

Análisis y Diseño del Sistema para el Control y Tratamiento de Datos (SISCOT).

M.G. Flores-Luevanos¹, E. Moreno-Nuñez¹, J.M. Arzola-Monreal¹

Resumen— La presente investigación se enfoca en el análisis y diseño del software “Sistema para el Control y Tratamiento de Datos (SISCOT)”.

Se detalla el objetivo el cual es: “Desarrollar el software que permita controlar y dar tratamiento a los datos obtenidos de una interfaz humano-máquina (HMI)”. Con estos datos se generarán los reportes correspondientes para obtener la estadística básica y las curvas de comportamiento de las variables: velocidad de giro, presión de entrada, consumo y amperaje; las cuales corresponden a un modelo clasificador de carbonato de calcio. El tratamiento de estos datos servirá para hacer eficiente el consumo de energía eléctrica en el proceso, así como para ajustar las calidades y capacidad de producción de carbonato de calcio grado técnico. Se presentan los productos principales del análisis y diseño del software y se describe la metodología utilizada para su desarrollo. Finalmente, se incluyen las conclusiones de este proceso.

Temas claves—proceso, análisis, desarrollo, software.

Abstract — This research focuses on the analysis and design software "System Control and Data Processing (SISCOT)".

The objective is detailed which is "To develop the software that permit to control and treat the data obtained from a human-machine interface (HMI)". With these data it will be generated the corresponding reports to obtain the basic statistics and the performance curves of the variables: rotational speed, inlet pressure, amperage and consumption; which correspond to a pattern calcium carbonate classifier. The processing of these data will be used to make efficient consumption of electric energy in the process, as well as the adjustment of the quality and production capabilities of technical grade calcium carbonate. It is presented the main products of analysis and software design and is described the methodology used for its development. Finally, are included the conclusions of this process.

Keywords — process, analysis, development, software.

I. INTRODUCCIÓN

La parte esencial de este proyecto es el desarrollo del Sistema para el Control y Tratamiento de los Datos (SISCOT). Su finalidad es el procesamiento y almacenamiento de los datos de las variables: velocidad de giro, presión de entrada, consumo y amperaje, obtenida de

una interfaz HMI, de un modelo clasificador de carbonato de calcio.

Dicho procesamiento permitirá la generación de curvas de comportamiento a través de los valores estadísticos resultantes.

Fundamento Teórico.

Se hará uso de las herramientas siguientes para el análisis y diseño del sistema: Microsoft Project®, StarUML, así mismo para el desarrollo se usarán, Microsoft Visual Studio®: C# y Microsoft SQL Server®.

UML.

Es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture), actualmente es de licencia abierta GNU/GPL. Es una de las mejores alternativas gratis que hay en Internet para el modelamiento del software.

Entre sus principales características:

- Soporte completo al diseño UML mediante el uso de.
 - Diagrama de casos de uso.
 - Diagrama de clase.
 - Diagrama de secuencia.
 - Diagrama de colaboración.
 - Diagrama de estados.
 - Diagrama de actividad.
 - Diagrama de componentes.
 - Diagrama de despliegue.
 - Diagrama de composición estructural (UML 2.0).
- Definir elementos propios para los diagramas, que no necesariamente pertenezcan al estándar de UML,
- La capacidad de generar código a partir de los diagramas y viceversa, actualmente funcionando para los lenguajes c++, c# y java.
- Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas.
- Patrones GoF (Gang of Four), EJB (Enterprise JavaBeans) y personalizados.
- Plantillas de proyectos.
- Posibilidad de crear plugins para el programa

Microsoft Project®.

Microsoft Project ayuda a planificar proyectos y a colaborar con otras personas fácilmente teniendo todo

organizado y realizando el seguimiento de los proyectos con el único sistema de administración de proyectos diseñado para trabajar sin ningún problema con otras aplicaciones de Microsoft y servicios en la nube. [1], [2]

Ciclo de vida.

El ciclo de vida elegido para el desarrollo del software SISCOT es de tipo incremental.

“En un proceso de desarrollo incremental, los clientes identifican, a grandes rasgos, los servicios que proporciona el sistema. Identifican qué servicios son más importantes y cuáles menos. Entonces se definen varios incrementos en donde cada uno proporciona un subconjunto de la funcionalidad del sistema... tan pronto como se completan los nuevos incrementos, se integran en los existentes de tal forma que la funcionalidad del sistema mejora con cada incremento entregado.”

[3], [4]

Una vez identificados los incrementos en el sistema, se definen con detalle los requerimientos para los servicios que se van a entregar en el primer incremento y se desarrolla la primera versión del sistema. Durante el desarrollo es posible llevar a cabo un análisis adicional para los requerimientos posteriores sin cambiar los requerimientos para el incremento en el que esté trabajando actualmente.

