

Universidad de las Ciencias Informáticas

Facultad 4



**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Título:

Programa para la simulación del proceso de difusión molecular en láminas delgadas y membranas de zeolitas, por método de Monte Carlo Directo.

Autor:

Leonardo Soto Borroto

Tutor:

Dr. Carlos R. González González

Lic. Edisel Navas Conyedo

Ciudad de La Habana, Cuba

Junio del 2009



“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente.”

AGRADECIMIENTOS

En la vida existen muchas personas que te enseñan y guían, otras que te ayudan o agradecen el estar a tu lado, cada una representa una pequeña parte, pero indispensable, en la construcción de la persona que he llegado a ser hoy en día. A todas ellas por su cariño, apoyo y ejemplo brindado quisiera agradecerles.

A la Revolución Cubana, también quisiera agradecerle, por el esfuerzo hecho por llevar el sueño UCI a la realidad y por sobre todo brindar Educación Universitaria a todos por igual sin diferencia social.

DEDICATORIA

A mis padres Leonardo Soto Romero, Elaine María Borroto González y Alberto Alvarez Sousa por su ejemplo y ayuda en todo momento.

A mi abuelo que no pudo ser universitario y soñó que su familia lo fuera, por sus consejos y cariño.

A Checho que fue un padre en una etapa de mi vida y Mirita que ha sido como una hermana para mí.

A mi hermano.

A Belkis Yadira Torres González por su ayuda y dedicación.

A mis amigos que siempre estuvieron a mi lado en los momentos malos y buenos.

A mi grupo de la universidad.

Y a cada una de esas personas que me ayudaron en la conformación y redacción del trabajo.

A todos gracias.

RESUMEN

El desarrollo tecnológico ha posibilitado que se simulen diferentes procesos de difusión de mezclas de gases en sólidos policristalinos. La simulación en estos procesos reduce el tiempo de espera para lograr un resultado considerablemente y aumenta la confiabilidad de los resultados. En el trabajo se presenta un algoritmo del programa TRANSMOL, destinado a la simulación de los procesos difusivos de mezclas de gases en el interior de sustancias sólidas microporosas como las membranas policristalinas de Zeolitas. Las Zeolitas están compuestas por aluminio, silicio, oxígeno, cationes intercambiables tales como Na, K, Ca y un gran número variable de moléculas absorbidas; entre las que se encuentran el agua y otras sustancias gaseosas. Las sustancias que la componen así como sus propiedades hacen de éstas un material importante en la industria química por sus propiedades adsorbentes catalíticas y de intercambio iónico.

La simulación se basa en un modelo simplificado de la membrana zeolítica, compuesta por una microcolumna de sección transversal pequeña, tomada a todo lo ancho de la membrana, en la dirección donde debe ocurrir el flujo difusivo molecular. La microcolumna estaría formada por cristalitas idénticas de zeolitas, con diversas orientaciones, y dos tipos de defectos volumétricos (atajo y barrera), con lo cual se representaría un sólido microporoso policristalino real; y no ideal.

El desarrollo del algoritmo de simulación en una aplicación informática permitirá avanzar en el estudio de la Zeolita, en su aplicación en la industria del petróleo y en la química fina en nuestro país.

Palabras Claves:

Difusión, Membrana Zeolítica, Método de Monte Carlo, Separación de Gases, Simulación.

INDICE

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
INTRODUCCION	VII
CAPITULO I: FUNDAMENTACION TEORICA	1
1.1 Introducción	1
1.2 Sistemas de Simulación de procesos de difusión molecular vinculados al campo de acción.	1
1.2.1 Antecedentes Nacionales	1
1.2.2 Investigaciones Internacionales.....	2
1.2.3 Softwares vinculados al Campo de Acción	3
1.3 Metodología a Utilizar	4
1.3.1 Lenguaje Unificado de Modelado (UML)	4
1.3.2 El Proceso Unificado del Rational	6
1.3.3 Método de Desarrollo de Sistemas Dinámicos (DSDM)	9
1.3.4 Programación Extrema (XP)	10
1.3.5 Comparación de DSDM con otras metodologías de desarrollo rápido	11
1.3.6 Selección de la Metodología de Desarrollo	12
1.4 Modelación del Espacio Poroso	12
1.4.1 Constitución de la Microcolumna	12
1.4.2 Selección de los TDA	13
1.4.3 Modelación de las Interacciones	14
1.5 Definición de un Lenguaje de Programación	14
1.5.1 Lenguaje de Programación	14
1.5.2 Clasificación de los Lenguajes de Programación	15
1.5.3 Selección de un Lenguaje de Programación	17
1.6 Sistemas de Gestión de Bases de Datos (SGBD)	21
1.6.1 PostgreSQL Sistema de Gestión de Bases de Datos.....	22
1.7 Selección de la herramienta CASE a utilizar	26
1.8 Conclusiones.....	26
CAPITULO II: CARACTERISTICAS DEL SISTEMA	27
2.1 Introducción	27
2.2 Modelo del Dominio.....	27
2.2.1 Conceptos del Dominio	28
2.2.2 Eventos del Dominio	29
2.2.3 Diagrama de Clases del Dominio	29
2.3 Propuesta del sistema	30
2.4 Requerimientos	31
2.4.1 Requerimientos Funcionales.....	31
2.4.2 Requerimientos No Funcionales	33
2.5 Modelo del Sistema.....	35

2.5.1	Actores del Sistema.....	35
2.5.2	Caso de Uso.....	35
2.5.3	Descripciones no expandidas de los casos de uso.....	36
2.5.4	Diagrama de Caso de Uso del Sistema.....	39
2.6	Descripción de los Casos de Uso del Sistema.....	40
2.6.1	Descripción Caso de Uso Controlar Acceso	40
2.6.2	Descripción Caso de Uso Gestionar Usuario.....	42
2.6.3	Descripción Caso de Uso Gestionar Zeolita	51
2.6.4	Descripción Caso de Uso Gestionar Gas	60
2.6.5	Descripción Caso de Uso Gestionar Mezcla Gas.....	68
2.6.6	Descripción Caso de Uso Gestionar Estudio	76
2.6.7	Descripción Caso de Uso Gestionar Barrera	82
2.6.8	Descripción Caso de Uso Gestionar Atajo.....	85
2.6.9	Descripción Caso de Uso Detener Estudio.....	88
2.6.10	Descripción Caso de Uso Guardar Resultado.....	92
2.6.11	Descripción Caso de Uso Entregar Resultado	93
2.7	Conclusiones.....	96
CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION		93
3.1	Introducción:	93
3.2	Análisis.....	93
3.2.1	Diagramas de Clases del Análisis.	93
3.2.2	Diagramas de Colaboración de Clases Análisis.	94
3.3	Diseño.....	94
3.3.1	Clases del Diseño.....	94
3.4	Arquitectura Agil	95
3.4.1	Diagrama de Despliegue	98
3.4.2	Diagrama de Componentes	98
3.5	Diseño de la Base de Datos.	99
3.5.1	Diagrama de Clases Persistentes.	99
3.5.2	Diagrama Entidad Relación.	101
3.5.3	Descripción de las Tablas del Modelo Entidad Relación.....	101
3.6	Pruebas.....	110
3.6.1	Pruebas Unitarias	111
3.7	Conclusiones	113
Conclusiones:		116
Recomendaciones:		116
Bibliografía:		117
Citada:		117
Consultada:		117
Glosario de Términos:.....		119
Anexos		120

INTRODUCCION

La investigación científica de la Zeolita constituye unos de los retos del mundo actual, donde la búsqueda constante de propiedades químicas y físicas en ésta es la base de sus investigaciones. La Zeolita por su composición y estructura tiene propiedades que hacen de ella un adsorbente selectivo en los procesos de separación de gases. Un gas a través de un cristal de Zeolita puede separarse en sus componentes moleculares, lo cual es muy utilizado en la industria para la producción en derivados del petróleo y del gas natural. Con el desarrollo de las Tecnologías de la Informática y de las Comunicaciones se puede lograr el desarrollo de un software que ilustre y optimice el proceso de difusión de mezclas de gases en sólidos policristalinos, formados por cristalitos microporosos tales como las zeolitas, incluyendo defectos de superficie y volumétricos, utilizando métodos de Monte Carlo cinéticos.

Para su creación se hace necesario realizar un estudio detallado, garantizando así una gestión eficiente de los diferentes tipos de Zeolitas; además de una configuración visual amena donde el usuario navegue con facilidad, permitiendo el acceso de forma rápida, dinámica y precisa a los datos almacenados.

Los procesos del petróleo para obtener sus derivados está en constante estudio logrando niveles cada vez mayores de eficiencia en su obtención. En Cuba durante los últimos años ha cobrado fuerza la creación de sistemas informáticos para dar solución a disímiles problemas de la industria, y otros sectores de la economía y la sociedad cubana; siendo los de la industria petroquímica una prioridad. "Cubapetróleo" es la entidad encargada del manejo, estudio y análisis del petróleo y sus derivados, el estudio de estos ocurre de forma experimental en laboratorios y centros de investigación nacional. Si Cuba quisiera utilizar software para simular y estudiar la difusión de los gases, tendría que invertir grandes sumas de dinero, no solo en la adquisición, si no también, en el soporte que este requiere. Con el desarrollo de este programa el personal de soporte y mantenimiento sería nacional, y su aplicación no estaría sujeta a mecanismos de mercado impuesto por las grandes compañías.

Esta tesis se estructura de tres capítulos, cada uno de los cuales brinda una información detallada del desarrollo de la aplicación, para lograr en su conjunto su entendimiento y aprendizaje. El primer capítulo aborda la fundamentación teórica, el estado del arte de los sistemas informáticos vinculados al

objeto de estudio, las técnicas, tecnologías, tipos de datos abstractos utilizados, herramientas empleadas y la metodología para el desarrollo del software empleado.

Durante el segundo capítulo se plantea una propuesta concreta de la aplicación a desarrollar, se define el modelo de negocio, se obtienen los requerimientos y se determinan los casos de usos y actores del sistema.

Para un mayor entendimiento se describen estos casos de usos y se muestra la matriz de trazabilidad.

En el tercer capítulo se crean las clases del análisis, diagramas de colaboración del análisis, diagrama de clase del diseño, el modelo de datos y las pruebas realizadas al sistema.

Una vez hecho este análisis se puede decir que el **problema a resolver** es: la carencia de una herramienta capaz de simular la difusión de gases en membranas zeolíticas reales, compuestas de múltiples cristalitas de zeolitas de diferentes orientaciones espaciales, unidas entre sí, pero con numerosos defectos de tipo barrera y atajo: o sea, regiones impenetrables y canales de libre tránsito.

Presentándose el **problema científico**: ¿Cómo modelar una herramienta capaz de simular la difusión de gases en membranas zeolíticas reales, compuestas de múltiples cristalitas de zeolitas de diferentes orientaciones espaciales, unidas entre sí, pero con numerosos defectos de tipo barrera y atajo: o sea, regiones impenetrables y canales de libre tránsito?

El desarrollo de las Tecnologías de la Informática y las Comunicaciones han permitido la programación de algoritmos cada vez más complejos, logrando el desarrollo de aplicaciones que simulan diferentes procesos difusivos. Estos procesos en membranas zeolíticas son flujos netos de masa que ocurren entre las dos caras a distintas concentraciones.

Con la investigación y posterior implementación de este trabajo se propone construir una herramienta capaz de simular los procesos de difusión molecular en un estrato de Zeolita que permita un mayor rendimiento en la producción de los derivados de la industria química y de la petroquímica, logrando que la aplicación se convierta en un sitio de referencia obligatorio.

El **Objeto de Estudio** es: el proceso de difusión molecular en membranas policristalinas de Zeolitas en estratos no perfectos.

El **Campo de Acción** lo constituyen: los sistemas de simulación de difusión molecular en membranas Zeolíticas, con aplicación en la industria química y la investigación científica.

Como **hipótesis** se parte de la idea de que, si se contara con una herramienta capaz de simular el proceso de difusión molecular en membranas zeolíticas utilizando métodos de Monte Carlo Directo, se mejoraría en el resultado y la modelación de estos procesos difusivos.

El **Objetivo General** del presente trabajo es: desarrollar una herramienta que permita la simulación de procesos de difusión molecular de mezclas de gases en membranas policristalinas de Zeolitas con orientaciones espaciales aleatorias en un estrato no perfecto, que pueda acoplarse al sistema de generación de estructuras zeolíticas.

Como **objetivos específicos** se definen:

- Realizar un estudio sobre formas y mecanismos de simulación de procesos difusivos de mezclas moleculares en sistemas porosos sólidos con defectos volumétricos del tipo de macroporos o mesoporos entre los granos cristalinos, y fronteras de granos por el método de Monte Carlo.
- Aplicar el algoritmo de cálculos de los procesos de modelación por Monte Carlo Directo y otros procesos afines de comprobación de modo que se optimice el tiempo y la memoria dedicada a la ejecución.
- Desarrollar una aplicación donde se utilicen los cálculos del método de Monte Carlo Directo para el sistema estudiado en un estrato.

Con el objetivo de resolver la situación problemática planteada y cumplir con nuestros Objetivos Específicos y Generales se han trazado un grupo de **Tareas**:

- El análisis de los estudios y sistemas de simulación de procesos de difusión molecular nacional e internacionalmente.
- La revisión y estudio de los métodos que se utilizan en el desarrollo de la aplicación.
- El estudio de las metodologías de análisis y diseño de sistemas informáticos que faciliten la creación y garanticen la calidad de la aplicación.
- La elección de las herramientas y lenguaje de programación en la realización del proyecto, así como la elección de la plataforma.

- La definición de las funcionalidades que tendrá el sistema de simulación de procesos de difusión molecular.
- El estudio de la ingeniería de software con el fin de conocer todo lo relacionado con los hardwares y softwares que se deben utilizar para abordar del modo más eficiente la elaboración de los programas de computación necesarios para la modelación. Las **ventajas** que posee el diseño de esta aplicación constituyen un punto de referencia en la simulación al vincular aspectos no analizados por software anteriormente, además de permitir:
 - El estudio de minerales en un tiempo muy inferior al experimental y costo muy bajo.
 - Permite la ejecución en tiempo real obteniendo un resultado en cualquier instante de tiempo sin alterar la marcha del proceso.
 - Permite estudiar situaciones extremas y tendencias, lo cual es muy difícil de lograr en la práctica experimental.

¿Como verificar los resultados?

Los resultados del proyecto se validarán mediante los cálculos que se efectúen con el programa de simulación TRANSMOL a través de:

- Los cálculos con TRANSMOL de sistemas simples conocidos y estudiados en la literatura con soluciones aceptadas por la generalidad de la comunidad científica; y comparación de los resultados.
- Comprobación de los resultados de TRANSMOL en estudios de situaciones complejas, con mediciones experimentales puntuales confiables.

CAPITULO I: FUNDAMENTACION TEORICA

1.1 Introducción

En el presente capítulo se brinda una panorámica general de los métodos de modelación de los procesos de simulación de difusión de mezclas de gases en sólidos policristalinos. Se describen herramientas, técnicas, lenguajes de programación y gestores de Base de Datos a utilizar con el fin de obtener una visión más exacta. La descripción de las definiciones asociadas al dominio del problema se tiene en cuenta, para comprender el negocio y la propuesta de solución.

La valoración de los software existentes en procesos de simulación brindan un punto de partida para la realización de este trabajo, por lo que una comparación entre estos sistemas y el que se propone es necesario. Durante el desarrollo del capítulo se realiza la descripción de la metodología de análisis y diseño escogida, así como las herramientas a emplear en la configuración y confección del sistema que permita crear un software compatible con los módulos del sistema Transmol.

1.2 Sistemas de Simulación de procesos de difusión molecular vinculados al campo de acción.

1.2.1 Antecedentes Nacionales

En nuestro país se han realizados numerosos estudios sobre las zeolitas naturales y sintéticas en importantes instituciones nacionales a partir de 1976, que dieron lugar a la ejecución de un Programa Nacional entre 1980 y 1985. Como parte de éstos estuvieron las aplicaciones industriales de las zeolitas que tenían que ver con la separación de gases, la refrigeración, la descontaminación ambiental, entre otras. Por otra parte, se han realizado muchos trabajos referidos a la obtención y aplicaciones del carbón activado. Este tipo de material tiene también notables propiedades adsorptivas y entre otras aplicaciones, puede usarse en la separación de gases. En los últimos años se han estado sintetizando membranas y monolitos de distintos tipos útiles en la separación de gases aprovechando las diferencias en las movilidades moleculares debidas a las diferentes interacciones con el medio poroso. Otros trabajos que resultan antecedentes a las investigaciones que se proponen tienen que ver con las tecnologías de separación de gases por los métodos de PSA y VSA empleando zeolitas, en los cuales se han obtenido notables resultados prácticos, pero la modelación teórica se ha referido

solamente a la simulación de los procesos a escala macro, asumiendo los fenómenos moleculares de forma muy esquemática y limitada.

La inmensa mayoría de los estudios se han dedicado a la caracterización empírica de las sustancias y el ajuste a modelos existentes; solamente algunos investigadores se han dedicado a modelar la estructura de los materiales buscando propiedades predeterminadas. En estos casos se ha hecho uso de los métodos de la Dinámica Molecular. No se tienen antecedentes del empleo en investigaciones desarrolladas en el país de los métodos de Monte Carlo Directo con este fin relacionados con la Química y la Ingeniería, aunque sí se han empleado en otros estudios.

En Cuba se han estudiado a profundidad un total de 16 yacimientos de zeolitas naturales, que totalizan unos 20,9 millones de toneladas de recursos medidos, 69,95 millones de toneladas en recursos indicados y 214,61 millones de toneladas en recursos inferidos. Los principales yacimientos se encuentran ubicados en Villa Clara, Camagüey y Holguín [1] (Ver Anexo 1).

1.2.2 Investigaciones Internacionales

Los trabajos realizados de difusión internacionalmente responden a dos vertientes principales. Una de ellas es la modelación de los fenómenos difusivos a partir de la Ley de Fick y sus formulaciones en derivadas parciales, con condiciones iniciales y de frontera concretas, cuya expresión más completa son los trabajos sobre la teoría de Maxwell-Stefan realizados por Krishna y colaboradores. La Ley de Fick lleva a ecuaciones diferenciales de segundo orden para llegar a la solución del problema, si las condiciones de frontera son sencillas. Se ha llegado a la solución de sistemas difusivos de zeolita con defectos de tamaño mesoscópico (Auerbach), sin embargo cuando los defectos son macroscópicos las condiciones de frontera se hacen tan complejas que las soluciones mediante ecuaciones diferenciales se vuelven prohibitivas hasta el momento actual. Otra vertiente es la simulación de los movimientos moleculares a través de los métodos de Monte Carlo Directo, este método se basa en el principio del caminante aleatorio, en el cual se considera que cada elemento del sistema realiza saltos individuales aleatoriamente, interaccionando con los otros elementos del sistema y con el ámbito circundante. Su principal ventaja es que no analiza condiciones fronteras para defectos macroscópicos, sino en la interacción de estos en la estructura. Sin embargo, en todos los trabajos publicados se estudian los sistemas bajo hipótesis simplificadoras que restringen notablemente el alcance de la descripción de la difusión. La más coercitiva de las medidas es el tratamiento bidimensional del movimiento, lo cual impide el análisis de la forma de los canales de las zeolitas. Además, se aborda en todos los casos el

movimiento en sistemas idealmente periódicos, soslayando así la consideración de defectos estructurales superficiales y volumétricos de distinto tipo que pudieran tener influencia en la permeación de los componentes de las mezclas de gases.

Los estudios de la difusión molecular de mezclas en láminas adsorbentes microporosas se justifican entonces con el objetivo de precisar la respuesta de los sistemas percoladores adsorbentes porosos ante la modificación de los elementos de la celda elemental de los cristales, la distribución y orientación de los cristalitos, y la cantidad, distribución y orientación de los espacios intergranulares y de fronteras.

Monte Carlo Directo ofrece ventajas fundamentales claras; al poder dársele un seguimiento sistemático al estudio del fenómeno de la difusión microscópica de mezclas en los microporos. Sin embargo, es necesaria una comparación cuidadosa entre los resultados teóricos y computacionales, y los experimentales.

1.2.3 Softwares vinculados al Campo de Acción

Durante las investigaciones realizadas se encontraron algunos softwares que por su importancia se debe señalar, estos simulan soluciones a distintos aspectos de la realidad aunque no representan una solución total a nuestro problema [2]:

- APBS: representa un paquete de programas informáticos para modelar la disolución biomolecular a través de la solución de la ecuación de Poisson-Boltzmann (PBE). Tiene gran importancia en el estudio de procesos biomoleculares, en cálculo de disoluciones y energía de los enlaces; así como permite evaluar eficientemente las características electrostáticas. Escrito por Nathan Baker tiene gran aplicación en la ciencia biomolecular, pero no tiene en cuenta la difusión en membranas Zeolíticas, siendo esta la principal limitante para el desarrollo del problema.
- Alemán: El programa simula el movimiento de partículas cargadas y los campos magnéticos correspondientes a estas partículas, exhibe la simulación tridimensional en tiempo real. Este software no representa la simulación de moléculas en membranas Zeolíticas, aunque si es un acercamiento visual a la solución.
- ENGRID: Es un software de la generación de acoplamiento con usos de la Dinámica Fluida de Cómputo (CFD); es una de las ramas de los fluidos mecánicos que utiliza métodos y algoritmos

numéricos para solucionar y para analizar los problemas que implican los flujos. Apoya las rejillas prismáticas automáticas de la capa de límite para las ecuaciones de Navier-Stokes (conjunto de ecuaciones en derivadas parciales no lineales que describen el movimiento de un fluido). El software es de referencia y estudio obligatorio en el desarrollo de nuestra solución, ya que simula procesos difusivos aunque no los describe en membranas Zeolíticas por lo que no constituye una solución al problema.

- Open PaperOpt: Plataforma de la simulación de código abierto utilizando Monte Carlo para la simulación del nivel de la partícula de la dispersión luminosa de las estructuras de papel generadas. Este software fue estudiado para el análisis del método de Monte Carlo, pero no constituye una solución.
- Ggrand Random Numbers: Es la programación de una biblioteca para la obtención de números al azar. Previsto para la simulación, de los algoritmos Monte-Carlo. Este fue de gran ayuda en la programación y el uso del algoritmo de Monte Carlos en la solución al problema. Aunque no constituye un software de simulación de procesos difusos.

1.3 Metodología a Utilizar

En la actualidad es un hecho vigente la informatización de la sociedad. Los principales procesos en entidades, empresas y corporaciones son llevadas a cabo por sistemas informáticos. En la elaboración de estos sistemas deben tener en cuenta la calidad, el soporte que se le puede brindar, las consideraciones del cliente y el mantenimiento.

La elección de la metodología adecuada para el desarrollo del proceso brinda una guía en la elaboración del sistema y organiza adecuadamente el desarrollo del proceso, logrando un producto eficiente, de acuerdo con los requerimientos del cliente.

1.3.1 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y expandido por las empresas y compañías en la actualidad, debido a que, está pensado para desarrollarse con todos los métodos de desarrollo, etapas de ciclo de vida, dominios de aplicación y medios.

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de Bases de Datos y componentes de software reutilizables. Permite unir los modelos en paquetes, logrando que equipos muy grandes puedan dividirse en grupos más específicos con un desarrollo común, trabaja con estructuras estáticas y dinámicas, obteniéndose la información referente a las mismas.

Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

El Lenguaje Unificado de Modelado no es un lenguaje formal, es un lenguaje de propósito general que capta los aspectos importantes del sistema a modelar.

Vistas y Diagramas del UML

Las vistas del Lenguaje Unificado de Modelado son un subconjunto en el lenguaje, que modela las construcciones que representan un aspecto del sistema. Los diagramas proporcionan una notación visual para los conceptos de cada vista.

En el nivel superior, las vistas se pueden dividir en tres áreas: clasificación estructural, comportamiento dinámico, y gestión del modelo.

- Clasificación estructural: describe los elementos del sistema y sus relaciones con otros elementos:

Vista Estática

- Diagrama de Clases

Vista de Casos de Uso

- Diagrama de componentes

Vista de Implementación y Despliegue:

- Diagrama de Casos de Usos del Sistema
- Diagrama de Despliegue
- Comportamiento dinámico: describe el comportamiento de un sistema en el tiempo:

Vista de Máquinas de Estado:

- Diagrama de Estados

Vista de Actividad

- Diagrama de Actividad

Vista de Interacción:

- Diagrama de Secuencia
- Diagrama de Colaboración

- Gestión del modelo: describe la organización de los propios modelos en unidades jerárquicas.

Vista de Gestión de Modelo

- Diagrama de Clases [3]

1.3.2 El Proceso Unificado del Rational

El Proceso Unificado de Rational (Rational Unified Process en inglés, RUP) es un proceso de desarrollo de software que contempla cinco principios claves:

- 1- Adaptar al proceso (El proceso deberá adaptarse a las características propias del proyecto u organización)
- 2- Balancear prioridades (Debe encontrarse un balance que satisfaga los diversos requerimientos)
- 3- Demostrar valor iterativamente (En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados)

- 4- Elevar el nivel de abstracción (Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas)
- 5- Enfocarse en la calidad (El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción)

Características de RUP

Iterativo e Incremental:

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Centrado en la Arquitectura:

La arquitectura de un sistema software se representa a través de las diferentes vistas del sistema en construcción (Vista de Casos de Uso, Vista Lógica, Vista de Procesos, Vista de Implementación, Vista de Despliegue).

La arquitectura del software abarca:

- La organización del sistema software.
- Los elementos estructurales que compondrán el sistema y sus interfaces, con sus comportamientos.
- El estilo arquitectónico que guía esta organización.

Guiado por Casos de Uso:

En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). Está compuesto por la fase de Inicio, Elaboración, Construcción y Transición; en las cuales se establece oportunidad y alcance, se identifica las entidades externas o actores con las que se trata e identifica los casos de uso.

Artefactos de RUP

En cada fase RUP propone un grupo de artefactos que ayudan a comprender el análisis y el diseño del sistema, entre los principales artefactos se encuentran:

Fase de Inicio:

- Documento Visión
- Especificación de Requerimientos

Fase de Elaboración:

- Diagramas de caso de uso

Fase de Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:

Vistas del RUP

Vista Lógica:

- Diagrama de clases
- Modelo E-R (Si el sistema así lo requiere)

Vista de Implementación:

- Diagrama de Secuencia
- Diagrama de Estados

- Diagrama de Colaboración

Vista Conceptual:

- Modelo de dominio

Vista física:

- Mapa de comportamiento a nivel de hardware

El Proceso Racional Unificado o RUP (Rational Unified Process), junto con el Lenguaje Unificado de Modelado UML, constituyen la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos [4].

1.3.3 Método de Desarrollo de Sistemas Dinámicos (DSDM)

DSDM nació en enero de 1994 con el objetivo de crear una metodología RAD unificada, aunque su primera versión fue completada en enero de 1995 y publicada en febrero del mismo año.

Es un marco basado originalmente alrededor del desarrollo de aplicaciones rápidas (RAD), apoyado por la implicación continua del usuario, presenta un desarrollo iterativo y acercamiento incremental, que permite la satisfacción a tiempo de los requerimientos del negocio. En DSDM el equipo de desarrollo y los usuarios trabajan juntos lo cual evita que no se cumplan los requerimientos o no funcionen correctamente.

Principios Básicos de DSDM

El Método de desarrollo de sistemas dinámicos está compuesto por nueve principios básicos [5]:

1. Participación del usuario activo.
2. El equipo toma decisiones.
3. Frecuentes entregas del producto.
4. Ajustarse a los objetivos del negocio.
5. Desarrollo iterativo e incremental.
6. Cambios reversibles.
7. Especificar requerimientos globales.

8. Pruebas integradas durante todo el ciclo de vida.
9. Cooperación entre el equipo, usuarios y stakeholders es esencial.

Fases de DSDM

DSDM está compuesto por tres fases y cinco etapas que guían el ciclo de desarrollo del equipo:

Fases:

1. Proyecto previo.
2. Ciclo de vida del proyecto.
3. Proyecto posterior.

Etapas:

1. Estudio de viabilidad
2. Estudio de negocio
3. Iteración funcional del modelo
4. Iteración de diseño y construcción
5. Implementación

Ver Anexo 2.

1.3.4 Programación Extrema (XP)

XP se basa en la programación rápida o extrema, que tiene como requisito para llegar al éxito tener como parte del equipo al usuario final. Es adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.

Características de XP

La metodología de XP se basa en tres características fundamentales:

1. Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de forma tal que puedan hacer pruebas de fallas que pudieran ocurrir en el futuro.
2. Refabricación: se basa en la reutilización de código.

3. Programación en pares: se basa en la programación en pares por parte de los desarrolladores, lo cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Lo fundamental en este tipo de metodología es la comunicación entre los usuarios y los desarrolladores, la simplicidad al desarrollar y codificar los módulos del sistema y la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Durante el desarrollo de XP no se definen tantos artefactos como RUP, ni se desarrolla por casos de uso, se basa mas bien en la solución de las funcionalidades que necesita el cliente, no en la documentación de éstas, aunque si se haga.

1.3.5 Comparación de DSDM con otras metodologías de desarrollo rápido

	CMM	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Características Metodología (CM)								
- Resultados	2	5	5	4	4	4	5	5
- Simplicidad	1	4	4	3	5	3	5	5
- Adaptabilidad	2	5	5	3	3	4	4	3
- Excelencia técnica	4	3	3	4	4	4	3	4
- Prácticas de colaboración	2	5	5	4	3	3	4	5
Media CM	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
Media Total	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

Figura 1. Ranking de Agilidad

1.3.6 Selección de la Metodología de Desarrollo

Durante el estudio realizado de las distintas metodologías de desarrollo rápido o ágil (RAD) y tradicionales se escogieron para el desarrollo del proyecto DSDM y RUP, ya que ambas pueden relacionarse por ser iterativas y por proponer artefactos necesarios para comprender el problema. DSDM define las reglas MoSCow (M: vitales para el proyecto, S: obtener el máximo de beneficio, implementarse si el tiempo lo permite, C: pueden dejarse para otro momento), así como un grupo de principios necesarios en el proyecto. RUP es la metodología tradicional más utilizada y expandida que presenta como principal dificultad, el estar definida para proyectos de gran tamaño por lo que no puede ser aplicada en su totalidad para el desarrollo del presente trabajo, aunque sí, se hace necesario el uso de varios artefactos no definidos por DSDM para una mayor comprensión por parte del cliente.

1.4 Modelación del Espacio Poroso

La creación de una estructura a nivel computacional conlleva a un estudio detallado de las dimensiones que pueden adquirir en su forma natural. En este trabajo se definen estructuras fáciles de modelar y de ubicar en una región del espacio, ejemplo: cilindros y cubos. Cada estructura será modelada a través de tres dimensiones x, y, z y ubicadas en una microcolumna. La microcolumna representa el volumen a través del cual se difunden las moléculas de gas.

