

Universidad de las Ciencias Informáticas
Facultad 6



**Título: BioSyS. Desarrollo de Portlets para la Gestión de
la Información en la Simulación de Sistemas Biológicos**

**Trabajo de Diploma para optar por el título de
Ingeniero Informático**

Autores:

Lisbeth León Téllez
Alain Meneses Jiménez

Tutores:

Lic. Haydee Noemí Vidal
Ing. Yudel Juan Quintana

Junio 2008

“La calidad nunca es un accidente, siempre es el resultado de un esfuerzo de la inteligencia.”

John Ruskin

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Lisbeth León Téllez

Autor

Alain Meneses Jiménez

Autor

Ing. Yudel Juan Quintana

Tutor

Lic. Haydee Noemí Vidal

Tutor

AGRADECIMIENTOS

A la Revolución y a nuestro Comandante por darnos la oportunidad de estudiar y superarnos.

A nuestros padres por su apoyo incondicional, por confiar siempre en nosotros y esperar con paciencia la llegada de este día.

A Edel, Keiler, Noel, Liusmila, Rolando, Linet y Yudel por su ayuda y paciencia.

A nuestros amigos, profesores y todo aquel que estuvo a nuestro lado en las buenas y en las malas.

DEDICATORIA

Alain

A mi madre, por ser mi inspiración en todos los pasos que he dado y que daré en la vida, por guiarme por el camino correcto, por los consejos, por ser un ejemplo para el mundo de cómo debe ser una madre y porque le debo todo lo que soy.

A mi padre, por el ejemplo que me ha sabido dar, por estar siempre presente cada vez que lo he necesitado y por sus sabias enseñanzas.

A Toyo, por ayudarme y quererme como un hijo, por sus consejos y por estar con mi madre en los buenos y en los malos momentos.

A mi hijo, por darme inconscientemente ese último impulso que tanto necesité. Además por obligarme a ser un ejemplo para él.

A mi mujer, por la paciencia y comprensión, por la madurez con que me ha esperado, por amarme y por regalarme ese hijo que tanto anhelaba.

A mis tías, por su ayuda incondicional a mis primos, que de ellos he sacado enseñanzas también.

A mi suegra, por ayudarme y orar todas las noches por mí. A María Antonia por cuidar de mi hijo cuando lo ha necesitado, a Andrés por cuidar de todos. A Hirán que de cada discusión siempre saqué alguna enseñanza.

En fin, a toda mi familia y amigos que todos han aportado un grano de arena en mi formación como persona.

Lisbeth

A mis abuelos, por su amor. A pipa y mima, por educarme y por siempre estar conmigo en todo momento.

A mis padres, Maysa y Gaspar por todo su apoyo y comprensión, por amarme y enseñarme siempre cual es el camino correcto.

A mi hermano, que le sirva de inspiración.

RESUMEN

En el marco de los proyectos productivos de la facultad 6 y considerando la necesidad de incluir en el Portal de Servicios Bioinformáticos la plataforma de BioSyS, se implementan 6 Portlets que gestionan la información necesaria para la Simulación de Sistemas Biológicos, la información se centra fundamentalmente en la relación que existe entre un Sistema Biológico y sus Modelos Matemáticos asociados, de manera que los investigadores cuenten con una interfaz amigable, personalizada y adaptable a futuras versiones, que de forma ágil e integrada le permita crear un escenario para la realizar una simulación.

En este documento se hace un análisis comparativo acerca de las tecnologías existentes y se seleccionan las más apropiadas, se abordan conceptos importantes como Portlets, Sistemas Biológicos, Modelos Matemáticos, se muestran resultados de la propuesta y finalmente se hacen una serie de recomendaciones para el desarrollo futuro del mismo.

ÍNDICE DE CONTENIDO

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN	IV
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Biología de Sistemas.....	6
1.1.1 <i>Definición de Biología de Sistemas.</i>	6
1.2 Modelos Matemáticos.	7
1.2.1 <i>Modelación Matemática.</i>	7
1.2.2 <i>Sistemas de Ecuaciones Diferenciales.</i>	8
1.2.3 <i>Los Sistemas de Ecuaciones Diferenciales en la Modelación de Sistemas Biológicos.</i>	8
1.3 <i>Tendencia de las aplicaciones de software en la Simulación de Sistemas Biológicos.</i>	9
1.3.1 <i>Importancia del desarrollo de Portlets para la gestión de la información en la Simulación de Sistemas Biológicos.</i>	10
1.4.1 <i>¿Cómo funciona un Portlet?</i>	11
1.4.2 <i>Modos del Portlet.</i>	12
1.4.3 <i>Manejo de Peticiones de Portlets.</i>	13
1.5 Contenedor de Portlets.	14
1.5.1 <i>JetSpeed.</i>	15
1.5.2 <i>Open Portal Portlet Container Driver.</i>	15
1.5.3 <i>Gridsphere.</i>	15
1.5.4 <i>¿Por qué Open Portal Portlet Container Driver?</i>	15
1.6 Sistema Operativo.....	15

1.6.1 ¿Por qué Linux?	16
1.7 Metodología de desarrollo.....	17
1.7.1 XP.....	18
1.7.2 OpenUP.....	19
1.8 Herramientas de Modelado de Visual.	20
1.8.1 JUDE Community.....	20
1.8.2 Omondo.....	21
1.8.3 Umbrello.....	21
1.8.4 Visual Paradigm.....	21
1.8.4 ¿Por qué Visual Paradigm?.....	22
1.9 Lenguajes de Programación.	22
1.9.1 PHP.....	22
1.9.2 JSP.....	23
1.9.3 Lenguajes utilizados.....	23
1.10 Herramientas de Programación.	24
1.10.1 NetBeans IDE.....	24
1.10.2 Eclipse.....	25
1.10.3 ¿Por qué NetBeans IDE?.....	25
1.11 Gestor de Base de Datos.....	25
1.11.1 PostGreSQL.....	26
1.11.2 MySQL.....	27
1.11.3 ¿Por qué MySQL?.....	27
1.12 Arquitectura Cliente-Servidor.....	28

1.12.1 Ventajas de la Arquitectura Cliente-Servidor(C/S)	28
1.13 Servidor Web.....	29
1.13.1 Cherokee.....	30
1.13.2 Tomcat.....	30
1.13.4 GlassFish.....	31
1.13.5 ¿Por qué GlassFish?.....	31
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	32
2.1 Requisitos.....	32
2.1.1 Requisitos Funcionales.....	32
2.1.2 Requisitos No Funcionales.....	33
2.2 Definición de los Actores del Sistema.....	34
2.3 Definición de Casos de Uso del Sistema.....	35
2.4 Descripción de los CU del Sistema.....	36
Conclusiones del capítulo.....	48
CAPÍTULO 3: DISEÑO DEL SISTEMA.....	49
3.1 Descripción de Estilos Arquitectónicos.....	49
3.1.1 Patrón "Modelo-Vista-Controlador".....	49
3.2 Patrones de Diseño.....	49
3.2.1 ¿Por qué utilizar patrones de diseño?.....	49
3.2.2 Patrones GRASP.....	50
3.3 Diagramas de Clases del Diseño.....	51
3.4 Diagramas de Interacción.....	57
3.4.1 Diagrama de secuencia.....	57

3.5. Modelo de Despliegue.....	74
Conclusiones del capítulo	74
CAPÍTULO 4: IMPLEMENTACIÓN.....	75
4.1 Diagrama de Componentes	75
4.2 Principales pantallas de la aplicación.....	83
Conclusiones del capítulo	89
CONCLUSIONES	90
RECOMENDACIONES	91
BIBLIOGRAFÍA	92
REFERENCIAS BIBLIOGRÁFICAS	94

ÍNDICE DE TABLAS

Tabla 2. 1 Actores del Sistema	35
Tabla 2. 2 Descripción del CU Insertar Solicitud para la creación de un proyecto	37
Tabla 2. 3 Descripción del CU Gestionar Solicitud.	39
Tabla 2. 4 Descripción del CU Gestionar Proyecto.	40
Tabla 2. 5 Descripción del CU Eliminar Proyecto.	41
Tabla 2. 6 Descripción del CU Gestionar Sistema Biológico.	43
Tabla 2. 7 Descripción del CU Eliminar Sistema Biológico.....	45
Tabla 2. 8 Descripción del CU Gestionar Modelo Matemático.	47
Tabla 2. 9 Descripción del CU Eliminar Modelo Matemático.	48

ÍNDICE DE FIGURAS

Figura 3. 1 Clase Acceso Datos	52
Figura 3. 2 Gestionar Sistema Biológico.....	53
Figura 3. 3 Eliminar Sistema Biológico	54
Figura 3. 4 Gestionar Modelo Matemático.....	55
Figura 3. 5 Eliminar Modelo Matemático.	56
Figura 3. 6 Insertar Solicitud para la creación de un Proyecto	58
Figura 3. 7 Gestionar Solicitud. Escenario (Buscar)	59
Figura 3. 8 Gestionar Solicitud. Escenario (Aceptar).....	60
Figura 3. 9 Gestionar Solicitud. Escenario (Rechazar).....	61
Figura 3. 10 Gestionar Proyecto. Escenario (Buscar)	62
Figura 3. 11 Gestionar Proyecto. Escenario (Modificar)	63
Figura 3. 12 Eliminar Proyecto. Escenario (Buscar).....	64
Figura 3. 13 Eliminar Proyecto.....	65
Figura 3. 14 Gestionar Sistema Biológico. Escenario (Insertar).....	66
Figura 3. 15 Gestionar Sistema Biológico. Escenario (Buscar).....	67
Figura 3. 16 Gestionar Sistema Biológico. Escenario (Modificar).....	68
Figura 3. 17 Eliminar Sistema Biológico.	69

Figura 3. 18 Gestionar Modelo Matemático. Escenario (Insertar)	70
Figura 3. 19 Gestionar Modelo Matemático. Escenario (Buscar)	71
Figura 3. 20 Gestionar Modelo Matemático. Escenario (Modificar)	72
Figura 3. 21 Gestionar Modelo Matemático. Escenario (Eliminar)	73
Figura 3. 22 Modelo de Despliegue.	74
Figura 4. 1 Diagrama de Componentes (Insertar Solicitud).....	76
Figura 4. 2 Diagrama de Componentes (Gestionar Proyecto).....	77
Figura 4. 3 Diagrama de Componentes (Insertar Sistema Biológico).....	78
Figura 4. 4 Diagrama de Componente (Modificar Sistema).....	79
Figura 4. 5 Diagrama de Componentes (Insertar Modelo Matemático)	80
Figura 4. 6 Diagrama de Componentes (Modificar Modelo Matemático)	81
Figura 4. 7 Diagrama de Componentes (Eliminar)	82
Figura 4. 8 Insertar Solicitud.	83
Figura 4. 9 Gestionar Solicitud.....	83
Figura 4. 10 Buscar Proyecto.	84
Figura 4. 11 Eliminar Proyecto(s).	84
Figura 4. 12 Mostar Proyecto(s).	84
Figura 4. 13 Modificar Proyecto(s).....	85
Figura 4. 14 Insertar Sistema(s) Biológico(s).....	85

Figura 4. 15 Modificar Sistema(s) Biológico(s).....	85
Figura 4. 16 Eliminar Sistema(s) Biológico(s).....	86
Figura 4. 17 Insertar Modelo(s) Matemático(s).....	86
Figura 4. 18 Mostrar Modelo(s) Matemático(s).....	87
Figura 4. 19 Eliminar Modelo(s) Matemático(s).....	87
Figura 4. 20 Modificar Modelo Matemático.....	88
Figura 4. 21 Buscar para eliminar.....	88

INTRODUCCIÓN

La Biotecnología, en un sentido amplio se puede definir como la aplicación de organismos vivos, componentes o Sistemas Biológicos para la obtención de bienes y servicios.

La actual Biotecnología es una empresa intensamente interdisciplinaria, caracterizada por la reunión de conceptos y metodologías procedentes de numerosas ciencias, para aplicarlas tanto a la investigación básica como a la resolución de problemas prácticos.

Algunas de las ramas del conocimiento implicadas en la Biotecnología son:

- Microbiología.
- Bioquímica.
- Genética.
- Biología Celular.
- Química.
- Ingeniería Bioquímica.
- Ingeniería Mecánica.
- Ciencia y Tecnología de Alimentos.
- Electrónica.
- Informática.

La Biotecnología incluye líneas de investigación dirigidas a resolver diversos problemas en diferentes esferas, tales como: la salud, la agricultura, el medio ambiente entre otras. Su campo de acción es inmenso como por ejemplo:

- Aplicaciones terapéuticas:
 - Productos farmacéuticos:
 - ✓ Antibióticos.
 - ✓ Vacunas.
 - Terapias génicas.
- Diagnósticos:
 - Diagnósticos para salud humana.
 - Diagnósticos para agricultura y ganadería.
 - Ensayos para calidad de alimentos.
 - Ensayos para calidad ambiental.
- Alimentación y medio ambiente.

La calidad de las producciones biotecnológicas ha interesado a empresas de países altamente desarrollados como Japón y Francia. Dentro de las ramas que estudia la Biotecnología, una de las más jóvenes y que ha comenzado a ganar terreno dentro de los científicos que a nivel mundial se dedican a su estudio, es la Biología de Sistemas.

La Biología de Sistemas es un área de investigación científica que se preocupa del estudio de procesos biológicos usando un enfoque sistémico. Comenzó a desarrollarse en los años 60 y su institucionalización académica no se produjo hasta el año 2000.

En Cuba, la Biotecnología ha alcanzado avances relevantes que sobrepasan los obtenidos por países altamente desarrollados y exhibe resultados de primer nivel en la obtención de vacunas humanas y veterinarias, medios de diagnósticos, anticuerpos monoclonales, fármacos, interferones y bioproductos.

La Revolución Cubana para garantizar la formación de profesionales de la informática preparados y desarrollar la creciente industria del software en el país, creó en el 2002 la Universidad de las Ciencias Informáticas (UCI), la cual esta compuesta por 10 facultades que desarrollan perfiles diferentes. La facultad 6 tiene como perfil la Bioinformática y desarrolla productos de calidad para esta importante rama de la economía. Se trabaja en la actualidad en un proyecto llamado BioSyS, él cual consta de 6 módulos:

- Modelación.
- Editor de Ecuaciones.
- Simulación.
- Análisis.
- Estimación de Parámetros.
- Base de datos.

La limitante de este sistema, es que todos los módulos están conectados a una plataforma de cálculo distribuido, que cuenta con una infraestructura que requiere una serie de recursos informáticos, o sea, una vez instalado el software en un Centro, solamente tendrán acceso a los servicios, los trabajadores del Centro. Si alguien más necesita de los servicios que brinda la plataforma tendría que instalar el software y para eso necesitaría todos los recursos informáticos que requiere BioSyS. Por lo anterior expuesto surge la necesidad de crear un Portal de Servicios que incluya entre las funcionalidades de la plataforma BioSyS, entre ellas se encuentra realizar simulación. Para realizar una simulación es indispensable gestionar información fundamentalmente de Sistemas Biológicos y sus Modelos Matemáticos asociados, marco en el que se basa la investigación.

Teniendo en cuenta lo presentado anteriormente se ha planteado el siguiente **problema científico**: ¿Cómo gestionar la información que permita realizar simulaciones de Sistemas Biológicos en el Portal de Servicios Bioinformáticos?

Objeto de estudio: El desarrollo de Portlets para la gestión de la información en la Biología de Sistemas.

Campo de acción: El desarrollo de Portlets para la gestión de la información en la Simulación de Sistemas Biológicos.

Tiene como **objetivo general:** Desarrollar Portlets que permitan gestionar la información para la Simulación de Sistemas Biológicos.

Para lograr el objetivo general del trabajo se trazaron los siguientes **objetivos específicos:**

- Análisis de los Portlets para la gestión de la información en la simulación de Sistemas Biológicos.
- Diseñar los Portlets para la gestión de la información en la simulación de Sistemas Biológicos.
- Implementar los Portlets para la gestión de la información en la simulación de Sistemas Biológicos.

Para darle solución a la problemática planteada y lograr el cumplimiento de los objetivos se proponen las siguientes **tareas:**

- Estudio general del módulo Simulación de la plataforma BioSyS.
- Entrevista a los responsables del módulo, así como al líder del Proyecto.
- Estudio y definición de la metodología y herramientas actuales para el desarrollo de los Portlets.
- Obtención, descripción y validación de las funcionalidades.
- Realización del diseño de los Portlets para la gestión de la información en la simulación de Sistemas Biológicos.
- Implementación de los Portlets para la gestión de la información en la simulación de Sistemas Biológicos.

Con la implementación de la solución informática que se propone, se espera obtener como resultados:

- Portlets que faciliten la gestión de la información para la Simulación de Sistemas Biológicos.
- Los investigadores podrán crear su propio proyecto con la información que necesiten para realizar simulaciones.

El trabajo tiene la estructura siguiente:

Capítulo1: En este capítulo se explica detalladamente los fundamentos teóricos principales de una parte fundamental de la información que se gestiona para la simulación de Sistemas Biológicos, aborda el estado del arte de aplicaciones que son utilizadas para simular Sistemas Biológicos, además se referencian las tendencias de las tecnologías y metodologías seleccionadas para la concepción de la solución informática propuesta.

Capítulo2: En este capítulo se describen los requisitos funcionales y no funcionales, los actores que intervienen en el sistema, los Casos de Uso y sus descripciones.

Capítulo3: En este capítulo se abordan los estilos arquitectónicos y patrones del diseño, los diagramas de clases del diseño, los diagramas de interacción y el modelo de despliegue.

Capítulo4: En este capítulo se representa el diagrama de componentes y se presentan pantallas de la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se explican detalladamente los principales fundamentos teóricos de la información que se gestiona para la Simulación de Sistemas Biológicos. Se aborda el estado del arte de aplicaciones que son utilizadas para simularlos y de qué manera gestionan la información necesaria para hacerlo, además se referencian las tendencias de las tecnologías y metodologías seleccionadas para la concepción de la solución informática propuesta.

1.1 Biología de Sistemas.

Usando herramientas de modelación, comparación y simulación, esta nueva rama de la Biotecnología, a diferencia de los métodos históricos utilizados por los biólogos para las investigaciones se centra en estudiar los Sistemas Biológicos como un todo.

1.1.1 Definición de Biología de Sistemas.

La Biología de Sistemas es una disciplina académica que pretende integrar diferentes niveles de información con el fin de entender cómo funcionan los Sistemas Biológicos. Intenta crear modelos comprensibles de sistemas mediante el estudio de las relaciones y las interacciones entre las diferentes partes de un Sistema Biológico por ejemplo, las redes génicas y las redes de interacción de proteínas implicadas en la señalización celular, las rutas metabólicas, los orgánulos, las células, los sistemas fisiológicos, los organismos, entre otros. (1)

En esta nueva ciencia el uso de los Modelos Matemáticos cobra gran importancia, debido a que permiten predecir el comportamiento del proceso como un sistema dinámico.

Ejemplos de su utilización práctica:

- Para la enfermedad de La Gota solo existe un medicamento, que por cierto, trae consigo algunos efectos secundarios, es por eso que la Biología de Sistemas enfoca sus estudios en crear nuevos medicamentos que acaben con la enfermedad y a su vez disminuyan los efectos secundarios.

- En el caso del Cáncer de Hígado, una célula del mismo hígado transformándose puede ser la que provoque el tumor, de allí que la Biología de Sistemas se dedique a modificar el funcionamiento de la célula para detener el desarrollo del tumor.

La Biología de Sistemas implica el análisis de todos los componentes del Sistema Biológico, que incluye un análisis profundo de cómo se expresan los genes y sus interacciones complejas dentro de una célula, tejido, órgano o todo el organismo.

1.2 Modelos Matemáticos.

Un Modelo Matemático se define como una descripción desde el punto de vista de las matemáticas de un hecho o fenómeno del mundo real, desde el tamaño de la población, hasta fenómenos físicos como la velocidad, aceleración o densidad. El objetivo del Modelo Matemático es entender ampliamente el fenómeno y tal vez predecir su comportamiento en el futuro. (2)

Si antes del estudio de cualquier Sistema Biológico se realiza primero un Modelo Matemático que lo represente, sin dudas el estudio se realizara de una manera más factible.

¿Para que sirven?

- Para estudiar o comprender el sistema real.
- Predecir su comportamiento.
- Proyectar alguna modificación o intervención.
- Probar cosas imposibles en el caso real.

1.2.1 Modelación Matemática.

La modelación matemática relacionada con sistemas de representaciones integra símbolos, signos, figuras, gráficas y construcciones geométricas. Éstos expresan el concepto y suscriben en sí mismos el modelo con el cual es posible interpretar y predecir comportamientos de fenómenos físicos. La simulación y la modelación son representaciones de un objeto matemático que está vinculado a una situación física o real. Un modelo es una representación de estructuras. El término modelo se refiere a la generalización

conceptual que se abstrae de un grupo de experiencias con el propósito de categorizar y sistematizar nuevas experiencias. (3)

Para la simulación de Sistemas Biológicos se hace uso de diferentes Modelos Matemáticos, en dependencia del sistema que se quiera simular. Existen diferentes tipos de Modelos Matemáticos, se centrará la atención en el modelo Sistemas de Ecuaciones Diferenciales.

1.2.2 Sistemas de Ecuaciones Diferenciales.

Una Ecuación Diferencial es una ecuación que relaciona de manera no trivial a una función desconocida y una o más derivadas de esta función con respecto a una o más variables independientes. Si la función desconocida depende de una sola variable la Ecuación Diferencial se llama Ordinaria, por el contrario, si depende de más de una variable, se llama Parcial. (4)

Existen distintas maneras de clasificar a las Ecuaciones Diferenciales: Una de ellas es la clasificación por el orden de la ecuación (que viene dado por la derivada de mayor orden que aparezca en la expresión) y otra por la linealidad. Ésta última es quizás una de las clasificaciones más importantes en el sentido de que permite caracterizar inmediatamente la facilidad con que se pueden hallar sus soluciones. De allí que se pueden clasificar en Lineales y no Lineales.

1.2.3 Los Sistemas de Ecuaciones Diferenciales en la Modelación de Sistemas Biológicos.

Independientemente de los avances logrados en la Bioinformática no se puede negar la necesidad de un mayor esfuerzo en la producción de software orientado a los expertos en Biología y Química, siendo este desafío el reto mas importante en esta área.

En ese sentido, los cálculos basados en restricciones gozan de unas características muy atractivas. Por estas razones, el trabajo futuro en los cálculos basados en restricciones se ciernen promisorio. En el campo aplicativo, la inclusión de estructuras y/o mecanismos que induzcan la noción de tiempo parece ser el paso inicial.

En este contexto, el desarrollo de sistemas de restricciones que incorporen Ecuaciones Diferenciales es una alternativa interesante por su alta aplicabilidad. Normalmente los procesos de una red metabólica generalmente se expresan en términos de Ecuaciones Diferenciales.

1.3 Tendencia de las aplicaciones de software en la Simulación de Sistemas Biológicos.

En la actualidad existen una amplia gama de software que se utilizan para la simulación de Sistemas Biológicos los cuales han contribuido a las investigaciones de la Biología de Sistemas, algunos de estos software son:

- **ByoDyn:** Usando datos experimentales este software es capaz de estimar parámetros cinéticos. Permite simular un modelo, analiza la sensibilidad del sistema con relación a los parámetros. Entre sus principales ventajas se encuentran la utilización de formatos estándares que permiten al programa operar con modelos desarrollados por otros científicos y brinda soporte a los cálculos complicados.
- **Cellware:** Es un software diseñado fundamentalmente para modelar y simular reacciones bioquímicas de la célula, es capaz de importar y exportar modelos en formato SBML. Por la intención de ser multiplataforma fue diseñado en lenguaje Java La interfaz de usuario es amigable para el biólogo que la utilice y tiene herramientas que permiten construir adecuadamente los modelos de células deseados. Gestiona la información mediante un fichero donde se almacenan los datos correspondientes al modelo y a las simulaciones realizadas.
- **ImmunoGrid:** Simula procesos a escala natural. Logra la conexión de las interacciones a nivel molecular con los modelos a nivel de sistema. Para la gestión de información hace uso de bases de datos especializadas para almacenar, manipular y modelar datos inmunológicos, lo que facilita la búsqueda de Modelos Matemáticos y predictivos. Debido a que utiliza bases de datos especializadas y a su vez éstas utilizan estándares, puede obtener información más detallada, con un alto nivel de calidad y confiabilidad.

Después de haber realizado un estudio acerca de las aplicaciones desarrolladas en el mundo, dedicadas a la simulación de sistemas biológicos, se ratificó la importancia de desarrollar software Webs de alta calidad para esta importante rama de la ciencia. Con la creación del Portal de Servicios Bioinformáticos se pondrá al alcance de la comunidad científica los servicios de la plataforma BioSys, que para su funcionamiento consta entre otros módulos, el encargado de la gestión de la información para simular Sistemas Biológicos. Este software incluye las principales funcionalidades de los ya existentes, facilita el

uso de las potencialidades de cómputo de una red corporativa, en la solución de aquellos problemas que así lo requieran e incorpora algoritmos de análisis que ayuda a los usuarios a arribar a conclusiones, puede acelerar las investigaciones en esta área.

1.3.1 Importancia del desarrollo de Portlets para la gestión de la información en la Simulación de Sistemas Biológicos.

Debido a la cantidad de información que se maneja en el estudio de los Sistemas Biológicos, el desarrollo de un sistema que gestione información para la Simulación de Sistemas Biológicos requiere de varias interfaces que se comuniquen entre sí.

Desarrollar una aplicación Web que conste de dichas interfaces, sería una de las vías para resolver el problema, la desventaja de esta solución es que la persona que interactúe con el sistema podría realizar solamente la gestión de un tipo de información, para luego gestionar otra. El desarrollo de Portlets contribuye a eliminar esta limitante, debido a que estos brindan la oportunidad mediante una sola interfaz de poder gestionar varias informaciones a la vez. Desde el punto de vista del usuario final, los Portlets ayudan de manera eficiente a mejorar la usabilidad de un portal, ya que cada usuario podrá ajustar a su manera la distribución y la apariencia de los componentes con los que desea trabajar. Además, el administrador del portal podrá decidir qué puede ver y qué no cada usuario. Esta característica puede lograrse, evidentemente, sin la necesidad del uso de Portlets, pero implica para el desarrollador mayor tiempo de trabajo, que puede simplificarse con la utilización de Portlets, sin tener en cuenta que además, se estará siguiendo un estándar abierto. Por esta razón se propone realizar una serie de Portlets que permitan la gestión de la información necesaria para la Simulación de Sistemas Biológicos.

