

Universidad de Cienfuegos “Carlos Rafael Rodríguez”

Facultad de Informática

Carrera de Ingeniería Informática



Titulo: “Sistema Informático para la Gestión de la información
del laboratorio industrial del CAI Antonio Sánchez
en período de zafra”

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

Autor

Yurdani Martínez Gil

Tutor

MsC. Oscar J. Alejo Machado

Consultante

MsC. Oscar Luís Muñoz

Cienfuegos, Cuba

Curso 2008 - 2009

Declaración de autoría

Declaro que yo Yurdani Martínez Gil soy el único autor de este trabajo y autorizo al Laboratorio Industrial del CAI Antonio Sánchez y a la Facultad de Informática de la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo la presente a los 9 días del mes de 7 del 2009.

Yurdani Martínez Gil
Nombre completo del primer autor

Ms.C. Oscar Alejo Machado
Nombre completo del primer tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

Firma Autor

Firma Tutor

Firma ICT

Firma Vicedecano

Agradecimientos

A mis padres por todo el apoyo que me han brindado para poder realizar este sueño.

A mi familia por haber estado atentos de mis estudios.

A mi amigo Juan que se a portado como un padre y tanto le debo esta carrera.

A mis compañeros de trabajo y estudio por todo el aliento que me supieron dar para poder llegar a la recta final.

A todos mis amigos y profesores que de una forma o de otra han apoyado para el desarrollo y culminación de esta carrera.

A la revolución y la tarea Álvaro Reinoso por la oportunidad de dejar realizar mi sueño.

A todos los antes Mencionados, hagan suyo estos resultados por haber contribuido de una forma u otra a la realización del mismo.

DEDICATORIA

*A mis dos niñas queridas,
Que son Toda mi fuente de inspiración.*

Resumen

El presente trabajo investigativo (**SIGILIAS**) se enmarca en un **Sistema Informático** para la **Gestión de la Información del Laboratorio Industrial del CAI Antonio Sánchez** en período de zafra; esta información incluye todos los análisis que se le realizan a la caña y a todos los materiales intermedios que se desarrollan a lo largo del proceso de producción de azúcar, como a todos los materiales o productos finales. Este sistema permite agilizar el desempeño de los químicos, calculista y jefe de laboratorio, así como ayudar a los diferentes dirigentes y técnicos en la toma de decisiones posteriores pudiendo corregir los errores posibles en el proceso de producción de azúcar. Por otra parte llevar un control estricto de las paradas diarias de la industria con el objetivo de elaborar el plan de reparaciones y lograr una disminución de las mismas en las zafra futuras, para el desarrollo del sistema se utilizó la metodología RUP y en la confección de la solución propuesta se utilizó C++ como lenguaje de programación de alto nivel.

Índice

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	4
1.1 Introducción.	4
1.1.6 Sistemas Existentes o Vinculados.	5
1.2 Tendencias tecnologías actuales a tener en cuenta.	5
1.2.1 Introducción.	5
1.2.2 Metodologías y herramientas utilizadas.	6
1.2.3 Fundamentación del lenguaje y software utilizado.....	12
1. 3 Conclusiones.	14
Capítulo 2: Construcción de la solución propuesta	15
2.1 Modelo del dominio.....	15
2.1.1 Introducción.	15
2.1.2 Descripción del modelo del dominio.....	15
2.1.3 Reglas del negocio a considerar	15
2.1.4 Representación del modelo del dominio	16
2.2 Modelo de sistema.....	17
2.2.1Introducción.	17
2.2.2 Descripción del modelo del sistema.....	17
2.2.3 Modelación del modelo de sistema.....	18
2.2.4 Requerimientos funcionales.....	18
2.2.5 Requerimientos no funcionales	22
2.2.6 Actores del modelo de sistema	25
2.2.7 Diagramas de casos de uso del sistema.....	26
2.2.8 – Descripción de los casos de usos del sistema.....	28
2.3 – Construcción del sistema.....	60
2.3.1 – Diagrama de clases del diseño	61
2.3.2 – Diagramas del modelo lógico de datos	61
2.3.3 – Diagramas del modelo físico de datos	62
2.3.4 – Diagramas de implementación.....	62
2.4 – Principios de diseño del sistema.....	63

2.4.1 - Estándares en la interfaz de la aplicación.....	63
2.4.2 - Concepción general de la ayuda.....	64
2.4.3 - Estándares de codificación.	64
2.5 – Conclusiones	66
Capítulo 3: Estudio de Factibilidad	67
3.1 Introducción.	67
3.2 Planificación.....	67
3.3 Costos.....	74
3.4 Beneficios tangibles e intangibles.	77
3.5 Análisis de costos y beneficios.	78
3.6 Conclusiones.	79
Conclusiones.....	80
Recomendaciones.....	81
Referencias Bibliográficas	82
Bibliografía.	83
Glosario de Términos	84

Introducción

Como todos conocen la industria azucarera surgió en Cuba en la Etapa Colonial (España) desarrollándose esta en la etapa Neo-colonial (EE-UU). En todo este periodo fue uno de las principales industrias del País y continuo siendo después del triunfo de la revolución. En la actualidad aunque se han desarrollado otras esferas sigue siendo de gran importancia.

En la provincia contamos con siete centrales activos ya que esta es una de las provincias de gran auge en este sentido y uno de estos y el más diversificado es el central “Antonio Sánchez” ya que además de azúcar el obtiene otros derivados como son: Alcohol, Ron, CO₂ y Levadura Torula. Esto demanda un laboratorio capaz de aplicar los métodos analíticos que determinen los parámetros acordados. Se tiene como **situación problemática** que la información del laboratorio Industrial del central azucarero “Antonio Sánchez” en Periodo de zafra se realiza de forma manual. Este flujo informativo requiere de un gran consumo de tiempo y es vulnerable a errores. A esto se le agrega que no se lleva un control riguroso de las paradas de la industria, para preparar el plan de reparaciones y el plan de capacitación. Por lo que se plantea como problema científico de esta investigación arroja la siguiente interrogante.

Teniéndose como **problema científico** la carencia de un sistema informático que permita la gestión de la información del laboratorio industrial del central Antonio Sánchez de modo más eficiente, disminuyendo la vulnerabilidad a los errores en el flujo informático.

El **objeto de estudio** de este trabajo se centra en la información que se genera en un CAI en período de zafra, y como **campo de acción** la gestión de la información del laboratorio industrial del CAI Antonio Sánchez en período de zafra.

Trazándose como **idea a defender** que con la confección y utilización de un software que gestione la información del laboratorio del CAI “Antonio Sánchez” se erradicarán los errores en el flujo informativo de este laboratorio y se contribuirá a la inmediatez de la información procesada.

Para esto se tiene como **objetivo general** Elaborar un sistema informático que permita gestionar de modo eficiente el flujo informativo relacionado con los datos del laboratorio industrial del CAI “Antonio Sánchez”.

Teniéndose como **objetivos específicos**:

- Analizar los elementos a automatizar en el sistema.
- Diseñar los elementos a automatizar en el sistema.
- Implementar el sistema.

Para el cumplimiento de éstos objetivos se trazaron las siguientes **tareas científicas**:

- Consultar y entrevistar tanto al personal que trabaja en el procesamiento y desarrollo de los análisis del laboratorio, como al personal que desarrolla los cálculos secundarios de los mismos.
- Recopilación bibliográfica relacionada con el marco teórico de la investigación.
- Definición de las estructuras de datos necesarias, definición de clases, diseño e implementación del software.
- Análisis y Diseño del Sistema propuesto.
- Implementación del Sistema propuesto.

Como **aportes prácticos** se tiene que:

- Brindará una herramienta capaz de gestionar la información del laboratorio industrial del CAI Antonio Sánchez en período de zafra.
- Esta aplicación constituye desde el punto de vista informático un primer peldaño en el perfeccionamiento empresarial para dicha institución;

pues implantará las bases para un desarrollo estratégico, que persigue la digitalización de la información en la empresa.

El presente Documento se encuentra estructurado básicamente en resumen, introducción, tres capítulos y conclusiones. De los capítulos se muestra a continuación su contenido de forma resumida:

Capítulo 1: *Fundamentación Teórica.* Fundamenta la base teórica del tema a desarrollar. Plantea los objetivos generales de la tarea propuesta, se describen los sistemas existentes vinculados al campo de acción y se especifican las tareas para el cumplimiento de los objetivos. Se hace referencia a las diferentes tendencias y metodologías existentes en la actualidad, se describen sus características esenciales y se expresa cuales fueron seleccionadas para la modelación e implementación del sistema propuesto.

Capítulo 2: *Construcción de la Solución Propuesta.* Se describe, a través del Modelo de Dominio, una representación de la solución propuesta, además, se presenta una descripción detallada de las reglas de negocio que el objeto de automatización debe seguir para asegurar el cumplimiento de las restricciones que existen en el dominio. Se proyectan los requerimientos funcionales y no funcionales, se definen los actores del sistema, los casos de uso del sistema y la descripción de los mismos, así como los diagramas de clases. Y se plantean los detalles relacionados con el diseño del sistema que se propone.

Capítulo 3: *Estudio de factibilidad.* Se describe lo relacionado con la planificación, costo, beneficios tangibles e intangibles, análisis de costo y beneficios en el desarrollo de la aplicación a desarrollar.

Capítulo 1: Fundamentación teórica

1.1 Introducción.

Durante los últimos años se ha incrementado la exigencia de la calidad, tanto en los productos destinados al consumidor como los que sirven de materia prima para otras industrias. Este hecho reviste tremenda importancia socioeconómica, en especial para los países en vías de desarrollo, ya que origina una alta competitividad en la lucha por los mercados. Esta realidad constituye una llamada de alerta al productor, resumida en la alternativa: producir con calidad que demanda el mercado o sucumbir.

La experiencia global, necesita de todos los procesos productivos, enseña que es imposible optimizar la calidad sin la previa optimización del control del proceso que la origina; y en la industria azucarera, por su complejidad, esta enseñanza requiere la más absoluta observancia. El control de la calidad, además de una vieja tradición, altísima disciplina y gran profesionalismo, requiere de medios y métodos que, por su eficiencia y actualidad, permitan alcanzar este objetivo. Por lo que un control adecuado de la calidad requiere un laboratorio capaz de aplicar los métodos analíticos que determinen los parámetros acordados.

