



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**FACULTAD DE INGENIERÍA**

**Sistema inteligente asesor de la  
gestión energética en pequeñas y  
medianas empresas PyMES**

**TESIS**

Que para obtener el título de  
**Ingeniero en Computación**

**P R E S E N T A**

Karla Angélica García Cortés

**DIRECTOR DE TESIS**

Dr. Nicolás Ceferino Kemper Valverde



**Ciudad Universitaria, CDMX, 2019**



*A Carlos y Esperanza por el apoyo, sacrificio y amor*



## **AGRADECIMIENTOS**

A mis padres y hermanos, por acompañarme y apoyarme en todos mis proyectos y en el cumplimiento de mis metas. Agradezco a la Universidad Nacional Autónoma de México por brindarme la educación y un sinfín de experiencias y oportunidades. A Nicolás Ceferino Kemper Valverde, Luis Ochoa Toledo y Hugo Flores Huerta por el apoyo y el asesoramiento de mi tesis. Al equipo Cansat Siqueiros de la UNAM y al doctor Alejandro Farah Simón, por enseñarme el verdadero significado de la ingeniería y el trabajo en equipo. A los profesores y compañeros del Instituto de Ciencias Aplicadas y Tecnología y a la Facultad de Ingeniería por el apoyo y la confianza para realizar este proyecto en sus instalaciones.

# ÍNDICE DE CONTENIDO

Resumen .....	1
<i>Abstract</i> .....	2
Índice de ilustraciones y cuadros .....	3
<b>1. Introducción.....</b>	<b>5</b>
1.1.    Antecedentes.....	5
1.2.    Problema de investigación .....	7
1.3.    Objetivo.....	9
1.4.    Metodología.....	10
1.5.    Organización por capítulos .....	11
<b>2. Marco teórico .....</b>	<b>12</b>
2.1.    Concepto de ahorro y consumo energético .....	12
2.2.    Auditoría energética .....	12
2.3.    La industria en México .....	14
<b>3. Marco conceptual: Sistemas expertos.....</b>	<b>17</b>
3.1.    Concepto de sistema experto .....	17
3.2.    Ingeniería del conocimiento .....	19
3.3.    Representación del conocimiento.....	20
3.4.    Codificación de la información.....	22
<b>4. Sistema asesor de la gestión energética en Pequeñas y Medianas Empresas     (PyMES) .....</b>	<b>24</b>

4.1.	Adquisición del conocimiento .....	24
4.1.1.	Equipos industriales .....	25
4.2.	Identificación de módulos.....	27
4.3.	Arquitectura del sistema .....	30
4.4.	Diagrama general de casos de uso .....	36
4.5.	Actores y diagramas de secuencia .....	37
4.6.	Modelado y desarrollo de la base de datos .....	50
4.7.	Codificación de algoritmos por proceso industrial .....	55
4.8.	Modelado y desarrollo de árboles de decisión destinado a recomendaciones.....	71
4.9.	Desarrollo de interfaz web .....	81
4.10.	Pruebas de usuario .....	85
<b>5.</b>	<b>Conclusiones y trabajo futuro .....</b>	<b>88</b>
<b>6.</b>	<b>Glosario .....</b>	<b>91</b>
<b>A.</b>	<b>Apéndices .....</b>	<b>92</b>
A.1.	Formatos de levantamiento .....	93
A.1.1.	Caldera .....	93
A.1.2.	Motor eléctrico .....	94
A.1.3.	Bomba .....	95
A.1.4.	Iluminación.....	96
A.1.5.	Compresor .....	97
A.1.6.	Horno de flama .....	98
A.2.	Base de datos.....	99
A.3.	Algoritmos de análisis energético por equipos .....	100
A.3.1.	Caldera .....	100
A.3.2.	Motor eléctrico .....	104
A.3.3.	Bomba .....	107
A.3.4.	Iluminación.....	115

A.3.5.	Compresor .....	127
A.3.6.	Horno de flama .....	134
A.4.	Interfaz del sistema .....	141
A.4.1.	Vista Inicio.....	141
A.4.2.	Vista Nosotros.....	142
A.4.3.	Vista Mejores prácticas .....	143
A.4.4.	Vista Casos de éxito .....	144
A.4.5.	Vista Contacto .....	145
A.4.6.	Vista Registro.....	146
A.4.7.	Vista Inicio de sesión .....	147
A.4.8.	Vista Sistema con ingreso .....	147
A.4.9.	Vista Encuesta de la empresa .....	148
A.4.10.	Vista Formulario de caldera .....	149
A.4.11.	Vista Diagnóstico de caldera .....	149
A.4.12.	Vista Formulario de motor eléctrico.....	150
A.4.13.	Vista Diagnóstico de motor eléctrico.....	150
A.4.14.	Vista Formulario de sistema de bombeo .....	151
A.4.15.	Vista Diagnóstico de sistema de bombeo .....	151
A.4.16.	Vista Formulario de iluminación .....	152
A.4.17.	Vista Diagnóstico de iluminación .....	152
A.4.18.	Vista Formulario de compresor .....	153
A.4.19.	Vista Diagnóstico de compresor.....	153
A.4.20.	Vista Formulario de horno de flama .....	154
A.4.21.	Vista Diagnóstico de horno de flama.....	154

**7. Referencias.....155**



## **RESUMEN**

Cada vez toma más importancia el tema del ahorro energético por la necesidad de aprovechar eficientemente los recursos que aportan energía eléctrica. En México, poco más del 85% de los energéticos provienen de recursos naturales no renovables, esto obliga a buscar alternativas para el ahorro de los recursos.

El Sistema inteligente asesor de la gestión energética ayuda a la gestión energética en Pequeñas y Medianas Empresas (Pymes) las cuales consumen poco más del 30% de energía total suministrada en el país. El sistema está desarrollado para obtener un diagnóstico energético por cada equipo perteneciente en la Pyme, actualmente el sistema contempla el algoritmo de 6 equipos diferentes tales como caldera, motor eléctrico, bomba, iluminación, compresor y horno de flama. Estos equipos están relacionados con la rama industrial a la que pertenece la empresa.

Utilizando la técnica de ingeniería del conocimiento se obtuvo un algoritmo por cada equipo industrial, estos algoritmos se adquirieron de expertos en el tema de eficiencia y auditoría energética.

A cada diagnóstico se le relaciona una serie de recomendaciones para que el usuario mejore la eficiencia de su equipo, estas recomendaciones son inferidas de un sistema experto desarrollado a partir de reglas proporcionadas por los expertos.

Las etapas del funcionamiento del sistema están basadas en las fases de una auditoría energética en la industria.

## **ABSTRACT**

Currently, energy saving is more and more important, for this reason, we need to use efficiently the resources that contribute to energy. In Mexico, more than 85% of the energetics come from non-renewable natural resources This forces us to look for alternatives to save up resources.

The intelligent system helps the energy management in small and medium-sized enterprises, which consume more than 30% of the total energy supplied in the country. The system uses artificial intelligence techniques such as knowledge engineering and expert systems. The phases of its operation are based on energy audits in the industrial sector.

# ÍNDICE DE ILUSTRACIONES Y CUADROS

Figura 1	Categorización de empresas según su tamaño.....	14
Figura 2	Distribución de pequeñas y medianas empresas Pymes.....	15
Figura 3	Arquitectura de sistema experto .....	18
Figura 4	Representación de conocimiento por reglas .....	21
Figura 5	Representación de conocimiento por árboles de decisión .....	22
Figura 6	Tipos de razonamiento en máquina de inferencia .....	23
Figura 7	Principales fases de sistema asesor de gestión energética .....	25
Figura 8	Matriz de aparición ramas industriales – equipos .....	27
Figura 9	Identificación de módulos del sistema .....	29
Figura 10	Arquitectura Modelo-Vista-Controlador (MVC).....	30
Figura 11	Diagrama general de casos de uso.....	36
Figura 12	Diagrama de secuencia: Ver información general del sitio.....	39
Figura 13	Diagrama de secuencia: Registro.....	40
Figura 14	Diagrama de secuencia: Iniciar sesión.....	41
Figura 15	Diagrama de secuencia: Agregar información de la empresa ..	42
Figura 16	Diagrama de secuencia: Agregar información de equipos .....	46
Figura 17	Diagrama de secuencia: Consultar diagnósticos energéticos ..	49
Figura 18	Diagrama de bloques: Base de Datos.....	50
Figura 19	Clasificación de tareas en estructura MVC .....	56
Figura 20	Diagrama de flujo: Caldera.....	59
Figura 21	Diagrama de flujo: Motor eléctrico .....	62
Figura 22	Diagrama de flujo: Bomba.....	64
Figura 23	Diagrama de flujo: Iluminación .....	66
Figura 24	Diagrama de flujo: Compresor .....	68
Figura 25	Diagrama de flujo: Horno de flama .....	70

Figura 26	Diagrama de secuencia: Unión de módulos Diagnóstico - Recomendaciones.....	71
Figura 27	Árbol de decisión: Caldera.....	77
Figura 28	Árbol de decisión: Motor eléctrico.....	77
Figura 29	Árbol de decisión: Bomba.....	78
Figura 30	Árbol de decisión: Compresor.....	79

# INTRODUCCIÓN

## 1.1. Antecedentes del ahorro energético

- Gran crisis petrolera de 1973

En la década de los setenta no existía la cultura de la gestión energética, la mayoría de los sistemas utilizados eran manuales, es decir, un usuario responsable era el encargado de manejar los equipos utilizados en la industria. La gestión de dichos equipos no contemplaba algunos factores como el clima, producción, estado de los equipos, etc. Estos factores actualmente son muy importantes para la eficiencia energética.

En 1973 inició la gran crisis petrolera, en esta etapa hubo aumentos repentinos en los precios de combustibles y, por lo mismo, problemas en el suministro de energía.

La crisis petrolera dio pie a que emergieran nuevas tecnologías para poder sobrellevar este evento. Se tomaron medidas de ahorro en energía, la mayoría se centraban en la promoción del ahorro energético en los hogares, sin embargo, estas fueron deficientes.

- Inicio de la gestión energética

En la década de los ochenta y hasta mediados de la década de los noventa se desarrollaron por primera vez los modelos de gestión eficaz. El propósito de estos modelos era ayudar a la gestión energética en la industria.

Llegó la era de las computadoras portátiles y con ellas se abrió paso al monitoreo y focalización de los equipos utilizados en la industria. Los análisis hechos por computadora ya tomaban en cuenta factores importantes como la temperatura ambiente, los niveles de producción y estado de los equipos.

Durante este periodo apareció la “consultoría de la gestión energética” en donde varias empresas realizaban auditorías, ejecutaban proyectos dirigidos a la gestión y ofrecían programas de comunicación y promoción del ahorro energético.

En 1987 se utilizó por primera vez el término *Desarrollo Sostenible* en el informe socioeconómico Brundtland elaborado para la Organización de la Naciones Unidas (ONU). Desarrollo sostenible se definió como “Satisfacer las necesidades del presente sin comprometer las necesidades de las futuras generaciones” [10].

- El abrir de ojos, problemas en el medio ambiente

Al inicio del siglo XXI, la gestión de la energía como disciplina comenzó a decaer gracias a dos factores: La reducción de los precios y la disminución del tamaño general de las empresas. La causa fue porque los empresarios podían lograr mayores ahorros con menos riesgo si adquirirían equipos eficientes, en menor cantidad, en lugar de pagar por una gestión energética.

En la primera década del nuevo siglo empezó a existir la necesidad de reducir el empleo del carbón y cada vez se tomaron más en cuenta los términos como calentamiento global, huella de carbono, cambio climático, etc. En este periodo el daño al medio ambiente empezó a tomar relevancia y muchas empresas reincorporaron la gestión energética aplicando iniciativas ambientales generales.

- Protocolo Kioto

El protocolo Kioto fue adoptado el 11 de diciembre de 1997 en Kioto, Japón, pero no entró en vigor hasta el 16 de febrero de 2005 [11]. En total, 141 países se comprometieron a reducir en un 5.2% la emisión de gases de efecto invernadero global sobre los niveles que tenían en 1990 para el periodo 2008-2012 [12]. El protocolo Kioto cobró protagonismo convirtiendo al cambio climático en un foco principal y la energía se hizo un tema de alto nivel e importancia, a partir de este tratado muchas empresas se comprometieron a reducir el consumo de hidrocarburos.

- Eficiencia energética en los últimos años

La eficiencia energética comenzó a crecer a nivel mundial. En los últimos años ha aumentado el interés por la financiación de la gestión de la energía. El mercado necesita más desarrollo y cada vez hay más avances tecnológicos que ayudan a las medidas de ahorro energético y a la eficiencia energética [8].

## **1.2. Problema de investigación**

En los últimos años, ha cobrado importancia el tema del ahorro energético por la necesidad de aprovechar eficientemente los recursos que aportan energía

eléctrica y térmica, ya que algunos de estos recursos no son renovables y la manera de cómo se transportan, se extraen y se consumen afectan al medio ambiente.

En México, poco más del 85% de los energéticos provienen de recursos naturales no renovables, principalmente hidrocarburos y carbón. Esto obliga a buscar alternativas que permitan contribuir en la preservación de dichos recursos naturales ya sea en el ahorro de éstos o en su sustitución.

La cultura del ahorro y eficiencia energética en México ha tenido avances en los últimos 20 años, sin embargo, todavía no está a la altura de países primermundistas como Alemania, Italia, Japón, Francia, entre otros.

De acuerdo al informe de Indicadores de las Regulaciones para la Energía Sostenible RISE (*Regulatory Indicators for Sustainable Energy*) 2016, “... México tiene la máxima puntuación en acceso a energía, con 100 puntos. Sin embargo, en Eficiencia energética y Energía renovables sus marcadores son de 79 y 72, respectivamente. Esto revela que aún falta camino y la necesidad de establecer mandatos para cumplir con una producción, distribución y uso eficiente de energía ...”.

Tener un control en el consumo energético en la industria ayuda a la optimización de procesos, a la competitividad, al ahorro económico destinado a combustibles y electricidad, y también ayuda a la prevención oportuna de problemas relacionados con los equipos utilizados en la industria.

Las auditorías energéticas ayudan a las empresas a tener un control y un monitoreo sobre el consumo en energía y éstas son hechas por empresas certificadas. El costo de una auditoría va de los 250 000 a los 2 500 000 MXN<sup>1</sup>, por lo tanto, para que una empresa quiera tener un control en el consumo y eficiencia energética deberá tener contemplado el gasto dirigido a una



auditoria, esto puede ser una complicación sobre todo a las Pequeñas y Medianas empresas ya que sus ingresos no son tan considerables como los ingresos de las empresas grandes.

Existen empresas que desarrollan sistemas de ahorro y eficiencia energética dirigidos a equipos desarrollados por ellos mismos, lo cual obliga al consumidor a adquirir equipos de una sola marca o empresa para obtener los beneficios energéticos.

### **1.3. Objetivo**

El principal objetivo de este trabajo es modelar y desarrollar un sistema asesor de gestión energética dirigido a pequeñas y medianas empresas.

El sistema asesor de eficiencia energética tiene como propósito general dar un diagnóstico de tipo energético a partir de los datos adquiridos por el sistema, los datos son obtenidos de la placa de los equipos utilizados en los procesos industriales y de las mediciones hechas por un responsable a dichos equipos. La estructura del sistema abarca todas las partes que conlleva hacer una auditoría energética, tales como:

1. Recopilación de datos de empresa.
2. Recopilación de datos relacionados con la planta.
3. Registro de levantamiento de datos. (Datos de placa)
4. Toma de mediciones por un responsable.
5. Análisis de datos obtenidos.
6. Elaboración de diagnósticos energéticos.
7. Áreas de oportunidad y recomendaciones.
8. Mantenimiento de los sistemas.

El sistema inteligente proporcionará un diagnóstico por equipo del consumo y eficiencia energética, este deberá ser comparable a uno hecho en una auditoría industrial. Cada diagnóstico está relacionado con una serie de recomendaciones, las cuales serán brindadas al usuario para un uso y mantenimiento óptimo del equipo industrial. Se ha desarrollado un prototipo funcional el cual estará listo para futuras pruebas en la industria en ocho ramas industriales.

#### **1.4. Metodología**

Los sistemas expertos son considerados un subconjunto de la inteligencia artificial por este motivo hacen uso de métodos y técnicas específicas relacionadas con sistemas inteligentes. En una primera fase se utilizará ingeniería del conocimiento (*Knowledge engineering*), la cual es una disciplina moderna cuya principal aplicación es el diseño y desarrollo de sistemas expertos. El objetivo principal de la ingeniería del conocimiento es representar el conocimiento y razonamiento humano en un sistema artificial, en este caso se quiere recolectar toda la experiencia y experticia de las personas especialistas para poder representar y codificar dicho conocimiento de manera que pueda ser procesado por el sistema.

En una segunda etapa se hará la representación del conocimiento adquirido, para poder hacer esta representación se utilizará Representación por reglas RBR (*Rule Based Reasoning*) y árboles de decisión (modelo de clasificación ID3) los cuales son modelos de clasificación capaz de tomar decisiones con gran precisión.

Se utilizará la arquitectura Modelo-Vista-Controlador (MVC) para el modelado y desarrollo del sistema asesor de gestión energética. Esta arquitectura distribuye los datos del sistema, la interfaz con el usuario y la lógica de control en tres componentes distintos.

Otras metodologías se emplearán tales como el modelado y representación de bases de datos (BD) y la plataforma .NET para el desarrollo del sistema web.

## **1.5. Organización por capítulos**

En el capítulo 2 se encuentran las definiciones a distintos conceptos relacionados con la gestión energética, tales como ahorro energético, consumo energético y auditoría. También describe de manera general la industria en México y la distribución de empresas según su tamaño y sector.

En el capítulo 3 refiere a todo lo relacionado con el marco conceptual tal como la definición y explicación de sistema experto e ingeniería del conocimiento. Este capítulo describe los pasos necesarios en la realización de un sistema experto, como la adquisición y representación del conocimiento para finalmente hacer la codificación.

En el capítulo 4 muestra todo lo relacionado al modelado y al desarrollo del sistema inteligente asesor de la gestión energética. En el subcapítulo 4.1 se describe la adquisición del conocimiento brindado por el experto. En el subcapítulo 4.2 se muestra la identificación de módulos, estos módulos fueron necesarios para el desarrollo del sistema. El subcapítulo 4.3 refiere a la arquitectura MVC, la cual fue seleccionada para la codificación del sistema. El subcapítulo 4.4 muestra el diagrama general de casos de uso. El subcapítulo 4.5 explica cada caso de uso y su respectivo diagrama de secuencia. El subcapítulo 4.6 refiere al modelado y desarrollo de la base de datos. El subcapítulo 4.7 muestra la codificación de los equipos. En el subcapítulo 4.8 se describe el modelado y desarrollo del sistema experto asociado al módulo de recomendaciones en el sistema. El subcapítulo 4.9 explica el desarrollo de la interfaz del sistema y el subcapítulo 4.10 muestra las pruebas del sistema. El capítulo 5 describe las conclusiones en la realización del sistema y el trabajo futuro.

## **2. MARCO TEÓRICO**

### **2.1. Concepto de ahorro y eficiencia energética**

El objetivo principal de la eficiencia energética en la industria es potencializar los procesos y hacer buen uso de los equipos para ahorrar energía sin dejar a un lado la calidad de producción.

La eficiencia energética mejora la eficiencia en actividades como la producción, transformación, fabricación, entre otros, promueve el uso de equipos donde el valor de la eficiencia energética sea alto. También ayuda con la disminución del consumo energético y por lo tanto también a los costos dirigidos a energía.

El objetivo del ahorro energético es optimizar el consumo energético con el fin de disminuir el uso de energía sin alterar el objetivo final, es decir, el ahorro energético ayuda a obtener los mismos beneficios consumiendo menos energía.

### **2.2. Auditoría Energética**

La Auditoría energética es una herramienta para el proceso de la gestión energética, esta evalúa el desempeño de los equipos y sistemas consumidores de energía en la industria con ayuda del análisis de parámetros de operación.

Es una actividad multidisciplinaria con la finalidad de mejorar el desempeño de equipos y sistemas. Gracias a las auditorías energéticas es posible tomar mejores decisiones para incrementar la eficiencia en los procesos.

Una auditoría energética tiene una metodología bien estructurada donde se recopilan los datos, se analizan y se da un diagnóstico general y un diagnóstico por cada equipo. A continuación, se encuentran los pasos descritos de forma breve para la ejecución de una auditoría.

- Programación de los recursos y el tiempo.  
Durante esta etapa se toman los datos generales de la empresa, esto con el fin de concentrar la información requerida de la planta.
- Recopilación de datos y recorrido de la planta  
En este paso se recopila los datos relacionados con la energía de la planta tal como, historial de producción y consumo energético. A continuación, se muestran las partes que lo conforman.
  - Historial de consumo y producción  
Reúne los datos relacionados con el consumo de energía y producción como combustible utilizado, capacidad, poder calorífico, factura eléctrica, factor de potencia, etc.
  - Inspección visual  
En esta etapa se hace una revisión visual y se recopilan datos acerca de los principales equipos consumidores de energía.
  - Registro de levantamiento de datos.  
Se registran los datos de placa de cada uno de los equipos.
- Toma de Mediciones.  
En esta etapa se hace la medición de parámetros de operación de los equipos.

- **Análisis de datos**

En este paso se analizan todos los datos que se recopilaron, después se elabora los balances de energía y las medidas de ahorro, este paso es crucial para el mejoramiento de la eficiencia energética en la planta.

- **Elaboración de cartera de proyectos**

Una vez que se han identificado los problemas y las áreas de oportunidad, se elabora una cartera de proyectos donde se presenta la situación actual, el cálculo de ahorros, la evaluación económica, las medidas propuestas para el mejoramiento de la planta, entre otros [2].

### **2.3. La industria en México**

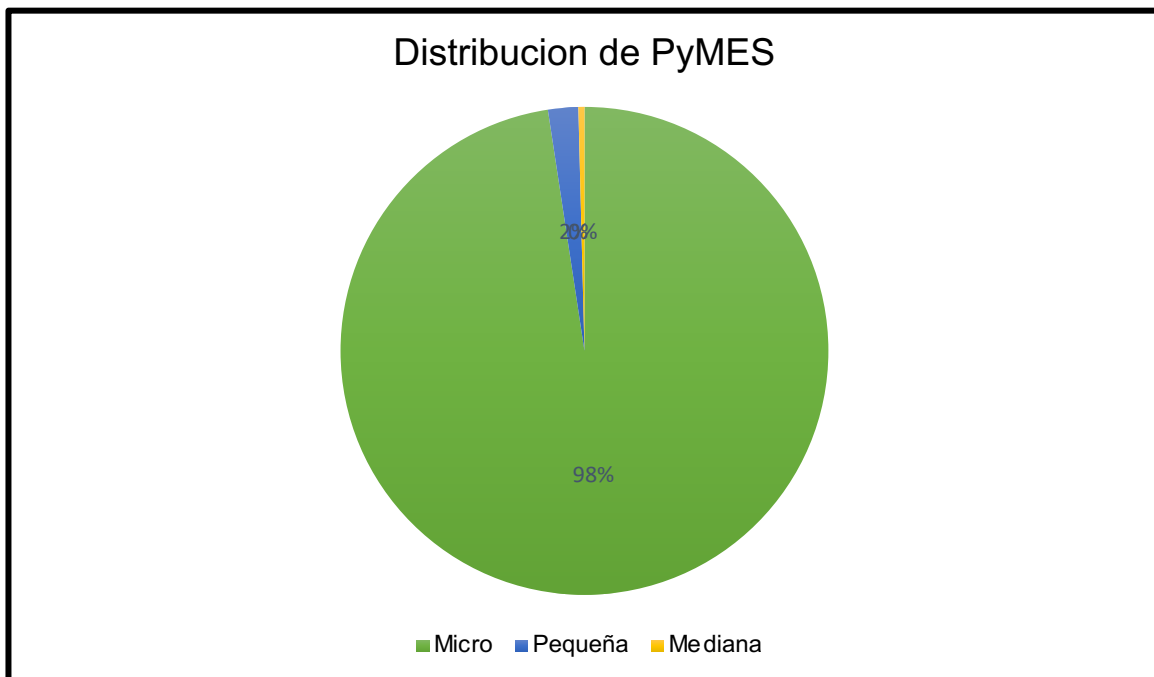
El sector industrial en México es pieza fundamental en el desarrollo económico, actualmente existen 4.2 millones de unidades económicas. De ese universo el 99.8% son consideradas Pequeñas y Medianas Empresas (PyMES).



*FIGURA 1. Categorización de empresas según su tamaño*

De acuerdo a las cifras reveladas por el Instituto Nacional de Estadística y Geografía al tercer trimestre del 2017, la industria manufacturera en su conjunto representó 16% del Producto Interno Bruto (PIB) [13].

Las Pequeñas y Medianas Empresas (PyMES) aportan 42% del Producto Interno Bruto (PIB) y generan el 78% del empleo en el país [14]. Existe una categoría propia para las Pequeñas y medianas empresas según su tamaño, en la siguiente gráfica se muestra la gráfica de su distribución. La imagen revela que existe en su mayoría empresas clasificadas como “Micro empresa” 98% y restante 2% se distribuye entre “Pequeñas empresas” 1.6% y “Medianas empresas” con un 0.4%.



*FIGURA 2. Distribución de pequeñas y medianas empresas PyMES*

A continuación, se muestra los principales sectores de los que están conformados las PyMES.

- Alimentos, bebidas, tabaco y confitería.
- Electrodomésticos.

- Electrónicos de comunicación.
- Equipo médico e instrumentos de medición.
- Equipo y maquinaria para la industria.
- Industria textil.
- Plásticos y hules.
- Productos de Madera.
- Productos para la construcción.
- Servicios de apoyo a los negocios.
- Servicios turísticos.
- Productos farmacéuticos.
- Industria química. [1]

De acuerdo con las cifras dadas por la Dirección General de Planeación e Información energéticas en 2016, la industria consume poco más del 30% de la energía total suministrada [9]. Esto revela la importancia de crear herramientas que aporten beneficios energéticos a las Pequeñas y Medianas Empresas las cuales sin duda forman gran parte del desarrollo económico y social del país.



# **3. MARCO CONCEPTUAL: SISTEMAS EXPERTOS**

## **3.1. Concepto de sistema experto**

En inteligencia artificial, un sistema experto es un sistema computacional que emula la capacidad de tomar decisiones de un humano experto. Los sistemas expertos hacen inferencias semejantes a las humanas y proporcionan respuestas sobre un área específica. La inferencia se hace basada en conocimientos obtenidos de distintas fuentes como libros, internet o una persona experta en el tema.

Para que un sistema actúe como un experto es importante que el sistema reúna ciertas características.

- Habilidad para reunir conocimiento.
- Fiabilidad en los datos obtenidos.
- Solidez en el dominio del conocimiento.
- Capacidad de resolver problemas [15].

En la siguiente figura se muestra la estructura básica de un sistema experto.

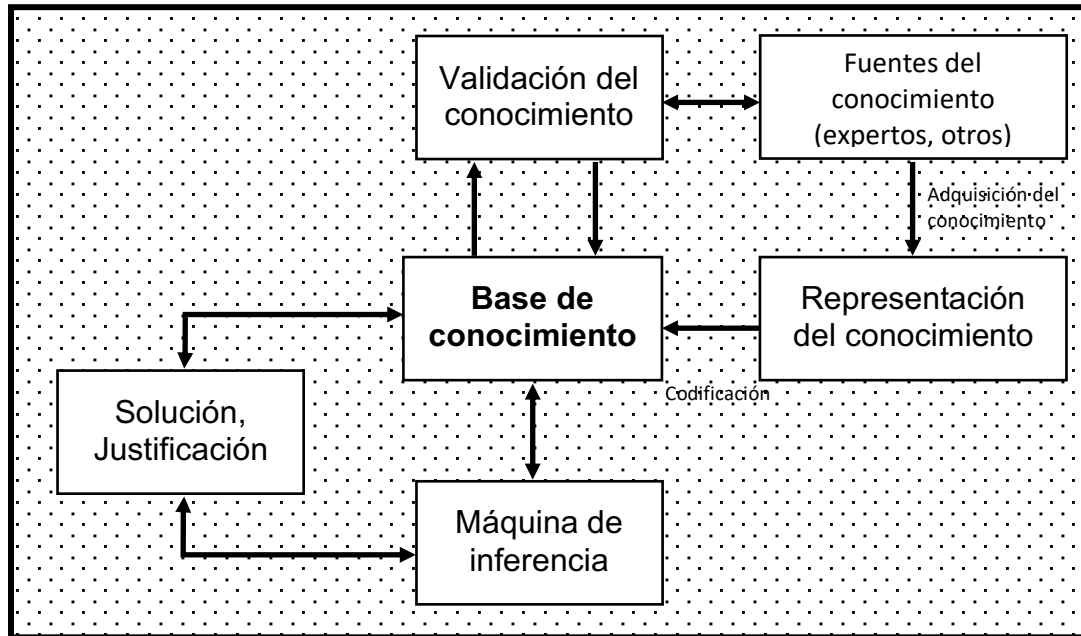


FIGURA 3. Arquitectura de Sistema Experto

Existen tres tipos de sistemas expertos:

- Basado en reglas preestablecidas RBR (Rule Based Reasoning)  
Este tipo de sistemas expertos utilizan reglas heurísticas basadas en lógica difusa para su evaluación y aplicación.
- Basado en casos CBR (Case Based Reasoning)  
Este tipo de sistema experto aplica el razonamiento basado en casos. Para su desarrollo se utilizan las soluciones a problemas presentados anteriormente.
- Basados en redes bayesianas  
Este tipo aplica redes bayesianas, las cuales, se basa en estadísticas como el teorema de Bayes.

## 3.2. Ingeniería del conocimiento

A el proceso de adquirir conocimiento de expertos y construir una base de conocimientos es llamado ingeniería del conocimiento (*Knowledge engineering*).

La ingeniería del conocimiento envuelve la cooperación de humanos expertos en un tema con ingenieros del conocimiento (*Knowledge engineer*) con el objetivo de codificar y hacer explícitas reglas o procedimientos que humanos expertos usan para resolver problemas reales.

A continuación, se muestra el proceso de ingeniería del conocimiento.

### 1. Adquisición del conocimiento.

La adquisición del conocimiento se encarga de obtener el conocimiento de humanos expertos, libros, documentos, sensores o archivos de computadora. El conocimiento puede ser específico o general a un tema, también puede ser el procedimiento para resolver un problema.

### 2. Representación del conocimiento.

El conocimiento adquirido es organizado y manejado para que se pueda utilizar, a esta actividad se le llama representación del conocimiento (*Knowledge representation*).

Esta actividad se encarga de la preparación de los datos adquiridos después dichos datos serán descritos en un mapa el cual estará almacenado en una base de conocimiento.

### 3. Validación del conocimiento.

La parte de validación del conocimiento hace, como su nombre lo indica, la validación y verificación del conocimiento obtenido, esto gracias a

algunas pruebas hechas para verificar que la calidad del conocimiento obtenido anteriormente es aceptable.

#### 4. Inferencia.

Esta actividad implica el desarrollo del software para que la computadora pueda hacer inferencias basadas en el conocimiento almacenado. El sistema debe ofrecer respuestas a problemas generales y específicos al tema.

#### 5. Explicación y justificación.

Este paso se encarga de poder explicar cómo es que el sistema funciona y cuáles son los pasos que hace la computadora para poder obtener una resolución [16].

### **3.3. Representación del conocimiento**

Una vez adquirido y validado el conocimiento, el ingeniero del conocimiento debe representar los datos en un formato entendible para los humanos y ejecutable para la computadora. Existen métodos diferentes para la representación del conocimiento, los más conocidos son reglas, árboles de decisión y objetos.

Para fines de la investigación nos centraremos en la producción de reglas y en los árboles de decisión.

- Producción de reglas

La producción de reglas es la forma más popular de representación del conocimiento para sistemas expertos. El conocimiento es representado en forma de condición y acción. A continuación, se muestra cómo es que trabaja la producción de reglas.

**SÍ/IF** esta condición ocurre  
**ENTONCES/THEN** esta acción o resultado puede ocurrir.

*FIGURA 4. Representación de conocimiento por reglas*

Esta forma de representación es adecuada para cuando se tiene un conjunto de acciones independientes a otras. Las reglas pueden verse como el modelo del comportamiento de los humanos expertos.

#### Ventajas y limitaciones de las reglas

La representación de reglas es especialmente aplicable cuando se necesita recomendar una acción basada en eventos observables.

Aquí se muestran sus principales ventajas.

- Las reglas son fáciles de entender, esto se debe a que son una forma natural de conocimiento.
- Las inferencias y explicaciones pueden ser fácilmente derivables.
- Las modificaciones y el mantenimiento del sistema experto son relativamente fácil.
- Se pueden combinar con las reglas casos en donde se tenga incertidumbre.
- Cada regla es comúnmente independiente a las otras.

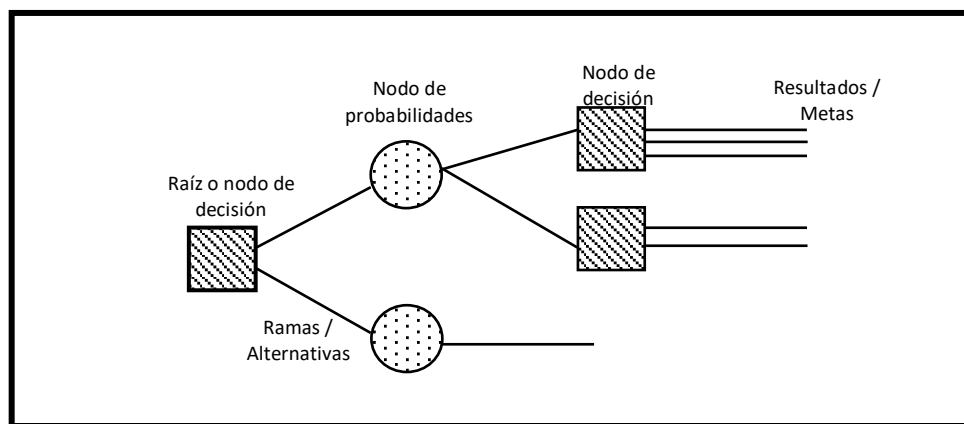
#### Desventajas

- Es difícil seguir jerarquías.
- Es ineficiente para sistemas largos.
- No todo el conocimiento puede ser representado en reglas.
- Tiene una pobre representación de la estructura del conocimiento.

- Árboles de decisión

Los árboles de decisión es una de las maneras más populares de representar a los sistemas expertos. Están compuestos por nodos y por metas, los cuales son unidos por líneas que representan las decisiones. Las principales ventajas de los árboles de decisión es que simplifican el proceso de adquisición del conocimiento y fácilmente el diagrama puede convertirse en reglas. Una de las limitaciones es que entre más extenso sea el árbol de decisión más tiempo le tomará a la computadora obtener una resolución.

A continuación, se muestra cómo es que se compone un árbol de decisión.



*FIGURA 5. Representación de conocimiento por árboles de decisión*

### **3.4. Codificación de la información**

Cuando la representación del conocimiento en la base del conocimiento esta completa o al menos tiene el suficiente nivel de precisión, entonces esta lista para ser usada.

Para que el sistema experto pueda hacer una inferencia necesita un programa de computadora que permita el acceso al conocimiento almacenado. Este programa es un algoritmo que controla el proceso de razonamiento, también es llamado máquina de inferencia o programa de control. El programa de

control decide cual regla investigar, cual alternativa eliminar y cual atributo tomar.

La máquina de inferencia utiliza más de un método de razonamiento para tomar una alternativa. A continuación, se muestran los distintos métodos.

<b>MÉTODOS DE RAZONAMIENTO</b>	
<b>Método</b>	<b>Descripción</b>
<i>Razonamiento deductivo</i>	Toma un principio general y hace una inferencia de tipo específica.
<i>Razonamiento inductivo</i>	Toma hechos establecidos y hace a partir de ellos conclusiones generales.
<i>Razonamiento analógico</i>	El razonamiento analógico hace un proceso de aprendizaje interno el cual, se basa en hechos pasados (experiencia).
<i>Razonamiento formal</i>	Manipula sintácticamente una estructura de datos para deducir nuevos hechos siguiendo reglas o inferencias preestablecidas.
<i>Razonamiento procedural</i>	Usa modelos matemáticos o simulaciones.
<i>Razonamiento "metalevel"</i>	Tiene conocimiento acerca de lo que se conoce.

FIGURA 6. Tipos de razonamiento en máquina de inferencia

## **4. SISTEMA ASESOR DE LA GESTIÓN ENERGÉTICA EN PEQUEÑAS Y MEDIANAS EMPRESAS (PYMES)**

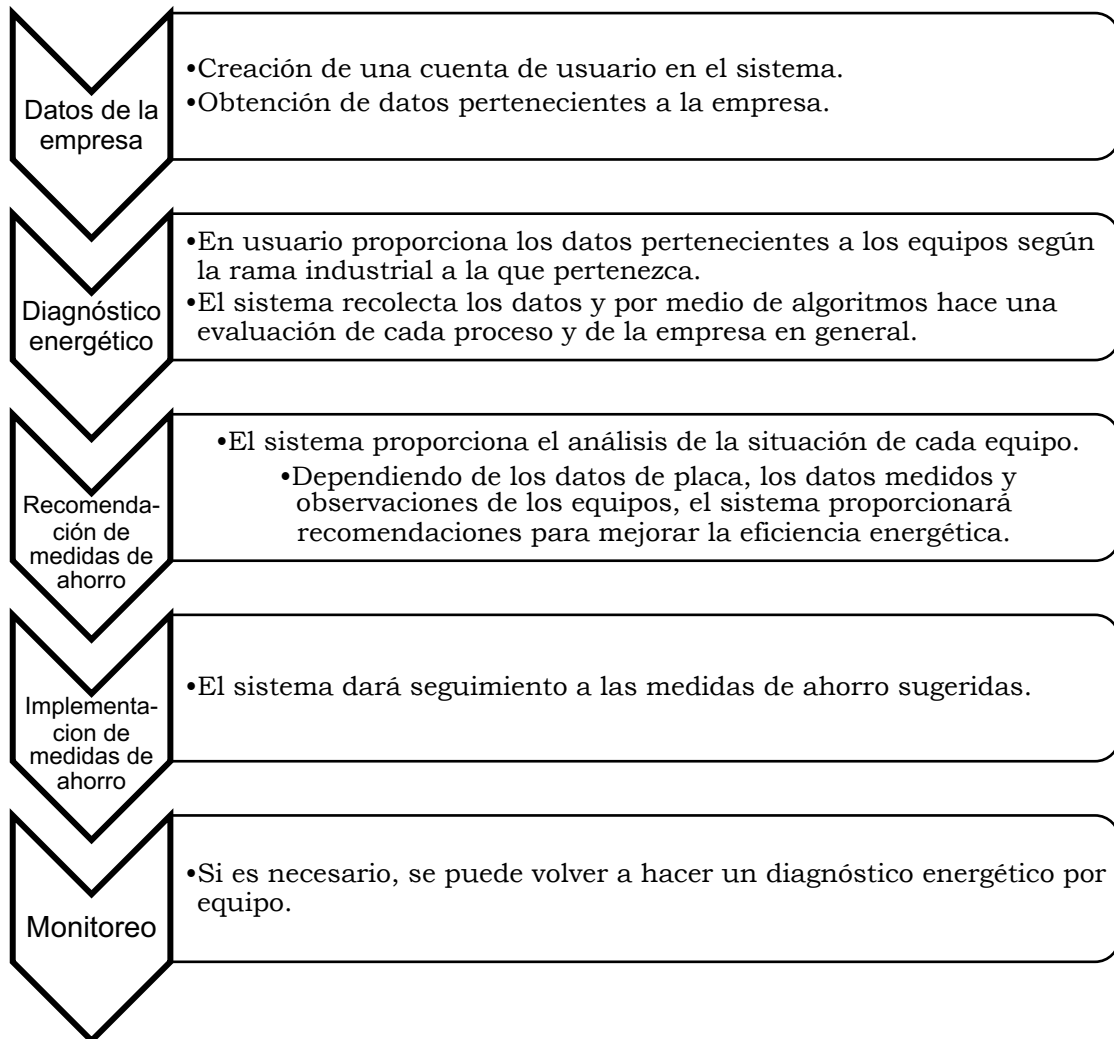
### **4.1. Adquisición del conocimiento**

La obtención de todos los datos, algoritmos de análisis y evaluación de medidas de ahorro fueron proporcionados por expertos en el área de ahorro de energía en las diferentes ramas industriales seleccionadas.

