

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SIMULACIÓN PARA LA ENSEÑANZA DE GRAVITACIÓN UNIVERSAL DIRIGIDO A ESTUDIANTES DE SECUNDARIA Y PRE-GRADO UNIVERSITARIO

Tesis para optar por el Título de Ingeniero Informático, que presenta el bachiller:

MIGUEL JAVIER VALDEZ MEJIA

ASESOR: LUIS ALBERTO FLORES

LIMA, noviembre del 2011

RESUMEN

El tema de Gravitación Universal define el comportamiento que tienen los cuerpos físicos frente a otros, describiendo las causas de los cambios en sus movimientos. Para entender este tema se requiere realizar experimentos en los laboratorios.

Para realizar experimentos de Gravitación Universal en el dominio de física en los laboratorios, se presentan los siguientes problemas: Alto costo de infraestructura del ambiente y los materiales; falta de demostración de los experimentos relacionados con el tema de Gravitación Universal y complicados procesos de preparación y de desarrollo de los experimentos físicos.

El presente proyecto de tesis consiste en analizar, diseñar e implementar un sistema de simulación para la enseñanza de Gravitación Universal dirigido a estudiantes de secundaria y pre-grado universitario, que permita simular ambientes o escenarios de experimentos físicos.

El sistema planteado es una herramienta educativa que permite al alumno analizar los experimentos físicos creados por el mismo, desarrollando su creatividad y fomentando el aprendizaje en los temas de física.

Los estudiantes que tienen problemas con los conceptos físicos pueden encontrar en este sistema, el aprendizaje interactivo que les permita entender fácilmente los temas.

Las simulaciones de experimentos de laboratorio no intentan sustituir los experimentos reales, su propósito es servir como preparación de estos experimentos. Sin embargo, los experimentos reales pueden ser costosos, peligrosos o difíciles de instalar en un laboratorio escolar.

El sistema muestra una animación en la que los estudiantes pueden cambiar los escenarios de acuerdo a los parámetros ingresados. Esta animación permite a los estudiantes desarrollar mejores métodos de resolución de problemas y de estímulo para resolverlos.

FACULTAD DE
**CIENCIAS E
 INGENIERÍA**
 ESPECIALIDAD DE
 INGENIERÍA INFORMÁTICA

 PONTIFICIA
**UNIVERSIDAD
 CATÓLICA**
 DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

TÍTULO: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SIMULACIÓN PARA LA ENSEÑANZA DE GRAVITACIÓN UNIVERSAL DIRIGIDO A ESTUDIANTES DE SECUNDARIA Y PRE-GRADO UNIVERSITARIO

ÁREA: MULTIMEDIA

PROPONENTE: Ing. Hugo Rodríguez

ASESOR: Ing. Luis Flores García
 Ing. Hugo Rodríguez

ALUMNO: Miguel Javier Valdez Mejía

CÓDIGO: 20019116

TEMA N°: 391

FECHA: San Miguel, 15 de marzo de 2011

DESCRIPCIÓN

La educación que se le da a una persona es muy importante debido a que en ella se forjan los valores, actitudes y sentimientos que expresa ante la sociedad. La educación incentiva al alumno a pensar, imaginar y crear.

Una de las mejores formas de aprendizaje y enseñanza es dejar que el alumno libere su imaginación, que el mismo descubra, analice y entienda el objetivo del problema planteado. Para ello, el alumno requiere de un entorno de aprendizaje que permita interactuar con el problema, así como entender problemas complejos que el alumno con un método tradicional no intentaría desafiar o motivar su resolución, tales como algunos temas del curso de física.

Para un mejor entendimiento de los temas del curso de física se requiere realizar experimentos en los laboratorios, pero algunos de estos experimentos resultan costosos ya que se debe tomar en cuenta la infraestructura del ambiente y los materiales, y en algunos casos son imposibles de efectuar, por tal motivo el aprendizaje de la física resulta dificultoso para el alumnado debido a que no le encuentra un sentido práctico y demostrativo. Todo el aprendizaje se basa solo en la teoría, de este modo se desanima y se desorienta al alumnado hacia una posible investigación acerca de este tema.

 Av. Universitaria 1801
 San Miguel, Lima - Perú

 Apartado Postal 1761
 Lima 100 - Perú

 Teléfono:
 (511) 626 2000 Anexo 4801

FACULTAD DE
**CIENCIAS E
INGENIERÍA**
ESPECIALIDAD DE
INGENIERÍA INFORMÁTICA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

Uno de los temas del curso de física es la ley de gravitación universal, el cual permite definir el comportamiento que tienen los cuerpos físicos frente a otros, en este caso sobre las causas de los cambios en sus movimientos, en particular este tema no es comprendido completamente por el alumno ya que no es posible representar estos fenómenos en una situación real en los laboratorios.

Actualmente la tecnología de la información permite tomar nuevos retos en el desarrollo de la educación, ya que proporciona una gama inmensa de herramientas y recursos para la investigación de diversos temas hacia el alumno, permitiendo nuevos métodos de aprendizaje y enseñanza tales como los sistemas de simulación educativos, que están tomando importancia en el proceso de educación de hoy y en el futuro.

Bajo este esquema se plantea la implementación de un sistema de simulación educativo en el tema de gravitación universal, el cual permitirá facilitar la enseñanza de este tema de manera creativa y divertida, de tal forma que el alumno verifique en forma interactiva la aplicación de esta ley física.

OBJETIVO GENERAL

El objetivo general es el análisis, diseño e implementación de un sistema de simulación educativo para las leyes físicas en el tema de la gravitación universal, que permita al alumno poder simular sus propios ambientes de experimentación física, analizar y experimentar la relación de los cuerpos u objetos frente a las leyes de gravitación universal.

OBJETIVOS ESPECÍFICOS

El proyecto abarca los siguientes objetivos específicos:

- Elaborar el análisis y diseño del sistema de simulación que cubra todos los requerimientos y procesos de la experimentación física gravitatoria, análisis de sus resultados así como sus respectivos diagramas.

Av. Universitaria 1801
San Miguel, Lima - Perú

Apartado Postal 1761
Lima 100 - Perú

Teléfono:
(511) 626 2000 Anexo 4801

FACULTAD DE
**CIENCIAS E
 INGENIERÍA**
 ESPECIALIDAD DE
 INGENIERÍA INFORMÁTICA

 PONTIFICIA
**UNIVERSIDAD
 CATÓLICA**
 DEL PERÚ

- Automatizar el proceso de experimentación física a través de un sistema amigable y con alto desempeño que mejore la satisfacción del alumno.
- Elaborar una arquitectura que permita que el sistema sea robusto, portable y configurable.
- Elaborar diagramas que faciliten un mejor entendimiento de los resultados por parte de los alumnos.
- Elaborar un modelo de base de datos que permita guardar las características de un experimento físico para la reutilización en la simulación.

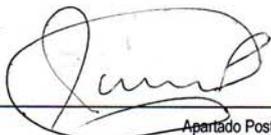
ALCANCE

- El sistema permitirá que el alumno pueda construir un ambiente de experimentación que demuestre las leyes físicas de gravitación universal.
- El sistema contendrá ejemplos didácticos incluidos que le servirán como demostración para que el alumno pueda visualizarlos y ejecutarlos.
- El sistema permitirá crear, modificar y eliminar los cuerpos y las propiedades asociadas de cada uno que se utilizarán en la simulación física.
- El sistema permitirá que el alumno pueda seleccionar, incluir o combinar cualquier tipo de cuerpo, definido en el sistema, en el ambiente de la simulación.
- El sistema permitirá realizar diagramas gráficos que demuestren los resultados que se han obtenido por la simulación del experimento físico, de forma automática o según el criterio del alumno para visualizarlo.
- El sistema permitirá grabar los ambientes construidos por el alumno a una base de datos.
- El sistema permitirá al alumno abrir cualquier experimento físico simulado que haya guardado.

 Av. Universitaria 1801
 San Miguel, Lima – Perú

 Apartado Postal 1761
 Lima 100 – Perú

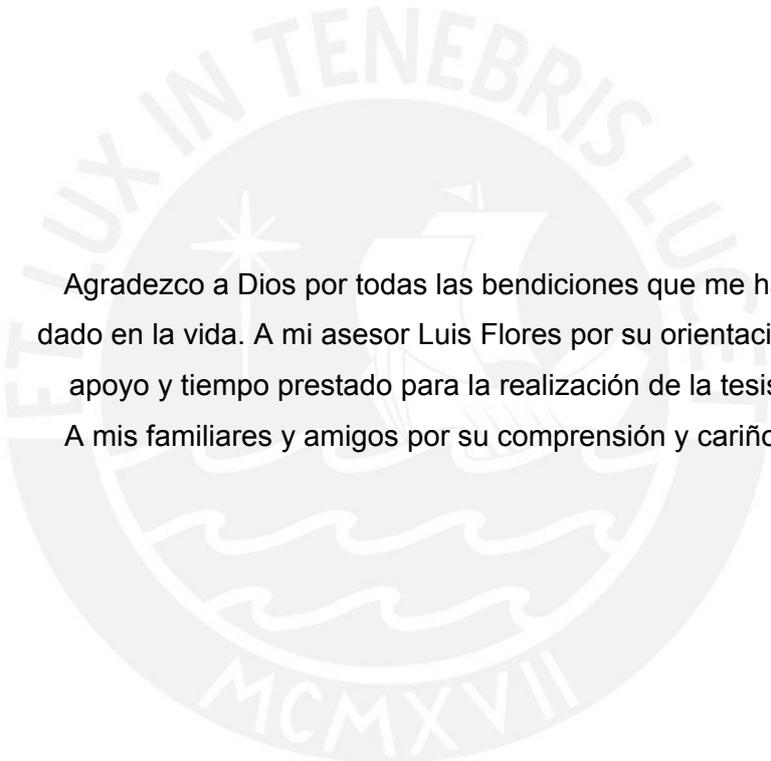
 Teléfono:
 (511) 626 2000 Anexo 4801



Máximo: 100 páginas







Agradezco a Dios por todas las bendiciones que me ha dado en la vida. A mi asesor Luis Flores por su orientación, apoyo y tiempo prestado para la realización de la tesis. A mis familiares y amigos por su comprensión y cariño.

