



Instituto Superior Minero Metalúrgico
“Dr. Antonio Núñez Jiménez”.

Facultad de Geología - Minas
Departamento de Informática
Moa, Holguín

Trabajo de Diploma

En opción al Título de Ingeniería Informática

*Sistema para el análisis termoenergético de
secadores cilíndricos rotatorios horizontales en la
Empresa “Comandante Ernesto Che Guevara”.*

Autora: *Keily Caraballo Mondelo*

Tutores: *Ing. Yodelkis Delgado Drubey*

MSc. Roiky Rodríguez Noa

Moa, 2012

“Año 54 de la Revolución”

Declaración de autoría:

Declaro que soy la única autora del trabajo: Sistema para el análisis termoenergético de secadores cilíndricos rotatorios horizontales en la Empresa “Comandante Ernesto Che Guevara” y certifico su propiedad a favor del Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, el cual podrá hacer uso del mismo con la finalidad que estime conveniente.

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

Keily Caraballo Mondelo
Firma de la Autora

Ing. Yodelkis Delgado Drubey
Firma del Tutor

MsC. Roiky Rodríguez Noa
Firma del Tutor



“El futuro de Cuba tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento.”

Fidel Castro Ruz

Agradecimientos

Un especial agradecimiento a todas las personas que de una forma u otra han contribuido con su ayuda en la realización de este trabajo, en especial:

A mis padres quienes con su amor, apoyo y comprensión incondicional me han guiado a lo largo de todo el camino, gracias a ustedes soy quien soy, y cada día me enorgullezco más de ser su hija.

A mis hermanos Danny y Egllys y a mi cuñada Katie por brindarme su amor, su cariño, su dedicación, por consentirme tanto y ser una guía en mi futuro.

A mi segunda familia Esmerido, Yamilet, Tito, Ernesto, Javier y al pequeñito de la casa Ernesto Alejandro y en especial a mi segunda mamá Elisa, gracias por soportarme estos 5 años, por tu paciencia y comprensión, siempre estarás en mi corazón.

A mis amigas Arletis, Sandra, Diosdada, Yuli, Dayamis, Nela, Yenía y a mis amigos Osiel, Evelio, Alexander y Eddy por todos los momentos lindos que hemos compartido, por su apoyo, cariño, comprensión, por ser tan especiales y siempre estar ahí en los buenos y malos momentos, siempre serán muy especiales para mí.

A mis tutores Yodelkis y Roiky por la confianza que depositaron en mí, por su paciencia y apoyo incondicional.

A mis compañeros de aula con los que compartí muchos momentos de alegría, por la ayuda y la comprensión mostrada.

A mis primas Yunia, Yaimi y mi primo Yonny, a mis abuelos, mis tios y tias en general a toda la familia, gracias por ser tan especiales y brindarme todo su amor y cariño.

A mi amiga Anais por brindarme su amistad y cariño, aunque estemos lejos y quizás no nos veamos más, siempre estarás en mi corazón.

A los profesores del departamento, por ayudarnos a formarnos como los profesionales en que hoy nos convertimos.

Al profesor Roilber por su apoyo y comprensión.

A mis vecinos por ser parte de mi familia.

*A mis hermanos
A las personas que confiaron en mí
A mis padres, A mi familia, A mis amigos*

Resumen

Entre los numerosos procesos industriales que ocurren en las empresas del níquel, resulta vital el estudio del secado de mineral laterítico en cilindros rotatorios horizontales; en el cuál se están haciendo varias modificaciones, por lo que es necesario trabajar en soluciones globales para que las empresas mejoren su nivel de automatización.

Por estas razones la presente investigación se centro en él se desarrollo una aplicación informática, que automatiza los modelos matemáticos que intervienen en el proceso de secado establecido por los estudios experimentales en la región de Moa, permitiendo que se agilice el cálculo de secado del mineral y motiven a investigadores de la materia a desarrollar mejores técnicas para el uso del secado.

Se utilizaron herramientas de software libre y la metodología de desarrollo de software RUP. Se realizó un estudio de factibilidad y sostenibilidad sobre el sistema, apoyado en el modelo COCOMO II; se obtuvo los beneficios tangibles e intangibles, que permiten hacer un análisis de los costos del desarrollo de este sistema arrojando que el proyecto es útil para que la entidad logre sus objetivos.

Abstract

Among the numerous industrial processes that happen in the companies of the nickel, it is important the study of the drying of mineral laterítico in horizontal rotational cylinders; in the which they are becoming several modifications, for what is necessary to work in global solutions so that the companies improve their automation level.

For these reasons the present investigation was carried out in the you development a computer application that automates the mathematical models that intervene in the drying process settled down by the experimental studies in the region of Moa, allowing that the calculation of drying of the mineral is speeded up and motivate investigators of the matter to develop better techniques for the use of the drying.

Tools of free software and the methodology of software development RUP were used. Besides was carried out a study of feasibility and sustainable on the system, supported in the pattern COCOMO II; it was obtained the tangible and intangible benefits that allow to make an analysis of the costs of the development of this system throwing that the project is useful so that the entity achieves its objectives.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	7
1.1. Trabajos precedentes sobre los procesos de secado	7
1.2. Descripción del flujo tecnológico de la planta	9
1.2.1. Descripción del proceso de secado en los secadores cilíndricos	10
1.3. Sistemas automatizados existentes vinculados al campo de acción	11
1.3.1. Tendencias en tecnología de lenguajes de Programación	12
1.3.2. Lenguaje de Programación C	13
1.3.3. Lenguaje de Programación C++	15
1.3.4. Lenguaje de Programación Java	16
1.4. Fundamentación del lenguaje a utilizar.....	17
1.4.1. Ambiente de Desarrollo Integrado (IDE) elegido: NetBeans 7.0.	19
1.5. Patrones Arquitectónicos	19
1.6. Bases de Datos Embedded.....	21
1.6.1. Apache Derby.....	22
1.6.2. Firebird	24
1.6.3. Berkeley DB (BDB).....	25
1.7. Fundamentación de la Base de Datos a utilizar.....	25
1.8. Embarcadero ER/Studio 8.0.0.....	26
1.9. Metodología de desarrollo.....	27
1.9.1. Metodologías tradicionales.....	27
1.9.1.1. Rational Unified Process (RUP en español Proceso Unificado Racional)	28
1.9.1.2. Microsoft Solution Framework (MSF en español Marco de Soluciones Microsoft)	29
1.9.2. Metodologías ágiles	30
1.9.2.1. Scrum.....	30
1.9.2.2. Extreme Programming (XP en español Programación Extrema)	31
1.9.3. Fundamentación de la metodología a utilizar	32
1.10. Herramientas de modelado	32
1.10.1. Visual Paradigm	33
1.10.2. Rational Rose	33
1.10.3. Fundamentación de la herramienta de modelado a utilizar	34
1.11. Conclusiones del Capítulo.....	35
Capítulo 2. Modelo del Dominio y Requisitos	37

2.1. ¿Por Qué Modelo de Dominio?	37
2.2. Definición de las entidades y conceptos fundamentales	37
2.3. Representación del diagrama de clases del Modelo del Dominio	38
2.4. Requerimientos	39
2.4.1. Requerimientos funcionales	39
2.4.2. Requerimientos no funcionales	40
2.4.2.1. Apariencia o interfaz externa	41
2.4.2.2. Requerimiento de disponibilidad	41
2.4.2.4. Requerimiento de rendimiento	41
2.4.2.5. Requerimiento de confiabilidad	42
2.4.2.6. Requerimiento de software	42
2.4.2.7. Requerimiento de hardware	42
2.4.2.8. Requerimiento de portabilidad	42
2.4.2.9. Requerimiento de diseño e implementación	42
2.4.2.10. Requerimiento de usabilidad	42
2.5. Conclusiones del capítulo	43
Capítulo 3. Diseño e Implementación del Sistema	45
3.1. Identificación de los actores del sistema a automatizar	45
3.2. Paquetes y sus relaciones	46
3.3. Diagrama de caso de uso del sistema	47
3.4. Descripciones de los casos de uso del sistema	49
3.5. Diagrama de clases del diseño	50
3.6. Descripción de las clases del diseño	52
3.7. Principios de diseño	53
3.7.1. Prototipo de Interfaz de usuario	53
3.7.2. Formato de salida de los reportes	55
3.7.3. Ayuda	55
3.8. Tratamientos de errores	55
3.9. Diseño de la base de datos	56
3.9.1. Modelo lógico de datos	57
3.9.1.1. Descripciones textuales de las clases persistentes	59
3.10. Diagrama de secuencia	61
3.11. Modelo de despliegue	62

3.12. Diagrama de Componente	63
3.13. Conclusiones del capítulo	68
Capítulo 4. Estudio de Factibilidad	70
4.1. Método COCOMO II	70
4.2. Pasos a seguir en el método Post Arquitectura	70
4.2.1. Desarrollo del método Post Arquitectura	71
4.3. Beneficios tangibles e intangibles	81
4.4. Análisis de costos y beneficios	81
4.5. Conclusiones del capítulo	82
Conclusiones Generales	83
Recomendaciones.....	84
Referencias Bibliográficas	85
Anexo # 1: Descripciones de los Casos de Uso del Sistema	88
Anexo # 2: Descripciones de las Clases del Diseño.....	101
Anexo # 3: Descripciones de las Tablas que Conforman el Modelo Físico de Datos	113
Anexo # 4: Diagramas de Secuencia de los Casos de Uso del Sistema	122
Glosario de Términos	135

Índice de Tablas

Tabla 3.1. Actores del sistema.....	45
Tabla 3.2. Descripción del caso de uso “Cálculo del calor aportado por el diesel (Q_c)”.....	49
Tabla 3.3. Descripción de la clase Calor_Apor_por_Combustible.....	52
Tabla 3.4. Descripción de la tabla Calor_Apor_por_Combustible.....	59
Tabla 4.1. Entradas Externas.....	71
Tabla 4.2. Salidas externas.....	72
Tabla 4.3. Ficheros Lógicos Internos.....	73
Tabla 4.4. Puntos de función desajustados.....	74
Tabla 4.5. Factores escala.....	76
Tabla 4.6. Valores específicos de los multiplicadores de esfuerzo.....	77
Tabla 4.7. Cantidad de hombres.....	79
Anexo # 1: Descripciones de los casos de uso	
Tabla 1. Descripción del caso de uso “Administrar Base de Datos”.....	88
Tabla 2. Descripción del caso de uso “Cálculo de los volúmenes de la combustión”.....	88
Tabla 3. Descripción del caso de uso “Cálculo del calor físico aportado del aire primario de combustión (Q_{fa})”.....	89
Tabla 4. Descripción del caso de uso “Cálculo aportado por el aire parásito (Q_{ap})”.....	89
Tabla 5. Descripción del caso de uso “Cálculo del calor físico del aire de pulverización (Q_{pu})”.....	90
Tabla 6. Descripción del caso de uso “Cálculo del calor aportado por el mineral húmedo (Q_{ch})”.....	90
Tabla 7. Descripción del caso de uso “Cálculo del calor que sale con el mineral húmedo (Q_{cs})”.....	91
Tabla 8. Descripción del caso de uso “Cálculo del calor sensible de los gases a la salida del secador (Q_{gs})”.....	91
Tabla 9. Descripción del caso de uso “Cálculo del calor sensible y latente de vaporización del agua contenida en el mineral (Q_{va})”.....	92
Tabla 10. Descripción del caso de uso “Cálculo de las pérdidas por transferencia de calor por cámara de paso (QP_1)”.....	92
Tabla 11. Descripción del caso de uso “Cálculo de las pérdidas por transferencia de calor por tambor secador (QP_2)”.....	93
Tabla 12. Descripción del caso de uso “Cálculo de las pérdidas por transferencia de calor por botella (QP_3)”.....	93
Tabla 13. Descripción del caso de uso “Cálculo del Rendimiento de la Instalación”.....	94
Tabla 14. Descripción del caso de uso “Mostrar Reporte de (Q_c)”.....	94
Tabla 15. Descripción del caso de uso “Mostrar Reporte de (Q_{fa})”.....	95
Tabla 16. Descripción del caso de uso “Mostrar Reporte de Volumen de Combustión”.....	95
Tabla 17. Descripción del caso de uso “Mostrar Reporte de (Q_{ap})”.....	96
Tabla 18. Descripción del caso de uso “Mostrar Reporte de (Q_{pu})”.....	96
Tabla 19. Descripción del caso de uso “Mostrar Reporte de (Q_{ch})”.....	97

Tabla 20. Descripción del caso de uso “Mostrar Reporte de (Q_{cs})”	97
Tabla 21. Descripción del caso de uso “Mostrar Reporte de (Q_{gs})”	98
Tabla 22. Descripción del caso de uso “Mostrar Reporte de (Q_{va})”	98
Tabla 23. Descripción del caso de uso “Mostrar Reporte de (QP_1)”	99
Tabla 24. Descripción del caso de uso “Mostrar Reporte de (QP_2)”	99
Tabla 25. Descripción del caso de uso “Mostrar Reporte de (QP_3)”	100
Tabla 26. Descripción del caso de uso “Mostrar Reporte del Rendimiento de Instalación”	100
Tabla 27. Descripción del caso de uso “Graficar con respecto a los flujos”	101

Anexo # 2: Descripción de las clases del diseño

Tabla 1. Descripción de la clase Calculo_Aportado_Aire_Parasito	101
Tabla 2. Descripción de la clase Calor_Aportado_Mineral_Húmedo	102
Tabla 3. Descripción de la clase Calor_Físico	102
Tabla 4. Descripción de la clase Calor_Físico_Aire_Pulverización	103
Tabla 5. Descripción de la clase Calor_Sale_Mineral_Húmedo	103
Tabla 6. Descripción de la clase Calor_Sensible_Gases_Salida	104
Tabla 7. Descripción de la clase Calor_Sensible_y_Latente_Agua	104
Tabla 8. Descripción de la clase Botella	105
Tabla 9. Descripción de la clase Camara_Paso	106
Tabla 10. Descripción de la clase Tambor_Secador	108
Tabla 11. Descripción de la clase Perdidas_Tranf_Calor	109
Tabla 12. Descripción de la clase Volumenes_Combustible	110
Tabla 13. Descripción de la clase Conexion_Derby	111
Tabla 14. Descripción de la clase Balance_Corrientes	112
Tabla 15. Descripción de la clase Proceso_Secado	112
Tabla 16. Descripción de la clase Graficar	112
Tabla 17. Descripción de la clase Rendimiento_Inst	112

Anexo # 3 : Descripción de las tablas del Modelo físico de datos

Tabla 1. Descripción de la tabla Volumenes_Combustible	113
Tabla 2. Descripción de la tabla Calculo_Aportado_Aire_Parasito	114
Tabla 3. Descripción de la tabla Calor_Aportado_Mineral_Húmedo	114
Tabla 4. Descripción de la tabla Calor_Físico_Aire_Pulverización	115
Tabla 5. Descripción de la tabla Calor_Físico	115
Tabla 6. Descripción de la tabla Calor_Sale_Mineral_Húmedo	116
Tabla 7. Descripción de la tabla Calor_Sensible_y_Latente_Agua	116
Tabla 8. Descripción de la tabla Calor_Sensible_Gases_Salida	117
Tabla 9. Descripción de la tabla Balance_Corrientes	117
Tabla 10. Descripción de la tabla Rendimiento_Inst	117
Tabla 11. Descripción de la tabla Perdidas_Tranf_Calor	118
Tabla 12. Descripción de la tabla Camara_Paso	118
Tabla 13. Descripción de la tabla Tambor_Secador	119
Tabla 14. Descripción de la tabla Botella	120

Índice de Figuras

Figura 1.1. Proceso de Secado del Mineral	10
Figura 1.2. Funcionamiento del patrón MVC	21
Figura 2.1. Diagrama de Modelo de Dominio	38
Figura 3.1. Diagrama de Paquetes	46
Figura 3.2. Diagrama de Casos de Uso del Paquete Conexión	47
Figura 3.3. Diagrama de Casos de Uso del Paquete Balance de Corrientes.....	48
Figura 3.4. Diagrama de Casos de Uso del Paquete Perdida por Transferencia de Calor	48
Figura 3.5. Diagrama de Casos de Uso del Paquete Reportes.....	49
Figura 3.6. Diagrama de Clases del Diseño	51
Figura 3.7. Interfaz gráfica para guardar la base de datos	53
Figura 3.8. Interfaz gráfica para calcular el calor aportado por el combustible	54
Figura 3.9. Interfaz gráfica para calcular el calor aportado por el combustible	54
Figura 3.10. Interfaz gráfica reporte del calor aportado por el combustible	55
Figura 3.11. Cálculo del Calor Aportado por el Combustible	56
Figura 3.12. Diagrama de Clases Persistentes.....	58
Figura 3.13. Modelo Físico de Datos	60
Figura 3.15. Diagrama de secuencia del CU “Cálculo del calor aportado por el diesel”	61
Figura 3.16. Diagrama de Despliegue	62
Figura 3.17. Diagrama de Componentes General	63
Figura 3.18. Diagrama de Componentes del Paquete de Balance_Corrientes	64
Figura 3.19. Diagrama de Componentes del Paquete Conexión_Derby.....	65
Figura 3.20. Diagrama de Componentes del Paquete Pérdida_Tranf_Calor	66
Figura 3.21. Diagrama de Componentes del Paquete Reportes	67
Anexo # 4 : Diagramas de Secuencia de los Casos de Uso del Sistema	
Figura 1. Diagrama de secuencia del CU “Calcular Calor Físico de Combustible”	122
Figura 2. Diagrama de secuencia del CU “Calculo del Aire Parásito”	122
Figura 3. Diagrama de secuencia del CU “Calor Físico de Pulverización”	123
Figura 4. Diagrama de secuencia del CU “Calcular Calor Aportado por Mineral Húmedo”	123
Figura 5. Diagrama de secuencia del CU “Calcular Calor Sale del Mineral Húmedo”	124
Figura 6. Diagrama de secuencia del CU “Calcular Calor Sensible a Gases de Salida”	124
Figura 7. Diagrama de secuencia del CU “Calcular Calor Sensible Latente Mineral”	125
Figura 8. Diagrama de secuencia del CU “Calcular Cámara de Paso”	125
Figura 9. Diagrama de secuencia del CU “Calcular Tambor Secador”	126
Figura 10. Diagrama de secuencia del CU “Calcular Botella”	126
Figura 11. Diagrama de secuencia del CU “Calcular Rendimiento de la Instalación”	127
Figura 12. Diagrama de secuencia del CU “Graficar respectos a flujos”	127
Figura 13. Diagrama de secuencia del CU “Calcular Volúmenes de Combustible”	128
Diagrama de secuencia del CU “Administrar Base de Datos”	129
Figura 14.1. Sección “Guardar Base de Datos”	129
Figura 14.2. Sección “Abrir Bases de Datos”	129
Figura 14.3. Sección “Cerrar Conexión de Bases de Datos”	129
Figura 15. Diagrama de secuencia del CU “Mostrar Reporte Calor Aportado por Combustible”	130
Figura 16. Diagrama de secuencia del CU “Mostrar Reporte Calor Físico del Combustible” ..	130
Figura 17. Diagrama de secuencia del CU “Mostrar Reporte para Calcular Aire Parásito”	131
Figura 18. Diagrama de secuencia del CU “Mostrar Reporte Calor Físico de Pulverización” ..	131
Figura 19. Diagrama de secuencia del CU “Mostrar Reporte Calor del Mineral Húmedo”	131

Figura 20. Diagrama de secuencia del CU “Mostrar Reporte Calor Sale del Mineral Húmedo” 131

Figura 21. Diagrama de secuencia del CU “Mostrar Reporte Calor Gases Salida del Secador” 132

Figura 22. Diagrama de secuencia del CU “Mostrar Reporte Calor Sensible y Latente del Mineral” 132

Figura 23. Diagrama de secuencia del CU “Mostrar Reporte Cámara de Paso” 132

Figura 24. Diagrama de secuencia del CU “Mostrar Reporte Tambor Secador” 133

Figura 25. Diagrama de secuencia del CU “Mostrar Reporte Botella” 133

Figura 26. Diagrama de secuencia del CU “Mostrar Reporte del Rendimiento de la Instalación” 133

Figura 27. Diagrama de secuencia del CU “Mostrar Reporte de Volúmenes de Combustible” 134

Introducción

La informática o computación es la ciencia del tratamiento automático de la información mediante un computador (llamado también ordenador o computadora).

Entre las tareas más populares que ha facilitado esta tecnología se encuentran: elaborar documentos, enviar y recibir correo electrónico, dibujar, crear efectos visuales y sonoros, maquetar folletos y libros, manejar la información contable en una empresa, reproducir música, controlar procesos industriales y jugar.

La informática es un amplio campo que incluye los fundamentos teóricos, el diseño, la programación y el uso de las computadoras (ordenadores).

El creciente desarrollo en el campo de la informática, ha permitido que las computadoras estén presentes en todos los sectores de la sociedad. Las computadoras no constituyen una solución mágica a los problemas de un ciudadano, empresa e incluso el país, pero sí contribuyen a aumentar la superación intelectual y profesional, abrir nuevas perspectivas de vida y hasta abaratar costos en la economía personal, a nivel empresarial y de nación. En el caso de Cuba se puede decir que quiere asumir los desafíos de la era digital, pues si no cuenta con ese vital recurso es imposible aspirar a crecer económicamente y ser competitivo. Existe la voluntad política y el capital humano requerido, solo quedaría trabajar en el mejoramiento de la infraestructura tecnológica y organizativa.

La Informática aplicada a la industria nos permite reducir costes y aumentar la productividad, además de incrementar la calidad del producto. Los conocimientos en esta área permiten a las empresas que quieran mejorar y optimizar sus procesos, contar con aplicaciones que resultan de la integración de algoritmos, procedimientos, y funciones que están derivadas del comportamiento físico de dichos procesos industriales. Entre los numerosos procesos industriales que ocurren en las empresas del níquel, resulta vital el estudio del secado de mineral laterítico en cilindros rotatorios horizontales; en el cuál se están haciendo varias modificaciones, por lo que es necesario

trabajar en soluciones globales para que las empresas mejoren su nivel de automatización.

El secado de mineral es una práctica ampliamente difundida, que origina diversos problemas que afectan los costos y la calidad del mismo.

Los secaderos cilíndricos de la planta de preparación de mineral cuya tecnología está basada en el esquema carbonato-amoniaco del mineral reducido, es un proceso continuo, utilizado en las condiciones de presión atmosférica y el equipamiento tecnológico se distingue por su sencillez y amplia utilización en los aparatos conocidos (hornos de soleras múltiples, espesadores y columnas de destilación). Estas características favorecen la creación de una producción con alto nivel de mecanización y automatización llamado proceso Carón.

En la empresa “Comandante Ernesto Che Guevara”, la reducción del consumo de portadores energéticos y la optimización de sus procesos inciden considerablemente en la eficiencia de los diferentes equipos e instalaciones que la componen. El secado, desde el punto de vista industrial, es un proceso de transferencia de calor y de masa, mediante el cual se hace pasar un producto desde un estado inicial húmedo a otro estado final menos húmedo, lo que implica extraer el líquido que posee este, obteniendo al final un producto diferente, ya sea como producto final o como intermedio en un proceso de fabricación dado. Este líquido suele ser agua natural o mezclada, también puede tratarse de otro tipo de disolvente orgánico e inorgánico.

En la actualidad para facilitar el cálculo del secado de mineral laterítico en cilindros rotatorios horizontal se utiliza Microsoft Excel y Calculadora, pero los cálculos necesarios son demasiado extensos y reiterativos por lo que también es necesario apoyarse de forma manual, pues estos software integran todo el proceso del cálculo de los modelos matemáticos del secado. Los parámetros que influyen en estos modelos matemáticos, varían constantemente sus resultados, ya que su valor dependen de variaciones climatológicas, al realizar estos cálculos a mano trae como consecuencia poca confiabilidad en los

resultados, debido a que al realizar cálculos manuales esta propensa a errores y demora para obtener sus resultados.

Este proceso de secado antes expuesto en la empresa, ofrece una gran cantidad de datos a partir de los cuales se pueden analizar el estado en el que está trabajando la instalación.

A partir de los aspectos antes mencionados se declara como **problema:**

¿Cómo facilitar el análisis energético de los secadores cilíndricos rotatorios horizontales en la Empresa “Comandante Ernesto Che Guevara”?

Como **objeto de estudio** de la investigación se plantea: Informatización del procedimiento de cálculo en los secadores cilíndricos rotatorios horizontales.

Campo de acción: Automatización de procesos industriales en la empresa “Comandante Ernesto Che Guevara”.

Sobre la base del problema a resolver se establece la siguiente **la idea a defender:** El diseño e implementación de una aplicación que concentre los datos generados de todas las variables durante el proceso de secado, utilizando las metodologías, tecnologías y herramientas adecuadas; permitirá el análisis del comportamiento energético de los secadores cilíndricos rotatorios en la Empresa “Comandante Ernesto Che Guevara”.

En correspondencia con la idea a defender planteada, se define como **objetivo del trabajo:** Desarrollar una aplicación para la empresa “Comandante Ernesto Che Guevara” que permitirá, de forma rápida y fiable, el análisis del comportamiento energético de los secadores cilíndricos rotatorios.

Para lograr el cumplimiento del objetivo propuesto, se plantean las siguientes **tareas del trabajo:**

1. Analizar las bibliografías de las literaturas científicas relacionadas con el secado de mineral laterítico.
2. Establecer los modelos matemáticos del proceso estudiando.
3. Seleccionar el lenguaje de programación para el desarrollo de la aplicación.

4. Diseñar e implementar un sistema informático para el cálculo de secado del mineral laterítico.
5. Validar el sistema creado.
6. Confeccionar el Manual de usuario del sistema.

Para dar respuesta a estas tareas propuestas se están utilizando los **métodos científicos de la investigación: teórico y empírico**.

Los métodos empíricos de investigación estudian las características y relaciones esenciales del objeto que son accesibles directamente desde la percepción sensorial. (21)

Los métodos teóricos de investigación se aplican durante el proceso de explicación, predicción, interpretación y comprensión de la esencia del objeto. Posibilitan la interpretación conceptual de los datos empíricos, revelan las relaciones esenciales del objeto de investigación que no son observados a simple vista, participan en la construcción del modelo y la hipótesis de la investigación. (21)

Entre los **Métodos Empíricos** se encuentran:

Entrevista: Necesaria en la recopilación de la información para el conocimiento del problema en general y la incorporación de las nuevas funcionalidades del sistema que se pretende diseñar.

Observación: Se utilizó para ver como los usuarios interactúan con las diferentes aplicaciones de su escritorio.

Entre los **Métodos Teóricos** se encuentra:

Hipotético-Deductivo: Este método permitió realizar el debido análisis para el posterior desarrollo del sistema que dará solución al problema existente.

Análisis y síntesis: Fue empleado en la recopilación y el procesamiento de la información obtenida en los métodos empíricos y de esta forma arribar a las conclusiones de la investigación.

Causa-Efecto: En la formulación del problema científico y la situación problemática, que desencadenan la investigación así como el desarrollo del objeto de estudio.

Capítulo 1: Fundamentación teórica: se aborda el objeto de estudio del presente trabajo, trabajos precedentes del proceso de secado, incluyendo las comparaciones entre herramientas, lenguajes de programación y bases de datos, para poder escoger cual se adecua más a las características de esta aplicación.

Capítulo 2. Modelo del dominio y requisitos: se definen las entidades y conceptos principales y partiendo de ellos se representa el modelo de dominio, siendo este el punto de partida para hacer el análisis del sistema a desarrollar. Se especifican las funcionalidades del sistema y se describen detalladamente.

Capítulo 3. Diseño e implementación del sistema: se abordan en detalles los flujos de trabajos relacionados a estas etapas de diseño e implementación que van encaminados a la construcción de la solución; se modelan los diagramas de clases del diseño, modelos de datos, entre otros que construyen las funcionalidades descritas en el capítulo anterior.

Capítulo 4. Estudio de factibilidad: se realiza un estudio de factibilidad y sostenibilidad sobre el sistema, apoyado en el modelo COCOMO II; se obtienen los beneficios tangibles e intangibles, que permiten hacer un análisis de los costos del desarrollo de este sistema.

Para finalizar se muestran las Conclusiones Generales a las que se arribaron, las Recomendaciones que se proponen, la referencia bibliográfica consultada y los Anexos con información necesaria sobre el trabajo de diploma y el Glosario de Términos.

Capítulo 1

Fundamentación Teórica

El secado, desde el punto de vista industrial es un proceso de transferencia de calor y de masa no isotérmico, este proceso permite una reducción en el contenido de humedad del material, lo cual provoca una disminución en el consumo de combustible y las emisiones de gases producto de la combustión en los secaderos. La inserción de un sistema que ayude a mejorar el trabajo en este proceso, así como un mejor estudio del comportamiento de la humedad sin duda será un paso de avance en el mejoramiento de las técnicas utilizadas. Para la elaboración de un software para automatizar el análisis de la humedad durante el secado es necesario el estudio de los trabajos precedentes, así como un análisis de las tecnologías actuales para diseñar, implementar y poner en marcha productos informáticos que se adapten a las exigencias de estos procesos.

En este sentido el **objetivo** del capítulo es:

Establecer el marco teórico que sustenta la investigación, a partir del análisis de los trabajos precedentes y la selección del lenguaje de programación a emplear en la automatización del proceso.

Capítulo 1. Fundamentación Teórica

1.1. Trabajos precedentes sobre los procesos de secado

El secado con el empleo de la energía térmica se ha desarrollado vertiginosamente en el mundo, de acuerdo con las fuentes bibliográficas consultadas (19) ;(31) se destacan dos tendencias fundamentales: el secado bajo parámetros controlados y el secado solar. A este trabajo ofrece particular interés la primera de ellas.

Respecto al primer elemento se encontraron numerosos trabajos experimentales encaminados a describir el comportamiento de la transferencia de calor y masa en el secador y caracterizar la cinética del proceso de secado y la humedad de equilibrio. Los trabajos más pertinentes se analizan a continuación:

Lykov (1954), demostró experimentalmente que el proceso de secado depende del mecanismo de transporte de la humedad en el interior del sólido bajo la acción de un gradiente térmico, de la energía de vaporización y de la difusión del vapor desde la superficie del material hacia los alrededores, a través de la capa límite (23). Este descubrimiento es válido para el secado de los materiales capilaroporosos, grupo al que corresponde el mineral laterítico según la clasificación dada por Kasatkin (1987). (18)

En Cuba se han desarrollado diferentes investigaciones relacionadas con el secado de materiales y productos alimenticios tales como: café, cacao, cebolla, madera, cemento, zeolita, entre otros, todos con resultados satisfactorios para cada uno de estos materiales. Dentro de ellos se puede destacar el trabajo desarrollado por Novoa et. al (1995) para el secado de zeolita a escala de laboratorio, en este se obtienen ecuaciones empíricas que describen el comportamiento del proceso de secado del producto (28) siguiendo el modelo propuesto por Hall (1983), obteniendo además, los valores de humedad de equilibrio para diferentes temperaturas y humedad inicial; finalmente con todos los resultados alcanzados realizaron un software para simular el proceso y predecir su comportamiento bajo determinadas condiciones de operación. (12)

Por otra parte Fitt (1996), afirma que muchos modelos matemáticos de procesos industriales pueden ser escritos como sistemas $n \times n$ de leyes de conservación en términos de n variables independientes.

Toirac et al. (1997) obtiene el tonelaje de combustible por mineral secado y agua evaporada para un flujo de mineral alimentado de 140 t/h.

Eich-Soellner et al. (1997) señalan que la modelación matemática es una herramienta indispensable en el diseño y operación de las plantas químicas, ofreciendo un método numérico para la solución de grandes sistemas de ecuaciones derivados de la modelación de una planta de producción.

Mora (1999) realiza un balance térmico y de masa de los secadores de mineral considerando los gases de hornos y el aire frío como aire secundario o de disolución. También se realizó una valoración económica del proceso para demostrar cómo influye el empleo de gases calientes como aire secundario en el costo de producción y calculó el consumo de combustible en estas condiciones. (25)

También Bachir (2001) realiza un balance térmico y de masa para determinar el consumo de combustible tomando el aire frío y los gases de hornos como aire secundario.

Varios trabajos investigativos se han llevado a cabo en los secadores de la empresa productora de níquel “Comandante Ernesto Che Guevara”, Moya (2008) realiza un análisis de las pérdidas y su influencia en el gasto económico por concepto de consumo de combustible en dicha empresa. Dichos secadores son similares para ambas empresas.

Como se puede observar de este análisis, en nuestro país la literatura recoge pocos intentos de estudios experimentales de los secadores rotatorios al parecer por la dificultad de hacer mediciones en dicho agregado, particularmente en condiciones industriales, ya que carecen de una data precisa y adecuada de la transferencia de calor, masa y del movimiento de la partícula por el interior del equipo, todos estos aspectos son los que han motivado la realización de esta investigación con vista a profundizar en los

parámetros de explotación y funcionamiento en los tambores cilíndricos rotatorios.

1.2. Descripción del flujo tecnológico de la planta

En la Planta de Preparación de Mineral es donde se inicia el proceso productivo de la fábrica según la tecnología carbonato-amoniaca. La misma está formada por las siguientes secciones: Patio de homogenización, Secaderos y Molinos.

Los secaderos cilíndricos rotatorios se dividen en dos tipos, conforme a las direcciones de la corriente de material a ser secado y de los gases de secado. La elección entre un secador rotatorio a contra-corriente o no se efectúa de conformidad con las propiedades del material a secar:

- Grado de secado
- Calor específico
- Capacidad de evaporación de humedad
- Inflamabilidad
- Propiedades de sinterización¹

El mineral procedente de la mina se transporta hacia la Planta de Preparación de Mineral a través de las grúas Gantry o directamente por los transportadores de enlace (TR-14 Y TR-15).

La alimentación a los secaderos se realiza a través de unos desviadores que se encuentran justamente sobre las tolvas de los secaderos. La alimentación al depósito Interior se realiza de la misma forma, o sea, mediante desviadores de mineral que se encuentran situados sobre tres correas colocadas entre los secaderos 2 y 3, 4 y 5 y al final del edificio.

¹ Proceso industrial por el cual se consigue crear piezas que son complicadas de obtener por otros procedimientos como el forjado o el mecanizado. Se basa en reducir el material base a polvo para luego comprimirlo en un molde a una determinada presión y calentarlo a una temperatura controlada.

El mineral almacenado en el depósito interior se usara cuando, por cualquier tipo de avería, se interrumpa el suministro de la materia prima a esta sección. La capacidad de este almacén posibilita a la planta trabajar durante 4 días.

Este mineral que llega a la planta con una humedad promedio de 40% es secado en los tambores cilíndricos rotatorios hasta valores de 4,5-5,5% de humedad, luego es llevado por transportadores de bandas hacia la sección de molienda donde es sometido a un proceso de molienda en los molinos cilíndricos de bolas hasta valores de granulometría para el 85-87% de la cantidad de mineral.