Cuando se completa y entrega, el cliente puede ponerlo en servicio es decir obtiene una entrega temprana de parte de las funcionalidades del sistema.

De esta manera pueden experimentar con el sistema para así clarificar sus requerimientos para los incrementos posteriores y las últimas versiones del incremento actual.

En la Figura 1, se muestra el diagrama correspondiente a este ciclo de vida.

II. PARTE TÉCNICA DEL ARTÍCULO

A. Metodología

La metodología propuesta para este proyecto es la orientada a objetos. Ésta consiste en armar módulos basados en componentes y cada componente es independiente del otro. Esto permite que el código sea reutilizable y es más fácil de mantener pues los cambios están localizados en cada componente. Su principal diferencia con la metodología estructurada es que ésta última comprende los procesos como funciones y cada función a realizar por el sistema se descompone en pequeños módulos individuales y finalmente se unen las soluciones para crear la solución al problema.

La ISO (International Organization for Standardization) en su norma 12207 define al ciclo de vida de un software como un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando desde la definición hasta la finalización de su uso.

Ciclo de vida propuesto: incremental.

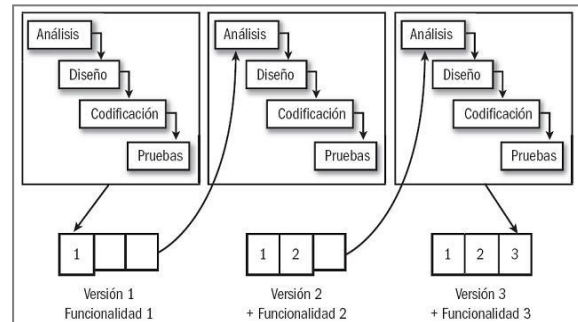


Figura 1. Ciclo de vida de software.

Especificación de requisitos según el estándar de IEEE 830.

Después de la recopilación y verificación de los requisitos, y tomando en cuenta las características necesarias para el correcto funcionamiento del software de acuerdo a las expectativas del mismo.

Esta especificación se estructura basada en las directrices dadas por la última versión del estándar “IEEE Recommended Practice for Software Requirement Specifications ANSI/IEEE 830-1998”

B. Análisis orientado a objetos.

Con base en el documento formal de requerimientos, se realiza el análisis orientado a objetos; en principio se desarrolla, utilizando el Lenguaje Unificado de Modelado (UML) y la herramienta StarUML, el diagrama de casos de uso general del sistema. “El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir.” [5].

Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML incluye conceptos semánticos, notación y principios generales, tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero si está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. [5]

A continuación se presenta en la Figura 2, el diagrama de casos de uso, en donde se observan las funciones siguientes como administrador:

- Iniciar sesión.
- Cargar datos.
 - o desde archivo.
 - o desde el historial.
- Generar gráficas.
 - o por variable.
 - o agrupadas.
- Personalizar gráficas.
- Vista detallada de los gráficos.
- Elaborar reportes.
- Salir del sistema.

Modelado de clases.

Se muestran las clases presentes en el sistema SISCOT mediante el diagrama de clases de la Figura 3.

Se crearon entre otras, las siguientes:

- Cambiar contraseña.
- Login.
- Base de Datos.
- Gráfica.
- Cifrado.
- Historial.
- Crear Reporte.

