

# IMPLEMENTACIÓN DE UN SOFTWARE DE GESTIÓN AMBIENTAL PARA EL SISTEMA DE GESTIÓN AMBIENTAL DE LA EMPRESA CONSTRUCTORA ECM4.

Ing. Rubén Hayron Rodríguez Espinosa<sup>1</sup>, Dr.C. Elpidia Caridad Cruz Cabrera<sup>2</sup>

1. Constructora ECM4 de Matanzas

2. Universidad de Matanzas, [caridad.cruz@umcc.cu](mailto:caridad.cruz@umcc.cu)

## Resumen

Este trabajo tiene como objetivo implementar una herramienta de software para el Sistema de Gestión Ambiental de la empresa constructora ECM4. El desarrollo de este programa permitirá generar procedimientos con el fin de garantizar el mejoramiento ambiental de la entidad a través del análisis y el diagnóstico de los aspectos ambientales enmarcados en la misma. Se realizará la implementación de esta herramienta para la cual se pasará por cada una de las etapas del proceso de desarrollo de software, Al finalizar, se arribará a la conclusión de que la implementación de este sistema garantiza el mejoramiento ambiental de la empresa ya que permitirá la identificación y la evaluación de las causas que originan el impacto negativo con el objetivo de minimizarlo o evitar que este vuelva a ocurrir.

**Palabras claves:** Empresa constructora; medio ambiente; software de gestión ambiental; sistema de gestión ambiental.

---

## Introducción

En la actualidad las entidades del sector empresarial se encuentran cada vez más interesadas en medir y controlar el impacto que provocan sus actividades, productos y servicios en el medio ambiente. Este comportamiento viene impulsado por el desarrollo de políticas económicas, legislaciones y otras medidas, cada vez más exigentes, concebidas para fomentar la protección y cuidado del medio ambiente, por el incremento de la preocupación a nivel global de la escasez de los recursos naturales y por el futuro y la subsistencia del hombre, abogando por impulsar el desarrollo sostenible. El sector empresarial de la construcción, presenta una serie de particularidades que hacen que implementar un Sistema de Gestión Ambiental (SGA) sea un tanto difícil, debido a que las constructoras presentan características particulares como son la especificidad de cada obra y la gran magnitud, complejidad y heterogeneidad que los productos finales presentan.

En la empresa constructora ECM4 no existe un software que gestione los aspectos ambientales que esta genera. Actualmente el sistema dedicado a la gestión ambiental en la empresa lleva el control del agua, desechos, papel, reciclaje, combustible y los gastos de energía eléctrica que la misma produce. Estas acciones están encaminadas a mitigar los efectos negativos que ocasionan daños medioambientales o podrían ocasionarlos en el futuro, a minimizar los residuos y a utilizar los recursos naturales y la energía de manera racional. Entre los problemas que presenta el especialista encargado de controlar el sistema de gestión ambiental en la ECM4 se encuentran: que existe una dispersión de tablas Excel que hace que el gestionar y visualizar la información contenida en ellos sea un proceso lento, y la saturación de archivos no permiten tratar de manera eficaz la información almacenada por mucho tiempo.

A través del software de gestión ambiental el proceso de gestión, control y fiscalización del sistema será mucho más ágil, ya que permitirá gestionar e integrar todos los subsistemas de gestión ambiental en una misma herramienta para introducir y visualizar los datos, analizarlos de forma gráfica y realizar planes de acción preventiva o de mitigación. Esta herramienta fue ideada debido a la necesidad de la empresa de tener un programa hecho a su medida, que reuniera toda la información generada por cada subsistema de gestión ambiental vigente en la entidad. Por tanto, esta investigación tiene como objetivo llevar a cabo la implementación de un software de apoyo al sistema de gestión ambiental que permita controlar e identificar los aspectos ambientales que puedan generar un determinado impacto en la entidad.

El tema propuesto desde el punto de vista económico, representa para la empresa constructora un ahorro de recursos en cuanto a combustible, papel, reutilización de escombros, ahorro de energía eléctrica y agua, gracias a que nos permitirá establecer medidas de tendencias y de comportamiento de los indicadores en cada uno de estos subsistemas de gestión a lo largo del tiempo, para poder estimar y prevenir gastos innecesarios en el futuro.