Tabla de Contenido

Introducción	1
1. Generalidades	3
1.1. Definición del Problema.....	3
1.1.1. Estudio de la Situación Actual	5
1.1.2. Solución de la Situación Actual:	5
1.2. Marco Conceptual.....	5
1.3. Plan del Proyecto.....	8
1.3.1. Metodología del Proyecto.....	8
1.3.2. Diagrama WBS.....	10
1.3.3. Estimación del Proyecto	14
1.3.4. Cálculo Total de los Puntos de Función.....	16
1.3.5. Cronograma de Gantt.....	17
1.3.6. Variación entre el Tiempo Estimado y el Tiempo Real.....	19
1.4. Estado del Arte	20
1.5. Descripción y Sustentación de la Solución.....	25
2. Análisis	27
2.1 Metodología Utilizada para la Solución	27
2.2 Identificación de Requerimientos.....	31
2.3 Análisis de la Solución.....	33
2.3.1 Viabilidad del Sistema.....	33
2.3.2 Análisis Técnico y Económico.....	34
2.3.3 Asignar Funciones	35
2.4 Definición del Sistema	36
2.5 Descripción de los Casos de Uso.....	36
2.5.1 Especificación de Requerimientos de Software (ERS).....	37
2.6 Diagrama de Clases (Análisis)	40
3. Diseño	41
3.1 Arquitectura de la Solución.....	41
3.1.1 Descripción	41
3.1.2 Diagrama de Arquitectura	42
3.1.3 Diagrama de Paquetes	44
3.1.4 Diagrama de Clases de Diseño	45
3.1.5 Diagrama de Componentes	51
3.1.6 Diagrama de Despliegue.....	52

3.1.7	Diagrama de Secuencia.....	52
3.1.8	Modelo de Base de Datos.....	53
3.1.9	Diccionario de Datos.....	54
3.2	Diseño de Interfaz Gráfica.....	55
3.2.1	Estándares de Interfaz.....	55
3.2.2	Pantalla Principal.....	61
3.2.3	Pantalla Secundaria.....	62
3.2.4	Pasos para ejecutar una Simulación y mostrar sus resultados.....	63
4.	Construcción.....	66
4.1	Construcción del Sistema.....	66
4.1.1	Lenguaje de Programación Java.....	66
4.1.2	Applets.....	67
4.1.3	Java Data Base Connectivity (JDBC).....	68
4.1.4	SQLite – Base de Datos Compacta.....	68
4.1.5	Entorno de Desarrollo Integrado (IDE).....	69
4.1.6	Java Development Kit (JDK).....	69
4.1.7	Librerías.....	70
4.1.8	Bibliotecas Gráficas.....	70
4.1.9	Estándares de Programación.....	70
4.2	Pruebas.....	74
4.2.1	Selección del Tipo de Prueba.....	74
4.2.2	Plan de Pruebas.....	75
4.3	Algoritmo de Simulación.....	82
4.4	Variables Consideradas en el Algoritmo:.....	87
5	Observaciones, Conclusiones y Recomendaciones.....	89
5.1	Observaciones.....	89
5.2	Conclusiones.....	90
5.3	Recomendaciones Para Trabajos Futuros.....	91
6	Referencias.....	93

Índice de Figuras

Figura 1.1: Formula de la Ley de Gravitación Universal.....	7
Figura 1.2: Diagrama de WBS 1	11
Figura 1.3: Diagrama de WBS 2	12
Figura 1.4: Diagrama de WBS 3	13
Figura 2.1: Diagrama de Casos de Uso.....	30
Figura 2.2: Diagrama de Clases de Análisis	40
Figura 3.1: Diagrama de Arquitectura.....	43
Figura 3.2: Diagrama de Paquetes	44
Figura 3.3: Diagrama de Clases de Diseño de la Capa de Interfaz 1.....	47
Figura 3.4: Diagrama de Clases de Diseño de la Capa de Interfaz 2.....	48
Figura 3.5: Diagrama de Clases de Diseño de la Capa de Lógica de Negocios.....	49
Figura 3.6: Diagrama de Clases de Diseño Componente de Entidades de Negocio. ..	50
Figura 3.7: Diagrama de Clases de Diseño de la Capa de Acceso a Base de Datos. .	51
Figura 3.8: Diagrama de Componentes.....	52
Figura 3.9: Diagrama de Despliegue.....	52
Figura 3.10: Diagrama de Secuencia – Generar Simulación.....	53
Figura 3.11: Diagrama de Base de Datos.....	54
Figura 3.12: Estándar de interfaz, Pagina Web del Simulador.....	55
Figura 3.13: Estándar de Interfaz, Ventana de Búsqueda de Simulación.....	56
Figura 3.14: Estándar de Interfaz, Ventana de Registrar ó Modificar Simulación	57
Figura 3.15: Estándar de Interfaz, Ventana de Registrar ó Modificar Objetos.....	58
Figura 3.16: Estándar de Interfaz, Ventana de Eliminar Objeto.....	59
Figura 3.17: Estándar de Interfaz, Ventana de Reportes.....	60
Figura 3.18: Diseño de Interfaz Gráfica, Pantalla Inicial	63
Figura 3.19: Diseño de Interfaz Gráfica, Pantalla Inicial Insertar Objeto.....	64
Figura 4.1: Método Runge – Kutta, Forma Generalizada.....	83
Figura 4.2: Método Runge – Kutta, Función Incremento.....	84
Figura 4.3: Método Runge – Kutta, Cálculo de las k.....	84
Figura 4.4: Método Runge – Kutta, Ecuación de Cuarto Orden.....	84

Índice de Tablas

Tabla 1.1: Puntos de Función Sin Ajustar de ILF y EIF	15
Tabla 1.2: Puntos de Función Sin Ajustar para la Primera Iteración.	15
Tabla 1.3: Puntos de Función Sin Ajustar para la Segunda Iteración.....	15
Tabla 1.4: Puntos de Función Sin Ajustar para la Tercera Iteración.....	16
Tabla 1.5: Cálculo de los Puntos de Función por Iteración.....	16
Tabla 1.6: Cronograma de Gantt.	19
Tabla 1.7: Cuadro de Semejanzas.....	22
Tabla 1.8: Cuadro de Diferencias No Funcionales.	22
Tabla 1.9: Cuadro de Diferencias Funcionales.....	25
Tabla 2.1: Análisis de la Solución, Costos del Proyecto	35
Tabla 3.1: Íconos de la Pantalla Principal	62
Tabla 4.1: Estándar de Programación de Clases	71
Tabla 4.2: Estándar de Programación de Atributos	72
Tabla 4.3: Estándar de Programación de Clases como Atributos	72
Tabla 4.4: Estándar de Programación de Controles como Atributos	73
Tabla 4.5: Pruebas Unitarias, Capa Lógica.	76
Tabla 4.6: Pruebas Unitarias, Componente de Entidades del Negocio.....	78
Tabla 4.7: Pruebas Unitarias, Capa de Acceso a Base de Datos.....	79
Tabla 4.8: Pruebas de Sistema, Administrar Simulación	80
Tabla 4.9: Pruebas de Sistema, Asignar Objetos	80
Tabla 4.10: Pruebas de Sistema, Ejecutar Simulación	81
Tabla 4.11: Pruebas de Sistema, Cargar Simulación	81
Tabla 4.12: Variables en el Algoritmo	88

Introducción

El presente proyecto de fin de carrera trata sobre un sistema de simulación para la enseñanza de la Gravitación Universal, el cual permite ayudar a los alumnos en el análisis de fenómenos físicos y en el fomento del aprendizaje, tanto en el aspecto teórico como en el práctico.

La Gravitación Universal propuesta por Newton permite predecir el movimiento de los objetos o cuerpos en el espacio, según las fuerzas que actúan sobre dicho cuerpo. Ver en [Newton 1687]. En base a esta teoría, se construye un simulador de física.

Para el desarrollo de experimentos físicos en los laboratorios se resaltan algunos problemas, tales como: el alto costo de la infraestructura del ambiente y los materiales, la falta de demostración de los experimentos físicos en relación con el tema de Gravitación Universal, los complicados procesos de preparación y de desarrollo de los experimentos físicos. Una alternativa para solucionar estos problemas consiste en utilizar sistemas de simulación.

El simulador de física propuesto permite diseñar escenarios, incorporando en ellos diversos objetos o cuerpos y asignándoles propiedades físicas. De este modo es posible estudiar la interacción entre distintas variables físicas presentes en el movimiento y visualizarlas mediante diagramas.

Los diagramas del simulador son velocidad versus tiempo, posición versus tiempo y fuerza versus tiempo. Estos diagramas permiten comparar el comportamiento entre diversos movimientos. Todas estas acciones son llevadas a cabo utilizando una interfaz gráfica amigable.

Se incluye además la formulación física con la que se construyó el sistema, según la teoría de Gravitación Universal, así como ejemplos típicos que permiten afianzar esta teoría.

La teoría de Gravitación Universal de Newton, describe el movimiento de los cuerpos en el espacio, la cual dice que: “Toda partícula en el universo atrae a otra partícula con una fuerza atractiva que es proporcional a las masas de los cuerpos e inversamente proporcional al cuadro de las distancias entre ellas”. Ver en [Newton 1687].

La presente monografía contiene cinco capítulos. Cada capítulo está dividido en secciones, las cuales describen las características del proyecto según el tema correspondiente.

En el primer capítulo se describe la definición del problema, tales como el estudio de la situación actual y su solución, además del marco conceptual, es decir una breve introducción teórica donde se tratan los principales conceptos físicos que cubre el simulador, el plan de desarrollo del simulador, el estado del arte es decir los proyectos anteriores que siguen el mismo objetivo y la descripción y la sustentación de la solución propuesta.