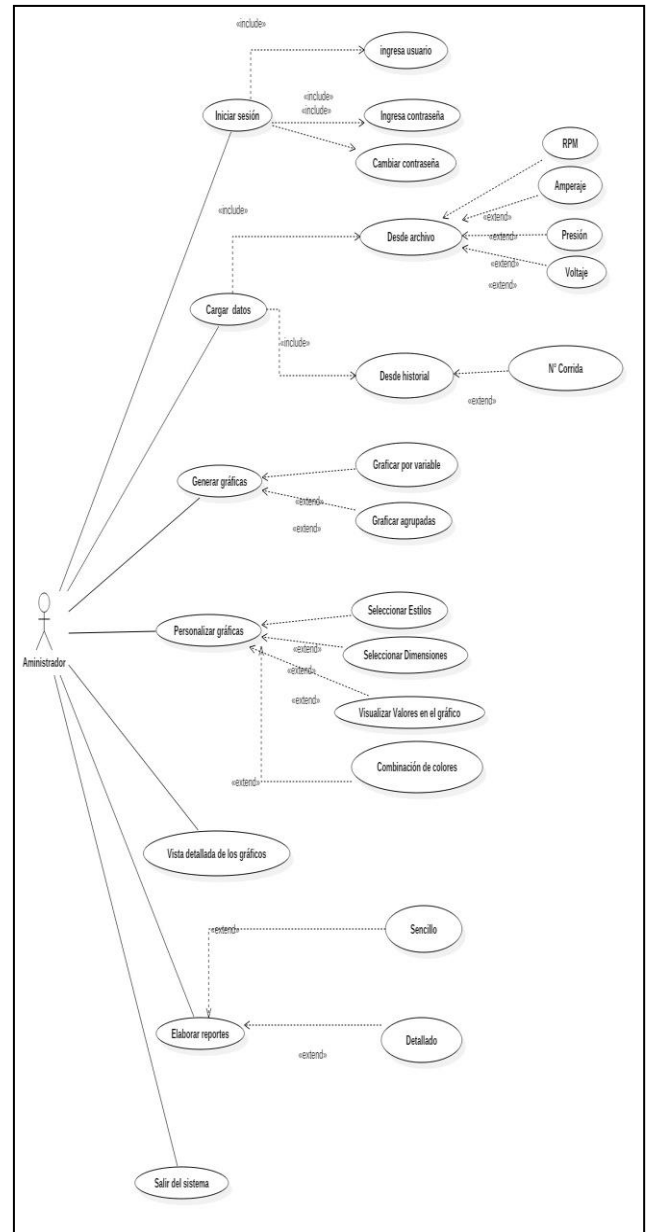


Figura 2. Diagrama de casos de uso

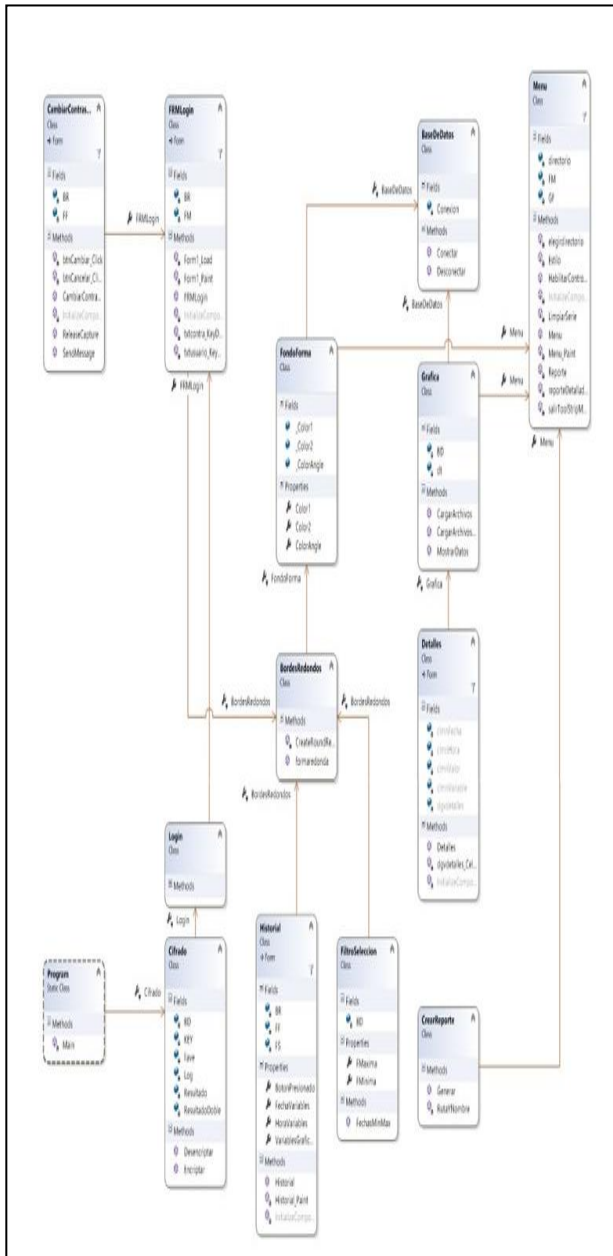


Figura 3. Diagrama de clases.

Modelo Relacional.

Mediante el Modelo Relacional mostrado en la Figura 4 podemos visualizar la forma en que se almacenará y relacionará la información manejada por el sistema. Se crearon las siguientes tablas:

- Variables.
- Login.

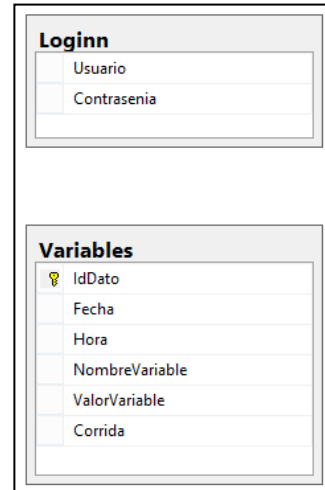


Figura 4. Esquema relacional de la base de datos.