1.4.1 Constitución de la Microcolumna

La microcolumna estará conformada por una serie de estratos y los mismos representan una pequeña porción de la microcolumna en aras de dividir la columna en varias partes, con el fin de minimizar el número de moléculas a cantidades manejables, de esta manera, el estrato será representativo de toda la membrana zeolítica; esta simplificación es imprescindible porque ningún dispositivo de hardware es capaz de dar seguimiento a todas las moléculas que pueden estar contenidas en una membrana real. El estrato estará constituido por nueve cristalitas cúbicas del mismo tamaño pero con orientaciones diferentes, y éstas a su vez por trescientas celdas elementales, cada una de las celdas posee 27 000 000 de puntos como máximo. Cada celda contiene la información básica de la estructura cristalina, la cual se puede reproducir a partir de traslaciones periódicas de la celda elemental. En la microcolumna, existen canales mesoporosos cilíndricos (atajos) y zonas impenetrables (barreras) con dimensiones seleccionadas y orientaciones aleatorias.

1.4.2 Selección de los TDA

Un Tipo de dato abstracto (TDA) es un conjunto de datos u objetos al cual se le asocian operaciones. El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en como estén implementadas dichas operaciones. Esto quiere decir que un mismo TDA puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad. Los TDA poseen como principal características el ocultamiento de la información.

Tipo de TDA más comunes:

1. Listas
2. Árboles
3. Tablas hash

Orden de complejidad para las búsquedas:

Tipo de dato	Método	Orden
Listas enlazadas	Buscar	$O(n)$
Árboles	Buscar	$O(\log n)$
Tablas hash	Buscar	$O(1)$ (peor caso $O(n)$)
Listas vectores	Buscar	$O(1)$ conocer la posición a buscar

Orden de complejidad para las inserciones:

Tipo de dato	Método	Orden
Listas enlazadas	Insertar	$O(n)$
Árboles	Insertar	$O(\log n)$
Tablas hash	Insertar	$O(1)$
Listas vectores	Insertar	$O(1)$ conocer la posición a buscar

Un algoritmo será más eficiente que otro siempre que su complejidad sea menor, por estas razones para la construcción de la celda elemental se utilizó la lista de vectores pues se tiene una cantidad predefinida de puntos a agregar (no malgasta ram) y la complejidad temporal para acceder a algún elemento dada la posición es de $O(1)$, para almacenar las moléculas que atravesarán el estrato se emplea la lista doblemente enlazada, con un iterador implementado.

1.4.3 Modelación de las Interacciones

Las moléculas interactúan entre sí de forma completamente elástica y de dos en dos (no existen choques triples ni de orden mayor). A los efectos de simplificar los cálculos, se aproximaron los choques tridimensionales a unidimensionales, asumiendo que las cantidades de movimiento de las moléculas durante el choque se modifican en la misma magnitud que en el unidimensional. Las nuevas direcciones del movimiento se toman entonces aleatoriamente y de forma conjugada.

La validez de esta suposición se sometió a prueba mediante un experimento simulado por métodos de Monte Carlo Directo, en el cual se generaban choques estocásticos valorando la variación de la cantidad de movimiento mediante dos algoritmos y comparándolos estadísticamente. Los choques con las paredes son también elásticos excepto cuando el sitio sea de absorción, en cuyo caso la molécula queda retenida varias iteraciones, en dependencia del tipo de sitio. Cuando el sitio está ocupado, las demás moléculas interactúan de forma elástica. Para modelar la velocidad inicial de las moléculas creadas se tuvo en cuenta las ecuaciones de La ley de distribución de las velocidades moleculares de Maxwell-Boltzmann. Ver Anexo 3

1.5 Definición de un Lenguaje de Programación

1.5.1 Lenguaje de Programación

Una secuencia de posiciones de llaves mecánicas constituía una acción, estas debían desconectarse y conectarse; donde conectada era 1 y desconectada era 0. La sucesión de ceros y unos era una o varias instrucciones en el ordenador en el que se estaba trabajando; por tanto una sucesión de llaves en cualquiera de sus dos posiciones definía una secuencia de unos y ceros. A esta forma primitiva de crear y diseñar programas para un ordenador se le denominó lenguaje máquina o código máquina.

Al existir acciones que eran muy comunes y necesarias surge la necesidad de no tener que programarlas cada vez que se necesiten, así surgen algunas con nombres fáciles de memorizar y asociar: ADD (sumar), SUB (restar), MUL (multiplicar), CALL (Ejecutar subrutina), entre otras. A esta secuencia de posiciones se le denominó "instrucciones", y a este conjunto de instrucciones se le llamó lenguaje ensamblador.

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina.

Aunque muchas veces se usan los términos 'lenguaje de programación' y 'lenguaje informático' como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como, por ejemplo, el HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación).

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje Léxico. Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa.

Los procesadores usados en las computadoras son capaces de entender y actuar según lo indican programas escritos en un lenguaje fijo llamado lenguaje de máquina. Todo programa escrito en otro lenguaje puede ser ejecutado de dos maneras:

- Mediante un programa que va adaptando las instrucciones conforme son encontradas. A este proceso se lo llama interpretar y a los programas que lo hacen se los conoce como intérpretes.
- Traduciendo este programa a su equivalente escrito en lenguaje máquina. A ese proceso se le llama compilar y al programa traductor se le denomina compilador.

1.5.2 Clasificación de los Lenguajes de Programación

Tipos de Clasificaciones:

- Según nivel de abstracción
- Según su forma de ejecución
- Según el paradigma de programación que posea

Según nivel de abstracción:

- Lenguaje Máquina: Lenguaje escrito directamente legible por la máquina.
- Lenguaje de Bajo nivel: Son lenguajes que se acercan al funcionamiento de una computadora.
- Lenguaje de Medio Nivel: Son lenguajes que tienen ciertas características que lo acercan a los lenguajes de bajo nivel pero teniendo al mismo tiempo cualidades que lo hacen un lenguaje mas cercano al humano y, por tanto, de alto nivel.
- Lenguaje de Alto Nivel: son normalmente fáciles de aprender porque están formados por elementos de lenguajes naturales, como el inglés.

Según la forma de Ejecución:

- Lenguaje Compilado: Los compiladores son aquellos cuya función es traducir un programa escrito en un determinado lenguaje a un idioma que la computadora entienda (lenguaje máquina con código binario).
- Lenguajes Interpretados: El intérprete elimina la necesidad de realizar una de compilación después de cada modificación del programa cuando se quiere agregar funciones o corregir errores; pero es obvio que un programa objeto compilado con antelación deberá ejecutarse con mucha mayor rapidez que uno que se debe interpretar a cada paso durante una ejecución del código

Según el paradigma de programación que posea:

Un paradigma de programación representa un enfoque particular o filosofía para la construcción del software. No es mejor uno que otro sino que cada uno tiene ventajas y desventajas.

- El paradigma imperativo o por procedimientos es considerado el más común y está representado, por ejemplo, por el C o por BASIC.
- El paradigma funcional está representado por la familia de lenguajes LISP (en particular Scheme), ML o Haskell.
- El paradigma lógico, un ejemplo es PROLOG.
- El paradigma orientado a objetos. Un lenguaje completamente orientado a objeto es Smalltalk [6].

1.5.3 Selección de un Lenguaje de Programación

Lenguaje de programación C:

C es un lenguaje de propósito general, orientado a la implementación de Sistemas Operativos, concretamente Unix, siendo el lenguaje de programación de sistemas por excelencia. C es muy eficiente en el código que produce, es de alto nivel y permite programar con instrucciones de lenguaje de propósito general.

Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones que permiten un control a muy bajo nivel.

El lenguaje C tiene una gran cantidad de ventajas sobre otros lenguajes y constituyen precisamente la razón fundamental de que después de casi dos décadas de uso siga siendo uno de los lenguajes más populares, utilizados en empresas, organizaciones y fábricas de software de todo el mundo [7].

Ventajas y Características de C:

- Alta velocidad de ejecución.
- Una nueva sintaxis para declarar funciones. Una declaración puede añadir una descripción de los argumentos de la función.
- Asignación de estructuras (registros) y enumeraciones.
- Preprocesador más sofisticado.
- Una nueva definición de la biblioteca que acompaña a C. Entre otras funciones se incluyen: acceso al sistema operativo (por ejemplo, lectura / escritura de archivos), entrada y salida con formato, asignación dinámica de memoria, manejo de cadenas de caracteres. Una colección de cabeceras estándar que proporciona acceso uniforme a las declaraciones de funciones y tipos de datos.
- Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado no llevado al extremo (permitiendo ciertas licencias rupturistas).

Desventajas de C:

- Carece de instrucciones de entrada/salida, de instrucciones para el manejo de cadenas de caracteres, con lo que el trabajo queda para la librería de rutinas, con la consiguiente pérdida de transportabilidad.
- La excesiva libertad en la escritura de los programas puede llevar a errores que en la programación, por ser correctos sintácticamente no se detectan a simple vista.
- Las precedencias de los operadores convierten a veces las expresiones en pequeños rompecabezas.

Lenguaje de programación C++:

C++ es un lenguaje orientado a objetos. Creado para añadirle cualidades y características de las que carecía C. Manteniendo una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción.

C++ es un lenguaje de propósito general basado en el C, al que se han añadido nuevos tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacios de nombres, funciones inline, sobrecarga de operadores, referencias, operadores para manejo de memoria persistente, y algunas utilidades adicionales de librería[8].

Ventajas:

- Mantiene las ventajas de C.
- Mantiene una alta compatibilidad con C, lo que le permite la reutilización del código existente.
- C++ introduce nuevas palabras clave y operadores para manejo de clases.
- Es un lenguaje de propósito general que se adapta a múltiples situaciones.

Principal desventaja:

- Es un lenguaje muy complejo, ya que permite que se pueda trabajar tanto a bajo como a alto nivel, de ahí que sea muy complejo; además realiza un minucioso tratamiento de errores aumentando más su complejidad.

Lenguaje de programación Java:

Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de C y C++. Este último es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido aprendizaje.

Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es una tarea de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria [9].

Ventajas:

- Es robusto.
- Es distribuido
- De arquitectura neutral.
- Es seguro.
- Es Neutral.

Desventajas:

- Dado que la máquina virtual de java es un intérprete y redundante en una falta de rendimiento con relación a aplicaciones equivalentes escritas en código máquina nativo.
- El poder reducir los problemas de acceso a memoria y liberación automática hacen de java un lenguaje poco apropiado para desarrollar aplicaciones de base como Sistemas Operativos.

Para el desarrollo de la aplicación se decidió utilizar el lenguaje de programación C++ debido a la potencialidad que representa, al ser una evolución del C que es uno de los más difundidos actualmente. El C++ permite el manejo de puntero, excepciones y posee una amplia gama de librerías. El manejo de datos es otra de las características que lo hace muy superior ante otros lenguajes. Además de no tener que poseer una máquina virtual y lograr así un mayor rendimiento con aplicaciones semejantes.

Qt

Qt son un conjunto de librerías multi-plataforma para el desarrollo del esqueleto de aplicaciones GUI (Interfaz Gráfica de Usuario), escritas en código C++. Esta completamente orientado a objetos. Con Qt se pueden desarrollar ricas aplicaciones gráficas, incluye soporte de nuevas tecnologías como OpenGL, XML, Bases de Datos, programación para redes, internacionalización, etc. Qt dispone de una amplia gama de herramientas que facilitan la creación de formas, botones y ventanas de diálogo con el uso del ratón. Las aplicaciones creadas con Qt son muy elegantes, se ven y se operan mejor que las aplicaciones nativas. Qt dispone de tres grandes ventajas ante las librerías de ventanas rivales:

- Qt es completamente gratuito para aplicaciones de código abierto.
- Las herramientas, librerías y clases están disponibles para casi todas las plataformas Unix y sus derivados (como Linux, MacOS X y Solaris) como también para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código y la aplicación se verá y actuará mejor que una aplicación nativa.
- Qt tiene una extensa librería con clases y herramientas para la creación de ricas aplicaciones. Estas librerías y clases están bien documentadas, son muy fáciles de usar y tienen una gran herencia de programación orientada a objetos.

Qt Designer: Es una herramienta muy potente que permite diseñar de una forma muy sencilla y rápida ventanas de diálogo con las librerías Qt. Esta herramienta es una aplicación mediante la cual se puede realizar el diseño de aplicaciones GUI de forma gráfica y muy intuitiva.

Qt Creator es un IDE creado por Trolltech, nuevo, ligero y con una plataforma de ambiente de desarrollo integrado (IDE) y diseñada para el desarrollo de aplicaciones con las librerías Qt.

Los sistemas operativos que soporta son [10]:

- GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además hay una versión para Linux con gcc 3.3.
- Mac OS X 10.4 o superior, requiriendo Qt 4.x
- Windows XP y Vista, requiriendo el compilador MinGW y Qt 4.4.3 para MinGW

Para el desarrollo de la aplicación se ha seleccionado las librerías Qt por la gran potencialidad que representa con el lenguaje c++ y poseer una interfaz gráfica muy eficiente. En el año 2008 Nokia adquiere las acciones sobre Trolltech, cambiándose el nombre de Trolltech por Qt Software. Se espera que en el año 2009 se lance una versión definitiva del Qt Creator. Incluye un paquete de ayuda muy completo, además de incluir un qtdemo que ofrece programas con códigos fuentes que sirven de mucha ayuda.

1.6 Sistemas de Gestión de Bases de Datos (SGBD)

Los Sistemas de Gestión de Base de Datos (SGBD - Data Base Management System DBMS) son aplicaciones que permite a los usuarios definir, crear, mantener y proporcionar el acceso controlado a la Base de Datos. Los SGBD contienen una colección de datos interrelacionados y un conjunto de programas para acceder a estos.

El Objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la Base de Datos.

Un Sistema Gestor de Base de Datos debe ser capaz de aceptar las definiciones de datos en versión fuente y convertirlas en la versión objeto. El SGBD debe incluir componentes procesadores para cada uno de los lenguajes de definición de datos. El Gestor debe atender las solicitudes de los usuarios para extraer, actualizar, adicionar o suprimir datos, además de incluir un componente procesador del Lenguaje de manipulación de datos.

El principal objetivo de la implantación de una Base de Datos es poner a disposición de un gran número de usuarios en conjunto integrado de datos. Estos datos podrán ser manipulados por los diferentes usuarios y es ahora cuando se debe garantizar la coherencia de los datos después de las diversas manipulaciones. El SGBD debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el Administrador de la Base de Datos.

Un Sistema de Gestión de Base de Datos se divide en módulos que tratan cada una de las responsabilidades del sistema general. Los componentes funcionales que incluyen son:

- Procesador de Consultas. Traduce sentencias en un lenguaje de consultas a instrucciones de bajo nivel que entiende el Gestor de la Base de Datos.

- Gestor de la Base de Datos: Proporciona la interfaz entre los datos de bajo nivel almacenados en la Base de Datos, los programas de aplicación y las consultas que se hacen en el sistema.
- Gestor de Archivos: Gestiona la asignación de espacio en la memoria del disco y de las estructuras de datos usadas para representar la información almacenada en disco.
- Compilador del Lenguaje de Definición de Datos DDL: Convierte sentencias en DDL en un conjunto de tablas metadatos o “datos sobre datos”.
- Gestor del Diccionario de Datos: Almacena metadatos sobre la estructura de la Base de Datos [11].

La información en los Sistemas Gestores de Base de Datos es representada a través de tuplas, las cuales describen al fenómeno, proceso o ente de la realidad objetiva que se está analizando y se representan a través de tablas. Los SGBD más utilizados en la actualidad son Oracle como número uno, Microsoft SQL Server y PostgreSQL.

Oracle es el primer SGBD a nivel mundial en cuanto a rendimiento, velocidad, estabilidad y seguridad de una Base de Datos. Presenta como inconveniente que no es ni gratuito ni libre, y su licencia es muy cara a nivel mundial. SQL Server es un software producido por Microsoft con licencia para su uso y mono plataforma; hecho y distribuido para Windows. PostgreSQL es una alternativa al software propietario con alto rendimiento y estabilidad en el procesamiento y almacenaje de datos, disminuyendo su velocidad de consultas a medida que aumenta el volumen de datos almacenados con respecto a otros SGBD.

1.6.1 PostgreSQL Sistema de Gestión de Bases de Datos

PostgreSQL es un Sistema de Gestión de Base de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977.

PostgreSQL está considerado como el sistema de Base de Datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

¿Qué características tiene PostgreSQL que lo hace nuestra elección?

PostgreSQL ofrece muchas ventajas para una compañía o negocio, respecto a otros sistemas de Base de Datos, en cuanto a:

1- Instalación ilimitada:

Es frecuente que las Bases de Datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran esto la principal fuente de incumplimiento. Con PostgreSQL, nadie puede demandar por violar estos acuerdos, puesto que no hay costo asociado a la misma.

Esto trae consigo ventajas adicionales como:

- Modelos de negocios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.

2- Ahorros considerables en costos de operación:

- Este software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento.
- Además de esto, los programas de entrenamiento son mucho más reconocidos, costo-efectivos, manejables y prácticos en el mundo real que aquellos de los principales proveedores comerciales.

3- Estabilidad y confiabilidad legendarias:

- En contraste a muchos sistemas de Bases de Datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

4- Extensible:

- El código fuente estará disponible para todos sin costo. Si un equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

5- Multiplataforma:

- PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

6- Diseñado para ambientes de alto volumen:

- PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC (Acceso concurrente multiversión, por sus siglas en inglés) para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de Bases de Datos comerciales usan esta tecnología, por las mismas razones.

7- Herramientas gráficas de diseño y administración de Base de Datos.

Existen varias herramientas gráficas de alta calidad para administrar las Bases de Datos (pgAdmin, pgAccess) y para hacer diseños de Base de Datos (Tora, Data Architect). Además PostgreSQL ofrece una serie de características técnicas, al igual que otros gestores, que permiten un mejor trabajo con las Bases de Datos; entre estas se puede encontrar:

- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de Bases de Datos maestras en múltiples sitios de réplica.
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby.
- Reglas
- Vistas
- Triggers
- Unicode
- Secuencias
- Herencia
- Outer Joins

- Sub-selects
- Una API abierta
- Procedimientos almacenados
- Soporte nativo SSL
- Lenguajes procedurales
- Indices parciales y funcionales
- Soporte para consultas con UNION, UNION ALL y EXCEPT
- Extensiones para SHA1, MD5, XML y otras funcionalidades
- Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL.
- Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario y rápido desarrollo de nuevos tipos.
- Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL.

Herramientas CASE

Las Herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software.

¿Por qué se debería usar herramientas CASE de modelado con UML?

Las herramientas de modelado con UML ofrecen beneficios para todos los que participan en un proyecto, en la medida que los sistemas que se construyen se vuelven complejos. Estas permiten aplicar la metodología de análisis y diseño orientados a objetos, en un nivel donde la arquitectura y el diseño se tornan obvios y fáciles de entender y modificar. En presencia de un proyecto grande, es imprescindible el uso de una herramienta CASE.

Ejemplos de herramientas CASE:

- ASADAL - Herramienta CASE especializada en Sistemas de Tiempo Real.

- System Architect, herramientas CASE para Análisis y Diseño, incluye técnicas estructuradas y orientadas a objetos.
- CRADLE, conjunto de herramientas CASE integradas que dan soporte a la Planificación estratégica, Análisis y Diseño.
- PowerDesigner 7.0: herramienta CASE de Análisis y Diseño incluye capacidades de generación relacional y con orientación a objetos.
- SilverRun: Conjunto integrado de herramientas CASE para el modelado de negocios.
- Rational Rose, herramienta CASE para Análisis y Diseño basándose en el Proceso Unificado de Rational (RUP).
- Visual Paradigm, herramienta CASE para Análisis y Diseño, utiliza el Lenguaje Unificado de Modelado (UML).

1.7 Selección de la herramienta CASE a utilizar

La herramienta CASE seleccionada fue el Visual Paradigm porque brinda un entorno agradable de creación de diagramas para UML, diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad; posibilita el uso de un lenguaje común para todo el equipo de desarrollo facilitando la comunicación, brinda capacidades de ingeniería directa e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, da la disponibilidad de múltiples versiones para cada necesidad y la de integrarse en los principales IDEs y es multiplataforma.

1.8 Conclusiones

En el actual capítulo se realiza un estudio de las tendencias actuales, así como una revisión detallada de lo que ha logrado el mundo en esta materia y en especial Cuba. Además de la selección del lenguaje de programación, el gestor de Base de Datos, las herramientas, las metodologías y tecnologías a utilizar en el análisis y desarrollo de la aplicación propuesta.

CAPITULO II: CARACTERISTICAS DEL SISTEMA

2.1 Introducción

En el actual capítulo se describen las características que presenta el sistema, para esto se verán los procesos del negocio que tienen que ver con el objeto de estudio, a partir del análisis de éstos se puede percibir que debido a la falta de información existente y a la poca estructuración de esos procesos, se necesita definir conceptos que se pueden agrupar en un Modelo de Dominio para capturar correctamente los requisitos y poder construir un sistema eficiente.

La metodología RUP es una guía que permite entender el contexto del problema y definir los procesos del negocio; la metodología DSDM define un grupo de principios a seguir para el desarrollo de un pequeño proyecto, siendo estas las guías para el desarrollo realizado. Se percibe que los procesos están poco definidos y estructurados debido a falta de información existente, por lo que se deben definir conceptos que puedan ser agrupados en un Modelo del Dominio; esto, a su vez, permite capturar más eficientemente los requisitos del sistema, los cuales constituyen la base en la construcción de un sistema eficaz. Los requisitos funcionales y no funcionales delimitan las funciones y características del sistema, mostrando una idea o concepción de lo que se quiere lograr. Se identifican los actores y casos de uso, creándose un diagrama de caso de uso del sistema y las descripciones de los casos de uso, exponiendo las relaciones entre los actores y la secuencia de acciones en el sistema.

Este capítulo guía a los desarrolladores del sistema en la implementación, predefiniendo una arquitectura ágil, que permite trabajar de forma iterativa e incremental, y dar cumplimiento al objetivo general del presente trabajo.

2.2 Modelo del Dominio

En la actualidad la información se necesita cada vez más rápida y confiable, surgiendo como respuesta los sistemas automatizados que simulan eventos o procesos de la realidad. En la petroquímica o centros de investigación de minerales este hecho es aun más evidente, ante la necesidad de disímiles usuarios de obtener la descripción del flujo de un gas determinado, a través de, membranas de minerales en un tiempo relativamente corto y en los medios de cómputo convencionales. Hasta la

actualidad no se tienen referencias de que existan programas que simulen este proceso dada una estructura zeolítica, aunque si existen software que simulan el movimiento de partículas.

Para responder a los objetivos del trabajo se organizaron las actividades por medio de un conjunto de procesos del negocio o modelo del dominio. Estos procesos están sujetos a un conjunto de reglas del Negocio que determinan la estructura y la política a seguir. La información resultante de estos es almacenada una vez que el proceso haya concluido o durante un tiempo determinado solicitado por el usuario autorizado, permitiendo la entrada de nuevos datos, así como el envío y muestra de los mismos.[13]

En el negocio antes estudiado existe un bajo nivel de estructuración, con soluciones muy diversas y dispersas, aunque todas llevan el mismo propósito. Por ello se utilizara un modelo del dominio, ya que permite documentar y visualizar los principales conceptos que se manejan en el dominio del sistema. Esto ayuda a los usuarios, clientes y desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

Para la captura correcta de los requisitos y la construcción de un sistema eficiente se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

2.2.1 Conceptos del Dominio

Entidad: Universidades, Centros de Investigación, Empresas de Proyectos de Ingeniería Química que solicitan el estudio o análisis del gas a través de la membrana de Zeolita.

Especialista: Licenciados en Física, Licenciados en Química, Ingenieros Químicos o personal calificado, que estén autorizados a manipular la muestra que se requiere investigar, estos deben tener conocimientos mínimos de informática y vasto en los análisis físicos a realizar.

Aplicación Informática: Sistema informático utilizado por el especialista para realizar un estudio.

Datos: Conjunto organizado e integrado de información almacenada y clasificada en computadora o papel para su posterior consulta.

Formato Digital: Datos almacenados en formato digital.

Formato Duro: Datos impresos en papel.

2.2.2 Eventos del Dominio

Solicita: proceso mediante el cual el especialista busca la información necesaria para dar respuesta a una petición dada del usuario.

Configura: proceso mediante el cual el especialista transforma una aplicación informática para dar respuesta a una petición de información.

Almacena: proceso mediante el cual la aplicación almacena la información del resultado de la petición del usuario.

2.2.3 Diagrama de Clases del Dominio

El modelo de Dominio o Modelo Conceptual del Dominio representa conceptos del mundo real. Es descrito mediante un diagrama de clases conceptuales significativas en el dominio del problema y es representado mediante diagramas UML.

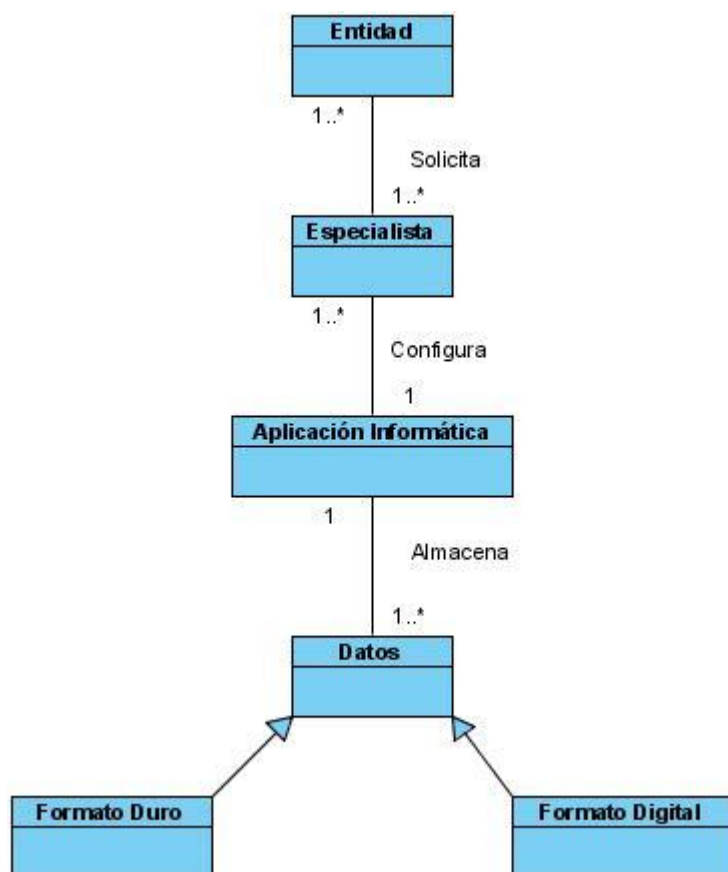


Figura 2. Modelo de Dominio.

2.3 Propuesta del sistema

Se implementará una aplicación desktop para la simulación de los procesos de difusión molecular en membranas policristalinas de Zeolitas en estratos no perfectos, que tendrá que acoplarse a un sistema generador de estructuras zeolíticas para su funcionamiento. Este proceso se llevará a cabo de manera eficiente y rápida, se utilizarán tipos de datos abstractos en la modelación y para el almacenamiento de los Datos Persistentes el gestor de Base de Datos Postgres. La aplicación presentará una interfaz amigable para facilitar el trabajo de los usuarios. Esto permitirá que los usuarios sólo necesiten tener un conocimiento básico para la manipulación de la aplicación desde el punto de vista informático, pues para realizar los estudios si deben tener un conocimiento de la materia en cuestión.

El usuario debe autenticarse en la Base de Datos, no solo para aumentar la seguridad, sino para almacenar las pruebas hechas y sus resultados, con lo que se identifica al personal implicado en cada

simulación o estudio realizado. El sistema debe permitir escoger que tipo de gases en las mezclas se utilizará. Las características físicas de los minerales serán almacenados en la Base de Datos, reproduciendo el sistema cada molécula del gas dentro de un estrato que tendrá forma aproximadamente cúbica. El movimiento será simulado por un algoritmo pseudoaleatorio, partiendo de velocidades de magnitud y dirección estocásticas generadas aleatoriamente mediante la distribución de Maxwell-Boltzmann. A continuación el movimiento de las moléculas continuará de forma a determinista considerando las mismas como cuerpos rígidos que chocan elásticamente entre sí y con las paredes del recinto, aunque la excentricidad de los choques se simula también aleatoriamente.

Durante el transcurso del proceso las moléculas que logren atravesar el estrato serán almacenadas para ponerlas a disposición del usuario, y durante la ejecución las que permanezcan estarán en tipos de datos abstractos para su uso. Los informes de las pruebas serán generados una vez concluido el proceso o interrumpido por el usuario.

La aplicación será desarrollada en la plataforma Linux, ya que es una prioridad para Cuba emigrar a un software sin restricciones, como lenguaje de programación será utilizado el C++, lenguaje muy potente en el procesamiento de datos.

2.4 Requerimientos

2.4.1 Requerimientos Funcionales.

Los Requerimientos funcionales especifican acciones que la aplicación debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por lo general se describen mejor a través del modelo de Casos de uso y las descripciones de los Casos de uso. Por lo tanto los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema. [12]

R1 Autenticarse

- 1.1 Confirmar Usuario y contraseña
- 1.2 Bloquear Acceso

R2 Gestionar Usuario

- 2.1 Adicionar Usuario

- 2.2 Modificar Usuario
- 2.3 Eliminar Usuario
- 2.4 Mostrar Datos de Usuario

R3 Gestionar Zeolita

- 3.1 Adicionar Zeolita
- 3.2 Modificar Zeolita
- 3.3 Eliminar Zeolita

- 3.4 Mostrar Características de la Zeolita

R4 Gestionar Gas

- 4.1 Adicionar Gas
- 4.2 Modificar Gas
- 4.3 Eliminar Gas
- 4.4 Mostrar Gases

R5 Gestionar mezcla de gases

- 5.1 Relacionar moléculas de la mezcla
- 5.2 Adicionar Mezcla
- 5.3 Modificar Mezcla
- 5.4 Eliminar Mezcla
- 5.5 Mostrar Mezclas

R6 Gestionar Estudio

- 6.1 Crear Estudio
- 6.2 Eliminar Estudio
- 6.3 Mostrar Estudio

R7 Gestionar Atajos

- 7.1 Adicionar Atajos

7.2 Eliminar Atajos

7.3 Mostar Atajos

R8 Gestionar Barreras

8.1 Adicionar Barreras

8.2 Eliminar Barreras

8.3 Mostar Barreras

R9 Detener estudio

9.1 Registrar motivo de la interrupción

R10 Almacenar Temporalmente Moléculas de Gas

R11 Guardar Resultados

11.1 Guardar en Formato Digital

11.1 Guardar en Formato Duro

R12 Entregar Resultados

12.1 Imprimir Resultados

2.4.2 Requerimientos No Funcionales

Un requisito no funcional describe no sólo lo que el software hará, sino como lo hará, por ejemplo: de rendimiento. Los requisitos no funcionales son difíciles de verificar, y por ello son evaluados subjetivamente. Son características del sistema, suelen ser más críticos que los requisitos funcionales y de no cumplirse el software puede que no funcione o no sea aceptado por el cliente.