En el segundo capítulo se incluye la definición de la metodología utilizada para la solución, la identificación de los requerimientos, el análisis de la solución, la definición del simulador, la descripción de los casos de uso y los diagramas de clases de análisis.

En el tercer capítulo se incluye la descripción de la arquitectura de la solución, utilizando diversos diagramas tales como de arquitectura, de paquetes, de clases de diseño, de componentes, de despliegue, de secuencia y de base de datos. Además, se describe el diccionario de datos y el diseño de la interfaz gráfica, el cual contiene sus estándares, la descripción y la visualización de la pantalla principal y las pantallas secundarias y una breve explicación de los pasos para la ejecución de una simulación y mostrar los resultados obtenidos.

En el cuarto capítulo se describen las tecnologías utilizadas para la construcción del sistema, tales como el lenguaje de programación java, los applets, el JDBC, el SQLite, el IDE, el JDK, las librerías, las bibliotecas gráficas y los estándares de programación. Además, las pruebas realizadas al simulador y la descripción del algoritmo de simulación y las variables consideradas.

Finalmente en el quinto capítulo se presentan las observaciones, conclusiones y algunas recomendaciones importantes para trabajos futuros relacionados al tema del proyecto.

1. Generalidades

El presente capítulo describe la definición del problema, el estudio del marco conceptual, el plan del proyecto, el estado del arte y la justificación de la solución.

1.1. Definición del Problema

Como todas las ciencias, la física parte de observaciones experimentales y mediciones cuantitativas. El principal objetivo de la física es encontrar el limitado número de leyes que gobiernan los fenómenos naturales para desarrollar teorías que puedan predecir los resultados de futuros experimentos. Las leyes fundamentales empleadas en el desarrollo de teorías se expresan en el lenguaje de las matemáticas, herramientas que proporciona un puente entre la teoría y el experimento.

Cuando surge una discrepancia entre la teoría y el experimento, deben formularse nuevas teorías para eliminar la discrepancia. Muchas veces una teoría es satisfactoria solo en condiciones limitadas; una teoría más general podría ser satisfactoria sin estas limitaciones. Por ejemplo, las leyes del movimiento descubiertas por Isaac Newton (1642 - 1727) en el

siglo XVII describen con presión el movimiento de los cuerpos con rapidez normal. Ver en [Serway 2000].

Para el alumno los temas del dominio de física pueden ser difíciles de entender, para ello el alumno requiere de un entorno de aprendizaje que permitiría interactuar con el problema para poder entenderlo. Utilizar un método tradicional, sin interacción, hace que el alumno no intente desafiar o motivar su resolución, tal como ocurre con algunos temas del dominio de física.

Para un mejor entendimiento de los temas del dominio de física se requiere realizar experimentos en los laboratorios, pero algunos de estos experimentos resultan costosos ya que se debe tomar en cuenta la infraestructura del ambiente y los materiales, los cuales en algunos casos son difíciles de efectuar. Por tal motivo el aprendizaje de la física resulta dificultoso para el alumnado debido a que no le encuentra un sentido práctico y demostrativo. La mayor parte del aprendizaje se basa solo en la teoría, de este modo se desanima y se desorienta al alumnado hacia una posible investigación acerca de este tema. Ver en [Auris Rodríguez 2001].

Uno de los temas del dominio de física es la Gravitación Universal, la cual permite definir el comportamiento que tienen los cuerpos físicos frente a otros, en este caso sobre las causas de los cambios en sus movimientos: En particular este tema no es comprendido completamente por el alumno ya que no es posible representar estos fenómenos en una situación real en los laboratorios.

La tecnología de la información que actualmente se presenta permite tomar nuevos retos en el desarrollo de la educación, debido a que le proporciona una gama inmensa de herramientas y recursos para la investigación de diversos temas hacia el alumno y permite nuevos métodos de aprendizaje y de enseñanza, tales como los sistemas de simulación educativos, que toman importancia en el proceso de educación de hoy y en el futuro.

Bajo este esquema se plantea la implementación de un sistema de simulación educativo en el tema de Gravitación Universal, el cual permitiría facilitar la enseñanza de este tema de manera creativa y divertida, de tal forma que el alumno verifique en forma interactiva la aplicación de este tema físico.

1.1.1. Estudio de la Situación Actual

Para realizar experimentos de Gravitación Universal en el dominio de física en los laboratorios, se destacan los siguientes problemas:

- Alto costo de la infraestructura del ambiente y los materiales.
- Falta de demostración de los experimentos relacionados con el tema de Gravitación Universal.
- Complicados procesos de preparación y de desarrollo de los experimentos físicos.

1.1.2. Solución de la Situación Actual:

Para solucionar los problemas descritos en la sección anterior 1.1.1 se propone la construcción de un simulador de experimentos físicos en el tema de Gravitación Universal.

El simulador tienen las siguientes ventajas:

- Elimina los gastos operativos y logísticos en el manejo de instrumentos o material de laboratorio.
- Facilita la tarea del educador, sustituyéndole parcialmente en su labor.
- Se adaptan al ritmo de trabajo del estudiante, menos demora en el proceso para realizar el experimento.

1.2. Marco Conceptual

La presente sección muestra las definiciones relacionadas con el tema del proyecto de fin de carrera: la teoría de la Gravitación Universal. Se explica algunos conceptos físicos, como también algunos aspectos del ambiente de desarrollo de los experimentos, tanto en un laboratorio como en un simulador.

- Cantidad Vectorial: Es una cantidad física que requiere la especificación tanto de una dirección como de una magnitud. Ver en [Serway 2000]. Por ejemplo, el desplazamiento y la velocidad de un objeto es una cantidad vectorial.
- Cantidad Escalar: Es una cantidad física que tiene magnitud pero no dirección. Ver en [Serway 2000]. Por ejemplo, la rapidez es una cantidad escalar.
- Masa: Es una cantidad escalar, es una propiedad inherente de un cuerpo y es independiente de los alrededores del cuerpo y del método utilizado para medirla. Unidad fundamental de la masa SI, el kilogramo (kg). Ver en [Serway 2000] Por ejemplo, la masa de la tierra es de 5.9736×10^{24} Kg.
- Desplazamiento: Es el cambio en la posición. Ver en [Serway 2000]. Por ejemplo, la luna tiene un desplazamiento circular.
- Velocidad: La velocidad promedio de una partícula se define como el desplazamiento de la partícula dividido entre el intervalo de tiempo durante el cual ocurre el desplazamiento. Ver en [Serway 2000]. Por ejemplo, la velocidad de la tierra es 29.78 km/s.
- Rapidez: La rapidez promedio de una partícula, es una cantidad escalar, se define como el cociente entre la distancia total recorrida y el tiempo total que lleva viajar esa distancia. Ver en [Serway 2000].
- Aceleración: Es el cambio en velocidad dividido entre el intervalo de tiempo en que ocurre dicho cambio. Ver en [Serway 2000].
- Fuerza: Ocasiona que un cuerpo se acelere. Ver en [Serway 2000]. Por ejemplo, la luna gira alrededor de la tierra debido a la fuerza de atracción de la tierra.
- Fuerza de Gravedad: Es el producto de la masa, el cual es una cantidad escalar, y la aceleración de caída libre. Ver en [Serway 2000].
- Primera Ley de Newton: Un cuerpo en reposo permanece en reposo y un cuerpo en movimiento uniforme en línea recta mantiene ese movimiento. Ver en [Serway 2000].

- Segunda Ley de Newton: La aceleración de un objeto es directamente proporcional a la fuerza que actúa sobre él e inversamente proporcional a su masa. Ver en [Serway 2000].
- Tercera Ley de Newton: Si dos cuerpos, cuerpo 1 y cuerpo 2 interactúan, la fuerza ejercida por el cuerpo 1 sobre el cuerpo 2 es igual en magnitud y opuesta en dirección a la fuerza ejercida por el cuerpo 2 sobre el cuerpo 1. Ver en [Serway 2000]. La fuerza de la Gravitación Universal se calcula tal como se muestra en la figura 1.1.

$$F = G \frac{m_1 m_2}{d^2}$$

Figura 1.1: Formula de la Ley de Gravitación Universal

- Movimiento: Representa el cambio continuo en la posición de un objeto. Ver en [RAE 2011]. Por ejemplo, la tierra tiene un movimiento circular.
- Cuerpo: Aquello que tiene extensión limitada, perceptible por los sentidos. Ver en [RAE 2011]. Por ejemplo, un planeta, una estrella, un cometa.
- Programa informático: Es un conjunto unitario de instrucciones que permite a un ordenador realizar funciones diversas, tales como el diseño de gráficos, la resolución de problemas matemáticos, el manejo de bancos de datos, etc. Ver en [RAE 2011].
- Modelo: Es un esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja. Ver en [RAE 2011]. Por ejemplo, un modelo matemático que permita predecir el movimiento planetario según diversos criterios.
- Experimentar: Operaciones destinadas a descubrir, comprobar o demostrar determinadas fenómenos o principios científicos. Ver en [RAE 2011]. Por ejemplo, en el sistema de simulación de Gravitación Universal, un experimento sería crear una nueva simulación, insertar objetos y ejecutar la simulación.
- Laboratorio: Lugar dotado de los medios necesarios para realizar investigaciones, experimentos y trabajos de carácter científico o técnico. Ver en [RAE 2011]. Por ejemplo, un laboratorio de experimentos físicos de un colegio o academia.

