



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA
Centro de Ciencias Aplicadas y Desarrollo Tecnológico

Sistema Web para la Gestión de la Eficiencia Energética en Hoteles

TESIS PROFESIONAL
para obtener el título de
INGENIERO EN COMPUTACIÓN
ÁREA
Bases de Datos

PRESENTA:
JONATHAN BENÍTEZ JIMÉNEZ

DIRECTOR DE TESIS
Dr. Nicolas Ceferino Kemper Valverde



Ciudad Universitaria, México, Noviembre 2012.

ÍNDICE

I. LA GESTIÓN ENERGÉTICA Y SU IMPACTO SOCIAL, AMBIENTAL Y ECONÓMICO.....	3
I.1 INTRODUCCIÓN	3
I.2 EL CONCEPTO DE GESTIÓN ENERGÉTICA	3
I.3 FACTORES Y CONSECUENCIAS DE UNA GESTIÓN INADECUADA	6
I.4 LOS ESPACIOS HOTELEROS Y SU GESTIÓN ENERGÉTICA ACTUAL EN MÉXICO.....	8
I.5 LA NECESIDAD DE IMPLEMENTAR UN SISTEMA PARA LA GESTIÓN EFICIENTE DE LA ENERGÍA...9	
II. TECNOLOGÍAS WEB: PROGRAMACIÓN EN LA NUBE.....	10
II.1 INTRODUCCIÓN	10
II.2 COMIENZOS.....	10
II.3 BENEFICIOS.....	14
II.4 DESVENTAJAS	15
II.5 CAPAS	16
II.5.1 SOFTWARE COMO SERVICIO (SAAS – SOFTWARE AS A SERVICE)	16
II.5.2 PLATAFORMA COMO SERVICIO (PAAS – PLATFORM AS A SERVICE).....	18
II.5.3 INFRAESTRUCTURA COMO SERVICIO (IAAS – INFRASTRUCTURE AS A SERVICE)	23
II.6 TIPOS DE NUBE	27
II.7 COMPARACIONES.....	32
II.8 LENGUAJES PARA PROGRAMAR EN LA NUBE.....	35
III. UN SISTEMA WEB INNOVADOR PARA LA GESTIÓN DE LA EFICIENCIA ENERGÉTICA EN HOTELES ..46	
III.1 NECESIDAD ACTUAL: AUTOMATIZACIÓN Y AHORRO DE LAS FUENTES ENERGÉTICAS. LA IDEA DE UTILIZAR UN SISTEMA COMPUTARIZADO PARA LA GESTIÓN.....	46
III.2 EXPOSICIÓN Y EXPLICACIÓN DEL CÓDIGO.....	58
III.3 PUESTA A PRUEBA DEL SISTEMA.....	90
III.4 REFINAMIENTO DEL SISTEMA	104
IV. VENTAJAS DE UTILIZAR UN SISTEMA COMPUTARIZADO PARA LA GESTIÓN EFICIENTE DE LA ENERGÍA VS. EL ANÁLISIS EMPÍRICO	113

IV.1 LA COMPUTACIÓN, LÓGICA Y MATEMÁTICAS COMO HERRAMIENTAS DE ANÁLISIS.....	113
IV.2 ENFOQUE EMPÍRICO PARA LA IMPLEMENTACIÓN DE UN SISTEMA GESTOR.....	114
IV.3 OTRAS ÁREAS Y PROFESIONES QUE INTERVIENEN PARA LA IMPLEMENTACIÓN DE UN SISTEMA GESTOR DE ENERGÍA.....	116
IV.4 RELACIÓN COSTO-BENEFICIO-TIEMPO.....	118
V. CONCLUSIONES.....	120
VI. BIBLIOGRAFÍA.....	121

I. LA GESTIÓN ENERGÉTICA Y SU IMPACTO SOCIAL, AMBIENTAL Y ECONÓMICO

I.1 INTRODUCCIÓN

Día a día la demanda energética que requiere la sociedad en sus diversos sectores de desarrollo va en aumento, haciendo que los requerimientos de los recursos necesarios para su producción también sea mayor.

Esto afecta no solo al medio ambiente, sino que el sector económico y social también se ve perjudicado. Además si observamos la situación actual, notaremos que los cambios se hacen cada vez más notorios y prominentes, lo que conlleva a tomar medidas ya no como una opción sino como a una necesidad.

Para lograr programas y planes de ahorro energético, es necesaria una gestión energética confiable y eficiente, pero sobre todo se requiere que la gente adquiera una conciencia económica, social y ambiental, que haga los cambios palpables y notorios.

Sin embargo, que mejor si como punto de partida conocemos lo que es la Gestión Energética y los beneficios que ésta nos proporciona.

I.2 EL CONCEPTO DE GESTIÓN ENERGÉTICA

La Gestión Energética es la suma de medidas planificadas y llevadas a cabo para conseguir el objetivo de utilizar la mínima cantidad posible de energía mientras se mantienen los niveles de confort y los niveles de producción.

Se puede aplicar a una fábrica, a un edificio de oficinas, a un centro deportivo, a una vivienda, y a cualquier tipo de edificio dónde se requiera un uso de energía. Para hacer un uso eficiente de la energía y, como consecuencia, para ahorrarla, las acciones se centran en:

- ✓ Conservación de la Energía
- ✓ Recuperación de la Energía
- ✓ Sustitución de la Energía

¿Por qué es necesaria la Gestión Energética?

- ❖ Para reducir las emisiones de CO₂ a la atmósfera es necesario mejorar la eficiencia energética y utilizar más las fuentes de energía renovable.
- ❖ Ahorraremos dinero si hacemos un uso más eficiente de las instalaciones y equipos o utilizando equipo más eficiente.
- ❖ Es un camino claro de identificar todos los costos que genera el consumo de la energía. [7]

¿Con qué factores está relacionada la Gestión Energética?

Personas

- **Comportamiento**
Normalmente, el principal problema es que las personas tienen hábitos de consumo de la energía que se deben mejorar. Cambiando estos hábitos, utilizaremos la energía más eficientemente y, como consecuencia, ahorraremos energía y dinero. Todos los que utilizan el espacio donde se aplica una gestión energética pueden hacer algo para mejorar su uso. Una de las tareas más importantes de la Gestión Energética es precisamente informar a la personas con el objetivo de cambiar sus hábitos para el uso eficiente de la energía. [7]
- **Ocupación**
El número de horas que el edificio está ocupado es un factor que influye en la demanda de energía. Muchas veces también se deben tomar en cuenta las actividades que se desarrollan en los espacios ocupados para poder implementar las medidas de ahorro que sean las más apropiadas y eficientes. [7]

Edificio

Existen algunos aspectos del edificio que influyen en las necesidades energéticas:

- *Luz natural*: Deberíamos utilizar tanta luz del sol como sea posible. De esta forma, es seguro que necesitaremos menos luz artificial, consumiendo menor cantidad de energía para iluminar los espacios.
- *Calor del sol*: En verano podemos utilizar menos aire acondicionado si se protege utilizando persianas o cortinas. [7]

- *Controles del sistema energético:* Los aparatos de control como termostatos e interruptores deben ser fácilmente accesibles por los usuarios. Esto permitirá que las personas haga un uso más efectivo de la energía.
- *Distribución del sistema energético:* Con una distribución más eficiente de los sistemas energéticos podemos reducir el consumo de energía. [7]

Tipo de energía utilizada

Algunos servicios energéticos se pueden conseguir con diferentes tipos de energía. La calefacción se puede obtener con el gas, combustibles sólidos, petróleo, electricidad o el sol. Se puede considerar el cambio del tipo de energía que se utiliza para conseguir un ahorro energético. El costo de la energía se debe incluir en los costos de los equipos. Generalmente, el equipo de calefacción eléctrica es más barato que los paneles solares, pero el beneficio para el medio ambiente es significativo, además que se amortizan los costos en el consumo eléctrico. [7]

Control de Carga

Controlando el tiempo de las cargas eléctricas se puede reducir el costo de la electricidad sin reducir el consumo. Podemos evitar las puntas de carga o el consumo en niveles de tarifa altos. [7]

Equipo Instalado

El equipo instalado es uno de los solicitantes de energía más importantes del edificio. Dependiendo de su eficiencia y comportamiento, habrá diferentes niveles de demanda en energía, el mantenimiento que se le da al equipo es un factor que también hay que considerar para reducir el consumo. [7]

Factores Externos

Hay algunos factores externos, como las condiciones meteorológicas, que influyen en la demanda energética y, por tanto, en la Gestión Energética. Por ejemplo, cuando en invierno las temperaturas externas son bajas, la demanda de calefacción normalmente es alta. Es aquí en donde el equipo instalado es un factor importante en el ahorro energético.

Teniendo ya un panorama de lo que es la Gestión Energética y algunos de los beneficios que ésta acarrea, a continuación se tratará de darle un enfoque Empresarial y de la problemática que existe en este sector para su aplicación. [7]

I.3 FACTORES Y CONSECUENCIAS DE UNA GESTIÓN INADECUADA

La gestión energética beneficia a la empresa, contribuyendo a garantizar la calidad de los productos y servicios, reduciendo costos de producción y elevando su competitividad; al país aplazando los requerimientos de financiamiento para la infraestructura energética, promoviendo nuevas tecnologías y la modernización del sector empresarial, y reduciendo la importación de bienes de capital para el desarrollo energético; a la sociedad, conservando recursos para las futuras generaciones, disminuir las emisiones contaminantes al medio ambiente y contribuyendo a la formación de una cultura energética y ambiental.

Gestión Energética Empresarial

La administración de la energía necesita un enfoque gerencial coherente e integral. La experiencia demuestra que los ahorros de energía sólo son significativos y perdurables en el tiempo cuando se logran en el marco de un sistema integral de gestión energética, que trate a la energía como un recurso más bajo el control de la administración de la empresa.

El consumo de energía es un gasto controlable, y la función de una buena gerencia es mantener este control.

La experiencia en materia de administración de energía indica que:

- I. Es más importante para una empresa contar con un sistema continuo de identificación de oportunidades que la detección de una oportunidad aislada.
- II. La administración de energía debe concebirse como un proceso de mejoramiento continuo de la eficiencia energética.
- III. Los esfuerzos deben concentrarse en el control de las principales funciones energéticas (energéticos, áreas, equipos y personal clave).
- IV. Debe controlarse el costo de las funciones o servicios energéticos, y no solo el costo de la energía primaria.
- V. El costo de las funciones o servicios energéticos debe controlarse como parte del costo del producto o servicio.
- VI. Deben orientarse los programas al logro de resultados y metas concretas.

- VII. Es necesario realizar el mayor esfuerzo en la instalación de equipos de medición (no se puede administrar lo que no se conoce). [5]

En el siguiente esquema, se muestran las causas principales que caracterizan el bajo nivel de gestión energética existente:

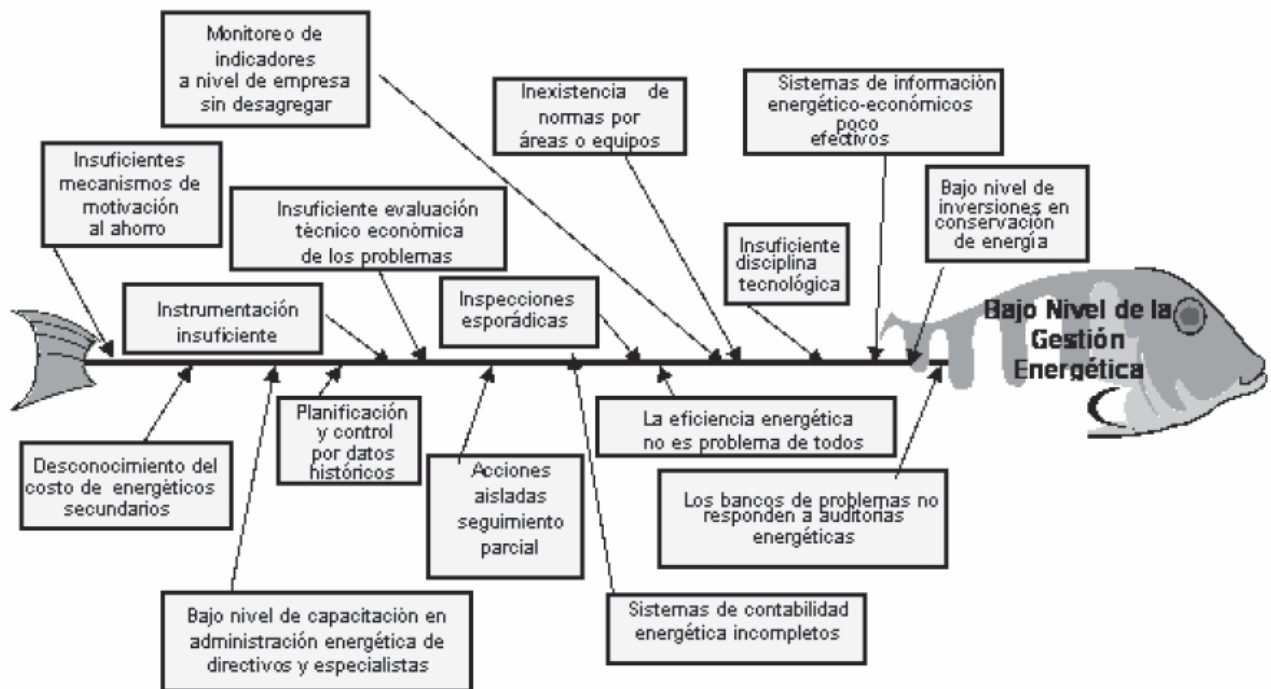


Figura 1. Causas que caracterizan el bajo nivel de la Gestión Energética [5]

La gestión energética implica monitoreo, registro, evaluación y acción correctiva continua sobre los equipos, áreas, procesos y personal clave, para reducir los consumos y gastos energéticos.

El monitoreo y control de la energía, la mejoría en la prácticas de operación y mantenimiento y la implementación de medidas técnico organizativas simples, de bajo costo, lograrán mucho más en el corto plazo. A menudo es más sencillo quejarse por la falta de dinero para invertir en medidas de ahorro de energía, que admitir que la administración presenta puntos débiles.

Si no se cuenta con una administración y un mantenimiento efectivo, no se podrán alcanzar los máximos beneficios de las inversiones en tecnología eficiente.

De todo esto podemos deducir que el llevar una Gestión Energética inadecuada, incrementará en gran medida nuestros gastos económicos en los pagos de energía, desperdiciaremos gran cantidad de ella y estaremos afectando al medio ambiente.

Ahora toca el punto de hablar sobre la gestión energética que existe actualmente en el sector hotelero en México, este tema se tratará basado en la experiencia personal en pláticas que se han tenido con personal encargado desde administrar, mantener, ampliar, etc. Hoteles en diversas regiones del país. [5]

I.4 LOS ESPACIOS HOTELEROS Y SU GESTIÓN ENERGÉTICA ACTUAL EN MÉXICO

Los Espacios Hoteleros en México

En México, el problema que existe en los espacios hoteleros es que muchos de ellos no están destinados para tal fin, es decir, son adaptados para convertirse en hoteles. Esto hace que la planificación de una estrategia de administración de energía se dificulte; desde que los materiales con los que está construido el edificio cambia radicalmente, hasta las adaptaciones que se hacen para proporcionar servicios de: aire acondicionado/condicionamiento, distribución del sistema de agua, instalación de calentadores para el agua, etc. En general, dichas adaptaciones están efectuadas sin tomar en consideración una gestión energética eficiente.

En algunos espacios adaptados a hotel, para satisfacer la creciente demanda de energía eléctrica se opta por la instalación de un banco de capacitores, sin antes haber realizado un estudio y haber identificado los factores que más influyen en el consumo energético, que van desde el cambio de lámparas hasta problemas en la instalación eléctrica; tal vez con el simple cambio de luminarias se disminuya el consumo evitando la adquisición de equipo eléctrico adicional, o convenga más un cambio de tarifa con la compañía de electricidad. Este sería un ejemplo de aplicación en la gestión de energía, que permita reducir gastos de consumo eléctrico y desperdicios en la misma.

Gestión Energética actual en México

En México, la gestión energética en hoteles pequeños y medianos es casi nula, solo existen algunas políticas de ahorro energético, pero no se lleva un plan continuo con el fin de optimizar y hacer más eficiente el consumo.

Muchas veces la falta de información hace que no haya interés por parte del sector hacia la implementación de un buen plan de gestión energético, pero cuando se informa a la gente y los beneficios que acarrea una buena gestión energética, algunos se entusiasman en gran medida, mientras que otros simplemente lo toman como una plática informativa más, sin embargo por aquellos que están entusiasmados, vale la pena trabajar.

El sector hotelero en México, tiene mucho potencial de desarrollo, solo hay que saber como empezar, y que mejor si se empieza por una buena gestión energética, con esto, no solo se logrará un ahorro, sino que se disminuirá el impacto que se tiene sobre el ambiente y también se pondrá en alto el nombre de nuestro país. Todos ganamos.

Es claro que no existe una gestión energética actual, ya que la poca o nula información que se tiene sobre el tema no ayuda, es necesario establecer las bases que den comienzo a una conciencia social y las buenas prácticas.

Es por esto que en este trabajo se pretende dar un paso en lo que a gestión energética se refiere, construyendo una herramienta computacional que ayude al sector hotelero en México.

I.5 LA NECESIDAD DE IMPLEMENTAR UN SISTEMA PARA LA GESTIÓN EFICIENTE DE LA ENERGÍA

Como ya hemos mencionado, hoy en día la gestión eficiente de la energía ya no es una opción, sino una necesidad. Con el avance tecnológico que existe actualmente ya no hay excusa para no tener implementado algún sistema de gestión, independientemente si está basado en las TI (Tecnologías de la Información) o no.

Pero ¿qué es lo que hace a uno pensar que se necesita de un sistema para la gestión de la energía?

La respuesta a esta interrogante no es más que: la información. Cuando se está informado sobre los beneficios que acarrea una buena administración energética es cuando se despierta el interés. Que mejor resultado que ver nuestro consumo de energía disminuido y que éste se vea reflejado en el recibo de luz, y a la vez el factor económico, que, probablemente sea el más importante para nosotros, también se vea beneficiado.

Con un sistema eficiente para la gestión energética, ahorramos energía, dinero y cuidamos el medio ambiente. Pequeños cambios pueden hacer una gran diferencia, desde cambiar el tipo de luminarias que existen en el hotel, hasta modernizar nuestro sistema de aire acondicionado.

Si tenemos un sistema de gestión energético eficiente, podemos estar seguros que no perderemos, muchas veces la inversión hace ver que se pierde, pero se debe estar consciente que las ganancias vienen de mediano a largo plazo. Esta también es una de las causas principales por las que no se quiera invertir en un sistema de gestión, que como las ganancias no son a corto plazo, mejor se opte por otras alternativas o seguir pagando lo mismo como se ha ido haciendo sin pensar en disminuir costos.

Finalmente, ya no podemos darnos el lujo de tener instalaciones sin o con poco mantenimiento, equipo ineficiente, fugas y desperdicio de energía eléctrica. Ya que todo esto hace menos atractivo el lugar hacia el sector turismo, generando pérdidas y dando mala imagen, es momento de generar consciencia y comenzar a implementar medidas de ahorro en beneficio de todos.

II. TECNOLOGÍAS WEB: PROGRAMACIÓN EN LA NUBE

II.1 INTRODUCCIÓN

La programación web en la nube es una realidad que día a día se va perfeccionando, actualmente se tienen aplicaciones, plataformas y sistemas trabajando en su totalidad o parcialmente en internet, esto quiere decir que no requieren de estar instaladas físicamente en la computadora del usuario para su uso, basta con una conexión a internet y una cuenta asociada para explotar la tecnología. A esto es lo que se le conoce como computación en la nube, proporcionar servicios de cómputo a través de internet.

Con computadoras que poseen mayor capacidad de procesamiento y almacenamiento, es posible proporcionar los servicios de cómputo necesarios a través de internet que hacen de la computación en la nube una tecnología que llegó para quedarse.

II.2 COMIENZOS

Para conocer cómo es que se llegó a la computación en la nube, es necesario hacer un recuento de cómo es que era gestionada la información y los recursos de computo antes de formalizar el nuevo modelo al que hoy en día conocemos como: cloud computing (computación en la nube de internet).

La Internet no es más que la interconexión de computadoras alrededor de todo el mundo. Las personas de hoy, dependen muy fuertemente del Internet, ya que puede ser usado

para compartir recursos, envío y recepción de correo electrónico (e-mail), pláticas y conferencias en línea (chatting), y sobre todo para la búsqueda de información. La forma de usar el Internet va cambiando día con día.

Para los inicios de la Internet, los servicios que se ofrecían eran limitados, pero con el paso del tiempo, éstos fueron creciendo y ahora, muchos de los desarrolladores de la IT se están enfocando para proporcionar todo tipo de servicio a través de la Internet.

Otro aspecto importante, es que las computadoras personales tienen un límite en cuanto almacenamiento, memoria y capacidad de procesamiento. Si un determinado usuario quisiera almacenar una gran cantidad de imágenes, videos y documentos teniendo cierto límite de espacio en disco y suponiendo que también quisiera instalar software que consume gran cantidad de procesamiento y memoria, entonces la computadora personal se convierte en una herramienta inadecuada para dichas tareas. La computación en la nube busca dar solución a todo esto y romper dichas barreras. [2]

La computación en la nube puede verse como una extensión de: redes de computadoras (grid computing), computación distribuida (distributed computing), computación en paralelo (parallel computing). En la computación en la nube, todo recurso puede ser proporcionado como servicio. [2]

La computación ha venido a cambiar la forma de trabajar. Resuelve el problema en cuanto al tiempo requerido para dar solución a infinidad de tareas. Tan pronto como la forma de proporcionar servicios cambia, las tareas se simplifican.

En los inicios de la era de la computación, el uso de computadoras centrales (mainframes) era muy popular. Estas computadoras eran usadas en grandes organizaciones para procesar gran cantidad de datos, poseían grandes gabinetes para hospedar procesadores (CPU's) y memoria que trabajaban en conjunto. Sin embargo, este tipo de computadoras no era financieramente adecuada para un individuo en específico. Es aquí cuando surge el concepto de "computadora personal" en donde se reduce el costo de los equipos de cómputo y da inicio a la nueva era de computadoras. [1] [2]

La computadora personal es un equipo adaptado para el uso de un solo individuo. Individual se refiere a que no es necesario entrenamiento especial, o de algún operador en particular para su uso. Una computadora personal puede ser desde una computadora de escritorio, laptop, Tablet PC, o computadora de mano (handheld PC). El software puede ser instalado y usado fácilmente. [1]

Las computadoras personales reducen costos, sin embargo, si una organización requiere de la implementación de aplicaciones y uso de bases de datos en cada una de sus equipos de cómputo de manera individual, esto puede incrementar los costos de implementación, y a la vez hacer de ésta un proceso complejo y muy rígido.

La arquitectura cliente-servidor resuelve el problema de implementación. En este paradigma, el cliente y el servidor son diferentes entidades conectados a través de una red. Las bases de datos y lógica de programación son gestionadas por el servidor, mientras que las interfaces son implementadas en el equipo del cliente. Con este tipo de arquitectura se reducen costos, se incrementa el desempeño, facilidad de mantenimiento, es posible escalar para incrementar el número de equipos interconectados, considerable disponibilidad y reduciendo el tiempo de implementación de las aplicaciones para su uso. Sin embargo, en esta arquitectura los recursos de información son limitados y muchas veces no es posible expandir ni compartirlos a través de la Internet de forma eficiente y efectiva. Es por eso que se dio paso a la cuarta generación del cómputo. [2]

Para la cuarta generación de computadoras, la WWW permitía compartir y proveer información de una manera eficiente y efectiva. WWW significa World Wide Web y fue fundada por Tim Berners-Lee. La Web es un sistema de archivos de hipertexto interconectados que son accedidos vía Internet. En el Internet no hay puntos de control, puntos específicos de información, dueños y usuarios únicos, o proveedores de servicio; es decir, hay pocas regulaciones para limitar la gran cantidad de cosas que se pueden lograr haciendo el uso del Internet. El Internet fue usado principalmente para propósitos militares, pero actualmente se ha convertido de uso casi estricto en el día a día. Provee información de una manera distribuida, en donde los datos se encuentran almacenados en diversos servidores y los recursos son proporcionados cuando el usuario los requiere. [3]

Sistemas completos construidos hasta la fecha poseen ciertas limitantes, como capacidad de almacenamiento, memoria y procesamiento. Estas limitantes de la cuarta generación de computadoras dieron paso a la quinta generación, mejor conocida como la Computación en la Nube. La Computación en la Nube no se limita a la computación en red (grid computing), a la computación en paralelo (parallel computing) o a la computación distribuida (distributed computing). [1] [3]

La computación en red (grid computing) provee de recursos a los usuarios solamente cuando son requeridos. La computación en paralelo (parallel computing), ejecuta instrucciones en paralelo para una rápida respuesta al realizar una tarea que el usuario solicitó, en este tipo de cómputo solo se provee de una rápida respuesta, no se presenta el almacenamiento, ni la memoria como recurso para ser solicitado. Finalmente en la computación distribuida (distributed computing), la información o datos se encuentran distribuidos en diversos servidores, inclusive situados en diferentes áreas geográficas. La Computación en la Nube involucra las ventajas de cada uno de estos paradigmas para formar una fuente de recursos a cualquier nivel, en donde todo puede ser proporcionado como servicio: almacenamiento, capacidad de procesamiento y memoria. [2] [3]

En la computación en la nube los recursos son gestionados de forma dinámica y automática por medio de hardware y software.

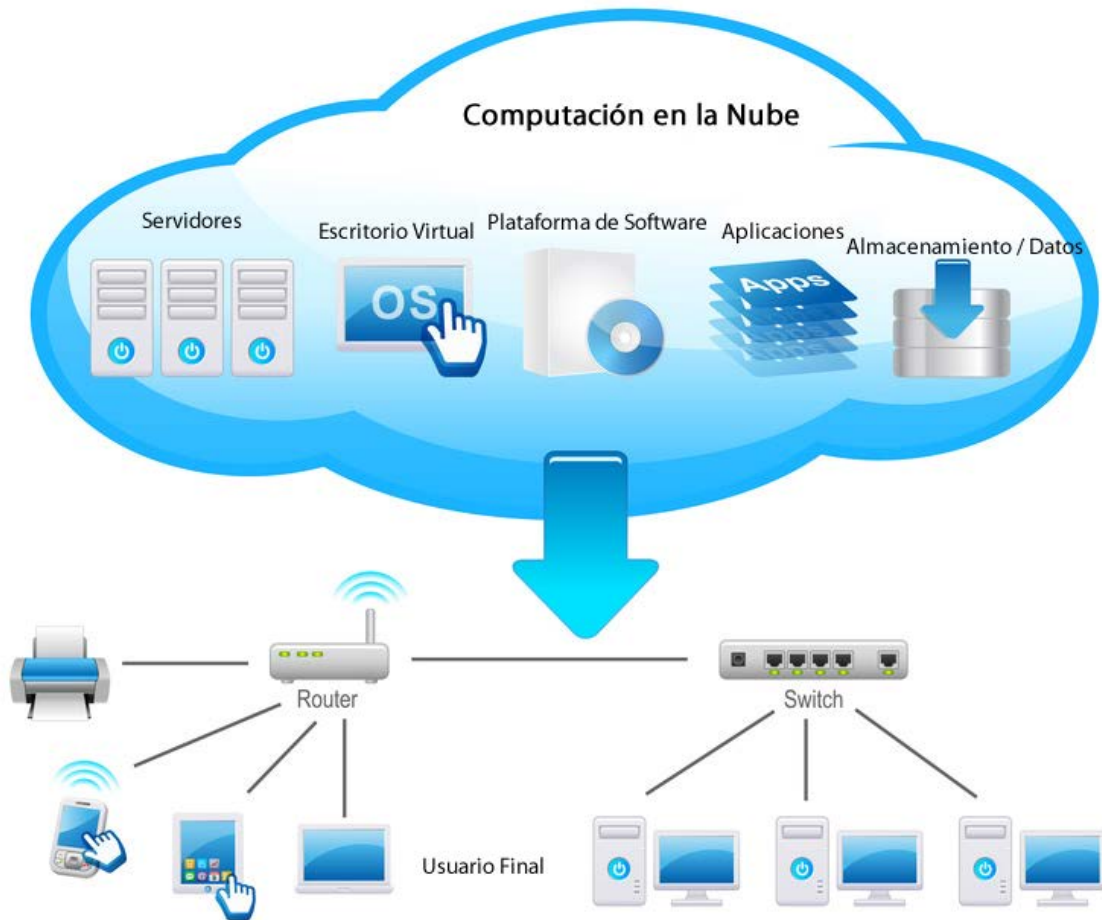


Figura 2. Computación en la Nube [8]

Como se observa en la figura 2, en la computación en la nube se crea una fuente de recursos virtuales como es almacenamiento, uso de capacidad de procesamiento, memoria y redes para satisfacer la demanda de recursos por parte del usuario final y a la vez provee de hardware y software on demand (únicamente cuando se es requerido y bajo un esquema de “pago por uso”) sin dificultad alguna. La computación en la nube es: computación dinámica, ya que provee de los recursos solicitados (dinámicamente) cuando estos son requeridos.

II.3 BENEFICIOS

Hagamos un análisis de los principales beneficios que nos proporciona la computación en la nube.

✓ Capacidad de hacer de la Internet un entorno de Escritorio.

Esta característica permite simular el trabajo que se realiza con las aplicaciones instaladas en una computadora pero desde Internet, es decir, no es necesario que se instalen aplicaciones en el equipo del usuario para su utilización. El almacenamiento y capacidad de procesamiento se deja en manos de la computación en la nube. Para acceder a este tipo de aplicaciones no se requiere más que de una conexión a internet y a lo menos un navegador Web.