Generalmente muchos de los métodos de ensayos desarrollados requieren de un gran consumo de tiempo. y si se detecta alguna anomalía en el proceso la solución a esta llegara con un determinado tiempo de retraso. Pero si se disminuye el tiempo de cálculo de algunos de estos ensayos y técnicas analíticas, se logra una mejor eficacia en todo este proceso, ya que se podrá realizar un mayor numero de estos y eso implica más control y por ende más calidad.

Por otra parte estos cálculos que algunos son extensos y otros que son sencillos pero reiterados pueden comprender un error humano que influye negativamente en el proceso y por ende en la calidad del proceso y producto final.

Otra de las cosas que afecta la calidad son las reiteradas paradas de la industria que generalmente son ocasionadas principalmente por dos razones:

- 1 Mala Reparación de la Fábrica.
- 2 Mala Capacitación del personal.

Por lo que en el periodo de no zafra se desarrollan dos planes importantes para erradicar estos problemas, el plan de reparaciones y el plan de capacitación del personal. Ambos planes se desarrollan a partir de la Norma Técnica 37 (NT37).

1.1.6 Sistemas Existentes o Vinculados.

Existe un sistema implantado en todas las industrias, ***IPLUS*** (***Industrial Plus***). que permite el cierre diario de los indicadores de este departamento o lo que es lo mismo la contabilidad de laboratorio del central azucarero, pero este se limita a los datos diarios de cierre, ósea después de haberse realizado un sin numero de cálculos pequeños y muy reiterados en el resto del día que cuenta con una ponderación de un sin numero de indicadores y una totalización de otro numero, cálculos que se realizan muy a menudo y representan malgasto de tiempo e incremento de errores que influyen en la toma de decisiones de forma correcta.

1.2 Tendencias tecnologías actuales a tener en cuenta.

1.2.1 Introducción.

En este capítulo hago referencia a las diferentes tecnologías y metodologías a utilizar para la creación del software que propongo, de ellas se describen sus características esenciales y se fundamenta el motivo por el cual fueron seleccionadas algunas de ellas para la modelación e implementación del sistema propuesto.

1.2.2 Metodologías y herramientas utilizadas.

En aras de garantizar calidad en el desarrollo de una herramienta de software es necesario e importante seguir las indicaciones de alguna metodología. Como, hacer un estudio de cuales son las tecnologías actuales, conocidas o no, con el fin de seleccionar y utilizar la más conveniente. En este capítulo se exponen las características fundamentales de la metodología más conocida por el autor de este trabajo, para el desarrollo de software, el Proceso Unificado de Desarrollo. Se justifican las razones por las cuales ha sido seleccionada esta para guiar el proceso de desarrollo de la herramienta propuesta. Adicionalmente se hace referencia a dos gestores de bases de datos y se explican sus principales funcionalidades, constituyendo un paso obligatorio en el proceso investigativo para la selección adecuada del sistema a utilizar. Además, se describen las características fundamentales de algunas de las tecnologías existentes para el desarrollo de software justificando la seleccionada para la implementación de dicha propuesta.

UML (Lenguaje de Modelamiento Unificado).

El Lenguaje de Modelación Unificado (UML – Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. (Craig Larman)

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelación visual que permite una abstracción del sistema y sus componentes.

Entre sus objetivos fundamentales se encuentran:

1. Ser tan simple como sea posible, pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.
2. Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son el encapsulamiento y los componentes.
3. Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
4. Imponer un estándar mundial.

UML surge como respuesta al primer problema reseñado para contar con un lenguaje estándar que permita escribir planos de software. Muchos han creído ver UML como solución para todos sus problemas sin saber en muchos casos de lo que se trataba en realidad.

El Lenguaje Unificado de Modelado, UML es una notación estándar para el modelado de sistemas software, resultado de una propuesta de estandarización promovida por el consorcio OMG (Object Management Group), del cual forman parte las empresas más importantes que se dedican al desarrollo de software en 1996.

UML es una notación, es decir, de una serie de reglas y recomendaciones para representar modelos. Permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos y entre sus características principales tenemos:

1. Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
2. Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos.
3. Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
4. Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
5. Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
6. Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
7. UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado de Desarrollo, fue creado por el mismo grupo de expertos que crearon UML, Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998.

El objetivo que se perseguía con esta metodología era producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Como se expresaba anteriormente, esta

metodología concibió desde sus inicios el uso de UML como lenguaje de modelado.

Es un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

Como Son:

1. Desarrollo de software en forma iterativa.
2. Manejo de requerimientos.
3. Utiliza arquitectura basada en componentes.
4. Modela el software visualmente.
5. Verifica la calidad del software.
6. Controla los cambios.

Para desarrollar el trabajo de RUP y UML se ha creado en el año 2000 una herramienta denominada CASE (Computer Assisted Software Engineering) por la compañía norteamericana Rational Corporation. Esta aplicación llamada Rational Rose completa todos los elementos que compone la metodología para cubrir el ciclo de vida de un proyecto y facilita el trabajo con RUP y UML para los desarrolladores de software. Por estas razones es que se ha resuelto el uso de estos instrumentos de trabajo en nuestra investigación y se utiliza Rational Rose como asistente en su versión del 2003.

Sistema Gestor de bases de datos utilizado.

Como proceso indispensable en el diseño del software propuesto procedí al análisis y comparación entre dos diferentes gestores de bases de datos de los

más utilizados dando la posibilidad de elegir el más óptimo para llevar a cabo el trabajo propuesto. Estas plataformas son: Microsoft SQL Server y Microsoft Access.

Microsoft SQL–Server.

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales con arquitectura cliente/servidor el cual está altamente integrada con el sistema operativo Windows y basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Este sistema utiliza Transact–SQL, para enviar las peticiones entre sí y el cliente. Con Transact–SQL se puede acceder a los datos, consultarlos, actualizarlos y gestionar sistemas de bases de datos relacionales. Usando SQL Server se pueden desarrollar aplicaciones modernas que separan la aplicación cliente de las bases de datos.

Entre sus características figuran:

1. Soporte de transacciones.
2. Gran estabilidad.
3. Gran seguridad.
4. Escalabilidad.
5. Soporta procedimientos almacenados.
6. Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
7. Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
8. Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET.

Este gestor, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

El Instituto Nacional Americano de Estándares (ANSI) y la Organización internacional de Estándares (ISO) han definido sus estándares para el SQL. El Transact-SQL soporta los últimos estándares ANSI SQL publicados además de muchas extensiones que le proveen elevadas funcionalidades.

Entre las ventajas del SQL Server se encuentran los procedimientos almacenados, Los disparadores, el número tan elevado de conexiones y transacciones que soporta y el tamaño de las bases de datos pueden alcanzar los terabytes. Su rendimiento depende solamente del hardware con que se cuente y corre sobre plataforma de Windows NT–2003 Server. En el caso del SQL– Server 2005.

Microsoft Access.

Microsoft Access es un sistema de gestión de bases de datos creado y modificado por Microsoft para uso personal o de pequeñas organizaciones. Es un componente de la suite Microsoft Office aunque no se incluye en el paquete básico. Su principal función es ser una potente base de datos, capaz de trabajar en sí misma o bien con conexión hacia otros lenguajes de programación, tales

como Visual Basic 6.0 o Visual Basic .NET. Pueden realizarse consultas directas a las tablas contenidas mediante instrucciones SQL. Internamente trae consigo el lenguaje Visual Basic for Application el cual es similar en forma al Visual Basic 6.

Permite el ingreso de datos de tipos: Numéricos, Texto, Fecha, Sí/No, OLE, Moneda, Memo y Boolean. Pueden desarrollarse aplicaciones completas basadas en Microsoft Access, pues trae consigo las herramientas necesarias para el diseño y desarrollo de formularios para el ingreso y trabajo con datos e informes para visualizar e imprimir la información requerida. Su funcionamiento se basa en un motor llamado Microsoft Jet, y permite el desarrollo de pequeñas aplicaciones autónomas formadas por formularios Windows y código VBA (Visual Basic para Aplicaciones). Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. Entre las principales funcionalidades de Access se encuentran:

1. Crear tablas de datos indexadas.
2. Modificar tablas de datos.
3. Relaciones entre tablas (creación de bases de datos relacionales).
4. Creación de consultas y vistas.
5. Consultas referencias cruzadas.
6. Consultas de acción (INSERT, DELETE, UPDATE).
7. Formularios.
8. Informes.
9. Llamadas a la API de windows.
10. Interacción con otras aplicaciones que usen VBA (resto de aplicaciones de Microsoft Office, Autocad, etc.).
11. Macros.

1.2.3 Fundamentación del lenguaje y software utilizado.

Antes de llevar a cabo el desarrollo del software propuesto se realizó por parte del autor un estudio de dos de las tecnologías, tanto lenguajes como herramientas de desarrollo existentes. En este análisis se profundizó en aquellas que han sido utilizadas en la elaboración del software del tipo propuesto. Estas plataformas son: Borland Delphi 7 y Borland C++ las que serán descritas a continuación:

Borland Delphi v 7.0.

El Object Oriented Pascal es el lenguaje que Delphi utiliza para crear las aplicaciones orientadas a objetos. Debido a que Delphi pertenece a la empresa Borland, la potencia de éste puede compararse con el compilador de C++. **(Pero no es el compilador C++)** Borland Delphi es un ambiente de desarrollo rápido de aplicaciones (RAD) muy flexible y fácil de usar. Estos últimos años ha tenido una gran repercusión dentro del mundo de la programación visual.

Presenta un ambiente visual de desarrollo para aplicaciones controlados por eventos de usuario sobre interfaces gráficas. Proporciona una jerarquía muy extensa de clases de objetos re-usables. En cuanto a información sobre técnicas de programación en Delphi, existe un vasto de opciones a elegir, tales como miles de páginas web, muchos foros de debate, sitios FTP que contienen una enorme cantidad de librerías, y mucha más información que puede ser obtenida a través de Internet. Delphi es una herramienta de propósito general, se puede programar tanto a bajo nivel, como a alto nivel (simplemente usando controles y ajustando propiedades) y tiene buenas capacidades gráficas. Las aplicaciones creadas en Delphi solo funcionan sobre la plataforma de trabajo Windows.

Borland C++ Builder v 6.0.