Apariencia o interfaz externa:

La interfaz del sistema debe ser amigable.

El entorno de simulación tendrá colores fuertes, fácilmente identificables a la vista.

Usabilidad:

El sistema deberá estar siempre disponible.

El análisis de una muestra no debe exceder las 24 horas.

El sistema podrá ser usado por personas con conocimiento básicos de informática.

Soporte:

El sistema necesita un servidor de Base de Datos que soporte grandes volúmenes de datos.

El servidor de Base de Datos deberá tener alta Velocidad de Procesamiento de Datos.

Los tiempos de respuesta en el procesamiento de la información deberán ser mínimos.

Portabilidad:

El sistema debe ser desarrollado para Linux en una primera fase.

Seguridad:

El usuario deberá autenticarse. (Contraseña de acceso).

A las funcionalidades del sistema solo tendrán acceso los usuarios autenticados.

Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos (Creación de sesiones).

Externo:

El sistema no revelará información personal acerca de los clientes a usuarios que no tienen acceso al mismo.

Hardware:

Procesador de dos núcleos o superior.

Memoria RAM 512Gb.

Capacidad de Almacenaje de 80Gb.

Tarjeta de red.

Software:

Sistema operativo Linux.

Gestor de Base de Datos PostgreSQL.

2.5 Modelo del Sistema

Un sistema representa un conjunto de funciones bien definidas. Constituye una guía a seguir.

El modelo de sistema es la plataforma sobre la cual se despliegan componentes o partes que se relacionan entre sí. Se describe a través de diagramas que definen las características propias del sistema, pudieran desglosarse en diagramas de flujo de datos. Se representan los subsistemas y módulos por los que esta compuesto el sistema. Permitiendo así una mayor comprensión del sistema por los analistas.

2.5.1 Actores del Sistema

Los actores del sistema son agentes externos que interactúan con él, solicitando una funcionalidad. Estos no constituyen parte del sistema aunque si guardan relación con el sistema y además se benefician de él. Un actor puede estar representado por otros sistemas que interactúan externamente o por clientes que demanden una funcionalidad.

Actor	Descripción
Administrador	Representa a un agente externo con acceso total a la aplicación.

Actor	Descripción
Especialista	Representa a un agente externo con acceso limitado al módulo de usuario de la aplicación.

2.5.2 Caso de Uso

Los casos de uso son artefactos que describen las interacciones que se establecen entre los actores y el sistema. Las interacciones son descritas a través de acciones y reacciones desde el punto de vista

del usuario. Los casos de usos muestran varios escenarios donde un usuario o sistema pudiera interactuar con el sistema a desarrollar. La representación se realiza a través de diagramas que muestran la respuesta a eventos del sistema identificados en los requerimientos del sistema.

2.5.3 Descripciones no expandidas de los casos de uso.

CU-1	Controlar Acceso
Actor	Usuario
Descripción	El caso de uso se inicia cuando el usuario llena los campos Identificador y Contraseña para su respectiva autenticación, de esta forma se limita el acceso de los usuarios al sistema. El caso de uso termina cuando el usuario cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R1

CU-2	Gestionar Usuario
Actor	Administrador
Descripción	El caso de uso se inicia cuando el administrador desea actualizar los datos del usuario. El sistema muestra los usuarios registrados. El administrador selecciona el usuario que desea modificar, además de poder eliminarlo y/o adicionar algún otro. El caso de uso termina cuando el administrador cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R2

CU-3	Gestionar Zeolita
Actor	Usuario
Descripción	El caso de uso se inicia cuando el usuario desea actualizar los datos de la zeolita. El sistema muestra las zeolitas registradas. El usuario selecciona la zeolita que desea modificar, además de poder eliminarla y/o adicionar alguna otra. El caso de uso termina cuando el usuario cierra el sistema o

CAPITULO II: CARACTERISTICAS DEL SISTEMA

	accede a otras opciones de la aplicación.
Referencia	R3

CU-4	Gestionar Gas
Actor	Especialista
Descripción	El caso de uso se inicia cuando el usuario desea actualizar los datos del gas. El sistema muestra los gases registrados. El usuario selecciona el gas que quiere modificar, además de poder eliminarlo y/o adicionar algún otro. El caso de uso termina cuando el usuario cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R4

CU-5	Gestionar Mezcla Gas
Actor	Especialista
Descripción	El caso de uso se inicia cuando el especialista desea actualizar los datos de la mezcla de gas. El sistema muestra las mezclas registradas en el sistema. El especialista selecciona la mezcla de gas que quiere modificar, además de poder eliminarla y/o adicionar alguna otra, así como relacionar las moléculas de la mezcla. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Referencia	R5

CU-6	Gestionar Estudio
Actor	Especialista
Descripción	El caso de uso se inicia cuando el especialista desea actualizar los datos de un estudio. El sistema permite mostrar o eliminar los estudios realizados. EL especialista selecciona la mezcla y la zeolita con que piensa realizar el

CAPITULO II: CARACTERISTICAS DEL SISTEMA

	estudio. El caso de uso termina cuando el especialista comienza la ejecución del estudio.
Referencia	R6,R10

CU-7	Gestionar Barrera
Actor	Especialista
Descripción	El caso de uso se inicia cuando el sistema muestra las opciones para adicionar las características de una o varias barreras. El sistema permite eliminar o mostrar las barreras relacionadas a un estudio determinado. El caso de uso finaliza cuando el especialista accede a otras opciones del sistema, que pueden ser ir al menú principal o ejecutar un estudio.
Referencia	R7

CU-8	Gestionar Atajo
Actor	Especialista
Descripción	El caso de uso se inicia cuando el sistema muestra las opciones para adicionar las características de uno o varios atajos. El sistema permite eliminar o mostrar los atajos relacionados a un estudio determinado. El caso de uso finaliza cuando el especialista accede a otras opciones del sistema, que pueden ser ir al menú principal o ejecutar un estudio.
Referencia	R8

CU-9	Detener Estudio
Actor	Especialista
Descripción	El caso de uso se inicia cuando el especialista solicita la interrupción del estudio. El caso de uso permite registrar el motivo y la fecha de la interrupción, así como eliminar la interrupción de un estudio determinado. El

	caso de uso finaliza cuando el usuario registra la interrupción en el sistema.
Referencia	R9

CU-10	Guardar Resultado
Actor	Especialista
Descripción	El caso de uso se inicia cuando el especialista desea guardar los resultados parciales o totales de la corrida del estudio. El caso de uso permite también eliminar o mostrar un resultado dado un estudio determinado. El caso de uso finaliza cuando se accede a otras opciones del sistema.
Referencia	R11

CU-11	Entregar Resultado
Actor	Especialista
Descripción	El caso de uso se inicia cuando le solicitan al especialista los resultados del estudio. El caso de uso permite entregar los resultados impresos o en memoria. El caso de uso finaliza cuando el especialista accede a otras opciones del sistema.
Referencia	R12

2.5.4 Diagrama de Caso de Uso del Sistema

Un diagrama de casos de uso del sistema representa gráficamente a los actores y su interacción con los distintos procesos del sistema. Cada caso de uso debe comunicarse con al menos un actor, si no aparece ningún actor que se comunique con un caso de uso esto indica error en el modelo de caso de uso o en los requerimientos planteados.

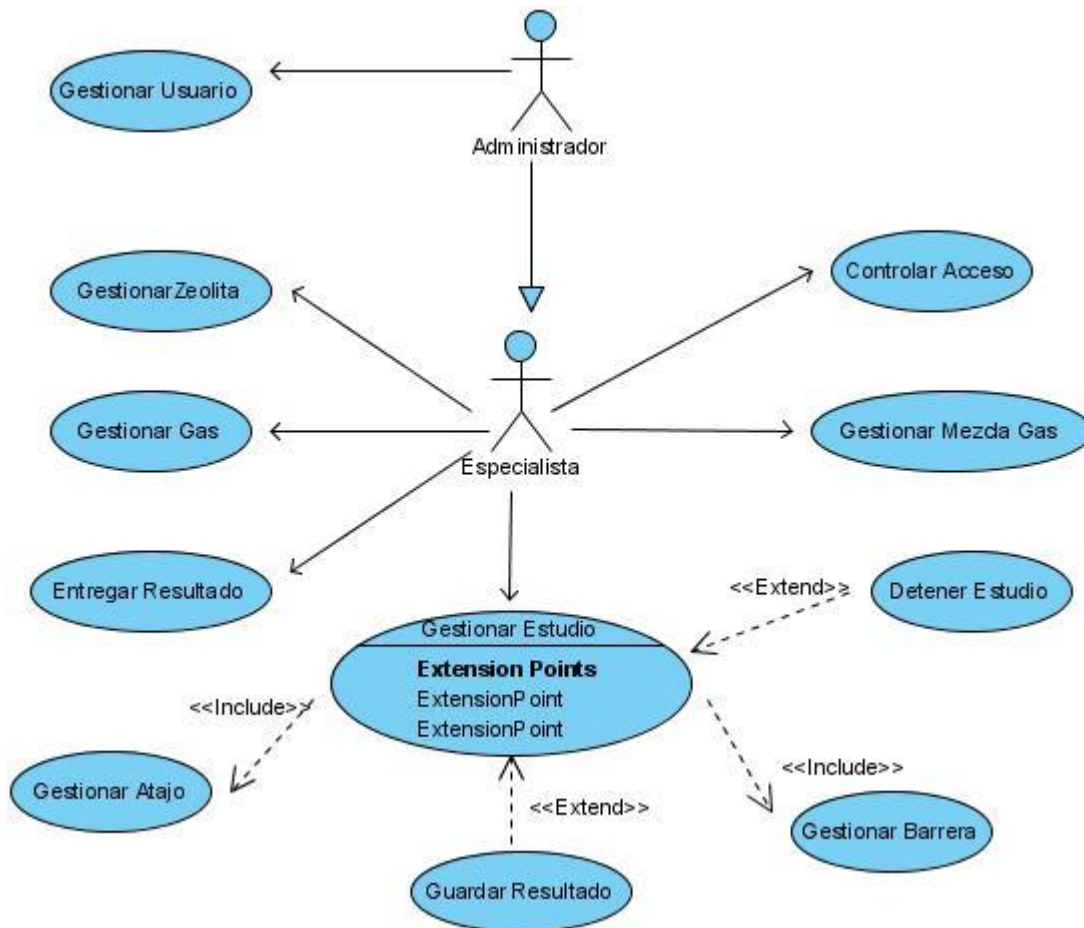


Figura 3. Diagrama de Casos de Uso.

2.6 Descripción de los Casos de Uso del Sistema

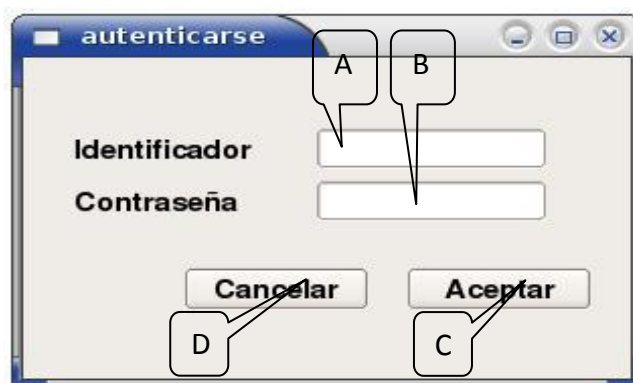
La descripción de los casos de uso del sistema permite comprender la interacción entre el sistema y los usuarios.

2.6.1 Descripción Caso de Uso Controlar Acceso

Nombre del Caso de uso:	Controlar Acceso
Actores:	Especialista
Propósito:	Limitar el acceso al sistema a usuarios registrados.
Resumen	El caso de uso se inicia cuando el especialista llena los

	campos Identificador y Contraseña para su respectiva autenticación, de esta forma se limita el acceso de los usuarios al sistema. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser usuario registrado en el sistema.
Tipo:	Real.
Referencia:	R1
Casos de uso relacionados:	

Interfaz 1



- A: nombre: identificador, tipo: varchar** (text para insertar el identificador de usuario asignado)
- B: nombre: contraseña, tipo: varchar** (text para insertar la clave de acceso al sistema del usuario)
- C: nombre: Aceptar** (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)
- D: nombre: Cancelar** (botón para indicar al sistema que regrese a las opciones iniciales)

Curso Normal de Eventos

Acción del actor	Respuesta del sistema
2: Introduce los datos	1: Muestra Interfaz 1.

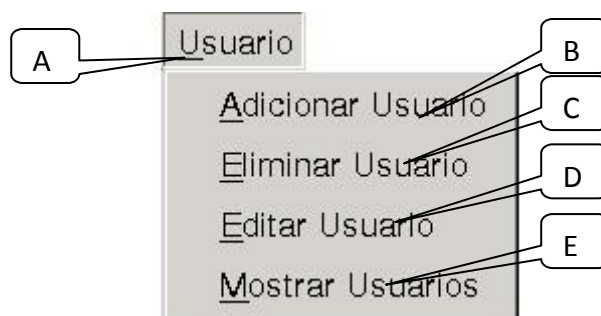
<p>solicitados por el sistema.</p> <p>3: Presiona el botón Aceptar.</p>	<p>4- Chequea si el especialista ha dejado algún campo, en los datos a llenar, vacío.</p> <p>5- Comprueba que el Identificador y la Contraseña del especialista existan y coincidan.</p> <p>6- Muestra mensaje: “El especialista fue registrado correctamente”.</p>
Cursos Alternos	
<p>Línea 3: Presiona el botón Cancelar.</p>	<p>3.1 La aplicación se cierra.</p>
	<p>Línea 4: Muestra mensaje: “Por favor llene todos los campos.”</p> <p>4.1 Regresa a la línea 1.</p>
	<p>Línea 5: Muestra mensaje: “Por favor verifique su Identificador o Contraseña.”</p> <p>5.1 Regresa a la línea 1.</p>
<p>Poscondiciones:</p>	<p>El sistema muestra las opciones, brinda el acceso a las funcionalidades del mismo al especialista.</p>

2.6.2 Descripción Caso de Uso Gestionar Usuario

Nombre del Caso de uso:	Gestionar Usuario
Actores:	Administrador
Propósito:	Adicionar, Eliminar, Modificar usuarios que trabajarán directamente con el sistema, así como, Mostrar los datos de los mismos.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción Usuario. El administrador puede

	seleccionar la acción de Adicionar, Eliminar, Editar o Mostrar usuarios. El caso de uso termina cuando el administrador cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser administrador registrado en el sistema.
Tipo:	Real.
Referencia:	R2
Casos de uso relacionados:	

Interfaz 2



- A:** menú desplegable que muestra las diferentes opciones del usuario.
- B:** indica la opción de adicionar un usuario al sistema.
- C:** indica la opción de eliminar al usuario del sistema.
- D:** indica la opción de editar usuario.
- E:** indica la opción de mostrar todos los usuarios registrados en el sistema.

Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1- Selecciona en el menú principal la opción "Usuario".(2-A) 3- Selecciona la opción:	2- Muestra menú con las opciones "Adicionar Usuario", "Eliminar Usuario", "Editar Usuario" y "Mostrar Usuarios".

<p>“Adicionar Usuario”.</p> <p>Sección 1.</p> <p>“Eliminar Usuario”.</p> <p>Sección 2.</p> <p>“Modificar Usuario”.</p> <p>Sección 3.</p> <p>“Mostrar Usuarios”.</p> <p>Sección 4.</p>	
---	--

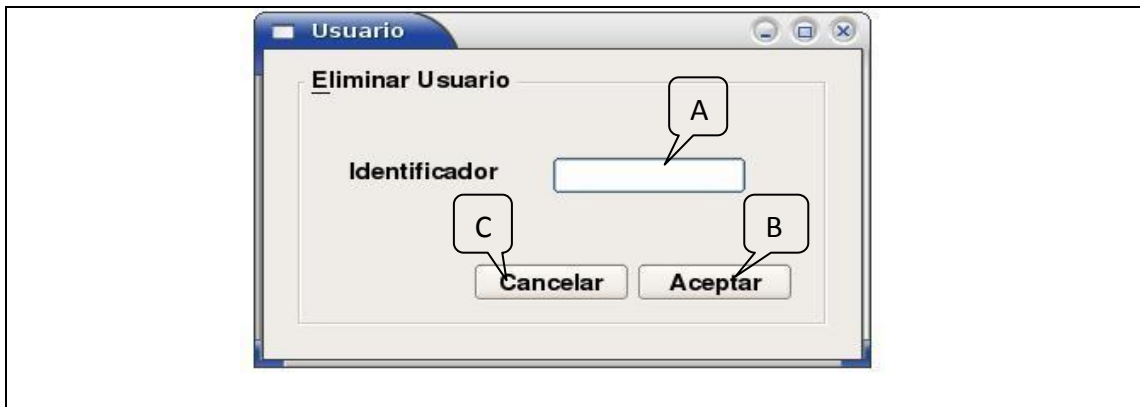
Sección 1: Adicionar Usuario

Interfaz 3



- A: nombre: identificador, tipo: varchar** (text para insertar el identificador del usuario)
- B: nombre: nombre, tipo: varchar** (text para insertar el nombre del usuario)
- C: nombre: apellido, tipo: varchar** (text para insertar el apellido del usuario)
- D: nombre: tipo_usuario, tipo: varchar** (text para escoger si es administrador o especialista)
- E: nombre: Provincia, tipo: varchar** (text para insertar la dirección del usuario)
- F: nombre: Teléfono, tipo: int** (text para insertar el teléfono del usuario)
- G: nombre: Especialidad, tipo: varchar** (text para insertar la especialidad del usuario)
- H: nombre: Carnet_identidad, tipo: varchar** (text para insertar número de carnet de identidad del usuario)

<p>I: contraseña: Usuario, tipo: varchar (text para insertar la contraseña del usuario)</p> <p>J: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)</p> <p>K: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Aceptar.</p>	<p>3- Muestra Interfaz 3.</p> <p>6- Chequea si el administrador ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Comprueba que solo se haya introducido entero en el campo de número de carnet de identidad y que no se exceda de 11 caracteres.</p> <p>8- Comprueba que solo se haya introducido números en el campo del teléfono y que no se exceda de 15 caracteres.</p> <p>9- Chequea que el identificador del usuario no esté repetido.</p> <p>10- Chequea que el número de carnet del usuario no esté repetido.</p> <p>11- Registra al nuevo usuario creado en el sistema.</p> <p>12- Muestra mensaje: "El usuario fue adicionado correctamente".</p>
<p>Sección 2: Eliminar Usuario</p>	
<p>Interfaz 4</p>	



A: nombre: identificador, tipo: varchar (text para insertar el identificador de del usuario)

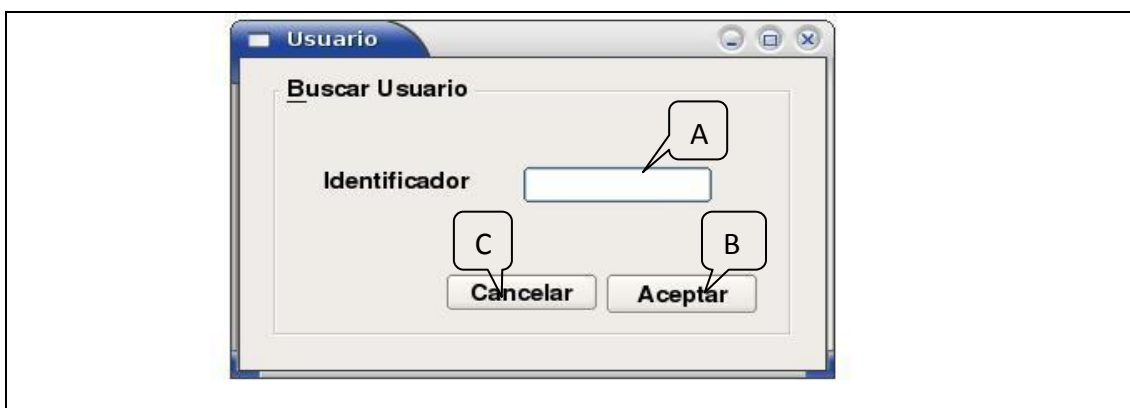
B: nombre: Eliminar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)

C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Eliminar.</p>	<p>3- Muestra Interfaz 4.</p> <p>6- Chequea si el usuario ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Chequea que el identificador del usuario exista.</p> <p>8- Muestra mensaje: “El usuario “id_usuario” fue correctamente eliminado”.</p>
---	--

Sección 3: Modificar Usuario

Interfaz 5



A: nombre: identificador, tipo: varchar (text para insertar el identificador de del usuario)

B: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)

C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

Interfaz 6



A: nombre: identificador, tipo: varchar (text para insertar el identificador de del usuario)

B: nombre: nombre, tipo: varchar (text para insertar el nombre del usuario)

C: nombre: apellido, tipo: varchar (text para insertar el apellido del usuario)

D: nombre: tipo_usuario, tipo: varchar (text para escoger si es administrador o especialista)

E: nombre: Provincia, tipo: varchar (text para insertar la dirección del usuario)

<p>F: nombre: Teléfono, tipo: int (text para insertar el teléfono del usuario)</p> <p>G: nombre: Especialidad, tipo: varchar (text para insertar la especialidad del usuario)</p> <p>H: nombre: Carnet_identidad, tipo: varchar (text para insertar número de carnet de identidad del usuario)</p> <p>I: contraseña: Usuario, tipo: varchar (text para insertar la contraseña del usuario)</p> <p>J: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)</p> <p>K: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Buscar.</p> <p>9- Cambia los datos necesarios del usuario.</p> <p>10- Presiona el botón Aceptar.</p>	<p>3- Muestra Interfaz 5.</p> <p>6- Chequea que no existan campos vacíos.</p> <p>7- Chequea que el usuario este registrado.</p> <p>8- Muestra Interfaz 6.</p> <p>11- Chequea que no existan campos vacíos.</p> <p>12- Muestra mensaje: "El usuario fue actualizado".</p>
<p>Sección 4: Mostrar Usuarios</p>	
<p>Interfaz 7</p>	

Id	Tipo Usuario	Especialidad	Nombre	Apellido	Provincia	Teléfono	# carnet	
1	btorresg	Administra...	Ing. Química	Belkis	Torres	Santiago	653042	4556898712
2	Isoto	Especialista	Lic. Física	Leonardo	Soto	Camaguey	263229	8510021634
3	1	Especialista	Ing. Química	I	I	I	3456	1234567890
4	dagmar	Administra...	Ing. Química	Dagmar	Fernandez	Santiago	234567	1245781245

A: nombre: Salir (botón para indicar al sistema la conformidad con los datos y regresar al menú principal)

<p>4- Presiona el botón Salir.</p>	<p>3- Muestra Interfaz 7.</p> <p>5- Regresa a las opciones de la aplicación.</p>
------------------------------------	--

Cursos Alternos

Sección 1:	
<p>Línea 5: Presiona el botón Cancelar.</p>	<p>5.1: Regresa a las opciones de la aplicación.</p>
	<p>Línea 6: Muestra mensaje: "Por favor llene todos los campos".</p> <p>6.1 Regresa a la línea 3.</p>
	<p>Línea 7: Muestra mensaje: "El número de</p>

	<p>carnet debe ser un entero y tener 11 caracteres”.</p> <p>7.1 Regresa a la línea 3.</p>
	<p>Línea 8: Muestra mensaje: “El número de teléfono debe ser un número entero y tener menos de 15 caracteres.”</p> <p>8.1 Regresa a la línea 3.</p>
	<p>Línea 9: Muestra mensaje: “El identificador del usuario ya existe en el sistema, debe cambiarlo.”</p> <p>9.1 Regresa a la línea 3.</p>
	<p>Línea 10: Muestra mensaje: “Existe un usuario con igual número de carnet, debe cambiarlo.”</p> <p>10.1 Regresa a la línea 3.</p>
Sección 2:	
<p>Línea 5: Selecciona el botón Cancelar.</p>	<p>5.1 Regresa a las opciones del sistema.</p>
	<p>Línea 6: Muestra mensaje “Por favor llene todos los campos.”</p> <p>6.1 Regresa a la línea 3.</p>
	<p>Línea 7: Muestra mensaje: “El usuario no está registrado en el sistema”.</p> <p>7.1 Regresa a la línea 3.</p>
Sección 3:	
<p>Línea 5: Presiona el botón Cancelar.</p>	<p>5.1 Regresa a las opciones del sistema.</p>
	<p>Línea 6: Muestra mensaje “Por favor llene todos los campos”.</p> <p>6.1 Regresa a la línea 3.</p>

	<p>Línea 7: Muestra mensaje: “El usuario no está registrado en el sistema”.</p> <p>7.1 Regresa a la línea 3.</p>
<p>Línea 10: Presiona el botón Cancelar.</p>	<p>10.1 Regresa a las opciones del sistema.</p>
	<p>Línea 11: Muestra mensaje “Por favor llene todos los campos”.</p> <p>11.1 Regresa a la línea 8.</p>
<p>Poscondiciones:</p>	<p>El usuario es registrado en el sistema.</p>

2.6.3 Descripción Caso de Uso Gestionar Zeolita

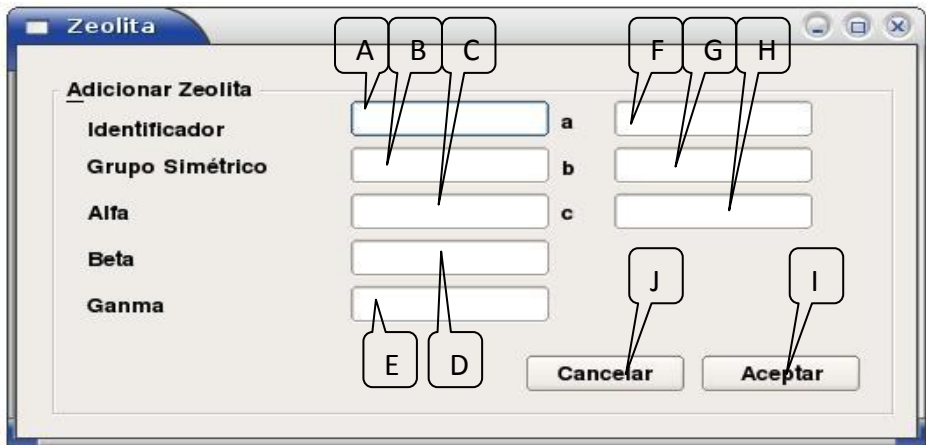
Nombre del Caso de uso:	Gestionar Zeolita
Actores:	Especialista
Propósito:	Adicionar, Eliminar, Modificar las zeolitas que se utilizarán en los estudios en el sistema, así como, Mostrar los datos de las mismas.
Resumen	El caso de uso se inicia cuando el especialista selecciona la opción Zeolita. El especialista puede seleccionar la acción de Adicionar, Eliminar, Editar o Mostrar zeolitas. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser especialista registrado en el sistema.
Tipo:	Real.
Referencia:	R3
Casos de uso relacionados:	

Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
<p>1- Selecciona en el menú principal la opción "Zeolita". (8-A)</p> <p>3- Selecciona la opción: "Adicionar Zeolita". Sección 1. "Eliminar Zeolita". Sección 2. "Editar Zeolita". Sección 3. "Mostrar Zeolita". Sección 4.</p>	<p>2- Muestra menú con las opciones "Adicionar Zeolita", "Eliminar Zeolita", "Editar Zeolita" y "Mostrar Zeolita".</p>

Interfaz 8



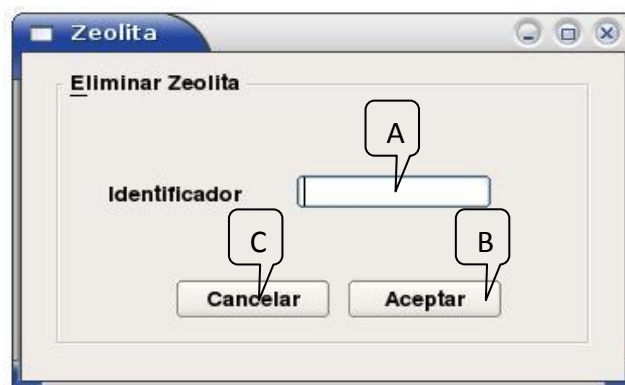
- A:** menú desplegable que muestra las diferentes opciones de la zeolita.
- B:** indica la opción de adicionar una zeolita al sistema.
- C:** indica la opción de eliminar la zeolita del sistema.
- D:** indica la opción de editar la zeolita.
- E:** indica la opción de mostrar todas las zeolitas registradas en el sistema.