## Desarrollo

El uso de las tecnologías ha sido un paso de avance para impulsar el desarrollo de los sistemas de gestión. En el mundo existen diversos software dedicados a la gestión ambiental que implementan de manera eficiente los sistemas de gestión ambiental de diversas entidades, un ejemplo de esto es el software EcoGestor de la empresa ENVIRA y la plataforma tecnológica ISOTools la cual facilita la implementación, automatización y mantenimiento de muchas de las normas ISO entre estas la de los Sistemas de Gestión de Medio Ambiente ISO 14001. Estos softwares son basados en su mayoría en tecnologías web y garantizan para facilitar el control y gestión de los datos su permanencia en la nube, condición por la cual teniendo en cuenta las normas de seguridad de la empresa ECM4 imposibilita el acceso a estas plataformas. Entre los tipos de software de SGA que no cumplen con esta condición se encuentra el programa de la empresa americana Metricstream, esta empresa se especializa en el desarrollo de herramientas de software de gestión empresarial y la misma contiene una aplicación para la ISO 14001, la cual posee numerosas herramientas, entre ellas un módulo para el cálculo de riesgo ambiental. El acceso a este tipo de herramientas trae consigo gastos económicos por concepto de compra y mantenimiento del software, razón por la cual la empresa desvía su atención hacia el uso de software libre.

Existen en la actualidad, un gran número de herramientas de software libre que se dedican a la esfera medioambiental. Muchas de estas se especializan en campos como el análisis de datos ambientales, sistemas de información geográfica, de modelación de flujos de aguas subterráneas y de la calidad del aire entre otros. Aunque existen numerosas herramientas de software de aplicaciones ambientales, ninguna tiene en cuenta las normativas locales en las que se enmarca la ECM4 ni el enfoque que esta quiere obtener, por lo que se requiere de un software hecho a la medida, teniendo en cuenta las características que una empresa de construcción presenta para poder realizar su gestión ambiental.

La implementación se llevó a cabo primeramente realizando un estudio detallado del Sistema de Gestión Ambiental de la empresa constructora ECM4. Luego se procedió realizando un estudio de los softwares más reconocidos a nivel mundial, así como los más usados en el país por el sector de la construcción, con el fin de establecer un nexo que permita mantener las ideas previamente existentes a nivel global desde el punto de vista de la gestión ambiental y las particularidades que la ECM4 requiere.

## Metodología de desarrollo de Software

La selección de una metodología apropiada, resulta en ocasiones un factor esencial para obtener un software con la calidad esperada.

Luego de la evaluación de las metodologías de desarrollo y el estudio de sus características, etapas de desarrollo y ventajas que posibilitan, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de

software; el cliente es parte del equipo de desarrollo, el equipo de desarrollo es de una persona, además de la dificultad para un equipo de desarrollo pequeño el adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo, lo cual permitió identificar la Programación Extrema (XP) como una alternativa acertada (Joskowicz, 2008).

### **Metodología de software XP**

La metodología XP además de ser una de las más conocidas y más usadas de las metodologías ágiles fue seleccionada para guiar el proceso de desarrollo de la presente investigación, debido a que se está en presencia de un proyecto pequeño, de corta duración, con un reducido equipo de desarrollo, donde el cliente forma parte activa del mismo, permitiendo la comunicación y retroalimentación frecuente entre ambas partes. Una de las prácticas más significativas que posee esta metodología, es que con ella es posible simplificar el diseño para agilizar el desarrollo, así como facilitar el mantenimiento del sistema. También es recomendable utilizar esta metodología debido a que los requisitos tienen altas probabilidades de cambiar con el tiempo (Sánchez, 2004).

### **Lenguaje de Programación**

Para el desarrollo de la solución se escogió el lenguaje de programación Java debido a que comprende las siguientes características.