Es el homologo del Delphi pero con estructura de clases de C++, por lo que tiene las mismas características descritas en el Delphi solo que el compilador es de C++ y permite la importación de código C++ existente. Posee una mayor cantidad de

clases y objetos re-usables. Existe mucha mas documentación referente a la programación en esta plataforma.

1. 3 Conclusiones.

Con el diseño de este software se logró agilizar, consolidar y viabilizar toda la información del laboratorio del central azucarero, en aras de apoyar en la toma de decisiones de una forma más rápida con el fin de aumentar la cantidad de ensayos de laboratorio diario y con una mejor confiabilidad de los mismo para poder mejorar la calidad del producto final (azúcar).

En este capitulo damos por sentado las bases de por que se escoge UML como metodología a seguir en la documentación del software propuesto y en su proceso de desarrollo a RUP, como gestor de bases de datos el SQL Server 2005 y para la implementación el lenguaje C++, utilizando como herramienta de programación el Borland C++ Builder v. 6.0.

Capítulo 2: Construcción de la solución propuesta

2.1 Modelo del dominio

2.1.1 Introducción.

En el presente capítulo tomando como guía la metodología RUP, se utiliza uno de los artefactos que brinda dicha metodología: el modelo de dominio, el cual ayuda a modelar y describir la solución propuesta. Además, se presenta una descripción del modelo de objeto del dominio, así como el enunciado y descripción de sus reglas para garantizar las restricciones existentes en el mismo.

2.1.2 Descripción del modelo del dominio.

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos. La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema.

En el modelo del dominio empleado en este trabajo se definen las siguientes entidades y clases principales: Usuarios, Codificadores, Diarios y Reportes. Una vez que el usuario codifica el sistema al iniciar la zafra, La esencia del funcionamiento de este dominio radica en la constante labor de este, desarrollando los ensayos y cálculos indispensables de los mismos con el objetivo de optimizar su uso y disminuir los tiempos de desarrollo de las operaciones. Para poder hacer uso de los reportes para la toma de dediciones de una forma más rápida y efectiva.

2.1.3 Reglas del negocio a considerar

Las reglas de negocio regulan y describen las principales políticas que deben cumplirse para el adecuado funcionamiento del negocio.

- La información necesaria para el procesamiento debe ser lo más real posible y adecuarse a las normas regidas en los laboratorios azucareros.
- Para llevar a cabo el trabajo de recopilación de la información se deben cumplir lo mas estrictamente con el esquema de control diseñado.
- Repetir los ensayos, pertinentes en caso de no estar de acuerdo con los resultados para tener una mejor confiabilidad.

2.1.4 Representación del modelo del dominio

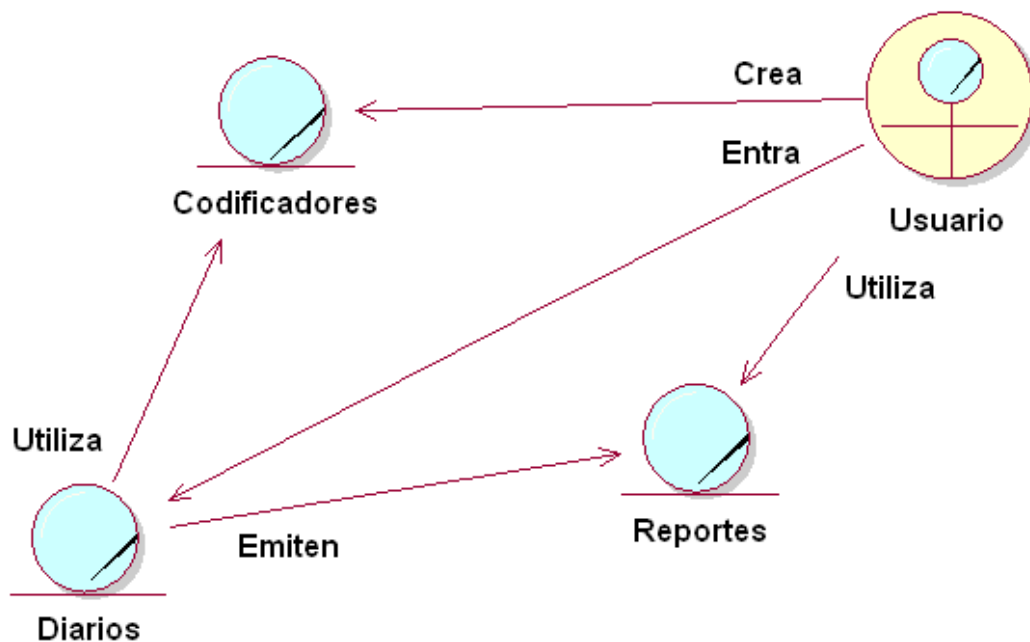


Figura 2.1 Diagrama de clases del modelo de objetos del dominio

2.2 Modelo de sistema.

2.2.1 Introducción.

En el presente capítulo se describe y analiza el modelo de sistema del objeto de automatización sobre la base de las especificaciones de la metodología RUP. Se identifican los requerimientos funcionales y no funcionales, se definen a los actores del sistema y a los servicios o funcionalidades que a su disposición se colocan (los casos de uso del sistema).

Además, se plantean y detallan una serie de diagramas que nos ayudan y guían en la implementación del modelo de sistema, como son: el diagrama de casos de uso del sistema, el diagrama de clases del diseño, el diagrama del modelo físico y lógico de datos y el diagrama de implementación.

2.2.2 Descripción del modelo del sistema.

El sistema propuesto tiene como objetivo la gestión de la información del laboratorio de un central azucarero durante el periodo de zafra y controlar las paradas ocasionadas para una buena elaboración del plan de reparaciones y el de capacitación. Para llevar a cabo esta tarea se necesita la evaluación de un gran volumen de codificadores interrelacionados que elaboran el esquema de control de los diversos indicadores a gestionar entre ellos tenemos:

1. Información diaria de los ensayos del laboratorio y esquema de control.

- Grupos de Control.
- Objetos a Controlar.
- Puntos de Muestreos.
- Tipos de Muestreos.
- Análisis a Realizar.
- Secciones.
- Depósitos y Tipos de Depósitos.
- Horas de Trabajo.

- Departamentos.
 - Áreas.
 - Brigadas.
 - Turnos de Trabajo.
2. Información diaria de Tiempo Perdido.
- NT37 Codificador Principal de Tiempo Perdido.
 - NT37 Codificador General de Tiempo Perdido.
3. Codificador de vinculación del Esquema de Control vs datos de entrada diarios del Sistema IPLUS (Industrial Plus) Sistema de Procesamiento Diario usado Actualmente.

2.2.3 Modelación del modelo de sistema

El modelado de casos de uso es la técnica más efectiva y a la vez la más simple que emplean los desarrolladores de software para modelar los requisitos del sistema desde la perspectiva del usuario. El modelo de casos de uso consiste en actores y casos de uso. Los actores representan usuarios y otros sistemas que interaccionan con el sistema y los casos de uso representan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor. En esencia, el modelado de Casos de Uso nos describe lo que hace el sistema para cada tipo de usuario y nos ofrece un medio correcto para el análisis, el diseño y las pruebas.

2.2.4 Requerimientos funcionales

Los requerimientos funcionales son declaraciones de los servicios o funciones que Proveerá el sistema, de la manera en que éste reaccionará a entradas particulares. Estos dependen del tipo de software y del sistema que se desarrolle y de los posibles usuarios del software. Cuando se expresan como requerimientos del usuario, habitualmente se describen de forma general mientras que los

requerimientos funcionales del sistema describen con detalle la función de éste, sus entradas y salidas, excepciones, etc.

En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer.

Listado de los requerimientos funcionales:

- Modificar los datos de la empresa.
- Insertar Departamento.
- Modificar Departamento.
- Eliminar Departamento.
- Listar Departamentos.
- Insertar Área.
- Modificar Área.
- Eliminar Área.
- Listar Áreas.
- Insertar Brigada.
- Modificar Brigada.
- Eliminar Brigada.
- Listar Brigadas.
- Insertar Turno Trabajo.
- Modificar Turno Trabajo.
- Eliminar Turno de Trabajo.
- Listar Turnos de Trabajo.
- Insertar Hora.
- Modificar Hora.
- Eliminar Hora.
- Listar Horas.
- Insertar Sección.
- Modificar Sección.

- Eliminar Sección.
- Listar Secciones.
- Insertar Tipo de Deposito.
- Modificar Tipo de Deposito.
- Eliminar Tipo de Deposito.
- Listar Tipos de Depósitos.
- Insertar Depósito.
- Modificar Depósito.
- Eliminar Depósito.
- Listar Depósitos.
- Insertar Grupo de Control.
- Modificar Grupo de Control.
- Eliminar Grupo de Control.
- Listar Grupos de Control.
- Insertar Objeto a Controlar.
- Modificar Objeto a Controlar.
- Eliminar Objeto a Controlar.
- Listar Objetos a Controlar.
- Insertar Análisis.
- Modificar Análisis.
- Eliminar Análisis.
- Listar Análisis.
- Insertar Punto de Muestreo.
- Modificar Punto de Muestreo.
- Eliminar Punto de Muestreo.
- Listar Puntos de Muestreo.
- Insertar Tipo de Muestreo.
- Modificar Tipo de Muestreo.
- Eliminar Tipo de Muestreo.
- Listar Tipos de Muestreos.

- Insertar NT37 Principal.
- Modificar NT37 Principal.
- Eliminar NT37 Principal.
- Listar NT37 Principales.
- Insertar NT37 General.
- Modificar NT37 General.
- Eliminar NT37 General.
- Listar NT37 Generales.
- Insertar Brix.
- Modificar Brix.
- Eliminar Brix.
- Listar Brix.
- Insertar Humedad.
- Modificar Humedad.
- Eliminar Humedad.
- Listar Humedades.
- Insertar Relación Brix/Humedad.
- Modificar Relación Brix/Humedad.
- Eliminar Relación Brix/Humedad.
- Listar Relaciones Brix/Humedad.
- Insertar Esquema de Control.
- Modificar Esquema de Control.
- Eliminar Esquema de Control.
- Listar Esquema de Control.
- Insertar Categoría.
- Modificar Categoría.
- Eliminar Categoría.
- Listar Categorías.
- Insertar Vinculo IPLUS vs Esquema de Control.
- Modificar Vinculo IPLUS vs Esquema de Control.