1.4 ¿Qué es un Portlet?

Un Portlet es un componente Web hecho en Java y manejado a través de un contenedor de Portlets que procesa las peticiones de los clientes y produce contenido dinámico. El contenido generado por un Portlets es llamado fragmento, una pieza de código HTML (Hypertext Markup Language), XHTML (Extensible Hypertext Markup Language), WML (Wireless Markup Language) adherida a ciertas reglas. Un fragmento puede ser agregado a otros fragmentos a fin de formar un documento completo, por ejemplo un portal es un conjunto de fragmentos generados por diversos Portlets.

El contenido generado por un Portlets puede variar de un usuario a otro dependiendo de cómo haya configurado el usuario el Portlets. A diferencia de los Servlets, los Portlets no tienen interacción directa con los clientes Web. En su lugar, los clientes Web interactúan con el portal a través de un mecanismo de solicitud/entrega aplicado por un contenedor de Portlets el cual también maneja el ciclo de vida de los Portlets. Generalmente, los Portlets tienen una clara separación entre el contenido y la presentación la cual es manejada por una o más clases de Java que contienen la aplicación lógica. Los portales usan los Portlets como componentes modulares para interfaz de usuario. (5)

Surgen como respuesta a la necesidad de simplificar el uso de dichas aplicaciones haciendo uso de interfaces amigables además de aprovechar la seguridad y el beneficio de la tecnología Grids.

1.4.1 ¿Cómo funciona un Portlet?

Teniendo en cuenta ciertas condiciones de proceso, un Portlet procesa una petición y responde con la salida apropiada a cada caso. Si se trata de un portal con varios módulos, la encargada de gestionar y controlar la ejecución, comunicación y respuesta de todos los Portlets activos, es la interfaz de programación de aplicaciones API (Applications Programming Interface).

Naturalmente, también es posible procesar peticiones mediante un protocolo simple de acceso a objetos SOAP (Simple Object Access Protocol) dentro de un Portlet a través de aplicaciones de terceros. De esta forma, se abre el camino para los Web Services a través del estándar UDDI (Universal Description Discovery and Integration) e invocar los servicios descritos en el WSDL (Web Service Description Language).

1.4.2 Modos del Portlet.

Los Portlets desempeñan diferentes tareas y crean contenidos de acuerdo a su función actual. Un modo de Portlet indica la función que está desempeñando en cierto momento, especifica el tipo de tarea que debería desempeñar y qué contenido debería generar. Cuando se invoca a un Portlet, el contenedor de Portlets provee el modo para el actual requerimiento al Portlet. Estos pueden cambiar su modo mientras procesan una petición de acción.

Se definen 3 categorías de modos de Portlet:

- **Edit:** Muestra una o más vistas que permiten al usuario personalizar los parámetros del Portlet.
- **Help:** Muestra pantallas de ayuda.
- **View:** Muestra la salida del Portlet.

Estados de las Ventanas.

El estado de la ventana le indica al Portlet cuánto espacio se le está asignando a él. Los siguientes son los estados de ventanas especificados por el API:

- **Normal:** El Portlet compartirá la pantalla con otros Portlets. Esto significa que el Portlet debe limitar su marco.
- **Minimizado:** El Portlet debe proveer poco o ninguna salida.
- **Maximizado:** El Portlet no comparte la ventana con otros Portlets.

1.4.3 Manejo de Peticiones de Portlets.

Hay dos tipos de peticiones que se pueden realizar a un Portlet la cuales son: los action request y los render request, estas peticiones están acompañada por su tipo de URL: actions URL y renders URL. Un action URL tiene como objetivo el método processAction del Portlets, mientras que el URL render apunta al método render.

Si la petición de un cliente es un action request, entonces este puede apuntar a un solo Portlet, el cual debe ser ejecutado de primero. No se puede ejecutar otro action request en los demás Portlets, solo render request.

El Contenedor de Portlets ejecutará el processAction en el Portlet objetivo, espera hasta que este finalice antes de ejecutar el render en los restantes Portlets. El llamado de los métodos render de los demás Portlets se pueden hacer en cualquier orden y en paralelo.

Action Request.

El Action request maneja el cambio de estado de un Portlet basado en los parámetros del action request. Esto es hecho usando el método processAction, el cual toma un objeto Action Request y un objeto ActionResponse como parámetros.

El objeto Action Request, provee:

- Los parámetros del action request.
- El modo del Portlet.
- La sesión del Portlet.
- El estado de la ventana.
- El objeto de preferencias del Portlet.
- El contexto del portal.

Para cambiar el modo del Portlet o el estado de la ventana, se llamará el método apropiado en el objeto `ActionResponse`. El cambio se llevará a cabo cuando se llame al método `render` al finalizar de procesar el método `processAction`.

RenderRequest.

El `RenderRequest` genera un fragmento desde el estado actual del Portlet. El objeto `RenderRequest` provee el acceso a lo siguiente:

- Los parámetros de render request.
- El modo del Portlet.
- La sesión del Portlet.
- El estado de la ventana.
- El objeto de preferencia del Portlet.

1.5 Contenedor de Portlets.

Un contenedor de Portlets corre los Portlets y les provee con el ambiente de ejecución adecuado, alberga los Portlets y gestiona sus ciclos de vida. También provee mecanismos de almacenamiento persistentes para las preferencias de los Portlets.

Ejemplos:

- JetSpeed.
- Open Portal Portlet Container Driver.
- Gridsphere.

1.5.1 JetSpeed.

JetSpeed es una de las muchas soluciones existentes en el mercado que implementa la especificación de Portlets, componentes Web habilitados para Java. Con JetSpeed puede crear sus propios Portlets o emplear algunos de los que ya incorpora esta solución de uso libre.

1.5.2 Open Portal Portlet Container Driver.

El contenedor de Portlets Open Portal Portlet Container permite la creación y utilización de Web Services, framework de administración, acceso a dispositivos móviles, integración con Java Server Pages. Controlan el ciclo de vida de los Portlets. Su modelo basado en Portlets les da a esos usuarios una interfaz flexible, sencilla y fácil de usar.

1.5.3 Gridsphere.

Gridsphere es una herramienta Código Abierto (Open-Source) para crear portales. Permite a los desarrolladores generar y empaquetar rápidamente aplicaciones basadas en Web Portlets que pueden ser ejecutadas y administradas con el Contenedor de Portlets Gridsphere. Este provee un ambiente altamente personalizable para un amplio rango de usuarios solo que una de sus grandes desventajas es que para crear un Portlet se requiere de una serie de pasos lógicos inviolables que hacen que sea muy engorroso el proceso de configuración.

1.5.4 ¿Por qué Open Portal Portlet Container Driver?

Open Portal Portlet Container Driver es un Contenedor de Portlet que se caracteriza por ser ágil y fácil de configurar, brinda al usuario una interfaz visual para administrar los Portlets de una manera rápida y segura, además de tener un excelente rendimiento para gestionar varios Portlets a la vez y para la creación de un Portlet es mucho más sencillo que el Gridsphere.

1.6 Sistema Operativo.

Un Sistema Operativo (SO) es un software básico que controla una computadora. SO es en sí mismo un programa de computadora. Sin embargo, es un programa muy especial, quizá el más complejo e

importante en una computadora. El SO despierta a la computadora y hace que reconozca a la CPU, la memoria, el teclado, el sistema de vídeo y las unidades de disco. Además, proporciona la facilidad para que los usuarios se comuniquen con la computadora y sirve de plataforma a partir de la cual se corran programas de aplicación.

- *Windows:*

Microsoft Windows es un Sistema Operativo para el trabajo en ordenadores digitales de 32 bits, sustituto del MSDOS y resultado de la evolución lógica de versiones anteriores de Windows (que surgen como ambiente gráfico del DOS) y que procura llevar la fuerza de la computación a los usuarios más comunes. Windows integra un sistema de funciones previamente ejecutados por MS DOS. Es un Sistema Operativo poderoso y fácil de usar en un ambiente gráfico, con capacidad de realizar acciones multitareas y usa interfaz orientada a objeto.

- *Linux:*

Es un Sistema Operativo de software libre distribuido bajo la Licencia Pública General de GNU (GNU GPL), tiene todas las prestaciones que se pueden esperar de un Unix moderno y completamente desarrollado: multitarea real, memoria virtual, bibliotecas compartidas, carga de sistemas a-demanda, compartimiento y soporte de redes TCP/IP.

1.6.1 ¿Por qué Linux?

- *Es más seguro.*

- ✓ Ya que la gran mayoría de los ataques de hackers' son dirigidos a servidores Windows al igual que los virus los cuales se enfocan principalmente a servidores con éste Sistema Operativo.
- ✓ La plataforma Linux es más robusta lo cual hace más difícil que algún intruso pueda violar el sistema de seguridad.

- *Es más rápido.*

- ✓ Al tener una plataforma más estable, esto favorece el desempeño de aplicaciones de todo tipo tales como: bases de datos, aplicaciones XML, multimedia, etcétera.
- ✓ La eficiencia de su código fuente hace que la velocidad de las aplicaciones Linux sean superiores a las que corren sobre Windows, lo cual se traduce en velocidad de su página.
- *Es más económico*
 - ✓ Ya que requieren menor mantenimiento. En servidores Windows es más costoso debido a que es necesaria una frecuente atención y monitoreo contra ataques de virus, hackers y errores de código, instalación y actualización de parches y Paquetes de Servicios (Service Packs).
 - ✓ El software Linux así como también el sin número de aplicaciones que brinda, son de código abierto.

No requieren supervisión tan estrecha ni pagos de pólizas de mantenimiento necesarias para obtener los Service Packs. (6)

Por lo anteriormente expuesto, se utilizará el sistema Operativo Linux para el desarrollo de la investigación.

1.7 Metodología de desarrollo.

La metodología de desarrollo es un aspecto de gran importancia para el proceso de desarrollo del software, manifiesta más trascendencia si el proyecto a realizar es complejo, puesto que todo desarrollo de software es riesgoso y difícil de controlar. Si no se lleva una metodología adecuada, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Las metodologías se clasifican en ágiles, flexibles y robustas.

Las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de

producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

Las metodologías robustas es más adaptable para proyectos a largo plazo, es basada en la documentación, en la que no son deseables todos esos cambios volátiles de las metodologías ágiles. Cuenta con diferentes elementos de planificación (plan de desarrollo, plan de iteración, plan de calidad, etc.) con los que se controla el desarrollo del software. A través de un esquema predefinido de escalabilidad y gestión de riesgos, se pueden reconocer previamente problemas y fallos de forma temprana y prevenirlos/corregirlos.

Mientras que las flexibles son adaptables a las otras .

1.7.1 XP.

Esta metodología se basa en:

- **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo el futuro, se puedan realizar hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara para conocer los posibles errores.
- **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** Una particularidad de esta metodología es la propuesta de programación en pares, consiste en la participación de dos desarrolladores de un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Objetivos:

- Añade funcionalidad con retroalimentación continua.

- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

1.7.2 OpenUP.

Es una versión de RUP, se utiliza preferentemente en proyectos pequeños. Cuando se dice proyectos pequeños se está haciendo referencia a proyectos de 3 a 6 personas. OpenUP contiene las características esenciales de RUP, que incluye el desarrollo iterativo de Casos de Uso, además de proporcionar un acercamiento a la arquitectura central del sistema. Han sido incluidas la mayoría de las opciones de RUP y muchos elementos han sido unidos. El resultado es un proceso mucho más simple que sigue fielmente los principios de RUP.

¿Cómo se organiza OpenUP?

Se organiza en 2 dimensiones diferentes: Método y Proceso.

- **Método:** Los roles, las tareas, los artefactos están definidos, sin tener en cuenta cómo ellos son aplicados en el ciclo de vida del proyecto.
- **Proceso:** Es cuando los elementos del proceso son aplicados en el sentido conductual, donde el mismo brinda los roles, las tareas, los artefactos. Pueden crearse ciclos de vida diferentes para proyectos diferentes.

OpenUP está enfocado en las siguientes disciplinas: requerimientos, arquitectura, desarrollo, prueba, dirección de proyecto y cambio de dirección.

Roles:

- **Analista:** Responsable de recoger los requerimientos y documentarlos como es necesario.

- **Arquitecto:** Responsable de la arquitectura del software, de realizar un diseño global y de la implementación del sistema.
- **Desarrollador:** Responsable de crear una solución(o parte de ella) a partir del diseño, implementación y prueba, uniendo todos los componentes.
- **Probador:** Responsable de probar el sistemas de diferentes perspectivas que el diseñador hace, asegurándose de que el sistema trabaje como se definió y de que el cliente lo acepte.
- **Jefe de proyecto:** Planea y dirige el proyecto, coordina la integración de los stakeholders y le informa a los integrantes del equipo, el enfoque del proyecto.

1.7.3 ¿OpenUP o XP?

Se escoge como metodología de desarrollo OpenUP sobre XP, pues ésta es una particularidad de RUP. Una gran parte de las funcionalidades de RUP han sido heredadas por OpenUP. La ventaja fundamental que tiene sobre XP es que asigna roles a las personas que trabajan en el proyecto, por lo que el trabajo se hace más eficiente y seguro.

1.8 Herramientas de Modelado de Visual.

UML (Unified Modeling Language) es un lenguaje de modelado visual, que permite modelar sistemas orientados a objetos. Ejemplos de software para el modelado en UML son: Java e IDL, JUDE Community, Omondo, Oracle JDeveloper, Visual Paradigm y Umbrello.

1.8.1 JUDE Community.

Es la libre edición con características básicas. Es rico en funcionalidad y ofrece características como la edición y la impresión de diagramas, la importación / exportación de código fuente en java, y la producción de gráficos.

Desventajas:

- El logotipo de JUDAS siempre aparece en la cabecera de los documentos impresos.

- Cuando se corre en el Sistema Operativo Linux, algunas funciones se bloquean.

1.8.2 Omondo.

Es un plugins de eclipse para poder dibujar UML. Es un software ágil y fácil de utilizar.

Desventaja:

- Solo es compatible con el IDE Eclipse.

1.8.3 Umbrello.

Umbrello es una herramienta libre para crear y editar diagramas UML, que ayuda en el proceso del desarrollo de software, maneja gran parte de los diagramas estándar UML pudiendo crearlos, además de manualmente, importándolos a partir de código en C++, Java, Python, IDL, Pascal/Delphi, Ada, o también Perl (haciendo uso de una aplicación externa). Así mismo, permite crear un diagrama y generar el código automáticamente en los lenguajes antes citados, entre otros.

Desventaja:

- Las imágenes que genera son demasiado sencillas.

1.8.4 Visual Paradigm.

Visual Paradigm para el Lenguaje Unificado de Modelado (VP-UML) es una herramienta completamente equipada para el UML, diseñada para la construcción de sistemas de software de gran escala de forma fiable a través de la utilización del enfoque orientado a objetos. VP - UML soporta los últimos estándares de Java y UML anotaciones. Además, VP - UML está integrada con todos los principales IDEs para apoyar la aplicación del desarrollo de software. El análisis de las transiciones de diseño a la aplicación se encuentran perfectamente integradas en la herramienta de modelado visual, reduciendo así significativamente los esfuerzos en todas las etapas del desarrollo del ciclo de vida del software.

1.8.4 ¿Por qué Visual Paradigm?

Visual Paradigm utiliza el enfoque orientado a objeto, tiene gran compatibilidad con la mayoría de los principales IDEs de desarrollo. Además su diseño es centrado en casos de uso y enfocado al negocio, lo cual genera un software de mayor calidad y ofrece el uso de un lenguaje estándar común a todo el equipo de desarrollo, lo cual facilita la comunicación y tiene disponibilidad en múltiples plataformas.

1.9 Lenguajes de Programación.

El lenguaje de programación permite crear la funcionalidad entre las interfaces gráficas para llevar a cabo las actividades de gestión del sistema. Entre la amplia gama de lenguajes existentes para el desarrollo Web, algunos de los más utilizados son PHP, JSP, JavaScript y Ajax.

1.9.1 PHP.

El lenguaje PHP (HyperText Pre-processor) es un lenguaje de programación de estilo clásico, es decir es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. (7)

Está diseñado especialmente para desarrollo Web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Es un lenguaje multiplataforma. Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL:

Desventajas:

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización.
- No soporta los Portlets.

1.9.2 JSP.

JSP es un acrónimo de Java Server Pages. Es una tecnología orientada a crear páginas Web con programación en Java. Es una tecnología que permite mezclar HTML estático con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas donde la CGI (Common Gateway Interface) es casi estática, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los Servlets, hacen que se genere la página completa mediante el programa, incluso aunque la mayoría de ella sea siempre lo mismo. JSP nos permite crear dos partes de forma separada.

Con JSP se crean aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP se escriben con el editor HTML/XML habitual.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones Web, almacenando en clases java las partes que consumen más recursos (así como las que requieren más seguridad) y dejando la parte encargada de formatear el documento HTML en el archivo JSP. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información.

Los Servlets y las Java Server Pages (JSPs) son dos métodos de creación de páginas Web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, los CGIs, programas que generan páginas Web en el servidor. (8)

1.9.3 Lenguajes utilizados.

Los lenguajes de programación que se describieron anteriormente son los más usados para el desarrollo Web. Cada uno de ellos tiene características que los hacen únicos, pero hasta el momento la especificación de Portlets no soporta PHP, por lo que se hace imposible utilizarlo para el desarrollo de este trabajo, por lo que se escogió JSP debido a que es una tecnología que permite mezclar HTML estático con HTML generado dinámicamente, permite integrarse con clases Java (.class). Combinado además con el lenguaje Java Script para las validaciones de los campos, debido a que éste brinda la

posibilidad de validar en el lado del cliente por lo que las operaciones se realizan más rápido. Es además un lenguaje orientado a objetos y dispone de herencia, si bien ésta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código Java Script integrado dentro de las páginas Web.

Se utilizó además la técnica de desarrollo Web Java Script Asíncrono y XML (AJAX), acrónimo de (Asynchronous Java Script And XML), para el envío de formularios, esto hace que la aplicación sea más dinámica y en caso de errores permite mostrar mensajes con el contenido del error, esto último no se logra sin la utilización de este lenguaje.

1.10 Herramientas de Programación.

1.10.1 NetBeans IDE.

El NetBeans es un IDE (Integrated Development Environment) para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. (9)

Tiene buen soporte de Portal en Perl para Automatización de Web (Webapps) (war, jsp y Servlets) y además en la actualidad cuenta con el plugin **Portal Pack** para la realización de Portlets.

Características:

- Autocompletación de código.
- Utilidades de búsqueda y reemplazo muy intuitivos, amigables y sobre todo realmente útiles para el programador.
- Coloración de código.
- Refactorización inteligente.

- Diseñadores de interfaces gráficas para Swing y JSF.
- Asistentes para escritura de código rápido de acceso a datos basados en JPA (Java Persistence API).
- Asistentes para escritura de servicios Web.
- Soporte para múltiples máquinas virtuales.
- Soporte para múltiples servidores contenedores J2EE.
- Soporte para librerías de terceros.

También es importante recordar la variedad de plugins (componentes de software conectables a NetBeans) disponibles para incrementar las funcionalidades del IDE.

1.10.2 Eclipse.

Eclipse es un entorno de desarrollo integrado de código abierto, Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados. Algunas de sus características fundamentales: Editor de texto, resaltado de sintaxis, compilación en tiempo real. Como desventaja tiene que para poder desarrollar entornos de diseño visual requiere la instalación de plugins por ejemplo el Visual Edition.

1.10.3 ¿Por qué NetBeans IDE?

Actualmente el NetBeans cuenta con un plugin para desarrollar Portlets. Además los Portlets necesitan un contenedor de Portlets que maneje su ciclo de vida y un servidor Web, debido a la compatibilidad que existe entre el Open Portal Portlet Container Driver y la herramienta de desarrollo NetBeans, se utilizará la misma.

1.11 Gestor de Base de Datos.

Un sistema Gestor o Manejador de Base de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una Base de Datos (BD) por lo tanto el SGBD es un software de propósito

general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones pueden ser de propósito general o específico.

1.11.1 PostGreSQL.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostGreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostGreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Características:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, cadenas de bits, etc.
- También permite la creación de tipos propios. Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Soporta el uso de índices, reglas y vistas. Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Desventajas

PostGreSQL es un magnífico gestor de bases de datos, capaz de competir con muchos gestores comerciales, aunque carezca de alguna característica casi imprescindible. La velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas es un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, cosa que resulta loable.

1.11.2 MySQL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software Libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Características:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multi-hilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.). Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

1.11.3 ¿Por qué MySQL?

PostgreSQL posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Pero consume gran cantidad de recursos, tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento, es de 2 a 3 veces más lento que MySQL.

Por otro lado MySQL cuenta con una velocidad increíble a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento, tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está. Además las utilidades de administración de este gestor son envidiables para

muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación, se escoge como gestor de base de datos MySQL.

1.12 Arquitectura Cliente-Servidor.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Una disposición muy común son los sistemas multicapas en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras, aumentando así el grado de distribución del sistema.

1.12.1 Ventajas de la Arquitectura Cliente-Servidor(C/S).

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.

Cada caso del software del cliente puede enviar peticiones a un servidor. Los tipos específicos de servidores incluyen los servidores Web, los servidores del uso, los servidores de archivo, los servidores terminales, y los servidores del correo. Mientras que sus propósitos varían, la arquitectura básica sigue siendo igual.

Características de un servidor:

- Voz pasiva (esclavo).
- Espera para las peticiones.
- Cuando recibe peticiones, las procesa y entonces los servicios son contestados.

Características de un cliente:

- Voz activa (amo).
- Envía peticiones, las esperas y recibe contestaciones del servidor.

Los servidores pueden ser apátridas o stateful. Un servidor apátrida no guarda ninguna información entre las peticiones. Un servidor stateful puede recordar la información entre las peticiones. El alcance de esta información puede ser global o sesión-específico. Un servidor del HTTP para las páginas estáticas del HTML es un ejemplo de un servidor apátrida mientras que Apache Tomcat es un ejemplo de un servidor stateful.

La interacción entre el cliente y el servidor se describe a menudo usando diagramas de secuencia. Los diagramas de secuencia se estandarizan en el UML.

Ventajas:

- Todos los datos se almacenan en los servidores, por lo que tienen un mayor control de la seguridad.
- Contienen tecnologías que aseguran seguridad y la facilidad de empleo.
- Cualquier elemento de la red C/S puede ser aumentado.

1.13 Servidor Web.

Un servidor es una computadora que entrega a otras computadoras (clientes), una información que ellos requieren bajo un lenguaje común, denominado protocolo. Por lo tanto al ver una página Web es porque el servidor les entrega una página HTML vía protocolo HTTP (Hypertext Transport Protocol), a través de una conexión TCP/IP por el puerto 80. Por lo tanto en el Servidor Web es donde se almacena la información estática accedida y/o las aplicaciones que la generan.

1.13.1 Cherokee.

El servidor HTTP Cherokee es un servidor Web libre, multiplataforma, abierto bajo Licencia Pública General (GPL). Es un servidor Web rápido que soporta las funcionalidades más comunes de un servidor además soporta registro y autenticación de usuarios. Cherokee puede también realizar redirecciones y soporta la configuración de Servidores Virtuales. (11)

1.13.2 Tomcat.

Tomcat es un servidor Web con soporte de Servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en Servlets. El motor de Servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache.

Tomcat puede funcionar como servidor Web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier Sistema Operativo que disponga de la máquina virtual Java.

1.13.3 Apache.

Apache está diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios Web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero no contiene una interfaz gráfica que ayude en su configuración.

1.13.4 GlassFish.

GlassFish es un servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito y de código libre. Es un proyecto Open Source modular que permite incluir sus librerías como parte de otros frameworks, y productos. GlassFish es la base de código de las distribuciones estables, certificadas y con opción de contratar soporte y mantenimiento del Servidor de Aplicaciones de Sun. GlassFish incluye gran parte de TOMCAT en el Web Container, incluyendo Jasper y Catalina, pero con especial atención en los temas de rendimiento. GlassFish incluye las nuevas librerías de Web Services (JAX-WS 2.0) y es la base de las nuevas plataformas SOA en JAVA. Es un servidor Web flexible y potente.

1.13.5 ¿Por qué GlassFish?

Para la configuración del Open Portal Portlet Container Driver que se utilizará como contenedor de Portlet, es necesario utilizar este servidor Web.

Conclusiones del Capítulo.

En el presente capítulo se explicaron los fundamentos teóricos de los Sistemas Biológicos y los Modelos Matemáticos los cuales forman parte de la información que se necesita gestionar para simular Sistemas Biológicos, además se llegó a la conclusión que el sistema que se va a desarrollar gestiona de una manera más eficiente, ágil y personalizada dicha información en comparación con los software existentes. Se seleccionó para dar solución a la problemática planteada el uso de un nuevo estándar, Portlets debido a las ventajas que brinda. Para desarrollar dicho estándar, después de analizar las diferentes técnicas y herramientas, se seleccionó como contenedor de Portlets el Open Portal Portlet Container Driver, el Sistema Operativo Linux, la metodología de desarrollo OpenUP, el lenguaje de modelado visual UML, la herramienta Case Visual Paradigm, el lenguaje de programación JSP donde podemos insertar código Java Script y Ajax y finalmente como IDE de desarrollo el NetBeans, además como gestor de base de datos el MySQL, utilizando una arquitectura C/S.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

En este capítulo se abordará de forma general los requisitos funcionales y no funcionales, los actores que intervienen en el sistema, el diagrama de Casos de Uso del sistema, así como las respectivas descripciones de los Casos de Uso.

2.1 Requisitos.

Los requisitos definen de forma precisa el producto de software que se va a construir. Se dividen en:

- **Requisitos Funcionales:** Son capacidades o condiciones que el sistema debe cumplir.
- **No Funcionales:** Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como características que hacen al producto atractivo, rápido, usable o confiable.