1.2.1. Descripción del proceso de secado en los secadores cilíndricos

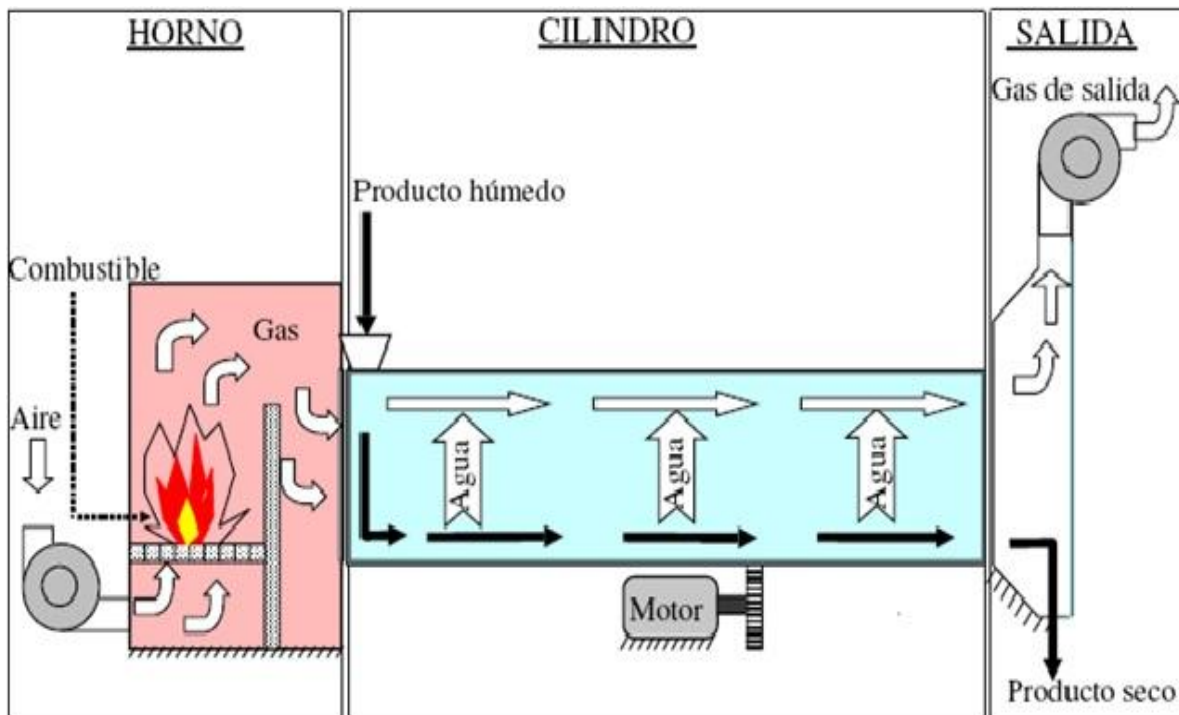


Figura 1.1. Proceso de Secado del Mineral

El mineral al entrar al secadero tiene una humedad de 40 % aproximadamente y debe salir del mismo con 4,5 - 5,5%. Para lograr esto cada secadero posee una cámara de combustión dotada de un quemador de petróleo, a la cual se le suministra aire de combustión, aire de pulverización, aire secundario o gases procedentes de la planta de hornos.

Los gases quemados dentro de la cámara de combustión alcanzan una temperatura de 1500 °C que disminuye hasta 800-850 °C al entrar en contacto

con el aire en exceso que se suministra y que sirve para aumentar el volumen de gases necesarios para secar el mineral. Los gases entran al tambor secador con esta última temperatura (800-850 °C) y salen del mismo a una temperatura de 80 -100 °C.

La entrada de los gases al tambor secador se realiza en dirección a corriente con el mineral alimentado, de forma que el contacto entre los gases calientes y el mineral permita que este último se vaya secando para obtener al final del secadero un producto con las características adecuadas.

Estos gases calientes pueden atravesar el secadero debido a la succión que crea un ventilador centrífugo de tiro inducido de doble entrada (BM-20), situado a la salida del electrofiltro² que posee cada secadero individualmente.

Además, estos gases son capaces de arrastrar con ellos el 28 % del polvo que entra con el mineral o que se forma durante el proceso de secado, y que por lo general posee una granulometría de $-0,074$ mm, el cual será introducido al sistema de colección.

1.3. Sistemas automatizados existentes vinculados al campo de acción

En Mathcad se han implementado fórmulas que no acaparan todo el contenido en el proceso de secado, pero que han ayudado de cierta forma a los investigadores del tema con ayuda de otras herramientas como el Excel de Windows, y el OpenOffice.org Math que crea o edita fórmulas y ecuaciones, por citar algunos ejemplos. Los cuales independientemente que son eficientes y de elevada eficacia en sus soluciones, no cuentan con una integración total de los modelos matemáticos del secado de los cilíndricos rotatorios, por lo que muchos cálculos se deben ser llevados a mano. Además de que algunas de estas herramientas no son gratis pues requieren del pago de licencias y no son multiplataforma.

Como se puede apreciar existe una amplia variedad en cuanto a las publicaciones relacionadas con el secado de materiales sólidos. En los trabajos

² precipitador electrostático

consultados se estudian diferentes aspectos del proceso y sistemas informáticos vinculados a su estudio. Del análisis de estos se derivan las siguientes conclusiones:

- Los aportes científicos de los diferentes autores están encaminados a establecer las leyes matemáticas que rigen la transferencia de calor y masa en el proceso de secado.
- El secado de mineral laterítico en cilindros rotatorios horizontales es un tema novedoso en Cuba por lo que existen pocos trabajos dedicados a su estudio. En la empresa “Comandante Ernesto Che Guevara” se han realizado pocos estudios experimentales debido a lo antes explicado y porque los resultados obtenidos en ellos en otras partes del mundo, no pueden ser generalizados al mineral laterítico debido a que el mismo tiene diferentes propiedades termofísicas y la región de Moa muestra un microclima que difiere del que presentan otras regiones del mundo donde se ha experimentado el proceso.
- Los estudios estadísticos realizado se han llevados a cabo a mano y con ayuda de algunas herramientas informáticas, siendo un método engorroso, pero el uso de un sistema informático que integre todo lo relacionado con el proceso de secado y almacenes sus registros, facilitará el análisis de sus resultados para la creación y desarrollo de nuevas técnicas de evaluación energética.

Los elementos antes expuestos exigen el desarrollo de una aplicación informática, que automatice los modelos matemáticos establecidos por los estudios experimentales en la región de Moa.

1.3.1. Tendencias en tecnología de lenguajes de Programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, especialmente una computadora. Permite a los programadores especificar sobre qué datos la computadora debe operar, cómo estos deben ser almacenados, y qué acciones debe tomar ante cada circunstancia previamente definida. Al ser un estándar de escritura

permite a más de un programador trabajar de forma colaborativa en la construcción de un programa.

En el transcurrir de los años y en la medida en que la tecnología ha ido avanzado, han venido surgiendo diferentes lenguajes de programación, cada uno con características y objetivos específicos distintos pero todos con la misma finalidad, la comunicación hombre-máquina a través de una estructura sintáctica similar al lenguaje común utilizado en la vida diaria, como por ejemplo la utilización de métodos numéricos para resolver problemas.

Los métodos numéricos son herramientas muy poderosas para la solución de problemas usando operaciones aritméticas combinadas con programación por ejemplo, se encarga de diseñar algoritmos para, a través de números y reglas matemáticas simples, simular procesos matemáticos más complejos y puede ser aplicados para resolver prodecidimientos matemáticos en: calculo de derivadas, integrales, ecuaciones diferenciales, operaciones con matrices, interpolaciones, ajuste de curvas, polinomios, etc. Además son de gran utilidad en áreas como: ingeniería informática, ingeniería industrial, ingeniería química, ingeniería civil, ingeniería mecánica, ingeniería eléctrica, etc. Por estas razones es de gran importancia el estudio de esta materia.

En este trabajo se analizarán tres de ellos: C, C++ y Java. A continuación se enuncian las características más relevantes de cada uno según la bibliografía consultada en AmericaTI (2006). (1)

1.3.2. Lenguaje de Programación C

El lenguaje C es creado entre los años 1972-1973 por un equipo de programadores de los antiguos Laboratorios Bell de AT&T, como un lenguaje de propósito general. Es un lenguaje que aunque por muchos años se le consideró sumamente expresivo y muy económico debido a la cantidad reducida de palabras claves y a las potencialidades de algunos de sus operadores como los punteros, en la actualidad al enfrentarse con estructuras de programación cada vez más complejas las implementaciones en lenguaje C tienden a tornarse no tan sencillas e inseguras frente a equivalentes en otros lenguajes. Fue considerado por mucho tiempo un buen ejemplo de un lenguaje

consistente y sin ambigüedades notorias, especialmente entre otros de su misma época. Sin embargo sus propios creadores reconocen en él ciertos inconvenientes en la notación, que aunque no es estrictamente un error y puede ser evitable, originan confusiones y uno de los principales problemas son los aspectos que son dejados a criterio del implementador, por ejemplo el tamaño de los tipos de datos, lo que puede convertirse en problemas de portabilidad. Sin embargo fue uno de los primeros “casos de éxito” de portabilidad gracias a estándares abiertos.

Los mecanismos de este lenguaje para proporcionar tipos de datos estructurados se consideran rudimentarios. Se suelen utilizar estructuras para definir tipos de datos complejos (unión de varios tipos de datos), arreglos para colecciones homogéneas de longitud fija, sin embargo carece de tipos de datos para representar cadenas de texto, las cuales son manejadas mediante un arreglo de caracteres.

En cuanto al nivel de empaquetamiento, sólo proporciona dos niveles de componentes visibles dentro del archivo de código fuente, y componentes visibles globalmente (funciones y variables).

Aunque no es necesariamente el lenguaje más conveniente para el acceso a los sistemas de bases de datos, permite un uso casi ilimitado, puesto que casi la totalidad de estas aplicaciones que proporcionan una interfaz de programación permiten el acceso mediante él.

Por poseer una estructura de datos muy simple, la traducción al “lenguaje de máquina”, resulta generalmente directa, el programador está muchas veces cerca de escribir en un lenguaje similar al que el CPU (Unidad Central de Procesamiento) comprende, lo que hace que este programa tenga un excelente rendimiento.

Este lenguaje no fue concebido para ser sencillo de aprender, su objetivo principal era ser eficiente, sin embargo suele ser utilizado para construir componentes básicos o de bajo nivel.

Dispone de una amplia variedad de opciones en cuanto a librerías para diversos propósitos. El único aspecto cuestionable radica en que muy pocas de

estas librerías están estandarizadas del mismo modo que el lenguaje. Por ejemplo, el estándar de lenguaje C actualmente incluye una librería (conocida como la "librería estándar") que permite realizar una gran cantidad de tareas esenciales (mostrar un mensaje, grabar en un archivo de disco, calcular funciones trigonométricas, procesar cadenas de caracteres, etc.) pero que para muchas aplicaciones modernas, resulta francamente insuficiente. (1)

1.3.3. Lenguaje de Programación C++

En 1983 el lenguaje C se rebautiza como C++. Posteriormente el lenguaje fue estandarizado (ISO C++). Desde sus inicios, intentó ser un lenguaje que incluyera completamente al lenguaje C (quizás el 99% del código escrito en C es válido en C++), pero al mismo tiempo incorpora muchas características nuevas útiles en diversos contextos, tales como: POO, excepciones, sobrecarga de operadores, templates o plantillas. Sin embargo por ser un fruto del propio C, comparte alguno de sus inconvenientes y uno de ellos es el mencionado problema con definición del tamaño de los datos por parte de los implementadores, lo cual puede convertirse en un problema de portabilidad, problemas que van siendo cada vez menos frecuentes.

En cuanto a los tipos de datos proporcionados, brinda facilidades que permiten la creación de estructuras de datos muy poderosas y fuertemente integradas en el lenguaje, un ejemplo de ello lo constituyen los tipos de datos creados por el desarrollador con diversas operaciones asociadas, resultando una extensión natural de los tipos de datos primitivos, alcanzando con ello un alto grado de claridad.

Además de contener funciones y variables, proporciona dos niveles adicionales de empaquetamiento, las clases y los namespace (espacio de nombres).

Por otra parte, muchos sistemas de bases de datos proporcionan una interfaz mejorada orientada a objetos disponible en este lenguaje, lo que hace bien amplia su utilización para el acceso a los sistemas de bases de datos.

Actualmente los compiladores de C++ generan un código muy difícil de superar por una hábil implementación equivalente en C y su rendimiento es también envidiable.

Al igual que el C, no fue creado para ser sencillo de aprender, pero sí para ser eficiente. Este lenguaje dispone de una biblioteca más extensa la cual incluye de hecho a la "librería estándar de C" así como la famosa "STL" (Standard Template Library) que implementa diversas estructuras de datos de manera genérica, así como muchos algoritmos de uso común. De igual modo, muchas librerías de terceros están disponibles para propósitos más especializados (por citar dos ejemplos, las bibliotecas Xerces de procesamiento de documentos XML, y Boost de propósito general). (1)

1.3.4. Lenguaje de Programación Java

Java, originalmente denominado "Oak". Sus inicios datan de 1991 cuando James Gosling encabezó un proyecto cuyo objetivo original era implementar una máquina virtual ampliamente portable y un lenguaje de programación, ambos orientados a dispositivos "embedded" (procesadores incorporados en diversos dispositivos de consumo masivo como tostadoras, VCR's (Video Cassette Recorder) y teléfonos móviles.).

La sintaxis del lenguaje heredó características de C y C++, adoptando una muy similar a la del C++. En la actualidad su uso es promovido para el desarrollo de aplicaciones empresariales del lado del servidor. Actualmente, dentro de los lenguajes populares es uno de los mejores en cuanto a definición, debido a que goza de total independencia del implementador del lenguaje y de sus clases auxiliares. Proporciona los tipos de datos primitivos similares a los de C++, sólo que carece de punteros y mediante su biblioteca de clases estándar proporciona todas las estructuras contenedoras "clásicas" antes mencionadas.

Tiene cuatro niveles de empaquetamiento: variables y funciones, al igual que los lenguajes anteriores y otros dos propios de él, denominados: clases y paquetes.

Este lenguaje cuenta con una interfaz orientada a objetos para acceder de un modo portable a cualquier base de datos, promoviendo la portabilidad. Es poseedor de una extensa biblioteca utilitaria y la portabilidad alcanzada es cualitativamente superior a la que se puede obtener con los lenguajes C/C++.

Desde sus inicios se concibió como un lenguaje muy fácil de comprender y utilizar, sin que esto signifique que su aprendizaje sea trivial.

Java como los otros dos lenguajes analizados es de propósito general, sin embargo es muy utilizado en el contexto de servidores Web, acompañado en muchos casos de una arquitectura multicapas. (1)

1.4. Fundamentación del lenguaje a utilizar

En la actualidad Cuba está inmersa en un proceso de cambio de su estructura tecnológica, a pesar de las dificultades impuestas por el bloqueo económico, nuestro país ha tenido que tomar decisiones para enfrentar este problema, por lo que se ha tomado la medida de ir emigrando cada vez más al software libre. Una solución que han adoptado hoy muchos desarrolladores cubanos es el hacer software en el lenguaje de programación Java capaz de funcionar en varios entornos.

Teniendo en cuenta el conocimiento que se ha adquirido durante el estudio de la carrera y siendo la política de nuestro centro de utilizar el lenguaje de programación Java para desarrollar aplicaciones de escritorio hemos seleccionado éste. Zukowski, (2006), planteó una breve descripción de las **características generales:**

Simple. Java tiene mucha similitud con C/C++, pero es menos complicado. Java incorpora nuevas tareas como un recolector de basura y elimina aspecto de C++ confuso y muy pocos utilizados como la sobre carga de operadores. Otro aspecto de la simplicidad es que no hay nada nuevo en Java, todo procede de algún otro lugar.

Orientada a Objeto. Java está orientado a objetos, ya que un buen diseño conlleva a componentes reutilizables, extensibles y sostenible. Esto permite controlar los cambios que puedan producirse a lo largo del tiempo, dado que el trabajo principal de estos objetos es intercambiar mensajes entre sí.

Distribuido. Inicialmente, Java podía acceder a objetos distribuidos mediante protocolos estándares en TCP/IP como HTTP y FTP. En las últimas versiones de Java Platform, se pueden invocar métodos en un equipo remoto tan

fácilmente e invisible como podría hacerlo en su mismo espacio, mediante protocolos comunes como CORBA y RMI, además de incorporar servidores Web.

Interpretados. En lugar de ser compilados en ejecutables nativo, el código de java es traducido en código bytes no asociados a una plataforma y ser ejecutados a través de una Máquina Virtual de Java (JVM).

Robusto. Java contiene varias funciones de integridad que mejora la fiabilidad de un programa entre las que se puede mencionar, que no tiene punteros, lo que no se puede hacer referencia a un puntero de la memoria, corrompiendo accidentalmente la memoria. Realiza la recolección de elementos no reutilizados.

Seguro. Muchas de las cosas que hacen que Java se robusto también garantizan la seguridad del programa que se está ejecutando, pues java tiene muchas funciones de seguridad integrada. Esto garantiza la seguridad del código que se está ejecutando y evita que código no seguro realice operaciones seguras.

De arquitectura neutral. Esto hace referencia a código de bytes neutrales de Java. En lugar de compilarlos en binarios específicos de cada plataforma, los programas de Java se realizan para que se ejecuten en cualquier equipo sin recompilarlos o revincularlos.

Portátil. El objetivo de Java es ser capaz de tomar el mismo programa y ejecutarlos en cualquier arquitectura.

Alto rendimiento. Los códigos de bytes de plataformas neutra realmente pueden convertirse en tiempo de ejecución al código particular de la maquina que se esté ejecutando, ejecutando casi tan rápidamente como compilado de forma nativa C y C++.

Multitarea. Permite ejecutar varias tareas o subprocesos simultáneamente. El lenguaje de programación Java y las bibliotecas estándar incluyen muchos recursos que permiten compartir memoria y la comunicación entre los subprocesos. (35)

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

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. (27)

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. La plataforma NetBeans es un IDE de código abierto escrito completamente en Java. Tiene soporte para varios lenguajes como PHP, JavaScript, HTML, CSS. Es un producto libre y gratuito sin restricciones de uso. Esta plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. (32)

El IDE NetBeans IDE 7.0 corre en diferentes sistemas operativos entre los que podemos mencionar linux, Mac OS X, Solaris y Windows. Como se puede observar se ha lanzado con soporte para la mayoría de las plataformas. Como requerimiento se necesita tener previamente instalado el JDK 5.0 o 6.0.

1.5. Patrones Arquitectónicos

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

(MVC). Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

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

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

Ventajas del Modelo Vista Controlador

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

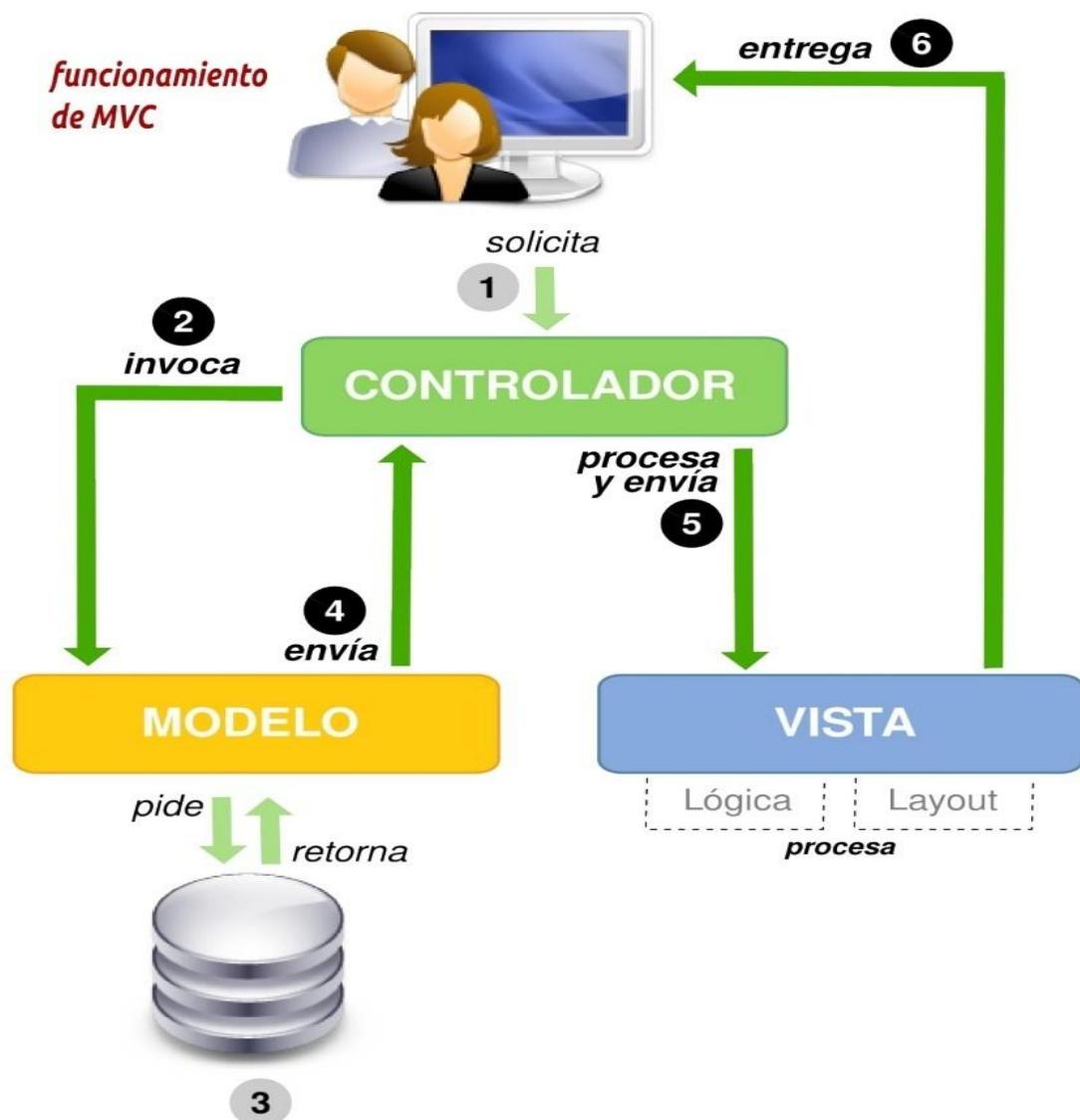


Figura 1.2. Funcionamiento del patrón MVC

1.6. Bases de Datos Embedded

Cierto es que generalmente los requerimientos de las bases de datos para almacenar información nos empujan a utilizar motores donde el rendimiento, la integridad referencial, la concurrencia, etc. son factores determinantes en la selección de un Sistema Gestor de Base de Datos.

Sin embargo, hay ocasiones en donde los requerimientos son otros, como la portabilidad, escasa memoria, rapidez, entre otros. Es allí donde han surgido varias opciones que se han hecho un hueco en el mercado al soportar todos

estos requisitos, e incluso dotando pequeños motores con las características de algunos grandes.

Este tipo de motores son por lo general denominados Bases de Datos Embedded (empotradas, embebidas), que se alinean a los requerimientos antes mencionados. Como ejemplos de este tipo de bases de datos están Apache Derby (descendiente directo de Cloudscape de IBM³), H2, HSQLDB, Firebird Embedded, SQLite, entre otros, que merecen explicar sus características cada una en un artículo independiente. Por lo general, este tipo de bases de datos vienen del movimiento Open Source⁴, aunque también hay algunos motores embedded de origen propietario, como SQL Server Mobile, Oracle Embedded (en varios sabores: Berkeley DB, Timesten, Oracle Database Lite), entre otros. (4)

1.6.1. Apache Derby

Apache Derby tiene su origen en la empresa Cloudscape Inc, en Oakland, California que se fundó en 1996 para desarrollar una tecnología de base de datos para Java.

La primera versión del motor de base de datos, que por entonces se llamó JBMS, tuvo lugar en 1997. Posteriormente el producto fue renombrado como Cloudscape y aparecieron versiones nuevas cada seis meses. En 1999 Informix Software, Inc., adquirió Cloudscape, Inc. En 2001 IBM adquirió los activos de Informix Software, incluyendo Cloudscape. El motor de base de datos fue renombrado a IBM Cloudscape y continuaron apareciendo versiones, enfocadas principalmente a usos embebidos en productos Java de IBM y middleware.

En agosto de 2004 IBM cedió el código a la Apache Software Foundation para Derby, un proyecto patrocinado por el proyecto Apache DB. En julio de 2005 el

³ El mayor fabricante de ordenadores del mundo. IBM es el inventor del PC y con sus sistemas medios (AS/400) y sus grandes ordenadores (mainframes) han revolucionado el mundo de la empresa. Es también uno de los mayores vendedores de software, servicios y equipos de comunicaciones.

⁴ Código abierto.

proyecto Derby continuó desarrollándose como subproyecto base de datos de alto nivel en Apache. Sun se unió al proyecto

Derby con el objetivo de utilizar como componente en sus propios productos, y con el lanzamiento de Java 6 en diciembre de 2006, Sun comenzó a empaquetar Derby en el JDK llamado Java DB.

En marzo de 2007 IBM anunció que podría comenzar a hacer marketing y dar soporte para el producto Cloudscape, pero que continuaría contribuyendo con el proyecto Apache Derby. (33)

Según la bibliografía estudiada (Wikipedia, 2009), Derby posee las siguientes **características** (33):

- APIs para JDBC y SQL. Soporta todas las características de SQL92 y la mayoría de SQL99. La sintaxis SQL usada proviene de IBM DB2.
- Su código mide alrededor de 2000KB comprimido.
- Soporta cifrado completo, roles y permisos. Además posee SQL SCHEMAS para separar la información en una única base de datos y control completo de usuarios.
- Soporta internamente procedures, cifrado y compresión.
- Trae soporte multilenguaje y localizaciones específicas.
- A partir de la versión 10.4 trae un sistema simple de replicación maestro-esclavo.
- Transacciones y recuperación ante errores ACID.
- Posee tres productos asociados a la marca:
 - Derby Embedded Database Engine: El motor propiamente dicho.
 - Derby Network Server: Permite convertir Derby en una base de datos que sigue el modelo cliente-servidor tradicional.
 - Database Utilities: Un paquete de utilidades.

1.6.2. Firebird

Firebird se deriva del código fuente de InterBase 6.0, de Borland. Es de código abierto y no tiene licencias duales. Tanto si lo usas en aplicaciones comerciales o de código abierto, ¡es totalmente LIBRE!

La tecnología de Firebird ha estado en uso por 20 años, lo que lo hace un producto muy estable y maduro. Es un poderoso y completo RDBMS. Puede manejar bases de datos desde solo unos cuantos KB hasta muchos Gigabytes con muy buen desempeño y prácticamente libre de mantenimiento.

Sus principales características son (6):

- Completo soporte para Procedimientos Almacenados y Disparadores.
- Transacciones 100% ACID.
- Integridad Referencial.
- Arquitectura multi-generacional.
- Bajo consumo de recursos.
- Completo lenguaje interno para procedimientos almacenados y disparadores (PSQL).
- Soporte para Funciones Externas (UDFs)
- Poca o ninguna necesidad de DBAs especializados.
- Prácticamente no requiere configuración - solamente instalas y ¡comienzas a usarla!
- Gran comunidad y muchos sitios donde puedes encontrar excelente soporte gratuito.
- Versión incrustada - ideal para crear catálogos en CDROM, versiones mono usuario, de evaluación o portátiles de las aplicaciones.
- Docenas de herramientas de terceros, como herramientas de administración gráficas, herramientas de replicación, etc.
- Escritura segura - recuperación rápida, ¡sin requerir logs de transacciones!

- Muchas formas de acceder a tu base de datos: nativo/API, drivers dbExpress, ODBC, OLEDB, proveedor .Net, driver JDBC nativo tipo 4, módulo Python, PHP, Perl, etc.
- Soporte nativo para todos los principales sistemas operativos, incluyendo Windows, Linux, Solaris, MacOS.
- Copias de seguridad incrementales.
- Disponibilidad de binarios en arquitectura de 64bits.
- Implementación completa de cursores en PSQL.
- Tablas de Monitoreo.
- Disparadores a nivel de Conexión y Transacción.
- Tablas Temporales.

1.6.3. Berkeley DB (BDB)

BDB se creó con la intención de eliminar código propietario de AT&T en BSD. Hace unos años el lado oscuro compró SleepyCat (la empresa que hace BDB) y le agregó funcionalidades súper-útiles como una versión que almacena los datos en XML así podemos hacer consultas con XQuery, una versión casi igual a BDB pero 100% Java, soporte para SQL (que en realidad es un SQLite que almacena en BDB). Las db pueden ser de hasta 256 terabytes. (5)

1.7. Fundamentación de la Base de Datos a utilizar

Un Sistema de gestión de Base de Datos consiste en un conjunto de programas, procedimientos y lenguajes que nos proporcionan las herramientas necesarias para trabajar con una base de datos. Incorporar una serie de funciones que nos permita definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos.

Derby es una fuente abierta de pura Java Database Management System (DBMS). Lo tome como el DBMS porque está libremente disponible, integra perfectamente con NetBeans, ejecuta en todas las plataformas que se ejecuta en NetBeans, y lo más importante, es mucho más simple de instalar y administrar que los tradicionales DBMS.

1.8. Embarcadero ER/Studio 8.0.0

Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. ER/Studio se diseña para hacer más fácil de entender el estado actual de los datos de las empresas. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. (9,10)

Potencial de ER/Studio: Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, ER/Studio se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia. Con el potencial y la facilidad de empleo de ER/Studio's que modela el ambiente, será productivo rápidamente y podrá casi demostrar resultados inmediatamente después de la instalación.

Diagramas: La creación de diagramas es clara y rápida. Tiene la posibilidad de realizar diagramas con desempeño rápido. También es posible cambiar el estilo de las líneas, los colores, tipos de letra, niveles de acercamiento, y modelos de despliegue. Es posible crear subvistas para separar y manejar áreas importantes. ER/Studio automáticamente mantiene todas las dependencias entre subvistas y el diagrama completo. El Explorer Navigation facilita el trabajo hasta con los diagramas más grandes. Si se está trabajando con un modelo largo de Datos, ER/Studio ofrece un aumento en la ayuda y fácil navegación en sus modelos. La Apreciación global (overview). Se usa el browser Explorer para encontrar y seleccionar entidades. Un solo clic inmediatamente enfoca una ventana de diagrama. (10)

Ventajas:

- Soporta el proceso de diseño interactivo inherente en el ciclo de vida de la aplicación.
- ER/Studio puede documentar automáticamente un diagrama entero, generando un conjunto integrado de reportes HTML sofisticados que múltiples usuarios pueden compartir en Internet.

- Calidad de presentación en los reportes. Además de los reportes de HTML, ER/Studio puede generar reportes de alta calidad con un formato de texto amplio que está disponible para presentaciones profesionales.
- Soporta metodología de Yourdon, con diagramas relación-entidad y modelos IDEF1.

1.9. Metodología de desarrollo

Según plantea Sergi Blanco Cuaresma la metodología de desarrollo constituye una colección de documentos formales referentes a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software. El núcleo de cualquier metodología de desarrollo se encuentra constituido por documentos que detallan cada una de las fases de desarrollo. (7)

A consideración de la autora una metodología de desarrollo constituye una guía para la realización de un proyecto encaminada a ayudar a los desarrolladores, ofreciendo un conjunto de procedimientos, técnicas, y documentos que guardan un historial del proceso de desarrollo del software.

1.9.1. Metodologías tradicionales

Las metodologías tradicionales son también denominadas metodologías clásicas, las cuales están guiadas por una fuerte planificación durante todo el proceso de desarrollo basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, las mismas generan un gran número de artefactos⁵ que registran lo que se ha desarrollado en las distintas etapas del software. Además estas metodologías ofrecen una cierta resistencia a los cambios, siguen un proceso controlado por numerosas políticas y contemplan un contrato prefijado sobre las funcionalidades que deberá poseer la solución informática dada. Cuentan a su vez con la peculiaridad de que el cliente interactúa con el equipo de desarrollo mediante reuniones en las cuales se evalúan los adelantos del software. (15)

⁵ Documentos, diagramas, modelos, etc.

1.9.1.1. Rational Unified Process (RUP en español Proceso Unificado Racional)

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. Fue creado por Software Rational Corporación, una división de IBM desde 2003. (34)

El ciclo de vida de RUP se caracteriza por:

Dirigido por Casos de Uso (CU): Los CU reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los CU guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los CU. (8)

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML. (8)

Iterativo e incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. (8)

Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son miniproyectos. (30)

Ventajas de RUP (30):

- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.
- No se necesita interacción con el cliente en todo el ciclo de desarrollo.
- Gran número de documentación que proporciona un control estricto de todo lo que pasó a lo largo del proyecto.
- Se define el alcance del proyecto desde un principio.

Desventajas de RUP (30):

- La evaluación de riesgos es compleja.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

1.9.1.2. Microsoft Solution Framework (MSF en español Marco de Soluciones Microsoft)

Se basa principalmente en los modelos espiral⁶ y cascada⁷ (hitos y fases). Fue desarrollado por Microsoft, tiene como principios fundamentales la comunicación (entre cliente/usuario y entre desarrolladores), una capacitación de las personas (disciplina de disponibilidad) es decir cumple con el proceso de formación de personal al compartir los roles entre todo el equipo de trabajo. MSF es un proceso donde se debe crear versiones para el negocio de cada cliente. Otro principio es la inversión de calidad (tiempo, trabajo, dinero), ya que cada proyecto es una inversión no un gasto. (14)

MSF se compone de 2 modelos y 3 disciplinas. Modelo de Equipo y de Proceso. Disciplina de Administración de Proyecto, Administración de Riesgos y Administración de la Preparación (14)

Ventajas de MSF (29):

- Tiene facilidad de soporte y mantenimiento.

⁶ El modelo espiral de los procesos software es un modelo del ciclo de meta-vida. En este modelo, el esfuerzo de desarrollo es iterativo. Tan pronto como uno completa un esfuerzo de desarrollo, otro comienza.

⁷ Contempla el desarrollo de software a través de una secuencia simple de fases, donde cada fase tiene un conjunto de metas bien definidas.

- Está basado en el trabajo en equipo.
- Se puede utilizar para proyectos de cualquier magnitud.
- Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto.
- Gracias a Microsoft, este modelo cuenta con plantillas que nos ayudan en el proceso de documentación.

Desventajas de MSF (29):

- Este modelo utiliza demasiada documentación en sus fases.
- El análisis de riesgo suele llevar mucho tiempo frenando el avance del proyecto.
- Al ser un modelo de Microsoft implica que se tiene que utilizar herramientas solo de Microsoft.

1.9.2. Metodologías ágiles

Cuando el desarrollo de un software se realiza de manera incremental⁸, cooperativa⁹, sencilla y adaptable¹⁰ se está en presencia de una metodología ágil. Las metodologías ágiles se encuentran preparadas para cambios durante el proyecto, siguen un proceso menos controlado, con pocos principios y sin un contrato tradicional y de existir este es bastante flexible. Requiere que el cliente sea parte activa del equipo de desarrollo. En este tipo de metodologías son generados pocos artefactos. (15)

1.9.2.1. Scrum

La metodología Scrum¹¹ tiene sus orígenes en el año 1986, en la misma se asume que el proceso de desarrollo de software es impredecible, y lo trata como a una caja negra controlada, en vez de manejarlo como un proceso completamente definido. (26)

⁸ Entregas pequeñas de software, con ciclos rápidos.

⁹ Cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación.

¹⁰ Permite realizar cambios de último momento.

¹¹ Hace honor a una estrategia de juego que se utiliza en el fútbol rugby.

Scrum no trata el proceso de desarrollo de software como un proceso lineal, en el que se sigue la secuencia de análisis, diseño, codificación y prueba. En Scrum, el proyecto puede iniciarse con cualquier actividad, y cambiar de una a otra en cualquier momento. Un proyecto administrado mediante Scrum se organiza en iteraciones, llamadas sprints¹², que normalmente tienen entre dos y cuatro semanas de duración. Al principio de cada sprint se establece una lista de requerimientos llamada backlog¹³, que debe completarse cuando éste finalice. A diario se realizan breves reuniones del equipo de desarrollo, en las que se exponen los avances y los problemas encontrados, y se señalan posibles caminos para resolverlos. (26)

El proceso de desarrollo Scrum se compone de cinco actividades principales: revisión de los planes de release¹⁴, distribución, revisión y ajuste de los estándares de producto, sprint, revisión de sprint, y cierre. (26)

1.9.2.2. Extreme Programming (XP en español Programación Extrema)

Surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. Es una de las llamadas metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes. La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. Además alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo. La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance. Uno de sus requerimientos es tener al cliente disponible durante todo el proyecto, formando parte del grupo de desarrollo. (17)

Desventajas: Una de las críticas a XP es la dificultad de estimar cuánto va a costar un proyecto. Dado que el alcance del mismo no está completamente

¹² Término que denomina a una iteración que está acotada en el tiempo, durante la cual el equipo trabaja para convertir elementos del backlog del producto en un incremento del producto potencialmente productivo.