Uno de los objetivos es que el usuario de la plataforma pueda utilizar la herramienta sin tener un conocimiento profundo de gestión en eficiencia energética ni de auditorías, por esta razón el sistema proporciona una serie de instrucciones para que el usuario pueda hacer las mediciones y el llenado de los datos correctamente.

El sistema trabajará en 5 etapas para diagnosticar la eficiencia energética de cada proceso. A continuación, se muestran las fases.





*FIGURA 7. Principales fases de Sistema Asesor de Gestión energética*

#### **4.1.1. Equipos industriales**

Las ramas industriales con las cuales trabajará el sistema fueron seleccionadas por los expertos. En cada sector industrial se ha analizado los procesos, los cálculos de análisis, recomendaciones y mantenimiento de cada equipo. A continuación, se muestra la lista de dichas ramas industriales.

- Alimentos procesados
- Plásticos
- Metal-Mecánica

- Metal-Mecánica (Herrería y estructura)
- Impresión e industrias conexas
- Lácteos
- Otros productos químicos
- Fabricación de pinturas y recubrimientos.

Cada rama industrial tiene asociado un conjunto de equipos, es decir, según la naturaleza de la rama industrial hay procesos específicos que se realizan en ella. Los expertos nos han brindado los algoritmos de los siguientes equipos, los cuales ayudarán a dar un diagnóstico energético por proceso industrial.

- Caldera
- Motor eléctrico
- Sistema de bombeo
- Iluminación
- Compresor
- Horno de flama

En la etapa de adquisición de conocimiento se obtuvo la siguiente matriz de aparición de Ramas-Equipos (marcado con 1 si es que pertenece en la rama industrial y 0 si no pertenece).

<b>Equipos- Ramas</b>	<i>Alimentos procesados</i>	<i>Plásticos</i>	<i>Metal-Mecánica</i>	<i>Metal-Mecánica (Herrería y estructura)</i>	<i>Impresión e industrias conexas</i>	<i>Lácteos</i>	<i>Otros productos químicos</i>	<i>Fabricación de pinturas y recubrimientos</i>
<i>Bombas</i>	1	1	1	1	1	1	1	1
<i>Iluminación</i>	1	1	1	1	1	1	1	1
<i>Motores Eléctricos</i>	1	1	1	1	1	1	1	1
<i>Compresores</i>	1	1	1	1	1	1	1	1
<i>Calderas</i>	1	1	0	0	0	1	1	0
<i>Hornos de flama</i>	0	0	0	0	0	0	0	1

FIGURA 8. Matriz de aparición Ramas Industriales - Equipos

En el apéndice A.1 se encontrarán los formatos de cada equipo, los cuales son utilizados para el desarrollo de auditorías energéticas. Cada formato muestra los datos necesarios para hacer el análisis energético (datos de placa o medidos).

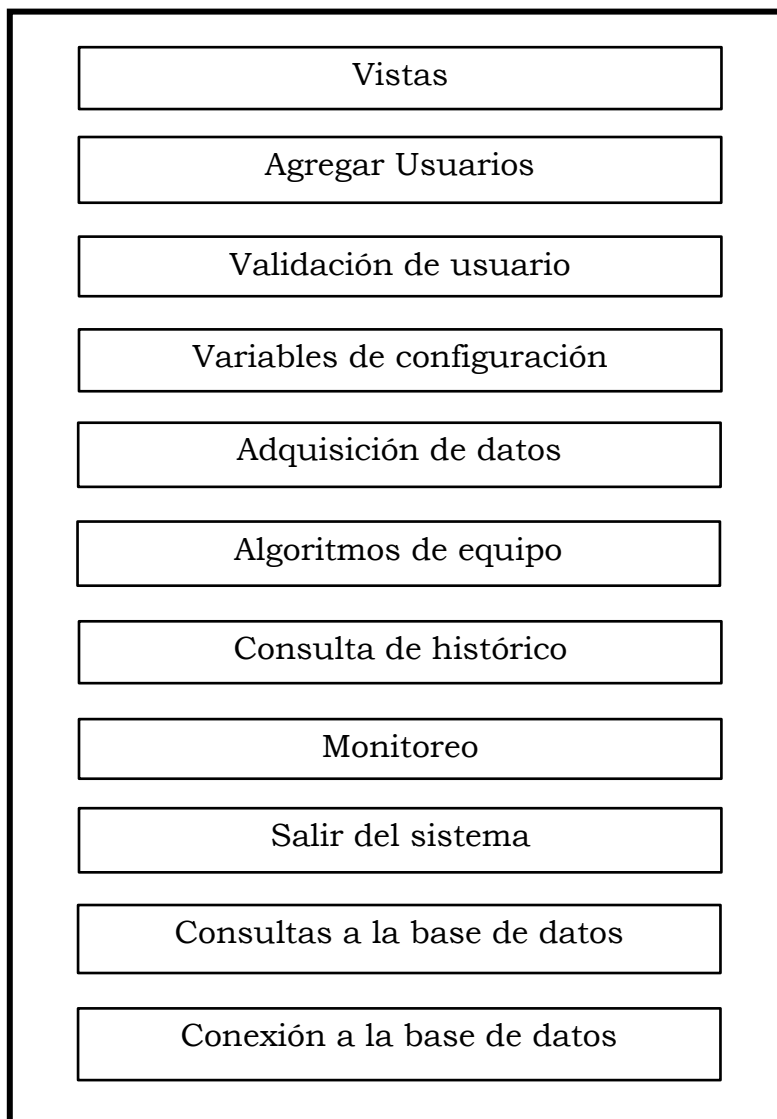
## 4.2. Identificación de Módulos

- Ingreso-registro al sistema

El sistema deberá contar con una conexión a base de datos para poder almacenar los nombres de usuario y contraseña. Se desplegará el formulario en el sistema web de una manera fácil para el usuario.

- Obtención de datos de la empresa  
Una vez ingresado al sistema el usuario deberá llenar la encuesta de empresa, para el almacenamiento de este formulario se tendrá una conexión a la base de datos y su respectiva vista en el sistema web.
- Formulario de equipo perteneciente a la empresa  
Cada formulario será desplegado en una vista diferente para facilitar el llenado de datos al usuario, los datos obtenidos (placa y medidos) se guardan en la base de datos y a su vez se mandan a la sección de algoritmos para su análisis.
- Algoritmos de análisis de cada equipo  
Los algoritmos de análisis de datos de cada equipo no son desplegados en vista para el usuario, después del análisis los datos calculados se almacenan en la base de datos.
- Despliegue de diagnóstico de equipo y recomendaciones  
Después de hacer el análisis de los datos calculados se desplegará en una vista el resultado relacionado con la eficiencia energética en el equipo analizado y las recomendaciones para la mejora de la eficiencia energética y del uso del equipo.
- Salir del sistema  
El sistema contendrá diferente información dependiendo si el usuario ha ingresado al sistema o no. Esta información podrá ser editada por el administrador del sistema.

En la siguiente figura se muestran los módulos identificados. Los módulos identificados ayudan a tener una primera vista de los componentes necesarios para el desarrollo del sistema.



*FIGURA 9. Identificación de módulos del sistema*

### 4.3. Arquitectura del sistema

Se empleará la arquitectura Modelo-Vista-Controlador (MVC) la cual distribuye los datos del sistema, la interfaz con el usuario y la lógica de control en tres componentes distintos [17].

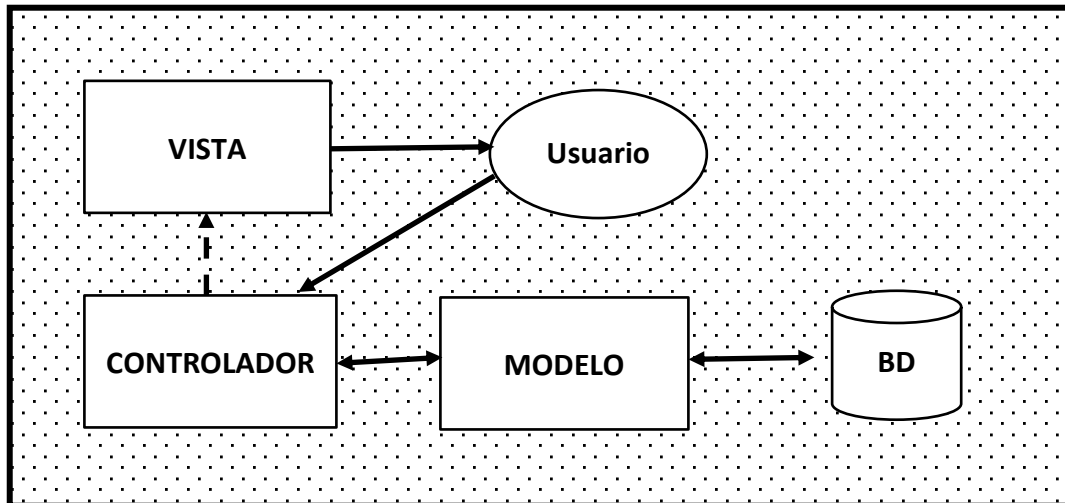


FIGURA 10. Arquitectura Modelo-Vista-Controlador (MVC)

- **Modelo**

El modelo contiene una representación de los datos que maneja el sistema por lo cual es el responsable de acceder a la base de datos. Este componente lleva un registro de las vistas y controladores del sistema y también define las reglas de funcionalidad del sistema.

- **Vista**

La vista recibe los datos del modelo y lo muestra al usuario. Su principal función es contener la información que se envía al cliente así como su interacción con el usuario.

- **Controlador**

La función del controlador es servir como intermediario entre la capa de Modelo y la capa de Vista, gestiona la información entre ellos y adapta

los datos según las necesidades de cada capa. En controlador recibe los eventos de entrada y contiene las reglas y los algoritmos para la gestión de eventos.

### **Arquitectura aplicada al sistema de gestión energética en PYMES**

- Modelos del sistema

A continuación, se muestra una lista con los componentes de la capa de Modelo y una breve explicación de sus funciones. Los componentes están divididos según a la sección que pertenezcan.

- Registro e ingreso al sistema

<b>Modelo</b>	<b>Descripción</b>
<i>Usuario</i>	Es la representación de los datos del usuario.
<i>SistemUser</i>	Es la representación de las llaves del usuario para ingresar al sistema.
<i>Empresa</i>	Es la representación de la información de la empresa
<i>Estado</i>	Es la representación de la información de los estados de la República Mexicana.
<i>RamoEmpresa</i>	Es la representación de la información sobre los ramos industriales.

- Caldera

<b>Modelo</b>	<b>Descripción</b>
<i>CalderaPlaca</i>	Es la representación de los datos de placa de la caldera.

<i>Combustible</i>	Es la representación de los datos del combustible utilizado en la caldera.
<i>MedicionCaldera</i>	Es la representación de los datos de medición en la caldera.

○ Motores eléctricos

<b>Modelo</b>	<b>Descripción</b>
<i>MotoresDatosIn</i>	Es la representación de los datos de placa del motor eléctrico.
<i>Tipos</i>	Es la representación de los datos de los tipos de motor eléctrico
<i>MedicionMotores</i>	Es la representación de los datos de medición del motor eléctrico

○ Sistema de bombeo

<b>Modelo</b>	<b>Descripción</b>
<i>BombasDatosIn</i>	Es la representación de los datos del sistema de bombeo
<i>Aplicaciones</i>	Es la representación de los datos de los tipos de bombas.
<i>Catalogo</i>	Es la representación de los datos de la clasificación de bombas.

○ Iluminación

<b>Modelo</b>	<b>Descripción</b>
<i>Iluminacion</i>	Es la representación de los datos de las áreas con iluminación.
<i>Luminaria</i>	Es la representación de los datos de las luminarias y lámparas.



<i>TipoArea</i>	Es la representación de los datos de clasificación de áreas.
<i>TipoIlumina</i>	Es la representación de los datos de clasificación de luminarias.
<i>TipoLampara</i>	Es la representación de los datos de clasificación de lámparas.
<i>Mantenimiento</i>	Es la representación de los datos de mantenimiento de las luminarias.

- Compresor

<b>Modelo</b>	<b>Descripción</b>
<i>Compresor</i>	Es la representación de los datos del compresor.
<i>periodoCompresor</i>	Es la representación de los datos del periodo de mantenimiento.
<i>ubicaCompresor</i>	Es la representación de los datos de la ubicación del compresor

- Horno de Flama

<b>Modelo</b>	<b>Descripción</b>
<i>HornosDatosIn</i>	Es la representación de los datos de los Hornos.

- Vistas del sistema

A continuación, se muestran las vistas del sistema con su respectivo identificador, cada vista está asociada a una sección dependiendo de su función.

- Home (Primera sección del sistema)

- Inicio (Inicio)

- Aviso (Aviso)
- Casos de Éxito (Casos)
- Nosotros (Nosotros)
- Mejores Prácticas (Mejores)
- Contacto (Contacto)
  
- Registro
  - Login (Login)
  - Registrado (Registrado)
  - Usuario (Usuario)
  
- Sitio
  - Inicio (Inicio)
  - Empresa (Empresa)
  
- Caldera
  - Caldera (Caldera)
  - Caldera Combustible (CalderaComb)
  - Caldera Diagnóstico (CalderaDiag)
  - Caldera Mediciones (CalderaMed)
  
- Motor eléctrico
  - Motores (Motores)
  - Motores Diagnóstico (MotoresDiag)
  - Motores Mediciones (MotoresMed)
  
- Sistema de bombeo
  - Bombas (Bombas)
  - Bombas Diagnóstico (BombasDiag)
  
- Iluminación

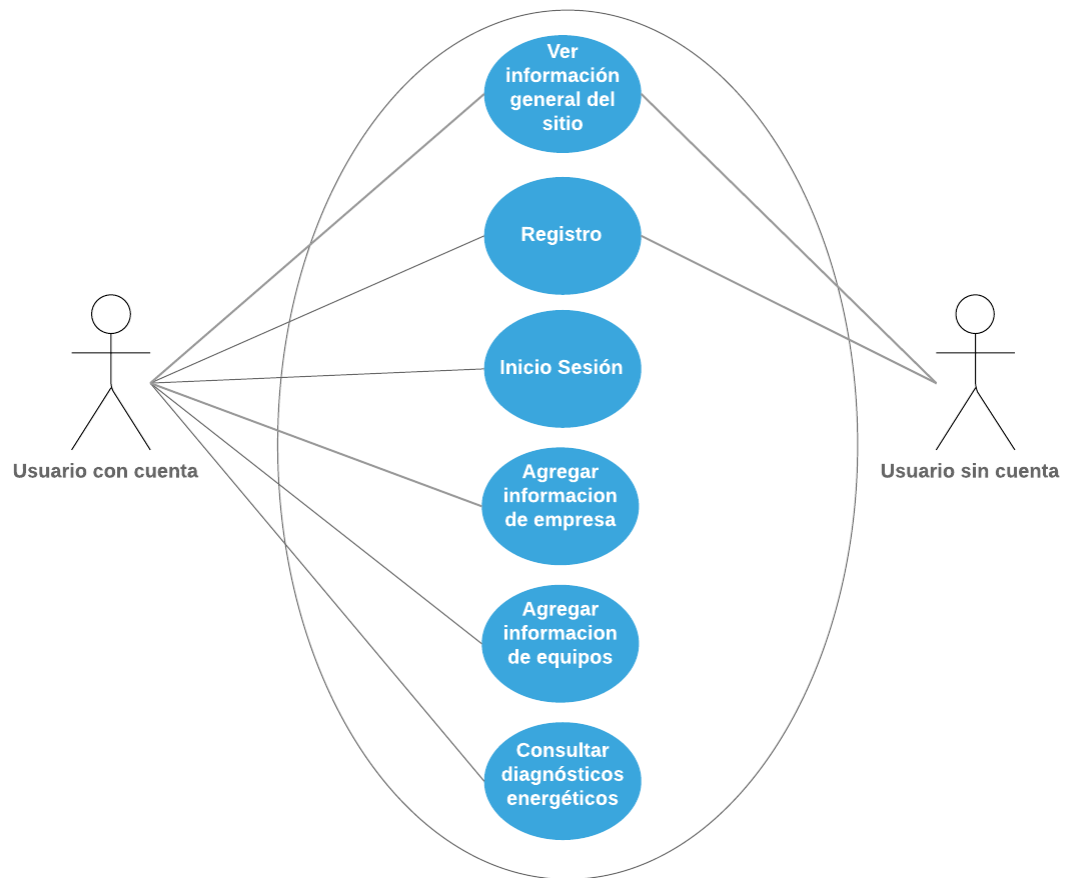
- Iluminación (Iluminacion)
  - Iluminación Análisis (iluAnalisis)
  - Luminaria (Luminaria)
  - Iluminación Resultados (resultadosIlu)
  - Iluminación Resumen (resumenIlu)
- Compresor
  - Compresor (Compresor)
  - Compresor Diagnóstico (CompresorDiag)
- Horno de flama
  - Hornos (HornoMain)
  - Hornos Diagnóstico (HornoDiag)
- Controladores del sistema

A continuación, se muestra una lista de los controladores utilizados y su identificador para el funcionamiento del sistema. Cada controlador hace de intermediario entre el modelo y sus vistas según la sección a la que pertenezcan.

- Controlador de Home (HomeController)
- Controlador de Registro (RegistroController)
- Controlador del Sitio (SitioController)
- Controlador de Caldera (CalderaController)
- Controlador del Motor eléctrico (MotoresElController)
- Controlador del Sistema de bombeo (BombasController)
- Controlador de Iluminación (IluminacionController)
- Controlador del Compresor (CompresorController)
- Controlador de Horno de Flama(HornosController)

#### 4.4. Diagrama general de Casos de Uso

Los diagramas de caso de uso son importantes para modelar el comportamiento de un sistema, especifican la interacción que toma lugar entre el actor y el sistema. El caso de uso mostrado a continuación expone una vista general de las actividades hechas por el usuario.



*FIGURA 11. Diagrama general de casos de uso*

## 4.5. Actores y diagrama de secuencia

### Actores

Un actor modela un tipo de rol, el cual interactúa con el sistema. Los actores pueden representar roles jugados por usuarios humanos, hardware externo y otros roles.

Se han especificado dos roles distintos, cada uno está descrito en las siguientes tablas.

<b>Actor</b>	<b>Usuario con cuenta</b>
<i>Casos de uso</i>	Registro, Inicio Sesión, Agregar información empresa, Consultar históricos, Agregar información de equipos, Consultar diagnósticos energéticos.
<i>Tipo</i>	Primario
<i>Descripción</i>	El usuario representa cualquier persona que utiliza el sistema para obtener diagnósticos energéticos de los equipos relacionados con la rama industrial a la que pertenece la empresa. Puede ingresar los datos de la empresa, los datos de los equipos, consultar los diagnósticos, los históricos y las recomendaciones.

<b>Actor</b>	<b>Usuario sin cuenta</b>
<i>Casos de uso</i>	Registro, Ver información general del sistema
<i>Tipo</i>	Secundario
<i>Descripción</i>	El Usuario sin cuenta representa a cualquier persona que ingrese al sistema y desee ver la

información general del sistema asesor de la gestión energética.
--

### Diagramas de Secuencia

El diagrama de secuencia describe la dinámica del sistema, representa la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores.

A continuación, se listan los casos de uso junto con sus respectivos diagramas de secuencia.

- Ver información general del sitio.

<b>Caso de uso Ver información general del sitio</b>	
<i>Actores</i>	Usuario con cuenta, Usuario sin cuenta
<i>Tipo</i>	Básico
<i>Propósito</i>	Es el primer acercamiento del sistema con los actores, brinda información sobre el sistema, sus desarrolladores y sus contactos.
<i>Resumen</i>	Inicia como punto de arranque del sistema, el usuario puede navegar a través de las vistas para obtener información general del sistema.
<i>Precondicion es</i>	
<i>Flujo principal</i>	El sistema presenta al usuario la vista de “Inicio” (Apéndice A.4.1). El usuario podrá navegar por las vistas “Nosotros”, “Mejores prácticas”, “Casos de éxito”, “Contacto”, “Inicio de sesión” y “registro”
<i>Flujo secundario</i>	Si se elige “Inicio de sesión” se va al caso de uso Iniciar Sesión.

Si se elige "Registro" se va al caso de uso Registro.

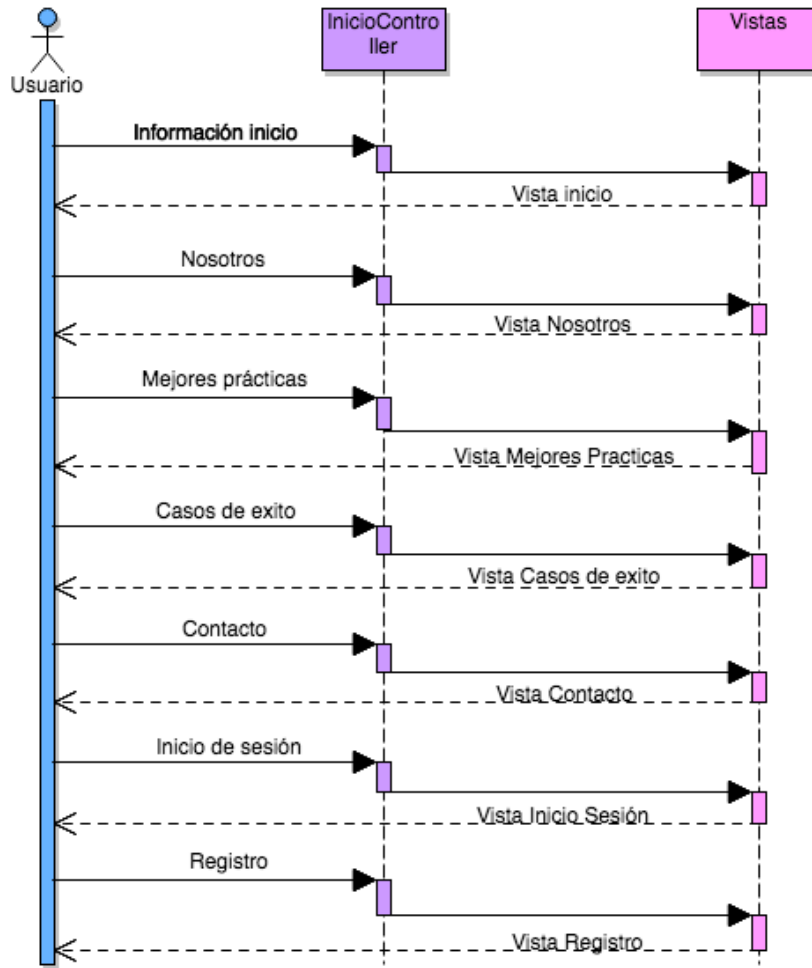


FIGURA 12. Diagrama de secuencia: Ver información general del sitio

- Registro

<b>Caso de uso Registro</b>	
Actores	Usuario con cuenta, Usuario sin cuenta
Tipo	Básico
Propósito	Es el primer paso para ingresar a las actividades del sistema, crea una cuenta al usuario.
Resumen	Inicia como punto de arranque del sistema, el usuario puede crear una cuenta ingresando sus

	datos personales, un correo electrónico y una contraseña.
<i>Precondicion es</i>	El usuario no deberá estar registrado anteriormente.
<i>Flujo principal</i>	El sistema presenta al usuario la vista de “Registro” (Apéndice A.4.6). El usuario ingresa su nombre, nombre de usuario, apellidos, correo electrónico, contraseña, verificación de contraseña y puesto en la empresa.

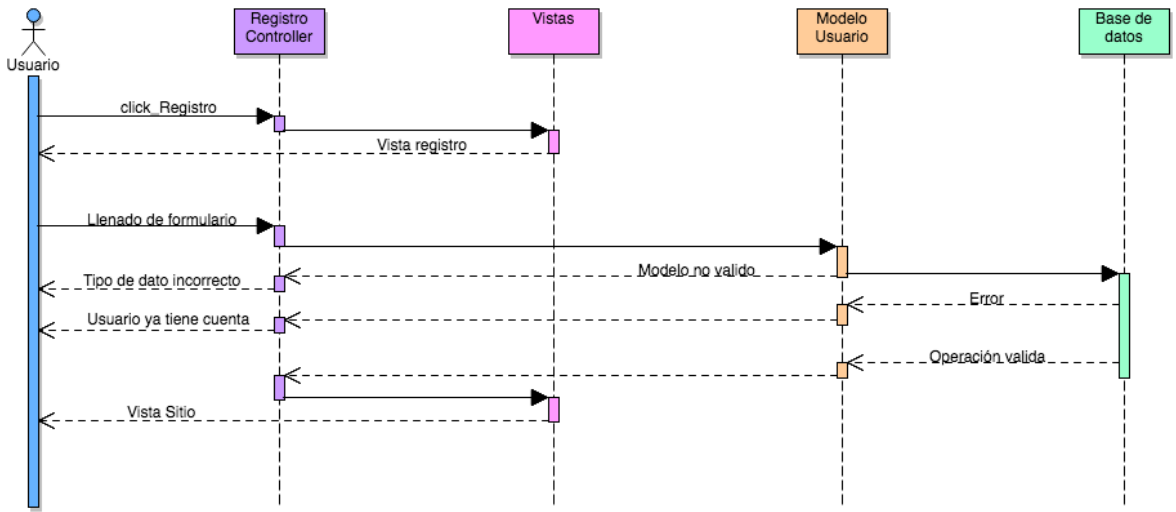


FIGURA 13. Diagrama de secuencia: Registro

- Iniciar sesión

<b>Caso de uso Iniciar sesión</b>	
<i>Actores</i>	Usuario con cuenta
<i>Tipo</i>	Básico
<i>Propósito</i>	Es el arranque del sistema de gestión de eficiencia energética, el sistema le brinda los permisos para ingresar los datos pertenecientes a la empresa y a los equipos.



<i>Resumen</i>	Cuando un usuario inicia sesión el sistema habilita las secciones de “Cuestionario de empresa”, y si es que ya contestó el cuestionario, habilita la sección de ingresar información de equipos.
<i>Precondicion es</i>	El usuario debe estar registrado.
<i>Flujo principal</i>	El sistema presenta al usuario la vista de “Inicio de sesión” (Apéndice A.4.7). El usuario escribe el nombre de usuario y contraseña, si estos son correctos puede ingresar al sistema.

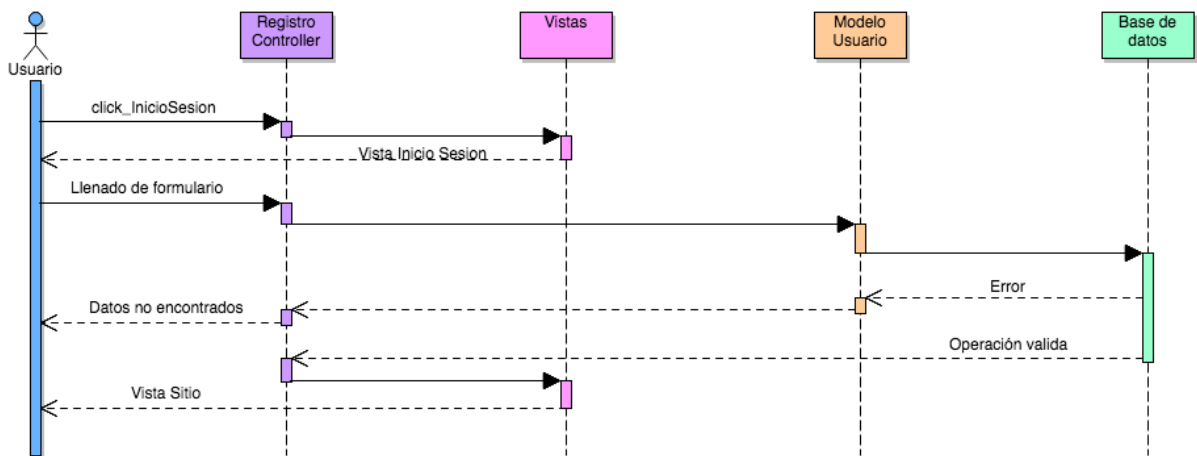


FIGURA 14. Diagrama de secuencia: Iniciar Sesión

- Agregar información de empresa

<b>Caso de uso Agregar información de la empresa</b>	
<i>Actores</i>	Usuario con cuenta
<i>Tipo</i>	Básico
<i>Propósito</i>	La información de la empresa ayuda al sistema a saber qué tipo de procesos y equipos pertenecen a la empresa según su rama industrial.

<i>Resumen</i>	El sistema recopila información importante para la empresa, esta información es utilizada por el sistema para el análisis y habilita otras secciones del sistema.
<i>Precondicion es</i>	El usuario debe iniciar sesión y no haber contestado la encuesta de empresa previamente.
<i>Flujo principal</i>	El sistema presenta al usuario la vista de “Encuesta de empresa” (Apéndice A.4.9). El usuario debe ingresar la información pedida por el cuestionario: nombre de la empresa, RFC, Estado, Delegación o municipio, Calle, Colonia, Código Postal, Teléfono, Página web y Ramo industrial.
<i>Flujo secundario</i>	Una vez contestada la encuesta de empresa se habilitan los formularios de los equipos pertenecientes a la Rama industrial señalada.

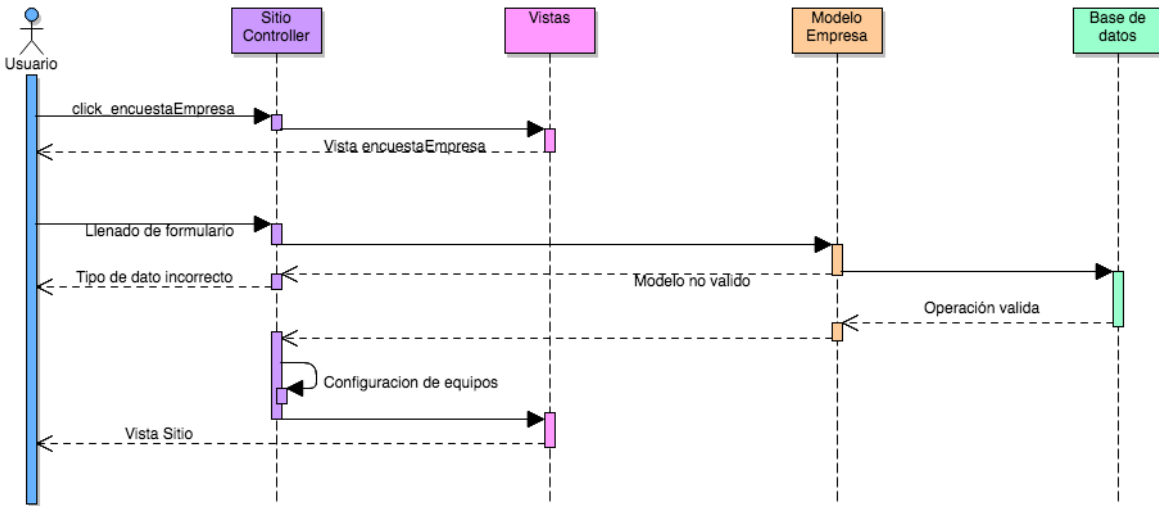


FIGURA 15. Diagrama de secuencia: Agregar información de la empresa

- Agregar información de equipos

<b>Caso de uso Agregar información de equipos</b>	
<i>Actores</i>	Usuario con cuenta
<i>Tipo</i>	Básico
<i>Propósito</i>	El propósito es obtener los datos de placa y datos medidos de los equipos que pertenecen a la empresa.
<i>Resumen</i>	El usuario puede elegir el equipo que desee llenar de un menú localizado en el costado del sistema. Después deberá llenar los datos pedidos en el formulario de dicho equipo para poder tener un diagnóstico energético.
<i>Precondiciones</i>	El usuario deberá iniciado sesión y deberá llenar la encuesta de Empresa.
<i>Flujo principal</i>	El sistema presenta al usuario la vista del sitio (Apéndice A.4.8) en donde el usuario hace la elección de equipo a llenar en el menú al costado del sistema, dependiendo de la selección el sistema se llama al subflujo elegido I, II, III, IV, V y VI . El usuario deberá llenar del formulario del equipo siguiendo la secuencia del sistema, en esta secuencia puede solicitar hacer el diagnóstico o guardar los datos y volver a la vista del sitio.
<i>Flujo secundario</i>	<p>I. Caldera</p> <p>El sistema presenta al usuario la vista “Caldera” (Apéndice A.4.10) donde el usuario debe llenar los datos pedidos en el formulario. El usuario puede elegir “Diagnóstico” o “Guardar y salir”.</p>

Si se elige “Diagnóstico” se va al caso de uso Consultar diagnóstico energético.

Si se elige “Guardar y salir” se va al caso de uso Inicio de sesión.

## II. Motor eléctrico

El sistema presenta al usuario la vista “Motor eléctrico” (Apéndice A.4.11) donde el usuario debe llenar los datos pedidos en el formulario. El usuario puede elegir “Diagnóstico” o “Guardar y salir”.

Si se elige “Diagnóstico” se va al caso de uso Consultar diagnóstico energético.

Si se elige “Guardar y salir” se va al caso de uso Inicio de sesión.

## III. Sistema de bombeo

El sistema presenta al usuario la vista “Sistema de bombeo” (Apéndice A.4.12) donde el usuario debe llenar los datos pedidos en el formulario. El usuario puede elegir “Diagnóstico” o “Guardar y salir”.

Si se elige “Diagnóstico” se va al caso de uso Consultar diagnóstico energético.

Si se elige “Guardar y salir” se va al caso de uso Inicio de sesión.

## IV. Iluminación

El sistema presenta al usuario la vista “Iluminación” (Apéndice A.4.13) donde el usuario debe llenar los datos pedidos en el formulario. El usuario puede elegir “Diagnóstico” o “Guardar y salir” o “Agregar luminaria” o “Agregar área”. Si se elige “Diagnóstico” se va al caso de uso Consultar diagnóstico energético.

Si se elige “Guardar y salir” se va al caso de uso Inicio de sesión.

Si se elige “Agregar luminaria” se va al caso de uso Agregar información de equipos con uso de la vista “Luminaria”.

Si se elige “Agregar Área” se va al caso de uso Agregar información de equipos con uso de la vista “Iluminación” (Apéndice A.3.13).

#### V. Compresor

El sistema presenta al usuario la vista “Compresor” (Apéndice A.4.14) donde el usuario debe llenar los datos pedidos en el formulario. El usuario puede elegir “Diagnóstico” o “Guardar y salir”. Si se elige “Diagnóstico” se va al caso de uso Consultar diagnóstico energético.

Si se elige “Guardar y salir” se va al caso de uso Inicio de sesión.

## VI. Horno de flama

El sistema presenta al usuario la vista “Horno de flama” (Apéndice A.4.15) donde el usuario debe llenar los datos pedidos en el formulario. El usuario puede elegir “Diagnóstico” o “Guardar y salir”.

Si se elige “Diagnóstico” se va al caso de uso Consultar diagnóstico energético.

Si se elige “Guardar y salir” se va al caso de uso Inicio de sesión.

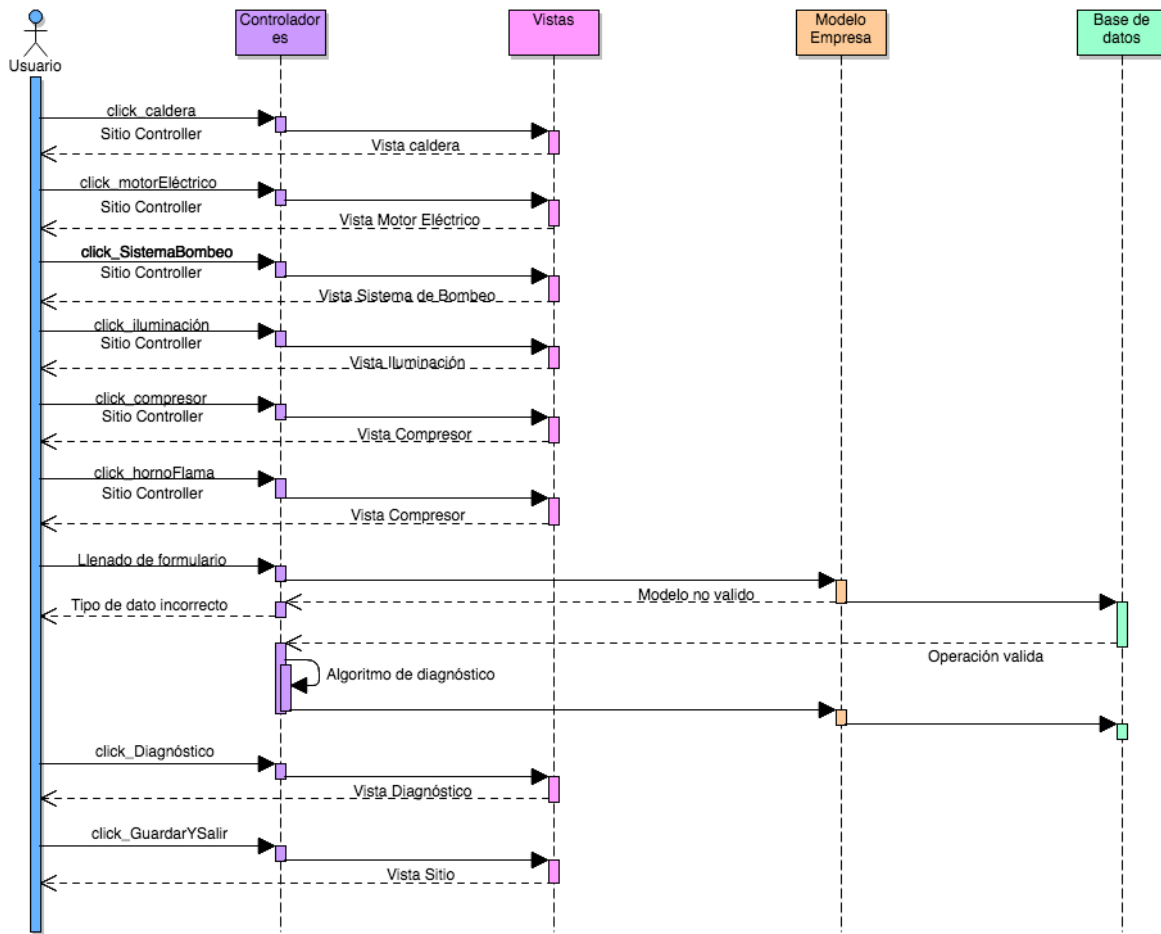


FIGURA 16. Diagrama de secuencia: Agregar información de equipos

- Consultar diagnósticos y recomendaciones energéticas

<b>Caso de uso Consultar Diagnósticos Energéticos</b>	
<i>Actores</i>	Usuario con cuenta
<i>Tipo</i>	Básico
<i>Propósito</i>	El sistema brinda el diagnóstico energético y una serie de recomendaciones hechas por el sistema.
<i>Resumen</i>	El sistema brinda el diagnóstico energético y junto a él una serie de recomendaciones para uso del equipo después de que el usuario haya llenado el formulario.
<i>Precondicion es</i>	El usuario deberá hacer llenado el formulario del equipo del cual quiere conocer el diagnóstico.
<i>Flujo principal</i>	Una vez llenado el formulario del equipo, el usuario podrá saber el diagnóstico energético de dicho equipo si es que selecciona al final del formulario “Diagnóstico”. Dependiendo del equipo el sistema seleccionará alguno de los subflujos I, II, III, IV, V o VI.
<i>Flujo secundario</i>	<p>I. Diagnóstico Caldera</p> <p>El sistema presenta al usuario la vista “Diagnóstico Caldera” donde el usuario puede observar el diagnóstico del equipo y sus recomendaciones. El usuario puede presionar “Guardar y salir”.</p> <p>Si se elige “Guardar y salir” se va al caso de uso Iniciar sesión.</p> <p>II. Diagnóstico Motores eléctricos</p>

El sistema presenta al usuario la vista “Diagnóstico Motores Eléctricos” donde el usuario puede observar el diagnóstico del equipo y sus recomendaciones. El usuario puede presionar “Guardar y salir”.