Ejemplos de este tipo de aplicaciones ya han sido y están siendo implementados por Google (Google Docs), Amazon (Amazon Cloud Drive), Microsoft (Windows Azure) y otras empresas enfocadas ya sea para proporcionar servicios globales o privados.

Si un usuario en particular carece de espacio de almacenamiento en su computadora, puede hacer uso de las soluciones en almacenamiento proporcionadas por la computación en la nube evitando la preocupación de quedarse sin espacio. Así como este caso, podemos mencionar muchos otros ejemplificando el amplio rango de soluciones que existen en la nube, sin embargo, por su extensión, nos limitamos al concepto.

✓ Servicios de almacenamiento.

A la vez de proporcionar una sensación de trabajo local (de escritorio), la computación en la nube permite al usuario almacenar archivos de forma remota y que estos puedan ser accedidos desde cualquier computadora conectada a Internet. Con esto es posible no solo simplificar el mantenimiento, creación y edición de archivos, sino que también se tiene la seguridad de que estos no se perderán y son actualizados en tiempo real.

Ejemplo de este tipo de enfoque es Dropbox (<https://www.dropbox.com>), servicio que permite el almacenamiento y gestión de archivos en la nube.

✓ Virtualización.

El concepto de virtualización consiste en hacer de una computadora un entorno virtual para otros sistemas de computadoras. La base de un sistema de virtualización es el componente conocido como monitor de máquinas virtuales (VMM – Virtual Machine Monitor). El VMM se encarga de la creación y ejecución de múltiples máquinas (entornos) virtuales (VMs) en una sola computadora física, y multiplexa un solo recurso físico en múltiples recursos virtuales. Cada máquina

virtual (VM) posee su propio sistema operativo (comúnmente llamado: sistema operativo invitado), a la vez que las aplicaciones que son ejecutadas por esta VM también poseen sus propios recursos virtuales, como procesador, memoria, tarjeta de red, tarjeta de video, etc.

El engine (motor) utilizado por GoogleApps utiliza virtualización en la nube para distribuir la carga a través de varios servidores y bases de datos en sus aplicaciones. [3]

Y dentro de las ventajas a grandes rasgos, podemos mencionar:

- ✓ Capacidad de proporcionar servicios de computo rápidos, eficientes y a bajo costo
- ✓ Elasticidad en el uso de los recursos, ya que éstos son utilizados y liberados dependiendo de la demanda que se tenga
- ✓ Optimización en general al utilizar los recursos de cómputo
- ✓ Se tiene relativamente ilimitado espacio de almacenamiento, ya que es fácil escalarlo con otros servidores
- ✓ Todo recurso es proporcionado como servicio
- ✓ Consumo de energía reducido por el hardware y software para el proveedor del servicio, ya que la infraestructura necesaria se ve reducida
- ✓ Alta disponibilidad y escalabilidad
- ✓ Reducido el riesgo de pérdida de datos e información

II.4 DESVENTAJAS

Aunque no lo parezca, la computación en la nube aún tiene todavía que enfrentar varios retos como lo es en los aspectos de seguridad, confiabilidad, interoperabilidad, disponibilidad y acuerdos para los niveles de servicio.

Todo usuario requiere seguridad en la información, sobre todo cuando se trata de información sensible: información sobre identidad, cuentas de e-mail para uso de negocios, datos bancarios, etc. En la nube, todo debe estar asegurado, previniendo ataques (hacking) y fallas técnicas tanto en software y hardware, por la forma a como se escalan los recursos y servicios, lograr la seguridad requerida puede ser una tarea compleja.

Confianza significa delegar responsabilidades en alguien o algo. Es por eso que proponer un esquema en la nube a los usuarios debe ser lo suficientemente convincente para que pongan su confianza de los servicios que requieran en nuestras manos. Un esquema enfocado en la nube debe sobrepasar los esquemas tradicionales de computo (cliente-servidor, grid, computación distribuida, etc.)

Gran cantidad de recursos son proporcionados como servicio, por lo que éstos deben ser interoperables para lograr un escalamiento acorde a la demanda. Resolver interrogantes como: ¿Si la comunicación con el servidor es interrumpida, sea la razón que fuese (fallas técnicas, hacking, etc.), que tan rápido estaría disponible otra vez la información para el usuario y evitando pérdidas? debe lograrse gracias a la interoperabilidad en los recursos. Nuevamente, la implementación de interoperabilidad en los recursos es una tarea compleja, por las variantes que existen en configuración, conectividad y gestión; es necesario homogeneizar dichas variantes para lograr una interoperabilidad escalable. [1]

Destacando otras desventajas que encontramos en la computación en la nube, podemos mencionar:

- ✓ La disponibilidad de los servicios y aplicaciones está ligada a la disponibilidad del acceso a Internet
- ✓ A medida que más usuarios empiecen a compartir la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentará, si la empresa no posee un esquema de crecimiento óptimo puede llevar a degradaciones en el servicio.

II.5 CAPAS

Los proveedores de recursos en la nube pueden ofrecer sus servicios bajo diferentes capas, simulando las funciones ejecutadas por aplicaciones, sistemas operativos y el hardware físico. Podemos segregar estos servicios en tres tipos:

- ❖ Software como Servicio (SaaS – Software as a Service)
- ❖ Plataforma como Servicio (PaaS – Platform as a Service)
- ❖ Infraestructura como Servicio (IaaS – Infrastructure as a Service)

II.5.1 SOFTWARE COMO SERVICIO (SAAS – SOFTWARE AS A SERVICE)

Este enfoque ofrece aplicaciones completas las cuales se encuentran disponibles para el usuario final a través de un cliente ligero (típicamente, pero no necesariamente, un navegador web). Ejemplos prominentes de SaaS: Gmail, Google Docs y Salesforce.com. El usuario final no está involucrado en el diseño de la aplicación (salvo algunas

personalizaciones y serie de configuraciones), servidores, redes, o estructura de almacenamiento. Sin embargo, es posible recuperar cierta retroalimentación sobre el uso de las aplicaciones para introducir mejoras, cambios de imagen, automatización, etc. [4]

Es una nueva forma de distribuir el software, en donde el proveedor aloja la aplicación en sus propios servidores, dándole al cliente acceso a ella a través de una licencia. La empresa provee del servicio de mantenimiento diario, almacenamiento de la información y seguridad de los datos, liberando así al cliente del mantenimiento y los gastos asociados: servidores, problemas, costos, etc. [4]

El Software como Servicio (SaaS) está revolucionando el mercado de las aplicaciones para empresas. Algunas de las implicaciones o ventajas del uso de "Saas" son:

- ✓ Sus datos siempre estarán seguros al ser tratados por empresas absolutamente dedicadas a ello y siempre cumpliendo la normativa vigente.
- ✓ No se necesita de instalación, mantenimiento, ni personal calificado trabajando en la empresa. Puede controlar su aplicación sin demasiados conocimientos.
- ✓ Inversión inicial mucho menor: el costo de la aplicación se reparte en el tiempo, siendo mucho más accesible que un desembolso inicial de golpe. El costo, además es casi siempre menor que la compra de la aplicación.

- ✓ Fácil de usar: las aplicaciones se vuelven más sencillas al utilizar la interfaz de los sitios web. En pocos días el usuario estará capacitado y trabajando.
- ✓ Cualquier dispositivo con conexión a Internet permite que sus empleados trabajen a pleno rendimiento y en el lugar del mundo donde se encuentren. Solo con abrir un navegador web se accede a la información.
- ✓ Totalmente configurables, se ofrece como un software estándar, pero permite una personalización absoluta a la empresa cliente. Incluso se pueden hacer desarrollos personalizados sobre la aplicación a un costo muy inferior al desarrollo a la medida tradicional.
- ✓ Totalmente escalable: solo pagas por las necesidades del momento. Si su empresa crece, crece el número de licencias, y si decrecen, sus necesidades también el costo de la aplicación.
- ✓ Evita la renovación de equipos a corto plazo. Todos los equipos soportan una conexión a internet, y al menos un navegador web, por lo que puede aguantar con sus equipos funcionando sin necesidad de aumentar potencia o capacidad, durante mucho más tiempo. [9]



Figura 3. Esquema de negocio – Software como Servicio (SaaS) [10]

En la Figura 3 se observa que los clientes (usuarios finales), se suscriben a las aplicaciones en lugar de adquirirlas, bajo un esquema de pago establecido por la empresa proveedora del servicio. En este método de pago por uso de software, los clientes pueden disfrutar de un modelo flexible y de bajo costo que les permitirá concentrarse en la dirección de su negocio en lugar de realizar grandes inversiones en la compra de aplicaciones, además de no destinar tanto tiempo y recursos en parches y actualizaciones de software.

Cabe mencionar que no todo el Software como Servicio es proporcionado bajo un esquema económico; la remuneración se puede lograr, por ejemplo, adaptando un esquema de marketing publicitario, en donde a los usuarios que hagan uso del servicio se le presenten pequeños anuncios, ofertas y promociones de terceros acorde a los intereses del usuario (basados en historial de navegación, consultas y visitas de los sitios web), tal y como lo hace Gmail de Google.

II.5.2 PLATAFORMA COMO SERVICIO (PAAS – PLATFORM AS A SERVICE)

Bajo este esquema se ofrece un Sistema Operativo (OS) junto con colecciones de diversos lenguajes de programación y kits de desarrollo de software para que el usuario final pueda desarrollar sus propias aplicaciones. Existen diversos planes de contratación, en donde el usuario tiene la flexibilidad de adquirir solo los servicios que requiera.

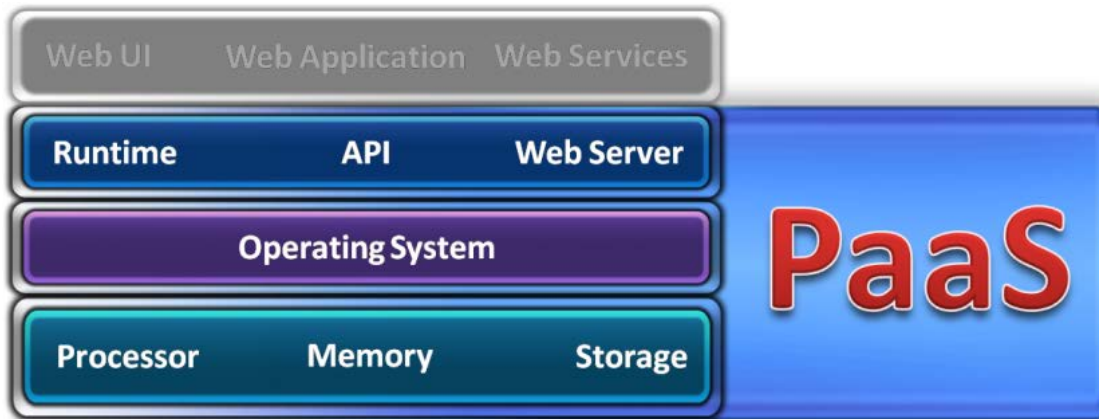


Figura 4. Encapsulamiento de una Plataforma como Servicio (PaaS) [21]

Ejemplos de PaaS: Microsoft Windows Azure y Google App Engine. En un esquema PaaS, el usuario tiene control sobre el diseño de la aplicación, pero no sobre la infraestructura física.

Analicemos ahora las características de un esquema PaaS, y de cómo algunos proveedores implementan dichas soluciones.

En un esquema PaaS, se crea una plataforma virtual para el desarrollo y distribución de aplicaciones. Los proveedores de un servicio PaaS normalmente ofrecen diversas herramientas que permiten al usuario u organización controlar a diferentes niveles el uso que se le da al servicio PaaS, se pueden monitorear factores como el número de veces que es usada cierta aplicación, el tiempo que se pasa con cierto programa, la actividad que se desarrolla con él, la eficiencia del software y los diversos problemas que se puedan experimentar. [22]

En algunas plataformas PaaS, como es el caso de Bungee Connect (<http://www.bungeelabs.com>), los desarrolladores pueden construir aplicaciones requiriendo de una conexión a internet. Mientras que en otras, como es el caso de Google App Engine (<https://developers.google.com/appengine>), el sistema PaaS crea una instancia del entorno de servicio online localmente, permitiendo al desarrollador trabajar sin una conexión a internet, para después subirla al proveedor del servicio PaaS, sincronizándola y almacenándola en el servidor remoto. [22]

Como se mencionó, PaaS integra lenguajes de programación y kits de desarrollo; la mayoría de los proveedores de servicio integran los lenguajes más comunes: C/C++, Java y PHP, así como herramientas drag-and-drop (arrastrar-y-colocar) que integran bloques de código predefinido, reduciendo tiempos y carga de trabajo.

Sin embargo, algunos proveedores como Bungee y Salesforce.com han desarrollado sus propios lenguajes. Argumentando que simplifican en gran medida el proceso de desarrollo y que son relativamente sencillos de aprender. [22]

Los defensores del servicio PaaS, argumentan los siguientes beneficios: incremento en la productividad, permite a las organizaciones poner a la venta sus aplicaciones más rápido y reducir costos en licencias de software. También se elimina la necesidad de que el desarrollador tenga que configurar su entorno de trabajo (servidor, red, almacenamiento, configuración, escalamiento, etc.), así como el trabajo de mantenimiento, seguridad y actualizaciones al servidor y sistema operativo. Todo esto es gestionado por el proveedor de servicio PaaS. [22]

Construir, depurar, probar y distribuir aplicaciones dentro de un solo entorno, reduce riesgos, al eliminar problemas causados cuando los programas son desarrollados bajo cierta configuración, pero ejecutados sobre otra. [22]

Otra ventaja que se tiene al utilizar PaaS es que se integra la colaboración entre los desarrolladores, ya que el código es gestionado en línea y se facilita el acceso, modificación y actualización.

ENFOQUES

Las plataformas PaaS pueden tomar diversos enfoques, desde el relativamente simple, en donde el desarrollador básicamente solo ensambla bloques de código generado por el proveedor; hasta un enfoque más complejo en donde se involucra el diseño y generación de código por parte del desarrollador.

Veamos algunos casos de plataformas PaaS y las soluciones que proponen.

ETELOS PLATFORM - <http://www3.etelos.com>

La plataforma Etelos permite a los desarrolladores crear aplicaciones utilizando el Etelos Development Environment (Entorno de Desarrollo Etelos) en lenguajes como PHP, Java Server Pages (JSP), C#, y el lenguaje propio de la compañía English Application Scripting Engine. La tecnología caching de Etelos permite al desarrollador trabajar sin conexión a internet para después sincronizar su trabajo con la aplicación en línea. Una vez terminada la aplicación, la organización puede hacer uso de ella o ponerla a la venta a través del servicio Etelos Marketplace.

BUNGEE LABS – <http://www.bungeelabs.com>

La plataforma Bungee Labs' Bungee Connect está enfocada a las pequeñas y medianas empresas (PyMES), así como a los departamentos dentro de una larga empresa. Incluye una interfaz de componentes drag-and-drop (arrastrar-y-colocar) así como su propio lenguaje basado en C, Bungee Logic.

La plataforma de la compañía permite distribuir las aplicaciones en servidores gestionados por Bungee o en servidores virtuales siendo ejecutados en Amazon's Elastic Computed Cloud (EC2). EC2 provee de una interfaz web de servicio escalable mediante la cual el desarrollador puede solicitar máquinas virtuales (VMs) en donde se puede cargar software.

Finalmente, las organizaciones pueden optar por comprar servidores Bungee, los cuales estarán autorizados para integrarse a la infraestructura de la empresa, proporcionando un alto nivel de control sobre el servidor y mayor flexibilidad en la auditoría interna.

SALESFORCE – <http://www.salesforce.com/platform>

La plataforma Force.com, perteneciente a Salesforce.com provee de alrededor de 60,000 programas que los usuarios (suscriptores) pueden usar e integrar para conformar una aplicación más robusta y/o compleja. También se proveen de las herramientas para construir una aplicación desde cero.

Force.com hace uso de un lenguaje de programación personalizado llamado Apex, que está enfocado en el modelado de bases de datos. Todas las aplicaciones construidas son específicas para su ejecución bajo la plataforma Force.com, sin embargo el usuario tiene la facilidad de respaldar localmente. [22]

GOOGLE APP ENGINE – <http://appengine.google.com>

La plataforma ofrece un entorno de desarrollo y de distribución escalable. Usa el popular lenguaje de script Python y como característica adicional, se incluye un sistema de autenticación para validar la identidad de los programadores que están participando en el proceso de desarrollo. La organización también puede utilizar este sistema para autenticar a los usuarios que hagan uso de la aplicación una vez terminada.

Aspectos en los que se debe trabajar e incertidumbres en las plataformas PaaS

Si el sistema PaaS se cae, si se carece de una conexión a internet cuando el proveedor no permite el trabajo local o más aún si el proveedor deja de proporcionar el servicio a causa de cierre del negocio, la organización se verá afectada, y particularmente si se trabajan con aplicaciones que son de vital importancia para las actividades cotidianas y de misión crítica.

Además, los desarrolladores están limitados a trabajar únicamente con la plataforma que el proveedor les proporciona. La aplicación una vez terminada, estará configurada para ser ejecutada únicamente por la plataforma elegida. Por tanto, si la organización desea migrar de plataforma, será una tarea compleja y tediosa, es por ello que antes de escoger una plataforma sea conveniente hace un estudio detallado y minucioso sobre las que existen en el mercado.

Dependiendo de los requerimientos y recursos con los que cuente determinada organización, se puede optar por la utilización de una plataforma PaaS para el desarrollo de aplicaciones, o desarrollar en un ambiente híbrido en donde las aplicaciones de misión crítica y de vital importancia sean ejecutadas dentro de la infraestructura de la organización, mientras que las aplicaciones para distribución de carga o que requieran de una mayor colaboración se trabajen en un esquema PaaS. A final de cuentas, la organización decidirá cómo y dónde desarrollará sus aplicaciones, teniendo a las plataformas PaaS como una opción más.

Finalmente observemos un diagrama que encapsula Software como Servicio (SaaS) y Plataforma como Servicio (PaaS) en Windows Azure.



Figura 5. Software como Servicio (SaaS) y Plataforma como Servicio (PaaS) en Windows Azure [23]

Se observa que las soluciones SaaS trabajan encima de la plataforma PaaS: Windows Azure.

II.5.3 INFRAESTRUCTURA COMO SERVICIO (IAAS – INFRASTRUCTURE AS A SERVICE)

Bajo este esquema se ofrece al usuario final acceso directo a recursos de procesamiento, almacenamiento, entre otros; a la vez que se les permite configurar y ejecutar sistemas operativos y software acorde a sus necesidades.

Los servicios IaaS son típicamente provistos por centros de datos (data centers), que rentan hardware moderno a los usuarios, liberando la preocupación que se acarrea por la depreciación, constante actualización y configuración. Una de las principales características de IaaS es la virtualización, que da la facilidad al usuario de ejecutar su propio sistema operativo huésped por encima del software virtualizado que proporciona el proveedor.

La virtualización en IaaS es un paso clave en la distribución, automatización y de una escalable distribución, instalación, y mantenimiento del software.

La siguiente tabla muestra las diferencias entre una IaaS y otros servicios.

	IaaS	Servicios Administrados	Data Centers Convencionales
Disponibilidad (Aprovisionamiento)	Bajo Demanda, instantáneo	Bajo demanda, de días a semanas	De semanas a meses
Escalamiento	Instantáneo, hacia arriba o hacia abajo	De semanas a meses, hacia arriba o hacia abajo	Meses, sólo hacia arriba
Costos	Sin pago por adelantado, “paga mientras consumes”	Pago por adelantado, contrato por un mínimo de tiempo, pagos mensuales fijos	Costo por equipo adquirido o rentado, cargos de mantenimiento, cargos extra por uso de ancho de banda
¿Automatizado?	Sí, vía consola en el escritorio	No	No automatizado

Tabla 1. Comparaciones de IaaS con otros servicios. [24]

La computación en la nube, al menos conceptualmente, ha venido adquiriendo cierta popularidad como un modelo para la IT (Tecnología de la Información), estratégico y sin muchas dificultades de acceso. Sin embargo, uno de los problemas que ha acarreado esta popularidad, es que los proveedores han comenzado a dar sus propias definiciones de lo que compromete un modelo basado en la nube, habitualmente basado en los servicios que ellos mismos ofrecen. [24]

Para romper toda confusión, el Instituto Nacional para Estándares y Tecnología de los Estados Unidos de América (U.S. National Institute for Standards and Technology – NIST) ha establecido cinco características esenciales que deben estar presentes en un servicio para que se puedan considerar en la nube:

1. *Bajo demanda y automatizado.* El cliente puede aprovisionarse de forma unilateral recursos de cómputo cuando lo requiera tales como, tiempos de servidor y almacenamiento, sin la necesidad de interacción con otra persona por parte del proveedor del servicio, es decir, todo el proceso debe ser automatizado. Esta característica es posiblemente la más importante para un servicio en la nube. [24]
2. *Acceso vía red amplia.* Las capacidades y el control de un servicio en la nube deben estar disponibles vía Internet (u otras redes, como las privadas) utilizando protocolos estándar, y bajo una amplia gama de plataformas, incluyendo dispositivos móviles. [24]
3. *Fuente de recursos.* Recursos físicos y virtuales son asignados y re-asignados de forma dinámica dependiendo de la demanda, resultando en ahorro de costos para el cliente. [24]
4. *Elasticidad eficiente.* Aprovisionamiento de recursos de cómputo casi inmediatos, para un rápido escalamiento hacia arriba, hacia abajo, y acorde a la demanda. [24]
5. *Servicio medido.* Los recursos de cómputo utilizados por el cliente son monitoreados, controlados, reportados y cargados en completa transparencia, habilitando un convenio de “paga-a-como-consumes”. [24]

Diferencias entre una IaaS y los otros modelos en la nube: PaaS y SaaS

PaaS es similar a IaaS, con la excepción de que en una PaaS se provee de un conjunto de lenguajes de programación y herramientas (la plataforma). Generalmente un servicio PaaS está enfocado a la comunidad de desarrolladores, PaaS se puede ver como una analogía a los servidores de aplicaciones, sólo que ofreciendo características de la nube. Debido a que el software de la plataforma depende del proveedor del servicio, PaaS limita el tipo de aplicaciones que se pueden construir.

SaaS es esencialmente la entrega de aplicaciones para la IT y los usuarios a través de Internet. Un servicio SaaS es análogo a un modelo cliente/servidor, con la excepción de que el servidor es sustituido por el centro de datos del proveedor, los clientes son simplemente los navegadores web en las PC de escritorio (o dispositivos móviles), y el servicio ofrece los beneficios de la nube, como la elasticidad y un plan de paga-a-como-consumes.

Los tres servicios de cómputo en la nube pueden ser vistos como una pila, con cada capa incrementando el nivel de especificidad, pero decrementando el control en las capas inferiores. Las tres capas de servicio se encuentran sobre una capa de virtualización, que a su vez, se sitúa sobre una capa física: servidores, almacenamiento, hardware de red, etc.

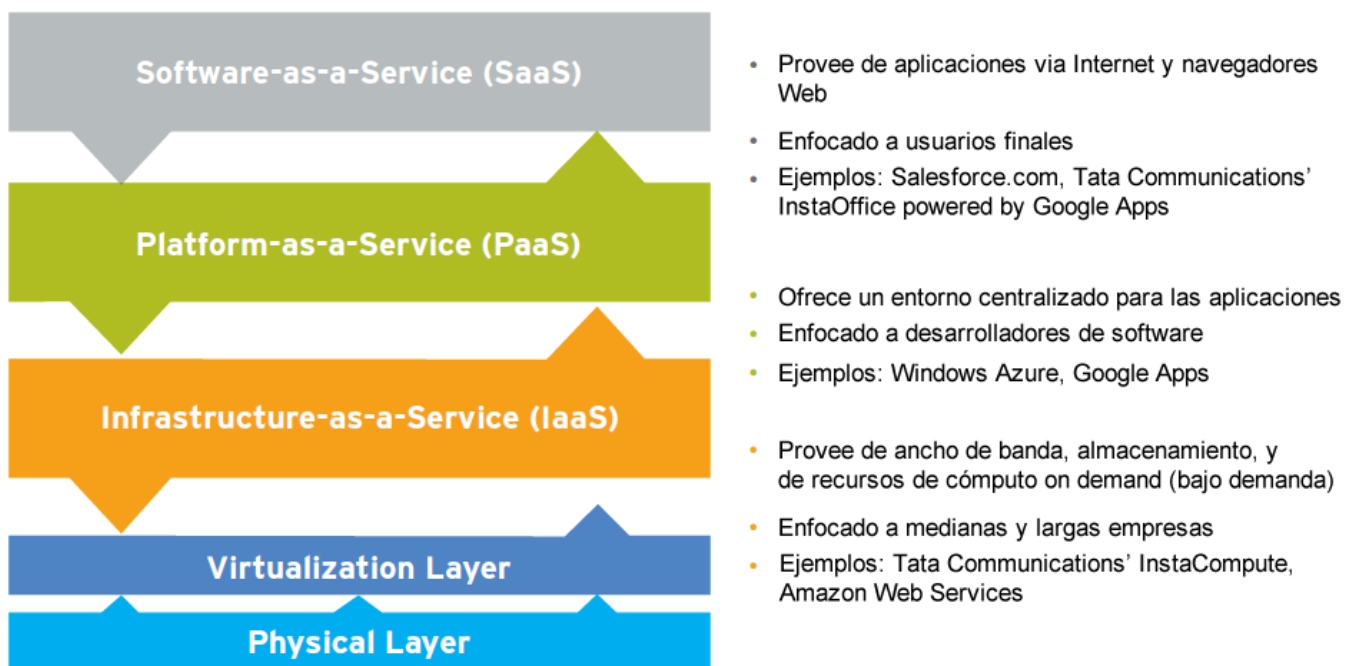


Figura 6. Pila de servicios en la nube: SaaS, PaaS, IaaS [24]

El término virtualización se refiere a la abstracción de cierta tecnología que se encuentra fuera de su entorno original. El beneficio radica en que se tiene la capacidad de proporcionar recursos de forma virtual (o lógica) con la misma funcionalidad y rendimiento que el original, pero con gran flexibilidad.

Un servicio de virtualización, no se debe confundir con un servicio IaaS, ya que carece de ciertas características de un modelo en la nube: aprovisionamiento bajo demanda, automatización y forma de cuantificar su uso. [24]

Algunas ventajas de adoptar un modelo IaaS son las siguientes:

- ✓ Disminución de costos en general
 - Los proveedores de servicio IaaS generalmente cuentan con un mayor poder adquisitivo que las pequeñas y medianas empresas
 - Ancho de banda con redistribución (y mejor eficiencia en su utilización) de la fuente de recursos de cómputo
- ✓ Evitar tecnología obsoleta
 - Con servicios de IT de alta eficiencia como base del negocio, los proveedores de servicio IaaS tienden a tener equipo de última generación y las últimas tecnologías del mercado
- ✓ Personal cualificado y capacitado
 - La escala de un proveedor de servicio IaaS le permite reclutar profesionales especialistas según requiera el negocio, mientras que en las pequeñas y medianas empresas algunas veces esto no es posible.

¿Por qué IaaS?

Las empresas lo que buscan es la reducción de costos. Una encuesta realizada por Yankee Group (<http://www.yankeegroup.com>), enfocada en este aspecto, ilustra los cinco motivantes principales para adquirir un servicio IaaS. En la siguiente gráfica se muestran los resultados obtenidos.

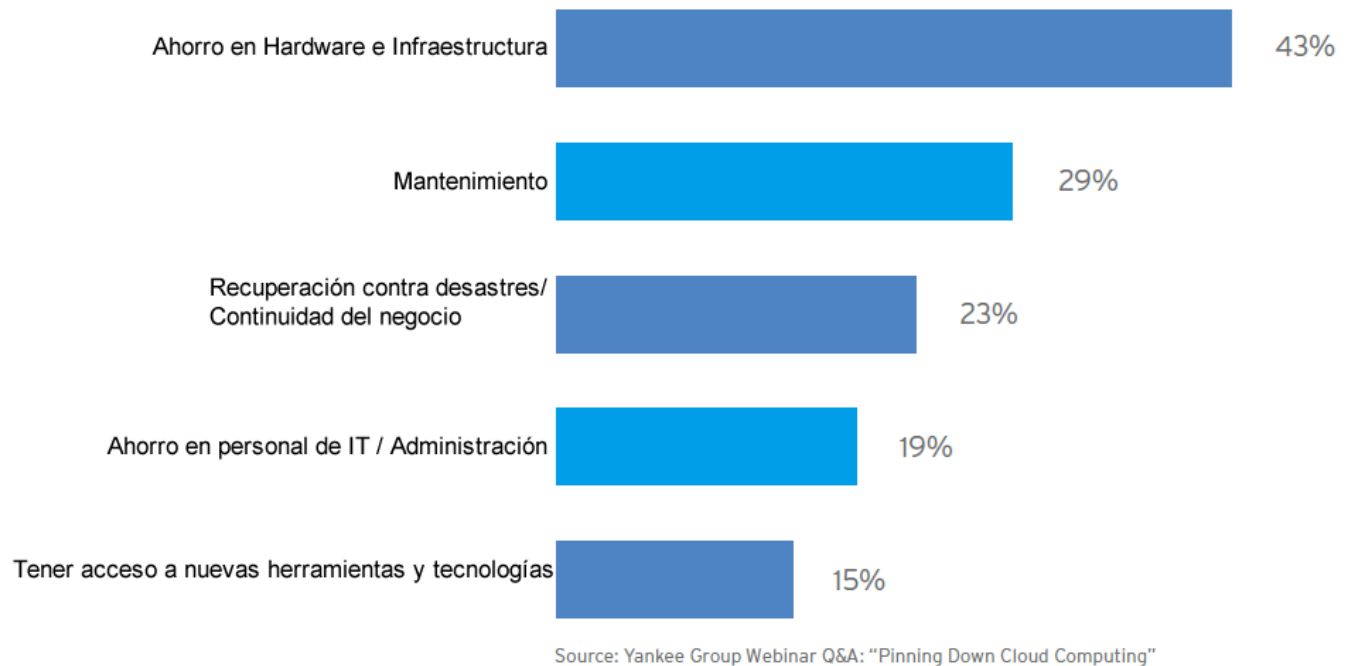


Figura 7. Los cinco principales motivantes para usar IaaS [24]

II.6 TIPOS DE NUBE

En la computación en la nube, se pueden tener tres tipos de nube:

- ❖ Nube privada
- ❖ Nube pública
- ❖ Nube híbrida

NUBE PRIVADA

En este tipo de nube, todos los servicios provistos son distribuidos a través de un centro de datos (data center) privado perteneciente a la organización. En estas nubes privadas se pueden distribuir aplicaciones y tecnologías propias, e inaccesibles al conjunto de servicios en la nube pública.