2.1.1 Requisitos Funcionales.

R1 Insertar solicitud para la creación de un proyecto.

R2 Gestionar Solicitud.

R2.1 Buscar solicitud.

R2.2 Mostrar solicitud.

R2.3 Aprobar solicitud.

R2.4 Rechazar solicitud.

R3 Gestionar Proyecto.

R3.1 Buscar proyecto.

R3.2 Mostrar datos del proyecto.

R3.3 Modificar proyecto.

R4 Eliminar Proyecto.

R5 Gestionar Sistema Biológico.

R5.1 Insertar sistema biológico.

R5.2 Buscar sistema biológico.

R5.3 Mostrar sistema biológico.

R5.4 Modificar sistema biológico.

R6 Eliminar Sistema Biológico.

R7 Gestionar Modelo Matemático.

R7.1 Insertar Modelo Matemático.

R7.2 Buscar Modelo Matemático.

R7.3 Modificar Modelo Matemático.

R8 Eliminar Modelo Matemático.

2.1.2 Requisitos No Funcionales.

1RNF de Usabilidad: El sistema está concebido para ser utilizado por cualquier investigador independientemente del lugar donde se encuentre. Los investigadores deben tener conocimientos básicos del manejo de las computadoras y del ambiente Web en general.

2RNF de Confiabilidad: El sistema debe garantizar la confidencialidad de los datos gestionados por el investigador.

3RNF de Rendimiento: El sistema debe ser capaz de gestionar grandes volúmenes de información sin presentar fallos.

4RNF de Apariencia o Interfaz Externa: El sistema no tendrá muchas imágenes y tendrá pocos colores.

5RNF de Extensibilidad: Se debe lograr un diseño adaptable, con la capacidad de poder soportar funcionalidades adicionales o modificar las funcionalidades existentes sin impactar el resto de los requerimientos contemplados en el sistema.

6RNF de Disponibilidad: Se garantizará que la aplicación se mantenga funcionando, con el menor tiempo posible de recuperación de fallos.

7RNF Restricciones en el Diseño y la Implementación: El análisis y diseño de la aplicación será basado en la Metodología OpenUP haciendo uso del lenguaje de modelado UML. Se usará como herramienta CASE el Visual Paradigm 5.3 para el modelado de los artefactos que se generan en cada uno de los flujos de trabajo definidos por OpenUP. Para el diseño de las interfaces se utilizará NetBeans IDE. Se usará como lenguaje de programación JSP, Ajax y Java Script. Se usará como Gestor de Base de Datos MYSQL. Podrán ser utilizados varios estándares como HTTP, HTML, XML, SOAP, UDDI. Además se utiliza como contenedor de Portlets Open Portal Portlet Container Driver y como servidor Web el GlassFish.

8RNF Portabilidad: El sistema debe ser capaz de correr en cualquier sistema operativo, pues las funcionalidades del sistema están implementadas en Java, la cual es multiplataforma.

9RNF Extensibilidad: El sistema debe ser capaz de integrarse con otros módulos, además de permitir la inserción de cambios.

2.2 Definición de los Actores del Sistema.

Los actores del sistema:

- No forman parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar un rol que juega una o varias personas, un equipo o un sistema automatizado.

Nombre del actor	Descripción
Trabajador del proyecto.	Persona que realiza las funcionalidades siguientes: gestionar un proyecto, gestionar un Sistema Biológico y gestionar un Modelo Matemático.
Líder del proyecto.	Persona que dirige el proyecto, es quien solicita la creación de un nuevo proyecto, una vez creado puede: gestionar un Proyecto, eliminar un proyecto, gestionar un Sistema Biológico, eliminar un Sistema Biológico, gestionar un Modelo Matemático y eliminar un Modelo Matemático
Administrador	Persona que administra el sistema puede realizar todas las funcionalidades que brinda el sistema, además es el encargado de aprobar o rechazar una solicitud para la creación de un nuevo proyecto.

Tabla 2. 1 Actores del Sistema

2.3 Definición de Casos de Uso del Sistema.

Los Casos de Uso (CU) son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Debido a la gran cantidad de CU, se agrupan en paquetes como se muestra a continuación:

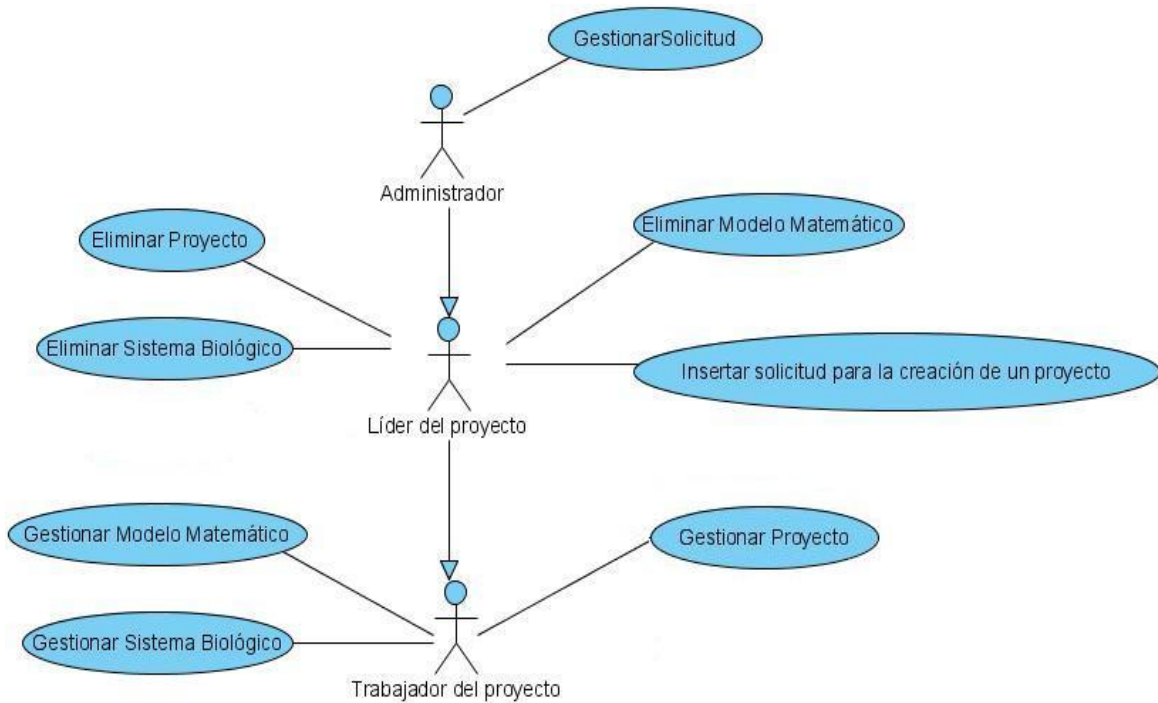


Figura 2. 1 CU del Sistema

2.4 Descripción de los CU del Sistema.

Para entender la funcionalidad asociada a cada caso de uso no es suficiente con la representación gráfica del Diagrama CU del Sistema por lo que se han descrito textualmente a continuación:

Insertar Solicitud para la creación de un proyecto

Nombre del CU	Insertar solicitud para la creación de un proyecto.
Actores	Líder del proyecto.
Propósito	Solicitar la creación del proyecto
Resumen	El Caso de Uso se inicia cuando el Líder del proyecto selecciona la opción solicitar la creación de un proyecto que no es más que insertar una solicitud.
Referencias	R1

Prototipos	Ver Figura 4.6
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	Queda insertada la solicitud en el sistema.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Líder del proyecto decide insertar una solicitud de creación de un proyecto.	2. El sistema muestra el formulario con los datos a insertar de la nueva solicitud.
3. El Líder del proyecto introduce el nombre de la solicitud, el líder del proyecto, los trabajadores del proyecto y una descripción del proyecto.	4. El sistema verifica que los datos introducidos sean correctos.
	5. Si los datos son correctos el sistema inserta la solicitud con los datos introducidos por el Líder del proyecto, muestra un mensaje de confirmación y termina el caso de uso.
Curso Alternativo de Eventos	
	5.1 Si existe algún dato incorrecto o algún campo vacío, el sistema muestra un mensaje de error señalando el campo incorrecto y finaliza el caso de uso.
Prioridad	Crítico

Tabla 2. 2 Descripción del CU Insertar Solicitud para la creación de un proyecto

Gestionar Solicitud.

Nombre del CU	Gestionar Solicitud
Actores	Administrador
Propósito	Gestionar solicitud.
Resumen	El caso de uso se inicia cuando el administrador decide gestionar una

	solicitud en el sistema. El administrador puede ver las solicitudes existentes, aceptarlas o rechazarlas.	
Referencias	R2.1, R2.2, R2.3, R2.4.	
Prototipos	Ver Figura 4.7	
Precondiciones	El usuario debe estar autenticado.	
Poscondiciones	Quedan actualizadas las solicitudes en el sistema.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El administrador decide gestionar una solicitud.	2. El sistema muestra la(s) solicitud(es) existentes junto a las opciones de Aceptar y Rechazar.	
3. El administrador puede: a) visualizar las solicitudes. Las visualiza y termina el caso de uso. b) aceptar una solicitud para insertar un nuevo proyecto. Ir a la Sección Aceptar Solicitud. c) rechazar una solicitud. Ir a la Sección Rechazar Solicitud.		
Sección Aceptar Solicitud		
4. El administrador selecciona la solicitud para insertar ese nuevo proyecto.	5. El sistema inserta el nuevo proyecto.	
	6. El sistema elimina la solicitud y finaliza el caso de uso.	
Sección Rechazar Solicitud		
7. El administrador selecciona la solicitud que desea rechazar.	8. El sistema elimina la solicitud y finaliza el caso de uso.	
Curso Alternativo de Eventos		
7.1 El administrador selecciona la opción	7.2 El sistema muestra el mensaje que debe	

Aceptar/Rechazar sin haber seleccionado una solicitud.	marcar una solicitud previamente.
Prioridad	Crítico

Tabla 2. 3 Descripción del CU Gestionar Solicitud.

Gestionar Proyecto.

Nombre del CU	Gestionar Proyecto.	
Actores	Trabajador del proyecto.	
Propósito	Gestionar proyecto.	
Resumen	El caso de uso se inicia cuando el trabajador del proyecto decide gestionar un proyecto en el sistema. El trabajador del proyecto puede ver los proyectos existentes y modificarlos.	
Referencias	R3.1, R3.2, R3.3.	
Prototipo	Ver figuras 4.8, 4.10, 4.11.	
Precondiciones	El usuario debe estar autenticado.	
Poscondiciones	Quedan actualizados los proyectos en el sistema.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El trabajador del proyecto decide gestionar un proyecto.	2. El sistema muestra el formulario para buscar el(los) proyecto(s).	
3. El Trabajador del proyecto puede: a) Buscar un determinado proyecto. b) Buscar todos los proyectos existentes.	4. El sistema muestra el(los) proyecto(s) junto a las opciones de Aceptar y Rechazar.	
5. El trabajador del proyecto puede: a) visualizar los proyectos. Los visualiza y termina el caso de uso. b) modificar un proyecto. Ir a la Acción 6		

6. El trabajador del proyecto selecciona el proyecto que desea modificar.	7. El sistema busca los datos del proyecto y los muestra para ser modificados.
8. El trabajador del proyecto modifica, de los datos mostrados, aquellos que desea modificar.	9. El sistema verifica que los datos introducidos sean correctos.
	10. Si los datos son correctos, el sistema actualiza los datos del proyecto y finaliza el caso de uso.
Curso Alternativo de Eventos	
	10.1 Si los datos son incorrectos, el sistema envía un mensaje de error señalando los datos introducidos incorrectamente y finaliza el caso de uso.
Prioridad	Crítica

Tabla 2. 4 Descripción del CU Gestionar Proyecto.

Eliminar Proyecto.

Nombre del CU	Eliminar Proyecto.
Actores	Líder del proyecto.
Propósito	Eliminar un proyecto.
Resumen	El caso de uso se inicia cuando el líder del proyecto decide eliminar un proyecto en el sistema.
Referencias	R4.
Prototipo	Ver figuras 4.19, 4.9.
Precondiciones	El usuario debe estar autenticado. Debe existir un proyecto en el sistema.
Poscondiciones	Queda eliminado un proyecto en el sistema.
Curso Normal de los Eventos	

Acciones del Actor	Respuesta del Sistema
1. El líder del proyecto decide eliminar un proyecto.	2. El sistema muestra el formulario para buscar proyectos.
3. El líder del proyecto selecciona la opción de buscar los proyectos.	4. El sistema busca y muestra el(los) proyecto(s) existente(s) junto a la opción de Eliminar.
5. El líder del proyecto selecciona el proyecto que desea eliminar junto con la opción de Eliminar.	6. El sistema elimina el proyecto seleccionado por el líder del proyecto y finaliza el caso de uso.
Curso Alternativo de Eventos	
5.1 El líder del proyecto selecciona la opción de Eliminar.	5.2 El sistema muestra un mensaje que debe marcar un proyecto previamente y finaliza el caso de uso.
Prioridad	Crítico

Tabla 2. 5 Descripción del CU Eliminar Proyecto.

Gestionar Sistema Biológico.

Nombre del CU	Gestionar Sistema Biológico.
Actores	Trabajador del proyecto.
Propósito	Gestionar sistema biológico.
Resumen	El caso de uso se inicia cuando el trabajador del proyecto decide gestionar un sistema biológico en el sistema. El trabajador del proyecto puede insertar, ver los Sistemas Biológicos y modificarlos.
Referencias	R5.1, R5.2, R5.3, R5.4.
Prototipo	Ver figuras 4.12, 4.13.
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	Quedan actualizados los Sistemas Biológicos en el sistema.
Curso Normal de los Eventos	

Acciones del Actor	Respuesta del Sistema
<p>1. El trabajador del proyecto decide gestionar un sistema biológico. El trabajador puede:</p> <p>a) insertar uno nuevo. Ir a la acción 2 del Curso Normal de los Eventos.</p> <p>b) visualizar o modificar algún sistema biológico. Ir a la Sección Modificar Sistema Biológico.</p>	<p>2. El sistema muestra el formulario para insertar un sistema biológico.</p>
<p>3. El trabajador del proyecto introduce el nombre del sistema biológico y el SBML que desea insertar.</p>	<p>4. El sistema verifica que los datos estén correctos.</p>
	<p>5. Si los datos son correctos, el sistema guarda el nuevo sistema biológico, envía un mensaje de confirmación y finaliza el caso de uso.</p>
<p>Curso Alternativo de Eventos</p> <p>Curso Alternativo 1</p>	
	<p>5.1 El sistema envía mensaje de error pues los datos introducidos son incorrectos y finaliza el caso de uso.</p>
<p>Sección Modificar Sistema Biológico.</p>	
<p>1. El trabajador del proyecto selecciona la opción de buscar los sistemas biológicos.</p>	<p>2. El sistema muestra el(los) sistema(s) biológico(s) junto a la opción de Modificar.</p>
<p>3. El trabajador del proyecto puede:</p> <p>a) visualizar los sistemas mostrados. Los visualiza y finaliza el caso de uso.</p>	

b) modificar algún sistema mostrado. Ir a la Acción 4.	
4. El trabajador del proyecto selecciona el sistema biológico junto con la opción de Modificar.	5. El sistema busca los datos del sistema biológico y los muestra para ser modificados.
6. El trabajador del proyecto modifica los datos que desea.	7. El sistema verifica que los datos introducidos estén correctamente.
	8. Si los datos son correctos, actualiza el sistema biológico y finaliza el caso de uso.
Curso Alternativo de Eventos	
Curso Alternativo 2	
	2.1 El sistema no muestra pues no existe algún sistema biológico.
Curso Alternativo de Eventos	
Curso Alternativo 3	
	8.1 Si los datos no son correctos, muestra un mensaje de error señalando los campos incorrectos y finaliza el caso de uso.
Prioridad	Crítico

Tabla 2. 6 Descripción del CU Gestionar Sistema Biológico.

Eliminar Sistema Biológico.

Nombre del CU	Eliminar Sistema Biológico.
Actores	Líder del proyecto.
Propósito	Eliminar proyecto.
Resumen	El caso de uso se inicia cuando el líder del proyecto decide eliminar un sistema biológico en el sistema
Referencias	R6.
Prototipo	Ver figuras 4.14, 4.19.

Precondiciones	El usuario debe estar autenticado. Debe existir al menos un sistema biológico en el sistema.	
Poscondiciones	Queda eliminado un sistema biológico en el sistema.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El líder del proyecto decide eliminar un sistema biológico.	2. El sistema muestra el formulario para buscar sistemas biológicos.	
3. El líder del proyecto selecciona la opción de buscar los sistemas biológicos.	4. El sistema busca y muestra el(los) sistema(s) biológico(s) existente(s) junto a la opción de Eliminar.	
5. El líder del proyecto selecciona el sistema biológico que desea eliminar junto con la opción de Eliminar.	6. El sistema elimina el sistema biológico seleccionado por el líder del proyecto si no tiene asociado algún Modelo Matemático y finaliza el caso de uso.	
Curso Alternativo de Eventos		
Curso Alternativo 1		
5.1 El líder del proyecto selecciona la opción de Eliminar.	5.2 El sistema muestra un mensaje que debe marcar un sistema biológico previamente y finaliza el caso de uso.	
Curso Alternativo de Eventos		
Curso Alternativo 2		
	4.1. El sistema busca y no muestra nada pues no existe ningún sistema biológico y finaliza el caso de uso.	
Curso Alternativo de Eventos		
Curso Alternativo 3		
	6.1 Si el sistema biológico tiene algún Modelo Matemático asociado, el sistema muestra un mensaje de alerta, no lo elimina y finaliza el	

	caso de uso.
Prioridad	Crítico

Tabla 2. 7 Descripción del CU Eliminar Sistema Biológico.

Gestionar Modelo Matemático.

Nombre del CU	Gestionar Modelo Matemático	
Actores	Trabajador del proyecto.	
Propósito	Gestionar Modelo Matemático.	
Resumen	El caso de uso se inicia cuando el trabajador del proyecto decide gestionar un Modelo Matemático en el sistema. El trabajador del proyecto puede insertar, ver los datos de un Modelo Matemático y modificarlo.	
Referencias	R7.1, R7.2, R7.3.	
Prototipo	Ver figuras 4.15, 4.16, 4.18.	
Precondiciones	El usuario debe estar autenticado.	
Poscondiciones	Quedan actualizados los Modelos Matemáticos en el sistema.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El trabajador del proyecto decide gestionar un Modelo Matemático. El trabajador del proyecto puede: a) insertar uno nuevo. Ir a la acción 2 del Curso Normal de Eventos. a) decide visualizar o modificar alguno existente. Ir a la Sección Modificar Modelo Matemático.	2. El sistema muestra el formulario para insertar un Modelo Matemático.	
3. El trabajador del proyecto selecciona el sistema biológico, introduce el nombre del	4. Si los datos son correctos, el sistema guarda el Modelo Matemático y finaliza el caso de uso.	

Modelo Matemático, los parámetros y las variables.	
Curso Alternativo de Eventos	
Curso Alternativo 1	
	4.1 Si los datos no son correctos, el sistema muestra un mensaje de error señalando los datos incorrectos y finaliza el caso de uso.
Sección Modificar Modelo Matemático	
1. El trabajador del proyecto selecciona la opción de buscar los Modelos Matemáticos.	2. El sistema muestra el(los) Modelo(s) Matemático(s) junto a la opción de Modificar.
3. El trabajador del proyecto puede: a) visualizar los modelos mostrados. Los visualiza y finaliza el caso de uso. b) modificar algún modelo mostrado. Ir a la Acción 4.	
4. El trabajador del proyecto selecciona el Modelo Matemático junto con la opción de Modificar.	5. El sistema muestra una interfaz con el nombre del modelo seleccionado por el trabajador del proyecto.
6. El trabajador del proyecto selecciona el nombre del Modelo Matemático.	6. El sistema busca los datos del Modelo Matemático y los muestra para ser modificados.
7. El trabajador del proyecto modifica los datos que desea.	8. El sistema verifica que los datos introducidos estén correctamente.
	9. Si los datos son correctos, actualiza el Modelo Matemático y finaliza el caso de uso.
Curso Alternativo de Eventos	
Curso Alternativo 2	
	2.1 El sistema no muestra pues no existe algún sistema biológico.
Curso Alternativo de Eventos	

Curso Alternativo 3	
	8.1 Si los datos no son correctos, muestra un mensaje de error señalando los campos incorrectos y finaliza el caso de uso.
Prioridad	Crítico

Tabla 2. 8 Descripción del CU Gestionar Modelo Matemático.

Eliminar Modelo Matemático.

Nombre del CU	Eliminar Modelo Matemático.	
Actores	Líder del proyecto.	
Propósito	Eliminar Modelo Matemático.	
Resumen	El caso de uso se inicia cuando el líder del proyecto decide eliminar un Modelo Matemático en el sistema.	
Referencias	R8.	
Prototipo	Ver figura 4.19, 4.17.	
Precondiciones	El usuario debe estar autenticado. Debe existir al menos un Modelo Matemático en el sistema.	
Poscondiciones	Queda eliminado un Modelo Matemático en el sistema.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El líder del proyecto decide eliminar un Modelo Matemático.	2. El sistema busca los Sistemas Biológicos existentes y los muestra.	
3. El líder del proyecto escoge el sistema biológico en el cual se encuentra el Modelo Matemático a eliminar.	2. El sistema muestra el(los) modelo(s) matemático(s) que se corresponden al sistema biológico seleccionado junto con la opción de Eliminar.	
4. El líder del proyecto selecciona el Modelo Matemático y selecciona la opción	5. El sistema elimina el Modelo Matemático seleccionado por el líder del proyecto y finaliza	

de eliminar.	el caso de uso.
Curso Alternativo de Eventos	
Curso Alternativo 1	
4.1 El líder del proyecto selecciona la opción de eliminar.	4.2 El sistema muestra mensaje de error pues debe seleccionar un Modelo Matemático previamente y finaliza el caso de uso.
Prioridad	Crítico

Tabla 2. 9 Descripción del CU Eliminar Modelo Matemático.

Conclusiones del Capítulo.

En este capítulo se describieron los requisitos funcionales y no funcionales, se identificaron los actores y los CU que intervienen en el sistema, mostrando las relaciones entre ellos a través del diagrama de CU del sistema y sus respectivas descripciones. Estas actividades permitirán diseñar las clases con vista a la implementación.

CAPÍTULO 3: DISEÑO DEL SISTEMA

En este capítulo se describen los estilos arquitectónicos y patrones del diseño evidenciando su utilización. Además de los diferentes artefactos que se generan en el diseño del sistema como son: diagramas de clases, diagramas de interacción y modelo de despliegue.

3.1 Descripción de Estilos Arquitectónicos.

3.1.1 Patrón "Modelo-Vista-Controlador".

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

- **Modelo:** Esta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario. (muestra información del modelo de usuario)
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

3.2 Patrones de Diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño.

3.2.1 ¿Por qué utilizar patrones de diseño?

Los patrones de diseño son muy importantes pues permiten la producción de software más resistentes al cambio, permiten la reutilización del código, ayudan a especificar interfaces y al uso de una documentación estándar.

3.2.2 Patrones GRASP.

Los patrones GRASP (Patrones de software para la asignación General de Responsabilidad) describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. (12) Se pueden destacar en este trabajo los siguientes patrones:

- **Experto** es un patrón que se utiliza para asignar una responsabilidad al experto en la información. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Este patrón provee un bajo nivel de acoplamiento lo que favorece el hecho de tener sistemas de fácil mantenimiento y más robustos.

Problema: ¿Cómo asignar responsabilidades, de la forma más eficiente?

Solución: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

- **Bajo acoplamiento** soporta el diseño de clases más independientes que reducen el impacto a primeros cambios que aumentan la oportunidad de una mayor productividad e influye en la decisión de asignar responsabilidades, son fáciles de entender por separado y fáciles de reutilizar.

Problema: ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización?

Solución: Asignar una responsabilidad para mantener bajo acoplamiento.

- **Controlador** asigna la responsabilidad del manejo de mensajes de los eventos del sistema. Además favorece un mayor potencial de los componentes reutilizables. Garantiza que la empresa o los procesos de dominio sean manejados por la capa de los objetos de dominio y no por la de la interfaz.

Problema: ¿Quién debería encargarse de atender un evento del sistema?

Solución: Asignar la responsabilidad del manejo de un mensaje de los eventos del sistema a una clase que represente una de las siguientes opciones: Controlador de fachada, controlador de tareas o controlador de Casos de Uso.

3.3 Diagramas de Clases del Diseño

Es una representación más concreta que el diagrama de clases del análisis. Representa la parte estática del sistema, las clases y sus relaciones. A continuación se representan los diagramas de clases del diseño de los CU Gestionar Sistema Biológico, Eliminar Sistema Biológico, Gestionar Modelo Matemático y Eliminar Modelo Matemático, el resto de los diagramas se encuentran en los Anexos.