Si se elige “Guardar y salir” se va al caso de uso Iniciar sesión.

### III. Diagnóstico Sistema de bombeo

El sistema presenta al usuario la vista “Diagnóstico Sistema de bombeo” donde el usuario puede observar el diagnóstico del equipo y sus recomendaciones. El usuario puede presionar “Guardar y salir”.

Si se elige “Guardar y salir” se va al caso de uso Iniciar sesión.

### IV. Diagnóstico Iluminación

El sistema presenta al usuario la vista “Diagnóstico Iluminación” donde el usuario puede observar el diagnóstico del equipo y sus recomendaciones. El usuario puede presionar “Guardar y salir”.

Si se elige “Guardar y salir” se va al caso de uso Iniciar sesión.

### V. Diagnóstico Compresor



El sistema presenta al usuario la vista “Diagnóstico Compresor” donde el usuario puede observar el diagnóstico del equipo y sus recomendaciones. El usuario puede presionar “Guardar y salir”.

Si se elige “Guardar y salir” se va al caso de uso Iniciar sesión.

#### VI. Diagnóstico Horno de Flama

El sistema presenta al usuario la vista “Diagnóstico Horno” donde el usuario puede observar el diagnóstico del equipo y sus recomendaciones. El usuario puede presionar “Guardar y salir”.

Si se elige “Guardar y salir” se va al caso de uso Iniciar sesión.

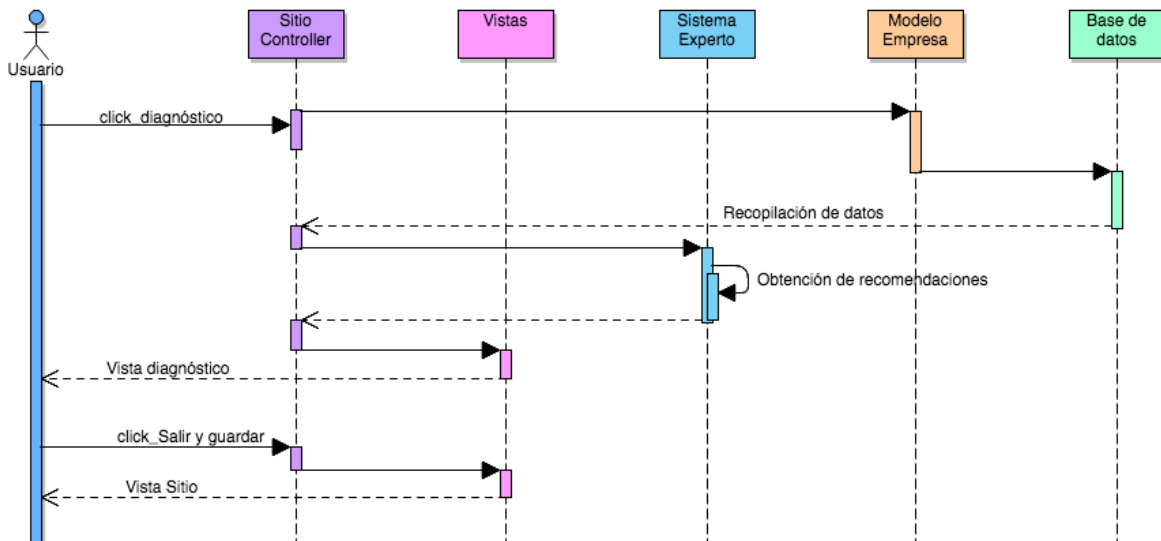


FIGURA 17. Diagrama de secuencia: Consultar diagnósticos energéticos

## 4.6. Modelado y desarrollo de la Base de Datos

Una vez adquiridos los datos y procesos de análisis se hizo el modelo y el desarrollo de la base de datos (BD). Se pueden distinguir 4 principales sectores de datos los cuales son: Datos de la empresa, datos del usuario, datos relacionados con la rama industrial y con sus equipos, así como la serie de recomendaciones y áreas de oportunidad.

A continuación, se muestra el diagrama de los principales componentes de la base de datos.

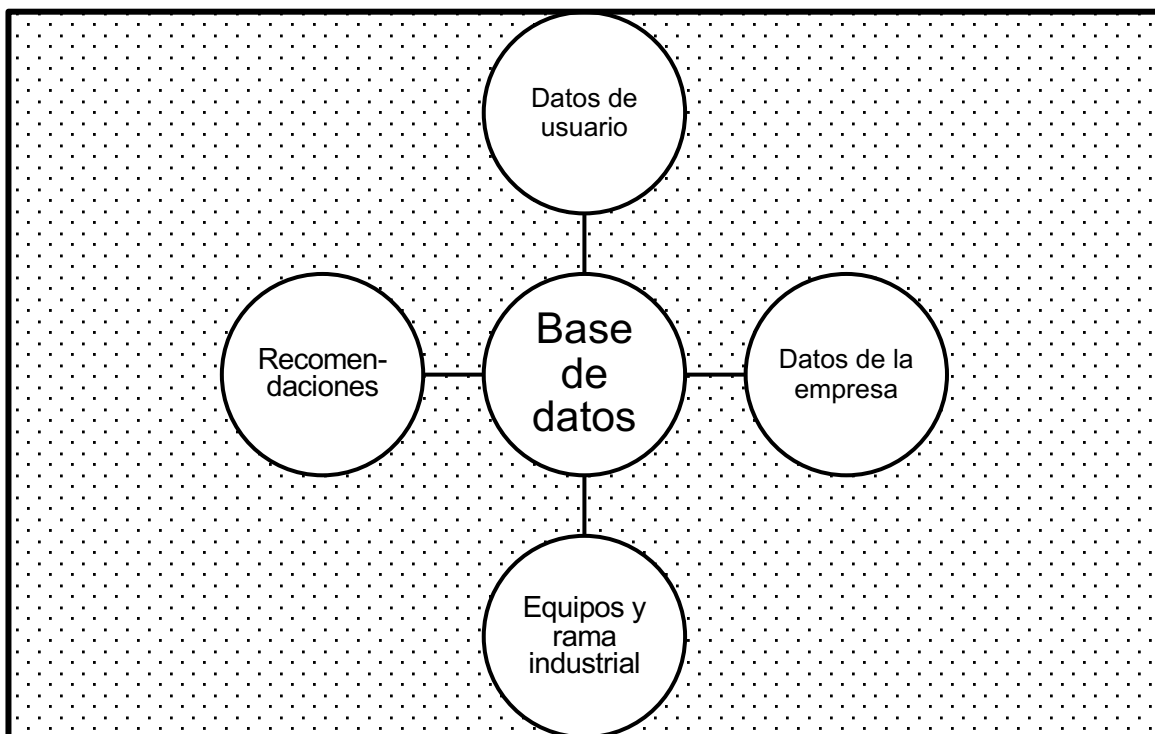


FIGURA 18. Diagrama de Bloques: Base de datos

Se eligió *Postgresql* como manejador de la BD. Una de las principales razones es porque es un proyecto de software libre (*Open Source*) y esta liberado por la licencia BSD (*Berkeley Software Distribution*).

El modelo de desarrollo de la base de datos es el Modelo Relacional, se eligió por ser un modelo robusto y por ser el más utilizado para implementar BD actualmente.

De acuerdo con las características del sistema se definieron 12 tablas principales donde se almacenan los datos. Cada equipo a su vez necesita catálogos, estos catálogos almacenan datos necesarios para el análisis de los equipos tales como factores, unidades, características de equipo etc.

En la siguiente tabla se muestra las entidades utilizadas en el desarrollo de la BD. En el apéndice A.2 se puede visualizar el modelo Entidad-Relación (ER) de la base de datos.

<b>Entidad</b>	<b>Descripción</b>	<b>Atributos</b>
<i>Usuario</i> ( <i>usuariosinfo</i> )	Almacena la información perteneciente al usuario.	idUser nombres paterno materno email puesto
<i>Empresa</i>	Almacena la información perteneciente a la empresa.	idEmp idUser nombre rfc estado municipio calle colonia codigoPostal teléfono paginaWeb ramoIndustrial
<i>Credencial de Usuario</i> ( <i>userlogs</i> )	Almacena los datos que identifican al usuario, estos datos permiten ingresar al sistema.	idUser userName pwd

<i>Rama industrial (ramo_industrial)</i>	Almacena el nombre e identificador de cada rama industrial.	idRi nombre
<i>Relación Rama- Equipo (ramosbyequipo)</i>	Almacena la matriz de aparición. Cada rama industrial tiene asociado cierto numero de equipos.	idRe ramo equipo aplica
<i>Estado (entidad_cat)</i>	Almacena las características del estado al que pertenece la empresa.	idEnt nombre temp humedad
<i>Caldera (caldera)</i>	Almacena los datos pertenecientes a la caldera.	idCaldera marca modelo capacidad área fecha unidadAT bombaMarca bombaPot unidadPT ventiladorMarca ventiladorPot tipoCal
<i>Motor Eléctrico (motorEl)</i>	Almacena los datos pertenecientes al motor eléctrico.	idMotor capacidad tensión corriente revoluciones factorPotencia eficiencia tipo horas reparaciones días antigüedad costo unidadPT

<p><i>Sistema de bombeo (bombas)</i></p>	<p>Almacena los datos pertenecientes al sistema de bombeo.</p>	<p>idBomba  aplicación  operación  volCont  tiempoLlenado  unidadVolCont  unidadTime  numServicios  diámetro  calFricción  presionSucc  fluidoBomb  presionMan  materialTub  hMax  qMax  hOp  qMin  potMot  corriente  eficiencia  tensión  costo  factPot  horas  días</p>
<p><i>Iluminación (iluminación)</i></p>	<p>Almacena los datos pertenecientes al sistema de iluminación (áreas, luminarias y lámparas).</p>	<p>idArea  nombreArea  numArea  tipoArea  nom25  dpea  largoArea  anchoArea  altoArea  colorTecho  colorPared  colorPiso  alturaTrabajo  alturaMontaje  horasDia  diasAnio  lumxArea  potxArea  lumUtilesArea  iluCalculado  dpeaCalculado  consumoEnergético</p>

<p><i>Compresor (compresor)</i></p>	<p>Almacena los datos pertenecientes al sistema de compresor.</p>	<p>idCompresor  tipoCom  potNominal  tipoTrans  presionMax  flujo  presionOp  temp  tempSalida  horasDia  diasAnio  costoEner  enfriamiento  cambioA  limpiezaC  ajusteTrans  ubicaciónC  AniosOp  flujoOp  fugas  efiNomMotor  efiRealmotor  potDemanda  efiTrans  eficompresion  ahorroenerMan  ahorroEcoMan  motoCom  disAhorro  EnerAnual  costoAnual</p>
<p><i>Horno de Flama (hornos)</i></p>	<p>Almacena los datos pertenecientes al horno de combustible.</p>	<p>Clave  unidadProd  tipo  producto  year  combustible  aplicación  consumo  tiempo  días  gasescomb  gasesChimenea  tAmbiente  tAire  tprodIni  tprodFin  tInterior</p>

		masa
		o2
		co2
		co

#### 4.7. Codificación de algoritmos por proceso industrial.

Una vez validado el conocimiento adquirido, se codificaron los algoritmos de análisis. Cada algoritmo está asociado a un equipo, es importante mencionar que todos los diagnósticos energéticos son diferentes ya que se adaptan al equipo y a los datos brindados por el experto. En la siguiente tabla se muestra el nombre del algoritmo asociado a cada equipo.

<b><i>Equipo o proceso industrial</i></b>	<b><i>Nombre del algoritmo en el sistema</i></b>
<i>Caldera</i>	CalderaController
<i>Motor Eléctrico</i>	MotoresElController
<i>Sistema de bombeo</i>	BombasController
<i>Iluminación</i>	IluminacionController
<i>Compresor</i>	CompresorController
<i>Hornos de flama</i>	HornosController

El objetivo del algoritmo de análisis es hacer los cálculos con los datos ingresados por el usuario, analizar los resultados de dichos cálculos y mandar los diagnósticos al área de recomendaciones.

A continuación, se muestra un diagrama general de las tareas pertenecientes a cada capa de la arquitectura MVC del sistema destinado al análisis de los equipos.

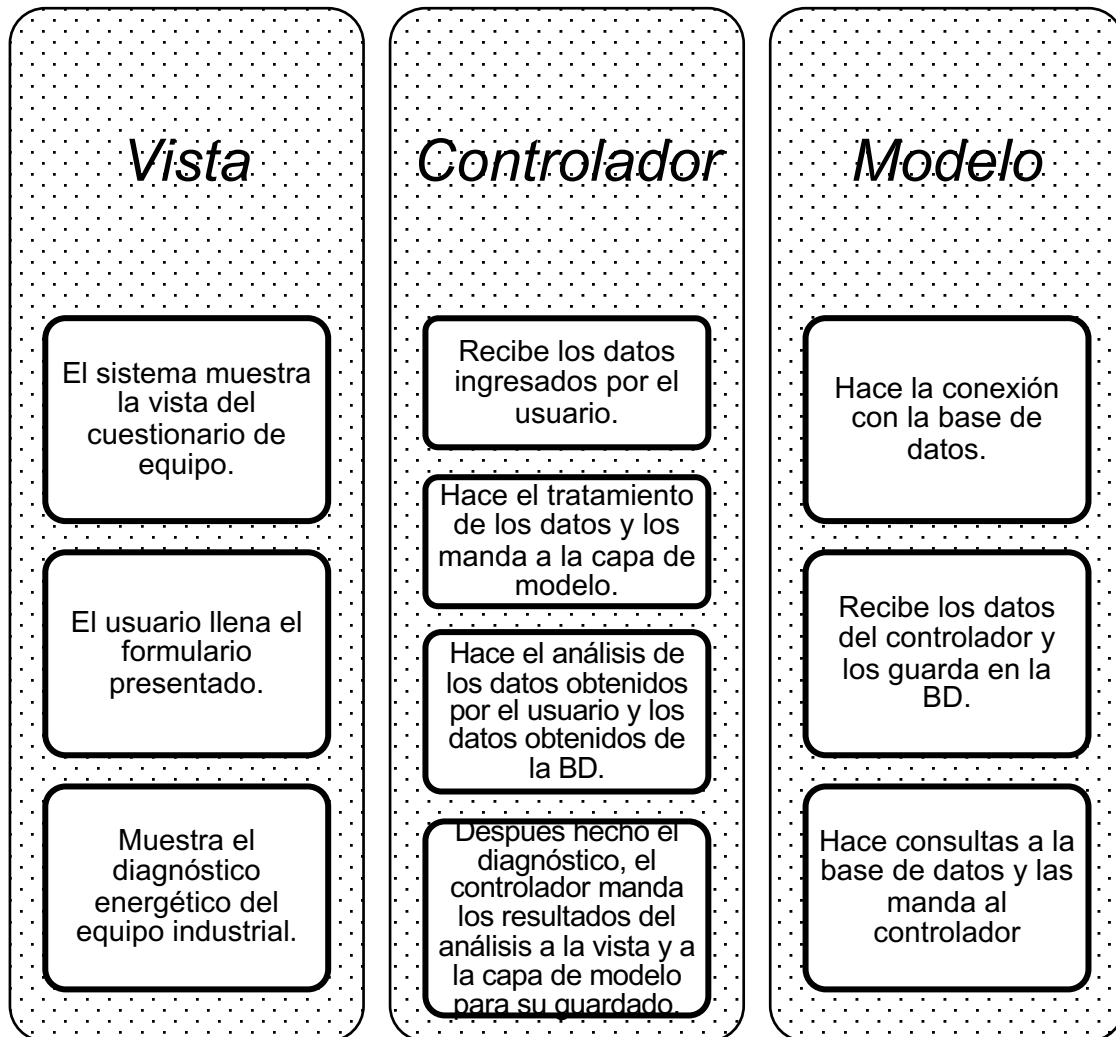


FIGURA 19. Clasificación de tareas en estructura MVC

En la siguiente sección se listan los equipos, los pseudocódigos, sus diagramas de flujo y sus respectivos códigos en el lenguaje de programación C#.

- Caldera

La estructura del algoritmo de caldera contempla datos de entrada y datos de salida. Los datos de entrada se dividen en dos categorías: datos de placa y datos medidos, esta información es proporcionada por el usuario. En la siguiente tabla se muestran los datos de entrada y salida del algoritmo de caldera.



<b>Datos de entrada</b>		
	<i>Nombre</i>	<i>Identificativo</i>
<i>Datos de placa y combustible</i>	Tipo de caldera Marca Modelo Capacidad Área de transferencia Fecha de instalación Marca de la bomba Modelo de la bomba Potencia de la bomba Marca de ventilador Modelo de ventilador Potencia del ventilador Tipo de combustible Consumo de combustible Temperatura del combustible Bandera vapor sobrecalentado Presión del vapor Temperatura del vapor Estado del aislante Temperatura del aislante	tipoCal marca modelo capacidad área fecha bombaMarca bombaModelo bombaPot ventiladorMarca ventiladorModelo ventiladorPot comb consumo temperatura presionSobre presionVP temperaturaVP estadoAS aislanteTemp
<i>Datos medidos</i>	Presión agua alimentación (AA) Temperatura AA Flujo AA Bandera flujómetro Oxígeno Dióxido de carbono Óxido de nitrógeno Temperatura Exceso de aire	presionAA temperaturaAA flujo auxFlujo o2 co2 nox gasesTemp aire

<b>Datos de salida</b>	
<i>Nombre</i>	<i>Identificativo</i>
<i>Eficiencia energética</i> <i>Recomendaciones generales</i> <i>Recomendaciones de aislante</i> <i>Recomendaciones de temperatura del aislante</i>	Eficiencia Recomendaciones Raislante Rtemp

Después de obtener la información, el sistema hace los cálculos necesarios para obtener los datos de salida. A continuación, se muestra el pseudocódigo, es cual esta diseñado para su fácil lectura y que ejemplifica la secuencia de los datos y los cálculos de ellos.

```

programa caldera

    mostrar vista caldera
    obtener tipoCal, marca, modelo, capacidad, area, fecha,
    bombaMarca, bombaModelo, bombaPot, ventiladorMarca,
    ventiladorModelo, ventiladorPot
    mostrar formulario de combustible
    obtener comb, consumo, presion, temperatura, presionSobre,
    presionVP, temperaturaVP

    Si presionSobre es verdadero entonces (significa que se utiliza
    vapor sobrecalentado)
        obtener presionVP, temperaturaVP
    Si no entonces
        presionVP <- presion
        temperaturaVP <- temperatura
    fin del Si

    mostrar formulario de medición
    obtener presionAA, temperaturaAA, flujo, auxFlujo
    Si auxFlujo es falso entonces (significa que el usuario no
    tiene calorímetro ni flujometro)
        obtener o2, co2, nox, gasesTemp, aire
        Perdida_por_gases <- obtener de tabla
        gases_vs_temperaturaAgua
        Perdida_por_inquemados <- (21/(21-o2)) / (co2/3100 +
        nox/1000)
        Perdida_por_purgas <- 5
        Perdida_por_radiación <- obtener de tabla
        capacidadCaldera_vs_radiacion
        Eficiencia <- 100 - Perdida_por_gases -
        Perdida_por_inquemados - Perdida_por_purgas -
        Perdida_por_radiacion
    Si no entonces
        obtener flujo
        Eficiencia <- (flujo * (entalpiaE_VaporSaturado -
        estalpiaE_AguaSaturada)) / poderCalorífico_combustible
        * consumo
    fin del Si
    Guardar datos en base de datos
    Mandar datos a sistema de recomendaciones

fin programa caldera

```

El diagrama de flujo muestra el funcionamiento del programa de manera gráfica antes de realizar el código final. A continuación, se muestra el diagrama de flujo y en el Apéndice A.3.1 se puede visualizar el código completo.

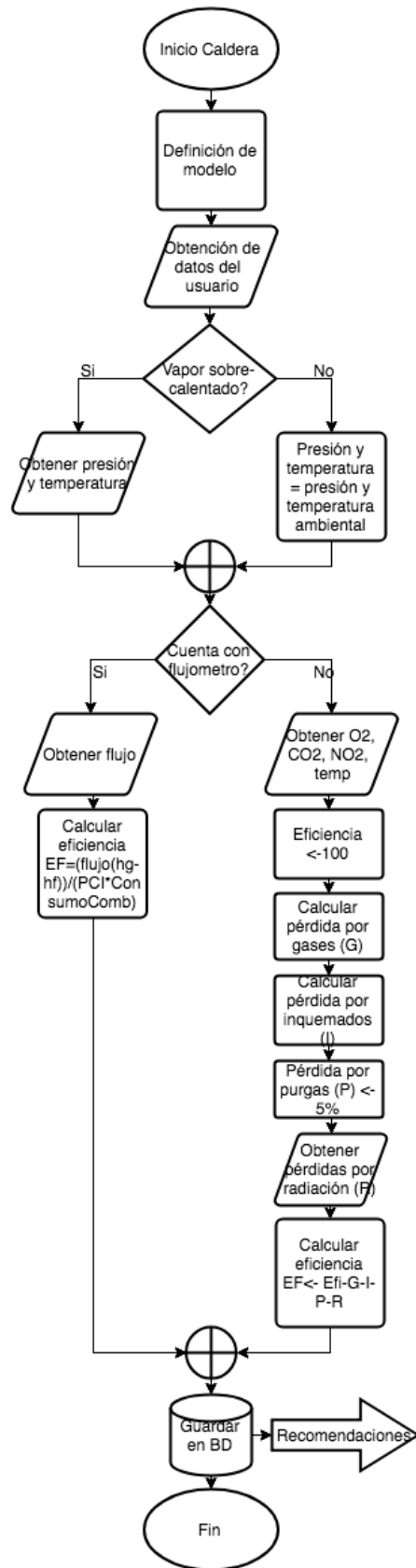


FIGURA 20. Diagrama de flujo: Caldera

- Motor eléctrico

En la siguiente tabla se muestra los datos de entrada (Datos de placa y datos medidos) y los datos de salida del algoritmo de motor eléctrico.

<b>Datos de entrada</b>		
	<i>Nombre</i>	<i>Identificativo</i>
<i>Datos de placa</i>	Capacidad mecánica Tensión Corriente nominal RPM Factor de potencia Eficiencia de placa Tipo de motor	Capacidad tensión corriente revoluciones factorPotencia eficiencia tipo
<i>Datos de operación</i>	Horas de uso al día Reparaciones o rebobinados Días de operación al año Antigüedad Costo medio anual	horas reparaciones días antigüedad costo
<i>Datos medidos</i>	Corriente fase 1 Corriente fase 2 Corriente fase 3 Tensión fase 1 Tensión fase 2 Tensión fase 3 Factor de potencia 1 Factor de potencia 2 Factor de potencia 3	Corriente1 Corriente2 Corriente3 Voltaje1 Voltaje2 Voltaje3 Fp1 Fp2 Fp3

<b>Datos de salida</b>	
<i>Nombre</i>	<i>Identificativo</i>
<i>Eficiencia</i>	Eficiencia
<i>Recomendaciones Generales</i>	Rgeneral
<i>Recomendaciones de Antigüedad</i>	Ranti
<i>Recomendaciones de rebobinados</i>	Rrebobinado

El siguiente pseudocódigo describe el funcionamiento del algoritmo del motor eléctrico.

```

programa motor

  mostrar vista motor
  obtener capacidad, tension, corriente, revoluciones,
  factorPotencia, eficienciaPla, tipo
  mostrar formulario de operación
  obtener horas, reparaciones, dias, antigüedad, costo
  
```

```

mostrar formulario de medición
obtener corriente1, corriente2, corriente3, voltaje1,
voltaje2, voltaje3, fp1, fp2, fp3

Si tipo es std entonces (estándar)
    obtener eficiencia de tabla motores_std
    obtener factor_carga de tabla motores_std
Si no entonces
    obtener eficiencia de tabla motores_altaFrec
    obtener factor_carga de tabla motores_altaFrec
fin del Si

desbalance <- (MAX (voltaje1, voltaje2, voltaje3) -
PROMEDIO(voltaje1, voltaje2, voltaje3)) /
PROMEDIO(voltaje1, voltaje2, voltaje3)
factor_desbalance <- 1 - desbalance * (0.0113 +
0.0073 * desbalance)
VV <- (PROMEDIO(voltaje1, voltaje2, voltaje3) /
tension) - 1
diferencia_voltaje <- VV * (0.07 - 1.334 * VV) -
0.0009

Si antigüedad es mayor a 10 entonces
    factor_antigüedad <- 0.01
Si no entonces
    factor_antigüedad <- 0.0
fin del Si

Si reparaciones es mayor a 2
    factor_rebobinado <- 0.025
Si no entonces
    factor_rebobinado <- 0.0
fin del Si

Eficiencia <- factor_desbalance * (eficienciaPla +
diferencia_voltaje - factor_antigüedad -
factor_rebobinado)

fin del programa motor

```

A continuación, se muestra el diagrama de flujo perteneciente al algoritmo del motor eléctrico. Se puede visualizar el programa completo en el Apéndice A.3.2.

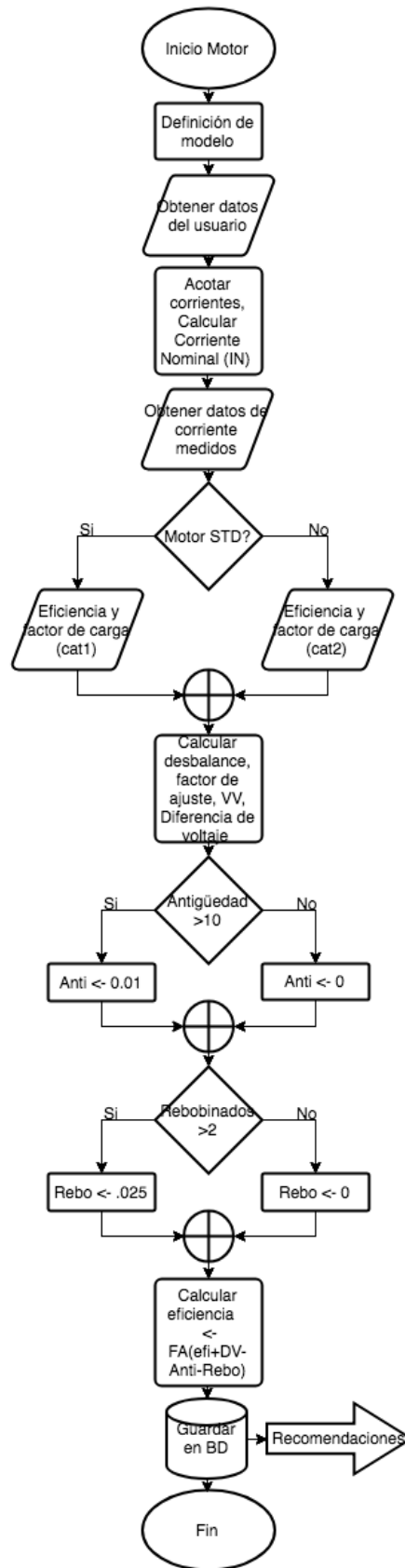


FIGURA 21. Diagrama de flujo: Motor Eléctrico

- Sistema de bombeo

En la siguiente tabla se muestra los datos de entrada: datos de placa y datos medidos, así mismo de muestran los datos de salida, los cuales son calculados por el sistema.

<b>Datos de entrada</b>		
	<i>Nombre</i>	<i>Identificativo</i>
<i>Datos del sistema</i>	Aplicación Bandera de flujo de operación Volumen del contenedor Tiempo de llenado del contenedor Número de servicios Diámetro de la conexión de entrada Criterio para cálculo de fricción Criterio de presión en la succión	aplicación operación volCont tiempoLlenado numServicios diámetro calFriccion presionSucc
<i>Datos de placa de la bomba</i>	Fluido bombeado Presión manométrica Material de tubo Carga máxima (Hmax) Gasto máximo (Qmax) Carga óptima (Hop) Gasto mínimo (Qmin)	fluidobomb presionMan materialTub hMax qMax hOp qMin
<i>Datos de operación eléctrica</i>	Potencia del motor Corriente medida Eficiencia del motor Tensión medida Costo de la energía Factor de potencia	potMot corriente eficiencia tension costo facPot
<i>Datos de operación hidráulica</i>	Gasto de operación Horas de operación al día Presión de operación Días de operación al año Distancia vertical Distancia horizontal	qOp horas presionOp dias distVert distHorz

<b>Datos de salida</b>	
<i>Nombre</i>	<i>Identificativo</i>
<i>Carga (H)</i>	Carga
Eficiencia conjunto bomba-motor	EficienciaConj
Gasto (Q)	Gasto
Eficiencia de bomba estimada	EficienciaBom
Potencia Hid	Potencia
Demanda eléctrica	Demanda1
CNPS disponible	CNPS
Energía aprovechada por el sistema	EnerAp

A continuación, se muestra el diagrama de flujo perteneciente al algoritmo de bomba. Se puede visualizar el pseudocódigo y el programa completo en el Apéndice A.3.3.

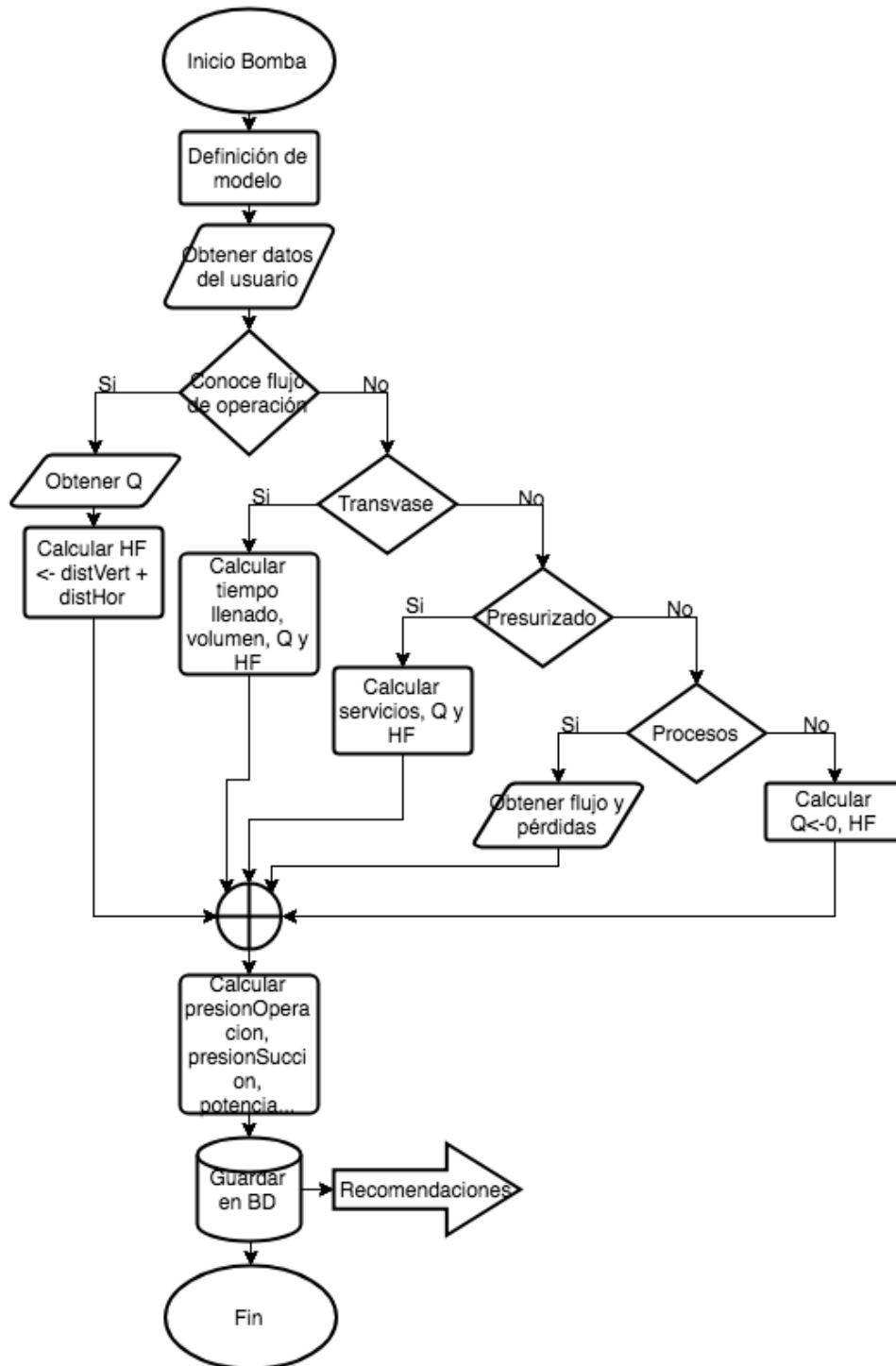


FIGURA 22. Diagrama de flujo: Bomba



- Iluminación

En la siguiente tabla se muestra los datos de entrada y los datos de salida del algoritmo de iluminación.

<b>Datos de entrada</b>		
	<i>Nombre</i>	<i>Identificativo</i>
<i>Datos del área de trabajo</i>	Nombre del área Tipo de área Largo Ancho Altura Color del techo Color de la pared Color del piso Altura del plano de trabajo Altura de montaje del luminaria	nombreArea tipoArea largoArea anchoArea altoArea colorTecho colorPared colorPiso alturaTrabajo alturaMontaje
<i>Datos de la luminaria</i>	Horas de operación Días de operación Tipo de luminaria Total de luminarias Tipo de lámpara Número de lámparas Lúmenes de la lámpara Potencia de la lámpara Tipo de balastro Factor de balastro Potencia de línea Mantenimiento	horasDia diasAnio tipoIllumina totalIllumina tipoLampara lampxLumina lumenLamp potLamp tipoBalastro factorBalastro potLinea mantenimiento

<b>Datos de salida</b>	
<i>Nombre</i>	<i>Identificativo</i>
Luminarias por área Potencia instalada Lúmenes útiles en el área Nivel de iluminación calculado Nivel de DPEA calculado Consumo energético anual en área	lumxArea potxArea lumUtilesArea iluCalculado dpeaCalculado consumoEnergético

A continuación, se muestra el diagrama de flujo perteneciente al algoritmo de iluminación. Se puede visualizar el pseudocódigo y el programa completo en el Apéndice A.3.4.

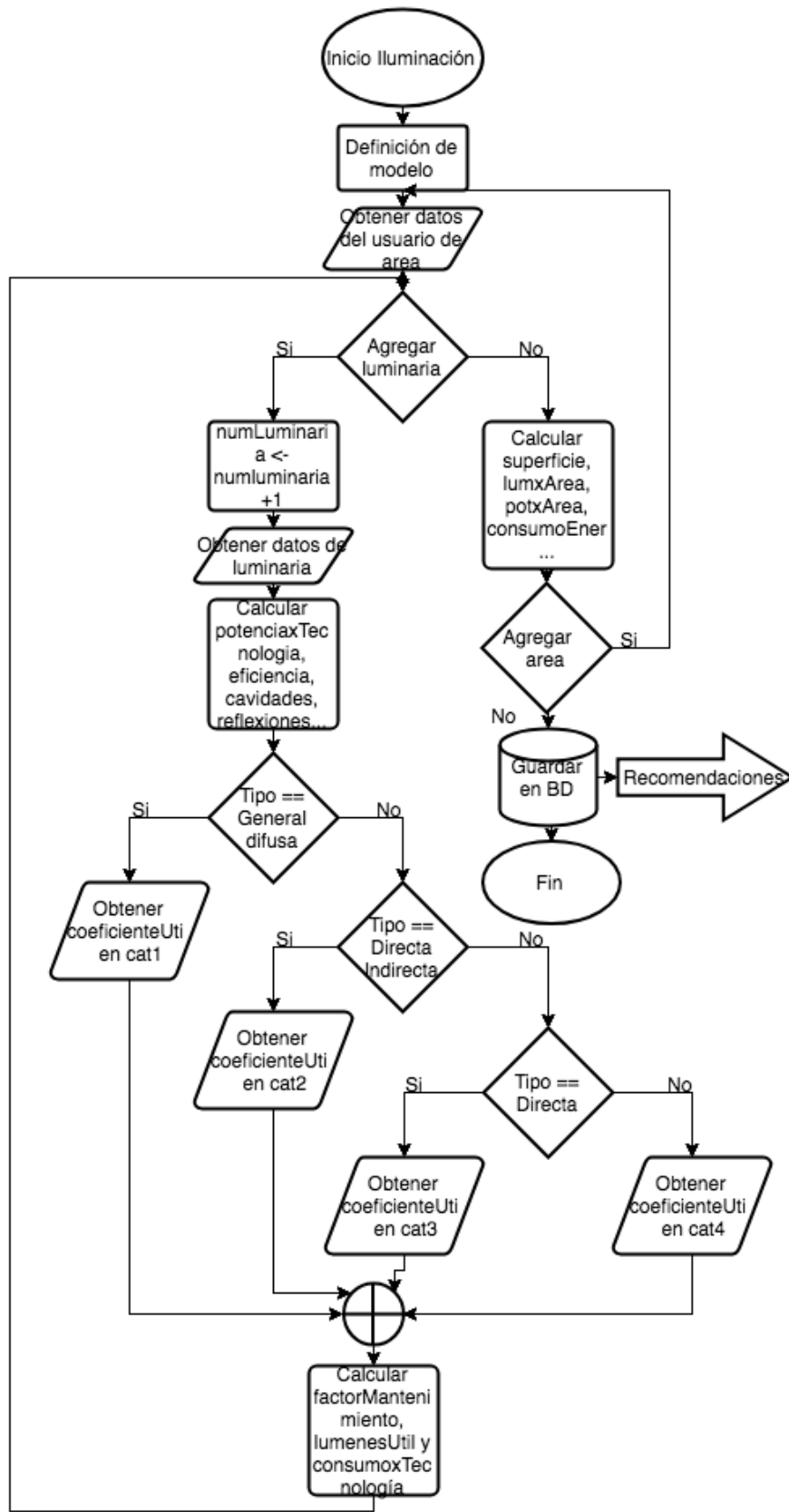


FIGURA 23. Diagrama de flujo: Iluminación

- Compresor

En la siguiente tabla se muestra los datos de entrada y los datos de salida del algoritmo del compresor.

