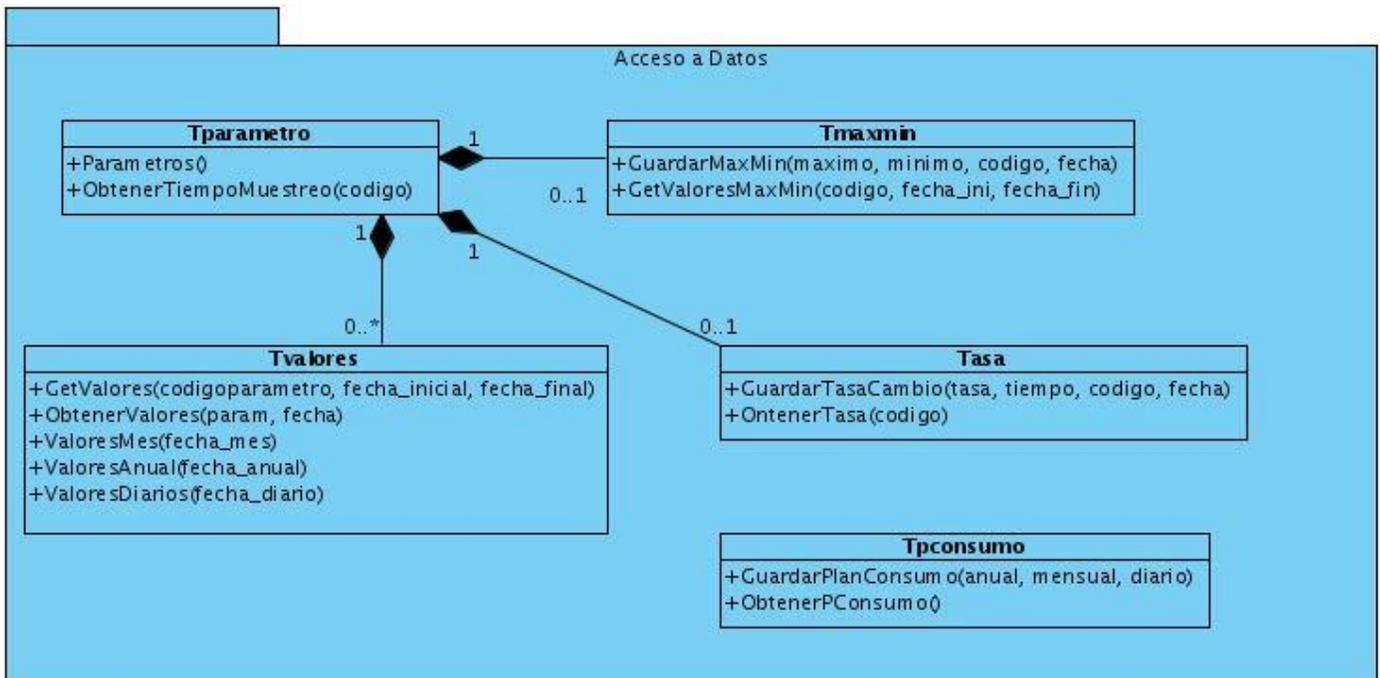


Figura 15: Paquete Acceso a Datos del Caso de Uso “Analizar Tendencia Tiempo Real”.

Los restantes diagramas de clases de diseños de los casos de uso Establecer Plan de Consumo, Establecer Tasa de Cambio y Establecer Máximo y Mínimo se encuentran en el [Anexo 1](#).

3.3.3 -Diagrama de Interacción.

Los Diagramas de Interacción no son más que la forma de representar el modo en que interactúan los objetos en una secuencia de tiempo, además de los mensajes que se envían entre ellos ordenadamente. En UML los Diagramas de Interacción pueden representarse a través de los Diagramas de Secuencia y/o Diagramas de Colaboración.

Para la construcción de los Diagramas de Interacción para los casos de uso, se seleccionó el Diagrama de Secuencia, porque muestra de forma más detallada la interacción entre los objetos.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

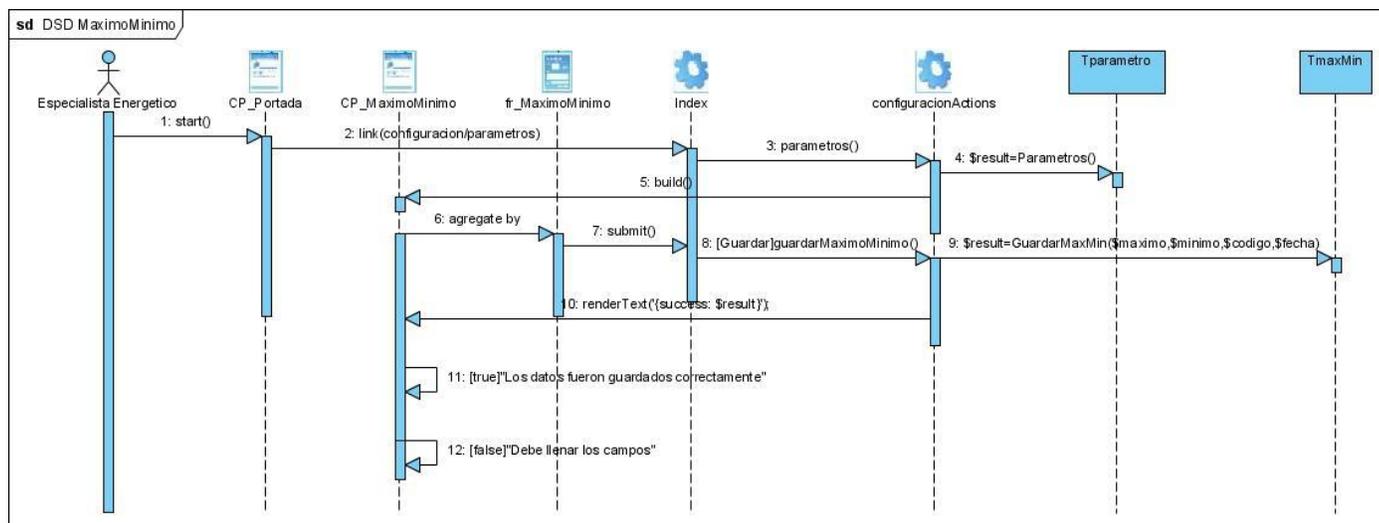


Figura 16: Diagrama de Secuencia Caso de Uso Establecer Máximo y Mínimo.

Los Diagramas de Secuencia de los restantes casos de uso se encuentran en el [Anexo 2](#).

3.4 -Descripción de las clases.

Como parte de los artefactos generados por el proyecto Supervisión Energética, se encuentra la documentación del código fuente de la aplicación, el cual fue generado con la herramienta phpDocumentor. El mismo brinda una descripción detallada de todas las clases con sus métodos y atributos (Ver adjunto “Documentación Análisis Tendencia”).

3.5 -Diseño de la Base de Datos.

3.5.1-Diagrama Entidad Relación de la BD.

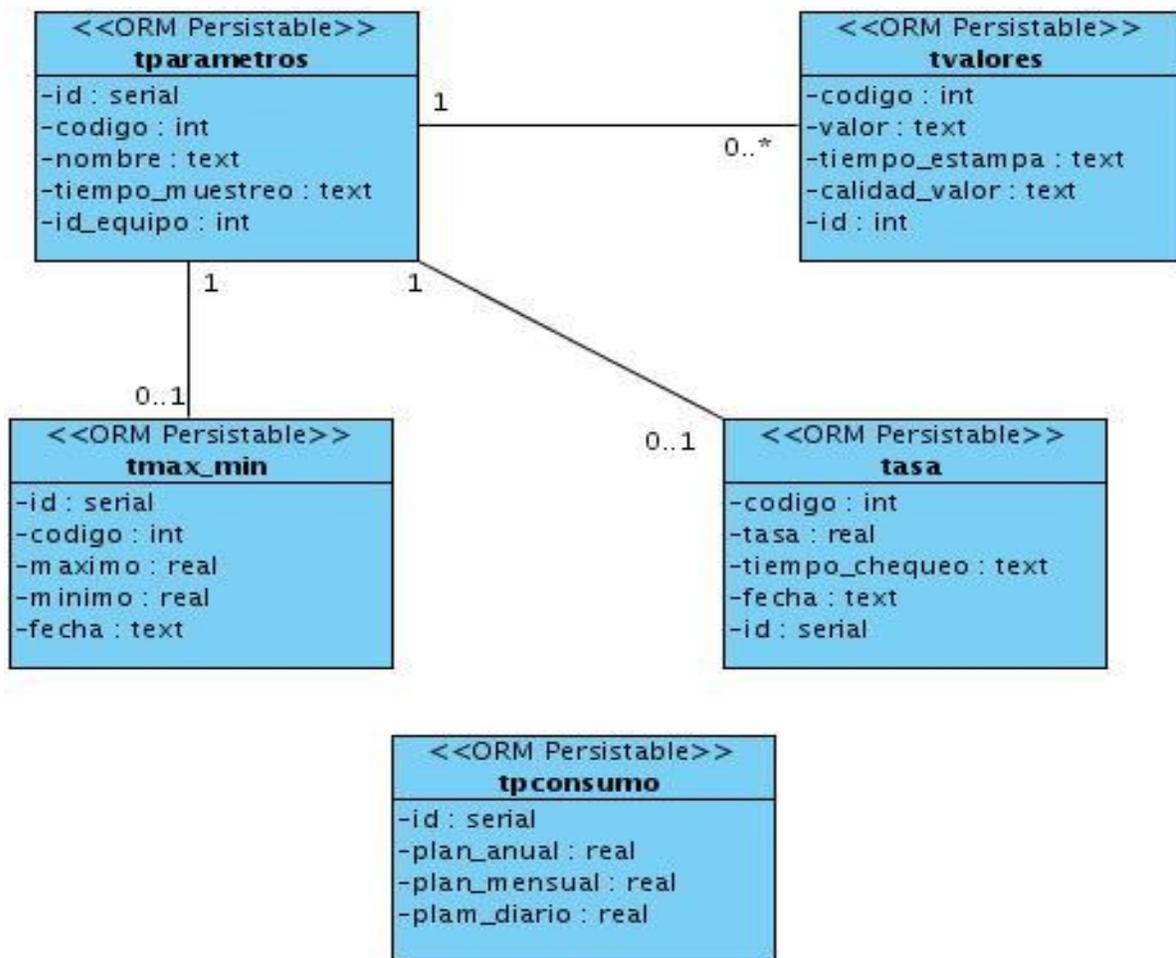


Figura 17: Diagrama de Clases Persistentes.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

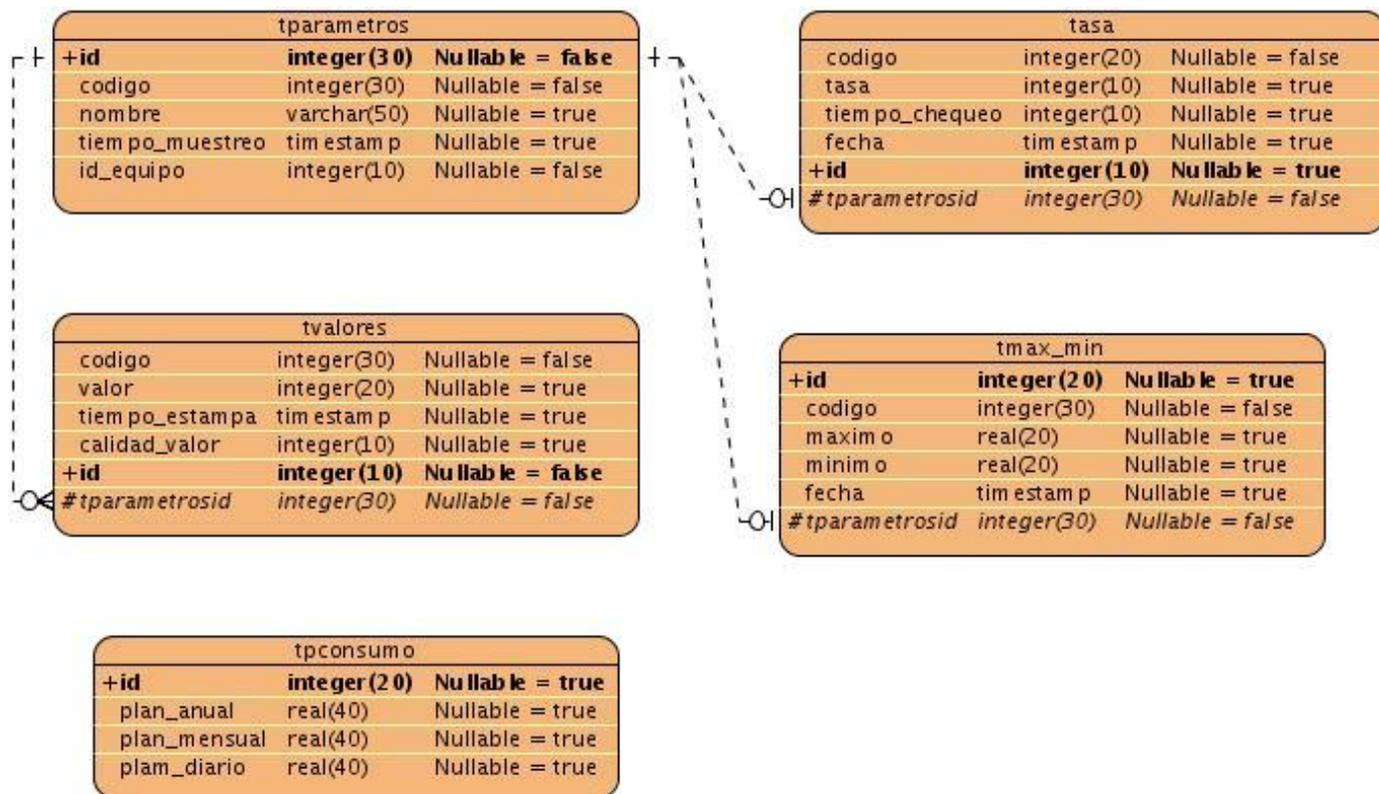


Figura 18: Diagrama Entidad Relación.