En Los diagramas cuando se hace referencia a la clase Acceso a Datos se refiere a la que se representa a continuación:

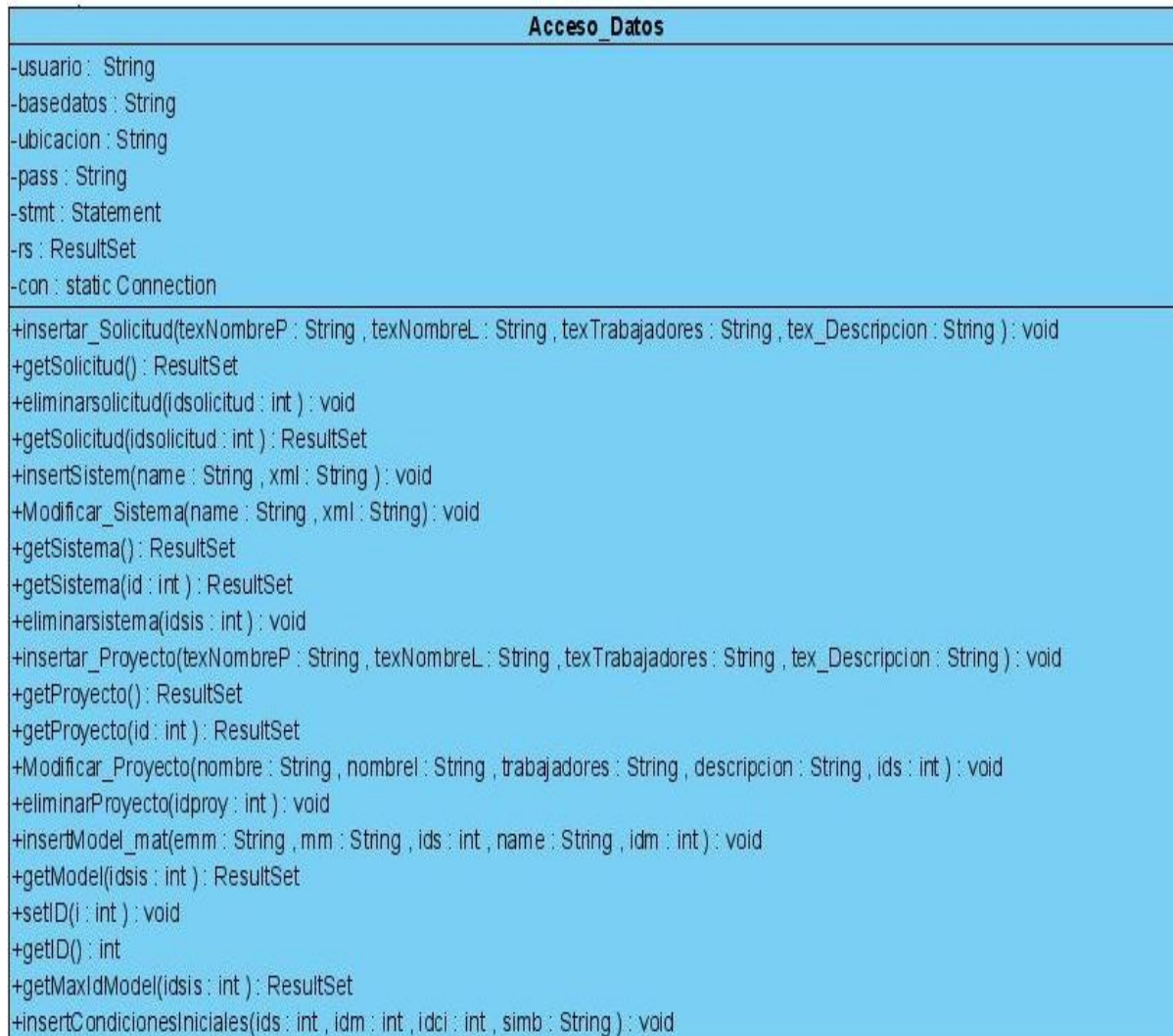


Figura 3. 1 Clase Acceso Datos

Gestionar Sistema Biológico

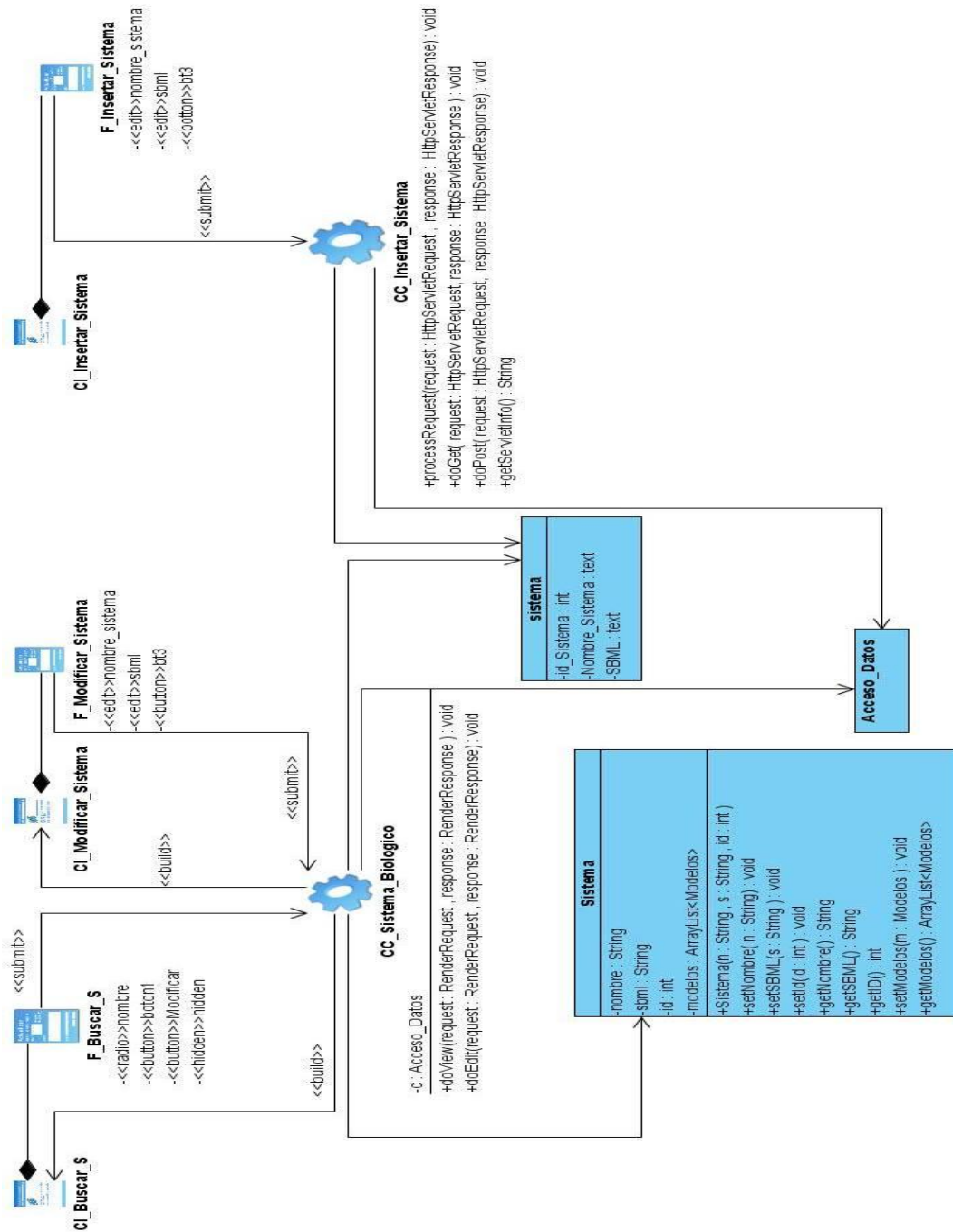


Figura 3. 2 Gestionar Sistema Biológico

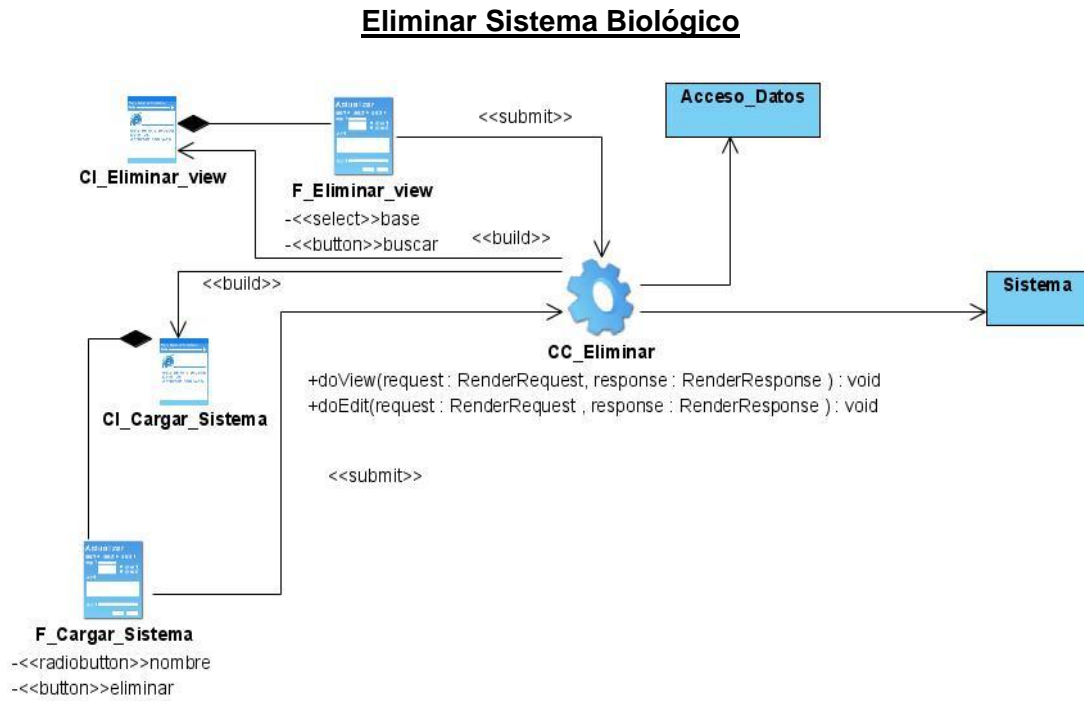


Figura 3. 3 Eliminar Sistema Biológico

Gestionar Modelo Matemático

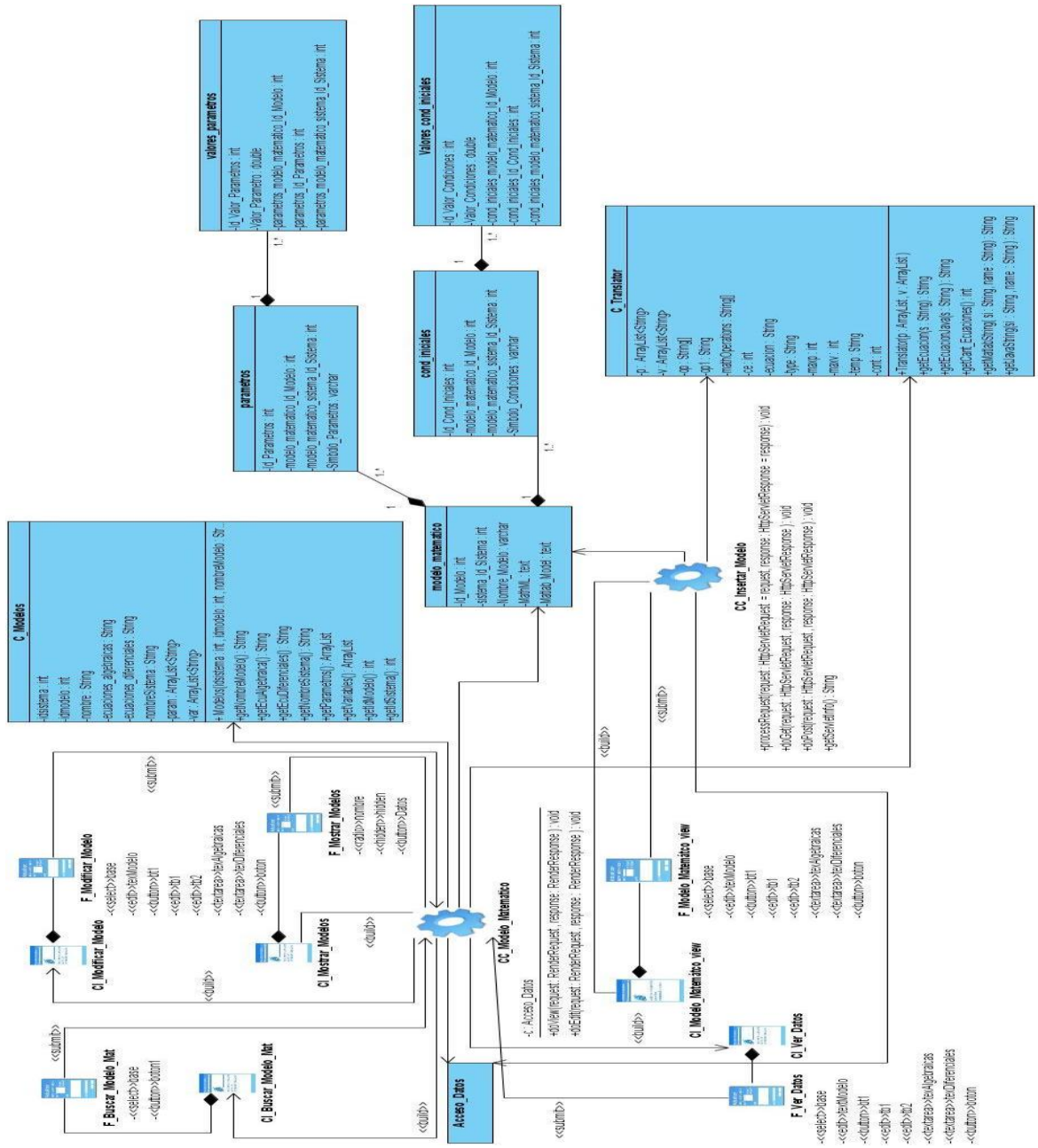


Figura 3. 4 Gestionar Modelo Matemático.

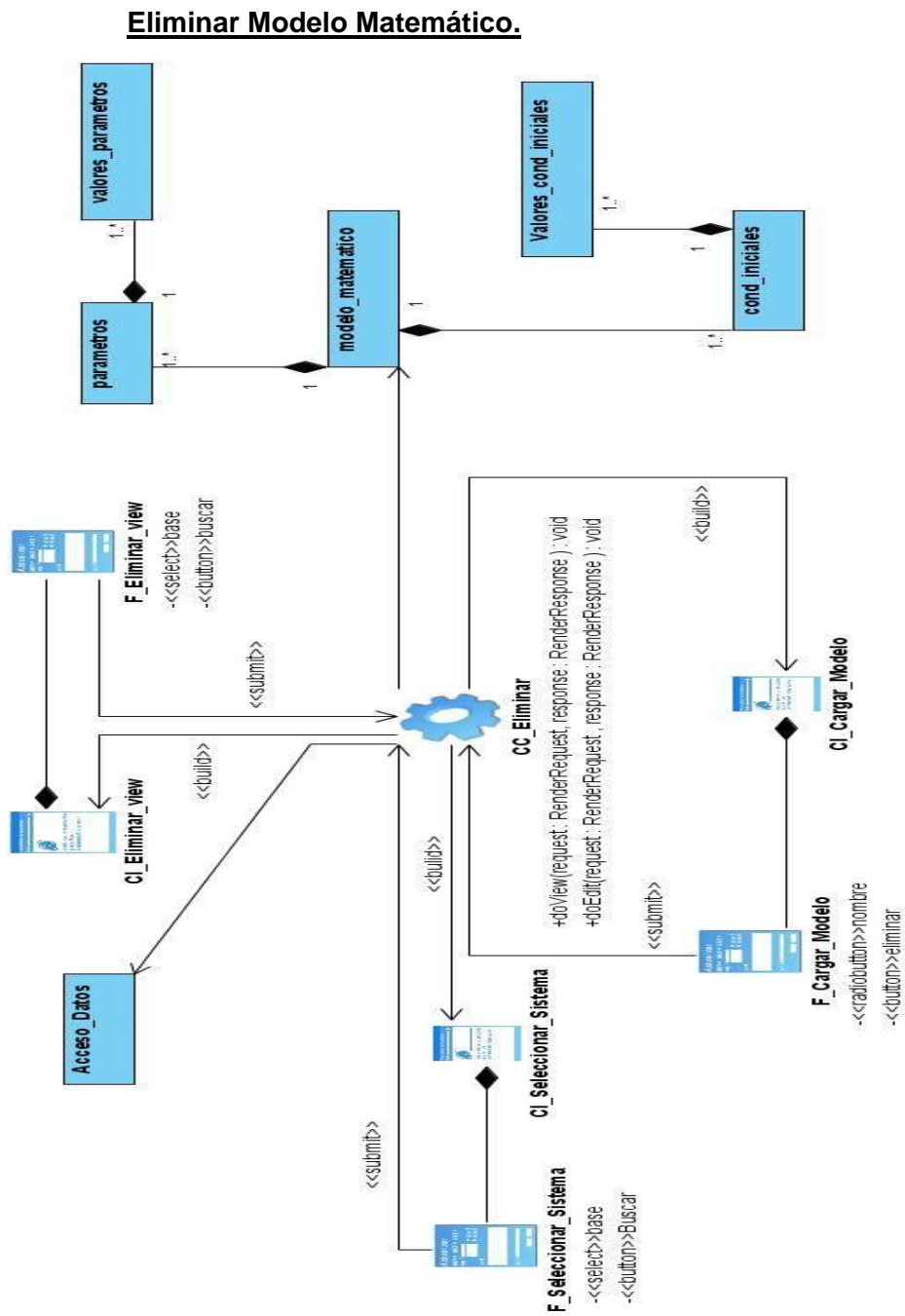


Figura 3. 5 Eliminar Modelo Matemático.

3.4 Diagramas de Interacción.

El diagrama de interacción del sistema es el encargado de presentar de manera gráfica la secuencia de acciones en un CU ya que el objetivo es encontrar secuencias de interacciones detalladas y ordenadas. Estos se representan a través de diagramas de secuencia y/o colaboración.

3.4.1 Diagrama de secuencia.

Un diagrama de secuencia muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto. Si los Casos de Uso tienen varios flujos o sub-flujos distintos, es útil crear un diagrama de secuencia para cada uno de ellos, a continuación se muestran los diagramas de interacción correspondientes a los CU Gestionar Sistema Biológico, Eliminar Sistema Biológico, Gestionar Modelo Matemático y Eliminar Modelo Matemático.

Insertar Solicitud para la creación de un Proyecto.

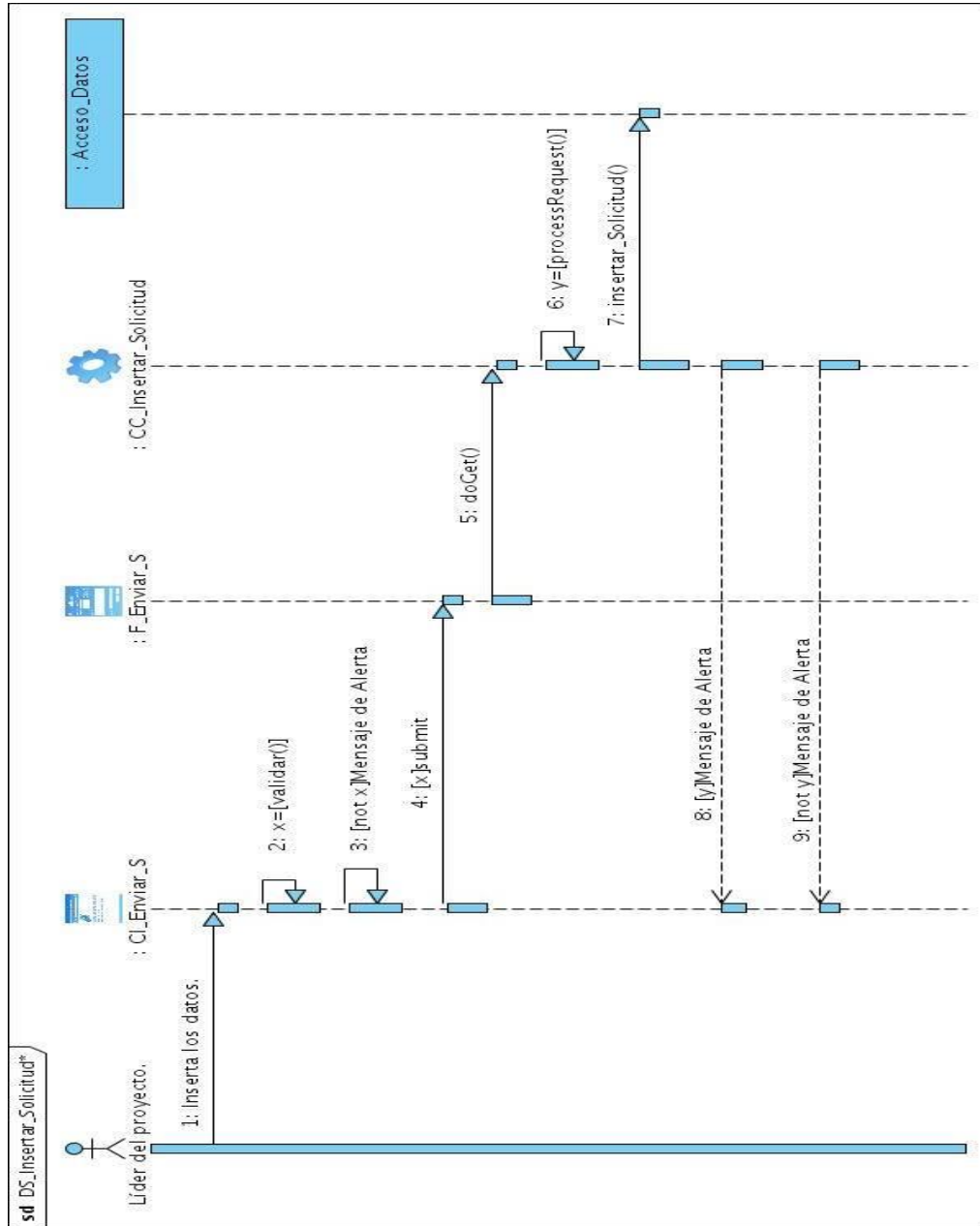


Figura 3. 6 Insertar Solicitud para la creación de un Proyecto

Gestionar Solicitud. Escenario (Buscar)

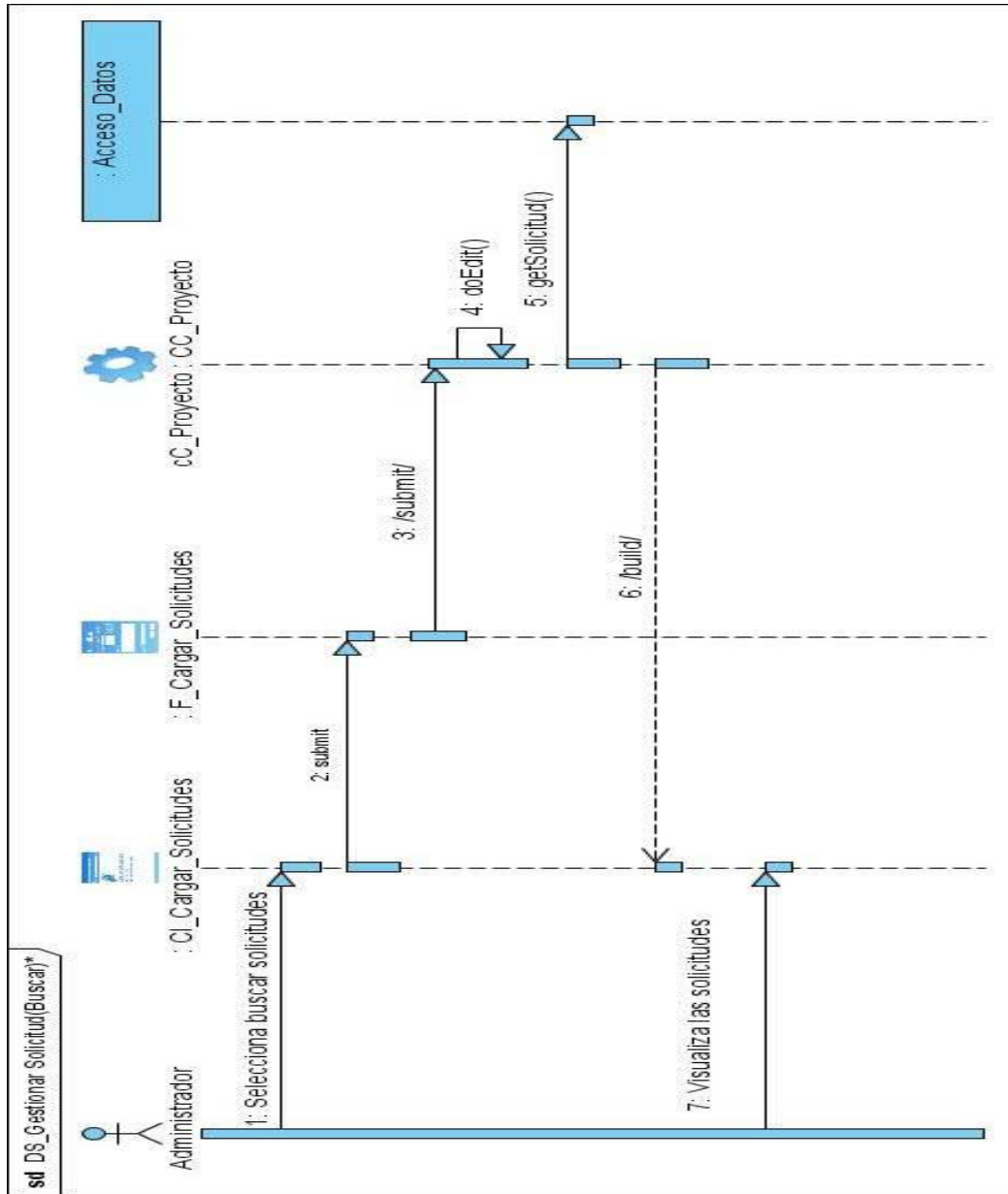


Figura 3. 7 Gestionar Solicitud. Escenario (Buscar)

Gestionar Solicitud. Escenario (Acepta)