<b>Datos de entrada</b>		
	<i>Nombre</i>	<i>Identificativo</i>
<i>Datos de placa y operación</i>	Tipo de compresor Tipo de transmisor Potencia nominal del compresor Presión máxima del compresor Flujo o caudal nominal Presión de operación Temperatura ambiente Bandera temperatura del aire Temperatura del aire de salida Horas de operación al día Días de operación al año Costo ponderado de la energía	tipoCom tipoTrans potNominal presionMax flujo presionOp temp valida tempSalida horasDia diasAnio costoEner
<i>Parámetros de operación y mantenimiento</i>	Tipo de enfriamiento Frecuencia en el cambio del aceite Frecuencia en limpieza de carcasa Frecuencia de ajuste de transmisión Ubicación del compresor Años de operación Flujo o caudal de operación Número de fugas	Enfriamiento cambioA limpiezaC ajusteTrans ubicacionC aniosOp flujoOp fugas
<i>Datos del motor</i>	Eficiencia nominal del motor Eficiencia real del motor Potencia eléctrica demandada	efiNomMotor efiRealMotor potDemanda

<b>Datos de salida</b>	
<i>Nombre</i>	<i>Identificativo</i>
Energía anual requerida Potencia demandada Costo anual Posible ahorro de energía en mantenimiento Posible ahorro económico en mantenimiento Pérdidas por fugas (porcentaje) Aire fugado por fugas Ahorro en energía Ahorro económico	enerAnual potDemanda costoAnual ahorroEnerMan ahorroEcoMan perdidasfugas aireFugado disEner disAhorro

A continuación, se muestra el diagrama de flujo perteneciente al algoritmo del compresor. Se puede visualizar el pseudocódigo y el programa completo en el Apéndice A.3.5

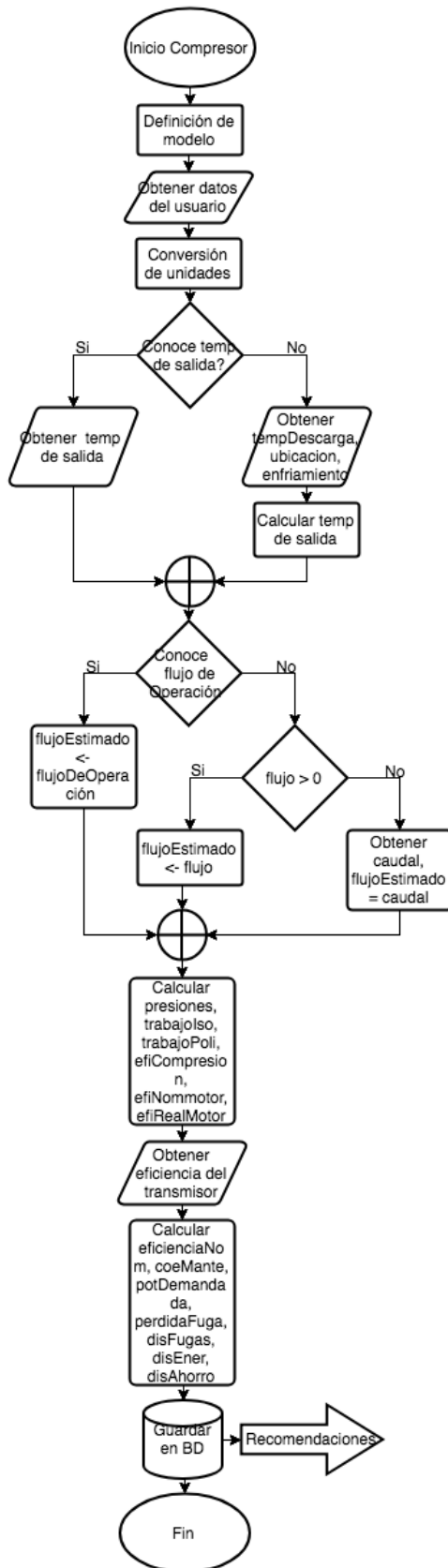


FIGURA 24. Diagrama de flujo: Compresor

- Horno de flama

En la siguiente tabla se muestra los datos de entrada y los datos de salida del algoritmo del horno de flama.

<b>Datos de entrada</b>		
	<i>Nombre</i>	<i>Identificativo</i>
<i>Datos del horno y de servicio</i>	Tipo de horno	tipo
	Año de instalación	year
	Aplicación o proceso	aplicación
<i>Datos requeridos para método indirecto</i>	Consumo de combustible	consumo
	Tiempo de Operación	tiempo
	Días de Operación	días
	Temperatura de gases de combustión	gasesComb
	Temperatura de gases de salida	gasesChimenea
	Temperatura ambiente	tAmbiente
	Temperatura de aire de entrada	tAire
	Temperatura inicial del producto	tProdIni
	Temperatura final del producto	tProdFin
	Temperatura interior del horno	tInterior
	Masa del producto horneado	masa
<i>Análisis de gases de combustión</i>	O2	O2
	CO2	CO2
	CO	CO

<b>Datos de salida</b>	
<i>Nombre</i>	<i>Identificativo</i>
N2	NN2
Exceso de Aire	exAire
Eficiencia del horno	Eficiencia

A continuación, se muestra el diagrama de flujo perteneciente al algoritmo del horno de flama. Se puede visualizar el pseudocódigo y el programa completo en el Apéndice A.3.6

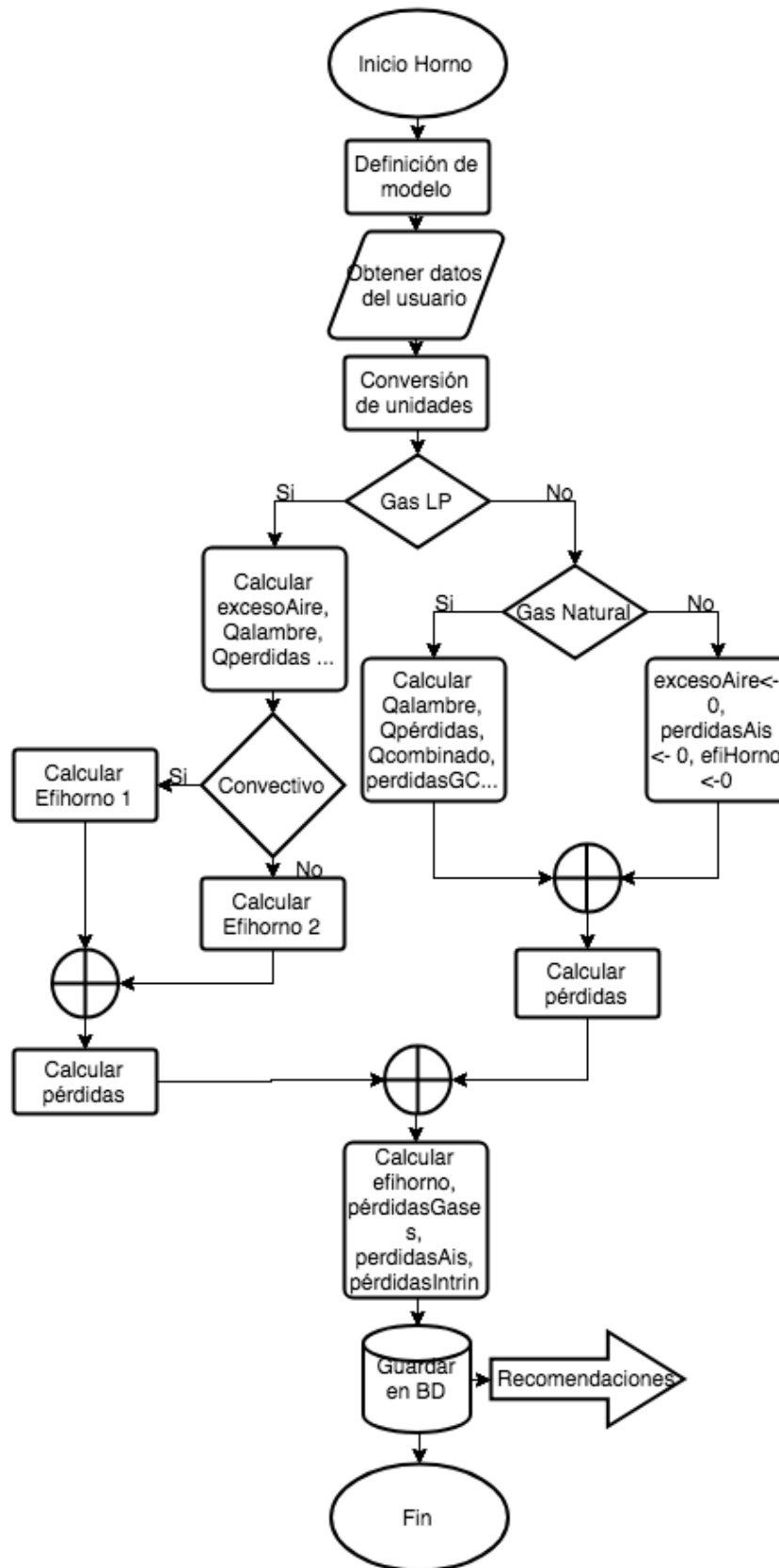


FIGURA 25. Diagrama de flujo: Horno de flama

## 4.8. Modelado y desarrollo de sistema experto destinado a recomendaciones

El objetivo principal de mostrar recomendaciones por equipo es que el usuario pueda visualizar las áreas de oportunidad y las recomendaciones. Con la realización de las recomendaciones el usuario podrá mejorar la eficiencia energética y por lo tanto el ahorro de energía.

El sistema experto forma la parte final del proceso de análisis y diagnóstico de un equipo. En el siguiente diagrama de secuencia se muestra la unión de los módulos de diagnóstico con el sistema experto y la salida entrada del usuario.

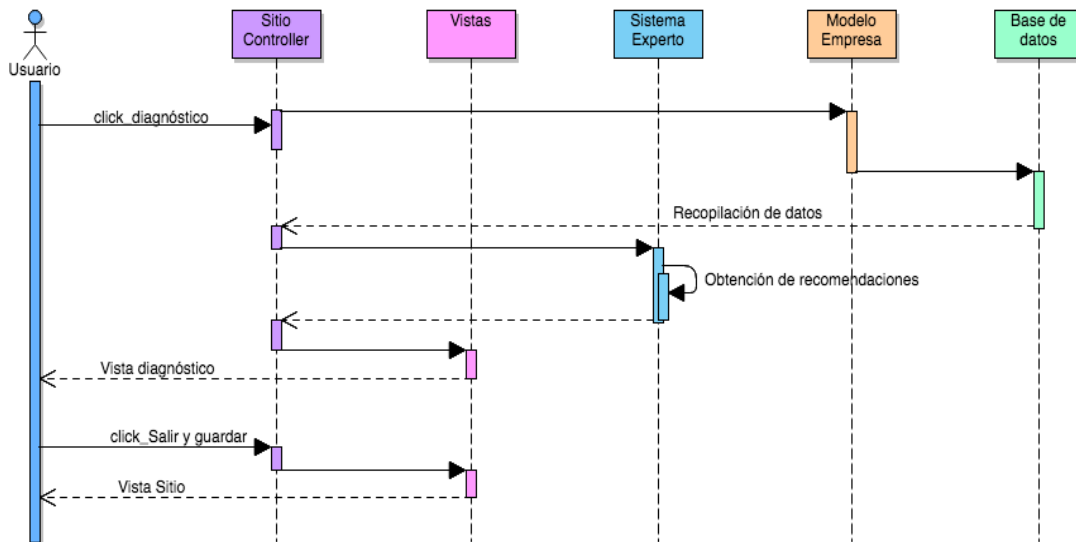


FIGURA 26. Diagrama de secuencia: Unión de módulos Diagnóstico - Recomendaciones

### Adquisición del conocimiento

Los expertos proporcionaron una serie de recomendaciones asociadas a ciertas características del equipo tales como antigüedad, mantenimiento, uso del equipo, etc. A continuación, se muestran las características del equipo y sus recomendaciones.

- Caldera

<b>Característica</b>	<b>Recomendación</b>
<i>Esta en buen estado el aislante.</i>	Mantener con la revisión continua del aislante para evitar disminuir la eficiencia.
<i>Esta en mal estado el aislante</i>	Se debe cambiar el aislante
<i>La temperatura de la tubería es mayor a 50° C.</i>	Se recomienda aislar, puede que la eficiencia de tu caldera no sea la óptima.
<i>La temperatura de la tubería esta entre 40° y 50° C.</i>	Revisa el aislante de la tubería, un aislante en mal estado, disminuye la eficiencia de la caldera.
<i>La temperatura de la tubería es menor de 40° C.</i>	La temperatura de la tubería esta en los parámetros recomendables, no olvides darle mantenimiento.

- Motor eléctrico

<b>Característica</b>	<b>Recomendación</b>
<i>El motor eléctrico tiene una antigüedad mayor a 10 años.</i>	Se recomienda cambiar el motor cuando este tiene más de 10 años, puede que la eficiencia de tu motor no sea la adecuada.
<i>El motor eléctrico tiene más de dos rebobinados.</i>	Más de dos rebobinados disminuye la eficiencia energética del motor.
<i>El motor eléctrico tiene menos de dos rebobinados.</i>	El motor eléctrico esta en los parámetros recomendables, no olvides darle mantenimiento.



- Sistema de bombeo

<b>Característica</b>	<b>Recomendación</b>
<i>La bomba tiene cargas altas y flujos pequeños.</i>	Es recomendable usar una bomba de flujo radial.
<i>La bomba tiene cargas pequeñas y flujos altos.</i>	Es recomendable usar una bomba de flujo axial.
<i>La bomba tiene cargas y caudales intermedios.</i>	Es recomendable usar una bomba diagonal.

- Iluminación

<b>Característica</b>	<b>Recomendación</b>
<i>Balastro en lámparas de tipo electromagnéticos.</i>	Sustituir lámparas por con balastro electrónico.
<i>Utilización de lámpara incandescente.</i>	Reemplazar lámpara por lámpara LED o fluorescente compacta.
<i>Luminaria con bajo coeficiente de utilización.</i>	Reemplazar la luminaria.
<i>Colores oscuros en pared techo o piso.</i>	Utilizar colores claros.
<i>La luminaria utiliza Fluorescente T12.</i>	Reemplazar lámpara por tecnología T8.
<i>La luminaria utiliza lámparas de vapor de mercurio.</i>	Reemplazar lámpara por aditivos metálicos.
<i>La luminaria utiliza Fluorescente T5 o T8</i>	Reemplazar lámpara por lámpara LED.
<i>No cumple con norma NOM007 si cumple con NOM025.</i>	Redistribuir luminarias, cambiar lámparas con características similares, pero con menor potencia.

<i>Si cumple con norma NOM007 no cumple con NOM025.</i>	Cambiar lámparas con la misma potencia, pero con diferentes características, también puedes cambiar el color de las paredes y el techo por colores claros.
<i>No cumple con ninguna norma.</i>	Debes cambiar las lámparas y luminarias ya que esta afectando la eficiencia energética de tu empresa.

- Compresor

<b>Característica</b>	<b>Recomendación</b>
<i>La eficiencia del motor está por debajo de la mínima aceptable.</i>	Se sugiere realizar el mantenimiento correspondiente o considerar la sustitución del motor.
<i>La eficiencia del motor es aceptable.</i>	Se sugiere establecer un programa de mantenimiento para mantener en buen estado el motor.
<i>La eficiencia de la transmisión está por debajo de la mínima aceptable.</i>	Se sugiere realizar los ajustes necesarios (tensados de bandas, rectificación de engaños etc.) o sustituir los elementos de la transmisión.
<i>La eficiencia de la transmisión es aceptable.</i>	Se sugiere revisar los elementos de transmisión continuamente y ajustarlos cuando sea necesario para mantener la eficiencia.

<i>La eficiencia del compresor está por debajo de la mínima aceptable.</i>	Se sugiere realizar una limpieza superficial del mismo, verificar el nivel de aceite y el estado de las partes móviles.
<i>La eficiencia del compresor es aceptable.</i>	La eficiencia del cabezal es aceptable, no olvides darle mantenimiento.

- Horno de flama

<b>Característica</b>	<b>Recomendación</b>
<i>Valores de O<sub>2</sub> bajo y CO<sub>2</sub> alto.</i>	Los valores indican que el aire introducido en el horno es insuficiente. Aumentar la compuerta de paso de aire del quemador.
<i>CO<sub>2</sub> bajo y O<sub>2</sub> alto.</i>	Existe exceso de aire. Se recomienda disminuir la apertura de la compuerta de paso de aire del quemador.
<i>CO<sub>2</sub> alto y O<sub>2</sub> alto.</i>	Mezcla de aire y combustible inadecuada. Se recomienda desmontar el inyector, llevar a cabo una limpieza o sustituirlo.
<i>Temperatura de la pared es mayor a la temperatura ambiente.</i>	Se recomienda sustituir el aislamiento.
<i>Horno tiene una antigüedad menor a 14 años.</i>	Se recomienda dar mantenimiento por lo menos cada año.

<i>Horno tiene antigüedad mayor o igual a 15 años.</i>	Se recomienda sustituir el horno ya que puede estar afectando la eficiencia energética.
<i>El factor de carga es mayor o igual a 90%.</i>	Muy bien, continuar con el funcionamiento del horno, no olvides darle mantenimiento.
<i>El factor de carga es menor a 80%.</i>	Dale mantenimiento a tu equipo puede que la eficiencia energética esté disminuyendo.
<i>Eficiencia del equipo es mayor al 80%.</i>	El horno tiene buena eficiencia. Continúa controlando los parámetros principales para tener mejores rendimientos.
<i>Eficiencia del equipo es menor al 80%.</i>	Mal desempeño del equipo, te recomendamos dar mantenimiento y corregir con las recomendaciones dadas.

### Representación del conocimiento

Una vez obtenidas las recomendaciones de cada equipo, se hizo la representación del conocimiento. Como se mostró en el capítulo anterior las representaciones que se seleccionaron son: Representación por medio de reglas y representación por medio de árboles de decisión.

Debido a la naturaleza de cada equipo el tipo de representación de conocimiento cambia, esto principalmente por la conexión entre los parámetros o características del equipo, si las características están relacionadas entre sí se optó por un árbol de decisión, sin embargo, si sus características son independientes unas a otras se optó por una representación basada en reglas.

- Caldera

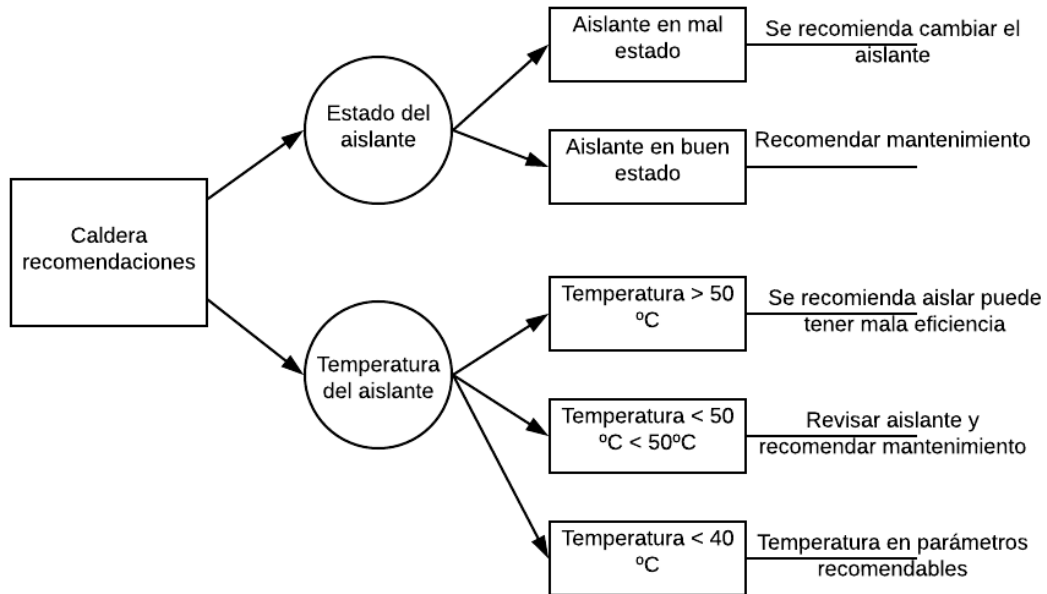


FIGURA 27. *Árbol de decisión: Caldera*

- Motor eléctrico

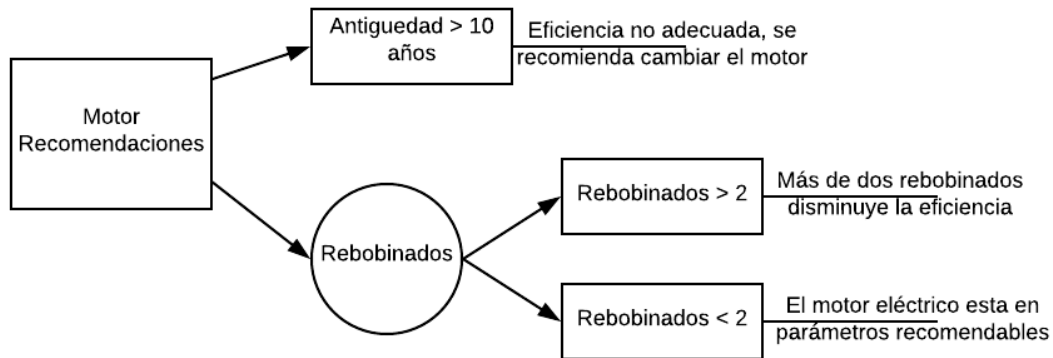


FIGURA 28. *Árbol de decisión: Motor eléctrico*

- Sistema de bombeo

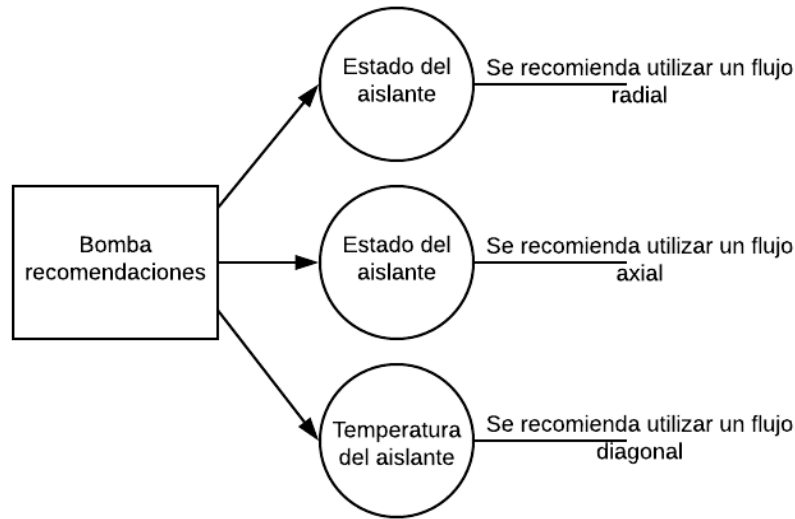


FIGURA 29. Árbol de decisión: Bomba

- Iluminación

REGLA 1 **IF** “Balastro de lámpara de tipo electromagnético” **THEN** “Sustituir lámparas por con balastro electrónico”.

REGLA 2 **IF** “Lámpara tipo incandescente” **THEN** “Reemplazar lámpara incandescente por lámpara LED o fluorescente compacta”.

REGLA 3 **IF** “Lámpara con bajo coeficiente de utilización” **THEN** “Reemplazar la luminaria”.

REGLA 4 **IF** “Color de techo es oscuro” **OR** “Color de piso es oscuro” **OR** “Color de pared es oscuro” **THEN** “Utilizar colores claros”.

REGLA 5 **IF** “Lámpara tipo fluorescente T12” **THEN** “Reemplazar lámpara por tecnología T8”.

REGLA 6 **IF** “Lámpara tipo vapor de mercurio” **THEN** “Reemplazar por lámpara de aditivos metálicos”.

REGLA 7 **IF** “Lámpara tipo fluorescente T5” **OR** “Lámpara tipo fluorescente T8” **THEN** “Reemplazar lámpara por lámpara LED”.

REGLA 8 **IF** “No cumple con norma NOM007” **AND** “Si cumple con norma NOM025” **THEN** “Redistribuir luminarias, cambiar lámparas con características similares, pero con menor potencia”.

REGLA 9 **IF** “Si cumple con norma NOM007” **AND** “No cumple con NOM025” **THEN** “Cambiar lámparas con la misma potencia, pero con diferentes características, también puedes cambiar el color de las paredes y el techo por colores claros”.

REGLA 10 **IF** “No cumple con norma NOM007” **AND** “No cumple con norma NOM025” **THEN** “Debes cambiar las lámparas y luminarias ya que esta afectando la eficiencia energética de tu empresa.”.

- Compresor

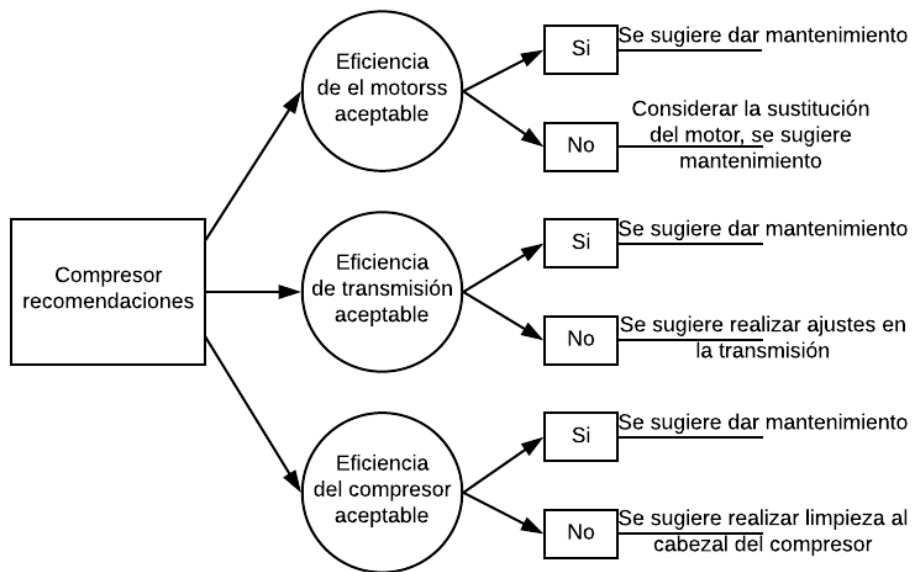


FIGURA 30. *Árbol de decisión: Compresor*

- Horno de flama

REGLA 1 **IF** “Valor O2 bajo” **AND** “Valor CO2 alto” **THEN** “Los valores indican que el aire introducido en el horno es insuficiente. Aumentar la compuerta de paso de aire del quemador”.

REGLA 2 **IF** “Valor O2 alto” **AND** “Valor CO2 bajo” **THEN** “Existe exceso de aire. Se recomienda disminuir la apertura de la compuerta de paso de aire del quemador”.

REGLA 3 **IF** “Valor O2 alto” **AND** “Valor CO2 alto” **THEN** “Mezcla de aire y combustible inadecuada. Se recomienda desmontar el inyector, llevar a cabo una limpieza o sustituirlo”.

REGLA 4 **IF** “Temperatura de la pared mayor a temperatura ambiente” **THEN** “Se recomienda sustituir el aislamiento”.

REGLA 5 **IF** “Antigüedad es menor a 14 años” **THEN** “Se recomienda dar mantenimiento por lo menos cada año.”

REGLA 6 **IF** “Antigüedad es mayor a 15 años ” **THEN** “Se recomienda sustituir el horno ya que puede estar afectando la eficiencia energética”.

REGLA 7 **IF** “Factor de carga mayor a 90%” **THEN** “Muy bien, continuar con el funcionamiento del horno, no olvides darle mantenimiento”.

REGLA 8 **IF** “Factor de carga menor a 80%” **THEN** “Dale mantenimiento a tu equipo puede que la eficiencia energética esté disminuyendo.”.

REGLA 9 **IF** “Eficiencia mayor a 80%” **THEN** “El horno tiene buena eficiencia. Continúa controlando los parámetros principales para tener mejores rendimientos”.

REGLA 10 **IF** “Eficiencia menor a 80%” **THEN** “Mal desempeño del equipo, te recomendamos dar mantenimiento y corregir con las recomendaciones dadas”.

### Codificación de máquina de inferencia

El objetivo de la máquina de inferencia es manejar el conocimiento obtenido y tomar la mejor decisión a partir de las reglas establecidas en los pasos anteriores. La codificación de la máquina de inferencia se hizo en el lenguaje C#.

Existen dos estrategias de razonamiento utilizadas en una máquina de inferencia, la primera es llamada encadenamiento hacia adelante o *botton-up* y la segunda es encadenamiento hacia atrás o *top-down*. En la siguiente lista se explica brevemente la descripción de cada una.

- Encadenamiento hacia adelante  
Toma un resultado (meta) y analiza si es que cumple alguna condición, si es que la cumple ejecuta una nueva hipótesis (nodo



de decisión o raíz). Se utiliza cuando se tiene las metas claras pero no la raíz o hipótesis.

- Encadenamiento hacia atrás

Toma una hipótesis y trata de analizar las condiciones que se cumplen para llegar a una meta. Se utiliza cuando se tiene la hipótesis clara pero no los resultados o metas.

La estrategia de razonamiento utilizada en el sistema experto es encadenamiento hacia atrás, la principal razón es porque se busca acciones, en este caso recomendaciones al usuario, a partir de ciertas condiciones e hipótesis de los equipos.

#### 4.9. Desarrollo de interfaz web

Para el desarrollo de la interfaz de usuario se utilizo el lenguaje descriptivo HTML 5 y C#. Se desarrolló una vista diferente para cada caso de uso mostrado en el subcapítulo 4.5. En la siguiente tabla se muestra una breve descripción de cada vista.

<b><i>Vista</i></b>	<b><i>Descripción</i></b>	<b><i>Muestra en apéndice</i></b>
<i>Inicio</i>	Muestra la información general del sistema tal como su funcionamiento, las ramas industriales con las que trabaja y las instituciones involucradas.	A.4.1
<i>Nosotros</i>	Da una breve introducción a las instituciones involucradas en el sistema.	A.4.2

<i>Mejores prácticas</i>	El objetivo de esta sección es mostrar recomendaciones generales para hacer el uso eficiente de energía.	A.4.3
<i>Casos de éxito</i>	La vista casos de éxito hace referencia a la experiencia de las empresas que utilicen el sistema.	A.4.4
<i>Contacto</i>	Muestra datos para el contacto de la institución administradora del sistema.	A.4.5
<i>Registro</i>	Muestra el formulario para el registro de un nuevo usuario.	A.4.6
<i>Inicio de sesión</i>	Muestra el formulario de para el ingreso al sistema, como identificador y contraseña.	A.4.7
<i>Sistema con Ingreso</i>	Es la primera vista que se muestra al usuario cuando ha ingresado al sistema.	A.4.8
<i>Encuesta de empresa</i>	El objetivo de esta vista es mostrar el formulario de las características generales de la empresa.	A.4.9
<i>Formulario de caldera</i>	Muestra el formulario de los datos pertenecientes a la caldera. Dependiendo de la Rama industrial esta vista puede o no aparecer.	A.4.10
<i>Diagnóstico de caldera</i>	Al terminar el formulario de caldera y guardar los datos,	A.4.11

		aparece esta vista. Aquí se muestra el diagnóstico de la caldera y sus respectivas recomendaciones.	
<i>Formulario de motor eléctrico</i>	<i>de</i>	Muestra el formulario de los datos pertenecientes al motor eléctrico. Dependiendo de la Rama industrial esta vista puede o no aparecer.	A.4.12
<i>Diagnóstico de motor eléctrico</i>	<i>de</i>	Al terminar el formulario del motor eléctrico y guardar los datos, aparece esta vista. Aquí se muestra el diagnóstico del motor y sus respectivas recomendaciones.	A.4.13
<i>Formulario de sistema de bombeo</i>	<i>de</i>	Muestra el formulario de los datos pertenecientes al sistema de bombeo. Dependiendo de la Rama industrial esta vista puede o no aparecer.	A.4.14
<i>Diagnóstico de sistema de bombeo</i>	<i>de</i>	Al terminar el formulario del sistema de bombeo y guardar los datos, aparece esta vista. Aquí se muestra el diagnóstico del sistema de bombeo y sus respectivas recomendaciones.	A.4.15
<i>Formulario de iluminación</i>	<i>de</i>	Muestra el formulario de los datos pertenecientes a la iluminación. Dependiendo de	A.4.16

		la Rama industrial esta vista puede o no aparecer.	
<i>Diagnóstico de iluminación</i>	<i>de</i>	Al terminar el formulario de iluminación y guardar los datos, aparece esta vista. Aquí se muestra el diagnóstico de la iluminación y sus respectivas recomendaciones.	A.4.17
<i>Formulario compresor</i>	<i>de</i>	Muestra el formulario de los datos pertenecientes al compresor. Dependiendo de la Rama industrial esta vista puede o no aparecer.	A.4.18
<i>Diagnóstico de compresor</i>	<i>de</i>	Al terminar el formulario del compresor y guardar los datos, aparece esta vista. Aquí se muestra el diagnóstico del compresor y sus respectivas recomendaciones.	A.4.19
<i>Formulario de horno de flama</i>	<i>de</i>	Muestra el formulario de los datos pertenecientes al horno de flama. Dependiendo de la Rama industrial esta vista puede o no aparecer.	A.4.20
<i>Diagnóstico de horno de flama</i>	<i>de</i>	Al terminar el formulario del horno de flama y guardar los datos, aparece esta vista. Aquí se muestra el diagnóstico del horno y sus respectivas recomendaciones.	A.4.21

#### **4.10. Pruebas de usuario**

##### Prueba de aceptación

Después del desarrollo de los algoritmos de análisis de cada equipo, se pidió a los usuarios expertos que validaran el Sistema asesor de la gestión energética. La validación del sistema consistió en ingresar datos reales de auditorías hechas en equipos en la industria y observar que los resultados obtenidos del sistema fueran similares que los análisis hechos por los expertos en la auditoría.

De la misma forma se mostró el diseño y la navegación por el sistema web a expertos y personas allegadas al tema de eficiencia energética y se tuvo una alimentación para la mejora de la interfaz del sistema.

##### Prueba de tolerancia a fallas

Esta prueba tiene como propósito identificar las fallas al hacer uso del sistema en situaciones anormales o inesperadas. El primer escenario consiste en ingresar al sistema información que rompe con la integridad de los datos en el sistema, lo que obtuvimos fue que el sistema no acepta estos datos ya que primero los evalúa antes de ingresarlos al área de análisis y almacenarlos en la BD.

##### Prueba de operación

El propósito de esta prueba es verificar el funcionamiento continuo del sistema en condiciones normales, los puntos que se toman en cuenta para la evaluación de esta prueba son: rapidez de respuesta, funcionalidades utilizadas al mismo tiempo y concurrencia. El sistema actualmente está alojado en un servidor perteneciente al Instituto de Ciencias Aplicadas y Tecnología (ICAT), el cual soporta hasta 20 usuarios, sin embargo se probó que su funcionamiento se hace más lento conforme a las tareas utilizadas y a los usuarios conectados, el problema se puede solucionar mejorando ciertos

parámetros del sistema y también se puede alojar el sistema en un servidor con más recursos.

### Evaluación del sistema experto

Para hacer la evaluación del sistema experto se hizo una serie de pruebas a los algoritmos y recomendaciones dadas por el sistema. Estos resultados fueron comparados por los cálculos y algoritmos realizados por los expertos para la evaluación de cada uno de los equipos industriales.

Los resultados obtenidos de estas pruebas son: Verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Cada uno de ellos son explicados en la siguiente lista.

- Verdaderos positivos (VP). Son clasificadas de tal manera a las recomendaciones que son acordes a la entrada A y que fueron arrojadas por el sistema.
- Verdaderos negativos (VN). Son clasificadas de tal manera a las recomendaciones no acordes a la entrada A y que no fueron arrojadas por el sistema.
- Falsos positivos (FP). Son clasificadas de tal manera a las recomendaciones no acordes a la entrada A y que fueron arrojadas por el sistema.
- Falsos negativos (FN). Son clasificadas de tal manera a la recomendaciones acordes a la entrada A y que no fueron arrojadas por el sistema.

Los parámetros a medir son: Exactitud, precisión y exhaustividad. Estos parámetros son descritos a continuación.

- Exactitud. La exactitud es la fracción de predicciones que el modelo realizó correctamente. Formalmente la exactitud tiene la siguiente definición.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$$

- Precisión. La precisión identifica la proporción de resultados positivos que fueron correctos. Su definición se muestra a continuación.

$$Precisión = \frac{VP}{VP + FP}$$

- Exhaustividad. La exhaustividad identifica la proporción de positivos reales que se identificaron correctamente. Su definición se muestra a continuación.

$$Exhaustividad = \frac{VP}{VP + FN}$$

Una vez identificados los parámetros se hicieron 10 secuencias en cada algoritmo y se identificaron los parámetros mencionados. Los resultados de las pruebas a cada algoritmo se muestra a continuación.

<i>Equipo industrial</i>	<i>Exactitud</i>	<i>Precisión</i>	<i>Exhaustividad</i>
<i>Caldera</i>	0.8	0.87	0.87
<i>Motor eléctrico</i>	0.9	0.87	1
<i>Sistema de bombeo</i>	0.9	0.88	1
<i>Iluminación</i>	0.9	1	0.9
<i>Compresor</i>	1	1	1
<i>Horno de flama</i>	0.9	1	0.88

## **5. CONCLUSIONES Y TRABAJO FUTURO**

El sistema desarrollado es una herramienta que ayuda a las PyMES en la gestión del consumo energético. El usuario puede tener los beneficios de una auditoría energética ya que el sistema esta basado en cada una de sus etapas.

Una de las principales ventajas es que el usuario puede obtener los beneficios energéticos en cualquier lugar y en el momento que lo desee ya que el sistema fue desarrollado como un sistema web.

Actualmente, se cuenta con los algoritmos de seis equipos utilizados en ocho ramas industriales, los cuales fueron obtenidos en la fase de adquisición del conocimiento y otorgados por expertos en el área energética.

La adquisición del conocimiento es un proceso largo y el ingeniero del conocimiento necesita que el experto en el tema pueda brindarle la información necesaria para la codificación de los algoritmos, por esta razón es de suma importancia la elección del experto que brindará los datos ya que se necesita una constante comunicación de ambas partes.

Cabe recalcar que los resultados brindados por el sistema al usuario son una imagen de los datos brindados por el experto, es decir, la experiencia y la experticia del experto en el tema están reflejados en el algoritmo de análisis. El sistema no tendrá contemplado ambientes y casos de uso que el experto no halla tomado en



cuenta para la realización de los algoritmos codificados por el ingeniero del conocimiento.

El sistema experto destinado a recomendaciones garantiza que el usuario identifique los parámetros que están fuera de los rangos operativos y así podrá tomar medidas de monitoreo y precaución en el manejo del equipo.

Para la validación del conocimiento obtenido y codificado es muy importante la constante comunicación con los expertos y con las personas allegadas al área energética, para tener contante alimentación de posibles mejoras en los algoritmos y en la parte visual del sistema web.