Sección 1: Adicionar Zeolita	
<p>Interfaz 9</p>  <p>The screenshot shows a window titled 'Zeolita' with a sub-header 'Adicionar Zeolita'. It contains several input fields and two buttons. Callouts A, B, and C point to the 'Identificador', 'Grupo Simétrico', and 'Alfa' fields respectively. Callouts D and E point to the 'Beta' and 'Gamma' fields. Callouts F, G, and H point to the 'a', 'b', and 'c' parameters. Callout J points to the 'Cancelar' button and callout I points to the 'Aceptar' button.</p>	
<p>A: nombre: identificador, tipo: varchar (text para insertar el identificador de de la zeolita)</p> <p>B: nombre: Grupo Simétrico, tipo: varchar (text para insertar el grupo simétrico de la zeolita)</p> <p>C: nombre: Alfa, tipo: integer (text para insertar el ángulo alfa)</p> <p>D: nombre: Beta, tipo: integer (text para insertar el ángulo beta)</p> <p>E: nombre: Gamma, tipo: integer (text insertar el ángulo ganma)</p> <p>F: nombre: a, tipo: int (text para insertar el parámetro a de la celda estructural de la zeolita)</p> <p>G: nombre: b, tipo: varchar (text para insertar el parámetro b de la celda estructural de la zeolita)</p> <p>H: nombre: c, tipo: varchar (text para insertar el parámetro c de la celda estructural de la zeolita)</p> <p>I: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)</p> <p>J: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón</p>	<p>3- Muestra Interfaz 9.</p>

<p>Aceptar.</p>	<p>6- Chequea si el usuario ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Comprueba que solo se haya introducido valores enteros en los campos de Alfa, Beta, Gamma.</p> <p>8- Comprueba que solo se haya introducido valores float en los campos de a, b, c.</p> <p>9- Chequea que el identificador de la zeolita no esté repetido.</p> <p>10- Registra la nueva zeolita creada en el sistema.</p> <p>11- Muestra mensaje: “La zeolita fue adicionada correctamente”.</p>
-----------------	--

Sección 2: Eliminar Zeolita

Interfaz 10



A: nombre: identificador, tipo: varchar (text para insertar el identificador de la zeolita)

B: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)

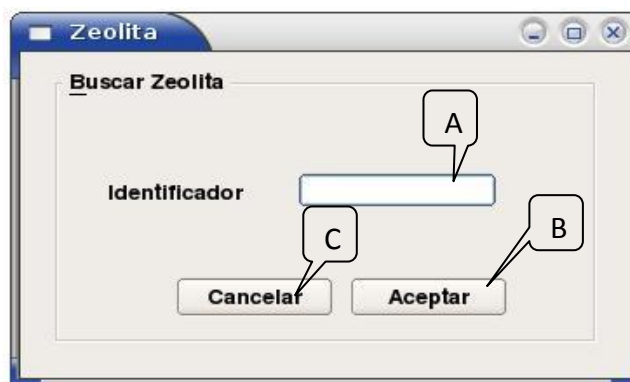
C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

<p>4- Introduce los datos</p>	<p>3- Muestra Interfaz 10.</p>
-------------------------------	--------------------------------

<p>solicitados por el sistema.</p> <p>5- Presiona el botón Eliminar.</p>	<p>6- Chequea si el usuario ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Chequea que el identificador de la zeolita exista.</p> <p>8- Muestra mensaje: “La zeolita “id_zeolita” fue correctamente eliminada”.</p>
--	--

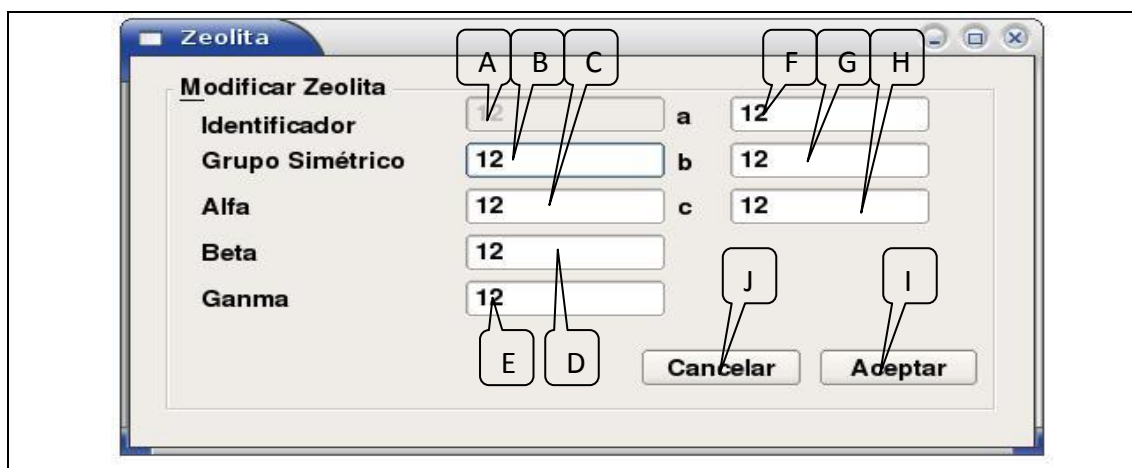
Sección 3: Editar Zeolita

Interfaz 11



- A: nombre: identificador, tipo: varchar** (text para insertar el identificador de la zeolita)
- B: nombre: Aceptar** (botón para indicar a la aplicación la conformidad con los datos y dar curso a las acciones)
- C: nombre: Cancelar** (botón para indicar al sistema que regrese a las opciones iniciales)

Interfaz 12



A: nombre: identificador, tipo: varchar (text para ver el identificador de de la zeolita)

B: nombre: Grupo Simétrico, tipo: varchar (text para insertar el grupo simétrico de la zeolita)

C: nombre: Alfa, tipo: integer (text para insertar el ángulo alfa)

D: nombre: Beta, tipo: integer (text para insertar el ángulo beta)

E: nombre: Ganma, tipo: integer (text insertar el ángulo ganma)

F: nombre: a, tipo: int (text para insertar el parámetro a de la celda estructural de la zeolita)

G: nombre: b, tipo: varchar (text para insertar el parámetro b de la celda estructural de la zeolita)

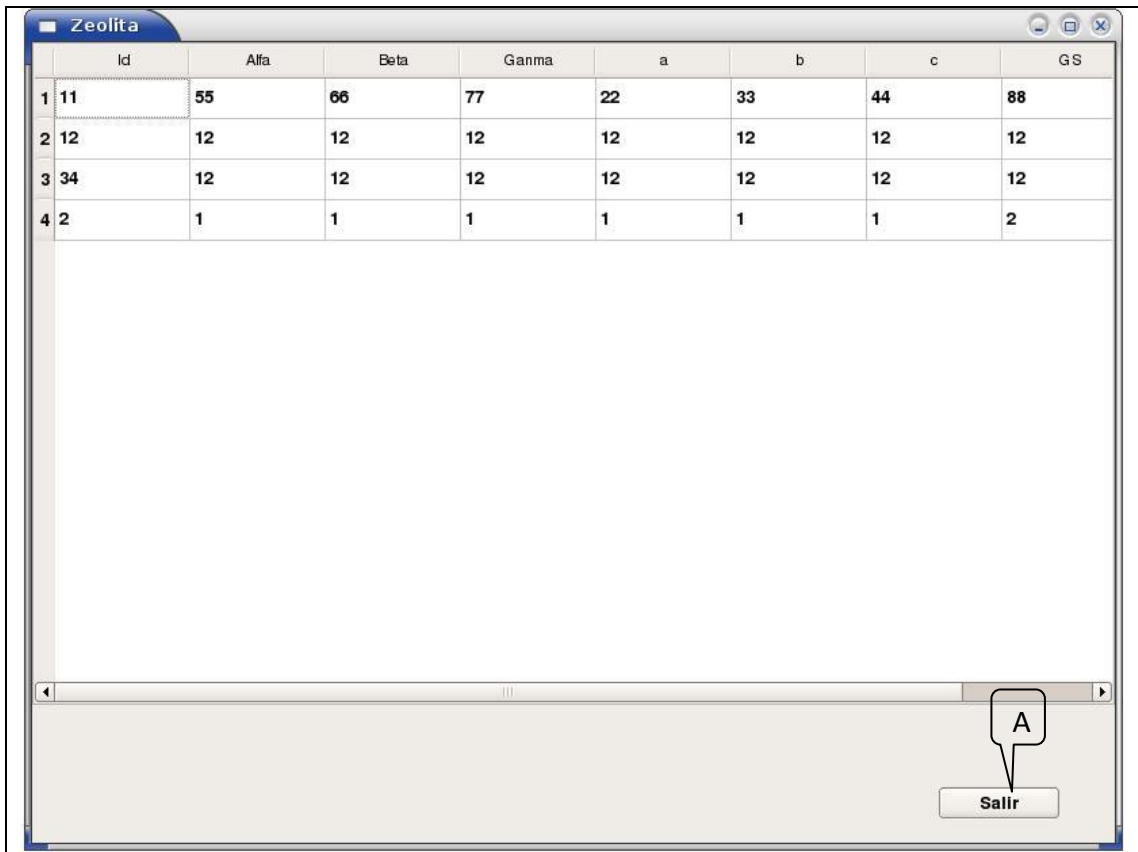
H: nombre: c, tipo: varchar (text para insertar el parámetro c de la celda estructural de la zeolita)

I: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)

J: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

<p>4- Introduce los datos solicitados por la aplicación.</p> <p>5- Presiona el botón Buscar.</p>	<p>3- Muestra Interfaz 11.</p>
--	--------------------------------

<p>9- Cambia los datos necesarios de la zeolita.</p> <p>10- Presiona el botón Aceptar.</p>	<p>6- Chequea que no existan campos vacíos.</p> <p>7- Chequea que la zeolita está registrada.</p> <p>8- Muestra Interfaz 12.</p> <p>11- Chequea que no existan campos vacíos.</p> <p>12- Comprueba que solo se haya introducido valores enteros en los campos de Alfa, Beta, Gamma.</p> <p>13- Comprueba que solo se haya introducido valores float en los campos de a, b, c.</p> <p>14- Muestra mensaje: “La zeolita fue actualizada”.</p>
<p>Sección 4: Mostrar Zeolita</p>	
<p>Interfaz 13</p>	



A: nombre: Salir (botón para indicar al sistema la conformidad con los datos y regresar al menú principal)

4- Presiona el botón Salir.	3- Muestra Interfaz 13. 5- Regresa a las opciones de la aplicación.
-----------------------------	--

Cursos Alternos

Sección 1:	
<p>Línea 5: Presiona el botón Cancelar.</p>	<p>5.1 Regresa a las opciones de la aplicación.</p>
	<p>Línea 6: Muestra mensaje: "Faltan campos por llenar".</p> <p>6.1 Regresa a la línea 3.</p>
	<p>Línea 7: Muestra mensaje: "Alfa, Beta y</p>

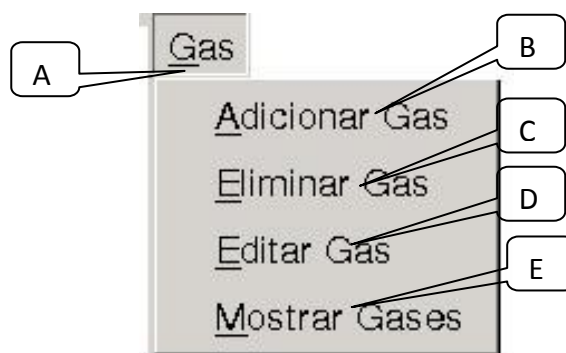
	Gamma deben tener valores enteros”. 7.1 Regresa a la línea 3.
	Línea 8: Muestra mensaje: “a, b, c deben tener valores float”. 8.1 Regresa a la línea 3.
	Línea 9: Muestra mensaje: “El identificador de la zeolita ya existe”. 9.1 Regresa a la línea 3.
Sección 2:	
Línea 5: Selecciona la opción Cancelar.	5.1 Regresa a las opciones de la aplicación.
	Línea 6: Muestra mensaje “Faltan campos por llenar”. 6.1 Regresa a la línea 3.
	Línea 7: Muestra mensaje: “La zeolita no está registrada”. 7.1 Regresa a la línea 3.
Sección 3:	
Línea 5: Selecciona la opción Cancelar.	5.1 Regresa a las opciones de la aplicación.
	Línea 6: Muestra mensaje “Por favor llene todos los campos”. 6.1 Regresa a la línea 3.
	Línea 7: Muestra mensaje: “La zeolita no está registrada en el sistema”. 7.1 Regresa a la línea 3.
Línea 10: Presiona el botón Cancelar.	10.1 Regresa a las opciones de la aplicación.

	<p>Línea 11 Muestra mensaje “Por favor llene todos los campos”.</p> <p>11.1 Regresa a la línea 8.</p>
	<p>Línea 12: Muestra mensaje: “Alfa, Beta y Gamma deben tener valores enteros”.</p> <p>12.1 Regresa a la línea 8.</p>
	<p>Línea 13: Muestra mensaje: “a, b, c deben tener valores float”.</p> <p>13.1 Regresa a la línea 8.</p>
Poscondiciones:	La zeolita es registrada en el sistema.

2.6.4 Descripción Caso de Uso Gestionar Gas

Nombre del Caso de uso:	Gestionar Gas
Actores:	Especialista
Propósito:	Adicionar, Eliminar, Modificar los gases que se utilizarán en los estudios en el sistema, así como, Mostrar los datos de los mismos.
Resumen	El caso de uso se inicia cuando el especialista selecciona la opción Gas. El especialista puede seleccionar la acción de Adicionar, Eliminar, Editar o Mostrar gases. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser especialista registrado en el sistema.
Tipo:	Real.
Referencia:	R4
Casos de uso relacionados:	

Interfaz 14



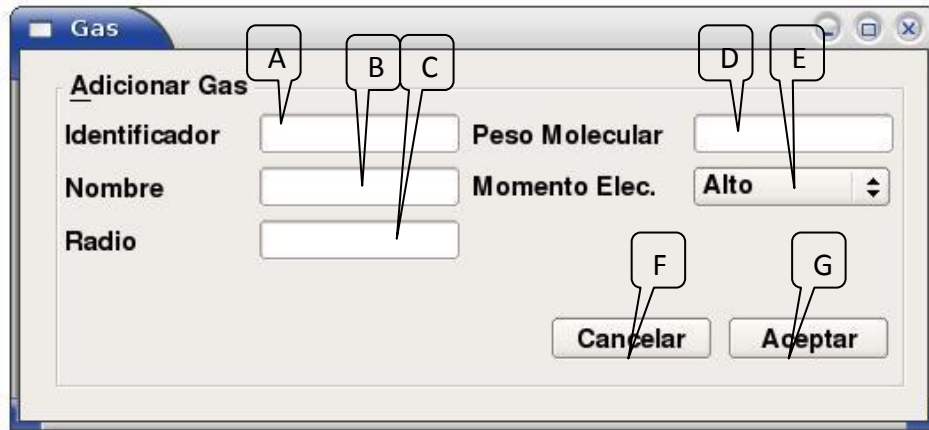
- A:** menú desplegable que muestra las diferentes opciones del gas.
- B:** indica la opción de adicionar un gas al sistema.
- C:** indica la opción de eliminar el gas del sistema.
- D:** indica la opción de editar el gas.
- E:** indica la opción de mostrar todos los gases registrados en el sistema.

Curso Normal de los Eventos

Acción del actor	Respuesta del sistema
<p>1- Selecciona en el menú principal la opción "Gas". (14-A)</p> <p>3- Selecciona la opción:</p> <ul style="list-style-type: none"> "Adicionar Gas". Sección 1. "Eliminar Gas". Sección 2. "Editar Gas". Sección 3. "Mostrar Gas". Sección 4. 	<p>2- Muestra menú con las opciones "Adicionar Gas", "Eliminar Gas", "Editar Gas" y "Mostrar Gas".</p>

Sección 1: Adicionar Gas

Interfaz 15



A: nombre: **identificador**, tipo: **varchar** (text para insertar el identificador del gas)

B: nombre: **nombre**, tipo: **varchar** (text para insertar el nombre del gas)

C: nombre: **radio**, tipo: **integer** (text para insertar el radio que ocuparán aproximadamente las moléculas de gas)

D: nombre: **peso molecular** tipo: **float** (text para insertar el peso molecular)

E: nombre: **momento eléctrico** tipo: **integer** (text insertar escoger el momento eléctrico)

G: nombre: **Aceptar** (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)

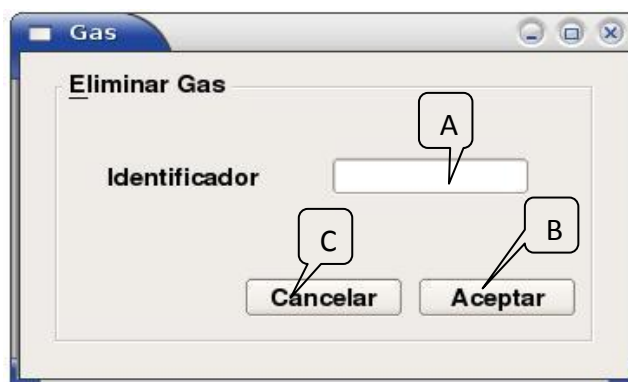
F: nombre: **Cancelar** (botón para indicar al sistema que regrese a las opciones iniciales)

<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Aceptar.</p>	<p>3- Muestra Interfaz 15.</p> <p>6- Chequea si el especialista ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Comprueba que pm sea un valor float.</p> <p>8- Comprueba que el radio sea un valor float.</p>
--	---

	<p>9- Chequea que el identificador del gas no este repetido.</p> <p>10- Registra el nuevo gas en el sistema.</p> <p>11- Muestra mensaje: "El gas fue adicionado correctamente".</p>
--	---

Sección 2: Eliminar Gas

Interfaz 16



A: nombre: identificador, tipo: varchar (text para insertar el identificador del gas)

B: nombre: Aceptar (botón para indicar a la aplicación la conformidad con los datos y dar curso a las acciones)

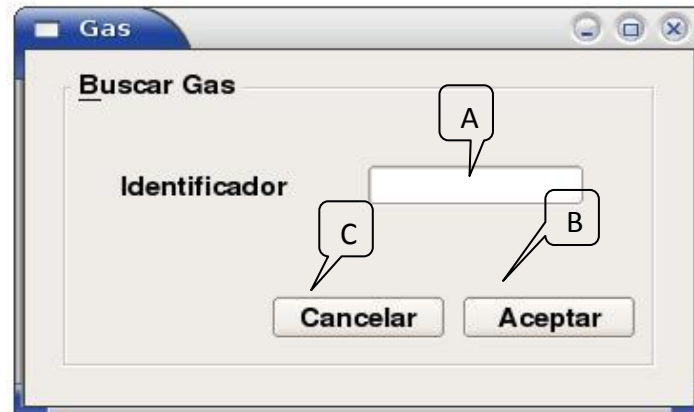
C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Eliminar.</p>	<p>3- Muestra Interfaz 16.</p> <p>6- Chequea si el especialista ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Chequea que el identificador del gas exista.</p> <p>8- Chequea que ninguna mezcla tenga contenido al gas.</p>
---	---

9- Muestra mensaje: "El gas "id_gas" fue correctamente eliminado".

Sección 3: Editar Gas

Interfaz 17

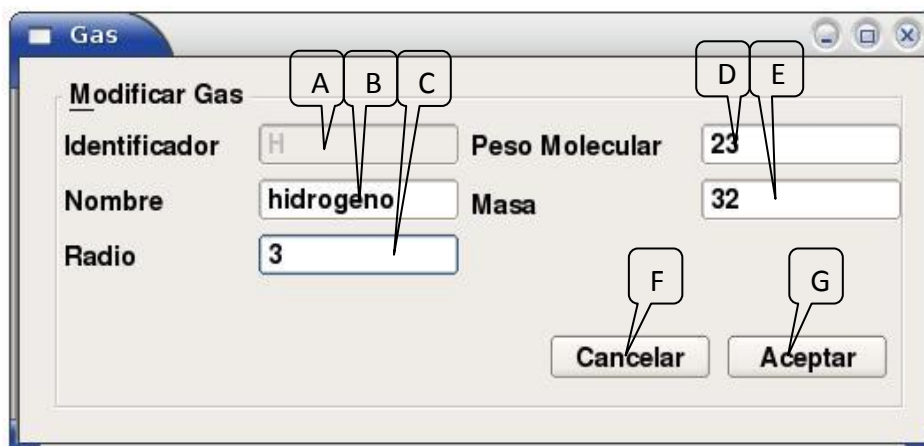


A: nombre: identificador, tipo: varchar (text para insertar el identificador del gas)

B: nombre: Aceptar (botón para indicar a la aplicación la conformidad con los datos y dar curso a las acciones)

C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

Interfaz 18



A: nombre: identificador, tipo: varchar (text para ver el identificador del gas)

B: nombre: nombre, tipo: varchar (text para cambiar el nombre del gas)

<p>C: nombre: radio, tipo: integer (text para cambiar radio de las moléculas de gas)</p> <p>D: nombre: peso molecular tipo: integer (text para cambiar el peso molecular)</p> <p>E: nombre: masa tipo: integer (text cambiar la masa)</p> <p>G: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)</p> <p>F: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Buscar.</p> <p>9- Cambia los datos necesarios del gas.</p> <p>10- Presiona el botón Aceptar.</p>	<p>3- Muestra Interfaz 17.</p> <p>6- Chequea que no existan campos vacíos.</p> <p>7- Chequea que el gas este registrado.</p> <p>8- Muestra Interfaz 18.</p> <p>11- Chequea que no existan campos vacíos.</p> <p>12- Comprueba que pm sea un valor float.</p> <p>13- Comprueba que el radio sea un valor float.</p> <p>14- Muestra mensaje: “El gas fue actualizado”.</p>
<p>Sección 4: Mostrar Gas</p>	
<p>Interfaz 19</p>	



A: nombre: Salir (botón para indicar al sistema la conformidad con los datos y regresar al menú principal)

<p>4- Presiona el botón Salir.</p>	<p>3- Muestra Interfaz 19.</p> <p>5- Regresa a las opciones del sistema.</p>
------------------------------------	--

Cursos Alternos

Sección 1:	
<p>Línea 5: Presiona el botón Cancelar.</p>	<p>5.1 Regresa a las opciones del sistema.</p>
	<p>Línea 6: Muestra mensaje: "Faltan campos por llenar".</p> <p>6.1 Regresa a la línea 3.</p>
	<p>Línea 7: Muestra mensaje: "El peso molecular debe ser un valor float".</p>

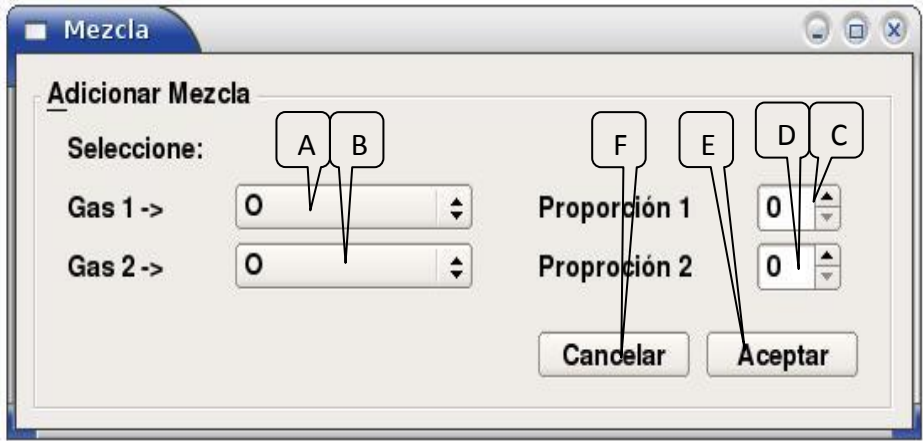
	7.1 Regresa a la línea 3.
	Línea 8: Muestra mensaje: “El radio debe ser un valor float”. 8.1 Regresa a la línea 3.
	Línea 9: Muestra mensaje: “El identificador del gas ya existe” 9.2 Regresa a la línea 3.
Sección 2:	
Línea 5: Selecciona la opción Cancelar.	5.1 Regresa a las opciones del sistema.
	Línea 6: Muestra mensaje “Faltan campos por llenar”. 6.1 Regresa a la línea 3.
	Línea 7: Muestra mensaje: “El gas no está registrado”. 7.1 Regresa a la línea 3.
	Línea 8: Muestra mensaje: “El gas no puede ser eliminado, pues está contenido en mezclas”. 8.1 Regresa a la línea 3.
Sección 3:	
Línea 5: Selecciona la opción Cancelar.	5.1 Regresa a las opciones del sistema.
	Línea 6: Muestra mensaje “Por favor llene todos los campos”. 6.1 Regresa a la línea 3.
	Línea 7 Muestra mensaje: “El gas no está registrado en el sistema”.

	7.1 Regresa a la línea 3.
Línea 10: Presiona el botón Cancelar.	10.1 Regresa a las opciones del sistema.
	Línea 11: Muestra mensaje “Por favor llene todos los campos”. 11.1 Regresa a la línea 8.
	Línea 12: Muestra mensaje: “El peso molecular debe ser un valor float”. 12.1 Regresa a la línea 8.
	Línea 13: Muestra mensaje: “El radio debe ser un valor float”. 13.1 Regresa a la línea 8.
Poscondiciones:	El Gas es registrado en el sistema.

2.6.5 Descripción Caso de Uso Gestionar Mezcla Gas

Nombre del Caso de uso:	Gestionar Mezcla Gas
Actores:	Especialista
Propósito:	Adicionar, Eliminar, Modificar las mezclas de los gases que se utilizarán en los estudios, en la aplicación, así como, Mostrar los datos de las mismas.
Resumen	El caso de uso se inicia cuando el especialista selecciona la opción Mezcla. El especialista puede seleccionar la acción de Adicionar, Eliminar, Editar o Mostrar mezclas. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser Especialista registrado en el sistema.
Tipo:	Real.
Referencia:	R5
Casos de uso	

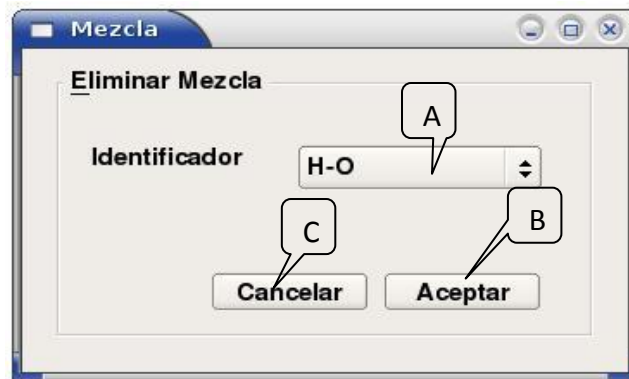
relacionados:	
Interfaz 20	
<p>The screenshot shows a menu titled 'Mezclas' (A) with four options: 'Adicionar Mezcla' (B), 'Eliminar Mezcla' (C), 'Editar Mezcla' (D), and 'Mostrar Mezcla' (E). Each option is underlined.</p>	
<p>A: menú desplegable que muestra las diferentes opciones de la mezcla. B: indica la opción de adicionar una mezcla al sistema. C: indica la opción de eliminar una mezcla del sistema. D: indica la opción de editar la mezcla. E: indica la opción de mostrar todas las mezclas registradas en el sistema.</p>	
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
<p>1- Selecciona en el menú principal la opción "Mezcla". (20-A)</p> <p>3- Selecciona la opción: "Adicionar Mezcla". Sección 1. "Eliminar Mezcla". Sección 2. "Editar Mezcla". Sección 3. "Mostrar Mezcla".</p>	<p>2- Muestra menú con las opciones "Adicionar Mezcla", "Eliminar Mezcla", "Editar Mezcla" y "Mostrar Mezcla".</p>

Sección 4.	
Sección 1: Adicionar Mezcla	
Interfaz 21	
	
<p>A: nombre: gas 1, tipo: varchar (text para insertar el identificador del gas1 de la mezcla)</p> <p>B: nombre: gas 2, tipo: varchar (text para insertar el identificador del gas2 de la mezcla)</p> <p>C: nombre: proporción 1, tipo: integer (text para insertar la proporción a la que va a estar representado por el gas 1)</p> <p>D: nombre: proporción 2 tipo: integer (text para insertar la proporción a la que va a estar representado por el gas 2)</p> <p>E: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)</p> <p>F: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Aceptar.</p>	<p>3- Muestra Interfaz 21.</p> <p>6- Chequea si el especialista ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Comprueba que solo se haya introducido</p>

	<p>números enteros en las proporciones del los gases.</p> <p>8- Chequea que el identificador del gas1 y el gas2 no estén registrados como mezcla.</p> <p>9- Chequea que el identificador del gas1 y el gas2 estén registrados como gas.</p> <p>10- Registra la nueva mezcla en el sistema.</p> <p>11- Muestra mensaje: “La mezcla fue adicionada correctamente”.</p>
--	--

Sección 2: Eliminar Mezcla

Interfaz 22



A: nombre: identificador, tipo: varchar (text para insertar el identificador de la mezcla)

B: nombre: Aceptar (botón para indicar a la aplicación la conformidad con los datos y dar curso a las acciones)

C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Eliminar.</p>	<p>3- Muestra Interfaz 22.</p> <p>6- Chequea si el especialista ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Chequea que la mezcla exista.</p>
---	---

8- Muestra mensaje: "La mezcla "id_gas1" y "id_gas2" fue correctamente eliminada".