- Interacción: Acción que se ejerce recíprocamente entre dos o más objetos, agentes, fuerzas, funciones, etc. Ver en [RAE 2011]. Por ejemplo, dos cuerpos interactúan entre sí, con la fuerza de gravedad, atrayéndose uno hacia el otro.
- Simulador: Aparato que reproduce el comportamiento de un sistema en determinadas condiciones. Ver en [RAE 2011]. Por ejemplo, realizar experimentos de simulación sobre la fuerza de gravedad de los objetos.
- Simular: Representar algo, fingiendo o imitando lo que no es. Ver en [RAE 2011]. Por ejemplo, simular el movimiento planetario en un sistema informático.
- Entorno: Ambiente, lo que rodea. Ver en [RAE 2011]. Por ejemplo, los planetas, las estrellas, los cometas que están alrededor de un planeta.
- Computarizar: Someter datos al tratamiento de una computadora. Ver en [RAE 2011]. Por ejemplo, un sistema que permita automatizar la información y simular el movimiento de los planetas.

1.3. Plan del Proyecto

La presente sección muestra la metodología utilizada en el proyecto de fin de carrera, el diagrama WBS, la estimación del proyecto, el cálculo total de puntos de función, el cronograma de Gantt y la variación entre el tiempo estimado y el tiempo real.

1.3.1. Metodología del Proyecto

Para el proyecto de fin de carrera se utiliza el PMBOK (Project Management Body of Knowledge) publicado por el Project Management Institute como guía para la gestión del proyecto de fin de carrera, ya que cuenta con las mejores prácticas en los procesos y áreas de conocimiento de gestión de proyectos. En el caso del proyecto se pone énfasis en la gestión de alcance, tiempo, calidad y comunicación.

De acuerdo al PMBOK la gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir sus requerimientos. Ver en [PMI 2009].

El PMBOK define nueve áreas de conocimiento, las cuales ocho son aplicados en el proyecto de fin de carrera de la siguiente manera:

- Integración del proyecto: Incluye los procesos y actividades del proyecto. Entre las actividades del proceso para el desarrollo del simulador se tiene: Administrar simulaciones, asignar objetos, cargar simulación, ejecutar simulación, establecer efectos visuales y generar reportes.
- Alcance del proyecto: Incluye procesos para garantizar el éxito del proyecto. Entre los procesos principales del proyecto se encuentran: La arquitectura del proyecto, la cual será la de una aplicación Web que utilice una base de datos. La aplicación debe realizar el proceso de simulación del movimiento de los objetos según la teoría de Gravitación Universal. El proceso de experimentación física es automatizada desde la creación de la simulación hasta la generación de los reportes a través de un navegador Web. El sistema es portable para que sea utilizada en diversas plataformas. El sistema genera diversos diagramas o tabulaciones. Para el sistema se utiliza una base de datos compacta, el cual almacena los datos en un archivo binario en el servidor.
- Plazos del proyecto: Incluye procesos para administrar la finalización del proyecto en el plazo establecido. Según los requerimientos del sistema y las actividades que se muestran en los diagramas WBS, tal como se muestra en la figura 1.2, 1,3 y 1.4, se definen las iteraciones de entrega del simulador. La primera iteración son los mantenimientos de las simulaciones y los objetos. La segunda iteración es la carga, ejecución, pausa y detención de los objetos de la simulación, cuyos movimientos se establecen según la teoría Gravitación Universal. La tercera y última iteración es realizar los efectos visuales y los reportes según la simulación realizada. Durante cada iteración se realizan las pruebas para verificar su correcto funcionamiento. Para el proyecto de fin de carrera, el tiempo planificado aproximado es de 700 horas / hombre. El detalle de esta estimación se muestra en la sección 1.3.3 del presente capítulo.
- Costes del proyecto: Incluye procesos para estimar, presupuestar y controlar los costos del proyecto. Para el desarrollo se debe contar con una computadora para realizar la codificación del simulador y una impresora. Los programas para el desarrollo son libres y se descargan de internet para ser utilizados sin ninguna restricción, adicionalmente el cliente debe contar con un servidor Web, el cual debe estar con conexión a internet. La

descripción del costo total se encuentra en la sección 2.3.2, análisis técnico y económico.

- Calidad del proyecto: Incluye procesos que determinan los objetivos y políticas de calidad. Se debe definir una planificación de controles, es decir, establecer las iteraciones en que se realizarán los entregables del simulador para su posterior revisión. En las iteraciones se debe revisar y anotar los resultados obtenidos por el simulador además de los comentarios del progreso o fallas en las pruebas.
- Recursos humanos del proyecto: Incluye procesos que organizan, gestionan y conducen el proyecto. Para los recursos humanos del proyecto se cuenta con una persona, que es el tesista, quien cumple la función de analista - programador. La función de analista - programador se encarga de la codificación de la aplicación y del estudio de los requerimientos de los usuarios para la aplicación.
- Comunicaciones del proyecto: Incluye procesos para garantizar la disposición de la información del proyecto. El documento de tesis es una monografía que describe de manera sencilla en lenguaje formal el contenido del proyecto de fin de carrera, de esta manera se asegura la oportuna y apropiada difusión, recolección, almacenamiento y disposición del proyecto. La comunicación con el asesor de tesis es a través de reuniones semanales y de envíos de correos electrónicos, la finalidad de cada reunión es la presentación de los avances del documento de tesis.
- Riesgos del proyecto: Incluye los procesos para identificar, analizar y planificar la respuesta de los riesgos. Las amenazas o riesgos durante el proceso de desarrollo fueron de probabilidad baja o impacto bajo, debido a la claridad de los alcances del proyecto, así como la utilidad de las tecnologías aplicadas para su construcción. Un riesgo para el proyecto sería la falta de experiencia del analista – programador con las tecnologías empleadas y para ello se requiere un tiempo de investigación en dichas tecnologías para saber emplearlas.

1.3.2. Diagrama WBS

La presente sección muestra el diagrama WBS, el diagrama se muestra en las figuras 1.2, 1.3 y 1.4. El diagrama presenta los trabajos que se deben realizar en el sistema de

simulación educativo, entre los cuales se destacan: los entregables de tesis, la elaboración de los capítulos, el desarrollo y las pruebas del sistema.