Los diagnósticos fueron probados con los expertos en el área energética para asegurar que los resultados obtenidos son acordes a los hechos en una auditoría energética, también se hicieron pruebas de escritorio para poder asegurar que los resultados dados por el sistema fueran acordes a los cálculos hechos de una forma manual, es decir, sin ayuda de un sistema computacional tal y como son mostrados en la matriz del capítulo 4.10.

### Trabajo futuro

El sistema asesor de la gestión energética se encuentra en la primera etapa de desarrollo, es decir, se contempla por parte de los expertos y del equipo de desarrollo en añadir más equipos industriales, de igual manera se piensa agregar más ramas industriales una vez que se investigue los procesos y equipos industriales relacionados con cada una.

Se contempla agregar más recomendaciones e información de buenas prácticas al sistema experto. Con esto se asegura que el usuario tenga la mayor información sobre el estado de su equipo y de las áreas de oportunidad a realizar por el usuario.

Cabe resaltar que al agregar nueva información de los equipos, ramas industriales y recomendaciones se contempla expandir la base de datos, por lo tanto el servidor en el que este alojado deberá tener los recursos necesarios para el sistema.

Como trabajo futuro se contempla hacer un diagnóstico general de tipo económico para que el usuario obtenga una mejor noción del ahorro energético y del dinero que puede ahorrar al poner en marcha las recomendaciones dadas.

## 6. GLOSARIO Y ACRÓNIMOS

- **BD** – Base de Datos
- **CBR** - Case Based Reasoning
- **ER** – Modelo Entidad-Relación
- **FN** – Falso negativo
- **FP** – Falso positivo
- **HTML 5** – Lenguaje descriptivo utilizado para la realización de sistemas web
- **Licencia BSD** – Licencia otorgada a proyectos de software libre para que cualquier usuario pueda utilizar dicho software.
- **MVC** – Modelo-Vista-Controlador
- **Open source** – Software Libre
- **PIB** - Producto Interno Bruto.
- **PyMES** - Pequeñas y medianas empresas.
- **RBR** - Rule Based Reasoning
- **VN** – Verdadero negativo
- **VP** – Verdadero positivo

## **A. APÉNDICES**

# A.1. Apéndice 1. Formatos de levantamiento

## A.1.1. Caldera

			
<b>FORMATO DE LEVANTAMIENTO DE GENERADOR DE CALOR</b>			
		Ubicación: _____	
		Fecha: _____	
Empresa: _____			
Rama industrial: _____	ID Equipo: <u>Cal01</u>		
<b>DATOS DE PLACA DEL GENERADOR DE CALOR</b>			
Marca: _____	Año de fabricación: _____		
Modelo: _____	Potencia termica: _____	[CC]	
Tipo: _____	Presión de diseño: _____	[Kg/cm <sup>2</sup> ]	
Serie: _____	Presión de operación: _____	[Kg/cm <sup>2</sup> ]	
Área de transferencia: _____	[m <sup>2</sup> ]	Ciclo de operación: _____	hrs/día
		_____	días/año
		Eficiencia de diseño: _____	
<b>TIPO DE GENERADOR DE CALOR</b> _____			
Sistema de control de operación del generador: _____			
<b>AGUA DE ALIMENTACIÓN</b>			
El agua es tratada? _____	Otro _____		
<b>COMBUSTIBLE</b>			
Combustibles disponibles _____			
Combustible principal: _____			
Combustible auxiliar: _____			
Costo de combustible: _____			
Gasto total del combustible: _____			
Gasto de combustible del generador de calor: _____			
Tipo de control de combustión: _____			
Forma de almacenamiento del combustible: _____	Período de suministro: _____		
¿Tienen tanque de día?: _____			
Potencia de la bomba de alimentación del combustible: _____			
¿Se precalienta el combustible? _____			
<b>GASES DE COMBUSTIÓN</b>			
Tipo de tiro: _____	Potencia del ventilador de tiro: _____		
Cuenta con filtro: _____			
Relación de aire combustible de operación: _____			
<b>TRAMPAS DE VAPOR</b>			
Existen trampas de vapor: _____			
<b>MANTENIMIENTO</b>			
Existe programa de mantenimiento: _____			
Antigüedad de operación: _____	Cuenta con bitácora: _____		
	Color de la flama: _____		
	Estado de la casa de máquinas: _____		
	Código de colores: _____		
Observaciones: _____			
Foto:	<div style="border: 1px solid black; width: 100%; height: 100%;"></div>		

Aut  
Automático en  
Se


## A.1.2. Motor Eléctrico

EVALUACIÓN ENERGÉTICA DE MOTORES ELÉCTRICOS TRIFÁSICOS			
Favor de ingresarlos datos solicitados.			
<b>DATOS DE LA EMPRESA</b>			
NOMBRE DE LA EMPRESA RAMA INDUSTRIAL	PYME UNAM		
	MANUFACTURA		
<b>DATOS DE PLACA DEL MOTOR</b>			
Capacidad	Estándar	HP	RPM FUERA DE RANGO. FAVOR DE INGRESAR UN VALOR ENTRE ENTRE 700 Y 3600 RPM
Tensión		V	
Velocidad		RPM	
Eficiencia		%	
Tipo de eficiencia		Hz	
Frecuencia		p.u	
Factor de potencia			
<b>DATOS DE OPERACIÓN</b>			
Años de operación	Menos de 10 años		
Número de rebobinados	0		
Horas de operación al día	6		
Días de operación al año	300		
Costo medio kWh	\$	2.20	
<b>DATOS DE MEDICIÓN</b>			
Tensión L1 - L2	400.00	V	
Tensión L2 - L3	400.00	V	
Tensión L3 - L1	400.00	V	
Corriente L1	1.00	A	#¡DIV/0!
Corriente L2	1.00	A	#¡DIV/0!
Corriente L3	5.00	A	#¡DIV/0!
<b>RESULTADOS Y PROPUESTAS</b>			
Eficiencia calculada	#¡REF!	%	
Diferencia con respecto a la eficiencia nominal	#¡REF!	%	
Factor de carga	#N/A	%	
Potencia de salida	#N/A	kW	
Potencia de entrada	#N/A	kW	
Pérdidas	#N/A	kW	
Energía de operación	#N/A	kWh	
Energía de pérdidas	#N/A	kWh	
Costo de operación	#N/A	anual	
Costo de pérdidas	#N/A	anual	
<b>Factor que decrementa la eficiencia</b>		<b>Resultados de la operación le motor</b>	
Diferencia de tensión	El valor de tensión con el que opera su motor es aceptable		
Desbalance de tensión	El desbalance de tensión que presenta su motor es aceptable		
Antigüedad	¡Su motor no presenta pérdidas por antigüedad! Se le recomienda que conserve un adecuado mantenimiento		
Rebobinados	Su motor no presenta pérdidas por rebobinados, se le recomienda que conserve un adecuado mantenimiento		
Desbalance de corriente	¡La eficiencia de su motor está disminuyendo! Se recomienda revisar las terminales o uniones del conductor eléctrico del motor (ya que puede haber un falso contacto), así como revisar que el calibre del conductor sea el adecuado		
Frecuencia	La frecuencia de operación de su motor es la adecuada, se le recomienda que conserve un adecuado mantenimiento		
Factor de carga	#N/A		

### A.1.3. Sistema de bombeo

EVALUACIÓN ENERGÉTICA DE SISTEMAS DE BOMBEO					
DATOS DE LA EMPRESA					
Empresa					
ID Equipo					
Lugar					
DATOS DEL SISTEMA					
Aplicación	Procesos industriales (distribución a equipos industriales...)				
¿Conoce el flujo de operación?	<input type="radio"/> Sí <input checked="" type="radio"/> No				
Diametro de la conexión de entrada que utiliza el fluido					[in]
Criterio para Cálculo de fricción	Criterio de presión en la succión				
Fluido bombeado					
Material de tub					
DATOS DE PLACA DE LA BOMBA					
Carga max ( $H_{max}$ )			Gasto max ( $Q_{max}$ )		
Carga op ( $H_{op}$ )			Gasto min ( $Q_{min}$ )		
DATOS DE OPERACIÓN ELÉCTRICA					
Potencia del motor			Corriente medida	[A]	
Eficiencia motor	[%]		Tensión medida	[V]	
Costo de la energía	[\$/kWh]		Factor de potencia	[%]	
DATOS DE OPERACIÓN HIDRÁULICA					
Presión op ( $P_{op}$ )			Horas de operación al día	[h/día]	
Distancia Vertical al usuario final			Días de operación al año	[día/año]	
Distancia Horizontal al usuario final					
RESULTADOS					
Carga (H)	#N/A	[m]	$\eta$ conjunto bba-mot (calculada)	#N/A	[%]
Gasto (Q)	#N/A	[m <sup>3</sup> /s]	$\eta$ bomba (estimada)	#N/A	[%]
Potencia hid	#N/A	[kW]	Demanda eléctrica	0	[kWh/año]
				0	[MJ/año]
CNPS disponible	#N/A	[m]		0	[\$/año]
			Energía aprovechada por el sistema	#N/A	[kWh/año]
GRÁFICA					
<b>Demanda eléctrica</b>					
<ul style="list-style-type: none"> <li><span style="color: blue;">■</span> Energía aprovechada por el sistema</li> <li><span style="color: blue;">■</span> Energía perdida por ineficiencias</li> </ul>					
RECOMENDACIONES					
Recomendaciones de mantenimiento: Precauciones de cavitación: Recomendaciones para las tuberías: Recomendaciones de bomba:					

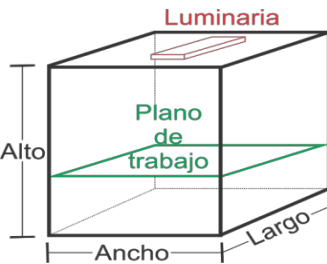
## A.1.4. Iluminación



**EVALUACIÓN DEL SISTEMA DE ALUMBRADO INTERIOR**

**Nombre del área de trabajo:** \_\_\_\_\_ **N° de área:** 02

**Tipo de área de trabajo:** Áreas de ventas



**CARACTERÍSTICAS DEL ÁREA DE TRABAJO**

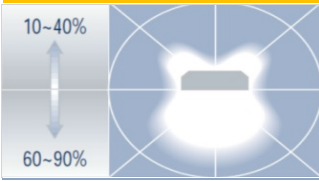
Nivel mínimo de iluminación, NOM-025-STPS-2008: 300 [lx]  
 Nivel máximo de DPEA, NOM-007-ENER-2014: 18.08 [W/m2]

Largo: \_\_\_\_\_ [m]      Techo: Muy claro  
 Ancho: \_\_\_\_\_ [m]      Pared: Claro  
 Alto: \_\_\_\_\_ [m]      Piso: Claro

Altura de plano de trabajo: \_\_\_\_\_ [m]  
 Altura de montaje de luminaria: \_\_\_\_\_ [m]


Horas de operación por día: \_\_\_\_\_ [Horas]  
 Días de operación al año: \_\_\_\_\_ [Días]

**Tipo de iluminación:** Semi Directa



**CARACTERÍSTICAS DE ILUMINACIÓN**

Total de luminarias: \_\_\_\_\_ **N° Luminaria:** 01



**Tipo de Lámpara:**

Lámparas por luminaria: \_\_\_\_\_  
 Lúmenes por lámpara: \_\_\_\_\_ [lm]  
 Potencia de lámpara: \_\_\_\_\_ [W]

**Nivel de mantenimiento:**

**Sustituto para:**

**SUSTITUCIÓN DE TECNOLOGÍA**

Potencia de lámpara:  [W]  
 Lúmenes por lámpara:  [lm]

N° de Lámparas necesarias: Faltan Datos 300 [lx]  
 Lámparas por luminaria: Faltan Datos  
 Total de luminarias: Faltan Datos

NUEVO FORMATO

RESULTADOS

NUEVA ÁREA

AGREGAR LUMINARIA



## A.1.5. Compresor



DATOS DE PYME			
Empresa			
ID Equipo			
Lugar	Buenavista, CDMX		
DATOS DE ENTRADA USUARIO			
LADO DEL SUMINISTRO			
		Presión máxima	
		Flujo o caudal del aire	
		Potencia del compresor	
Datos de placa			
	Tipo de compresor	Pistones	Presión de operación MAX. 11 [Bar], potencia máxima 25 hp
	Tipo de transmisión	Banda y poleas	-
Pot	Potencia nominal del compresor	10.00	[HP]
P <sub>max</sub>	Presión máxima del compresor		[Psi]
V <sub>placa</sub>	Flujo o caudal nominal		[ft <sup>3</sup> /min]
Datos de operación			
P <sub>ope</sub>	Presión de operación del compresor		[Psi]
T <sub>1</sub>	Temperatura ambiente		[°C]
¿Se conoce la temperatura real de aire de salida del compresor? (si se tiene un termómetro a la salida de aire, de lo contrario dejar en blanco)		No	
Horas de operación al día del motor			[hrs/día]
Días de operación del motor al año			[días]
Costo ponderado de la energía			[\$/kWh]
Parámetros de operación y mantenimiento			
Tipo de enfriamiento		Aire (Ventilador integrado)	
Frecuencia de cambio de aceite		3 - 6 meses	
Frecuencia de limpieza de carcasa y filtros		6 - 12 meses	
Frecuencia de ajuste de transmisión		No se ha realizado/no se tiene información	
Ubicación del compresor		Cuarto de compresores con ventilación	
Años de operación del compresor		15 años o más	[años]
Flujo o caudal de operación	<i>Sustituir por el valor de caudal de operación si es conocido por el usuario, de lo contrario dejar en blanco. (La hoja de cálculo provee un valor de flujo de operación estimado)</i>		[ft <sup>3</sup> /min]
Método de detección de fugas			
Preparar una solución simple de agua jabonosa. Aplicar la solución con una brocha en uniones y válvulas. Observar en busca de puntos donde se formen burbujas, las burbujas indican fugas de aire. Marcar y contabilizar las fugas.			
Número de fugas contabilizadas en el sistema			
Datos de motor			
η <sub>motor</sub>	Eficiencia nominal del motor (De placa)	88.00	[%]
<i>Se debe llenar previamente el formato de motores</i>			
η <sub>motor</sub>	Eficiencia real del motor	90.00	[%]
P <sub>dem</sub>	Potencia eléctrica demandada	7.40	[kW]
RESULTADOS			
η <sub>compresión</sub>	Eficiencia del compresor	# DIV/0!	[%]
η <sub>transmisión</sub>	Eficiencia de transmisión	80.00	[%]
η <sub>total</sub>	Eficiencia del conjunto motor-compresor	# DIV/0!	[%]

### Distribución de la energía suministrada al compresor

Componente	Porcentaje
Motor	36%
Transmisión	64%
Potencia disponible	0%
Cabezal de compresión	0%

### Eficiencias del compresor

Componente	Tipología	Valor
Motor	Estimadas	0.90
	Nominales	0.88
Transmisión	Estimadas	0.80
	Nominales	0.95
Cabezal de compresión	Estimadas	0.00
	Nominales	0.84

## A.1.6. Horno de flama

Evaluación energética de hornos de flama		
Nombre de la empresa		
Rama industrial		
Fecha		
Localidad		
Datos del horno y servicio		
Clave del equipo		Unidad de producto <input type="text" value="Másica"/>
Tipo	<input type="text" value="Convectivo"/>	Producto <input type="text" value="Hierro"/>
Año de instalación		Combustible <input type="text" value="Gas LP"/>
Aplicación o proceso		
Datos requeridos: Método Indirecto		
Designación	Unidades	Valor
Consumo combustible	m <sup>3</sup> /mes	20,231
Tiempo de operación	horas	22
Días de operación	días/mes	20
Temperatura gases de combustión	°C	395
Temperatura gases salida chimenea	°C	150
Temperatura ambiente	°C	25
Temperatura aire de entrada	°C	43
Temperatura inicial del producto	°C	32.5
Temperatura final del producto	°C	820
Temperatura interior del horno		
Masa del producto horneado por día	kg/día	23,760
Análisis de gases de combustión		
O <sub>2</sub>	%	13.9
CO <sub>2</sub>	%	4
CO	ppm	2718
N <sub>2</sub>	%	81.83
Exceso de aire	%	205.64
Eficiencia energética del horno		
Eficiencia del horno	%	40.62

La eficiencia del horno es:

40.62

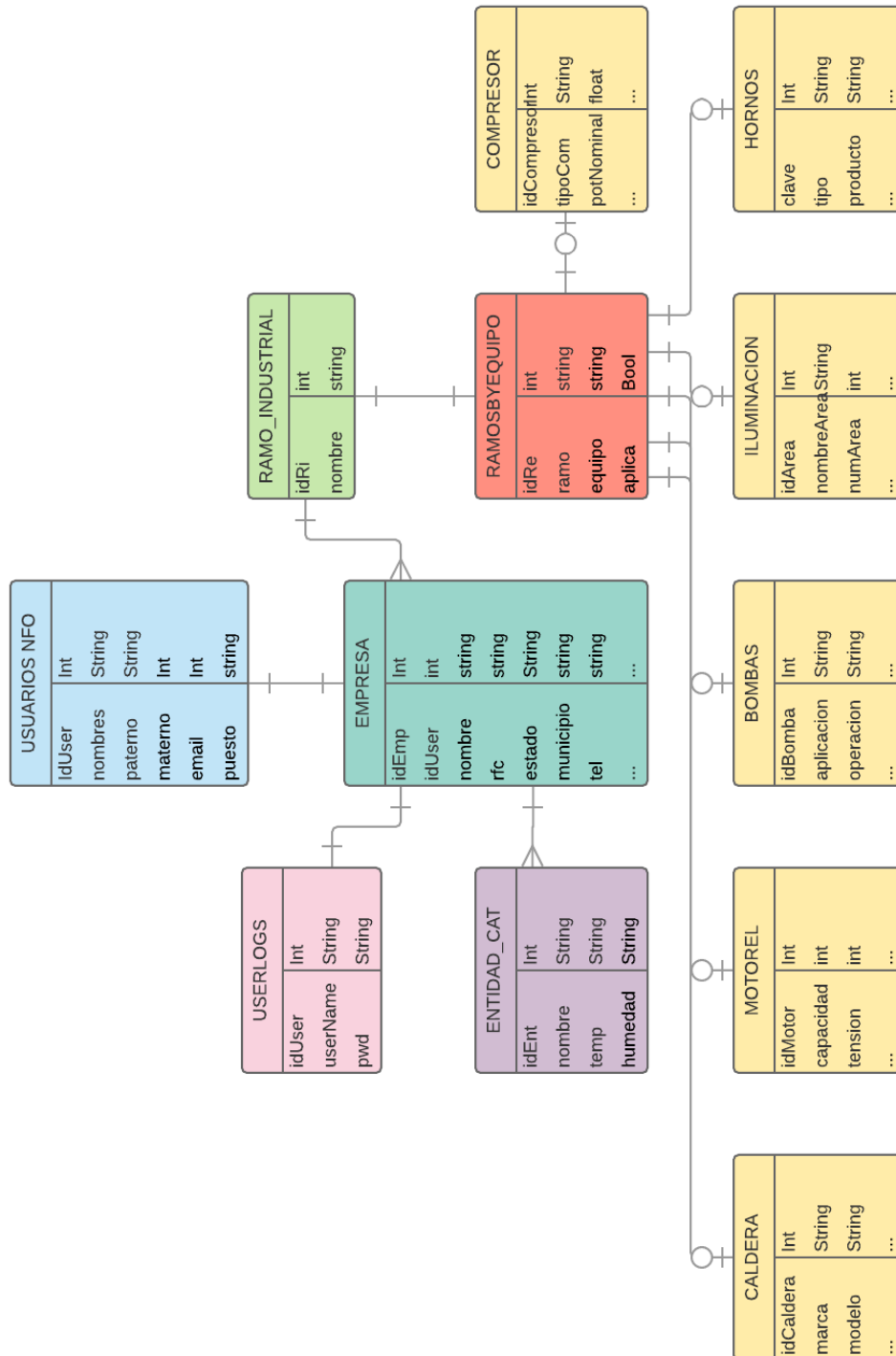
Buenas prácticas		
<p><b>¿Se tiene control de la calidad de materia que entra al horno?</b></p> <p>La calidad con la que se produce aumenta, disminuyendo el consumo de combustible y optimizando el proceso.</p>	SI	NO
	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<p><b>¿Se tiene bitácora de mantenimiento y operación?</b></p> <p>Es importante programar el mantenimiento preventivo cada 6 meses para evitar paros imprevistos.</p>	SI	NO
	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<p><b>¿Existen fugas de calor?</b></p> <p>Revisar periódicamente las condiciones de los aislantes, para evitar desgaste del equipo por sobrecalentamiento. Actualmente un buen aislante aumentará de 2% a 3% la eficiencia de nuestro equipo</p>	SI	NO
	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<p><b>¿Se registran altas temperaturas en las paredes exteriores?</b></p> <p>Posiblemente tenga una falla en los aislamientos, sellos y/o refractarios. Una termografía puede mostrarnos las variaciones de temperatura en la zona. Por lo cual se recomienda cambiarlos.</p>	SI	NO
	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Mantenimiento
<ul style="list-style-type: none"> <li>• Se debe de realizar un mantenimiento periódico de acuerdo a lo establecido por el proveedor.</li> <li>• Revisar cada 3 meses la varilla del quemador, es importante que se encuentre trabajando de forma correcta para mantener la temperatura deseada.</li> <li>• Es importante observar la presencia de suciedad o cascarrillas.</li> <li>• Se debe observar la uniformidad del color o existencia de puntos calientes, así como deformaciones.</li> <li>• Se debe verificar el estado general del aislamiento interior (refractario, manta cerámica, etc.) y exterior (principalmente en acometidas al horno).</li> <li>• Los quemadores deben conservarse limpios y en buen estado. Todas las partes ajustables deben poder moverse libremente.</li> </ul>

## A.2. Apéndice 2. Base de Datos

### A.2.1. Diagrama general de Base de datos



## A.3. Apéndice 3. Algoritmos de análisis energético por equipo

### A.3.1. Código de Caldera

```
namespace sitioSener.Controllers
{
    public class CalderaController : Controller
    {
        // GET: Caldera
        [HttpGet]
        public ActionResult Caldera()
        {
            Session["UsuarioNombre"] = "Prueba";
            Session["encuesta"] = "true";
            Session["editarPerfil"] = "true";
            Session["inicioACT"] = "noCurrent";
            Session["empresaACT"] = "current";

            List<Unidades> liUnidadAT = new List<Unidades>();

            liUnidadAT.Add(new Unidades ()
            {
                idUnidad = "metros",
                nombre = "m^2"
            });

            liUnidadAT.Add(new Unidades ()
            {
                idUnidad = "pies",
                nombre = "ft^2"
            });

            ViewBag.unidadAT = liUnidadAT;

            List<Unidades> liUnidadPT = new List<Unidades>();

            liUnidadPT.Add(new Unidades ()
            {
                idUnidad = "hp",
                nombre = "HP"
            });

            liUnidadPT.Add(new Unidades ()
            {
                idUnidad = "kw",
                nombre = "KW"
            });

            ViewBag.unidadPT = liUnidadPT;

            List<Catalogo> liCatTipo = new List<Catalogo>();

            liCatTipo.Add(new Catalogo () { idCat = "Piro tubular", nombre = "Piro tubular" });
            liCatTipo.Add(new Catalogo () { idCat = "Acuatubular", nombre = "Acuatubular" });

            ViewBag.catTipo = liCatTipo;

            return View();
        }

        [HttpPost]
        public ActionResult Caldera(CalderaPlaca model)
        {
            if (ModelState.IsValid)
            {
                return RedirectToAction("CalderaComb", "Caldera");
            }

            List<Unidades> liUnidadAT = new List<Unidades>();

            liUnidadAT.Add(new Unidades ()
```

```

    {
        idUnidad = "metros",
        nombre = "m^2"
    });

    liUnidadAT.Add(new Unidades ()
    {
        idUnidad = "pies",
        nombre = "ft^2"
    });

    ViewBag.unidadAT = liUnidadAT;

    List<Unidades> liUnidadPT = new List<Unidades>();

    liUnidadPT.Add(new Unidades ()
    {
        idUnidad = "hp",
        nombre = "HP"
    });

    liUnidadPT.Add(new Unidades ()
    {
        idUnidad = "kw",
        nombre = "KW"
    });

    ViewBag.unidadPT = liUnidadPT;

    List<Catalogo> liCatTipo = new List<Catalogo>();

    liCatTipo.Add(new Catalogo() { idCat = "Piro tubular", nombre = "Piro tubular" });
    liCatTipo.Add(new Catalogo() { idCat = "Acuatubular", nombre = "Acuatubular" });

    ViewBag.catTipo = liCatTipo;

    return View(model);
}

public ActionResult CalderaComb()
{
    Session["UsuarioNombre"] = "Prueba";
    Session["encuesta"] = "true";
    Session["editarPerfil"] = "true";
    Session["inicioACT"] = "noCurrent";
    Session["empresaACT"] = "current";

    List<Unidades> liUnidadCS = new List<Unidades>();

    liUnidadCS.Add(new Unidades ()
    {
        idUnidad = "kg hora",
        nombre = "kg/hr"
    });

    liUnidadCS.Add(new Unidades ()
    {
        idUnidad = "litro hora",
        nombre = "L/hr"
    });

    ViewBag.unidadCS = liUnidadCS;

    List<Unidades> liUnidadTM = new List<Unidades>();

    liUnidadTM.Add(new Unidades ()
    {
        idUnidad = "C",
        nombre = "°C"
    });

    liUnidadTM.Add(new Unidades ()

```

```

    {
        idUnidad = "F",
        nombre = "°F"
    });

ViewBag.unidadTM = liUnidadTM;

List<Unidades> liUnidadPR = new List<Unidades>();

liUnidadPR.Add(new Unidades() { idUnidad = "Psatm", nombre = "Psatm" });
liUnidadPR.Add(new Unidades() { idUnidad = "bar", nombre = "bar" });
liUnidadPR.Add(new Unidades() { idUnidad = "pascal", nombre = "Kpa" });
liUnidadPR.Add(new Unidades() { idUnidad = "kg por centimetro", nombre = "kg/cm3" });

ViewBag.unidadPR = liUnidadPR;

return View();
}

[HttpPost]
public ActionResult CalderaComb(Combustible model)
{
    if (ModelState.IsValid)
    {
        return RedirectToAction("CalderaMed", "Caldera");
    }

List<Unidades> liUnidadCS = new List<Unidades>();

liUnidadCS.Add(new Unidades() { idUnidad = "kg hora", nombre = "kg/hr" });
liUnidadCS.Add(new Unidades() { idUnidad = "litro hora", nombre = "L/hr" });

ViewBag.unidadCS = liUnidadCS;

List<Unidades> liUnidadTM = new List<Unidades>();

liUnidadTM.Add(new Unidades() { idUnidad = "C", nombre = "°C" });
liUnidadTM.Add(new Unidades() { idUnidad = "F", nombre = "°F" });

ViewBag.unidadTM = liUnidadTM;

List<Unidades> liUnidadPR = new List<Unidades>();

liUnidadPR.Add(new Unidades() { idUnidad = "Psatm", nombre = "Psatm" });
liUnidadPR.Add(new Unidades() { idUnidad = "bar", nombre = "bar" });
liUnidadPR.Add(new Unidades() { idUnidad = "pascal", nombre = "Kpa" });
liUnidadPR.Add(new Unidades() { idUnidad = "kg por centimetro", nombre = "kg/cm3" });

ViewBag.unidadPR = liUnidadPR;

return View(model);
}

public ActionResult CalderaMed()
{
    Session["UsuarioNombre"] = "Prueba";
    Session["encuesta"] = "true";
    Session["editarPerfil"] = "true";
    Session["inicioACT"] = "noCurrent";
    Session["empresaACT"] = "current";

List<Unidades> liUnidadTM = new List<Unidades>();

liUnidadTM.Add(new Unidades() { idUnidad = "C", nombre = "°C" });
liUnidadTM.Add(new Unidades() { idUnidad = "F", nombre = "°F" });

ViewBag.unidadTM = liUnidadTM;

List<Unidades> liUnidadPR = new List<Unidades>();

liUnidadPR.Add(new Unidades() { idUnidad = "Psatm", nombre = "Psatm" });

```

```

    liUnidadPR.Add(new Unidades() { idUnidad = "bar", nombre = "bar" });
    liUnidadPR.Add(new Unidades() { idUnidad = "pascal", nombre = "Kpa" });
    liUnidadPR.Add(new Unidades() { idUnidad = "kg por centimetro", nombre = "kg/cm3" });

    ViewBag.unidadPR = liUnidadPR;

    List<Unidades> liUnidadFL = new List<Unidades>();

    liUnidadFL.Add(new Unidades() { idUnidad = "kg por s", nombre = "kg/s" });
    liUnidadFL.Add(new Unidades() { idUnidad = "kg por h", nombre = "kg/hr" });
    liUnidadFL.Add(new Unidades() { idUnidad = "galon por min", nombre = "gal/min" });

    ViewBag.unidadFL = liUnidadFL;

    List<Apariencia> liEstadoAS = new List<Apariencia>();

    liEstadoAS.Add(new Apariencia() { idStatus = "buena", nombre = "Buen Estado" });
    liEstadoAS.Add(new Apariencia() { idStatus = "mala", nombre = "Mal Estado" });

    ViewBag.estadoAS = liEstadoAS;

    return View();
}

[HttpPost]
public ActionResult CalderaMed(MedicionCaldera model)
{
    if (ModelState.IsValid)
    {
        return RedirectToAction("CalderaDiag", "Caldera");
    }

    List<Unidades> liUnidadTM = new List<Unidades>();

    liUnidadTM.Add(new Unidades() { idUnidad = "C", nombre = "°C" });
    liUnidadTM.Add(new Unidades() { idUnidad = "F", nombre = "°F" });

    ViewBag.unidadTM = liUnidadTM;

    List<Unidades> liUnidadPR = new List<Unidades>();

    liUnidadPR.Add(new Unidades() { idUnidad = "Psatm", nombre = "Psatm" });
    liUnidadPR.Add(new Unidades() { idUnidad = "bar", nombre = "bar" });
    liUnidadPR.Add(new Unidades() { idUnidad = "pascal", nombre = "Kpa" });
    liUnidadPR.Add(new Unidades() { idUnidad = "kg por centimetro", nombre = "kg/cm3" });

    ViewBag.unidadPR = liUnidadPR;

    List<Unidades> liUnidadFL = new List<Unidades>();

    liUnidadFL.Add(new Unidades() { idUnidad = "kg por s", nombre = "kg/s" });
    liUnidadFL.Add(new Unidades() { idUnidad = "kg por h", nombre = "kg/hr" });
    liUnidadFL.Add(new Unidades() { idUnidad = "galon por min", nombre = "gal/min" });

    ViewBag.unidadFL = liUnidadFL;

    List<Apariencia> liEstadoAS = new List<Apariencia>();

    liEstadoAS.Add(new Apariencia() { idStatus = "buena", nombre = "Buen Estado" });
    liEstadoAS.Add(new Apariencia() { idStatus = "mala", nombre = "Mal Estado" });

    ViewBag.estadoAS = liEstadoAS;

    return View(model);
}

public ActionResult CalderaDiag()
{
    Session["UsuarioNombre"] = "Prueba";
    Session["encuesta"] = "true";
    Session["editarPerfil"] = "true";
}

```

```

        Session["inicioACT"] = "noCurrent";
        Session["empresaACT"] = "current";

        ViewData["Eficiencia"] = model.diag;

        return View();
    }
}

```

### A.3.2. Motor Eléctrico

```

namespace sitioSener.Controllers
{
    public class MotoresElController : Controller
    {
        // GET: MotoresEl
        public ActionResult Motores()
        {
            Session["UsuarioNombre"] = "Prueba";<
            Session["encuesta"] = "true";
            Session["editarPerfil"] = "true";
            Session["inicioACT"] = "noCurrent";
            Session["empresaACT"] = "current";

            List<Unidades> liUnidadPT = new List<Unidades>();
            liUnidadPT.Add(new Unidades() { idUnidad = "Watts", nombre = "Watts" });
            liUnidadPT.Add(new Unidades() { idUnidad = "HP", nombre = "HP" });
            ViewBag.unidadPT = liUnidadPT;

            List<Tipos> liTipo = new List<Tipos>();
            liTipo.Add(new Tipos() { idTipo = "STD", nombre = "Estandard" });
            liTipo.Add(new Tipos() { idTipo = "HI", nombre = "Alta" });
            ViewBag.tiposMot = liTipo;

            return View();
        }

        // POST: Motores
        [HttpPost]
        public ActionResult Motores(MotoresDatosIn model)
        {
            if (ModelState.IsValid)
            {
                Random r = new Random();
                int r1 = r.Next(0, 9);
                int r2 = r.Next(0, 9);
                int r3 = r.Next(0, 9);
                int l = r.Next(0, 25);
                string letra = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

                string key = "MTE" + r1 + r2 + r3 + Session["UsuarioNombre"].ToString() +
System.DateTime.Now.ToString("yyMMdd") + letra[l];

                if(model.unidadPT == "Watts")
                {
                    model.capacidad = model.capacidad * Convert.ToDecimal(0.00134102);
                }

                string query = "INSERT INTO snr.tmp_motoresEl (idtemp, capacidad, tension, corriente,
revoluciones, fp, eficiencia," +
                    "tipo, horas, reparaciones, dias, antiguedad, costo) VALUES('" + key + "','" +
model.capacidad + "','" + model.tension + "','" + model.corriente + "','" + model.revoluciones + "','" +
+ model.factorPotencia + "','" + model.eficiencia + "','" + model.tipo + "','" + model.horas + "','" +
model.revoluciones + "','" + model.dias + "','" + model.antiguedad + "','" + model.costo + "')";
                ejecutar(query);
                return RedirectToAction("MotoresMed", "MotoresEl", new { keyIn = key });
            }

            List<Unidades> liUnidadPT = new List<Unidades>();
            liUnidadPT.Add(new Unidades() { idUnidad = "Watts", nombre = "Watts" });

```



```

    liUnidadPT.Add(new Unidades() { idUnidad = "HP", nombre = "HP" });
    ViewBag.unidadPT = liUnidadPT;

    List<Tipos> liTipo = new List<Tipos>();
    liTipo.Add(new Tipos() { idTipo = "STD", nombre = "Estandard" });
    liTipo.Add(new Tipos() { idTipo = "HI", nombre = "Alta" });
    ViewBag.tiposMot = liTipo;

    return View(model);
} // termina post motores

// GET: MotoresMed
public ActionResult MotoresMed(string keyIn)
{
    Session["UsuarioNombre"] = "Prueba";
    Session["encuesta"] = "true";
    Session["editarPerfil"] = "true";
    Session["inicioACT"] = "noCurrent";
    Session["empresaACT"] = "current";

    ViewBag.keyMTE1 = keyIn;

    return View();
}

// POST: Motores
[HttpPost]
public ActionResult MotoresMed(MedicionMotores m, string key)
{
    Session["UsuarioNombre"] = "Prueba";
    Session["encuesta"] = "true";
    Session["editarPerfil"] = "true";
    Session["inicioACT"] = "noCurrent";
    Session["empresaACT"] = "current";

    if (ModelState.IsValid)
    {
        string auxQuery = "";
        string query = "SELECT idtemp, capacidad, tension, corriente, revoluciones, fp,
eficiencia," +
            "tipo, horas, reparaciones, dias, antiguedad, costo FROM snr.tmp_motoresEl WHERE
idtemp = '" + key + "'";
        DataTable dtDatos = datos(query);

        decimal n = Convert.ToDecimal(dtDatos.Rows[0][6].ToString()) / Convert.ToDecimal(100),
hp = Convert.ToDecimal(dtDatos.Rows[0][1].ToString());
        decimal FpP = Convert.ToDecimal(dtDatos.Rows[0][5].ToString());
        decimal revoluciones = Convert.ToDecimal(dtDatos.Rows[0][4].ToString()), VOLp =
Convert.ToDecimal(dtDatos.Rows[0][2].ToString());
        int rebobinados = Convert.ToInt32(dtDatos.Rows[0][9].ToString()), antiguedad =
Convert.ToInt32(dtDatos.Rows[0][11].ToString());
        string tipo = dtDatos.Rows[0][7].ToString();
        decimal VOLm, Im, FPm, Pot, FC, FC1 = 0, FC2 = 0, Ncal;

        VOLm = (m.voltaje1 + m.voltaje2 + m.voltaje3) / 3;
        Im = (m.corriente1 + m.corriente2 + m.corriente3) / 3;
        FPm = (m.fp1 + m.fp2 + m.fp3) / 300;

        Pot = (Convert.ToDecimal(Math.Sqrt(3)) * VOLm * Im * FPm) / 1000;
        FC = (Pot * n) / (hp * Convert.ToDecimal(0.746));

        if (FC <= Convert.ToDecimal(0.25))
        {
            auxQuery = "ef_25, ef_50"; FC1 = 0; FC2 = Convert.ToInt32(0.25);
        }
        else if (FC > Convert.ToDecimal(0.25) && FC <= Convert.ToDecimal(0.5))
        {
            auxQuery = "ef_25, ef_50"; FC1 = Convert.ToDecimal(0.25); FC2 =
Convert.ToDecimal(0.5);
        }
        else if (FC > Convert.ToDecimal(0.5) && FC <= Convert.ToDecimal(0.75))
    }
}

```

```

    {
        auxQuery = "ef_50, ef_75"; FC1 = Convert.ToDecimal(0.5); FC2 =
Convert.ToDecimal(0.75);
    }
    else if(FC > Convert.ToDecimal(0.75))
    {
        auxQuery = "ef_75, ef_100"; FC1 = Convert.ToDecimal(0.75); FC2 =
Convert.ToDecimal(1.0);
    }

    query = "SELECT " + auxQuery + " FROM snr.cat_curvaseficiencia WHERE potencia = '" + hp
+ "' AND tipo = '" + tipo + "' AND " +
        "carcaza = 'abierto' AND rpm = '" + revoluciones + "'";
    DataTable dtEficiencia = datos(query);

    decimal Ninf = Convert.ToDecimal(dtEficiencia.Rows[0][0].ToString()), Nsup =
Convert.ToDecimal(dtEficiencia.Rows[0][1].ToString());

    Ncal = ((FC - FC1) / (FC2 - FC1)) * (Nsup - Ninf) + Ninf;

    decimal VOLstd = VOLm / VOLp - 1, FAvv = VOLstd * (Convert.ToDecimal(0.07) -
Convert.ToDecimal(1.334) * VOLstd) - Convert.ToDecimal(0.0009);
    decimal VOLmax = 0, VOLmin = 0, DVstd;

    if(m.voltaje1 > m.voltaje2 && m.voltaje1 > m.voltaje3)
    {
        VOLmax = m.voltaje1;

        if(m.voltaje2 > m.voltaje3)
        {
            VOLmin = m.voltaje3;
        }
        else
        {
            VOLmin = m.voltaje2;
        }
    }
    else if(m.voltaje1 > m.voltaje2 && m.voltaje1 < m.voltaje3)
    {
        VOLmax = m.voltaje3;
        VOLmin = m.voltaje2;
    }
    else if(m.voltaje1 < m.voltaje2 && m.voltaje2 > m.voltaje3)
    {
        VOLmax = m.voltaje2;

        if (m.voltaje1 > m.voltaje3)
        {
            VOLmin = m.voltaje3;
        }
        else
        {
            VOLmin = m.voltaje1;
        }
    }
    }

    if (VOLmax - VOLm >= VOLm - VOLmin)
    {
        DVstd = (VOLmax - VOLm) / VOLm;
    }
    else
    {
        DVstd = (VOLm - VOLmin) / VOLm;
    }

    decimal FAdv = Convert.ToDecimal(1) - DVstd * (Convert.ToDecimal(0.0113) -
Convert.ToDecimal(0.0073) * DVstd), FAre = 0, Nstd_ajus;

    if(rebobinados > 0)
    {
        FAre = Convert.ToDecimal(0.025);
    }