¹³ Lista de tareas que define el trabajo a realizar por los miembros del equipo de Scrum durante un sprint. En español significa acumulación de trabajo.

¹⁴ Revisión o versión.

definido al comienzo, y que esta metodología es expresamente abierta a los cambios durante todo el proceso, por lo que se torna sumamente difícil estimar un presupuesto previo. (17)

Otra crítica a XP está dada por su baja documentación, pensando en el posterior mantenimiento del sistema. Si bien durante el proyecto el equipo tiene en mente todas sus particularidades, hay que prever que pasará luego de entregado, cuando el equipo se disuelva y sea necesario realizar alguna mejora.

1.9.3. Fundamentación de la metodología a utilizar

Después de realizar un análisis e investigación de estas metodologías, se selecciona RUP para el desarrollo del software, porque es la más completa y abarcadora, pues como señalan algunos autores, las otras metodologías son casos particulares de esta. Además, XP y MSF presentan algunas debilidades, lo que representa riesgos considerables, como dificultades a la hora de una buena obtención de requisitos para el sistema. RUP es adaptable, según el tipo de proyecto, así serán las características del mismo, haciéndose énfasis en aquellos flujos de trabajo durante la vida del software, que reporten más importancia y sean indispensables. RUP permite trazarse planes de riesgos y pruebas durante el ciclo de vida. Contiene artefactos que son diseñados durante las diferentes fases, los cuales describen detalladamente las características del software desde que se realiza el análisis del problema hasta la entrega final del producto.

1.10. Herramientas de modelado

Las herramientas de modelado fueron creadas con el fin de desarrollar programas, utilizando técnicas de diseño y metodologías bien definidas, soportadas por herramientas automatizadas. Suelen llamarse herramientas CASE, significado de las siglas: Computer Aided Software Engineering, en español Ingeniería de Software Asistida por Computadora. (22)

Es un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. (22)

La autora del presente trabajo de diploma sostiene que una herramienta CASE es un instrumento que permite desarrollar soluciones informáticas abarcando todo el ciclo de vida de un software¹⁵, por lo que constituye un apoyo para los desarrolladores del mismo.

1.10.1. Visual Paradigm

Visual Paradigm for UML (VP-UML) es una herramienta de diseño Unified Modeling Language (UML) cuyo significado en español es Lenguaje Unificado de Modelado y además constituye una herramienta CASE diseñada para la ayuda al desarrollo de software. Es una herramienta UML profesional que soporta el ciclo de vida completo de desarrollo de un programa: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (11)

Características de Visual Paradigm (11):

- Licencia: gratuita y comercial.
- Producto de calidad.
- Soporta aplicaciones web.
- Varios idiomas.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

1.10.2. Rational Rose

Rational Rose es una herramienta CASE que propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del mismo. Permite crear y refinar estas vistas estableciendo de esta forma un modelo completo que representa el dominio del problema y de las características del sistema. (24)

¹⁵ Programa.

Rational Rose utiliza un proceso de desarrollo iterativo controlado. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que estos se hagan mínimos. Cuando la implementación pasa todas las pruebas que se determinan en el proceso, esta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. (24)

Características (24).

- Capacidad para proporcionar el desarrollo iterativo.
- Mantiene la consistencia de los modelos del sistema.
- Chequeo de la sintaxis UML.
- Generación de código a partir de los modelos.
- Ingeniería Inversa (crear modelo a partir código).
- No es gratuito, se debe hacer un previo pago para poder adquirir el producto.
- Admite la integración con otras herramientas de desarrollo.

1.10.3. Fundamentación de la herramienta de modelado a utilizar

Para el modelado del sistema se utiliza como herramienta CASE a Visual Paradigm en su versión 6.4, puesto que es una herramienta potente y distribuida bajo una licencia gratuita y comercial, es fácil de instalar y actualizar permitiendo una excelente compatibilidad entre ediciones, lo que facilita el trabajo de modelado de la aplicación. La herramienta Rational Rose no se escoge puesto que no es libre ni gratuita.

1.11. Conclusiones del Capítulo

En la revisión bibliográfica se evidencia pocos trabajos de investigaciones realizadas en los cilindros rotatorios horizontales de la Empresa “Comandante Ernesto Che Guevara”; ninguno con el fin de automatizar el procedimiento para su evaluación energética.

La herramienta CASE Visual Paradigm en su versión 6.4, es la más adecuada para el modelado del sistema por su potencia, compatibilidad y licencia gratis.

Capítulo 2

Modelo del Dominio y Requisitos

El modelado de dominio contribuye a la comprensión del contexto del sistema, así como de sus requisitos, el cual captura los tipos más importantes de objetos del dominio para representar las “cosas” que existen o los eventos que suceden en el entorno en el que se trabaja. Todo ello ayuda a los usuarios a entender el contexto en el que va dirigido el sistema. Es decir la selección de un modelo de dominio puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño del sistema, en el cual se precisa los requisitos funcionales y no funcionales además se definen las principales entidades del dominio.

El objetivo de este capítulo es:

Establecer el diagrama del modelo de dominio, los requerimientos funcionales y no funcionales del sistema.

Capítulo 2. Modelo del Dominio y Requisitos

2.1. ¿Por Qué Modelo de Dominio?

Teniendo en cuenta que el negocio tiene un bajo nivel de estructuración y que estudiaremos cómo se representa el cálculo del secado de mineral laterítico en cilindros rotatorios horizontales a partir de datos medidos por los diferentes dispositivos electrónicos y no cuáles son los procedimientos llevados a cabo por los investigadores para obtener estos datos, se realiza un modelo del dominio. No es de nuestro interés saber cómo se adquieren los datos que se utilizan para la obtención del modelo, ni comprender las funcionalidades de la empresa que realiza el secado del mineral, lo que nos interesa es, una vez obtenido estos, cómo se evalúa el cálculo para el secado del mineral.

2.2. Definición de las entidades y conceptos fundamentales

Entre las entidades y conceptos que se utilizan en el dominio encontramos las siguientes:

- **CYTEC:** Es un software de tipo SCADA que almacena los datos medidos por los diferentes dispositivos electrónicos (termopares, termoresistencia, etc.) que se encargan de medir en distintos puntos requeridos en el secador rotatorio.
- **Jefe de Operaciones:** Especialista que se encarga de verificar el funcionamiento del CYTEC y a partir de ello genera un reporte para su posterior procesamiento.
- **Reporte:** Forma de representación de la información almacenada en la base de datos a partir de las mediciones realizadas por el CYTEC.
- **ExcelClient:** Es un software que proporciona una gran variedad de opciones de cálculo y gráficos, para visualizar y manipular los datos de las tablas que se encuentran en el reporte.

2.3. Representación del diagrama de clases del Modelo del Dominio

El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema por medio de un Diagrama de Clases el cual se muestra seguidamente:

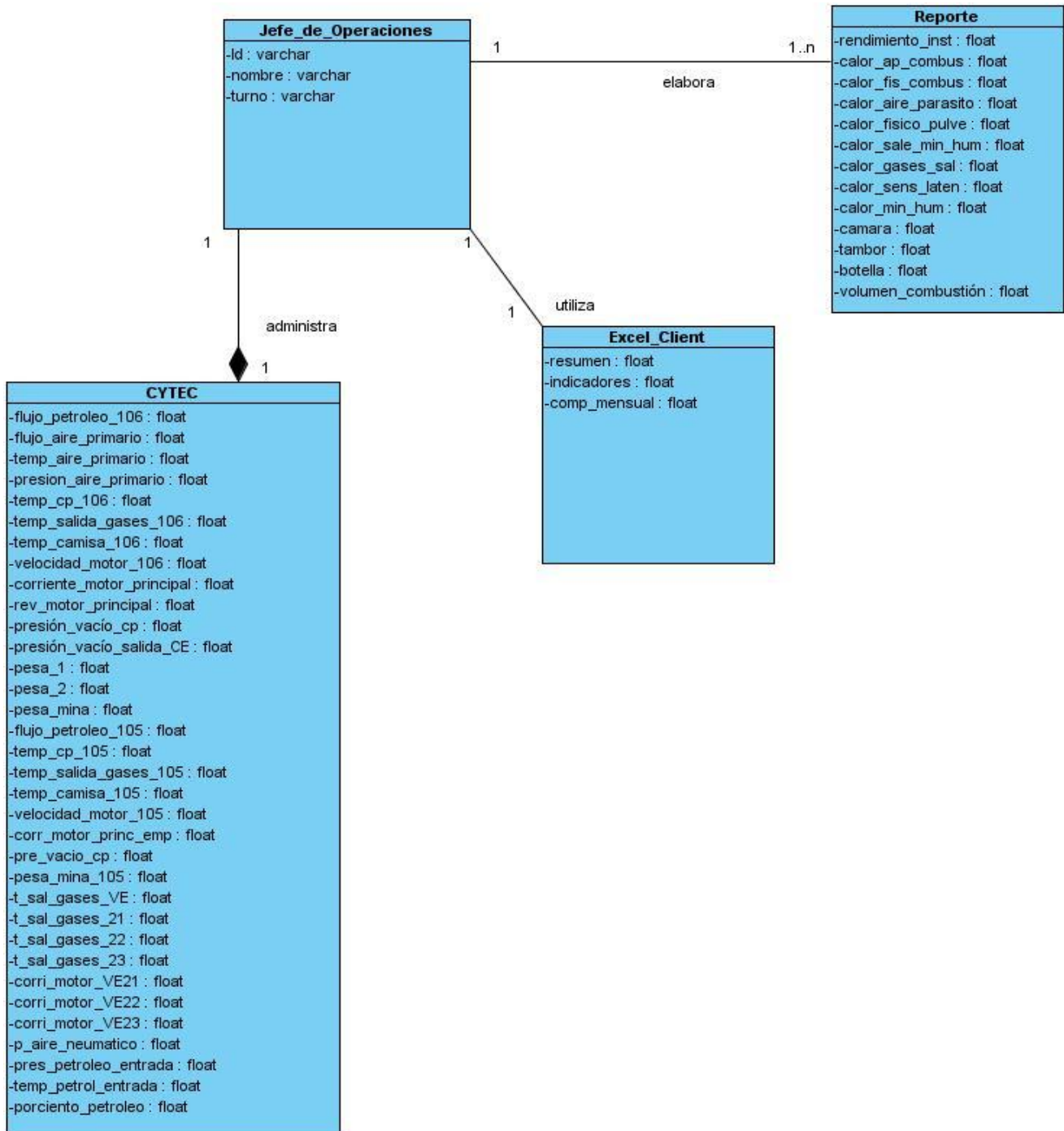


Figura 2.1. Diagrama de Modelo de Dominio

2.4. Requerimientos

Los requerimientos para un sistema de software determinan lo que hará el sistema y definen las restricciones de su operación e implementación. (27)

2.4.1. Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los mismos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importarle con que propiedades o cualidades se relacionen con el producto. (28)

La aplicación a desarrollar debe ser capaz de:

RF- 1. Administrar Base de Datos

1.1. Guardar Base de Datos

1.2. Abrir Base de Datos

1.3. Cerrar Conexión de la Base de Datos

RF- 2. Calcular el calor aportado por el diesel (Q_c)

RF- 3. Calcular los volúmenes de la combustión

RF- 4. Calcular el calor físico aportado del aire primario de combustión (Q_{fa})

RF- 5. Calcular el calor aportado por el aire parásito (Q_{ap})

RF- 6. Calcular el calor físico del aire de pulverización (Q_{pu})

RF- 7. Calcular el calor aportado por el mineral húmedo (Q_{ch})

RF- 8. Calcular el calor que sale con el mineral húmedo (Q_{cs})

RF- 9. Calcular el calor sensible de los gases a la salida del secador (Q_{gs})

RF- 10. Calcular el calor sensible y latente de vaporización del agua contenida en el mineral (Q_{va})

- RF- 11. Calcular las pérdidas por transferencia de calor Cámara de paso (QP_1)
- RF- 12. Calcular las pérdidas por transferencia de calor Tambor secador (QP_2)
- RF- 13. Calcular las pérdidas por transferencia de calor Botella (QP_3)
- RF- 14. Calcular el Rendimiento de Instalación
- RF- 15. Mostrar Reporte de (Q_c)
- RF- 16. Mostrar Reporte de Volumen de Combustión
- RF- 17. Mostrar Reporte de (Q_{fa})
- RF- 18. Mostrar Reporte de (Q_{ap})
- RF- 19. Mostrar Reporte de (Q_{pu})
- RF- 20. Mostrar Reporte de (Q_{ch})
- RF- 21. Mostrar Reporte de (Q_{cs})
- RF- 22. Mostrar Reporte de (Q_{gs})
- RF- 23. Mostrar Reporte de (Q_{va})
- RF- 24. Mostrar Reporte de (QP_1)
- RF- 25. Mostrar Reporte de (QP_2)
- RF- 26. Mostrar Reporte de (QP_3)
- RF- 27. Mostrar Reporte del Rendimiento de Instalación
- RF- 28. Graficar con respecto a los flujos

2.4.2. Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener.

Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto.

Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. (28)

2.4.2.1. Apariencia o interfaz externa

Los usuarios pueden ser personas que no estén familiarizadas con herramientas informáticas, por lo que el sistema debe contar con una interfaz externa amigable que sea fácil de usar y que brinde una correcta navegación para el usuario. Los colores deben ser claros procurando de esta forma que no se dificulte la visualización de la misma. Se evita el uso excesivo de imágenes que pueden dificultar el rendimiento de la aplicación. La interfaz debe utilizar preferentemente conceptos que sean manejados por los usuarios para que les sea fácil el uso de la herramienta y su aprendizaje.

2.4.2.2. Requerimiento de disponibilidad

El sistema deberá estar disponible las 24 horas del día para todos los usuarios.

2.4.2.3. Requerimiento de integridad

La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes por los usuarios.

2.4.2.4. Requerimiento de rendimiento

Con el desarrollo de la aplicación se pretende manipular el procesamiento de datos con una buena eficiencia, precisión y disponibilidad para que el éxito del producto se vea reflejado en la velocidad de respuesta del servidor ante una solicitud de información. Se realiza el tratamiento de errores para evitar un posible bloqueo de la aplicación.

2.4.2.5. Requerimiento de confiabilidad

La seguridad e integridad de los datos es garantizada por la base de datos embebida Apache Derby, seleccionada para el desarrollo de la aplicación.

2.4.2.6. Requerimiento de software

- ✓ Sistema Operativo Windows 2000 Advanced Server o Superior.
- ✓ Apache Derby como gestor de base de datos relacional.
- ✓ Máquina Virtual 6.0 (JDK).
- ✓ Multiplataforma.
- ✓ Herramienta para reportes: iReport.

2.4.2.7. Requerimiento de hardware

- ✓ Pentium II con 128 MB de RAM y un microprocesador a 300 MHz, 6 Gb de disco duro.

2.4.2.8. Requerimiento de portabilidad

Tanto el producto de software como el servidor de base de datos pueden ser usados bajo cualquier sistema operativo, como por ejemplo Linux, Windows, etc.

2.4.2.9. Requerimiento de diseño e implementación

Para la construcción del sistema se requiere:

- ✓ Lenguaje de programación para la implementación: Java.
- ✓ Como artefactos para el diseño se usan los que propone RUP apoyado en el estándar de notaciones de UML.
- ✓ Como gestor de base de datos Apache Derby.

2.4.2.10. Requerimiento de usabilidad

Los usuarios deben poseer conocimientos básicos correspondientes al manejo de computadoras. Se muestran mensajes de error correspondientes al llenado de los datos para garantizar que el usuario se sienta guiado en todo momento en su navegación por la aplicación. Además debe tener una opción de ayuda

sobre las principales operaciones que se realizan, para lograr un menor tiempo de adaptación.

2.5. Conclusiones del capítulo

- En el diagrama del modelo de dominio se representaron 4 clases conceptuales, cuyos atributos pertenecen al procedimiento analítico establecido en la literatura especializada en el tema. (29)
- Además se obtuvieron 28 requisitos funcionales y 10 no funcionales. Estos constituirán la base para estructurar el diagrama de casos de uso del sistema, que será descrito más adelante.

Capítulo 3

Diseño e Implementación del Sistema

Durante la construcción de una propuesta de solución, se presenta el diagrama de clases por casos de uso y se realiza una descripción detallada para cada uno. Se deben tener en cuenta los principios para el diseño de la aplicación, entre los que se pueden mencionar estándares de interfaz, concepción general de la ayuda y el tratamiento de errores. Además se deben presentar el diagrama de clases de diseño y el diagrama de clases persistentes, este último para generar el modelo físico de datos con el uso de la herramienta Embarcadero, se describe cada tabla de la base de datos con sus respectivos atributos. También son mostrados los diagramas de secuencia. Por otro lado, se ofrece una representación gráfica del modelo de despliegue y el de componente.

En tal sentido el objetivo de este capítulo es:

Realizar el diseño y la implementación de la aplicación que tiene como fin la evaluación automática del proceso de secado en cilindros rotatorios horizontales de la “Empresa Comandante Ernesto Che Guevara”.

Capítulo 3. Diseño e Implementación del Sistema

3.1. Identificación de los actores del sistema a automatizar

Una de las situaciones más difíciles a la hora de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan compleja como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con las personas, máquinas y otros sistemas. Este establecimiento de los requisitos es el punto de partida para la identificación de los principales casos de uso y actores del sistema en cuestión. 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. Por lo tanto, establece un acuerdo entre los clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. Por lo tanto podemos plasmar todos los requerimientos en un modelo de casos de uso del sistema y a su vez identificar los principales actores que interactúan con el mismo. Un actor no es parte del sistema, sino un rol que se juega dentro del sistema, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema.

Utilizando las facilidades que nos brinda el UML, vamos a capturar los requisitos funcionales del sistema y a representarlos mediante un diagrama de casos de uso. Para ello tenemos que definir cuáles serían los actores que van a interactuar con el sistema y los casos de uso que van a representar las funcionalidades.

Tabla 3.1. Actores del sistema

Actor	Justificación
Usuario	Especialista que puede obtener información mediante el cálculo referente al balance de corrientes y pérdidas por transferencia de calor, visualizar reportes dados los resultados y graficar el proceso respecto a cada flujo del secado del mineral laterítico.

	RF asociado: 1-28
--	-------------------

3.2. Paquetes y sus relaciones

Los paquetes son abstracciones que organizan un modelo, el cual se considera un mecanismo de propósito general para agrupar elementos por algún objetivo común, logrando así organizar los elementos en los modelos con el fin de poder comprenderlos más fácilmente. (32)

En el sistema se definieron los siguientes paquetes:

- **Conexión Derby:** Controla el inicio de la aplicación encargándose de las funcionalidades correspondientes a la base de datos, para que sea posible el almacenamiento de los datos calculados por el usuario en el sistema. y la representación gráfica de las diferentes funcionalidades.
- **Balance de Corrientes:** Agrupa el comportamiento relacionado con las corrientes que se evalúan y los volúmenes aplicados en unas de ellas.
- **Pérdidas por Transferencia de Calor:** Agrupa el comportamiento relacionado con las corrientes de transferencias de calor que se calculan.
- **Reportes:** Lista toda la información necesaria para el trabajo del usuario en el sistema.

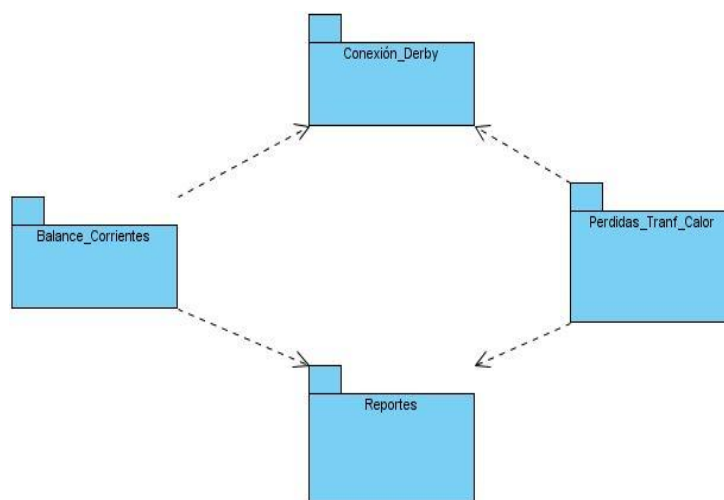


Figura 3.1. Diagrama de Paquetes

3.3. Diagrama de caso de uso del sistema

El diagrama de CUS es una representación de los casos de uso que se generan a raíz de los requisitos funcionales y la relación que tienen estos con los distintos actores que interactúan con el sistema. Los casos de uso pueden presentar varios tipos de relaciones: generalización/especialización¹⁶, incluye¹⁷ y extend¹⁸; los actores solo pueden presentar la relación de generalización/especialización; es importante resaltar que un caso de uso puede ser inicializado por un solo actor.

3.3.1. Diagrama de Casos de Uso del Paquete Conexión

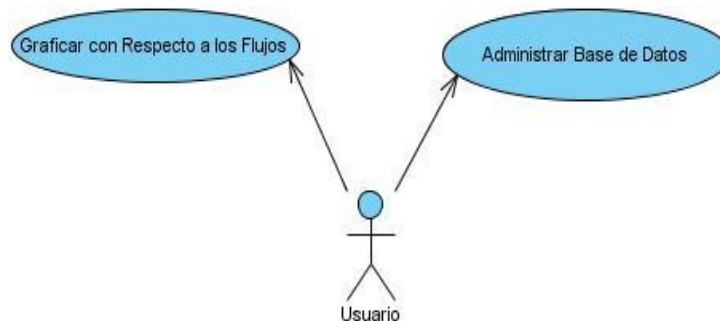


Figura 3.2. Diagrama de Casos de Uso del Paquete Conexión

¹⁶ Se comparten estructuras, propósitos y comportamientos.

¹⁷ El caso de uso de inclusión se inserta explícitamente dentro del comportamiento definido para el caso de uso base.

¹⁸ El caso de uso de extensión tiene un comportamiento opcional u optativo, puede ser usado o no por caso de uso base.

3.3.2. Diagrama de Casos de Uso del Paquete Balance de Corrientes

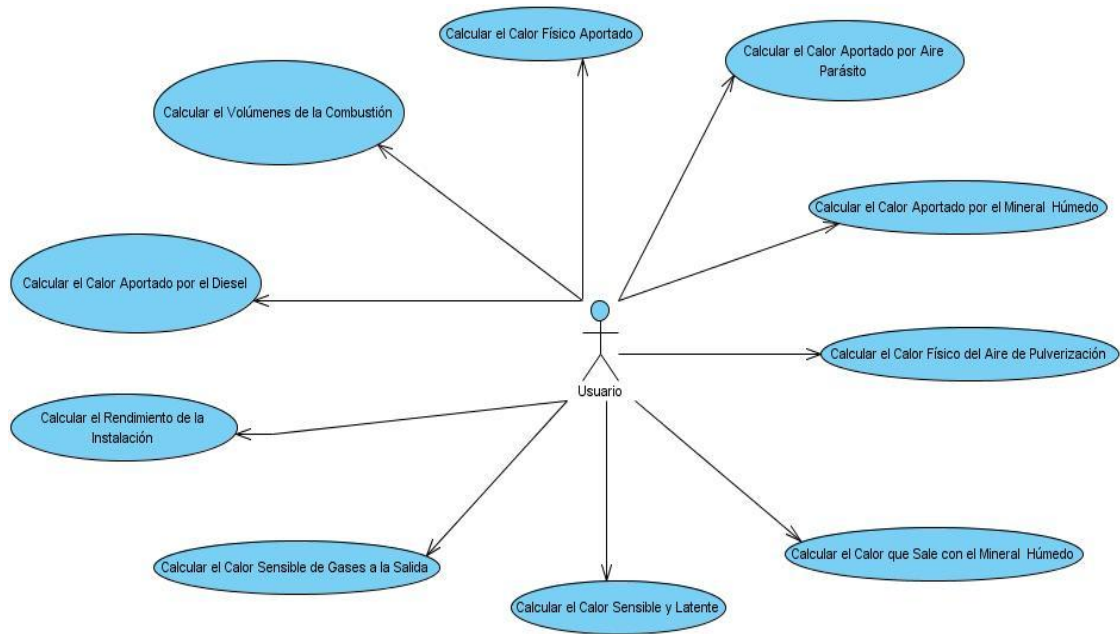


Figura 3.3. Diagrama de Casos de Uso del Paquete Balance de Corrientes

3.3.3. Diagrama de Casos de Uso del Paquete Perdida por Transferencia de Calor

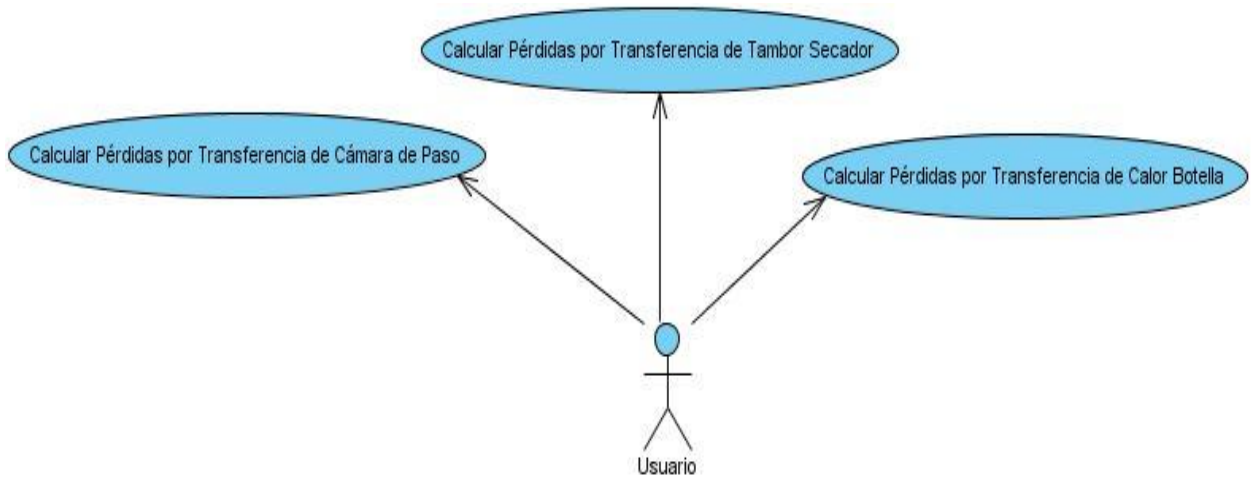


Figura 3.4. Diagrama de Casos de Uso del Paquete Perdida por Transferencia de Calor

3.3.4. Diagrama de Casos de Uso del Paquete Reportes

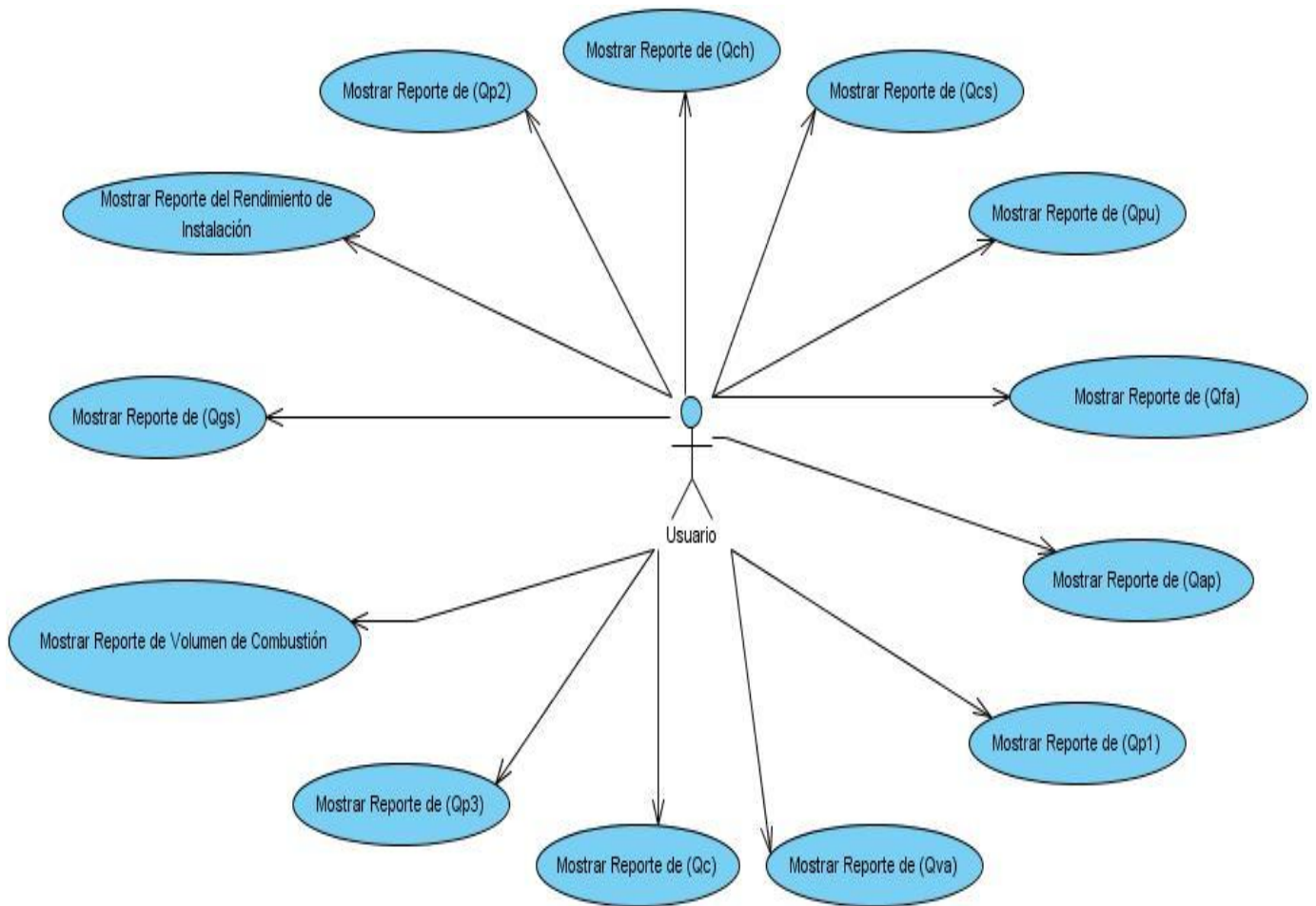


Figura 3.5. Diagrama de Casos de Uso del Paquete Reportes

3.4. Descripciones de los casos de uso del sistema

Tabla 3.2. Descripción del caso de uso “Cálculo del calor aportado por el diesel (Q_c)”.

Caso de uso:	Cálculo del calor aportado por el diesel (Q_c).
Actor:	Usuario (inicia).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los valores para calcular el calor de reacción aportado por el diesel.
Referencia:	RF-2.
Precondiciones:	El usuario debe haber entrado los valores dentro de los

	rangos correspondientes.
Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Las restantes descripciones de caso de uso se encuentran en el [Anexo 1](#).

3.5. Diagrama de clases del diseño

Este describe las clases que componen el sistema con mayor detalle. Incluye los atributos particulares de cada clase y las relaciones existentes entre ellas. Se crean para cumplimentar los requisitos funcionales y no funcionales determinados en el capítulo 2. Se muestra el diagrama de clases del diseño en la figura 3.6.

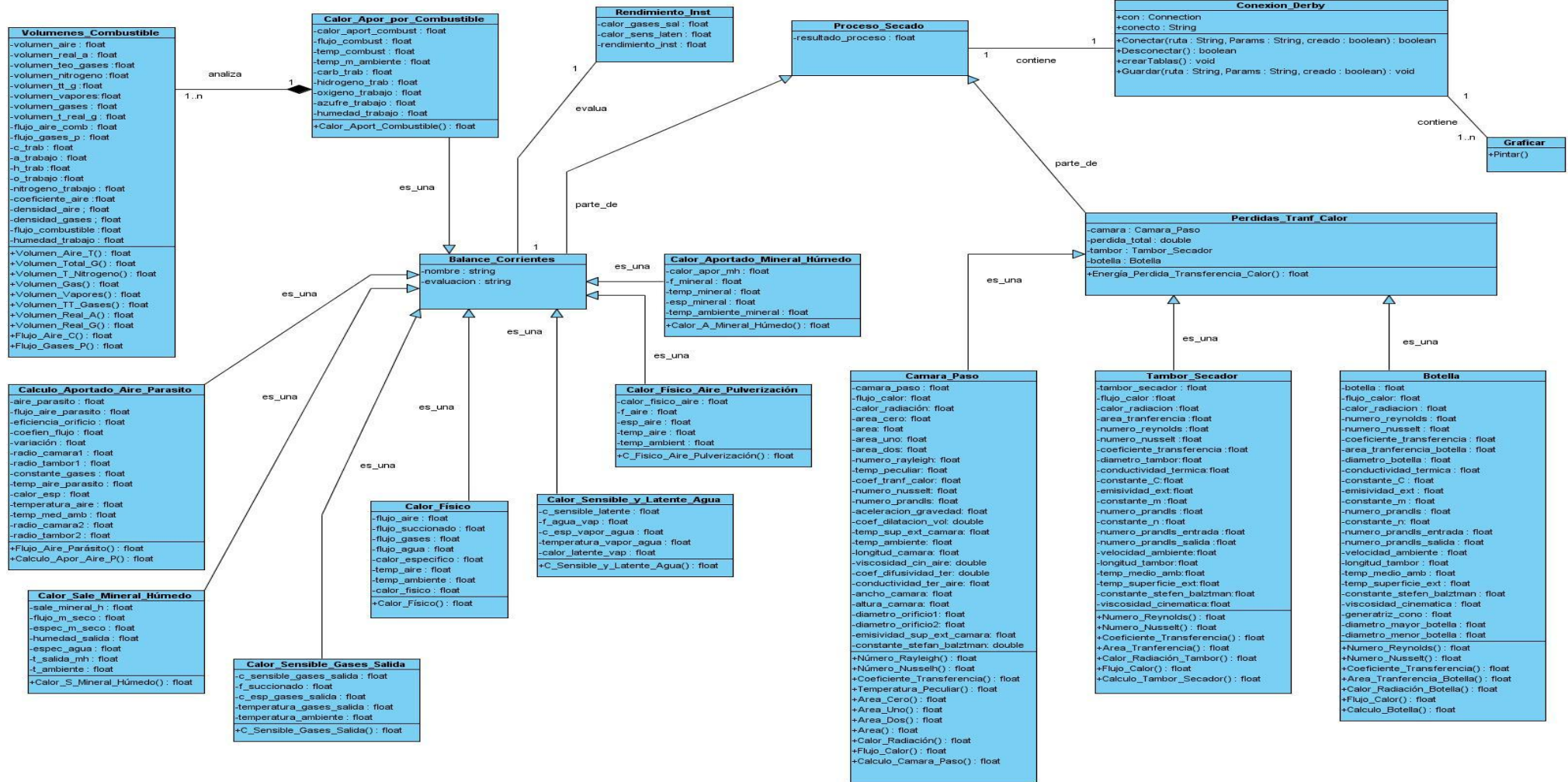


Figura 3.6. Diagrama de Clases del Diseño

3.6. Descripción de las clases del diseño

En este epígrafe se describe cada clase representada en el diagrama anterior, Se define el tipo de clase que es cada una (entidad o controladora), las funciones que realizan, los atributos y métodos que la conforman.

Tabla 3.3. Descripción de la clase Calor_Apor_por_Combustible.

Nombre de la clase	Calor_Apor_por_Combustible	
Tipo de clase:	Controladora	
Resumen: Esta clase permite realizar el cálculo del calor aportado por el combustible.		
Atributos	Tipo	
calor_aport_combust	Float	
flujo_combust	Float	
temp_combust	Float	
temp_m_ambiente	Float	
carb_trab	Float	
hidrogeno_trab	Float	
oxigeno_trabajo	Float	
azufre_trabajo	Float	
humedad_trabajo	Float	
Responsabilidades		
Nombre.	Descripción.	
Calor_Aport_Combustible()	Este método se encarga de calcular el calor aportado por el combustible.	