3.5.2-Descripción de las tablas.

Nombre:	tparámetros.	
Descripción:		
Atributo	Tipo	Descripción
id	integer(30)	Identificador del parámetro
código	integer(30)	Código del parámetro

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

nombre	varchar(50)	Nombre del parámetro
tiempo_muestreo	timestamp	Tiempo de muestreo del parámetro
id_equipo	integer(10)	Id del equipo de donde fue leído el parámetro

Nombre:	tvalores	
Descripción:		
Atributo	Tipo	Descripción
código	integer(30)	Código del parámetro
valor	integer(20)	Valor del parámetro
Tiempo_estampa	timestamp	Tiempo en el que fue insertado el valor en la Base de Datos.
calidad_valor	integer(10)	Calidad del valor (1 o 2)
id	integer(10)	Id del valor

Nombre:	tpconsumo	
Descripción:		
Atributo	Tipo	Descripción
id	integer(30)	Id del plan de consumo
plan_anual	real(40)	Valor del plan anual convenido con la UNE
plan_mensual	real(40)	Valor del plan mensual convenido con la UNE
plan_diario	ireal(40)	Valor del plan diario convenido con la UNE

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Nombre:	tasa	
Descripción:		
Atributo	Tipo	Descripción
código	integer(30)	Código del parámetro
tasa	integer(10)	Tasa de cambio del parámetro
tiempo_chequeo	integer(10)	Cada que tiempo se chequeara la tasa de cambio
fecha	timestamp	Fecha de introducción de la tasa en la BD.
id	integer(10)	Id de la tasa.

Nombre:	tmax_min	
Descripción:		
Atributo	Tipo	Descripción
código	integer(30)	Id del plan de consumo
máximo	real(20)	Valor del plan anual convenido con la UNE
mínimo	real(20)	Valor del plan mensual convenido con la UNE
fecha	timestamp	Valor del plan diario convenido con la UNE
id	integer(20)	Id del máximo y mínimo.

3.6-Patrones de diseño utilizados por Symfony.

Según Christopher Alexander, “un *patrón* describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema de tal modo que se puede aplicar esta solución un millón de veces, sin hacer lo mismo dos veces” [13].

- **Gang of Four (Patrones de diseño orientado a objetos)**

- **Singleton:**

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es usado en Symfony, para permitir el acceso desde cualquier lugar de la aplicación a los objetos relacionado con el núcleo del framework.

- **Chain of Responsibility:**

Evita acoplar el emisor de una petición a su receptor, al dar a más de un objeto la posibilidad de responder a la petición. Crea una cadena con los objetos receptores y pasa la petición a través de la cadena hasta que esta sea tratada por algún objeto. Permite que más de un objeto pueda manejar una petición.

- **Patrones GRASP (Patrones de Software para la Asignación de Responsabilidades)**

- **Creador:**

En las clases Actions se encuentran las acciones definidas para el Análisis de Tendencia de parámetros energéticos y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

- **Experto:**

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Este es uno de los más utilizados, puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsular la lógica de los datos, y generar las clases con todas las funcionalidades comunes de las entidades.

- Alta Cohesión:

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo, la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

- Bajo Acoplamiento:

La clase Action hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

● Otros:

- Front Controller:

Todas las peticiones Web son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo, con la URL entrada por el usuario.

- Row Data Gateway:

Brinda una instancia por registro en una tabla o vista, a diferencia de Table Data Gateway que tiene un solo objeto para todos los registros de la tabla.

- Table Data Gateway:

Propone brindar un objeto que actúe como Gateway a una tabla, vista o conjunto de tablas, en una base de datos manejando todos los registros de esa tabla. El Gateway contiene un conjunto de operaciones (crear, actualizar, eliminar) con el código SQL (Lenguaje de Consulta Estructurado) necesario para manipular los datos en la tabla correspondiente.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.7-Tratamiento de errores.

Para garantizar la estabilidad y confiabilidad de un sistema, es imprescindible llevar a cabo un detallado tratamiento de excepciones y realizar validaciones, siempre que se introduzcan datos al sistema, de esta forma se impide que el mismo pueda presentar comportamientos inesperados, o falle al tratar de procesar datos incorrectos.

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que en éste se registra y muestra.

Algunos errores serán mostrados en forma de mensajes de texto en la misma página donde se ejecutó la acción, de forma que el usuario pueda corregir más fácilmente el error y continuar. Se utilizan mensajes de confirmación para acciones de guardados de datos.

Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios. Cuando se introduce información en un formulario y faltan datos, sale un cuadro de alerta indicando que faltan campos por llenar. Cuando se deja un campo que es obligatorio llenar, se marca el campo en rojo y se muestra un mensaje cuando se pasa el puntero del mouse por encima, anunciando que el campo es obligatorio y así no se deja ejecutar la acción en vano. En los campos numéricos no se permite introducir letras.

3.8 -Seguridad.

El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios en la aplicación, identificando al usuario antes de que este pueda realizar cualquier acción sobre el mismo; garantizando que la información sea vista y manipulada únicamente por quien tiene derecho a hacerlo. Antes de ser ejecutada, cada acción pasa por un filtro especial que verifica si el usuario actual tiene privilegios de acceder a la acción requerida.

Symfony posee un mecanismo de escape, que puede ser activado mediante un parámetro en un fichero de configuración, para evitar la inyección de códigos hechos en JavaScript que pueden impedir el uso normal de una aplicación. El uso de Propel ayuda a prevenir el tipo de ataque conocido como SQL

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Injection, que permite realizar acciones con el objetivo de corromper la información de un sistema, o ganar acceso al mismo través de la entrada de sentencias SQL maliciosas por medio de formularios, o como parámetros de la petición en la URL (Uniform Resource Locator).

3.9 - Interfaz.

El correcto diseño de una interfaz de usuario es una tarea que ha adquirido una gran relevancia en el desarrollo de sistemas, pudiera definirse como interfaz de usuario a: el conjunto de trabajos y pasos que seguirá el usuario durante todo el tiempo que se encuentre en relación con el software, detallando lo que verá y escuchará en cada momento, las acciones que realizará, así como las respuestas que el software brindará.

La calidad con que sea desarrollada la interfaz de usuario es uno de los principales motivos para lograr el éxito o el fracaso de una aplicación, es por ello que en gran medida uno de los principales aspectos de la usabilidad es la consistencia de su interfaz de usuario y por tanto, la misma debe ser lo más amigable posible y debe proporcionarle funciones fáciles de aprender al usuario, para que se sienta mejor identificado con la misma, siendo esta, el medio por el cual se interactúa con el sistema.

Para el desarrollo de la interfaz de usuario de la siguiente aplicación se tuvieron en cuenta los siguientes aspectos:

- Lograr que los usuarios realicen las acciones de forma fácil.
- Interfaz sencilla para que el usuario en ningún momento se sienta desorientado.
- Previsiones de errores y manejo de errores de entrada de datos por parte del usuario.
- Interfaz sencilla con colores discretos, predominando en el fondo el azul claro, así como para las ventanas; el color blanco y negro para el menú y el color de las letras se reduce a negro.
- Se utilizan la familia de fuentes Verdana, Arial, Helvetica, San-Serif de tamaño 14 y 16 puntos para diferenciar los títulos, los subtítulos, las cabeceras de tabla y el texto, logrando con esto, uniformidad y claridad a la hora de mostrar la información, un estilo apropiado y un entorno agradable para el usuario.

3.10 - Consideraciones finales del capítulo.

En este capítulo fueron tratadas las concepciones básicas para el análisis y diseño del sistema. Se desarrollaron los diagramas de clases del análisis y del diseño. Se realizaron los diagramas de interacción

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

del diseño, además de describir los principios de diseño seguidos, los estándares de la interfaz y la concepción del tratamiento de errores.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1-Introducción.

En el capítulo anterior se modelaron los diagramas de clases de análisis y los diagramas de clases Web. En este capítulo se pasará a la implementación del Software en el lenguaje de programación escogido, se modelarán los diagramas de Despliegues y Componentes para una mayor vista para la implementación del producto, además de realizar casos de pruebas al software obtenido en busca de errores y fallas en el mismo.

4.2-Implementación.

La implementación es la fase más esperada en un proceso de desarrollo de un producto software, es donde se hacen realidad todas las ideas y artefactos que han sido modelados por el equipo de trabajo responsable de la solución; no constituye una etapa independiente y formalmente delimitada en el proceso, ya que la metodología de desarrollo utilizada permite que partes de la solución que conceptualmente estaban definidas y que se podrían ir modelando de forma funcional, se le diera la posibilidad de la implementación mucho antes de que estos artefactos fueran modelados.

4.2.1- Diagrama de despliegue.

El modelo de despliegue contiene los nodos que conforman la topología de hardware sobre la que se ejecuta el sistema. Muestra las relaciones entre el hardware y el software en el sistema final. Se representa como un grafo de nodos unidos por conexiones de comunicación.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes.

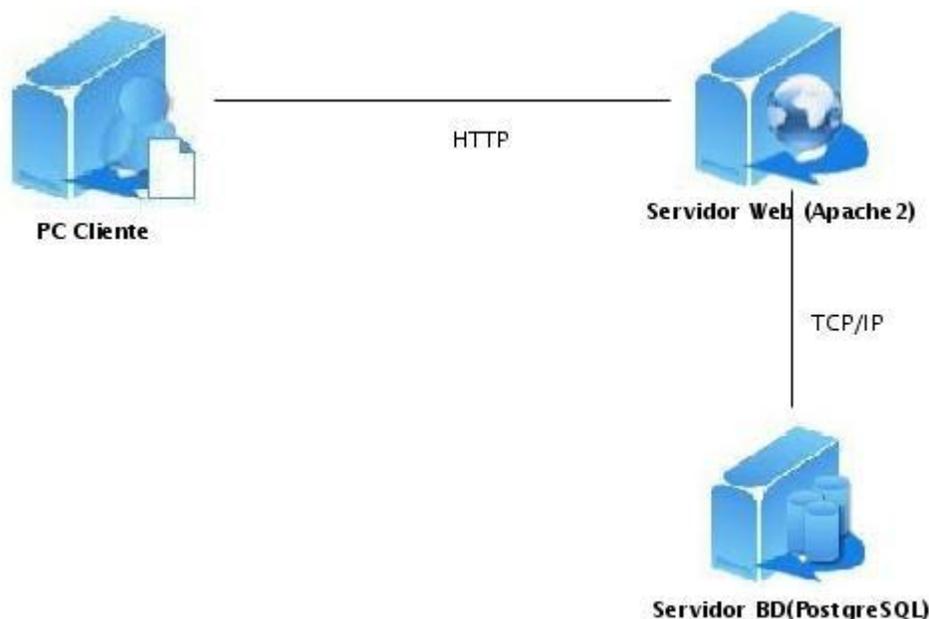


Figura 19: Diagrama de Despliegue.

4.3 - Diagrama de Componentes.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Normalmente contienen componentes, interfaces y relaciones entre ellos, y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo. [14]

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

Dado que los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases, o un diagrama de casos de uso.

Un diagrama de componentes se representa como un grafo de componentes software, unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también que un componente software contiene una interfaz, es decir, la soporta.