```

```

        if(antiguedad > 10)
        {
            Nstd_ajus = FAdv * (Ncal + FAvv - FAre) - Convert.ToDecimal(0.01);
        }
        else
        {
            Nstd_ajus = FAdv * (Ncal + FAvv - FAre);
        }

        return RedirectToAction("MotoresDiag", "MotoresEl", new { Nstd_res = Nstd_ajus });
    }

    return View(m);
} // termina post motoresMed

public ActionResult MotoresDiag(decimal Nstd_res)
{
    Session["UsuarioNombre"] = "Prueba";
    Session["encuesta"] = "true";
    Session["editarPerfil"] = "true";
    Session["inicioACT"] = "noCurrent";
    Session["empresaACT"] = "current";

    decimal result = Nstd_res * 100;

    ViewData["Eficiencia"] = result.ToString("0.##");
    return View();
} // termina get for motoresDiag
}
}

```

### A.3.3. Bomba

- Pseudocódigo

```

programa bomba
mostrar vista bomba
obtener aplicación, operación, volCont, tiempoLlenado,
numServicios, diametro, calFricción, presionSucc,
fluidobomb, presionMan, materialTub, hMax, qMax, hOp,
qMin, potMot, corriente, eficiencia, tension, costo,
facPot, qOp, horas, presionOp, dias, distanciaVert,
distanciaHorz

Si operacion = "no" entonces (Quiere decir que no sabe
el flujo de operación)
    Si aplicación = "trasvase" entonces
        gasto <- volCont / tiempoLlenado
        hf <- (disHorz * disVert) * 0.1
    Si no si aplicación = "presurizado" entonces
        auxServicios <- numServicios * 5.5
        gasto <- -0.000002161 * auxServicios^3 *
        0.093254272 * 49.737719298
        hf <- (disHorz * disVert) * 0.98
    Si no si aplicación = "procesos" entonces
        buscar en base de datos gasto y hf a partir de
        diametro de tuberia
    Si no entonces
        gasto <- 0
        hf <- (disHorz * disVert)
    fin del Si
Si no entonces
    gasto <- qOp

```

```

        hf <- (disHorz * disVert)
fin del Si

Si presioSucc = "Manometrico" entonces
    presionEvacua <- presionMan
Si no entonces
    presionEvacua <- 0
fin del Si

Si calFriccion = 5 entonces
    V2 <- 1
Si no entonces
    V2 <- 2.2
fin del Si

p1 <- 101.3 * (1 - (0.000022557) * alturaCiudad) ^ 5.256
p1q <- p1 - presionEvacua
p2q <- presionOp - p1
Gh <- 9.8 * 6370000 ^ 2 / (6370000 - alturaCiudad) ^ 2
Buscar la densidad del fluido en base de datos
Y <- Gh * densidad
deltaZ <- distVert
Carga <- (p2q - p1q * 1000) / Y + deltaZ + V2 ^ 2 / (2 * Gh) +
hf
Potencia <- (gasto * Carga * Y) / 1000
CNPS <- (p1 - 3.16975) / Y + deltaZ - hf
potenciaE = (corriente * tension * (facPot / 100) + RAIZ(3)) /
1000
EficienciaConj <- (Potencia / potenciaE) * 100
EficienciaBom <- EficienciaConj / (eficiencia / 100)
Demandal <- potenciaE * horas * dias
EnerAp <- Potencia * horas * dias

```

**fin de programa bomba**

- Código

```

namespace sitioSener.Controllers
{
    public class BombasController : Controller
    {
        // GET: Bombas
        public ActionResult Bombas()
        {
            Session["UsuarioNombre"] = "Prueba";
            Session["encuesta"] = "true";
            Session["editarPerfil"] = "true";
            Session["inicioACT"] = "noCurrent";
            Session["empresaACT"] = "current";

            List<Aplicaciones> liAplic = new List<Aplicaciones>();
            liAplic.Add(new Aplicaciones() { idApli = "Trasvase", nombre = "Trasvase (trasiego,
transporte,...)" });
            liAplic.Add(new Aplicaciones() { idApli = "Presurizado", nombre = "Presurizado (serv.
sanitarios, ...)" });
            liAplic.Add(new Aplicaciones() { idApli = "Procesos", nombre = "Procesos industriales
(distribución a equipos industriales...)" });
            ViewBag.aplicacion = liAplic;

            List<Unidades> liUnidadVol = new List<Unidades>();

```

```

liUnidadVol.Add(new Unidades() { idUnidad = "m3", nombre = "m^3" });
liUnidadVol.Add(new Unidades() { idUnidad = "ft3", nombre = "ft^3" });
liUnidadVol.Add(new Unidades() { idUnidad = "in3", nombre = "In^3" });
liUnidadVol.Add(new Unidades() { idUnidad = "Ltrs", nombre = "Ltrs" });
ViewBag.unidadVol = liUnidadVol;

List<Unidades> liUnidadTime = new List<Unidades>();
liUnidadTime.Add(new Unidades() { idUnidad = "hrs", nombre = "hrs" });
liUnidadTime.Add(new Unidades() { idUnidad = "min", nombre = "min" });
liUnidadTime.Add(new Unidades() { idUnidad = "seg", nombre = "seg" });
ViewBag.unidadTime = liUnidadTime;

List<Catalogo> liCatFriccion = new List<Catalogo>();
liCatFriccion.Add(new Catalogo() { idCat = "5", nombre = "5%" });
liCatFriccion.Add(new Catalogo() { idCat = "10", nombre = "10%" });
ViewBag.catFricc = liCatFriccion;

List<Catalogo> liCatPre = new List<Catalogo>();
liCatPre.Add(new Catalogo() { idCat = "Atmosférico", nombre = "Atmosférico" });
liCatPre.Add(new Catalogo() { idCat = "Manométrico", nombre = "Manométrico" });
ViewBag.catPre = liCatPre;

List<Catalogo> li = new List<Catalogo>();
string query = "SELECT idden, fluido FROM snr.cat_densidades order by fluido";

DataTable dtCat = datos(query);
for (int i = 0; i < dtCat.Rows.Count; i++)
{
    li.Add(new Catalogo()
    {
        idCat = dtCat.Rows[i][0].ToString(),
        nombre = dtCat.Rows[i][1].ToString()
    });
}
ViewBag.catFluidos = li;

List<Catalogo> li2 = new List<Catalogo>();
query = "SELECT idDm, diametro FROM snr.cat_diametros";
DataTable dtCatDm = datos(query);
for (int i = 0; i < dtCatDm.Rows.Count; i++)
{
    li2.Add(new Catalogo()
    {
        idCat = dtCatDm.Rows[i][0].ToString(),
        nombre = dtCatDm.Rows[i][1].ToString()
    });
}
ViewBag.catDiametro = li2;

List<Unidades> liUnidadMan = new List<Unidades>();
liUnidadMan.Add(new Unidades() { idUnidad = "Bar", nombre = "Bar" });
liUnidadMan.Add(new Unidades() { idUnidad = "kpa", nombre = "kpa" });
liUnidadMan.Add(new Unidades() { idUnidad = "Psi", nombre = "Psi" });
liUnidadMan.Add(new Unidades() { idUnidad = "InH2O", nombre = "InH2O" });
liUnidadMan.Add(new Unidades() { idUnidad = "kg_cm", nombre = "kg/cm^2" });
liUnidadMan.Add(new Unidades() { idUnidad = "Atm", nombre = "Atm" });
liUnidadMan.Add(new Unidades() { idUnidad = "mmHg", nombre = "mmHg" });
liUnidadMan.Add(new Unidades() { idUnidad = "cmHg", nombre = "cmHg" });
ViewBag.unidadMan = liUnidadMan;
List<Catalogo> liCatMat = new List<Catalogo>();

liCatMat.Add(new Catalogo() { idCat = "Acero", nombre = "Acero" });
liCatMat.Add(new Catalogo() { idCat = "Plástico (PE, PVC)", nombre = "Plástico (PE, PVC)"
});
liCatMat.Add(new Catalogo() { idCat = "Poliester reforzado con fibra de vidrio", nombre =
"Poliester reforzado con fibra de vidrio" });
liCatMat.Add(new Catalogo() { idCat = "Hierro galvanizado", nombre = "Hierro galvanizado"
});
ViewBag.catMat = liCatMat;

List<Unidades> liUnidadCarga = new List<Unidades>();
liUnidadCarga.Add(new Unidades() { idUnidad = "m", nombre = "m" });

```

```

    liUnidadCarga.Add(new Unidades() { idUnidad = "ft", nombre = "ft" });
    ViewBag.unidadH = liUnidadCarga;

    List<Unidades> liUnidadGasto = new List<Unidades>();
    liUnidadGasto.Add(new Unidades() { idUnidad = "m3/s", nombre = "m^3/s" });
    liUnidadGasto.Add(new Unidades() { idUnidad = "gpm", nombre = "gpm" });
    liUnidadGasto.Add(new Unidades() { idUnidad = "L/s", nombre = "L/s" });
    liUnidadGasto.Add(new Unidades() { idUnidad = "L/min", nombre = "L/min" });
    ViewBag.unidadQ = liUnidadGasto;

    List<Unidades> liUnidadPot = new List<Unidades>();
    liUnidadPot.Add(new Unidades() { idUnidad = "HP", nombre = "HP" });
    liUnidadPot.Add(new Unidades() { idUnidad = "kW", nombre = "kW" });
    ViewBag.unidadPot = liUnidadPot;

    List<Unidades> liUnidadPre = new List<Unidades>();
    liUnidadPre.Add(new Unidades() { idUnidad = "Bar", nombre = "Bar" });
    liUnidadPre.Add(new Unidades() { idUnidad = "kpa", nombre = "kpa" });
    liUnidadPre.Add(new Unidades() { idUnidad = "Psi", nombre = "Psi" });
    liUnidadPre.Add(new Unidades() { idUnidad = "InH20", nombre = "InH20" });
    liUnidadPre.Add(new Unidades() { idUnidad = "kg_cm", nombre = "kg/cm^2" });
    ViewBag.unidadPre = liUnidadPre;

    List<Unidades> liUnidadDist = new List<Unidades>();
    liUnidadDist.Add(new Unidades() { idUnidad = "m", nombre = "m" });
    liUnidadDist.Add(new Unidades() { idUnidad = "ft", nombre = "ft" });
    liUnidadDist.Add(new Unidades() { idUnidad = "Inch", nombre = "Inch" });
    ViewBag.unidadDist = liUnidadDist;

    return View();
}

// POST: Bombas
[HttpPost]
public ActionResult Bombas(BombasDatosIn model)
{
    string query;
    if (ModelState.IsValid)
    {
        Random r = new Random();
        int r1 = r.Next(0, 9);
        int r2 = r.Next(0, 9);
        int r3 = r.Next(0, 9);
        int l = r.Next(0, 25);
        string letra = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        string key = "BOM" + r1 + r2 + r3 + Session["UsuarioNombre"].ToString() +
System.DateTime.Now.ToString("yyMMdd") + letra[l];
        decimal t, V, Hf, dV, dH, auxServ, Q; //Q es Gasto (Q) de la tabla de resultados

        DataTable dt1;
        if (model.unidadDistV == "ft")
        {
            dV = model.distVert * Convert.ToDecimal(0.3048);
        }
        else if (model.unidadDistV == "Inch")
        {
            dV = model.distVert * Convert.ToDecimal(0.0254);
        }
        else
        {
            dV = model.distVert;
        }

        if (model.unidadDistH == "ft")
        {
            dH = model.distHorz * Convert.ToDecimal(0.3048);
        }
        else if (model.unidadDistH == "Inch")
        {
            dH = model.distHorz * Convert.ToDecimal(0.0254);
        }
    }
}

```

```

else
{
    dH = model.distHorz;
}

if (model.operacion == "No")
{
    switch (model.aplicacion)
    {
        case "Trasvase":

            if (model.unidadTime == "hrs")
            {
                t = model.tiempoLlenado * 60 * 60;
            }
            else if (model.unidadTime == "min")
            {
                t = model.tiempoLlenado * 60;
            }
            else
            {
                t = model.tiempoLlenado;
            }

            if (model.unidadVolCont == "ft3")
            {
                V = model.volCont * Convert.ToDecimal(0.0283);
            }
            else if (model.unidadVolCont == "in3")
            {
                V = model.volCont * Convert.ToDecimal(0.0000164);
            }
            else if (model.unidadVolCont == "Ltrs")
            {
                V = model.volCont * Convert.ToDecimal(0.001);
            }
            else
            {
                V = model.volCont;
            }

            Q = V / t;
            Hf = (dV + dH) * Convert.ToDecimal(0.1);

            break;

        case "Presurizado":
            auxServ = model.numServicios * Convert.ToDecimal(5.5);
            Q = Convert.ToDecimal(-0.000002161) * auxServ * auxServ +
Convert.ToDecimal(0.093254272) * auxServ +
            Convert.ToDecimal(49.737719298);
            Hf = (dV + dH) * Convert.ToDecimal(0.098);
            break;

        case "Procesos":
            query = "SELECT iddm, diametro, flujo, perdidas FROM snr.cat_diametros
WHERE iddm = '" + model.diametro + "'";
            dt1 = datos(query);
            Q = Convert.ToDecimal(dt1.Rows[0][2].ToString());
            Hf = (dV + dH) * Convert.ToDecimal(dt1.Rows[0][3].ToString());
            break;

        default:
            Q = 0;
            Hf = (dV + dH);
            break;
    }
} // termina if para cuando no conoce el flujo de operacion
else
{
    if (model.unidadQop == "m3/s")
    {

```

```

        Q = model.qOp * Convert.ToDecimal(15852.056);
    }
    else if (model.unidadQop == "L/s")
    {
        Q = model.qOp * Convert.ToDecimal(15.85);
    }
    else if (model.unidadQop == "L/min")
    {
        Q = model.qOp * Convert.ToDecimal(0.2642);
    }
    else
    {
        Q = model.qOp;
    }

    Hf = (dV + dH);
} //termina caso para cuando se conoce el flujo de operacion

decimal PO = Convert.ToDecimal(101.3), Pop;

if (model.unidadPop == "Bar") { Pop = model.presionOp * Convert.ToDecimal(100.0); }
else if (model.unidadPop == "Psi") { Pop = model.presionOp * Convert.ToDecimal(6.9); }
else if (model.unidadPop == "InH2O") { Pop = model.presionOp * Convert.ToDecimal(0.25); }
}
else if (model.unidadPop == "kg_cm") { Pop = model.presionOp * Convert.ToDecimal(98.1); }
}
else if (model.unidadPop == "Atm") { Pop = model.presionOp * Convert.ToDecimal(101.3); }
}
else if (model.unidadPop == "mmHg") { Pop = model.presionOp * Convert.ToDecimal(0.13); }
}
else if (model.unidadPop == "cmHg") { Pop = model.presionOp * Convert.ToDecimal(1.33); }
}
else { Pop = model.presionOp; }

double A = 536.0; //A debe ser obtenido de la altitud del estado o ciudad, falta ligar
este valor

decimal P1 = PO * Convert.ToDecimal(Math.Pow(1 - (0.000022557) * A, 5.256));
decimal Pvacuo;

if (model.presionSucc == "Manométrico")
{
    if (model.unidadMan == "Bar") { Pvacuo = model.presionMan *
Convert.ToDecimal(100.0); }
    else if (model.unidadMan == "Psi") { Pvacuo = model.presionMan *
Convert.ToDecimal(6.9); }
    else if (model.unidadMan == "InH2O") { Pvacuo = model.presionMan *
Convert.ToDecimal(0.25); }
    else if (model.unidadMan == "kg_cm") { Pvacuo = model.presionMan *
Convert.ToDecimal(98.1); }
    else if (model.unidadMan == "Atm") { Pvacuo = model.presionMan *
Convert.ToDecimal(101.3); }
    else if (model.unidadMan == "mmHg") { Pvacuo = model.presionMan *
Convert.ToDecimal(0.13); }
    else if (model.unidadMan == "cmHg") { Pvacuo = model.presionMan *
Convert.ToDecimal(1.33); }
    else { Pvacuo = model.presionMan; }
}
else
{
    Pvacuo = 0;
}

decimal Plq = P1 - Pvacuo, P2q = Pop - P1, Gh = Convert.ToDecimal(9.8) *
Convert.ToDecimal(Math.Pow(6370000, 2)) / Convert.ToDecimal(Math.Pow(6370000 - A, 2));

query = "SELECT densidad FROM snr.cat_densidades WHERE idden = '" + model.fluidoBomb +
""";

dt1 = datos(query);
decimal Y = Gh * Convert.ToDecimal(dt1.Rows[0][0].ToString()), deltaZ = dV;
double V2;

if(model.calFriccion == "5") { V2 = 1; }

```



```

        else { V2 = 2.2; }
        decimal Ht = ((P2q - P1q) * Convert.ToDecimal(1000)) / Y + deltaZ +
(Convert.ToDecimal(Math.Pow(V2, 2))) / (2 * Gh) + Hf; //Corresponde al Carga (H) de la tabla de
resultados
        decimal Ph = (Q * Ht * Y) / Convert.ToDecimal(1000); // Corresponde a Potencia hid de
la tabla de resultados;
        decimal CNPS = (P1 - Convert.ToDecimal(3.16975)) / Y + deltaZ - Hf; //Corresponde a
CNPS de la tabla de resultados

        decimal Pe = (model.corriente * model.tension * (model.facPot / 100) +
Convert.ToDecimal(Math.Sqrt(3))) / 1000;

        decimal Ncal = (Ph / Pe) * 100; //corresponde a η conjunto bba-mot (calculada) de la
tabla de resultados
        decimal Nbom = Ncal / (model.eficiencia / 100); //corresponde a η bomba (estimada) de
la tabla de resultados
        decimal demanda = Pe * model.horas * model.dias; //corresponde a demanda elecetrica
[kWh/año] de la tabla de resultados
        decimal energia = Ph * model.horas * model.dias; // corresponde a Energia aprovechada
por el sistema de la tabla de resultados

        return RedirectToAction("BombasDiag", "Bombas", new { keyIn = key, gasto = Q, carga =
Ht,
                potencia = Ph, cnps = CNPS, Nconjunto = Ncal, Nbomba = Nbom, demElect = demanda,
energia = energia,
                costo = model.costo});

    } // termina de calcular cuando el modelo es valido

    List<Aplicaciones> liAplic = new List<Aplicaciones>();
    liAplic.Add(new Aplicaciones() { idAplic = "Trasvase", nombre = "Trasvase (trasiego,
transporte,...)" });
    liAplic.Add(new Aplicaciones() { idAplic = "Presurizado", nombre = "Presurizado (serv.
sanitarios, ...)" });
    liAplic.Add(new Aplicaciones() { idAplic = "Procesos", nombre = "Procesos industriales
(distribución a equipos industriales...)" });
    ViewBag.aplicacion = liAplic;

    List<Unidades> liUnidadVol = new List<Unidades>();
    liUnidadVol.Add(new Unidades() { idUnidad = "m3", nombre = "m^3" });
    liUnidadVol.Add(new Unidades() { idUnidad = "ft3", nombre = "ft^3" });
    liUnidadVol.Add(new Unidades() { idUnidad = "in3", nombre = "In^3" });
    liUnidadVol.Add(new Unidades() { idUnidad = "Ltrs", nombre = "Ltrs" });
    ViewBag.unidadVol = liUnidadVol;

    List<Unidades> liUnidadTime = new List<Unidades>();
    liUnidadTime.Add(new Unidades() { idUnidad = "hrs", nombre = "hrs" });
    liUnidadTime.Add(new Unidades() { idUnidad = "min", nombre = "min" });
    liUnidadTime.Add(new Unidades() { idUnidad = "seg", nombre = "seg" });
    ViewBag.unidadTime = liUnidadTime;

    List<Catalogo> liCatFriccion = new List<Catalogo>();
    liCatFriccion.Add(new Catalogo() { idCat = "5", nombre = "5%" });
    liCatFriccion.Add(new Catalogo() { idCat = "10", nombre = "10%" });
    ViewBag.catFricc = liCatFriccion;

    List<Catalogo> liCatPre = new List<Catalogo>();
    liCatPre.Add(new Catalogo() { idCat = "Atmosférico", nombre = "Atmosférico" });
    liCatPre.Add(new Catalogo() { idCat = "Manométrico", nombre = "Manométrico" });
    ViewBag.catPre = liCatPre;

    List<Catalogo> li = new List<Catalogo>();
    query = "SELECT idden, fluido FROM snr.cat_densidades order by fluido";

    DataTable dtCat = datos(query);
    for (int i = 0; i < dtCat.Rows.Count; i++)
    {
        li.Add(new Catalogo()
        {
            idCat = dtCat.Rows[i][0].ToString(),
            nombre = dtCat.Rows[i][1].ToString()
        });
    }

```

```

    }
    ViewBag.catFluidos = li;

    List<Catalogo> li2 = new List<Catalogo>();
    query = "SELECT idDm, diametro FROM snr.cat_diametros";
    DataTable dtCatDm = datos(query);
    for (int i = 0; i < dtCatDm.Rows.Count; i++)
    {
        li2.Add(new Catalogo()
        {
            idCat = dtCatDm.Rows[i][0].ToString(),
            nombre = dtCatDm.Rows[i][1].ToString()
        });
    }
    ViewBag.catDiametro = li2;

    List<Unidades> liUnidadMan = new List<Unidades>();
    liUnidadMan.Add(new Unidades() { idUnidad = "Bar", nombre = "Bar" });
    liUnidadMan.Add(new Unidades() { idUnidad = "kpa", nombre = "kpa" });
    liUnidadMan.Add(new Unidades() { idUnidad = "Psi", nombre = "Psi" });
    liUnidadMan.Add(new Unidades() { idUnidad = "InH2O", nombre = "InH2O" });
    liUnidadMan.Add(new Unidades() { idUnidad = "kg_cm", nombre = "kg/cm^2" });
    liUnidadMan.Add(new Unidades() { idUnidad = "Atm", nombre = "Atm" });
    liUnidadMan.Add(new Unidades() { idUnidad = "mmHg", nombre = "mmHg" });
    liUnidadMan.Add(new Unidades() { idUnidad = "cmHg", nombre = "cmHg" });
    ViewBag.unidadMan = liUnidadMan;

    List<Catalogo> liCatMat = new List<Catalogo>();
    liCatMat.Add(new Catalogo() { idCat = "Acero", nombre = "Acero" });
    liCatMat.Add(new Catalogo() { idCat = "Plástico (PE, PVC)", nombre = "Plástico (PE, PVC)"
});
    liCatMat.Add(new Catalogo() { idCat = "Poliester reforzado con fibra de vidrio", nombre =
"Poliester reforzado con fibra de vidrio" });
    liCatMat.Add(new Catalogo() { idCat = "Hierro galvanizado", nombre = "Hierro galvanizado"
});

    ViewBag.catMat = liCatMat;

    List<Unidades> liUnidadCarga = new List<Unidades>();
    liUnidadCarga.Add(new Unidades() { idUnidad = "m", nombre = "m" });
    liUnidadCarga.Add(new Unidades() { idUnidad = "ft", nombre = "ft" });
    ViewBag.unidadH = liUnidadCarga;

    List<Unidades> liUnidadGasto = new List<Unidades>();
    liUnidadGasto.Add(new Unidades() { idUnidad = "m3/s", nombre = "m^3/s" });
    liUnidadGasto.Add(new Unidades() { idUnidad = "gpm", nombre = "gpm" });
    liUnidadGasto.Add(new Unidades() { idUnidad = "L/s", nombre = "L/s" });
    liUnidadGasto.Add(new Unidades() { idUnidad = "L/min", nombre = "L/min" });
    ViewBag.unidadQ = liUnidadGasto;

    List<Unidades> liUnidadPot = new List<Unidades>();
    liUnidadPot.Add(new Unidades() { idUnidad = "HP", nombre = "HP" });
    liUnidadPot.Add(new Unidades() { idUnidad = "kW", nombre = "kW" });
    ViewBag.unidadPot = liUnidadPot;

    List<Unidades> liUnidadPre = new List<Unidades>();
    liUnidadPre.Add(new Unidades() { idUnidad = "Bar", nombre = "Bar" });
    liUnidadPre.Add(new Unidades() { idUnidad = "kpa", nombre = "kpa" });
    liUnidadPre.Add(new Unidades() { idUnidad = "Psi", nombre = "Psi" });
    liUnidadPre.Add(new Unidades() { idUnidad = "InH2O", nombre = "InH2O" });
    liUnidadPre.Add(new Unidades() { idUnidad = "kg_cm", nombre = "kg/cm^2" });
    ViewBag.unidadPre = liUnidadPre;

    List<Unidades> liUnidadDist = new List<Unidades>();
    liUnidadDist.Add(new Unidades() { idUnidad = "m", nombre = "m" });
    liUnidadDist.Add(new Unidades() { idUnidad = "ft", nombre = "ft" });
    liUnidadDist.Add(new Unidades() { idUnidad = "Inch", nombre = "Inch" });
    ViewBag.unidadDist = liUnidadDist;

    return View(model);
}

// GET: Bombas

```

```

public ActionResult BombasDiag( string keyIn, decimal gasto, decimal carga, decimal potencia,
decimal cnps,
    decimal Nconjunto, decimal Nbomba, decimal demElect, decimal energia, decimal costo)
{
    Session["UsuarioNombre"] = "Prueba";
    Session["encuesta"] = "true";
    Session["editarPerfil"] = "true";
    Session["inicioACT"] = "noCurrent";
    Session["empresaACT"] = "current";
    ViewData["Carga"] = carga.ToString("0.##");
    ViewData["Gasto"] = gasto.ToString("0.##");
    ViewData["Potencia"] = potencia.ToString("0.##");
    ViewData["CNPS"] = cnps.ToString("0.##");
    decimal EfConj = Nconjunto * 100;
    ViewData["EficienciaConj"] = EfConj.ToString("0.##");
    decimal EfBomba = Nbomba * 100;
    if(Nbomba > Convert.ToDecimal(0.9))
    {
        ViewData["EficienciaBom"] = "Verificar";
    }
    else
    {
        ViewData["EficienciaBom"] = EfBomba.ToString("0.##");
    }

    ViewData["Demandal"] = demElect.ToString("0.##");
    decimal dem2 = demElect * Convert.ToDecimal(3.6);
    ViewData["Demanda2"] = dem2.ToString("0.##");
    decimal dem3 = demElect * costo;
    ViewData["Demanda3"] = dem3.ToString("0.##");
    ViewData["EnerAp"] = energia.ToString("0.00");
    decimal energiaPer = energia - demElect;

    ViewBag.energiaAp = energia;
    ViewBag.energiaPd = energiaPer;

    return View();
}

public ActionResult GraficaResult(string EnergiaAp, string EnergiaPr)
{
    List<Grafica> data = new List<Grafica>();

    data.Add(new Grafica()
    {
        etiqueta = "Energia Aprovechada por el Sistema",
        valor = Convert.ToDecimal(EnergiaAp),
    });
    data.Add(new Grafica()
    {
        etiqueta = "Energia Perdida por irreversibilidades",
        valor = Convert.ToDecimal(EnergiaPr),
    });

    return Json(data.ToList(), JsonRequestBehavior.AllowGet);
}

```

### A.3.4. Iluminación

- Pseudocódigo

```

programa Iluminación
mostrar vista iluminacion
obtener nombreArea, tipoArea, largoArea, anchoArea,
altoArea, colorTecho, colorPared, colorPiso,
alturaTrabajo, alturaMontaje, horasDia, diasAnio

```

```

Si click en "Agregar luminaria" entonces
  mostrar vista luminaria
  obtener tipoIlumina, totalIlumina, tipoLampara,
  lampxlumina, lumenLamp, potLamp, tipoBalastro,
  factorBalastro, potLinea, mantenimiento

  numLumina <- numlumina + 1

  Si tipoLumina es igual a "incandescente" ó "LED"
  ó "Florescente compacta" ó "Inducción" entonces
    lumTecnologia <- totalLumina * lampxLumina
    * lumenLamp
    potlinea <- potLamp * lampxLumina
  Si no entonces
    lumTecnologia <- totalLumina * lampxLumina *
    lumenLamp * factorBalastro
  fin del sí

  potTecnologia <- totalLumina * potlinea
  efiTecnologia <- lumTecnologia / potTecnologia
  RCT <- (5 * (alto - alturaMontaje) * (largo +
  ancho)) / (largo * ancho)
  RCL <- (5 * (alMontaje - alTrabajo) * (largo +
  ancho)) / (largo * ancho)
  RCP <- (5 * (alTrabajo) * (largo + ancho)) /
  (largo * ancho)
  obtener reflectanciaPared de base de datos
  obtener reflectanciaTecho de base de datos
  obtener reflectanciaPiso de base de datos
  obtener reflectanciaEfectivaP de base de datos
  obtener reflectanciaEfectivaT de base de datos
  obtener reflectanciaEfectivaPi de base de datos

  Según tipoIlumina obtener coeUtilizacion de base
  de datos
  Según Mantenimiento obtener factorMantenimiento de
  base de datos
  lumUtiles <- lumTecnologia * coeUtilizacion *
  factorMantenimiento
  consumoTecnologia <- potTecnologia * horasdia *
  diasAnio

fin del Si

superficie <- largoArea * anchoArea
obtener numLumi de base de datos

x <- 1
Mientras x < numlumi hacer
SumaPotxArea <- potxArea + SumaPotxArea
x <- x + 1
fin del Mientras

x <- 1
Mientras x < numlumi hacer
SumaLumxArea <- LumxArea + SumaLumxArea
x <- x + 1
fin del Mientras

```

```

x <- 1
Mientras x < numlumi hacer
SumalumUtilesArea <- lumUtilesArea + SumalumUtilesArea
x <- x + 1
fin del Mientras

Según tipoArea obtener nom25 de base de datos
Según tipoArea obtener dpea de base de datos
iluCalculado <- lumUtilesArea / superficie
dpeaCalculado <- potxArea / superficie

x <- 1
consumoEnergetico <- 1
Mientras x < numLumi hacer
consumoEnergetico <- CconsumoEnergetico +
consumoTecnologia
x <- x + 1
fin del Mientras

```

**fin programa Iluminación**

- Código

```

namespace sitioSener.Controllers
{
    public class IluminacionController : Controller
    {
        private double interpolar(double a, int bandera)
        {
            if (bandera == 0)
            {
                double tmp = (a % 1);
                if (tmp == 0.1 || tmp == 0.3 || tmp == 0.5 || tmp == 0.7 || tmp == 0.9)
                    a += 0.1;
            }
            else if (bandera == 1)
            {
                a = Math.Ceiling(a);
            }
            else if (bandera == 2)
            {
                if (a <= 10)
                    a = 10;
                else if (a > 10 && a <= 30)
                    a = 30;
                else if (a > 30 && a <= 50)
                    a = 50;
                else if (a > 50 && a <= 70)
                    a = 70;
                else
                    a = 80;
            }
            return a;
        }
        //Seccion de Areas de iluminacion
        public ActionResult iluminacion()
        {
            Session["UsuarioNombre"] = "Prueba";
            Session["encuesta"] = "true";
            Session["editarPerfil"] = "true";
            Session["inicioACT"] = "noCurrent";
            Session["empresaACT"] = "current";
        }
    }
}

```

```

if (Session["UsuarioNombre"] != null)
{
    List<TipoArea> litipoArea = new List<TipoArea>();
    List<Techo> liTecho = new List<Techo>();
    List<Pared> liPared = new List<Pared>();
    List<Piso> liPiso = new List<Piso>();

    string query = "SELECT count(*) FROM snr.iluminacion WHERE usuario = '" +
Session["UsuarioNombre"] + "'";
    DataTable data = datos(query);

    if (data.Rows[0][0].ToString().Equals('0'))
    {
        ViewData["Contador"] = "1";
    }
    else
    {
        ViewData["Contador"] = (Convert.ToInt32(data.Rows[0][0].ToString()) +
1).ToString();
    }

    /* Lista de Tipo de Area*/
    query = "SELECT idTipo,tipo FROM snr.cat_TipoArea order by tipo";

    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        litipoArea.Add(new TipoArea()
        {
            idTipo = Convert.ToInt32(data.Rows[i][0].ToString()),
            tipo = data.Rows[i][1].ToString()
        });
    }
    ViewBag.TipoArea = litipoArea;

    /*Lista de color de techo*/
    query = "SELECT idTecho,color FROM snr.cat_Techo order by color";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        liTecho.Add(new Techo()
        {
            idTecho = Convert.ToInt32(data.Rows[i][0].ToString()),
            color = data.Rows[i][1].ToString()
        });
    }
    ViewBag.Techo = liTecho;

    /*Lista de color de pared*/
    query = "SELECT idPared,color FROM snr.cat_Pared order by color";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        liPared.Add(new Pared()
        {
            idPared = Convert.ToInt32(data.Rows[i][0].ToString()),
            color = data.Rows[i][1].ToString()
        });
    }
    ViewBag.Pared = liPared;

    /*Lista de color de piso*/

    query = "SELECT idPiso,color FROM snr.cat_Piso order by color";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {

```

```

        liPiso.Add(new Piso()
        {
            idPiso = Convert.ToInt32(data.Rows[i][0].ToString()),
            color = data.Rows[i][1].ToString()
        });
    }
    ViewBag.Piso = liPiso;
    return View();
}
else
{
    return RedirectToAction("Login", "Registro");
}
}

[HttpPost]
public ActionResult iluminacion(iluminacion model)
{
    if (Session["UsuarioNombre"] != null)
    {
        if (ModelState.IsValid)
        {
            Random r = new Random();
            int r1 = r.Next(0, 9);
            int r2 = r.Next(0, 9);
            int r3 = r.Next(0, 9);
            int l = r.Next(0, 25);
            string letra = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

            string idArea = "ILU" + r1 + r2 + r3 + Session["UsuarioNombre"].ToString() +
System.DateTime.Now.ToString("yyMMdd") + letra[l];

            //Modificar para que aqui aparezca el numero de empresa a la que esta agregando la
area
            string query2 = "INSERT INTO snr.iluminacion (idArea, usuario,
nombreArea,numArea, tipoArea, nom25, dpea, largoArea, anchoArea, altoArea, colorTecho, colorPared, colorPiso, " +
" alturaTrabajo, alturaMontaje, horasDia, diasAnio) VALUES ('" + idArea + "','" +
Session["UsuarioNombre"] + "','" + model.nombreArea + "','" + model.numArea + "','" + model.tipoArea +
 "','" + model.nom25 + "','" +
model.dpea + "','" + model.largoArea + "','" + model.anchoArea + "','" +
model.altoArea + "','" +
model.colorTecho + "','" + model.colorPared + "','" + model.colorPiso + "','" +
model.alturaTrabajo + "','" + model.alturaMontaje + "','" + model.horasDia + "','" + model.diasAnio +
 "')";

            ejecutar(query2);
            if (salir)
            {
                salir = false;
                return RedirectToAction("Inicio", "Sitio");
            }
            else if (lumi)
            {
                lumi = false;
                return RedirectToAction("luminaria", "Iluminacion", new { idArea = idArea});
            }
        }
    }

    List<TipoArea> litipoArea = new List<TipoArea>();
    List<Techo> liTecho = new List<Techo>();
    List<Pared> liPared = new List<Pared>();
    List<Piso> liPiso = new List<Piso>();

    string query = "SELECT count(*) FROM snr.iluminacion WHERE usuario = '" +
Session["UsuarioNombre"] + "'";
    DataTable data = datos(query);

    if (data.Rows[0][0].ToString().Equals('0'))
    {
        ViewData["Contador"] = "1";
    }
    else

```

```

        {
            ViewData["Contador"] = (Convert.ToInt32(data.Rows[0][0].ToString()) +
1).ToString();
        }

        /* Lista de Tipo de Area*/
        query = "SELECT idTipo,tipo FROM snr.cat_TipoArea order by tipo";
        data = datos(query);

        for (int i = 0; i < data.Rows.Count; i++)
        {
            litipoArea.Add(new TipoArea()
            {
                idTipo = Convert.ToInt32(data.Rows[i][0].ToString()),
                tipo = data.Rows[i][1].ToString()
            });
        }
        ViewBag.TipoArea = litipoArea;

        /*Lista de color de techo*/
        query = "SELECT idTecho,color FROM snr.cat_Techo order by color";
        data = datos(query);

        for (int i = 0; i < data.Rows.Count; i++)
        {
            liTecho.Add(new Techo()
            {
                idTecho = Convert.ToInt32(data.Rows[i][0].ToString()),
                color = data.Rows[i][1].ToString()
            });
        }
        ViewBag.Techo = liTecho;

        /*Lista de color de pared*/
        query = "SELECT idPared,color FROM snr.cat_Pared order by color";
        data = datos(query);

        for (int i = 0; i < data.Rows.Count; i++)
        {
            liPared.Add(new Pared()
            {
                idPared = Convert.ToInt32(data.Rows[i][0].ToString()),
                color = data.Rows[i][1].ToString()
            });
        }
        ViewBag.Pared = liPared;

        /*Lista de color de piso*/
        query = "SELECT idPiso,color FROM snr.cat_Piso order by color";
        data = datos(query);

        for (int i = 0; i < data.Rows.Count; i++)
        {
            liPiso.Add(new Piso()
            {
                idPiso = Convert.ToInt32(data.Rows[i][0].ToString()),
                color = data.Rows[i][1].ToString()
            });
        }
        ViewBag.Piso = liPiso;

        /*Imagen de tipo de iluminacion*/
        return View(model);
    }
    else
    {
        return RedirectToAction("Login", "Registro");
    }
}

public ActionResult luminaria(string idArea)
{

```



```

if (Session["UsuarioNombre"] != null)
{
    List<TipoIlumina> litipoIlumina = new List<TipoIlumina>();
    List<TipoLampara> litipoLampara = new List<TipoLampara>();
    List<Mantenimiento> liMantenimiento = new List<Mantenimiento>();

    string query = "SELECT count(*) FROM snr.luminaria WHERE idArea = '" + idArea + "'";
    DataTable data = datos(query);

    if (data.Rows[0][0].ToString().Equals('0'))
    {
        ViewData["Contador"] = "1";
    }
    else
    {
        ViewData["Contador"] = (Convert.ToInt32(data.Rows[0][0].ToString()) +
1).ToString();
    }

    /* Lista de Tipo de Luminaria*/
    query = "SELECT idTipoIlu,tipo FROM snr.cat_TipoIlumina order by tipo";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        litipoIlumina.Add(new TipoIlumina()
        {
            idTipoIlu = Convert.ToInt32(data.Rows[i][0].ToString()),
            tipo = data.Rows[i][1].ToString()
        });
    }
    ViewBag.TipoIlumina = litipoIlumina;

    /*Lista de tipo de lampara*/
    query = "SELECT idLampara,tipo FROM snr.cat_tipolampara order by tipo";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        litipoLampara.Add(new TipoLampara()
        {
            idLampara = Convert.ToInt32(data.Rows[i][0].ToString()),
            tipo = data.Rows[i][1].ToString()
        });
    }
    ViewBag.TipoLampara = litipoLampara;

    /*Lista de estado de mantenimiento*/
    query = "SELECT idMante,estadoM FROM snr.cat_mantenimiento order by estadoM";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        liMantenimiento.Add(new Mantenimiento()
        {
            idMante = Convert.ToInt32(data.Rows[i][0].ToString()),
            estadoM = data.Rows[i][1].ToString()
        });
    }
    ViewBag.Mantenimiento = liMantenimiento;
    return View();
}
else
{
    return RedirectToAction("Login", "Registro");
}

}

[HttpPost]
public ActionResult luminaria(luminaria model)