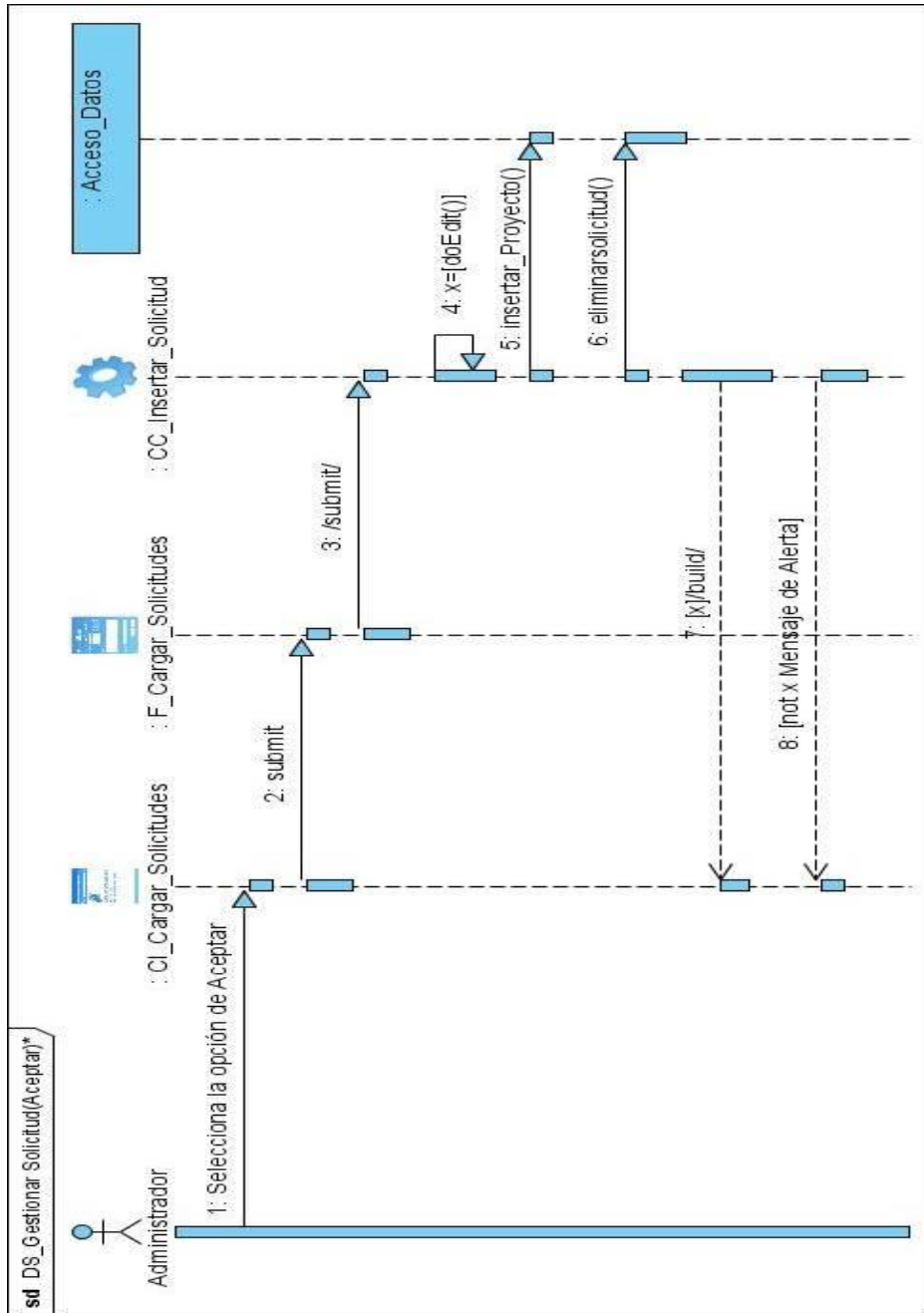


Figura 3. 8 Gestionar Solicitud. Escenario (Acepta)

Gestionar Solicitud. Escenario (Rechazar)

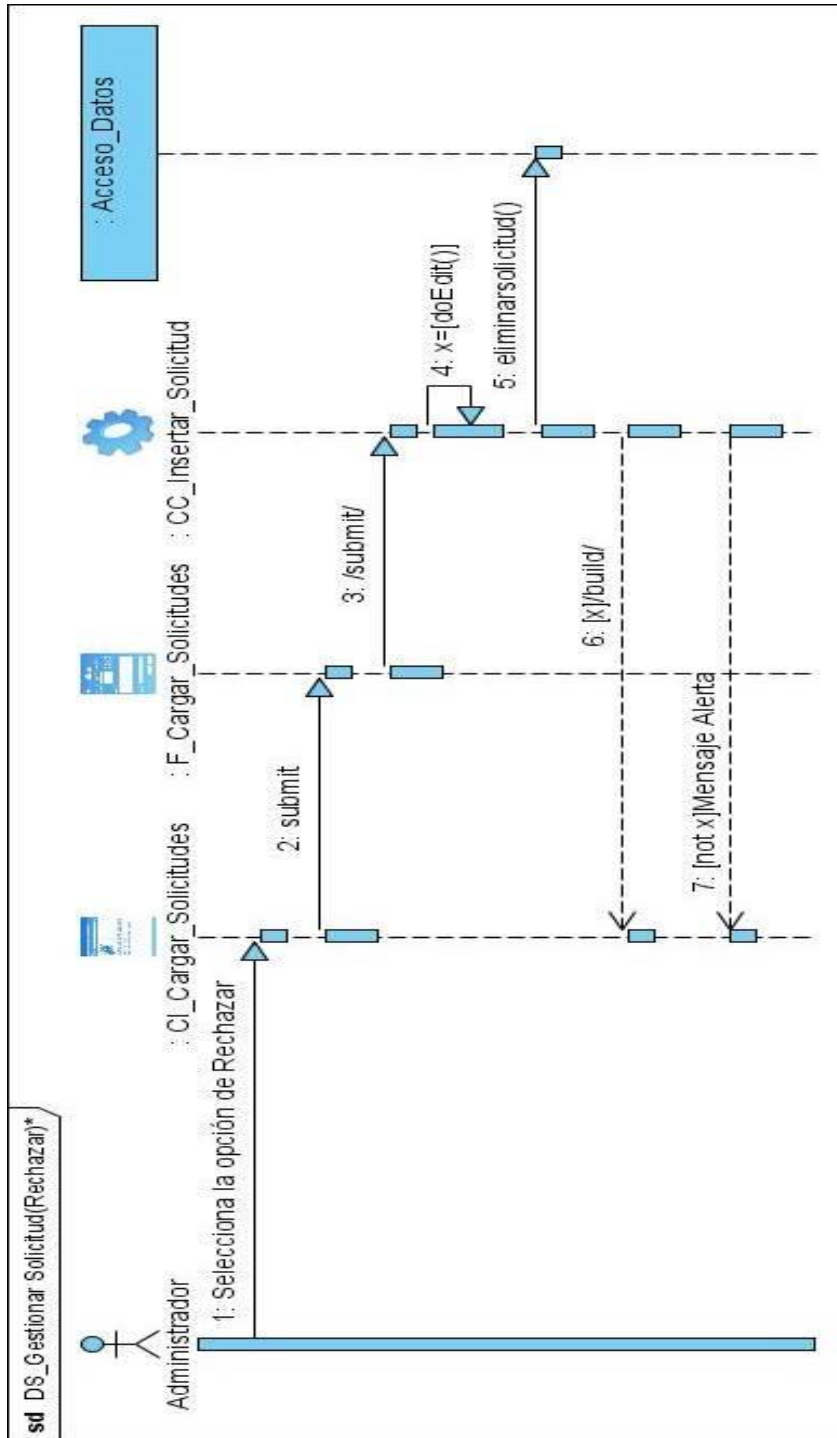


Figura 3. 9 Gestionar Solicitud. Escenario (Rechazar)

Gestionar Proyecto. Escenario (Buscar)

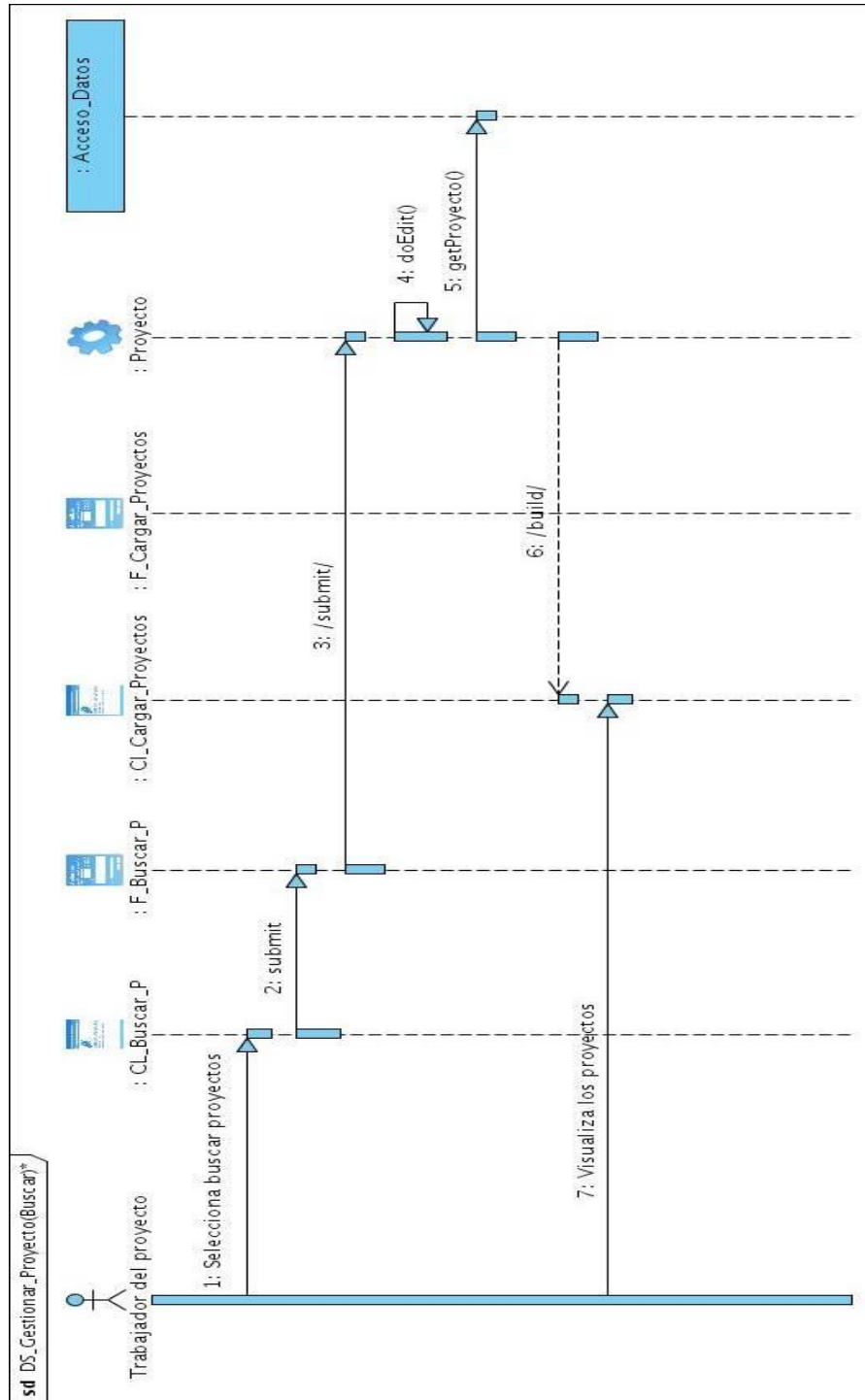


Figura 3. 10 Gestionar Proyecto. Escenario (Buscar)

Gestionar Proyecto. Escenario (Modificar)

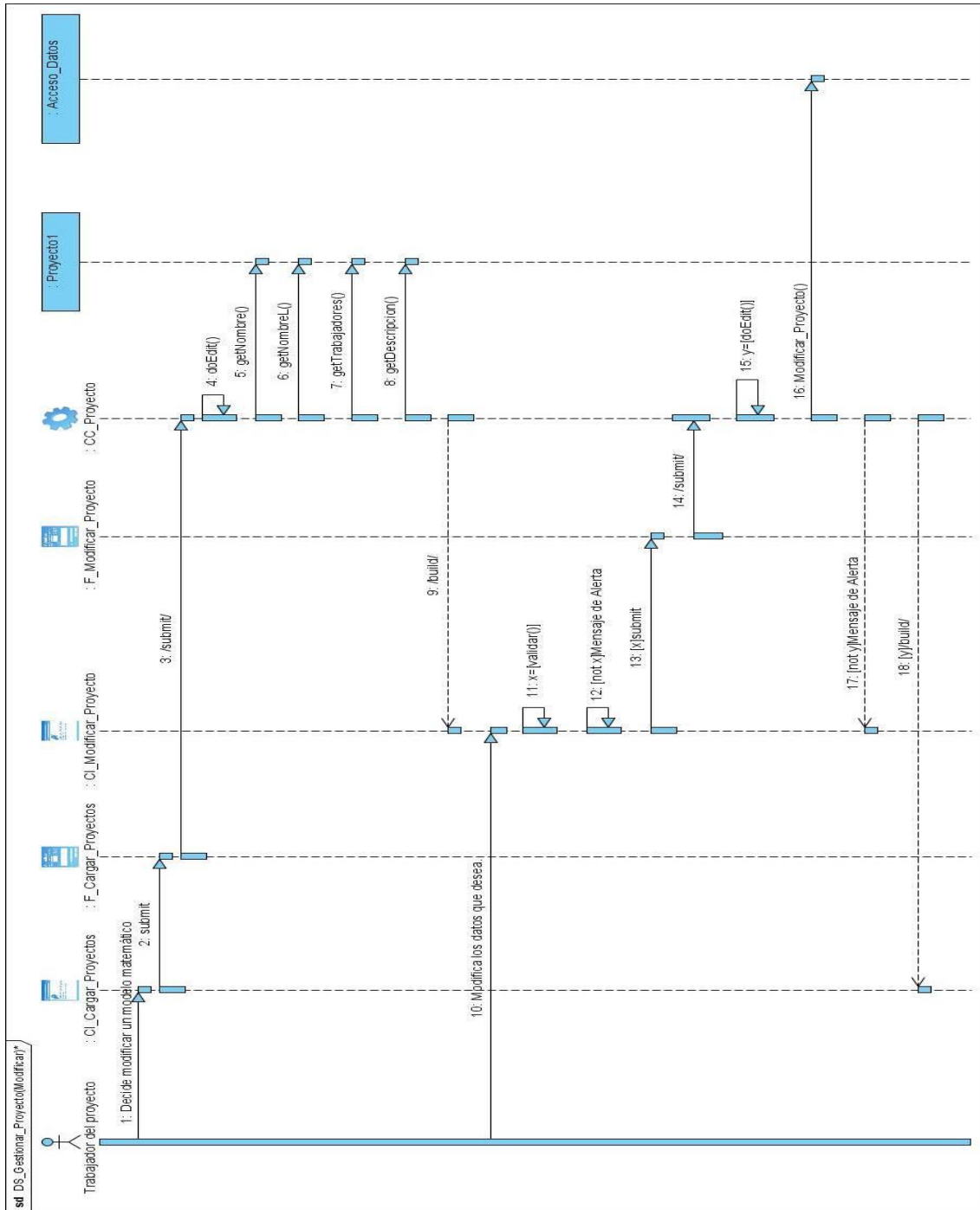


Figura 3. 11 Gestionar Proyecto. Escenario (Modificar)

Eliminar Proyecto. Escenario (Buscar)

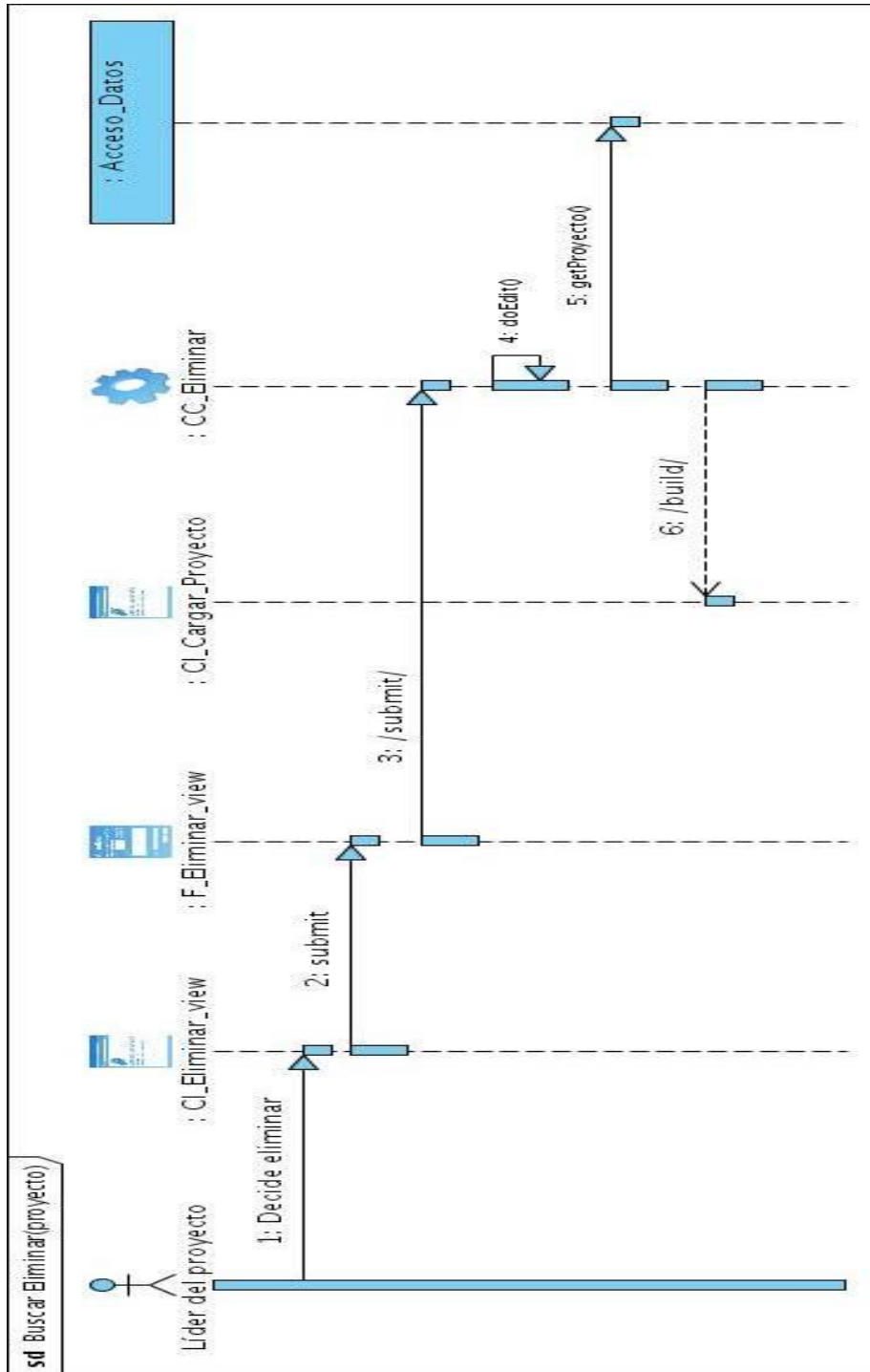


Figura 3. 12 Eliminar Proyecto. Escenario (Buscar)

Eliminar Proyecto

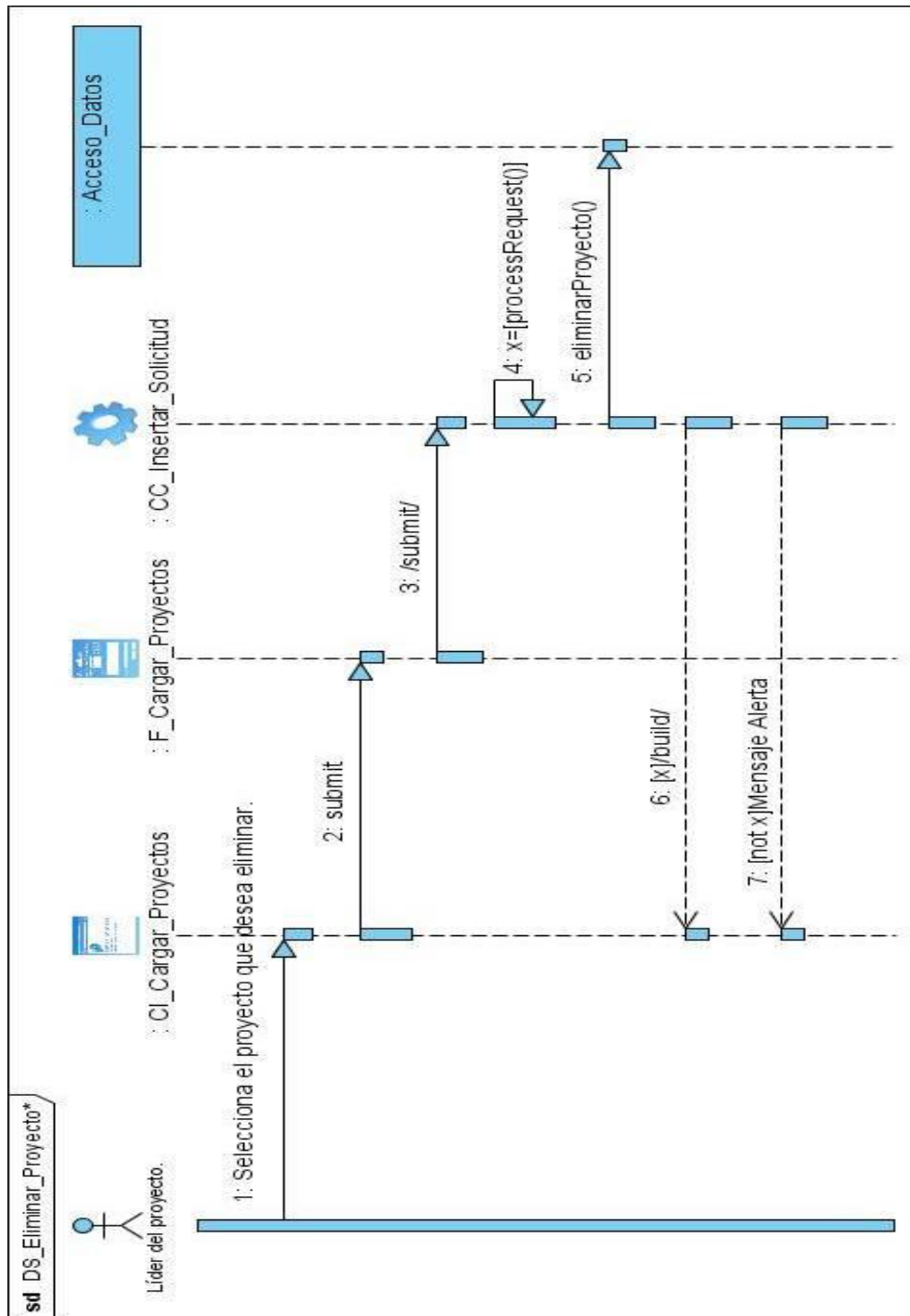


Figura 3. 13 Eliminar Proyecto

Gestionar Sistema Biológico. Escenario (Insertar)

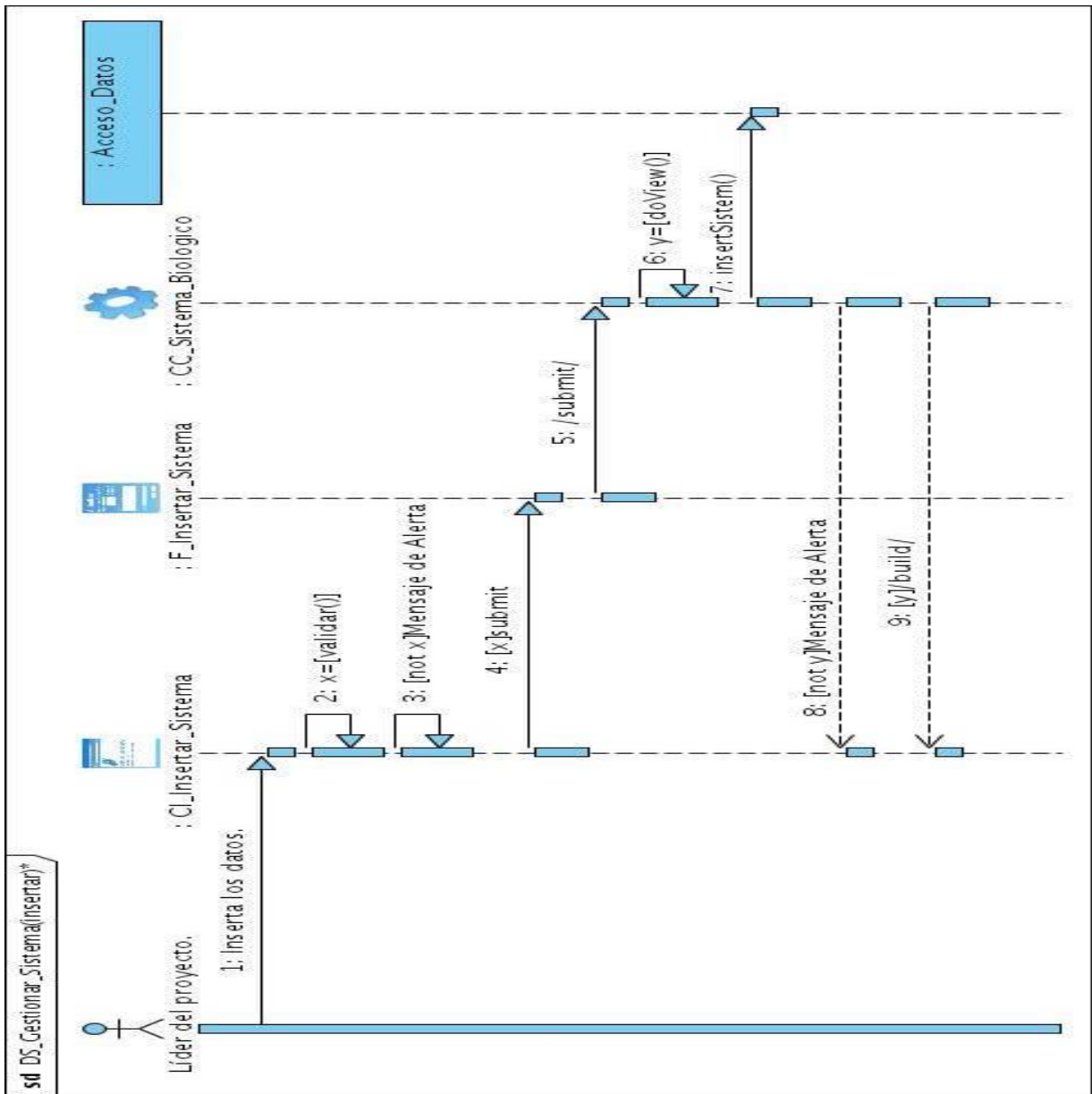


Figura 3. 14 Gestionar Sistema Biológico. Escenario (Insertar).