Normalmente los diagramas de componentes se utilizan para modelar código fuente, versiones ejecutables, bases de datos físicas, entre otros. Para la construcción de la solución propuesta se decidió dividir el sistema en un conjunto de paquetes que se muestran a continuación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

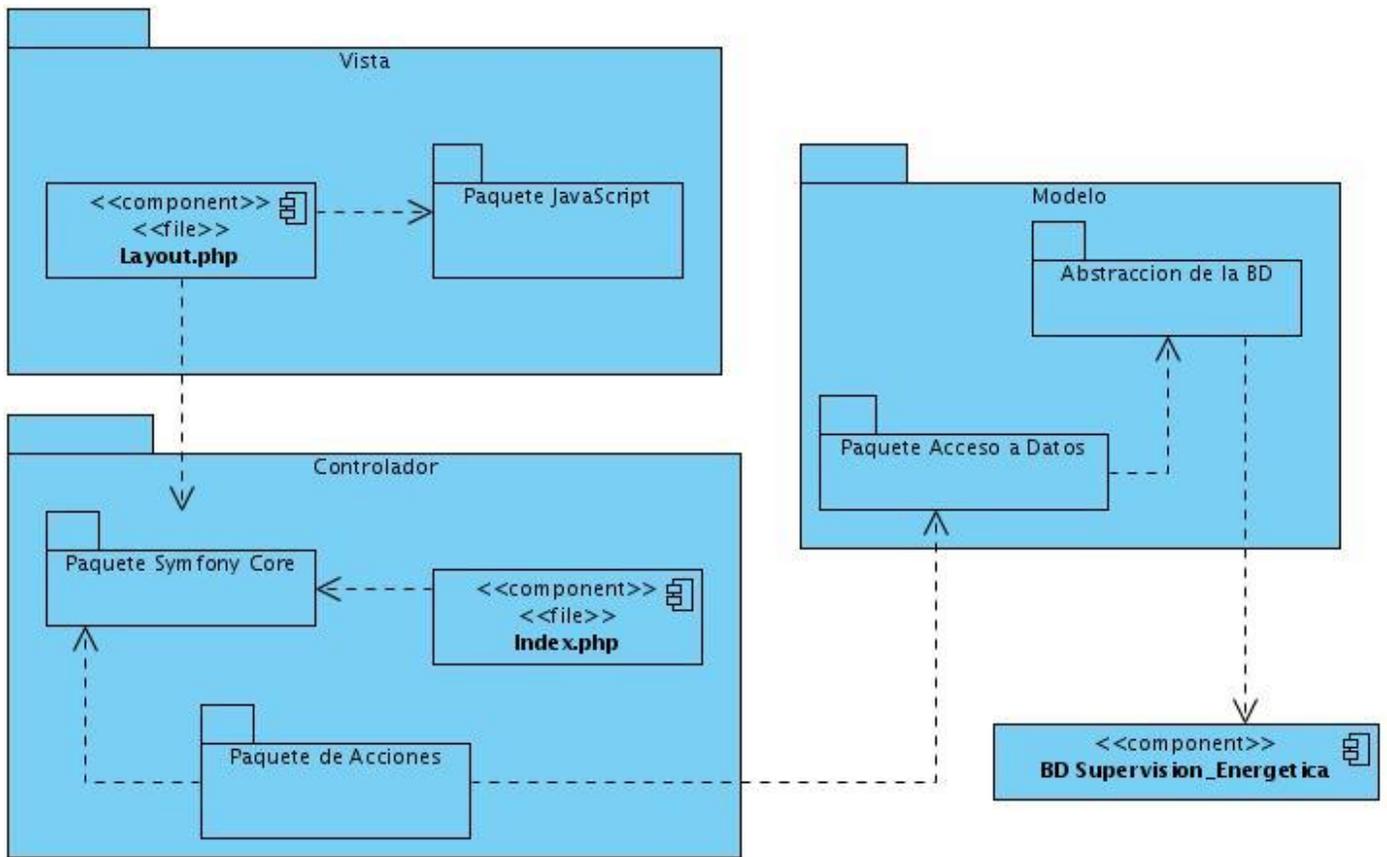


Figura 20: Diagrama de Componentes General.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

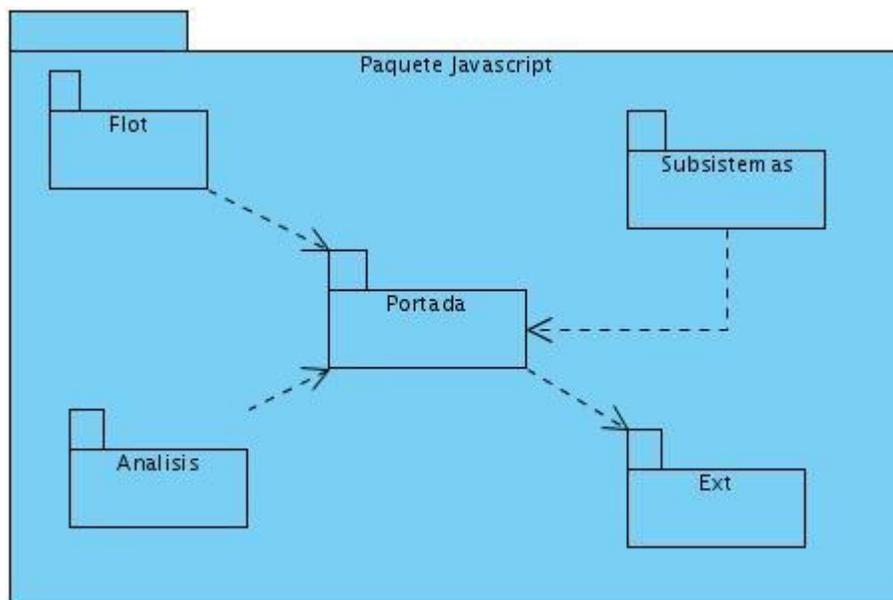


Figura 21: Paquete JavaScript.

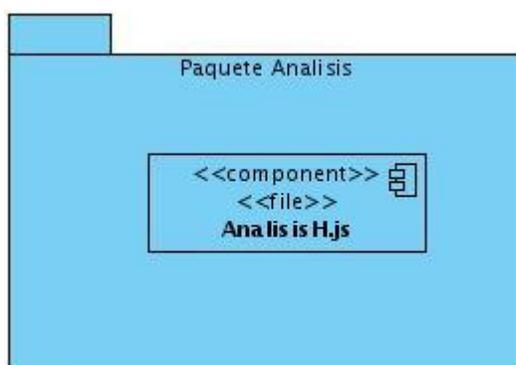


Figura 22: Paquete Análisis.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

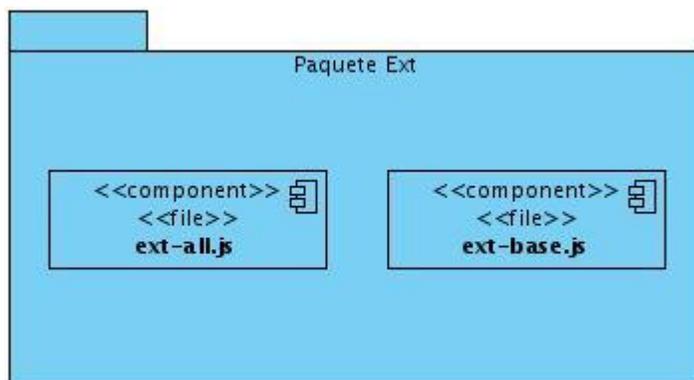


Figura 23: Paquete Ext.

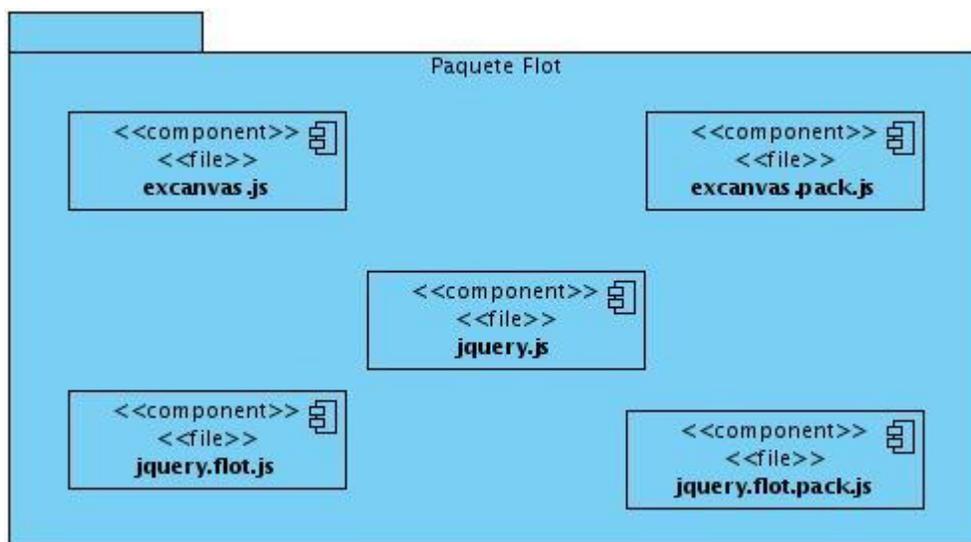


Figura 24: Paquete Flot.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

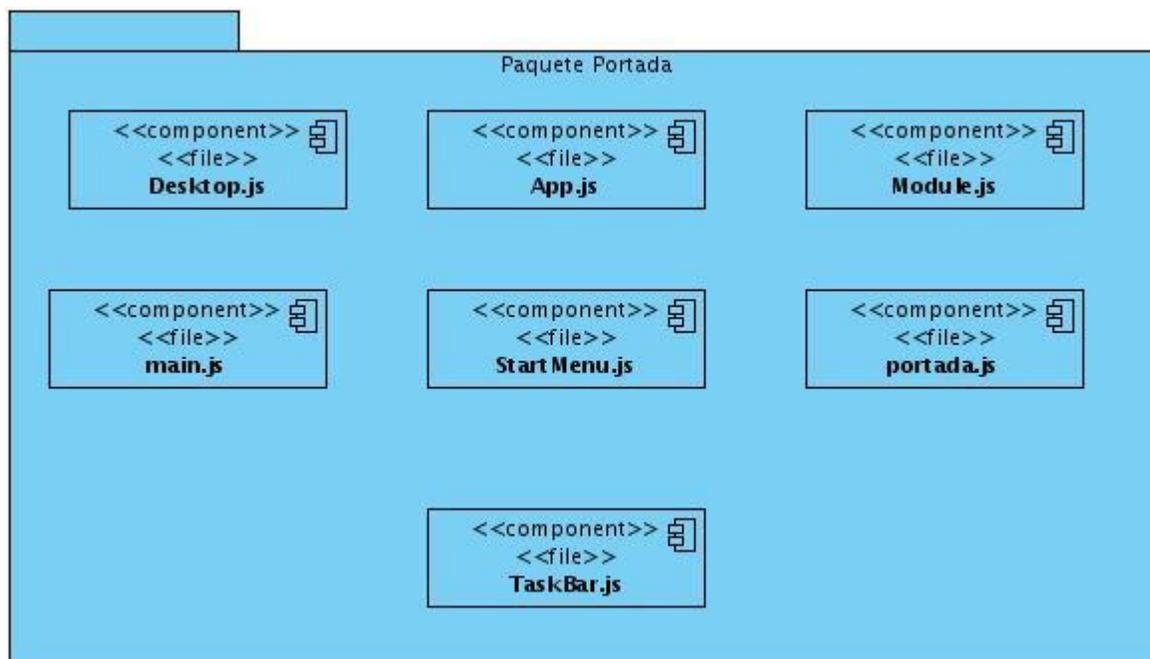


Figura 25: Paquete Portada.