- Eliminar Vinculo IPLUS vs Esquema de Control.
- Listar Vínculos IPLUS vs Esquema de Control.
- Insertar Corrección Brix.
- Modificar Corrección Brix.
- Eliminar Corrección Brix.
- Listar Correcciones Brix.
- Activar Día Zafra.
- Agregar Día Zafra.
- Listar días Zafra.
- Establecer Turnos.
- Listado de Turnos.
- Capturar Paradas Diarias.
- Capturar Análisis Diarios.
- Visualizar NT37.
- Listar la NT37.
- Visualizar Datos Diarios del Laboratorio.
- Listar Datos Diarios del Laboratorio.
- Corregir Pol en Bagazo.
- Corregir Brix.
- Visualizar Entradas IPLUS.
- Listar Entradas IPLUS.

2.2.5 Requerimientos no funcionales

Los requerimientos no funcionales describen las restricciones del sistema o del proceso de desarrollo; no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida, en cuanto a prestaciones, atributos de calidad y la representación de datos que se utiliza en la interfaz del sistema.

Listado de los requerimientos no funcionales del sistema

Requisitos de usabilidad.

1. La información original del software sólo podrá ser modificada por especialistas en la materia. Se le permitirá al usuario introducir modificaciones en los valores originales. Pero queda bajo responsabilidad del usuario la creación de las formulas del sistema.
2. La explotación del sistema agilizará la actividad científica del laboratorio donde se ponga en práctica gracias a las comodidades brindadas por esta aplicación, la rapidez y organización, presentado en sus cálculos.

Requisitos de Rendimiento

1. Se requiere de una capacidad de procesamiento alta para ejecutar algoritmos complejos.

Apariencia o interfaz externa.

1. La interfaz se ajustará al estándar de ventanas que el sistema operativo Windows ha establecido e internacionalizado.
2. Tendrá una documentación básica que comprenda los aspectos generales a tener en cuenta para trabajar con la aplicación.
3. La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. Se cuidará porque la aplicación sea lo más interactiva posible. La interfaz será uniforme.

CAPÍTULO 2: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

4. El sistema podrá ser usado por aquellas personas que no tengan experiencia en el uso de la computadora, sólo necesitarían un ligero entrenamiento sobre el funcionamiento de los principales elementos de una interfaz estándar en el ambiente del sistema operativo Windows (uso del Mouse, manejo de menús, botones, cuadros de texto, etc.)

Requisitos de Soporte

- Las pruebas del sistema se realizarán en la Empresa Azucarera “Antonio Sánchez”. Dichas pruebas permitirán evaluar en la práctica la funcionalidad y las ventajas de este nuevo producto.
- El sistema deberá dar las posibilidades a futuras mejoras y nuevas opciones que se le quieran incorporar.

Requisitos Legales

- Este software es propiedad intelectual de la Universidad de Cienfuegos y solo es permitida su comercialización con el consentimiento del autor y la entidad involucrada.

Requisitos de Portabilidad.

- La plataforma seleccionada para desarrollar la aplicación fue Windows

Requisitos de Confiabilidad

- Es importante que el sistema presente un mecanismo de respuesta rápida ante fallos y que en caso de ocurrencia se minimicen las pérdidas de información.

Requisitos de Ayuda y Documentación en Línea

CAPÍTULO 2: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

- Debe disponerse de una ayuda bien detallada sobre las principales opciones del sistema. Además, se debe tener disponible otros documentos para consulta general.

Requerimiento de Hardware (Mínimos)

- Procesador Pentium
- 128 Mb. de RAM
- Mouse.
- 1 Gb. de HDD Libre.
- UPS o fuente de corriente ininterrumpida.

Requerimiento de Software

- Se debe disponer de Windows 95 o superior, para la instalación de la aplicación.

2.2.6 Actores del modelo de sistema

Un actor es aquel que interactúa con el sistema, sin ser parte de él y puede asumir el rol que juega una o varias personas, un equipo o un sistema automatizado.

A continuación se define el actor identificado en el sistema propuesto:

Actor	Descripción
-------	-------------

Usuario	Cualquier usuario que interactúe con el sistema, puede ser un Químico de Turno, 1er Químico, Jefe de turno o Jefe de laboratorio. Este usuario tendrá acceso a todos los requerimientos funcionales del sistema.
---------	--

Tabla1 Descripción de los actores del sistema.

2.2.7 Diagramas de casos de uso del sistema

Los actores interactúan y usan el sistema a través de casos de uso. Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. En nuestro trabajo los casos de uso del sistema quedan representados por:

1. Modificar Datos de la empresa.
2. Definir los Departamentos de la Empresa.
3. Definir las Áreas de la Industria.
4. Definir las Brigadas de las Industria.
5. Definir los Turnos de Trabajo.
6. Definir las Horas.
7. Definir las Secciones de Tiempo.
8. Definir los Tipos de depósitos.
9. Definir los Depósitos.
10. Definir los Grupos de Control.
11. Definir los Objetos a Controlar.
12. Definir los Análisis a Realizar
13. Definir los puntos de muestreo.
14. Definir los Tipos de Muestreos.
15. Definir la NT37 Codificador Principal.
16. Definir la NT37 Codificador General.
17. Definir el listado de Brix más Frecuentes.

CAPÍTULO 2: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

18. Definir el Listado de las Humedades más Frecuentes.
19. Definir la Relación Humedad Vs Brix.
20. Crear el Esquema de Control
21. Definir las categorías del IPLUS.
22. Crear Relaciones IPLUS Vs GILIPZ.
23. Crear la Tabla de Corrección de Brix.
24. Activar o Crear día zafra.
25. Establecer turnos del Día.
26. Modificar las Paradas Diarias.
27. Modificar los Análisis Diarios.
28. Visualizar y reportar la NT37.
29. Visualizar y reportar datos del Laboratorio.
30. Visualizarla tabla de Entrada del IPLUS.
31. Reportar e Imprimir NT37.
32. Reportar Datos Diarios.
33. Reportar e Imprimir Tabla IPLUS.

Figura 2.2 Diagramas de casos de usos del sistema GILIPZ.

2.2.8 – Descripción de los casos de usos del sistema

CAPÍTULO 2: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Después de haber representado los casos de usos del sistema pasaremos a describirlos. Estos serán referenciados a través de los requerimientos funcionales planteados en el epígrafe 4.3.1.

Caso de Uso	Modificar datos de la empresa
Actores:	Usuario
Propósito:	Permite que el usuario modifique todos los datos que identifican a la empresa.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la modificación de los datos de la empresa, introducirá los datos solicitados en el formulario y se modificaran estos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 2. Descripción del caso de uso de Modificar datos de la Empresa.

Caso de Uso	Definir los Departamentos de la empresa
Actores:	Usuario
Propósito:	Permite que el usuario defina todos los departamentos de la Empresa.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Departamentos” y comienza a definir los departamentos en que se divide la empresa y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la	

CAPÍTULO 2: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 3. Descripción del caso de uso Definir los Departamentos de la empresa.

Caso de Uso	Definir las Áreas de la industria
Actores:	Usuario
Propósito:	Permite que el usuario defina todas las Áreas de la Industria.
<u>Resumen</u>	
El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Áreas” y comienza a definir las áreas en que se divide la industria y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 4. Descripción del caso de uso Definir Áreas de la industria.

Caso de Uso	Definir las Brigadas de la industria
Actores:	Usuario
Propósito:	Permite que el usuario defina todas las Brigadas de trabajo de la Industria.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Brigadas” y comienza a definir las diferentes brigadas que existen en la industria y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 5. Descripción del caso de uso Definir las Brigadas de la industria.

Caso de Uso	Definir los Turnos de Trabajo
Actores:	Usuario
Propósito:	Permite que el usuario defina los turnos de trabajo de la Industria.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Turnos de trabajo” y comienza a definir los turnos de trabajo de la industria y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 6. Descripción del caso de uso Definir los Turnos de Trabajo.

Caso de Uso	Definir las Horas
Actores:	Usuario
Propósito:	Permite que el usuario defina las Horas de cierre de información.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Horas” y comienza a definir las horas diarias y plasmar los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 7. Descripción del caso de uso Definir las Horas.

Caso de Uso	Definir las Secciones de tiempo
Actores:	Usuario
Propósito:	Permite que el usuario defina las Secciones de cierre de información.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Secciones” y comienza a definir las secciones y plasmar los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 8. Descripción del caso de uso Definir las Secciones de tiempo.

Caso de Uso	Definir los Tipos de Depósitos
Actores:	Usuario
Propósito:	Permite que el usuario defina los Tipos de Depósitos a Utilizar.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Tipos de Depósitos” y comienza a definir los tipos de depósitos donde se puedan encontrar los productos y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 9. Descripción del caso de uso Definir los Tipos de Depósitos.

Caso de Uso	Definir los Depósitos
Actores:	Usuario
Propósito:	Permite que el usuario defina los Depósitos a Utilizar.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Depósitos” y comienza a definir los diferentes depósitos donde se pueden encontrar los productos y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU8 (extend)
Precondiciones:	Deben estar Definidos los Tipos de Depósitos
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	La tabla Tipos de Depósitos no debe estar vacía.
Prototipo	Anexo A

Tabla 10. Descripción del caso de uso Definir los Depósitos.

Caso de Uso	Definir los Grupos de Control
Actores:	Usuario
Propósito:	Permite que el usuario defina los Grupos de control de la empresa.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Grupos de Control” y comienza a definir los diferentes grupos de control existentes y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 11. Descripción del caso de uso Definir los Grupos de Control.

Caso de Uso	Definir los Objetos a controlar
Actores:	Usuario
Propósito:	Permite que el usuario defina los Objetos a controlar.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Objetos de Control” y comienza a definir los objetos que se van a controlar y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU10 (extend)
Precondiciones:	Deben estar definidos los Grupos de Control
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	La tabla de Grupos de control no debe estar vacía.
Prototipo	Anexo A

Tabla 12. Descripción del caso de uso Definir los Objetos de Control.

Caso de Uso	Definir los Análisis a Realizar
Actores:	Usuario
Propósito:	Permite que el usuario defina los Análisis que se le realizaran a los diferentes Objetos de Control.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Análisis a Realizar” y comienza a definir los análisis a realizar a los objetos que se van a controlar y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 13. Descripción del caso de uso Definir los Análisis a Realizar.