Diccionario de Datos de las Tablas principales.

Mediante el Diccionario de datos, podemos listar las características lógicas de los datos que se van a utilizar en el sistema, esto se observa en la Figura 5.

Index		
Sr	Object	Type
1	[dbo].[Datos]	Table
2	[dbo].[Loginn]	Table
3	[dbo].[Variables]	Table

Table:[dbo].[Datos]				
Nro Filas	Reservado	Tamaño de los Datos	Tamaño de los Indices	No Usado
0	0 KB	0 KB	0 KB	0 KB

Description

Sr.	Name	Datatype	Nullable	Description	Value
1	IdDato	INT	N	Es el identificador de la variable de tipo entero.	
2	NombreVariable	VARCHAR(15)	Y	En esta campo se especifica el nombre de la variable a utilizar.	
3	Fecha	DATE	Y	Se especifica la fecha en que se realizo dieta corrida.	
4	Hora	TIME	Y	Se especifica la hora en que se realizo la corrida.	
5	Dato	DECIMAL(10,2)	Y	Es el valor correspondiente a la variable con 2 decimales de precisión.	

Indeaxes

Sr.	Name	Type	Columns
1	DatosPK	Clustered	IdDato

Table:[dbo].[Loginn]				
Nro Filas	Reservado	Tamaño de los Datos	Tamaño de los Indices	No Usado
0	0 KB	0 KB	0 KB	0 KB

Description

Sr.	Name	Datatype	Nullable	Description	Value
1	Usuario	VARCHAR(30)	Y	El campo usuario guarda un nombre que se le especifica.	
2	Contra	TEXT	Y	Se especifica una contraseña correspondiente al usuario.	

Figura 5. Diccionario de datos de tablas principales del Sistema

Conclusiones

El proceso de análisis y diseño de este sistema, permitirá la codificación e implementación exitosa del software para el almacenamiento y tratamiento de los datos mencionados, que consiste en: respaldo de la información obtenida de la HMI a través de un dispositivo externo de almacenamiento, selección de lecturas por fecha y/o por tipo de variable, acceso inmediato a registros de lecturas almacenados anteriormente, visualización dinámica de gráficas de los valores de las variables mencionadas, con la opción de selección de una hasta la totalidad de éstas, generación de reportes dinámicos.



Juan Martín Arzola Monreal. Nació en Ciudad Lerdo, Durango. México, es Licenciado en Informática. Egresado del Instituto Tecnológico Superior de Lerdo. Ciudad Lerdo, Durango. México (2002). Estudió la Maestría en Sistemas Computacionales. Instituto Tecnológico de la Laguna. Torreón, Coahuila. México. (2007). Actualmente es docente del Instituto Tecnológico Superior de Lerdo, Ciudad Lerdo. Durango. México. Sus áreas de interés son las redes computacionales, investigación de operaciones y el diseño CAD y vectorial.

III. AGRADECIMIENTOS

Nuestro agradecimiento es para el Instituto Tecnológico Superior de Lerdo por su apoyo a este proyecto.

IV. REFERENCIAS

- [1] Chatfield, C., Johnson, T. (2013). Microsoft Project 2013, Step by Step. Washington: Microsoft Press.
- [2] Microsoft. (2010). Microsoft Project 2010: guía interactiva de la correspondencia entre los menús y la cinta de opciones.
- [3] Sommerville, I. (2005). Ingeniería del Software. España: Pearson.
- [4] Pressman, R. S. (1993). Ingeniería del Software Un Enfoque Práctico. Madrid, España: Mc Graw Hill.
- [5] Schmuller, J. (2001.). Aprendiendo UML en 24 horas. Prentice Hall.

V. BIOGRAFÍA



María Guadalupe Flores Luévanos. Nació en Torreón, Coahuila de Zaragoza, México, es Ingeniero en Sistemas Computacionales. Egresada del Instituto Tecnológico de la Laguna. Torreón, Coah., México (2002). Estudió la Maestría en Administración. Universidad Autónoma de Coahuila. Torreón, Coah., México. (2004).

Ella es docente del Instituto Tecnológico Superior de Lerdo, Cd. Lerdo. Durango, México. Sus áreas de interés son el Desarrollo software en el área de Investigación.



Elda Moreno Núñez. Nació en Gómez Palacio, Durango, México, es Ingeniero en Sistemas Computacionales. Egresada del Instituto Tecnológico de la Laguna. Torreón, Coah., México (1995). Estudió la Maestría en Administración. Universidad Autónoma de Coahuila. Torreón, Coah., México. (2004).

Actualmente es docente del Instituto Tecnológico Superior de Lerdo, Cd. Lerdo. Durango, México. Sus áreas de interés son la ingeniería de software y los sistemas geográficos.