- **Simple:** elimina la complejidad de otros lenguajes como C y se enfoca en el contexto de los lenguajes orientados a objetos.
- **Robusto:** maneja la memoria de la computadora para que el programador no se tenga que preocupar por ello, además de realizar verificaciones en busca de errores lo mismo en tiempo de compilación que en tiempo de ejecución.
- **Portable:** un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java, ya que su código compilado es interpretado.
- **Multiproceso:** puede ejecutar diferentes líneas de código al mismo tiempo.
- **Dinámico:** no es necesario que compile todas las clases de un programa para que funcione. Al efectuar al menos un cambio en alguna de las clases, Java se encarga de realizar un enlace dinámico o una carga dinámica para encontrar las clases.
- **Interpretado:** corre en máquina virtual.

## Entorno integrado de desarrollo

Para el desarrollo de la solución, como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) se seleccionó el NetBeans 8.2. Es libre, de código abierto y contiene todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. Es conocido por la integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, coloreo de sintaxis, permite la utilización y edición de los componentes visuales de forma sencilla. Incluye también el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente. Además es un entorno de desarrollo disponible para varios sistemas operativos como Windows, Mac, Linux y Solaris (Oracle, 2015):

## Herramienta para el modelado

Como herramienta para el modelado de la solución se seleccionó *Visual Paradigm* para UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés). Esta herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. El VP-UML en su versión v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, que brinda a los equipos de desarrollo de software todo lo necesario para la planificación de software, captura de requisitos, planificación de controles y modelado de clases y de datos. Esta herramienta ofrece gran interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo, además de hacer posible la generación de código Java desde los diagramas (Paradigm, 2016).

## Sistema Gestor de Base de Datos.

Como sistema gestor de base de datos (SGBD) se utilizó SQLite, el cual es un SGBD relacional que está contenido en una pequeña biblioteca (~500kb) construida en el lenguaje C. Es un proyecto de dominio público creado por Dr. Richard Hipp. A diferencia de los SGBD cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones (SQLite, 2015).

Esto reduce el tiempo de acceso a la base de datos (BD), debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la BD (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de BD al inicio de cada transacción. Este SGBD dispone de una completa interfaz orientada a

objetos, con distintas funciones que facilitan la manipulación de datos y posee funciones muy similares a las que se pueden manejar con MySQL.

### Descripción de la solución

Para elaborar la propuesta de solución en este artículo se describe solo el módulo de Gestión de la Electricidad. A continuación, se muestra el diagrama de actividades que modela la secuencia de pasos del sistema.