La nube privada es como "traer la computación en la nube a la casa" en vez de convertirla en un servicio provisto vía Internet. La idea es que se adquieren todos los beneficios de la computación en la nube, pero sin ceder control, seguridad, y costos a algún proveedor de servicio.

Las tecnologías involucradas en una nube privada son obvias, y de las que ya hemos tratado: virtualización (para una escalabilidad simple y sencilla, administración de recursos flexible, y utilización del hardware al máximo) y automatización del centro de datos (principalmente para el auto-provisionamiento de servidores físicos). [11]

Por una parte, la nube privada es un paso en la forma que se gestionan y administran los servicios y recursos de la IT, en donde el personal interino selecciona un servicio disponible dentro de una infraestructura virtualizada, en lugar de solicitar y asignar recursos permanentes a sus necesidades; por otro lado, la organización o empresa tiene la facilidad de implementar una nube híbrida para uso con clientes y asociados, acelerando el proceso de negociación y comercio en gran medida. [11]

Sin lugar a dudas, la nube privada es una opción viable para optimizar y generar comodidad en el uso de servicios dentro de la organización, sentando sus bases sobre una buena gestión de recursos virtualizados y teniendo el potencial de reducir costos, mantenimiento y constante monitoreo de la infraestructura.

En la siguiente figura se muestra un esquema de la infraestructura de una nube privada, en la que restringe el acceso utilizando un cortafuegos (firewall).

LA NUBE PRIVADA VISTA DESDE UN CENTRO DE DATOS PRIVADO

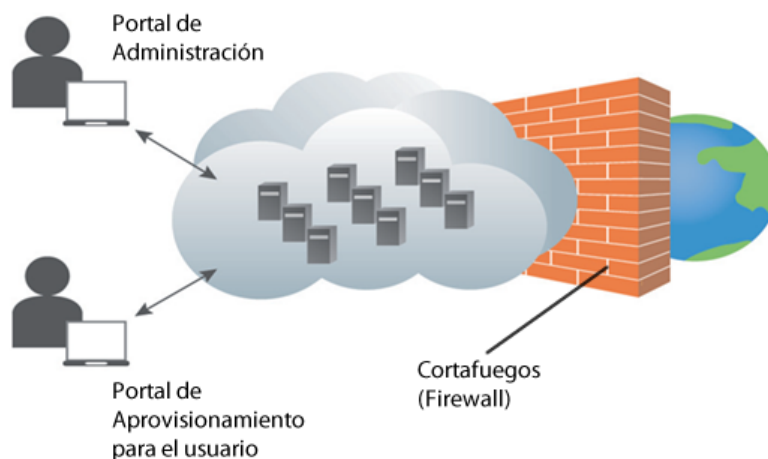


Figura 8. Infraestructura de una nube privada [12]

NUBE PÚBLICA

Si nos enfocamos en el término, público se refiere a que puede ser accedido por cualquiera desde cualquier parte del mundo a través de Internet. Es por eso que, en una nube pública se consolida el concepto de almacenamiento ilimitado y de un incremento en el ancho de banda a través de Internet para el negocio. Generalmente se trata de un sistema abierto que se encuentra disponible al público en general vía World Wide Web (WWW) o Internet. Algunos ejemplos de nubes públicas ya han sido mencionados y ejemplificados como: Amazon Elastic Compute Cloud (EC2), Google App Engine, Blue Cloud por IBM y Azure Services Platform por Microsoft. [13] [14]

La nube pública enfatiza su utilización al público en general, mientras que la nube privada se limita a un subconjunto de individuos dentro de la organización o negocio, y estableciendo permisos de uso público o privado interno. [13] [14]

Las características de este tipo de nube son las mismas que las de una nube privada e híbrida, que a su vez son heredadas del modelo de computación en la nube y que ya han sido tratadas, para evitar redundancia en el trabajo no se tratarán en este apartado. [13]

Sin embargo, para explicar mejor el concepto de automatización y aprovisionamiento veamos un caso de uso:

Caso: Supóngase que una organización no desea compartir información o almacenar archivos confidenciales en las estaciones de trabajo que posee. Entonces, ¿cuál sería la solución para compartir esta información con sus clientes? ¿Sería adecuado enviar e-mails masivos? Desde luego que no habría problema alguno en crear una lista de contactos de todos nuestros clientes y mandar un e-mail a todos ellos con un par de clics, sin embargo, ¿se podría cambiar la forma de interactuar con el cliente?, Si se desea proporcionar un servicio más personalizado, ya no se ve tan amigable el panorama utilizando métodos y servicios tradicionales de IT. Sin embargo, debe existir un modo eficiente para tratar este tipo de situaciones de la forma más sencilla posible. [13]

Solución: Adoptar y utilizar un servicio en la nube pública: Google App Engine. Creando una aplicación que permita interactuar con el cliente en dirección a nuestro negocio, en donde se cambie la forma de interactuar con el cliente, dándole la sensación de una atención personalizada. Se tiene un control y se provee de información sobre eventos, conferencias, compartición y acceso de archivos, etc. Rompiendo los tradicionales métodos de IT como es el caso del correo electrónico, en donde no se tiene un control sobre el tato con el cliente. [13]

Diagrama de representación de una nube pública:

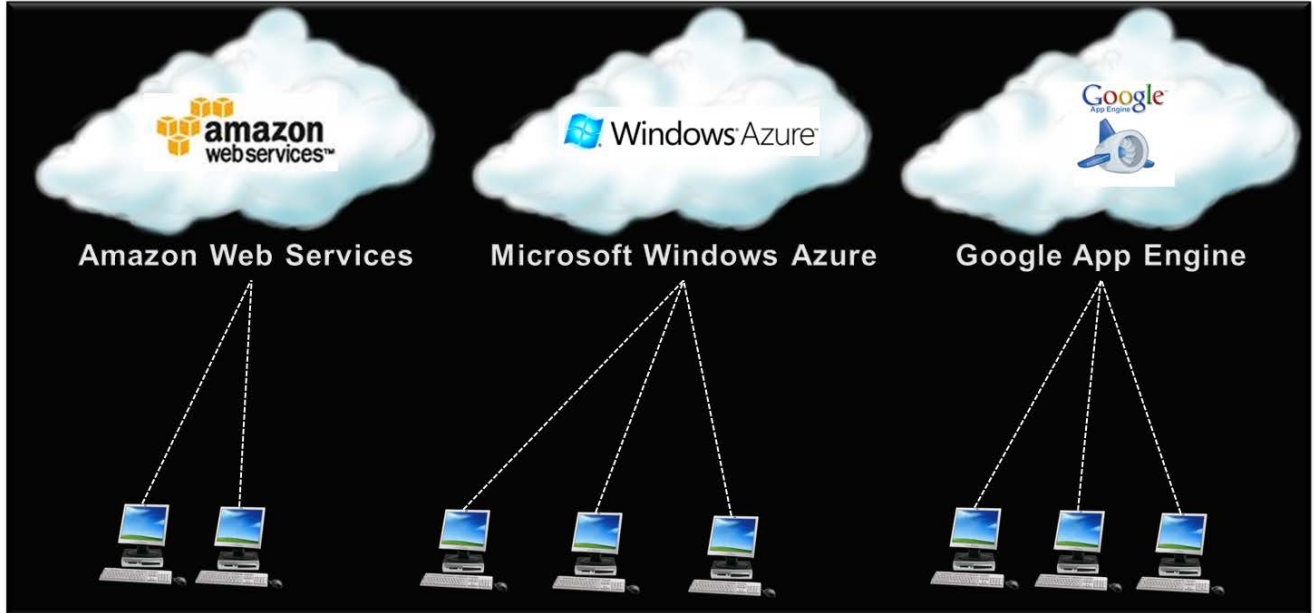


Figura 9. Servicios en la nube pública [15]

NUBE HÍBRIDA

En una nube pública, la organización o negocio distribuye sus servicios en dos partes, una a través de nube pública y la otra con una nube privada. Normalmente se tiene un centro de datos (data center) propio, pero en lugar de optar por expandir los recursos de cómputo físicamente, se contrata un servicio en la nube para tal fin. [16]

En otras palabras, una nube híbrida no es más que la combinación de nube pública y privada. Si citamos la definición del Instituto Nacional de Estándares y Tecnología (NIST): “Una nube híbrida, es la combinación de una nube privada y pública enlazadas entre sí por tecnología ya sea estandarizada o propietaria, que permita la portabilidad de los datos y las aplicaciones”. [16]

Una nube híbrida ofrece la escalabilidad y ahorro en costos de una nube pública, a la vez que el control y seguridad de una nube privada.

Casos de uso para una nube híbrida

Existen muchos escenarios que denotan el uso de nubes híbridas, pero a continuación citaremos los más característicos.

- Utilizar una nube privada para aplicaciones de misión crítica, mientras que las restantes puedan estar en una nube pública. Por ejemplo, una compañía puede utilizar una nube pública para pruebas y desarrollo, y una nube privada para producción y distribución. Otro ejemplo sería el de utilizar a la nube pública para aplicaciones de imagen pública, y una nube privada para aplicaciones internas, enfocadas a la operación y colaboración entre los desarrolladores. [16]
- Incremento en la eficiencia por cierto período de tiempo (cloudbursting). Hacer que una aplicación sea distribuida dinámicamente, originalmente ejecutada en una nube privada, pero por cuestiones de demanda, migrarla para su ejecución en una nube pública, como sería el caso de una compañía dedicada a ventas, que necesita de un mayor ancho de banda por el incremento en el tráfico durante las ventas en días festivos. [16]
- Pruebas contra Recuperación de Desastres (Disaster Recovery –DR) de forma no destructiva. Las compañías y organizaciones pueden probar si su entorno de producción está listo contra Recuperación de Desastres (DR), recurriendo a la nube pública sin afectar su entorno privado y producir interrupciones en los servicios. [16]

Las nubes híbridas ofrecen de una gran flexibilidad en los negocios en términos de seguridad y control. También son implementadas por organizaciones que requieren de distribuir su carga de trabajo, o para una integración más rápida de los proyectos.

Finalmente mostramos la figura que ejemplifica en dónde se posiciona una nube híbrida con respecto a una privada y una pública.

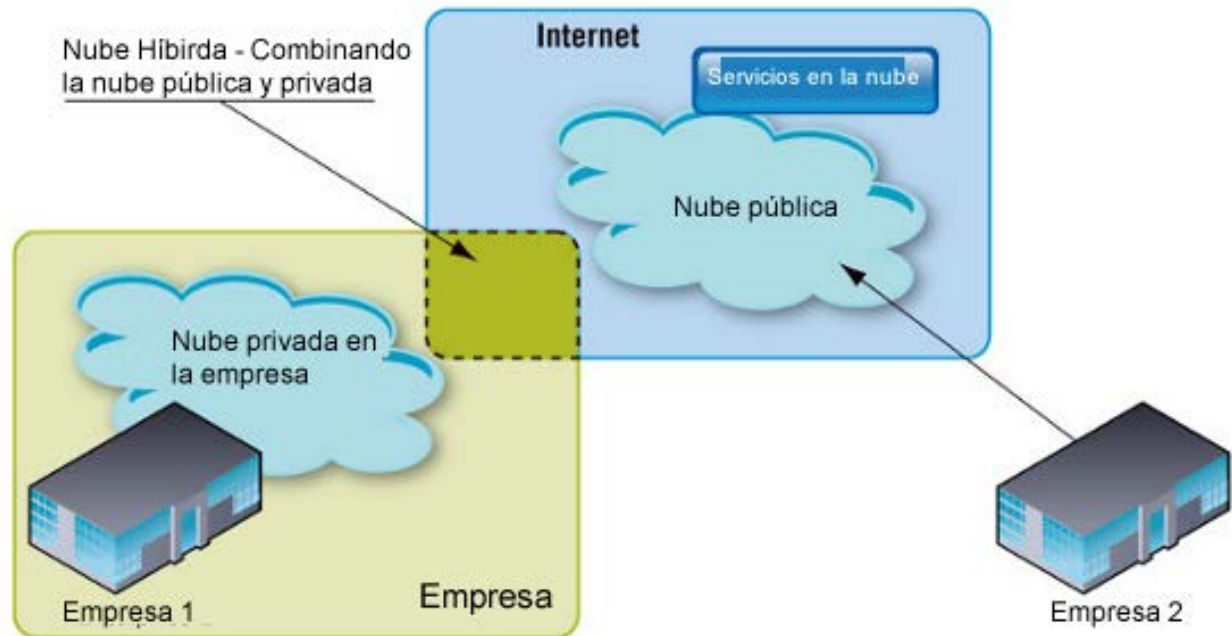


Figura 10. La nube híbrida [17]

II.7 COMPARACIONES

La computación en la nube está basada en otros modelos, de los que destacan: la computación en cluster (cluster computing) y computación en grid (grid computing).

Para que podamos comparar el cómputo en la nube, describamos brevemente los otros dos modelos.

COMPUTACIÓN EN CLUSTER (CLUSTER COMPUTING)

Por muchos años, las supercomputadoras fueron líderes en el campo de poder de cómputo. Pero debido a los problemas que iban surgiendo en las áreas de la ciencia, ingeniería, y negocios, que no eran resueltos de forma eficiente utilizando supercomputadoras; se optó por reemplazar a las supercomputadoras por clusters (arreglos de computadoras interconectadas), alcanzando el mismo poder de cómputo pero a un costo mucho menor. [25]

Definición. Un cluster es una colección de computadoras interconectadas en paralelo o de forma distribuida utilizando redes de alta velocidad, como gigabit Ethernet. [25]

Trabajan en conjunto para la ejecución de tareas intensivas que requieren una alta capacidad en procesamiento y carga de datos, y que no sería eficiente procesarlas en una sola computadora. [25]

Los clusters son utilizados principalmente para tener alta disponibilidad, balanceo de cargas de trabajo y trabajo de procesamiento. Cuando múltiples computadoras se encuentran enlazadas en el cluster, se distribuye la carga de trabajo entre ellas pero como si se tratara de una sola computadora virtual. Normalmente un cluster está enfocado a trabajos de procesamiento y no para actividades que requieran interacción con el usuario, como peticiones de entrada-salida. [25]

COMPUTACIÓN EN GRID (GRID COMPUTING)

En grid computing se combinan equipos de cómputo que están administrados por diferentes dominios, para alcanzar una meta en común, como resolver una simple tarea, y desaparecer tan rápido se termine de procesarla. [25]

En grid computing también se involucra el cómputo distribuido, lo que a su vez integra sistemas de cluster a gran escala. El tamaño de un grid puede variar, desde una red de computadoras dentro de la organización, hasta la colaboración a través de varias redes y compañías. [25]

Definición. Es un tipo de sistema paralelo y distribuido que permite colaborar, seleccionar y agregar recursos de cómputo geográficamente distribuidos y autónomos, de forma dinámica, en tiempo real y acorde a la disponibilidad, capacidad, eficiencia, costo y requerimientos del usuario. [25]

En la siguiente tabla se muestran las diferencias y similitudes de la computación en la nube con los dos modelos descritos.

	Cluster	Grid	Cloud (Nube)
Acuerdo de Nivel de Servicio (SLA)	Limitado	Si	Si
Asignación	Centralizada	Descentralizada	Ambas
Manejo de Recursos	Centralizado	Distribuido	Ambos
Acoplamiento flexible	No	Ambos	Si
Protocolos / API	MPI, Parallel Virtual	MPI, MPICH-G, GIS, GRAM	TCP/IP, SOAP, REST, AJAX
Confiabilidad	No	Media	Total
Seguridad	Si	Media	No
¿Amigable para el usuario?	No	Media	Si
Virtualización	Media	Media	Si
Interoperabilidad	Si	Si	Media
Estandarización	Si	Si	No
Modelo de Negocio	No	No	Si
Carga de trabajo	Pequeña a larga	Pequeña a larga	Pequeña y mediana
SOA	No	Si	Si
Eficiencia del Sistema	Mejora	Mejora	Mejora
Automatización	No	Si	Si
Servicio de Computo	Procesamiento	Procesamiento en su totalidad	Bajo demanda
Heterogeneidad	No	Si	Si
Escalabilidad	No	Media	Si
Bajo costo	No	No	Si
Localidad de datos explotados	No	No	Si
Aplicaciones	HPC, HTC	HPC, HTC, Batch	Aplicaciones interactivas en PyMES
Costo de migración	Bajo	Bajo	Alto
Valor añadido de los servicios	No	Medio	Si

Tabla 2. Comparativas: Computación en la nube, grid y cluster [25]

II.8 LENGUAJES PARA PROGRAMAR EN LA NUBE

La mayoría de los lenguajes de programación pueden ser usados para programar en la nube, o mejor dicho, para programar en la Web. Sin embargo, no todos ellos son los más adecuados para ello.

Por ejemplo, C/C++ puede integrarse para una programación Web utilizando lenguajes de apoyo para su integración, como Lua.

A continuación describiremos los lenguajes y librerías más poderosas y comunes que podemos utilizar para programar en Web de la forma más robusta, segura y eficiente.

HTML - XHTML

Definiéndolo de forma sencilla, "HTML es lo que se utiliza para crear todas las páginas web de Internet". Más concretamente, HTML es el lenguaje con el que se "escriben" la mayoría de páginas web. [18]

Los diseñadores utilizan el lenguaje HTML para crear sus páginas web, los programas que utilizan los diseñadores generan páginas escritas en HTML y los navegadores que utilizan los usuarios muestran las páginas web después de leer su contenido HTML. [18]

Aunque HTML es un lenguaje que utilizan los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas. HTML son las siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto). [18]

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C (<http://www.w3.org>). Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. [18]

El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online (e-commerce) y banca electrónica. [18]

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML (Standard Generalized Markup Language), mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML). [18]

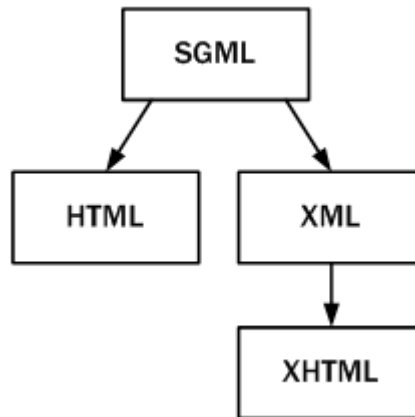


Figura 11. Esquema de la evolución de HTML y XHTML [18]

Las páginas y documentos creados con XHTML son muy similares a las páginas y documentos HTML. Las discusiones sobre si HTML es mejor que XHTML o viceversa son recurrentes en el ámbito de la creación de contenidos web, aunque no existe una conclusión ampliamente aceptada. [18]

Actualmente, entre HTML 4.01 y XHTML 1.0, la mayoría de diseñadores escogen XHTML. En un futuro cercano, si los diseñadores deben elegir entre HTML 5 y XHTML 1.1 o XHTML 2.0, quizás la elección sea diferente. [18]

El lenguaje HTML original era muy permisivo en su sintaxis, por lo que era posible escribir sus etiquetas y atributos de muchas formas diferentes. Las etiquetas por ejemplo podían escribirse en mayúsculas, en minúsculas e incluso combinando mayúsculas y minúsculas. El valor de los atributos de las etiquetas se podían indicar con y sin comillas ("). Además, el orden en el que se abrían y cerraban las etiquetas no era importante. [18]

La flexibilidad de HTML puede parecer un aspecto positivo, pero el resultado final son páginas con un código HTML desordenado, difícil de mantener y muy poco profesional. Afortunadamente, XHTML soluciona estos problemas añadiendo ciertas normas en la forma de escribir las etiquetas y atributos. [18]

A continuación se muestran las cinco restricciones básicas que introduce XHTML respecto a HTML en la sintaxis de sus etiquetas: [18]

1. Las etiquetas se tiene que cerrar de acuerdo a como se abren:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con <a>un enlace</a></p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con <a>un enlace</p></a>
```

2. Los nombres de las etiquetas y atributos siempre se escriben en minúsculas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con  
    <a href="http://www.google.com">un enlace</a>  
</p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<P>Este es un párrafo con  
    <A HREF="http://www.google.com">un enlace</A>  
</P>
```

3. El valor de los atributos siempre se encierra con comillas:

Ejemplo correcto en XHTML:

```
<p>Este es un párrafo con  
    <a href="http://www.google.com">un enlace</a>  
</p>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<p>Este es un párrafo con  
    <a href=http://www.google.com>un enlace</a>  
</p>
```

4. Los atributos no se pueden comprimir:

Ejemplo correcto en XHTML:

```
<dl compact="compact">...</dl>
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<dl compact>...</dl>
```

Este tipo de atributos en los que el nombre coincide con su valor no son muy habituales.

5. Todas las etiquetas deben cerrarse siempre:

La mayoría de etiquetas HTML encierran un contenido de texto entre la etiqueta de apertura y la etiqueta de cierre. Sin embargo, algunas etiquetas especiales llamadas "etiquetas vacías" no necesitan encerrar ningún texto. [18]

La etiqueta `
` por ejemplo, se utiliza para indicar el comienzo de una nueva línea, tal y como se verá más adelante. Por sus características, la etiqueta `
` nunca encierra ningún contenido de texto.

Como el estándar XHTML obliga a cerrar todas las etiquetas abiertas, siempre que se incluya la etiqueta `
` se debería cerrar de forma seguida: `
</br>`. Para que el código resulte más cómodo de escribir, XHTML permite en estos casos escribir de forma abreviada una etiqueta que se abre y se cierra de forma consecutiva.

En lugar de abrir y cerrar de forma consecutiva la etiqueta (`
</br>`) se puede utilizar la sintaxis `
` para indicar que es una etiqueta vacía que se abre y se cierra en ese mismo punto. En la forma compacta es habitual equivocarse con la posición del carácter `/`. [18]

Ejemplo correcto en XHTML:

```
<br />
```

Ejemplo incorrecto en XHTML (pero correcto en HTML):

```
<br>
```

CSS (CASCAIDING STYLE SHEETS) – HOJAS DE ESTILO EN CASCADA

Originalmente, las páginas HTML sólo incluían información sobre sus contenidos de texto e imágenes. Con el desarrollo del estándar HTML, las páginas empezaron a incluir también información sobre el aspecto de sus contenidos: tipos de letra, colores y márgenes.

Incluir en una misma página HTML los contenidos, el diseño y la programación complica en exceso su mantenimiento. Normalmente, los contenidos y el diseño de la página web son responsabilidad de diferentes personas, por lo que es conveniente separarlos.

CSS es el mecanismo que permite separar los contenidos definidos mediante XHTML y el aspecto que deben presentar esos contenidos: [18]

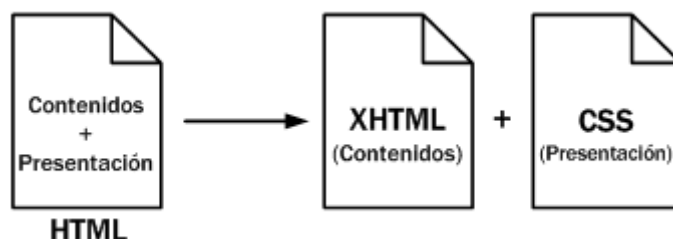


Figura 12. Esquema de la separación de los contenidos y su presentación [18]

Una ventaja añadida de la separación de los contenidos y su presentación es que los documentos XHTML creados son más flexibles, ya que se adaptan mejor a las diferentes plataformas: pantallas de ordenador, pantallas de dispositivos móviles, impresoras y dispositivos utilizados por personas discapacitadas. [18]

JAVASCRIPT / JQUERY

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Javascript puede ser un lenguaje complejo ya que está ligado directamente con la manipulación del DOM (Document Object Model) de una página. Es por eso que para su simplificación se utilizan librerías que faciliten dicha manipulación, una de las más usadas y adoptadas actualmente es jQuery (<http://jquery.com/>).

jQuery es una librería JavaScript rápida y concisa que simplifica la forma en que se recorre un documento HTML, el manejo de eventos, animación, e interacciones AJAX para un desarrollo web rápido y efectivo. En otras palabras: "jQuery está diseñado para cambiar la forma en que se escribe Javascript".

PHP – HYPERTEXT PREPROCESSOR

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.

Todo el código PHP embebido en una página o documento HTML será ejecutado en el servidor (servidor web) antes de que se envíe la página al usuario que la requiera, de esta manera, se pueden generar estructuras de programación de alto nivel que requieran de una lógica más compleja.

Algunas compañías y organizaciones hacen uso de PHP a través de un framework de desarrollo, que integra un modelo o patrón de abstracción para el desarrollo de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos.

La mayoría de los frameworks de desarrollo para PHP incorporan el modelo MVC (modelo-vista-controlador).

Como el sistema desarrollado en este trabajo utiliza PHP, describiremos brevemente las diferencias que existen entre PHP y otros lenguajes.

PHP vs. ASP

ASP no es realmente un lenguaje en sí, pero es un acrónimo para Active Server Pages; los actuales lenguajes usados para programar ASP incluyen Visual Basic Script, JScript, and C#, entre otros. El mayor inconveniente de ASP es ser un sistema propietario y que es nativo y usado únicamente por Microsoft Internet Information Server (IIS). Esto limita su disponibilidad únicamente a servidores con WIN32. Hay un par de proyectos que permiten a ASP funcionar con otros ambientes y webservers. InstantASP de Halcyon (Comercial - <http://www.instantasp.co.uk/>) Chili!Soft ASP de Chili!Soft (Comercial) y Mono (open-source - http://www.mono-project.com/Main_Page). Se dice que ASP es más lento y más incómodo que PHP, con menos estabilidad. Una de las ventajas de ASP es que, al usar VBScript, es relativamente fácil de aprender si se es bueno con Visual Basic. ASP es habilitado por defecto en IIS, haciéndolo fácil de obtener y hacerlo funcionar. Sin embargo, los componentes hechos en ASP son realmente limitados, así que si se necesitan características "avanzadas" (como interactuar con servidores FTP), se necesitará comprar componentes adicionales. [26]

PHP vs. ColdFusion

Se dice comúnmente que PHP es más rápido y más eficiente para tareas de programación compleja, así como para probar nuevas ideas, y es considerado por muchos el más estable y el que usa menos recursos intensivos. Mientras que ColdFusion tuvo en algún momento un buen manejo de los errores, abstracción de bases de datos, y manejo de fechas, la abstracción de bases de datos dejó que desear cuando fue abordado en PHP 4. Una particular cualidad que es listada como una de las fuerzas de ColdFusion es su excelente motor de búsqueda, pero muchos argumentos sólidos sugieren que un motor de búsqueda no es algo que debería ser incluido en un lenguaje de scripts para la WEB. Además, PHP se ejecuta en casi todas las plataformas modernas que existen, mientras que ColdFusion está disponible únicamente para Windows, Solaris, Linux, MacOS, y AIX.

ColdFusion tiene un buen IDE y es generalmente rápido para dar resultados para los nuevos programadores, mientras que PHP requiere de más conocimiento de programación. ColdFusion ha sido diseñado teniendo en mente a los que no son programadores, mientras que PHP está enfocado en los programadores. [26]

PHP vs. Perl

La mayor ventaja de PHP sobre Perl es que PHP fue diseñado para hacer scripts para la web, mientras que Perl fue diseñado para muchas otras tareas. Por esto, Perl puede volverse complicado. La flexibilidad / complejidad de Perl pueden hacerlo de difícil colaboración por parte de desarrolladores de diferentes niveles. PHP es menos confuso y estricto en su formato, pero sin perder la flexibilidad. PHP es también fácil de integrar en un HTML existente a diferencia de Perl. En gran medida, PHP tiene todas las bondades de Perl - construcciones, sintaxis, etc. - sin hacer las cosas tan complicadas. Además, el intérprete de la línea de comandos de PHP (CLI por sus siglas en inglés) es suficientemente poderoso para realizar tareas de alto nivel en la misma forma en que Perl ha sido tradicionalmente empleado. Perl es uno de los lenguajes más usados y un verdadero lenguaje, y se ha mantenido firme desde los 80's, pero PHP ha madurado y evolucionado rápido, y continúa haciendo un progreso fantástico. [26]

Finalmente, en el sistema se hace uso de una base de datos relacional, y por tanto es necesario un lenguaje para la consulta de datos, en este caso, SQL.

SQL – STRUCTURED QUERY LANGUAGE

Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella. [19]

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros —y no a registros individuales— permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a una o más sentencias que se utilizarían en un lenguaje convencional de alto nivel para efectuar la misma operación. [19]

En SQL existen dos clasificaciones para las sentencias que manipulan la base de datos.