Caso de Uso	Definir los Puntos de Muestreos
Actores:	Usuario
Propósito:	Permite que el usuario defina los Puntos donde se recogerán las muestras de los análisis para el esquema de control.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Puntos de Muestreos” y comienza a definir los puntos donde se recogerán las muestras y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 14. Descripción del caso de uso Definir los Puntos de Muestreos.

Caso de Uso	Definir los Tipos de Muestreos
Actores:	Usuario
Propósito:	Permite que el usuario defina los Diferentes tipos de muestreos o forma en recoger las muestras de los análisis para el esquema de control.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Tipos de muestreos” y comienza a definir de qué modo se tomaran las muestras y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 15. Descripción del caso de uso Definir los Tipos de Muestreos.

Caso de Uso	Definir NT37 Codificador Principal
Actores:	Usuario
Propósito:	Permite que el usuario definir el codificador Principal de Tiempo Perdido (NT37).
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/NT37 Principal” y comienza a definir el codificador principal de tiempo perdido y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 16. Descripción del caso de uso Definir el NT37 Codificador Principal.

Caso de Uso	Definir el NT37 Codificador General.
Actores:	Usuario
Propósito:	Permite que el usuario definir el codificador General de Tiempo Perdido (NT37).
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/NT37 Codificador” y comienza a definir el codificador general de tiempo perdido y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU15 (extend)
Precondiciones:	Se debe Definir Primero el NT37 Codificador Principal
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	La tabla del NT37 Codificador Principal No debe estar Vacía.
Prototipo	Anexo A

Tabla 17. Descripción del caso de uso Definir NT37 Codificador General.

Caso de Uso	Definir Listado de Brix más frecuentes
Actores:	Usuario
Propósito:	Permite que el usuario Defina el Listado de los Brix más frecuentes en la corrección de la pol en Bagazo
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Brix” y comienza a definir el listado de los brix mas frecuentes para luego hacerlo relacionar con la humedad para corregir la pol en bagazo y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 18. Descripción del caso de uso Definir Listado de Brix mas frecuentes.

Caso de Uso	Definir Listado de Humedad en Bagazo mas frecuentes.
Actores:	Usuario
Propósito:	Permite que el Usuario Defina el Listado de las Humedades en Bagazo más frecuentes en la corrección de Pol en Bagazo.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Humedad” y comienza a definir el listado de las humedades mas frecuentes para poder corregir la pol y plasmar los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 19. Descripción del caso de uso Definir Listado de Humedad en bagazo mas frecuentes.

Caso de Uso	Definir la Relación Humedad Vs Brix
Actores:	Usuario
Propósito:	Permite que el usuario cree la relación Humedad Vs Bagazo para poder corregir la pol en Bagazo
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Humedad en Bagazo” y comienza definir la relación del listado de brix y humedad y entrar el factor de corrección de la pol en bagazo para corregir la misma y seguidamente plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU17(extend), CU18(extend)
Precondiciones:	Deben definirse antes la lista de Brix y la de Humedad en Bagazo.
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	La tabla de Brix y de Humedad no deben estar vacías.
Prototipo	Anexo A

Tabla 20. Descripción del caso de uso Definir la Relación Humedad Vs Brix.

Caso de Uso	Crear el Esquema de Control
Actores:	Usuario
Propósito:	Permite al usuario crear el esquema de control por el que se registrará en toda la zafra.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Esquema de control” y comienza la relación entre los Análisis, Objetos de Control, Puntos de Muestreos, Tipos de Muestreos, Secciones esta se guardara en la base de datos para regir el control de los analíticos en todo el periodo. Se definen aquí las fórmulas de los que son calculados la forma en que acumulan si son totalizados o ponderados. El caso de uso termina cuando se presione el botón de salir de la forma.	
Referencia:	CU7, CU11,CU12,CU13,CU14
Precondiciones:	Deben definirse antes las secciones, los objetos de control, los análisis a realizar, los Puntos de Muestreos y Tipos de Muestreos.
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	Las tablas relacionadas con los casos de uso de referencia no deben estar vacías
Prototipo	Anexo A

Tabla 21. Descripción del caso de uso Crear esquema de control.

Caso de Uso	Definir las Categorías del IPLUS
Actores:	Usuario
Propósito:	Permite que el usuario Defina las Categorías Usadas por el IPLUS.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Categorias” y comienza a definir las diferentes categorías que presenta el IPLUS y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 22. Descripción del caso de uso Definir las Categorías del IPLUS.

Caso de Uso	Relacionar o Vincular IPLUS vs GILIPZ.
Actores:	Usuario
Propósito:	Permite que el usuario relacionar el Codificador de entrada del IPLUS con el Esquema de Control del GILIPZ.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Vincular Con IPLUS” y comienza relacionar los diferentes indicadores del esquema de control con los indicadores de entrada diarios del IPLUS para una importación diaria y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 23. Descripción del caso de uso Vincular IPLUS vs GILIPZ.

Caso de Uso	Crear Tabla de Corrección de Brix.
Actores:	Usuario
Propósito:	Permite que el usuario Cree La tabla de corrección de brix para poder corregir el brix mediante la temperatura.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Codificadores/Corrección de Brix” y comienza a crear la tabla de corrección de brix con respecto a la temperatura y plasma los datos en la base de datos. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 24. Descripción del caso de uso Crear tabla de corrección de brix.

Caso de Uso	Activar o Crear Día de Zara
Actores:	Usuario
Propósito:	Permite que el usuario Active un día de zafra o cree uno nuevo según la necesidad
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Operaciones/Activar Día” y el usuario escoge o crea un nuevo día (consecutivo) que este quedara activo. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 25. Descripción del caso de uso Activar Día.

Caso de Uso	Establecer turnos del día vs Brigadas
Actores:	Usuario
Propósito:	Permite que el usuario relacione las brigadas con los turnos existentes
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Operaciones/Establecer Turnos” y el usuario escoge o crea una nueva relación, entre los turnos diarios y las brigadas, ósea que brigada trabajara en cada turno ya que esto es variable. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU5, CU4 (extend)
Precondiciones:	Haber definido primero las brigadas y los turnos de trabajo.
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	Las tablas vinculadas al CU4 y CU5 no deben estar Vacías
Prototipo	Anexo A

Tabla 26. Descripción del caso de uso Establecer turnos del día vs Brigadas.

Caso de Uso	Insertar o modificar las paradas diarias
Actores:	Usuario
Propósito:	Permite que el usuario adicione o modifique una parada realizada por la industria.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Operaciones/Establecer Turnos” y el usuario inserta o modifica una parada que haya realizado el central, se plasma porque causa, cuantas veces lo hizo y el tiempo que duro la misma. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU15, CU16 (extend)
Precondiciones:	Se deben definir primero la NT37 Codificador General y la NT37 Codificador Principal
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	Las tablas relacionadas con el CU15 y CU16 no deben estar Vacías.
Prototipo	Anexo A

Tabla 27. Descripción del caso de uso insertar o modificar las paradas diarias.

Caso de Uso	Modificar o Insertar Análisis de laboratorio
Actores:	Usuario
Propósito:	Permite que el usuario inserte o modifique los análisis diarios realizados.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Operaciones/Diarios Laboratorio” y el usuario inserta o modifica un análisis en dependencia de su objetivo. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	-----
Precondiciones:	-----
Poscondiciones:	La información entrada por el usuario es almacenada en la base de datos
Requisitos Especiales:	-----
Prototipo	Anexo A

Tabla 28. Descripción del caso de uso Modificar o Insertar Análisis.

Caso de Uso	Visualizar y Reportar Datos del Laboratorio.
Actores:	Usuario
Propósito:	Permite que el usuario visualice o reporte los consolidados diarios.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Salidas/Laboratorio” y el usuario puede visualizar o crear los reportes necesarios. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU32 (include)
Precondiciones:	La información necesitada por el usuario debe estar almacenada en la base de datos
Poscondiciones:	-----
Requisitos Especiales:	Las tablas relacionadas a este caso de uso no deben estar vacías.
Prototipo	Anexo A

Tabla 29. Descripción del caso de uso Visualizar y Reportar Datos del Laboratorio.

Caso de Uso	Visualizar o Reportar Tabla de Entrada del IPLUS.
Actores:	Usuario
Propósito:	Permite que el usuario Visualice o Reporte la tabla de los datos diarios de Entrada del IPLUS.
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Salidas/IPLUS” y el usuario puede visualizar o crear el reporte de Entrada del IPIUS. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU33 (include)
Precondiciones:	La información necesitada por el usuario debe estar almacenada en la base de datos
Poscondiciones:	-----
Requisitos Especiales:	Las tablas relacionadas a este caso de uso no deben estar vacías.
Prototipo	Anexo A

Tabla 30. Descripción del caso de uso Visualizar o Reportar Tabla de Entrada del IPLUS.

Caso de Uso	Visualizar y Reportar Datos de la NT37.
Actores:	Usuario
Propósito:	Permite que el usuario Visualice o Reporte todo lo referente al Tiempo perdido Industrial o Norma Técnica 37 (NT37).
<u>Resumen</u> El caso de uso se inicia cuando el usuario solicita la opción “Salidas/NT37Salidas” y el usuario puede visualizar o crear los reportes referentes al tiempo perdido industrial o Norma Técnica 37 (NT37). El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	CU31 (include)
Precondiciones:	La información necesitada por el usuario debe estar almacenada en la base de datos
Poscondiciones:	-----
Requisitos Especiales:	Las tablas relacionadas a este caso de uso no deben estar vacías.
Prototipo	Anexo A

Tabla 31. Descripción del caso de uso Visualizar y Reportar Datos de la NT37.

Caso de Uso	Reportar e Imprimir NT37.
Actores:	Usuario
Propósito:	Permite que el Usuario cree los reportes necesarios y referentes a esta materia.
<u>Resumen</u> El caso de uso se inicia cuando el CU28 lo active para crear el reporte referente a este y con las especificaciones requeridas por el usuario. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	Deben existir Datos referentes a este CU.
Precondiciones:	La información referente a este CU de estar almacenada en la base de datos.
Poscondiciones:	-----
Requisitos Especiales:	Las tablas relacionadas con este CU no deben estar Vacías.
Prototipo	Anexo A

Tabla 32. Descripción del caso de uso Reportar e Imprimir NT37.