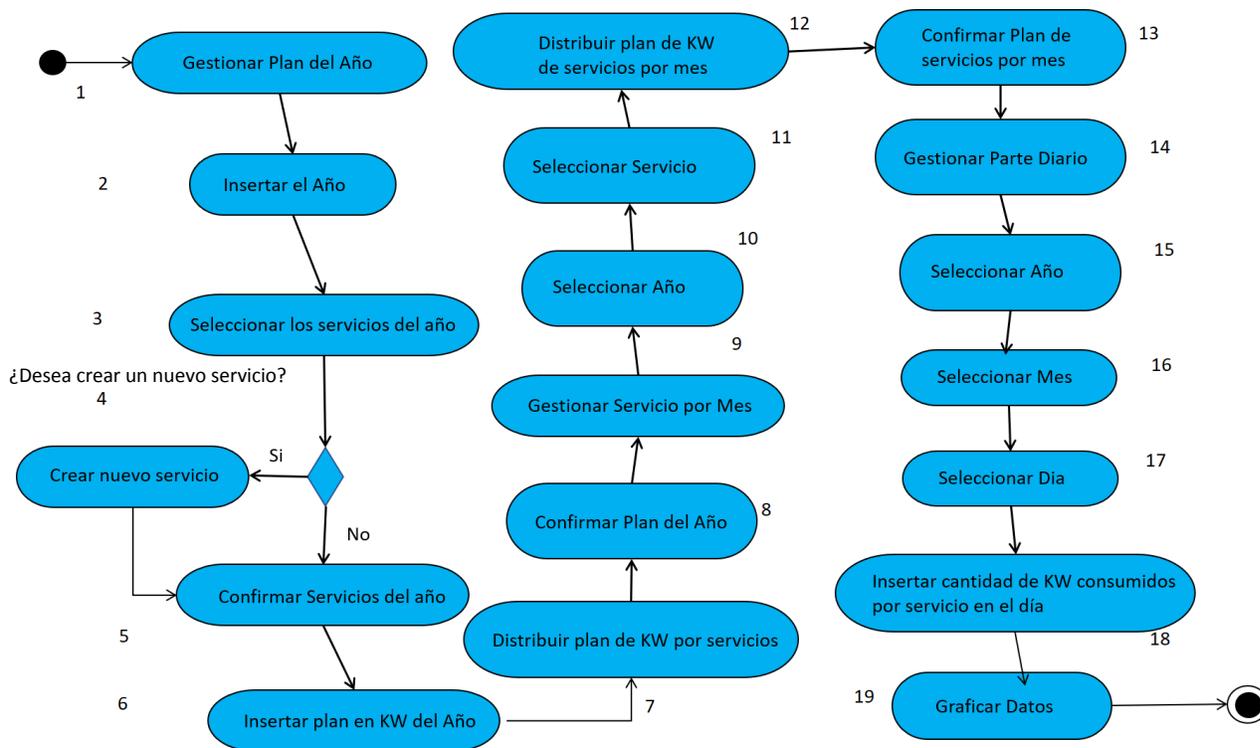


Ilustración 1: Diagrama de flujo del Módulo Gestión de Electricidad

1. En esta etapa se selecciona el botón Gestionar Plan del Año
2. Inserción del año a gestionar.
3. Selección de los servicios del año.
4. El usuario puede crear un nuevo servicio si este no existe en la lista de servicios actuales.
5. Selección de los servicios que funcionarán en el año previsto.

6. El usuario debe entrar la cantidad de kW planificada para el año previsto.
7. Se le brinda la posibilidad al usuario de generar la misma distribución de kW por cada servicio en partes iguales según el plan del año o de forma manual.
8. Confirmar el Plan del Año.
9. Selección del botón de Gestionar Servicios por Mes.
10. Selección del Año.
11. Selección del servicio a planificar.
12. En esta fase se le brinda la posibilidad al usuario de generar la misma distribución de kW por servicio en cada mes en partes iguales según el plan del año por servicio, o de forma manual.
13. Confirmación del plan de Servicios por Mes.
14. Selección del botón de Gestionar Parte Diario.
15. Selección del Año.
16. Selección del Mes.
17. Selección del Día.
18. El usuario debe Insertar la cantidad de kW consumidos por cada servicio en el día.
19. Se brinda la posibilidad de graficar los datos del consumo de kW hasta la fecha por cada servicio.

## **Historias de Usuario**

Uno de los artefactos definidos por la metodología XP son las HU las cuales son utilizadas para especificar las funcionalidades que brindará el sistema. Cada HU es una representación de un requerimiento de software escrito en una o dos frases, utilizando el lenguaje común del usuario. Estas representan una forma rápida de administrar los requerimientos de los usuarios, sin necesidad de elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos (Beck, 2002). En correspondencia con la prioridad asignada por el cliente a cada HU y teniendo en cuenta la complejidad y riesgo determinado por el programador, se realiza la estimación de una de las identificadas.

A continuación, se muestra una de las Historias de Usuario identificadas.

Tabla 1: Historia de usuario # 8

Historia de usuario	
<b>Número:</b> 8	<b>Nombre:</b> Graficar consumo de kW
<b>Iteración asignada:</b> 3	
<b>Prioridad en negocio:</b> Alta (Alta / Media / Baja)	<b>Puntos estimados:</b> 2 semanas
<b>Riesgo en desarrollo:</b> Alto (Alto / Medio / Bajo)	<b>Puntos reales:</b> 2 semanas
<b>Descripción:</b> Dado los resultados obtenidos en la gestión del consumo diario por servicios el sistema muestra la graficación del consumo de kW diario por servicio hasta la fecha.	
<b>Observaciones:</b>	