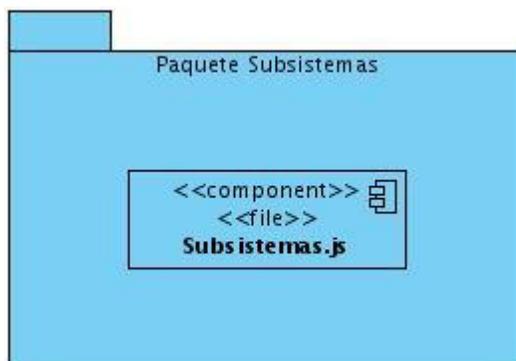


Figura 26: Paquete Subsistemas.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

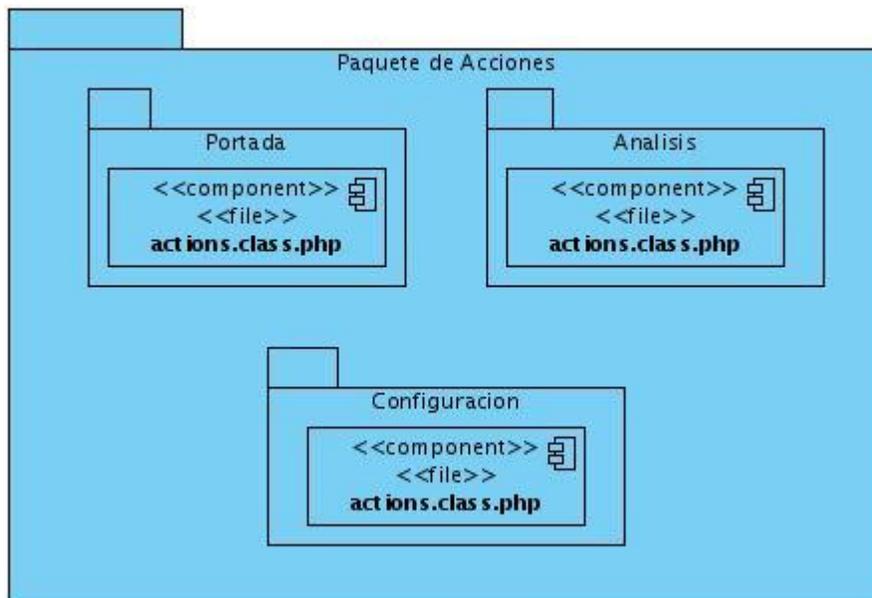


Figura 27: Paquete de Acciones.

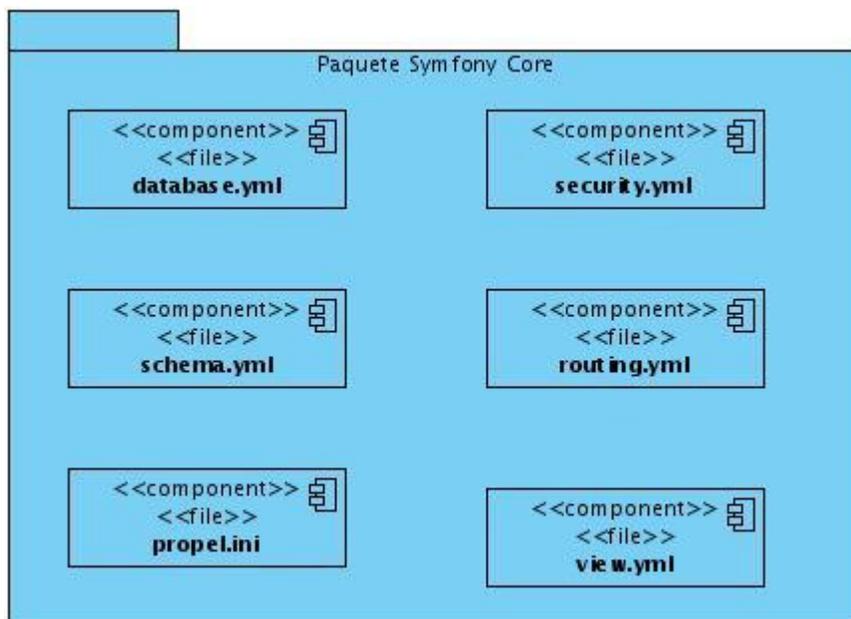


Figura 28: Paquete Symfony Core.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

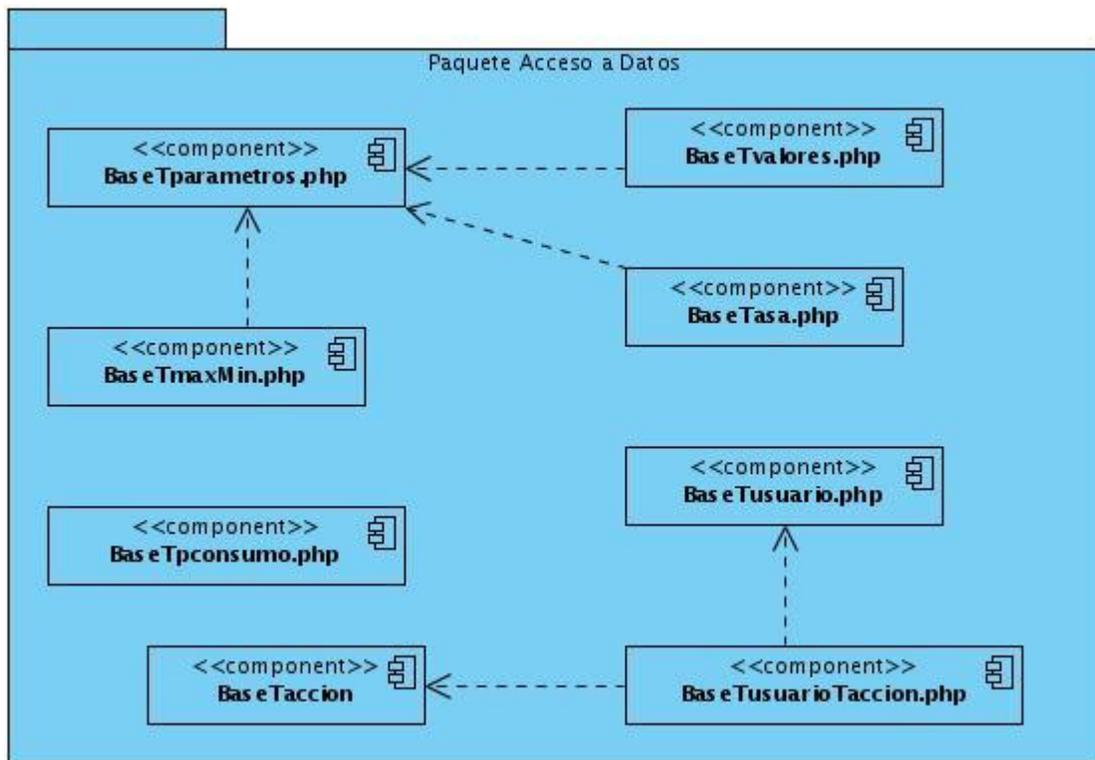


Figura 29: Paquete Acceso a Datos.

4.4 - Prueba.

Las pruebas de software son un elemento crítico para la garantía de calidad del software, y representa una revisión final de las especificaciones, del diseño y de la codificación. Las pruebas se realizan a lo largo del desarrollo del sistema y no simplemente al final. Esto significa sacar a la luz problemas no conocidos y no demostrar la perfección de programas manuales o equipo. Un buen caso de prueba es aquel que tiene una probabilidad muy alta de descubrir un nuevo error y tiene éxito si lo descubre. Se debe diseñar y ejecutar juegos de prueba que, de forma sistemática, detecten distintos tipos de error en el menor tiempo y esfuerzo posible.

Cualquier proceso de ingeniería puede ser probado de mas de dos formas: Se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa. Se pueden desarrollar pruebas que aseguren que la operación interna se ajusta a las especificaciones y que todos los componentes

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

internos se han comprobado de forma adecuada. La primera aproximación se denomina prueba de la caja negra y la segunda prueba de la caja blanca.

Diferentes tipos de pruebas:

La prueba de unidades se plantea a pequeña escala, y consiste en ir probando uno a uno los diferentes módulos que constituyen una aplicación.

Las pruebas de integración se centran en probar la coherencia semántica entre los diferentes módulos, tanto de semántica estática (se importan los módulos adecuados; se llama correctamente a los procedimientos proporcionados por cada módulo), como de semántica dinámica (un módulo recibe de otro lo que esperaba). Normalmente estas pruebas se van realizando por etapas, englobando progresivamente más y más módulos en cada prueba.

Por último, las pruebas de aceptación son las que se plantea el cliente final, que decide qué pruebas va a aplicarle al producto antes de darlo por bueno y pagarlo. De nuevo, el objetivo del que prueba es encontrar los fallos lo antes posible, en todo caso antes de pagarlo y antes de poner el programa en producción.

Al software implementado durante todo el desarrollo del presente trabajo se le aplicaron las pruebas de Caja Negra. [15]

4.4.1-Pruebas de la Caja Negra.

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Las pruebas de caja negra están especialmente indicadas en

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, ficheros, canales de comunicaciones, etc, etc.).

Las pruebas de la Caja Negra se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretende demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta, y la integridad de la información externa se mantiene.
- Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa.
- La prueba de la caja negra intenta encontrar errores de las siguientes categorías:
 - Funciones incorrectas o ausentes.
 - Errores de interfaz.
 - Errores en estructuras de datos o en accesos a bases de datos externas.
 - Errores de rendimiento.
 - Errores de inicialización y de terminación.

Limitaciones:

Lograr una buena cobertura con pruebas de caja negra es un objetivo deseable; pero no suficiente a todos los efectos. Un programa puede pasar con holgura millones de pruebas y sin embargo tener defectos internos que surgen en el momento más inoportuno. Las pruebas de caja negra nos convencen de que un programa hace lo que queremos; pero no de que haga otras cosas menos aceptables.

Las pruebas de la Caja Negra aplicadas al software implementado se encuentran en el [Anexo 3](#).

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4. 5 - Consideraciones finales del capítulo.

En este capítulo es mostrada la topología de hardware sobre la que estará corriendo el sistema a través del modelo de despliegue y fue definida la organización de las clases y objetos en componentes. Además, se le realizaron pruebas de caja negra a cada caso de uso en vista de descubrir posibles errores en la interfaz de la aplicación. A partir de este momento el sistema cuenta con funcionalidad operativa.

CONCLUSIONES GENERALES

El desarrollo de este trabajo ha estado encaminado al desarrollo de una aplicación web para el sistema Supervisor Energético que se desarrolla en la Facultad 5 de la Universidad de la Ciencias Informáticas, con el objetivo de realizar un análisis tanto histórico como en tiempo real suave del comportamiento de los parámetros energéticos en diferentes intervalos de tiempo a partir de mediciones realizadas con anterioridad. La implantación de esta aplicación traerá al país un ahorro de dinero puesto que sustituye en gran medida el tipo de análisis antes realizado, además de que brindará en tiempo real la información necesaria para tomar decisiones que eviten fallas graves en las instalaciones. La aplicación obtenida posibilitará realizar sobre las gráficas históricas que se pueda realizar un *zoom*⁴, posibilitando así saber en detalle cómo se comportó un parámetro en un intervalo de tiempo específico. Al ser una aplicación web no dependerá de aplicaciones Desktop que puedan ralentizar el proceso de análisis de los parámetros eléctricos.

Para el desarrollo de este proyecto se utilizó RUP como metodología de desarrollo y UML como lenguaje representativo, lo cual permitió el modelado de los procesos de negocio y la realización de los artefactos propuestos por RUP que fueron necesarios para la correcta concepción y documentación del proyecto. La implementación se realizó utilizando herramientas de desarrollo en software libre sobre Linux. El uso de la interfaz gráfica de la aplicación no requiere grandes conocimientos informáticos y se brinda la mayor cantidad de información posible en cada una de las pantallas. Por todo lo antes expuesto se concluye que los objetivos propuestos para el presente proyecto se han alcanzado satisfactoriamente. Se incluyen además una serie de recomendaciones que pueden ser útiles para el posterior desarrollo de nuevas funcionalidades.

⁴ Opción que permite la ampliación o reducción del tamaño de una imagen que aparece en pantalla

RECOMENDACIONES

Se debe decir que a pesar de haberle dado cumplimiento a los objetivos planteados al inicio de este trabajo, no se implementaron todas las funcionalidades con las que cuentan herramientas de este tipo por los problemas de tiempo además el propio evolucionar de las tecnologías hace necesario que se recomienden se hagan las siguientes acciones:

- Extender los parámetros y las variables prestas a analizar.
- Confeccionar un sistema de alarmas que cuente con más funcionalidades.
- Hacer que la herramienta pueda realizar acciones sobre el sistema en caso de fallas críticas.
- Implantar este sistema en las entidades que deseen controlar como se están comportando las variables energéticas en tiempo real y no tener que depender de los Analizadores Trifásicos.
- Proseguir con la investigación para garantizar nuevas mejoras en futuras versiones del sistema.

BIBLIOGRAFÍA

Alvarez, Miguel Ángel. Desarrolloweb.com. *Qué es Java Script?* [Online] 2003. [Cited: febrero 16, 2009.] <http://www.desarrolloweb.com/articulos/25.php>.

Accesomedia.com. *Nota de Prensa.* [Online] 2007. [Cited: marzo 10, 2009.] http://www.accesomedia.com/display_release.html?id=38950.

Catalini, Ezequiel. Arquitectura Modelo Vista Controlador. [Online] 2007. [Cited: febrero 16, 2009.] <http://ezequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador>.