Caso de Uso	Reportar e Imprimir Datos Diarios.
Actores:	Usuario
Propósito:	Permite que el Usuario cree los reportes necesarios y referentes a esta materia.
<u>Resumen</u> El caso de uso se inicia cuando el CU29 lo active para crear el reporte referente a este y con las especificaciones requeridas por el usuario. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	Deben existir Datos referentes a este CU.
Precondiciones:	La información referente a este CU de estar almacenada en la base de datos.
Poscondiciones:	-----
Requisitos Especiales:	Las tablas relacionadas con este CU no deben estar Vacías.
Prototipo	Anexo A

Tabla 33. Descripción del caso de uso Reportar e Imprimir Datos Diarios.

Caso de Uso	Reportar e Imprimir Tabla IPLUS.
Actores:	Usuario
Propósito:	Permite que el Usuario cree los reportes necesarios y referentes a esta materia.
<u>Resumen</u> El caso de uso se inicia cuando el CU30 lo active para crear el reporte referente a este y con las especificaciones requeridas por el usuario. El caso de uso terminara cuando estos datos sean modificados y se presione el botón salir de la Forma.	
Referencia:	Deben existir Datos referentes a este CU.
Precondiciones:	La información referente a este CU de estar almacenada en la base de datos.
Poscondiciones:	-----
Requisitos Especiales:	Las tablas relacionadas con este CU no deben estar Vacías.
Prototipo	Anexo A

Tabla 34. Descripción del caso de uso Reportar e Imprimir Tabla IPLUS.

2.3 – Construcción del sistema

En el presente epígrafe se realiza una descripción de la construcción de la solución propuesta. En esta descripción se ha utilizado el Diagrama de Clases del Diseño como artefacto propuesto por la Metodología de RUP. Se plantean los diagramas del modelo lógico y físico de datos para una mayor comprensión del funcionamiento de la base de datos. Se describen los principios de diseño utilizados, mostrando ejemplos de cómo se presentan estos principios al usuario y la concepción general de la ayuda. También son descritas las consideraciones de codificación que se tuvieron en cuenta en la implementación de este sistema. Para

describir los elementos fundamentales de la implementación se muestra el Diagrama de Implementación.

2.3.1 – Diagrama de clases del diseño

Un diagrama de clases de diseño muestra la especificación para las clases de una aplicación. Incluye la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Navegabilidad.
- Dependencias.

A diferencia del modelo conceptual, un diagrama de clases de diseño muestra definiciones de entidades software más que conceptos del mundo real.

2.3.2 – Diagramas del modelo lógico de datos

El modelo lógico de la base de datos determina cómo se estructuran los datos de forma lógica mediante tablas y relaciones. Este diseño puede tener también una gran repercusión en el rendimiento de la aplicación.

Gestion de Información del Laboratorio en Periodo de Zafra.

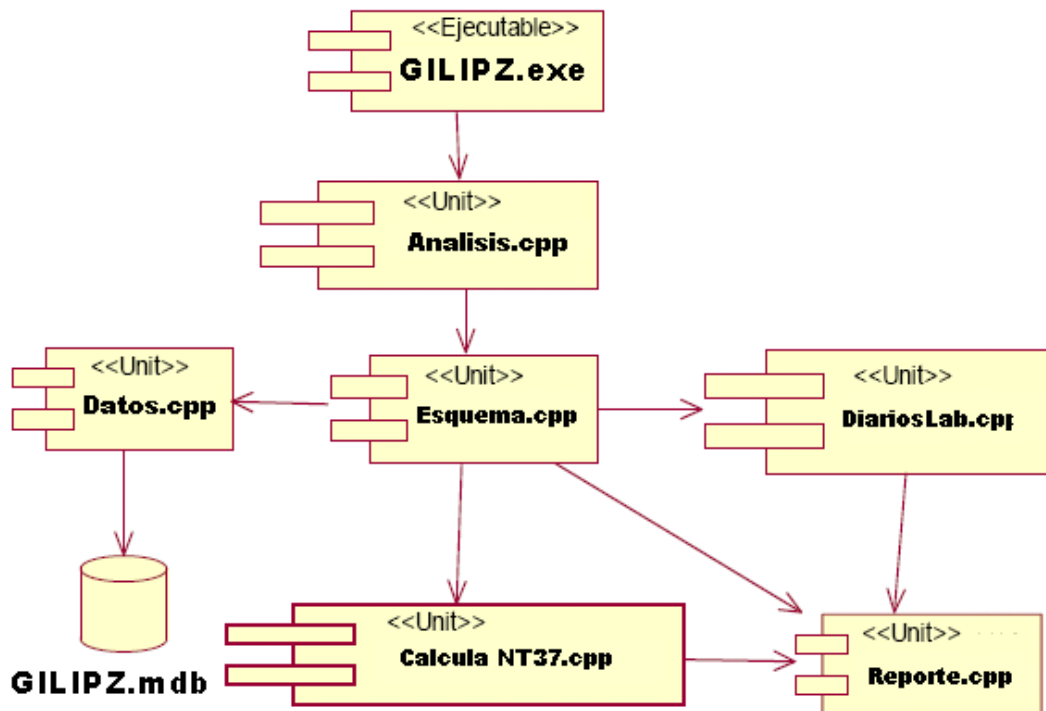


Figura 4.3. Diagrama de implementación

2.4 – Principios de diseño del sistema

2.4.1 - Estándares en la interfaz de la aplicación.

La interfaz diseñada para el sistema está basada en el estándar de ventanas de Windows. El tipo de letra a utilizar será arial de estilo normal y tamaño 9. La carga visual esta concebida para especialistas en la materia, para los cuales las opciones de acceso se han hecho de forma muy familiar teniendo en cuenta los objetivos buscados por los usuarios. También se muestra información referente a la hora del sistema, día, turno y brigada activo, así como los datos referentes a la empresa. Los mensajes de error e informativos se hacen sencillos pero precisos para que el usuario lo pueda identificar con facilidad.

2.4.2 - Concepción general de la ayuda.

Una parte importante del sistema lo constituye la ayuda, en este caso, el usuario la tendrá disponible en cada momento y podrá acceder a la misma con solo presionar la tecla *F1* o a través del menú de opciones en la correspondiente opción de Ayuda. La misma constará de una parte teórica y otra funcional. Esto tiene el objetivo de que el usuario no solo tenga la explicación funcional del sistema sino que también pueda contar con todo un conjunto de materiales que lo ayudarán en la preparación teórica de los contenidos abordados en el mismo. Además, el sistema también brinda una ayuda contextual que se muestra al pasar el Mouse sobre cada objeto, lo cual permite que el usuario pueda saber que operación puede realizar con cada uno de ellos.

2.4.3 - Estándares de codificación.

Para llevar a cabo el ejercicio de la buena ingeniería de software debe seguirse un buen estilo de código. Es necesario escribir código que sea fácil de entender y que reduzca el tiempo y esfuerzo a la hora de realizar alguna modificación al mismo.

Existen varios aspectos que pueden hacer un código más legible; algunos de estos son el empleo de nombres descriptivos, el uso de una indentación coherente y de comentarios informativos, entre otros. Se describen a continuación algunas convenciones tomadas con relación a estos aspectos.

Convención de Nombres

Los nombres de las variables, los controles, los procedimientos y funciones fueron adoptados lo más explicativos posibles, siempre respondiendo a su propósito.

Para cada tipo de control se tuvo en cuenta el uso de los prefijos más utilizados, una muestra de esto se muestra en la siguiente tabla:

Control	Prefijo	Ejemplo
Buttom	B	Bsalida
Form	Form	FormAbriendo
Menú	Menú	MenuPrincipal
Label	L	Lobjeto
Edit	E	Eobjeto
SkinData	Skin	SkinGILIPZ
DBNavegator	DBN	DBNEsquema
DBGGrid	DBG	DBGDepartamentos

Tabla: Ejemplos de controles y sus prefijos utilizados en la codificación.

En el caso de las variables comenzarán todas con la letra “v” seguido del nombre deseado, por ejemplo: .vFormula

Comentarios

Un buen comentario añade información al código de una manera clara y ayuda a entender el objetivo del mismo. Se tomó como regla, por tanto, comentar todos los procedimientos y funciones al principio de los mismos para explicar como se deben usar sin necesidad de leer el código. Se comentaron además algunos algoritmos que pudieran resultar de difícil comprensión.

Indentación

En el caso de la indentación, se establece que todas las líneas dentro de un procedimiento o función, estarán indentadas con respecto a la instrucción que encabeza a este, lo mismo ocurre con todas las líneas que conformen el cuerpo de un ciclo estructural condicional.

2.5 – Conclusiones

A través de este capítulo se obtiene de forma detallada toda la información del modelo del dominio se definen las entidades y sus conceptos principales dando como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema.

Se describió el modelo de sistema del objeto de automatización sobre la base de las especificaciones de la metodología RUP. Se identificaron los requerimientos funcionales y no funcionales, se definieron los actores del sistema y los servicios o funcionalidades que a su disposición se colocan (los casos de uso del sistema).

Además, se detallaron una serie de diagramas que colaboran con la implementación del modelo de sistema, como son: el diagrama de casos de uso del sistema, el diagrama de clases del diseño, el diagrama del modelo físico y lógico de datos y el diagrama de implementación.

Capítulo 3: Estudio de Factibilidad

3.1 Introducción.

En el presente capítulo se aborda el tema relativo al estudio de la factibilidad del producto, se ofrece una descripción de la planificación del proyecto, así como los costos asociados al mismo, los beneficios tangibles e intangibles que reportaría su elaboración y finalmente el análisis entre los costos y los beneficios para concluir si es o no factible el desarrollo del sistema.

Es necesario para la realización de un proyecto estimar el esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo. Estas estimaciones pueden realizarse a través del método de puntos de función del modelo de COCOMO II.

3.2 Planificación.

Se utilizó para el cálculo de la estimación del esfuerzo, el tiempo de desarrollo y el costo del proyecto el método de puntos de características.

Para realizar el cálculo de los costos de desarrollo del sistema se deben obtener primero las instrucciones fuentes. Analizándose para esto las cantidades de entradas, salidas, peticiones, archivos lógicos e interfases externas preliminares que tiene el sistema. Para calcular la cantidad de instrucciones fuentes hay que tener en cuenta también que la conversión al *C++ Builder*, y *Microsoft Office Access 2003* herramientas seleccionadas para implementar este software, es de 29 y 37 puntos respectivamente.