### *Estimación de esfuerzos por Historias de Usuario*

Tras la definición de prioridades de cada HU se realiza una estimación del esfuerzo necesario de cada una de ellas. Donde la unidad de estimación es el punto, un punto equivale a una semana ideal de programación. Esta estimación arrojó los resultados siguientes:

Tabla 2: Puntos de estimación por historias de usuario

No	Historias de usuario	Puntos de estimación
1	Gestionar plan del Año	2
2	Gestionar plan del Mes	2
3	Gestionar Parte Diario	2
4	Graficar Datos	2

### Requisitos No Funcionales

Los requisitos no funcionales son aquellos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento que debe cumplir o cumple el producto de software (Andrew & Greene,2005). Estos requisitos tienen lugar por la demanda del sistema de ciertas funciones o cualidades, o porque el cliente desea que ese requerimiento forme parte del producto de software final.

**Requisitos de software:** Se debe tener instalada en la computadora la máquina virtual de Java en su versión 8 o superior.

**Requisitos de hardware:** La computadora deben tener como mínimo las siguientes características:

- 1 GB de memoria RAM.
- Más de un núcleo físico con frecuencia mayor de 1.6 GHz.

### Plan de iteraciones

Después de definir las HU y estimar el esfuerzo propuesto para la realización de cada una de ellas, se requiere establecer el contenido de trabajo de estas, y establecer el plan de iteraciones, regulando la cantidad de HU a implementar dentro del rango establecido por la estimación efectuada. Es decir, una iteración tiene como finalidad implementar las HU que se consideraron más necesarias atendiendo a su relevancia e impacto. Partiendo de los

aspectos tratados anteriormente, la aplicación que se pretende construir se desarrollará en 3 iteraciones, expuestas de forma detallada a continuación:

### **Iteración 1**

En esta iteración se implementaron las funcionalidades que permiten Gestionar los Planes y el consumo de Energía.

### **Iteración 2**

En esta iteración se realizan todas las funcionalidades relacionadas con la graficación del consumo de Energía Eléctrica.

*Tabla 3: Plan de duración de las iteraciones*

<b>Iteraciones</b>	<b>Historias de usuario</b>	<b>Duración</b>
Iteración 1	Gestionar plan del Año	6
	Gestionar plan del Mes	
	Gestionar Parte Diario	
Iteración 2	Graficar Datos	2

### **Diseño de la propuesta de solución**

En este apartado se hace alusión a los principales elementos que componen el diseño del sistema propuesto, donde siguiendo los planteamientos de la metodología XP este debe ser un diseño simple y de fácil interacción con el usuario, basándose fundamentalmente en el desarrollo de las tarjetas Clases, Responsabilidad y Colaboración (CRC).

### **Tarjetas CRC**

Las tarjetas CRC ayudan a definir actividades durante el diseño del sistema y evitan el enfoque procedimental destacando la orientación a objetos (Beck, 1999). Al poder trabajar con una metodología basada en objetos estas tarjetas permiten que el equipo de desarrollo completo contribuya también en la tarea del diseño. La estructura física de las tarjetas CRC

facilita la interacción entre los integrantes del proyecto, en sesiones en las que se aplican técnicas de grupos como, juego de roles o tormenta de ideas y se ejecutan escenarios a partir de la especificación de los requisitos, historias de usuarios o casos de uso. De esta forma, van surgiendo las entidades del sistema junto con sus responsabilidades y colaboraciones (Reinaga, 2009).

Tabla 4: Tarjeta CRC de la clase Principal

<b>Clase: Principal</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Permite mostrar la gráfica de los datos gestionados.	Gráfica
Permite a la aplicación conectarse a la base de datos.	Conexión Sqlite

### **Estilos de codificación**

Los estilos de codificación comprenden todos los aspectos de la generación de código, debiendo reflejar un estilo armónico en su confección (como si un único programador hubiera escrito todo el código). Estos estilos permiten entender de manera rápida y sencilla el código empleado en el desarrollo del software, es por eso que, al comenzar un proyecto, se debe establecer un estilo de codificación para asegurarse de que se trabaje de forma coordinada, permitiendo realizar el mantenimiento a un sistema de software, al establecer como operar con la base de código existente (García, 2011).