Sección 3: Editar Mezcla

Interfaz 23

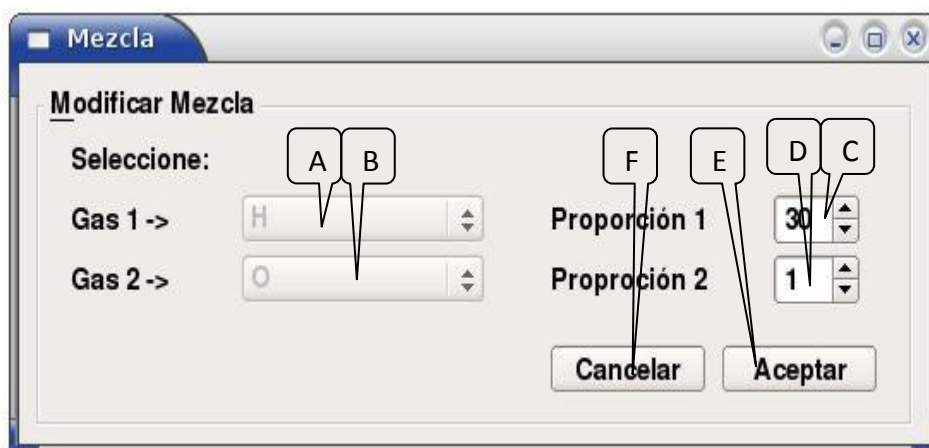


A: nombre: identificador, tipo: varchar (text para insertar el identificador de la mezcla)

B: nombre: Aceptar (botón para indicar a la aplicación la conformidad con los datos y dar curso a las acciones)

C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

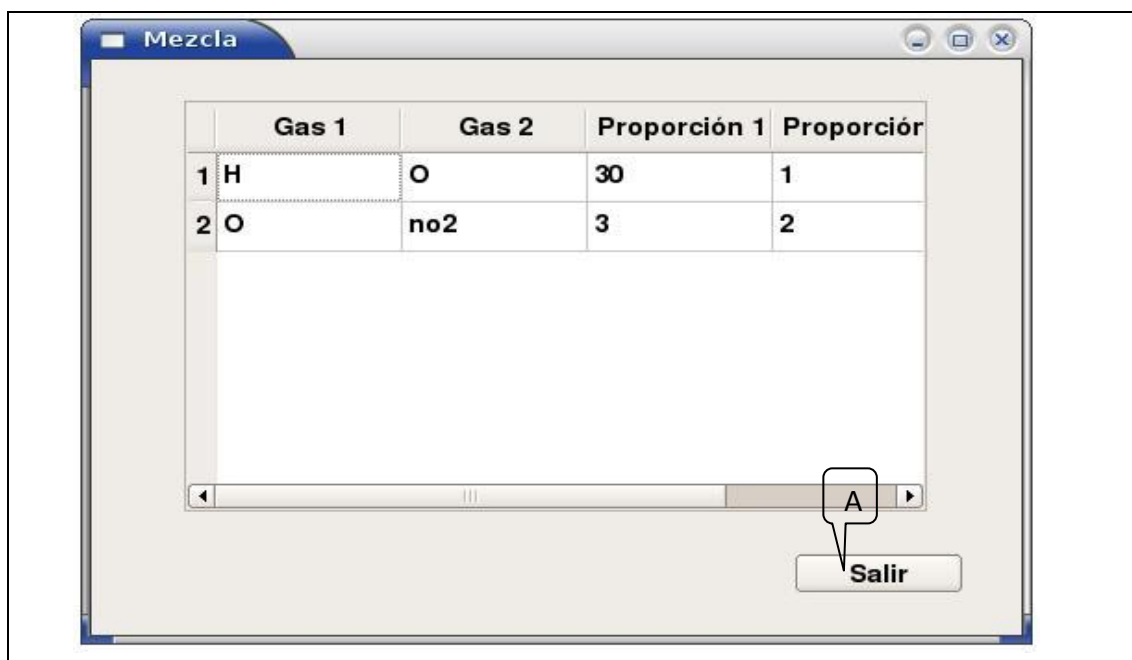
Interfaz 24



A: nombre: gas 1, tipo: varchar (text para mostrar el identificador del gas1 de la mezcla)

B: nombre: gas 2, tipo: varchar (text para mostrar el identificador del gas2 de la mezcla)

<p>C: nombre: proporción 1, tipo: integer (text para cambiar la proporción a la que va a estar representado el gas 1)</p> <p>D: nombre: proporción 2 tipo: integer (text para cambiar la proporción a la que va a estar representad el gas 2)</p> <p>E: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)</p> <p>F: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Buscar.</p> <p>9- Cambia los datos necesarios de la mezcla.</p> <p>10- Presiona el botón Aceptar.</p>	<p>3- Muestra Interfaz 23.</p> <p>6- Chequea que no existan campos vacíos.</p> <p>7- Chequea que la mezcla está registrada.</p> <p>8- Muestra Interfaz 24.</p> <p>11- Chequea que no existan campos vacíos.</p> <p>12- Chequea que se halla inserta números en los campos de las proporciones.</p> <p>13- Muestra mensaje: “La mezcla fue actualizada correctamente”.</p>
<p>Sección 4: Mostrar Mezclas</p>	
<p>Interfaz 25</p>	



A: nombre: Salir (botón para indicar al sistema la conformidad con los datos y regresar al menú principal)

<p>4- Presiona el botón Salir.</p>	<p>3- Muestra Interfaz 25.</p> <p>5- Regresa a las opciones del sistema.</p>
------------------------------------	--

Cursos Alternos

Sección 1:	
<p>Línea 5: Presiona el botón Cancelar.</p>	<p>5.1 Regresa a las opciones del sistema.</p>
	<p>Línea 6: Muestra mensaje: "Faltan campos por llenar"</p> <p>6.1 Regresa a la línea 3.</p>
	<p>Línea 7: Muestra mensaje: "Las proporciones deben ser números enteros".</p> <p>7.2 Regresa a la línea 3.</p>
	<p>Línea 8: Muestra mensaje: "La mezcla ya existe"</p>

	8.2 Regresa a la línea 3.
	Línea 9: Muestra mensaje: “Los gases escogidos deben estar registrados en el sistema”. 9.2 Regresa a la línea 3.
Sección 2:	
Línea 5: Selecciona la opción Cancelar.	5.1 Regresa a las opciones del sistema.
	Línea 6: Muestra mensaje: “Faltan campos por llenar”. 6.1 Regresa a la línea 3.
	Línea 7: Muestra mensaje: “La mezcla no existe”. 7.1 Regresa a la línea 3.
Sección 3:	
Línea 5: Selecciona la opción Cancelar.	5.1 Regresa a las opciones del sistema.
	Línea 6: Muestra mensaje “Por favor llene todos los campos”. 6.1 Regresa a la línea 3.
	Línea 7: Muestra mensaje: “La mezcla no”. 7.1 Regresa a la línea 3.
Línea 10: Presiona el botón Cancelar.	10.1 Regresa a las opciones iniciales del sistema.
	Línea 11: Muestra mensaje “Por favor llene todos los campos”. 11.1 Regresa a la línea 8.

	<p>Línea 12: Muestra mensaje “Por favor, las proporciones deben ser números enteros”.</p> <p>12.1 Regresa a la línea 8.</p>
Poscondiciones:	Se registra la mezcla en el sistema.

2.6.6 Descripción Caso de Uso Gestionar Estudio

Nombre del Caso de uso:	Gestionar Estudio
Actores:	Especialista
Propósito:	Adicionar, Eliminar o Mostrar los estudios en el sistema.
Resumen	El caso de uso se inicia cuando el especialista selecciona la opción Estudio. El especialista puede seleccionar la acción de Crear, Eliminar o Mostrar estudios. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser especialista registrado en el sistema.
Tipo:	Real.
Referencia:	R6,10
Casos de uso relacionados:	Gestionar Atajo (incluido), Gestionar Barrera (incluido), Guardar Resultado (extendido), Detener Estudio (extendido).

Interfaz 26

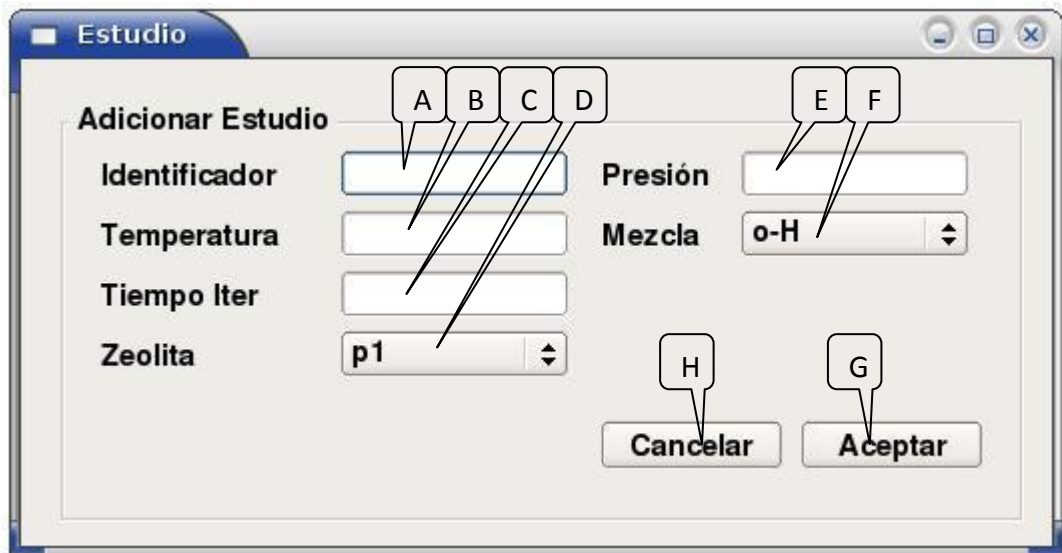


- A:** menú desplegable que muestra las diferentes opciones del estudio.
- B:** indica la opción de crear un estudio, que será registrado en el sistema.
- C:** indica la opción de eliminar un estudio del sistema.
- D:** indica la opción de mostrar todos los estudios realizados.

Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
<p>1- Selecciona en el menú principal la opción "Estudio". (26-A)</p> <p>3- Selecciona la opción: "Crear Estudio". Sección 1. "Eliminar Estudio". Sección 2. "Mostrar Estudios". Sección 3.</p>	<p>2- Muestra menú con las opciones "Crear Estudio", "Eliminar Estudio", "Mostrar Estudios".</p>

Sección 1: Crear Estudio

Interfaz 27



A: nombre: **identificador**, tipo: **varchar** (text para insertar el identificador del estudio)

B: nombre: **temperatura**, tipo: **varchar** (text para insertar la temperatura a

<p>utilizar en el estudio)</p> <p>C: nombre: tiempo iter, tipo: integer (text para insertar el tiempo de cada iteración en el estudio)</p> <p>D: nombre: zeolita, tipo: integer (text para insertar la presión a la que trabajará el estudio)</p> <p>E: nombre: Tiempo Iteración tipo: integer (text para insertar el tiempo de una iteración en el estudio)</p> <p>F: nombre: Mezcla, tipo: varchar (text para escoger la mezcla a utilizar en el estudio)</p> <p>G: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)</p> <p>H: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Aceptar.</p> <p>11- Presiona el botón Ejecutar.</p>	<p>3- Muestra Interfaz 27.</p> <p>6- Chequea si el especialista ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Comprueba que solo se haya introducido números en los campos de temperatura, presión y tiempo iteración.</p> <p>8- Chequea que el identificador del estudio no exista.</p> <p>9- Va a caso de uso Gestionar Barrera. (Sección 1)</p> <p>10- Va a caso de uso Gestionar Atajo. (Sección 1)</p> <p>12- Comienza a ejecutar el estudio.</p>

Sección 2: Eliminar Estudio

Interfaz 28



A: nombre: identificador, tipo: varchar (text para insertar el identificador del estudio)

B: nombre: Aceptar (botón para indicar a la aplicación la conformidad con los datos y dar curso a las acciones)

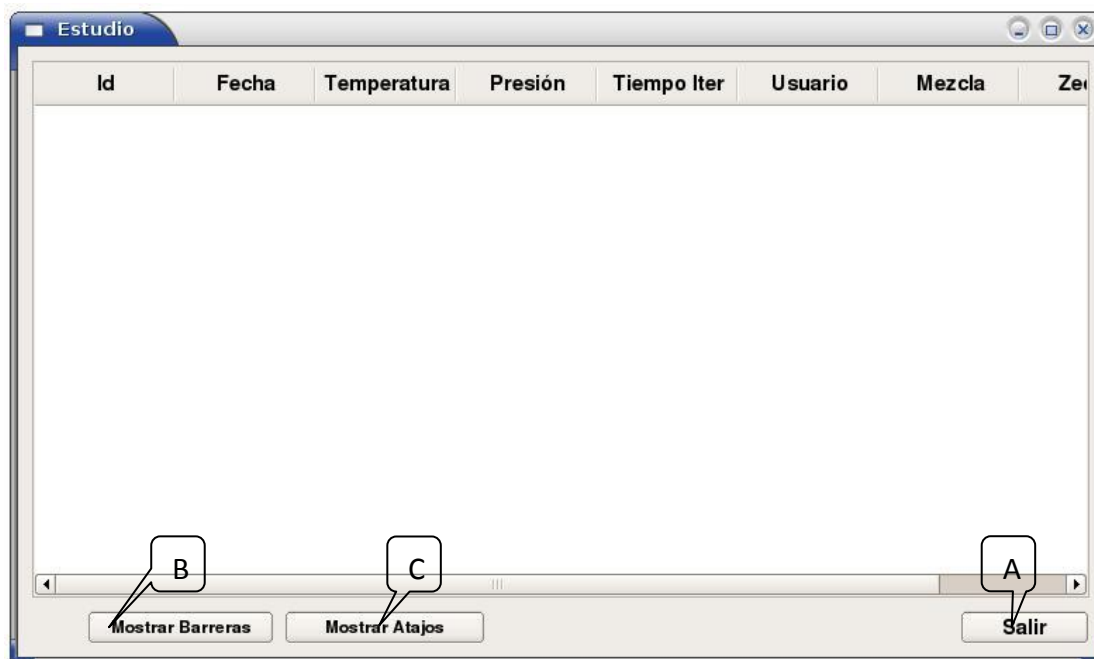
C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

<p>4- Introduce los datos solicitados por el sistema.</p> <p>5- Presiona el botón Aceptar.</p>	<p>3- Muestra Interfaz 28.</p> <p>6- Chequea si el especialista ha dejado algún campo vacío, en los datos a llenar.</p> <p>7- Chequea que el estudio exista.</p> <p>8- Llama al caso de uso: Gestionar Barreras. (Sección 2).</p> <p>9- Llama al caso de uso: Gestionar Atajos. (Sección 2)</p> <p>10- Llama al caso de uso: Detener Estudio. (Sección 2)</p> <p>11- Llama al caso de uso: Gestionar Atajos. (Sección 2)</p>
--	--

	<p>12- Llama al caso de uso: Detener Estudio. (Sección 2)</p> <p>13- Llama al caso de uso: Guardar Resultado. (Sección 3)</p> <p>14- Muestra mensaje: “El estudio fue correctamente eliminado”.</p>
--	---

Sección 3: Mostrar Estudios

Interfaz 29



A: nombre: Salir (botón para indicar al sistema la conformidad con los datos y regresar al menú principal)

B: nombre: Mostrar Barreras (botón para indicar al sistema la conformidad con los datos y mostrar las barreras registradas en el sistema)

C: nombre: Atajos (botón para indicar al sistema la conformidad con los datos y mostrar los atajos registrados en el sistema)

<p>4- Presiona el botón Salir.</p>	<p>3- Muestra Interfaz 29.</p> <p>5- Regresa a las opciones del sistema.</p>
------------------------------------	--

Cursos Alternos

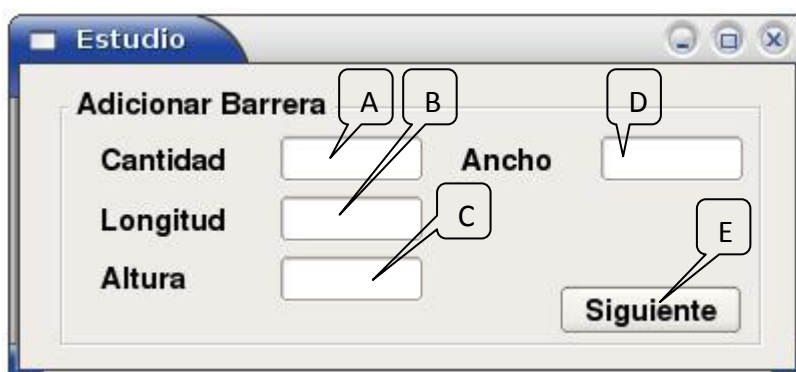
Sección 1:	
Línea 5: Presiona el botón Cancelar.	5.1- Regresa a las opciones del sistema.
	Línea 6: Muestra mensaje: "Faltan campos por llenar". 6.1 Regresa a la línea 3.
	Línea 7: Muestra mensaje: "La presión, la temperatura y el tiempo iteración deben ser valores enteros". 7.1 Regresa a la línea 3.
	Línea 8: Muestra mensaje: "El identificador del estudio ya existe en el sistema". 8.1 Regresa a la línea 3.
Línea 12: Presiona el botón Detener.	12.1 Va a caso de uso Detener Estudio. (Sección 3)
Sección 2:	
Línea 5: Presiona el botón Cancelar.	5.1- Regresa a las opciones del sistema.
	Línea 6: Muestra mensaje: "Faltan campos por llenar" 6.1 Regresa a la línea 3.
	Línea 7: Muestra mensaje: "Por favor revise el identificador, no está registrado". 7.1 Regresa a la línea 3.
Sección 3:	
Línea 4.1- Presiona el botón Mostrar Barreras.	4.1.1- Va a caso de uso Gestionar

	Barrera. (Sección 3)
Línea 4.1- Presiona el botón Mostrar Atajos.	4.1.1- Va a caso de uso Gestionar Atajo. (Sección 3)
Poscondiciones:	Se registra el estudio en el sistema.

2.6.7 Descripción Caso de Uso Gestionar Barrera

Nombre del Caso de uso:	Gestionar Barrera
Actores:	Especialista
Propósito:	Adicionar, Eliminar o Mostrar los defectos de tipo barrera que forman parte del estudio.
Resumen	El caso de uso se inicia cuando el sistema solicita los datos de configuración de los defectos de tipo barrera que se desea adicionar al sistema. El sistema permite mostrar y eliminar las barreras registradas en el sistema. El caso de uso termina cuando la barrera es registrada en el sistema o cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones	Ser especialista registrado en el sistema.
Tipo:	Real e incluido.
Referencia:	R7
Casos de uso relacionados:	(Gestionar Estudio, incluido)

Interfaz 30



A: nombre: cantidad, tipo: int (text para insertar la cantidad de tipo defecto barrera a agregar en el sistema)

B: nombre: longitud, tipo: int (text para insertar la longitud del tipo de defecto barrera)

C: nombre: altura, tipo: int (text para insertar la altura del tipo de defecto barrera)

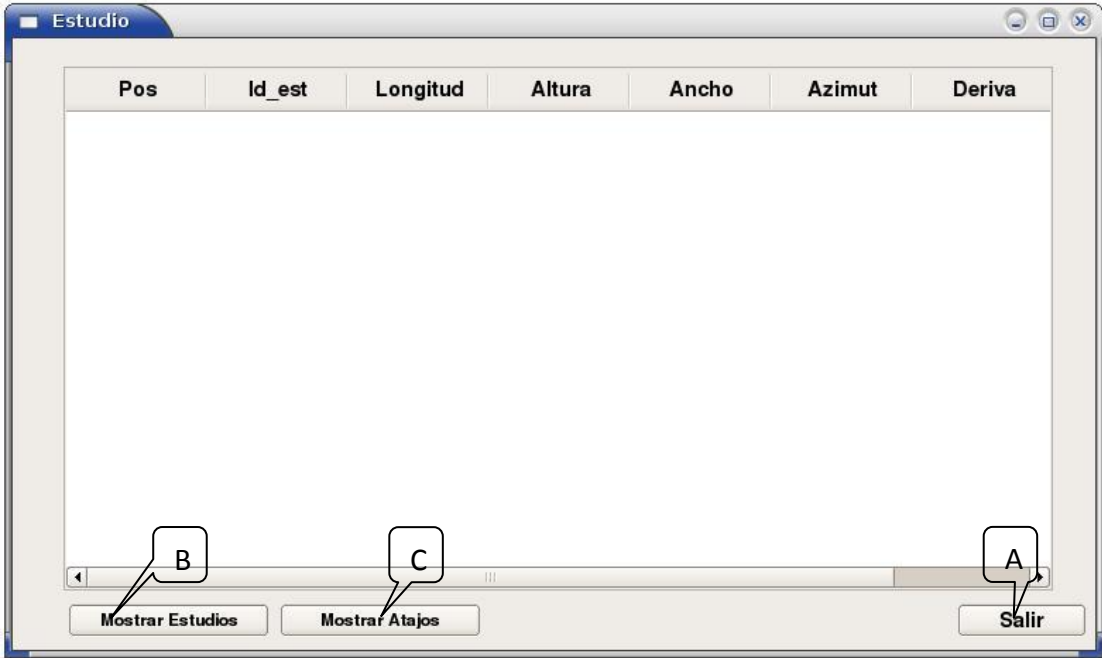
D: nombre: ancho, tipo: int (text para insertar el ancho del tipo de defecto barrera)

E: nombre: Siguiete (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones)

Curso Normal de Eventos

Sección 1: Adicionar Barrera

Acción del actor	Respuesta del sistema
2- Introduce los datos solicitados por el sistema. 3- Presiona el botón Aceptar.	1- Muestra Interfaz 30. 4- Chequea si el especialista ha dejado algún campo, en los datos a llenar, vacío. 5- Comprueba que se halla insertado valores enteros en la cantidad, el ancho, la

	<p>altura y largo del ortoedro.</p> <p>6- Genera valores aleatorios de pos y ángulos a las barreras.</p> <p>7- Quedan registrado los tipos de defectos barrera.</p>
Sección 2: Eliminar Barrera	
	<p>1- Elimina el tipo de defecto barrera que corresponde al estudio que lo solicita.</p> <p>2-La barrera queda eliminada del sistema.</p>
Sección 3: Mostrar Barrera	
Interfaz31	
	
<p>A: nombre: Salir (botón para indicar al sistema la conformidad con los datos y regresar al menú principal)</p> <p>B: nombre: Mostrar Estudios (botón para indicar al sistema la conformidad con los datos y mostrar los estudios registrados en el sistema)</p> <p>C: nombre: Atajos (botón para indicar al sistema la conformidad con los datos y mostrar los atajos registrados en el sistema)</p>	
2- Presiona el botón	1- Muestra Interfaz 31.

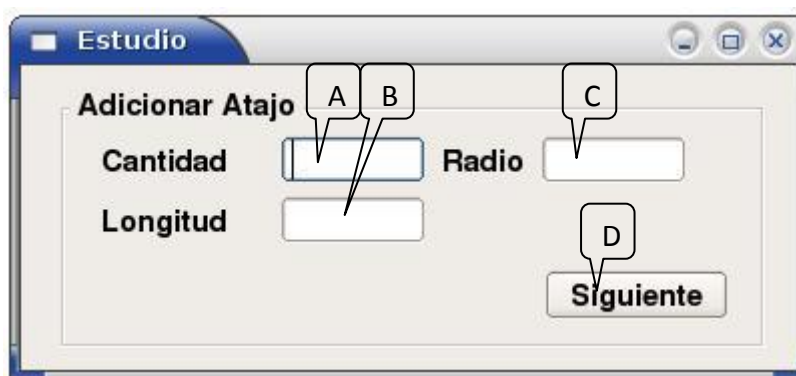
Salir.	
Cursos Alternos	
Sección 1:	
	<p>Línea 4: Muestra mensaje: “Faltan campos por llenar”</p> <p>4.1 Regresa a la línea 2.</p>
	<p>Línea 5: Muestra mensaje: “La cantidad, el ancho, el largo y la altura de la barrera deben ser valores enteros”.</p> <p>5.1 Regresa a la línea 2.</p>
Sección 3:	
Línea 2.1- Presiona el botón Mostrar Estudios.	<p>Línea 2.1.1- Muestra los estudios registrados en el sistema.</p>
Línea 2.2- Presiona el botón Mostrar Atajos.	<p>Línea 2.2.1- Muestra los atajos registrados en el sistema.</p>
Poscondiciones:	<p>La barrera queda registrada en el sistema, para uso del estudio.</p>

2.6.8 Descripción Caso de Uso Gestionar Atajo

Nombre del Caso de uso:	Gestionar Atajo
Actores:	Especialista
Propósito:	Adicionar, Eliminar o Mostrar los defectos de tipo atajo que forman parte del estudio.
Resumen	El caso de uso se inicia cuando el sistema solicita los datos de configuración del defecto de tipo atajo que se

	desea adicionar al sistema. El sistema permite mostrar y eliminar los atajos registrados en el sistema. El caso de uso termina cuando el especialista agrega el atajo al sistema, o cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser especialista registrado en el sistema.
Tipo:	Real e incluido.
Referencia:	R8
Casos de uso relacionados:	

Interfaz 32



A: nombre: cantidad, tipo: int (text para insertar la cantidad de defectos de tipo atajo a adicionar al estudio)

B: nombre: longitud, tipo: int (text para insertar la longitud de defectos de tipo atajo a adicionar al estudio)

C: nombre: radio, tipo: int (text para insertar el radio del tipo de defecto atajo a adicionar al estudio)

D: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y dar curso a las acciones).

Curso Normal de Eventos

Sección 1: Adicionar Atajo

Acción del actor	Respuesta del sistema
	1-Muestra Interfaz 32.

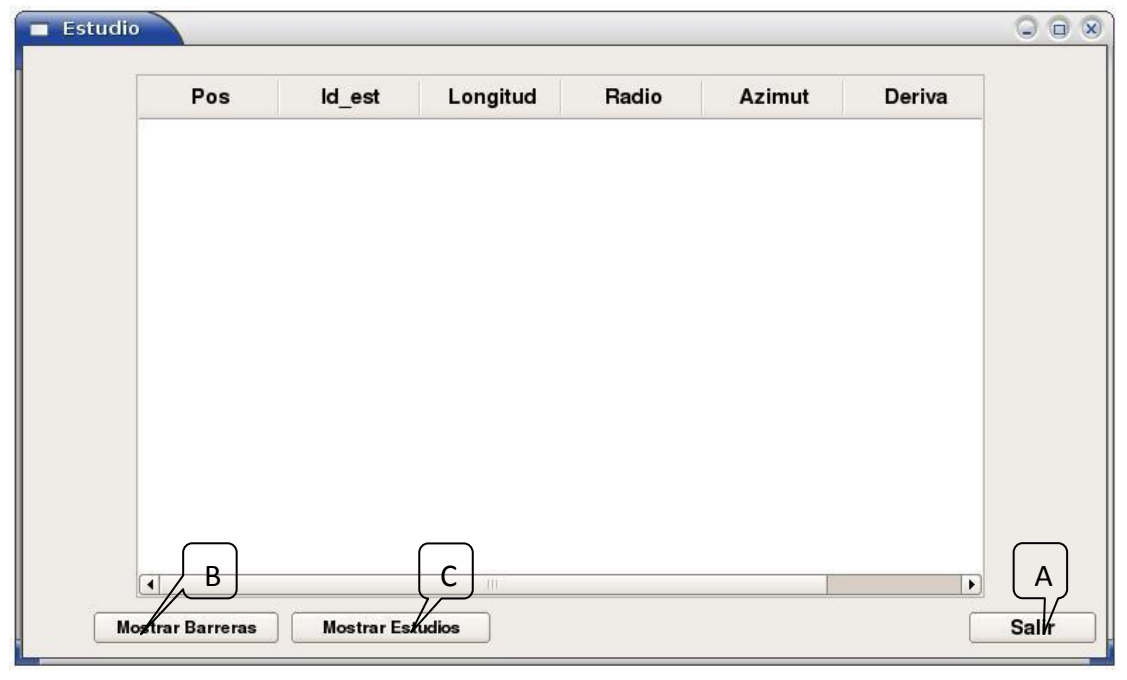
<p>2-Introduce los datos solicitados.</p> <p>3- Presiona el botón Aceptar.</p>	<p>4- Chequea si el especialista ha dejado algún campo, en los datos a llenar, vacío.</p> <p>5- Comprueba que se halla insertado valores enteros en la cantidad, el radio y longitud del atajo.</p> <p>6- Da valores aleatorios de pos y ángulos a los atajos.</p> <p>7- Quedan registrado los tipos de defectos atajo en el sistema.</p>
--	---

Sección 2: Eliminar Atajo

	<p>1- Elimina el tipo de defecto atajo que corresponde al estudio que lo solicita.</p> <p>2-El atajo queda eliminado del sistema.</p>
--	---

Sección 3: Mostrar Atajos


Interfaz 33




<p>A: nombre: Salir (botón para indicar al sistema la conformidad con los datos y regresar al menú principal)</p> <p>B: nombre: Mostrar Estudios (botón para indicar al sistema la conformidad con los datos y mostrar las barreras registradas en el sistema)</p> <p>C: nombre: Mostrar Estudios (botón para indicar al sistema la conformidad con los datos y mostrar los estudios registrados en el sistema)</p>	
2- Presiona el botón Salir.	1- Muestra Interfaz 33.
Cursos Alternos	
Sección 1:	
	<p>Línea 4: Muestra mensaje: “Faltan campos por llenar”.</p> <p>4.1 Regresa a la línea 2.</p>
	<p>Línea 5: Muestra mensaje: “La cantidad, el radio y la altura deben ser valores enteros”.</p> <p>5.1 Regresa a la línea 2.</p>
Sección 3:	
Línea 2.1- Presiona el botón Mostrar Barreras.	Línea 2.1.1- Muestra las barreras registradas en el sistema.
Línea 2.2- Presiona el botón Mostrar Estudios.	Línea 2.2.1- Muestra los estudios registrados en el sistema.
Poscondiciones:	El atajo queda registrado en el sistema, para uso del estudio.

2.6.9 Descripción Caso de Uso Detener Estudio

Nombre del	Detener estudio
------------	-----------------

Caso de uso:	
Actores:	Especialista
Propósito:	Detener la ejecución del estudio ante cualquier eventualidad tanto del usuario como del sistema.
Resumen	El caso de uso se inicia cuando se solicita la interrupción del estudio. El caso de uso permite registrar el motivo y la fecha de la interrupción. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser especialista registrado en el sistema.
Tipo:	Real y extendido.
Referencia:	R9
Casos de uso relacionados:	(Gestionar Estudio, extendido)
Curso Normal de Eventos	
Sección 1: Guardar Interrupción	
Interfaz 34	
	
A: nombre: Aceptar (botón para indicar al sistema la conformidad con los datos y almacenar la interrupción)	
Acción del actor	Respuesta del sistema
	1- Muestra interfaz 34.

<p>2- Introduce las causas de la interrupción.</p> <p>3- Presiona el botón Aceptar.</p>	<p>4- Chequea que no existan campos vacíos.</p> <p>5- Detiene el estudio.</p> <p>6- Registra la causa de la interrupción en el sistema.</p>
<p>Sección 2: Eliminar Interrupción</p>	
	<p>1- Elimina la interrupción del sistema.</p>
<p>Sección 3:Mostrar Interrupción</p>	
<p>Interfaz 35</p> 	
<p>A: nombre: identificador, tipo: varchar (text para insertar el identificador del estudio)</p> <p>B: nombre: Aceptar (botón para indicar a la aplicación la conformidad con los datos y mostrar las características de la interrupción)</p> <p>C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)</p>	
<p>Interfaz 36</p>	

A: nombre: Salir (botón para indicar al sistema que regrese a las opciones iniciales)

<p>2- Introduce los datos solicitados por el sistema.</p> <p>3- Presiona el botón Aceptar.</p> <p>7- Presiona el botón Salir.</p>	<p>1- Muestra Interfaz 35.</p> <p>4- Chequea que no existan campos vacíos.</p> <p>5- Chequea que el identificador del estudio exista.</p> <p>6- Muestra Interfaz 36.</p> <p>8- Regresa a las opciones iniciales del sistema.</p>
---	--

Cursos Alternos

Sección 1:

<p>Línea 3: Presiona el botón Cancelar.</p>	<p>3.1- Regresa a las opciones iniciales del</p>
--	--

	sistema.
	Línea 4: Muestra mensaje: "Faltan campos por llenar". 4.1- Regresa a línea 2.
Sección 3:	
Línea 3: Presiona el botón Cancelar.	3.1- Regresa a las opciones iniciales del sistema.
	Línea 4: Muestra mensaje: "Faltan campos por llenar". 4.1- Regresa a línea 2.
Poscondiciones:	Se registra la causa de la interrupción del estudio en el sistema. Se detiene el estudio.

2.6.10 Descripción Caso de Uso Guardar Resultado.

Nombre del Caso de uso:	Guardar Resultado
Actores:	Especialista
Propósito:	Guardar los resultados del estudio.
Resumen	El caso de uso se inicia cuando el especialista desea guardar resultados parciales o totales de la corrida del estudio. El caso de uso permite registrar los resultados en el sistema. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser especialista registrado en el sistema.
Tipo:	Real y extendido.
Referencia:	R11
Casos de uso relacionados:	(Gestionar Estudio, extendido)
Curso Normal de Eventos	
Sección 1: Guardar Resultado y Continuar.	