Desarrollo de Aplicaciones Web. [Online] 2009. [Cited: febrero 15, 2009.] <http://personales.ya.com/juriver/pfc/htmlhelp/ch01s03.html>.

D'Onofrio, Diego Lucio. elGuille. *Probando software y números de versión.* [Online] 2002. [Cited: marzo 15, 2009.] <http://www.elguille.info/Clipper/probando.htm>.

Eguiluz, Javier. Maestros del web. *El framework Symfony, una introducción práctica.* [Online] 2007. [Cited: febrero 18, 2009.] <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.

González Cristiano, Alberto. MasterMagazine. *AJAX y las nuevas aplicaciones Web.* [Online] 2006. [Cited: febrero 16, 2009.] <http://www.mastermagazine.info/articulo/9475.php>.

Kynetia. *RUP - Rational Unified Process.* [Online] 2007. [Cited: febrero 18, 2009.] <http://www.kynetia.es/calidad/rup-rational-unified-process.html>.

Moreno Martínez, Gerardo. Monografías.com. *Ingeniería de SoftwareUML.* [Online] 2006. [Cited: febrero 18, 2009.] <http://www.monografias.com/trabajos5/insof/insof.shtml>.

Raga, Charlis. Ilustrados.com. *Base de Datos.* [Online] 2004. [Cited: febrero 17, 2009.] <http://www.ilustrados.com/publicaciones/EpZVVZpAuAZcHFchUj.php>.

Symfony.es. *Symfony 1.0, la guía definitiva ,Capítulo 2. Explorando el interior de Symfony ,2.1. El patrón MVC .* [Online] 2008. [Cited: marzo 10, 2009.] <http://www.symfony.es>.

Van Der Henst, Cristhian. Maestros del Web. *¿Qué es el PHP?* [Online] 2001. [Cited: febrero 17, 2009.] <http://www.maestrosdelweb.com/editorial/phpintro/>.

Ventajas de las Herramientas CASE. [Online] 2000. [Cited: febrero 19, 2009.] <http://www.csi.map.es/csi/silice/Sgcase5.html>.

Visual Paradigm. *Visual Paradigm for UML.* [Online] 2007. [Cited: febrero 18, 2009.] <http://www.visual-paradigm.com>.

Wikipedia. *PostgreSQL.* [Online] 2009. [Cited: febrero 17, 2009.] <http://es.wikipedia.org/wiki/Postgres>.

Wikipedia. *Apache(servidor web).* [Online] 2009. [Cited: febrero 17, 2009.] [http://gl.wikipedia.org/wiki/Apache_\(servidor_web\)](http://gl.wikipedia.org/wiki/Apache_(servidor_web)).

Referencias Bibliográficas.

1. **Romagosa Cabús, Jaume.** Mini Proyecto Automatización Industrial. [En línea] 2004. [Citado el: 1 de marzo de 2009.] <http://bibliotecnica.upc.es/bustia/arxiu/40201.pdf>.
2. **Schneider Electric.** Sistemas de Supervisión. [Online] 2003. [Cited: Febrero 5, 2009.] <http://ares.unimet.edu.ve/electrica/automatizacion/PRESENTACIONES/Presentaci%F3n%2026-03-04/Unimet2.pdf>.
3. **El Foro Latino.** *Parámetros Eléctricos Característicos.* [Online] 2008. [Cited: junio 18, 2009.] <http://www.elforolatino.com/f244/parametros-electricos-caracteristicos-4307/>.
4. **Etap Real Time.** Etap Real Time Solución Empresarial. [En línea] 2007. [Citado el: 10 de febrero de 2009.] <http://etap.com>.
5. **Circuitor.** [Online] 2007. [Cited: febrero 12, 2009.] <http://www.circuitor.es>.
6. **Catalini, Ezequiel.** Arquitectura Modelo Vista Controlador. [Online] 2007. [Cited: febrero 16, 2009.] <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador>.
7. **Alvarez, Miguel Ángel.** Desarrolloweb.com. *Qué es Java Script?* [Online] 2003. [Cited: febrero 16, 2009.] <http://www.desarrolloweb.com/articulos/25.php>.
8. **González Cristiano, Alberto.** MasterMagazine. *AJAX y las nuevas aplicaciones Web.* [Online] 2006. [Cited: febrero 16, 2009.] <http://www.mastermagazine.info/articulo/9475.php>.
9. **Van Der Henst, Cristhian.** Maestros del Web. *¿Qué es el PHP?* [Online] 2001. [Cited: febrero 17, 2009.] <http://www.maestrosdelweb.com/editorial/phpintro/>.
10. **Flores, Hector.** slideshare. *Concepto UML.* [Online] 2009. [Cited: marzo 20, 2009.] <http://www.slideshare.net/hectorxxx/camtasia-getting-started-guide-1341569>.
11. **Acosta, Ariel Arencibia.** Monografías.com. *Diseño de un Software.* [Online] 2006. [Cited: febrero 15, 2009.] <http://www.monografias.com/trabajos42/propuesta-software/propuesta-software.shtml>.
12. **Scribd.** *Symfony 1.1 Guía Definitiva.* [En línea] 30 de julio de 2008. [Citado el: 25 de marzo de 2009.] <http://www.scribd.com/doc/4290865/symfony-1-1-guia-definitiva>.

13. **Navarro, Antonio.** Programadores de Hoy. *Arquitecturas: Patrones de diseño: el ser o no ser del modelado actual*. [Online] 2009. [Cited: abril 2, 2009.]
<http://www.programadoresdehoy.com/numeros/000/5.htm>.
14. **Fernandez Vilas, Ana.** *Diagrama de Componentes*. [En línea] 30 de marzo de 2001. [Citado el: 15 de abril de 2009.] <http://tvdi.det.uvigo.es/~avilas/UML/node49.html>.
15. **Mañas, Jose A.** Pruebas de Programas. [Online]
<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s11>.

ANEXO

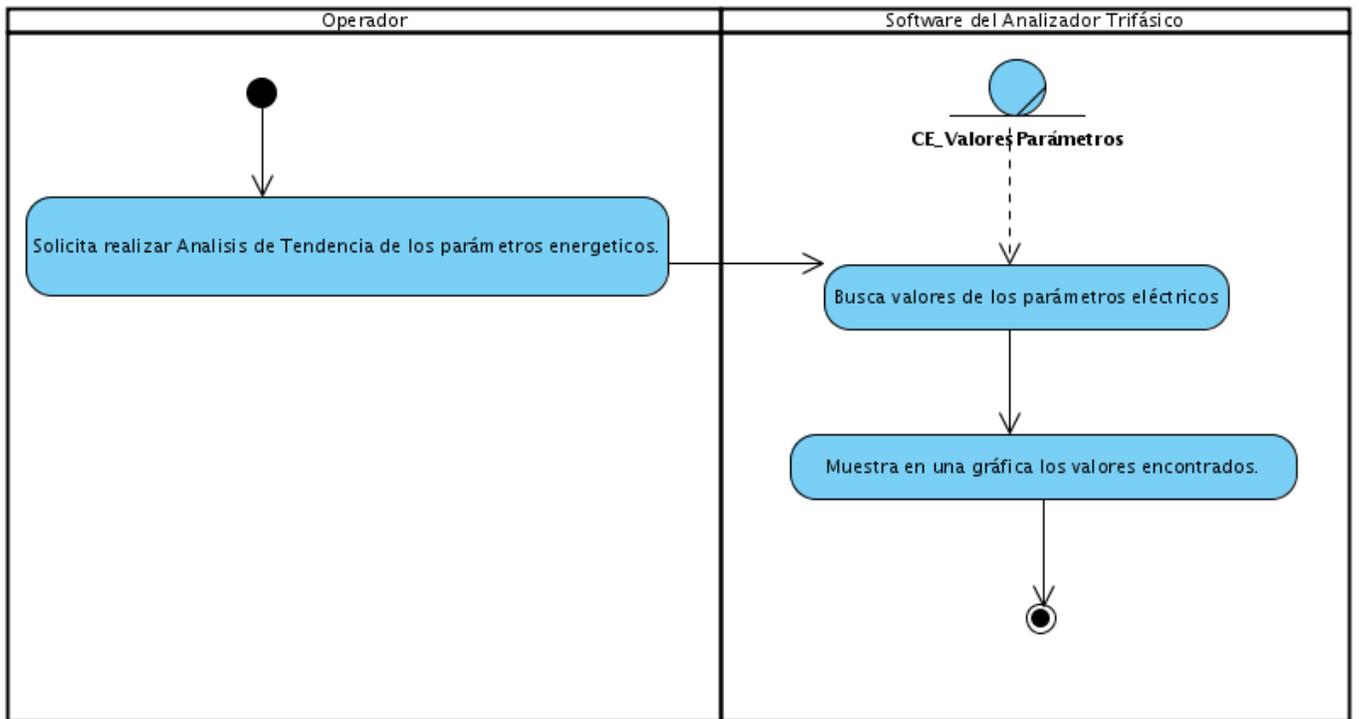


Figura 30: Diagrama de Actividad Caso Uso del Negocio “Analizar Tendencia”.

ANEXO 1

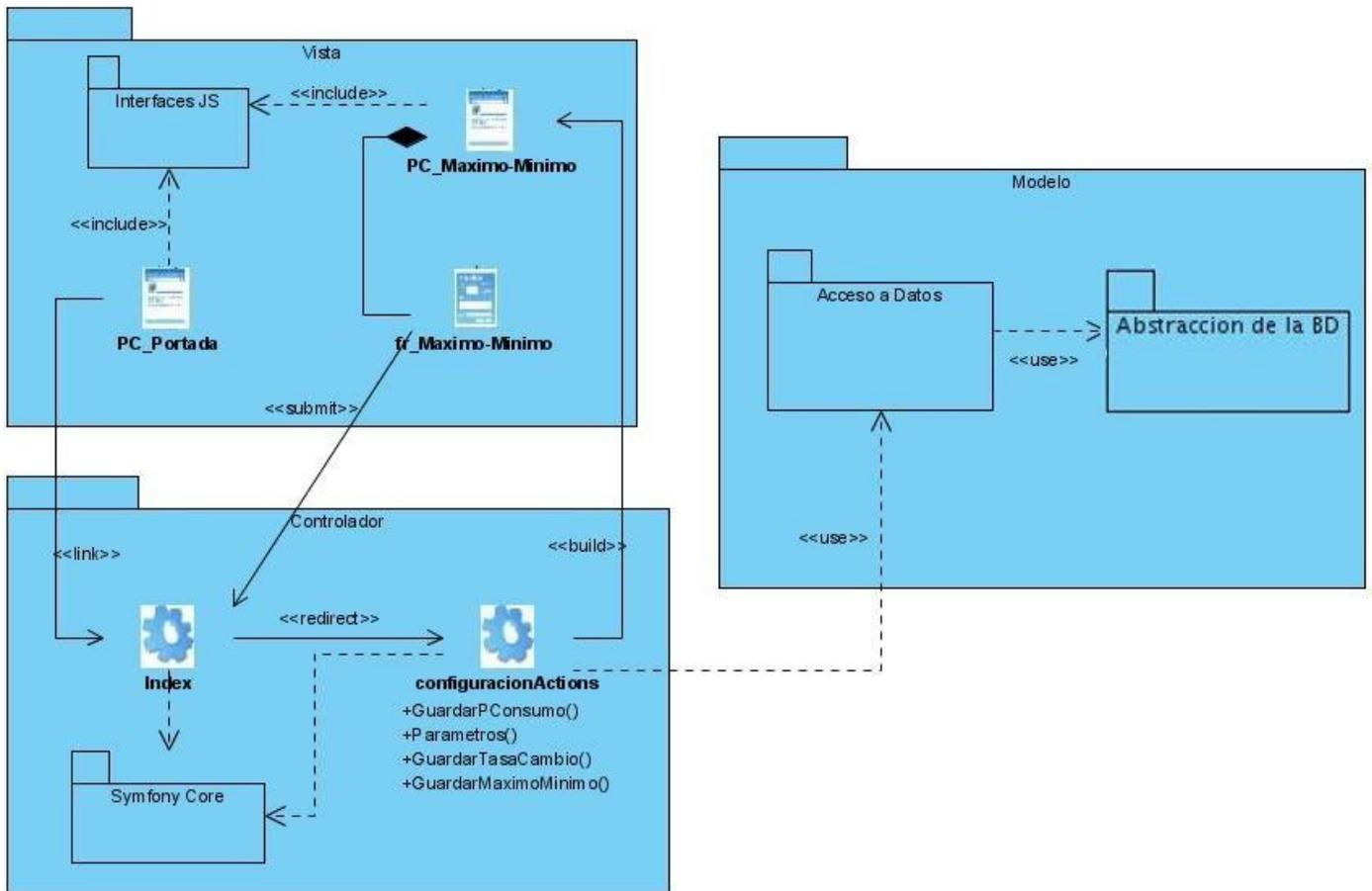


Figura 31: Diagrama de Clases del Diseño Caso de Uso “Establecer Máximo y Mínimo”.