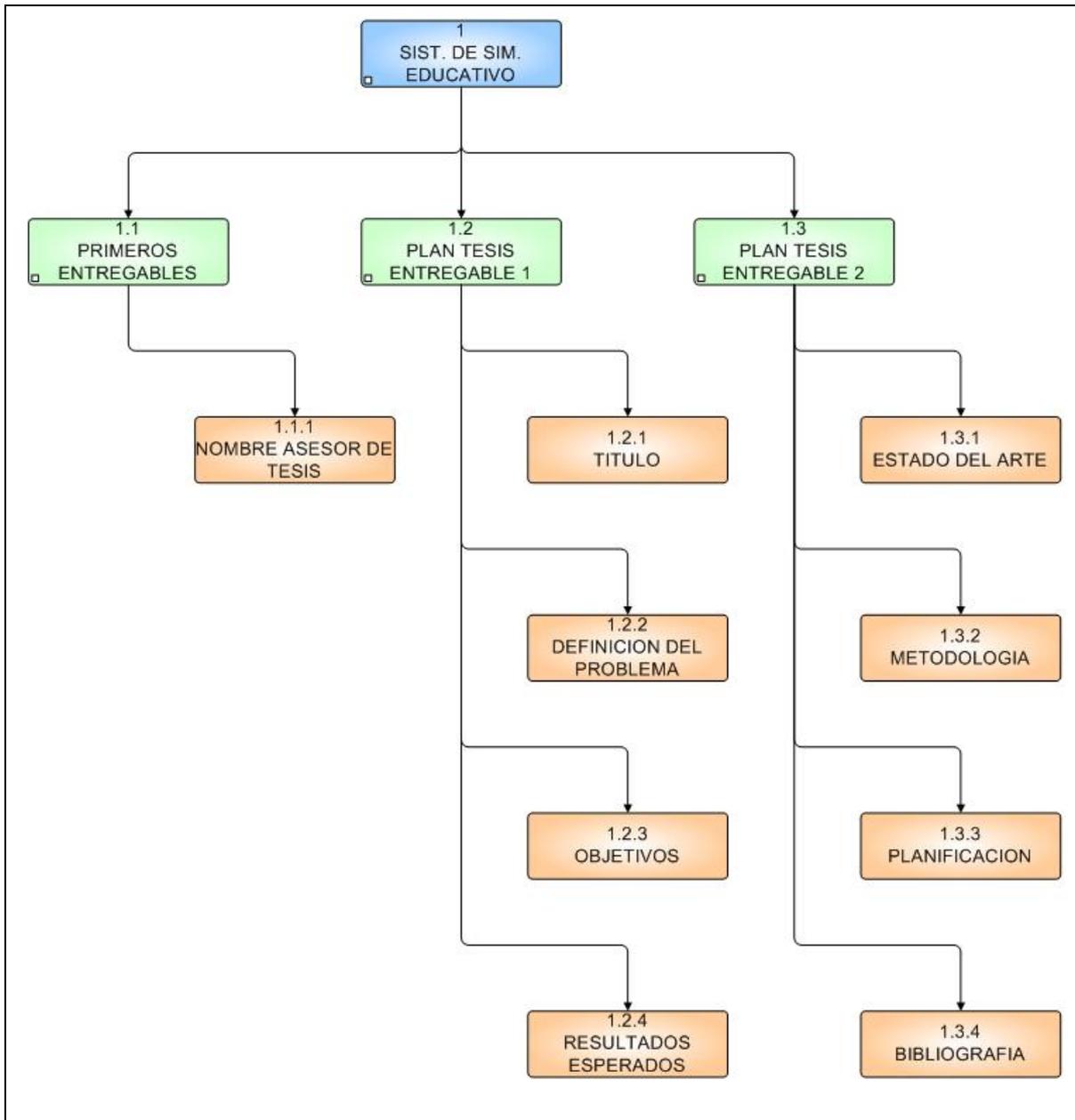


Figura 1.2: Diagrama de WBS 1

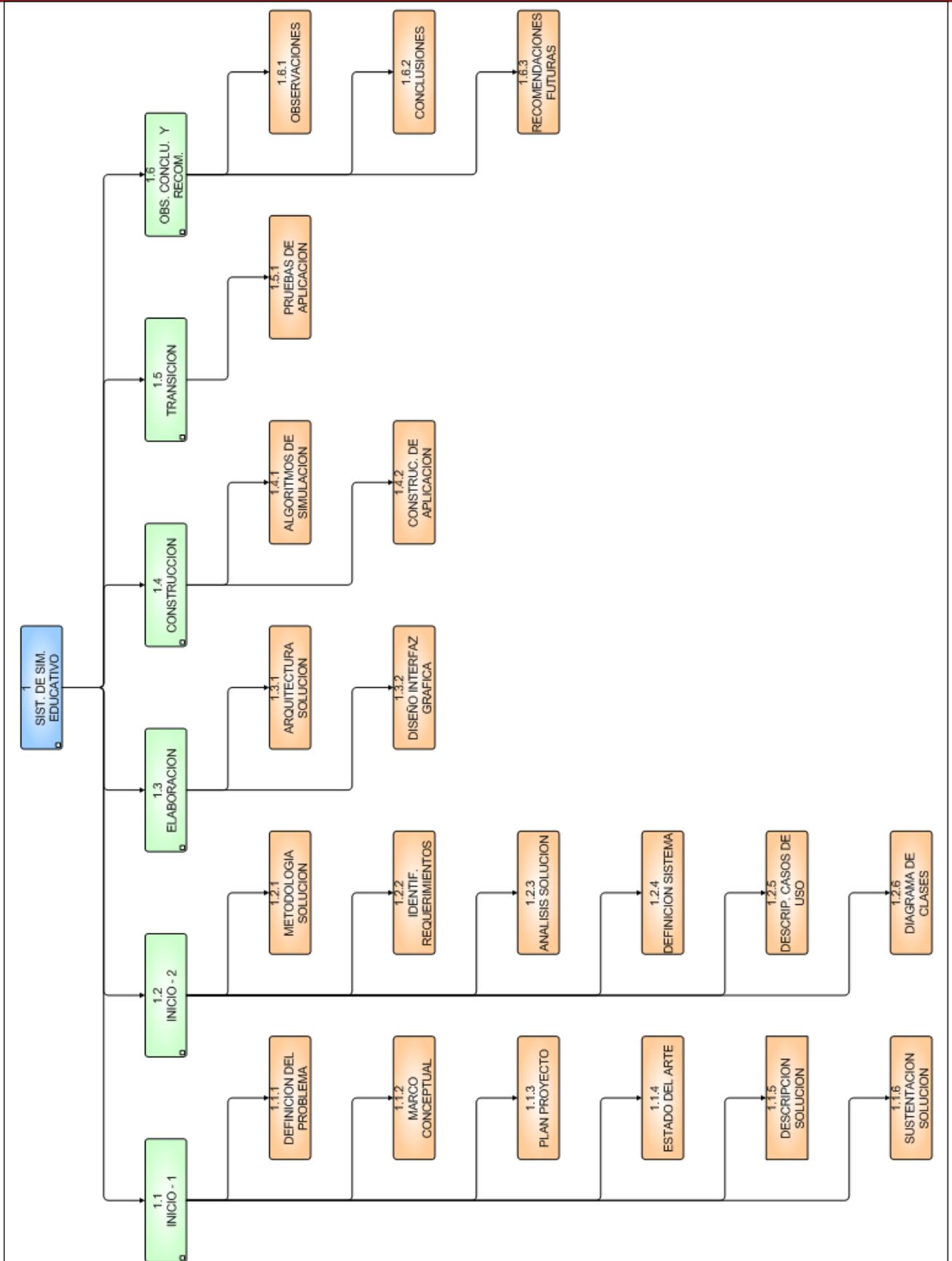


Figura 1.3: Diagrama de WBS 2

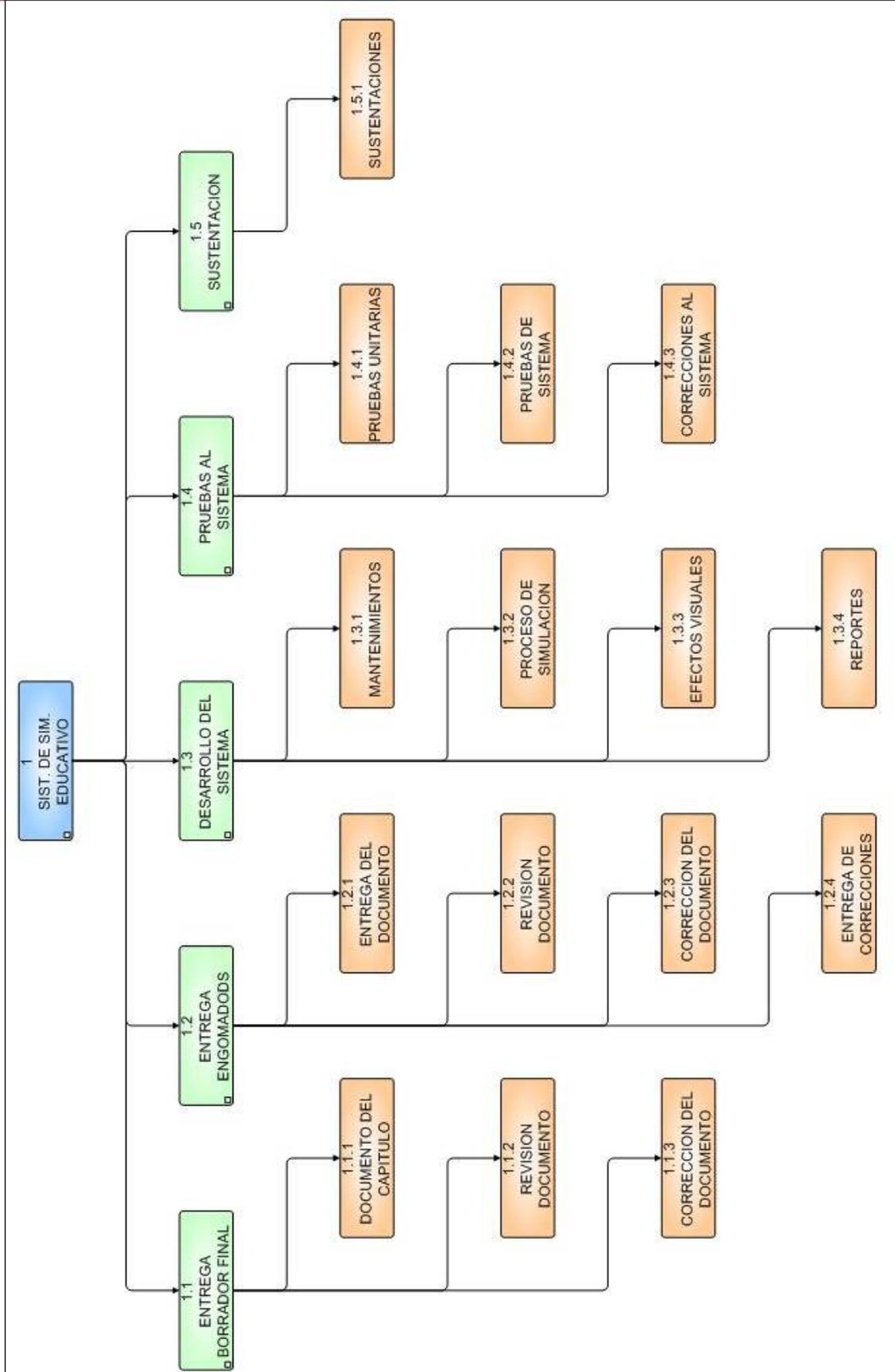


Figura 1.4: Diagrama de WBS 3

1.3.3. Estimación del Proyecto

En la presente sección se calcula la estimación de duración, coste y esfuerzo necesario para construir el producto del proyecto de fin de carrera.

Para calcular la estimación del producto se utiliza un modelo de estimación compuesto llamado: Puntos de Función.

Los puntos de función miden el software cuantificando y cualificando la funcionalidad que proporciona externamente, basándose en el diseño lógico del sistema. Para los subsistemas diseñados independientemente, los puntos de función se calculan para cada una de ellas y luego se suman. Ver en [IFPUG 1999].

Para calcular la estimación del software primero se debe determinar los casos de uso del sistema, los cuales se describen en el capítulo 2. Los casos de uso se implementarán por iteración, siendo el resultado final de esta tarea lo siguiente:

- Primera Iteración: Los casos de uso de la iteración son: Administrar Simulaciones y Asignar Objetos.
- Segunda Iteración: Los casos de uso de la iteración son: Cargar Simulación y Ejecutar Simulación.
- Tercera Iteración: Los casos de uso de la iteración son: Establecer Efectos Visuales y Generar Reportes.

El detalle del desarrollo de la estimación del sistema según los puntos de función se encuentra en los anexos, a continuación se muestra el resultado final.

Tal como se muestra en la tabla 1.1, se obtienen 21 puntos de función sin ajustar de ILF y EIF del sistema y son obtenidos de ficheros de complejidad baja y media.

<u>TIPO DE FUNCIÓN</u>	<u>COMPLEJIDAD FUNCIONAL</u>	<u>TOTAL DE COMPLEJIDAD</u>	<u>TOTAL DE TIPO DE FUNCIÓN</u>
ILF	<u>3</u> BAJA x 7 =	<u>21</u>	<u>21</u>
	<u>1</u> MEDIA x 10 =	<u>10</u>	
	<u>0</u> ALTA x 15 =	<u>0</u>	
EIF	<u>0</u> BAJA x 5 =	<u>0</u>	<u>0</u>
	<u>0</u> MEDIA x 7 =	<u>0</u>	
	<u>0</u> ALTA x 10 =	<u>0</u>	

Tabla 1.1: Puntos de Función Sin Ajustar de ILF y EIF

Para la primera iteración, tal como se muestra en la tabla 1.2, se obtienen 24 puntos de función sin ajustar.