Lenguaje de Definición de Datos (DDL) es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen cuatro operaciones básicas: CREATE, ALTER, DROP y TRUNCATE. [19]

Lenguaje de Manipulación de Datos (DML) permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado. Incluye las operaciones: SELECT, UPDATE, DELETE, INSERT al igual que algunas cláusulas de condicionamiento: WHERE, JOIN, ON, LIKE, IN. [19]

Generalmente los programadores y desarrolladores de software trabajan más con el DML, mientras que los administradores de bases de datos (DBAs) utilizan ambos, además de funciones avanzadas del lenguaje SQL como disparadores (TRIGGERS), procedimientos almacenados (STORED PROCEDURES) y funciones (FUNCTIONS). [19]

PYTHON

Python es un lenguaje de programación poderoso y dinámico utilizado para el desarrollo de un amplio dominio de aplicaciones. Algunas de sus características más representativas incluyen:

- ❖ Sintaxis limpia y legible
- ❖ Orientado a objetos de forma intuitiva
- ❖ Expresión natural de código procedimental
- ❖ Completamente modular, soportando paquetes de forma jerarca

- ❖ Resolución de errores bajo un sistema de excepciones
- ❖ Extensiones y módulos escritos en C, C++
- ❖ Embebido en aplicaciones como interfaz de script

Python puede ser ejecutado en la mayoría de los sistemas operativos actuales: Windows, Linux/Unix, OS/2, Amiga, entre otros.

La implementación de Python se encuentra bajo la licencia de software libre, que lo convierte de libre distribución y uso, inclusive para su uso comercial.

Para el desarrollo web, con Python podemos lograr:

- ✓ Escritura de scripts CGI
- ✓ Utilización de frameworks de desarrollo como Django y TurboGears
- ✓ Soluciones de alto nivel como Zope (servidor de aplicaciones)
- ✓ Avanzados sistemas de manejador de contenidos como Plone
- ✓ Procesamiento de E-mail y RSS
- ✓ Soporte para diversos protocolos de internet como FTP, POP3, SMTP, etc. [28]

JAVA SERVER PAGES (JSP)

La tecnología de Java Server Pages (JSP), permite a los desarrolladores web y diseñadores trabajar en aplicaciones Web dinámicas de forma rápida, eficiente, y de fácil mantenimiento. En JSP se separa la interfaz de usuario del generador de contenido, permitiendo, por ejemplo, cambiar el diseño de la página sin afectar el contenido que se genera dinámicamente.

Para un desarrollador web familiarizado con HTML, se puede:

- Utilizar la tecnología JSP sin necesidad de aprender Java
- Extender el lenguaje JSP implementando un API
- Facilidad de escritura y mantenimiento de las páginas

En JSP se utilizan etiquetas tipo XML que encapsulan la lógica que genera el contenido para las páginas. La lógica de aplicación puede residir en recursos del servidor y ser accedida por las páginas con las etiquetas de tipo XML.

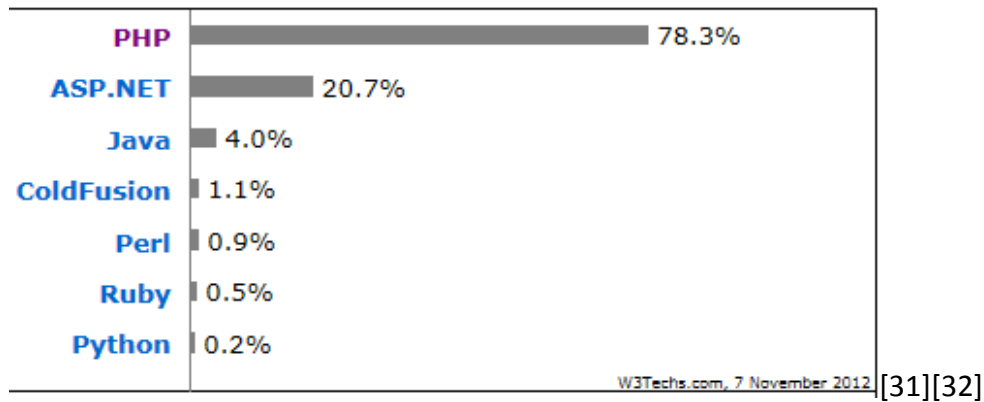
La tecnología JSP es una extensión de los Java Servlets. [29]

.NET

.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporation y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo. [30]

A continuación se presenta una gráfica de los lenguajes más utilizados actualmente para el desarrollo de aplicaciones Web, independientemente de si se encuentran trabajando bajo un esquema en la nube o no:



III. UN SISTEMA WEB INNOVADOR PARA LA GESTIÓN DE LA EFICIENCIA ENERGÉTICA EN HOTELES

III.1 NECESIDAD ACTUAL: AUTOMATIZACIÓN Y AHORRO DE LAS FUENTES ENERGÉTICAS. LA IDEA DE UTILIZAR UN SISTEMA COMPUTARIZADO PARA LA GESTIÓN

Habiendo descrito las tecnologías de la información (TI) con las que se construyó el sistema, toca el turno de observar la implementación de las mismas, pero antes, veamos el diagrama de bloques de las funciones que tiene el sistema, así como el modelo relacional de la base de datos a la que se ligarán los módulos del sistema.

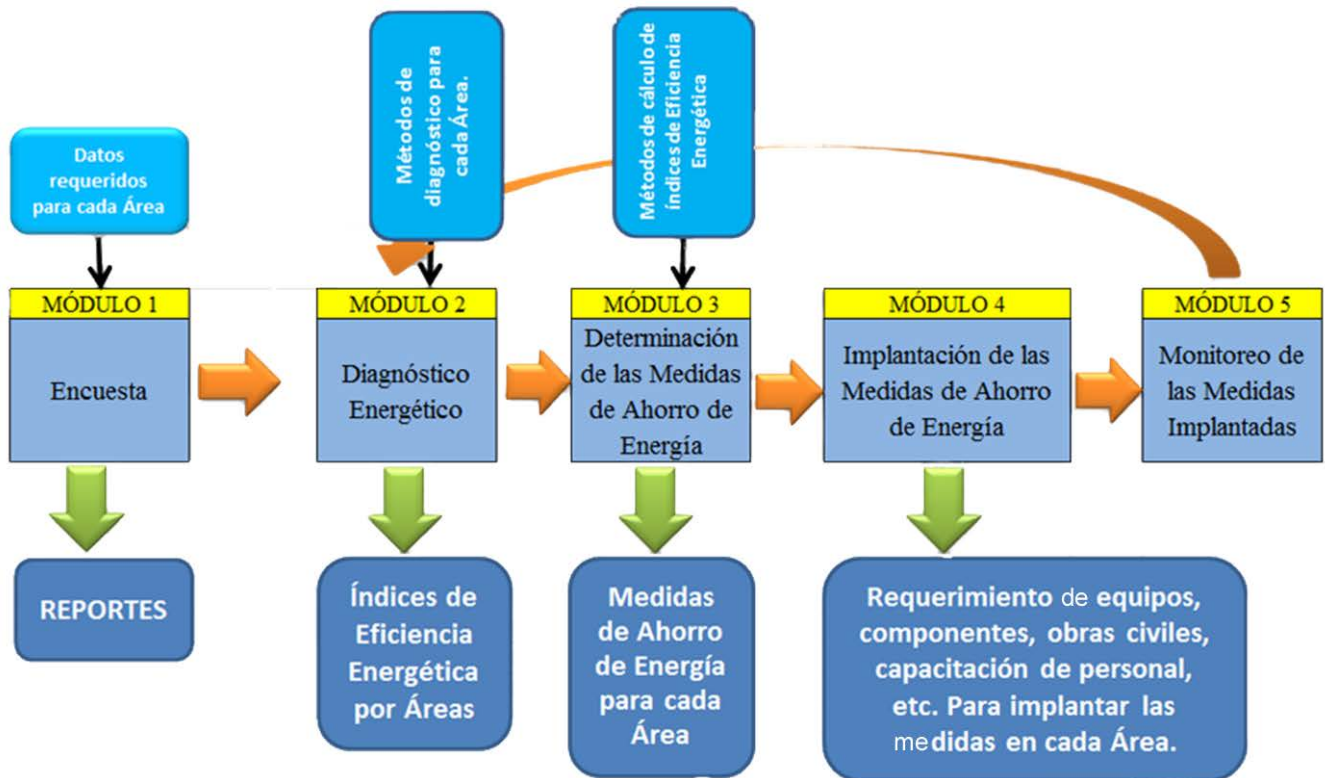
DIAGRAMA DE BLOQUES DE EFICIENCIA ENERGÉTICA EN HOTELES

Figura 13. Diagrama de bloques del sistema web

El sistema analizará las siguientes áreas, en dónde se aplicarán los cinco módulos presentados en el diagrama de la figura 13.

- Iluminación
 - Interior
 - Exterior
- Aire Acondicionado
- Calentamiento de Agua
- Refrigeración
- Elevadores
- Bombeo

En este trabajo, de todas las áreas analizadas, se explicará a detalle la implementación para Aire Acondicionado, ya que por su extensión y complejidad es el más adecuado para comprender el alcance y objetivo del sistema.

Con este sistema, los hoteles tendrán una herramienta que les permitirá determinar problemas de gestión en la eficiencia energética, evitando la contratación de servicios profesionales o de planta para levantamientos y auditorías energéticas. Ya que el sistema adapta el conocimiento de expertos que permitirá proporcionar y evaluar medidas de ahorro acorde a los datos proporcionados por los hoteleros para las diferentes áreas de análisis.

Antes de llegar a trabajar con los módulos del sistema, es necesario establecer un módulo de autenticación y registro de usuarios, así como de los hoteles respectivos, el cual estará ligado a cada usuario y hotel en particular, que a su vez se ligará con los cinco módulos propios del sistema.

Para entender mejor el flujo de registro y trabajo del sistema, analicemos el diagrama relacional de la base de datos del sistema ligado estrechamente con el área de aire acondicionado.

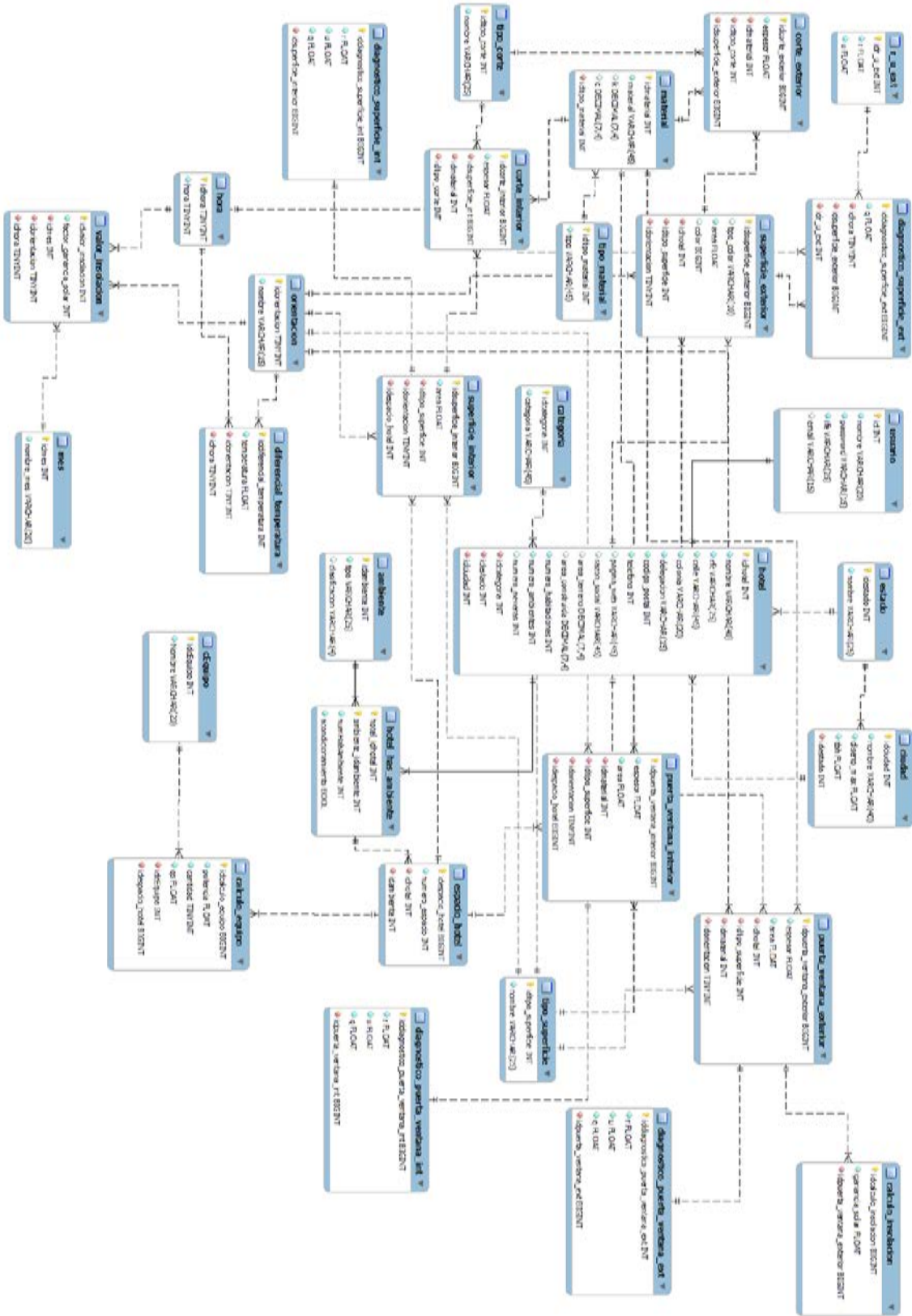


Figura 14. Modelo Entidad-Relación (EER) de la base de datos del sistema web

Se tiene que, de las relaciones más significativas:

- ❖ Un usuario solo podrá tener registrado un hotel, a la vez que un hotel solo podrá estar asociado con un solo usuario. Relación 1-1 entre Usuario – Hotel
- ❖ El hotel puede estar conformado por varios ambientes (habitaciones simples, habitaciones dobles, habitaciones triples, suites, cocinas, restaurantes, etc.), y un determinado ambiente se puede asociar con varios hoteles. Relación N-N entre Hotel – Ambiente
- ❖ Para cada ambiente que tiene el hotel, pueden existir varios espacios, (por ejemplo, hay 15 habitaciones dobles y 10 habitaciones simples), y un espacio pertenece a solo un ambiente. Relación 1-N entre Ambiente – Espacio
- ❖ Para el área de aire acondicionado se necesitan datos de paredes, techo y piso, así como de sus acabados, tanto interiores como exteriores.
- ❖ Las superficies exteriores se asocian al hotel, pudiendo tener varias superficies exteriores, pero cada una estará asociada solo con un hotel. Relación 1-N entre Hotel – Superficie Exterior
- ❖ Las superficies interiores estarán asociadas con cada espacio para cada ambiente, pudiendo tener varias superficies interiores, pero cada una estará asociada solo con un espacio en particular, independiente del ambiente al que pertenezca el espacio. Relación 1-N entre Espacio – Superficie Interior
- ❖ Los diagnósticos estarán asociados con sus respectivas superficies, ya sean interiores o exteriores, cada diagnóstico pertenecerá a una sola superficie, y la superficie podrá tener solo un diagnóstico. Relación 1-1 entre Diagnóstico – Superficie Interior/Exterior

El sistema está programado para ser ejecutado bajo la siguiente configuración:

- ✓ Equipo Servidor
 - Dell PowerEdge T710
 - Procesador: Intel Xeon serie 5500
 - Memoria: 12GB DDR3
- ✓ Base de Datos
 - Server Version: 5.5.15 MySQL Community Server (GPL)

- ✓ Servidor Web
 - Apache 2.2.20
- ✓ Lenguajes de programación y librerías
 - XHTML 1.0 Transitional
 - Javascript / jQuery 1.6.1
 - CSS 2.1
 - PHP 5.3.8

Sin embargo, ejecutando el sistema bajo un esquema cloud (en la nube) se tendría la siguiente configuración:

- ✓ Equipo para controlador Nodo de la Nube (Cloud Controller node), ejecuta y gestiona la red, almacenamiento, API, programador de tareas y servicios de imagen)
 - Procesador 64-bit ó x86
 - 12GB de Memoria RAM
 - Espacio en disco: 30GB (SATA o SAS o SSD)
 - Espacio en disco para volumen de almacenamiento en los nodos virtuales: dos discos con 2TB (SATA)
 - Red: una tarjeta de red (NIC) de 1GB
- ✓ Equipo para los nodos de Computo (Compute nodes), ejecuta las instancias virtuales
 - Procesador 64-bit ó x86
 - 32GB de Memoria RAM
 - Espacio en disco: 30GB (SATA)
 - Red: dos tarjetas de red (NIC) de 1GB

La topología a implementar sería como sigue, utilizando el sistema operativo OpenStack (<http://www.openstack.org/>) para la gestión en la nube:

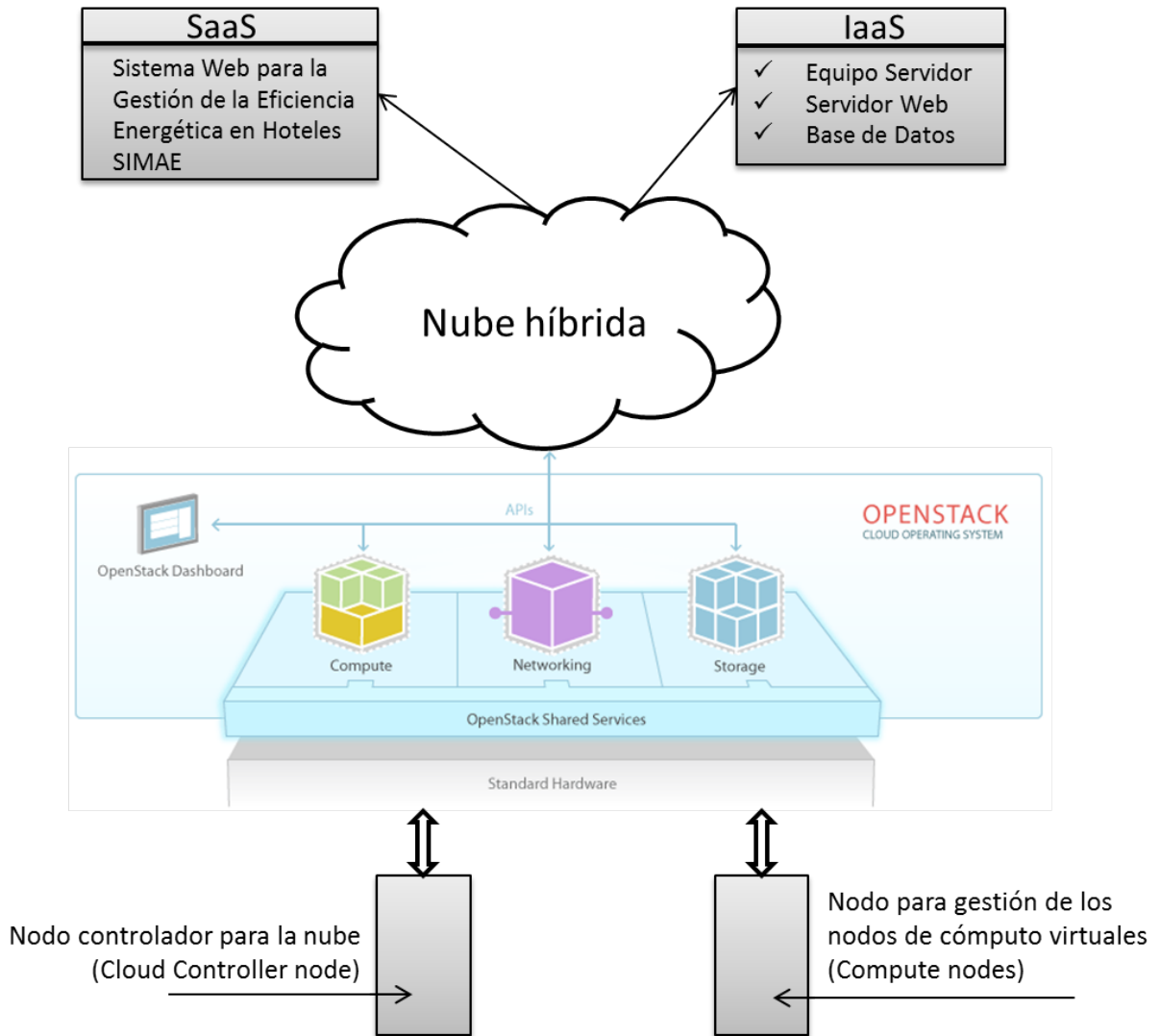


Figura 15. Topología en la nube del sistema web

Finalmente, el diagrama de flujo para el funcionamiento del sistema con el área de aire acondicionado sería como sigue:

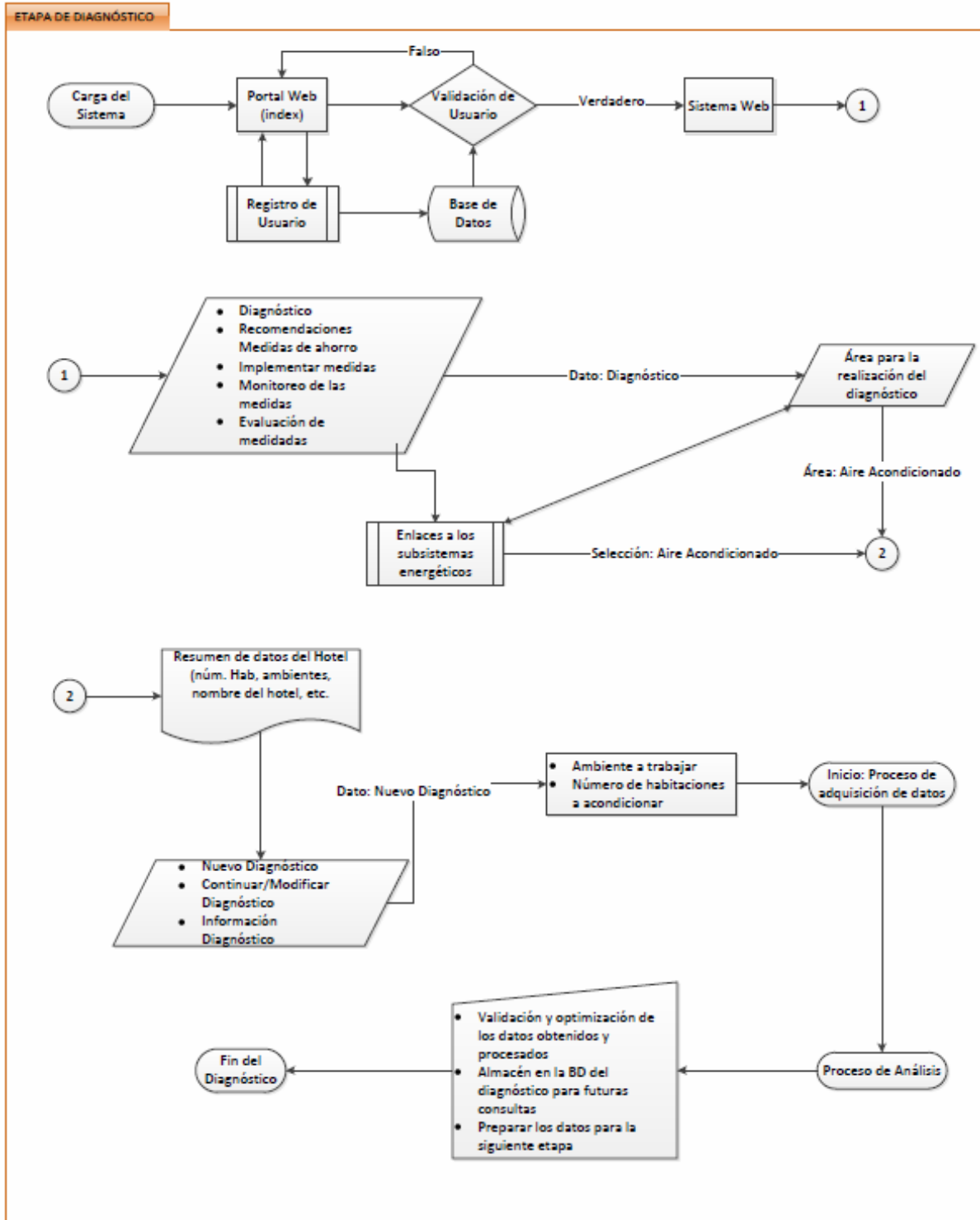


Figura 16. Diagrama de Flujo – Proceso General

El proceso más detallado para aire acondicionado es como sigue:

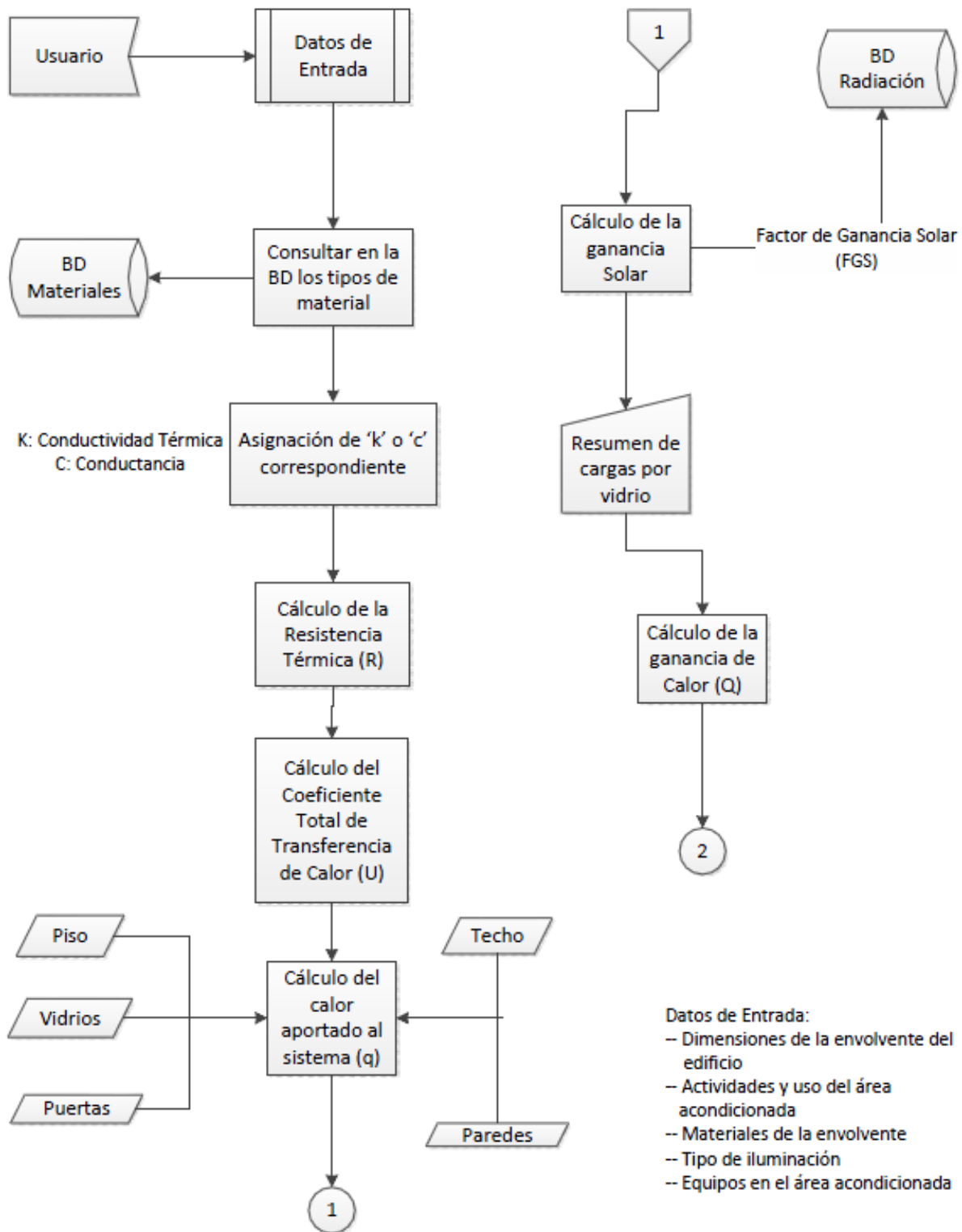


Figura 17. Diagrama de Flujo – Aire Acondicionado (parte 1)