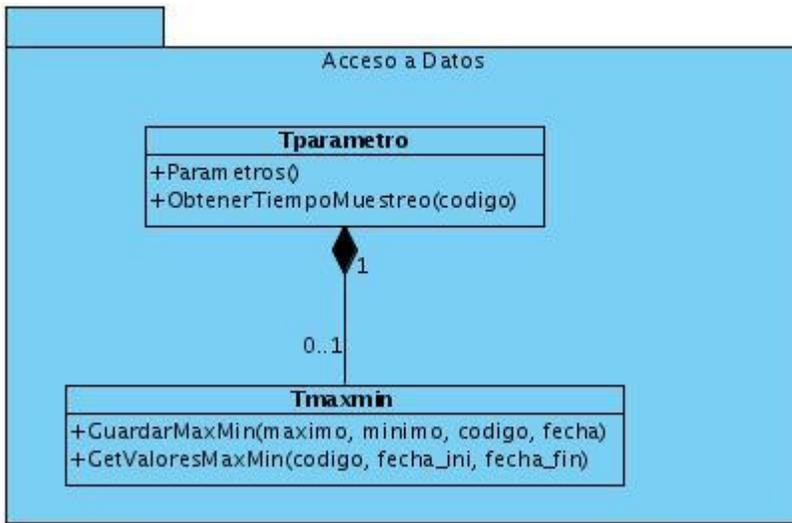


Figura 32: Paquete Acceso a Datos Caso de Uso "Establecer Máximo y Mínimo".

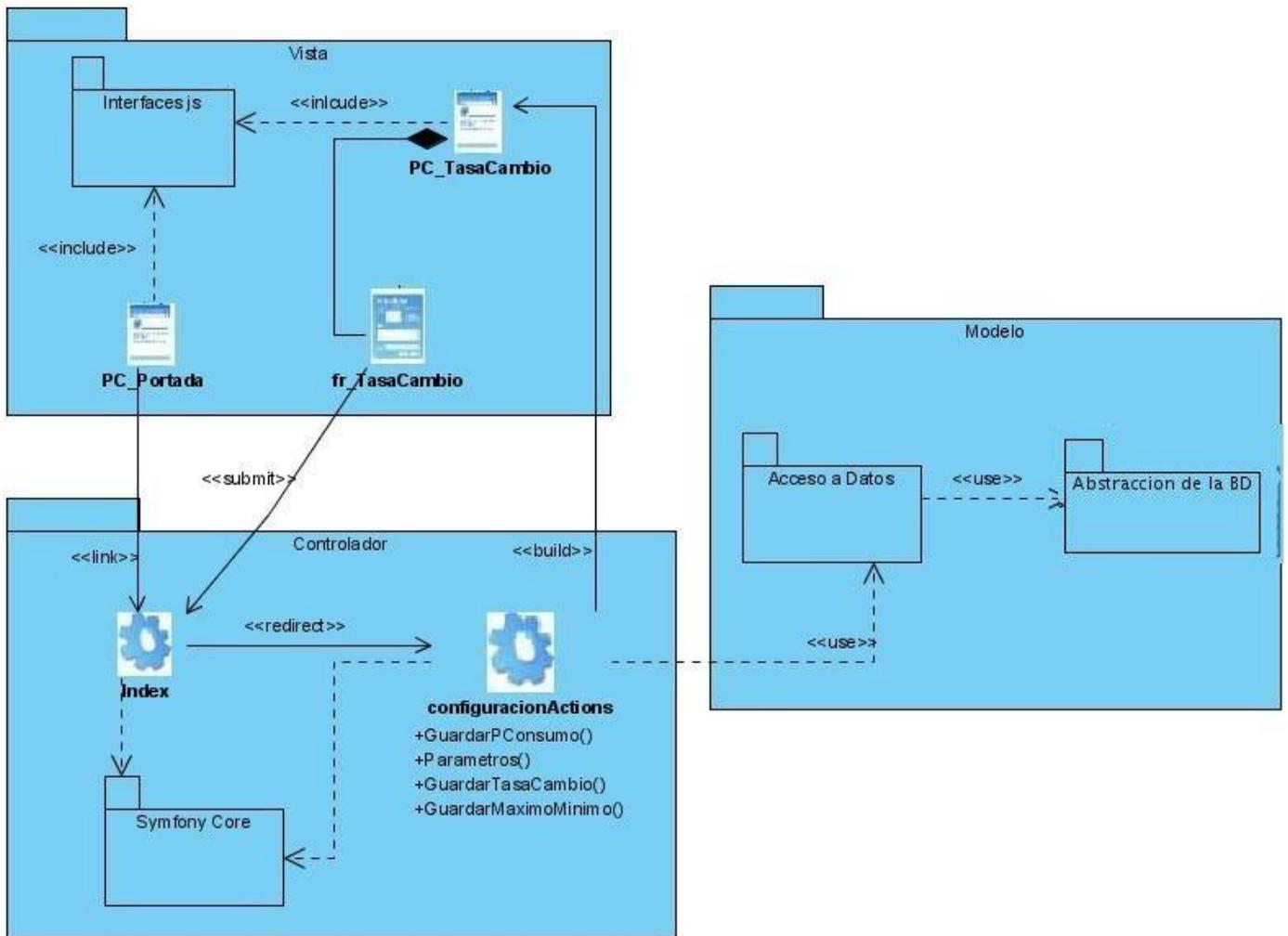


Figura 33: Diagrama de Clases del Diseño Caso de Uso "Establecer Tasa de Cambio".

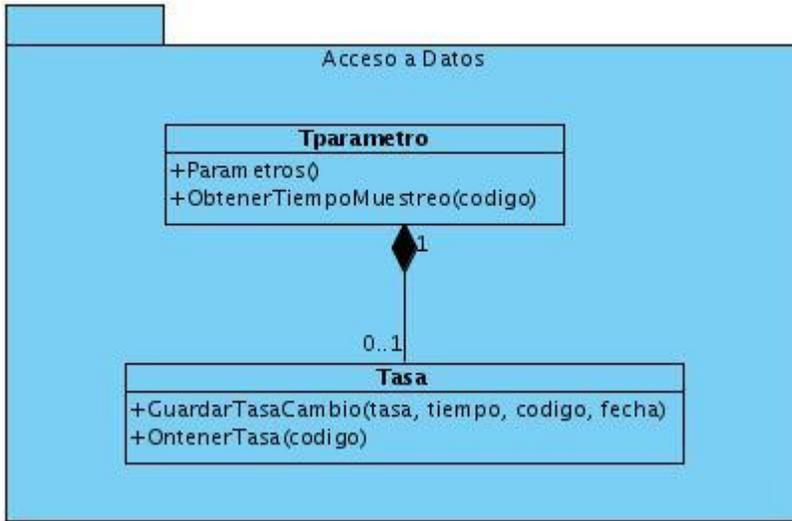


Figura 34: Paquete Acceso a Datos Caso de Uso "Establecer Tasa de Cambio".

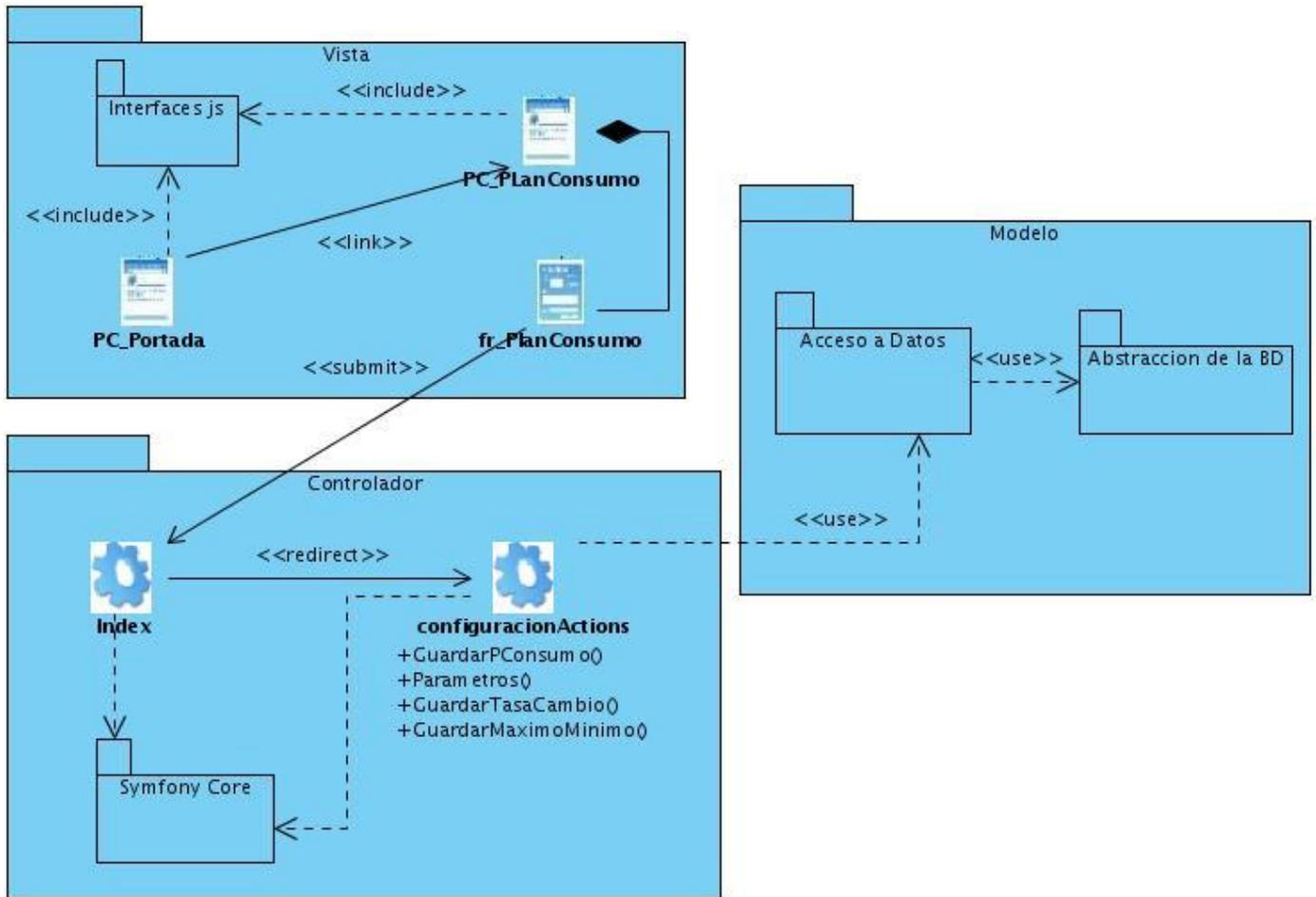


Figura 35: Diagrama de Clases del Diseño Caso de Uso "Establecer Plan de Consumo".

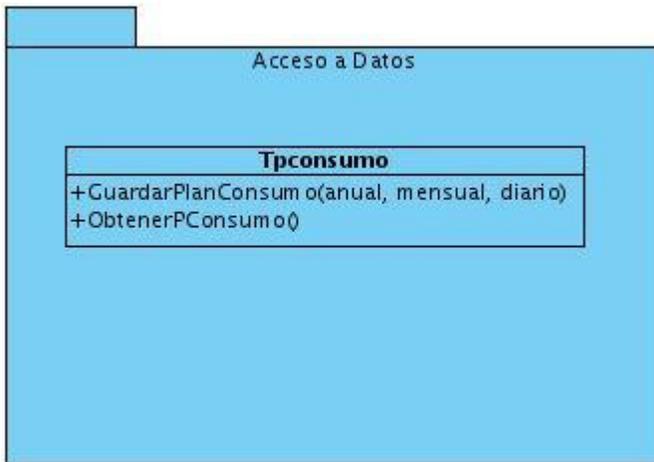


Figura 36: Paquete Acceso a Datos Caso de Uso "Establecer Plan de Consumo".