<u>TIPO DE FUNCIÓN</u>	<u>COMPLEJIDAD FUNCIONAL</u>	<u>TOTALES DE COMPLEJIDAD</u>	<u>TOTALES DE TIPO DE FUNCIÓN</u>
EI	<u>3</u> Baja x 3 =	<u>9</u>	<u>21</u>
	<u>3</u> Media x 4 =	<u>12</u>	
	<u>0</u> Alta x 6 =	<u>0</u>	
EQ	<u>1</u> Baja x 3 =	<u>3</u>	<u>3</u>
	<u>0</u> Media x 4 =	<u>0</u>	
	<u>0</u> Alta x 6 =	<u>0</u>	

Tabla 1.2: Puntos de Función Sin Ajustar para la Primera Iteración.

Para la segunda iteración, tal como se muestra en la tabla 1.3, se obtienen 20 puntos de función sin ajustar.

<u>TIPO DE FUNCIÓN</u>	<u>COMPLEJIDAD FUNCIONAL</u>	<u>TOTALES DE COMPLEJIDAD</u>	<u>TOTALES DE TIPO DE FUNCIÓN</u>
EI	<u>2</u> Baja x 3 =	<u>6</u>	<u>6</u>
	<u>0</u> Media x 4 =	<u>0</u>	
	<u>0</u> Alta x 6 =	<u>0</u>	
EO	<u>0</u> Baja x 4 =	<u>0</u>	<u>14</u>
	<u>0</u> Media x 5 =	<u>0</u>	
	<u>2</u> Alta x 7 =	<u>14</u>	

Tabla 1.3: Puntos de Función Sin Ajustar para la Segunda Iteración.

Finalmente para la tercera iteración, tal como se muestra en la tabla 1.4, se obtienen 14 puntos de función sin ajustar.

<u>TIPO DE FUNCIÓN</u>	<u>COMPLEJIDAD FUNCIONAL</u>	<u>TOTALES DE COMPLEJIDAD</u>	<u>TOTALES DE TIPO DE FUNCIÓN</u>
EI	<u>2</u> Baja x 3 =	<u>6</u>	<u>6</u>
	<u>0</u> Media x 4 =	<u>0</u>	
	<u>0</u> Alta x 6 =	<u>0</u>	
EO	<u>2</u> Baja x 4 =	<u>8</u>	<u>8</u>
	<u>0</u> Media x 5 =	<u>0</u>	
	<u>0</u> Alta x 7 =	<u>0</u>	

Tabla 1.4: Puntos de Función Sin Ajustar para la Tercera Iteración.

1.3.4. Cálculo Total de los Puntos de Función

La presente sección calcula los puntos de función por iteración y caso de uso, tal como se muestra en la tabla 1.5.

<u>ITERACIONES</u>	<u>CASOS DE USO</u>	<u>PUNTOS DE FUNCIÓN</u>
Primera iteración	Administrar Simulaciones	24
	Asignar Objetos	
Segunda iteración	Cargar Simulación	20
	Ejecutar Simulación	
Tercera iteración	Establecer Efectos Visuales	14
	Generar Reportes	

Tabla 1.5: Cálculo de los Puntos de Función por Iteración.

Tal como se muestra en la tabla 1.5, el resultado de los puntos de función se interpreta como el tamaño de cada iteración.

El resultado de la tabla 1.5 permite deducir que a mayor tamaño de la iteración, mayor será el tiempo que tomará en su desarrollo; es decir, la primera iteración tomará más tiempo de desarrollo, luego la segunda iteración y finalmente la tercera iteración.

1.3.5. Cronograma de Gantt

La presente sección muestra el cronograma de Gantt del proyecto de fin de carrera, tal como se muestra en la tabla 1.6. El tiempo aproximado para el proyecto es de 700 horas.

Para la estimación del tiempo de desarrollo de la documentación, es decir, los primeros entregables de tesis, sus capítulos y sus correcciones, se tomó como referencia los plazos dados por los cursos de Tesis 1 y Tesis 2. Esto fue un aproximado de 340 horas, considerando 3 horas de trabajo por día.

Para la estimación del tiempo de construcción de la aplicación, se tomó como referencia el tiempo de programación de la aplicación. Esto fue aproximadamente 360 horas. Considerando 8 horas de trabajo por día.

<u>TAREA</u>	<u>DETALLE</u>	<u>DIAS</u>	<u>HORAS DE TRABAJO / DIA</u>	<u>HORAS</u>	<u>CRONOGRAMA (DIAS)</u>
Plan de tesis primeros entregables	Nombre del asesor de tesis.	1	3	3	1
Plan de tesis – Entregable 1	Título. Definición del problema. Objetivos. Resultados esperados.	5	3	15	2 al 6
Plan de tesis – Entregable 2	Estado del arte. Metodología. Planificación. Bibliografía.	5	3	15	7 al 11
Entrega del Capítulo 1	Desarrollar el contenido del capítulo. Revisión del capítulo. Realizar correcciones del capítulo.	25	3	75	12 al 36

<u>TAREA</u>	<u>DETALLE</u>	<u>DIAS</u>	<u>HORAS DE TRABAJO / DIA</u>	<u>HORAS</u>	<u>CRONOGRAMA (DIAS)</u>
Entrega del Capítulo 2	Desarrollar el contenido del capítulo. Revisión del capítulo. Realizar correcciones del capítulo.	10	3	30	37 al 46
Entrega del Capítulo 3	Desarrollar el contenido del capítulo. Revisión del capítulo. Realizar correcciones del capítulo.	10	3	30	47 al 56
Entrega del Capítulo 4	Desarrollar el contenido del capítulo. Revisión del capítulo. Realizar correcciones del capítulo.	10	3	30	57 al 66
Entrega del Capítulo 5	Desarrollar el contenido del capítulo. Revisión del capítulo. Realizar correcciones del capítulo.	10	3	30	67 al 76
Entrega del borrador final	Revisión del capítulo. Realizar correcciones del capítulo.	15	3	45	77 al 91
Entrega de engomados	Revisión del capítulo. Realizar correcciones del capítulo. Entrega de las correcciones.	15	3	45	92 al 106

<u>TAREA</u>	<u>DETALLE</u>	<u>DIAS</u>	<u>HORAS DE TRABAJO / DIA</u>	<u>HORAS</u>	<u>CRONOGRAMA (DIAS)</u>
Primera iteración	Administrar Simulaciones. Asignar Objetos.	16	8	128	107 al 122
Segunda iteración	Cargar Simulación. Ejecutar Simulación.	21	8	168	123 al 143
Tercera iteración	Establecer Efectos Visuales. Generar Reportes.	11	8	88	144 al 154

Tabla 1.6: Cronograma de Gantt.

1.3.6. Variación entre el Tiempo Estimado y el Tiempo Real

Tal como se observa en la tabla 1.6, la segunda iteración fue la que tomó más tiempo de desarrollo, luego la primera iteración y finalmente la tercera iteración.

Mientras que en la tabla 1.5, se indica que la primera iteración, por ser de mayor tamaño, debería tomar más tiempo de desarrollo, luego la segunda iteración y finalmente la tercera iteración.

Esta contradicción se debe a que los puntos de función basan su estimación según el número de campos del caso de uso. Entonces es correcto afirmar que la primera iteración posee más campos, por lo tanto tiene un mayor tamaño y toma más tiempo.

Pero la primera iteración, son dos mantenimientos. Los mantenimientos son procesos que generalmente son más rápidos de implementar, debido al procedimiento mecánico para construirlos. En cambio, la segunda iteración a pesar de tener menos campos en comparación con la primera, su construcción es más compleja debido a que se desarrolla más efectos gráficos, además del desarrollo del algoritmo de movimiento según la teoría de la Gravitación Universal y la ubicación de los parámetros en dicho algoritmo.

Además, las pruebas que se realizaron al mantenimiento son más sencillas que las del proceso de simulación, ya que en un mantenimiento se verifica el ingreso de datos y que estos sean registrados correctamente en la base de datos, mientras que en el proceso de

simulación se debe verificar que cada uno de los parámetros cumpla con su función dentro del algoritmo. Así mismo se deben controlar los eventos gráficos, para evitar alguna distorsión en la imagen en el momento de la simulación.

1.4. Estado del Arte

En la actualidad existen varios enfoques con respecto a sistemas educativos. Existen desde programas orientados al aprendizaje de algunos temas específicos hasta sistemas operativos orientados a una educación más completa.

Además existen proyectos anteriores realizados en la misma línea que el proyecto de fin de carrera, las cuales se describen a continuación:

- Tesis: Simulador de física - Ley de Gravitación Universal: La tesis tiene como objetivo la implementación de un sistema de simulación físico en el tema de la Gravitación Universal postulada por Newton, que permite distinguir y describir correctamente el movimiento de los cuerpos en el espacio. El simulador de física permite diseñar escenarios incorporando en ello diversos cuerpos y asignándoles propiedades físicas. Es posible estudiar la interacción entre distintas variables físicas presentes en el movimiento mediante diagramas de variable versus variable y así también comparar el comportamiento entre diversos movimientos. Todas las acciones son llevadas a cabo utilizando una interfaz gráfica amigable al usuario y fácil de usar. Ver en [Del Mar Zarate 2001].

Existen diversas diferencias Funcionales y No Funcionales entre la tesis anterior y la tesis propuesta. Entre las mejoras No Funcionales realizadas en la tesis propuesta destacan: Es una aplicación Web, es portable en diversas plataformas y tiene una base de datos acorde a la aplicación. Entre las Funcionales destacan: El uso de un panel de información, mejoras en el proceso de visualización de las vistas y el posicionamiento de los objetos, el uso del proceso “drag and drop” y nuevas opciones para facilitar el proceso de creación, administración y ejecución de la simulación. Además de enviar el diagrama por mail, guardarlo como imagen y generar diferentes tipos de reportes.

Para una identificación más detallada de las diferencias entre la tesis anterior y la tesis propuesta se muestra en la tabla 1.9.