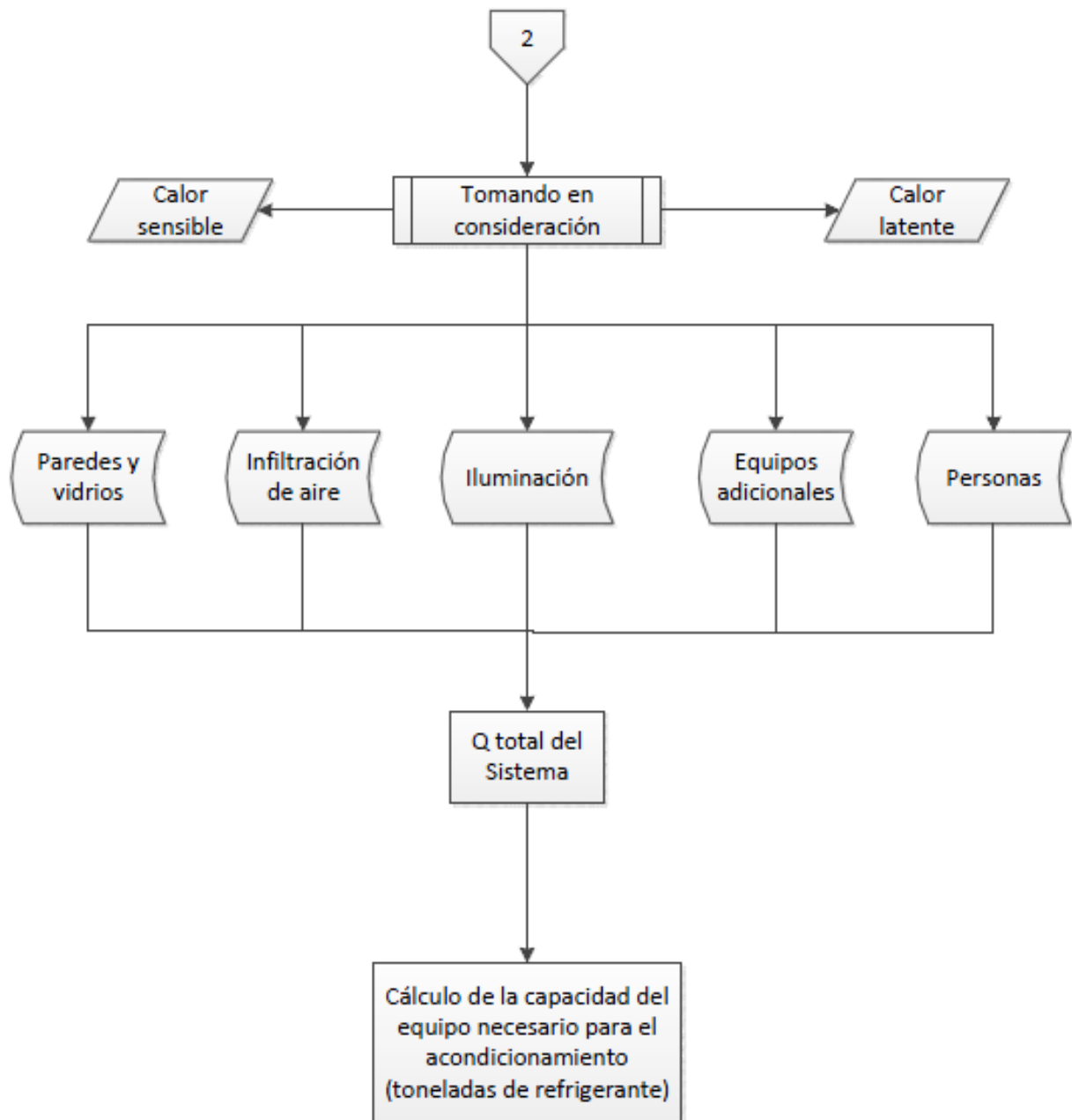
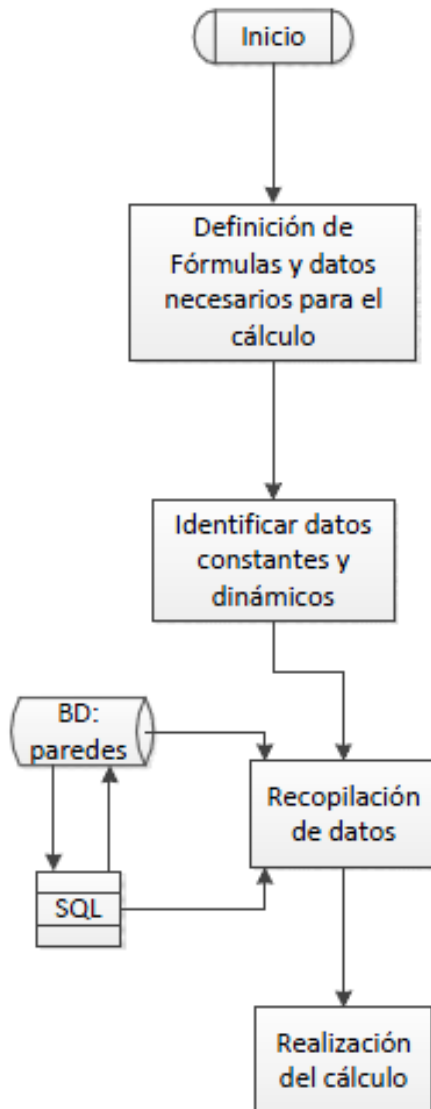


Figura 18. Diagrama de Flujo – Aire Acondicionado (parte 2)



$$R = 1 / h_e + X_1/K_1 + X_2/K_2 + \dots + 1/C_1 + 1/C_2 + \dots + 1/h_i$$

Donde:

H_i, h_e : Coef de transferencia de calor interior y exterior respectivamente

K : Conductividad térmica

C : Conductancia térmica

X : Espesor del material

1. h_i, h_e se consideran constantes (para todo tipo de material)
2. Se asocian K, C y X respectivamente acorde al tipo de material
3. Se calcula $1/h_i$ y $1/h_e$
4. Se calculan : $x_1/k_1, x_2/k_2, \dots, 1/c_1, 1/c_2, \dots$
5. Se realiza la suma de los resultados obtenidos en '3' y '4'
6. El resultado de '5' representa la 'R' del material

-- El algoritmo se aplica de igual manera para todos los materiales

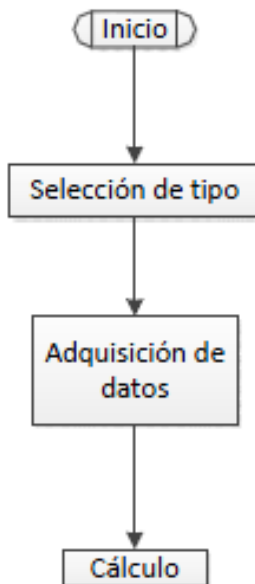
-- El proceso de asociación en el paso 2 es efectuado a nivel programación y consulta de datos (hacia la BD)

-- CÁLCULO DEL COEFICIENTE TOTAL DE TRANSFERENCIA DE CALOR (U):

1. A partir del resultado obtenido para $R \Rightarrow U = 1/R$

Figura 19. Diagrama de Flujo – Aire Acondicionado (parte 3)

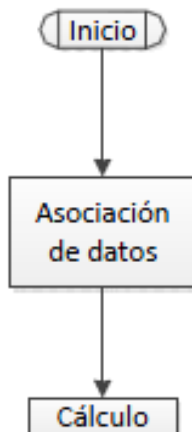
TECHOS



1. Selección de tipo
2. Asociación de datos acorde al tipo seleccionado
3. Se aplica la fórmula: $q = UADTc$ en donde:
 U = Energía interna del sistema
 A = área
 $DTc = (DT_{tabla}/1.8) + (DT_{real} - 8.3)$
4. La 'U' se obtiene de la BD correspondiente o se calcula aplicando el algoritmo correspondiente (cálculo de resistencia térmica)
5. La 'A' se obtiene de la BD correspondiente proporcionada inicialmente por el usuario
6. DT_{tabla} se obtiene de los datos proporcionados y asociados por '2'
7. DT_{real} se obtiene de la fórmula $DT_{real} = TBSE - TBSI$ en donde TBSE y TBSI se obtienen de los datos de entrada proporcionados al inicio por el usuario
8. Se calculan las operaciones en '7' y el cálculo para DTc
9. Se calcula '3' obteniendo la 'q' requerida para los techos

Figura 20. Diagrama de Flujo – Aire Acondicionado (parte 4)

PAREDES



1. Para la asociación de datos, se tiene una serie de información ordenada por hora
2. Para obtener (q) se deberán sumar los datos para cada pared en esa hora, multiplicar por su área y su energía interna (u)
 $q = UADT$
 en donde:
 DT : Diferencia de temperatura obtenida de tabla acorde la hora
3. El área (A) se obtiene de los datos de entrada iniciales, mientras que la energía interna (U) ya fue calculada previamente

Figura 21. Diagrama de Flujo – Aire Acondicionado (parte 5)

Como se observa, el cálculo está enfocado en determinar el calor que se absorbe en los espacios hoteleros para así poder determinar el equipo más adecuado para el aire acondicionado.

Analicemos el código necesario para la implementación de los diversos procesos que lleva a cabo el sistema, desde la etapa de registro/autenticación de usuarios y hoteles, hasta la obtención del diagnóstico respectivo para el aire acondicionado.

III.2 EXPOSICIÓN Y EXPLICACIÓN DEL CÓDIGO

En el siguiente apartado se explicará la parte del código más importante del sistema, ya que por la extensión del mismo, no es factible ponerlo todo, acarrearía problemas de lectura y comprensión.

Tampoco se analizará el código necesario para la creación y configuración de la base de datos, siendo un proceso trivial y sencillo de implementar teniendo el modelo relacional de la figura 14.

La puesta en funcionamiento del ambiente de trabajo, que se refiere a la configuración del servidor web y las extensiones para PHP y MySQL, son cuestiones técnicas que se encuentran fuera de la extensión de este trabajo.

Dejando en claro estos puntos, entremos de lleno con el código del sistema.

REGISTRO DE USUARIOS/HOTELES

Para registrar un hotel, primero es necesario registrar un usuario (`nuevoUsuario.php`), para que el hotel quede asociado con dicho usuario:

```
<table class="withBorder">
  <tr>
    <td>Nombre (*):</td>
    <td><input type="text" id="nombre" value="" /></td>
  </tr>
  <tr>
    <td>Usuario (*):</td>
    <td><input type="text" id="usuario" value="" /></td>
  </tr>
  <tr>
    <td>Contraseña (*):</td>
    <td><input type="password" name="pass" value="" /></td>
  </tr>
  <tr>
    <td>Vuelve a escribir tu contraseña (*):</td>
    <td><input type="password" name="pass" value="" /></td>
  </tr>
  <tr>
    <td>IFE (*):</td>
    <td><input type="text" id="ife" value="" /></td>
  </tr>
  <tr>
    <td>Correo Electrónico (e-mail):</td>
    <td><input type="text" id="email" value="" /></td>
  </tr>
  <tr>
    <td colspan="4" align="center">
      <input type="button" id="enviarRegistro" value="Registrar" />
    </td>
  </tr>
</table>
```

Con esto, se definen los campos de formulario necesarios para el registro, al momento de enviar los datos, éstos son procesados por un archivo Javascript (`registro.js`) encargado de ponerlos y verificarlos en la base de datos.

```
var peticion = null;

$(function() {
    $('#enviarRegistro').click(registrar);
});

function inicializa_xhr() {
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        return new ActiveXObject("Microsoft.XMLHTTP");
    } else {
        return;
    }
}

function registrar() {
    var nombre = document.getElementById("nombre").value;
    var usuario = document.getElementById("usuario").value;
    var password = $('[name*=pass]')[0].value;
    var password2 = $('[name*=pass]')[1].value;
    var ife = document.getElementById("ife").value;
    var email = document.getElementById("email").value;

    var eqPass = password && password2 && (password == password2);

    var obligatorios = nombre && usuario && eqPass && ife;

    var todosLosCampos = nombre && usuario && password && password2;
```

Para efectuar las instrucciones de inserción en la base de datos, es necesario hacer una petición AJAX. Se define una variable que almacena la referencia al objeto XMLHttpRequest, el cual permite realizar la petición y recibir la respuesta de la operación AJAX. Se establece que al pulsar el botón de envío de formulario se ejecuten las instrucciones de validación, las cuales, siendo validadas satisfactoriamente, sean pasadas al archivo que se encargará de introducir los datos en la base de datos (ajaxRegistro.php):

```
if (obligatorios) {
    var datos = "usuario=" + usuario +
        "&nombre=" + nombre +
        "&password=" + password +
        "&ife=" + ife +
        "&email=" + email;
    peticion = inicializa_xhr();
    peticion.open("POST", "/SistemaHotel/scripts/ajaxRegistro.php?nocache="+Math.random(), true);
    peticion.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    peticion.send(datos);

    peticion.onreadystatechange = procesaRegistro;
} else if (!todosLosCampos) {
    $('#obligatorios').text('Rellena los campos (*) faltantes.').show().fadeOut(5000);
} else {
    $('#obligatorios').text('Error: Los campos de Contraseña no son iguales.').show().fadeOut(5000);
}
}

function procesaRegistro() {
    if (peticion.readyState == 4 && peticion.status == 200) {
        var respuesta = peticion.responseText;
        if (respuesta == "true") {
            $('#obligatorios').text('Usuario Actualmente en la Base de Datos').show().fadeOut(5000);
        } else {
            $('#central').html('<p>Usuario Registrado Correctamente</p>');
            setTimeout('delayer("nuevoHotel.php)", 3000);
        }
    }
}
}
```

Si la petición fue realizada satisfactoriamente, es decir, si los datos fueron introducidos en la base de datos, y no se encuentra un usuario registrado con el mismo nombre, se procede a registrar un Hotel. El archivo encargado de procesar los datos de la petición (`ajaxRegistro.php`), es como sigue:

```

<?php
    session_start();
    require("conexionBD.php");
    require("funciones_db.php");

    //Consulta de los usuarios registrados
    $queryRegisteredUsers = "SELECT nombre FROM usuario ORDER BY nombre ASC";
    $resRegisteredUsers = mysql_query($queryRegisteredUsers, $conexion);

    // Datos del Usuario
    $nombre = $_POST["nombre"];
    $usuario = trim(strtolower($_POST["usuario"]));
    $pass = $_POST["password"];
    $ife = $_POST["ife"];
    $email = $_POST["email"];

    //Verificar que el usuario no se encuentre actualmente en la Base de Datos
    while ($row = mysql_fetch_array($resRegisteredUsers)) {
        if ($usuario == $row["nombre"]) {
            echo "true";
            exit;
        }
    }

    //Inserción de datos del Usuario
    $insertUser = "INSERT INTO usuario(nombre, password, ife) VALUES('$usuario', '$pass', '$ife)";
    mysql_query($insertUser, $conexion);

    $idQuery = "SELECT id FROM usuario ORDER BY id DESC LIMIT 1";
    $resId = mysql_query($idQuery, $conexion);
    $id = fetch_result($resId);

    if ($email) {
        mysql_query("UPDATE usuario SET email = '$email' WHERE id = ".$id[0][0], $conexion);
    }

    require("cerrarBD.php");

    $_SESSION["usuario"] = $id[0][0];
?>

```

Lo importante de este archivo es que define el ID de Usuario con el que se asociará el hotel que se vaya a registrar en el siguiente proceso, también se verifica que en la base de datos no exista un usuario ya registrado con el mismo nombre.

Para el registro del hotel se tiene lo siguiente:

```

<table class="withBorder">
  <tr>
    <th colspan="2">Datos del Hotel</th>
  </tr>
  <tr>
    <td>Nombre del Hotel(*):</td>
    <td><input type="text" id="nombreHotel" value="" /></td>
  </tr>
  <tr>
    <td>RFC Hotel(*):</td>
    <td><input type="text" id="rfc" value="" /></td>
  </tr>
  <tr>
    <td>Calle y Número(*):</td>
    <td><input type="text" id="calle" value="" /></td>
  </tr>
  <tr>
    <td>Estado(*):</td>
    <td><select id="estado">
      <option value="">--Estado--</option>
      <?php
        while ($row = mysql_fetch_array($resEstados)) {
          echo '<option value="'. $row['idestado'] . '">'. $row['nombre']. '</option>';
        }
      ?>
    </select></td>
  </tr>
  <tr>
    <td>Ciudad(*):</td>
    <td><select id="ciudad">
      <option value="">--Ciudad--</option>
    </select>
    </td>
  </tr>
  <tr>
    <td>Colonia(*):</td>
    <td><input type="text" id="colonia" value="" /></td>
  </tr>
  <tr>
    <td>Delegación o Municipio(*):</td>
    <td><input type="text" id="delegacion" value="" /></td>
  </tr>

```



```

<td>Categoría(*):</td>
<td><select id="categoria">
  <option value="">--Sel. Categoría--</option>
  <?php
    while ($row = mysql_fetch_array($resCategorias)) {
      echo '<option value="'. $row['idcategoria'] . '">'. $row['categoria']. '</option>';
    }
  ?>
</select>
</td>
</tr>
<tr>
<td>Página WEB:</td>
<td><input type="text" id="web" value="" /></td>
</tr>
<tr>
<td>Razón Social:</td>
<td><input type="text" id="rsocial" value="" /></td>
</tr>
<tr>
<td>Área del terreno (m2):</td>
<td><input type="text" id="aterreno" value="" /></td>
</tr>
<tr>
<td>Área construida (m2):</td>
<td><input type="text" id="aconstruida" value="" /></td>
</tr>
<tr>
<td>Zona de ubicación:</td>
<td><select id="ubicacion">
  <option value="">--Ubicación--</option>
  <option value="turistica">Turística</option>
  <option value="comercial">Comercial</option>
  <option value="industrial">Industrial</option>
  <option value="residencial">Residencial</option>
  <option value="natural">Natural</option>
  <option value="playa">Playa</option>
  <option value="centroHistorico">Centro Histórico</option>
</select></td>
</tr>

```

El código representa los campos para el registro del hotel, nótese que los campos de Estado, Categoría y Ciudad son obtenidos de la base de datos, siendo este último a través de una petición AJAX (nuevoRegistroHotel.js) dependiendo del Estado seleccionado, el código implementado para la petición es como sigue:

```

$(function () {
    $('#sendNewHotel').live('click', registraHotel);
    $('#estado').change(peticionCiudades);
});

function peticionCiudades() {
    if ($(this).val() == "")
        $('#ciudad').html('<option value="">--Ciudad--</ciudad>');
    else {
        $.ajax({
            url: "scripts/ajaxCargaCiudades.php",
            global: false,
            type: "POST",
            dataType: "html",
            data: ({idEstado: $(this).val()}),
            async: true,
            success: function(data) {
                $('#ciudad').html(data);
            }
        });
    }
}
}

```

Con esto se cargan las ciudades correspondientes al estado seleccionado, mostrándolas en el campo respectivo.

```

function registraHotel() {
    var nombreHotel = $('#nombreHotel').val();
    var rfcHotel = $('#rfc').val();
    var calleHotel = $('#calle').val();
    var estadoHotel = $('#estado').val();
    var ciudadHotel = $('#ciudad').val();
    var coloniaHotel = $('#colonia').val();
    var delegacionHotel = $('#delegacion').val();
    var cpHotel = $('#cp').val();
    var telefonoHotel = $('#tel').val();
    var categoriaHotel = $('#categoria').val();
    var paginaWebHotel = $('#web').val();
    var rsocialHotel = $('#rsocial').val();
    var aterrenoHotel = $('#aterreno').val();
    var aconstruidaHotel = $('#aconstruida').val();
    var ubicacionHotel = $('#ubicacion').val();
}

```

Se guardan los valores obtenidos del formulario de registro para posteriormente mandarlos junto con la petición AJAX para su inserción en la base de datos:

```
var datos = "nombreHotel=" + nombreHotel +
    "&rfcHotel=" + rfcHotel +
    "&calleHotel=" + calleHotel +
    "&estadoHotel=" + estadoHotel +
    "&ciudadHotel=" + ciudadHotel +
    "&coloniaHotel=" + coloniaHotel +
    "&delegacionHotel=" + delegacionHotel +
    "&cpHotel=" + cpHotel +
    "&telefonoHotel=" + telefonoHotel +
    "&categoriaHotel=" + categoriaHotel +
    "&paginaWebHotel=" + paginaWebHotel +
    "&rsocialHotel=" + rsocialHotel +
    "&aterrenoHotel=" + aterrenoHotel +
    "&aconstruidaHotel=" + aconstruidaHotel +
    "&ubicacionHotel=" + ubicacionHotel;

$.ajax({
    url: "scripts/ajaxProcesaNuevoHotel.php",
    global: false,
    type: "POST",
    data: datos,
    dataType: "html",
    async: true,
    success: function(data) {
        var codeRedirect = 'delayer("simae_1.php)';
        $('#warning').text('');
        setTimeout(codeRedirect, 500);
    }
});
```

Los datos son encapsulados una vez más en una sola variable; si la petición fue satisfactoria, entonces se procede a la página de carga de los módulos del sistema (simae_1.php).

Nótese que las peticiones AJAX realizadas para el registro de Hotel son efectuadas con jQuery, simplificando en gran medida el proceso, a diferencia como se hace con el registro de Usuarios. Se ponen ambos casos para observar las ventajas de utilizar jQuery.

El proceso de inserción de datos (ajaxProcesaNuevoHotel.php) para el Hotel es similar a como se hace con el registro de Usuarios.

AUTENTICACIÓN DE USUARIOS

Cuando el sistema ya tiene registrado un usuario, el proceso de autenticación está definido como se muestra a continuación:

```
<form class="medio" action="scripts/validaUsuario.php" method="post">
<fieldset>
<legend>Datos de Usuario</legend>
  <label for="usuario">Usuario: &nbsp;&nbsp;&nbsp;</label>
  <input type="text" id="usuario" name="usuario" value="" size="15" />
  <label for="pass">Contraseña: &nbsp;&nbsp;&nbsp;</label>
  <input type="password" id="pass" name="pass" value="" size="15" /> <
  <input type="submit" name="enviar" value="Entrar" /> &nbsp;&nbsp;&
</fieldset>
</form>
```

Formulario para verificar el nombre de usuario y la contraseña.

```
<?php
  session_start();
  require("conexionBD.php");
  require("funciones_db.php");

  $usuario = strtolower($_POST["usuario"]);
  $pass = $_POST["pass"];
  $valido = false;
  $hasHotel = false;

  $queryUser = mysql_query("SELECT id, nombre, password FROM usuario", $conexion);

  while ($row = mysql_fetch_array($queryUser)) {
    if ($usuario == $row["nombre"])
      if ($pass == $row["password"]) {
        $valido = true;
        $idUser = $row["id"];
        $_SESSION["usuario"] = $idUser;
        $queryHotel = "SELECT idhotel FROM hotel WHERE usuario_id = $idUser";
        $resHotel = mysql_query($queryHotel, $conexion);
        $fetchHotel = fetch_result($resHotel);
        if (count($fetchHotel) != 0)
          $hasHotel = true;
        break;
      }
  }
}
```

Se verifica que el usuario se encuentre en la base de datos y también que tenga un hotel registrado, ya que para hacer uso del sistema se necesita de un hotel.

```
if ($hasHotel) {
echo '<html>
<head><title>Datos Verificados</title>
<script type="text/javascript">
function delayer() {
window.location = "/SistemaHotel/simae_1.php";
}
</script>
</head>
<body onload="setTimeout(\'delayer()\', 3000)">
<table width=300>
<tr>
<td><strong>Nombre</strong></td>
</tr>
<tr>
<td>'. $usuario. '</td>
</tr>
</table>
<h2>Datos verificados</h2>
</body>
</html>';
} elseif ($valido) {
echo '<html>
<head><title>Hotel no Registrado</title>
<script type="text/javascript">
function delayer() {
window.location = "/SistemaHotel/nuevoHotel.php";
}
</script>
</head>
<body onload="setTimeout(\'delayer()\', 3000)">
<h2>Usted aun no tiene registrado un Hotel, Registrarlo para continuar</h2>
</body>
</html>';
} else {
echo '<html>
<head><title>Datos Incorrectos</title>
<script type="text/javascript">
function delayer() {
window.location = "/SistemaHotel/autenticacion.php";
}
</script>
</head>
```

Dependiendo del status del usuario se redirecciona al apartado correspondiente.

En caso de que las credenciales proporcionadas sean correctas, el usuario puede trabajar con los módulos del sistema (simea_1.php).

PROCESO DE ENCUESTA

Lo primero que se debe realizar antes de trabajar con las áreas energéticas del hotel será aplicar una encuesta sobre los aspectos generales del hotel, como el tipo de ambientes que existen, el número de espacios para cada uno y si cuentan o no con acondicionamiento.

Primero se cargan los posibles ambientes que puede tener un hotel de la base de datos:

```
<?php
    session_start();
    require("conexionBD.php");
    require("funciones_db.php");

    // $idHotel = $_GET["idHotel"];
    $idHotel = $_SESSION["idHotel"];
    $nombreHotel = $_SESSION["nombreHotel"];

    $queryAmbientes = "SELECT idambiente, tipo FROM ambiente";
    $resAmbientes = mysql_query($queryAmbientes, $conexion);
    $ambientes = fetch_result($resAmbientes);

    require("cerrarBD.php");
?>
```

Así como el ID del Hotel y su nombre, datos utilizados para registrar los ambientes en el hotel.

Teniendo todos los ambientes de la base de datos, éstos se presentan al usuario en una tabla en donde podrá seleccionar aquellos con los que cuenta el hotel:


```
$(function() {
  $(':checkbox.ambiente').live('click', muestraNoCuartos);
  $('#sendAmbientes').live('click', verificaAmbientes);
});

function muestraNoCuartos() {
  if ($(this).is(':checked')) {
    $('#th.encNoEspacios, th.encAcondicionamiento').show();
    $(this).closest('td').siblings('td').show();
    $('#warning').text('');
  } else {
    $(this).closest('td').siblings('td').hide();
    var numChecked = $('input:checked.ambiente').length;
    if (numChecked == 0) {
      $('#th.encNoEspacios, th.encAcondicionamiento').hide();
    }
  }
}

function verificaAmbientes() {
  var checked = $('input:checked.ambiente').length;
  if (checked == 0) {
    $('#warning').html('<p>* Selecciona al menos un ambiente</p>').show().fadeOut(5000);
    return false;
  }
}
```

Una vez que el usuario ha seleccionado al menos un tipo de ambiente para el hotel, así como el número de espacios para cada uno y su acondicionamiento, se procede a la inserción de datos por el siguiente script PHP:


```
<?php
require("conexionBD.php");

$ambientes = $_POST["ambientes"];
$idhotel = (int)$_POST["idhotel"];
$numAmbientes = count($ambientes);
$numHabTotales = 0;

mysql_query("UPDATE hotel SET numero_ambientes = $numAmbientes WHERE idhotel = $idhotel");
mysql_query("DELETE FROM hotel_has_ambiente WHERE hotel_idhotel = $idhotel");
mysql_query("UPDATE hotel SET numero_habitaciones = 0 WHERE idhotel = $idhotel");

foreach ($ambientes as $v) {
    $numHabAmb = (int)$_POST[$v];
    $idAmbiente = (int)$v;
    $hasAcond = (int)$_POST[$v.'_'].$v;
    mysql_query("INSERT INTO hotel_has_ambiente(hotel_idhotel, ambiente_idambiente, numHabAmbiente, acondicionamiento)".
        " VALUES($idhotel, $idAmbiente, $numHabAmb, $hasAcond)");
    $numHabTotales += $numHabAmb;
}

mysql_query("UPDATE hotel SET numero_habitaciones = $numHabTotales WHERE idhotel = $idhotel");

require("cerrarBD.php");
?>
```

Además de insertar los ambientes con los que cuenta el hotel, se actualiza la información con el número de ambientes con los que cuenta y el número de espacios totales.

Teniendo los ambientes a acondicionar registrados, se puede trabajar finalmente con el área de aire acondicionado.

El punto de partida, será el módulo de diagnóstico energético, que recopilará la mayoría de la información para cada área energética, con la que se realizará el análisis y generación de recomendaciones, evaluación e implementación de medidas de ahorro.

ADQUISICIÓN DE DATOS Y DIAGNÓSTICO – AIRE ACONDICIONADO

Para la obtención del diagnóstico en Aire Acondicionado, se requiere de un análisis de la envolvente del edificio así como de los espacios interiores.

Para espacios exteriores se analizan puertas, paredes y ventanas.

Para cada tipo de superficie se especifica el número que existe en el hotel (por ejemplo, se tienen 4 paredes exteriores y 3 puertas exteriores).

La información del formulario de datos es generada dinámicamente acorde al número de superficies especificadas.

Veamos el ejemplo de cómo se carga el formulario y los materiales de construcción requeridos para las paredes exteriores:

```
<?php
require("conexionBD.php");
require("funciones_db.php");

$query_acabado_ex = "SELECT idmaterial, material FROM material WHERE idtipo_material = 3 AND LOWER(material) NOT LIKE '%interior%'";
$query_aislamiento = "SELECT idmaterial, material FROM material WHERE idtipo_material = 2";
$query_albanileria = "SELECT idmaterial, material FROM material WHERE idtipo_material = 1";
$query_acabado_in = "SELECT idmaterial, material FROM material WHERE idtipo_material = 3 AND LOWER(material) NOT LIKE '%exterior%'";
$query_orientacion = "SELECT idorientacion, nombre FROM orientacion WHERE LOWER(nombre) NOT LIKE '%horizontal%'";

$result_acabado_ex = mysql_query($query_acabado_ex, $conexion);
$result_aislamiento = mysql_query($query_aislamiento, $conexion);
$result_albanileria = mysql_query($query_albanileria, $conexion);
$result_acabado_in = mysql_query($query_acabado_in, $conexion);
$result_orientacion = mysql_query($query_orientacion, $conexion);

$material_acabado_ex = fetch_result($result_acabado_ex);
$material_aislamiento = fetch_result($result_aislamiento);
$material_albanileria = fetch_result($result_albanileria);
$material_acabado_in = fetch_result($result_acabado_in);
$orientaciones = fetch_result($result_orientacion);
```

Los materiales son cargados de la base de datos, se guardan en variables como arreglos para ser insertados en el formulario para el usuario.

Dependiendo el número de paredes que se seleccionen, serán los campos de datos correspondientes, representados por una respuesta a una llamada AJAX:

```

$numParedesExt = (int)$_POST["numParedes"];

for ($i=0; $i < $numParedesExt; $i++) {
    echo '<fieldset style="display: inline;">
        <legend style="color: #CC0000;">Pared Exterior '.$(i+1).'

```

```

echo      '</select><br />
          Espesor (cm):
          <input type="text" name="espesorAisParedE[]" value="" size="5" />
        </fieldset><br />
        <fieldset style="display: inline;">
          <legend style="color: #990000;">Material de Albañilería</legend>
          Material:
          <select name="materialAlbParedE[]">' ;
          foreach ($material_albanileria as $val) {
            echo '<option value="'. $val[0]. '">'. $val[1]. '</option>';
          }
echo      '</select><br />
          Espesor (cm):
          <input type="text" name="espesorAlbParedE[]" value="" size="5" />
        </fieldset>
        <fieldset style="display: inline;">
          <legend style="color: #990000;">Acabado Interior</legend>
          Material:
          <select name="materialAiParedE[]">' ;
          foreach ($material_acabado_in as $val) {
            echo '<option value="'. $val[0]. '">'. $val[1]. '</option>';
          }
echo      '</select><br />
          Espesor (cm):
          <input type="text" name="espesorAiParedE[]" value="" size="5" />
        </fieldset>
      </fieldset><br /><br />';
echo '<script type="text/javascript">
      $(\'#color'. $i. '\').modcoder_excolor();
    </script>';

```

Como se observa, dependiendo el número de paredes que se tengan, será el número de campos generado. Se necesitan obtener datos para el acabado exterior e interior, aislamiento y el material de albañilería de la pared, al usuario se le presenta una lista de materiales predefinida que fueron cargados previamente.