```

```

{
    if (Session["UsuarioNombre"] != null)
    {
        List<TipoIlumina> litipoIlumina = new List<TipoIlumina>();
        List<TipoLampara> litipoLampara = new List<TipoLampara>();
        List<Mantenimiento> liMantenimiento = new List<Mantenimiento>();
        List<SelectListItem> litipoBalastro = new List<SelectListItem>();
        //float factorM = 0.0f;

        string query = "SELECT count(*) FROM snr.luminaria WHERE idArea = '" + model.idArea +
        """;
        DataTable data = datos(query);

        if (data.Rows[0][0].ToString().Equals('0'))
        {
            ViewData["Contador"] = "1";
        }
        else
        {
            ViewData["Contador"] = (Convert.ToInt32(data.Rows[0][0].ToString()) +
            1).ToString();
        }

        /* Lista de Tipo de Luminaria*/
        query = "SELECT idTipoIlu,tipo FROM snr.cat_TipoIlumina order by tipo";
        data = datos(query);

        for (int i = 0; i < data.Rows.Count; i++)
        {
            litipoIlumina.Add(new TipoIlumina()
            {
                idTipoIlu = Convert.ToInt32(data.Rows[i][0].ToString()),
                tipo = data.Rows[i][1].ToString()
            });
        }
        ViewBag.TipoIlumina = litipoIlumina;

        /*Lista de tipo de lampara*/
        query = "SELECT idLampara,tipo FROM snr.cat_TipoLampara order by tipo";
        data = datos(query);

        for (int i = 0; i < data.Rows.Count; i++)
        {
            litipoLampara.Add(new TipoLampara()
            {
                idLampara = Convert.ToInt32(data.Rows[i][0].ToString()),
                tipo = data.Rows[i][1].ToString()
            });
        }
        ViewBag.TipoLampara = litipoLampara;

        /*Lista de estado de mantenimiento*/
        query = "SELECT idMante,estadoM FROM snr.cat_Mantenimiento order by estadoM";
        data = datos(query);

        for (int i = 0; i < data.Rows.Count; i++)
        {
            liMantenimiento.Add(new Mantenimiento()
            {
                idMante = Convert.ToInt32(data.Rows[i][0].ToString()),
                estadoM = data.Rows[i][1].ToString(),
            });
        }
        ViewBag.Mantenimiento = liMantenimiento;

        /*Lista de Tipo de Balastro*/
        litipoBalastro.Add(new SelectListItem() { Text = "Electrónico", Value = "1" });
        litipoBalastro.Add(new SelectListItem() { Text = "Electromagnético", Value = "2" });
        ViewBag.tipoBalastro = litipoBalastro;

        /*Lumenes por tecnología, Factor de Balastro y potencia de linea dependiendo de tipo de
        lampara*/
    }
}

```

```

        if (model.tipoLampara == "Incandescente" || model.tipoLampara == "LED" ||
model.tipoLampara == "Fluorecente compacta" || model.tipoLampara == "Induccion")
        {
            model.lumTecnologia = model.totalLumina * model.lampxLumina * model.lumenLamp;
            model.deshabilita = true;
            /*Calculo de potencia de linea*/
            model.tipoBalastro = "";
            model.factorBalastro = 0.0f;
            model.potlinea = model.potLamp * model.lampxLumina;
        }
        else
        {
            model.deshabilita = false;
            model.lumTecnologia = model.totalLumina * model.lampxLumina * model.lumenLamp *
model.factorBalastro;
        }

        /* calculo de potencia por tecnología */
        model.potTecnologia = model.totalLumina * model.potlinea;

        /*Cálculo de eficacia por tecnología */
        model.efiTecnologia = model.lumTecnologia / model.potTecnologia;

        //Calculo de coeficiente de utilización :S este está difícil//
        query = "SELECT idArea,largoArea, anchoArea, altoArea, alturaTrabajo, alturaMontaje
FROM snr.iluminacion WHERE idArea =" + "'" + model.idArea + "'";
        data = datos(query);

        double largo = Convert.ToDouble(data.Rows[0][1].ToString());
        double ancho = Convert.ToDouble(data.Rows[0][2].ToString());
        double alto = Convert.ToDouble(data.Rows[0][3].ToString());
        double alTrabajo = Convert.ToDouble(data.Rows[0][4].ToString());
        double alMontaje = Convert.ToDouble(data.Rows[0][5].ToString());

        /*-----Calculo de cavidades-----*/
        double RCT = (5 * (alto - alMontaje) * (largo + ancho)) / (largo * ancho);
        double RCL = (5 * (alMontaje - alTrabajo) * (largo + ancho)) / (largo * ancho);
        double RCP = (5 * (alTrabajo) * (largo + ancho)) / (largo * ancho);
        /*-----Busqueda de reflexiones-----*/
        query = "SELECT idArea,colorPared, colorTecho, colorPiso FROM snr.iluminacion WHERE
idArea =" + "'" + model.idArea + "'";
        data = datos(query);
        string pared = data.Rows[0][1].ToString();
        string techo = data.Rows[0][2].ToString();
        string piso = data.Rows[0][3].ToString();

        query = "SELECT idPared,color, reflexion FROM snr.cat_Pared WHERE color =" + "'" +
pared + "'";
        data = datos(query);
        int rePared = Convert.ToInt32(data.Rows[0][2].ToString());

        query = "SELECT idTecho,color, reflexion FROM snr.cat_Techo WHERE color =" + "'" +
techo + "'";
        data = datos(query);
        int reTecho = Convert.ToInt32(data.Rows[0][2].ToString());

        query = "SELECT idPiso,color, reflexion FROM snr.cat_Piso WHERE color =" + "'" + piso
+ "'";
        data = datos(query);
        int rePiso = Convert.ToInt32(data.Rows[0][2].ToString());

        /*-----tratamiento de los datos (interpolacion)-----*/
        RCT = Math.Truncate(RCT * 10) / 10;
        RCL = Math.Truncate(RCL * 10) / 10;
        RCP = Math.Truncate(RCP * 10) / 10;

        RCT = interpolar(RCT, 0);
        RCL = interpolar(RCL, 1);
        RCP = interpolar(RCP, 0);

        /*-----Buscar reflectancia efectiva-----*/

```

```

        query = "SELECT idcat,cuefectiva FROM snr.cat_cuefectiva WHERE cavidad =" + "'" + RCT +
'" AND tepiso = "' + reTecho + "' AND pared = '" + rePared + "'";
        data = datos(query);
        double cuetecho = Convert.ToDouble(data.Rows[0][1].ToString());
        int efecTecho = Convert.ToInt32(interpolar(cuetecho, 2));

        /*-----Dependiendo del tipo de iluminacion checamos el CU-----*/
        double coeUtiliza = 0;
        if (model.tipoIlumina == "General difusa")
        {
            query = "SELECT idcat, general FROM snr.cat_cu WHERE rcl =" + "'" + RCL + "' AND
retecho = '" + efecTecho + "' AND repared = '" + rePared + "'";
            data = datos(query);
            coeUtiliza = Convert.ToDouble(data.Rows[0][1].ToString());
        }
        else if (model.tipoIlumina == "Directa indirecta")
        {
            query = "SELECT idcat, dindirecta FROM snr.cat_cu WHERE rcl =" + "'" + RCL + "' AND
retecho = '" + efecTecho + "' AND repared = '" + rePared + "'";
            data = datos(query);
            coeUtiliza = Convert.ToDouble(data.Rows[0][1].ToString());
        }
        else if (model.tipoIlumina == "Directa")
        {
            query = "SELECT idcat, directa FROM snr.cat_cu WHERE rcl =" + "'" + RCL + "' AND
retecho = '" + efecTecho + "' AND repared = '" + rePared + "'";
            data = datos(query);
            coeUtiliza = Convert.ToDouble(data.Rows[0][1].ToString());
        }
        else if (model.tipoIlumina == "Semi directa")
        {
            query = "SELECT idcat, semi FROM snr.cat_cu WHERE rcl =" + "'" + RCL + "' AND
retecho = '" + efecTecho + "' AND repared = '" + rePared + "'";
            data = datos(query);
            coeUtiliza = Convert.ToDouble(data.Rows[0][1].ToString());
        }
        }

        //Cálculo de lumenes utiles por tecnología --- hacer la consulta a la base de datos
para obtener coeu y factor Mante
        query = "SELECT idMante,estadoM, factorM FROM snr.cat_Mantenimiento WHERE estadoM =" +
        "'" + model.mantenimiento + "'";
        data = datos(query);
        //ver si es conveniente manejarlo en el modelo o no el factorM
        model.factorM = Convert.ToSingle(data.Rows[0][2].ToString());
        model.lumUtiles = model.lumTecnologia * coeUtiliza * model.factorM;

        /*Calculo de consumo energético por tecnología*/
        query = "SELECT idArea, horasDia, diasAnio FROM snr.iluminacion WHERE idArea =" + "'" +
model.idArea + "'";
        data = datos(query);
        int horas = Convert.ToInt32(data.Rows[0][1].ToString());
        int dias = Convert.ToInt32(data.Rows[0][2].ToString());
        model.consumoTecnologia = model.potTecnologia * horas * dias;

        if (ModelState.IsValid)
        {
            string query2 = "INSERT INTO snr.luminaria (idLumina,
idArea,numLumina,tipoIlumina,totalLumina,tipoLampara,lampxLamina,lumenLamp,potLamp,tipoBalastro,factorB
alastro, mantenimiento, reflexPared, potlinea, lumTecnologia, potTecnologia, efiTecnologia, lumUtiles,
factorM, consumoTecnologia) + VALUES (' + Session["idArea, idLumina"] + "','" +
            model.idArea + "','" + model.numLumina + "','" + model.tipoIlumina + "','" +
            model.totalLumina + "','" + model.tipoLampara + "','" +
            model.lampxLamina + "','" + model.lumenLamp + "','" + model.potLamp + "','" +
            model.tipoBalastro + "','" +
            model.factorBalastro + "','" + model.mantenimiento + "','" + model.reflexPared +
            "','" + model.potlinea + "','" + model.lumTecnologia + "','" + model.potTecnologia + "','" +
            model.efiTecnologia + "','" + model.lumUtiles + "','" + model.factorM + "','" + model.consumoTecnologia
            + "')";

            ejecutar(query2);
        }
    }
}

```

```

        if (resul)
        {
            resul = false;
            return RedirectToAction("resultadosIlu", "Iluminacion");
        }
        else if (lumi)
        {
            lumi = false;
            return RedirectToAction("luminaria", "Iluminacion");
        }
        else if (ilu)
        {
            ilu = false;
            return RedirectToAction("iluminacion", "Iluminacion");
        }
    }

    return View(model);
}
else
{
    return RedirectToAction("Login", "Registro");
}
}

[HttpPost]
public ActionResult resultadosIlu(iluminacion model)
{
    if (Session["UsuarioNombre"] != null)
    {
        /*calcula de la superficie*/
        float superficie = model.largoArea * model.anchoArea;

        string query = "SELECT idLumina, idArea , potTecnologia, lumUtiles FROM snr.luminaria
WHERE idArea =" + "'" + model.idArea + "'";
        DataTable data = datos(query);

        /*cálculo de luminarias por area --Consulta y suma de de base de datos*/
        model.lumxArea = data.Rows.Count;

        /*cálculo de potencia instalada por area*/
        for (int i = 0; i < data.Rows.Count; i++)
        {
            model.potxArea += Convert.ToDouble(data.Rows[i][2].ToString());
        }

        /*cálculo de lumenes utiles por area*/
        for (int i = 0; i < data.Rows.Count; i++)
        {
            model.lumUtilesArea += Convert.ToSingle(data.Rows[i][3].ToString());
        }

        /*Obtencion de los valores de las normas de iluminacion*/
        query = "SELECT idTipo, tipo, nom25, dpea FROM snr.cat_TipoArea WHERE tipo =" + "'" +
model.tipoArea + "'";

        data = datos(query);
        model.nom25 = Convert.ToInt32(data.Rows[0][2].ToString());
        model.dpea = Convert.ToSingle(data.Rows[0][3].ToString());

        /*calcula de nom25 calculado */
        model.iluCalculado = model.lumUtilesArea / superficie;

        /*calcula de dpa calculado */
        model.dpeaCalculado = model.potxArea / superficie;

        /*cálculo de consumo energético anual por área*/
        query = "SELECT idLumina, idArea, consumoTecnologia FROM snr.luminaria WHERE idArea ="
+ "'" + model.idArea + "'";
        data = datos(query);
    }
}

```

```

    for (int i = 0; i < data.Rows.Count; i++)
    {
        model.consumoEnergetico += Convert.ToSingle(data.Rows[i][2].ToString());
    }

    //Verificación de no errores en el modelo, guardado en la base de datos

    if (ModelState.IsValid)
    {
        string query2 = "INSERT INTO snr.iluminacion
(idArea,lumxArea,potxArea,lumUtilesArea,iluCalculado,dpeaCalculado,consumoEnergetico)" + " VALUES ('" +
Session["IdArea"] + "','" + model.lumxArea + "','" + model.potxArea + "','" + model.lumUtilesArea +
 "','" + model.iluCalculado + "','" +
        model.dpeaCalculado + "','" + model.consumoEnergetico + "')";
        ejecutar(query2);

        if (salir)
        {
            salir = false;
            return RedirectToAction("Inicio", "Sitio");
        }
        else if (ilu)
        {
            ilu = false;
            return RedirectToAction("iluminacion", "Iluminacion");
        }
        else if (resumen)
        {
            resumen = false;
            return RedirectToAction("resumenIlu", "Iluminacion");
        }
    }
    return View(model);
}
else
{
    return RedirectToAction("Login", "Registro");
}
}

[HttpPost]
public ActionResult resumenIlu(int idempresa)
{
    if (Session["UsuarioNombre"] != null)
    {
        List<iluminacion> table = new List<iluminacion>();

        string query = "SELECT numArea, nombreArea, tipoArea, largoArea, anchoArea, altoArea,
horasDia, diasAnio, lumxArea, potxArea, lumutilesarea, ilucalculado, dpeacalculado, consumoenergetico
FROM snr.iluminacion WHERE idempresa= " + idempresa + "'";
        DataTable data = datos(query);

        if (data.Rows.Count > 0)
        {
            table.Add(new iluminacion()
            {
                numArea = Convert.ToInt32(data.Rows[idempresa][0].ToString()),
                nombreArea = data.Rows[idempresa][1].ToString(),
                tipoArea = data.Rows[idempresa][2].ToString(),
                largoArea = Convert.ToSingle(data.Rows[idempresa][3].ToString()),
                anchoArea = Convert.ToSingle(data.Rows[idempresa][4].ToString()),
                altoArea = Convert.ToSingle(data.Rows[idempresa][5].ToString()),
                horasDia = Convert.ToInt32(data.Rows[idempresa][6].ToString()),
                diasAnio = Convert.ToInt32(data.Rows[idempresa][7].ToString()),
                lumxArea = Convert.ToInt32(data.Rows[idempresa][8].ToString()),
                potxArea = Convert.ToDouble(data.Rows[idempresa][9].ToString()),
                lumUtilesArea = Convert.ToDouble(data.Rows[idempresa][10].ToString()),
                iluCalculado = Convert.ToDouble(data.Rows[idempresa][11].ToString()),
                dpeaCalculado = Convert.ToDouble(data.Rows[idempresa][12].ToString()),
                consumoEnergetico = Convert.ToDouble(data.Rows[idempresa][13].ToString()),
            });
        }
    }
}

```

```

    });
    ViewBag.areaEmpresa = data.Rows[idempresa][0].ToString();
  }
  return View("resumenIlu", table);
}
else
{
  return RedirectToAction("Login", "Registro");
}
}

```

### A.3.5. Compresor

- Pseudocódigo

```

programa compresor
  mostrar vista compresor
  obtener tipoCom, tipoTrans, potNominal, presionMax,
  flujo, presionOp, temp, valida, tempSalida,
  horasDia, diasAnio, costoEner, enfriamiento,
  cambioA, limpiezaC, ajusteTrans, ubicacionC,
  aniosOp, flujoOp, fugas, efiNomMotor, efiRealMotor,
  potDemanda

  Si tempSalida = "no" entonces (No conoce temperatura
  de salida)
    obtener tempDescarga, factor, factorEnfriamiento,
    factorcambioA, factorUbicacion de base de datos
    factores <- factor + factorEnfriamiento +
    factorcambioA + factorUbicacion
    tempSalida <- tempDescarga + factores
  fin del si

  Si flujoOp > 0 entonces (Conoce flujo de operación)
    flujoEstimado <- flujoOp
  Si no entonces

    Si flujo > 0 entonces
      flujoEstimado <- flujo
    Si no entonces
      Según tipoCompresor obtener flujo de BD
    fin del Si

  fin del Si

  presionEnt <- (1.01 * (1 - (0.0000225557 * altura)) *
  5.256) * 100
  presionSal <- presionEnt + presionOp
  temp <- temp + 273.15
  lnTemp <- LogNatural(model.temp / model.tempSalida)
  lnPres <- LogNatural(presionSal / presionEnt)
  n <- 1 / (1 - lnTemp / lnPres)
  trabajoIso <- 0.287 * temp * lnPres
  trabajoPoli <- ((n * 0.287 * temp) / (n - 1)) *
  (presionSal / presionEnt) ^ ((n - 1) / n) - 1
  efiCompresion <- trabajoIso / trabajoPoli

  Si efiRealMotor < 100 y efiRealMotor > 1 entonces
    efiRealMotor <- efiRealMotor / 100

```

```

fin del Si

Si efiNomMotor < 100 y efiNomMotor > 1 entonces
    efiRealMotor <- efiRealMotor / 100
fin del Si

Si tipoTrans = "engranés" entonces
    obtener eficienciaTrans en tabla Engranés de
    base de datos
si no entonces
    obtener eficienciaTrans en tabla Banda de
    base de datos
fin del Si

potSalMotor <- efiRealMotor * potDemanda
potSalTrans <- eficienciaTrans * potSalMotor
potSalCompresor <- efiCompresion * potSalTrans
(eficiencias Nominales)
efiNMotor <- potDemanda - potSalMotor
efiNTrans <- potSalMotor - potSalTrans
efiNCompresor <- potSalTrans - potSalCompresor
(eficiencia compuesta)
motoCom <- efiTrans * efiCompresion * efiRealMotor
coeMantenimiento <- (0.9 - efiRealMotor) +
(0.93 - efiTrans) + (0.84 - efiCompresion)
horasAnio <- horasDia * model.diasAnio
EnerAnual <- potDemanda * horasAnio
costoAnual <- EnerAnual * costoEner
ahorroEnerMan <- EnerAnual * coeMantenimiento
ahorroEcoMan <- ahorroEnerMan * costoEner
    perdidasfugas = model.fugas * 0.2
flujoCom <- flujoestimado * 60000
aireFugado <- (perdidasfugas / flujoCom) * 100
disminucion <- (aireFugado - 10)
disEner <- EnerAnual * disminucion / 100
disAhorro <- disminucion * costoAnual / 100

```

**fin del programa compresor**

- Código

```

namespace sitioSener.Controllers
{
    public class CompresorController : Controller
    {
        public ActionResult Compresor()
        {
            Session["UsuarioNombre"] = "Prueba";
            Session["encuesta"] = "true";
            Session["editarPerfil"] = "true";
            Session["inicioACT"] = "noCurrent";
            Session["empresaACT"] = "current";

            if (Session["UsuarioNombre"] != null)
            {
                List<periodoCompresor> liPeriodo = new List<periodoCompresor>();
                List<ubicaCompresor> liubicacion = new List<ubicaCompresor>();
                List<SelectListItem> litipoCompresor = new List<SelectListItem>();
                List<SelectListItem> litipoTrans = new List<SelectListItem>();
            }
        }
    }
}

```



```

List<SelectListItem> lipotNominal = new List<SelectListItem>();
List<SelectListItem> lienfriamiento = new List<SelectListItem>();
List<SelectListItem> lianosOp = new List<SelectListItem>();
List<SelectListItem> livalida = new List<SelectListItem>();

/*Lista de frecuencia del mantenimiento*/
string query = "SELECT idperiodo,meses FROM snr.periodoCompresor order by meses";
DataTable data = datos(query);

for (int i = 0; i < data.Rows.Count; i++)
{
    liPeriodo.Add(new periodoCompresor()
    {
        idperiodo = Convert.ToInt32(data.Rows[i][0].ToString()),
        meses = data.Rows[i][1].ToString()
    });
}
ViewBag.liPeriodo = liPeriodo;

/*Lista de Ubicacion del compresor*/
query = "SELECT idubica,ubicacion FROM snr.ubicaCompresor order by ubicacion";
data = datos(query);

for (int i = 0; i < data.Rows.Count; i++)
{
    liubicacion.Add(new ubicaCompresor()
    {
        idubica = Convert.ToInt32(data.Rows[i][0].ToString()),
        ubicacion = data.Rows[i][1].ToString()
    });
}
ViewBag.liUbicacion = liubicacion;

/*Lista de Tipo de Compresor*/
litipoCompresor.Add(new SelectListItem() { Text = "Pistones", Value = "0" });
litipoCompresor.Add(new SelectListItem() { Text = "Tornillo", Value = "1" });
ViewBag.litipoCompresor = litipoCompresor;

/*Lista de Tipo de Transmisor*/
litipoTrans.Add(new SelectListItem() { Text = "Banda y poleas", Value = "0" });
litipoTrans.Add(new SelectListItem() { Text = "Engranajes", Value = "1" });
ViewBag.litipoTrans = litipoTrans;

/*Lista de Tipo de Potencia nominal*/
lipotNominal.Add(new SelectListItem() { Text = "5.5", Value = "5.5" });
lipotNominal.Add(new SelectListItem() { Text = "7.5", Value = "7.5" });
lipotNominal.Add(new SelectListItem() { Text = "10", Value = "10" });
lipotNominal.Add(new SelectListItem() { Text = "20", Value = "20" });
lipotNominal.Add(new SelectListItem() { Text = "25", Value = "25" });
ViewBag.lipotNominal = lipotNominal;

/*Lista de Tipo de Enfriamiento*/
lienfriamiento.Add(new SelectListItem() { Text = "Aire (Ventilador integrado)", Value =
"0" });
lienfriamiento.Add(new SelectListItem() { Text = "Agua", Value = "1" });
ViewBag.lienfriamiento = lienfriamiento;

/*Lista de Años de operacion*/
lianosOp.Add(new SelectListItem() { Text = "1 - 5 ", Value = "0" });
lianosOp.Add(new SelectListItem() { Text = "5 - 10", Value = "1" });
lianosOp.Add(new SelectListItem() { Text = "10 - 15", Value = "2" });
lianosOp.Add(new SelectListItem() { Text = "15 años o más", Value = "3" });
ViewBag.lianosOp = lianosOp;

/*Lista de validacion*/
livalida.Add(new SelectListItem() { Text = "Si", Value = "0" });
livalida.Add(new SelectListItem() { Text = "No", Value = "1" });
ViewBag.livalida = livalida;

return View();
}

```

```

else
{
    return RedirectToAction("Login", "Registro");
}
}

[HttpPost]
public ActionResult Compresor(Compresor model)
{
    if (Session["UsuarioNombre"] != null)
    {
        /*
        if (ModelState.IsValid)
        {
            string query2 = "INSERT INTO snr.empresa_info
(iduser,nombre,rfc,estado,municipio,calle,colonia," +
                "codigoPostal,telefono,paginaWeb,ramo) VALUES (' + Session["IdUser"] + ',' +
                model.nombre + ',' + model.rfc + ',' + model.estado + ',' +
                model.calle + ',' + model.colonia + ',' + model.codigoPostal + ',' +
+
                model.paginaWeb + ',' + model.ramo + ')";

            ejecutar(query);

            return RedirectToAction("Inicio", "Sitio");
        }*/

List<periodoCompresor> liPeriodo = new List<periodoCompresor>();
List<ubicaCompresor> liubicacion = new List<ubicaCompresor>();
List<converCompresor> liPresion = new List<converCompresor>();
List<converCompresor> liFlujo = new List<converCompresor>();
List<converCompresor> liPotencia = new List<converCompresor>();
List<SelectListItem> litipoCompresor = new List<SelectListItem>();
List<SelectListItem> litipoTrans = new List<SelectListItem>();
List<SelectListItem> lipotNominal = new List<SelectListItem>();
List<SelectListItem> lienfriamiento = new List<SelectListItem>();
List<SelectListItem> lianiosOp = new List<SelectListItem>();
List<SelectListItem> livalida = new List<SelectListItem>();

/*Lista de frecuencia del mantenimiento*/
string query = "SELECT idperiodo,meses FROM snr.periodoCompresor order by meses";
DataTable data = datos(query);

for (int i = 0; i < data.Rows.Count; i++)
{
    liPeriodo.Add(new periodoCompresor()
    {
        idperiodo = Convert.ToInt32(data.Rows[i][0].ToString()),
        meses = data.Rows[i][1].ToString()
    });
}
ViewBag.liPeriodo = liPeriodo;

/*Lista de Ubicacion del compresor*/
query = "SELECT idubica,ubicacion FROM snr.ubicaCompresor order by ubicacion";
data = datos(query);

for (int i = 0; i < data.Rows.Count; i++)
{
    liubicacion.Add(new ubicaCompresor()
    {
        idubica = Convert.ToInt32(data.Rows[i][0].ToString()),
        ubicacion = data.Rows[i][1].ToString()
    });
}
ViewBag.liUbicacion = liubicacion;

/*Lista de unidades de presion*/
query = "SELECT idconver,unidad FROM snr.converCompresor WHERE carac = 'Presion'";
data = datos(query);

for (int i = 0; i < data.Rows.Count; i++)

```

```

    {
        liPresion.Add(new converCompresor()
        {
            idconver = Convert.ToInt32(data.Rows[i][0].ToString()),
            unidad = data.Rows[i][1].ToString()
        });
    }
    ViewBag.liPresion = liPresion;

    /*Lista de unidades de Flujo*/
    query = "SELECT idconver,unidad FROM snr.converCompresor WHERE carac = 'Caudal'";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        liFlujo.Add(new converCompresor()
        {
            idconver = Convert.ToInt32(data.Rows[i][0].ToString()),
            unidad = data.Rows[i][1].ToString()
        });
    }
    ViewBag.liFlujo = liFlujo;

    /*Lista de unidades de Potencia*/
    query = "SELECT idconver,unidad FROM snr.converCompresor WHERE carac = 'Potencia'";
    data = datos(query);

    for (int i = 0; i < data.Rows.Count; i++)
    {
        liPotencia.Add(new converCompresor()
        {
            idconver = Convert.ToInt32(data.Rows[i][0].ToString()),
            unidad = data.Rows[i][1].ToString()
        });
    }
    ViewBag.liPotencia = liPotencia;

    /*Lista de Tipo de Compresor*/
    litipoCompresor.Add(new SelectListItem() { Text = "Pistones", Value = "0" });
    litipoCompresor.Add(new SelectListItem() { Text = "Tornillos", Value = "1" });
    ViewBag.litipoCompresor = litipoCompresor;

    /*Lista de Tipo de Transmisor*/
    litipoTrans.Add(new SelectListItem() { Text = "Banda y poleas", Value = "0" });
    litipoTrans.Add(new SelectListItem() { Text = "Engranajes", Value = "1" });
    ViewBag.litipoTrans = litipoTrans;

    /*Lista de Tipo de Potencia nominal*/
    lipotNominal.Add(new SelectListItem() { Text = "5.5", Value = "5.5" });
    lipotNominal.Add(new SelectListItem() { Text = "7.5", Value = "7.5" });
    lipotNominal.Add(new SelectListItem() { Text = "10", Value = "10" });
    lipotNominal.Add(new SelectListItem() { Text = "20", Value = "20" });
    lipotNominal.Add(new SelectListItem() { Text = "25", Value = "25" });
    ViewBag.lipotNominal = lipotNominal;

    /*Lista de Tipo de Enfriamiento*/
    lienfriamiento.Add(new SelectListItem() { Text = "Aire (Ventilador integrado)", Value =
"0" });
    lienfriamiento.Add(new SelectListItem() { Text = "Agua", Value = "1" });
    ViewBag.lienfriamiento = lienfriamiento;

    /*Lista de Años de operacion*/
    lianosOp.Add(new SelectListItem() { Text = "1 - 5 años", Value = "0" });
    lianosOp.Add(new SelectListItem() { Text = "5 - 10 años", Value = "1" });
    lianosOp.Add(new SelectListItem() { Text = "10 - 15 años", Value = "2" });
    lianosOp.Add(new SelectListItem() { Text = "15 años o más", Value = "3" });
    ViewBag.lianosOp = lianosOp;

    /*Lista de validacion*/
    livalida.Add(new SelectListItem() { Text = "Si", Value = "0" });
    livalida.Add(new SelectListItem() { Text = "No", Value = "1" });
    ViewBag.livalida = livalida;

```

```

//Declaracion de variables
float sumaFactores = 0.0f;
float tempdescarga = 0.0f;
float flujoestimado = 0.0f;
float alturaCDMX = 2239.00F;
double presionEnt = 0.0;
double presionSal = 0.0;
double efiTrans = 0.0;

//Conversiones de unidades
query = "SELECT idconver, oficial FROM snr.cat_convercompresor WHERE carac = 'Presion'
AND unidad = '" + model.uniPresionMax + "'";
data = datos(query);
model.presionMax = model.presionMax * Convert.ToSingle(data.Rows[0][1].ToString());

query = "SELECT idconver, oficial FROM snr.cat_convercompresor WHERE carac = 'Presion'
AND unidad = '" + model.uniPresionOp + "'";
data = datos(query);
model.presionOp = model.presionOp * Convert.ToSingle(data.Rows[0][1].ToString());

query = "SELECT idconver, oficial FROM snr.cat_convercompresor WHERE carac = 'Potencia'
AND unidad = '" + model.uniPotencia + "'";
data = datos(query);
model.potNominal = model.potNominal * Convert.ToSingle(data.Rows[0][1].ToString());

query = "SELECT idconver, oficial FROM snr.cat_convercompresor WHERE carac = 'Caudal'
AND unidad = '" + model.uniFlujo + "'";
data = datos(query);
model.flujo = model.flujo * Convert.ToSingle(data.Rows[0][1].ToString());

query = "SELECT idconver, oficial FROM snr.cat_convercompresor WHERE carac = 'Caudal'
AND unidad = '" + model.uniFlujoOp + "'";
data = datos(query);
model.flujoOp = model.flujoOp * Convert.ToSingle(data.Rows[0][1].ToString());

//Deshabilitacion de campo de vista *Conoce la temperatura de salida*
if (model.valida == "Si")
{
    model.deshabilita = false;
}
else
{
    model.deshabilita = true;
    query = "SELECT idtipo, tempdes FROM snr.cat_tipofactor WHERE carac = 'aniosOp' AND
descripcion = '" + model.aniosOp + "'";
data = datos(query);
tempdescarga = Convert.ToSingle(data.Rows[0][1].ToString());
query = "SELECT idtipo, factor FROM snr.cat_tipofactor WHERE carac = 'aniosOp' AND
descripcion = '" + model.aniosOp + "'";
data = datos(query);
sumaFactores = sumaFactores + Convert.ToSingle(data.Rows[0][1].ToString());
query = "SELECT idtipo, factor FROM snr.cat_tipofactor WHERE carac = 'enfriamiento'
AND descripcion = '" + model.enfriamiento + "'";
data = datos(query);
sumaFactores = sumaFactores + Convert.ToSingle(data.Rows[0][1].ToString());
query = "SELECT idtipo, factor FROM snr.cat_tipofactor WHERE carac = 'cambioA' AND
descripcion = '" + model.cambioA + "'";
data = datos(query);
sumaFactores = sumaFactores + Convert.ToSingle(data.Rows[0][1].ToString()) * 2;
query = "SELECT idubica, factor FROM snr.cat_ubicaCompresor WHERE ubicacion = '" +
model.ubicacionC + "'";
data = datos(query);
sumaFactores = sumaFactores + Convert.ToSingle(data.Rows[0][1].ToString());

    model.tempSalida = tempdescarga + sumaFactores;
}

//Flujo o caudal estimado
if (model.flujoOp > 0)
{
    flujoestimado = model.flujoOp;
}

```

```

    }
    else
    {
        if (model.flujo > 0)
        {
            flujoestimado = model.flujo;
        }
        else
        {
            query = "SELECT idtipo, caudal FROM snr.cat_tipocompresor WHERE tipo ='" +
model.tipoCom + "' AND pot = '" + model.potNominal + "'";
            data = datos(query);
            flujoestimado = Convert.ToSingle(data.Rows[0][1].ToString());
        }
    }

    //Calculo de presiones
    presionEnt = (1.01 * (1 - (0.0000225557 * alturaCDMX)) * 5.256) * 100;
    presionSal = presionEnt + model.presionOp;
    model.temp = model.temp + 273.15;

    //Logaritmos Naturales
    double lnTemp = Math.Log(model.temp / model.tempSalida);
    double lnPres = Math.Log(presionSal / presionEnt);
    double n = 1 / (1 - lnTemp / lnPres);
    double trabajoIso = 0.287 * model.temp * lnPres;
    double trabajoPoli = ((n * 0.287 * model.temp) / (n - 1)) * (Math.Pow((presionSal /
presionEnt), ((n - 1) / n)) - 1);
    model.efiCompresion = trabajoIso / trabajoPoli;

    //Calculo de eficiencias
    if (model.efiRealMotor < 100 && model.efiRealMotor > 1)
    {
        model.efiRealMotor = model.efiRealMotor / 100;
    }

    if (model.efiNomMotor < 100 && model.efiNomMotor > 1)
    {
        model.efiNomMotor = model.efiNomMotor / 100;
    }

    //Obtenemos eficiencia del transmisor
    if (model.tipoTrans == "Engranajes")
    {
        query = query = "SELECT idtipo, efiengran FROM snr.cat_tipofactor WHERE carac
='ajusteTrans' AND descripcion='Engranajes';
        data = datos(query);
        model.efiTrans = Convert.ToDouble(data.Rows[0][1].ToString());
    }
    else
    {
        query = query = "SELECT idtipo, efiBandas FROM snr.cat_tipofactor WHERE carac
='ajusteTrans' AND descripcion='Banda y poleas';
        data = datos(query);
        model.efiTrans = Convert.ToDouble(data.Rows[0][1].ToString());
    }

    //Obtenemos potencias de salida
    double potSalMotor = model.efiRealMotor * model.potDemanda;
    double potSalTrans = efiTrans * potSalMotor;
    double potSalCompresor = model.efiCompresion * potSalTrans;

    //Obtenemos eficiencias Nominales
    double efiNMotor = model.potDemanda - potSalMotor;
    double efiNTrans = potSalMotor - potSalTrans;
    double efiNCompresor = potSalTrans - potSalCompresor;

    model.motoCom = model.efiTrans * model.efiCompresion * model.efiRealMotor;

    //Obtenemos coeficiente de mantenimiento

```

```

        double coeMantenimiento = (0.9 - model.efiRealMotor) + (0.93 - efiTrans) + (0.84 -
model.efiCompresion);

        //Calculos finales y resultados
        /*Potencia demandada*/
        int horasAnio = model.horasDia * model.diasAnio;
        model.EnerAnual = model.potDemanda * horasAnio;
        /*Costo ponderado de energía*/
        model.costoAnual = model.EnerAnual * model.costoEner;
        model.ahorroEnerMan = model.EnerAnual * coeMantenimiento;
        model.ahorroEcoMan = model.ahorroEnerMan * model.costoEner;

        //Resultados teniendo en cuenta las fugas
        double perdidasfugas = model.fugas * 0.2;
        double flujoCom = flujoestimado * 60000;
        double aireFugado = (perdidasfugas / flujoCom) * 100;
        model.disminucion = (aireFugado - 10);
        model.disEner = model.EnerAnual * model.disminucion / 100;
        model.disAhorro = model.disminucion * model.costoAnual / 100;

        return View(model);
    }
    else
    {
        return RedirectToAction("Login", "Registro");
    }
} // httpost

```

### A.3.6. Horno de flama

- Pseudocódigo

```

programa Horno
mostrar vista de horno
obtener tipo, year, aplicacion, consumo, tiempo,
dias, gasesComb, gasesChimenea, tAmbiente, tAire,
tProdIni, tProdFin, tInterior, masa, O2, CO2, CO
alfa <- CO2 / 100
teta <- CO / 1000000
gama <- O2 / 100
N2 <- 1 - alfa - teta - gama
obtener cp del producto de base de datos

Si combustible = "gas LP" entonces
x <- (alfa + teta) / (0.664 * 3 + 0.336 * 4)
y <- (x * (0.664 * 8 + 0.336 * 10) / 2
g <- N2 / 3.762
F22 <- (0.0069 * (alfa + 0.5 * teta + 0.5 * y +
gama) / (x * 0.336)) * 32
H22 <- ((0.0069) + * N2) / (x * 0.336)) * 28
E17 <- ((0.0069 * (0.664 * 3 + 0.336 * 4 +
0.5 * (0.664 * 8 + 0.336 * 10) / 2 )) / 0.336)
* 32
G17 <- (0.0069 * 3.76 * ((0.664 * 3 + 0.336 * 4)
+ 0.5 * ((0.664 * 8 + 0.336 * 10) / 2)) /
0.336) * 28
excesoAire <- (((F22 + H22) / (E17 + G17)) - 1) *
100
Qalambre <- masa * cp * (Talambre - Tambiente)
Qperdidas <- 59545.06 * dias * tiempo * 120 * 0.001
aireComb <- F22 + H22

```

```

Qaire <- (aireComb * 1.005 * (TaireIn - Tambiente))
* (consumo / dias) * 0.737
Qcomb <- 45982.16 * (consumo / dias) * 0.737
Qgc = ((0.0069 * alfa) * 44 + (0.0069 * teta) *
28 + (0.0069 * N2) * 28 + (0.0069 * y) * 18 +
(0.0069 * gama) * 32) / (x * 0.339) * 1.0753 *
(Tgases - Tambiente) * (consumo / dias) * 0.737
Qttotal <- Qperdidas + Qalambre + Qaire + Qgc

Si tipo = "conventivo" entonces
    EfHorno <- ((Qalambre + Qperdidas + Qaire) /
    Qcomb) * 100
Si no entonces
    EfHorno <- ((Qalambre - Qperdidas) / Qcomb) *
    100
fin del Si

Si no si combustible = "Gas Natural" entonces
x <- (alfa + teta) / (0.937 + 0.051 * 2 + 0.012 *
3)
y <- (0.937 * 4 * x + 0.051 * 6 * x + 0.012 * 8 *
x) / 2
H22 <- ((0.001 * (alfa + 0.5 * teta + 0.5 * y +
gama)) / (x * 0.012)) * 32
J22 <- ((0.001 * N2) / (x * 0.012)) * 28
G17 <- (0.001 * (2 * (0.937 + 0.051 * 2 + 0.012 *
3) + (0.937 * 4 + 0.051 * 6 + 0.012 * 8) / 2) / 2)
/ 0.012) * 32
I17 <- (0.001 * 3.76 * ((2 * (0.937 + 0.051 * 2 +
0.012 * 3) + (0.937 * 4 + 0.051 * 6 + 0.012 * 8) /
2)) / 2) / 0.012) * 28
excesoAire <- (((H22 + J22) / (G17 + I17)) - 1) *
100
Qalambre <- masa * cp * (Talambre - Tambiente)
Qperdidas <- masa * cp * (TprodIni - Tambiente)
Qcomb <- 45000 * (consumo / dias) * 0.737
perdidasGC <- 0.02
perdidasAis <- 0.03
EfHorno <- ((Qalambre - Qperdidas) / Qcomb) * 100
si no entonces
excesoAire <- 0
perdidasGC <- 46
perdidasAis <- 0
EfHorno <- 0
fin del Si