- Tesis: Simulador de física - Electroestática: La tesis tiene como objetivo la implementación de un sistema de simulación físico en el tema de las leyes de la electrostática, que permiten distinguir y calcular las fuerzas eléctricas producidas por partículas cargadas. Del mismo modo, las leyes de la electrodinámica permiten describir correctamente el movimiento de las cargas al ser afectadas por campos eléctricos y magnéticos. Contiene una interfaz gráfica amigable e intuitiva que permite diseñar escenarios mediante la selección de cargas y asignación de diversas propiedades físicas. Es posible estudiar el comportamiento de los campos entre las cargas estáticas, así como la interacción entre las variables físicas presentes en el movimiento utilizando diagramas de variables versus variable. Ver en [Prado Lau 2001].
- Tesis: Simulador de física – Oscilaciones: La tesis tiene como objetivo la implementación de un sistema de simulación físico en el tema de movimientos oscilatorios mecánicos, que permite diseñar escenarios de simulación seleccionando diversos objetos y asignándole sus respectivas propiedades. De esta manera se hace posible la interacción entre variables físicas presentes en el movimiento, a través de gráficas de variables que permiten observar su comportamiento matemático, así mismo se permite almacenar y cargar diversos escenarios de simulación. El software de simulación incluye una serie de ejemplos típicos dentro de la física que sirven de guía para la construcción de los escenarios de simulación, finalmente se completa el sistema con una guía de ayuda que contiene la descripción teórica de los movimientos oscilatorios. Ver en [Auris Rodríguez 2001].
- Tesis: Simulador de física: Cantidad de movimiento y colisiones: La tesis tiene como objetivo la implementación de un sistema de simulación físico en el tema de colisiones y choques. Para su desarrollo se revisaron los conceptos físicos relacionados a colisiones y choques, así como diversos productos similares de simulaciones de varios temas de física. La interfaz del usuario es de fácil manejo. Es un software de simulación interactivo y amigable orientado principalmente a la visualización gráfica de los resultados de experimentos físicos de colisiones, además de los matemáticos, teniendo las funcionalidades de abrir, guardar y modificar las propiedades del escenario de simulación, así como modificar las propiedades de los cuerpos involucrados, ejecutar varias simulaciones a la vez y graficar los resultados. Ver en [Rodríguez Castillo 2001].

Las tesis anteriormente mencionadas son proyectos de desarrollo similares que resuelven diferentes ramas del dominio de la física tales como la Gravitación Universal, la electrostática, las oscilaciones y cantidad de movimiento y colisiones a través de

procedimientos computacionales. La tesis propuesta es un proyecto de desarrollo para la Gravitación Universal. Tal como se muestra en la tabla 1.7, se describen las semejanzas entre las tesis anteriores y el proyecto de fin de carrera.

<u>SEMEJANZAS</u>
Es un sistema de simulación físico.
Permite diseñar, almacenar, cargar y ejecutar escenarios de simulación.
Permite estudiar la interacción entre distintas variables físicas presentes en el movimiento.
Realiza diagramas de variable versus variable, el cual permite visualizar el comportamiento entre los objetos.

Tabla 1.7: Cuadro de Semejanzas.

En la tabla 1.8, se describen las diferencias No Funcionales entre las tesis anteriores y el proyecto de fin de carrera.

<u>DIFERENCIAS NO FUNCIONALES</u>	
<u>TESIS ANTERIORES</u>	<u>PROYECTO DE FIN DE CARRERA</u>
Es una aplicación cliente – servidor.	Es una aplicación Web.
Utiliza archivos para almacenar información.	Utiliza una base de datos para almacenar la información.
Se utiliza para su desarrollo el lenguaje pascal.	Se utiliza para su desarrollo, el lenguaje java para garantizar la portabilidad.

Tabla 1.8: Cuadro de Diferencias No Funcionales.

En la tabla 1.9, se describen las diferencias Funcionales entre la tesis de simulador de física de Gravitación Universal y el proyecto de fin de carrera.

<u>DIFERENCIAS FUNCIONALES</u>	
<u>TESIS ANTERIOR</u>	<u>PROYECTO DE FIN DE CARRERA</u>
En el panel de animación de la simulación no se muestra la posición del objeto.	En el panel de animación, se muestra junto al objeto su posición (x, y).

<u>DIFERENCIAS FUNCIONALES</u>	
<u>TESIS ANTERIOR</u>	<u>PROYECTO DE FIN DE CARRERA</u>
En el panel de animación de la simulación no se muestra la posición del objeto.	La posición es según la vista en la que se visualiza. Esta posición se muestra antes, durante y al finalizar la ejecución.
Las vistas no muestran una escala de valores que indiquen el espacio que cubre.	En cada vista (superior, frontal y lateral) se muestra como un plano cartesiano (x, y) con sus respectivas escalas valores y ejes, esto permite indicar el espacio que se cubre durante la simulación.
No cuenta con un panel de información.	Junto al panel de animación de la simulación, se cuenta con un panel de información. Este panel contiene pestañas con los valores de cada uno de los objetos de la simulación. Estos valores cambian en cada instante del tiempo de la ejecución.
Para cambiar la posición del objeto se requiere ir al menú, seleccionar la opción de "Modificar Objeto" luego seleccionar el objeto y colocar los valores de la nueva posición. Este método no permite obtener con exactitud la posición inicial deseada por el usuario, ya que debe predecir los valores de la posición del objeto en el panel de animación.	Para cambiar la posición de un objeto en el panel de animación solo se requiere presionar con el mouse sobre el objeto, arrastrarlo y soltarlo en la posición que se desea. (Acción Drag and Drop). De esta manera el usuario obtiene la exactitud de la posición que desea para el objeto.
No existe la opción "Deshabilitar". Si se desea que un objeto no participe en la simulación es necesario borrarlo de la simulación.	Existe la opción "Deshabilitar" esta opción permite que un objeto no participe en la simulación sin la necesidad de borrarlo de la simulación.
No existe la opción "Deshabilitar".	Si se desea deshabilitar un objeto de la simulación, solo es necesario hacer un clic derecho con el mouse sobre el objeto en el panel de animación y seleccionar la opción "Deshabilitar".

<u>DIFERENCIAS FUNCIONALES</u>	
<u>TESIS ANTERIOR</u>	<u>PROYECTO DE FIN DE CARRERA</u>
<p>Para insertar un objeto es necesario ir al menú, seleccionar la opción “Insertar Objeto”, para luego ingresar los valores deseados. Este método es más lento y poco amigable porque se requiere realizar muchos pasos para su operación.</p>	<p>Para insertar un objeto, solo es necesario hacer clic derecho con el mouse sobre el panel de animación y seleccionar la opción “Insertar Objeto” y luego ingresar los valores deseados. Este método permite de una manera más sencilla, rápida y amigable de agregar un objeto en la simulación.</p>
<p>Para modificar las propiedades del objeto es necesario ir al menú, seleccionar la opción “Modificar Objeto” y luego seleccionar el objeto, para luego ingresar los valores deseados. Este método es más lento y poco amigable porque se requiere realizar muchos pasos para su operación y que el usuario recuerde cuál es el objeto que se debe modificar.</p>	<p>Para modificar las propiedades del objeto, solo es necesario hacer clic derecho con el mouse sobre el objeto en el panel de animación y seleccionar la opción “Modificar Objeto” y luego modificar los valores deseados. Este método permite de una manera más sencilla, rápida y amigable encontrar un objeto que se desea modificar.</p>
<p>Para eliminar un objeto es necesario ir al menú, seleccionar la opción “Eliminar Objeto” y luego seleccionar el objeto. Este método es más lento y poco amigable porque se requiere realizar muchas pasos para su operación y que el usuario recuerde cuál es el objeto que se debe eliminar.</p>	<p>Para eliminar un objeto, solo es necesario hacer un clic derecho con el mouse sobre el objeto en el panel de animación y seleccionar la opción “Eliminar Objeto”. Este método permite de una manera más sencilla, rápida y amigable encontrar un objeto que se desea eliminar.</p>
<p>No existe la opción “Deshabilitar”.</p>	<p>Para habilitar un objeto de la simulación, solo es necesario buscar en el panel de información de la simulación, el objeto que se deshabilitó y presionar con el mouse sobre el objeto, arrastrarlo y soltarlo sobre el panel de animación y el objeto se colocará en su posición inicial.</p>

<u>DIFERENCIAS FUNCIONALES</u>	
<u>TESIS ANTERIOR</u>	<u>PROYECTO DE FIN DE CARRERA</u>
No cuenta con esta operación.	Cada reporte gráfico tiene una opción que permite guardar el reporte como una imagen, además de seleccionar la ruta de destino.
No cuenta con esta operación.	Cada reporte gráfico tiene una opción que permite enviar por mail la imagen del diagrama y añadir otros archivos más. La imagen y los otros archivos son enviados como archivos adjuntos hacia un destinatario.
No cuenta con esta operación.	La aplicación tiene una opción que permite exportar a un archivo Excel la lista de valores obtenidos en la simulación: velocidad vs tiempo, posición vs tiempo y fuerza vs tiempo.
No cuenta con esta operación.	La aplicación tiene una opción que permite exportar a un archivo PDF la lista de valores obtenidos en la simulación: velocidad vs tiempo, posición vs tiempo y fuerza versus tiempo.

Tabla 1.9: Cuadro de Diferencias Funcionales

1.5. Descripción y Sustentación de la Solución

Un sistema de simulación educativo es un entorno de aprendizaje que permite la elaboración de una actividad educativa, además de facilitar la enseñanza.

Un sistema de simulación educativo utiliza los modelos matemáticos para representar y reproducir los fenómenos físicos. Los modelos matemáticos de la Gravitación Universal permiten sustituir los experimentos hechos en los laboratorios por una simulación de ambiente computarizado.