Las descripciones de las demás clases se encuentran en el [Anexo 2.](#)

3.7. Principios de diseño

A continuación se presentan las interfaces gráficas de usuario diseñadas para la aplicación final. Para su presentación se ha realizado un resumen con las principales funcionalidades del sistema.

3.7.1. Prototipo de Interfaz de usuario

Respondiendo a las solicitudes del cliente se escogen colores tenues para el diseño del software, se emplean palabras familiarizadas con el entorno del cliente y diseña una interfaz sencilla, previendo que el usuario no se pierda en la navegación del software.

3.7.1.1. Guardar Base de Datos

La aplicación dispone de una primera página para guardar la información obtenida mediante el cálculo de las diferentes funcionalidades, con la condición de que, cuando el usuario compile la aplicación necesariamente tiene que guardar o abrir la base de datos (que es otra funcionalidad del sistema), para que los datos no se pierdan y sean almacenados.

El usuario puede determinar la dirección y el nombre de la base datos.

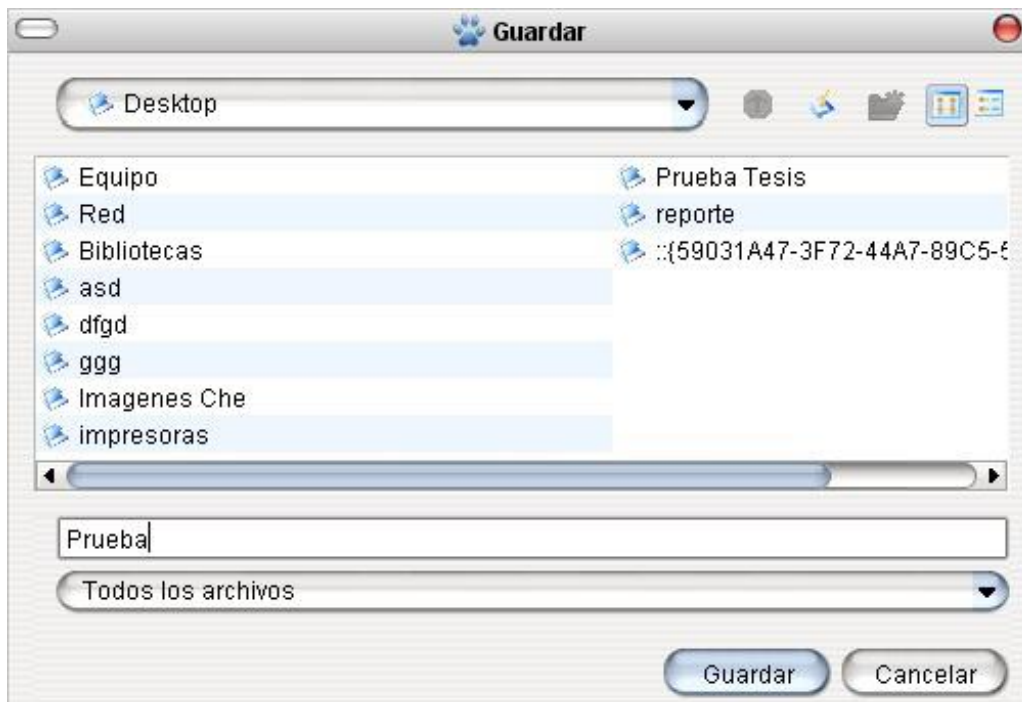


Figura 3.7. Interfaz gráfica para guardar la base de datos

3.7.1.2. Calcular calor aportado por el combustible

Mediante este requisito se permite calcular el calor aportado por el combustible.



Figura 3.8. Interfaz gráfica para calcular el calor aportado por el combustible

3.7.1.3. Calcular pérdidas de transferencia de calor cámara de paso

Mediante este requisito se permite calcular la pérdida de transferencia de calor cámara de paso.

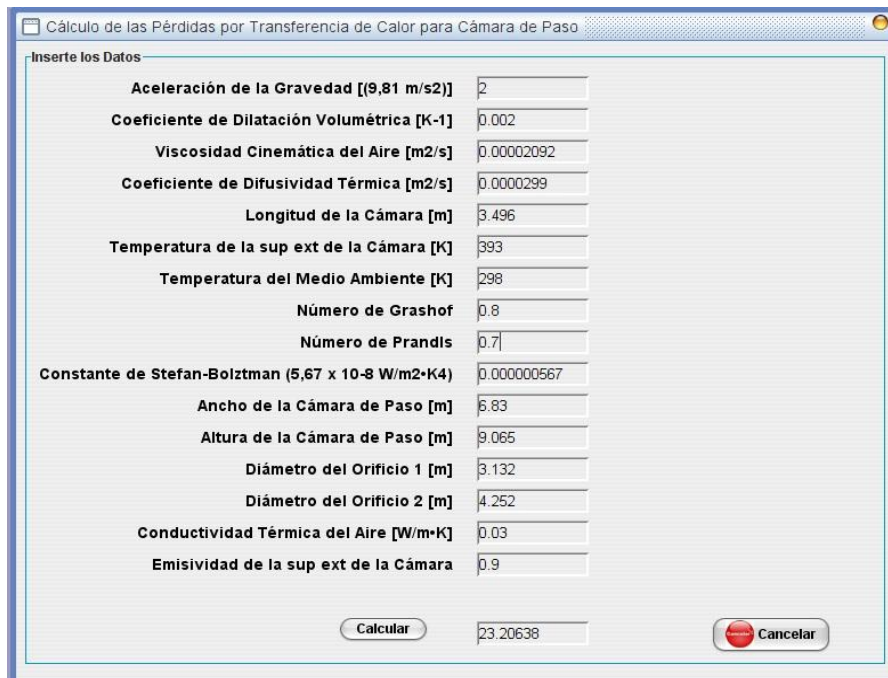


Figura 3.9. Interfaz gráfica para calcular el calor aportado por el combustible

3.7.2. Formato de salida de los reportes



Flujo	Temp	TempAmbiente	CTrabajo	HTrabajo	OTrabajo	Azufre Trab	Humedad	Calor Aportado
2222.0	98.0	35.0	5.0	3.0	4.0	8.0	7.0	9030.0

Figura 3.10. Interfaz gráfica reporte del calor aportado por el combustible

3.7.3. Ayuda

Se contará con un manual de ayuda para el usuario, en el que se explica detalladamente cómo se trabaja con la aplicación.

3.8. Tratamientos de errores

Para lograr que un sistema tenga un alto porcentaje de confiabilidad es de importancia al detectar los errores que ocurran y responder adecuadamente ante ellos con mensajes claros que orienten al usuario para poder corregirlos, de esta manera se evitan incongruencias que generen un resultado inadecuado.

Crear los mecanismos para evitar o manejar adecuadamente los errores que puedan ocurrir en un sistema es una norma elemental a la hora de diseñar e implementar un software apropiadamente.

Para cumplir con esa norma se utilizaron los mecanismos de control de excepciones *try* y *catch* mediante los cuales se detectan los errores que sean cometidos y envía un mensaje de error para que el usuario pueda corregirlos.

Igualmente se impide la inserción de datos ambiguos, como caracteres en campos numéricos netamente o dígitos en espacios de solo texto.

Los ejemplos típicos de error aparecen generalmente cuando se intenta insertar información que ya está creada, o se intentan insertar campos vacíos, en este caso que se muestra a continuación no puede realizarse una inserción de los datos para el cálculo del calor aportado por el combustible en el sistema hasta antes no haberse creado el mismo.

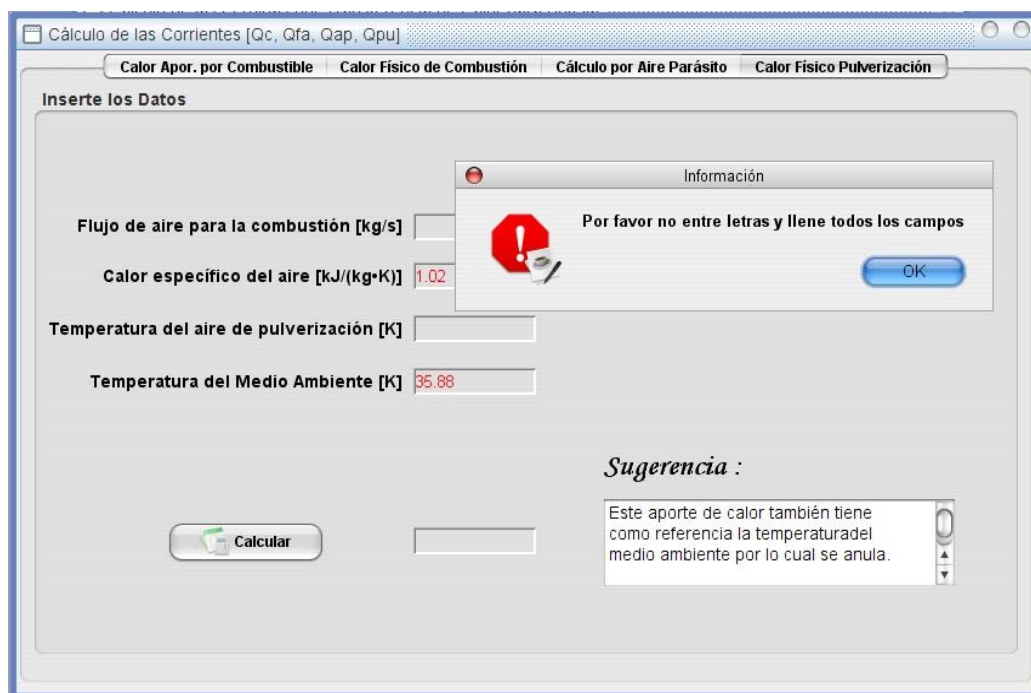


Figura 3.11. Cálculo del Calor Aportado por el Combustible

3.9. Diseño de la base de datos

La base de datos contiene información específica de las entidades que se manejan en el sistema, así como de las relaciones existentes entre ellas, de la calidad de su diseño depende el éxito de la obtención de los datos que residen en la misma.

En los siguientes subepígrafes se evidencian aspectos fundamentales referentes a la base de datos utilizada por el sistema, se muestran las tablas correspondientes a la información persistente en la misma, así como sus atributos, operaciones y la relación existente entre cada tabla.

3.9.1. Modelo lógico de datos

En el diagrama de clases persistentes mostrado en la figura 3.12, se aprecian las relaciones de asociación entre las clases y se incluyen las llaves primarias de cada clase.

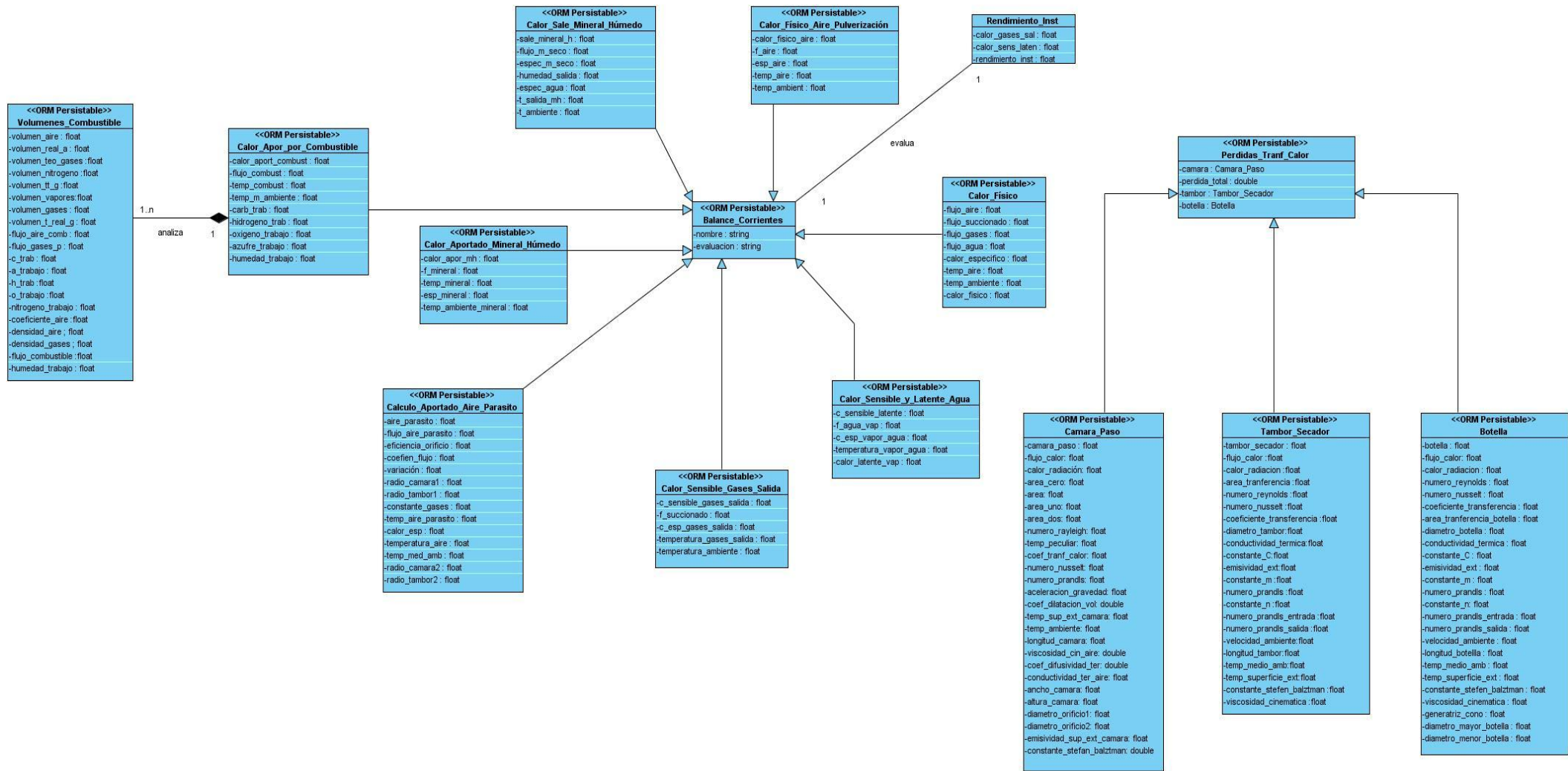


Figura 3.12. Diagrama de Clases Persistentes

3.9.1.1. Descripciones textuales de las clases persistentes

Tabla 3.4. Descripción de la tabla Calor_Apor_por_Combustible.

Nombre de la Clase	Calor_Apor_por_Combustible.
Propósito	Calcula y almacena el calor aportado por el combustible con los datos ingresados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
calor_aport_combust	Resultado del calor aportado por el combustible.
flujo_combust	Valor del flujo de combustible.
temp_combust	Valor de la temperatura de combustible.
temp_m_ambiente	Valor de la temperatura del medio ambiente.
carb_trab	Valor del carbono de trabajo.
hidrogeno_trab	Valor del hidrogeno de trabajo.
oxigeno_trabajo	Valor del oxigeno de trabajo.
azufre_trabajo	Valor del azufre de trabajo.
humedad_trabajo	Valor de la humedad de trabajo.

Las otras descripciones de las tablas aparecen en el [Anexo 3](#).

3.9.2 Modelo físico de datos

Este modelo, mostrado en la figura 3.13, describe la estructura lógica de la información que será almacenada en la base de datos. Es generado a partir del diagrama de clases persistentes mediante la herramienta Embarcadero. Más adelante se detalla la estructura de cada tabla que conforma el modelo.

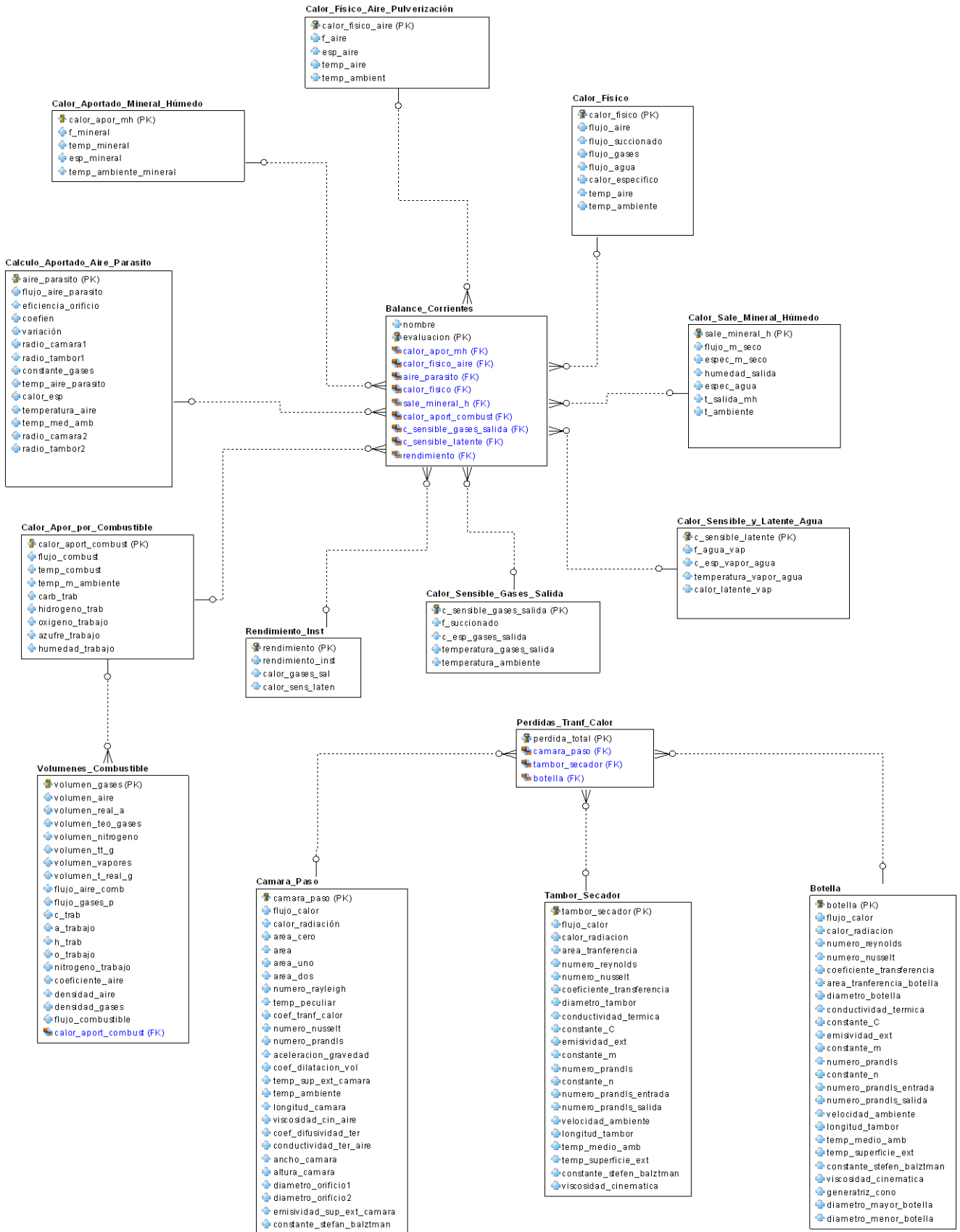


Figura 3.13. Modelo Físico de Datos

3.10. Diagrama de secuencia

Un diagrama de secuencia es una representación que muestra, en determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema.

A todos los sistemas se les trata como una caja negra; los diagramas se centran en los eventos que trascienden las fronteras del sistema y fluyen desde los actores a los sistemas. Estos están dirigidos a los desarrolladores. (28)

En la figura 3.15, se aprecia el diagrama de secuencia del caso de uso “Cálculo del calor aportado por el diesel”, los demás se encuentran en el [Anexo 4](#).

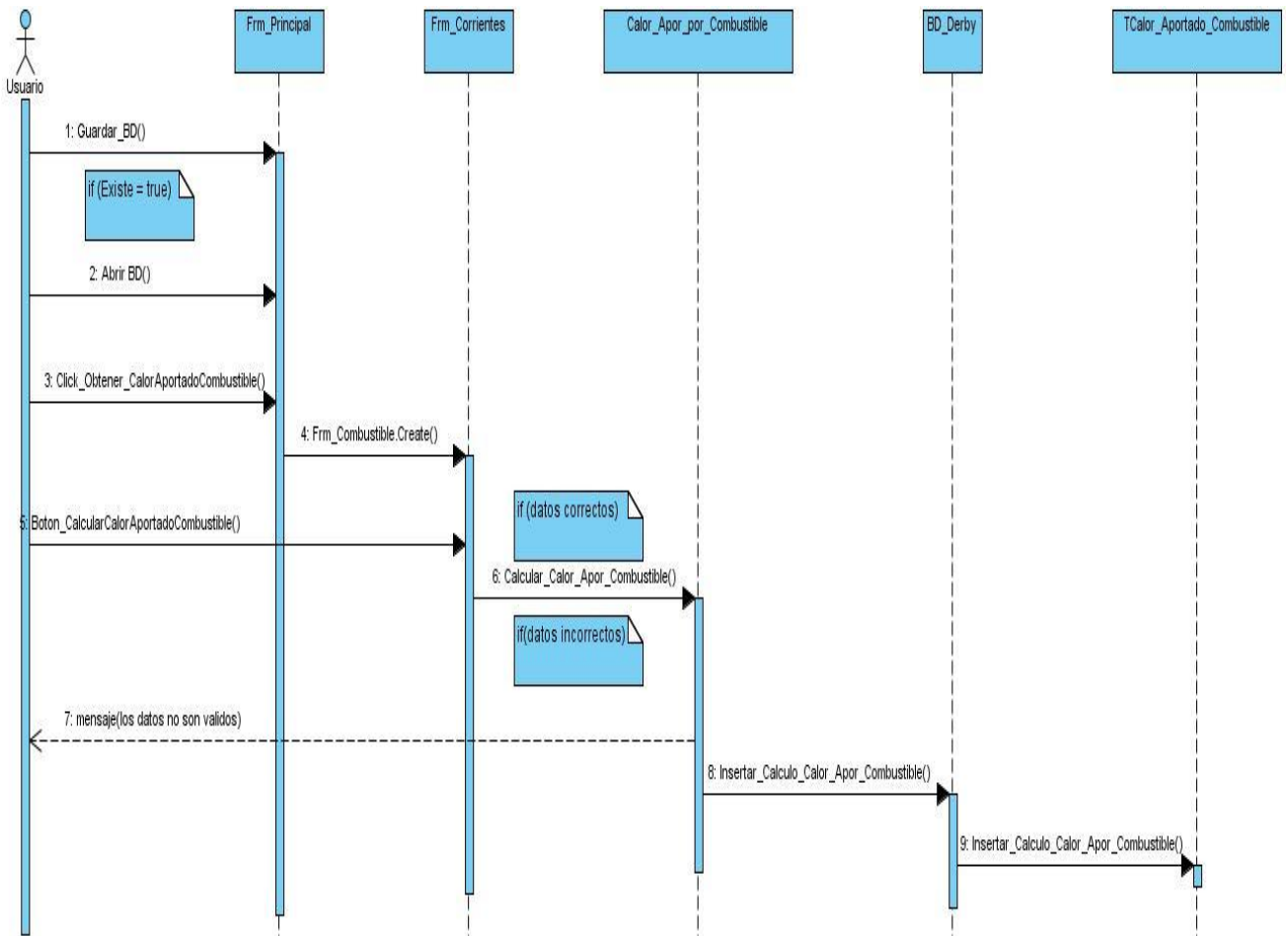


Figura 3.15. Diagrama de secuencia del CU “Cálculo del calor aportado por el diesel”

3.11. Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. (29)

En él se representan una serie de nodos y arcos, donde cada nodo representa un equipo de cómputo (procesador o equipo de hardware similar).

Entre los procesadores o dispositivos hardware a utilizar se presentan, el nodo de cómputo Aplicación el cual será usado por los usuarios que interactúen con el sistema, el mismo depende de la Base de Datos Embebida Apache Derby en donde se encontrará toda la información del sistema y actualizará todas las corrientes entrantes y salientes y las pérdidas de transferencia de calor en el proceso de secado del mineral.

Mediante el diagrama de despliegue se puede apreciar cómo se encuentran relacionados los nodos físicamente.

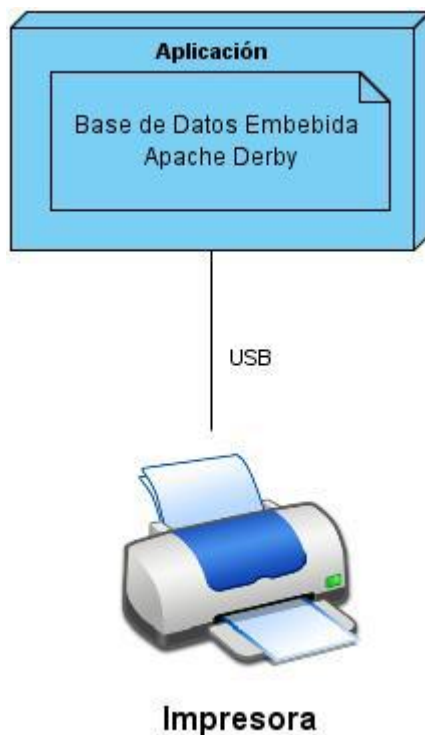


Figura 3.16. Diagrama de Despliegue

3.12. Diagrama de Componente

Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones.

Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean éstos componentes de código fuente, ejecutables, librerías, tablas o documentos.

El diagrama de componentes estará dividido por partes, donde cada uno de ellos describe un apartado del sistema.