Gestionar Sistema Biológico. Escenario (Buscar).

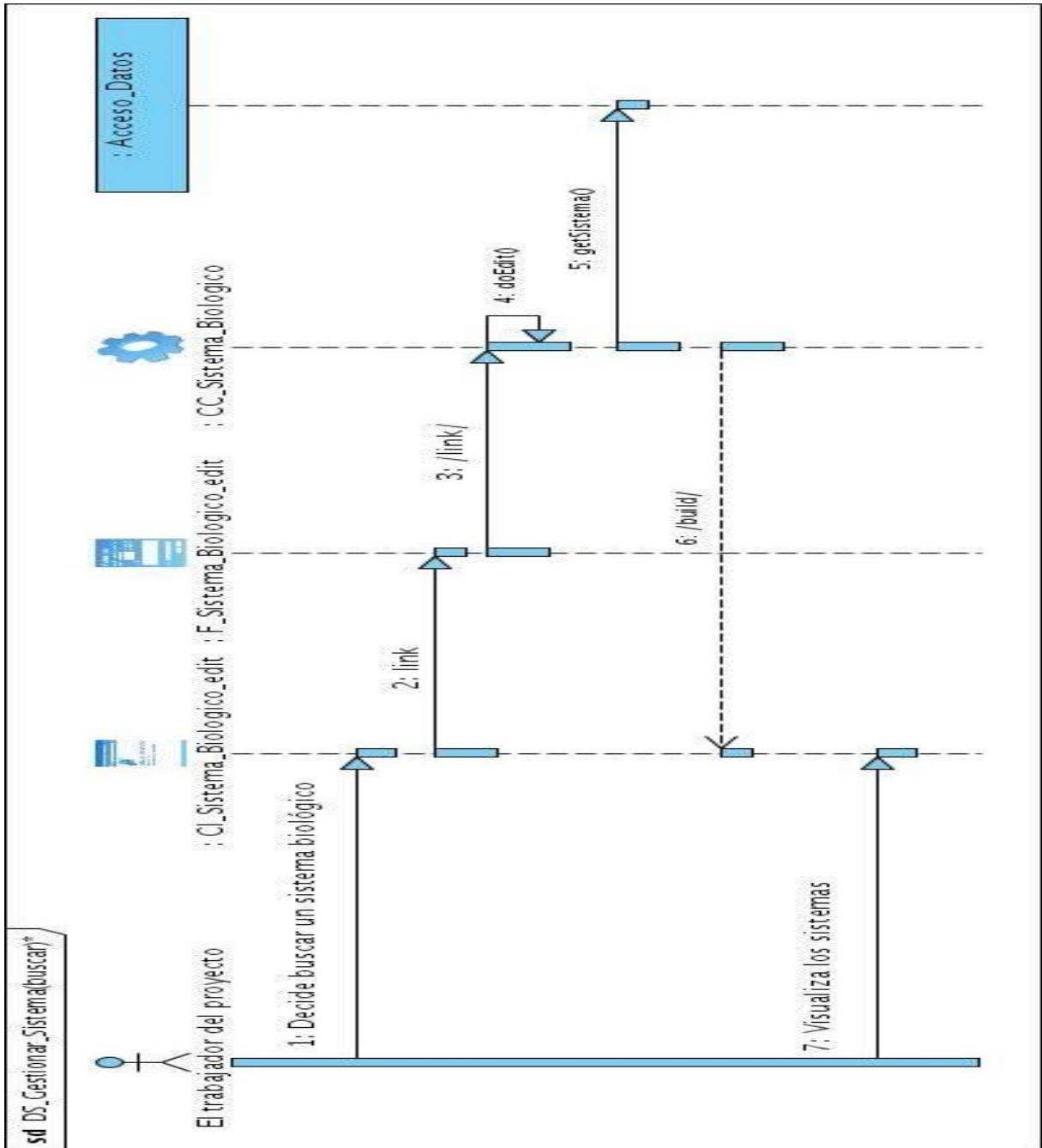


Figura 3. 15 Gestionar Sistema Biológico. Escenario (Buscar).

Gestionar Sistema Biológico. Escenario (Modificar).

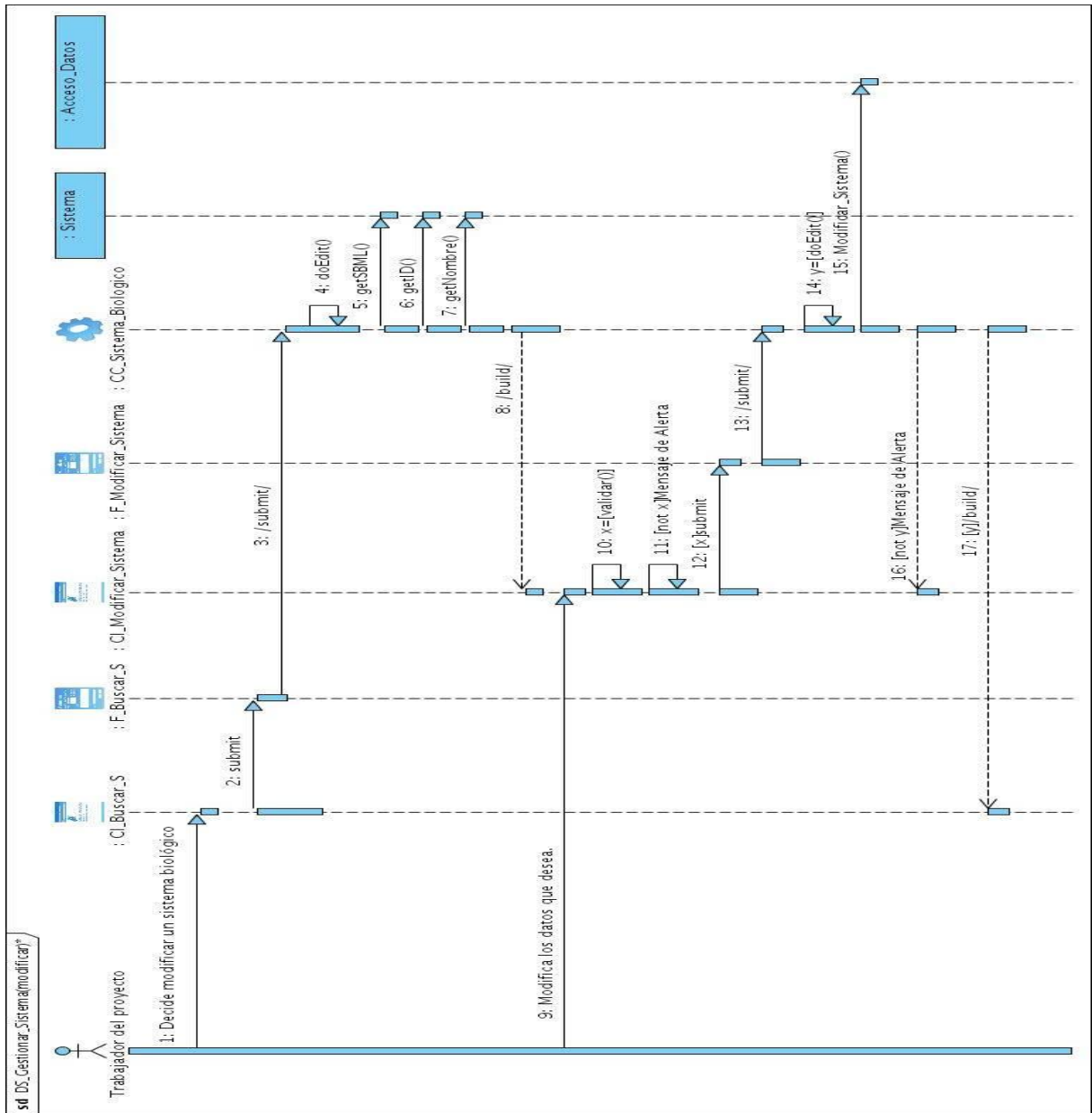


Figura 3. 16 Gestionar Sistema Biológico. Escenario (Modificar)

Eliminar Sistema Biológico.

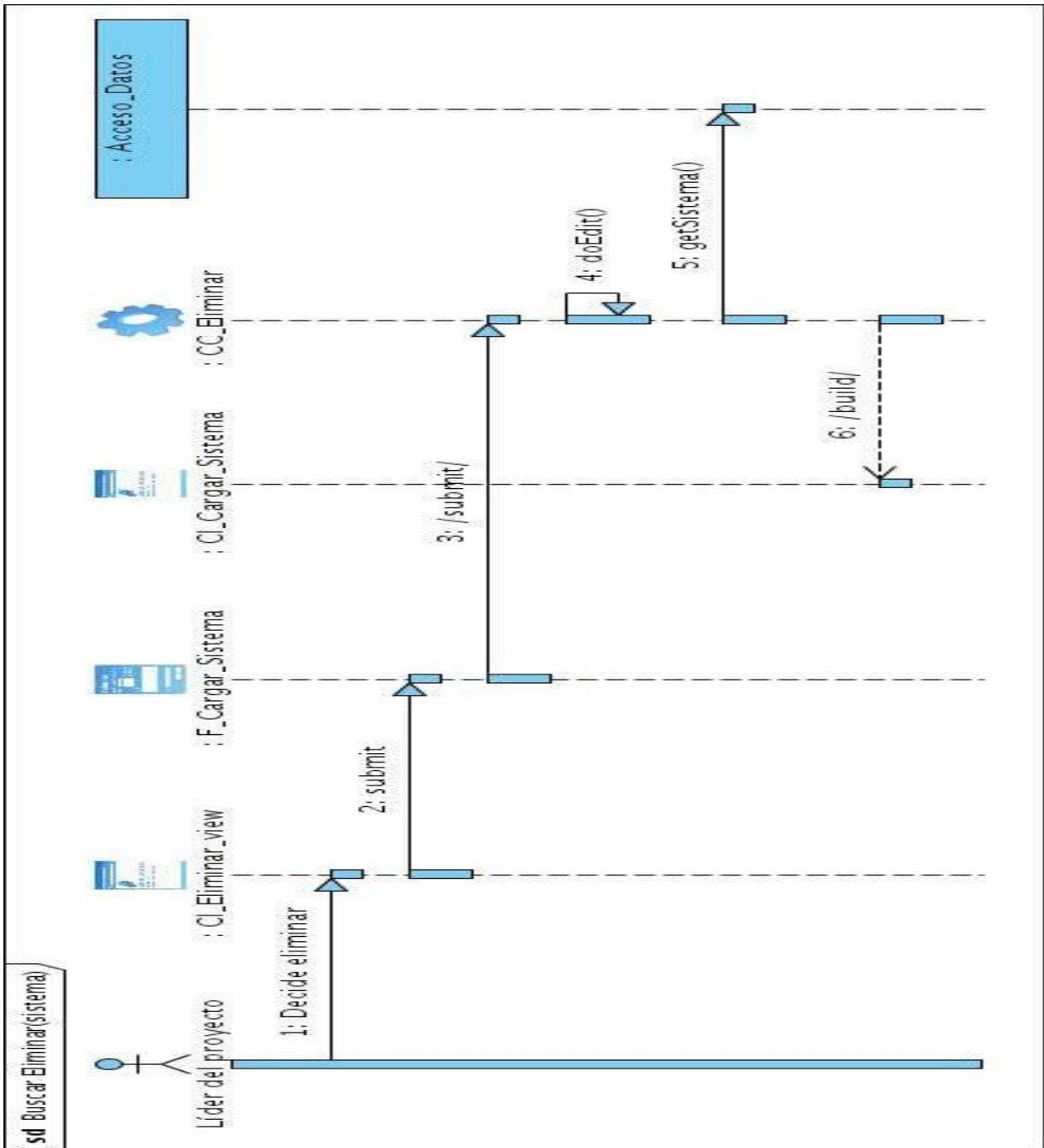


Figura 3. 17 Eliminar Sistema Biológico.

Gestionar Modelo Matemático. Escenario (Insertar)

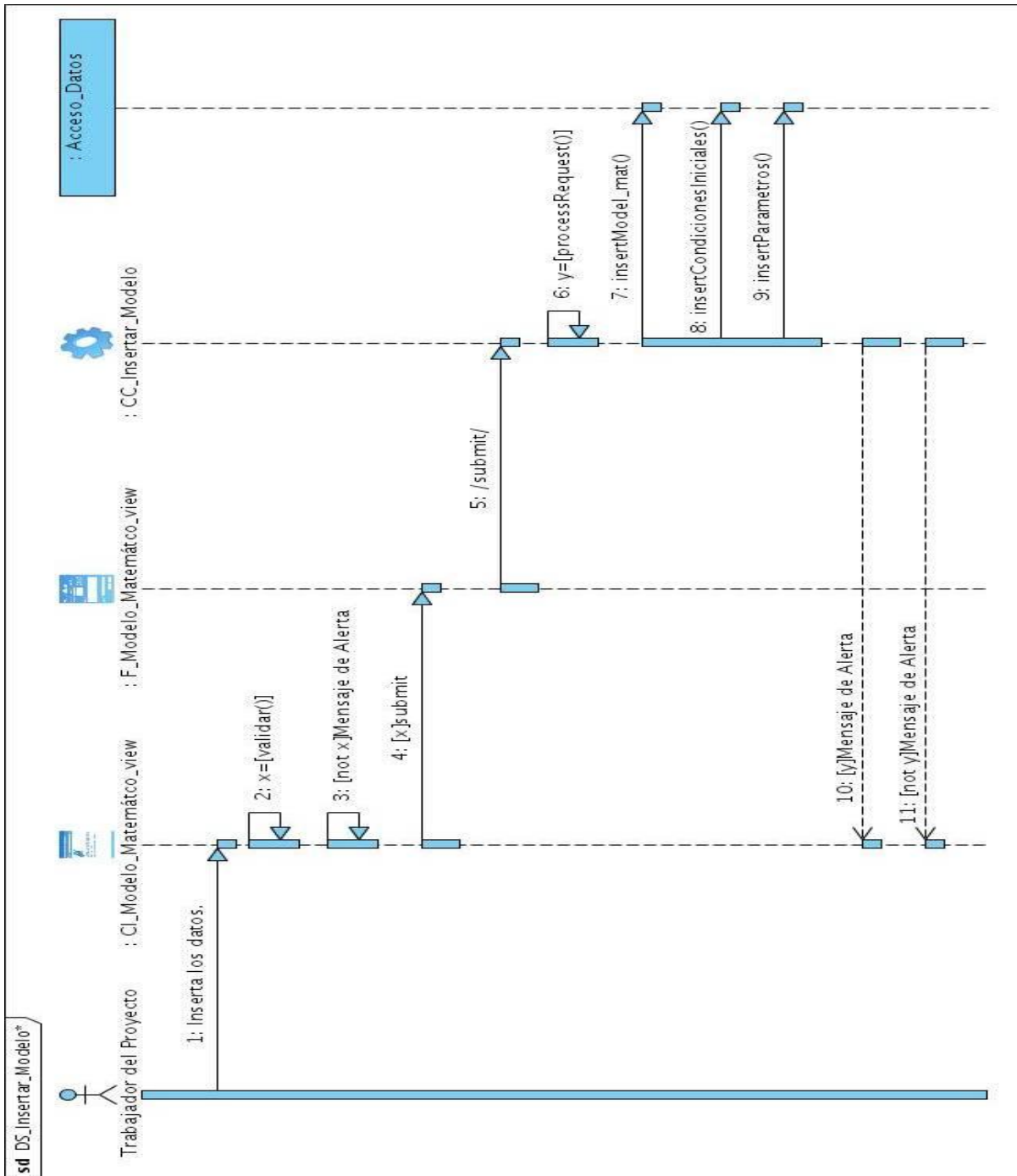


Figura 3. 18 Gestionar Modelo Matemático. Escenario (Insertar)

Gestionar Modelo Matemático. Escenario (Buscar)

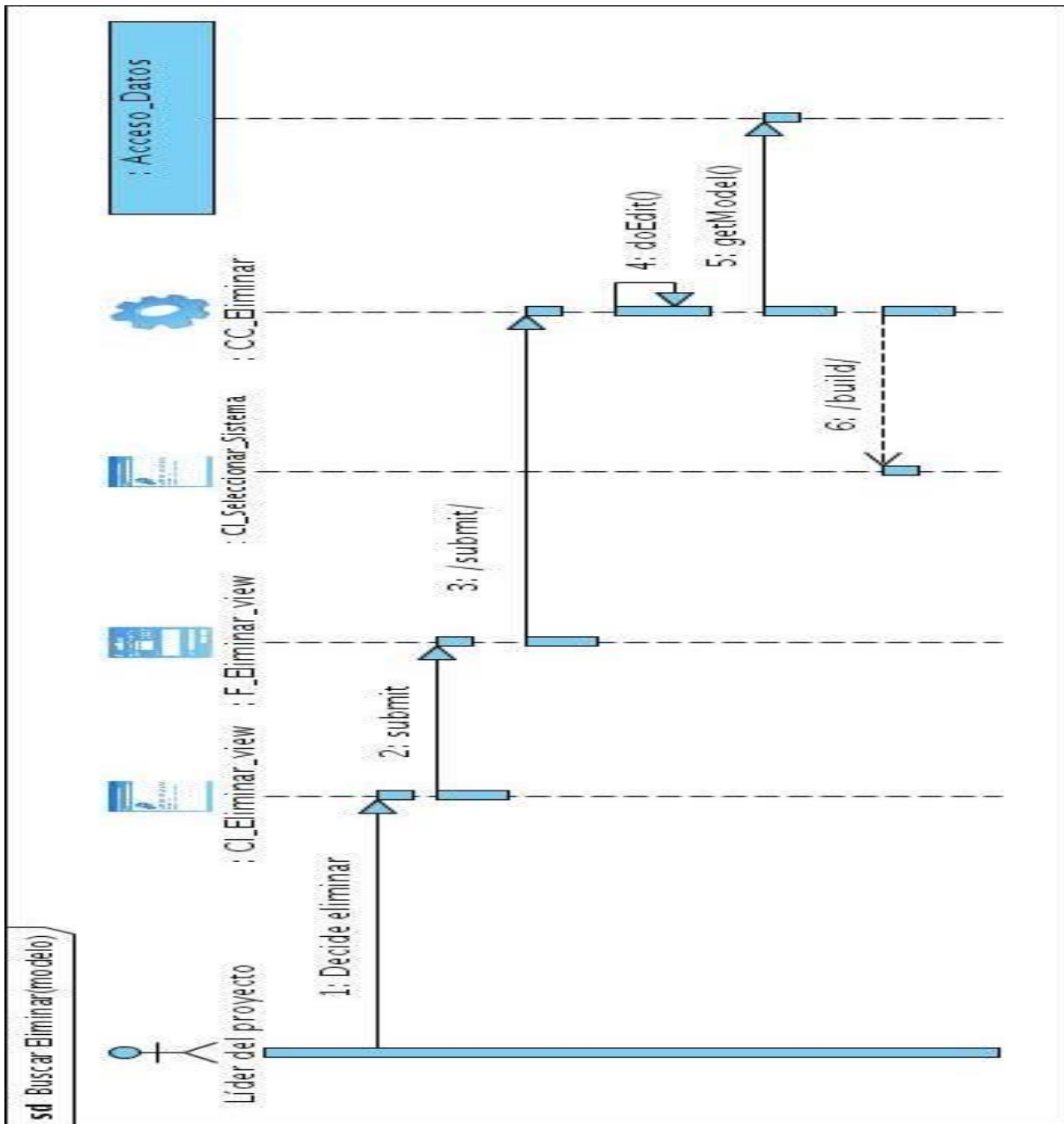


Figura 3. 19 Gestionar Modelo Matemático. Escenario (Buscar)

Gestionar Modelo Matemático. Escenario (Modificar)

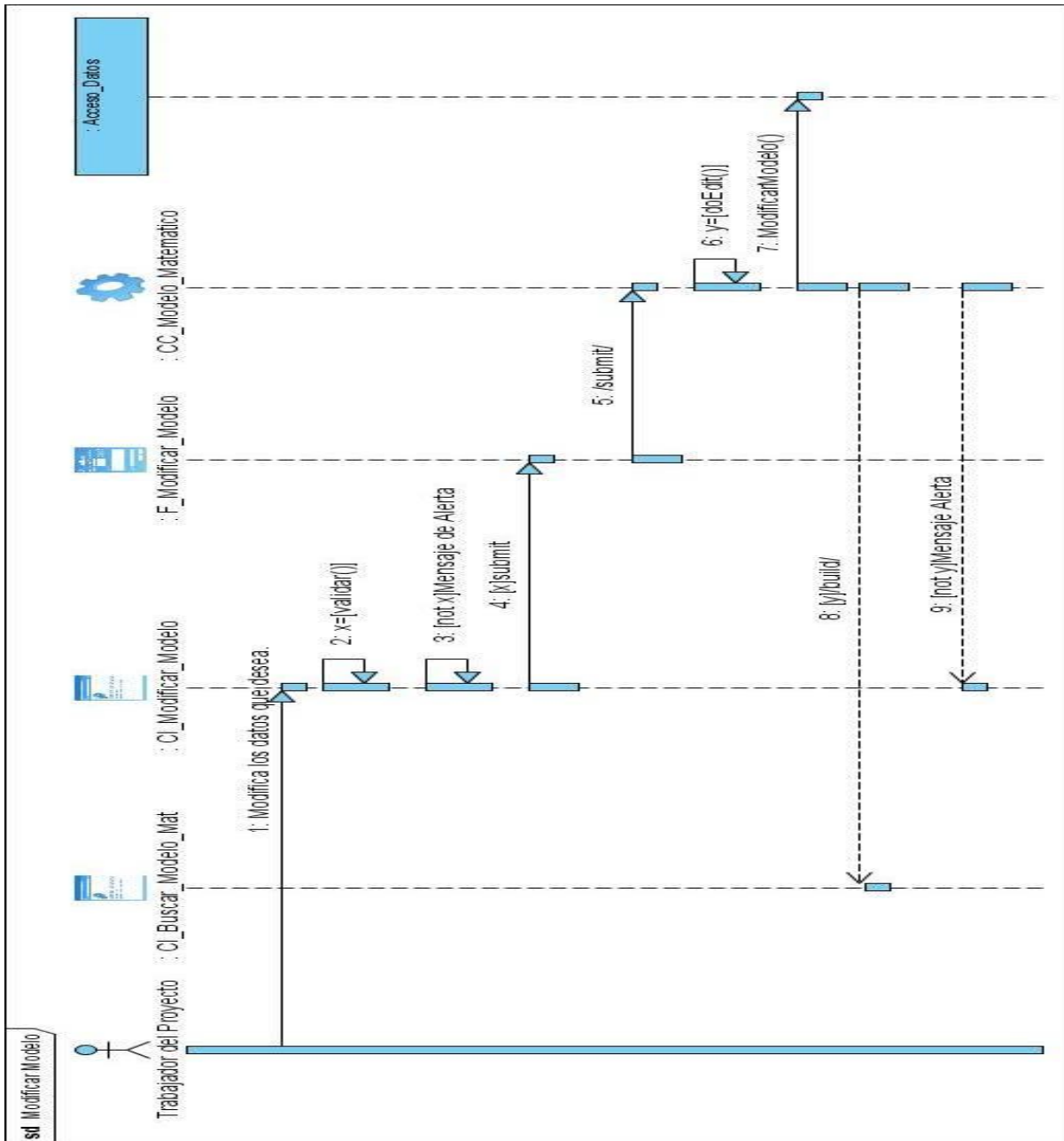


Figura 3. 20 Gestionar Modelo Matemático. Escenario (Modificar)

Gestionar Modelo Matemático. Escenario (Eliminar)

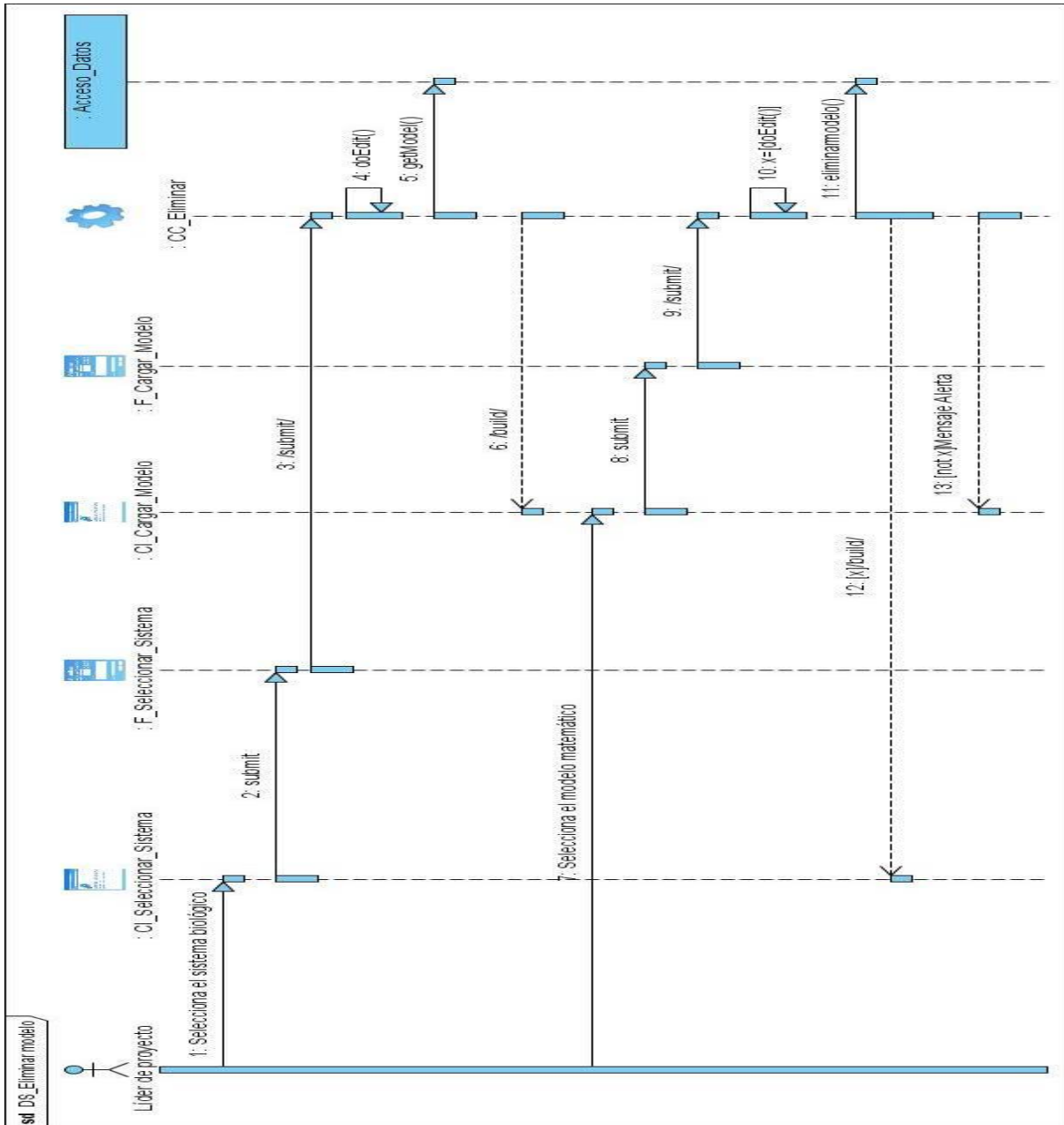


Figura 3. 21 Gestionar Modelo Matemático. Escenario (Eliminar)

3.5. Modelo de Despliegue

El modelo de despliegue describe la distribución física del sistema, esta compuesto por elementos como los procesadores, dispositivos y protocolos, lo cual permite visualizar la distribución del sistema y se utiliza como entrada fundamental en las actividades del diseño y la implementación.