Acción del actor	Respuesta del sistema
	1- Selecciona el identificador del estudio. 2- Captura el día y la hora en que se guardó el estudio. 3- Guarda los resultados del estudio. 4- Continúa ejecutándose el estudio.
Sección 2: Guardar Resultado y Finalizar.	
	1- Selecciona el identificador del estudio. 2- Captura el día y la hora en que se guardó el estudio. 3- Guarda los resultados del estudio. 4- Se detiene la ejecución el estudio.
Sección 3: Eliminar Estudio.	
	1- Elimina el resultado registrado en el sistema.
Poscondiciones:	Se registra el resultado en el sistema.

2.6.11 Descripción Caso de Uso Entregar Resultado

Nombre del Caso de uso:	Entregar Resultados
Actores:	Especialista
Propósito:	Entregar los resultados del estudio.
Resumen	El caso de uso se inicia cuando le solicitan al especialista los resultados del estudio. El caso de uso permite que al especialista mostrar e imprimir los resultados del estudio. El caso de uso termina cuando el especialista cierra el sistema o accede a otras opciones de la aplicación.
Precondiciones:	Ser especialista registrado en el sistema.
Tipo:	Real.

Referencia:	R12
Casos de uso relacionados:	

Interfaz 37

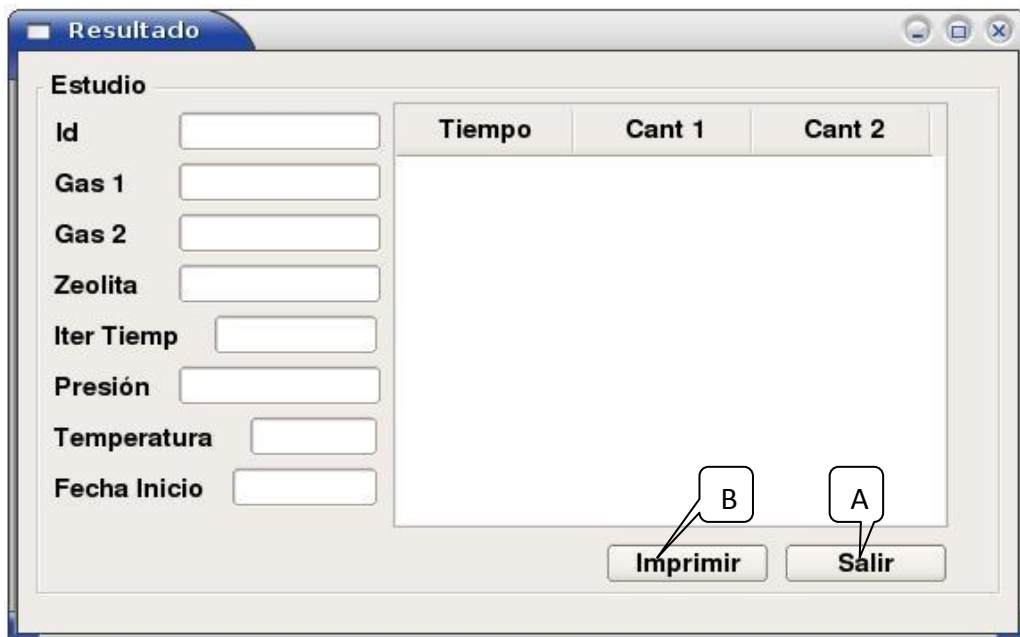


A: nombre: identificador, tipo: varchar (text para insertar el identificador del estudio)

B: nombre: Aceptar (botón para indicar a la aplicación la conformidad con los datos y mostrar los resultados)

C: nombre: Cancelar (botón para indicar al sistema que regrese a las opciones iniciales)

Interfaz 38



<p>A: nombre: Salir (botón para indicar al sistema que regrese a las opciones iniciales)</p> <p>B: nombre: Imprimir (botón para indicar al sistema que imprima los resultados)</p>	
Curso Normal de Eventos	
<p>1-Selecciona la opción “Mostrar Resultado”.</p> <p>3- Presiona el botón Aceptar.</p> <p>7- Presiona el botón Salir.</p>	<p>2- Muestra interfaz 37.</p> <p>4- Chequea que no existan campos vacios.</p> <p>5- Chequea que el identificador del estudio exista.</p> <p>6- Muestra interfaz 38.</p> <p>8- Regresa a las opciones iniciales del sistema.</p>
Cursos Alternos	
<p>Línea 3: Presiona el botón Cancelar.</p>	<p>3.2- Regresa a las opciones iniciales del sistema.</p>
	<p>Línea 4: Muestra mensaje:”Faltan campos por llenar”.</p> <p>4.2- Regresa a la línea 2.</p>
	<p>Línea 5: Muestra mensaje:”El identificador del estudio no está registrado”.</p> <p>5.2- Regresa a la línea 2.</p>
<p>Línea 7: Presiona el botón Imprimir.</p>	<p>7.1- Imprime los resultados.</p>

Poscondiciones:	Se entregan los resultados del estudio solicitado.
------------------------	--

Matriz de Trazabilidad de los Requerimientos. Ver Anexo 4.

2.7 Conclusiones

En este capítulo quedaron plasmadas las características que presenta el sistema, los problemas existentes en la entidad, y se hace una propuesta de solución a dichos problemas. Teniendo en cuenta lo dicho anteriormente se pudo apreciar que debido al bajo nivel de estructuración que presenta el negocio estudiado, se llegó a la conclusión de realizar un modelo de dominio ya que muestra los principales conceptos que se manejan en el sistema en desarrollo. Este trabajo ha expuesto cómo el modelado del dominio puede facilitar la identificación de los requisitos tanto funcionales como no funcionales del sistema. Además, de las descripciones de las funcionalidades del mismo.

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

3.1 Introducción:

El análisis, diseño, implementación y prueba se aborda en la segunda fase de desarrollo dentro de la metodología DSDM. En esta fase se crean los documentos indispensables para el desarrollo de la aplicación informática utilizando las extensiones del UML. En este capítulo es donde se ajusta el resultado del análisis a las tecnologías y lenguajes que serán utilizados, además de darle forma a la arquitectura del sistema.

En el capítulo finaliza con un grupo de pruebas que validan el trabajo realizado.

3.2 Análisis.

En el análisis es donde se identifican las necesidades del cliente, se definen los conceptos del sistema para establecer su viabilidad, se asigna funciones al Hardware, Software y otros elementos del Sistema, y se crea un sistema (prototipo del diseño) que se puede dar con seguridad a los usuarios finales para el uso diario, también para los propósitos de prueba.

3.2.1 Diagramas de Clases del Análisis.

Los diagramas de clases de análisis son elementos estáticos, como clases, interfaces y relaciones. Sus conexiones conforman un grafo. Este diagrama de clases ayuda a modelar el sistema.

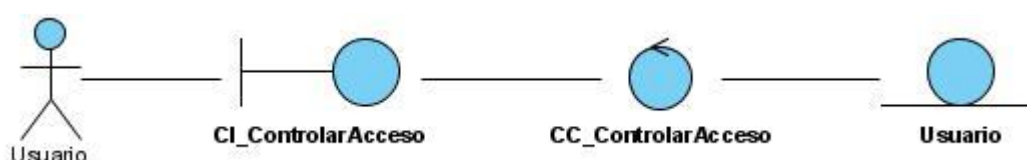


Figura 4. Diagrama de Clase del Análisis, Caso de Uso Controlar Acceso.

Diagramas de Clases del Análisis, Ver Anexos 5,6,7,8,9,10,11,12,13,14.

3.2.2 Diagramas de Colaboración de Clases Análisis.

Los diagramas de colaboración del análisis muestran la interacción entre las clases del análisis, usuarios y eventos del sistema. Posee eventos del análisis que define el comportamiento de las clases y sus funciones.

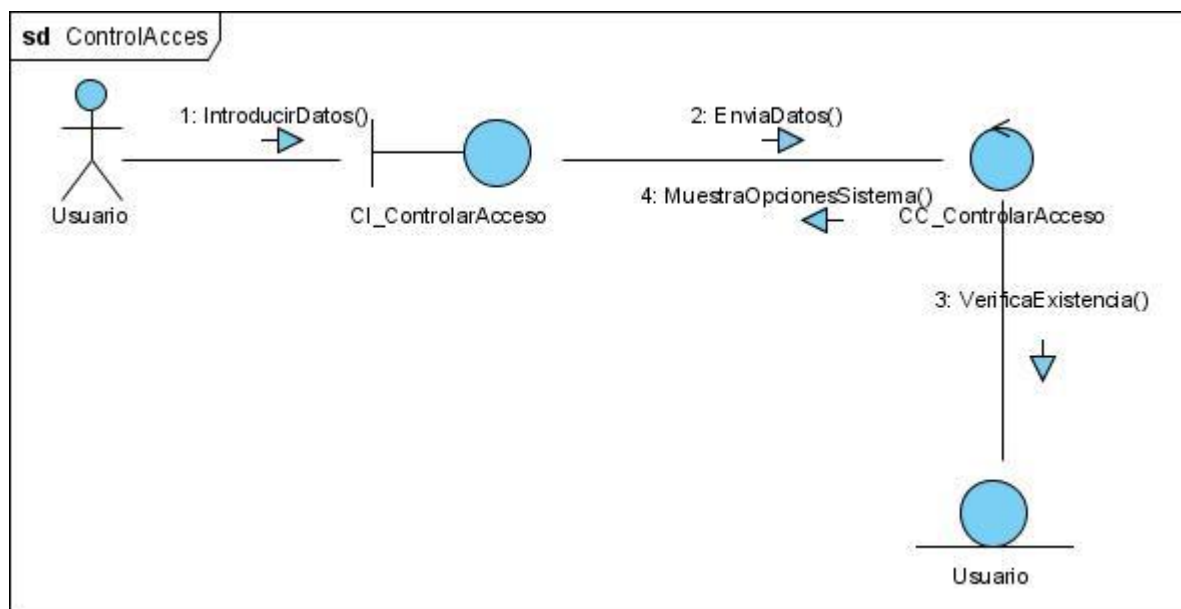


Figura 5. Diagrama de Colaboración de Clase del Análisis, Caso de Uso Controlar Acceso.

Diagramas de Colaboración de Clases del Análisis, Ver Anexos. 15, 16, 17, 18, 19, 20, 21, 22,23,24.

3.3 Diseño.

El diseño constituye una de las fases principales para el desarrollo de todo proyecto, este modela las interacciones de los mensajes, asignándosele responsabilidades a los diferentes objetos. El soporte, mantenimiento y calidad de los componentes reutilizables dependen en gran medida de las decisiones que aquí se tomen, para ello se aplican algunos principios básicos en la toma de las decisiones, los patrones resumen muchos de estos principios.

3.3.1 Clases del Diseño.

Los diagramas de clases del diseño muestran las relaciones entre las distintas clases creadas para la solución del problema. Constituyen un punto de partida para los implementadores a la hora de

construir los objetos y sus instancias. En el trabajo se muestra las principales clases creadas junto a una explicación de su responsabilidad dentro del sistema.

Nombre de la Clase	Responsabilidad
cestud	Es la clase interfaz entre los elementos del estudio y el especialista. Permite enviar las órdenes del especialista, así como verificar los datos insertado por el mismo.
simulador	Es la clase base del estudio, tiene un conjunto de métodos que permiten ejecutar el estudio, así como crearlo y eliminarlo.
generador	Se especializa en generar aleatoriamente los valores de los elementos o atributos del estudio, es muy utilizada ya que este es el principio básico del Método de Monte Carlo Directo.
punto	Clase que gestiona las funcionalidades de cada punto generado con X, Y y Z como atributos.
vectores	Clase que gestiona las funcionalidades de cada velocidad generada con vectores en X, Y y Z como atributos.
acceso_datos	Clase que permite la interacción entre las clases bases, las clases interfaces y la clase para la conexión a la Base de Datos.
conexion	Clase que permite la conexión y ejecución de las consultas con la Base de Datos.
chequeardatos	Clase que permite chequear tipos de datos, mostrar mensajes, rangos de valores de números y longitud de los datos.
variable	Clase que almacena el tipo de usuario y su rol, permite limitar el acceso de la aplicación.

3.4 Arquitectura Agil

A diferencia de lo que proponen las metodologías ágiles aquí se propone una arquitectura candidata en forma temprana, esto es basado en la curva del costo de cambio propuesta por Barry Boehm que indica que el costo de reparar un defecto descubierto en la fase de requerimientos es 200 veces menor que en la fase de mantenimiento.

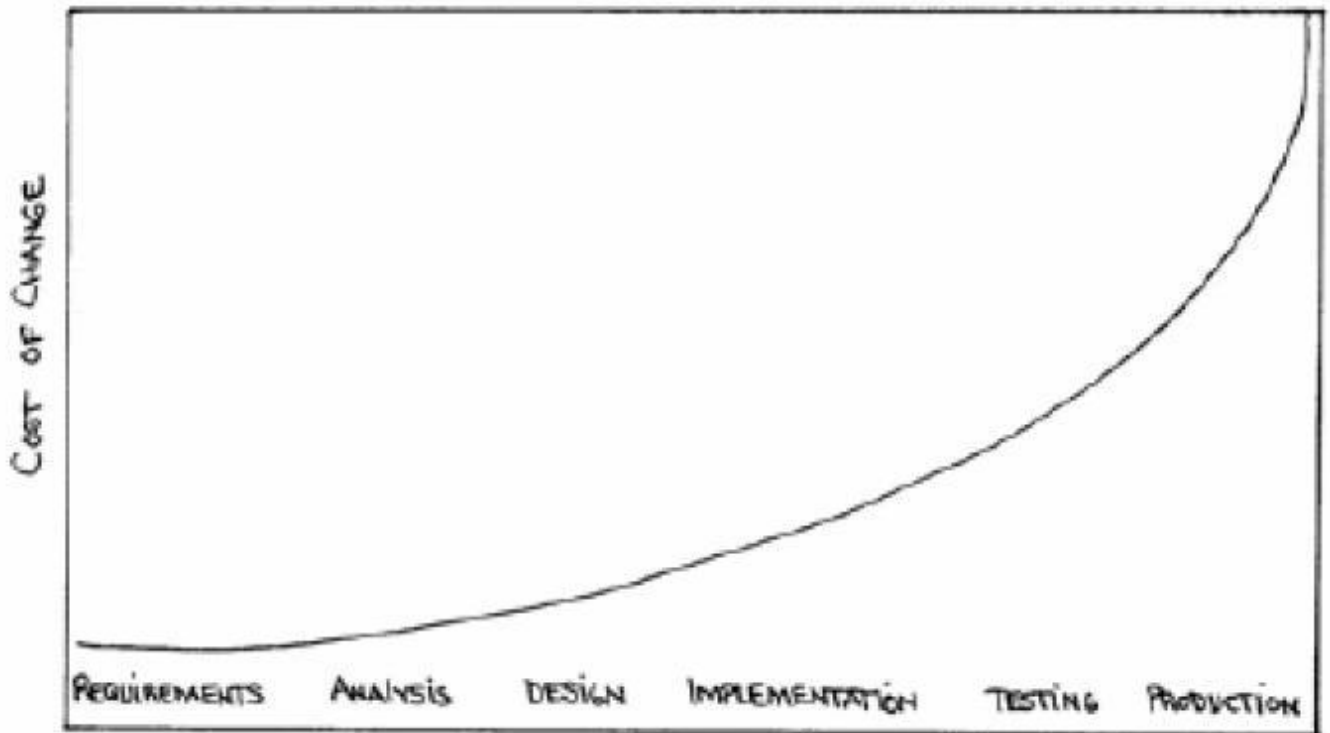


Figura 6. Curva del costo del cambio a los requerimientos en diferentes fases del ciclo de vida.

En la conformación estructuración de la arquitectura se tuvieron en cuenta los algunos de los patrones arquitectónicos de Alexander y los Grasp.

Patrones de Alexander:

1. Máxima Comunicación: Las metodologías ágiles tienen sentido si existe comunicación entre las personas, la comunicación representa las diversas formas de transmisión de datos, información, conocimiento y experiencia entre individuos. La metodología debe proveer la interacción de los desarrolladores, así el flujo de información entre los mismos será máximo.
2. Mínima Dispersión Física: El equipo de desarrollo debe estar en una única oficina que tenga una distribución centralizada.
3. Comunicación Interna del Equipo: Se basa en tres patrones fundamentales unidos entre sí, Mínima Dispersión, Interrupciones Mínimas y Radiadores de Información.
4. Participación Activa del Cliente: Plantea la necesidad de contar con una incondicional participación del cliente durante el proyecto de desarrollo. No hay modelo ni herramienta que pueda sustituir el contacto cara a cara entre el analista y los usuarios del sistema. El no

involucrar al usuario durante el desarrollo de software permite construir un producto de software que sólo cumpla las necesidades de aquellas personas que no van a utilizarlo.

5. Estimaciones Agiles: el objetivo es poder hacer una estimación de la funcionalidad a ser entregada en cada etapa en función de la productividad del equipo de desarrollo.
6. Enfoque en la Arquitectura: Definición temprana de un prototipo de arquitectura, mantiene un cierto nivel de abstracción dentro de la disciplina de diseño, no contempla los detalles de más bajo nivel, simplemente propone un conjunto de vistas que enfatizan distintos aspectos del software.

Patrones Grasp:

1. Experto: Asignar una responsabilidad a la clase que tiene la información necesaria para cumplirla. Si las asignaciones se hacen de forma adecuada los sistemas tienden a ser más fáciles de entender.
2. Creator: Asignarle a la clase B la responsabilidad de crear una instancia de clase A.
3. Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:
 - Sistema global.
 - Empresa u organización global.
 - Manejador artificial de todos los eventos del sistema.
 - Algo del mundo real activo.

La arquitectura ágil consiste en mantener una visión única, formulada por el equipo de proyecto de las decisiones consideradas críticas para el desarrollo, siendo el equipo el encargado de construir, mantener y obtener los beneficios de la misma. La arquitectura no va a ser un documento estático con una estructura determinada a la cual el equipo se deba ajustar, sino que irá evolucionando con el desarrollo de los componentes y será mantenida hasta el punto que contribuya a la comunicación dentro del grupo. Todo es regido por el principio del Proceso Orientado a Personas, el cual describe la necesidad de construir aquellos artefactos indispensables para el proyecto de software, aquellos que sirvan a algún propósito claro.

3.4.1 Diagrama de Despliegue

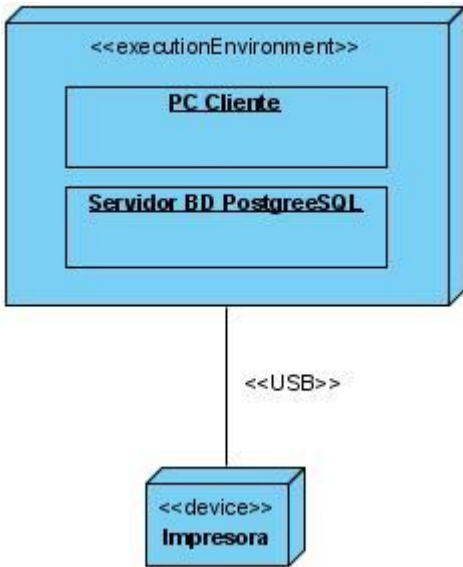


Figura 7. Diagrama de Despliegue

3.4.2 Diagrama de Componentes

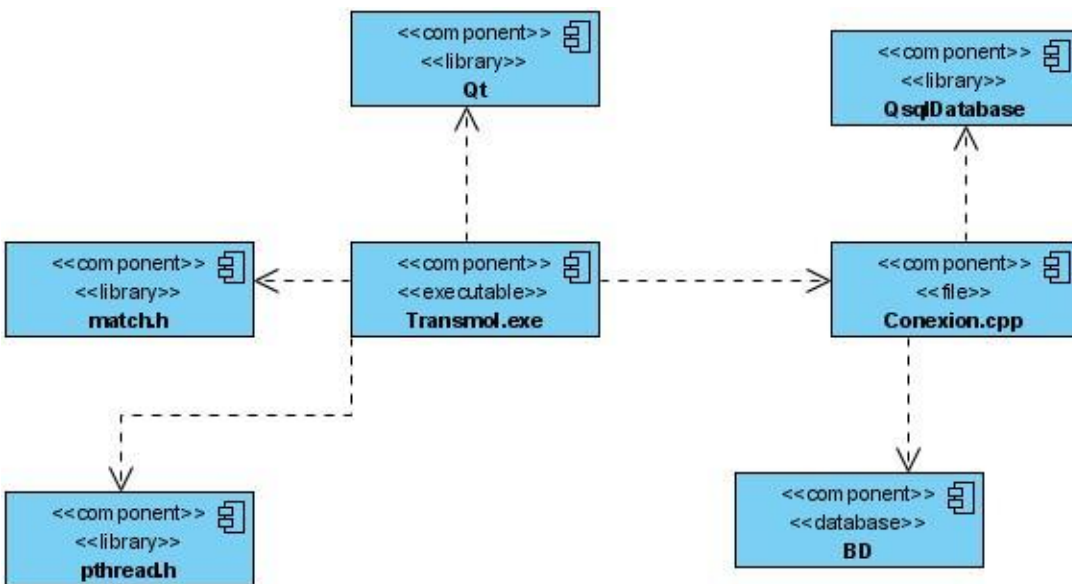


Figura 8. Diagrama de Componentes

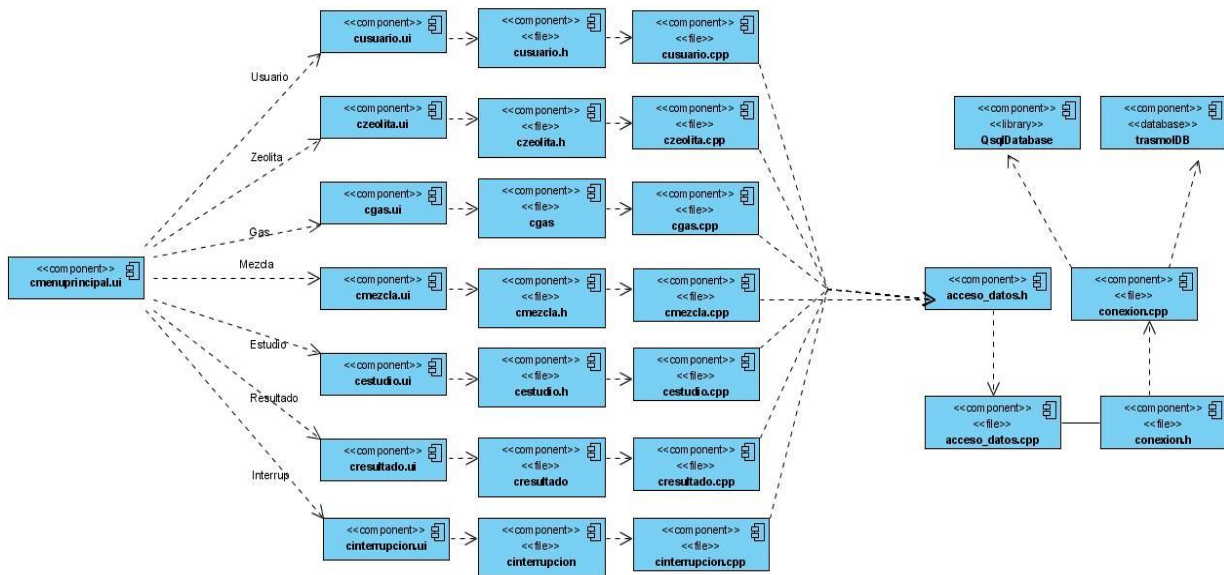


Figura 9. Diagrama de Componentes

3.5 Diseño de la Base de Datos.

En el diseño de la Base de Datos se modela el tratamiento de la información con carácter persistente dentro del sistema.

Varios son los métodos y alternativas para modelar la persistencia de los datos, incluyendo una gran variedad de herramientas de modelado. La propuesta actual es modelar la persistencia de los datos a partir de los diagramas de clase, con herramientas modernas que realizan una traducción del modelo de clases a un modelo de datos relacional.

3.5.1 Diagrama de Clases Persistentes.

Este modelo debe ser usado siempre que se modelen datos persistentes, en sistemas que manipulan gran cantidad de información contenida en medios de almacenamiento persistente.

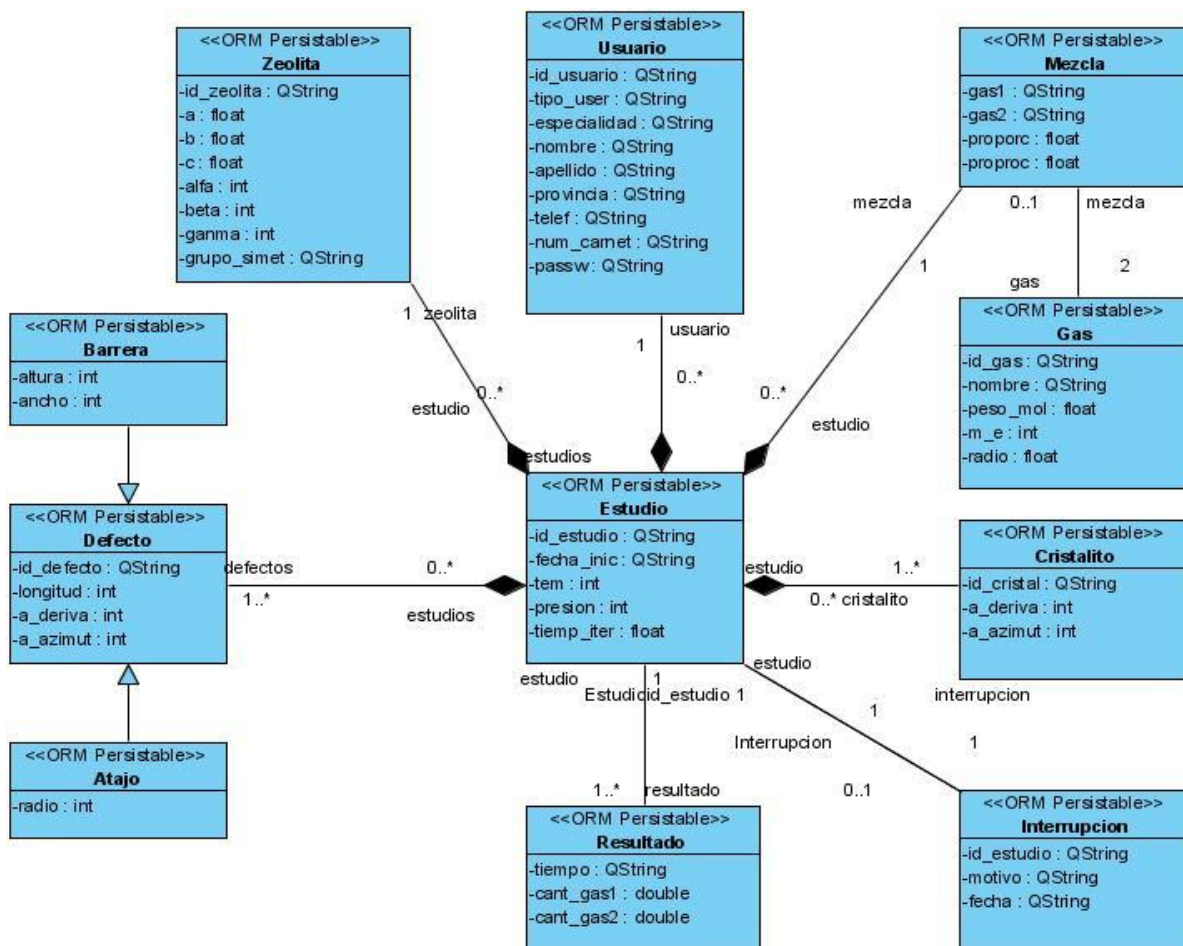


Figura 10. Diagrama de Clases Persistentes

3.5.2 Diagrama Entidad Relación.

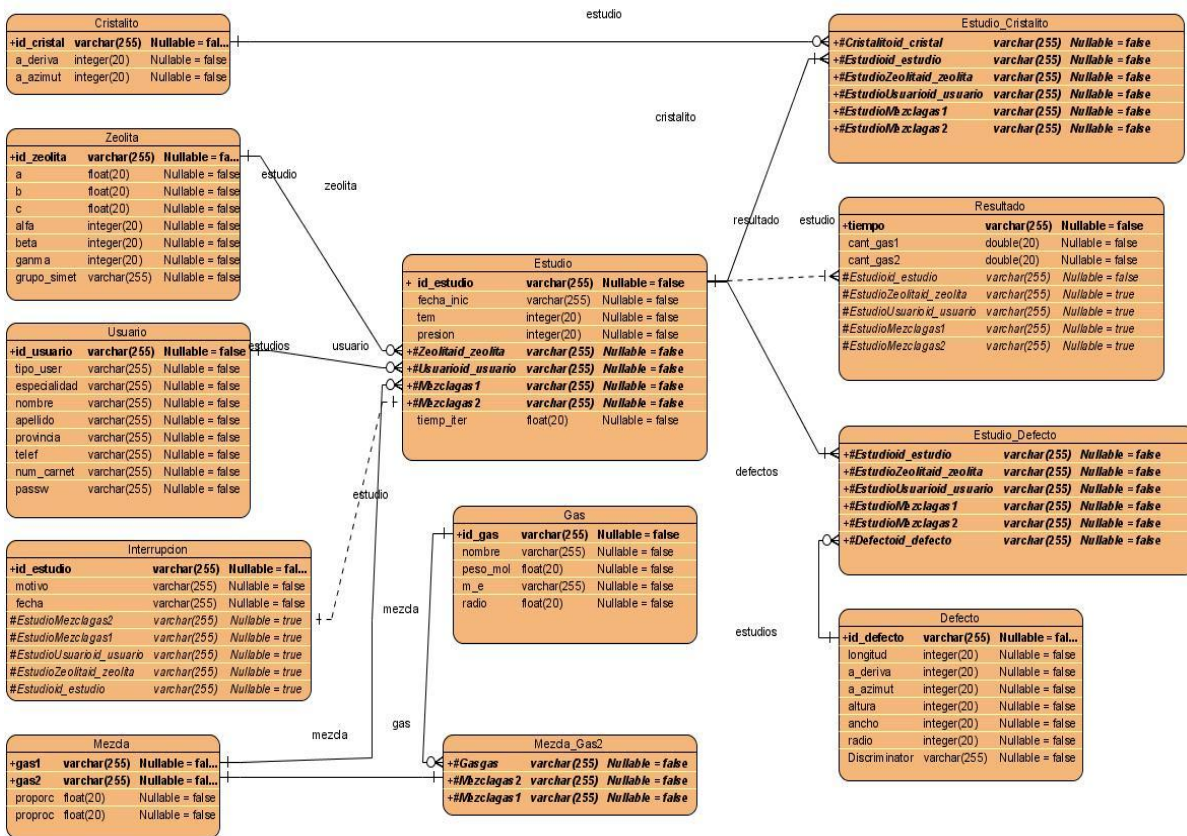


Figura 11. Diagrama Entidad Relación.