ANEXO 2

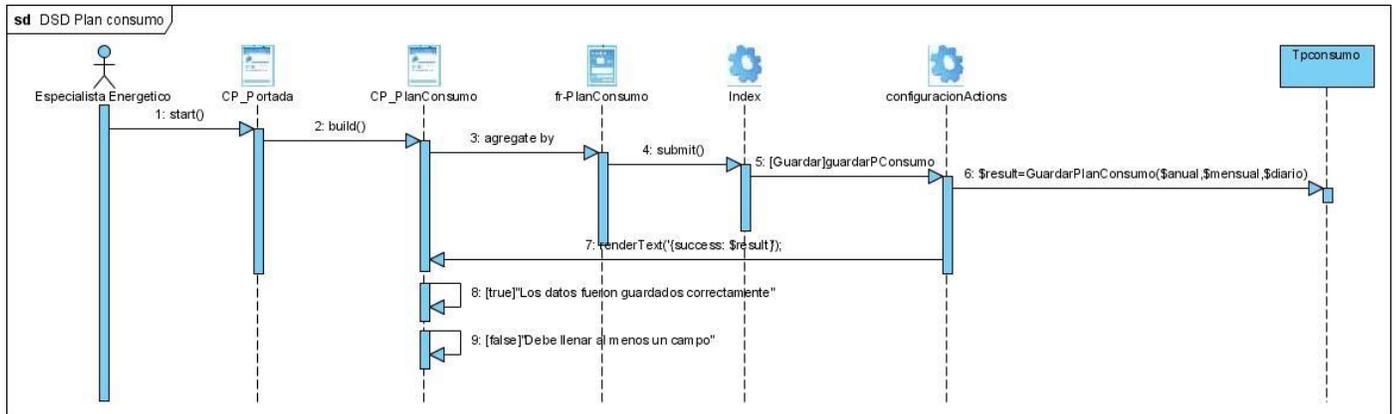


Figura 37: Diagrama de Secuencia Caso de Uso Establecer Plan de Consumo.

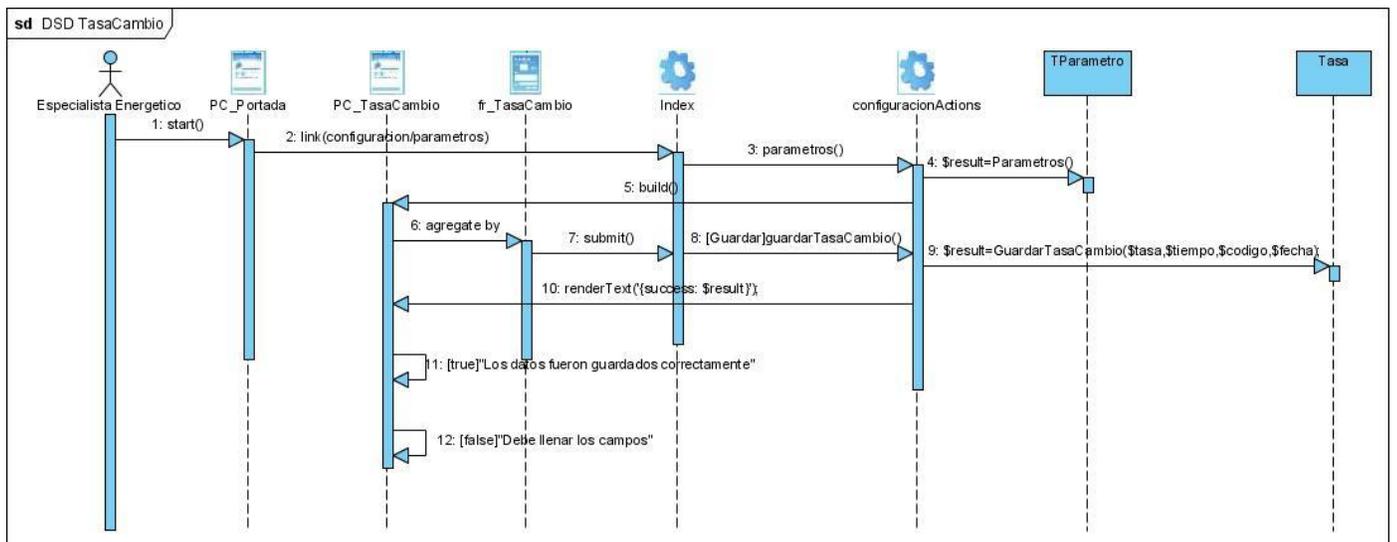


Figura 38: Diagrama de Secuencia Caso de Uso Establecer Tasa de Cambio.

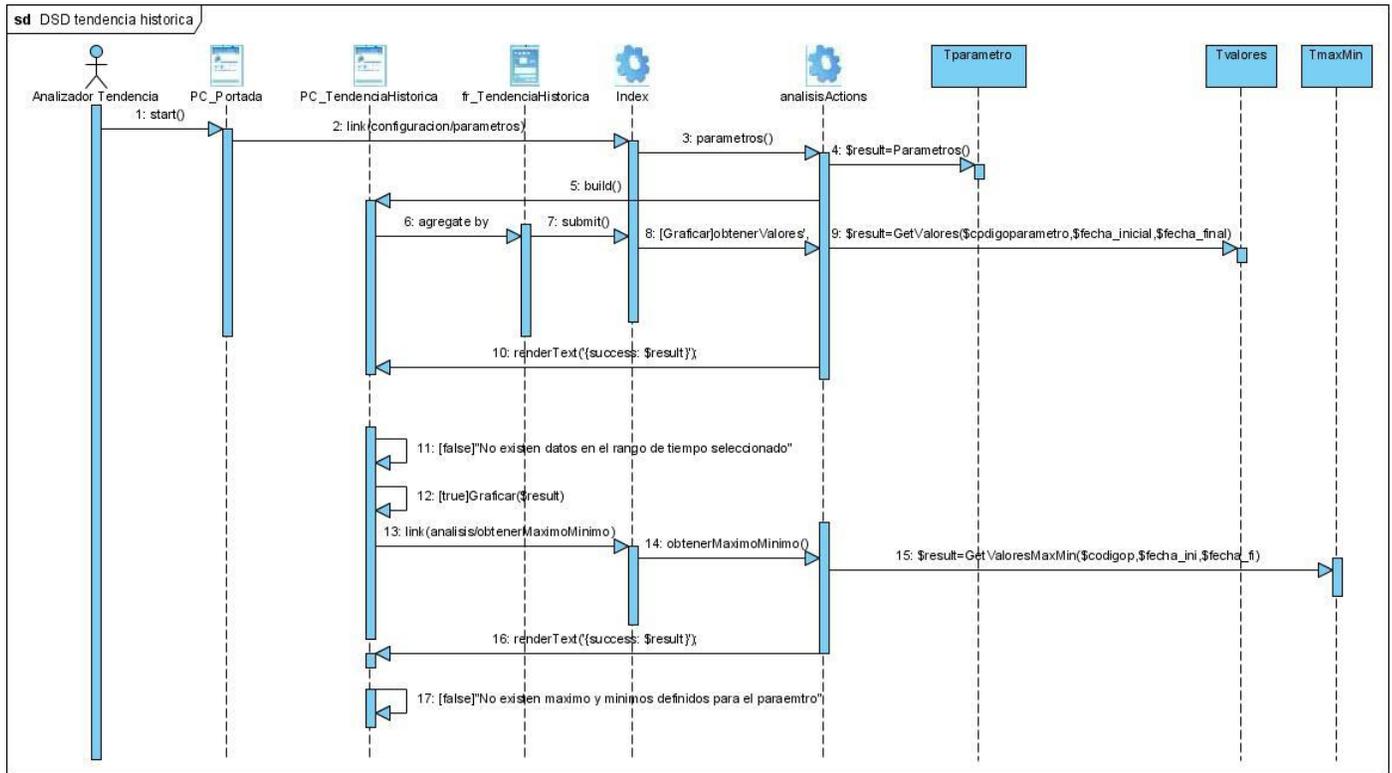


Figura 39: Diagrama de Secuencia Caso de Uso Analizar Tendencia Histórica.

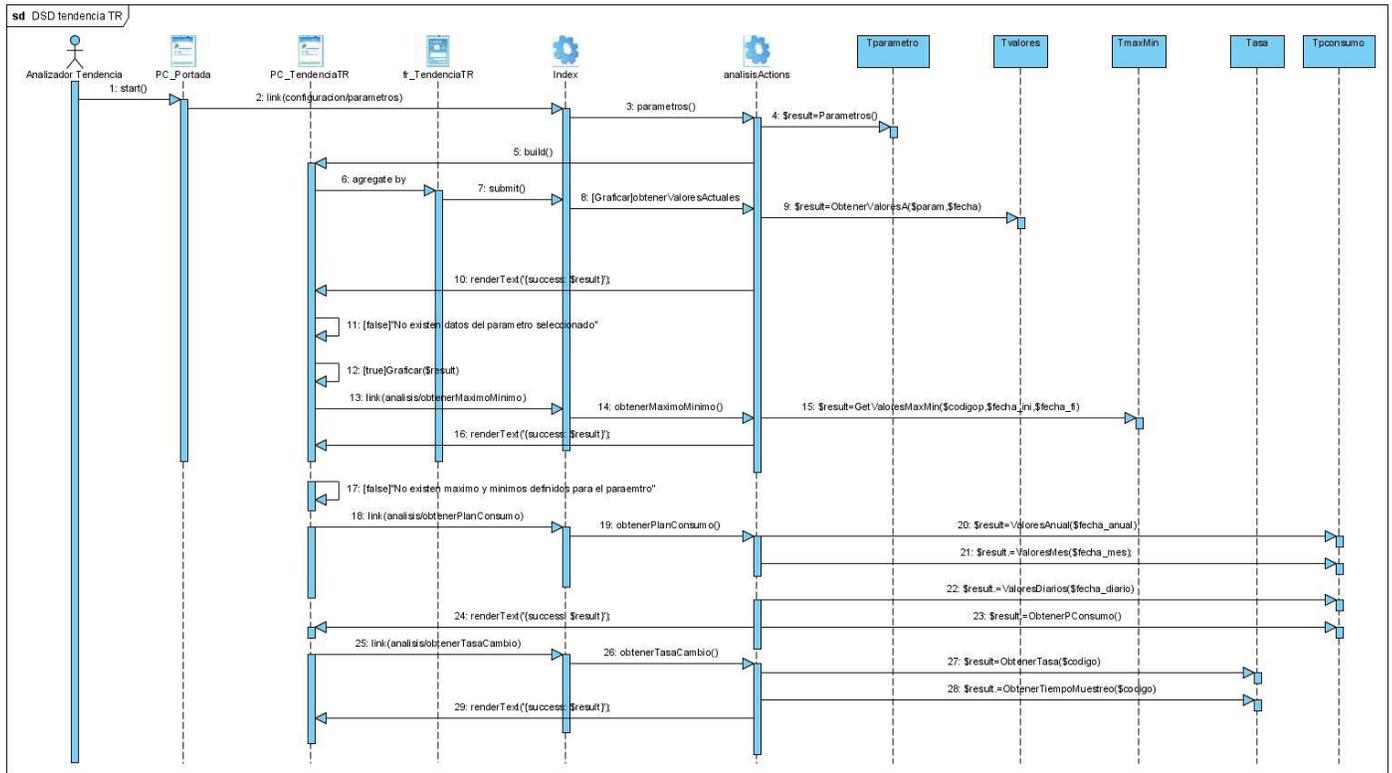


Figura 40: Diagrama de Secuencia Caso de Uso Analizar Tendencia Tiempo Real.

ANEXO 3

Diseño de Casos de Prueba.

Análisis de Tendencia de Parámetros Energéticos.

Caso de Uso Establecer Máximo_Mínimos.

1. Descripción general.

Este caso de uso le permite al usuario poder definir los límites máximos y mínimos para cada parámetro energético.

- CP. #1: Introducir máximo o mínimo.

2. CP. #1: Introducir Máximo o Mínimo.

2.1. Descripción.

Este caso de prueba le permite al usuario introducir máximos y mínimos para un parámetro energético determinado.

2.2. Flujo central.

1. El usuario selecciona la opción Configurar Máximos y Mínimos en el menú Inicio de la aplicación.
2. El sistema muestra una interfaz con un formulario donde el usuario debe seleccionar un parámetro de un listado presentado e introducir el máximo y el mínimo para dicho parámetro.
3. El usuario selecciona el parámetro.
4. El usuario introduce máximo y mínimo.

5. El usuario selecciona "Guardar".
6. El sistema verifica la información.
7. El sistema muestra un mensaje informando que los datos fueron guardados correctamente.