Una vez obtenidos los datos, éstos son procesados y cargados en la base de datos.

El control de eventos y las peticiones, son controladas con Javascript/jQuery:

```
$(function() {
    $('#numParedesExt').change(peticionParedesExt);
    $('#numPuertasExt').change(peticionPuertasExt);
    $('#numVentanasExt').change(peticionVentanasExt);
    $('#sendSuperficiesExt').hide();
});

function peticionParedesExt() {
    if ($(this).val() != "") {
        $.ajax({
            url: "scripts/ajaxParedesExteriores.php",
            global: false,
            type: "POST",
            data: ({numParedes: this.value}),
            dataType: "html",
            async: true,
            success: function(data) {
                $('#infoAdicionalParedesExt').html(data);
            }
        });
        $('#sendSuperficiesExt').show();
    } else {
        $('#infoAdicionalParedesExt').text('');
        if (($('#numPuertasExt').val() == "") && ($('#numVentanasExt').val() == "")) {
            $('#sendSuperficiesExt').hide();
        }
    }
}
```

El código solo muestra la petición para paredes exteriores, ya que las otras (para puertas y ventanas) son similares.

Después que los datos de entrada se insertan en la base de datos (al enviar el formulario), se procede a la realización del cálculo para la obtención del diagnóstico.

```
//Datos constantes: Coeficientes de transferencia de calor interior y exterior respectivamente
$hint = 8.03;
$hext = 29.3;
$modHint = 1.0 / $hint;
$modHext = 1.0 / $hext;

//Datos de las Puertas Exteriores
@$orientacionPuertaE = $_POST["orientacionPuertaE"];
@$materialPuertaE = $_POST["materialPuertaE"];
@$espesorPuertaE = $_POST["espesorPuertaE"];
@$areaPuertaE = $_POST["areaPuertaE"];
@$numPuertas = count($orientacionPuertaE);

//Datos de las Ventanas Exteriores
@$orientacionVe = $_POST["orientacionVe"];
@$materialVe = $_POST["materialVe"];
@$areaVe = $_POST["areaVe"];
$espesorVentanaE = 0;
@$numVentanas = count($orientacionVe);

//--- Datos de las Paredes Exteriores ---
@$tipo_colorParedE = $_POST["tipo_colorParedE"];
@$orientacionParedE = $_POST["orientacionParedE"];
@$areaParedE = $_POST["areaParedE"];
@$colorParedE = $_POST["colorParedE"];
@$numParedes = count($tipo_colorParedE);
//- Acabado Exterior -
@$materialAeParedE = $_POST["materialAeParedE"];
@$espesorAeParedE = $_POST["espesorAeParedE"];
//- Aislamiento -
@$materialAisParedE = $_POST["materialAisParedE"];
@$espesorAisParedE = $_POST["espesorAisParedE"];
//- Material de Albañilería -
@$materialAlbParedE = $_POST["materialAlbParedE"];
@$espesorAlbParedE = $_POST["espesorAlbParedE"];
//- Acabado Interior -
@$materialAiParedE = $_POST["materialAiParedE"];
@$espesorAiParedE = $_POST["espesorAiParedE"];
```

Datos enviados por el usuario, almacenados en variables locales para su procesamiento.

```
//Query para obtener la temperatura de diseño maximo acorde a la ciudad
//en donde se encuentre el hotel
$queryTempDisMax = "SELECT diseno_max FROM ciudad c JOIN hotel h ON(c.idciudad = h.idciudad)
                    WHERE h.idhotel = $idhotel";
$resTempDisMax = mysql_query($queryTempDisMax, $conexion);
$fetchedTempDisMax = fetch_result($resTempDisMax);
$tempDisMax = $fetchedTempDisMax[0][0];

//Diferencia de temperatura para parades exteriores:
//temp_diseño_max - temp_a acondicionar
//Por el momento se asume que se desea acondicionar a una temperatura de 25°C,
//sujeto a cambio o establecido por el usuario
$tempAcond = 25;
$dtreal = $tempDisMax - $tempAcond;

// Operación de truncamiento acorde a la documentación de excel
if ((11 - $dtreal) < 0) {
    $temp = abs(11 - $dtreal);
    $temp = floor($temp);
    $dttrunc = $temp * (-1);
} else {
    $dttrunc = floor(11 - $dtreal);
}
}
```

Se obtiene y calcula información adicional, como la temperatura de diseño acorde al estado y ciudad en donde se encuentre el hotel, la temperatura a la que se desea acondicionar y cálculos propios del algoritmo.

Para puertas y ventanas exteriores, los cálculos son efectuados por el manejador de base de datos a través de un procedimiento almacenado, al cual se le pasan como parámetros los datos de entrada.

```

// Inserción de datos y realización de cálculos para las Puertas Exteriores --tipo_superficie: 2--
// Temperatura a acondicionar: 25°C (sujeto a cambio o establecido por el usuario)
for ($i = 0; $i < $numPuertas; $i++) {
    if (!$areaPuertaE[$i])
        continue;
    if (!$espesorPuertaE[$i])
        $espesorPuertaE[$i] = 0;
    mysql_query("CALL puerta_ventana_ext_calcs(".(int)$orientacionPuertaE[$i].", ".(float)$espesorPuertaE[$i].", ".
        (float)$areaPuertaE[$i].", $idhotel, $puerta_exterior, ".(int)$materialPuertaE[$i].", ".(float)$tempAcond.");
}

// Inserción de datos y realización de cálculos para las Ventanas Exteriores --tipo_superficie: 4--
// Temperatura a acondicionar: 25°C (sujeto a cambio o establecido por el usuario)
for ($i = 0; $i < $numVentanas; $i++) {
    if (!$areaVe[$i])
        continue;
    mysql_query("CALL puerta_ventana_ext_calcs(".(int)$orientacionVe[$i].", $espesorVentanaE, ".
        (float)$areaVe[$i].", $idhotel, $ventana_exterior, ".(int)$materialVe[$i].", ".(float)$tempAcond.");
}

```

El bucle (for) se utiliza para procesar cada una de las puertas y ventanas exteriores. El código del procedimiento almacenado se encarga de insertar y realizar los cálculos para el diagnóstico:

```

BEGIN
    DECLARE last_puerta_ventana_ext BIGINT;
    DECLARE tipo_superficie_sp VARCHAR(40);
    DECLARE tipo_material_sp VARCHAR(15);
    DECLARE idciudad_sp INT;
    DECLARE diseno_max_sp FLOAT;
    DECLARE cond FLOAT;
    DECLARE r_sp FLOAT DEFAULT 0;
    DECLARE u_sp FLOAT;
    DECLARE q_sp FLOAT;

    DECLARE factor_forma FLOAT;
    DECLARE q_insolacion FLOAT;
    DECLARE mes_a_procesar INT;
    DECLARE hora_a_procesar INT;
    DECLARE fgs_sp INT;
    DECLARE idvalor_insolacion_sp INT;

    DECLARE dt FLOAT;
    DECLARE hint FLOAT;
    DECLARE hext FLOAT;

    SET @hint = 8.03;
    SET @hext = 29.3;

    SET @factor_forma = 0.7;
    SET @mes_a_procesar = 1;

```


Se declaran las variables necesarias y se procede a la inserción y cálculo:

```

INSERT INTO puerta_ventana_exterior(idorientacion, espesor, area, idhotel, idtipo_superficie, idmaterial)
VALUES (orientacion_sp, espesor_sp, area_sp, idhotel_sp, idtipo_superficie_sp, idmaterial_sp);

SET @idciudad_sp = (SELECT idciudad FROM hotel WHERE idhotel = idhotel_sp);
SET @diseno_max_sp = (SELECT diseno_max FROM ciudad WHERE idciudad = @idciudad_sp);

SET @cond = (SELECT k FROM material WHERE idmaterial = idmaterial_sp);

IF IFNULL(@cond, 0) = 0 THEN
    SET @cond = (SELECT c FROM material WHERE idmaterial = idmaterial_sp);
    SET @r_sp = (1.0 / @hext) + (1.0 / @cond) + (1.0 / @hint);
ELSE
    SET @r_sp = (1.0 / @hext) + (espesor_sp / @cond) + (1.0 / @hint);
END IF;

SET @u_sp = 1.0 / @r_sp;

SET @tipo_superficie_sp = (SELECT nombre FROM tipo_superficie WHERE idtipo_superficie = idtipo_superficie_sp);

IF (LOWER(@tipo_superficie_sp) = 'puerta exterior') THEN
    SET @dt = @diseno_max_sp - temp_a_acond;
ELSEIF (LOWER(@tipo_superficie_sp) = 'ventana exterior') THEN
    SET @dt = @diseno_max_sp - temp_a_acond - 5.0;
END IF;

SET @q_sp = @u_sp * area_sp * @dt;

SET @last_puerta_ventana_ext = (SELECT idpuerta_ventana_exterior FROM puerta_ventana_exterior
                                ORDER BY idpuerta_ventana_exterior DESC LIMIT 1);

INSERT INTO diagnostico_puerta_ventana_ext(r, u, q, idpuerta_ventana_ext)
VALUES (@r_sp, @u_sp, @q_sp, @last_puerta_ventana_ext);

```

Se inserta la puerta o ventana según sea el caso, y se calculan las variables de acuerdo al material de construcción empleado, finalmente insertando los datos calculados en la tabla de diagnóstico.

En ventanas y puertas de vidrio, se necesita calcular la insolación solar:

```

SET @tipo_material_sp = (SELECT tm.tipo FROM tipo_material tm JOIN material m ON(tm.idtipo_material = m.idtipo_material)
                        JOIN puerta_ventana_exterior pve ON(m.idmaterial = pve.idmaterial)
                        WHERE pve.idpuerta_ventana_exterior = @last_puerta_ventana_ext);

IF (LOWER(@tipo_material_sp) = 'vidrio') THEN

    WHILE @mes_a_procesar < 8 DO

        SET @hora_a_procesar = 6;

        WHILE @hora_a_procesar < 19 DO

            SELECT idvalor_insolacion, factor_ganancia_solar INTO @idvalor_insolacion_sp, @fgs_sp FROM valor_insolacion
            WHERE idmes = @mes_a_procesar AND idorientacion = orientacion_sp
            AND idhora = (SELECT idhora FROM hora WHERE hora = @hora_a_procesar);

            SET @q_insolacion = area_sp * @factor_forma * @fgs_sp;

            INSERT INTO calculo_insolacion(ganancia_solar, idpuerta_ventana_exterior, idvalor_insolacion)
            VALUES (@q_insolacion, @last_puerta_ventana_ext, @idvalor_insolacion_sp);

            SET @hora_a_procesar = @hora_a_procesar + 1;

        END WHILE;

        SET @mes_a_procesar = @mes_a_procesar + 1;

    END WHILE;
END IF;

```

El cálculo de insolación se hace por hora y mes, desde las 6:00 hasta las 18:00 hrs. Para los últimos 7 meses del año (Junio a Diciembre). De este proceso de cálculo se obtiene la ganancia solar, información almacenada en la tabla de cálculo correspondiente.

Con este procedimiento almacenado se tienen procesadas las puertas y ventanas exteriores. Para las paredes exteriores se regresa nuevamente al cálculo con PHP.

Se comienza por verificar que se tengan datos obtenidos por el usuario, para después insertarlos en la base de datos:

```

//Insercion de datos y realizacion de calculos para las Paredes Exteriores --tipo_superficie: 6--
for ($i = 0; $i < $numParedes; $i++) {
    if (!$areaParedE[$i])
        continue;

    if (!$espesorAeParedE[$i])
        $espesorAeParedE[$i] = 0;
    if (!$espesorAisParedE[$i])
        $espesorAisParedE[$i] = 0;
    if (!$espesorAlbParedE[$i])
        $espesorAlbParedE[$i] = 0;
    if (!$espesorAiParedE[$i])
        $espesorAiParedE[$i] = 0;

    $queryInsertParedExt = "INSERT INTO superficie_exterior(tipo_color, area, color, idhotel, idtipo_superficie, idorientacion)
        VALUES('.$tipo_colorParedE[$i].', ".(float)$areaParedE[$i].", ".hexdec($colorParedE[$i]).", $idhotel,
            $pared_exterior, ".(int)$orientacionParedE[$i].")";
    mysql_query($queryInsertParedExt, $conexion);

    $queryLastSuperficieExt = "SELECT idsuperficie_exterior FROM superficie_exterior ORDER BY idsuperficie_exterior
        DESC LIMIT 1";
    $resLastSuperficieExt = mysql_query($queryLastSuperficieExt, $conexion);
    $fetchLastSuperficieExt = fetch_result($resLastSuperficieExt);
    $lastSuperficieExt = $fetchLastSuperficieExt[0][0];

    $queryInsertAe = "INSERT INTO corte_exterior(espesor, idmaterial, idtipo_corte, idsuperficie_exterior)
        VALUES (".(float)$espesorAeParedE[$i].", ".(int)$materialAeParedE[$i].", $acabado_exterior, $lastSuperficieExt)";
    mysql_query($queryInsertAe, $conexion);

    $queryInsertAis = "INSERT INTO corte_exterior(espesor, idmaterial, idtipo_corte, idsuperficie_exterior)
        VALUES (".(float)$espesorAisParedE[$i].", ".(int)$materialAisParedE[$i].", $aislamiento, $lastSuperficieExt)";
    mysql_query($queryInsertAis, $conexion);

    $queryInsertAlb = "INSERT INTO corte_exterior(espesor, idmaterial, idtipo_corte, idsuperficie_exterior)
        VALUES (".(float)$espesorAlbParedE[$i].", ".(int)$materialAlbParedE[$i].", $albanileria, $lastSuperficieExt)";
    mysql_query($queryInsertAlb, $conexion);

    $queryInsertAi = "INSERT INTO corte_exterior(espesor, idmaterial, idtipo_corte, idsuperficie_exterior)
        VALUES (".(float)$espesorAiParedE[$i].", ".(int)$materialAiParedE[$i].", $acabado_interior, $lastSuperficieExt)";
    mysql_query($queryInsertAi, $conexion);
}

```

Los materiales de construcción para cada uno de los cortes de la pared (albañilería, acabado interior y exterior, aislamiento) son puestos en una misma tabla: CORTE_EXTERIOR e identificados por el ‘tipo de corte’.

```

$queryKcAe = "SELECT k, c FROM material WHERE idmaterial = ".$materialAeParedE[$i];
$queryKcAis = "SELECT k, c FROM material WHERE idmaterial = ".$materialAisParedE[$i];
$queryKcAlb = "SELECT k, c FROM material WHERE idmaterial = ".$materialAlbParedE[$i];
$queryKcAi = "SELECT k, c FROM material WHERE idmaterial = ".$materialAiParedE[$i];

$resKcAe = mysql_query($queryKcAe, $conexion);
$resKcAis = mysql_query($queryKcAis, $conexion);
$resKcAlb = mysql_query($queryKcAlb, $conexion);
$resKcAi = mysql_query($queryKcAi, $conexion);

$fetcheKcAe = fetch_result($resKcAe);
$fetcheKcAis = fetch_result($resKcAis);
$fetcheKcAlb = fetch_result($resKcAlb);
$fetcheKcAi = fetch_result($resKcAi);

if ($fetcheKcAe[0][0] != "") {
    $kcAe = $fetcheKcAe[0][0];
    $rAe = $espesorAeParedE[$i] / $kcAe;
} else {
    $kcAe = $fetcheKcAe[0][1];
    $rAe = 1.0 / $kcAe;
} if ($fetcheKcAis[0][0] != "") {
    $kcAis = $fetcheKcAis[0][0];
    $rAis = $espesorAisParedE[$i] / $kcAis;
} else {
    $kcAis = $fetcheKcAis[0][1];
    $rAis = 1.0 / $kcAis;
} if ($fetcheKcAlb[0][0] != "") {
    $kcAlb = $fetcheKcAlb[0][0];
    $rAlb = $espesorAlbParedE[$i] / $kcAlb;
} else {
    $kcAlb = $fetcheKcAlb[0][1];
    $rAlb = 1.0 / $kcAlb;
} if ($fetcheKcAi[0][0] != "") {
    $kcAi = $fetcheKcAi[0][0];
    $rAi = $espesorAiParedE[$i] / $kcAi;
} else {
    $kcAi = $fetcheKcAi[0][1];
    $rAi = 1.0 / $kcAi;
}

//Calculo de 'R: resistencia de transferencia de calor' y 'U: coeficiente total de transferencia de calor'
$r = $modHext + $rAe + $rAis + $rAlb + $rAi + $modHint;
$u = 1.0 / $r;

```

Se establecen los coeficientes de conductividad de acuerdo al material, para posteriormente calcular R (resistencia de transferencia de calor) y U (coeficiente total de transferencia de calor).

```

$queryInsertRu = "INSERT INTO r_u_ext(r, u) VALUES ($r, $u)";
mysql_query($queryInsertRu, $conexion);

$queryLastRu = "SELECT idr_u_ext FROM r_u_ext ORDER BY idr_u_ext DESC LIMIT 1";
$resLastRu = mysql_query($queryLastRu, $conexion);
$fetchedLastRu = fetch_result($resLastRu);
$lastRu = $fetchedLastRu[0][0];

if (strtolower($tipo_colorParedE[$i]) == "medio")
    $tipo_colorParedE[$i] = "claro";

$queryDts = "SELECT dt.temperatura FROM diferencial_temperatura dt JOIN orientacion o
            ON(dt.idorientacion = o.idorientacion) JOIN hora h ON(dt.idhora = h.idhora)
            WHERE dt.idorientacion = ".$orientacionParedE[$i]." AND
            LOWER(dt.intensidad_color) = '".strtolower($tipo_colorParedE[$i])."' ORDER BY h.hora";
$resQueryDts = mysql_query($queryDts, $conexion);
$dts = fetch_result($resQueryDts);

$contadorHora = 1;
foreach ($dts as $dttabla) {
    $dte = ($dttabla[0] / 1.8) + ($dttreal - 8.3) + ($dtttrunc / 2.0);
    $q = $areaParedE[$i] * $u * $dte;

    $queryInsertDiagnostico = "INSERT INTO diagnostico_superficie_ext(q, idhora, idsuperficie_exterior, idr_u_ext)
                              VALUES ($q, $contadorHora, $lastSuperficieExt, $lastRu)";
    mysql_query($queryInsertDiagnostico, $conexion);
    $contadorHora++;
}

```

R y U son insertados en la base de datos, para Q (calor aportado) se obtienen los datos por hora: desde las 8:00 hasta las 00:00 hrs. El diferencial de temperatura también es procesado acorde a la hora, la intensidad de color y la orientación de la pared.

Con esto se tienen las paredes exteriores procesadas, la diferencia con respecto a puertas y ventanas radica en la forma que se procesan los cálculos según el algoritmo.

Para procesar las superficies interiores, primero se obtiene la lista de aquellos ambientes que tienen o se desean acondicionar, para posteriormente procesar los espacios de cada uno.

```

<?php
    session_start();
    require("conexionBD.php");
    require("funciones_db.php");

    $nombreHotel = $_SESSION["nombreHotel"];
    $idHotel = $_SESSION["idHotel"];

    $queryAmbientesAcond = "SELECT a.tipo, hha.numHabAmbiente, hha.ambiente_idambiente FROM hotel_has_ambiente hha
        JOIN ambiente a ON(hha.ambiente_idambiente = a.idambiente)
        WHERE hha.hotel_idhotel = $idHotel AND hha.acondicionamiento = 1";
    $resAmbientesAcond = mysql_query($queryAmbientesAcond, $conexion);
    $fetchAmbientesAcond = fetch_result($resAmbientesAcond);

    mysql_free_result($resAmbientesAcond);

    require("cerrarBD.php");

    $ambientes = serialize($fetchAmbientesAcond);
?>

```

El resultado obtenido se encapsula en un arreglo para poder mandarlo al momento de enviar el formulario, con esto se podrá llevar un control de los ambientes y espacios procesados.

```

$queryEquipo = "SELECT idcEquipo, nombre FROM cEquipo";
$resEquipo = mysql_query($queryEquipo, $conexion);
$equipo = fetch_result($resEquipo);

// Si existen ambientes que tienen aire acondicionado o se pretenden acondicionar se guardan en una variable local
if (@$_POST["init_info"]) {
    $ambientes = unserialize($_POST["ambientes"]);

    // Obtención del nombre del primer ambiente y el número de espacios para ese ambiente
    $numAmbientes = count($ambientes);
    $nombreAmb = $ambientes[0][0]; // Nombre del ambiente almacenado en la primer columna
    $numEspacioAmb = $ambientes[0][1]; // Número de espacios para el ambiente en la segunda columna
    $idAmbiente = $ambientes[0][2]; // Id del ambiente en la tercer columna
}

```

Se carga el catálogo de equipo eléctrico, para registrarlo en caso de que el espacio correspondiente cuente con ello; también se obtiene la información del primer ambiente a procesar, así como el número de espacios para el mismo.

La forma de presentar el formulario de datos para las puertas, ventanas y paredes es la misma a como se hizo con las superficies exteriores, sin embargo, en superficies interiores se puede registrar equipo eléctrico a través de un formulario emergente gestionado a través de un widget de jQuery:

```
$(function() {
    var $dialog = $('#dialogEquipo').dialog({
        autoOpen: false,
        modal: true,
        width: 380,
        position: [200, 10],
        dialogClass: "dialogStyle"
    });

    $dialog.dialog("option", "buttons", [
        {
            text: "Ok",
            click: function() {
                var equipoChecked = $('#input:checked.equipo').length;
                if (equipoChecked == 0) {
                    $('#warningEquipo').html('<p>* Selecciona al menos un equipo</p>').show().fadeOut(5000);
                } else {
                    var url = $('#formEquipo').attr('action');
                    var datos = $('#formEquipo').serialize();
                    $.post(url, datos, function(respuesta){
                        $dialog.dialog("close");
                    });
                }
            }
        },
        {
            text: "Cerrar",
            click: function() {
                $(this).dialog("close");
            }
        }
    ]
    );

    $('#a#openDiagEquipo').click(function() {
        $dialog.dialog("open");
        return false;
    });
});
```

El formulario es incrustado en el documento XHTML para el registro de las superficies interiores, como si se tratara de cualquier otro, pero que gracias al código jQuery tiene un comportamiento totalmente diferente:

```
<form method="post" action="scripts/procesa-equipo.php" name="formEquipo" id="formEquipo">
  <table>
    <tr>
      <th>Equipo</th>
      <th class="cantEquipo" style="display: none;">Cantidad</th>
      <th class="potEquipo" style="display: none; padding-left: 20px;">Potencia (W)</th>
    </tr>
    <?php
      for ($i = 0, $size = count($equipo); $i < $size; $i++) {
        echo '<tr>';
        echo '<td>';
        <input class="equipo" type="checkbox" name="equipos[]"
          value="'. $equipo[$i][0].'" />' . $equipo[$i][1]. '</td>'
        <td style="display: none;">
          <select style="margin-left: 30px;" name="'. $equipo[$i][0].'">
            </select>
          </td>
          <td style="display: none;">
            <input type="text" style="margin-left: 30px;" size="6" value="100"
              name="potencia_'. $equipo[$i][0].'" />
            </td>';
        echo '</tr>';
      }
    <?>
  </table>
  <input type="hidden" name="idHotel" value="<?php echo $idHotel; ?>" />
  <input type="hidden" name="idAmbiente" value="<?php echo $idAmbiente; ?>" />
  <input type="hidden" name="numEspacioAmb" value="<?php echo $numEspacioAmb; ?>" />
</form>
```

Como se observa, el código representa un formulario común, inclusive con código embebido en PHP, pero su comportamiento definido en su totalidad por el widget jQuery.

El archivo para procesar los datos de equipo eléctrico se muestra a continuación:

```
$factorDeConversion = 0.86; // Factor de Conversión de potencia para convertir de W a Kcal/h

$queryEspacio = "SELECT idespacio_hotel FROM espacio_hotel WHERE numero_espacio = $noEspacio AND idhotel = $idHotel
                AND idambiente = $idAmbiente";
$resEspacio = mysql_query($queryEspacio, $conexion);
$lastEspacio;
while ($row = mysql_fetch_array($resEspacio)) {
    $lastEspacio = $row[0];
}

if (!$lastEspacio) {
    $queryInsertEspacio = "INSERT INTO espacio_hotel(numero_espacio, idhotel, idambiente)
                        VALUES($noEspacio, $idHotel, $idAmbiente)";
    mysql_query($queryInsertEspacio, $conexion);
    $queryLastEspacio = "SELECT idespacio_hotel FROM espacio_hotel
                        ORDER BY idespacio_hotel DESC LIMIT 1";
    $resLastEspacio = mysql_query($queryLastEspacio, $conexion);
    while ($row = mysql_fetch_array($resLastEspacio)) {
        $lastEspacio = $row[0];
    }
}

foreach ($equipos as $v) {
    $idEquipo = (int)$v;
    $cantidadEquipos = (int)$_POST[$v];
    $potenciaEquipos = (int)$_POST["potencia_". $v];
    if (!$potenciaEquipos)
        $potenciaEquipos = 0;
    $qsEquipo = $cantidadEquipos * $potenciaEquipos * $factorDeConversion;
    $queryInsertCalculo = "INSERT INTO calculo_equipo(potencia, cantidad, qs, idcEquipo, idespacio_hotel)
                        VALUES($potenciaEquipos, $cantidadEquipos, $qsEquipo, $idEquipo, $lastEspacio)";
    mysql_query($queryInsertCalculo, $conexion);
}
```

Se carga el factor de conversión requerido para el cálculo, y en caso de que sea primera vez que se registre equipo eléctrico para el espacio, se realizan los cálculos correspondientes, en caso contrario, se borran los datos que se tengan y se toman los nuevos valores:

```

} else {
    $queryRemoveEquipo = "DELETE FROM calculo_equipo WHERE idespacio_hotel = $lastEspacio";
    mysql_query($queryRemoveEquipo, $conexion);

    foreach ($equipos as $v) {
        $idEquipo = (int)$v;
        $cantidadEquipos = (int)$_POST[$v];
        $potenciaEquipos = (int)$_POST["potencia_".$v];
        if (!$potenciaEquipos)
            $potenciaEquipos = 0;
        $qsEquipo = $cantidadEquipos * $potenciaEquipos * $factorDeConversion;
        $queryInsertCalculo = "INSERT INTO calculo_equipo(potencia, cantidad, qs, idcEquipo, idespacio_hotel)
            VALUES($potenciaEquipos, $cantidadEquipos, $qsEquipo, $idEquipo, $lastEspacio)";
        mysql_query($queryInsertCalculo, $conexion);
    }
}

```

El bucle es utilizado para procesar el equipo eléctrico seleccionado, junto con su información adicional como la cantidad y la potencia.

Para procesar las paredes, puertas y ventanas interiores, se utiliza, al igual que para exteriores, un procedimiento almacenado:

```

// Procesa las puertas, paredes y ventanas para el espacio actual
// - Paredes Interiores -
for ($i = 0; $i < $numParedes; $i++) {
    if (!$areaPared[$i])
        continue;
    $procedurePared = "CALL pared_int_calcs(".$orientacionPared[$i].", ".$areaPared[$i].", ".$materialAePared[$i].", ".
        $espesorAePared[$i].", ".$materialAisPared[$i].", ".$espesorAisPared[$i].", ".
        $materialAlbPared[$i].", ".$espesorAlbPared[$i].", ".$materialAiPared[$i].", ".$espesorAiPared[$i].", ".
        $pared_interior.", ".$lastEspacioHotel.", ".$idHotel.", $temp_a_acond.");";
    mysql_query($procedurePared, $conexion);
}

// - Puertas Interiores -
for ($i = 0; $i < $numPuertas; $i++) {
    if (!$areaPuerta[$i])
        continue;
    $procedurePuerta = "CALL puerta_ventana_int_calcs(".$orientacionPuerta[$i].", ".$espesorPuerta[$i].", ".
        $areaPuerta[$i].", $idHotel, $puerta_interior, ".$materialPuerta[$i].", $temp_a_acond, ".
        $lastEspacioHotel.");";
    mysql_query($procedurePuerta, $conexion);
}

// - Ventanas Interiores -
for ($i = 0; $i < $numVentanas; $i++) {
    if (!$areaVentana[$i])
        continue;
    $procedureVentana = "CALL puerta_ventana_int_calcs(".$orientacionVentana[$i].", ".$espesorVentana.", ".
        $areaVentana[$i].", $idHotel, $ventana_interior, ".$materialVentana[$i].", $temp_a_acond, ".
        $lastEspacioHotel.");";
    mysql_query($procedureVentana, $conexion);
}

```

Esta vez, el procedimiento almacenado también incluye a las paredes, ya que no se requiere de procesamiento por hora. El código de dicho procedimiento sigue las mismas normas a como se hizo con superficies exteriores.

Finalmente se actualiza la información del espacio y ambiente a procesar:

```
$numEspacioAmb--;

if ($numEspacioAmb == 0) {
    $numAmbientes--;
    if ($numAmbientes == 0) {
        echo '<script type="text/javascript" src="/SistemaHotel/scripts/actualizaPagina.js"></script>';
        echo '<script type="text/javascript">
            setTimeout(\'delayer("resultadosInt.php")\', 200);
        </script>';
    } else {
        $nombreAmb = $ambientes[$numAmbientes][0];
        $numEspacioAmb = (int)$ambientes[$numAmbientes][1];
        $idAmbiente = $ambientes[$numAmbientes][2];
    }
}
```

Si los espacios para el ambiente actual ya fueron procesados, se decrementa el número de ambientes, y si todos los ambientes ya fueron procesados, se redirecciona a la síntesis de resultados obtenidos para el diagnóstico; en caso de que aún existan ambientes por procesar, se actualiza su identificador, el número de espacios y su nombre.