- Los nombres de las clases serán con mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma.
- Los comentarios irán encima del método correspondiente.
- La primera letra de los nombres de los métodos se escribirá en minúscula y si el nombre es compuesto las siguientes palabras empezarán con mayúscula.
- Se escribirá una declaración por línea.

- Los identificadores de las variables se escribirán en minúscula y en caso de ser un nombre compuesto las siguientes palabras empezarán con mayúscula.

## Patrones de diseño utilizados

Los patrones de diseño constituyen una descripción de la interacción entre las clases, con el objetivo de resolver un problema de diseño general en un determinado contexto. El uso de patrones permite la estandarización de la forma en que se realiza el diseño de manera que el código pueda ser reutilizable, y se le pueda dar mantenimiento cuando se requiera. En este proyecto se utilizaron los siguientes patrones de diseño:

**Experto:** El uso de este patrón se evidencia en la clase Grafica. Esta es la encargada de Graficar los datos.

**Creador:** Dentro de la aplicación este patrón se observa en las acciones del controlador Principal, en la cual se crean objetos para el procesamiento de los datos. Estas clases tienen la responsabilidad de instanciar objetos para cumplir sus funcionalidades.

## Interfaz del Programa

A continuación, se muestran las interfaces principales del software SGA-ECM4 (Sistema de Gestión Ambiental de la ECM4).

The screenshot shows the 'Gestión de la Energía' interface. At the top, there are three buttons: 'Gestionar Plan del Año', 'Gestionar Plan por Mes', and 'Gestionar Parte'. Below these are three dropdown menus for 'Año: 2019', 'Mes: 1', and 'Día: 1'. The main content is a table with the following data:

Servicio	Inicial	Final
UB MATANZAS (hotelería)(PREPAGO)		512.0
UB VIALES PONCHERA (PREPAGO)		123.0
UB VIALES TALLER		456.0
UB VIALES OFICINAS (PREPAGO)		464.0
CONSTRUCTORA MILITAR(EQUIPOS)		468.0
UB TOM - TALLER		986.0
UB TOM		479.0
UB INDUSTRIA 2		564.0
UB INDUSTRIA 1		358.0
OBRA VIVIENDAS GELPI		126.0
V. GELPI F. PERMANENTES		459.0
CAMPO TIRO TOM 220 V M-369		587.0
CAMPO TIRO (TOM) M-369 (ROZADORA)		785.0

Below the table is a green 'Confirmar' button. At the bottom right of the interface, there are three buttons: 'Generar Informe', 'Graficar Datos', and 'Gestionar Parte'.