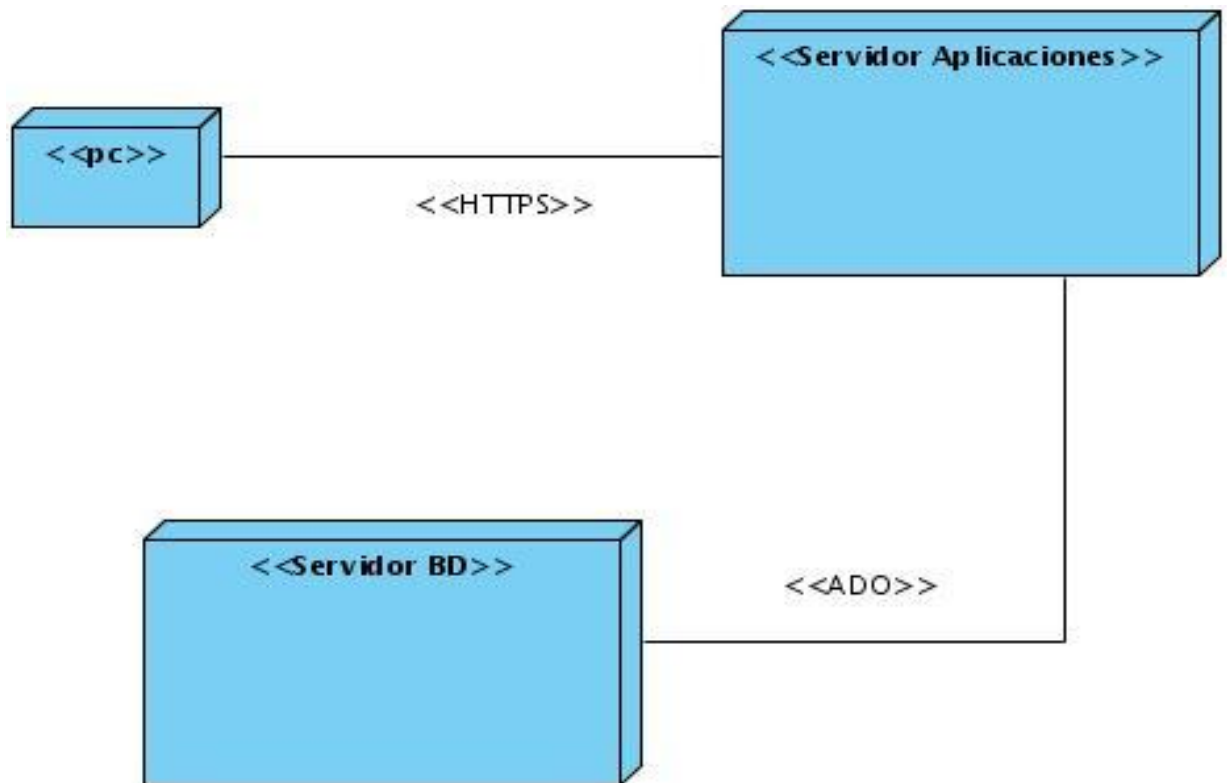


Figura 3. 22 Modelo de Despliegue.

Conclusiones del capítulo

En este capítulo se describieron los estilos arquitectónicos y patrones del diseño evidenciando su utilización. Además de los diagramas de clases, diagramas de interacción y modelo de despliegue.

CAPÍTULO 4: IMPLEMENTACIÓN

En este capítulo se realiza el diagrama de componente que pertenece al flujo de Implementación y se exponen algunas pantallas de la aplicación.

4.1 Diagrama de Componentes

El diagrama de componentes describe como los elementos del modelo de diseño se implementan en términos de componentes, ficheros de código fuente, ejecutables etc. Se utiliza para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software.

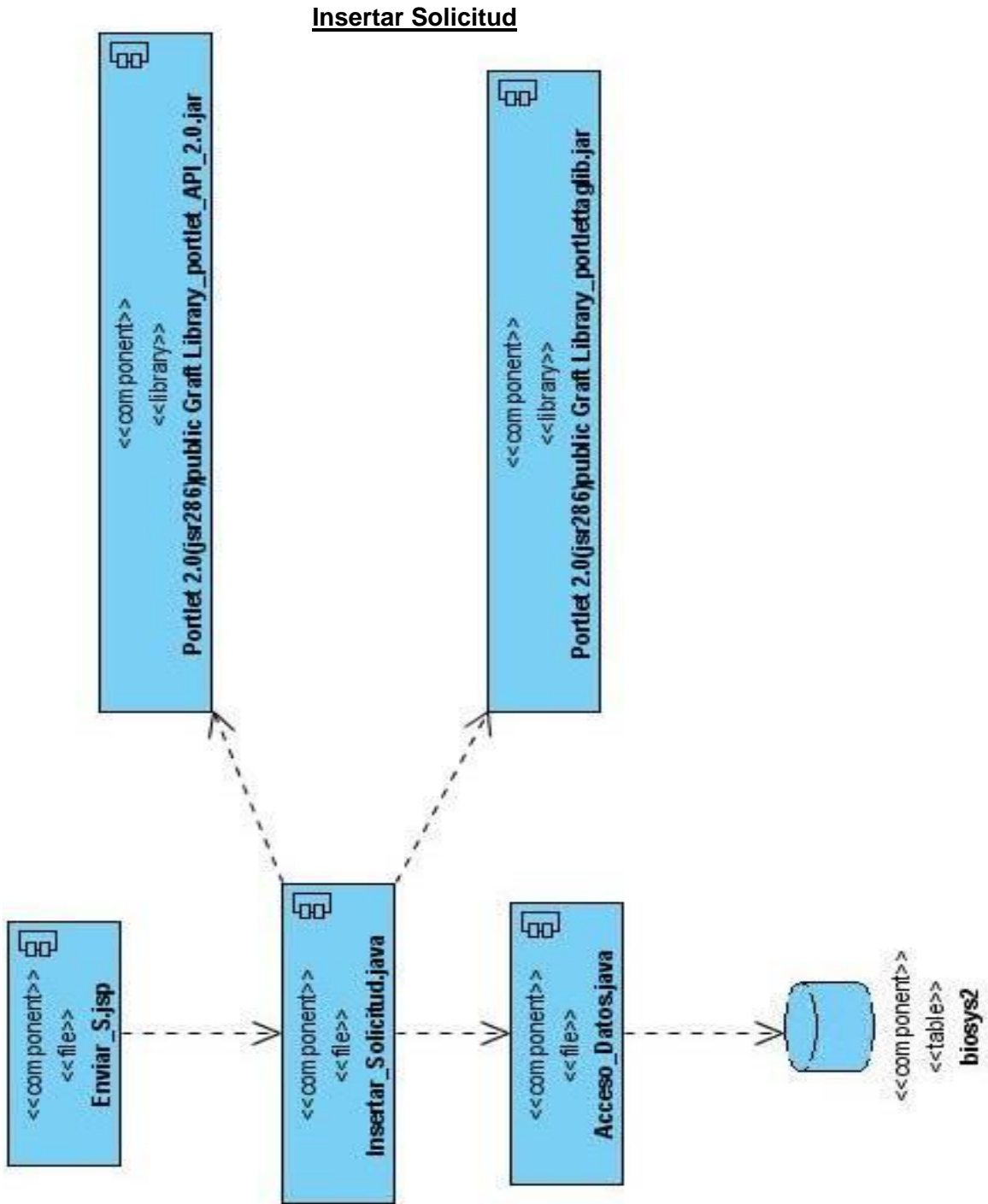


Figura 4. 1 Diagrama de Componentes (Insertar Solicitud).

Gestionar Proyecto

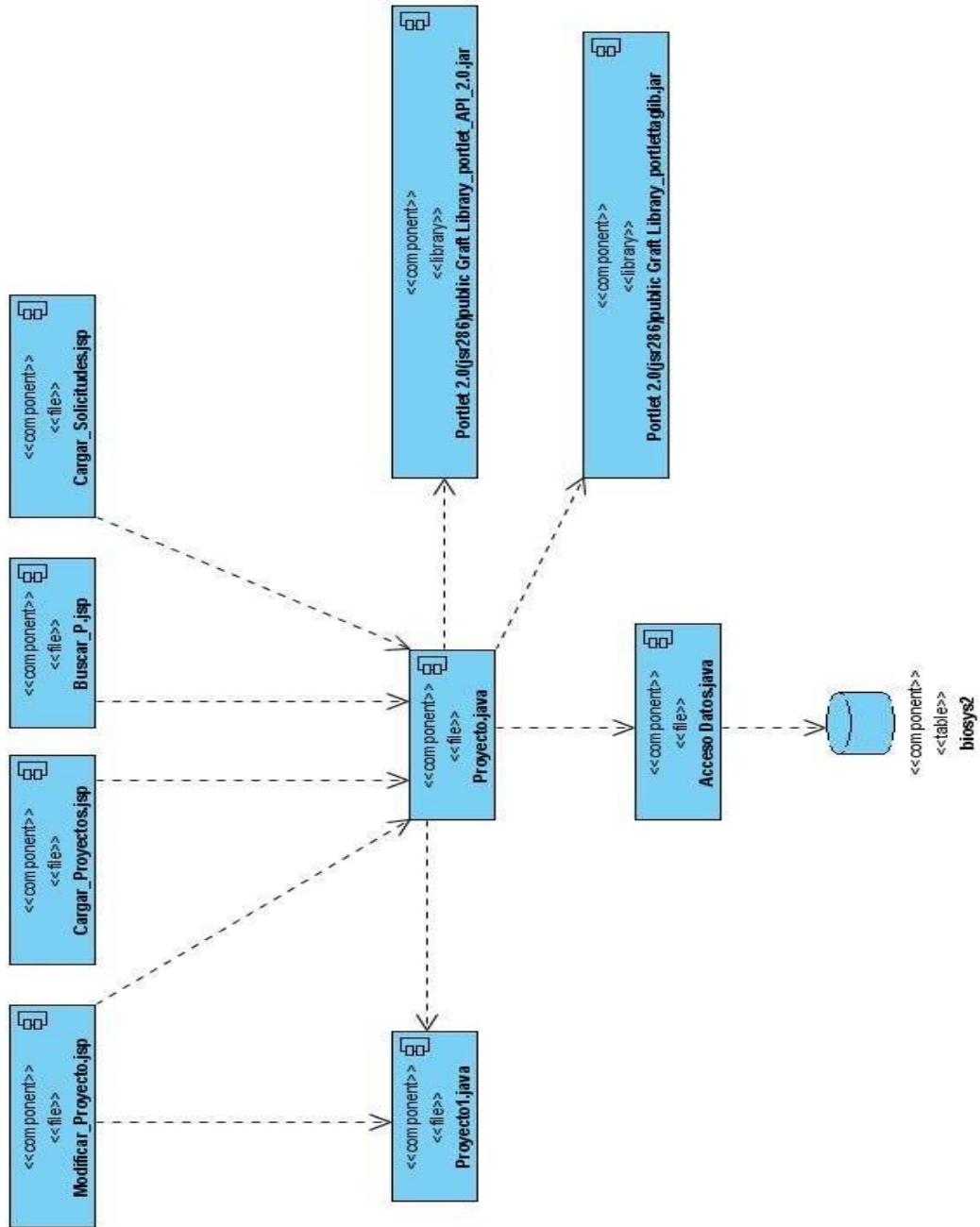


Figura 4. 2 Diagrama de Componentes (Gestionar Proyecto).

Insertar Sistema Biológico.

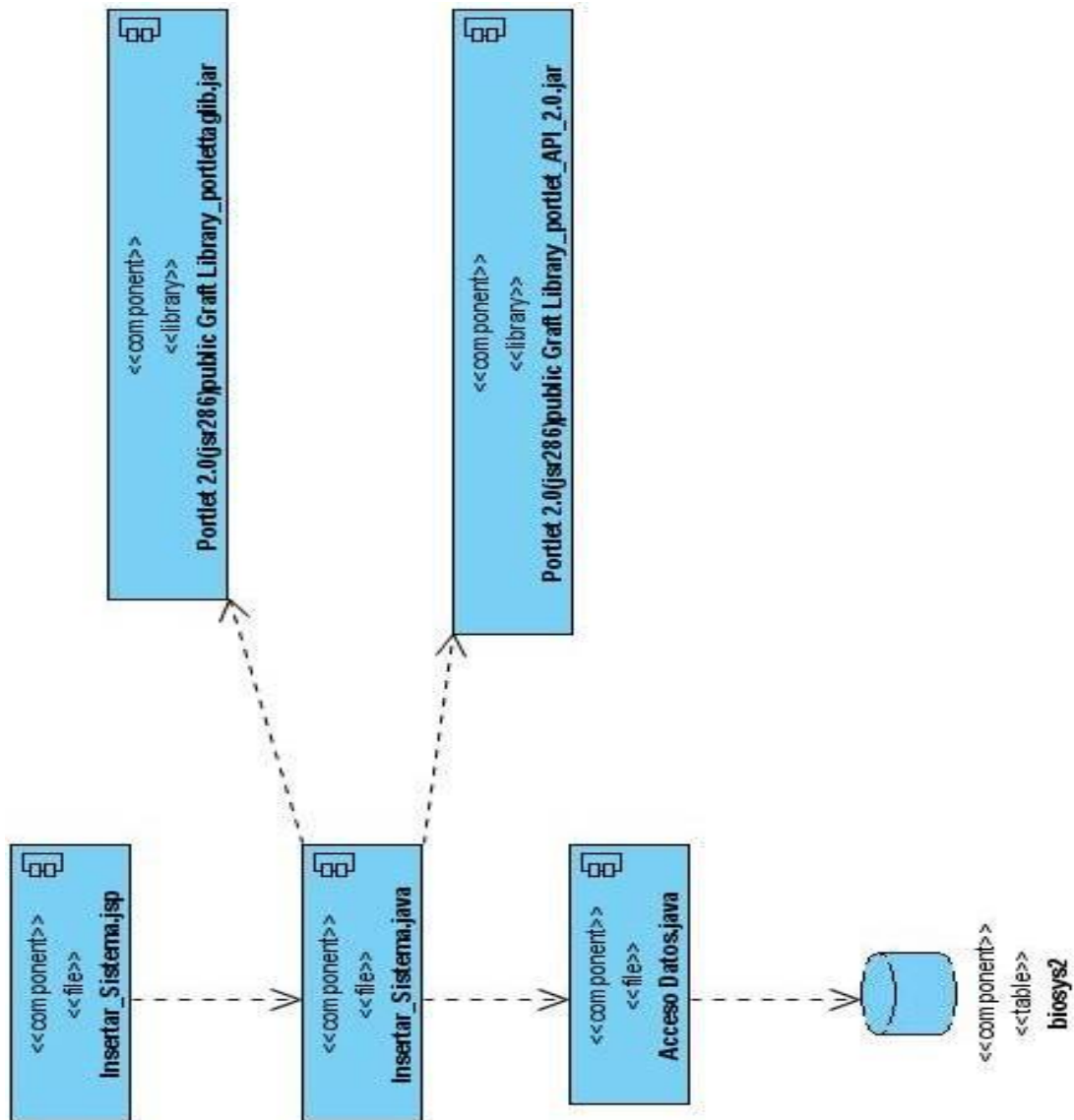


Figura 4. 3 Diagrama de Componentes (Insertar Sistema Biológico)

Modificar Sistema Biológico.

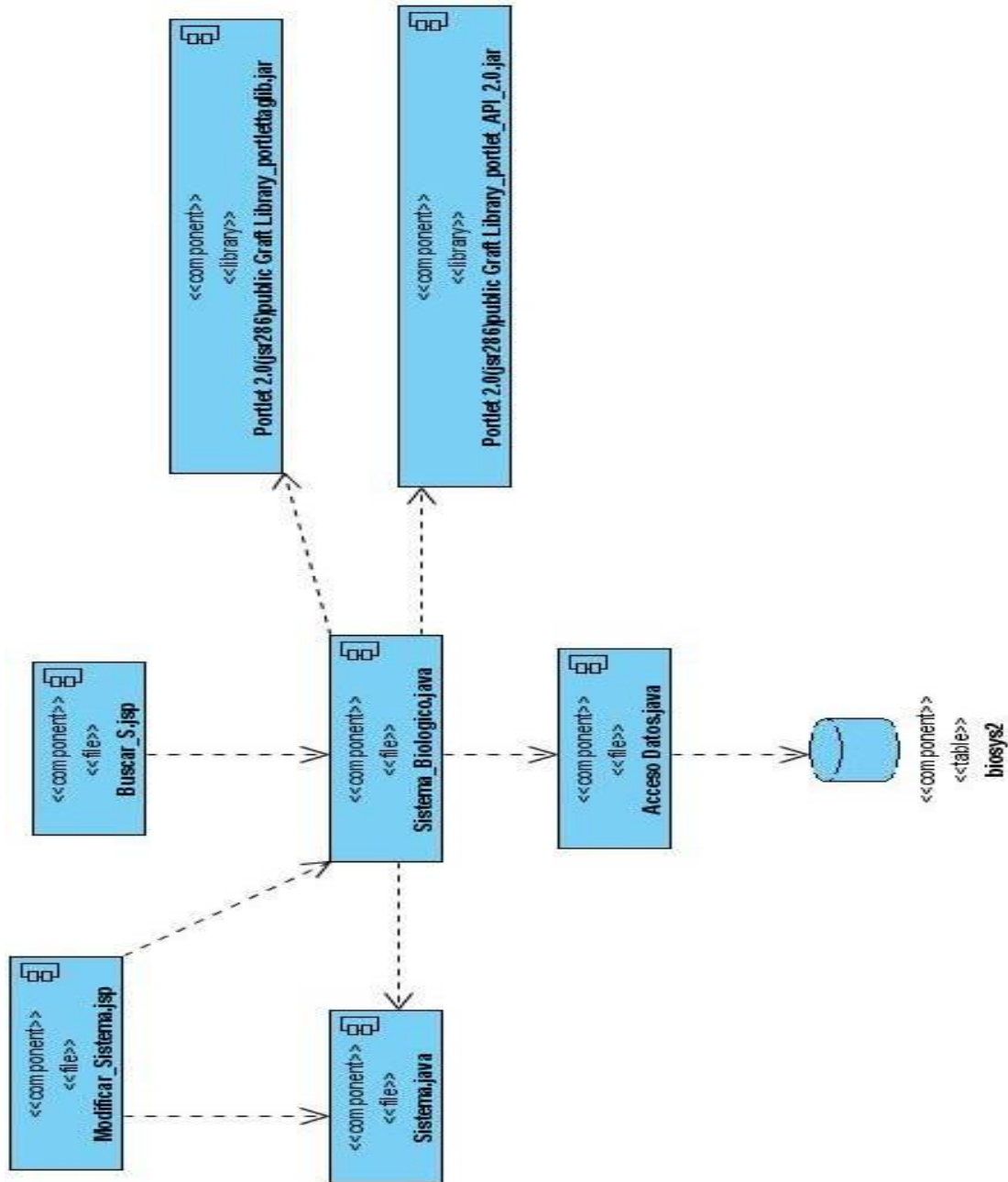


Figura 4. 4 Diagrama de Componente (Modificar Sistema)

Insertar Modelo Matemático

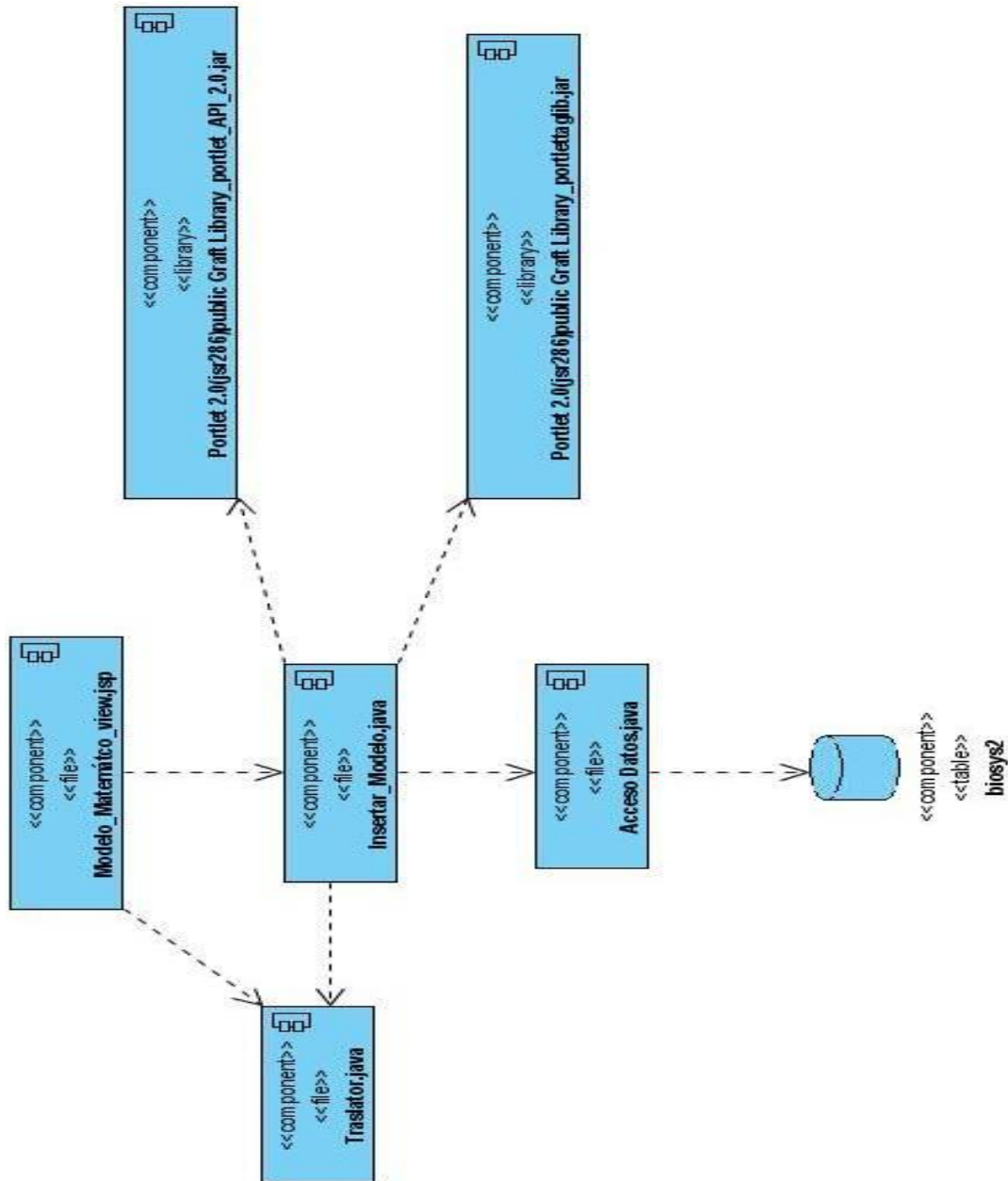


Figura 4. 5 Diagrama de Componentes (Insertar Modelo Matemático)