3.5.3 Descripción de las Tablas del Modelo Entidad Relación.

Nombre: Usuario		
Descripción: Almacena las características de los usuarios del sistema.		
Atributo	Tipo	Descripción
id_usuario	varchar	El atributo identificador del usuario en la Base de Datos como llave primaria.
tipo_user	varchar	Breve información sobre el rol del usuario en el sistema.

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

especialidad	varchar	Breve información sobre el grado científico del usuario.
nombre	varchar	Nombre del usuario en la Base de Datos.
apellidos	varchar	Apellidos del usuario en la Base de Datos.
provincia	varchar	Provincia del usuario en la Base de Datos.
telef	integer	Teléfono del usuario en la Base de Datos.
num_carnet	integer	Numero de carnet del usuario en la Base de Datos.
passw	varchar	Contraseña del usuario en la Base de Datos.

Nombre: Gas		
Descripción: Esta tabla almacena las características de los Gases.		
Atributo	Tipo	Descripción
id_gas	varchar	El atributo identificador de gas en la Base de Datos como llave primaria.
nombre	varchar	Breve información del nombre del gas.
peso_mol	float	Breve información del peso molecular del gas.
m_e	varchar	Breve información sobre el momento eléctrico.
radio	float	Breve información del radio de las moléculas del gas.

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

Nombre: Zeolita		
Descripción: Esta tabla almacena todas las características estructurales de la Zeolita.		
Atributo	Tipo	Descripción
id_zeolita	varchar	El atributo identificador de la Zeolita.
a	float	Parámetro a de la celda estructural de la zeolita, que corresponde al eje x.
b	float	Parámetro b de la celda estructural de la zeolita, que corresponde al eje y.
c	float	Parámetro c de la celda estructural de la zeolita, que corresponde al eje z.
alfa	int	Angulo de inclinación del eje a con respecto al eje x (altura) de la estructura de la Zeolita.
beta	int	Angulo de inclinación del eje b con respecto al eje y (largo) de la estructura de la Zeolita.
gamma	int	Angulo de inclinación del eje c con respecto al eje z (profundidad) de la estructura de la Zeolita.
grup_simet	varchar	Identifica el grupo de simetría al que pertenece la Zeolita.

Nombre: Defecto		
Descripción: Esta tabla almacena las características del tipo de defecto Barrera.		
Atributo	Tipo	Descripción

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

id_defecto	varchar	El atributo identificador del tipo de defecto barrera en la Base de Datos como llave primaria, representa la coordenada inicial de la ubicación del defecto.
longitud	int	Identifica la longitud del defecto.
altura	int	Identifica la altura del defecto.
ancho	int	Identifica el ancho del defecto.
radio	int	Identifica el radio del defecto.
a_deriva	int	Angulo de inclinación con respecto al plano xy del defecto.
a_azimut	int	Angulo de inclinación con respecto al eje z del defecto.
Discriminator	varchar	Para referir de que tipo es el defecto.

Nombre: Estudio_Defecto		
Descripción: Esta tabla almacena las características de la relación entre el defecto y el estudio.		
Atributo	Tipo	Descripción
Estudioid_estudio	varchar	El atributo identificador del identificador del estudio en la Base de Datos como llave foránea.
EstudioZeolitaid_zeolita	varchar	El atributo identificador del identificador de la zeolita en el estudio en la Base de Datos como llave foránea.
EstudioUsuarioid_usuario	varchar	El atributo identificador del identificador del usuario en el

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

		estudio en la Base de Datos como llave foránea.
EstudioMezclagas1	varchar	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.
EstudioMezclagas2	varchar	El atributo identificador del identificador del gas2 en el estudio en la Base de Datos como llave foránea.
Defectoid_defecto	varchar	El atributo identificador del identificador del defecto en el estudio en la Base de Datos como llave foránea.

Nombre: Estudio_Cristalito		
Descripción: Esta tabla almacena las características de la relación entre el cristalito y el estudio.		
Atributo	Tipo	Descripción
Cristalitoid_cristal	varchar	El atributo identificador del identificador del estudio en la Base de Datos como llave foránea.
Estudioid_estudio	varchar	El atributo identificador del identificador de la zeolita en el estudio en la Base de Datos como llave foránea.
EstudioZeolitaid_zeolita	varchar	El atributo identificador del identificador del usuario en el estudio en la Base de Datos como llave foránea.

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

EstudioUsuarioid_usaurio	varchar	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.
EstudioMezclagas1	varchar	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.
EstudioMezclagas2	varchar	El atributo identificador del identificador del gas2 en el estudio en la Base de Datos como llave foránea.

Nombre: Cristalito		
Descripción: Esta tabla almacena las características estructurales del Cristalito.		
Atributo	Tipo	Descripción
id_cristal	varchar	El atributo identificador del estudio al cual se vinculara la microcolumna como llave primaria.
a_deriva	int	Coseno del ángulo de inclinación con respecto al plano xy de las láminas de la cristalita.
a_azimut	int	Coseno del ángulo de inclinación con respecto al eje z de las láminas de la cristalita.

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

Nombre: Resultado		
Descripción: Esta tabla almacena las características del resultado.		
Atributo	Tipo	Descripción
tiempo	varchar	El atributo identificador tiempo del resultado como llave primaria en la Base de Datos.
cant_gas1	int	Cantidad de moléculas del gas1 que atravesaron el estrato.
cant_gas2	int	Cantidad de moléculas del gas2 que atravesaron el estrato.
Estudioid_estudio	varchar	El atributo identificador del identificador de la zeolita en el estudio en la Base de Datos como llave foránea.
EstudioZeolitaid_zeolita	varchar	El atributo identificador del identificador del usuario en el estudio en la Base de Datos como llave foránea.
EstudioUsuarioid_usuario	varchar	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.
EstudioMezclagas1	varchar	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.

Nombre: Interrupción		
Descripción: Esta tabla almacena las características de la interrupción.		
Atributo	Tipo	Descripción

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

id_estudio	varchar	El atributo identificador de la interrupción como llave primaria en la Base de Datos.
motivo	varchar	Breve información del motivo de la interrupción del estudio.
fecha	varchar	Breve información de la fecha de la interrupción del estudio.
Estudioid_estudio	varchar	El atributo identificador del identificador de la zeolita en el estudio en la Base de Datos como llave foránea.
EstudioZeolitaid_zeolita	varchar	El atributo identificador del identificador del usuario en el estudio en la Base de Datos como llave foránea.
EstudioUsuarioid_usaurio	varchar	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.
EstudioMezclagas1	varchar	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.

Nombre: Mezcla		
Descripción: Esta tabla almacena las características de la Mezcla.		
Atributo	Tipo	Descripción
id_gas	varchar	El atributo identificador del primer gas como llave primaria en la Base de Datos.

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

id_gas2	varchar	El atributo identificador del segundo gas como llave primaria en la Base de Datos.
proporc_gas1	float	Identifica la proporción a la que está el primer gas en la mezcla.
proporc_gas2	float	Identifica la proporción a la que está el segundo gas en la mezcla.

Nombre: Mezcla_Gas2		
Descripción: Esta tabla almacena las características de la Mezcla.		
Atributo	Tipo	Descripción
GasGas		El atributo identificador del primer Mezcla como llave primaria en la Base de Datos.
Mezclagas1	varchar	El atributo identificador como llave foránea que representa el gas1 de la mezcla.
Mezclagas2	varchar	El atributo identificador como llave foránea que representa el gas2 de la mezcla.

Nombre: Estudio		
Descripción: Esta tabla almacena las características del Estudio.		
Atributo	Tipo	Descripción
id_estudio	varchar	El atributo identificador del estudio como llave primaria en la Base de Datos.
fecha_inicio	varchar	Identifica la fecha de comienzo

		del estudio.
temperatura	int	Identifica la temperatura a la que se va a realizar el estudio.
presión	int	Identifica la presión a la que se va a realizar el estudio.
Tiempo_iter	int	Identifica la concentración a la que se va a realizar el estudio.
EstudioZeolitaid_zeolita	int	El atributo identificador del identificador del usuario en el estudio en la Base de Datos como llave foránea.
EstudioUsuarioid_usuario	int	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.
EstudioMezclagas1	int	El atributo identificador del identificador del gas1 en el estudio en la Base de Datos como llave foránea.
EstudioMezclagas2	int	El atributo identificador del identificador del gas2 en el estudio en la Base de Datos como llave foránea.

3.6 Pruebas

Las pruebas de software son la constancia de que el producto obtenido funciona según los requisitos del cliente. Estas pueden clasificarse en varios tipos, de verificación, validación, cubrimiento, condiciones, etc. Aquí se muestran algunas pruebas realizadas y documentadas durante el desarrollo del sistema.

3.6.1 Pruebas Unitarias

Las pruebas unitarias verifican el correcto funcionamiento de un modulo o método de un sistema. Debido a la metodología utilizada es que se realizaron este tipo de pruebas para verificar la funcionalidad principal del sistema, para ello se escogieron básicamente cinco métodos, los cuales constituyen ramificaciones de la estructura principal.

CAPITULO III: IMPLEMENTACION DE LA PROPUESTA DE SOLUCION

Prueba	Método	Valores de Entrada	Pertenece (clase)	Valor Esperado	Obs
1	generarcant1 (a,p) a->proporción de área libre p->proporción del gas en la mezcla.	Temperatura = 300 (K) Presión = 100000 (Pa) proporcion1 = 0.6 proporcion2 = 0.4	generador	Satisfactorio. cant1 = 1664 cant2 = 1109	Esta prueba se realizó para varios valores dando positivos los resultados.
2	gen_velocidad()	Temperatura = 300 (K) Presión = 100000 (Pa) proporcion1 = 0.6 proporcion2 = 0.4	generador	Satisfactorio. Se generaron los valores de la velocidad para cada una de las moléculas	En esta prueba se utilizaron los mismos valores de la prueba anterior para validando el correcto funcionamiento.
3	ubicar_mol()	Temperatura = 300 (K) Presión = 100000 (Pa) proporcion1 = 0.6 proporcion2 = 0.4	simulador	Satisfactorio. Se ubicaron todas las moléculas: Tipo uno: 1664 Tipo dos: 1109	Se realizó para una corrida inicial, utilizando los mismos valores de las pruebas anteriores para la configuración.
3	mover_mole()	Temperatura = 300 (K) Presión = 100000 (Pa) proporcion1 = 0.6 proporcion2 = 0.4	simulador	Insatisfactorio. No se chequeaban los puntos de análisis de la función espacio por lo que generaba valores fuera del rango de la microcolumna.	Este error fue corregido, ejecutándose posteriormente sin problemas.
4	choque_pared()	Vinic: x= -576.134053 y= 289.472931 z= 576.134033 Vector unitario: x= 0 y= 0 z= 1 punto avance: x= 156496 y= 152000 z= 1152	simulador	Satisfactorio La nueva velocidad final de la molécula fue: x= -576.134053 y= 289.472931 z= 1728.4021	Se chequearon manualmente diez choques, para corroborar los resultados.

5	avance()	Tiempo iter: 10 punto: x = 9200 y=145033 z=0 velocidad: x=141.9037 y=-163.1942 z=576.1340	simulador	Satisfactorio. La nueva posición del centro de la molécula fue: punto: x=10619 y=143401 z=5761	
---	----------	---	-----------	--	--

3.7 Conclusiones

En este capítulo se abarcó lo perteneciente a las vistas estáticas y de implementación correspondiente a la notación UML. De igual manera fueron señalados los contenidos que sobre los principios de diseño debían añadirse a los requisitos funcionales mencionados en el capítulo anterior, para de esta forma completar el análisis realizado.

Con este capítulo culmina la modelación completa de la primera versión del Programa para la simulación del proceso de difusión molecular en láminas delgadas y membranas de zeolitas, por método de Monte Carlo Directo, haciéndose un grupo de pruebas que validan el funcionamiento del mismo en su versión inicial.

Conclusiones:

Con este trabajo se hace una propuesta de solución a los problemas existentes en la simulación de la difusión de gases en membranas zeolíticas reales, compuestas de múltiples cristalitas de zeolitas de diferentes orientaciones espaciales, unidas entre sí, pero con numerosos defectos de tipo barrera y de tipo atajo, con la que se logrará una vía más fácil, rápida y precisa en el procesamiento de la difusión de los gases.

Por lo que se investigó de forma exhaustiva los procesos que se llevan a cabo para realizar la simulación del proceso de difusión molecular en láminas delgadas y membranas de zeolitas, por método de Monte Carlo Directo, áreas de conocimiento de ingeniería de software, metodologías ágiles y soluciones prácticas de desarrollo.

La propuesta del sistema se desarrolló siguiendo la unión de dos metodologías (RUP y DSDM), además se utilizaron representaciones para la modelación de todas las fases del proyecto. El sistema resultante está provisto de un ambiente cómodo, fácil de entender. Por todo lo expuesto anteriormente se concluye que se cumplieron los objetivos trazados.

Se incluyen una serie de recomendaciones a futuras iteraciones.

Recomendaciones:

Sin duda no es éste un trabajo estático; los modelos y la solución que se propone habrán de revisarse y actualizarse periódicamente para reflejar los cambios, que podrían garantizar sin duda mayor calidad y eficiencia. Ha quedado claro que la propuesta puede ser mucho más ambiciosa. Por ello se recomienda lo siguiente:

- Diseñar la aplicación para programación multihilo.
- Visualizar el proceso en 3D.
- Además de modelar el problema para un conjunto de máquinas en las que se pueda trabajar simultáneamente.

Bibliografía:**Citada:**

- [1] JIMÉNEZ, D. P. Y. M. M. Zeolita. ¿Donde está el mineral del siglo? Juventud Rebelde. , 2007.
- [2] SOURCEFORGE.NET. 1999. [2009]. Disponible en: <http://sourceforge.net/>
- [3] JAMES RUMBAUGH, I. J. Y. G. B. B. El Lenguaje Unificado de Modelado. Manual de Referencia. . p. Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>
- [4] IVAR JACOBSON, G. B. Y. J. R. El Proceso Unificado de Desarrollo de Software. La Habana. La Habana, 2004. p. Félix Varela.
- [5] Dynamic Systems Development Method. 2009. 15. Disponible en: <http://www.scribd.com/doc/14529506/Dynamic-Systems-Development-Method?autodown=pdf>
- [6] Paradigma de programación. 2008.
- [7] Programación en C. 2009.
- [8] El lenguaje C++.
- [9] EL Lenguaje Java.
- [10] Qt Creator: FAQ Disponible en: <http://www.qtsoftware.com/include/qt-4.5-staging-area/depreciated/qt-creator/faq>
- [11] Introducción a los sistema de Base de Datos y los SGBD, Jesús Tramullas y Kronos, 2000. [http://www.tramullas.com/documatica/2-4.html]
- [12] PRESSMAN, R. S. Ingeniería de software un enfoque VARELA, F., 2005. 26.

Consultada:

- [*] KRISHNA R. Standart. G.L. A multicomponent film model incorporating a general matrix method of solution to the Maxwell-Stefan equations. s.l: 1976. . p. AIChE J.
- [*] PASCHEK D., K. R. Kinetic Monte Carlo simulations of transport diffusivities of binary mixtures in zeolites. Phys. Chem. 2001. 2001. p.
- [*] DIFFUSION OF BINARY MIXTURES IN ZEOLITES: KINETIC MONTE CARLO VERSUS MOLECULAR DYNAMICS SIMULATIONS. Langmuir. 2001. p.
- [*] R, K. Predicting transport difusivities of binary mixtures in zeolites, Chem. Phys. Lett. . 2002. p.
- [*] SANCHEZ, M. A. M. Metodologías de desarrollo de software, 2004.
- [*] VAN BATEN J.M., K. R. ntropy effects in adsorption and diffusion of alkane isomers in mordenite: An investigation using CBMC and MD simulations, Microp. Mesop. Mater. . 2005. p.

-
- [*] KRISHNA R., V. B. J. M., GARCÍA-PÉREZ E., AND CALERO S. Incorporating the Loading Dependence of the Maxwell-Stefan Diffusivity in the Modeling of CH₄ and CO₂ Permeation Across Zeolite Membranes, Ind. Eng. Chem. Research. . 2007. p.
- [*] K., P. S. C. Y. N. H. Tutorial de UML 2008.
- [*] Ingeniería de Software. Proceso Unificado. . 2002.
- [*] FONTES, G. A. Y. A. D. B. Desarrollo Código Seguro, 2004.
- [*] Historia del Lenguaje C. Disponible en: <http://www.ib.cnea.gov.ar/~icom/CursoC/historia.shtml>
- [*] CODAZZI, A. Sistema de Gestión de Bases de Datos, 2004.
- [*] PENADÉS, P. L. Y. M. C. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), 2008. 8. Disponible en: <http://www.scribd.com/doc/8530565/Metodolgias-agiles>
- [*] MOSCARDÓ, J. M. DSDM Dynamic System Development Method 2002: 20.
- [*] CORTE, V. D. SOBRE LIBRERIAS Qt: 40. Disponible en: <http://www.elai.upm.es/spain/Investiga/GCII/personal/vcorte/informeqt.PDF>
- [*] Expendedora: Modelo del Dominio. Disponible en: http://iie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_clase2.pdf
- [*] La ley de distribución de las velocidades moleculares. Física Estadística y Termodinámica. Disponible en: <http://www.sc.ehu.es/sbweb/fisica/estadistica/maxwell/maxwell.html>

Glosario de Términos:

Atajos: Son defectos volumétricos presentes en la membrana de la zeolita.

Barreras: Son defectos volumétricos presentes en la membrana de la zeolita.

Ca: Símbolo químico del Calcio.

Cantidad de Movimiento: Producto de la masa por la velocidad.

Case: Computer Aided Software Engineering.

Cationes Intercambiables: Iones con carga positiva, que compensan la carga negativa del aluminio.

Choque Elástico: Choque entre dos partículas donde no existe pérdida de energía.

Difusión Molecular: Proceso mediante el cual las moléculas de una sustancia fluyen a través de un medio.

DSDM: Dynamic Systems Development Method.

Estocásticos: Aleatorio, casual.

Gestionar Estudio: Se refiere al análisis que puede realizar un especialista para analizar la incidencia de una mezcla determinada o zeolita en el proceso de difusión.

ICIMAF: Instituto de Cibernética, Matemática y Física (ICIMAF) fundado en 1964, es un Instituto de Investigaciones de alto nivel teóricas y aplicadas en ramas de la Cibernética, la Matemática y la Física.

Intergranulares: Espacio entre dos granos.

K: Símbolo químico del Potasio.

Na: Símbolo químico del Sodio.

Policristalinos: materiales cristalinos con diferentes orientaciones.

PSA: Pressure Swing Adsorption, método de separación de gases por adsorción en zeolitas, las presiones de trabajo son ambiental y a una máxima.

RAD: Rapid Application Development.

Sistemas Percoladores Adsorbentes Porosos: Sistemas en los que existen sitios de absorción a los cuales acceden los átomos y moléculas mediante procesos de difusión.

Sitio de Adsorción: Localización dentro de la estructura de un material donde se pueden fijar temporal o permanentemente las moléculas.

TDA: Tipo de dato abstracto, ejemplo listas y arboles.

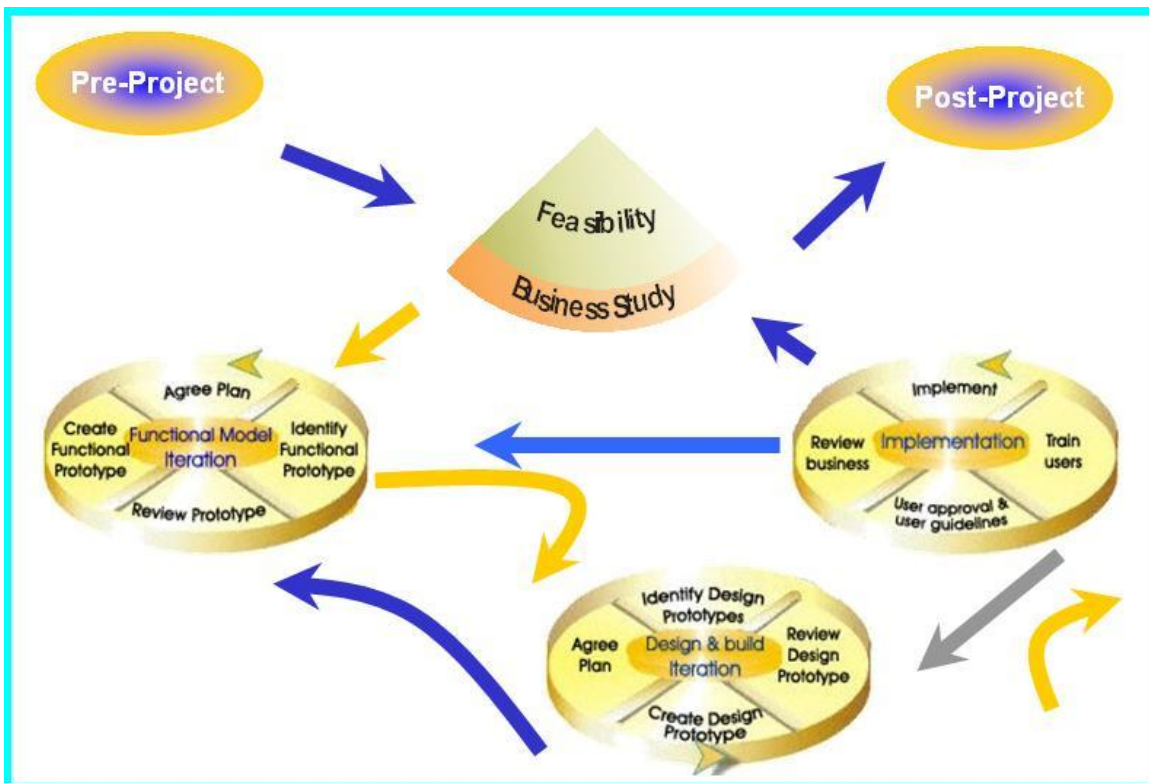
VSA: Vacuum Swing Adsorption, método de separación de gases por adsorción en zeolitas, las presiones de trabajo son vacío y una máxima.

XP: eXtreme Programming, metodología de desarrollo rápido.

Anexos



Anexo 1



Anexo 2.

Cálculo de las velocidades aleatorias según la distribución de Maxwell

a distribución de Maxwell es $F(v) = 4\pi \left(\frac{PM}{2\pi RT}\right)^{3/2} v^2 e^{-\frac{PM}{2RT}v^2}$

Entonces: $F(v)e^{\frac{PMv^2}{2RT}} = 4\pi \left(\frac{PM}{2RT}v^2\right) \left(\frac{PM}{2\pi RT}\right)^{3/2}$

ó $F(v)e^x = \frac{8\pi}{(2\pi)^{3/2}} x \sqrt{\frac{PM}{2RT}}$ donde $x = \frac{PM}{2RT}v^2 \rightarrow$

$$v = \sqrt{\frac{2RTx}{PM}} \quad (1)$$

$$\therefore F(v)e^{-x} = \frac{4N_A p}{RT} x \sqrt{\frac{PM}{2RT}}$$

ó

$$F(v) \frac{e^x}{x} = Q \quad \text{donde} \quad Q = 4 \sqrt{\frac{PM}{2\pi RT}} \quad (2)$$

En particular, para $x=1$ $F=F_{max}$, y entonces

$$F_{max} \frac{e}{1} = Q \rightarrow F_{max} = Q/e \rightarrow F(v) \frac{e^x}{x} = eF_{max} \rightarrow \frac{e^x}{x} = e \frac{F_{max}}{F} \rightarrow$$

$$\frac{e^{x-1}}{x} = \frac{F_{max}}{F} \quad (3)$$

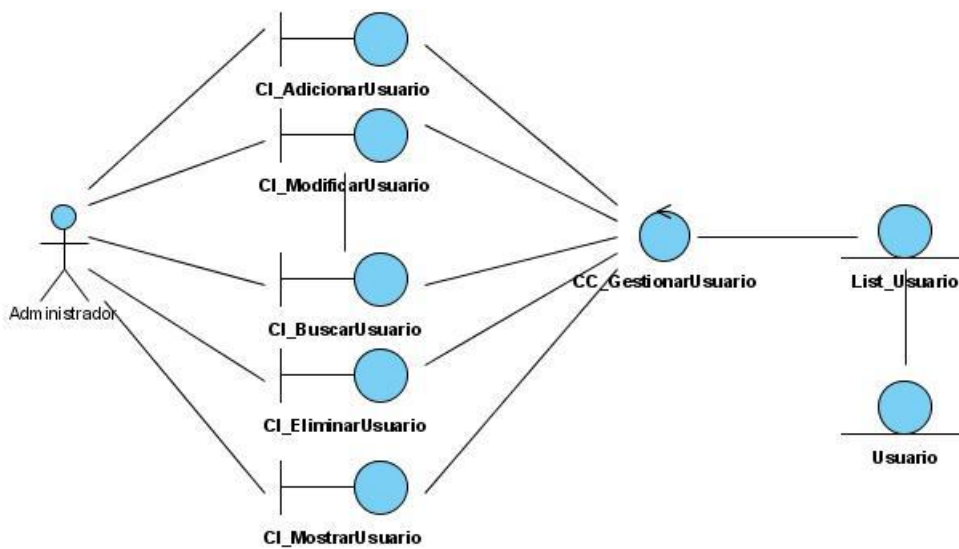
Tabla provatoria:

Nº aleatorio	0.98-1.00	0.94-0.97	0.89-0.93	0.82-0.88	0.73-0.87	0.62-0.72	0.49-0.61	0.34-0.48	0.18-0.33	0-0.17
$F(v)/F_{max}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$F_{max}/F(v)$	10	5	3.33	2.5	2	1.667	1.428	1.250	1.111	1.0
$x > 1$	4.9	4.0	3.4	3.0	2.7	2.4	2.1	1.8	1.5	1.0
$x < 1$	0.04	0.08	0.13	0.17	0.23	0.30	0.38	0.48	0.60	1.0
$v(x \geq 1)$	874	789	728	684	649	611	572	530	484	395
$v(x < 1)$	79	112	142	163	189	216	243	274	306	-

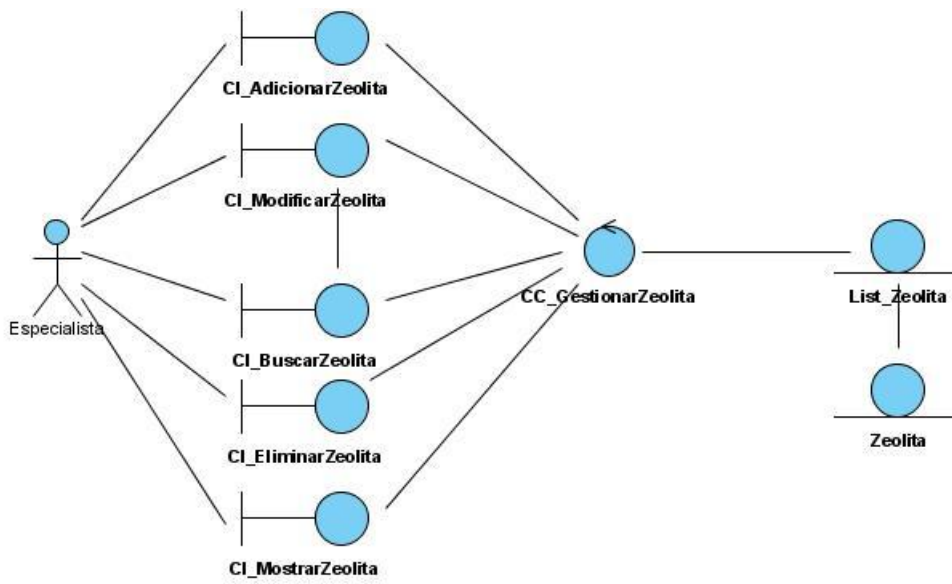
Anexo 3

CU/R	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
CU-1	X											
CU-2		X										
CU-3			X									
CU-4				X								
CU-5					X							
CU-6						X				X		
CU-7							X					
CU-8								X				
CU-9									X			
CU-10											X	
CU-11												X