Con esto se da por terminado el proceso de cálculo y adquisición de datos para el área de aire acondicionado. Las demás áreas siguen un proceso similar, con la diferencia del algoritmo aplicado para la realización de cálculos y las tablas en donde son almacenados.

III.3 PUESTA A PRUEBA DEL SISTEMA

A continuación se hace la prueba del sistema, desde el registro y autenticación del usuario con su respectivo hotel, hasta la generación de consultas para obtener los cálculos realizados para el diagnóstico energético.



Figura 22. Vista previa de la página de inicio del sistema web

La página de inicio será el punto de partida para comenzar a trabajar con el sistema, desde la visualización de noticias, eventos, páginas amarillas, etc. Es decir, información disponible a todo público, sin embargo para trabajar con los módulos y áreas energéticas, será necesario estar registrado dentro del sistema.

DATOS DEL RESPONSABLE DEL HOTEL

Nombre(*):	Manuel Estrada
Usuario(*):	mestrada
Contraseña(*):
Vuelve a escribir tu contraseña(*):
IFE(*):	EMHHDFG792502
Correo Electrónico (e-mail):	mestrada@gmail.com
<input type="button" value="Registrar"/>	

Los campos marcados con (*) son obligatorios.

Figura 23. Formulario de Registro de Usuario

Formulario de registro para los usuarios, en caso de faltar algún campo obligatorio o de que ambas contraseñas coincidan, se mostrará un mensaje de error temporal, impidiendo el registro del usuario hasta su corrección.

Al registrar el usuario, se procede a pedir la información del hotel:

Datos del Hotel	
Nombre del Hotel(*):	Las Torres Gemelas
RFC Hotel(*):	TGLRGDTS568426
Calle y Número(*):	Lerdo de Tejada No. 78
Estado(*):	GUERRERO
Ciudad(*):	Acapulco de Juarez
Colonia(*):	Vista Hermosa
Delegación o Municipio(*):	Buena Vista
Código postal(*):	784168
Teléfono(*):	7745521436
Categoría(*):	Cuatro estrellas
Página WEB:	http://www.ltg.org.mx
Razón Social:	
Área del terreno (m2):	750
Área construida (m2):	600
Zona de ubicación:	Turística
<input type="button" value="Registrar Hotel"/>	

Figura 24. Formulario de Registro de Hotel

Si ya se está registrado en el sistema, el formulario de autenticación es como sigue:

El formulario de autenticación tiene un fondo azul claro. En la parte superior, el título "REALIZA TU DIAGNÓSTICO" está subrayado con una línea naranja. Debajo del título, el texto "Datos de Usuario" precede a un recuadro que contiene dos campos de entrada: "Usuario:" y "Contraseña:", cada uno con un campo de texto blanco. En la parte inferior izquierda del recuadro, hay un botón rectangular con el texto "Entrar".

Figura 25. Formulario de Autenticación

Una vez registrado o autenticado se puede trabajar con los módulos del sistema:

La página principal muestra un menú de navegación a la izquierda con los siguientes ítems: INICIO, ENCUESTA (destacado con un recuadro amarillo), DIAGNÓSTICO ENERGÉTICO, RECOMENDACIÓN DE MEDIDAS DE AHORRO, EVALUACIÓN DE MEDIDAS DE AHORRO, IMPLEMENTACIÓN DE MEDIDAS DE AHORRO, MONITOREO DE LAS MEDIDAS DE AHORRO, FACTURACIÓN DE CONSUMO y PAGINAS AMARILLAS. En la parte superior derecha, se muestra "Hotel: 'Las Torres Gemelas'". El contenido principal de la página incluye el mensaje "BIENVENIDO A LA PÁGINA PRINCIPAL DE LOS MÓDULOS DEL SISTEMA" y una instrucción: "PARA COMENZAR DAR CLICK EN EL BOTON ACTIVADO EN EL MENU DE LA IZQUIERDA (RECUADRO AMARILLO)". En la esquina inferior derecha, hay un botón que dice "SALIR".

Figura 26. Página principal de los módulos del sistema web

Este es el punto de partida para comenzar a trabajar con las áreas energéticas que incorpora el sistema. No sin antes haber realizado una encuesta general sobre algunos aspectos del hotel como se muestra a continuación.

REGISTRO DE AMBIENTES

Seleccione los ambientes con los que cuenta el hotel, así como el número de espacios para cada uno y si cuenta o desea acondicionamiento.

	No. de Espacios	Acondicionamiento	
<input checked="" type="checkbox"/> Habitación Simple	2	Si <input checked="" type="radio"/>	No <input type="radio"/>
<input checked="" type="checkbox"/> Habitación Doble	2	Si <input checked="" type="radio"/>	No <input type="radio"/>
<input type="checkbox"/> Habitación Triple			
<input type="checkbox"/> Habitación Cuadruple			
<input checked="" type="checkbox"/> Suite	1	Si <input checked="" type="radio"/>	No <input type="radio"/>
<input type="checkbox"/> Bungalow			
<input checked="" type="checkbox"/> Lobby	1	Si <input type="radio"/>	No <input checked="" type="radio"/>
<input checked="" type="checkbox"/> Sala de Recepcion	1	Si <input type="radio"/>	No <input checked="" type="radio"/>
<input type="checkbox"/> Pasadizo			
<input checked="" type="checkbox"/> Bar	1	Si <input type="radio"/>	No <input checked="" type="radio"/>
<input type="checkbox"/> Restaurante			
<input type="checkbox"/> Auditorio			
<input checked="" type="checkbox"/> Cocina	2	Si <input type="radio"/>	No <input checked="" type="radio"/>
<input type="checkbox"/> Lavandería			
<input type="checkbox"/> Spa			
<input type="checkbox"/> Gimnasio			
<input type="checkbox"/> Cuarto de Maquinas			
<input type="checkbox"/> Sala Multiuso			
<input type="checkbox"/> Oficina			
<input checked="" type="checkbox"/> Sala de Internet	1	Si <input type="radio"/>	No <input checked="" type="radio"/>
<input type="checkbox"/> Alberca			
<input checked="" type="checkbox"/> Estacionamiento	1	Si <input type="radio"/>	No <input checked="" type="radio"/>
<input type="checkbox"/> Jardín			
<input type="checkbox"/> Cancha Deportiva			
<input type="checkbox"/> Estacionamiento Cubierto			

Figura 27. Registro de Ambientes – Proceso de Encuesta

En el formulario se muestra un ejemplo de los posibles ambientes que puede tener un hotel, el número de espacios para cada uno y si cuenta o no con acondicionamiento.

Por cuestiones de ejemplo no se registra una cantidad grande de espacios, sin embargo en un caso real no hay restricción alguna.

Teniendo la encuesta inicial procesada, se puede trabajar con los diversos subsistemas energéticos del hotel, para la obtención del diagnóstico:

ILUMINACIÓN	INTERIOR
	EXTERIOR
AIRE ACONDICIONADO	
CALENTAMIENTO DE AGUA	
REFRIGERACIÓN	
ELEVADORES	
BOMBEO	
SUBESTACIONES	

Hotel: "Las Torres Gemelas"

SIMAE: ÁREAS DISPONIBLES PARA EL DIAGNÓSTICO ENERGÉTICO

SALIR

Figura 28. Página de los subsistemas energéticos del sistema web

En nuestro caso se trabajará el subsistema de aire acondicionado, comenzando con las superficies exteriores.

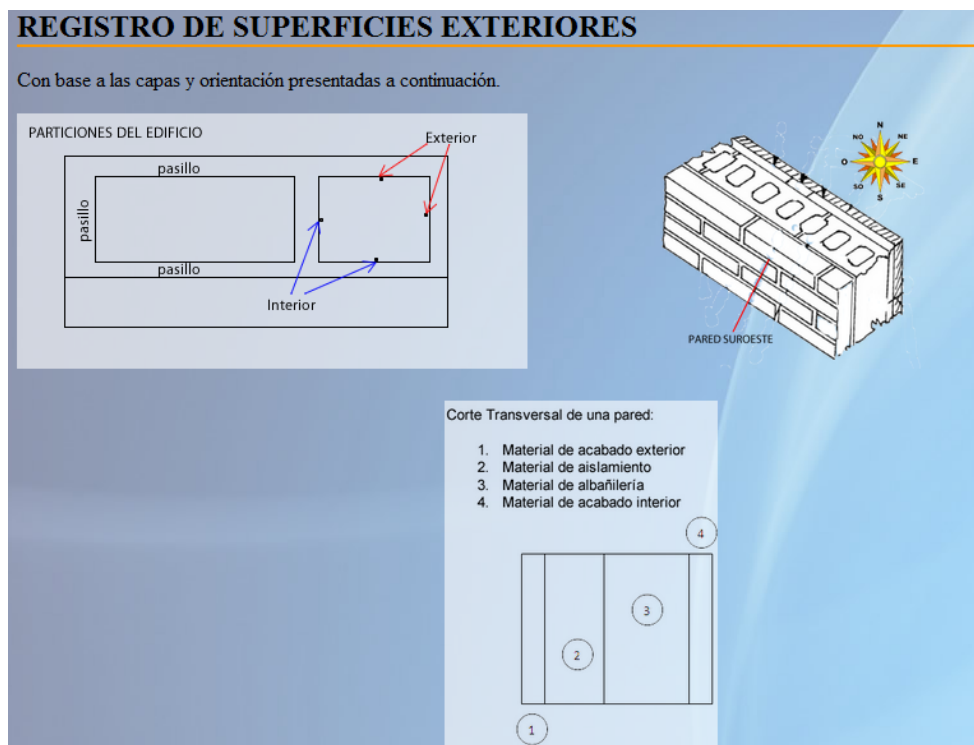


Figura 29. Diagrama explicativo de materiales de construcción, orientación y tipo de pared

Como se explicó en el código del sistema, se generan únicamente los formularios requeridos para las paredes, puertas y ventanas:

Proporciona la siguiente información para el registro de las superficies exteriores del edificio.

No. de Paredes Exteriores: 1

Pared Exterior 1

Tipo de color: Medio
Orientación: Norte
Área (m2): 24
Color: #baa759

Acabado Exterior
Material: Placa de asbesto-Cemento
Espesor (cm): 2

Aislamiento
Material: Capa de fibra de vidrio
Espesor (cm): 3

Material de Albañilería
Material: Ladrillo comun
Espesor (cm): 4

Acabado Interior
Material: Revestimiento de madera
Espesor (cm): 2

No. de Puertas Exteriores: 2

Puerta Exterior 1
Orientación: Norte
Material: Arce, Roble, Madera dura
Espesor (cm): 4
Área (m2): 5

Puerta Exterior 2
Orientación: Este
Material: Madera biselada de 1 x 8
Espesor (cm): 5
Área (m2): 5

No. de Ventanas Exteriores: --No. Ventanas Ext.--

Guardar Valores

Figura 30. Registro de Superficies Exteriores – Aire Acondicionado

Al introducir los datos, todo el proceso de cálculo para la obtención del diagnóstico es ejecutado y almacenado en la base de datos, presentando finalmente una síntesis del calor aportado al hotel por las superficies exteriores a lo largo del día:



Figura 31. Calor aportado al sistema por superficies exteriores

Si se desean procesar las superficies interiores, se presenta un sumario de los ambientes que tienen acondicionamiento y el número de espacios de cada uno para comenzar el análisis.

SUPERFICIES INTERIORES

A continuación se presentan los ambientes que tiene registrado el hotel que tienen Aire Acondicionado, o que se pretenden acondicionar.

Para la realización del diagnóstico será necesario procesar habitación por habitación cada uno de los ambientes, por lo cual puede llevar un tiempo capturar los datos. Esto con la finalidad de realizar un diagnóstico lo más efectivo posible.

AMBIENTE	NO. DE ESPACIOS
Habitacion Simple	2
Habitacion Doble	2
Suite	1

Figura 32. Ambientes y número de espacios a procesar con acondicionamiento

NO. DE ESPACIO: 2 **AMBIENTE: Habitación Simple**

Proporciona la siguiente información para el registro de las superficies interiores del edificio.

Registro de equipo Eléctrico/Electrónico

No. de Paredes Interiores: --No. Paredes Int.--

No. de Puertas Interiores: 1

Puerta Interior 1

Orientación: Sur

Material: Asbesto, Pino, Madera blanda

Espesor (cm): 3

Área (m²): 4.5

No. de Ventanas Interiores: 1

Ventana Interior 1

Orientación: Oeste

Material: Vidrio plano sencillo

Área (m²): 3.2

Guardar Valores

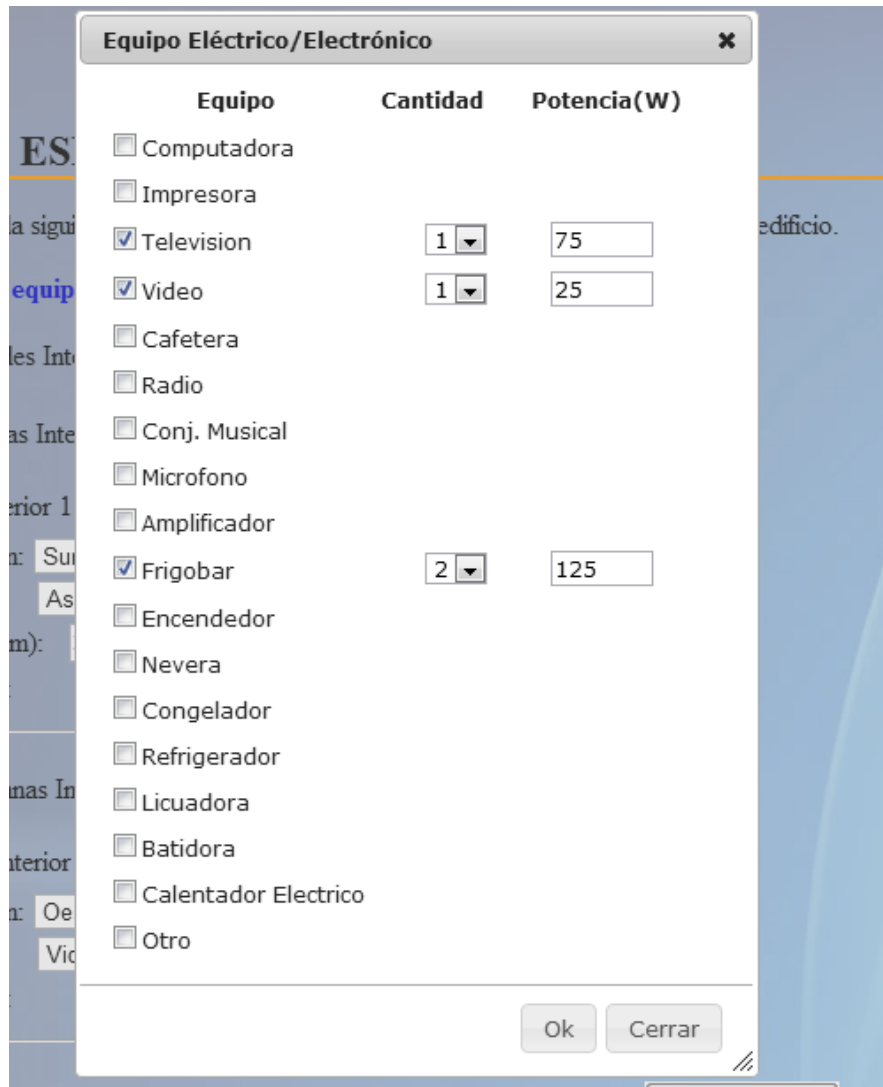
Figura 33. Adquisición de datos para superficies interiores – Aire Acondicionado

A diferencia de las superficies exteriores, aquí se proporciona información sobre el ambiente y el espacio que se está procesando, así como dar la opción de registrar equipo eléctrico/electrónico.

El tipo de datos requeridos es igual que para las superficies exteriores en paredes, puertas y ventanas.

El registro de equipo eléctrico/electrónico se hace por espacio, es decir, es exclusivo para el ambiente y número de espacio actual; cuando se procesen los demás espacios y ambientes, el registro será totalmente independiente inclusive tratándose del mismo ambiente.

El formulario para la adquisición de datos es efectuado a través de un widget:



The image shows a web widget titled "Equipo Eléctrico/Electrónico" with a close button (X) in the top right corner. The widget contains a table with three columns: "Equipo", "Cantidad", and "Potencia(W)". The table lists various types of equipment with checkboxes for selection, dropdown menus for quantity, and input fields for power. The following table represents the data visible in the widget:

Equipo	Cantidad	Potencia(W)
<input type="checkbox"/> Computadora		
<input type="checkbox"/> Impresora		
<input checked="" type="checkbox"/> Television	1	75
<input checked="" type="checkbox"/> Video	1	25
<input type="checkbox"/> Cafetera		
<input type="checkbox"/> Radio		
<input type="checkbox"/> Conj. Musical		
<input type="checkbox"/> Microfono		
<input type="checkbox"/> Amplificador		
<input checked="" type="checkbox"/> Frigobar	2	125
<input type="checkbox"/> Encendedor		
<input type="checkbox"/> Nevera		
<input type="checkbox"/> Congelador		
<input type="checkbox"/> Refrigerador		
<input type="checkbox"/> Licuadora		
<input type="checkbox"/> Batidora		
<input type="checkbox"/> Calentador Electrico		
<input type="checkbox"/> Otro		

At the bottom of the widget, there are two buttons: "Ok" and "Cerrar".

Figura 34. Widget para registro de equipo eléctrico/electrónico

Con esto, ya se tienen procesados los datos tanto de superficies interiores como exteriores, cálculos de insolación en ventanas y puertas de cristal exteriores, así como de equipo eléctrico/electrónico en interiores.

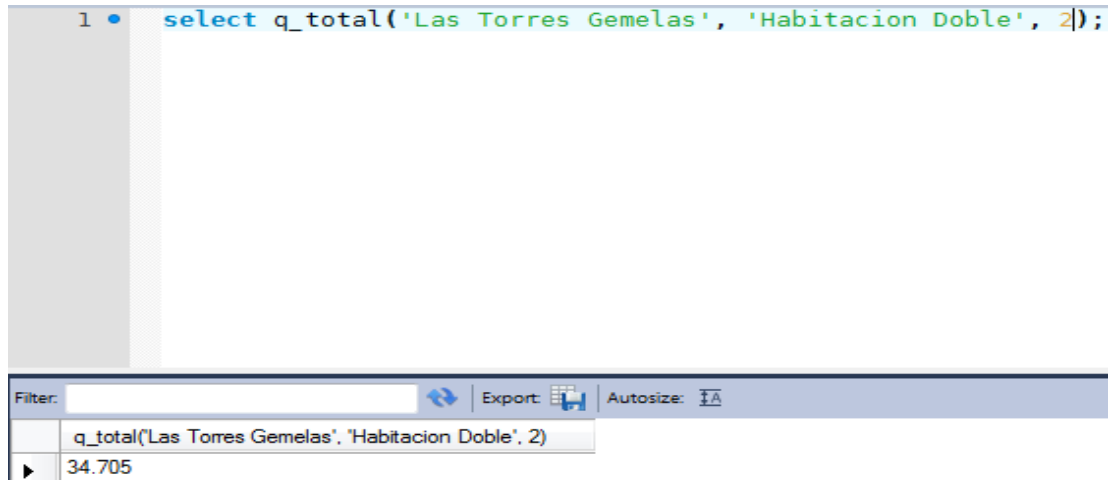
Con esta información ya es posible determinar el calor aportado al hotel por cualquier tipo de superficie, y bajo cualquier nivel de granularidad, desde un espacio en particular, hasta todos los espacios para todos los ambientes, incluyendo exteriores.

Veamos qué tipo de información podemos obtener de los datos adquiridos.

El calor aportado al sistema está dado en Kcal/h, este valor puede ser comparado con datos de placa de equipo de aire acondicionado para observar, por ejemplo, si estamos consumiendo más de lo requerido.

1. Calor aportado al hotel “Las Torres Gemelas” por la habitación doble No. 2

```
1 • select q_total('Las Torres Gemelas', 'Habitacion Doble', 2);
```

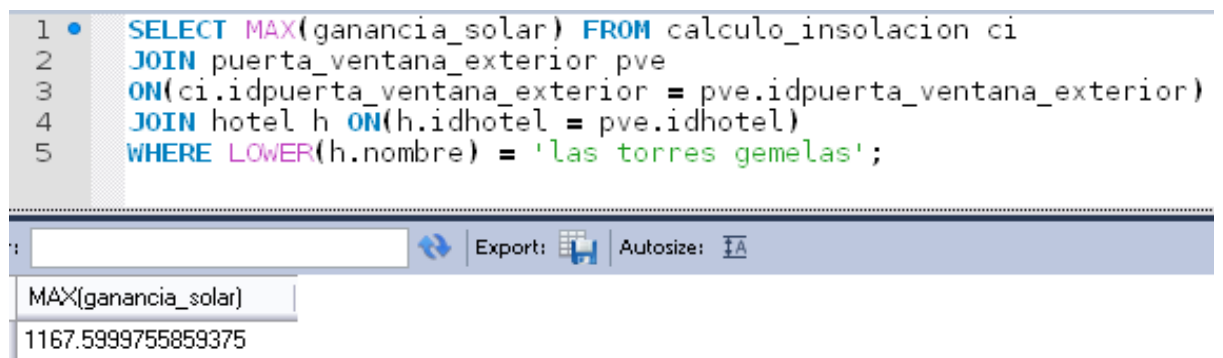


The screenshot shows a SQL query execution window. The query is: `select q_total('Las Torres Gemelas', 'Habitacion Doble', 2);`. The result table has one row with the value 34.705.

q_total('Las Torres Gemelas', 'Habitacion Doble', 2)
34.705

2. Calor absorbido por insolación en ventanas y puertas de vidrio para el hotel “Las Torres Gemelas”

```
1 • SELECT MAX(ganancia_solar) FROM calculo_insolacion ci
2 JOIN puerta_ventana_exterior pve
3 ON(ci.idpuerta_ventana_exterior = pve.idpuerta_ventana_exterior)
4 JOIN hotel h ON(h.idhotel = pve.idhotel)
5 WHERE LOWER(h.nombre) = 'las torres gemelas';
```

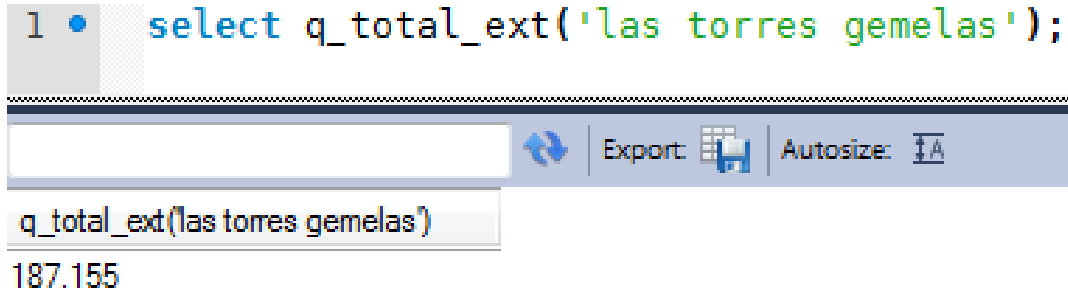


The screenshot shows a SQL query execution window. The query is: `SELECT MAX(ganancia_solar) FROM calculo_insolacion ci JOIN puerta_ventana_exterior pve ON(ci.idpuerta_ventana_exterior = pve.idpuerta_ventana_exterior) JOIN hotel h ON(h.idhotel = pve.idhotel) WHERE LOWER(h.nombre) = 'las torres gemelas';`. The result table has one row with the value 1167.5999755859375.

MAX(ganancia_solar)
1167.5999755859375

3. Calor absorbido por la envolvente del edificio (puertas, ventanas y paredes exteriores) en el hotel “Las Torres Gemelas”

```
1 • select q_total_ext('las torres gemelas');
```



The screenshot shows a SQL query execution interface. The query is `select q_total_ext('las torres gemelas');`. The interface includes a toolbar with 'Export' and 'Autosize' options. Below the query, the result is displayed as `q_total_ext(las torres gemelas)` with a value of `187.155`.

Como observamos el nivel de granularidad en la consulta 1 es mucho mayor que en las consultas 2 y 3, sin embargo, esto no quiere decir que las sentencias para recuperar los datos sean más simples para un nivel de granularidad mucho más bajo, ya que depende de cómo son calculados los datos acorde el algoritmo como lo demuestra el query para la consulta 3:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `q_total_ext`(nombre_hotel VARCHAR(40)) RETURNS decimal(7,3)
BEGIN
    DECLARE q_pve DECIMAL(7,3);
    DECLARE q_se DECIMAL(7,3);

    SELECT SUM(diag_pve.q) INTO q_pve FROM diagnostico_puerta_ventana_ext diag_pve
    JOIN puerta_ventana_exterior pve
    ON(diag_pve.idpuerta_ventana_ext = pve.idpuerta_ventana_exterior)
    JOIN hotel h ON(h.idhotel = pve.idhotel)
    WHERE
    LOWER(h.nombre) = LOWER(nombre_hotel);

    DROP TEMPORARY TABLE IF EXISTS q_max_se;

    CREATE TEMPORARY TABLE q_max_se AS
    (
    SELECT MAX(diag_se.q) AS q_max FROM diagnostico_superficie_ext diag_se
    JOIN superficie_exterior se ON(diag_se.idsuperficie_exterior = se.idsuperficie_exterior)
    JOIN hotel h ON(h.idhotel = se.idhotel)
    WHERE
    LOWER(h.nombre) = LOWER(nombre_hotel)
    GROUP BY diag_se.idsuperficie_exterior
    );

    SELECT SUM(q_max) FROM q_max_se INTO q_se;

    IF IFNULL(q_pve, 0) = 0 THEN
    SET q_pve = 0.0;
    END IF;

    IF IFNULL(q_se, 0) = 0 THEN
    SET q_se = 0.0;
    END IF;

    RETURN q_pve + q_se;
```

Si comparamos esta última sentencia con la que se utilizó para la consulta 1, de un nivel de granularidad mucho mayor, observaremos que casi son iguales en nivel de complejidad:

```
BEGIN

DECLARE q_pvi DECIMAL(7,3);
DECLARE q_si DECIMAL(7,3);

SELECT SUM(diag_pvi.q) INTO q_pvi FROM diagnostico_puerta_ventana_int diag_pvi
JOIN puerta_ventana_interior pvi
ON(diag_pvi.idpuerta_ventana_int = pvi.idpuerta_ventana_interior)
JOIN espacio_hotel eh ON(pvi.idespacio_hotel = eh.idespacio_hotel)
JOIN ambiente a ON(a.idambiente = eh.idambiente)
JOIN hotel h ON(h.idhotel = eh.idhotel)
WHERE
LOWER(h.nombre) = LOWER(nombre_hotel)
AND
LOWER(a.tipo) = LOWER(nom_ambiente)
AND
eh.numero_espacio = no_espacio;

SELECT SUM(diag_si.q) INTO q_si FROM diagnostico_superficie_int diag_si
JOIN superficie_interior si ON(diag_si.idsuperficie_interior = si.idsuperficie_interior)
JOIN espacio_hotel eh ON(si.idespacio_hotel = eh.idespacio_hotel)
JOIN ambiente a ON(a.idambiente = eh.idambiente)
JOIN hotel h ON(h.idhotel = eh.idhotel)
WHERE
LOWER(h.nombre) = LOWER(nombre_hotel)
AND
LOWER(a.tipo) = LOWER(nom_ambiente)
AND
eh.numero_espacio = no_espacio;

IF IFNULL(q_pvi, 0) = 0 THEN
    SET q_pvi = 0.0;
END IF;

IF IFNULL(q_si, 0) = 0 THEN
    SET q_si = 0.0;
END IF;