2.3. Condiciones de ejecución.

- ✓ El usuario previamente autenticado en el sistema.
- ✓ Existencia del formulario

2.4. Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>El usuario selecciona un parámetro e introduce el valor máximo y mínimo para ese parámetro y selecciona "Guardar"</p> <p>Ejemplo:</p> <p>Parámetro: Potencia Activa</p>		<p>Se espera que el sistema guarde los datos y muestre un mensaje informando que los datos fueron guardados correctamente.</p>	<p>Al introducir correctamente el máximo y el mínimo el sistema muestra un mensaje informando que el máximo y mínimo fueron guardados correctamente.</p>	

<p>Máximo: 200</p> <p>Mínimo: -200</p>				
	<p>El usuario introduce los valores y no selecciona un parámetro</p>	<p>El sistema debe mostrar un mensaje indicando el error.</p>	<p>El sistema marca en rojo el campo de selección de los parámetros y muestra el mensaje este campo es obligatorio.</p>	
	<p>El usuario introduce letras en los campos de máximo y de mínimo.</p> <p>Ejemplo:</p> <p>Parámetro: Frecuencia.</p> <p>Máximo: Prueba.</p> <p>Mínimo:</p>	<p>El sistema no debe permitir que se escriban letras en un campo numérico.</p>	<p>El sistema no permite que se escriban letras.</p>	

	Prueba.			
--	---------	--	--	--

Diseño de Casos de Prueba.

Análisis de Tendencia de Parámetros Energéticos.

Caso de Uso Establecer Plan de Consumo.

1- Descripción general.

Este caso de uso le permite al usuario configurar los planes de consumo Anual, Mensual y Diario convenidos con la UNE.

- CP. #1 Introducir planes de Consumo.

2- CP. #1: Introducir Planes de Consumo.

2.1- Descripción.

Este caso de prueba le permite al usuario establecer los planes de consumo para días, meses y el año.

2.2- Flujo central.

- El usuario selecciona la opción Configurar plan de consumo en el menú Inicio de la aplicación.
- El sistema muestra una interfaz con un formulario donde el usuario debe introducir los planes de consumo para el mes, el día o el año.
- El usuario introduce los planes de consumo.

- El usuario selecciona “Guardar”.
- El sistema verifica la información.
- El sistema muestra un mensaje informando que los datos fueron guardados correctamente.

2.3- Condiciones de ejecución.

El usuario previamente autenticado en el sistema.

Existencia del formulario

2.4- Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>El usuario introduce los valor para los campos Anual , Mensual o Diario y selecciona “Guardar”</p> <p>. Ejemplo:</p> <p>Anual: 80000552</p> <p>Mensual: 4000623</p>		<p>Se espera que el sistema guarde los datos y muestre un mensaje informando que los datos fueron guardados correctamente.</p>	<p>Al introducir correctamente los planes de consumo el sistema muestra un mensaje informando que los datos fueron guardados correctamente.</p>	

<p>Diario: 10000</p>				
	<p>El usuario no introduce los planes de consumo y selecciona "Guardar".</p>	<p>El sistema debe mostrar un mensaje indicando que no se pueden dejar campos vacios.</p>	<p>El sistema muestra un mensaje informando que debe de llenar al menos un campo.</p>	
	<p>El usuario introduce letras en los campos de los planes de consumo</p> <p>Ejemplo:</p> <p>Anual: lola</p> <p>Mensual: Prueba.</p> <p>Diario: Prueba.</p>	<p>El sistema no debe permitir que se escriban letras en un campo numérico.</p>	<p>El sistema no permite que se escriban letras en los campos de los planes de consumo.</p>	

Diseño de Casos de Prueba.

Análisis de Tendencia de Parámetros Energéticos.

Caso de Uso Establecer Tasa de Cambio.

1- Descripción general.

Este caso de uso le permite al usuario configurar la tasa de cambio para un parámetro determinado y establecer su tiempo de chequeo.

- CP. #1 Introducir tasa de cambio y tiempo de chequeo.

2- CP. #1: Introducir Tasa de Cambio y Tiempo de chequeo.

2.1- Descripción.

Este caso de prueba le permite al usuario establecer la tasa de cambio y el tiempo de chequeo para un parámetro determinado.

2.2- Flujo central.

- El usuario selecciona la opción Configurar Tasa de Cambio del menú Inicio de la aplicación.
- El sistema muestra una interfaz con un formulario donde el usuario debe seleccionar un parámetro de un listado presentado e introducir la tasa de cambio y el tiempo de chequeo para ese parámetro
- El usuario selecciona el parámetro.
- El usuario introduce la tasa de cambio y el tiempo de chequeo para ese parámetro.
- El usuario selecciona "Guardar".
- El sistema verifica la información.

- El sistema muestra un mensaje informando que los datos fueron guardados correctamente.

2.3- Condiciones de ejecución.

El usuario previamente autenticado en el sistema.

Existencia del formulario

2.4- Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>El usuario selecciona un parámetro e introduce los valor para la tasa de cambio y el tiempo de chequeo y selecciona "Guardar"</p> <p>. Ejemplo:</p> <p>Parámetro: Tención.</p> <p>Tasa de Cambio: 400.</p>		<p>Se espera que el sistema guarde los datos y muestre un mensaje informando que los datos fueron guardados correctamente.</p>	<p>Al seleccionar el parámetro e introducir correctamente la tasa de cambio y el tiempo de chequeo el sistema muestra un mensaje informando que los datos fueron guardados correctamente.</p>	

<p>Tiempo de Chequeo: 100.</p>				
	<p>El usuario no selecciona un parámetro e introduce la tasa de cambio y el tiempo de chequeo y selecciona "Guardar";</p>	<p>El sistema debe mostrar un mensaje indicando el error</p>	<p>El sistema marca en rojo el campo de selección de parámetro y muestra un mensaje que este campo es obligatorio.</p>	
	<p>El usuario introduce letras en los campos de la tasa de cambio y el tiempo de chequeo</p> <p>Ejemplo:</p> <p>Tasa de Cambio: lola</p>	<p>El sistema no debe permitir que se escriban letras en un campo numérico.</p>	<p>El sistema no permite que se escriban letras en los campos de los planes de consumo.</p>	

	Tiempo de Chequeo: Prueba.			
--	----------------------------------	--	--	--

Diseño de Casos de Prueba.

Análisis de Tendencia de Parámetros Energéticos.

Caso de Uso Analizar_Tendencia Histórica.

1- Descripción general.

Este caso de uso le permite al usuario seleccionar un parámetro y un rango de fecha para observar los valores del parámetro seleccionado en una gráfica en el rango de fecha introducido.

- CP. #1 Seleccionar Rango de fecha.

2- CP. #1: Seleccionar Rango de Fecha.

2.1- Descripción.

Este caso de prueba le permite al usuario seleccionar un rango de fecha en el cual se buscarán valores del parámetro seleccionado.

2.2- Flujo central.

- El usuario selecciona la opción Análisis del menú Inicio de la aplicación.
- El sistema muestra los dos tipos de análisis que se pueden realizar.

- El usuario selecciona la opción Analizar Tendencia Histórica.
- El sistema muestra una interfaz con un formulario donde el usuario debe seleccionar un parámetro de un listado presentado y escoger un rango de fecha en el cual se buscaran valores del parámetro seleccionado.
- El usuario selecciona el parámetro
- El usuario selecciona el rango de fecha.
- El usuario selecciona “Graficar”.
- El sistema verifica la información.
- El sistema muestra una gráfica con los valores del parámetro seleccionado en el intervalo de tiempo escogido.

2.3- Condiciones de ejecución.

El usuario previamente autenticado en el sistema.
Existencia del formulario

2.4- Iteraciones.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultad de la Prueba	Observaciones
El usuario selecciona un parámetro y selecciona un rango de fecha y selecciona		Se espera que el sistema muestre una gráfica con los valores del parámetro seleccionado en	Al seleccionar el parámetro y el rango de las fechas el sistema muestra una grafica con los	

<p>“Graficar” .</p> <p>Ejemplo:</p> <p>Parámetro: Tención.</p> <p>Fecha Inicial: 01/01/2009</p> <p>Fecha Final: 05/01/2009</p>		<p>el rango de las fechas seleccionadas.</p>	<p>valores obtenidos en el rango de fecha especificado.</p>	
	<p>El usuario no selecciona un parámetro y selecciona el rango de fechas y selecciona “Graficar”;</p>	<p>El sistema debe mostrar un mensaje indicando el error.</p>	<p>El sistema marca en rojo el campo de selección de parámetro y muestra un mensaje que este campo es obligatorio.</p>	
	<p>El usuario selecciona la fecha inicial mayor que la fecha final.</p> <p>Ejemplo:</p>	<p>El sistema no debe buscar valores en ese rango de fecha.</p>	<p>El sistema muestra un mensaje informando que el rango de fecha es incorrecto.</p>	

	Fecha Inicial: 05/01/2009			
	Fecha Final: 01/01/2009			

Diseño de Casos de Prueba.

Análisis de Tendencia de Parámetros Energéticos.

Caso de Uso Analizar Tendencia Tiempo Real Suave.

1- Descripción general.

Este caso de uso le permite al usuario seleccionar un parámetro y observar en una gráfica como se está comportando el parámetro seleccionado en tiempo real suave.

- CP. #1 Seleccionar el parámetro.

2- CP. #1: Seleccionar Parámetro.

2.1- Descripción.

Este caso de prueba le permite al usuario seleccionar un parámetro para observar cómo se está comportando en tiempo real suave.

2.2- Flujo central.

- El usuario selecciona la opción Análisis del menú Inicio de la aplicación.
- El sistema muestra los dos tipos de análisis que se pueden realizar.
- El usuario selecciona la opción Analizar Tendencia Tiempo Real Suave.
- El sistema muestra una interfaz con un formulario donde el usuario debe seleccionar un parámetro de un listado presentado
- El usuario selecciona el parámetro
- El usuario selecciona "Graficar".
- El sistema verifica la información.
- El sistema muestra una gráfica con los valores del parámetro seleccionado en tiempo real suave.

2.3- Condiciones de ejecución.

El usuario previamente autenticado en el sistema.

Existencia del formulario

2.4- Iteraciones.

Clases Válidas	Clases	Resultado	Resultado de la	Observaciones
----------------	--------	-----------	-----------------	---------------

	Inválidas	Esperado	Prueba	
<p>El usuario selecciona un parámetro y selecciona "Graficar".</p> <p>Ejemplo:</p> <p>Parámetro: Tención.</p>		<p>Se espera que el sistema muestre una gráfica con los valores del parámetro seleccionado en tiempo real suave.</p>	<p>Al seleccionar el parámetro el sistema muestra una grafica con los valores obtenidos en tiempo real suave.</p>	
	<p>El usuario no selecciona un parámetro y selecciona "Graficar";</p>	<p>El sistema debe mostrar un mensaje indicando el error</p>	<p>El sistema marca en rojo el campo de selección de parámetro y muestra un mensaje que este campo es obligatorio.</p>	

GLOSARIO

AJAX: Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML): Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano.

Apache: Servidor HTTP de dominio público basado en el sistema operativo Linux. Se desarrolló en 1995 y es actualmente uno de los servidores HTTP más utilizados en la red.

Aplicación Web: Una aplicación web es un sistema informático que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero.

Base de Datos (BD): Es un conjunto de datos, o registros dentro del mismo contexto que son almacenados sistemáticamente (ordenados, clasificados) para su posterior consulta, actualización o mantenimiento a través de aplicaciones específicas.

Casos de Uso: Proporcionan uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico, utilizando un lenguaje más cercano al usuario final. También, se trata de una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema.

Eclipse: Es principalmente una plataforma de programación, usada para crear entornos integrados de desarrollo (del Inglés IDE).

Eclipse PHP Development Tools (PDT): Es un conjunto de herramientas que mejora la productividad de los desarrolladores de PHP.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

HTML (Hypertext Markup Language): Es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

JavaScript: Es el lenguaje scripting por excelencia, es decir, es un lenguaje basado en scripts (guión o conjunto de instrucciones). Posee una sintaxis similar a la del lenguaje Java y el lenguaje C, aunque no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia. Está destinado al desarrollo de aplicaciones web como complemento del HTML.

JSON: acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos, es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Lenguaje Unificado de Modelado (UML): Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software

PHP (acrónimo de "PHP: Hypertext Preprocessor"): Es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

PostgreSQL: Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99.

Proceso Unificado de Rational (RUP): Es un proceso de desarrollo de software. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados.

Propel: Es un proyecto de software libre y es una de las mejores capas de abstracción de objetos/relacional disponibles en PHP 5. Propel está completamente integrado en Symfony.

Servidor: Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la red.

Tiempo Real Suave: Es un sistema en el que los plazos de tiempo asignados a las tareas no son críticos. Esto significa que el no cumplimiento de una tarea en su tiempo establecido no conlleva a fallas catastróficas o pérdidas de información en el sistema.

Visual Paradigm: Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

XML: Sus siglas vienen de Extensible Markup Language, no es más que un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes.