



Instituto Superior Minero Metalúrgico de Moa

“ Dr. Antonio Núñez Jiménez ”

Facultad de Geología-Minas

Departamento de Informática

Trabajo de diploma para optar por el título de
Ingeniero en Informática

TÍTULO: Sistema informático para mejorar la toma de decisiones en la explotación de los Grupos Electrógenos Diésel de Moa.

Autor: Angel Cutiño Guilarte

Tutor: Ing. Yisel Rivera Samón

Moa, Holguín,
Cuba, Julio 2012
Año del 54 Aniversario del Triunfo de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez” para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del 2012.

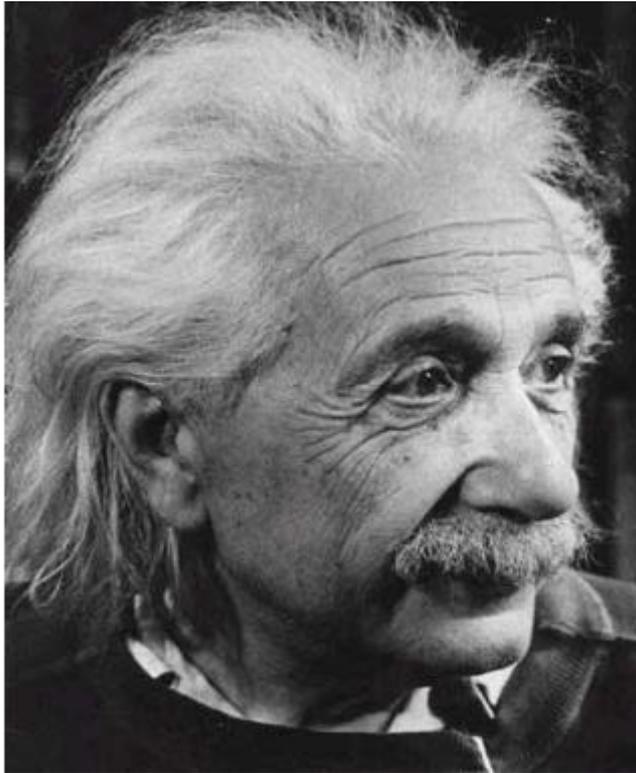
Angel Cutiño Guilarte

Ing. Yisel Rivera Samón

Firma del Autor

Firma del Tutor

Pensamiento.



“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber. “

A. Einstein

Dedicatoria

Este trabajo va dedicado especialmente a mis Padres, por darme vida y todo lo que ha sido necesario para que yo logre mis metas.

A mi madre Irlaide,

que con mucho amor ha sabido darme la educación y forjado valores que solo una madre sabe cómo hacerlo.

A mi padre Angel Manuel,

al igual que mi madre me ha dado su amor incondicionalmente, educación y siempre me ha inculcado valores de honestidad y perseverancia con los cuales he podido abrirme paso durante toda mi vida y así cumplir mis sueños.

A mi hermano Isnel,

que de una forma u otra me ha ayudado en momentos que solo un hermano sabe cómo hacerlo.

Agradecimientos

Quisiera comenzar agradeciendo: a mis padres, mi hermano y toda mi familia en general que son lo más importante de mi vida, que ha sido una familia donde nunca ha faltado el amor y siempre han estado en los buenos y malos momentos.

No puede faltar el agradecimiento a mis compañeros de aula y de cuarto que somos como una familia: Leo, Leonel (los hermanos isnagas), Daniar, Roilan, Lecusay, Yariel, Yuniel, Aluchy, Evelio, Rafael, Carlos, Hinojosa, Robin, Acel (los pastores).

Un agradecimiento a mi tutora Yisel, a la profe Alina por brindarme su ayuda incondicional, muchas gracias.

Resumen

Las empresas, en su afán por elevar la eficiencia de sus procesos de gestión, se han trazado estrategias en las que, debido al auge de las tecnologías, se ha hecho evidente la necesidad del uso de software para la optimización de cualquier actividad.

En este contexto, el presente trabajo tuvo como objetivo el desarrollo de una aplicación de escritorio para mejorar el proceso de toma de decisiones respecto al funcionamiento de los Grupos Electrónicos Diésel en Batería de Moa. Al mismo, se le dio cumplimiento a través de las diferentes etapas de la metodología XP, las cuales fueron expuestas en esta memoria escrita, además del estudio de factibilidad del proyecto.

Durante el desarrollo de la investigación se precisó que el software contribuye a la seguridad, disminución de errores y mejoras en el funcionamiento de los Grupos Electrónicos Diésel en Batería de Moa; además de que se presenta como punto de partida para desarrollar y/o adjuntar nuevos módulos.

Abstract

The companies, in their desire to elevate the efficiency of their administration processes, strategies have been traced in those that, due to the peak of the technologies, it has become evident the necessity of the software use for the optimization of any activity.

In this context, the present work had as objective the development of a desk application to improve the process of taking of decisions regarding the operation of the Groups Electrogenos Diesel in Battery of Moa. To the same one, he was given execution through the different stages of the methodology XP, which were exposed in this written memory, besides the study of feasibility of the project.

During the development of the investigation he/she was necessary that the software contributes to the security, decrease of errors and improvements in the operation of the Groups Electrogenos Diesel in Battery of Moa; besides that is presented as starting point to develop y/o to attach new modules.

INDICE

INTRODUCCIÓN	1
Estructura de la Tesis.....	4
Capítulo I Fundamentación Teórica	6
1.1 Introducción	6
1.2 Estado del arte.....	6
1.2.1 Antecedentes.....	6
1.3 Lenguajes de Programación	8
1.4 Arquitectura	11
1.5 Sistemas Gestores de bases de datos (SGBD).....	13
1.6 Metodologías para el desarrollo de software	14
XP	15
Fases de la Metodología XP	17
1.7 Propuesta de solución	17
1.8 Herramientas y metodología a emplear en la propuesta de solución.	17
1.8.1 MySQL como Sistema Gestor de Base de Datos	17
1.8.2 EMS MySQL Manager como Cliente	18
1.8.3 ¿Por qué XP como metodología?	18
1.8.4 ¿Por qué Java?	19
1.8.5 Ambiente de Desarrollo Integrado (IDE) elegido: NetBeans 7.0.	20
1.8.6 Visual Paradigm.....	20
1.8.7 Embarcadero ER/Studio.	21
Conclusiones del Capítulo	22
Capítulo II Planeación y Diseño	23
2.1 Introducción	23
2.2 Planificación.....	23
2.2.1 Personal relacionado con el sistema	24
Historias de usuarios	24
Planificación de entregas.....	25
2.3 Diseño.....	27
2.3.1 Clases, Responsabilidades y Colaboración.....	27
2.4 Conclusiones	28
Capítulo 3: Implementación y Pruebas.....	29
3.1 Introducción	29
3.2 Implementación	29
3.2.1 Desarrollo de las iteraciones.....	30
3.2.2 Tareas por Historias de Usuarios	30
3.3 Pruebas	32
3.3.1 Pruebas de aceptación	33
3.4 Conclusiones	34
Capítulo 4: Estudio de Factibilidad del Proyecto.	35
4.1 Introducción	35
4.2 Análisis Beneficio Costo ABC	35
4.2.1 Efectos Económicos	35
4.2.2 Beneficios y Costos Intangibles en el Proyecto	36
4.3 Gráfica de la solución manual y solución automatizada	39
4.4 Conclusiones	40

Conclusiones Generales.....	41
Recomendaciones.....	42
Referencias Bibliográficas.....	43
Anexos	44
Anexo 1 Historias de Usuarios.....	44
Anexo 2: Tarjetas CRC	49
Anexo 3: Interfaz de usuario	52
Anexo 4: Tarjetas de Tarea	56
Anexo 5: Pruebas de Aceptación.....	60

INTRODUCCIÓN

En la actualidad existe una fuerte voluntad del Gobierno por lograr la Informatización de la Sociedad Cubana. A tales efectos se han dado pasos importantes para materializar estos aspectos. Por mencionar algunos:

- La creación de Laboratorios de Computación en todas las Escuelas del país desde la enseñanza primaria.
- La creación de la Universidad de Ciencias de la Información (UCI).
- La creación de la Oficina Nacional de Informatización de la Sociedad (ONIS).

Estas son algunas acciones que han permitido lograr una infraestructura tecnológica adecuada para ser utilizadas en todas las esferas de la Sociedad y la formación de los recursos humanos capaces de utilizar la informática de forma eficiente en función del desarrollo socio-económico del país. Por otra parte, un elemento fundamental para la elevación de la satisfacción de todos es la utilización de las nuevas tecnologías en función de la sociedad. El desarrollo empresarial y las necesidades en la vida cotidiana de cada persona han sido fieles demandantes de este proceso revolucionario. Cuba ha utilizado una serie de estrategias para extender la informatización a toda la Isla, poniéndose en práctica en la salud, la educación, la generación de energía y otros. Esto se debe principalmente al hecho de que las políticas y tácticas seguidas por nuestro Estado, orientan el desarrollo de la informatización y la Informática para elevar los objetivos económicos y sociales en todas las manifestaciones y ramas. En la generación de energía, a pesar del bloqueo económico de Estados Unidos, Cuba se propuso invertir en la Generación Distribuida (GD) con el objetivo de garantizar el suministro de energía a los consumidores residenciales y económicos. La GD no es más que: **la generación o el almacenamiento de energía eléctrica a pequeña escala, lo más cercana al centro de carga, con la opción de interactuar con la red eléctrica, y en algunos casos, considerando la máxima eficiencia energética.**

La generación distribuida tiene innumerables aplicaciones, dentro de las cuales se pueden mencionar la generación en forma continua, su operación en

paralelo con la red de distribución nacional, como soporte a la red de distribución nacional cuando existen altas demandas en diversas épocas del año o por fallas en la red. Este tipo de generación incluye varios beneficios, tanto para los usuarios como para los suministradores. Entre los beneficios para los usuarios se pueden citar: “la reducción del número de interrupciones, un menor costo de la energía, el uso de energías renovables y una disminución en la emisión de gases contaminantes a la atmósfera” [1]. Las ventajas para el suministrador que en este caso es la Unión Nacional Eléctrica (UNE) son muchas, entre las cuales se destacan: “la reducción de pérdidas en transmisión y distribución, el abasto energético a zonas remotas, una menor saturación del sistema y una rebaja del índice de fallas”[1]. Los Grupos Electrógenos (GE) forman parte o se encuentran dentro de lo que conocemos como generación distribuida. Los GE son: **equipos formados por un motor primario de combustión interna y un generador sincrónico de corriente alterna acoplado mecánicamente para producir energía eléctrica. Para su funcionamiento puede consumir fuel, diésel o gas natural. Estos dispositivos pueden estar sincronizados al Sistema Electro-Energético Nacional (SEN) para solucionar los déficit de potencia y contingencias, aislados (pertenecientes a la Unión Eléctrica UNE) para suministrar energía eléctrica en lugares donde no llegue la red eléctrica nacional o de emergencia, ubicados en una entidad para operar en caso de fallo, desconexión o insuficiencias del fluido eléctrico en la red nacional [1].**

Los Grupos Electrógenos Diésel (GED) constituyen un importante componente en el actual Sistema Electroenergético Nacional (SEN). Para la explotación de estos sistemas se utilizan varios productos y servicios energéticos. El encarecimiento que los mismos han sufrido actualmente, ha convertido la gestión de los GED en una preocupación para los técnicos y económicos de la Unión Nacional Eléctrica (UNE). Después de una reflexión enmarcada en la gran utilidad e importancia que tienen los sistemas distribuidos, en particular, los GED, se reveló la necesidad de adoptar nuevas técnicas dirigidas a incrementar su rendimiento y mejorar su funcionamiento mediante un análisis detallado del comportamiento de los mismos durante un período de tiempo. Teniendo en cuenta que se tiene una gran cantidad de

información de los mismos, se ha decidido usar el Manager de Mysql ya que es un sistema de gestión de Base de Datos relacional, multihilo y multiusuario donde existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, Pascal, Delpis y Java, cada uno de estos utiliza una APIs específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos, además realizar una aplicación de escritorio en Java para el manejo de los datos y facilitar la toma de decisiones. La acumulación incesante de datos en todas las áreas del quehacer humano, requiere de métodos automatizados para la gestión de la información existente. De aquí se puede inferir como **problema científico** la inexistencia de una Base de Datos de los indicadores técnicos-productivos que facilite procesos eficientes de toma de decisiones para la explotación de los Grupos Electrógenos Diésel en Batería de Moa en las condiciones reales de trabajo.

Razón por la cual el **objeto de la investigación** se enmarca en la Bases de Datos de los Grupos Electrógenos Diésel en Batería de Moa.

El **campo de acción** es el desarrollo e implementación de una Base de Datos para la explotación de los Grupos Electrógenos Diésel en Batería de Moa.

Para dar solución al problema en cuestión, se tomó como **objetivo general** de la investigación establecer una Base de Datos de los indicadores técnicos productivos que facilite procesos eficientes de toma de decisiones para la explotación de los grupos Electrógenos Diésel de Batería de Moa en las condiciones reales de trabajo.

Como guía para la investigación se plantea la siguiente **hipótesis**: Una Base de Datos de los indicadores técnico-productivos de los GED en Batería de Moa, donde se analicen las regularidades de los indicadores técnicos-productivos y dichas regularidades se muestran mediante una aplicación de escritorio, entonces se facilitará el proceso eficiente de toma de decisiones para la

explotación de los Grupos Electrónicos Diésel en Batería de Moa en las condiciones reales de trabajo.

De acuerdo con esta propuesta se derivan los siguientes **objetivos específicos** de la investigación:

1. Estudiar el marco teórico de los Grupos Electrónicos Diésel en Batería de Moa en las condiciones reales de trabajo.
2. Análisis y descripción de la base de datos en explotación de los indicadores técnicos-productivos del municipio de Moa.
3. Elaborar y actualizar la Base de Datos de los indicadores técnicos-productivos del municipio de Moa.
4. Desarrollar un Software que cumpla con todos los estándares establecidos.

Para lograr el cumplimiento de los objetivos trazados fue necesario plantearse el siguiente sistema de **tareas**:

1. Establecer las tendencias y problemas encontrados en los anteriores sistemas informáticos utilizados en la explotación de los grupos electrónicos.
2. Crear la base de datos de los indicadores técnicos-productivos de los Grupos Electrónicos Diésel en Batería de los circuitos residenciales del municipio de Moa.
3. Proponer un modelo digital para la toma de decisiones de los Grupos Electrónicos Diésel en Batería de Moa.
4. Estudiar las herramientas que se utilizarán en el proceso de creación del Software.
5. Realización de la arquitectura, diseño e implementación del Software.
6. Realización del estudio de factibilidad.

Estructura de la Tesis.

El presente Trabajo consta con cuatro capítulos:

En el **Capítulo I** se realiza una descripción del proceso de toma de decisiones para la explotación en la Batería de Grupos Electrónicos Diésel de Moa. Se

establecen la metodología y las herramientas a usar. Además se documenta y desarrolla la Base de Datos de los indicadores técnicos-productivos del municipio.

En el **Capítulo 2, Planeación y Diseño**, se expone la descripción sobre las principales funcionalidades del sistema, las estimaciones de esfuerzo de las historias de usuarios y el plan de iteraciones necesario para su terminación.

En el **Capítulo 3, Desarrollo y Pruebas**, se describen en detalles los flujos de trabajos relacionados a las etapas de diseño e implementación de la metodología utilizada, XP.

En el **Capítulo 4, Estudio de Factibilidad**, se realiza un estudio de los beneficios y costos del desarrollo del proyecto, y se valora la sostenibilidad del producto.

Para terminar, se muestran las **Conclusiones** a las cuales se arriba, las **Recomendaciones** que se proponen, la **Bibliografía** utilizada y los **Anexos** con la información necesaria sobre el trabajo.

Capítulo I Fundamentación Teórica

1.1 Introducción

En este capítulo abordaremos los conceptos fundamentales para el desarrollo del proceso de toma de decisiones así como el funcionamiento de los Grupos Electrónicos Diésel (GED), sus principales características, así como las herramientas que se pueden utilizar para obtener un buen producto. Haremos un estudio de los diferentes Gestores de Base de Datos, lenguajes de programación y metodologías existentes para determinar cuáles son más convenientes.

1.2 Estado del arte

El uso racional y eficiente de la energía en los Sistemas de Generación Distribuida (GD), ha sido un tema intensamente discutido dentro del marco de la Revolución Energética. Entre dichos sistemas se encuentran los Grupos Electrónicos Diésel (GED) conectados a la red de distribución nacional. La utilización de los grupos en el municipio ha servido para garantizar el suministro de energía a los consumidores residenciales y económicos. Se ha decidido a partir de los grandes volúmenes de información que se tiene desde el comienzo de la utilización de los grupos, crear un modelo que permita mejorar la eficiencia energética y determinar las causas que puedan inducir altos costos de generación. Este estudio está dirigido fundamentalmente a la obtención de mejoras en la explotación y el funcionamiento de los grupos y el perfeccionamiento a la hora de la toma de decisiones, ya que muchas veces, en estos grupos no es sencillo obtener información valiosa de la cantidad colosal de datos que tienen almacenados, a pesar de intuir que el conocimiento que se podría extraer de ellos sería de gran ayuda en muchas de las áreas y facetas en que se desenvuelven (toma de decisiones, mejoras en la producción, etc).

1.2.1 Antecedentes

En la actualidad existen algunos sistemas automatizados relacionados con la

toma de decisiones como son:

- El Sistema de Supervisión y Control de Procesos (Eros) [13], es un sistema de adquisición de datos y supervisión (SCADA_ Supervision Control and Data Adquisition) que realiza un potente tratamiento estadístico y determinístico de las variables medidas, trabaja acoplado con diversos sistemas de colección de datos y se comunica con los dispositivos de medición mediante manejadores de comunicación (drivers), posee alarmas que notifican las fallas en la explotación y mantenimiento garantizando la seguridad de la toma de dato al operar individual los grupos o la instalación (emplazamiento o batería) en modo de manual o automático. Este SCADA permite al usuario entrar datos de mediciones opcionales, tal como el índice de consumo, de aceite, de líquido refrigerante, consumo de combustible del motor para mejorar el estimado de la eficiencia. El histórico que almacena los parámetros de explotación esta limitado por la capacidad de la memoria de la computadora por tanto, en la mayoría de los estudios no se tiene acceso a todos los datos necesarios y los datos almacenados no permiten un análisis detallado de la eficiencia de los grupos.
- Anterior a este SCADA se tenía el MVision [12][14] con características muy parecidas, censaba más parámetros, pero a diferencia de este no guardaba un histórico por lo que no era factible para el estudio de la eficiencia.
- El software Power System eXplorer (PSX) [15] es notable por su flexibilidad, baja invasividad y fácil uso. Este software se utiliza para estudios de regímenes, parte de los datos de chapa y operación de los elementos eléctricos de la instalación y datos de la carga instalada en cada hora para estimar la capacidad de la generación de los sistemas de GD en los regímenes paralelo e ISLA. Esta potente herramienta ha sido utilizada por los investigadores cubanos Abreu y col [10] y Castro y col [11] para determinar el comportamiento de la generación de los GE ante catástrofes o indisponibilidad del SEN.

1.3 Lenguajes de Programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. A continuación se realiza un estudio de los principales lenguajes de programación existentes que se utilizan para la creación de aplicaciones de escritorio, como son Delphi, C++, Java, entre otros.

Delphi

Es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal. Es producido comercialmente por la empresa estadounidense CodeGear adquirida en Mayo de 2008 por Embarcadero Technologies, una empresa del grupo Thoma Cressey Bravo, en una suma que ronda los 30 millones de dólares. En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET.

C++.

Es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C. El nombre C++ fue propuesto por Rick Masciatti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". Se puede decir que abarca tres paradigmas de la programación:

- La programación estructurada.
- La programación genérica.
- La programación orientada a objetos.[7]

Java.

Es un lenguaje de programación desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Las características principales que nos ofrece Java son:

- **Simple:** Ofrece toda la funcionalidad de un lenguaje potente. Debido a que C y C++ son los lenguajes más difundidos, Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. Elimina muchas de las características de otros lenguajes para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria.
- **Orientado a objetos:** Con el objetivo de mantener la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Sopor talas tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, necesitan ser construidas y destruidas en espacios de memoria. Incorpora funcionalidades como por ejemplo, la resolución dinámica de métodos mediante una interfaz específica llamada RTTI (RunTime Type Identification) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el runtime

que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

- **Distribuido:** Se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. Proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.
- **Robusto:** Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java.
- **Arquitectura neutral:** Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris 2.x, SunOs 4.1.x, Windows 95, Windows NT, Linux, Irix, Aix, Mac, Apple y probablemente haya grupos de desarrollo trabajando en el porting a otras plataformas.
- **Seguro:** La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o el casting se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa Java para crear un

navegador, se combinan las características del lenguaje con protecciones de sentido común aplicadas al propio navegador. El código Java pasa muchos tests antes de ejecutarse en una máquina. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto. El Cargador de Clases también ayuda a Java a mantener su seguridad, separando el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior. Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y luego en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida.

- **Multihebra:** Soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.[3]

1.4 Arquitectura

La Arquitectura es el esqueleto o base de una aplicación. Representa la organización fundamental de un sistema. Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". En la Web es muy común la utilización de la arquitectura "3-capas", "n-capas", "MVC", entre otras. El lenguaje de programación JAVA, seleccionado anteriormente, implementa a su

vez el patrón arquitectónico MVC, es por ello que adoptamos esta arquitectura para el desarrollo de la propuesta de solución. Modelo Vista Controlador (MVC). Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **Vista:** Presenta el modelo en un formato adecuado, como en una página Web que le permite al usuario interactuar con ella, usualmente un elemento de interfaz de usuario.
- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

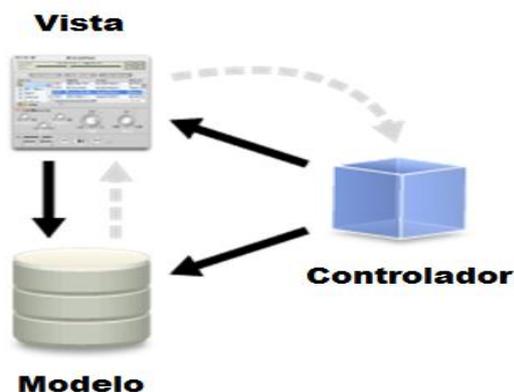


Figura # 1 Elementos del patrón MVC.

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

1.4.1 Ventajas del Modelo Vista Controlador

- La separación del Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Permite el escalamiento de la aplicación en caso de ser requerido.

1.5 Sistemas Gestores de bases de datos (SGBD).

MySQL.

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, Pascal, Delpis, Java cada uno de estos utiliza una APIs específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos. [5]

PostgreSQL.

Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Mediante un sistema denominado MVCC (Acceso concurrente multiversión por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel

mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. Está considerado como el gestor de base de datos de software libre más avanzado del mundo. Es un sistema objeto – relacional pues incluye aspectos del paradigma orientada a objetos, tales como la herencia, tipos de datos, funciones, restricciones, reglas e integridad transaccional, aunque no llega a ser un gestor con orientación a objetos pura.[6]

1.6 Metodologías para el desarrollo de software

Hoy en día, llevar a cabo el desarrollo de un buen software depende de un gran número de actividades y etapas donde elegir la mejor metodología para el equipo influye directamente en el futuro éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo. Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos. Las metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: las metodologías tradicionales (RUP, MSF) y las metodologías ágiles (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente. (Zulueta ,2010)

RUP

Es un proceso para el desarrollo de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como tres características esenciales está dirigido por casos de uso: que orientan al proyecto a la importancia para el usuario y lo que se quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido

el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini-proyectos donde los casos de uso y al arquitectura cumplen sus objetivos de manera depurada. RUP propone cuatro etapas para el desarrollo de un producto: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería per o con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros. El Proceso Unificado de Desarrollo tiene 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. Flujos de trabajo:

- Modelamiento del negocio
- Requerimientos
- Análisis y diseño
- Implementación
- Prueba (Testeo)
- Instalación
- Administración del proyecto
- Administración de configuración y cambios
- Ambiente

XP

La calidad en el desarrollo y mantenimiento del software, se ha convertido, hoy en día, en uno de los principales objetivos estratégicos de las organizaciones. Esto se debe a que, cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento.

En los últimos años, se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para el desarrollo de un proceso de software. De esta forma, se han definido varias metodologías para

llevar a cabo este proceso; algunas ágiles, entre las que se encuentra SCRUM y XP, y otras, denominadas tradicionales, como RUP.

En este contexto, se debe destacar que una metodología es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos. Por ello, seleccionar la más adecuada, constituye un paso importante para el éxito de cualquier proyecto.

En este sentido, se decide utilizar la metodología Programación Extrema (*eXtreme Programming* ó XP, por sus siglas en inglés). La misma, es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado, es ideal para grupos de desarrollo relativamente pequeños y donde el tiempo disponible para la entrega del proyecto es corto; como es el caso de la presente investigación.

Ventajas

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, ya conocen las fechas de entrega de funcionalidades. Vital para su negocio.
- Permite definir en cada iteración cuales son los objetivos de la siguiente.
- Permite la retroalimentación.
- La presión está a lo largo de todo el proyecto y no en una entrega final.

Desventajas

- Delimitar el alcance del proyecto con nuestro cliente.

Para mitigar esta desventaja se plantea definir un alcance a alto nivel basado en la experiencia

Esta metodología, propone seguir las siguientes fases.[4]

Fases de la Metodología XP



1.7 Propuesta de solución

Se desarrollará una aplicación de escritorio utilizando como lenguaje de programación Java, la cual guardará sus datos en una base de Datos de MySQL y se utilizará la metodología XP (Programación Extrema) para llevar a cabo el proceso de desarrollo del software.

1.8 Herramientas y metodología a emplear en la propuesta de solución.

1.8.1 MySQL como Sistema Gestor de Base de Datos

Los sistemas de gestión de bases de datos (*Data Base Manager Systems* DBMSs, por sus siglas en inglés) permiten almacenar, visualizar y modificar datos, así como hacer copias de seguridad y mantener la integridad de los mismos. Además, proporcionan una serie de funciones que facilitan el desarrollo de nuevas aplicaciones. Existen una gran variedad de sistemas Gestores de Base de Datos: SQL, MySQL, Oracle, PostgreSQL, etc. En general todos desempeñan la misma función pero tienen diferentes propiedades y herramientas que los distingue claramente unos de otros.

En este proyecto, se decide utilizar el Gestor de Base de Datos MySQL. El cual es un sistema de administración de bases de datos relacional, licenciado bajo la Licencia Pública General de GNU².

Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de

organización. Por lo que para una mejor gestión de la base de datos, se puede utilizar algún cliente para la administración y mantenimiento de las mismas.

1.8.2 EMS MySQL Manager como Cliente para la Administración y Mantenimiento del Sistema Gestor de Base de Datos MySQL.

Para completar con las herramientas, se decide utilizar *EMS MySQL Manager* como cliente para el gestor de bases de datos seleccionado. El mismo, se considera muy potente y efectivo para administrar y mantener este tipo de servidor. Además, posee características que fundamentan su utilización tales como:

- Interfaz gráfica de usuario que posibilita la creación/edición de bases de datos de una manera simple y sencilla.
- Soporta scripts SQL.
- Permite hacer construcciones visuales de consulta SQL.
- Extrae o imprime metadatos.
- Exporta/importa datos.
- Permite el mantenimiento y la gestión de privilegios de usuarios.

1.8.3 ¿Por qué XP como metodología?

XP y RUP son dos grandes metodologías que después de analizar sus principales características y los aspectos más sobresalientes de cada una de ellas, se ha determinado la implantación de XP, una metodología ligera, con menos requerimientos de documentación y planificación para el desarrollo de la aplicación. Si se escoge RUP traería grandes dificultades como son:

Multitud de artefactos: El hecho de realizar varios artefactos y mantenerlos actualizados consume mucho tiempo.

El poco personal de desarrollo: Al ser solo una persona a cargo del desarrollo de la aplicación, este tomaría varios roles en cada etapa y sería muy complejo cumplir con las actividades de cada uno de ellos. En futuras versiones: Se piensa añadirle otras funcionalidades, destacando que se desea lograr una funcionalidad mínima del software lo más pronto posible.

Se necesita: La documentación mínima necesaria para el futuro soporte y mantenimiento del producto final.

Requisitos cambiantes: Los cambios en un proceso de desarrollo son inevitables, al aparecer un nuevo requisito hace que se tenga que comenzar una nueva iteración para dar cumplimiento a su funcionalidad. Como el proyecto está en plena investigación y en un ambiente de desarrollo sujeto a cambios repentinos se sugieren una gran adaptabilidad y pronta respuesta, lo cual RUP no ofrece.

Planificación inexistente: La planificación que se realiza en las fases iniciales está sujeta a muchas variaciones en dependencia de los cambios que se experimenten en los requisitos. Por tanto se hace muy difícil planificar actividades específicas si no se tiene claro que se debe hacer realmente.

Los inconvenientes planteados pueden ser eliminados con la utilización de la metodología XP.

1.8.4 ¿Por qué Java?

Atendiendo las características de los lenguajes antes mencionados hemos seleccionado Java para el desarrollo de nuestra aplicación por contar con las siguientes características:

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- ✓ Aritmética de punteros.
- ✓ No existen referencias.
- ✓ Registros (struct).
- ✓ Definición de tipos (typedef).
- ✓ Macros (#define).
- ✓ Necesidad de liberar memoria (free).

Aunque, en realidad, lo que hace es eliminar las palabras reservadas (struct, typedef), ya que las clases son algo parecido. Tiene ventajas en cuanto a seguridad ya que al tener la característica de ser interpretado el código Java pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o el caso de un objeto. El Cargador de Clases, separando el espacio de nombres del sistema de ficheros local, del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior.

1.8.5 Ambiente de Desarrollo Integrado (IDE) elegido: NetBeans 7.0.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos. La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. El IDE NetBeans 7.0 es una herramienta para programadores, pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, JavaEE, web, EJB y aplicaciones móviles). Entre sus características se encuentran un sistema de proyectos basado en Ant, control de versiones y refactoring.

El IDE NetBeans IDE 6.0 corre en diferentes sistemas operativos entre los que podemos mencionar linux, Mac OS X, Solaris y Windows. Como se puede observar se ha lanzado con soporte para la mayoría de las plataformas. Como requerimiento se necesita tener previamente instalado el JDK 5.0 o 6.0.

1.8.6 Visual Paradigm.

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Está dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, lo cual garantiza la calidad del producto final. Permite modelar todos los tipos de

diagramas de clases, generar código desde diagramas, código inverso y generar documentación. Se prefiere su utilización en el desarrollo del módulo por ser una herramienta no propietaria, multiplataforma, amigable en su entorno, lo que facilita su interoperabilidad con otras aplicaciones además de que tiene un uso de un lenguaje estándar común a todo el equipo de desarrollo lo que facilita la comunicación.

Principales Características:

- Producto de calidad.
- Soporta diseño de aplicaciones Web.
- Soporta varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Licencia gratuita.
- Compatibilidad entre ediciones.

1.8.7 Embarcadero ER/Studio.

Simple y fácil al usuario, ayuda a las organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. [8]

Conclusiones del Capítulo

En este capítulo se abordaron los conceptos fundamentales asociados al dominio del problema, relacionados con el objeto de estudio y el campo de acción. Además se realizó un estudio de lo más utilizado en cuanto a los múltiples lenguajes, metodologías y tecnologías para el desarrollo de aplicaciones existentes, escogiendo así lo que presenta mayor ventaja con respecto a las características de nuestro sistema. Entre lo seleccionado se encuentra Java como lenguaje de programación y MySQL como gestor de Base de Datos; además, se decide realizar la aplicación sobre la base de la metodología ágil XP, pues nos permitirá obtener resultados funcionales observables a corto plazo.

Capítulo II Planeación y Diseño

2.1 Introducción

En este capítulo se expondrán las necesidades del cliente así como las principales funcionalidades que debe cumplir el sistema llevadas a cabo por las historias de usuarios, y se mostrarán también una estimación de esfuerzo para cada una de ellas. Se establecerá un plan de iteraciones y además se presentarán las tarjetas CRC la cuales son un paradigma de la programación orientada a objetos, que permitirá trabajar con objetos.

2.2 Planificación

Lista de reserva:

Después de conocer el personal relacionado e identificar los requisitos generales, se procede a realizar el análisis de las funcionalidades que debe cumplir la aplicación para dar respuesta a los mismos. Para ello se numerarán mediante una lista de reserva, las funcionalidades que el sistema debe ser capaz de cumplir.

1. Autenticar Usuario
2. Gestionar Usuario
 - 2.1 Insertar usuario
 - 2.2 Modificar usuario
 - 2.3 Eliminar usuario
3. Mostrar Generación Neta
4. Mostrar Generación Bruta
5. Mostrar Densidad
6. Mostrar Índice de Consumo
7. Mostrar consumo de Ventilador
8. Insertar Escritura
9. Mostrar Escritura
10. Mostrar Reporte

2.2.1 Personal relacionado con el sistema

Personas relacionadas con el sistema	Justificación
Especialista	Esta es la persona que tiene conocimiento en la toma de decisiones.
Desarrollador	Es la persona responsable de llevar a cabo la implementación del sistema.

Tabla 2.1 Personas relacionadas con el sistema.

Historias de usuarios

Las HU es una técnica que utiliza XP para detallar los requisitos que debe cumplir el sistema, requisitos que surgen a partir de las reuniones diarias entre el cliente y el desarrollador donde expresan sus ideas para el desarrollo del software.

A continuación se muestra una de las HU descritas por el cliente, las demás se encuentran en el [ANEXO 1: HISTORIAS DE USUARIO]

HU No.1.1: Autenticar usuario

Historia de usuario	
Número:1	Usuario: Especialista
Nombre: Autenticar usuario	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados:3	Iteración asignada:1
Programador responsable: Angel Cutiño Guilarte	
Descripción: Este sistema deberá autenticar a los usuarios cuando deseen pasar a modo administrativo o técnico.	
Observaciones:	

Tabla 2.2 HU No.1.1: Autenticar usuario

Planificación de entregas

En esta parte se establece la prioridad de cada historia de usuario, así como una estimación del esfuerzo necesario de cada una de ellas con el fin de determinar un cronograma de entregas en conjunto con el cliente, una entrega debe realizarse en un periodo de dos a tres meses. Las estimaciones de esfuerzo asociado a la implementación de las historias se establecen utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.

Historia de usuarios	Puntos Estimados
Autenticar Usuario	1 semanas
Gestionar Usuario	1.5 semanas
Mostrar Generación Neta	0.5 semanas
Mostrar Generación Bruta	0.5 semanas
Mostrar Densidad	0.5 semanas
Mostrar Índice de Consumo	0.5 semanas
Mostrar Consumo Ventilador	0.5 semanas
Insertar Escritura	1 semanas
Mostrar Escritura	1 semanas
Mostrar Reporte	0.5 semanas

Tabla 2.11 Estimación de esfuerzo por historia de usuario

2.2.2.1 Planificación de iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración.

En este plan se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un reléase, lo cual debería ser consistente con el contenido de una o dos iteraciones.

En relación con lo antes tratado se decide realizar el sistema en 3 iteraciones, las cuales se explican de forma detallada a continuación:

Iteración # 1 Esta iteración tiene como objetivo darle cumplimiento a las funcionalidades descritas en las HU 1, y 2 las cuales están relacionadas con la autenticación y gestión de usuario.

Iteración # 2 Introducción de datos. Esta iteración tiene como finalidad darle cumplimiento a la HU 8, 9 y 10.

Iteración # 3 Esta iteración tiene como finalidad darle cumplimiento a la HU 3, 4, 5, 6 y 7.

Plan de duración de las iteraciones

En la metodología de desarrollo de software XP, se debe crear un plan de duración de las iteraciones que tenemos, y las mismas se llevarán a cabo en el desarrollo. Este plan tiene como objetivo fundamental mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas.

Plan de duración de las iteraciones

Iteración	Historia de usuario	Duración total
Iteración 1	Autenticar Usuario Gestionar Usuario	$1+1.5=2.5$
Iteración 2	Insertar Escritura Mostrar Escritura Mostrar reporte	$1+1+0.5=2.5$
Iteración 3	Mostrar Generación Neta Mostrar Generación Bruta Mostrar Densidad Mostrar Índice de Consumo Mostrar consumo de Ventilador	$0.5+0.5+0.5+0.5+0.5=2.5$

Tabla 2.12 Plan de duración de las iteraciones

2.3 Diseño

2.3.1 Clases, Responsabilidades y Colaboración

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como tarjetas CRC, las cuáles se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño.

Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente.

Nota: En el capítulo solo ha sido representada la tarjeta **CRC No.1 Cliente** las demás están reflejadas en el [ANEXO 2: TARJETAS CRC].

Tarjeta CRC No.1 Grupo

grupo	
Description: guardar información de los grupos	
Attributes:	
Name	Description
id_grupo	identificar el grupo
grupo	
Responsibilities:	
Name	Collaborator
crear_grupo	
modificar_grupo	
mostrar_grupo	

Tabla 2.13 Tarjeta CRC Grupo

2.4 Conclusiones

En este capítulo de planeación y diseño se han expuesto las HU conjunto con al cliente a través de plantillas. Se realizó un plan de iteraciones para cada una de estas historias aplicando una estimación de esfuerzo de las mismas. Se presentaron además las principales clases que se utilizarán en el desarrollo de la aplicación a través de las llamadas tarjetas CRC, terminado de esta forma esta fase para pasar a la siguiente fase de desarrollo y pruebas.

Capítulo 3: Implementación y Pruebas.

3.1 Introducción

En este capítulo se siguen las fases de Desarrollo y pruebas conformadas en la Metodología XP. Se muestra un modelo de datos correspondiente a la aplicación concluyente, así como las interfaces de usuarios diseñadas para la naciente aplicación. Se realiza a través del desarrollo de las iteraciones un desglose de las tareas a realizar para cada una de las HU expuestas en el capítulo anterior. Por último se describen las pruebas realizadas, mostrando las diferentes respuestas al aplicarse cada una de las funcionalidades.

3.2 Implementación

Modelo de datos

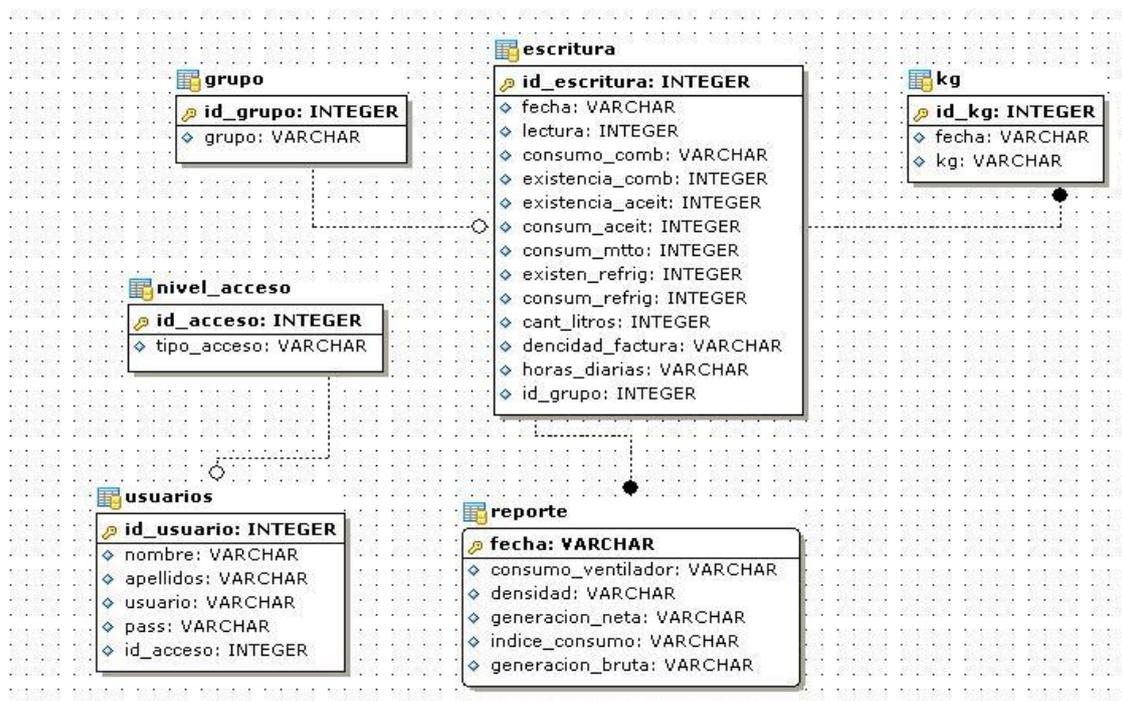


Figura 3.1 Modelo de datos del Sistema

Interfaces de Usuario

En el capítulo, solo ha sido representada la interfaz de usuario correspondiente a la portada del sistema las demás están reflejadas en el [ANEXO 4: INTERFAZ DE USUARIO].

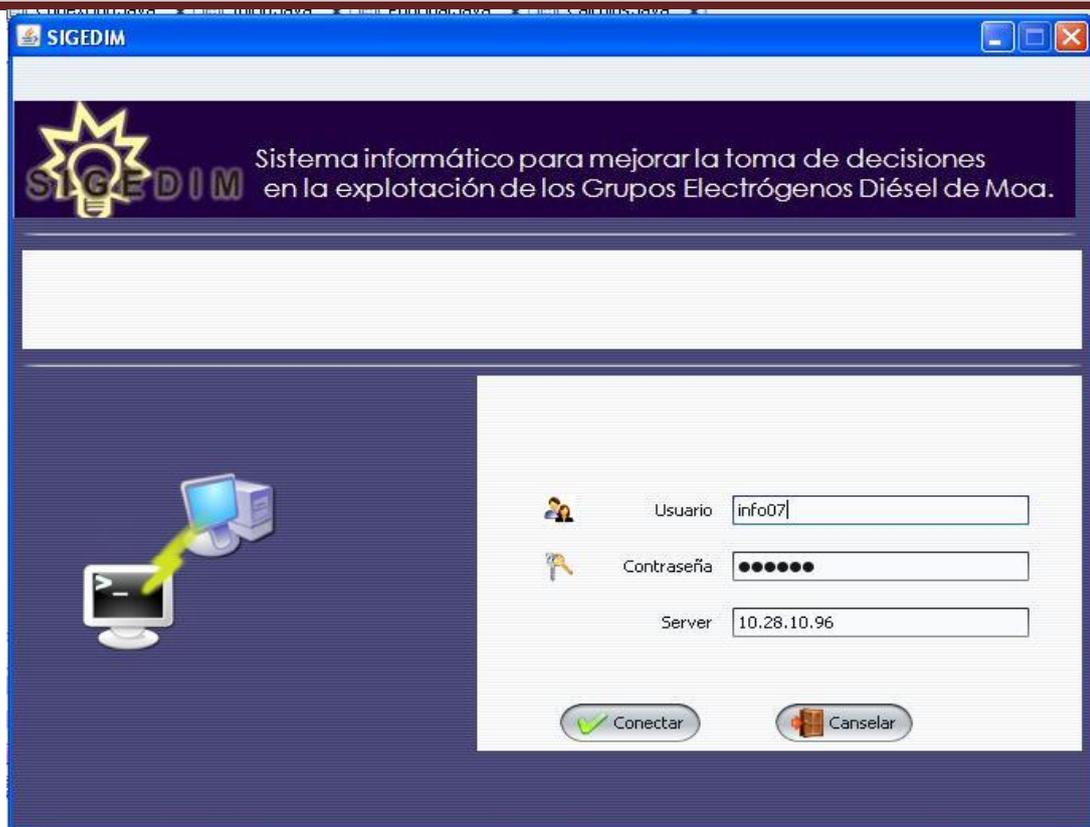


Figura 3.2 Acceso del usuario.

3.2.1 Desarrollo de las iteraciones

En el capítulo anterior se detallaron todas las historias de usuarios de cada una de las iteraciones en la que se va a desarrollar el sistema, y para la realización de cada una de las HU se realizará una revisión al plan de iteraciones, para ver si pueden ocurrir cambios.

3.2.2 Tareas por Historias de Usuarios

Las HU se dividen en tareas de programación o ingeniería que son asignadas al equipo de desarrollo para su implementación, tareas que solo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Las mismas se representan mediante las siguientes tarjetas de tareas:

Historia de usuarios	Tareas
Autenticar Usuario	Autenticar Usuario
Gestionar Usuarios	Insertar usuario

	Modificar usuario Eliminar Usuario
Mostrar Generación Neta	Mostrar Generación Neta
Mostrar Generación Bruta	Mostrar Generación Bruta
Mostrar Densidad	Mostrar Densidad
Mostrar índice de Consumo	Mostrar índice de Consumo
Mostrar Consumo Ventilador	Mostrar Consumo Ventilador
Insertar Escritura	Insertar Escritura
Mostrar Escritura	Mostrar Escritura
Mostrar Reporte	Mostrar Reporte

Tabla 3.1 Distribución de tareas por historia de usuario

Historias de usuario abordadas en la primera iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Autenticar usuarios	1	1
Gestionar usuarios	1.5	1.5

Tabla 3.2 Historias abordadas en la primera iteración

Historias de usuario abordadas en la segunda iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Insertar Escritura	1	1
Mostrar Escritura	1	1
Mostrar Reporte	0.5	0.5

Tabla 3.3 Historias abordadas en la segunda iteración

Historias de usuario abordadas en la tercera iteración

Historias de usuario	Tiempo de estimación (semanas)
----------------------	--------------------------------

	Estimación inicial	Real
Mostrar Generación Neta	0.5	0.5
Mostrar Generación Bruta	0.5	0.5
Mostrar Densidad	0.5	0.5
Mostrar Índice de Consumo	0.5	0.5
Mostrar Consumo Ventilador	0.5	0.5

Tabla 3.4 Historias abordadas en la tercera iteración

A continuación se expondrán representadas por tablas una de las diferentes tareas de las historias de usuarios definidas por el cliente en la seis iteraciones, las demás tareas se podrán ver [ANEXO 5: TARJETAS DE TAREAS].

Tarjeta de tarea No.1: Autenticar usuarios

Tarea ingeniería	
Número tarea: 1	Número historia: 1
Nombre tarea: Autenticar usuarios	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 1/02/2012	Fecha fin: 7/02/2012
Programador responsable: Angel Cutiño Guilarte	
Descripción: Esta tarea facilita que los usuarios al entrar al sistema sean auténticos.	

Tabla 3.5 Tarjeta de tarea No. 1 Autenticar usuarios

3.3 Pruebas

Una de las mejores características de la metodología XP es el proceso de pruebas. Esta metodología propone probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección.

También permite aumentar la seguridad de evitar efectos no deseados a la hora de realizar modificaciones y refactorizaciones. XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente.

3.3.1 Pruebas de aceptación

En esta plantilla se precisan las pruebas realizadas según la historia de usuario seleccionada para realizar la comprobación y validar las funcionalidades del sistema, y de esta forma saber si está apto para ser liberado.

A continuación se presentan una de las pruebas llevadas a cabo para verificar el buen funcionamiento del sistema, las demás plantillas verlas en el **[ANEXO 5: PRUEBAS DE ACEPTACIÓN]**.

Pruebas de aceptación para la HU: Autenticar usuarios

Prueba de aceptación
HU: Autenticar usuarios
Nombre: Prueba para comprobar la autenticación de un usuario.
Descripción: El usuario se autentica para entrar al sistema.
Condiciones de ejecución: Se necesita el usuario y contraseña.
Entrada/Pasos ejecución: El usuario intenta introducir usuario y contraseña.
Resultado: Se emite un mensaje de error en caso de que no es correcta la contraseña sino el usuario estará entrando al sistema según el modo y sus privilegios.
Evaluación de la prueba: Aceptada.

Tabla 3.13 PA: Prueba para comprobar la autenticación de un usuario.

3.4 Conclusiones

Al concluir este capítulo se pudo precisar la realización de las pruebas según la historia de usuario seleccionada para consumir la comprobación y validar las funcionalidades más importantes del sistema, se logró controlar y organizar cada una de las actividades a desarrollar por el equipo de trabajo, se confeccionaron las tarjetas CRC, así como el modelo de datos y las interfaces creadas durante el desarrollo.

Capítulo 4: Estudio de Factibilidad del Proyecto.

4.1 Introducción

Para estudiar la factibilidad de este proyecto se utilizará la Metodología Costo Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo, que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados
- La efectividad, que se entiende como la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo).

4.2 Análisis Beneficio Costo ABC

4.2.1 Efectos Económicos

- Efectos directos
- Efectos indirectos
- Efectos externos
- Intangibles

4.2.1.1 Efectos directos

1. POSITIVOS:

- Permitirá compartir información entre los usuarios del sistema.
- Se cuenta con una herramienta capaz de mantener la seguridad e integridad de los datos que se procesan.
- Se gestiona y controla el acceso de los usuarios.

2. NEGATIVOS:

- Para usar la aplicación es imprescindible el uso de un ordenador, aparejado a los gastos que este trae de consumo de corriente eléctrica y mantenimiento.

4.2.1.2 Efecto indirecto

Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

4.2.1.3 Efectos externos

Se contará con una herramienta disponible que permitirá la gestión y control de la información del funcionamiento de los Grupos Electrónicos Diesel en Batería de Moa y facilitará la toma de decisiones.

4.2.1.4 Intangibles

En la estimación financiera siempre hay elementos visibles por un grupo como perjuicio o beneficio, pero al momento de darle valor en unidades monetarias esto resulta difícil o prácticamente imposible. A fin de medir con precisión los efectos, deberán considerarse dos situaciones: la situación sin Proyecto y la situación con proyecto.

Situación sin Proyecto (Solución Manual):

Los especialistas son los encargados de entrar y calcular manualmente los datos sobre el funcionamiento de los GED. De tal manera, que pueden cometer errores debido a:

- Mucha información
- Muchos cálculos

Situación con el Producto

Para la entrada de los datos al sistema propuesto debemos seguir los siguientes pasos:

- Cargar la aplicación ejecutando el ejecutable del sistema.
- Entrar los datos correspondientes del usuario que hará uso de la aplicación.
- Insertar los datos del funcionamiento de los GED.
- Mostrar un formulario con reportes de las variables más importantes.
- Graficar.

4.2.2 Beneficios y Costos Intangibles en el Proyecto

Costos:

- Resistencia al cambio

Beneficios:

- Mayor comodidad, rapidez y eficiencia de los especialistas responsables a desarrollar la tarea
- Mayor control sobre los usuarios
- Aporta a la seguridad informática del centro

4.2.2.1 Ficha de Costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar Una Ficha De Costo de un Producto Informático [Dra. Ana María Gracia Pérez, UCLV]. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costo en Moneda Libremente Convertible.

Costos Directos

1. Compra de equipos de cómputo: No procede.
2. Alquiler de equipos de cómputo: No procede.
3. Compra de licencia de Software: No procede.
4. Depreciación de equipos: \$ 60.78.
5. Materiales directos: No procede.

Total: \$ 60.78.

Costos Indirectos

1. Formación del personal que elabora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. *Know How*: No procede.
5. Gastos en representación: No procede.

Total: \$0.00.

Gastos de distribución y venta.

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

Total: \$0.00.

Costos en Moneda Nacional:

Costos Directos

1. Salario del personal que laborará en el proyecto: \$100.00.
2. El 12% del total de gastos por salarios se dedica a la seguridad social: No procede.
3. El 0.09% de salario total, por concepto de vacaciones a acumular: No procede.
4. Gasto por consumo de energía eléctrica: \$ 5.94.
5. Gastos en llamadas telefónicas: No procede.
6. Gastos administrativos: No procede.

Costos Indirectos

1. Know How: \$ 108, 75.

Total: \$ 214.69.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo- Efectividad. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado por los especialistas para llevar a cabo el proceso de gestión de información para la toma de decisiones y la variable sería la complejidad de las pruebas que se realizan durante este proceso.

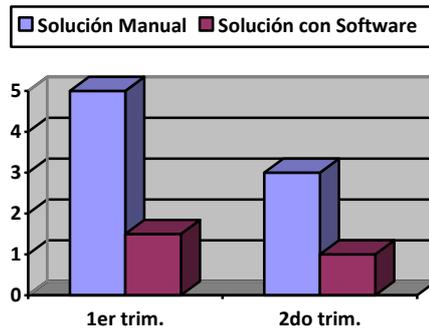
Valores de la variable (Solución manual):

- Realizar la inserción de los datos medibles en la escritura. 13 variables (5 min.).
- Obtener un reporte del comportamiento de las variables de observación. 5 variables (3 min.).

Valores de la variable (Solución con el software):

- Cargar el formulario para insertar los datos. 13 variables (1.5 min.).
- Cargar formulario con los datos necesarios para mostrar un reporte 5 variable (1 min.).

4.3 Gráfica de la solución manual y solución automatizada



Teniendo en cuenta los resultados reflejados en la gráfica queda demostrada la factibilidad del sistema el tiempo que demora la solución del mismo de forma manual y automatizada.

4.4 Conclusiones

En este capítulo se realizó el estudio de factibilidad mediante La Metodología Costo Efectividad (Beneficio), se analizó los efectos económicos, los beneficios y costos intangibles, así como se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$ 60.78 CUC. y \$ 214.69 MN demostrándose la factibilidad del proyecto.

Conclusiones Generales.

- Se estudió cómo se lleva a cabo el proceso de toma de decisiones en los Grupos Electrónicos Diésel en batería de Moa, lo que justificó el análisis, diseño e implementación del (SIGEDIM).

- Se llevó a cabo un estudio de las principales metodologías, lenguajes y herramientas que se consideraron factibles para el desarrollo del sistema.

Como resultado de la investigación se logró desarrollar un software en el que se da cumplimiento a las especificidades de los objetivos propuestos. Para simplificar la demora y errores que produce este proceso manual y elevar la eficacia del mismo.

Recomendaciones.

- El sistema y la BD implementada (MySql) se utilicen en la toma de decisiones de los Grupos Electrónicos Diesel en Batería de Moa.
- Se cree un módulo para los otros Grupos Electrónicos feul.oil.
- Se de un curso de capacitación a los especialistas y operarios que utilizan esta aplicación.

Referencias Bibliográficas.

- [1] – Generación Distribuida energía de calidad, 009. Disponible en:
[http:// www.conae.gob.mx](http://www.conae.gob.mx)
- [2] – Noa Ramírez, Aliniuska. Operación del Sistema de Generación Distribuida de Moa para situaciones excepcionales. ISMM, 2009.
- [3] Java. [en línea], 2004. [Consultado: 2012-03-20]. Disponible en:
<http://wikipedia.uo.edu.cu/es/articles/j/a/v/Java.html>.
- [4] Metodología XP. [en línea], 2008. [Consultado: 2012-01-15]. Disponible en:
http://www.metodologiaxpvsmetodologiarup.blogspot.com/2008/04/caracteristicas-de-la-metodologia-xp_25.html.
- [5] MySQL. [en línea], 2008. [Consultado: 2012-03-15]. Disponible en:
<http://wikipedia.uo.edu.cu/mysql>.
- [6] PostgreSQL. [en línea], 2006. [Consultado: 2012-03-15]. Disponible en:
http://wikipedia.uo.edu.cu/es/articles/p/o/s/PostgreSQL_399b.html.
- [7] C++. [en línea], 2004. [Consultado: 2012-03-20]. Disponible en:
<http://wikipedia.uo.edu.cu/es/articles/c/+/+/C++.html>.
- [8] **Bureaudeprensa.com. 2008.** Embarcadero ER/Studio. [En línea] 2008.
http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=1153755975
- [9] Delphi. [en línea], 2004. [Consultado: 2012-03-20]. Disponible en:
<http://wikipedia.uo.edu.cu/es/articles/d/e/l/Delphi.html>.
- [10] García Abreu, M., Pérez Saintgermes, J.C. Estudio y simulación de Islas utilizando el Power System eXplore. CIE'2007. 2007.
- [11] Castro, N., Money, M. R., Leizan, I.B. Posibilidades de explotación de los grupos electrógenos en la subestación Santiago Industrial. FIE'2008. Santiago de Cuba. Cuba. 2008.
- [12] Configuration Manual. M-Vision designer. DEIF A/S Frisenborgvej. 2006.
- [13] Manual de operación. Grupos de Desarrollo EROS. Nícaro. Cuba 2006.
- [14] Operator's Manual. DEIF A/S Frisenborgvej. 2006.
- [15] Manual para las aplicaciones docentes del PSX. Universidad Central de Las Villas. 2008.

Anexos

Anexo 1 Historias de Usuarios.