Después de este estudio se llegó a los siguientes resultados:

Tabla 3.1 Entradas Externas.

Nombre de la entrada externa	Cantidad de Ficheros	Cant de Elementos de Datos	Clasificación (Bajo, Medio, Alto)
Modificar los datos de la empresa.	1	8	Medio
Insertar Departamento.	1	2	Bajo
Modificar Departamento.	1	2	Bajo
Eliminar Departamento.	1	2	Bajo
Insertar Área.	1	2	Bajo
Modificar Área.	1	2	Bajo
Eliminar Área.	1	2	Bajo
Insertar Brigada.	1	2	Bajo
Modificar Brigada.	1	2	Bajo
Eliminar Brigada.	1	2	Bajo
Insertar Turno Trabajo.	1	4	Bajo
Modificar Turno Trabajo.	1	4	Bajo
Eliminar Turno de Trabajo.	1	4	Bajo
Insertar Hora.	1	4	Bajo
Modificar Hora.	1	4	Bajo
Eliminar Hora.	1	4	Bajo
Insertar Sección.	1	3	Bajo
Modificar Sección.	1	3	Bajo
Eliminar Sección.	1	3	Bajo
Insertar Tipo de Deposito.	1	2	Bajo
Modificar Tipo de Deposito.	1	2	Bajo
Eliminar Tipo de Deposito.	1	2	Bajo
Insertar Depósito.	1	3	Bajo
Modificar Depósito.	1	3	Bajo
Eliminar Depósito.	1	3	Bajo
Insertar Grupo de Control.	1	2	Bajo

CAPÍTULO 3: ESTUDIO DE FACTIBILIDAD

Modificar Grupo de Control.	1	2	Bajo
Eliminar Grupo de Control.	1	2	Bajo
Insertar Objeto a Controlar.	1	6	Medio
Modificar Objeto a Controlar.	1	6	Medio
Eliminar Objeto a Controlar.	1	6	Medio
Insertar Análisis.	1	2	Bajo
Nombre de la entrada externa (Continuación)	Cantidad de Ficheros	Cant de Elementos de Datos	Clasificación (Bajo, Medio, Alto)
Modificar Análisis.	1	2	Medio
Eliminar Análisis.	1	2	Medio
Insertar Punto de Muestreo.	1	2	Bajo
Modificar Punto de Muestreo.	1	2	Bajo
Eliminar Punto de Muestreo.	1	2	Bajo
Insertar Tipo de Muestreo.	1	2	Bajo
Modificar Tipo de Muestreo.	1	2	Bajo
Eliminar Tipo de Muestreo.	1	2	Bajo
Insertar NT37 Principal.	1	2	Bajo
Modificar NT37 Principal.	1	2	Bajo
Eliminar NT37 Principal.	1	2	Bajo
Insertar NT37 General.	1	3	Bajo
Modificar NT37 General.	1	3	Bajo
Eliminar NT37 General.	1	3	Bajo
Insertar Brix.	1	2	Bajo
Modificar Brix.	1	2	Bajo
Eliminar Brix.	1	2	Bajo
Insertar Humedad.	1	2	Bajo
Modificar Humedad.	1	2	Bajo
Eliminar Humedad.	1	2	Bajo
Insertar Relación Brix/Humedad.	1	3	Bajo
Modificar Relación Brix/Humedad.	1	3	Bajo
Eliminar Relación Brix/Humedad.	1	3	Bajo
Insertar Esquema de Control.	1	14	Alto
Modificar Esquema de Control.	1	14	Alto
Eliminar Esquema de Control.	1	14	Alto

CAPÍTULO 3: ESTUDIO DE FACTIBILIDAD

Insertar Categoría.	1	4	Bajo
Modificar Categoría.	1	4	Bajo
Eliminar Categoría.	1	4	Medio
Insertar Vinculo IPLUS vs Esquema de Control.	1	3	Medio
Modificar Vinculo IPLUS vs Esquema de Control.	1	3	Medio
Eliminar Vinculo IPLUS vs Esquema de Control.	1	3	Bajo
Insertar Corrección Brix.	1	3	Bajo

Nombre de la entrada externa (Continuación)	Cantidad de Ficheros	Cant de Elementos de Datos	Clasificación (Bajo, Medio, Alto)
Modificar Corrección Brix.	1	3	Bajo
Eliminar Corrección Brix.	1	3	Bajo
Agregar Día Zafra.	1	3	Bajo
Capturar Paradas Diarias.	1	4	Alto
Capturar Análisis Diarios.	1	5	Alto

Tabla 3.2 Salidas Externas.

Nombre de la salida externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
Visualizar la entrada IPLUS	2	10	Bajo
Visualizar Datos Diarios	8	17	Alto
Visualizar NT37	3	11	Medio

Tabla 3.3 Peticiones.

Nombre de la petición	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
Listar Entrada IPLUS	1	3	Bajo
Listar Datos Laboratorio Diario	1		Bajo
Listar NT37	1	3	Medio

CAPÍTULO 3: ESTUDIO DE FACTIBILIDAD

Listado de turnos	1	3	Bajo
Listar Días Zafra	1	3	Bajo
Listar Corrección Brix	1	3	Bajo
Listar Vinculo IPLUS	3	4	Medio
Listar Categorías	1	2	Bajo
Listar Esquema de Control	1	14	Alto
Listar Brix Humedad	1	3	Bajo
Listar Humedad	1	2	Bajo
Listar Brix	1	2	Bajo
Listar NT37 General	3	5	Alto
Listar NT37 Principal	1	2	Bajo
Nombre de la petición (Continuación)	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
Listar Tipos de Muestreos	1	2	Bajo
Listar Departamentos	1	2	Bajo
Listar Áreas	1	2	Bajo
Listar Brigadas	1	2	Bajo
Listar Turnos de Trabajo	1	4	Medio
Listar Horas	1	4	Medio
Listar Secciones	1	3	Bajo
Listar Tipos de Depósitos	1	2	Bajo
Listar Depósitos	3	5	Medio
Listar Grupos de Control	1	2	Bajo
Listar Objetos a Controlar	3	8	Medio
Listar Análisis	1	2	Bajo
Listar Puntos de Muestreos	1	2	Bajo

Tabla 3.4 Ficheros internos.

Nombre del fichero interno	Cantidad de record	Cantidad de Elementos de datos	Clasificación(Bajo, Medio y Alto)
---------------------------------------	-------------------------------	---	--

CAPÍTULO 3: ESTUDIO DE FACTIBILIDAD

BrixCorregido	1	2	Bajo
Categorías	1	2	Bajo
CdAreas	1	2	Bajo
CdBrigadas	1	2	Bajo
CdDeposito	1	3	Bajo
CdHoras	1	4	Medio
CdPuntoMuestreo	1	2	Bajo
CdSeecciones	1	3	Bajo
CdTpoDep	1	2	Bajo
CdTpoMuestreo	1	2	Bajo
CdTurnos	1	4	Bajo
CodEmpresa	1	13	Medio
CorrigeBrix	1	3	Bajo
Datos Diarios	8	9	Alto
Departamentos	1	2	Bajo
DiasZafra	1	4	Bajo
EstablecerTurno	1	5	Medio
Humedad	1	2	Bajo
HumedadBagazo	1	3	Bajo
LbAnálisis	1	2	Bajo
LbEsquema	1	17	Alto
LbGrupo	1	2	Bajo
LbObjeto	1	6	Medio
NT37Paradas	1	5	Medio
NT37Principales	1	4	Medio
NT37Reales	1	7	Alto
PrimeraCaptura	1	4	Medio

Tabla 3.5 Puntos de Función.

CAPÍTULO 3: ESTUDIO DE FACTIBILIDAD

Elementos	Bajos	X Peso	Medios	X Peso	Altos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	16	7	7	10	3	15	26
Ficheros de interfaces externas	0	5	0	7	0	10	0
Entradas externas	56	3	9	4	5	6	70
Salidas externas	1	4	1	5	1	7	3
Peticiones	19	3	6	4	2	16	27
Total							126

Tabla 3.6 Miles de instrucciones fuentes.

Características		Valor
Puntos de función desajustados		126
Lenguaje	SQL-Server	C++
Instrucciones fuentes por puntos de función	37	29
Por ciento de la aplicación en cuanto a requerimientos funcionales	45%	55%
Instrucciones fuentes	8621	10252
Total de Instrucciones fuentes	18873	

3.3 Costos.

Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Factores de escala.

Tabla 3.7 Factores de escala.

Cálculo de :	Valor	Justificación
PREC	2.48	El equipo de desarrollo posee una comprensión considerable de los objetivos del producto, no tiene experiencia en la realización de software de este tipo.
FLEX	3.04	Debe haber considerable cumplimiento de los requerimientos del sistema y de las especificaciones de interfaz externa.
RESL	7.07	Se está haciendo un estudio, no existe un plan definido.
TEAM	2.19	El equipo que va a desarrollar el software es muy cooperativo.
PMAT	7.80	Se encuentra en el nivel 1 (bajo).

Factores de escala.

SF =

$$\sum_{i=1}^5 SF_i = PREC + FLEX + RESL + TEAM + PMAT$$

SF =

$$\sum_{i=1}^5 SF_i = 3.72 + 3.04 + 7.07 + 2.19 + 7.80 = 23.82$$

Multiplicadores de esfuerzo.

Tabla 3.8 Multiplicadores de esfuerzo.

Cálculo de :	Valor	Justificación
RUSE	1.00	Tiene gran cantidad de código reusable.
PDIF	1.00	El software podrá estar trabajando varias horas, no tiene limitaciones de memoria impuesta y la plataforma de aplicación tiene gran estabilidad.
PREX	1.00	El equipo tiene dominio y conocimiento del lenguaje de programación. Con experiencia de aproximadamente 1 años.
FCIL	0.87	Se usarán técnicas modernas de programación.
SCED	1.00	La planificación se hace con moderada frecuencia.
RCPX	1.33	El software es confiable, la Base de Datos es grande.
PERS	0.83	La rotación del personal anualmente es casi nula.

Multiplicadores de esfuerzo.