Ilustración 2: Interface principal del SGA-ECM4

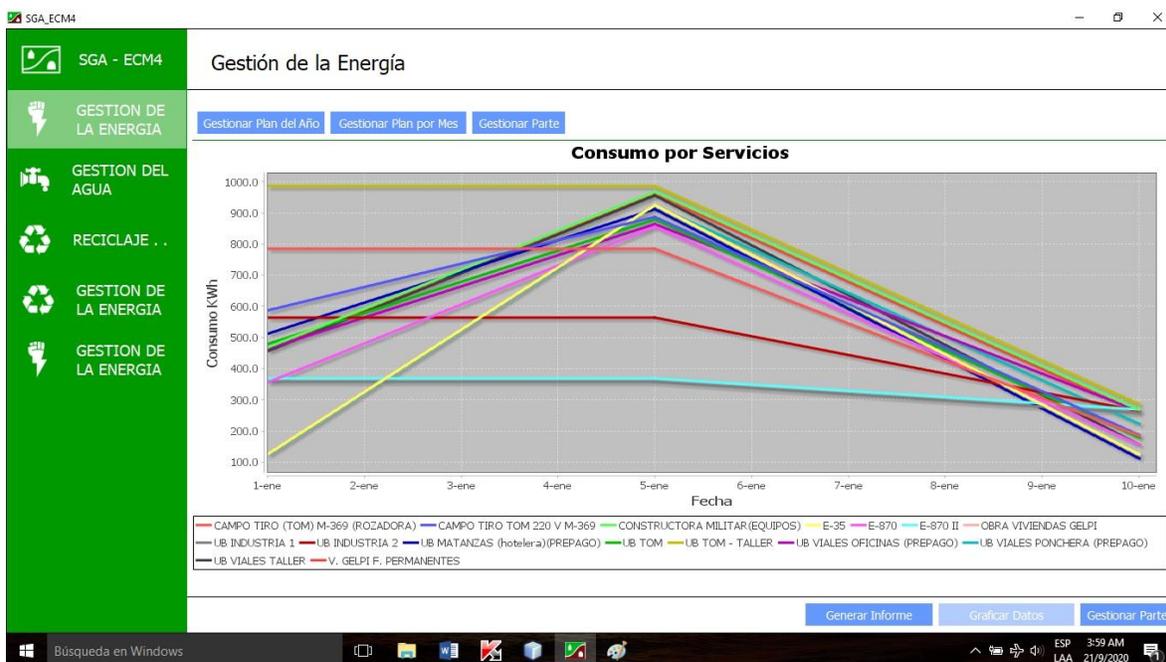


Ilustración 3: Interface de graficación del SGA-ECM4

## Conclusiones

Durante el desarrollo de este trabajo se describió la propuesta de solución según la metodología XP, generando un grupo de artefactos que caracterizan al sistema y describen los aspectos a tener en cuenta para el desarrollo de la solución propuesta. Se realizó también la estimación del esfuerzo necesario para la implementación de las HU. Se definieron las actividades implementadas durante la fase de diseño mediante las tarjetas CRC. Se describió el funcionamiento del sistema de gestión de la energía, y cada una de sus funcionalidades, así como el empleo de estándares de codificación y patrones de diseño, los cuales, permitieron realizar un diseño organizado y de fácil comprensión. Con esta herramienta de software se espera superar en cuanto a la optimización de tiempo de gestión al actual sistema. Por tanto, se desea obtener un software con un alto nivel de usabilidad, flexibilidad, y que sea adaptable para poder ajustarse fácilmente a los cambios y mejoras que pueda sufrir el sistema de gestión ambiental presente actualmente en la empresa constructora ECM4.

## Referencias bibliográficas

BECK, K. Embracing change with extreme programming. *Computer*, no 10 vol. 32, 1999, pp. 70-77.

BECK K. *Una explicación de la programación extrema: aceptar el cambio*. Massachusetts: Addison Wesley, 2002.

CASAS, S y REINAGA, H. Aspectos Tempranos: un enfoque basado en Tarjetas CRC. *Avances en Sistemas e Informática*, Medellín, no 1 vol. 6, 2009, pp. 85-92.

GARCIA, O. *Nomenclatura estándar del código Java*. . [En línea] [fecha de consulta : 22 de julio de 2020]. Disponible en: <http://www.elclubdelprogramador.com/2011/08/22/java-nomenclatura-estandar-del-codigo-java/>

JOSKOWICZ, J. *Reglas y prácticas en eXtreme Programming* . Pontevedra: Universidad de Vigo, 2008.

ORACLE. JAVA. [En línea]. Estados Unidos [fecha de consulta 20 de enero de 2020.]. Disponible en <http://www.java.com/es/about/>

SÁNCHEZ MENDOZA, M. A. *Metodologías de desarrollo de software*. 2004.

SQTLite. *SQTLiteManagenet.org*. [En línea] [fecha de consulta: 30 de junio 2020.]. Disponible en : <http://www.sqlitemanager.org>.

STELLMAN, A. y GREENE, J. *Applied software project management*. " O'Reilly Media, Inc.", 2005.

VISUAL, PARADIGM. *Visual Paradigm*. [En línea] Estados Unidos [fecha de consulta : 30 de junio de 2020]. Disponible en <https://www.visual-paradigm.com>.