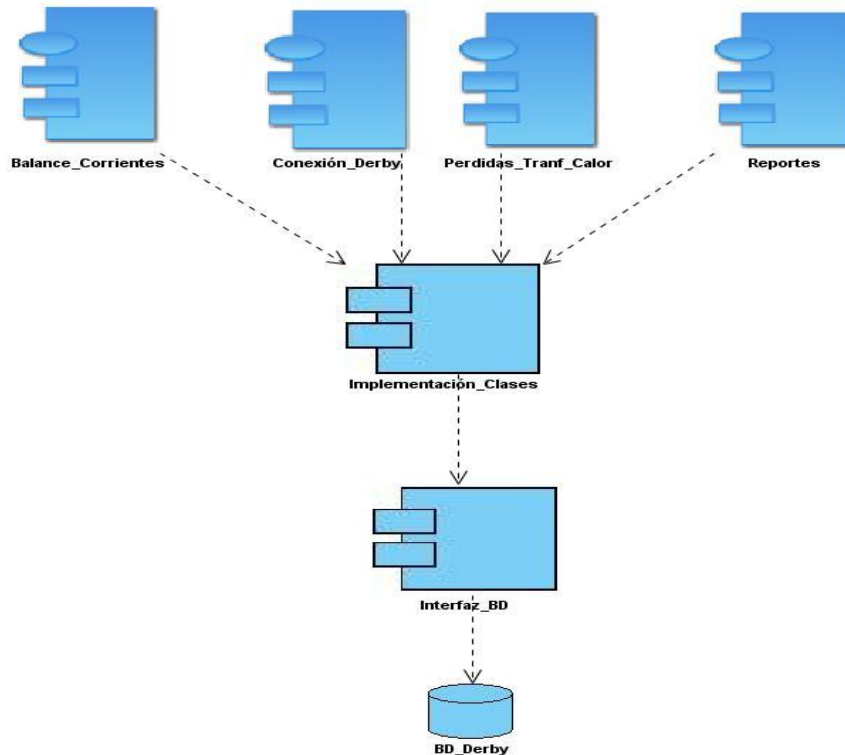


Figura 3.17. Diagrama de Componentes General

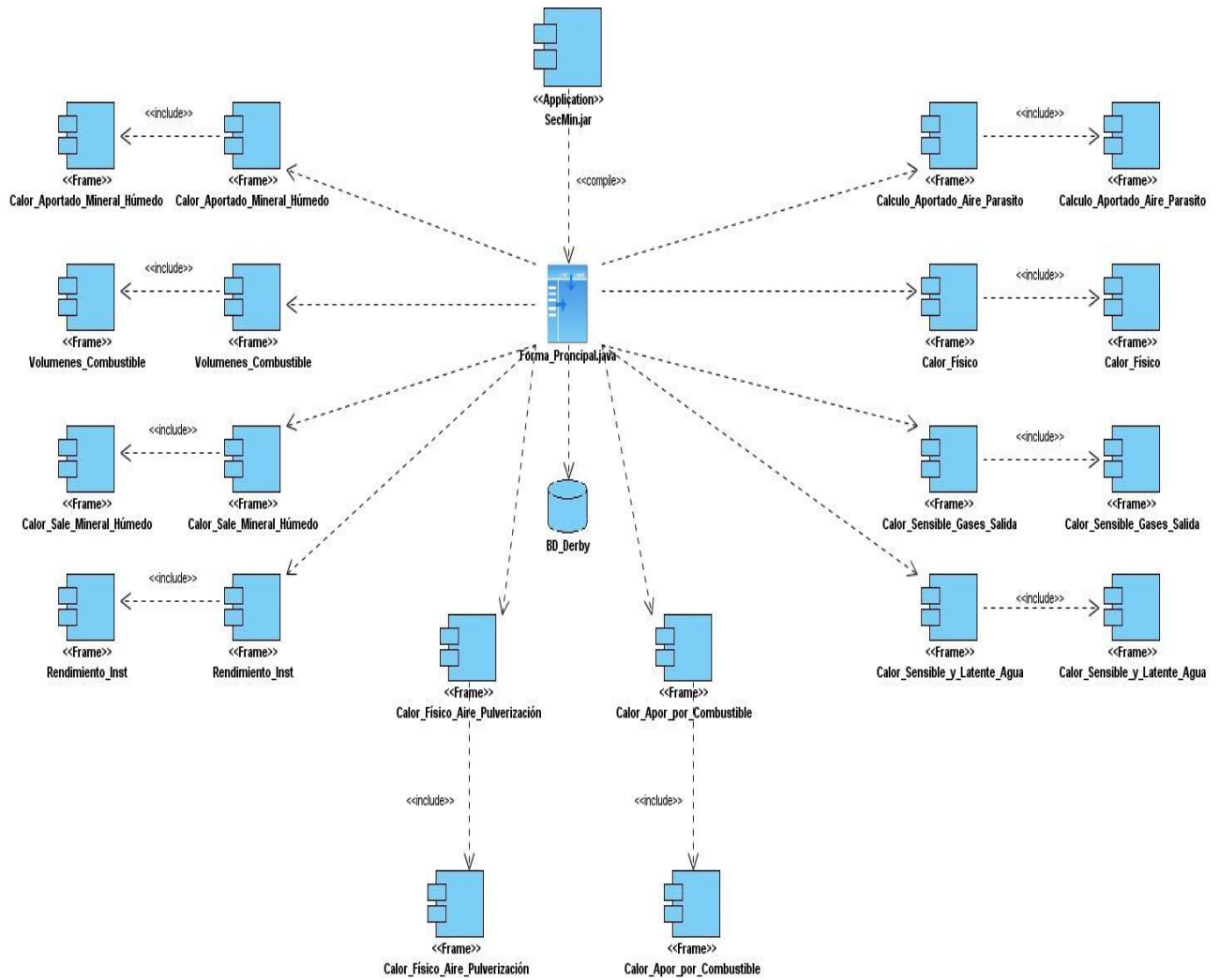


Figura 3.18. Diagrama de Componentes del Paquete de Balance_Corrientes

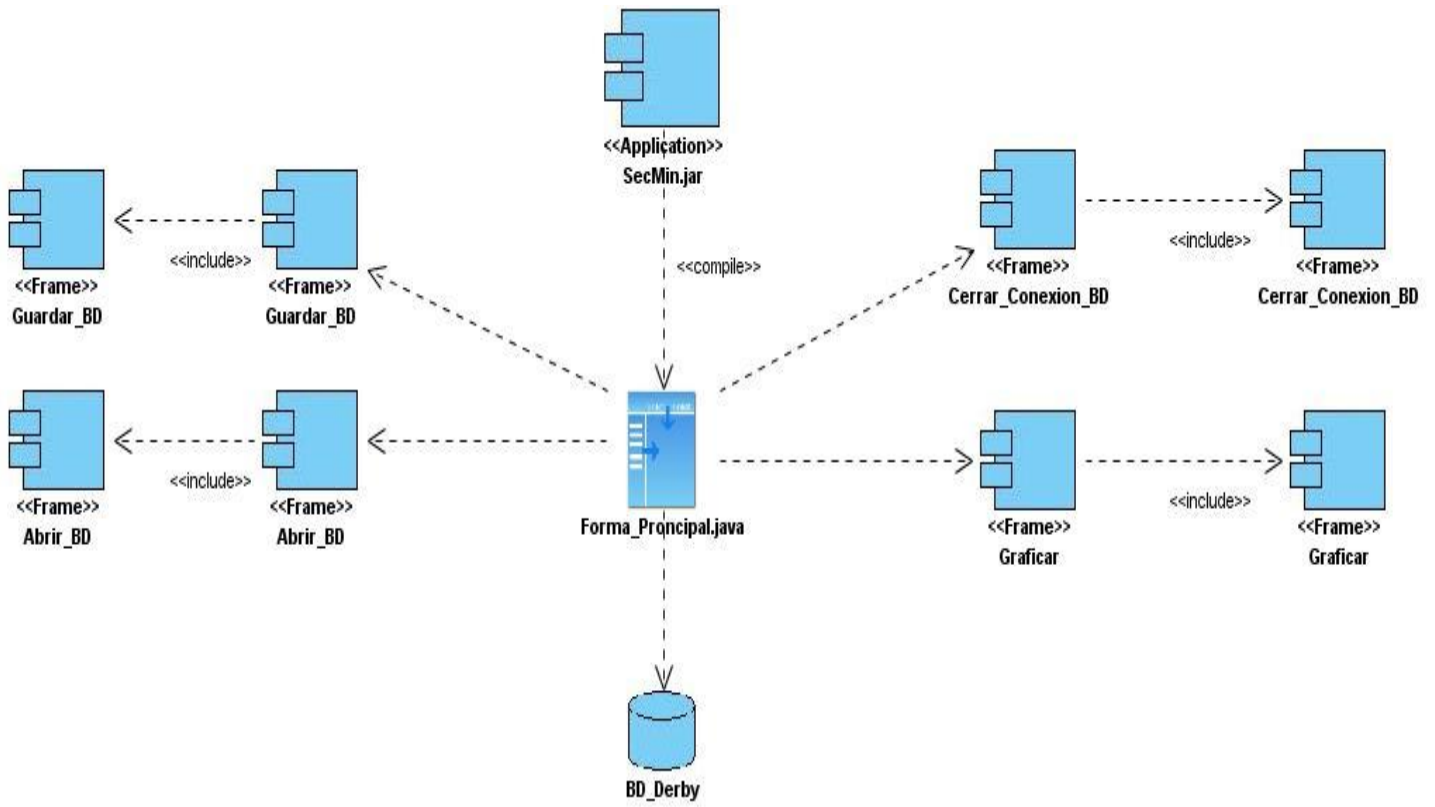


Figura 3.19. Diagrama de Componentes del Paquete Conexión_Derby

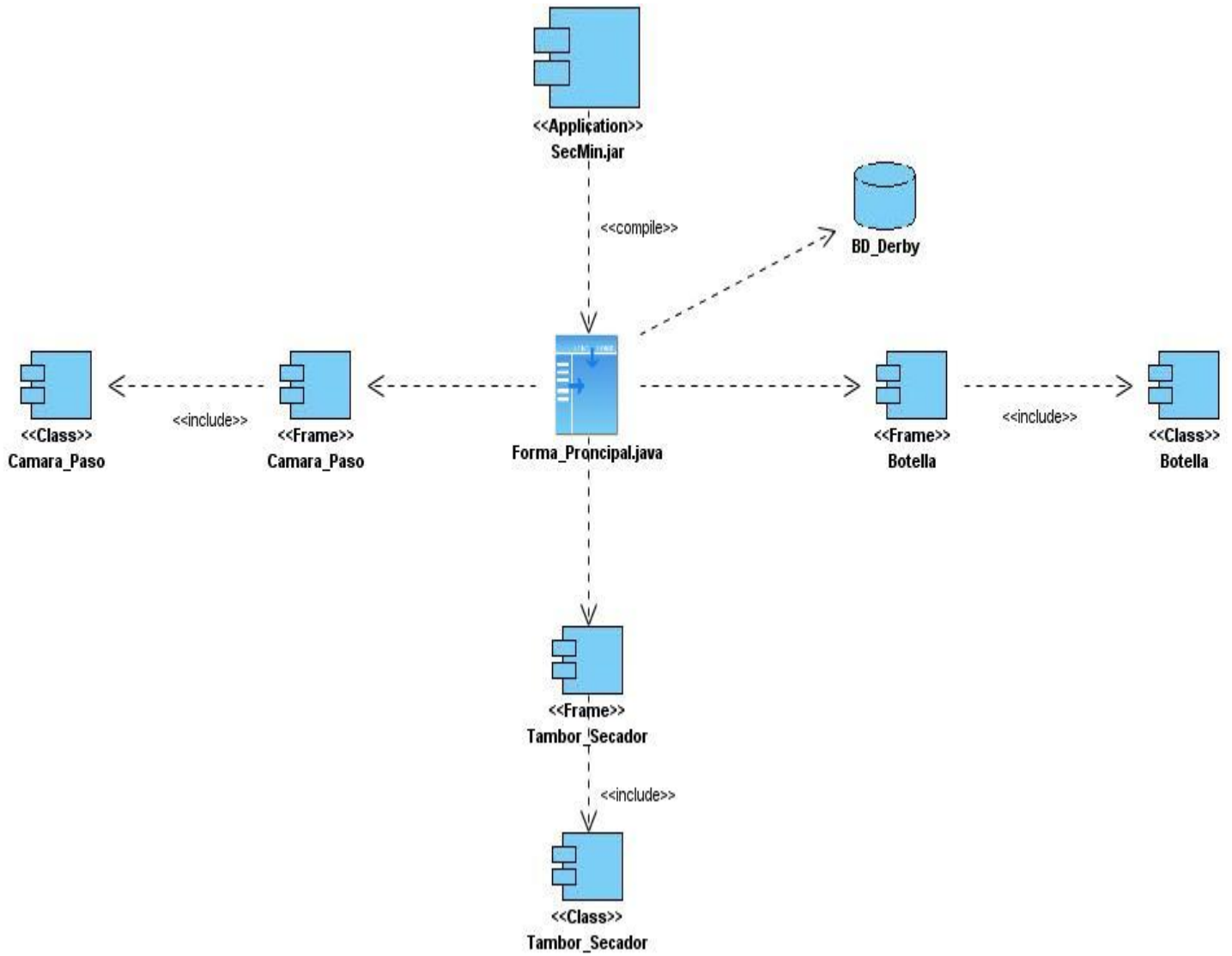


Figura 3.20. Diagrama de Componentes del Paquete Pérdida_Tranf_Calor

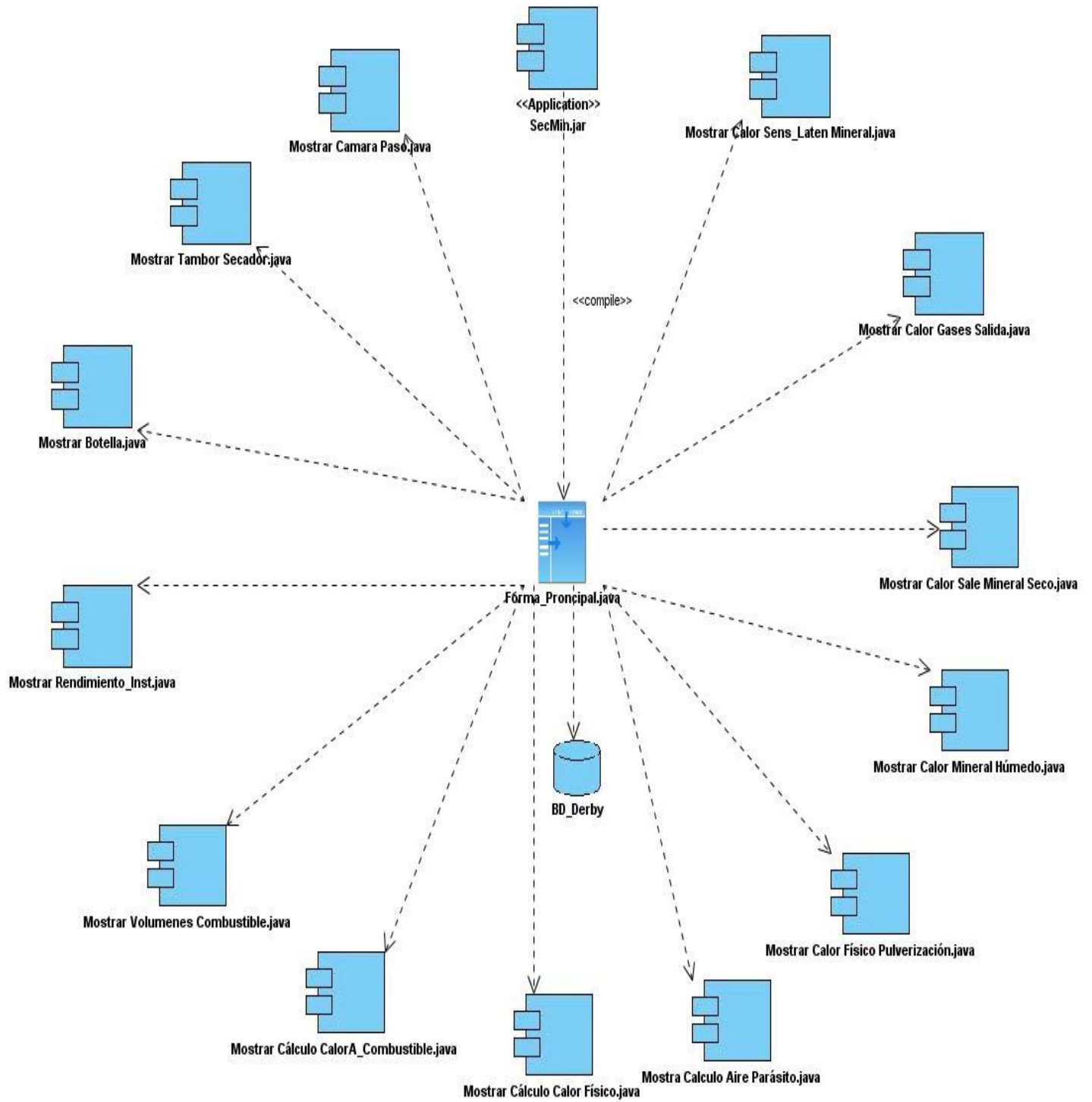


Figura 3.21. Diagrama de Componentes del Paquete Reportes

3.13. Conclusiones del capítulo

Concluida la etapa de diseño e implementación del sistema por medio de diferentes diagramas que reflejan la estructura del mismo, se ha hecho posible organizar el trabajo para la construcción y desarrollo de la aplicación. Se realizó el diseño de la base de datos, quedando establecidos la relación entre las tablas, así como los campos asociados a cada una de ellas.

Capítulo 4

Estudio de Factibilidad

Factibilidad se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados, apoyándose en tres aspectos básicos: operativo, técnico y económico. El éxito de un proyecto está determinado por el grado de factibilidad que se presente en cada uno de los tres aspectos anteriores.

Mediante el análisis de factibilidad se descubre cuáles son los objetivos de la organización, luego determinar si el proyecto es útil para que la entidad logre sus objetivos, por ello el estudio de la factibilidad brinda una información completa relacionada con el costo del producto, tiempo estimado de desarrollo, cantidad de personas que intervienen, entre otros.

Este capítulo tiene como objetivo estimar el esfuerzo (costes) de desarrollo del sistema, utilizando para ello el método COCOMO II.

Capítulo 4. Estudio de Factibilidad

4.1. Método COCOMO II

Este método posee tres modelos denominados **Composición de Aplicación, Diseño Temprano y Post Arquitectura**, cada uno orientado a sectores específicos del mercado de desarrollo de software y a las distintas etapas.

El **Modelo Composición de Aplicación**, es utilizado en los proyectos que se construyen a partir de componentes pre-empaquetados. En este caso, se emplean Puntos objetos para estimar el tamaño del software, lo cual está acorde al nivel de información que generalmente se tiene en la etapa de planificación, y el nivel de precisión requerido en la estimación de proyectos de esta naturaleza.

El **Modelo Diseño Temprano** se emplea en las primeras etapas del desarrollo en la que se evalúan las alternativas de hardware y software de un proyecto. En esta etapa se tiene poca información, lo que concuerda con el uso de Puntos de función, para estimar tamaño y el uso de un número reducido de factores de costo.

El **Modelo de Post Arquitectura** se aplica en la etapa de desarrollo, después que se define la arquitectura del sistema y en la etapa de mantenimiento. Es el más detallado.

El Modelo que se utiliza es el Post Arquitectura, escogido entre los demás, porque brinda valores más exactos y se realiza en una fase avanzada en el desarrollo.

4.2. Pasos a seguir en el método Post Arquitectura

Los pasos a seguir en este método son:

1. Obtener los puntos de función (UFP).
 - a. Identificación de las características.
 - b. Clasificación.

- c. Ponderación aplicando pesos.
- 2. Estimar la cantidad de instrucciones fuente. (SLOC).
 - a. Utilizar tabla de lenguajes.
- 3. Aplicar las fórmulas de Bohem.
 - a. Obtener esfuerzo (PM) y tiempo (TDEV).
 - b. Costo del Proyecto

4.2.1. Desarrollo del método Post Arquitectura

Paso 1. Obtener los puntos de función

Seguidamente se muestran las funcionalidades del sistema, las cuales se agrupan en: Entradas externas, Salidas externas y Ficheros lógicos internos y se clasifican según su nivel de complejidad en: Simple, Media y Compleja.

Tabla 4.1. Entradas Externas.

Entradas externas	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media, Compleja)
Calor aportado por combustible.	1	9	Simple
Volumen Combustible	10	26	Compleja
Calor físico del combustible.	1	8	Simple
Cálculo por aire parásito.	1	13	Simple
Calor físico pulverización.	1	5	Simple
Calor aportado por mineral húmedo.	1	5	Simple
Calor sale mineral húmedo.	1	7	Simple
Calor sensible gases salida.	1	5	Simple

Calor sensible y latente mineral.	1	5	Simple
Cámara de paso.	1	16	Media
Tambor secador	1	16	Media
Botella	1	19	Media
Rendimiento Instalación	1	5	Simple
Cantidad de ficheros			
Simple	Media		Compleja
9	3		1

Tabla 4.2. Salidas externas.

Salidas externas	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (Simple, Media, Compleja)
Mostrar reporte calor aportado combustible.	1	10	Simple
Mostrar reporte volumen combustible	1	10	Simple
Mostrar reporte calor físico del combustible.	1	9	Simple
Mostrar reporte cálculo por aire parásito.	1	14	Simple
Mostrar reporte calor físico pulverización.	1	6	Simple
Mostrar reporte calor aportado por mineral húmedo.	1	6	Simple

Mostrar reporte calor sale mineral húmedo.	1	8	Simple
Mostrar reporte calor sensible gases salida.	1	6	Simple
Mostrar reporte calor sensible y latente mineral.	1	6	Simple
Mostrar reporte cámara de paso.	1	9	Simple
Mostrar reporte tambor secador	1	10	Simple
Mostrar reporte botella	1	11	Simple
Mostrar reporte rendimiento Instalación	1	5	Simple
Cantidad de ficheros			
Simple	Media		Compleja
13	0		0

Tabla 4.3. Ficheros Lógicos Internos.

Ficheros lógicos internos	Cantidad de records	Cantidad de elementos de datos	Clasificación (Simple, Media, Compleja)
Volumenes_Combustible	6+	10	Media
Calculo_Aportado_Aire_Parasito	1	14	Simple
Calor_Apor_por_Combustible	1	10	Simple
Calor_Sale_Mineral_Húmedo	1	8	Simple
Calor_Aportado_Mineral_Húmedo	1	6	Simple

Calor_Sensible_Gases_Salida	1	6	Simple
Calor_Sensible_y_Latente_Agua	1	6	Simple
Balance_Corrientes	1	2	Simple
Calor_Físico_Aire_Pulverización	1	6	Simple
Calor_Físico	1	9	Simple
Rendimiento_Inst	1	5	Simple
Perdidas_Tranf_Calor	1	4	Simple
Camara_Paso	1	17	Simple
Tambor_Secador	1	17	Simple
Botella	1	20	Simple
Cantidad de ficheros			
Simple	Media		Compleja
14	1		0

Tabla 4.4. Puntos de función desajustados.

Elementos	Simple		Media		Compleja		
	No.	XPeso	No.	XPeso	No.	XPeso	
Entradas Externas	9	3	3	4	1	6	45
Salidas Externas	13	3	0	5	0	7	39
Ficheros Lógicos Internos	14	7	1	10	0	15	108
Total	36		4		1		
Puntos de Función Desajustados (UFP) Total							192

R/ El total de puntos de función desajustados (UFP) que se obtuvo es de 192.

Paso 2. Estimar la cantidad de instrucciones fuente. (SLOC)

Para el cálculo de las instrucciones fuentes (SLOC) se utiliza la ecuación:

$$\text{SLOC} = \text{UFP} * \text{ratio. (1)}$$

Donde:

UFP: Total de puntos de función desajustados, (calculados en el paso anterior).

Ratio: constante para las SLOC de cada lenguaje de programación en este caso tiene un valor para Java de 53.

Luego:

SLOC: Líneas de código fuente.

KSLOC: Miles de línea de código.

Sustituyendo los valores UFP =192 y ratio =53 en la ecuación (1) tendríamos:

$$\text{SLOC} = 10\ 176.$$

Obtenido el valor de SLOC calculamos KSLOC mediante la ecuación (2).

$$\text{KSLOC} = \text{SLOC}/1000. (2)$$

$$= 10\ 176/1000$$

$$= 10,176.$$

R/ EL valor obtenido de KSLOC es de 10,176.

Paso 3. Aplicando las fórmulas de Bohem

Una vez realizado el cálculo de la cantidad de instrucciones fuentes se procede a calcular el esfuerzo (PM).

➤ **Cálculo del esfuerzo (PM).**

$$\text{PM}_{\text{NS}} = A \times (\text{Size})^E \times \prod_{i=1}^n EM_i (3)$$

Donde:

$$E = B + 0,01 \times \sum_{j=1}^5 SF_j \quad (4)$$

Se conoce que:

A = 2,94 (constante).

B = 0.91 (constante).

Size: Tamaño estimado KSLOC.

EMi: Multiplicadores de Esfuerzo.

SF_j : Factores de escala.

Los valores de los factores de escala y los 17 multiplicadores de esfuerzo se muestran en la tabla 4.5 y 4.6 respectivamente.

Tabla 4.5. Factores escala.

Factor de escala	Valor	Justificación
PREC (Precedencia)	3.72	El software tiene aspectos novedosos para el desarrollador.
FLEX (Flexibilidad)	2.03	Hubo cierto grado de acuerdo con los requerimientos pre-establecidos y con las interfaces externas pre-existentes.
RESL (Riesgos)	2.83	Se considera muchos de los riesgos críticos y se establecieron hitos para resolverlos.
TEAM(Cohesión del Equipo)	0.00	No se requiere más de un desarrollador de la aplicación.
PMAT (Madurez de las capacidades)	4.68	Se estima un nivel de madurez normal para el desarrollo de la aplicación.

Tabla 4.6. Valores específicos de los multiplicadores de esfuerzo.

No.	Multiplicadores	Valor	Justificación
1	RELY	1.10	El cliente exige un producto que tenga un grado de fiabilidad muy alto.
2	DATA	1.00	El tamaño de la base de datos es normal, debido a que tiene exactamente 23 tablas.
3	CPLX	1.00	El producto tiene una moderada complejidad. La base de datos que se utiliza tiene un volumen mediano de información por lo que se considera de tamaño moderado.
4	RUSE	1.07	En la implementación del sistema existe una alta reusabilidad de códigos.
5	DOCU	1.00	Se considera que la documentación es normal por la cantidad promedio de artefactos que deben ser documentados.
6	TIME	1.11	Existen altas exigencias sobre capacidad de ejecución.
7	STOR	1.05	Se requiere de un almacenamiento de datos seguro y lo más confiable posible, por lo que las exigencias de almacenamiento son alta.
8	PVOL	0.87	La plataforma es poco volátil porque los cambios ocurren en prolongados períodos de tiempo.
9	ACAP	0.85	Se considera que la capacidad del analista es elevada.
10	PCAP	1.00	Se considera la capacidad del programador es normal.
11	PCON	1.29	La volatilidad de la persona es muy baja.

12	APEX	1.10	Existe una experiencia previa limitada en el área de la aplicación.
13	PLEX	0.85	Existía un conocimiento muy alto de la plataforma NetBeans.
14	LTEX	1.00	Existe una experiencia normal con las herramientas utilizadas.
15	TOOL	0.78	Son usadas diferentes herramientas de software por lo tanto es elevado el uso de las mismas.
16	SITE	1.22	No existe desarrollo de la aplicación en localidades distribuidas.
17	SCED	1.00	Las exigencias sobre el calendario son normales como cualquier producto de software entre más rápido sea el software en brindar una respuesta, más satisfecho queda el cliente.

Como resultado obtenemos:

$$\sum_{j=1}^5 SF_j = 13,26. \text{ y } \prod_{i=1}^n EM_i = 1,07.$$

Sustituyendo el valor de $\sum_{j=1}^5 SF_j$ en la ecuación (4) obtendríamos E:

$$E = 0,91 + 0,01 \times 13,26$$

$$= 1,0426.$$

Nos quedaría sustituir E, en la ecuación (3):

$$PM_{NS} = 2,94 \times (10,176)^{1,0426} \times 1,07.$$

$$= 35,3273 \text{ hombres-mes}$$

R/ El esfuerzo que realiza una persona para el desarrollo del software en un mes es de aproximadamente 35,3273 personas.

➤ **Cálculo de tiempo de desarrollo (TDEV).**

$$TDEV_{NS} = C \times (PM_{NS})^F \quad (5)$$

Donde:

$$F = D + 0,2 \times 0,01 \times \sum_{j=1}^5 SF_j \quad (6)$$

$$= D + 0,2 \times (E-B)$$

Se conoce que:

C= 3,67 y D=0,28. (Valores constantes)

Calculamos F y Tiempo de desarrollo.

$$F = 0,30652$$

$$TDEV_{NS} = 3,67 * 35,3273^{0,30652}$$

$$= 10,9366 \text{ meses.}$$

R/ El tiempo de duración del proyecto sería de 10,9366 meses aproximadamente.

➤ **Cantidad de hombres**

Una vez calculado el tiempo de desarrollo y el esfuerzo, se calcula la cantidad de personas (CH) que son necesarias para el desarrollo del software. En la tabla 4.7 se muestran los valores para el cálculo.

$$CH = PM/TDEV \quad (7)$$

$$= 35,3273 / 10,9366$$

$$= 3,2301.$$

Tabla 4.7. Cantidad de hombres.

Siglas	Indicador	Valor o fórmula
CH	Cantidad de hombres por mes	PM/TDEV
PM	Esfuerzo.	35,3273

TEDV	Tiempo de desarrollo	10,9366
------	----------------------	---------

Son necesarias 4 personas para realizar el software en 10 meses y como 1 persona es la que trabaja, se reajustan los cálculos para este valor:

CH * = 1 persona

TEDV = PM / CH*

$$= 35,3273 / 1$$

$$= 35,3273 \text{ meses.}$$

R/ Son necesarias 35,3273 meses para que una persona desarrolle el software.

➤ **Costos del Proyecto.**

El costo lo calculamos por la ecuación siguiente:

$$C = CHM * PM \text{ (8)}$$

Se conoce que:

C: Costo del proyecto

SP: Salario promedio.

CHM: Costo por hombre-mes

Donde:

$$CHM = 1 * SP. \text{ (9)}$$

Calculando obtenemos:

$$CHM = 1 * 365$$

$$= \$ 365,00$$

$$C = 365,00 * 35,3273$$

$$= \$ 12 894.464 \text{ CUP.}$$

R/ El costo real del proyecto es de \$ 12 894.464 CUP.

4.3. Beneficios tangibles e intangibles

Con la implementación de esta aplicación de escritorio, existirá una herramienta de uso habitual que logrará un alto grado de funcionalidad, usabilidad y seguridad; además de ser lo suficientemente flexible como para permitir modificaciones y actualizaciones de la información que en él se procesa. Se mejora la integridad de la información, realizando además reportes de las corrientes entrantes y salientes, reportes para las pérdidas de transferencia de calor, evitando atender contra la pérdida de información.

A través del uso de este sistema se pueden obtener los siguientes beneficios:

- ✓ Fácil acceso a la información actualizada por parte del usuario.
- ✓ Fácil obtención de información.
- ✓ Utilización de un gestor de base de datos muy potente que garantiza velocidad en el procesamiento, seguridad y protección de los datos.
- ✓ Tiene amplia facilidad de uso.
- ✓ Brinda varias ventajas como ahorro de tiempo y recursos.

4.4. Análisis de costos y beneficios

Con la desarrollo de este producto informático se reportan beneficios considerables al constituir un aporte al desarrollo e informatización eficiente y segura de problemas prácticos en áreas de secado de mineral producto a que resuelve y minimiza los riesgos en la gestión de la información.

Mediante este software es posible un óptimo aprovechamiento del tiempo en la búsqueda de información dándole un mejor uso a las tecnologías de la información y las comunicaciones, de manera que los procesos que se desarrollan en la Empresa Ernesto Che Guevara pueden controlarse y gestionarse con mejores resultados.

Por lo cual hay concordancia con los beneficios tangibles e intangibles que fueron especificados anteriormente, se puede determinar que la construcción de este sistema es totalmente factible no solo para la Empresa Ernesto Che

Guevara sino para otras empresas del níquel.

4.5. Conclusiones del capítulo

Se realizó el estudio de factibilidad llevado a cabo mediante el método COCOMO II, poniendo en práctica el Modelo de Post Arquitectura, se analizó los beneficios tangibles e intangibles, así como los costos y beneficios arrojando como resultado \$ 12 894.464 CUP lo que demuestra la conveniencia de la elaboración del sistema.

Conclusiones Generales

Luego de haber realizado un estudio de diferentes sistemas existentes, se concluye que los mismos no cumplen con las demandas y especificaciones que requiere el sistema a utilizar en la Empresa "Comandante Ernesto Che Guevara" porque presentan limitaciones en sus funcionalidades, por lo que se decidió desarrollar y aplicar un Sistema para el Análisis Termoenergético de Secadores Cilíndricos Rotatorios Horizontales "SecMin" para solventar estas deficiencias.

Con respecto a la metodología y a la herramienta de desarrollo se concluye que fueron escogidas atendiendo tanto a la licencia de uso y distribución que poseen como a sus prestaciones, lo que permitió comprender la lógica del negocio y definir las deficiencias existentes en la misma, por lo que se escogió como metodología RUP, como herramienta de modelado Visual Paradigm for UML, como SGBD Apache Derby, como Lenguaje de Programación Java y como IDE NetBeans.

A través del flujo de trabajo Análisis y Diseño se precisó la estructura que debe estar presente en el sistema y se obtuvo el diagrama de clases del diseño el cual permitió sentar las bases para una implementación exitosa donde se vincularon los componentes de la aplicación con la arquitectura definida.

Con la implementación del sistema "SecMin" se obtuvo una aplicación para la Empresa "Comandante Ernesto Che Guevara" capaz de realizar el análisis del procedimiento de secado del mineral que se manejan en la misma, de manera controlada y centralizada.

Se analizaron los beneficios tangibles e intangibles arrojando como resultado \$ 12 894.464 CUP lo que demuestra la conveniencia de la elaboración del sistema.

Recomendaciones

Tomando como punto de partida los resultados obtenidos en la investigación y con vista a lograr mejores resultados se recomienda:

- Continuar con el perfeccionamiento del sistema desarrollado de acuerdo con las necesidades futuras del cliente.
- Proponer el sistema a las empresas que utilizan la técnica del proceso de secado, como una herramienta para su investigación y control estadísticos del comportamiento de la humedad durante el secado del mineral.

Referencias Bibliográficas

- (1). **AMERICATI**. *Ventajas y Desventajas: Comparación de los Lenguajes C, C++ y Java*. [En línea] 2006. [Citado el: 26 de Febrero de 2012.] <http://www.americati.com>.
- (2). **Arquitectura**. *Programación por capas*. [En línea] [Citado el: 15 de Marzo de 2012.] http://es.wikipedia.org/wiki/Programacion_por_capas.
- (3). **AUTORES, COLETIVO DE**. *“Diseño Orientado a Objetos con UML”*. Versión . s.l. : Gripo EIDOS, 2000.
- (4). **Bases de Datos Empotradas (Embedded)**. [En línea] [Citado el: 20 de Enero de 2012.] <http://techerald.com/page/base-de-datos-empotradas-embedded.html>.
- (5). **Bases de datos embebidas**. *Berkeley DB (BDB)*. [En línea] [Citado el: 25 de Abril de 2012.] <http://ktulu.com.ar/blog/2010/09/29/bases-de-datos-embebidas/>.
- (6). **Conoce Firebird en 2 minutos**. [En línea] [Citado el: 25 de Abril de 2012.] http://www.firebirdnews.org/docs/fb2min_es.html.
- (7). **Cuaresma, Sergi Blanco**. *Metodologías de desarrollo*. [En línea] [Citado el: 10 de Noviembre de 2011.] <http://www.marblestation.com/?p=644>.
- (8). **EcuRed**. *Proceso Unificado de Desarrollo*. [En línea] [Citado el: 15 de Noviembre de 2011.] http://www.ecured.cu/index.php/Proceso_Unificado_de_Development#Fases.
- (9). **EmbarcaderoER/Studio**. [En línea] [Citado el: 10 de Abril de 2012.] http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_softwarre&key=1153755965.
- (10). **ER/Studio**. [En línea] [Citado el: 10 de Abril de 2012.] <http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml>.
- (11). **Free Download Manager- Sitio de descargas de software**. *Paradigma visual para UML (Plataforma Java) (Visual Paradigm for UML [Java Platform]) 6.0*. [En línea] [Citado el: 2 de Febrero de 2012.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
- (12). **Hall, C**. *Drying technollogy An International Journal*. New York : s.n., 1983.
- (13). **Incropera, F. y Witt, D. De**. *Fundamentals of Heat and Mass Transfer. Tercera edición*. La Habana : Pueblo y Educación, 2003 Tomo I, II y III. 723p.

- (14). **Intro Ingeniería Software**. *Microsoft Solution Framework 3.0 (MSF)*. [En línea] [Citado el: 20 de Noviembre de 2011.] <http://scruz334.blogspot.es/i2007-11/>.
- (15). **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley : s.n., 2000. 84-7829-036-2.
- (16). **JACOBSON, I., RUMBAUGH, J. y BOOCH, G**. *El Proceso Unificado de Desarrollo de Software*. La Habana : Félix Varela, 2004. 438 p.
- (17). **Joskowicz, Ing. José**. *Reglas y Prácticas en eXtreme Programming*. [En línea] [Citado el: 20 de Enero de 2012.] <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
- (18). **Kasatkin, A**. *Operaciones básicas y aparatos en la tecnología química. Segunda parte./pág 262-366*. s.l. : Pueblo y Educación, 1987.
- (19). **Krisher, O**. *Fundamentos científicos de la técnica del secado*. Moscú : Izdatinlit, 1961.
- (20). **LARMAN, C**. *UML y patrones. Introducción al análisis y diseño orientado a objeto.3 ed*. La Habana : Félix Varela, 2004.507 p.
- (21). **LEGRÁ LOBAINA, A. y RAMÓN SILVA, O**. *La investigación Científica: Conceptos y Reflexiones*.
- (22). **Luis Giraldo, Yuliana Zapata**. *HERRAMIENTAS DE DESARROLLO DE INGENIERIA DE SW PARA LINUX*. [En línea] [Citado el: 2 de Febrero de 2012.] http://hugolopez.phi.com.co/.../file=Giraldo-Zapata-Herramientas%20de%20ISW.pdf_id=17.
- (23). **Lykov, A**. *Fenómenos del transporte en cuerpos capilaroporosos*. Moscu : gostejizdat, 1954.
- (24). **Medina, Yennifer**. *Ingeniería de Software*. [En línea] [Citado el: 15 de Febrero de 2012.] <http://yennifermedina.blogspot.es/>.
- (25). **Mora, E**. *Evaluación de los gases residuales en el secado del mineral laterítico en tambores cilíndricos rotatorios*. s.l. : Trabajo de Diploma. ISMM Moa. Facultad de Metalurgia y Electromecánica, 1999.
- (26). **Mortier, Gustavo du**. *El método Scrum*. [En línea] [Citado el: 15 de Enero de 2012.] <http://www.mastersoft.com.ar/MsWeb/otros>.
- (27). **NetBeans**. [En línea] [Citado el: 15 de Abril de 2012.] <http://es.wikipedia.org/wiki/NetBeans>.

- (28). **Novoa, R.; Boizán, M.** *Secado de Zeolita a escala de laboratorio III*. No.1:24-29, Santiago de Cuba : Revista Tecnología Química, 1995.
- (29). **Pérez, Segundo.** Scribd. *Comparación de los modelos MSF Y Cascada*. [En línea] [Citado el: 10 de Enero de 2012.] <http://www.scribd.com/doc/35014145/Comparacion-Metodo-Msf-y-Metodologia-Cascada>.
- (30). **Roberth G. Figueroa, Camilo J. Solís, Armando A. Cabrera.** *Metodologías tradicionales vs. metodologías ágiles*. [En línea] [Citado el: 20 de Noviembre de 2011.] <http://adonisnet.files.wordpress.com>.
- (31). **Strumillo, C.** *Fundamentos de la teoría y práctica del secado*. Varsovia : W.N.T, 1975.
- (32). **Taringa.** *110 programas gratuitos en un solo link*. [En línea] [Citado el: 15 de Abril de 2012.] <http://www.taringa.net/posts/downloads/5084555/110-programas-gratuitos-en-un-solo-link.html>.
- (33). **Wikipedia.** *Apache Derby*. [En línea] [Citado el: 25 de Febrero de 2012.] <http://es.wikipedia.org/wiki/ApacheDerby>.
- (34). **Worldlingo.** *Proceso unificado racional de IBM*. [En línea] [Citado el: 15 de Noviembre de 2011.] http://www.worldlingo.com/ma/enwiki/es/IBM_Rational_Unified_Process.
- (35). **ZUKOWISKI, J.** *Programación Java2 J2SE 1.4*. La Habana : Félix Varela, 2006. Volumen 1.

*Anexos***Anexo # 1: Descripciones de los Casos de Uso del Sistema****Tabla 1. Descripción del caso de uso “Administrar Base de Datos”**

Caso de uso:	Administrar Base de Datos.
Actor:	Usuario (inicia).
Propósito:	Permitir guardar, abrir o cerrar conexión la base de datos.
Resumen:	El caso de uso inicia cuando el usuario del sistema solicita guardar, abrir o cerrar conexión de la Base de Datos, para ello debe especificar el nombre y ruta de la base de datos. Finaliza cuando el sistema realiza una de las acciones anteriores.
Referencia:	RF-1.
Precondiciones:	Para guardar la base de datos, no puede existir una base de datos con el mismo nombre. Para abrir la base de datos, debe de estar creada la misma.
Poscondiciones:	En caso de guardar, la base de datos ha sido guardada. En caso de abrir, la base de datos ha sido cargada. En caso de cerrar conexión, se ha cerrado la conexión de la base de datos.
Requisitos Especiales	-

Tabla 2. Descripción del caso de uso “Calcular los volúmenes de la combustión”

Caso de uso:	Calcular los volúmenes de la combustión.
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de los diferentes volúmenes de combustión.
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce y calcula los datos de los diferentes volúmenes de combustión.
Referencia:	RF-3.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.

Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 3. Descripción del caso de uso “Calcular el calor físico aportado del aire primario de combustión (Q_{fa})”

Caso de uso:	Calcular el calor físico aportado del aire primario de combustión (Q_{fa}).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (Q_{fa}).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para calcular el calor físico aportado del aire primario de combustión.
Referencia:	RF-4.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.
Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 4. Descripción del caso de uso “Calcular calor aportado por el aire parásito (Q_{ap})”

Caso de uso:	Calcular calor aportado por el aire parásito (Q_{ap}).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (Q_{ap}).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo aportado por el aire parásito.
Referencia:	RF-5.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.

Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 5. Descripción del caso de uso “Calcular el calor físico del aire de pulverización (Q_{pu})”

Caso de uso:	Calcular el calor físico del aire de pulverización (Q_{pu}).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (Q_{pu}).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo del calor físico del aire de pulverización.
Referencia:	RF-6.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.
Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 6. Descripción del caso de uso “Calcular el calor aportado por el mineral húmedo (Q_{ch})”

Caso de uso:	Calcular el calor aportado por el mineral húmedo (Q_{ch}).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (Q_{ch}).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo del calor aportado por el mineral húmedo.
Referencia:	RF-7.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.

Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 7. Descripción del caso de uso “Calcular el calor que sale con el mineral húmedo (Q_{cs})”

Caso de uso:	Calcular el calor que sale con el mineral húmedo (Q_{cs}).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (Q_{cs}).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo del calor que sale con el mineral húmedo.
Referencia:	RF-8.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.
Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 8. Descripción del caso de uso “Calcular el calor sensible de los gases a la salida del secador (Q_{gs})”

Caso de uso:	Calcular el calor sensible de los gases a la salida del secador (Q_{gs}).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (Q_{gs}).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo del calor sensible de los gases a la salida del secador.
Referencia:	RF-9.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.

Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 9. Descripción del caso de uso “Calcular el calor sensible y latente de vaporización del agua contenida en el mineral (Q_{va})”

Caso de uso:	Calcular el calor sensible y latente de vaporización del agua contenida en el mineral (Q_{va}).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (Q_{va}).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo del calor sensible y latente de vaporización del agua contenida en el mineral.
Referencia:	RF-10.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.
Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 10. Descripción del caso de uso “Calcular la pérdida por transferencia de calor por cámara de paso (QP_1)”

Caso de uso:	Calcular las pérdidas por transferencia de calor por cámara de paso (QP_1).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (QP_1).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo de la pérdida por transferencia de calor por cámara de paso.
Referencia:	RF-11.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.

Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 11. Descripción del caso de uso “Calcular la pérdida por transferencia de calor por tambor secador (QP_2)”

Caso de uso:	Calcular la pérdida por transferencia de calor por tambor secador (QP_2).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (QP_2).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo de la pérdida por transferencia de calor por tambor secador.
Referencia:	RF-12.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.
Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 12. Descripción del caso de uso “Calcular la pérdida por transferencia de calor por botella (QP_3)”

Caso de uso:	Calcular la pérdida por transferencia de calor por botella (QP_3).
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo de (QP_3).
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo de la pérdida por transferencia de calor por botella.
Referencia:	RF-13.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.

Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 13. Descripción del caso de uso “Calcular el Rendimiento de la Instalación”

Caso de uso:	Calcular el Rendimiento de la Instalación.
Actor:	Usuario (inicia).
Propósito:	Permitir que el usuario realice el cálculo del rendimiento de la instalación.
Resumen:	El caso de uso inicia cuando el usuario desea hacer uso de la aplicación e introduce los datos para el cálculo del rendimiento de la instalación.
Referencia:	RF-14.
Precondiciones:	El usuario debe haber entrado los valores dentro de los rangos correspondientes.
Poscondiciones:	El valor ha sido calculado y el sistema muestra el resultado y lo guarda en la Base de Datos.
Requisitos Especiales	-

Tabla 14. Descripción del caso de uso “Mostrar Reporte de (Q_c)”

Caso de uso:	Mostrar Reporte de (Q_c).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_c).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_c)”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Combustible</u> . Luego el sistema muestra los datos de (Q_c).
Referencia:	RF-15.
Precondiciones:	-

Poscondiciones:	Se muestra los datos de (Q_c).
Requisitos Especiales	-

Tabla 15. Descripción del caso de uso “Mostrar Reporte de (Q_{fa})”

Caso de uso:	Mostrar Reporte de (Q_{fa}).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_{fa}).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_{fa})”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Calor Fisico</u> . Luego el sistema muestra los datos de (Q_{fa}).
Referencia:	RF-17.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (Q_{fa}).
Requisitos Especiales	-

Tabla 16. Descripción del caso de uso “Mostrar Reporte de Volúmenes de Combustión”

Caso de uso:	Mostrar Reporte de Volúmenes de Combustión.
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de Volumen de Combustión.
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de Volumen de Combustión”, en esta opción aparecen todos los valores calculados por el usuario, además que existen en la tabla <u>Volúmenes</u> . Luego el sistema muestra los datos de (Q_{fa}).
Referencia:	RF-16.
Precondiciones:	-
Poscondiciones:	Se muestra los datos volumen de combustión.
Requisitos Especiales	-

Tabla 17. Descripción del caso de uso “Mostrar Reporte de (Q_{ap})”

Caso de uso:	Mostrar Reporte de (Q_{ap}).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_{ap}).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_{ap})”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Aire Parasito</u> . Luego el sistema muestra los datos de (Q_{ap}).
Referencia:	RF-18.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (Q_{ap}).
Requisitos Especiales	-

Tabla 18. Descripción del caso de uso “Mostrar Reporte de (Q_{pu})”

Caso de uso:	Mostrar Reporte de (Q_{pu}).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_{pu}).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_{pu})”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Aire Pulverizacion</u> . Luego el sistema muestra los datos de (Q_{pu}).
Referencia:	RF-19.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (Q_{pu}).
Requisitos Especiales	-

Tabla 19. Descripción del caso de uso “Mostrar Reporte de (Q_{ch})”

Caso de uso:	Mostrar Reporte de (Q_{ch}).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_{ch}).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_{ch})”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Mineral Humedo</u> . Luego el sistema muestra los datos de (Q_{ch}).
Referencia:	RF-20.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (Q_{ch}).
Requisitos Especiales	-

Tabla 20. Descripción del caso de uso “Mostrar Reporte de (Q_{cs})”

Caso de uso:	Mostrar Reporte de (Q_{cs}).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_{cs}).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_{cs})”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Sale Mineral Humedo</u> . Luego el sistema muestra los datos de (Q_{cs}).
Referencia:	RF-21
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (Q_{cs}).
Requisitos Especiales	-

Tabla 21. Descripción del caso de uso “Mostrar Reporte de (Q_{gs})”

Caso de uso:	Mostrar Reporte de (Q_{gs}).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_{gs}).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_{gs})”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Sensible Gases Salida</u> . Luego el sistema muestra los datos de (Q_{gs}).
Referencia:	RF-22.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (Q_{gs}).
Requisitos Especiales	-

Tabla 22. Descripción del caso de uso “Mostrar Reporte de (Q_{va})”

Caso de uso:	Mostrar Reporte de (Q_{va}).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (Q_{va}).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (Q_{va})”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Sensible Latente Agua</u> . Luego el sistema muestra los datos de (Q_{va}).
Referencia:	RF-23.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (Q_{va}).
Requisitos Especiales	-

Tabla 23. Descripción del caso de uso “Mostrar Reporte de (QP_1)”

Caso de uso:	Mostrar Reporte de (QP_1).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (QP_1).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (QP_1)”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Camara_Paso</u> . Luego el sistema muestra los datos de (QP_1).
Referencia:	RF-24.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (QP_1).
Requisitos Especiales	-

Tabla 24. Descripción del caso de uso “Mostrar Reporte de (QP_2)”

Caso de uso:	Mostrar Reporte de (QP_2).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (QP_2).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (QP_2)”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Tambor_Secador</u> . Luego el sistema muestra los datos de (QP_2).
Referencia:	RF-25.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (QP_2).
Requisitos Especiales	-

Tabla 25. Descripción del caso de uso “Mostrar Reporte de (QP_3)”

Caso de uso:	Mostrar Reporte de (QP_3).
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte de (QP_3).
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte de (QP_3)”, en esta opción aparecen todos los valores insertados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Botella</u> . Luego el sistema muestra los datos de (QP_3).
Referencia:	RF-26.
Precondiciones:	-
Poscondiciones:	Se muestra los datos de (QP_3).
Requisitos Especiales	-

Tabla 26. Descripción del caso de uso “Mostrar Reporte del Rendimiento de Instalación”

Caso de uso:	Mostrar Reporte del Rendimiento de Instalación.
Actor:	Usuario (inicia).
Propósito:	Mostrar Reporte del Rendimiento de Instalación.
Resumen:	El caso de uso comienza cuando el usuario elige la opción “Reporte del Rendimiento de Instalación”, en esta opción aparecen todos los valores entrados por el usuario, además del resultado del cálculo realizado que existen en la tabla <u>Rendimiento Inst.</u> Luego el sistema muestra los datos del rendimiento de la instalación.
Referencia:	RF-27.
Precondiciones:	-
Poscondiciones:	Se muestra los datos del rendimiento de la instalación.
Requisitos Especiales	-

Tabla 27. Descripción del caso de uso “Graficar con respecto a los flujos”

Caso de uso:	Graficar con respecto a los flujos.
Actor(es):	Usuario (inicia).
Propósito:	Graficar con respecto a los flujos.
Resumen:	Este caso de uso inicia cuando el usuario selecciona la opción Graficos. Como respuesta el sistema muestra un formulario con los flujos para graficar.
Referencia:	RF-28.
Precondiciones:	
Poscondiciones:	
Requisitos Especiales:	-

Anexo # 2: Descripciones de las Clases del Diseño

Tabla 1. Descripción de la clase Calculo_Aportado_Aire_Parasito

Nombre de la clase	Calculo_Aportado_Aire_Parasito
Tipo de clase:	Entidad
Resumen: Esta clase permite realizar el cálculo aportado por el aire parasito.	
Atributos	Tipo
aire_parasito	Float
flujo_aire_parasito	Float
eficiencia_orificio	Float
coefien_flujo	Float
variación	Float
radio_camara1	Float
radio_tambor1	Float
constante_gases	Float
temp_aire_parasito	Float
calor_esp	Float
temperatura_aire	Float
temp_med_amb	Float
radio_camara2	Float

radio_tambor2	Float
Responsabilidades	
Nombre.	Descripción.
Flujo_Aire_Parasito()	Este método se encarga de calcular el flujo de aire parasito.
Calculo_Apor_Aire_P()	Este método se encarga del cálculo aportado por el aire parasito.

Tabla 2. Descripción de la clase Calor_Aportado_Mineral_Húmedo

Nombre de la clase	Calor_Aportado_Mineral_Húmedo
Tipo de clase:	Entidad
Resumen: Esta clase permite realizar el cálculo del calor aportado por el mineral húmedo.	
Atributos	Tipo
calor_apor_mh	Float
f_mineral	Float
temp_mineral	Float
esp_mineral	Float
temp_ambiente_mineral	Float
Responsabilidades	
Nombre.	Descripción.
Calor_A_Mineral_Húmedo()	Este método se encarga de calcular el calor aportado por el mineral húmedo.

Tabla 3. Descripción de la clase Calor_Físico

Nombre de la clase	Calor_Físico
Tipo de clase:	Entidad
Resumen: Esta clase permite realizar el cálculo del calor físico aportado del aire primario de combustión.	
Atributos	Tipo
flujo_aire	Float
flujo_succionado	Float
flujo_gases	Float
flujo_agua	Float
calor_especifico	Float
temp_aire	Float

temp_ambiente	Float
calor_fisico	Float
Responsabilidades	
Nombre.	Descripción.
Calor_Físico()	Este método se encarga de calcular el calor físico aportado del aire primario de combustión.

Tabla 4. Descripción de la clase Calor_Físico_Aire_Pulverización

Nombre de la clase	Calor_Físico_Aire_Pulverización	
Tipo de clase:	Entidad	
Resumen: Esta clase permite realizar el cálculo del calor físico del aire de pulverización.		
Atributos	Tipo	
calor_fisico_aire	Float	
f_aire	Float	
esp_aire	Float	
temp_aire	Float	
temp_ambient	Float	
Responsabilidades		
Nombre.	Descripción.	
C_Fisico_Aire_Pulverización ()	Este método se encarga de calcular el calor físico del aire de pulverización.	

Tabla 5. Descripción de la clase Calor_Sale_Mineral_Húmedo

Nombre de la clase	Calor_Sale_Mineral_Húmedo	
Tipo de clase:	Entidad	
Resumen: Esta clase permite realizar el cálculo del calor que sale con el mineral húmedo.		
Atributos	Tipo	
sale_mineral_h	Float	
flujo_m_seco	Float	
espec_m_seco	Float	
humedad_salida	Float	
espec_agua	Float	
t_salida_mh	Float	

t_ambiente	Float
Responsabilidades	
Nombre.	Descripción.
Calor_S_Mineral_Húmedo ()	Este método se encarga de calcular el calor que sale con el mineral húmedo.

Tabla 6. Descripción de la clase Calor_Sensible_Gases_Salida

Nombre de la clase	Calor_Sensible_Gases_Salida	
Tipo de clase:	Entidad	
Resumen: Esta clase permite realizar el cálculo del calor sensible de los gases a la salida del secador.		
Atributos	Tipo	
c_sensible_gases_salida	Float	
f_succionado	Float	
c_esp_gases_salida	Float	
temperatura_gases_salida	Float	
temperatura_ambiente	Float	
Responsabilidades		
Nombre.	Descripción.	
C_Sensible_Gases_Salida ()	Este método se encarga de calcular el calor sensible de los gases a la salida del secador.	

Tabla 7. Descripción de la clase Calor_Sensible_y_Latente_Agua

Nombre de la clase	Calor_Sensible_y_Latente_Agua	
Tipo de clase:	Entidad	
Resumen: Esta clase permite realizar el cálculo del calor sensible y latente de vaporización del agua contenida en el mineral.		
Atributos	Tipo	
c_sensible_latente	Float	
f_agua_vap	Float	
c_esp_vapor_agua	Float	
temperatura_vapor_agua	Float	
calor_latente_vap	Float	
Responsabilidades		

Nombre.	Descripción.
C_Sensible_y_Latente_Agua ()	Este método se encarga de calcular el calor sensible y latente de vaporización del agua contenida en el mineral.

Tabla 8. Descripción de la clase Botella

Nombre de la clase	Botella
Tipo de clase:	Entidad
Resumen: Esta clase permite realizar el cálculo de las pérdidas por transferencia de calor de la botella.	
Atributos	Tipo
botella	Float
flujo_calor	Float
calor_radiacion	Float
numero_reynolds	Float
numero_nusselt	Float
coeficiente_transferencia	Float
area_tranferencia_botella	Float
diametro_botella	Float
conductividad_termica	Float
constante_C	Float
emisividad_ext	Float
constante_m	Float
numero_prandls	Float
constante_n	Float
numero_prandls_entrada	Float
numero_prandls_salida	Float
velocidad_ambiente	Float
longitud_tambor	Float
temp_medio_amb	Float
temp_superficie_ext	Float
constante_stefen_balztman	Float
viscosidad_cinematica	Float
generatriz_cono	Float

diametro_mayor_botella	Float
diametro_menor_botella	Float
Responsabilidades	
Nombre.	Descripción.
Numero_Reynolds()	Este método se encarga de calcular el número de reynolds.
Numero_Nusselt()	Este método se encarga de calcular el número de nusselt.
Coeficiente_Transferencia()	Este método se encarga de calcular el coeficiente de transferencia
Area_Tranferencia_Botella()	Este método se encarga de calcular el área de transferencia de la botella
Calor_Radiación_Botella()	Este método se encarga de calcular el calor de radiación de la botella
Flujo_Calor()	Este método se encarga de calcular el flujo de calor de la botella
Calculo_Botella()	Este método se encarga el cálculo general de la botella

Tabla 9. Descripción de la clase Camara_Paso

Nombre de la clase	Camara_Paso
Tipo de clase:	Entidad
Resumen: Esta clase permite realizar el cálculo de las pérdidas por transferencia de calor de la Camara_Paso.	
Atributos	Tipo
camara_paso	Float
flujo_calor	Float
calor_radiación	Float
area_cero	Float
area	Float
area_uno	Float
area_dos	Float
numero_rayleigh	Float
temp_peculiar	Float
coef_tranf_calor	Float
numero_nusselt	Float

numero_prandls	Float
aceleracion_gravedad	Float
coef_dilatacion_vol	Double
temp_sup_ext_camara	Float
temp_ambiente	Float
longitud_camara	Float
viscosidad_cin_aire	Double
coef_difusividad_ter	Double
conductividad_ter_aire	Float
ancho_camara	Float
altura_camara	Float
diametro_orificio1	Float
diametro_orificio2	Float
emisividad_sup_ext_camara	Float
constante_stefan_balztman	Double
Responsabilidades	
Nombre.	Descripción.
Número_Rayleigh()	Este método se encarga de calcular el número de reynolds.
Número_Nusselh()	Este método se encarga de calcular el número de nusselt.
Coeficiente_Transferencia()	Este método se encarga de calcular el coeficiente de transferencia
Temperatura_Peculiar()	Este método se encarga de calcular la temperatura peculiar de la cámara de paso.
Area_Cero()	Este método se encarga de calcular el área cero de la cámara de paso.
Area_Uno()	Este método se encarga de calcular el área uno de la cámara de paso.
Area_Dos()	Este método se encarga de calcular el área dos de la cámara de paso.
Area()	Este método se encarga de calcular el área de la cámara de paso.
Calor_Radiación()	Este método se encarga de calcular el calor de radiación de la cámara de paso.
Flujo_Calor()	Este método se encarga de calcular el flujo de calor de

	la cámara de paso.
Calculo_Camara_Paso()	Este método se encarga del cálculo general de la cámara de paso.

Tabla 10. Descripción de la clase Tambor_Secador

Nombre de la clase	Tambor_Secador
Tipo de clase:	Entidad
Resumen: Esta clase permite realizar el cálculo de las pérdidas por transferencia de calor de la tambor_secador.	
Atributos	Tipo
tambor_secador	Float
flujo_calor	Float
calor_radiacion	Float
area_tranferencia	Float
numero_reynolds	Float
numero_nusselt	Float
coeficiente_transferencia	Float
diametro_tambor	Float
conductividad_termica	Float
constante_C	Float
emisividad_ext	Float
constante_m	Float
numero_prandls	Float
constante_n	Float
numero_prandls_entrada	Float
numero_prandls_salida	Float
velocidad_ambiente	Float
longitud_tambor	Float
temp_medio_amb	Float
temp_superficie_ext	Float
constante_stefen_balztman	Float
viscosidad_cinematica	Float
Responsabilidades	

Nombre.	Descripción.
Numero_Reynolds()	Este método se encarga de calcular el número de reynolds.
Numero_Nusselt()	Este método se encarga de calcular el número de nusselt.
Coeficiente_Transferencia()	Este método se encarga de calcular el coeficiente de transferencia del tambor secador.
Area_Tranferencia()	Este método se encarga de calcular el área de transferencia del tambor secador.
Calor_Radiación_Tambor()	Este método se encarga de calcular el calor de radiación del tambor secador
Flujo_Calor()	Este método se encarga de calcular el flujo de calor del tambor secador.
Calculo_Tambor_Secador()	Este método se encarga el cálculo general del tambor secador.

Tabla 11. Descripción de la clase Perdidas_Tranf_Calor

Nombre de la clase	Perdidas_Tranf_Calor
Tipo de clase:	Controladora
Resumen: Esta clase permite realizar el cálculo total de las pérdidas por transferencia de calor.	
Atributos	Tipo
camara	Camara_Paso
tambor	Tambor_Secador
botella	Botella
perdida_total	Double
Responsabilidades	
Nombre.	Descripción.
Energía_Perdida_Transferencia_Calor()	Este método se encarga de sumar las pérdidas locales por convección y radiación en todas las secciones (cámara de paso, tambor secador y la botella).

Tabla 12. Descripción de la clase Volúmenes_Combustible

Nombre de la clase	Volúmenes_Combustible
Tipo de clase:	Entidad
Resumen: Esta clase permite realizar el cálculo de los volúmenes de la combustión.	
Atributos	Tipo
volumen_aire	Float
volumen_real_a	Float
volumen_teo_gases	Float
volumen_nitrogeno	Float
volumen_tt_g	Float
volumen_vapores	Float
volumen_gases	Float
volumen_t_real_g	Float
flujo_aire_comb	Float
flujo_gases_p	Float
c_trab	Float
a_trabajo	Float
h_trab	Float
o_trabajo	Float
nitrogeno_trabajo	Float
coeficiente_aire	Float
densidad_aire	Float
densidad_gases	Float
flujo_combustible	Float
Responsabilidades	
Nombre.	Descripción.
Volumen_Aire_T()	Este método se encarga de calcular el volumen de aire teórico.
Volumen_Total_G()	Este método se encarga de calcular el volumen total de gases triatómicos.
Volumen_T_Nitrogeno()	Este método se encarga de calcular el volumen teórico de nitrógeno.
Volumen_Gas()	Este método se encarga de calcular el volumen teórico

	de los gases secos.
Volumen_Vapores()	Este método se encarga de calcular el volumen teórico de los vapores.
Volumen_TT_Gases()	Este método se encarga de calcular el volumen teórico total de los gases.
Volumen_Real_A()	Este método se encarga de calcular el volumen total real de gases.
Volumen_Real_G()	Este método se encarga de calcular el volumen real de aire.
Flujo_Aire_C()	Este método se encarga de calcular el flujo de aire para la combustión.
Flujo_Gases_P()	Este método se encarga de calcular el flujo de gases.

Tabla 13. Descripción de la clase Conexion_Derby

Nombre de la clase	Conexion_Derby	
Tipo de clase:	Entidad	
Resumen: Esta clase permite conectar, guardar, abrir y desconectar la base de datos apache derby.		
Atributos	Tipo	
con	Connection	
conecto	String	
Responsabilidades		
Nombre.	Descripción.	
Conectar(ruta : String, Params : String, creado : boolean)	Este método se encarga de realizar la conexión de la base de datos.	
Desconectar()	Este método se encarga de desconectar la base de datos.	
crearTablas()	Este método se encarga de crear las tablas en la base de datos.	
Guardar(ruta : String, Params : String, creado : boolean)	Este método se encarga guardar la base de datos.	

Tabla 14. Descripción de la clase Balance_Corrientes

Nombre de la clase	Balance_Corrientes
Tipo de clase:	Controladora
Resumen: Esta clase permite llevar el control de todas las corrientes.	
Atributos	Tipo
nombre	String
evaluacion	String

Tabla 15. Descripción de la clase Proceso_Secado

Nombre de la clase	Proceso_Secado
Tipo de clase:	Controladora
Resumen: Esta clase permite llevar los datos de todo el proceso de secado.	
Atributos	Tipo
resultado_proceso	Float

Tabla 16. Descripción de la clase Graficar

Nombre de la clase	Graficar
Tipo de clase:	Entidad
Resumen: Esta clase permite graficar respecto a cada corrientes determinada.	
Responsabilidades	
Nombre.	Descripción.
Pintar()	Este método se encarga de graficar las corrientes en ejes coordenadas.

Tabla 17. Descripción de la clase Rendimiento_Inst

Nombre de la clase	Rendimiento_Inst
Tipo de clase:	Entidad
Resumen: Esta clase permite determinar el rendimiento de la instalación.	
Atributos	Tipo
calor_gases_sal	Float
calor_sens_laten	Float
rendimiento_inst	Float

Responsabilidades	
Nombre.	Descripción.
Rendimiento_Instalacion()	Este método se encarga de calcular el rendimiento de la instalación.

Anexo # 3: Descripciones de las Tablas que Conforman el Modelo Físico de Datos

Tabla 1. Descripción de la tabla Volumenes_Combustible

Nombre de la clase	Volumenes_Combustible
Propósito	Calcula y almacena los volúmenes de combustible con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
volumen_aire	Resultado del volumen de aire teórico.
volumen_real_a	Resultado del volumen real de aire.
volumen_teo_gases	Resultado del volumen teórico de los gases secos.
volumen_nitrogeno	Resultado del volumen teórico de nitrógeno.
volumen_tt_g	Resultado del volumen teórico total de los gases.
volumen_vapores	Resultado del volumen teórico de los vapores de agua
volumen_gases	Resultado del volumen total de gases triatómicos.
volumen_t_real_g	Resultado del volumen total real de gases.
flujo_aire_comb	Resultado del flujo de aire para la combustión.
flujo_gases_p	Resultado del flujo de gases producto de la combustión.
c_trab	Valor del carbono de trabajo.
a_trabajo	Valor del azufre de trabajo.
h_trab	Valor del hidrógeno de trabajo.
o_trabajo	Valor del oxígeno de trabajo.
nitrogeno_trabajo	Valor del nitrógeno de trabajo.
coeficiente_aire	Valor del coeficiente de exceso de aire
densidad_aire	Valor de la densidad del aire

densidad_gases	Valor de la densidad de los gases
----------------	-----------------------------------

Tabla 2. Descripción de la tabla Calculo_Aportado_Aire_Parasito

Nombre de la clase	Calculo_Aportado_Aire_Parasito.
Propósito	Calcula y almacena el cálculo aportado por el aire parásito con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
aire_parasito	Resultado del cálculo aportado por el aire parásito.
flujo_aire_parasito	Valor del flujo del aire parásito.
eficiencia_orificio	Valor de la eficiencia del orificio.
coefien_flujo	Valor del coeficiente de flujo.
variación	Valor de la variación.
radio_camara1	Valor del radio 1 de la cámara de paso.
radio_tambor1	Valor del radio 1 del tambor secador.
constante_gases	Constante particular de los gases.
temp_aire_parasito	Valor de la temperatura del aire parásito.
calor_esp	Valor del calor específico del aire.
temperatura_aire	Valor de la temperatura del aire.
temp_med_amb	Valor de la temperatura de medio ambiente.
radio_camara2	Valor del radio 2 de la cámara de paso.
radio_tambor2	Valor del radio 2 del tambor secador.

Tabla 3. Descripción de la tabla Calor_Aportado_Mineral_Húmedo

Nombre de la clase	Calor_Aportado_Mineral_Húmedo.
Propósito	Calcula y almacena el calor aportado por el mineral húmedo con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.

calor_apor_mh	Resultado del calor aportado por el mineral húmedo.
f_mineral	Valor del flujo de mineral húmedo que entra al sistema.
temp_mineral	Valor de la temperatura de entrada del mineral húmedo.
esp_mineral	Valor del calor específico del mineral húmedo.
temp_ambiente_mineral	Valor de la temperatura ambiente del mineral húmedo.

Tabla 4. Descripción de la tabla Calor_Físico_Aire_Pulverización

Nombre de la clase	Calor_Físico_Aire_Pulverización.
Propósito	Calcula y almacena el calor físico del aire de pulverización con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
calor_fisico_aire	Resultado del calor físico del aire de pulverización.
f_aire	Valor del flujo de aire para la combustión.
esp_aire	Valor del calor específico del aire.
temp_aire	Valor de la temperatura del aire de pulverización.
temp_ambient	Valor de la temperatura ambiente del aire de pulverización.

Tabla 5. Descripción de la tabla Calor_Físico

Nombre de la clase	Calor_Físico.
Propósito	Calcula y almacena el calor físico aportado del aire primario de combustión con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
flujo_aire	Valor del flujo de aire para la dilución.
flujo_succionado	Valor del flujo succionado por el ventilador de tiro inducido.
flujo_gases	Valor del flujo de gases productos de la combustión.

flujo_agua	Valor del flujo de agua evaporada.
calor_especifico	Valor del calor específico del aire.
temp_aire	Valor de la temperatura del aire.
temp_ambiente	Valor de la temperatura ambiente.
calor_fisico	Resultado del calor físico aportado del aire primario de combustión.

Tabla 6. Descripción de la tabla Calor_Sale_Mineral_Húmedo

Nombre de la clase	Calor_Sale_Mineral_Húmedo.
Propósito	Calcula y almacena el calor que sale con el mineral húmedo con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
sale_mineral_h	Resultado del calor que sale con el mineral húmedo.
flujo_m_seco	Valor del flujo de mineral seco que sale del sistema.
espec_m_seco	Valor del calor específico del mineral seco.
humedad_salida	Valor de la humedad de salida del mineral.
espec_agua	Valor del calor específico del agua.
t_salida_mh	Valor de la temperatura de salida del mineral seco.
t_ambiente	Valor de la temperatura ambiente.

Tabla 7. Descripción de la tabla Calor_Sensible_y_Latente_Agua

Nombre de la clase	Calor_Sensible_y_Latente_Agua.
Propósito	Calcula y almacena el calor sensible y latente de vaporización del agua contenida en el mineral con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
c_sensible_latente	Resultado del calor sensible y latente de vaporización del agua contenida en el mineral.

f_agua_vap	Valor del flujo de agua evaporada.
c_esp_vapor_agua	Valor del calor específico del vapor de agua.
temperatura_vapor_agua	Valor de la temperatura del vapor de agua.
calor_latente_vap	Valor del calor latente de vaporización.

Tabla 8. Descripción de la tabla Calor_Sensible_Gases_Salida

Nombre de la clase	Calor_Sensible_Gases_Salida.
Propósito	Calcula y almacena el calor sensible de los gases a la salida del secador con los datos entrados por los usuarios que hacen uso de la aplicación.
Atributos	Descripción de los atributos.
c_sensible_gases_salida	Resultado del calor sensible de los gases a la salida del secador.
f_succionado	Valor del flujo succionado por el ventilador de tiro inducido.
c_esp_gases_salida	Valor del calor específico de los gases a la salida.
temperatura_gases_salida	Valor de la temperatura de los gases a la salida.
temperatura_ambiente	Valor de la temperatura ambiente.

Tabla 9. Descripción de la tabla Balance_Corrientes

Nombre de la clase	Balance_Corrientes.
Propósito	Esta clase permite llevar el control de todas las corrientes de las que hereda.
Atributos	Descripción de los atributos.
nombre	Variable que determina del nombre de cada corriente.
evaluacion	Variable que determina la evaluación de cada corriente calculada.

Tabla 10. Descripción de la tabla Rendimiento_Inst

Nombre de la clase	Rendimiento_Inst.
Propósito	Calcula y almacena el rendimiento de la instalación.

Atributos	Descripción de los atributos.
calor_gases_sal	Resultado del calor sensible de los gases a la salida del secador.
calor_sens_laten	Resultado del calor sensible y latente del agua.
rendimiento_inst	Resultado del rendimiento de la instalación.

Tabla 11. Descripción de la tabla Perdidas_Tranf_Calor

Nombre de la clase	Perdidas_Tranf_Calor.
Propósito	Esta clase permite llevar el control de todas las pérdidas de transferencia de calor.
Atributos	Descripción de los atributos.
camara	Valor que determina el resultado de la pérdida de transferencia de calor cámara de paso.
perdida_total	Resultado de la suma del cálculo que casa perdida de transferencia de calor realizada.
tambor	Valor que determina el resultado de la pérdida de transferencia de calor tambor secador.
botella	Valor que determina el resultado de la pérdida de transferencia de calor botella.

Tabla 12. Descripción de la tabla Camara_Paso

Nombre de la clase	Camara_Paso
Propósito	Esta clase permite realizar el cálculo de la pérdida por transferencia de calor de la cámara paso.
Atributos	Descripción de los atributos.
camara_paso	Resultado de la pérdida de transferencia de calor cámara de paso.
flujo_calor	Valor del flujo de calor por convección.
calor_radiación	Valor del flujo de calor por radiación.
area_cero	Valor del area total de la cámara de paso.
area	Valor del área de transferencia de calor en la cámara de paso.
area_uno	Valor del área del orificio del diámetro 1.
area_dos	Valor del área del orificio del diámetro 2.
numero_rayleigh	Valor del número de Rayleigh.

temp_peculiar	Valor de la temperatura pelicular.
coef_tranf_calor	Valor del coeficiente de transferencia de calor por convección.
numero_nusselt	Valor del número de Nusselt.
numero_prandls	Valor del número de Prandls.
aceleracion_gravedad	Valor de la aceleración de la gravedad.
coef_dilatacion_vol	Valor del coeficiente de dilatación volumétrica.
temp_sup_ext_camara	Valor de la temperatura de la superficie exterior de la cámara.
temp_ambiente	Valor de la temperatura del medio ambiente.
longitud_camara	Valor de la longitud de la cámara de paso.
viscosidad_cin_aire	Valor de la viscosidad cinemática del aire.
coef_difusividad_ter	Valor del coeficiente de difusividad térmica.
conductividad_ter_aire	Valor de la conductividad térmica del aire.
ancho_camara	Valor del ancho de la cámara de paso.
altura_camara	Valor de la altura de la cámara de paso.
diametro_orificio1	Valor del diámetro del orificio 1.
diametro_orificio2	Valor del diámetro del orificio 2.
emisividad_sup_ext_camara	Valor de la emisividad de la superficie exterior de la cámara.
constante_stefan_balztman	Valor de la constante de Stefan-Boltzman.

Tabla 13. Descripción de la tabla Tambor_Secador

Nombre de la clase	Tambor_Secador
Propósito	Esta clase permite realizar el cálculo de la pérdida por transferencia de calor del tambor secador.
Atributos	Descripción de los atributos.
tambor_secador	Resultado de la pérdida de transferencia de calor tambor secador.
flujo_calor	Valor del flujo de calor por convección.
calor_radiacion	Valor del flujo de calor por radiación.
area_tranferencia	Valor del área de transferencia de calor del tambor secador.
numero_reynolds	Valor del número de Reynolds.
numero_nusselt	Valor del número de Nusselt.

coeficiente_transferencia	Valor del coeficiente de transferencia de calor.
diametro_tambor	Valor del diámetro del tambor secador.
conductividad_termica	Valor de la conductividad térmica del aire.
constante_C	Valor de la constante C.
emisividad_ext	Valor de la emisividad de la superficie exterior del tambor.
constante_m	Valor de la constante m.
numero_prandls	Valor del número de Prandls.
constante_n	Valor de la constante n.
numero_prandls_entrada	Valor de entrada del número de Prandls.
numero_prandls_salida	Valor de salida del número de Prandls.
velocidad_ambiente	Valor de la velocidad del aire ambiente.
longitud_tambor	Valor de la longitud del tambor secador.
temp_medio_amb	Valor de la temperatura del medio ambiente.
temp_superficie_ext	Valor de la temperatura de la superficie exterior del tambor.
constante_stefen_balztman	Valor de la constante de Stefan-Bolztman.
viscosidad_cinematica	Valor de la viscosidad cinemática del aire.