```

**fin del programa Horno**

- Código

```

namespace sitioSener.Controllers
{
    public class HornosController : Controller
    {
        // GET: Hornos
        public ActionResult HornoMain()
        {

```

```

Session["UsuarioNombre"] = "Prueba";
Session["encuesta"] = "true";
Session["editarPerfil"] = "true";
Session["inicioACT"] = "noCurrent";
Session["empresaACT"] = "current";

List<Catalogo> liCatUnidadProd = new List<Catalogo>();

liCatUnidadProd.Add(new Catalogo() { idCat = "Masica", nombre = "Másica" });
liCatUnidadProd.Add(new Catalogo() { idCat = "Volumetrica", nombre = "Volumetrica" });

ViewBag.catUnidadProd = liCatUnidadProd;

List<Catalogo> liCatTipo = new List<Catalogo>();

liCatTipo.Add(new Catalogo() { idCat = "Convectivo", nombre = "Convectivo" });
liCatTipo.Add(new Catalogo() { idCat = "Rotatorio", nombre = "Rotatorio" });
liCatTipo.Add(new Catalogo() { idCat = "Columpio", nombre = "Columpio" });
liCatTipo.Add(new Catalogo() { idCat = "Tunel", nombre = "Túnel" });

ViewBag.catTipo = liCatTipo;

List<Catalogo> li = new List<Catalogo>();

string query = "SELECT idpro, productos FROM snr.cat_prodHornos order by productos";

DataTable dtCat = datos(query);

for (int i = 0; i < dtCat.Rows.Count; i++)
{
    li.Add(new Catalogo()
    {
        idCat = dtCat.Rows[i][0].ToString(),
        nombre = dtCat.Rows[i][1].ToString()
    });
}

ViewBag.catProducto = li;

List<Catalogo> liCatComb = new List<Catalogo>();

liCatComb.Add(new Catalogo() { idCat = "GasNatural", nombre = "Gas Natural" });
liCatComb.Add(new Catalogo() { idCat = "GasLP", nombre = "Gas LP" });

ViewBag.catComb = liCatComb;

List<Unidades> liUnidadesTemp = new List<Unidades>();

liUnidadesTemp.Add(new Unidades() { idUnidad = "C", nombre = "°C" });
liUnidadesTemp.Add(new Unidades() { idUnidad = "K", nombre = "K" });
liUnidadesTemp.Add(new Unidades() { idUnidad = "F", nombre = "°F" });

ViewBag.unidadTemp = liUnidadesTemp;

return View();
}

// POST: HornosMain
[HttpPost]
public ActionResult HornoMain(HornosDatosIn m)
{
    string query;
    if (ModelState.IsValid)
    {
        decimal N2, alfa, teta, gama, x, y, g, F22, H22, E17, G17, excesoAire, J22, I17;

        alfa = m.CO2 / 100;
        teta = m.CO / 1000000;
        gama = m.O2 / 100;

        N2 = 1 - alfa - teta - gama; // N2 corresponde a resultado de gases N2
    }
}

```



```

    decimal Talambre, Tambiente, TaireIn, Tgases, TprodIni, EfHorno, Qalambre, cp,
    Qperdidas, Qaire, aireComb, Qcomb, Qgc,
    perdidasGC, perdidasAis, Qtotal;

    if (m.unidadTPF == "K") { Talambre = m.tProdFin - Convert.ToDecimal(273.15); }
    else if (m.unidadTPF == "F" ) { Talambre = (m.tProdFin - 32) / Convert.ToDecimal(5 /
9); }

    else { Talambre = m.tProdFin; }

    if (m.unidadTA == "K") { Tambiente = m.tAmbiente - Convert.ToDecimal(273.15); }
    else if (m.unidadTA == "F") { Tambiente = (m.tAmbiente - 32) / Convert.ToDecimal(5 /
9); }

    else { Tambiente = m.tAmbiente; }

    if (m.unidadTAr == "K") { TaireIn = m.tAire - Convert.ToDecimal(273.15); }
    else if (m.unidadTAr == "F") { TaireIn = (m.tAire - 32) / Convert.ToDecimal(5 / 9); }
    else { TaireIn = m.tAire; }

    if (m.unidadGC == "K") { Tgases = m.gasesComb - Convert.ToDecimal(273.15); }
    else if (m.unidadGC == "F") { Tgases = (m.gasesComb - 32) / Convert.ToDecimal(5 / 9); }
    else { Tgases = m.gasesComb; }

    if (m.unidadTPI == "K") { TprodIni = m.tProdIni - Convert.ToDecimal(273.15); }
    else if (m.unidadTPI == "F") { TprodIni = (m.tProdIni - 32) / Convert.ToDecimal(5 / 9);
}

    else { TprodIni = m.tProdIni; }

    query = "SELECT idpro, cp FROM snr.cat_prodHornos WHERE idpro = '" + m.producto + "'";
    DataTable dtAux = datos(query);

    cp = Convert.ToDecimal(dtAux.Rows[0][1].ToString());

    if (m.combustible.Equals("GasLP"))
    {
        x = (alfa + teta) / (Convert.ToDecimal(0.664 * 3 + 0.336 * 4));
        y = (x * Convert.ToDecimal(0.664 * 8 + 0.336 * 10)) / 2;
        g = N2 / Convert.ToDecimal(3.762);

        F22 = ((Convert.ToDecimal(0.0069) * (alfa + Convert.ToDecimal(0.5) * teta +
Convert.ToDecimal(0.5) * y + gama)) / (x * Convert.ToDecimal(0.336))) * 32;
        H22 = ((Convert.ToDecimal(0.0069) * N2) / (x * Convert.ToDecimal(0.336))) * 28;
        E17 = ((Convert.ToDecimal(0.0069) * (Convert.ToDecimal(0.664 * 3 + 0.336 * 4) +
Convert.ToDecimal(0.5 * (0.664 * 8 + 0.336 * 10) / 2 ))) /
Convert.ToDecimal(0.336)) * 32;
        G17 = Convert.ToDecimal(((0.0069 * 3.76 * ((0.664 * 3 + 0.336 * 4) + 0.5 * ((0.664 *
8 + 0.336 * 10) / 2)))) / 0.336) * 28;

        excesoAire = (((F22 + H22) / (E17 + G17)) - 1) * 100;

        Qalambre = m.masa * cp * (Talambre - Tambiente);
        Qperdidas = Convert.ToDecimal(59545.06) * m.dias * m.tiempo * 120 *
Convert.ToDecimal(0.001);
        aireComb = F22 + H22;
        Qaire = (aireComb * Convert.ToDecimal(1.005) * (TaireIn - Tambiente)) * (m.consumo
/ m.dias) * Convert.ToDecimal(0.737);
        Qcomb = Convert.ToDecimal(45982.16) * (m.consumo / m.dias) *
Convert.ToDecimal(0.737);
        Qgc = ((Convert.ToDecimal(0.0069) * alfa) * 44 + (Convert.ToDecimal(0.0069) * teta)
* 28 + (Convert.ToDecimal(0.0069) * N2) * 28 +
        (Convert.ToDecimal(0.0069) * y) * 18 + (Convert.ToDecimal(0.0069) * gama) * 32)
/ (x * Convert.ToDecimal(0.339)) * Convert.ToDecimal(1.0753) *
        (Tgases - Tambiente) * (m.consumo / m.dias) * Convert.ToDecimal(0.737);
        Qtotal = Qperdidas + Qalambre + Qaire + Qgc;

        perdidasGC = Convert.ToDecimal(0.46); // salida para graficas
        perdidasAis = Qperdidas / Qtotal; // salida para graficas

        if (m.tipo.Equals("Convectivo"))
        {
            EfHorno = ((Qalambre + Qperdidas + Qaire) / Qcomb) * 100; //Eficiencia del
horno salida en resultados
        }
    }

```

```

        else
        {
            EfHorno = ((Qalambre - Qperdidas) / Qcomb) * 100;
        }
    }
    else if(m.combustible.Equals("GasNatural"))
    {
        x = (alfa + teta) / (Convert.ToDecimal(0.937 + 0.051 * 2 + 0.012 * 3));
        y = ((Convert.ToDecimal(0.937 * 4) * x + Convert.ToDecimal(0.051 * 6) * x +
Convert.ToDecimal(0.012 * 8) * x)) / 2;
        //g = N2 / Convert.ToDecimal(3.762);

        H22 = ((Convert.ToDecimal(0.001) * (alfa + Convert.ToDecimal(0.5) * teta +
Convert.ToDecimal(0.5) * y + gama)) / (x * Convert.ToDecimal(0.012))) * 32;
        J22 = ((Convert.ToDecimal(0.001) * N2) / (x * Convert.ToDecimal(0.012))) * 28;
        G17 = ((Convert.ToDecimal(0.001) * (Convert.ToDecimal(2 * (0.937 + 0.051 * 2 +
0.012 * 3) +
            (0.937 * 4 + 0.051 * 6 + 0.012 * 8) / 2)) / 2) / Convert.ToDecimal(0.012)) *
32;
        I17 = Convert.ToDecimal(((0.001 * 3.76 * ((2 * (0.937 + 0.051 * 2 + 0.012 * 3) +
(0.937 * 4 + 0.051 * 6 + 0.012 * 8) / 2)) / 2) / 0.012) * 28;

        excesoAire = (((H22 + J22) / (G17 + I17)) - 1) * 100;

        Qalambre = m.masa * cp * (Talambre - Tambiente);
        Qperdidas = m.masa * cp * (TprodIni - Tambiente);
        Qcomb = Convert.ToDecimal(45000) * (m.consumo / m.dias) * Convert.ToDecimal(0.737);

        perdidasGC = Convert.ToDecimal(0.02); // salida para graficas
        perdidasAis = Convert.ToDecimal(0.03); // salida para graficas

        EfHorno = ((Qalambre - Qperdidas) / Qcomb) * 100;
    }
    else
    {
        excesoAire = 0;
        perdidasGC = 46; // salida para graficas
        perdidasAis = 0; // salida para graficas
        EfHorno = 0;
    }
}

return RedirectToAction("HornoDiag", "Hornos", new { N2 = N2, aire = excesoAire,
eficiencia = EfHorno, perdGC = perdidasGC, perdAis = perdidasAis,
comb = m.combustible });
}

List<Catalogo> liCatUnidadProd = new List<Catalogo>();

liCatUnidadProd.Add(new Catalogo() { idCat = "Masica", nombre = "Másica" });
liCatUnidadProd.Add(new Catalogo() { idCat = "Volumetrica", nombre = "Volumetrica" });

ViewBag.catUnidadProd = liCatUnidadProd;

List<Catalogo> liCatTipo = new List<Catalogo>();

liCatTipo.Add(new Catalogo() { idCat = "Convectivo", nombre = "Convectivo" });
liCatTipo.Add(new Catalogo() { idCat = "Rotatorio", nombre = "Rotatorio" });
liCatTipo.Add(new Catalogo() { idCat = "Columpio", nombre = "Columpio" });
liCatTipo.Add(new Catalogo() { idCat = "Tunel", nombre = "Túnel" });

ViewBag.catTipo = liCatTipo;

List<Catalogo> li = new List<Catalogo>();

query = "SELECT idpro, productos FROM snr.cat_prodHornos order by productos";

DataTable dtCat = datos(query);

for (int i = 0; i < dtCat.Rows.Count; i++)
{
    li.Add(new Catalogo()

```

```

        {
            idCat = dtCat.Rows[i][0].ToString(),
            nombre = dtCat.Rows[i][1].ToString()
        });
    }

    ViewBag.catProducto = li;

    List<Catalogo> liCatComb = new List<Catalogo>();

    liCatComb.Add(new Catalogo() { idCat = "GasNatural", nombre = "Gas Natural" });
    liCatComb.Add(new Catalogo() { idCat = "GasLP", nombre = "Gas LP" });

    ViewBag.catComb = liCatComb;

    List<Unidades> liUnidadesTemp = new List<Unidades>();

    liUnidadesTemp.Add(new Unidades() { idUnidad = "C", nombre = "°C" });
    liUnidadesTemp.Add(new Unidades() { idUnidad = "K", nombre = "°K" });
    liUnidadesTemp.Add(new Unidades() { idUnidad = "F", nombre = "°F" });

    ViewBag.unidadTemp = liUnidadesTemp;

    return View(m);
} // termina hornosMain post

// GET: Bombas
public ActionResult HornoDiag(decimal N2, decimal aire, decimal eficiencia, decimal perdGC,
decimal perdAis, string comb)
{
    decimal NN2 = N2 * 100;

    ViewData["NN2"] = NN2.ToString("0.##");
    ViewData["exAire"] = aire.ToString("0.##");
    ViewData["Eficiencia"] = eficiencia.ToString("0.##");
    decimal perdidasInt;

    if (comb.Equals("GasLP"))
    {
        perdidasInt = 0;
    }
    else
    {
        perdidasInt = 100 - perdGC * 100 - perdAis * 100 - eficiencia;
    }

    ViewBag.EficHorno = eficiencia;
    ViewBag.PerdGC = perdGC * 100;
    ViewBag.PerdAis = perdAis * 100;
    ViewBag.PerdInt = perdidasInt;
    ViewBag.Combust = comb;

    return View();
}

public ActionResult GraficaHorno(string Eficiencia, string GC, string Aislamiento, string
Intrinseco)
{
    List<Grafica> data = new List<Grafica>();

    data.Add(new Grafica()
    {
        etiqueta = "Eficiencia del Horno",
        valor = Convert.ToDecimal(Eficiencia),
    });
    data.Add(new Grafica()
    {
        etiqueta = "Perdidas por Gases de Combustión",
        valor = Convert.ToDecimal(GC),
    });
}

```

```
data.Add(new Grafica()
{
    etiqueta = "Perdidas por Aislamiento",
    valor = Convert.ToDecimal(Aislamiento),
});
data.Add(new Grafica()
{
    etiqueta = "Perdidas Intrinsecas",
    valor = Convert.ToDecimal(Intrinseco),
});
return Json(data.ToList(), JsonRequestBehavior.AllowGet);
```

## A.4. Apéndice 3. Interfaz del sistema

### A.4.1. Página de inicio



Registrarse Iniciar Sesión

#### SIGEEI

**Sistema Inteligente para la Gestión de la Eficiencia Energética en la Industria.**

Es una herramienta web para el diagnóstico y solución de problemas en la gestión de la eficiencia energética en la industria (Pymes).

#### OBJETIVO

Mejorar la forma del uso de la energía en las PyME, a través de la implementación de las sugerencias obtenidas de la herramienta en línea, la cual le permitirá al empresario identificar oportunidades de mejora y aprovecharlas con proyectos apropiados a cada caso.

#### RAMOS INDUSTRIALES SOPORTADOS



#### DESARROLLADO POR:



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.



UNIVERSIDAD AUTÓNOMA DE CAMPECHE.

Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

#### FINANCIAMIENTO POR:



CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA.



FONDO DE SUSTENTABILIDAD ENERGÉTICA.



SENER SECRETARÍA DE ENERGÍA.

#### COLABORACIÓN DE:



UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO.



COMISIÓN NACIONAL PARA EL USO EFICIENTE DE LA ENERGÍA



PROGRAMA LIDERAZGO AMBIENTAL PARA LA COMPETITIVIDAD DE LA PROCURADURÍA FEDERAL DE PROTECCIÓN AL AMBIENTE



CÁMARA NACIONAL DE LA INDUSTRIA DE LA TRANSFORMACIÓN.

[Aviso de Privacidad](#)

## A.4.2. Nosotros



[Inicio](#)

[Mejores Practicas](#)

[Nosotros](#)

[Casos de éxito](#)

[Contacto](#)

[Registrarse](#) [Iniciar Sesión](#)







El grupo de **Sistemas Inteligentes** el cual pertenece al **Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET)** tienen como objetivo alcanzar y consolidar capacidades de alto nivel teórico y práctico en el campo de las tecnologías del conocimiento e inteligencia artificial para promover el desarrollo de organizaciones inteligentes y automatizar la toma de decisiones. Lo anterior a través del diseño, desarrollo e implantación de sistemas inteligentes para solucionar problemas de diversas índoles y complejidades que ocurren en diferentes dominios tales como industria, medicina, finanzas, negocios, administración, ecología, energética, petróleo, educación, servicios, entre otros, desarrollando e integrando para ello técnicas, modelos, procesos y programas basados en Inteligencia Artificial, Inteligencia Computacional y Tecnologías de Información.

El grupo de ..... el cual pertenece a la **Facultad de Ingeniería** tiene como objetivo ....




El grupo de ..... el cual pertenece a la **Universidad Autónoma de Campeche** tiene como objetivo ....


El **Consejo Nacional de Ciencia y Tecnología (CONACYT)** fue creado en 1970, como un organismo público descentralizado de la Administración Pública Federal, integrante del Sector Educativo, con personalidad jurídica y patrimonio propio. Y es responsable de elaborar las políticas de ciencia y tecnología en México, coordinar y promover el desarrollo científico y tecnológico en México.

El **Fondo de Sustentabilidad Energética (FSE)** fue creado para ser la instancia que lidera la consolidación de las capacidades tecnológicas, institucionales y de talento en el sector energía de México. Su misión es identificar y apoyar oportunidades en el sector energía mediante la creación y fortalecimiento de capacidades tecnológicas, institucionales y de talento en la academia, sociedad, gobierno e industria a través de la implementación estratégica de programas, proyectos y actividades.

**DESARROLLADO POR:**




UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.




UNIVERSIDAD AUTÓNOMA DE CAMPECHE.

**FINANCIAMIENTO POR:**




CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA.




FONDO DE SUSTENTABILIDAD ENERGÉTICA.


**COLABORACIÓN DE:**




UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO.



COMISIÓN NACIONAL PARA EL USO EFICIENTE DE LA ENERGÍA



PROGRAMA LIDERAZGO AMBIENTAL PARA LA COMPETITIVIDAD DE LA PROCURADURÍA FEDERAL DE PROTECCIÓN AL AMBIENTE



CÁMARA NACIONAL DE LA INDUSTRIA DE LA TRANSFORMACIÓN.

[Aviso de Privacidad](#)

### A.4.3. Mejores Prácticas



[Inicio](#)

[Mejores Prácticas](#)

[Nosotros](#)

[Casos de éxito](#)

[Contacto](#)

[Registrarse](#) [Iniciar Sesión](#)

#### MEJORES PRÁCTICAS

La energía desempeña un papel fundamental en el desarrollo de todos los sectores productivos cuya utilización debería realizarse con alta eficiencia, bajo impacto medioambiental y al menor coste posible. El consumo de energía se ha ido incrementando unido a la producción de bienes y servicios.

Las empresas son grandes consumidoras de energía para proporcionar bienes y servicios a la sociedad, y en estos momentos disponen de opciones factibles para disminuir su consumo energético, mediante actuaciones que favorecen además la reducción de gastos, el aumento de la competitividad y la innovación tecnológica.

Podemos reducir nuestro consumo de energía utilizándola de forma más eficiente, invirtiendo en equipamiento energéticamente eficiente y en medidas de ahorro energético, así como adoptando un estilo de vida más sostenible con respecto al uso de la energía, es decir, cambiando nuestro comportamiento.

De ahí la importancia de aplicar las mejores prácticas en el consumo energético.

#### SECCIÓN EN CONSTRUCCIÓN



#### DESARROLLADO POR:

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

UNIVERSIDAD AUTÓNOMA DE CAMPECHE.

Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

#### FINANCIAMIENTO POR:

CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA.

FONDO DE SUSTENTABILIDAD ENERGÉTICA.

SECRETARÍA DE ENERGÍA.

#### COLABORACIÓN DE:

UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO.

COMISIÓN NACIONAL PARA EL USO EFICIENTE DE LA ENERGÍA

PROGRAMA LIDERAZGO AMBIENTAL PARA LA COMPETITIVIDAD DE LA PROCURADURÍA FEDERAL DE PROTECCIÓN AL AMBIENTE

CÁMARA NACIONAL DE LA INDUSTRIA DE LA TRANSFORMACIÓN.

[Aviso de Privacidad](#)

## A.4.4. Casos de Éxito



[Inicio](#)

[Mejores  
Practicas](#)

[Nosotros](#)

[Casos de  
éxito](#)

[Contacto](#)

[Registrarse](#) [Iniciar Sesión](#)

### CASOS DE ÉXITO

#### **Bebidas Santa Fe S.A DE C.V. - Hidalgo - México**

El diagnóstico efectuado por medio del sistema web SIGEEI identificó varias medidas de ahorro de energía, que se han propuesto y discutido con la dirección. Cuatro de estas medidas han sido aprobadas para su implementación, y estos son en las siguientes áreas: impulsos tecnológicos eléctricos, iluminación, ventilación, y configuraciones de equipo. La inversión total es de aproximadamente \$ 1,000,000 y que dará lugar a un ahorro energético anual de más de 292.000 kWh, lo que es un período de recuperación simple de sólo 1,9 años, y la reducción de CO2 de 239,8 toneladas / año!.

#### **Lordos Unidos Plásticos – Puebla - México**

Lordos Plásticos (fundada en 1959) es un proveedor líder de plásticos que proporciona una amplia gama de productos. La empresa consta de 5 edificios y casi 200 empleados. Su consumo anual de electricidad llega a 5.693 MWh. Después de varios diagnósticos en la herramienta SIGEEI, decidieron hacer cambios en el campo de la iluminación con el reemplazo de sus lámparas con LED. La iluminación representa un 4,85% del consumo total, y después de cambiar sus lámparas, se logró alcanzar un ahorro del 15%. El ahorro total asciende a 50.600 kWh, que es términos económicos son \$ 210,000 pesos.

#### DESARROLLADO POR:



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO.



UNIVERSIDAD AUTÓNOMA DE CAMPECHE.

#### Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

#### FINANCIAMIENTO POR:



CONSEJO NACIONAL DE CIENCIA Y  
TECNOLOGÍA.



FONDO DE SUSTENTABILIDAD ENERGÉTICA.



SENER SECRETARÍA DE ENERGÍA.

#### COLABORACIÓN DE:



UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO.



COMISIÓN NACIONAL PARA EL USO EFICIENTE DE LA ENERGÍA



PROGRAMA LIDERAZGO AMBIENTAL PARA LA COMPETITIVIDAD DE LA PROCURADURÍA FEDERAL DE PROTECCIÓN AL AMBIENTE



CÁMARA NACIONAL DE LA INDUSTRIA DE LA TRANSFORMACIÓN.

[Aviso de Privacidad](#)



## A.4.5. Contacto



[Inicio](#)

[Mejores Practicas](#)

[Nosotros](#)

[Casos de éxito](#)

[Contacto](#)

[Registrarse](#) [Iniciar Sesión](#)

### CONTACTO

Facultad de Ingeniería  
fi@unam.mx

Circuito Exterior S/N Av. Universidad  
CP 04510 Ciudad de México, México  
Tel: 55-55-55-55-55

#### DESARROLLADO POR:



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.



UNIVERSIDAD AUTÓNOMA DE CAMPECHE.

#### Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

#### FINANCIAMIENTO POR:



CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA.



FONDO DE SUSTENTABILIDAD ENERGÉTICA.



SENER SECRETARÍA DE ENERGÍA.

#### COLABORACIÓN DE:



UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO.



CONUEE COMISIÓN NACIONAL PARA EL USO EFICIENTE DE LA ENERGÍA



PROGRAMA LIDERAZGO AMBIENTAL PARA LA COMPETITIVIDAD DE LA PROCURADURÍA FEDERAL DE PROTECCIÓN AL AMBIENTE




CÁMARA NACIONAL DE LA INDUSTRIA DE LA TRANSFORMACIÓN.

[Aviso de Privacidad](#)

## A.4.6. Registro



<p>Registrarse es muy fácil proporcione los datos solicitados y comienza a utilizar el SIGEEI y ahorre en su consumo energético.</p> 	<b>REGISTRAR NUEVO USUARIO</b>	
	<p>Nombre(s): *</p> <input type="text"/>	<p>Nombre de Usuario: *</p> <input type="text"/>
<p>Apellido Paterno: *</p> <input type="text"/>	<p>Contraseña: *</p> <input type="password"/>	
<p>Apellido Materno:</p> <input type="text"/>	<p>Repita Contraseña: *</p> <input type="password"/>	
<p>Correo Electronico: *</p> <input type="text"/>	<p>Puesto en Empresa:</p> <input type="text"/>	
<p><b>Registrar</b></p> <p><small>Los campos marcados con * son obligatorios</small></p>		

[Aviso de Privacidad](#)

## A.4.7. Inicio de Sesión



<p>SIGEEI te ayuda a diagnosticar tu consumo de energía y te ofrece recomendaciones para el ahorro de energía en tu empresa.</p> 	<b>INICIAR SESIÓN EN SISTEMA</b>	
	<p>Nombre de Usuario:</p> <input type="text"/>	<p>Contraseña:</p> <input type="password"/>
<p><b>Iniciar Sesión</b></p> <p>¿No tiene una cuenta? <a href="#">Registrarse</a></p>		

[Aviso de Privacidad](#)

## A.4.8. Sistema con ingreso



CCADET



BIENVENIDO: Prueba

Cerrar Sesión

INICIO

CUENTA

FACTURACIÓN

DIAGNÓSTICO  
EQUIPO ELÉCTRICO

DIAGNÓSTICO  
EQUIPO  
COMBUSTIBLE

RECOMENDACIONES

INICIO

Sistema para diagnóstico energético en empresas. Para proceder al diagnóstico primero debe completar los datos sobre su empresa. [Ir a completar datos](#)



Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

## A.4.9. Encuesta de empresa



CCADET  
COMITÉ CONSULTIVO  
COMISIONADO DE ENERGÍA



BIENVENIDO: Prueba

Cerrar Sesión

INICIO

CUENTA

FACTURACIÓN

DIAGNÓSTICO  
EQUIPO ELÉCTRICO

DIAGNÓSTICO  
EQUIPO  
COMBUSTIBLE

RECOMENDACIONES

### Datos de la Empresa

Nombre de la Empresa: \*

RFC: \*

Estado: \*

Delegación o Municipio: \*

Calle: \*

Colonia: \*

Código Postal: \*

Teléfono:

Página Web:

Ramo Industrial: \*

*Los campos marcados con \* son obligatorios*

Guardar Datos

Sito Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

## A.4.10. Formulario de Caldera



BIENVENIDO: Prueba

Cerrar Sesión

INICIO

CUENTA

FACTURACIÓN

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

Diagnóstico de Calderas

Datos de Placa (Los puede obtener en las placas de sus equipos)

Datos de Caldera	Valor	Unidad
Tipo de Caldera	--Seleccionar--	
Marca		
Modelo		
Capacidad		HP
Área de transferencia		Uní
Fecha de Instalación		

Datos de la Bomba	Valor	Unidad
Marca		
Modelo		
Potencia		Unidad

Si su caldera tiene tiro natural inducido llenar los siguientes datos

Datos de motor del ventilador	Valor	Unidad
Marca		
Modelo		
Potencia		Unidad

SIGUIENTE >>>

Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

## A.4.11. Diagnóstico de Caldera



BIENVENIDO: Prueba

Cerrar Sesión

INICIO

CUENTA

FACTURACIÓN

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

Diagnóstico de caldera

La eficiencia energética de tu equipo es: 65.2 %

**Recomendaciones**

-Esta en buen estado tu aislante

Se recomienda mantener la revisión continua del aislante para evitar disminuir la eficiencia.

-La temperatura de la tubería es mayor a 50 ° C

Se recomienda aislar, puede que la eficiencia de tu caldera no sea la óptima.

SIGUIENTE >>>

Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

## A.4.12. Formulario de Motor Eléctrico







BIENVENIDO: Prueba Cerrar Sesión

INICIO

CUENTA

FACTURACION

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

**Diagnóstico de Motores**

Datos de Placa (Los puede obtener en las placas de sus equipos)

Datos de Motor	Valor	Unidad
Capacidad mecánica		Unidad
Tensión		Volts
Corriente nominal		Amperes
Revoluciones por minuto		RPM
Factor de Potencia		
Eficiencia		%
Tipo de motor	Tipo de Motor	

Datos de Operación (Aproximados acorde a la operación promedio observada)

Operación	Valor	Unidad
Horas de uso al día		hr
Reparaciones o rebobinados		
Días de operación al año		días
Antigüedad		años
Costo medio		MXN

[SIGUIENTE >>>](#)

Sitio Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322. Versión 2.0

## A.4.13. Diagnóstico de Motor Eléctrico







BIENVENIDO: Prueba Cerrar Sesión

INICIO

CUENTA

FACTURACION

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

**Diagnóstico de motor eléctrico**

La eficiencia energética de tu equipo es: 70.23 %

**Recomendaciones**  
-El motor tiene más de dos rebobinados  
Más de dos rebobinados disminuye la eficiencia energética del motor.

[SIGUIENTE >>>](#)

Sitio Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322. Versión 2.0

## A.4.14. Formulario de Sistema de bombeo



BIENVENIDO: Prueba Cerrar Sesión

INICIO

CUENTA

FACTURACIÓN

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

**Diagnóstico de Bombas**

DATOS DEL SISTEMA	
Aplicación:	--Seleccionar--
¿Conoce el flujo de operación?	Si <input checked="" type="radio"/> No <input type="radio"/>
Criterio para Cálculo de fricción	--Selección--
Criterio de presión en la succión	--Seleccionar--
Fluido bombeado	--Seleccionar--
Material de tub	--Seleccionar--

DATOS DE PLACA DE LA BOMBA	
Carga máxima (Hmax)	--Selec--
Gasto máximo (Qmax)	--Selec--
Carga op (Hop)	--Selec--
Gasto mínimo (Qmin)	--Selec--

DATOS DE OPERACIÓN ELÉCTRICA	
Potencia del motor	--Selec--
Corriente medida	Amp
Eficiencia motor	%
Tensión medida	V
Costo de la energía	\$/kWh
Factor de potencia	%

DATOS DE OPERACIÓN HIDRÁULICA	
Gasto op (Q_op)	--Selec--
Horas de operación al día	h/día
Presión op (Pop)	--Selec--
Días de operación al año	día/año
Distancia Vertical al usuario final	--Selec--
Distancia Horizontal al usuario final	--Selec--

**SIGUIENTE >>>**

Site Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322, Versión 2.0

## A.4.15. Diagnóstico de Sistema de bombeo



BIENVENIDO: Prueba Cerrar Sesión

INICIO

CUENTA

FACTURACIÓN

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

**Diagnóstico de bomba**

**RESULTADOS**

Carga (H):  
N conjunto bba-mot:  
Gasto (Q):  
N bomba (estimada):  
Potencia hid:  
Demanda eléctrica:

**Recomendaciones**  
-La bomba tiene cargas altas y flujos pequeños  
Es recomendable usar una bomba de flujo radial

**SIGUIENTE >>>**

Site Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322, Versión 2.0

## A.4.16. Formulario de Iluminación



BIENVENIDO: Prueba

Cerrar Sesión

- INICIO
- CUENTA
- FACTURACIÓN
- DIAGNÓSTICO EQUIPO ELÉCTRICO
- DIAGNÓSTICO EQUIPO COMBUSTIBLE
- RECOMENDACIONES

### Diagnóstico de Iluminación

#### Datos del área

Datos generales	Valor	Unidad
Nombre de área		
Tipo de área		
Altura		metros
Ancho		metros
Largo		metros
Altura de trabajo		metros
Altura de montaje de luminaria		metros

#### Datos de Luminaria

Datos de Luminaria	Valor	Unidad
Numero de luminaria		
Potencia		Watts
Lumenes		
Tipo de iluminación		
Tipo de lámpara		

SIGUIENTE >>>

Sitio Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322. Versión 2.0

## A.4.17. Diagnóstico de Iluminación



BIENVENIDO: Prueba

Cerrar Sesión

- INICIO
- CUENTA
- FACTURACIÓN
- DIAGNÓSTICO EQUIPO ELÉCTRICO
- DIAGNÓSTICO EQUIPO COMBUSTIBLE
- RECOMENDACIONES

### Diagnóstico de Iluminación

El consumo energético anual de tecnología 1 es: 1 274. 23  
 El consumo energético anual de tecnología 2 es: 1 274. 23  
 El consumo energético anual de tecnología 3 es: 1 274. 23  
 El consumo energético anual en el área 1 es : 3,822.69

#### Recomendaciones

-Utilización de lámpara incandescente  
 Reemplazar por una lámpara LED o fluorescente compacta.  
 -Colores oscuros en pared techo o piso.  
 Utilizar colores claros

SIGUIENTE >>>

Sitio Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322. Versión 2.0



## A.4.18. Formulario de Compresor



CCADET



BIENVENIDO: Prueba

Cerrar Sesión

INICIO

CUENTA

FACTURACIÓN

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

Diagnóstico de Compresor

Datos del área

Datos generales	Valor	Unidad
Años de antigüedad		
Ubicación		
Tipo compresor	Tipo	
Tipo de transmisión	Tipo	
Tipo de enfriamiento	Tipo	
Tipo de potencia nominal	Tipo	
Conoce el flujo de salida		Tipo

SIGUIENTE >>>

Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

## A.4.19. Diagnóstico de Compresor



CCADET



BIENVENIDO: Prueba

Cerrar Sesión

INICIO

CUENTA

FACTURACION

DIAGNÓSTICO EQUIPO ELÉCTRICO

DIAGNÓSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

Diagnóstico de Compresor

Las pérdidas por fugas son:  
 El flujo de combustible (estimado) es:  
 El aire perdido por fugas es :  
 La disminución por fugas es:  
 La energía anual es:  
 Costo que puede ahorrar con recomendaciones:

Recomendaciones

-La eficiencia del motor es aceptable.  
 Se sugiere establecer un programa de mantenimiento para mantener en buen estado el motor

SIGUIENTE >>>

Sitio Web por:

Copyright © 2018 ICAT - UNAM.

Derechos reservados

Conacyt Proyecto No 249322. Versión 2.0

## A.4.20. Formulario de Horno de Flama



BIENVENIDO: Prueba Cerrar Sesión

INICIO

CUENTA

FACTURACION

DIAGNOSTICO EQUIPO ELECTRICO

DIAGNOSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

Diagnóstico de Hornos

Datos del horno y servicio			
Clave del equipo	<input type="text"/>	Unidad de producto calentador	--Seleccionar--
Tipo	--Seleccionar--	Producto	--Seleccionar--
Año de instalación	<input type="text"/>	Combustible	--Seleccionar--
Aplicación o proceso	<input type="text"/>		

Datos requeridos: Método Indirecto		
Designación	Valor	Unidad
Consumo de combustible	<input type="text"/>	m <sup>3</sup> /mes
Tiempo de operación	<input type="text"/>	horas
Días de operación	<input type="text"/>	días/mes
Temperatura gases de combustión	<input type="text"/>	Unidac
Temperatura gases salida chimenea	<input type="text"/>	Unidac
Temperatura ambiente	<input type="text"/>	Unidac
Temperatura aire de entrada	<input type="text"/>	Unidac
Temperatura inicial del producto	<input type="text"/>	Unidac
Temperatura final del producto	<input type="text"/>	Unidac
Temperatura interior del horno	<input type="text"/>	Unidac
Masa del producto horneado por día	<input type="text"/>	kg/día

Análisis de gases de combustión		
O2	<input type="text"/>	%
CO2	<input type="text"/>	%
CO	<input type="text"/>	ppm

[SIGUIENTE >>>](#)

Sito Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322. Versión 2.0

## A.4.21. Diagnóstico de Horno de Flama



BIENVENIDO: Prueba Cerrar Sesión

INICIO

CUENTA

FACTURACION

DIAGNOSTICO EQUIPO ELECTRICO

DIAGNOSTICO EQUIPO COMBUSTIBLE

RECOMENDACIONES

Diagnóstico de Horno de flama

**RESULTADOS**

N2:  
Exceso de aire  
Eficiencia del horno es :

**Recomendaciones**  
-CO2 bajo y O2 alto  
Existe exceso de aire. Se recomienda disminuir la apertura de la compuerta del paso de aire del quemador

[SIGUIENTE >>>](#)

Sito Web por:  
Copyright © 2018 ICAT - UNAM.  
Derechos reservados  
Conacyt Proyecto No 249322. Versión 2.0

## 7. REFERENCIAS

1. INEGI. (2016). Se difunden estadísticas detalladas sobre las micro, pequeñas y medianas empresas del país. julio 25, 2018, de INEGI Sitio web: <https://goo.gl/S22nvF>
2. Ministerio de Economía de la República de El Salvador Banco Interamericano del desarrollo. (2011). Metodología de eficiencia energética en la industria. El salvador: Consejo Nacional de Energía
3. Realpozo, Pablo. (2007). Ahorro de Energía Eléctrica en México, Avances y prospectiva 2006-2012. México: Academia de Ingeniería.
4. Muerza, Alex. (2016). Los países más eficientes en energía del mundo. 2018, julio 30, de EROSKI Consumer Sitio web: <https://goo.gl/uP3K23>
5. Alonso, Lola. (2017). Auditoría energética: una herramienta de ahorro. 2018, julio 30, de BlacktoGreen consulting Sitio web: <https://goo.gl/jTjsuS>
6. Anónimo. (2018). Eficiencia Energética. agosto 7,2018, de Siemens Sitio web: <https://goo.gl/uoB88P>
7. Kreith, Frank. (2007). *Handbook of energy efficiency and renewable energy*. USA: CRC Prees.
8. Anónimo. (2017). Breve historia de eficiencia energética. Agosto 9, 2018, de Econoticias Sitio web: <https://goo.gl/TwRAky>
9. SENER. (2016). Sistema de Información energética. Agosto 10, 2018, de Secretaría de Energía Sitio web: <https://goo.gl/wX2rD9>

10. Vicente Juan. (2014). Eficiencia Energética Parte I: Antecedentes Históricos. Agosto 13, 2018, de Espacio Más Abierto Sitio web: <https://goo.gl/VmMfDm>
11. Anónimo. (2018). Protocolo de Kioto. Agosto 13, 2018, de Wikipedia Sitio web: <https://goo.gl/JnJKa5>
12. Muy Interesante. (Desconocido). ¿Qué es el protocolo de Kioto? Agosto 13, 2018, de Muy Interesante Sitio web: <https://goo.gl/aRLrFM>
13. Araujo, Montserrat. (2018). Sector industrial en el 2018. agosto 15, 2018, de El economista Sitio web: <https://goo.gl/j1nqfF>
14. Arana, David. (2018). Pymes mexicanas, un panorama para 2018. Agosto 15, 2018, de Forbes Sitio web: <https://goo.gl/VLorsL>
15. EcuRed. (2018). Sistemas Expertos. Agosto 27, 2018, de EcuRed Sitio web: <https://goo.gl/cayN5R>
16. Turban E, Aronson J, Liang TP. (2005). *Knowledge acquisition, Representation and Reasoning*. En *Decisión support systems and intelligent system* (177,178). USA: PEARSON.
17. Universidad de Alicante. (Desconocido). Modelo-Vista-Controlador. noviembre 8, 2018, de Universidad de Alicante Sitio web: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>