HU No.1.1: Autenticar usuarios

Historia de usuarios	
Número: 1	Usuario: Especialista
Nombre: Autenticar Usuario	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Angel Cutiño Guilarte	
Descripción: Este sistema deberá autenticar a los usuarios cuando deseen pasar a modo administrativo o Especialista.	
Observaciones:	

Tabla 2.2 HU No.1.1: Autenticar usuarios

HU No.1.2: Gestionar usuario

Historia de usuarios	
Número: 2	Usuario: Especialista
Nombre: Gestionar Usuario	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá insertar, modificar y eliminar	
Observaciones:	

Tabla 2.3 HU No.1.2: Gestionar usuarios

HU No.1.3: Mostrar Generación Neta

Historia de usuarios	
Número: 3	Usuario: Especialista
Nombre: Mostrar Generación Neta	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá mostrar la Generación Neta	
Observaciones:	

Tabla 2.4 HU No.1.3: Mostrar Generación Neta

HU No.1.4: Mostrar Generación Bruta

Historia de usuarios	
Número: 4	Usuario: Especialista
Nombre: Mostrar Generación Bruta	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá mostrar la Generación Bruta	
Observaciones:	

Tabla 2.5 HU No.1.4: Mostrar Generación Bruta

HU No.1.5: Mostrar Densidad

Historia de usuarios	
Número: 5	Usuario: Especialista
Nombre: Mostrar densidad	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá mostrar la Generación Bruta	
Observaciones:	

Tabla 2.6 HU No.1.5: Mostrar Densidad

HU No.1.6: Mostrar Índice de Consumo

Historia de usuarios	
Número: 6	Usuario: Especialista
Nombre: Mostrar Índice de Consumo	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá mostrar el Índice de Consumo	
Observaciones:	

Tabla 2.7 HU No.1.6: Mostrar Índice de Consumo

HU No.1.7: Mostrar Consumo de Ventilador

Historia de usuarios	
Número: 7	Usuario: Especialista
Nombre: Mostrar Consumo de Ventilador	

Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá mostrar el consumo de Ventilador	
Observaciones:	

Tabla 2.8 HU No.1.7: Mostrar Consumo de Ventilador

HU No.1.8: Insertar Escritura

Historia de usuarios	
Número: 8	Usuario: Especialista
Nombre: Insertar Escritura	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá insertar la Escritura	
Observaciones:	

Tabla 2.9 HU No.1.8: Insertar Escritura

HU No.1.9: Mostrar Escritura

Historia de usuarios	
Número: 9	Usuario: Especialista
Nombre: Mostrar Escritura	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 2

Programador responsable: Angel Cutiño Guilarte
Descripción: El especialista podrá Mostrar la Escritura
Observaciones:

Tabla 2.10 HU No.1.9: Mostrar Escritura

HU No.1.10: Mostrar Reporte

Historia de usuarios	
Número: 10	Usuario: Especialista
Nombre: Mostrar Reporte	
Prioridad en el negocio: Alta	Riesgo del desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Angel Cutiño Guilarte	
Descripción: El especialista podrá Mostrar la Escritura	
Observaciones:	

Tabla 2.11 HU No.1.10: Mostrar Reporte

Anexo 2: Tarjetas CRC

Tarjeta CRC No.1 Grupo

grupo	
Description: guardar información de los grupos	
Attributes:	
Name	Description
id_grupo	identificar el grupo
grupo	
Responsibilities:	
Name	Collaborator
crear_grupo	
modificar_grupo	
mostrar_grupo	

Tabla 2.13 Tarjeta CRC Grupo

Tarjeta CRC No.2 Usuarios

usuarios	
Description: guardar nformación de los usuarios	
Attributes:	
Name	Description
id_usuario	identificar el usuario
nombre	
apellidos	
usuario	
pass	
Responsibilities:	
Name	Collaborator
crear_usuario	nivel_acceso
modificar_usuario	nivel_acceso
mostrar_usuario	
ver_detalle	nivel_acceso

Tabla 2.14 Tarjeta CRC Usuarios

Tarjeta CRC No.3 Escritura

escritura	
Description: guardar nformación de la escritura	
Attributes:	
Name	Description
id_escritura	identificar la escritura
fecha	
lectura	
consumo_comb	
existencia_comb	
existencia_aceit	
consum_aceit	
consum_mtto	
existen_refrig	
consum_refrig	
cant_litros	
dencidad_factura	
horas_diarias	
Responsibilities:	
Name	Collaborator
insertar escritura	id_grupo
mostrar escritura	id_grupo

Tabla 2.15 Tarjeta CRC Escritura

Tarjeta CRC No.4 Nivel de acceso

Nivel_acceso	
Description: Guardar información del nivel de acceso	
Attributes:	
Name	Description
id_acceso	identificador del Nivel_acceso
tipo_acceso	
Responsibilities:	
Name	Collaborator
crear_nivel_acceso	
modificar_nivel_acceso	
mostrar_nivel_acceso	

Tabla 2.16 Tarjeta CRC Nivel de Acceso

Tarjeta CRC No.5 Reporte

reporte	
Description: Guardar datos de los reportes	
Attributes:	
Name	Description
fecha	identificar el reporte
generación_neta	
generación_bruta	
densidad	
indice_consumo	
consumo_ventilador	
Responsibilities:	
Name	Collaborator
crear_reporte	
mostrar_reporte	

Tabla 2.17 Tarjeta CRC Reporte

Tarjeta CRC No.6 KG

KG	
Description: guardar información del kg	
Attributes:	
Name	Description
fecha	identificar el kg
kg	
Responsibilities:	
Name	Collaborator
crear_kg	
Right click: add Responsibility;	

Tabla 2.18 Tarjeta CRC KG

Anexo 3: Interfaz de usuario

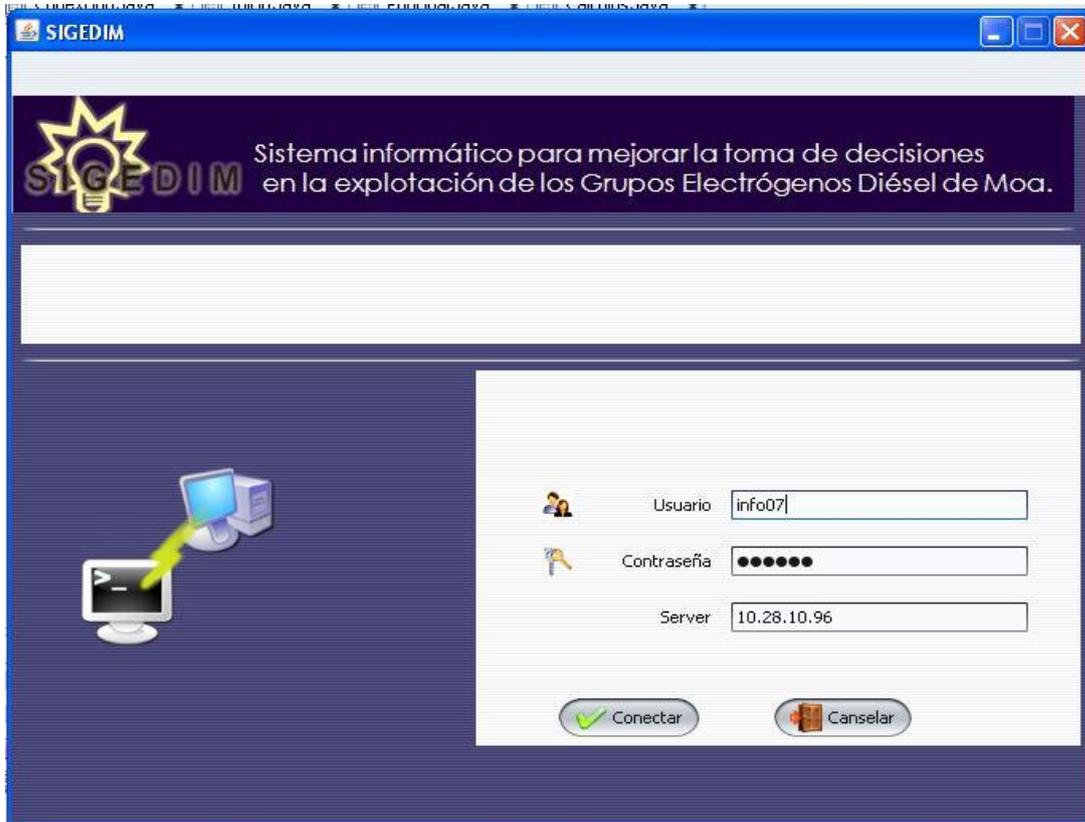


Figura 3.2 Acceso del usuario.



Figura 3.3 Opciones generales.

The screenshot shows a software window with a blue title bar and standard Windows window controls (minimize, maximize, close). The menu bar contains 'File' and 'Edit'. Below the menu bar are three tabs: 'Insertar Usuario', 'Modificar Usuario', and 'Eliminar Usuario'. The main area contains a form with the following fields:

- ID usuario:
- usuario:
- Nombre:
- Passwor:
- Apellidos:
- ID acceso:

An 'Insertar' button is located at the bottom right of the form area.

Figura 3.4 Gestionar usuario.

The screenshot shows a software window with a blue title bar and standard Windows window controls. The header area features a logo of a lightbulb with the text 'SIGEDIM' and the subtitle 'Sistema informático para mejorar la toma de decisiones en la explotación de los Grupos Electrógénos Diésel de Moa.' Below the header is a form with the following fields:

- ID_Grupo:
- existencia_aceit:
- cant_litros:
- Fecha:
- consum_aceit:
- dencidad*factura:
- Lectura:
- consum_mtto:
- horas_diarias:
- consumo_comb:
- existen_refrig:
- existencia_comb:
- consum_refrig:

An 'Insertar' button is located at the bottom right of the form area.

Figura 3.5 Insertar datos de escritura.

File Edit

Consumo Ventilador Densidad Generación Neta Índice de Consumo Generación Bruta

SIGEDIM Sistema informático para mejorar la toma de decisiones en la explotación de los Grupos Electrógenos Diésel de Moa.

Fecha

DiferenciaHoras Hora F

Escritura Hora F_A

Consumo Ventilador

Figura 3.6 Calcular.

File Edit

Consumo Ventilador Densidad Generación Neta Índice de Consumo Generación Bruta

SIGEDIM Sistema informático para mejorar la toma de decisiones en la explotación de los Grupos Electrógenos Diésel de Moa.