RETURN q_pvi + q_si;
```

Esto demuestra que todo depende de los cálculos realizados acorde el algoritmo, y no el nivel de granularidad lo que define la complejidad de la recuperación de los datos.

Es posible, sin embargo, observar la estructura de la base de datos para obtener la información bajo el nivel de granularidad que se desee.

Con esto se da por concluida la prueba del sistema, lo que demuestra el alcance del mismo y lo que puede aportar no sólo a la comunidad de hoteleros, sino también el uso estadístico y la minería de datos que puede aportar en el sector energético.

Una de las mejoras que se pueden hacer al sistema en el sentido tecnológico, es la migración hacia una arquitectura definida para el desarrollo de software como el Modelo Vista Controlador o MVC que separa los datos de la aplicación, la lógica del negocio y la interfaz de usuario en tres componentes distintos.

III.4 REFINAMIENTO DEL SISTEMA

El sistema presentado en este trabajo presenta una implementación parcial en MVC en ciertos módulos: autenticación, registro y control de usuarios, con la finalidad de observar las grandes ventajas de utilizarlo, pero antes de entrar en detalle sobre la implementación veamos cómo funciona una arquitectura MVC con el framework de desarrollo Yii (utilizado en el sistema) para PHP.

Yii implementa MVC con el objetivo de hacer más productivo el proceso de desarrollo, en donde los desarrolladores puedan trabajar con los diversos aspectos de la aplicación: interfaz de usuario, modelo de datos y lógica del negocio, de forma independiente para después integrar casi transparentemente el trabajo realizado, produciendo una aplicación robusta y mejor estructurada. [27]

En MVC, el modelo representa la información (los datos) y las reglas del negocio; la vista contiene elementos de la interfaz de usuario como etiquetas de texto y entradas de formularios; el controlador se encarga de gestionar la comunicación entre el modelo y la vista. [27]

Además de implementar MVC, Yii también introduce un front-controller (controlador-principal), llamado `Application`, que encapsula el contexto de ejecución para el procesamiento de peticiones. Este front-controller recopila cierta información sobre la petición del usuario para luego dispararla al controlador apropiado para tratamiento futuro. [27]

El siguiente diagrama muestra la estructura estática de una aplicación Yii.

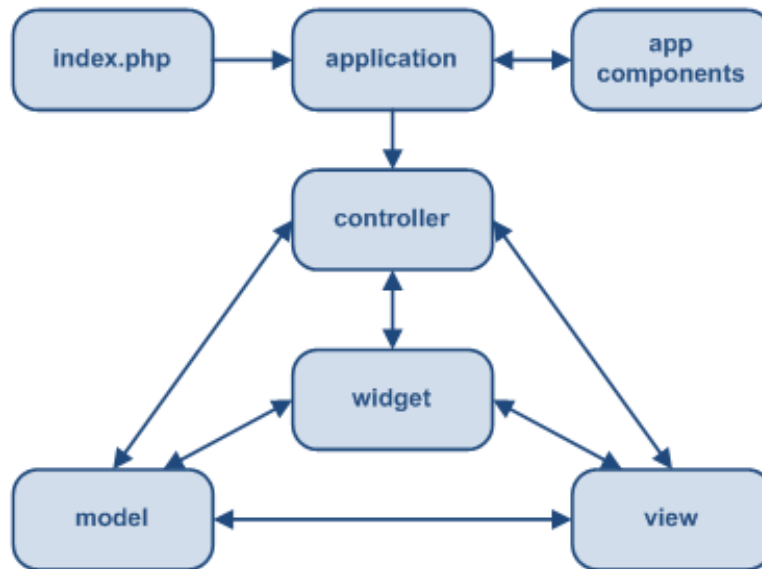


Figura 35. Estructura estática de una aplicación Yii [27]

UN FLUJO DE TAREAS TÍPICO

El siguiente diagrama muestra un flujo de tareas típico de una aplicación Yii cuando resuelve una petición de usuario:

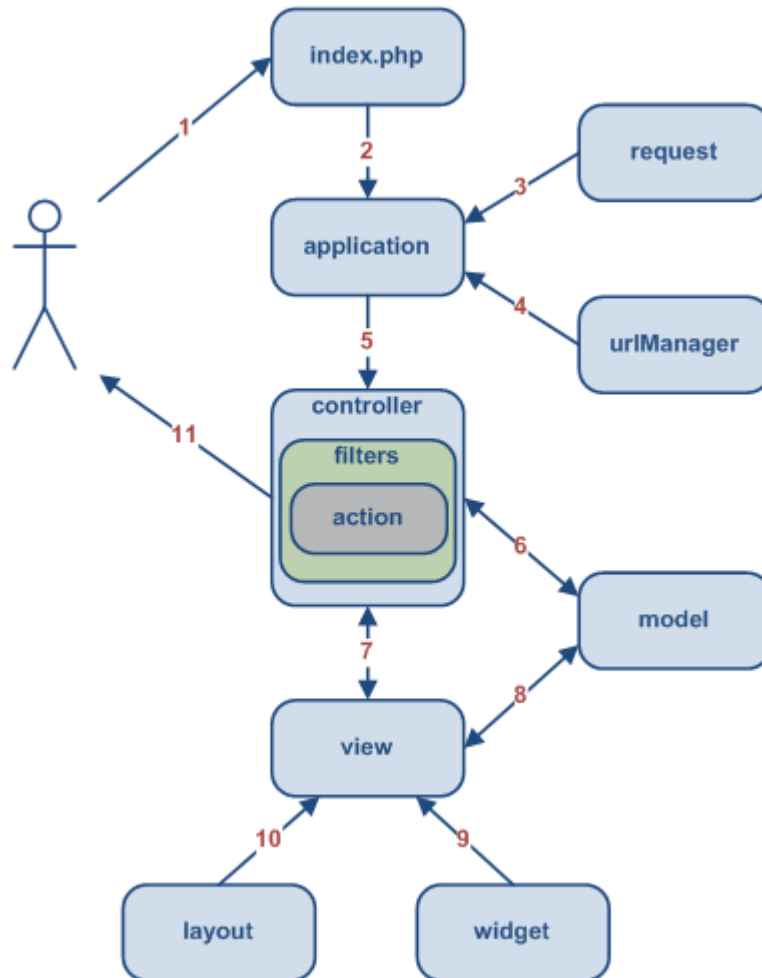


Figura 36. Flujo de tareas en una aplicación Yii [27]

1. Un usuario realiza una petición con la siguiente URL <http://www.ejemplo.com/index.php?r=post/show&id=1>, el servidor Web se encarga de la solicitud mediante la ejecución del script de arranque en `index.php`.
2. El script de entrada crea una instancia de `application` y la ejecuta.
3. La aplicación obtiene la información detallada del pedido del usuario a través del componente de aplicación `request`.
4. La aplicación determina el controlador y la acción requerida con ayuda del componente de aplicación llamado `urlManager`. Para este ejemplo el controlador es `post` que refiere a la clase `PostController` y la acción es `show` que su significado es determinado por el controlador.

5. La aplicación crea una instancia del controlador pedido para resolver la petición del usuario. El controlador determina que la acción `show` refiere al nombre del método `actionShow` dentro de la clase. Entonces crea y ejecuta los filtros asociados con esta acción (ejemplo: control de acceso, benchmarking). La acción es ejecutada si los filtros lo permiten.
6. La acción lee el modelo `Post` cuyo ID es 1 en la base de datos.
7. La acción realiza la vista llamada `show` con el modelo `Post`
8. La vista lee y muestra los atributos del modelo `Post`.
9. La vista ejecuta algunos `widgets`.
10. El resultado realizado es embebido en un esquema (layout).
11. La acción completa la vista realizada y se la muestra al usuario.

A gran escala, este es el procedimiento que se sigue en toda petición de usuario en una aplicación Yii. Por cuestiones de extensión no se tratarán a detalle las cuestiones técnicas de configuración de una aplicación Yii. [27]

Observemos ahora como sería el sistema web trabajando bajo un esquema MVC.

MÓDULO DE REGISTRO Y AUTENTICACIÓN

Bajo un esquema MVC se necesita de un modelo (representado por una clase o clases que hacen referencia a objetos con los que se accede a la base de datos), una vista (representada por la interfaz que se le presenta al usuario como los formularios) y de un controlador (clase que gestiona las peticiones de usuario y encargado de crear las instancias de los modelos y generar las vistas).

Para la autenticación de usuarios, a diferencia de como se hizo en el modelo tradicional presentado, el nivel de seguridad y control es gestionado por un componente del framework Yii denominado 'authManager', que nos permitirá aplicar un algoritmo de encriptación para validar los usuarios.

El modelo de validación se implementa a través de la clase 'CUserIdentity' como sigue:

```
class UserIdentity extends CUserIdentity
{
    private $_id;

    public function authenticate()
    {
        $username = strtolower($this->username);
        $pass = $this->password;
        $user = Usuario::model()->findByAttributes(array('id_usuario'=>$username));

        if ($user == null) {
            $this->errorCode = self::ERROR_UNKNOWN_IDENTITY;
        } else if ($user->validate_password($pass)) {
            $this->_id = $user->id;
            $this->username = $user->id_usuario;
            $this->errorCode = self::ERROR_NONE;
        }

        return $this->errorCode === self::ERROR_NONE;
    }

    public function getId() {
        return $this->_id;
    }
}
```

Mientras que los detalles sobre el algoritmo de encriptación se encuentran en el modelo del Usuario como sigue:

```
protected function beforeSave() {
    parent::beforeSave();
    if ($this->password && $this->isNewRecord) {
        $this->id_usuario = strtolower($this->id_usuario);
        $this->hashed_password = $this->hash_password($this->password);

        return true;
    }
    return true;
}

private function hash_password($password) {
    $salt = bin2hex(mcrypt_create_iv(32, MCRYPT_DEV_URANDOM));
    $hash = hash('sha256', $salt . $password);

    return $salt . $hash;
}

public function afterSave() {
    parent::afterSave();
    if ($this->isNewRecord) {
        $loginObj = new LoginForm;
        $loginObj->username = $this->id_usuario;
        $loginObj->password = $this->password;

        $loginObj->login();
    }
}

public function validate_password($password) {
    $salt = substr($this->hashed_password, 0, 64);
    $hash = substr($this->hashed_password, 64, 64);

    $password_hash = hash('sha256', $salt . $password);

    return $password_hash == $hash;
}
```

En este modelo, la contraseña es encriptada y almacenada en la base de datos con el algoritmo de Hash seguro de 256 bits.

Para el registro de usuarios y hoteles, se utilizan los modelos, las vistas y los controladores correspondientes.

Veamos los diferentes escenarios del sistema trabajando en MVC.

SIMAE: SISTEMA MULTIAGENTE PARA LA GESTIÓN DE LA EFICIENCIA ENERGÉTICA EN HOTELES

Inicio SIMAE Noticias Experiencias Enlaces Eventos

OBJETIVO

Sistema Web para la Gestión de la Eficiencia Energética en Hoteles. Que busca mejorar el consumo y eficiencia energética en el sector hotelero.

ORGANISMOS

- UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
- UNIVERSIDAD AUTÓNOMA DE CAMPECHE
- UNIVERSIDAD VERACRUZANA

CONTACTO

Sección que incluirá información de contacto, para el envío de comentarios y sugerencias sobre el uso del sistema, así como el reporte de posibles errores que puedan existir.

NOTICIAS SOBRE AHORRO DE ENERGÍA EN HOTELES

Sección dedicada a la publicación de experiencias e implementaciones que han tenido diversos hoteles con el sistema. Se presentan casos de éxito, en donde se resumen las medidas de ahorro más significativas que el sistema ha propuesto para las diversas áreas con las que cuenta el hotel.

Apartados adicionales para la presentación de los reportes más significativos efectuados en otros hoteles, que de igual manera han sido beneficiados, gracias a la utilización de los diversos módulos para la gestión de la eficiencia energética que implementa el sistema.

Operaciones - SIMAE

Sistema Multiagente para la Gestion de la Eficiencia Energetica en Hoteles

Paginas Amarillas para Ahorro Energético

Copyright © 2012 por CCADET - UNAM.
Todos los derechos reservados.

Figura 37. Página de inicio del sistema – MVC

Inicio SIMAE Noticias Experiencias Enlaces Eventos

Acceso - SIMAE

Por favor ingrese los datos requeridos para acceder.

Los campos marcados con * son obligatorios.

Id de Usuario *

Contraseña *

Mantener autenticado

Entrar

Nuevo usuario

Registro

Figura 38. Página de autenticación de usuarios – MVC

DATOS DEL RESPONSABLE DEL HOTEL

*Los campos marcados con * son obligatorios.*

Nombre *

Id de Usuario *

Contraseña *

Vuelva a escribir su contraseña *

Ife *

E-mail

Figura 39. Formulario para registro de usuarios – MVC

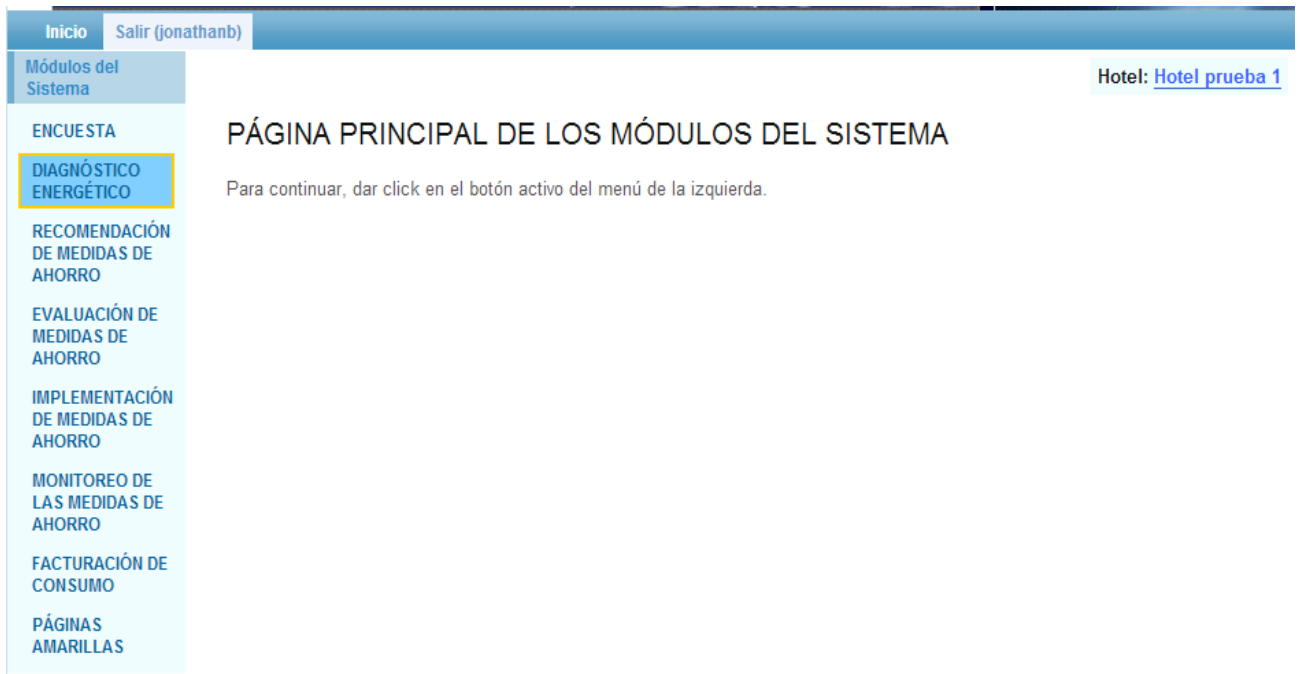


Figura 40. Página principal de los módulos del sistema – MVC

Finalmente, migrando el sistema a un esquema MVC se tiene un mejor nivel de organización en el código así como un mayor nivel de seguridad, desde el momento de validación de usuarios hasta la inserción de datos en el modelo de datos (base de datos); también se regula el acceso a áreas del sistema dependiendo las reglas del negocio y flujo de trabajo.

Destaquemos ahora los puntos más relevantes sobre las ventajas de utilizar un sistema computarizado para la gestión energética a diferencia de un análisis empírico.

IV. VENTAJAS DE UTILIZAR UN SISTEMA COMPUTARIZADO PARA LA GESTIÓN EFICIENTE DE LA ENERGÍA VS. EL ANÁLISIS EMPÍRICO

IV.1 LA COMPUTACIÓN, LÓGICA Y MATEMÁTICAS COMO HERRAMIENTAS DE ANÁLISIS

Desde el inicio de los tiempos, el hombre ha desarrollado sistemas para la resolución de problemas, desde sistemas numéricos hasta los avanzados sistemas de cómputo de hoy en día.

Sin embargo, muchos de estos sistemas dependen de otros para su funcionamiento, como es el caso de las computadoras, la lógica y las matemáticas; por ejemplo, un sistema de cómputo se basa en la lógica y matemáticas para la toma de decisiones y realización de cálculos.

Hoy en día estos sistemas son puestos en práctica para un sin fin de usos, aplicados a la educación, investigación, ciencia, desarrollo tecnológico, medicina, ingeniería, etc.

Con el avance tecnológico que existe actualmente, es posible delegar responsabilidades a los sistemas de cómputo para optimizar y hacer que las cargas de trabajo sean más prácticas y eficientes.

El sector energético no es la excepción en cuanto al uso de herramientas tecnológicas, sin embargo, en áreas como el turismo y los hoteles, no se cuenta con una herramienta computarizada que les permita analizar y tomar medidas para un mejor aprovechamiento de la energía que consumen. De ahí que, la Secretaría de Energía (SENER) y el CoNaCyt (Consejo Nacional de Ciencia y Tecnología) optaran por financiar este proyecto en donde se integra el conocimiento de expertos en gestión energética en una herramienta computarizada, evitando costos de contratación y tiempos de espera para la elaboración de auditorías energéticas, permitiéndoles determinar problemas potenciales en materia de energía más rápido y eficiente.

Cabe mencionar que para que un sistema trabaje adecuadamente, los datos que se le proporcionen deben ser lo más precisos posible, ya que todo su funcionamiento se basa con respecto a esta información proporcionada por el usuario.

Se espera que un buen diagnóstico energético va ir de la mano con datos reales sobre las condiciones energéticas actuales que existen en los hoteles, ante esta restricción, se espera que el usuario hotelero tome en serio la captura de datos y se forme una disciplina al implementar las recomendaciones que el sistema le genere.

Con este sistema se busca promover el ahorro energético así como la creación y utilización de herramientas computarizadas que beneficien a la sociedad y al medio ambiente.

IV.2 ENFOQUE EMPÍRICO PARA LA IMPLEMENTACIÓN DE UN SISTEMA GESTOR

Para construir un sistema inteligente capaz de analizar un sector como el energético, es necesario encapsular el conocimiento adquirido por expertos y buscar la forma de introducirlo en el sistema para que pueda ser explotado de forma práctica y eficiente.

Generalmente un experto tiene la suficiente experiencia que prueban la certeza y validez de sus resultados, es decir, se basan en un enfoque empírico para diagnosticar y resolver problemas, sin descartar el conocimiento teórico que poseen.

Hablando del sector energético en los hoteles y de la forma en que se determinan los problemas potenciales que existen, es necesario realizar una auditoría energética en donde los resultados obtenidos permiten al experto determinar las medidas necesarias para optimizar la eficiencia.

El experto cuenta con un amplio espectro de datos experimentales obtenidos a lo largo de su carrera, lo que permite establecer un rango para el sistema más reducido y preciso, aunque no se descarta la posibilidad de realizar un muestreo antes de arrojar los resultados.

El sistema incorpora una serie de tablas de datos que sirven para la realización de cálculos, como lo haría un experto, sin embargo, los resultados obtenidos son comparados con información proporcionada por el experto acorde una serie de parámetros e indicadores, según las condiciones del lugar, para finalmente poder determinar las recomendaciones pertinentes y su implementación. Es por eso que las encuestas para la recuperación de datos pueden ser tediosas y muy técnicas para el usuario final, pero es un proceso vital y necesario para el correcto funcionamiento del sistema inteligente.

La forma en que una computadora y un ser humano analizan y verifican la información es diferente, veamos un ejemplo:

SUDOKU DE COLORES [20]

En un simple juego de lógica, como Sudoku, las personas reaccionan de forma compleja: resuelven el problema usando una mezcla de percepción, expectativa y raciocinio; actúan de una forma diferente a la del ordenador. [20]

Para ayudar a las personas a entender el beneficio de una nueva visión de computación, la del modelado empírico, los investigadores del departamento de ciencia de la computación de la universidad de Warwick, en Inglaterra, desarrollaron un juego de sudoku con colores. Por medio del juego los usuarios consiguen modelar, simular y visualizar sistemas de forma general. Consiguen comprender mejor como los ordenadores hacen operaciones lógicas y cálculos. Eso es modelado empírico. [20]



Figura 41. Sudoku de Colores [20]

El sudoku de colores tiene cuadrados con números de colores. El jugador rellenará los cuadrados vacíos siguiendo los colores y los números coloreados de los otros cuadrados. El juego es una forma de explorar como la lógica (ordenador) y la percepción (humano) interaccionan. Este tipo de interacción, analizada por el modelado empírico, es vista como una solución para construir mejores programas informáticos. [20]

Como se explica, existen formas de hacer que un enfoque empírico sea de utilidad para el desarrollo e investigación a través del uso de las tecnologías de la información (IT).

IV.3 OTRAS ÁREAS Y PROFESIONES QUE INTERVIENEN PARA LA IMPLEMENTACIÓN DE UN SISTEMA GESTOR DE ENERGÍA

Como se trata de un sistema relacionado con la gestión de la eficiencia energética, es necesario trabajar en conjunto con profesionales de diversas áreas.

❖ Ingenieros Mecánicos

Son los que se encargan de proporcionar los datos necesarios para la realización de cálculos, así como pruebas de escritorio específicas para diversos casos de uso para su posterior generalización.

Verifican que los resultados arrojados por el sistema sean correctos, y son los que proponen las diversas medidas de ahorro que se incluirán dependiendo estos resultados. También proponen la viabilidad de soluciones de energía alternativa como calentadores solares, celdas fotovoltaicas y energía eólica.

❖ Arquitectos

Tienen a su cargo el análisis de materiales, sistemas pasivos y su repercusión en materia energética. Proponen técnicas para aprovechar mejor las condiciones climatológicas y del medio ambiente en beneficio del ahorro energético, así como de los materiales de construcción adecuados, siguiendo las diversas normas establecidas por los reglamentos.

❖ Ingenieros Eléctricos

Tienen a su cargo de verificar que el aprovechamiento de la energía eléctrica sea óptimo; proporcionan información sobre el tipo de focos a utilizar, mantenimiento a la instalación eléctrica, tarifas eléctricas, medición de niveles de carga, etc.

Con su apoyo, es posible recomendar horarios de uso para diversos aparatos eléctricos, logrando reducción de costos en la facturación y consumo.

❖ Termodinámica

Para entender las ecuaciones utilizadas en los algoritmos para las áreas energéticas que se analizan: aire acondicionado, refrigeración y bombeo, es necesario profundizar en conceptos como calor, energía interna, coeficientes de transferencia, trabajo, presión y temperatura.

❖ Diseño gráfico

Como toda aplicación web, es necesario añadir elementos decorativos para la presentación del sistema; se necesitan poseer conocimientos sobre el diseño gráfico en web a través de las hojas de estilo (CSS) para tener un entorno atractivo y funcional para el usuario. El proceso de diseño puede ser complejo, requiriendo la manipulación de imágenes y animaciones, obligando aprender paquetería específica para tales fines como Flash y Photoshop.

❖ Ingeniería de Software

Se tomaron como base algunas de sus metodologías para el desarrollo del sistema, como el modelo basado en prototipos y la arquitectura MVC. Sirvió también para seguir un orden y dar organización al desarrollo: definición de requerimientos, diseño del sistema, codificación y pruebas.

IV.4 RELACIÓN COSTO-BENEFICIO-TIEMPO

Para determinar las fallas que existen en cuanto al aprovechamiento de la energía es necesario llevar a cabo una auditoría energética, con el fin de mejorar el consumo.

Si analizamos los requerimientos de una auditoría energética, observamos lo siguiente:

- ✓ Personal capacitado en el ámbito energético (ingenieros eléctricos, mecánicos, técnicos, personal de apoyo, etc.)
- ✓ Equipo para la realización de mediciones
- ✓ Si se trata de una auditoría fuera del lugar de residencia, se deberán incluir gastos para: viáticos, hospedajes y comida
- ✓ Tiempos de traslado

En cambio si hacemos uso del sistema web proporcionado, se requiere de:

- ✓ Acceso a un equipo de cómputo con conexión a internet (se puede trabajar de manera local)
- ✓ Capacitación inicial y/o manual de usuario del sistema

Con el uso del sistema web se busca sustituir de cierta manera la auditoría energética, ahorrando los costos y tiempos que acarrea; si se utiliza en conjunto con la auditoría puede traer grandes beneficios a mediano y largo plazo en el ahorro de energía y facturación.

Apuntemos sin embargo, que se trata de un sistema nuevo para tratar la eficiencia energética que con el tiempo y uso que se le dé, irá mejorándose para tal vez convertirse en una herramienta de gran utilidad y de gran difusión, no sólo para los hoteles de México sino por qué no, para hoteles en todo el mundo.

V. CONCLUSIONES

Con lo expuesto en este trabajo, se tiene un panorama de un sistema web capaz de beneficiar al sector hotelero en materia de gestión eficiente de la energía. A pesar de que sólo fue expuesta la implementación para Aire Acondicionado no quiere decir que no se tengan las demás áreas energéticas implementadas en el sistema; tal vez algunas de ellas en fase inicial y todavía con mucho trabajo por realizar, pero que, con perseverancia y dedicación se tendrá un producto final funcional y efectivo.

La carga de trabajo que se tiene con el sistema requiere de un trabajo en equipo y de varias disciplinas. Aquí se expone la forma en que el sistema está pensado para su implementación: plataforma como servicio en la nube y el flujo de información que se sigue para trabajar con él.

Se espera que este trabajo despierte interés para el desarrollo de futuras aplicaciones en beneficio de la sociedad y el medio ambiente, demostrando que en las tecnologías de la información (IT) hay mucho potencial de desarrollo e innovación.

VI. BIBLIOGRAFÍA

1. Mikkilineni, R., Sarathy, V. (2009). *Cloud Computing and the Lessons from the Past. Enabling Technologies: Infrastructures for Collaborative Enterprises*, 57-62. [Agosto 2012]
2. Rajan, S., Jairath, A. (2011). *Cloud Computing: The Fifth Generation of Computing. Communication Systems and Network Technologies (CSNT)*, 665-667. [Agosto 2012]
3. Sadashiv, N., Kumar, S.M.D. (2011). *Cluster, grid and cloud computing: A detailed comparison. Computer Science & Education (ICCSE)*, 477-482. [Agosto 2012]
4. Lawton, G. (2008). *Developing Software Online With Platform-as-a-Service Technology. Computer*, 41(6), 13-15. [Julio 2012]
5. Borroto, A., Lapido, M., Monteagudo, J., Armas, M. (2005). *La gestión energética: una alternativa eficaz para mejorar la competitividad empresarial. Energética*, 33, 65-69. [Febrero 2012]
6. OpenStack Project. *Open source software for building private and public clouds*. 2012. < <http://www.openstack.org/>>. [Junio 2012]
7. Energy Office. *Gestión de la energía: Definición*. 2012. < <http://www.energyoffice.org/spain/tools/emanagement/definicion/main.html>>. [Enero 2012]
8. Global Digital Vision (GDV). *Cloud Computing*. 2011. < <http://www.gdv.com.au/cloud-computing.html>>. [Febrero 2012]
9. Conpas.net. *Definición SaaS: Software como Servicio en la empresa*. 2011. < http://www.conpas.net/por_que_saas.html>. [Enero 2012]
10. Tecnología Pyme. *SaaS: el software como servicio online*. 2011. < <http://www.tecnologiapyme.com/servicios-web/saas-el-software-como-servicio-online>>. [Febrero 2012]
11. Knorr, Eric. *What the 'private cloud' really means*. 2010. < <http://www.infoworld.com/t/cloud-computing/what-the-private-cloud-really-means-463?page=0,0>>. [Enero 2012]
12. Chou, Yung. *What is private cloud*. 2011. < <http://blogs.technet.com/b/yungchou/archive/2011/03/21/what-is-private-cloud.aspx>>. [Enero 2012]
13. CloudWays. *What is public cloud? and 4 basic features of it*. 2011. < <http://www.cloudways.com/blog/what-is-public-cloud/>>. [Febrero 2012]

14. McKendrick, Joe. *Private clouds are from mars, public clouds are from venus... sort of*. 2012. <<http://www.forbes.com/sites/joemckendrick/2012/05/22/private-clouds-are-from-mars-public-clouds-are-from-venus-sort-of/>>. [Febrero 2012]
15. Jani. *The tenets of the cloud*. 2009. <<http://www.janakiramm.net/blog/the-tenets-of-the-cloud>>. [Febrero 2012]
16. Subramanian, Krishnan. *Hybrid Clouds*. 2011. <http://www.trendmicro.com/cloud-content/us/pdfs/business/white-papers/wp_hybrid-clouds-analyst-ksubramanian.pdf>. [Marzo 2012]
17. Linkhead. *Cloud*. 2010. <http://en.linkhead.com/_d271714402.htm>. [Marzo 2012]
18. Eguíluz, J. *Introducción a XHTML*. 2009. <<http://www.librosweb.es/xhtml/index.html>>. [Marzo 2012]
19. Wikipedia. *SQL*. 2010. <<http://es.wikipedia.org/wiki/SQL>>. [Marzo 2012]
20. Department of Computer Science, University of Warwick. *Colour Sudoku*. 2009. <<http://www2.warwick.ac.uk/fac/sci/dcs/research/em/sudoku>>. [Septiembre 2012]
21. Heidi/sem. *Is 2012 the year of PaaS?* 2012. <<http://www.undertheradarblog.com/blog/is-2012-the-year-of-paas/>>. [Febrero 2012]
22. Lawton, G. (2008). *Developing Software Online with Platform-as-a-Service Technology*. Industry Trends, 13-15.
23. Saasmanía. *Windows Azure – Nueva paas de Microsoft*. 2008. <<http://www.saasmania.com/blog/2008/10/27/windows-azure-nueva-paas-de-microsoft/>>. [Enero 2012]
24. Tata Communications. *Infrastructure-a- a-Service, Fulfilling the Promise of Cloud Computing*. 2012. <http://www.tatacommunications.com/downloads/whitepapers/Tata_Communications_IaaS_WhitePaper_v2.0-web.pdf>. [Marzo 2012]
25. Sadashiv, N., Kumar, D. (2011). *Cluster, Grid and Cloud Computin: A Detailed Comparison*. The 6th International Conference on Computer Science & Education (ICCSE), 477-482. [Febrero 2012].
26. PHP: Hypertext Preprocessor. *PHP and other languages*. 2010. <<http://mx.php.net/manual/en/faq.languages.php>>. [Marzo 2012]
27. Yii Framework. *Model-View-Controller (MVC)*. 2011. <<http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>> [Septiembre 2012]
28. Python Programming Language. *About Python*. 2011. <<http://www.python.org>> [Noviembre 2012]

29. Java Server Pages (JSP). *JavaServer Pages Overview*. 2011.
<<http://www.oracle.com/technetwork/java/overview-138580.html> > [Noviembre 2012]
30. Microsoft .NET. *.NET Definition*. 2011.< http://es.wikipedia.org/wiki/Microsoft_.NET >
[Noviembre 2012]
31. VentureBeat /Dev. *Exclusive: PHP, the web's most popular programming language, is coming to mobile*. 2012. <<http://venturebeat.com/2012/10/02/php-andi-gutmans-future-mobile/> > [Noviembre 2012]
32. W3Techs. *Usage of server-side programming languages for websites*. 2012.
<http://w3techs.com/technologies/overview/programming_language/all >
[Noviembre 2012]