Tabla 14. Descripción de la tabla Botella

Nombre de la clase	Botella
Propósito	Esta clase permite realizar el cálculo de la pérdida por transferencia de calor de la botella.
Atributos	Descripción de los atributos.
botella	Resultado de la pérdida de transferencia de calor botella.
flujo_calor	Valor del flujo de calor por convección.
calor_radiacion	Valor del flujo de calor por radiación.
numero_reynolds	Valor del número de Reynolds.
numero_nusselt	Valor del número de Nusselt.
coeficiente_transferencia	Valor del coeficiente de transferencia de calor.
area_tranferencia_botella	Valor del área de transferencia de calor de la botella.
diametro_botella	Valor del diámetro de la botella.
conductividad_termica	Valor de la conductividad térmica del aire.

constante_C	Valor de la constante C.
emisividad_ext	Valor de la emisividad de la superficie exterior de la botella.
constante_m	Valor de la constante m.
numero_prandls	Valor del número de Prandls.
constante_n	Valor de la constante n.
numero_prandls_entrada	Valor de entrada del número de Prandls.
numero_prandls_salida	Valor de salida del número de Prandls.
velocidad_ambiente	Valor de la velocidad del aire ambiente.
longitud_botella	Valor de la longitud de la botella.
temp_medio_amb	Valor de la temperatura del medio ambiente.
temp_superficie_ext	Valor de la temperatura de la superficie exterior de la botella.
constante_stefen_balztman	Valor de la constante de Stefan-Bolztman.
viscosidad_cinematica	Valor de la viscosidad cinemática del aire.
generatriz_cono	Valor de la generatriz del cono.
diametro_mayor_botella	Valor del diámetro mayor de la botella
diametro_menor_botella	Valor del diámetro menor de la botella

Anexo # 4: Diagramas de Secuencia de los Casos de Uso del Sistema

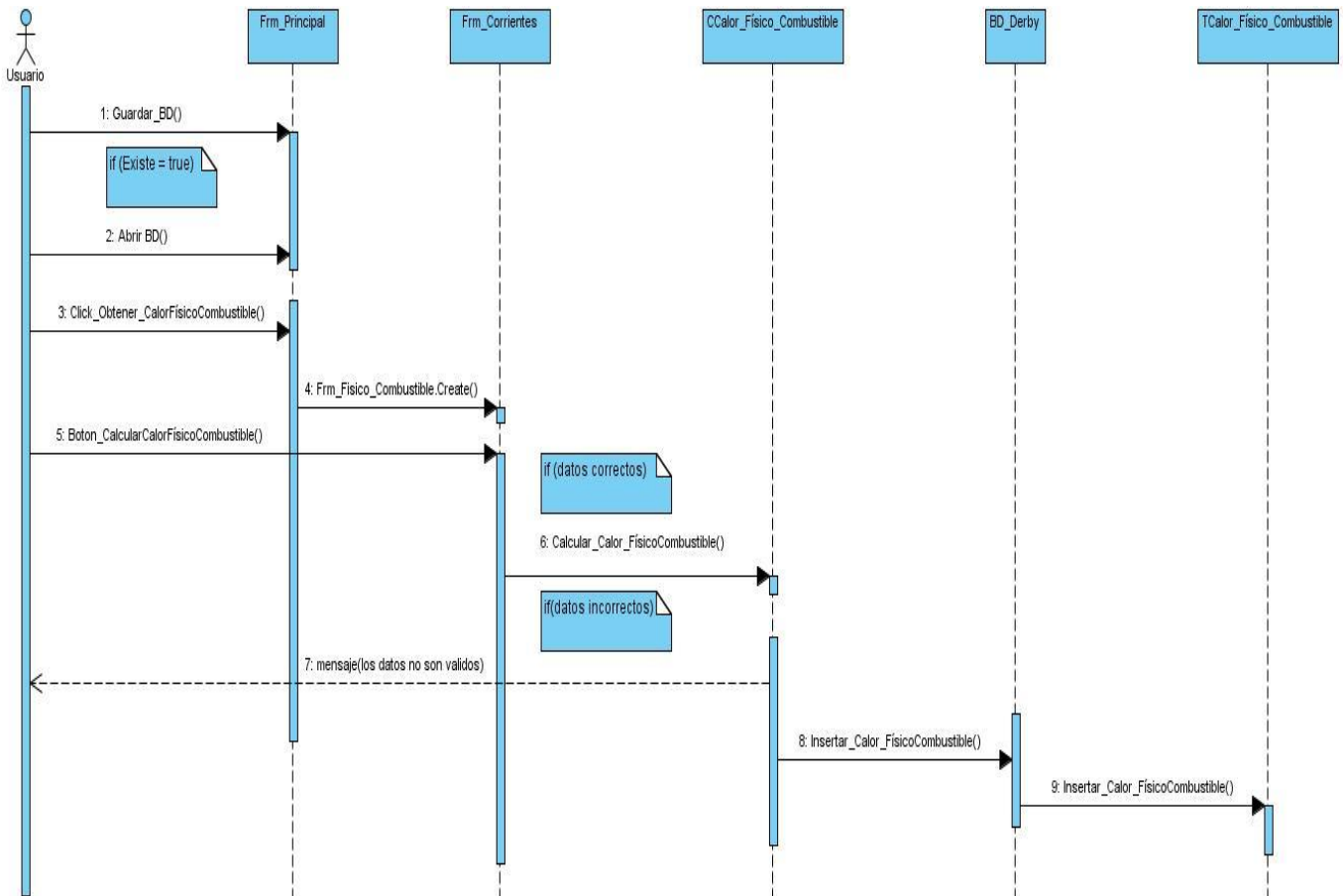


Figura 1. Diagrama de secuencia del CU “Calcular Calor Físico de Combustible”

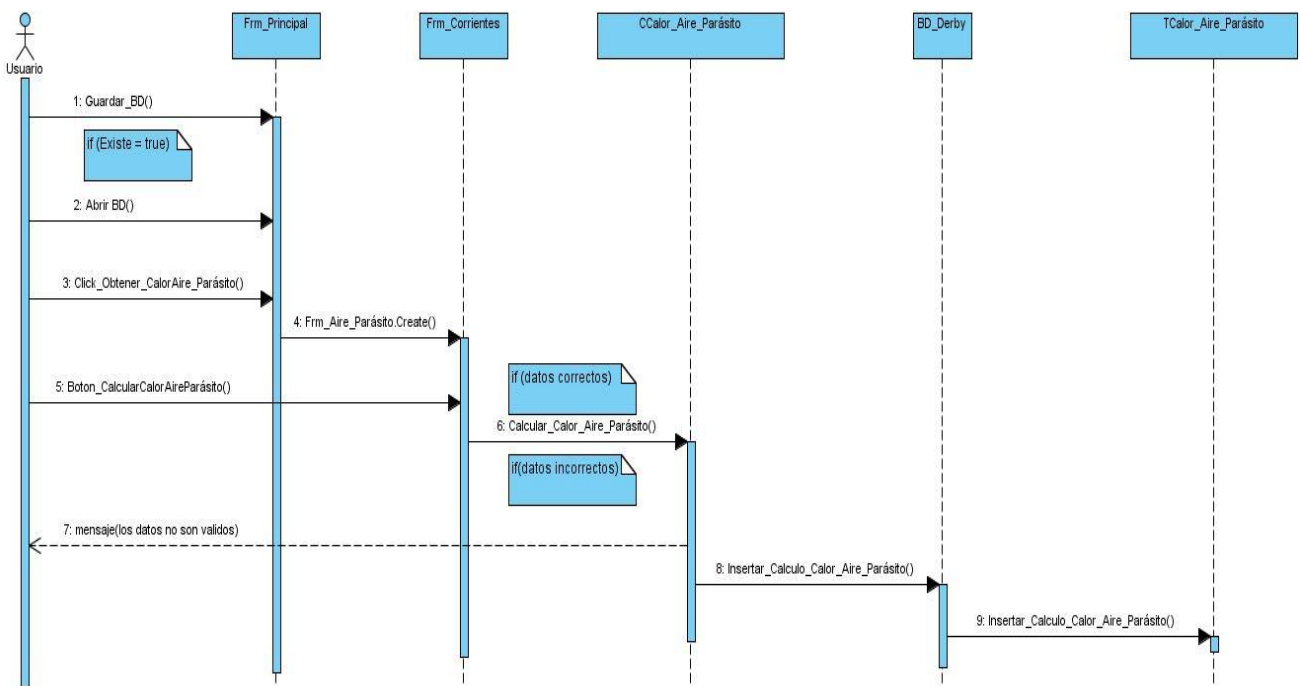


Figura 2. Diagrama de secuencia del CU “Calculo del Aire Parásito”

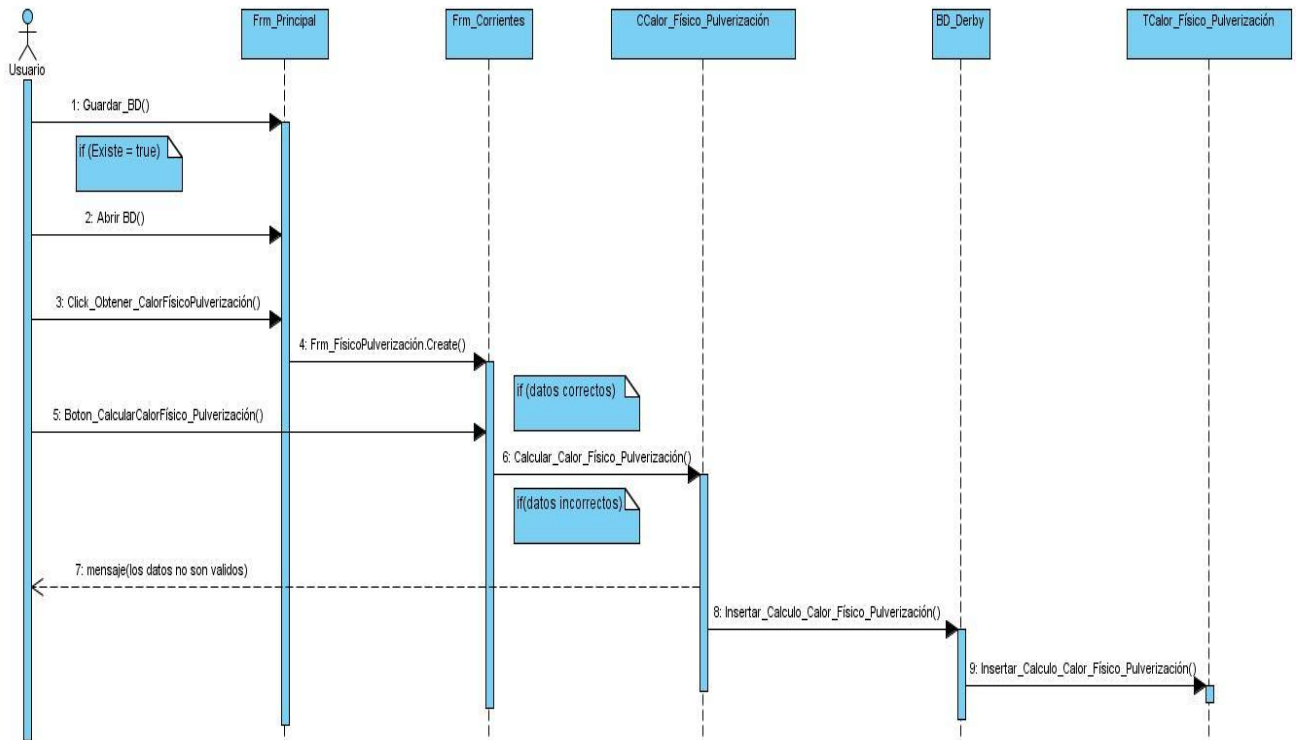


Figura 3. Diagrama de secuencia del CU “Calor Físico de Pulverización”

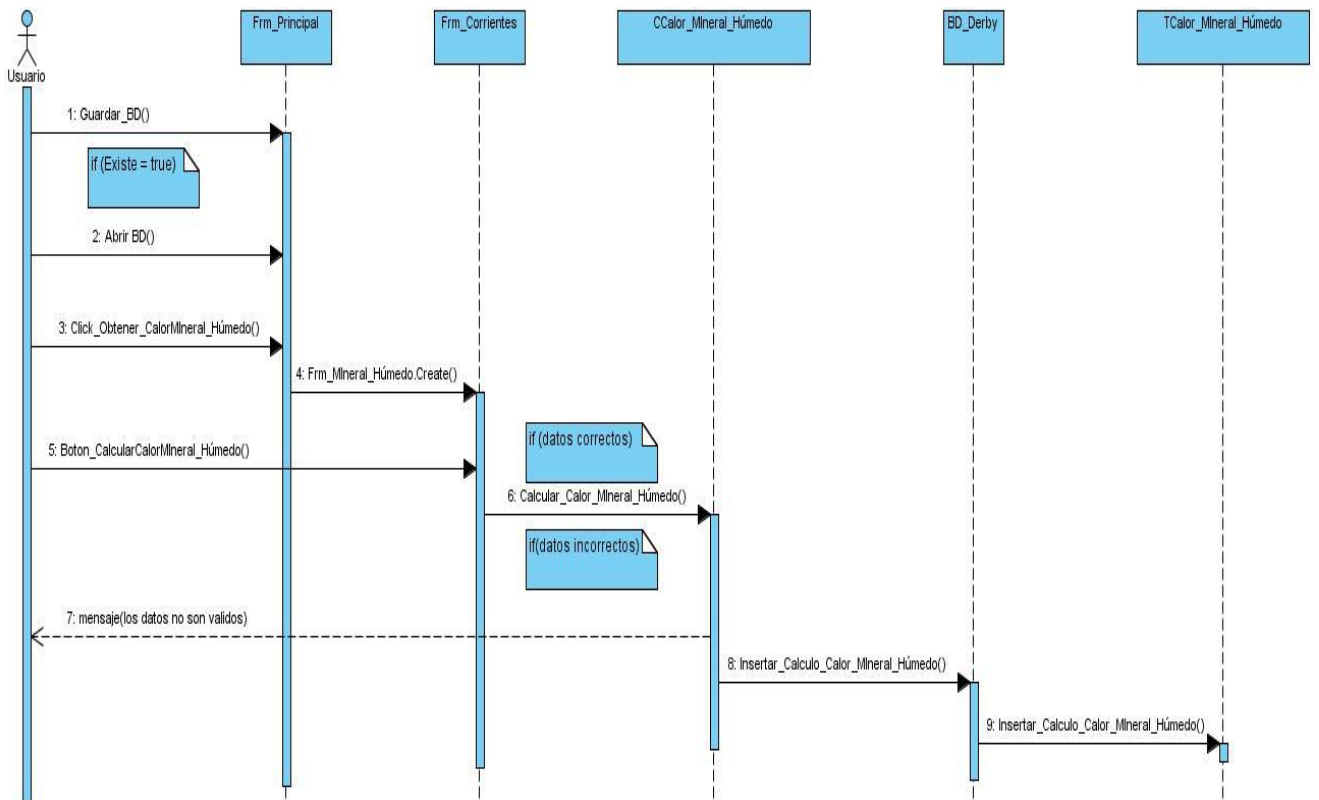


Figura 4. Diagrama de secuencia del CU “Calcular Calor Aportado por Mineral Húmedo”

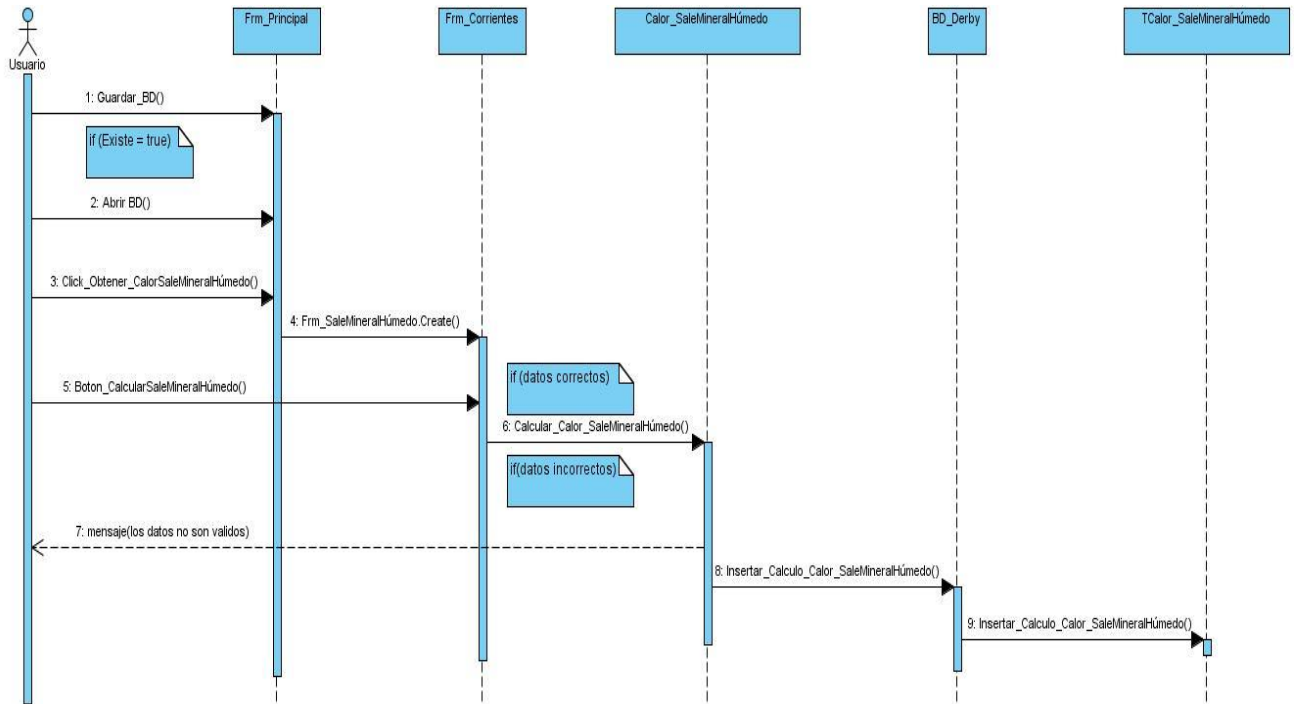


Figura 5. Diagrama de secuencia del CU “Calcular Calor Sale del Mineral Húmedo”

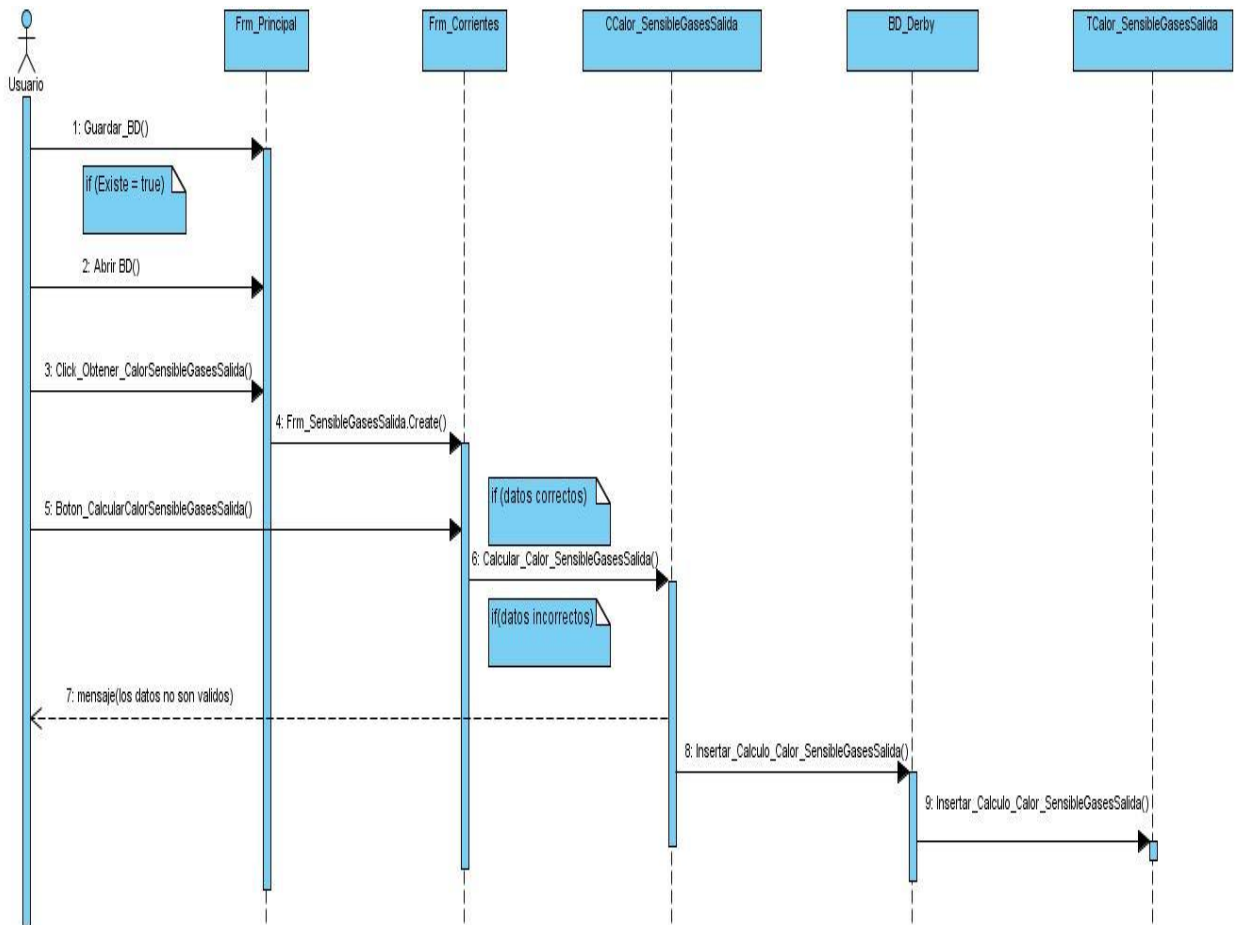


Figura 6. Diagrama de secuencia del CU “Calcular Calor Sensible a Gases de Salida”

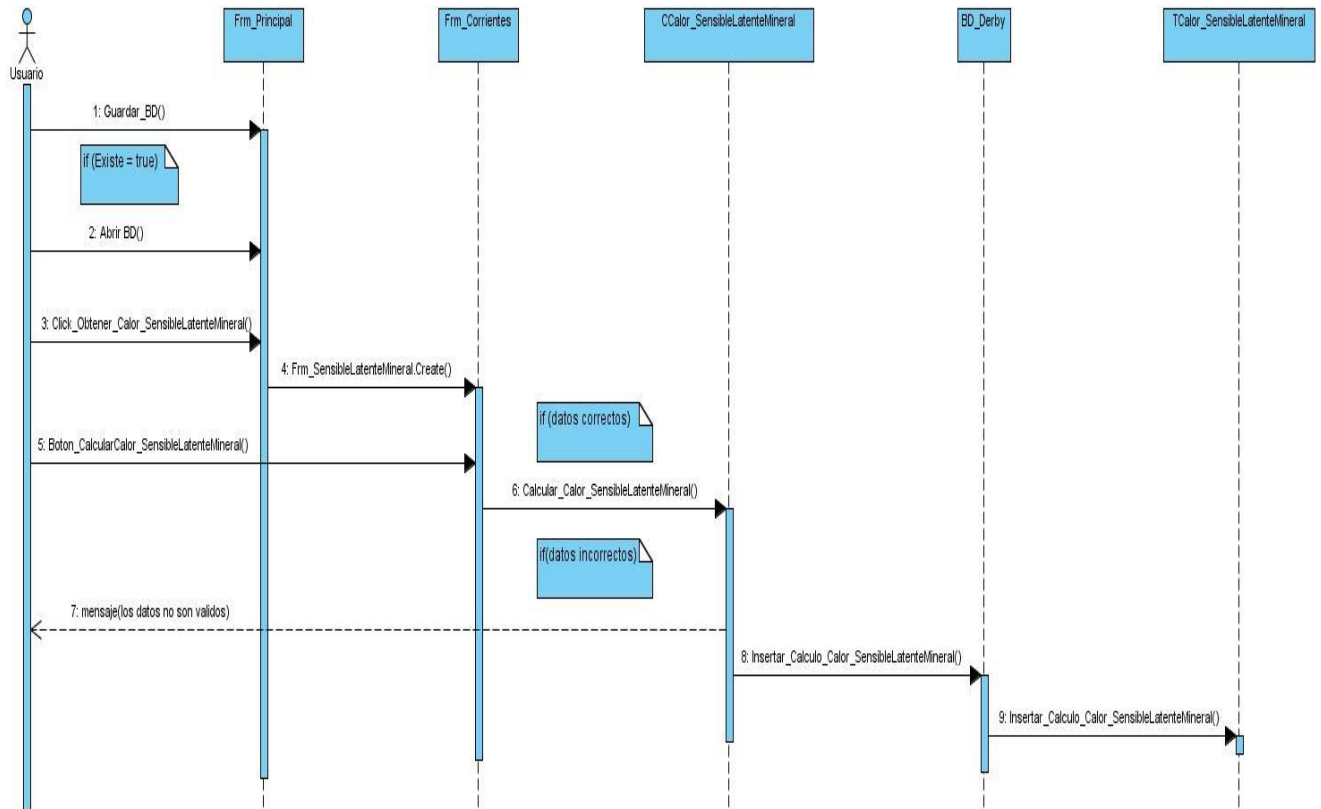


Figura 7. Diagrama de secuencia del CU “Calcular Calor Sensible Latente Mineral”

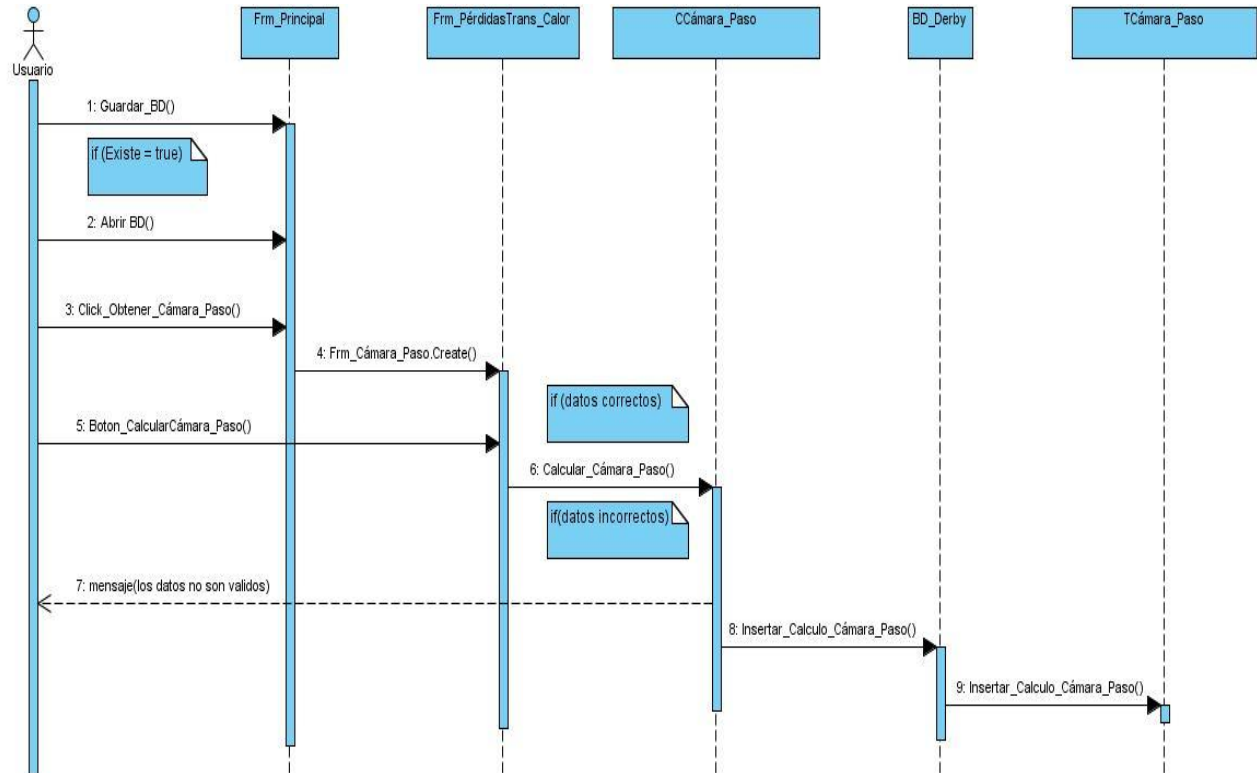


Figura 8. Diagrama de secuencia del CU “Calcular Cámara de Paso”

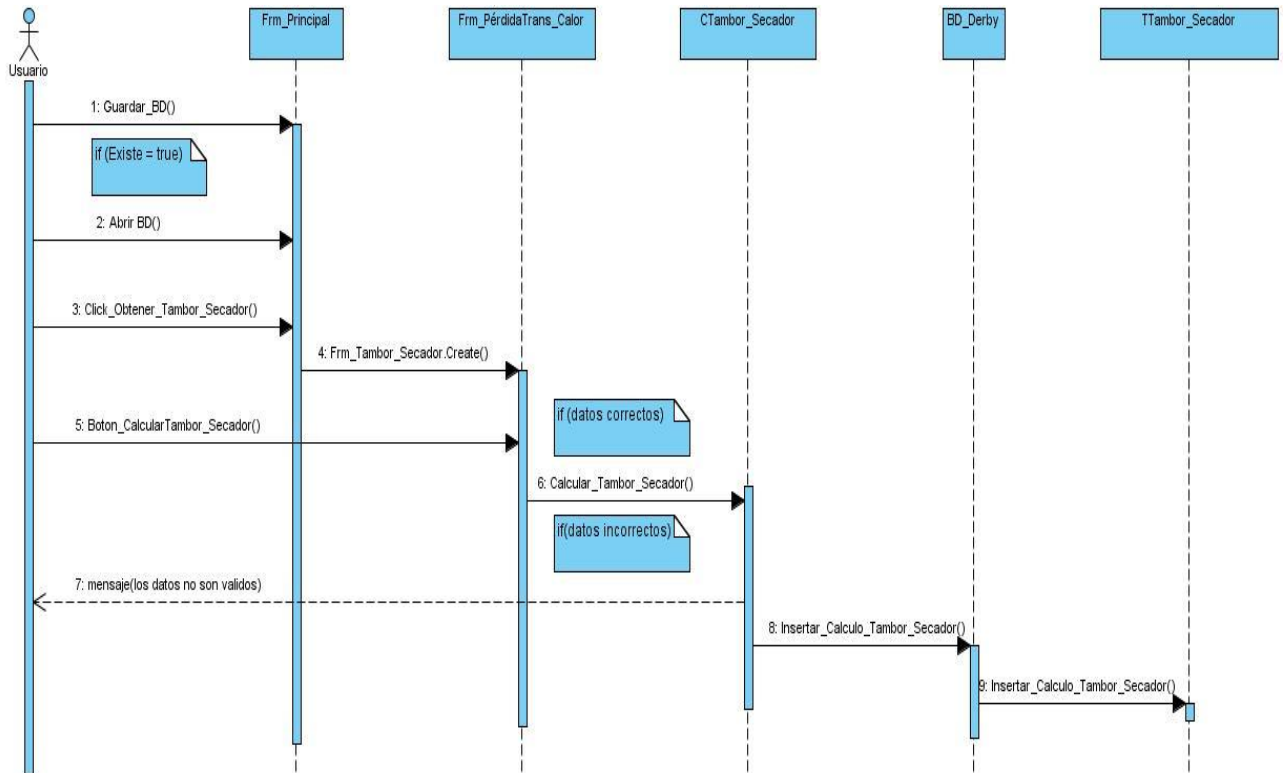


Figura 9. Diagrama de secuencia del CU “Calcular Tambor Secador”

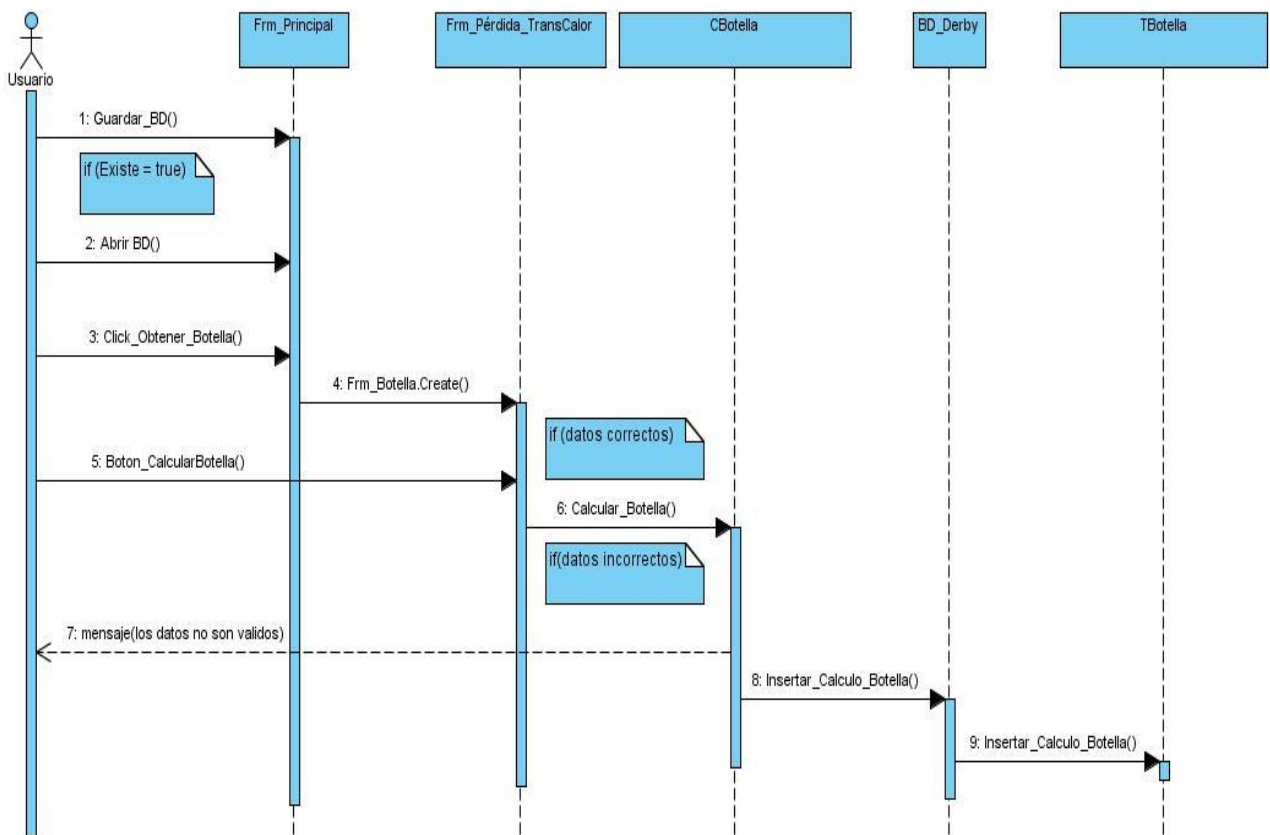


Figura 10. Diagrama de secuencia del CU “Calcular Botella”

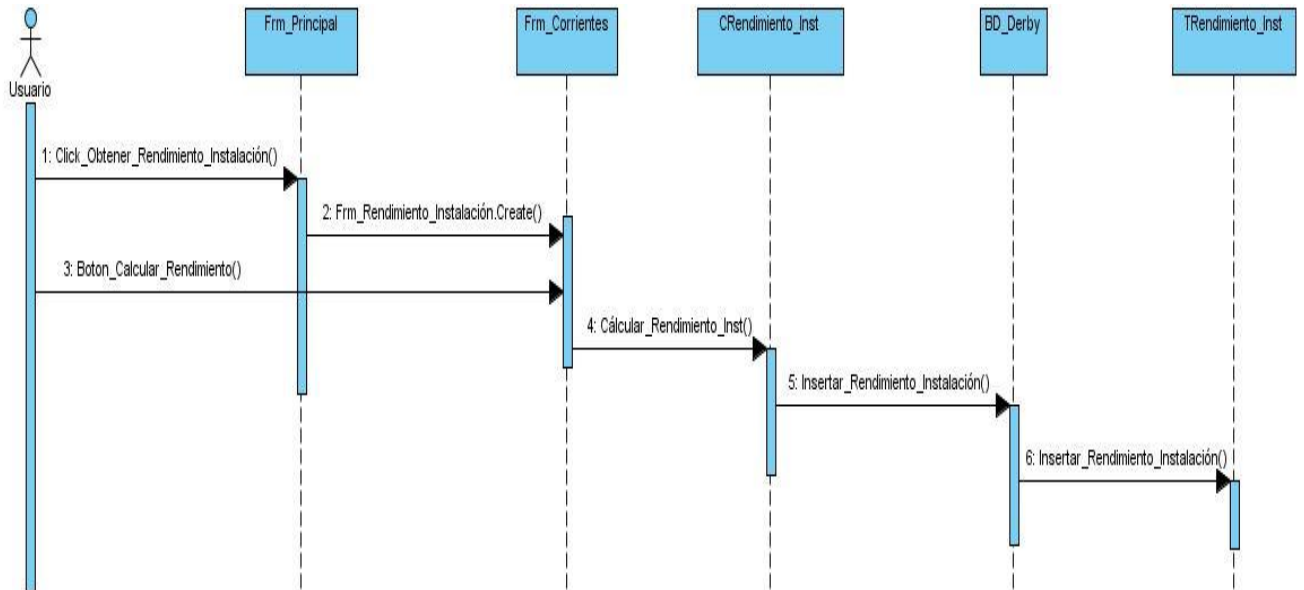


Figura 11. Diagrama de secuencia del CU “Calcular Rendimiento de la Instalación”

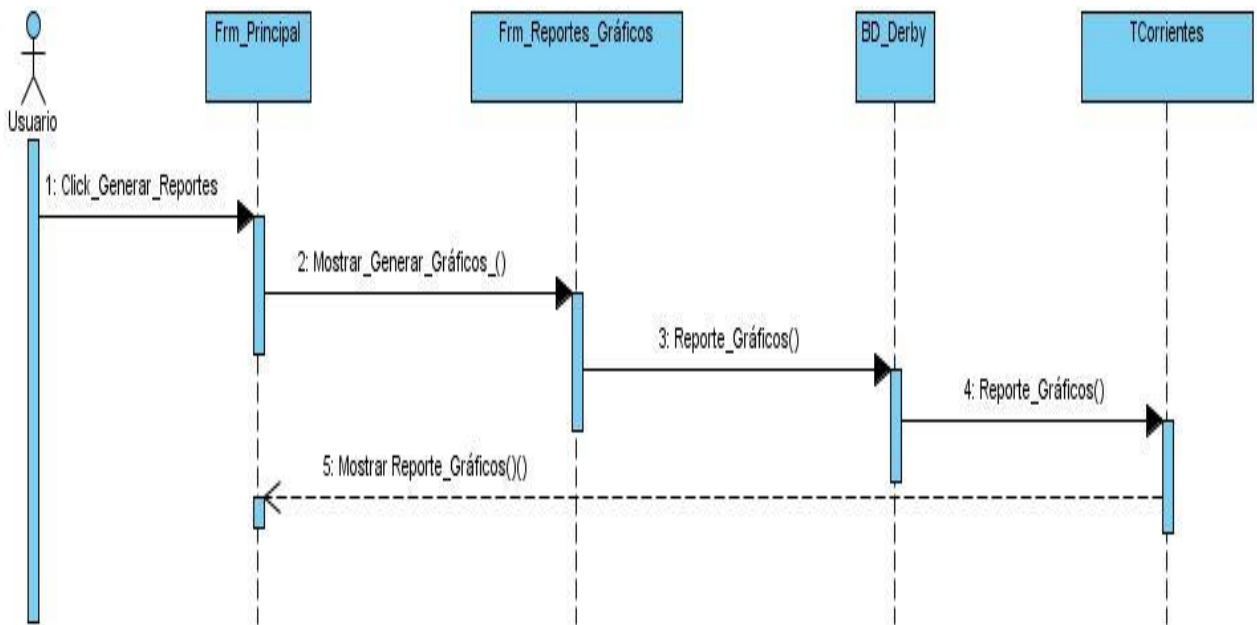


Figura 12. Diagrama de secuencia del CU “Graficar respecto a flujos”