Insertar Modelo Matemático

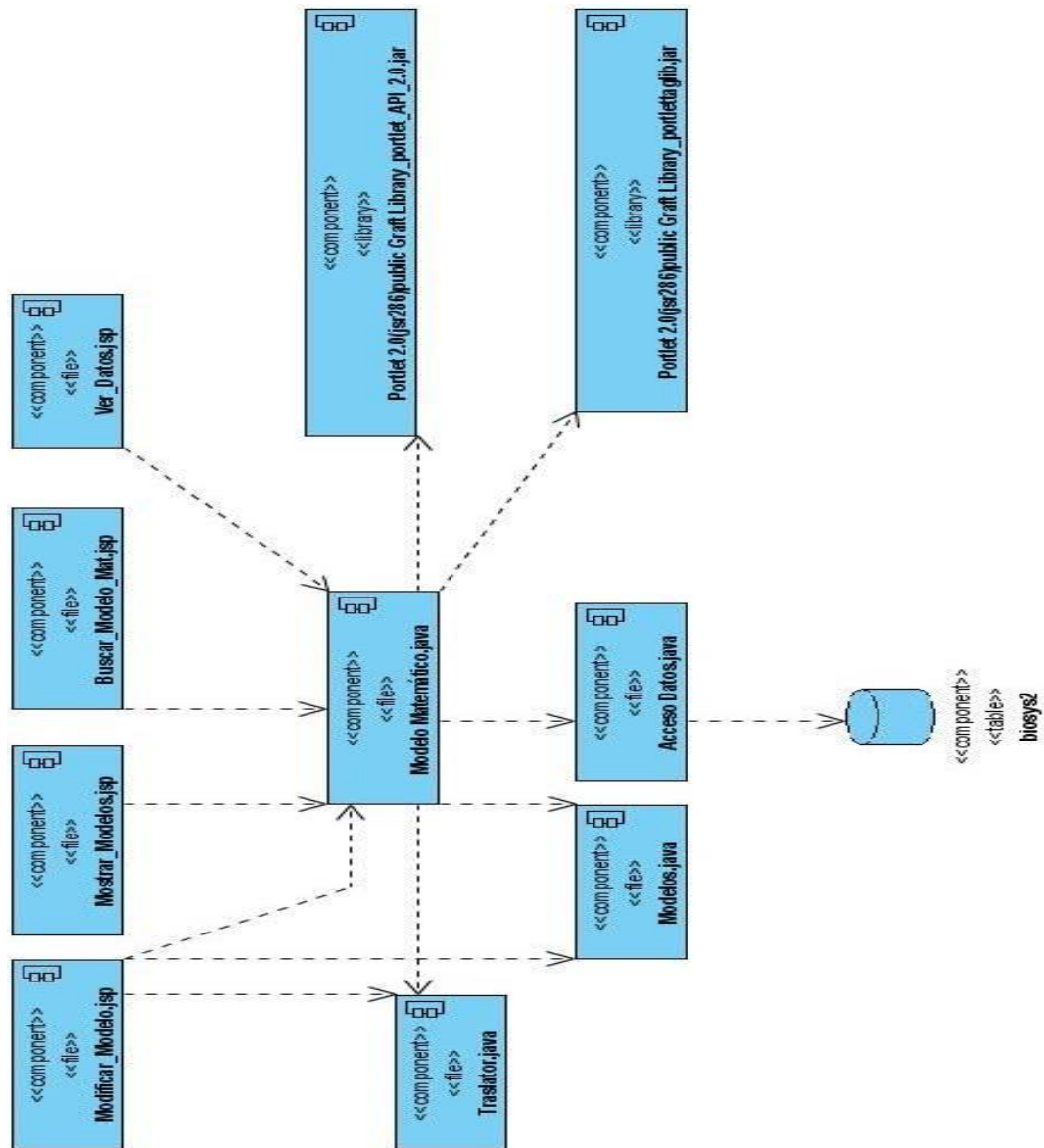


Figura 4. 6 Diagrama de Componentes (Modificar Modelo Matemático)

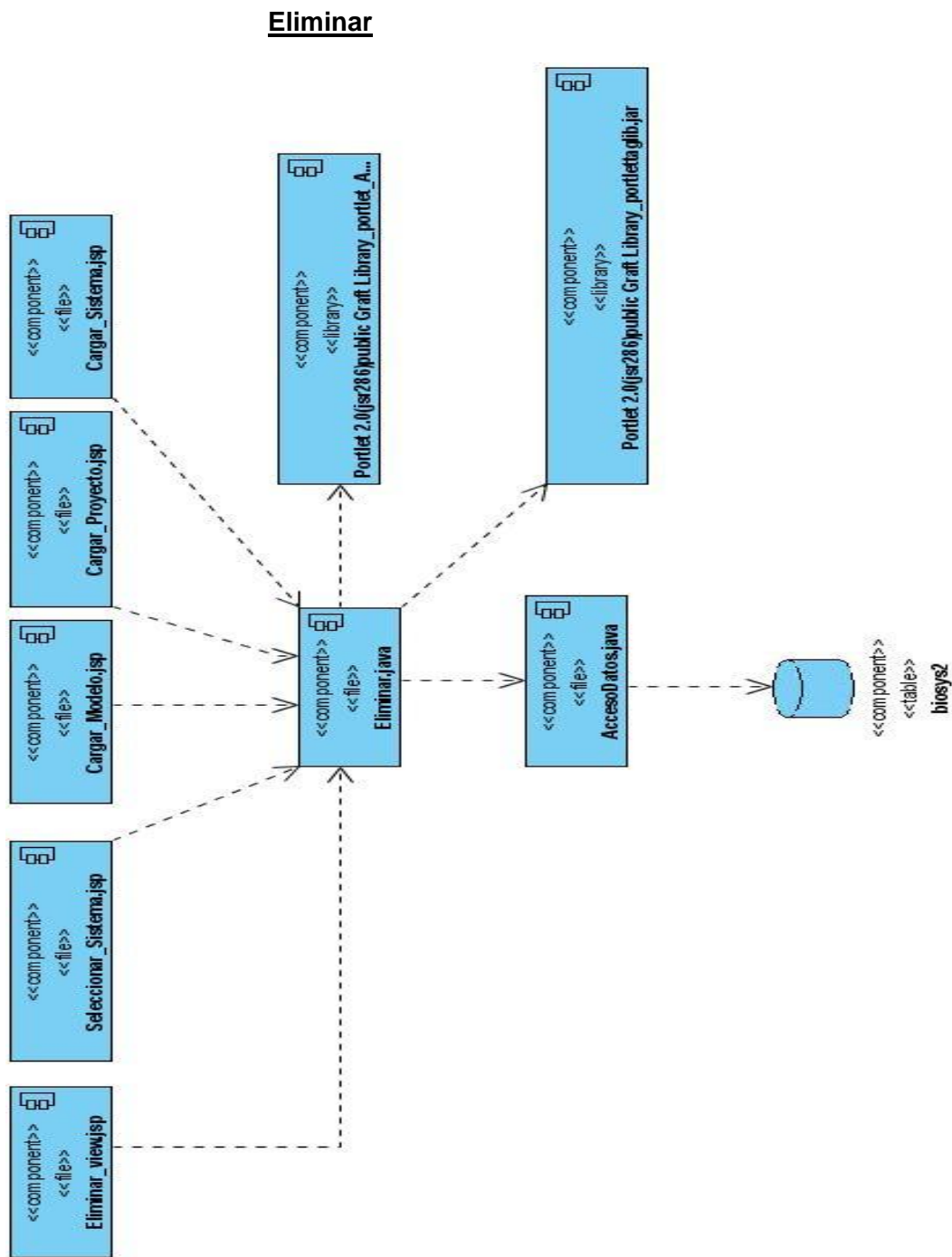


Figura 4. 7 Diagrama de Componentes (Eliminar)

4.2 Principales pantallas de la aplicación

A continuación se presentarán las principales pantallas de la aplicación:

The screenshot shows a window titled 'Solicitud' with a standard toolbar (minimize, maximize, close). Below the title bar, the text 'Enviar Solicitud' is displayed. The form contains four input fields: 'Nombre del Proyecto', 'Líder del Proyecto', 'Trabajadores', and 'Descripción'. A 'Solicitar' button is located at the bottom center of the form.

Figura 4. 8 Insertar Solicitud.

The screenshot shows a window titled 'Gestionar_Solicitud' with a standard toolbar. Below the title bar, there is a table with four columns: 'Nombre Solicitud', 'Nombre del Lider', 'Trabajadores', and 'Descripcion'. Each row in the table has a small circular icon to its right. At the bottom of the window, there are two buttons: 'Rechazar' and 'Aprobar'.

Nombre Solicitud	Nombre del Lider	Trabajadores	Descripcion
simulaciones	lisbeth	alain	biosys
Simulaciones estáticas	jorge	leonel edel	biosys
CIGB	Alain	MariaV	Simular
DESOFT	Alain	Yusleny Alain RaÃ¶l	BioSyS

Figura 4. 9 Gestionar Solicitud.

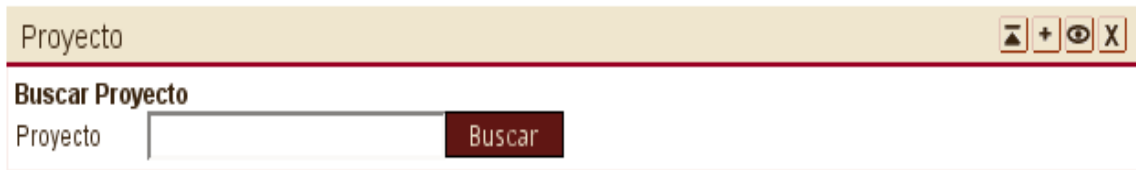


Figura 4. 10 Buscar Proyecto.



Figura 4. 11 Eliminar Proyecto(s).



Figura 4. 12 Mostar Proyecto(s).

The screenshot shows a window titled 'Proyecto' with a standard toolbar (minimize, maximize, close). The form contains the following fields and values:

Nombre del Proyecto	lisbeth
Nombre del lider del proyecto	alain
Trabajadores	lisbeth
Descrpcion	polito

Below the fields is a red 'Modificar' button and a link: 'Pulse [aquí](#) para volver atrás'.

Figura 4. 13 Modificar Proyecto(s).

The screenshot shows a window titled 'Sistema_Biologico' with a toolbar including a pencil icon. The form is titled 'Insertar Sistema Biológico' and contains:

- 'Proyecto' field with a dropdown menu showing 'Seleccionar'.
- 'Nombre del Sistema' text input field.
- 'SBML' text input field with a 'Browse' button next to it.
- A red 'Insertar' button at the bottom.

At the bottom, there is a link: 'Pulse [aquí](#) para volver atrás'.

Figura 4. 14 Insertar Sistema(s) Biológico(s).

The screenshot shows a window titled 'Sistema_Biologico' with a toolbar including a pencil icon. The form is titled 'Modificar Sistema Biológico' and contains:

- 'Sistema biológico' text input field with the value 'Interleuquina'.
- 'SBML' text input field with the value '/home/ameneses/NetBe...' and a 'Browse' button next to it.
- A red 'Modificar' button at the bottom.

At the bottom, there is a link: 'Pulse [aquí](#) para volver atrás'.

Figura 4. 15 Modificar Sistema(s) Biológico(s).

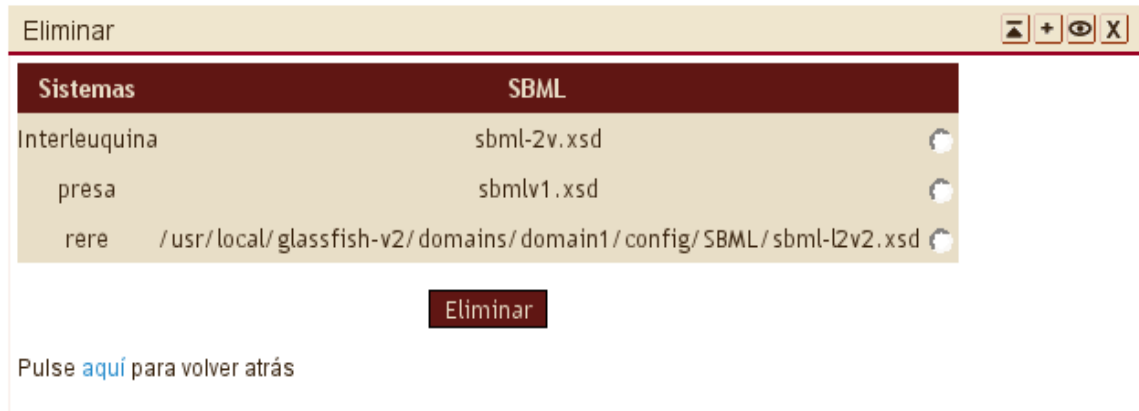


Figura 4. 16 Eliminar Sistema(s) Biológico(s).

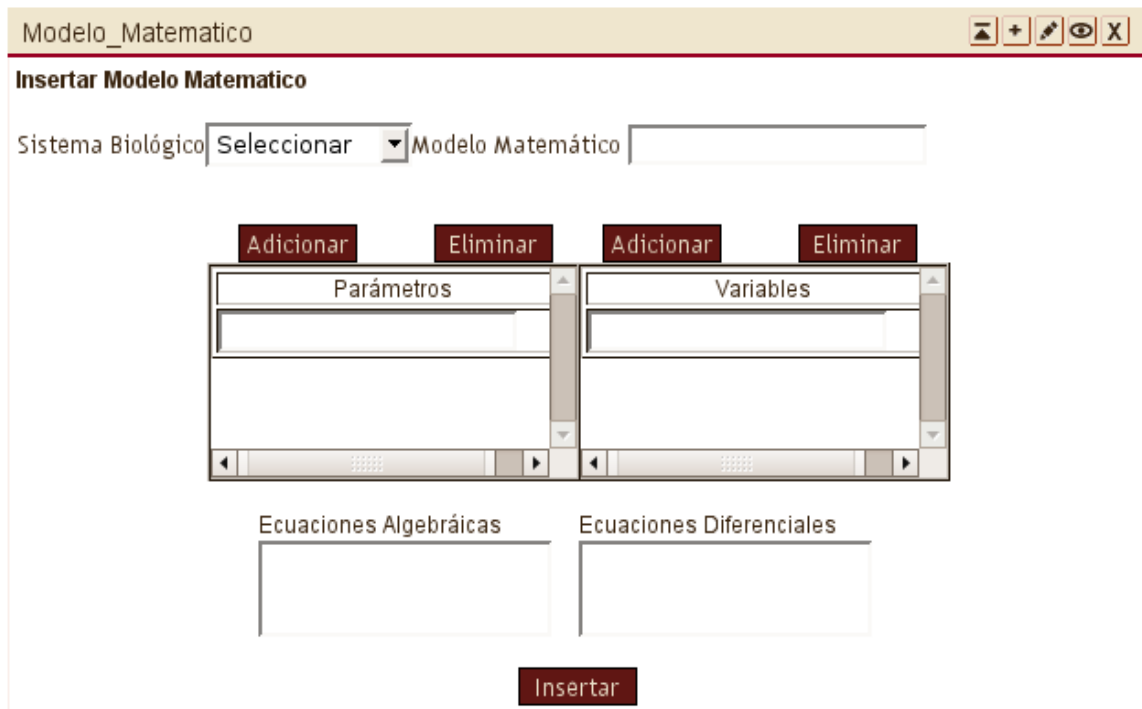


Figura 4. 17 Insertar Modelo(s) Matemático(s).

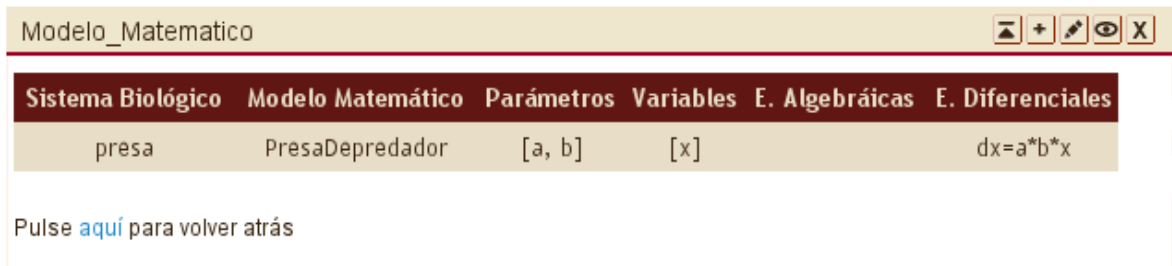


Figura 4. 18 Mostrar Modelo(s) Matemático(s).

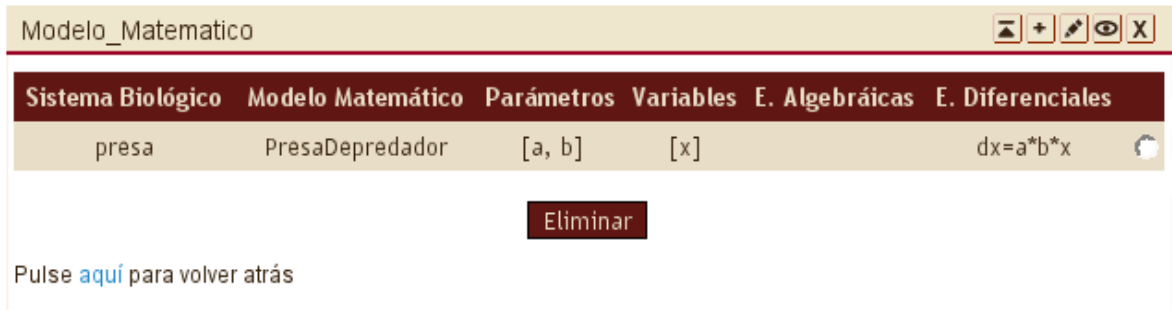


Figura 4. 19 Eliminar Modelo(s) Matemático(s).

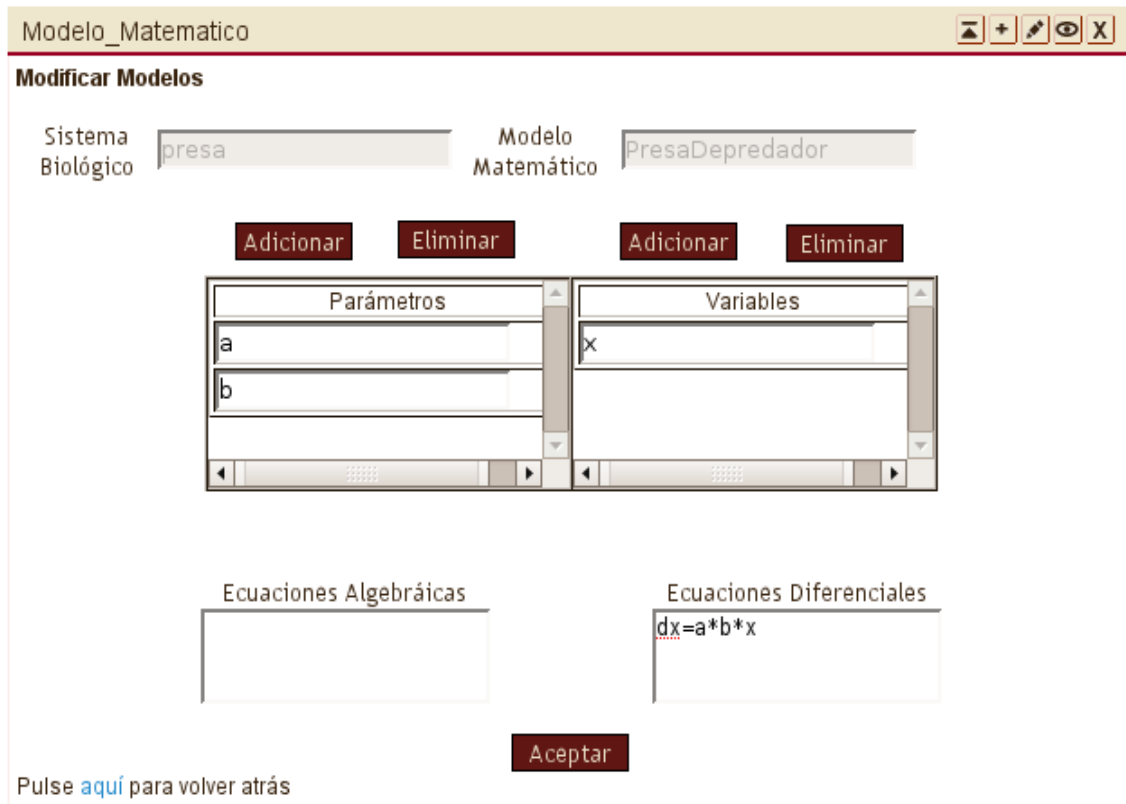


Figura 4. 20 Modificar Modelo Matemático.



Figura 4. 21 Buscar para eliminar.

Conclusiones del capítulo

En este capítulo se describió cómo fue implementado el sistema en términos de componentes y se expusieron algunas pantallas del sistema implementado.

CONCLUSIONES

Después de haber realizado el presente trabajo de diploma se llegó a las siguientes conclusiones:

- Se implementó un sistema capaz de gestionar la información necesaria para realizar simulaciones de Sistemas Biológicos, lo cual permite a los investigadores crear su propio proyecto, en el cual se guardará solo la información que ellos deseen.
- Se obtuvo la primera versión del sistema, lo cual contribuirá a mejorar las investigaciones en la Biología de Sistemas.

RECOMENDACIONES

Se recomienda:

- Continuar con el estudio de los Portlets para el desarrollo de la segunda versión del sistema.
- Agregar la funcionalidad **Simulación** a los Portlets.

BIBLIOGRAFÍA

1. Sitio de Ingeniería de software. [Online] <http://teleformación.uci.cu>.
2. Rojas, Freddy A. abril 2006. Metodología para la construcción de Portlets para aplicaciones científicas.
3. Pressman, Roger S. *Ingeniería de software, un enfoque práctico*. Cuarta Edición.
4. Övergaard, Gunnar y Palmkvist, Karin. 2004. *Use Cases Patterns and Blueprints* .s. l.: Addison Wesley Professional, 2004.
5. Larman, Craig. 2004. *UML y patrones, introducción al análisis y diseño orientado a objetos*. s.l. : Felix Valera, 2004.
6. Introducción a las Bases de Dato. Modelo Entidad Relación (ER). Conferencia 1, Universidad de las Ciencias Informáticas, 2007.p.
7. [En línea] [Citado el: 25 de 1 de 2008.] <http://www.uvic.cat/eps/dept/biologiasistemas/es/inici.html>..
8. [En línea] [Citado el: 1 de 2 de 2008.] <http://cremc.ponce.inter.edu/1raedicion/modelacion.htm>..
9. [En línea] [Citado el: 4 de 5 de 2008.] <http://www.cidse.itcr.ac.cr/cursos-nea/EcuacionesDiferenciales/EDO-Geo/edo-cap1-geo/node3.html>..
10. [En línea] [Citado el: 20 de 5 de 2008.] http://www.alzado.org/articulo.php?id_art=249.
11. [En línea] [Citado el: 23 de 4 de 2008.] <http://www.entmexico.com/hosting/windows-o-linux.html>..
12. [En línea] [Citado el: 29 de 4 de 2008.] <http://www.Webestilo.com/php/php00.phtml>.
13. [En línea] [Citado el: 1 de 5 de 2008.] http://www.programacion.com/tutorial/Servlets_jsp..
14. [En línea] [Citado el: 15 de 5 de 2008.] http://www.netbeans.org/index_es.html..
15. [En línea] [Citado el: 30 de 5 de 2008.] <http://www.cherokee-project.com/>..
16. [En línea] [Citado el: 2 de 6 de 2008.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>

17. *Introducción a las Bases de Dato. Modelo EntidadRelación(ER). Conferencia 1.* 2007. Universidad de las Ciencias Informáticas : s.n., 2007.

REFERENCIAS BIBLIOGRÁFICAS

1. [En línea] [Citado el: 25 de 1 de 2008.] <http://www.uvic.cat/eps/dept/biologiasistemas/es/inici.html>..
2. [En línea] [Citado el: 1 de 2 de 2008.] <http://cremc.ponce.inter.edu/1raedicion/modelacion.htm>..
3. [En línea] [Citado el: 5 de 4 de 2008.] <http://cremc.ponce.inter.edu/1raedicion/modelacion.htm>.
4. [En línea] [Citado el: 4 de 5 de 2008.] <http://www.cidse.itcr.ac.cr/cursos-nea/EcuacionesDiferenciales/EDO-Geo/edo-cap1-geo/node3.html>..
5. [En línea] [Citado el: 20 de 5 de 2008.] http://www.alzado.org/articulo.php?id_art=249.
6. [En línea] [Citado el: 23 de 4 de 2008.] <http://www.entmexico.com/hosting/windows-o-linux.html>..
7. [En línea] [Citado el: 29 de 4 de 2008.] <http://www.Webestilo.com/php/php00.phtml>.
8. [En línea] [Citado el: 1 de 5 de 2008.] http://www.programacion.com/tutorial/Servlets_jsp..
9. [En línea] [Citado el: 15 de 5 de 2008.] http://www.netbeans.org/index_es.html..
10. [En línea] [Citado el: 30 de 5 de 2008.] <http://www.cherokee-project.com/>..
11. [En línea] [Citado el: 2 de 6 de 2008.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>
12. *Introducción a las Bases de Dato. Modelo EntidadRelación(ER). Conferencia 1.* 2007. Universidad de las Ciencias Informáticas : s.n., 2007.

GLOSARIO DE TÉRMINOS

Artefacto: Es un fragmento de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los artefactos son los resultados tangibles del proyecto que se van creando y usando hasta obtener el producto final.

Bioinformática: Disciplina científica y tecnológica en la que interactúan en armonía los planteamientos investigadores de la Biología Genética y Molecular con los enfoques metodológicos y tecnológicos de la Ciencia de la Computación y la Ingeniería Informática. (Coltell et al., 2002).

CU: Especificación de las secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

Flujo de trabajo: Es una relación de actividades que producen resultados observables dado por una secuencia de actividades realizadas por los diferentes roles.

Frameworks: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Grids: La tecnología Grid surge del nuevo paradigma de computación distribuida propuesto por Ian Foster y Carl Kesselman a mediados de los 90. Se basa fundamentalmente en el acceso remoto a recursos computacionales. Su objetivo es permitir gestionar y distribuir la potencia de cálculo disponible, de tal forma que los usuarios se beneficien de la potencia de ordenadores infrautilizados que se encuentran dispersos geográficamente. Éstos ceden parte de sus recursos para procesar aplicaciones con funciones de cálculo avanzadas.

Lenguaje de Marcas Inalámbricas: Es una tecnología creada para ofrecer servicios y contenidos de Internet a través de conexiones inalámbricas.

Open Source: Es el término con el que se conoce al software distribuido y desarrollado libremente.

SBML: (Systems Biology Markup Language) es un lenguaje de marco en beneficio de la Biología de Sistemas esta basado en XML y creado para describir simulaciones en Biología de Sistemas.

Service Packs: Paquetes de servicios. Proporciona las actualizaciones más recientes para Microsoft.

Soap: Es un protocolo estándar creado por Microsoft, IBM y otros, define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

UDDI: Son las siglas del catálogo de negocios de Internet, donde el registro en el catálogo se hace en XML.

Unix: Es un sistema operativo portable, multitarea y multiusuario.

Web Services: Es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Webapps: El nombre Webapps es una abreviación de Portal en Perl para Automatización de Web. Disponible bajo la licencia GNU Licencia Pública General, Webapps es software libre.

XHTML: Es una versión más estricta y limpia de HTML. Lenguaje de Marcas Inalámbricas: es una tecnología creada para ofrecer servicios y contenidos de Internet a través de conexiones inalámbricas.