EM =

$$\prod_{i=1}^6 E_{mi} = RUSE * PDIF * PREX * FCIL * SCED * RCPX * PERS$$

EM =

$$\prod_{i=1}^6 E_{mi} = 1.00 * 1.00 * 1.00 * 0.87 * 1.00 * 1.33 * 0.83 = 0,960393$$

Valores de los coeficientes.

A = 2.94; B = 0.91; C = 3.67; D = 0.24

$$E = B + 0.01 * SF$$

$$E = 0.91 + 0.01 * 23.82$$

$$E = 1,1482$$

$$F = D + 0.2 * (E - B)$$

$$F = 0.24 + 0.2 * (1,1482 - 0.91)$$

$$F = 0,28764$$

Esfuerzo.

$$PM = A * MF^E * EM$$

$$PM = 2.94 * (18,873)^{1.1482} * 0,960393$$

$$PM = 82,36$$

Cálculo del tiempo de desarrollo.

$$TDEV = C * PM^F$$

$$TDEV = 3.67 * (82,36)^{0,28764}$$

$$TDEV = 13,05 \approx 13 \text{ meses}$$

Cálculo de la cantidad de hombres.

$$CH = PM / TDEV$$

$$CH = 82,36 / 13,05$$

$$CH = 6,31 \approx 6 \text{ hombres}$$

Como el software se realiza por dos personas, se haría en 22 meses aproximadamente, trabajando cada uno 11 meses.

Costo.

Se asume como salario promedio mensual 198\$

$CHM = 1 * \text{Salario Promedio}$

$CHM = 1 * 198$

$CHM = 198 \text{ \$/mes}$

$\text{Costo} = CHM * PM$

$\text{Costo} = \$198 * 82,36$

$\text{Costo} = \$ 16307,28$

El costo sería \$ 16307,28

Los costos en los que se incurriría de desarrollarse el sistema serían:

Tabla 3.9 Resumen de resultados obtenidos.

Cálculo de:	Valor
Esfuerzo(PM)	82,36
Tiempo de desarrollo	22 meses
Cantidad de hombres	1
Costo	\$ 16307,28
Salario medio	\$198.0
RCPX	1.33
RUSE	1.00
PDIF	1.00
PREX	1.00
FCIL	0.87
SCED	1.00

3.4 Beneficios tangibles e intangibles.

Los beneficios obtenidos con el desarrollo del *software* son fundamentalmente intangibles, ya que permite mantener el control más detallado y organizado sobre el proceso de fabricación de azúcar evitando errores de cálculo. También implica un ahorro del tiempo que se invierte en esta tarea, de manera que se pueden realizar un mayor número de ensayos y mayor calidad de dicho proceso que implica calidad del producto final.

3.5 Análisis de costos y beneficios.

Al desarrollo de todo producto informático va asociado un costo, el justificarlo depende de los beneficios tangibles e intangibles que produce.

La utilización de este nuevo sistema para gestionar la información del laboratorio industrial de un central azucarero, parte de la idea de concebir los ensayos como un recurso estratégico para evaluar la calidad en el proceso de producción de azúcar y consigo elevar la calidad de este producto partiendo del ahorro de tiempo entre análisis para un desarrollo de mayor cantidad de estos y a su vez evitar los errores humanos y agilizar la toma de decisiones , mejora considerablemente las condiciones de trabajo que con solo acceder al sistema, analiza la situación del proceso a través de la interacción con este y libre de errores y sin tantos cálculos humanos que podrían demorar mas tiempo en la toma de decisiones.

La nueva herramienta automatizada brinda numerosas facilidades para obtener reportes de la situación del proceso en cuanto el comportamiento de este en cada momento dinámicamente.

Para el Laboratorio de la industria reviste gran importancia la utilización de un sistema que Gestione la información. Así se espera que el personal que dentro de sus labores tenía la de calcular y acceder a la información, disponga de mayor tiempo, tiempo que puede ser invertido en la realización de un mayor numero de ensayos.

Es factible desarrollar una herramienta para informatizar el proceso de gestión en el laboratorio Industrial, porque es importante optimizar el tiempo para poder controlar mejor el proceso ya que con mayor calidad en el proceso, mayor calidad del azúcar.

3.6 Conclusiones.

La herramienta propuesta trae consigo una serie de beneficios sobre todo intangibles para el laboratorio, pero no menos necesarios e importantes, porque va a contribuir a mejorar su funcionamiento, lo que indica que es factible implementar la herramienta propuesta. Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de 22 meses para su construcción por 1 hombre y su costo asciende a \$ 16307,28.

Conclusiones

Se concluye lo siguiente de la presente investigación:

- Se cumplieron de manera satisfactoria las etapas de análisis y diseño propuestas por la metodología RUP para el desarrollo de este tipo de producto de software.
- Brinda una herramienta capaz de gestionar la información del laboratorio industrial del CAI Antonio Sánchez en período de zafra.
- Facilita el cálculo, análisis y elaboración de los diferentes ensayos de laboratorio.

Finalmente, se realizó un estudio de factibilidad mediante una estimación por puntos de función donde se pudo apreciar que era viable la implementación del sistema.

Recomendaciones

Después de realizar un análisis exhaustivo del presente trabajo en los capítulos referidos anteriormente y haber llegado a conclusiones, pueden ser planteadas las siguientes sugerencias:

- Se sugiere que el sistema propuesto sea enriquecido con nuevos reportes y análisis estadísticos; que permitan realizar comparaciones entre indicadores obtenidos en diversas zafras por un período de diez años.
- Proponer su utilización y generalización en los demás centrales azucareros de nuestro país.

Referencias Bibliográficas

- Díaz Antón, Maria Gabriela; Angélica Pérez, Maria. Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática. Tomado de:
<http://www.academia-interactiva.com/ise.pdf> , 10 marzo del 2008.
- Ferrá Grau, Xavier. Desarrollo orientado a objetos con UML. Tomado de:
<http://www.clikear.com/manuales/uml/introduccion.asp>, 10 de marzo del 2008.
- Piñero Suárez, Boris. MaCoSoft Software para la elaboración de Mapas Conceptuales. Boris Piñero Suárez; Danaysi Ruíz Bravo; Alfredo J. Simón Cuevas, tutor. – Trabajo de Diploma, CUJAE (CH), 2004. – 104h. : ilus.
- Ferrá Grau, Xavier “Desarrollo orientado a objetos con UML”/ Xavier Ferrá Grau, Universidad Politécnica de Madrid, 2004 Tomado de:
<http://www.clikear.com/manuales/uml/introduccion.asp>, 10 de marzo del 2008.
- Letelier Torres, Patricio "Desarrollo de Software Orientado a Objeto usando UML" /Patricio Letelier Torres. Departamento Sistemas informáticos y Computación, Universidad Politécnica de Valencia, España. Tomado de:
[URL:http://www.creangel.com/uml/intro.php](http://www.creangel.com/uml/intro.php) , 20 de abril del 2008.
- Díaz Antón, Maria Gabriela. “Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática”./Maria Gabriela Díaz Antón, Maria Angélica Pérez, 2004. Tomado de: <http://www.academia-interactiva.com/ise.pdf>, abril del 2008.
- Popkin Software and Systems. “Modelado de Sistema con UML” Tomado de:
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c124.html>, 20 de abril del 2008.
- Comunidad de desarrolladores. Diagrama de Clases de Diseño. Tomado De:
<http://www.clikear.com/manuales/uml/faseconstruccionbajonivel.asp> 9 de abril del 2008.

Bibliografía.

- Lázaro J. Blanco Encinosa e IDA R. GUTSZTAT GUTSZTAT, Sistemas Informáticos, Teoría, Métodos de Elaboración, Técnicas, Herramientas Tomo I y II.
- Craig Larman, UML y Patrones, Introducción al análisis y diseño orientado a objetos. Tomo I, II y III.
- Gregory L. Heileman, Estructuras de Datos, Algoritmos, y Programación Orientada a Objetos.
- MACU (Manual de Técnicas Analíticas).
- Revistas Cuaba azúcar.

Glosario de Términos

Azúcares Ecológicos: Azúcar que su producción causo el menor daño posible al ecosistema y a la salud humana, para ello no se utilizo ningún producto químico ni en el cultivo de la caña (fertilizantes, herbicidas, plaguicidas, etc.) ni para la fabricación de azúcar, (Hidróxido de cal, potasa, ácidos, etc.) solamente se utilizan productos orgánicos, como cachaza, Guano y Agentes biológicos, para el control de plagas, también se le llama azúcar orgánica.

Brix: Contenido de sólidos solubles que hay presente en una solución azucarada, en ella están incluida la pol o el contenido de sacarosa, se expresa en % y su valor como es lógico es mayor que el de la pol.

Color: Escala de unidades de color caramelo en la que se compara la solución de azúcar, en la medida que sea mayor la calidad del azúcar es menor.

Edulcorantes: Sustancia capaz de endulzar.

Edulcorantes hipocalóricos: Sustancia edulcorantes con bajo poder de calorías.

Ensayos: Prueba o examen de laboratorio con el objetivo de determinar, la precisión o contenido de una sustancia, (llamado normalmente análisis).

Mascabado: Nombre comercial del producto antes de separarse la miel del grano de azúcar, así se presentaba el producto en el mercado antes de patentizarse las centrifugas, que separan el azúcar y la miel.

NT37 : (Norma Técnica 37) En el sistema MINAZ, existen un grupo de Normas Técnicas, las cuales cumplen ciertos objetivos, están codificadas en el orden en que se crearon. La NT37 relaciona las paradas, o tiempo perdido del ingenio con las causas que la provocaron codificando cada causa con un código específico sujeto a ciertas reglas.

Pol: Debe su nombre a la propiedad que tienen las soluciones azucaradas de desviar cierto ángulo al plano de luz polarizada, en dependencia de la cantidad de concentración que ha en la misma. Significa el contenido de azúcar (o sacarosa) que hay presente en una solución azucarada. Se expresa en %.

Sucumbir: Caer Fatigado por una carga excesiva, rendirse, ceder, morir, perecer, dejar de existir.

VLC (Very Low Color): azúcar de muy bajo color, dado por requisitos del cliente, fundamentalmente los refinadores de azúcar.

GLOSARIO DE TÉRMINOS

VHP(Very Hight Pol): azúcar de alta pol o dolarización igual que la anterior es un requisito que imponen ciertos clientes, de acuerdo al uso que se le quiera dar.