Fecha

Generación Neta

Consumo Vent

Generación Bruta

Figura 3.7 Calcular y crear un reporte.

id_es...	fecha	lectura	cons...	exist...	exist...	consu...	cons...	exist...	cons...	cant_...	denci...	horas...	id_gr...
6589	2/01/...	50	54	54	54	54	54	54	54	54	54	54	1
6590	3/01/...	70	54	54	54	54	54	54	54	32	54	54	1
6591	4/01/...	85	54	54	54	54	54	54	54	32	35	54	1
6588	1/01/...	25	54	54	54	54	54	54	54	54	54	40	1
6592	5/01/...	100	54	65	87	54	21	32	98	54	65	65	2

Mostrar

Figura 3.8 Mostrar datos de escritura.

fecha	consumo_ventilador	densidad	generacion_neta	indice_consumo	generacion_bruta
1/01/2007	2480	0.18518518	25	370.37036	2505
2/01/2007	868	0.12962963	25	259.25925	893

Mostrar

Figura 3.9 Mostrar Reporte.

Anexo 4: Tarjetas de Tarea

Tarjeta de tarea No.1: Autenticar usuarios

Tarea ingeniería	
Número tarea: 1	Número historia: 1
Nombre tarea: Autenticar usuarios	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 1/02/2012	Fecha fin: 6/02/2012
Programador responsable: Angel Cutiño Guilarte	
Descripción: Esta tarea facilita que los usuarios al entrar al sistema sean auténticos.	

Tabla 3.5 Tarjeta de tarea No. 1 Autenticar usuarios

Tarjeta de tarea No.2: Insertar usuarios

Tarea ingeniería	
Número tarea: 2	Número historia: 2
Nombre tarea: Insertar usuarios	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 7/02/2012	Fecha fin: 9/02/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El administrador podrá insertar un nuevo usuario del sistema.	

Tabla 3.10 Tarjeta de tarea No. 2 Insertar usuarios

Tarjeta de tarea No.3: Modificar usuarios

Tarea ingeniería	
Número tarea: 3	Número historia: 2

Nombre tarea: Modificar usuarios	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 10/02/2012	Fecha fin: 12/02/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El administrador podrá modificar los datos de un usuario existente del sistema.	

Tabla 3.11 Tarjeta de tarea No. 3 Modificar usuarios

Tarjeta de tarea No.4: Eliminar usuarios

Tarea ingeniería	
Número tarea: 4	Número historia: 2
Nombre tarea: Eliminar usuarios	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 13/02/2012	Fecha fin: 15/02/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El administrador podrá eliminar un usuario existente del sistema.	

Tabla 3.12 Tarjeta de tarea No. 4 Eliminar usuarios

Tarjeta de tarea No.5: Mostrar Generación Neta

Tarea ingeniería	
Número tarea: 5	Número historia: 3
Nombre tarea: Mostrar Generación Neta	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 16/02/2012	Fecha fin: 18/02/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El administrador podrá eliminar un usuario existente del sistema.	

sistema.

Tabla 3.13 Tarjeta de tarea No. 5 Mostrar Generación Neta

Tarjeta de tarea No.6: Mostrar Generación Bruta

Tarea ingeniería	
Número tarea: 6	Número historia: 4
Nombre tarea: Mostrar Generación Bruta	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 19/02/2012	Fecha fin: 21/02/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El administrador podrá eliminar un usuario existente del sistema.	

Tabla 3.14 Tarjeta de tarea No. 6 Mostrar Generación Bruta

Tarjeta de tarea No.7: Mostrar Densidad

Tarea ingeniería	
Número tarea: 7	Número historia: 5
Nombre tarea: Mostrar Densidad	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 22/02/2012	Fecha fin: 24/02/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El administrador podrá eliminar un usuario existente del sistema.	

Tabla 3.15 Tarjeta de tarea No. 7Mostrar Densidad

Tarjeta de tarea No.8: Mostrar Índice de Consumo

Tarea ingeniería

Número tarea: 8	Número historia: 6
Nombre tarea: Mostrar Índice de consumo	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 25/02/2012	Fecha fin: 27/02/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El sistema podrá mostrar el Índice de Consumo.	

Tabla 3.16 Tarjeta de tarea No. 8 Mostrar Índice de Consumo

Tarjeta de tarea No.9: Mostrar Consumo de Ventilador

Tarea ingeniería	
Número tarea: 9	Número historia: 7
Nombre tarea: Mostrar Consumo de Ventilador	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 28/02/2012	Fecha fin: 1/03/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El sistema podrá Mostrar Consumo de Ventilador.	

Tabla 3.17 Tarjeta de tarea No. 9 Mostrar Consumo de Ventilador

Tarjeta de tarea No.10: Insertar Escritura

Tarea ingeniería	
Número tarea: 10	Número historia: 8
Nombre tarea: Insertar Escritura	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 2/03/2012	Fecha fin: 8/03/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El sistema podrá Insertar Escritura.	

Tabla 3.18 Tarjeta de tarea No. 10 Insertar Escritura

Tarjeta de tarea No.11: Mostrar Escritura

Tarea ingeniería	
Número tarea: 11	Número historia: 9
Nombre tarea: Mostrar Escritura	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 9/03/2012	Fecha fin: 14/03/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El sistema podrá Mostrar Escritura.	

Tabla 3.19 Tarjeta de tarea No. 11 Mostrar Escritura

Tarjeta de tarea No.12: Mostrar Reporte

Tarea ingeniería	
Número tarea: 12	Número historia: 10
Nombre tarea: Mostrar Reporte	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 15/03/2012	Fecha fin: 17/03/2012
Programador responsable: Angel Cutiño Guilarte.	
Descripción: El sistema podrá Mostrar Reporte.	

Tabla 3.20 Tarjeta de tarea No. 12 Mostrar Reporte

Anexo 5: Pruebas de Aceptación

Pruebas de aceptación para la HU: Autenticar usuarios

Prueba de aceptación
HU: Autenticar usuarios
Nombre: Prueba para comprobar la autenticación de un usuario.
Descripción: El usuario se autentica para entrar al sistema.
Condiciones de ejecución: Se necesita el usuario y contraseña.

Entrada/Pasos ejecución: El usuario intenta introducir usuario y contraseña.
Resultado: Se emite un mensaje de error en caso de que no es correcta la contraseña sino el usuario estará entrando al sistema según el modo y sus privilegios.
Evaluación de la prueba: Aceptada.

Tabla 3.21 PA: Prueba para comprobar la autenticación de un usuario.

Pruebas de aceptación para la HU: Gestionar usuarios

Prueba de aceptación
HU: Gestionar usuarios
Nombre: Prueba para comprobar la inserción de un nuevo usuario.
Descripción: El Administrador insertar un nuevo usuario del sistema estos tendrán prioridades de especialistas.
Condiciones de ejecución: Se necesita el nombre de usuario, contraseña y confirmar contraseña.
Entrada/Pasos ejecución: El administrador introduce el nombre de usuario, contraseña y su confirmación.
Resultado: Se emite un mensaje de error en caso de que no se pueda insertar el nuevo usuario o si la confirmación no es correcta, sino se emite un mensaje de información en caso de éxito.
Evaluación de la prueba: Aceptada.

Tabla 3.22 PA: Prueba para comprobar la inserción de un nuevo usuario.

Pruebas de aceptación para la HU: Gestionar usuarios

Prueba de aceptación
HU: Gestionar usuarios
Nombre: Prueba para comprobar la modificación de un usuario.
Descripción: El Administrador modificara el nombre o la contraseña de un usuario del sistema.
Condiciones de ejecución: Se necesita el nombre de usuario, el nuevo nombre de usuario o el mismo, contraseña y confirmar contraseña.
Entrada/Pasos ejecución: El administrador introduce el nuevo nombre de usuario, contraseña y su confirmación.
Resultado: Se emite un mensaje de error en caso de que no se pueda modificar el usuario, si la confirmación no es correcta, sino se emite un mensaje de información en caso de éxito.
Evaluación de la prueba: Aceptada.

Tabla 3.23 PA: Prueba para comprobar la modificación de un usuario.

Pruebas de aceptación para la HU: Gestionar usuarios

Prueba de aceptación

HU: Gestionar usuarios
Nombre: Prueba para comprobar la eliminación de un usuario.
Descripción: El Administrador elimina un usuario del sistema.
Condiciones de ejecución: Se necesita el nombre de usuario.
Entrada/Pasos ejecución: El administrador selecciona el nombre de usuario.
Resultado: Se emite un mensaje de confirmación si este es aceptado entonces se emite un mensaje de error en caso de que no se pueda eliminar el usuario, sino se emite un mensaje de información en caso de éxito.
Evaluación de la prueba: Aceptada.

Tabla 3.24 PA: Prueba para comprobar la eliminación de un usuario.

Pruebas de aceptación para la HU: Insertar Escritura

Prueba de aceptación
HU: Insertar Escritura
Nombre: Prueba para comprobar la inserción de los datos de una escritura.
Descripción: El Especialista inserta una nueva escritura al sistema.
Condiciones de ejecución: Se necesita que todos los campos estén llenos.
Entrada/Pasos ejecución: El especialista introduce los datos y su confirmación.
Resultado: Se emite un mensaje de error en caso de que no se pueda insertar la escritura.
Evaluación de la prueba: Aceptada.

Tabla 3.25 PA: Prueba para comprobar la inserción de una escritura.

Pruebas de aceptación para la HU: Mostrar Reporte

Prueba de aceptación
HU: Mostrar reporte
Nombre: Prueba para comprobar reporte.
Descripción: El Especialista valida la entrada de datos.
Condiciones de ejecución: El usuario debe de entrar a la Aplicación para poder obtener un reporte.
Entrada/Pasos ejecución: El usuario selecciona la opción mostrar reporte.
Resultado: Se muestra en un formulario toda la información.
Evaluación de la prueba: Aceptada.