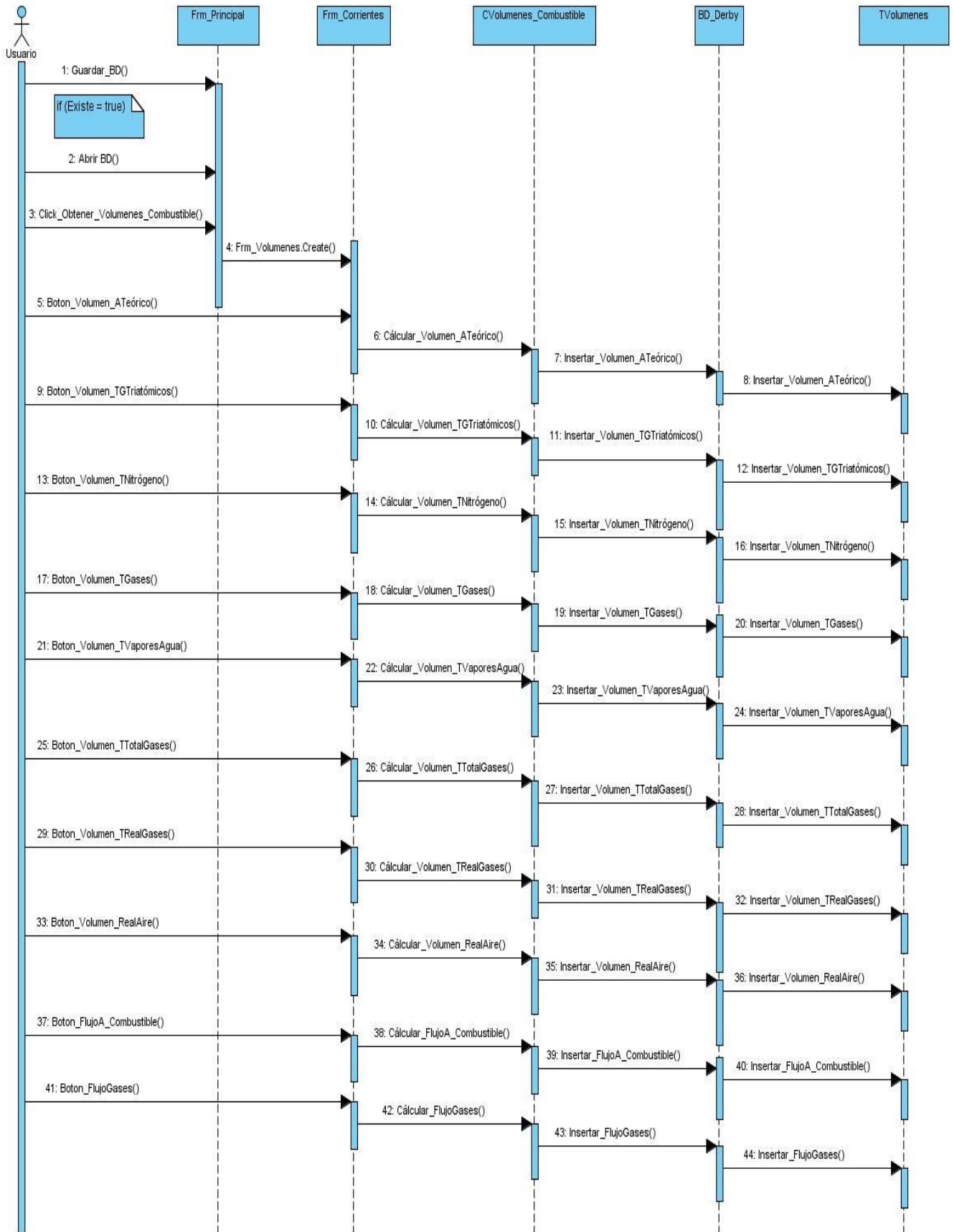


Figura 13. Diagrama de secuencia del CU “Calcular Volúmenes de Combustible”

Diagrama de secuencia del CU "Administrar Base de Datos"

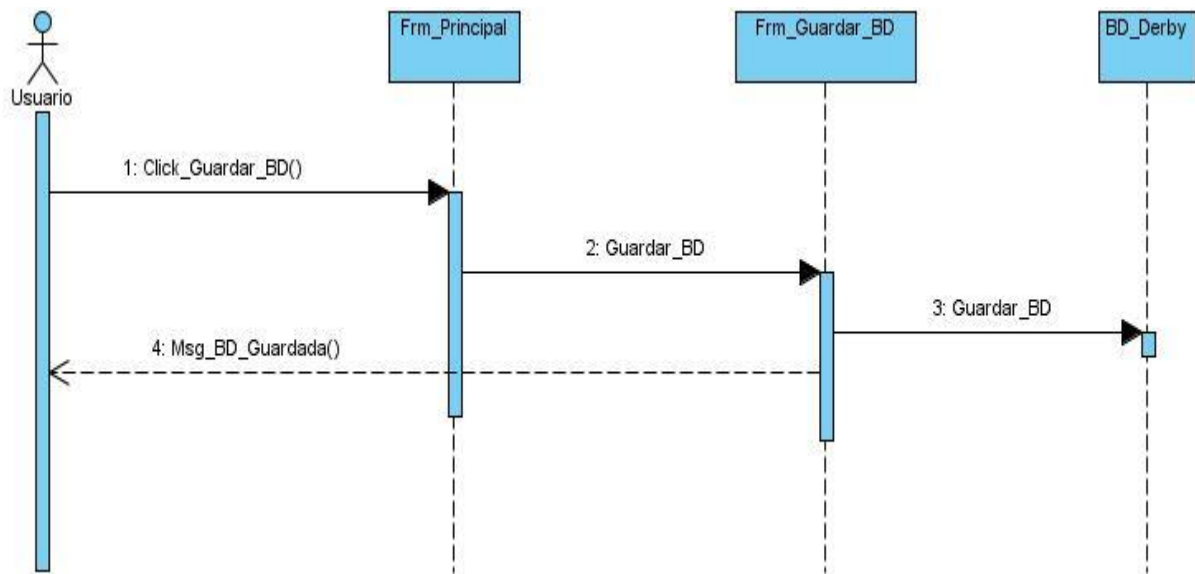


Figura 14.1. Sección "Guardar Base de Datos"

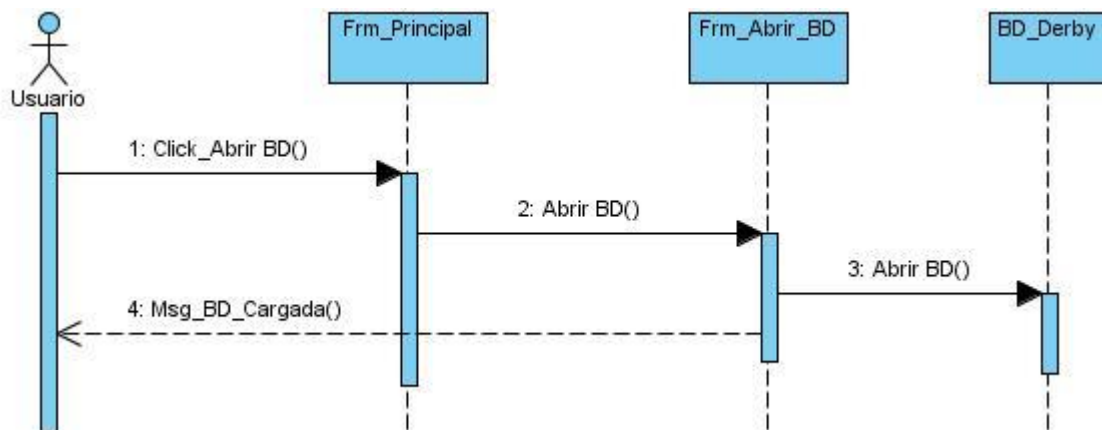


Figura 14.2. Sección "Abrir Bases de Datos"

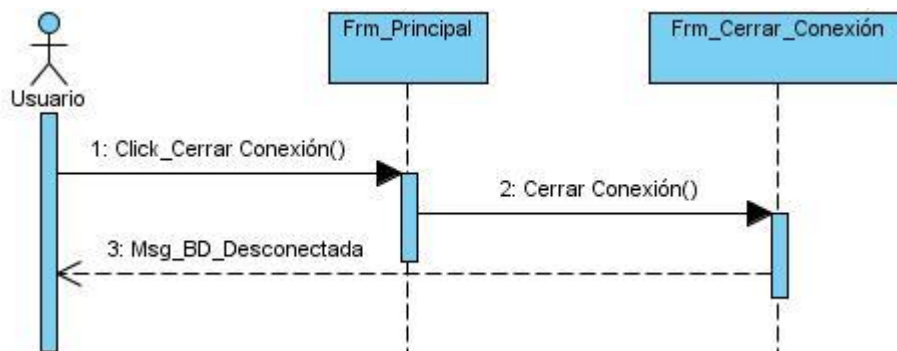


Figura 14.3. Sección "Cerrar Conexión de Bases de Datos"

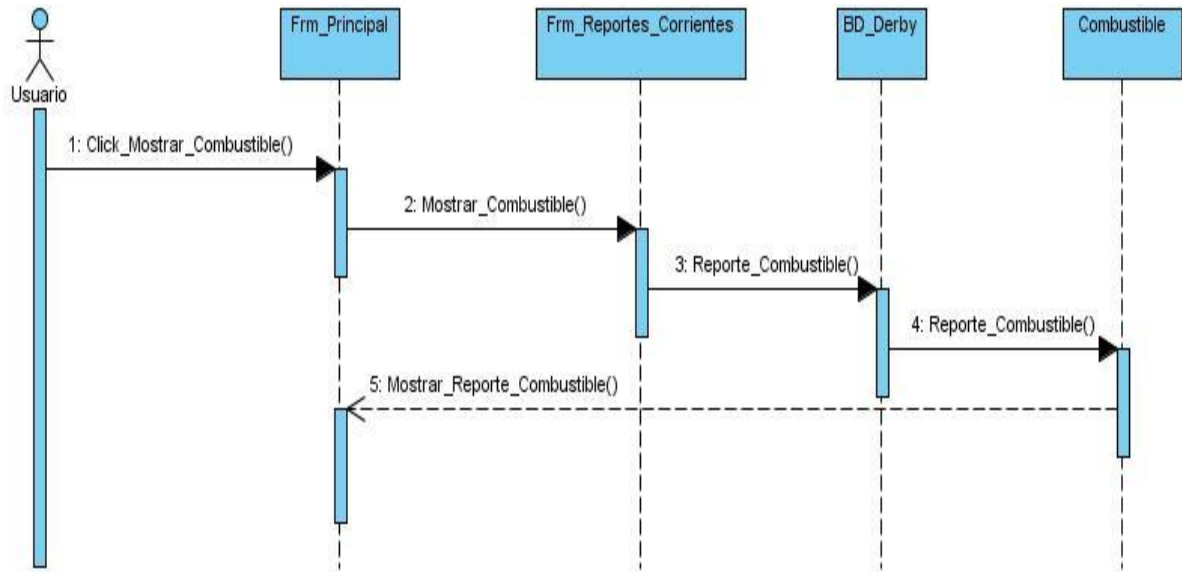


Figura 15. Diagrama de secuencia del CU “Mostrar Reporte Calor Aportado por Combustible”

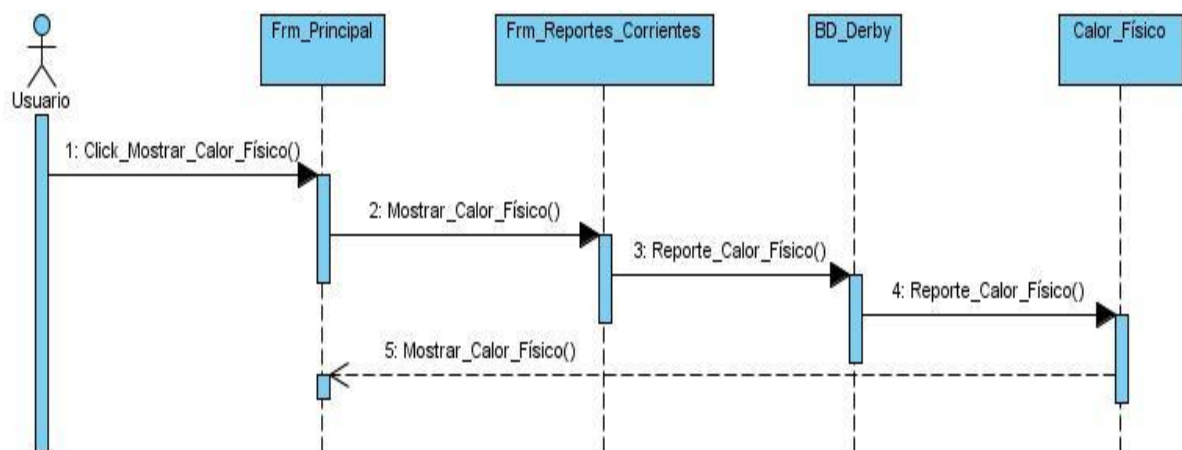


Figura 16. Diagrama de secuencia del CU “Mostrar Reporte Calor Físico del Combustible”

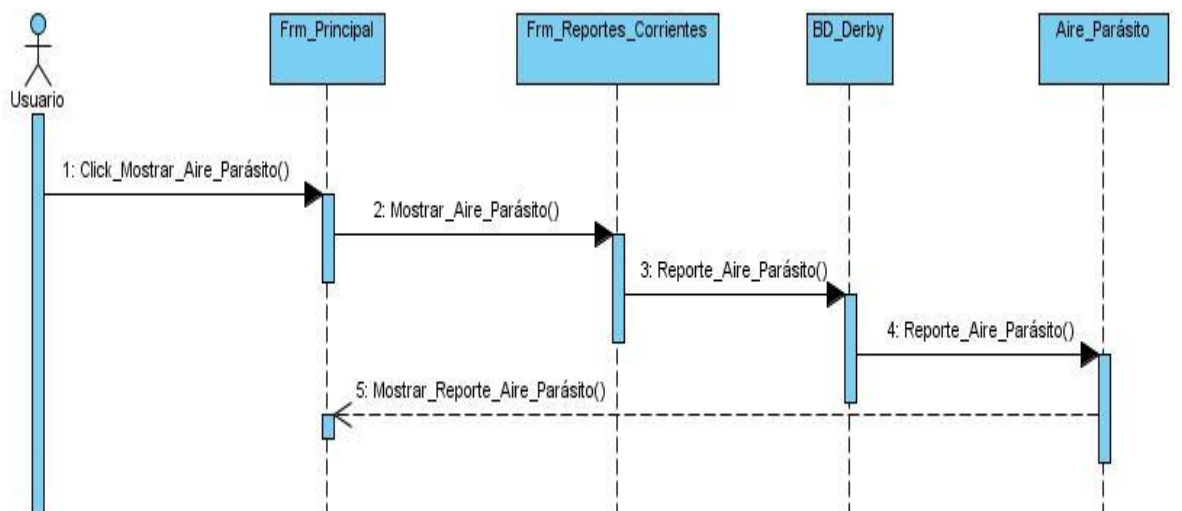


Figura 17. Diagrama de secuencia del CU “Mostrar Reporte para Calcular Aire Parásito”

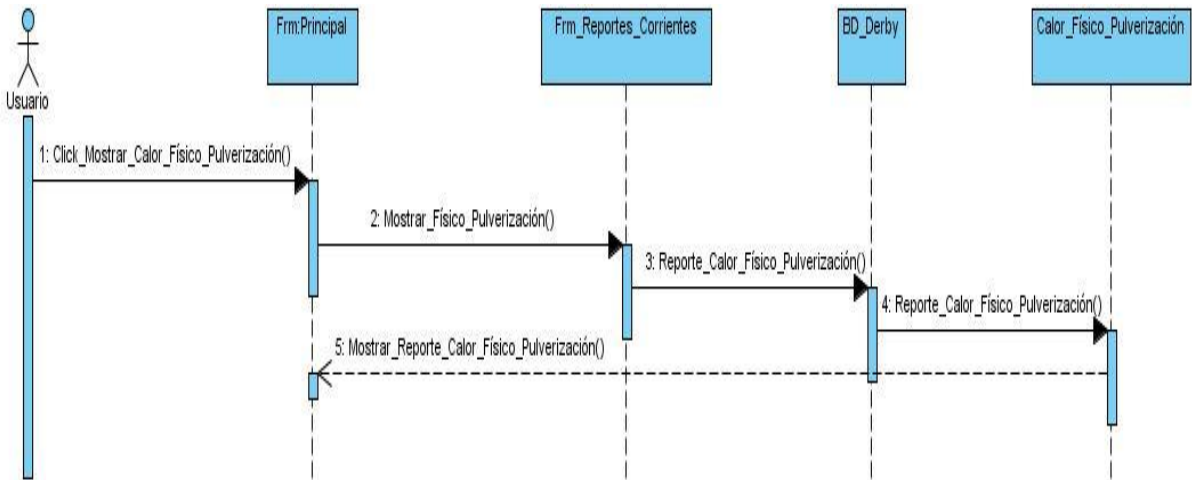


Figura 18. Diagrama de secuencia del CU “Mostrar Reporte Calor Físico de Pulverización”

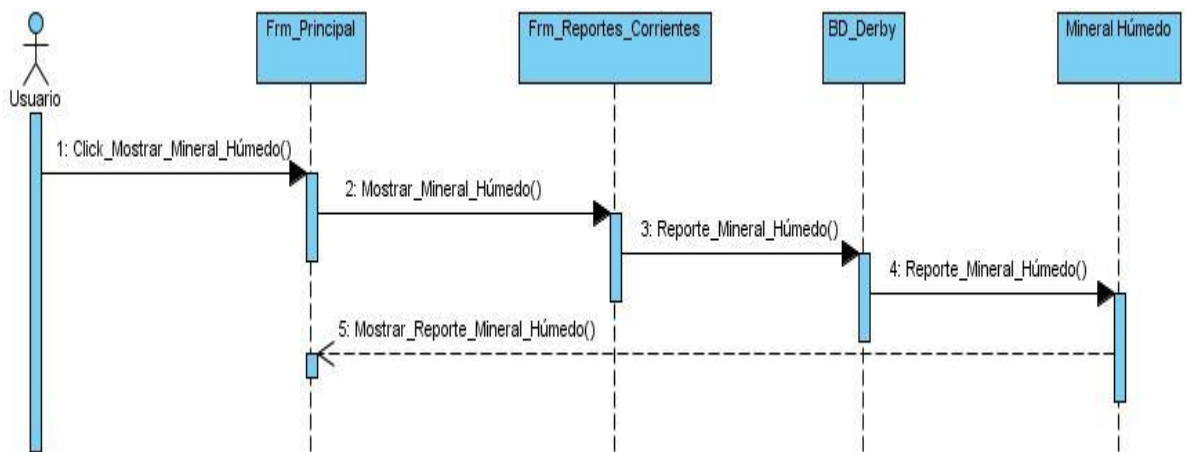


Figura 19. Diagrama de secuencia del CU “Mostrar Reporte Calor del Mineral Húmedo”

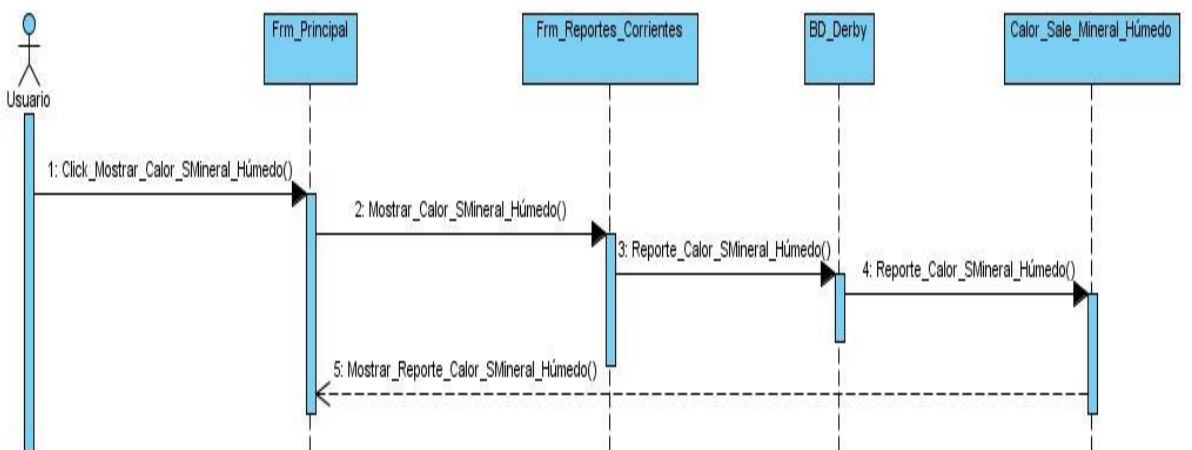


Figura 20. Diagrama de secuencia del CU “Mostrar Reporte Calor Sale del Mineral Húmedo”

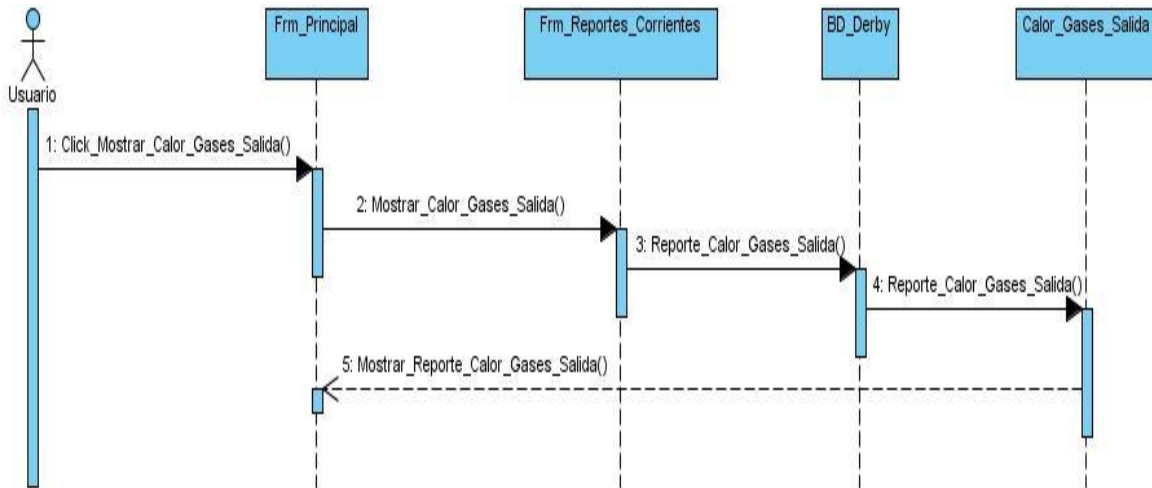


Figura 21. Diagrama de secuencia del CU “Mostrar Reporte Calor Gases Salida del Secador”

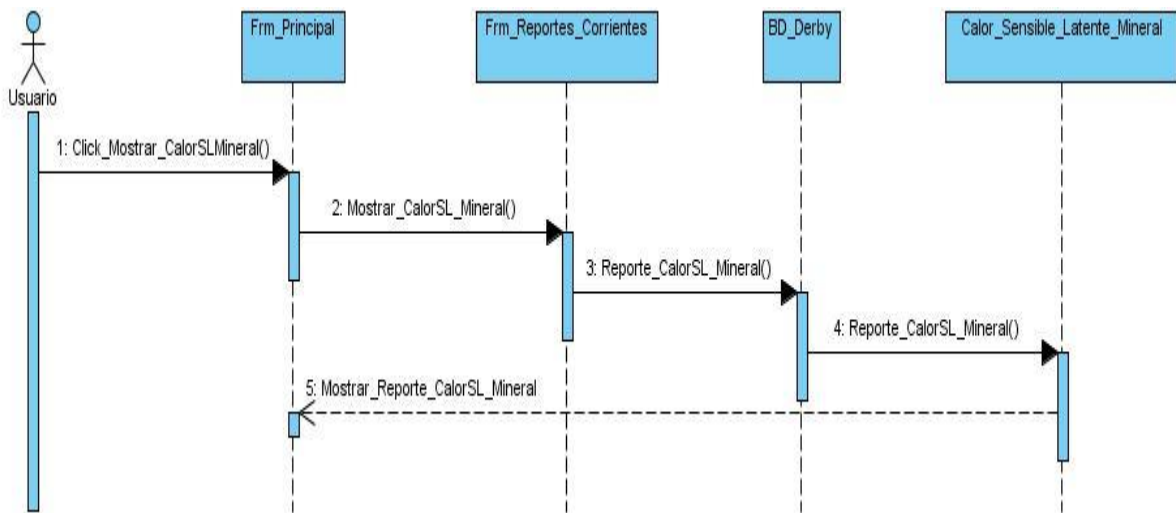


Figura 22. Diagrama de secuencia del CU “Mostrar Reporte Calor Sensible y Latente del Mineral”

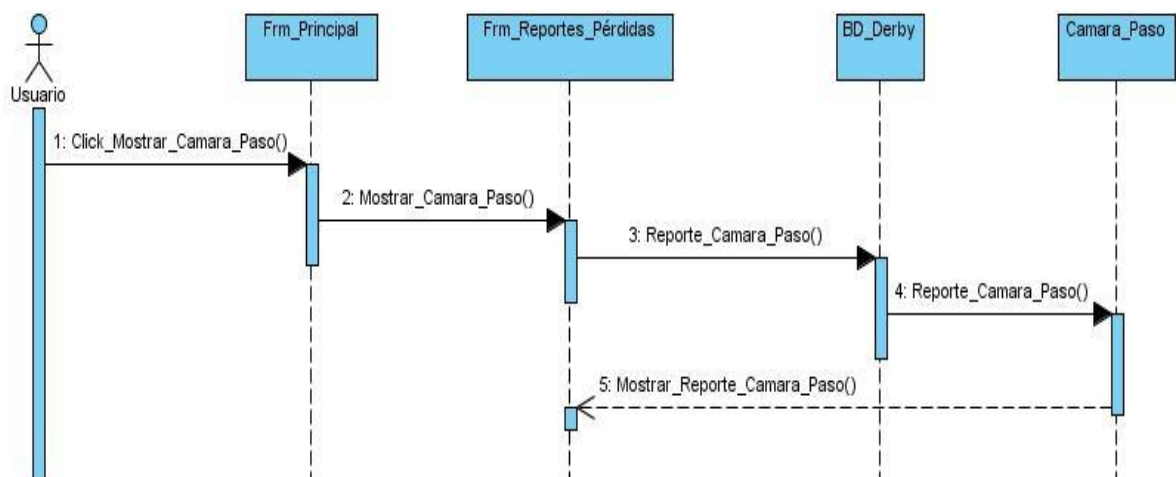


Figura 23. Diagrama de secuencia del CU “Mostrar Reporte Cámara de Paso”

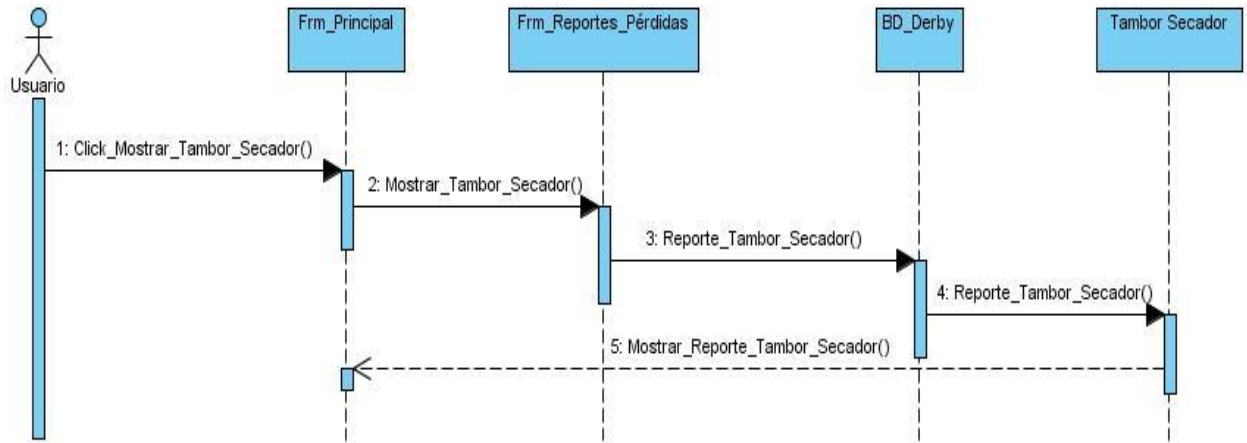


Figura 24. Diagrama de secuencia del CU “Mostrar Reporte Tambor Secador”

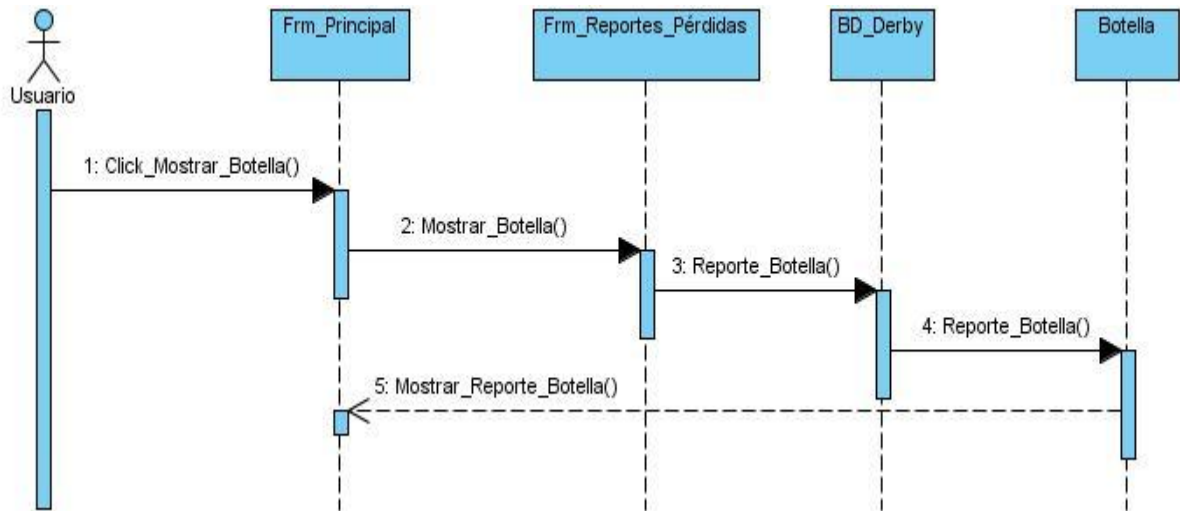


Figura 25. Diagrama de secuencia del CU “Mostrar Reporte Botella”

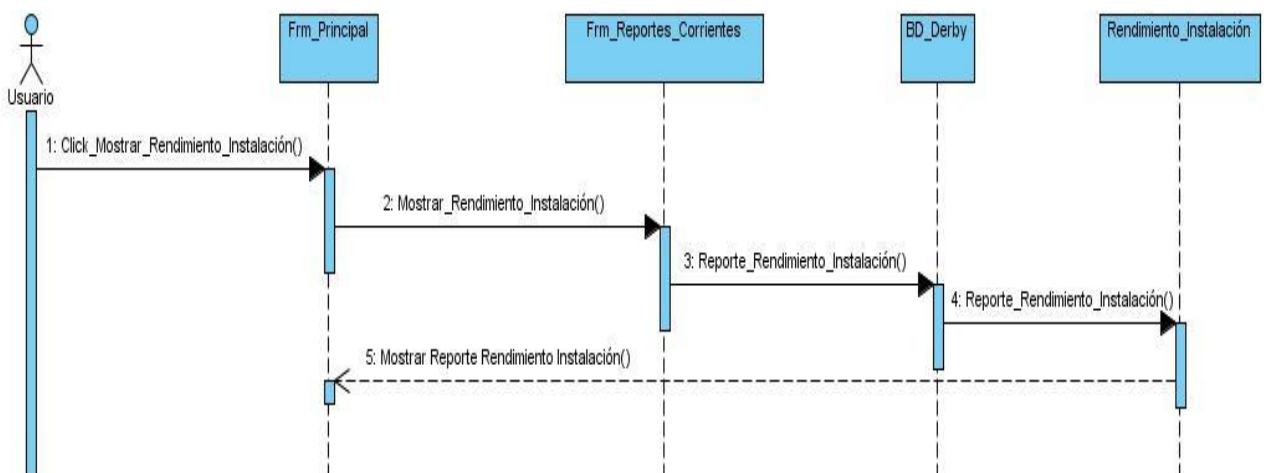


Figura 26. Diagrama de secuencia del CU “Mostrar Reporte del Rendimiento de la Instalación”

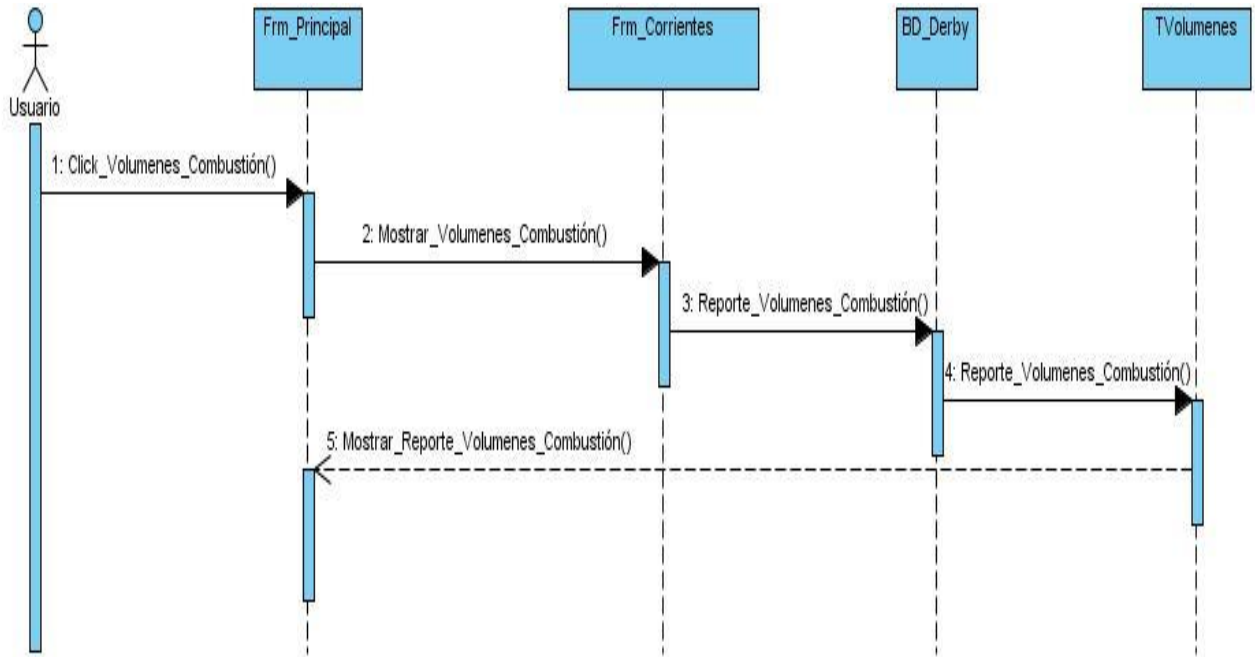


Figura 27. Diagrama de secuencia del CU “Mostrar Reporte de Volúmenes de Combustible”

Glosario de Términos

Aplicación: Es el programa que el usuario activa para trabajar en el ordenador. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

API: Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

JDBC: es el acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

RMI: (Java Remote Method Invocation) es un mecanismo ofrecido en Java para invocar un método remotamente. Al ser RMI parte estándar del entorno de ejecución Java, usarlo provee un mecanismo simple en una aplicación distribuida que solamente necesita comunicar servidores codificados para Java.

RUP: El Proceso Unificado Rational (RUP) es una metodología de desarrollo para la programación orientada a objetos. Según Rational (diseñadores de Rose Rational y el Idioma Modelado Unificado), RUP está como un mentor en línea que mantiene pautas, plantillas, y ejemplos de todos los aspectos y fases de desarrollo del programa. RUP es un software comprensivo que diseña herramientas que combinan los aspectos procesales de desarrollo (como las fases definidas, técnicas, y prácticas) con otros componentes de desarrollo (como los documentos, modelos, manuales, el código, y así sucesivamente) dentro de una armazón unificándose.

Metodologías de Desarrollo: Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

Multiplataforma: Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Programación Extrema(XP): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Software Libre: Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente, aunque conserve su carácter de libre, puede ser vendido comercialmente.