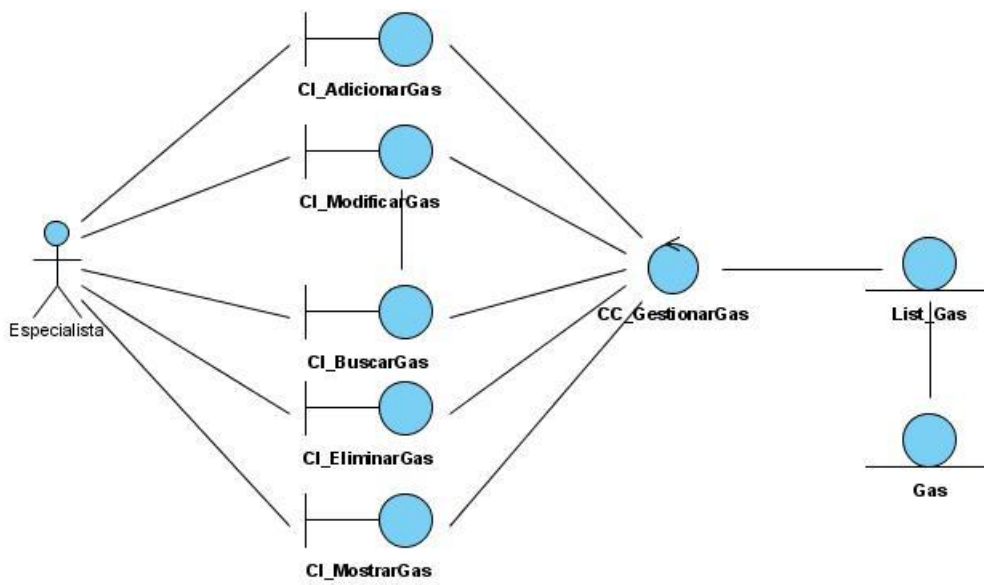
Anexo 4



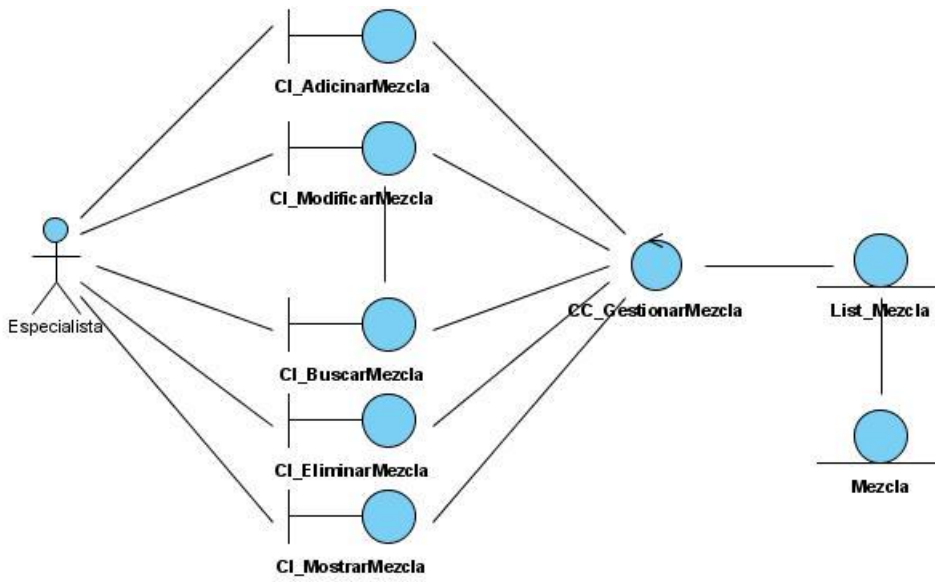
Anexo 5



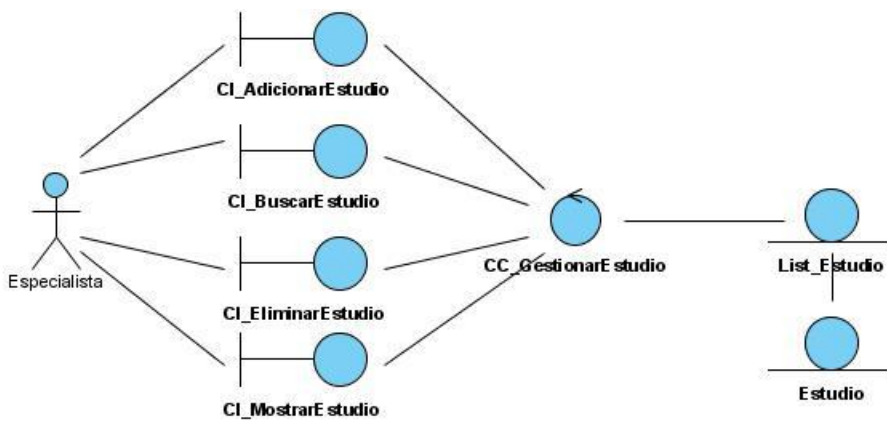
Anexo 6



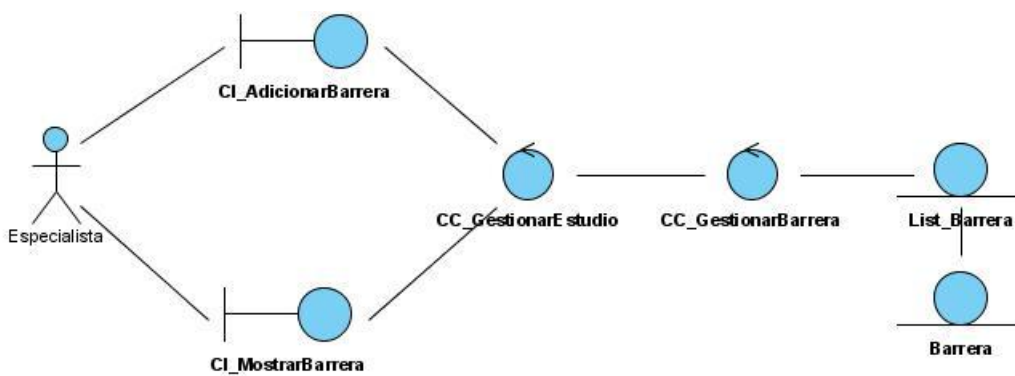
Anexo 7



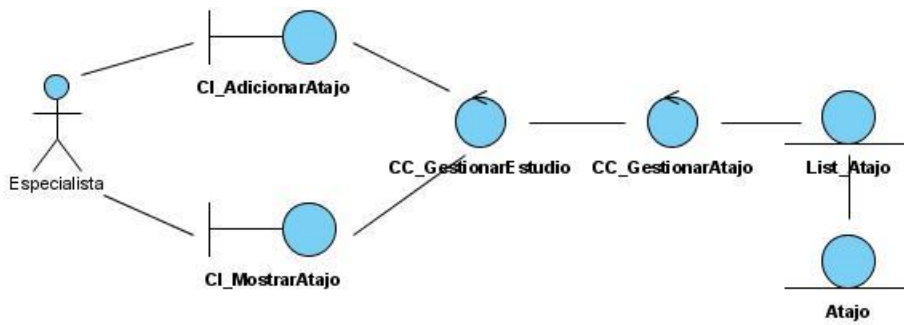
Anexo 8



Anexo 9



Anexo 10



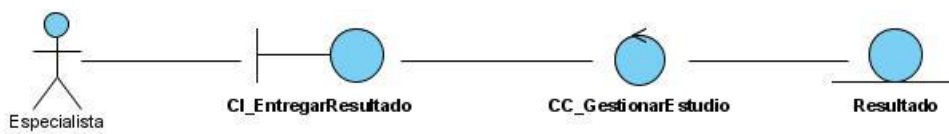
Anexo 11



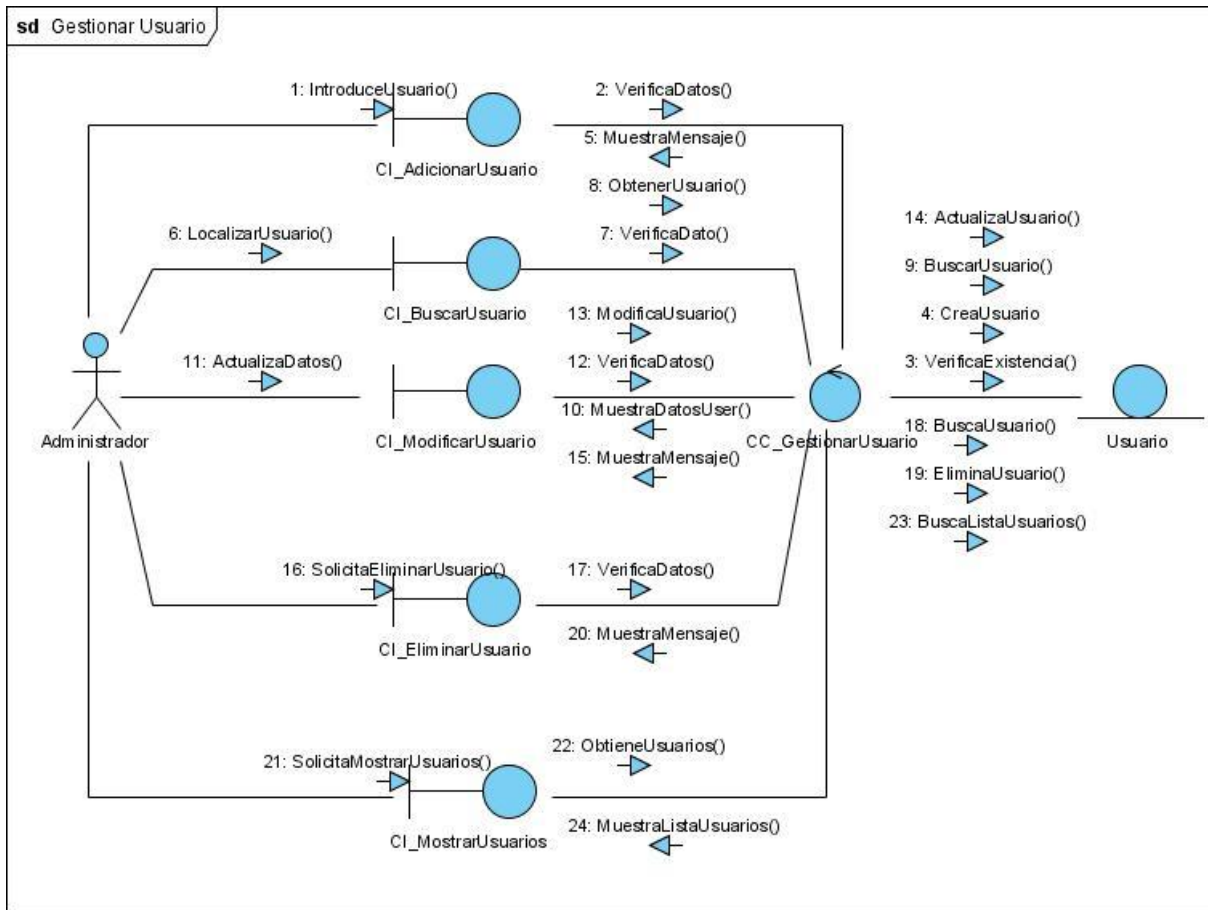
Anexo 12



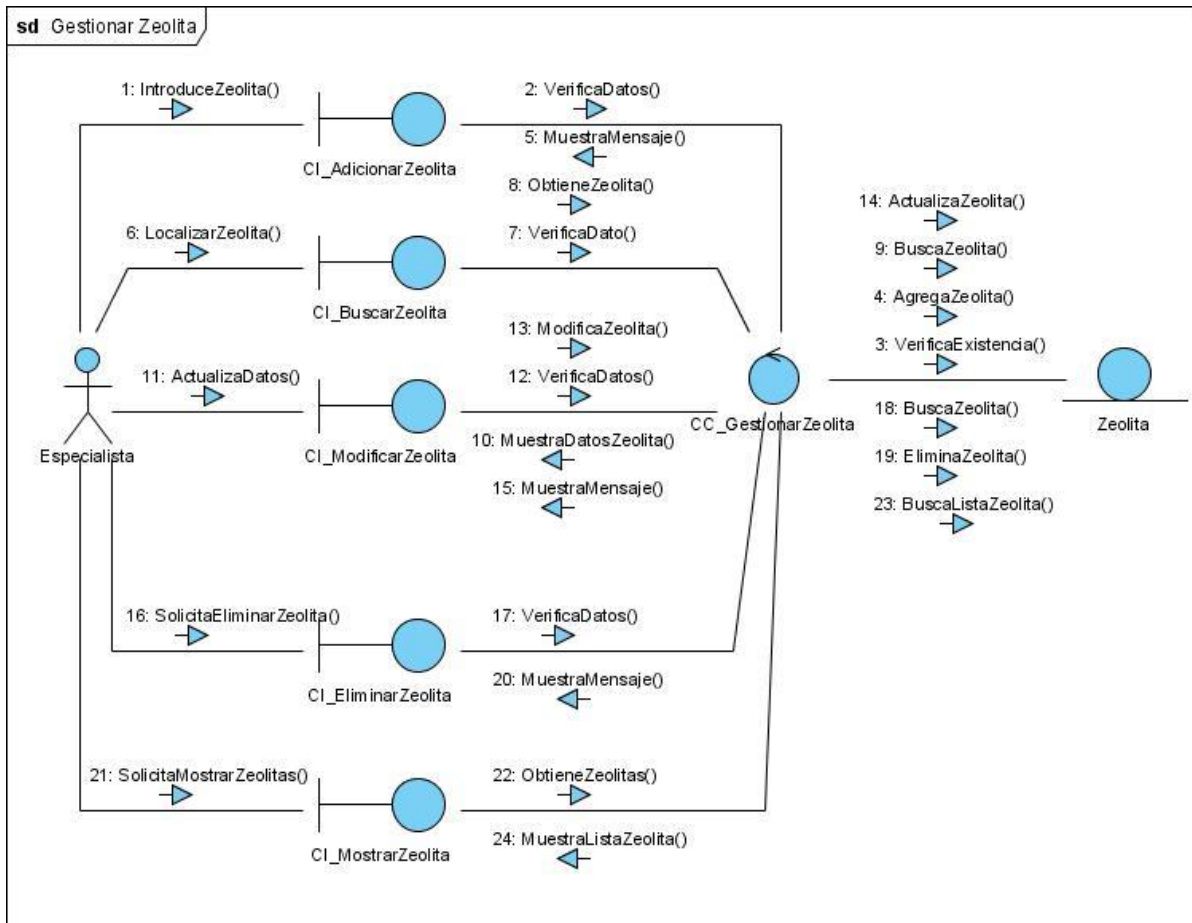
Anexo 13



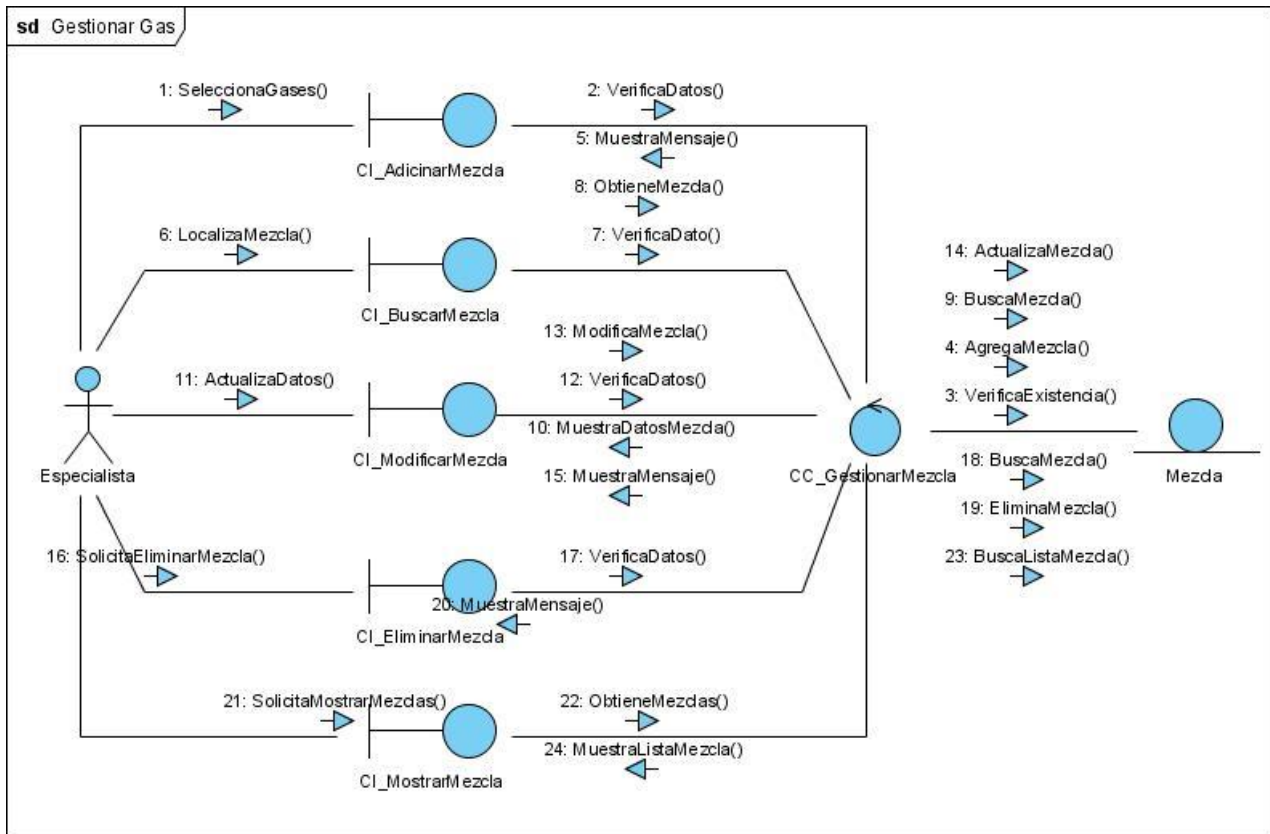
Anexo 14



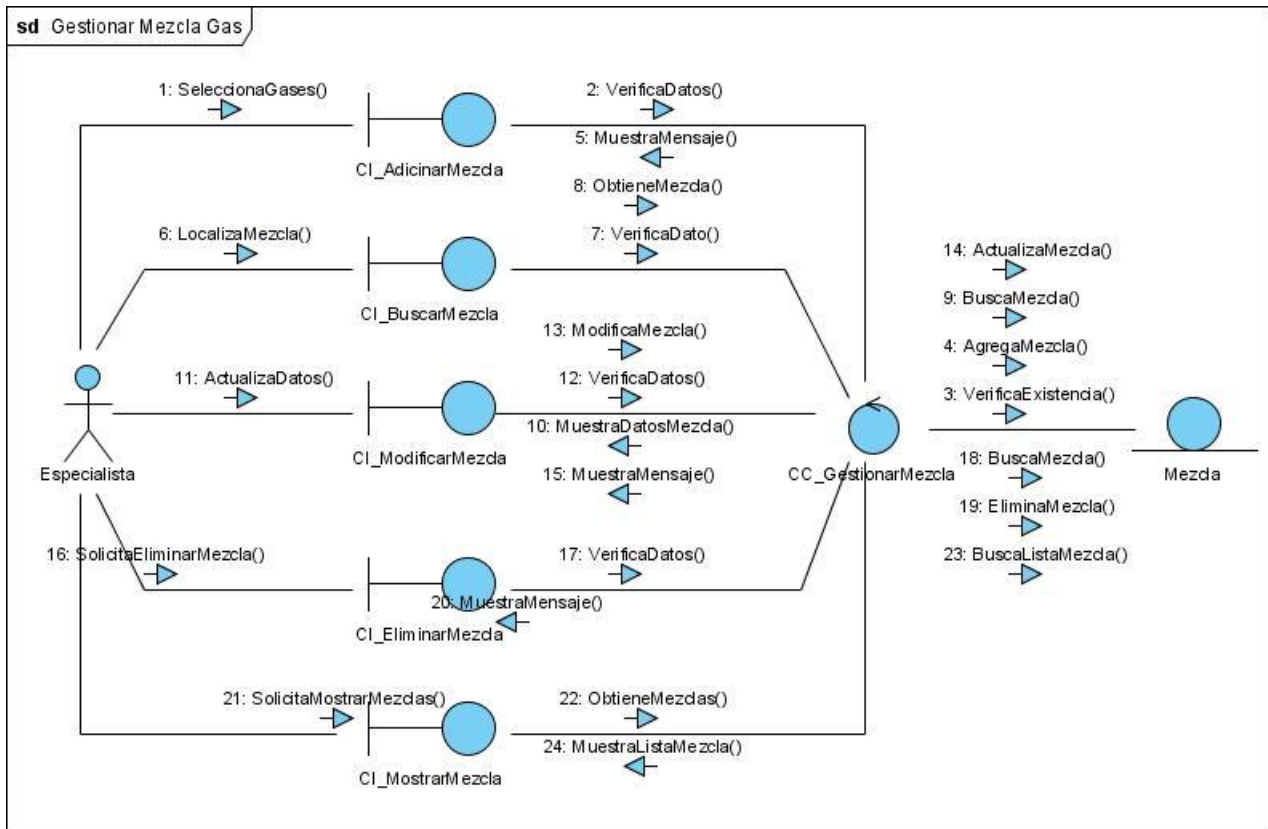
Anexo 15



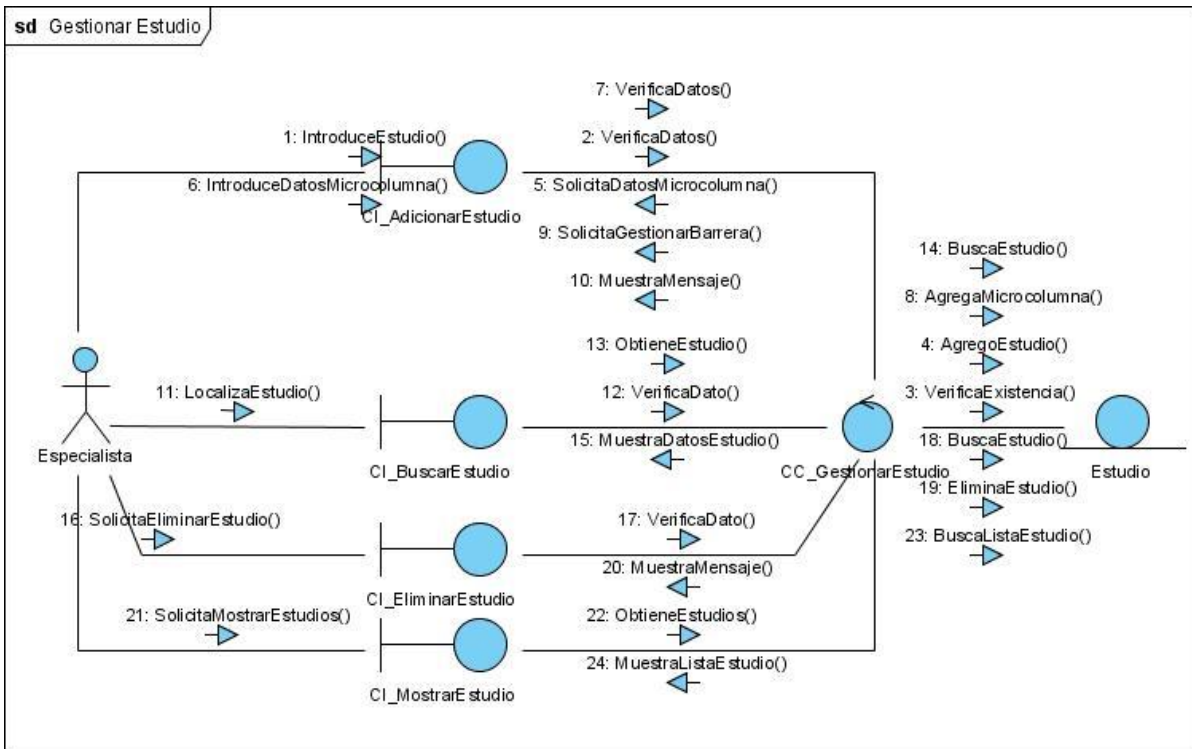
Anexo 16



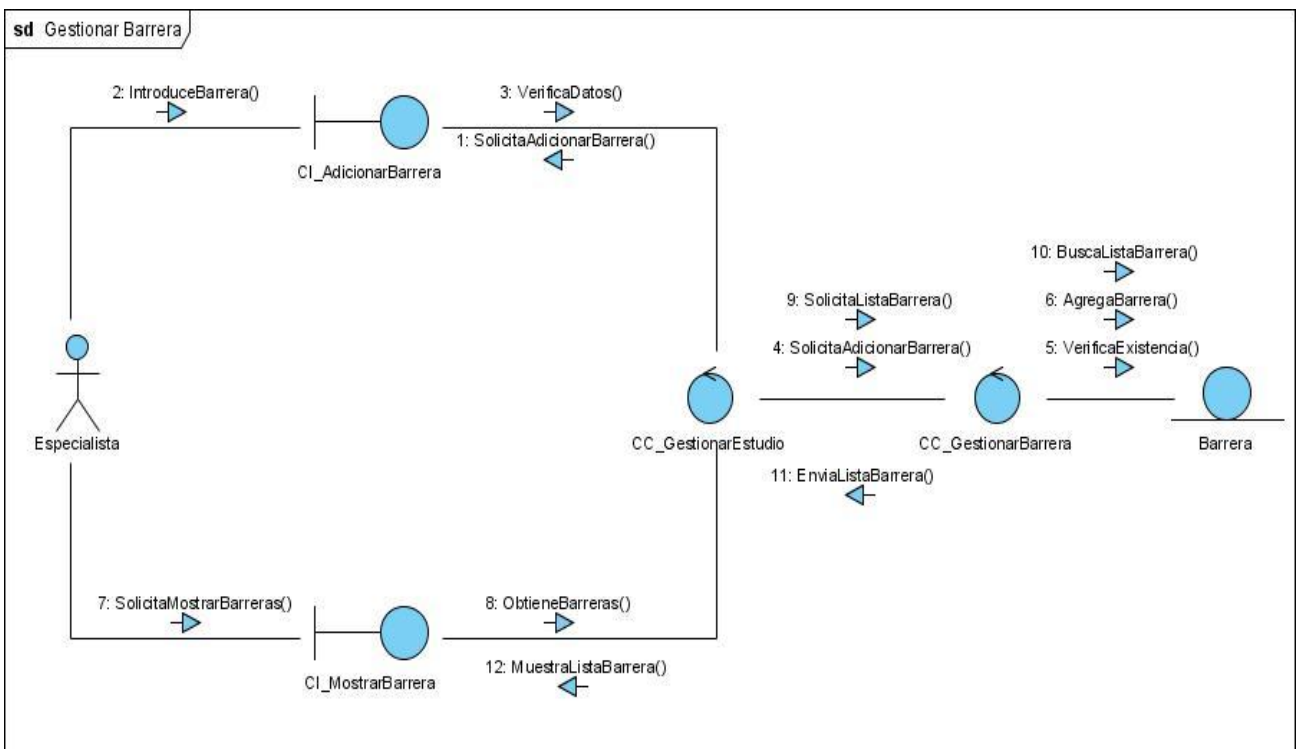
Anexo 17



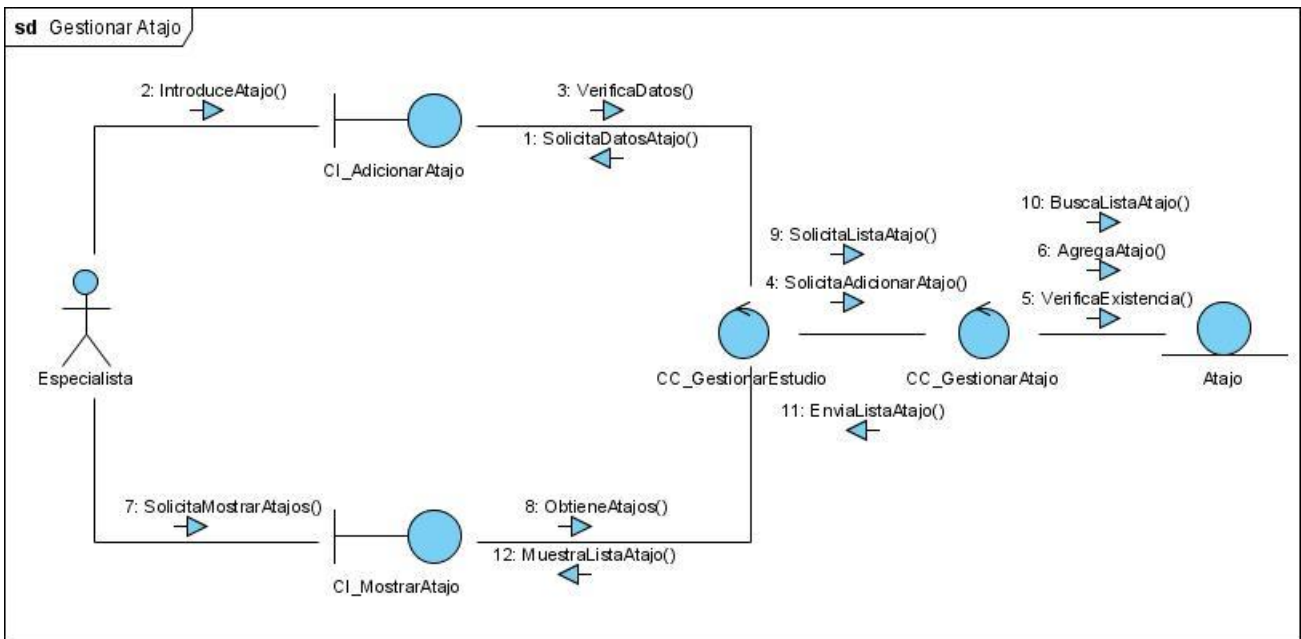
Anexo 18



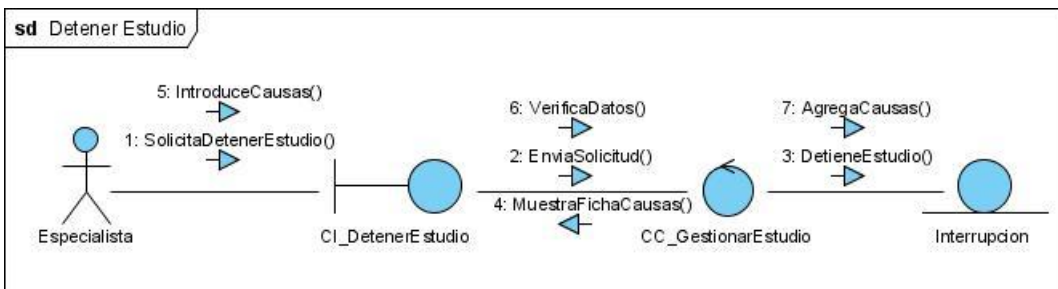
Anexo 19



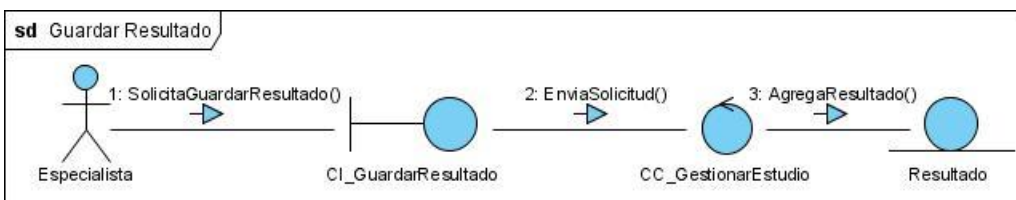
Anexo 20



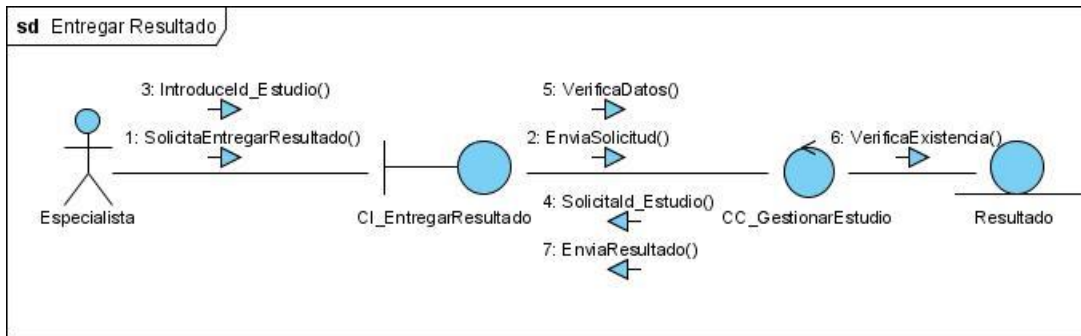
Anexo 21



Anexo 22



Anexo 23



Anexo 24