Otra de las principales ventajas de un sistema de simulación educativo es que se enfatiza a que el alumno pueda aprender de cada experimento físico. Esto permite que el alumno pueda entender con mayor facilidad el tema del experimento.

El sistema de simulación educativo elimina los problemas mencionados en la definición del problema.

Adicionalmente el sistema provee las siguientes características:

- El alumno puede crear su propio experimento para saber el por qué de un suceso. De esta manera, no se limita las capacidades en el análisis, así como la imaginación y la creatividad del alumno.
- El sistema permite que el alumno pueda construir un ambiente de experimentación física de manera simulada, en la que pueda ver, experimentar y analizar. Además permite la reflexión del alumno en el cómo se relacionan o cómo se comportan los cuerpos frente a otros, en relación a la teoría de Gravitación Universal. Adicionalmente el sistema contiene ejemplos didácticos que permiten que el alumno los utilice para visualizarlos y ejecutarlos.
- El alumno podrá crear, modificar y eliminar los objetos de la simulación y sus propiedades, además de seleccionarlos e incluirlos en el ambiente de simulación y ver sus características físicas durante el experimento físico.
- El alumno puede ver los resultados obtenidos por la simulación hecha en el sistema, con distintas posibilidades de visualizarlos, según el criterio del alumno tenga para realizar análisis.
- El sistema permite realizar gráficos que muestran los resultados del experimento simulado de forma automática.
- El usuario no pierde la experiencia realizada, debido a que el sistema permite guardar y abrir cualquier experimento físico para un posterior análisis. El sistema cuenta con una adecuada base de datos que permite almacenar los datos de los experimentos contruidos por el alumno de una forma consistente.

2. Análisis

El presente capítulo describe la metodología utilizada para el desarrollo de la solución, la identificación de los requerimientos, el análisis de la solución, la definición del sistema, la descripción de los casos de uso y el diagrama de clases.

2.1 Metodología Utilizada para la Solución

La metodología utilizada en el proyecto de fin de carrera es el Proceso Unificado Ágil (AUP). Esta metodología es una versión simplificada del Rational Unified Process (RUP), es decir, es una metodología ágil.

Esto permite desarrollar proyectos de una manera más simple y fácil de entender para el desarrollo de software de aplicación, utilizando las técnicas y conceptos que aún permanecen fieles al RUP. La metodología AUP trata de mantener la simplicidad en el enfoque y en su descripción. Ver en [Chin 2004].

Justificación:

La metodología permite que el desarrollo del proyecto de fin de carrera sea apto a los requerimientos de los usuarios. Este tipo de metodología enfatiza más en la comunicación que en la documentación. La comunicación entre el programador y los usuarios de la aplicación permite mostrar el proyecto a tiempo para poder realizar cambios. Se minimizan los riesgos ya que el desarrollo del software se realiza en cortos lapsos de tiempo.

A continuación se describe las fases, las iteraciones y los entregables de la metodología AUP aplicada al proyecto de fin de carrera.

1era Fase - Inicio:

En esta fase el objetivo es identificar el alcance inicial del simulador, su arquitectura y aceptación por parte de los usuarios.

- Documento de Alcance del Proyecto. Este documento permite saber cuáles son los requerimientos de los usuarios, en este caso son los profesores de física y los alumnos.
- Documento de Estimación de Costos y Planificación. Este documento permite calcular el costo y el tiempo que dura el proyecto.
- Documento de Riesgos. Este documento permite saber si el riesgo asociado al proyecto está dentro del alcance permitido.
- Documento de Viabilidad. Este documento permite saber si el costo del proyecto y la tecnología disponible cumple con los requerimientos de los usuarios en el tiempo establecido.

2da Fase – Elaboración:

En esta fase el objetivo es mostrar la arquitectura del sistema. Un conjunto de patrones que proporcionan un marco de referencia de lo necesario para la construcción del sistema.

- Seleccionar el uso de la tecnología, herramientas y recursos. En el proyecto de fin de carrera se requiere utilizar una tecnología multiplataforma y que sea en entorno Web, por lo tanto se debe contar con un servidor Web.
- Plantear el prototipo de arquitectura. La arquitectura del proyecto de fin de carrera se divide en tres capas (capa de presentación, capa de lógica de negocio y capa de acceso a base de datos) y una base de datos. Se cuenta con un servidor que almacena la aplicación Web y la base de datos.
- Calcular el trabajo en corto tiempo. Debido a que los requerimientos de la aplicación no son muy complejos, no toma mucho tiempo planificar el tiempo de construcción de la aplicación.

3ra Fase – Construcción:

En esta fase el objetivo es construir el software trabajando en forma iterativa e incremental, de acuerdo a los requerimientos del usuario.

En esta fase se asignan las tareas de programación y la organización de la entrega. Para el proyecto de fin de carrera se cuenta con una sola persona que se encarga de las tareas de análisis y programación. El desarrollo de la aplicación debe ser simple, funcional y estandarizado. Además de la construcción se debe realizar la validación del sistema, es decir, se desarrolla las pruebas a la aplicación. Debido a que las funcionalidades de la aplicación no son demasiado extensas, las pruebas se realizan para las funcionalidades generales. El cambio de estructura de un código no tiene por qué cambiar su funcionamiento.

Para una ordenada construcción del proyecto, se identifican los siguientes casos de uso:

- Administrar Simulación.
- Asignar Objetos.
- Cargar Simulación.
- Ejecutar Simulación.
- Establecer Efectos Visuales.
- Generar reportes.

Tal como se muestra en la figura 2.1, el proyecto de fin de carrera se divide en 6 casos de uso y se identifica como usuario de la aplicación al estudiante y el profesor.

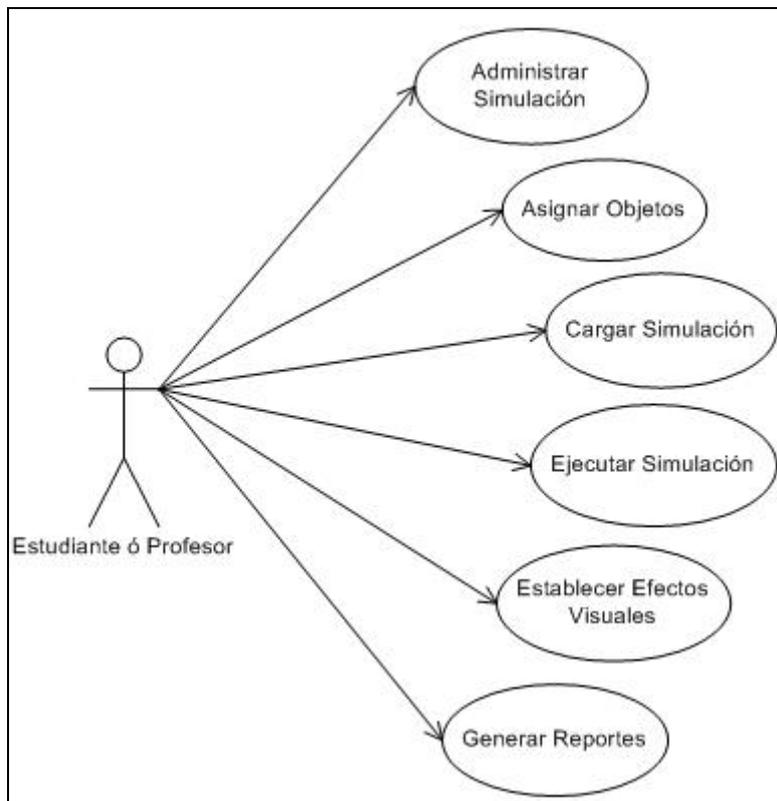


Figura 2.1: Diagrama de Casos de Uso.

La entrega del proceso de construcción del proyecto, se realiza a través de iteraciones, las cuales son:

1era iteración:

- Realizar los mantenimientos de las simulaciones y de los objetos. Los casos de uso involucrados son: Administrar Simulaciones y Asignar Objetos
- Realizar pruebas al mantenimiento de la simulación y los objetos, tales como evitar números alfabéticos o negativos en algunos campos.

2da iteración:

- Generar una simulación a partir de dos o más objetos creados, en sus tres vistas. Los casos de uso involucrados son: Cargar Simulación y Ejecutar Simulación.

- Realizar pruebas a la ejecución de la simulación, entre las principales pruebas se encuentran la ubicación de un objeto y evitar el ingreso de datos durante la simulación.

3ra iteración:

- Realizar los reportes de acuerdo a la simulación realizada. El caso de uso involucrado es: Generar Reportes.
- Establecer los efectos visuales en la simulación, tales como el acercamiento, el alejamiento, el aumento y disminución de la velocidad. El caso de uso involucrado es: Establecer Efectos Visuales.
- Realizar pruebas a los reportes del simulador, tales como validar si la simulación realizada esta acorde con el reporte mostrado.
- Realizar pruebas a los efectos visuales al simulador, tales como validar la velocidad máxima y mínima en la simulación.

2.2 Identificación de Requerimientos

El software del proyecto de fin de carrera cubre los requerimientos según la identificación del problema.

El sistema permitirá crear, almacenar o abrir una simulación realizada por el alumno, de esta manera el alumno puede crear y grabar el experimento físico en el momento que éste lo desee para analizarlo después.

El almacenamiento de la simulación permite abrirlo posteriormente y el sistema permite administrar las propiedades de la simulación.

Además el sistema cuenta con ejemplos explicativos que le permitan familiarizarse con su uso para el desarrollo de sus propios experimentos.

El manejo de creación de un modelo de escenario de simulación es fácil e intuitivo. El simulador contiene tres vistas: superior, frontal y lateral. Las vistas tridimensionales se utilizan para una mejor apreciación de la simulación.

El sistema cuenta con un panel de información que muestra los datos de los objetos que están involucrados durante el proceso de simulación, dicha información va cambiando con respecto al tiempo de la ejecución.

El sistema permite seleccionar, arrastrar y soltar los objetos con el mouse, para realizar diferentes operaciones. Los objetos se pueden arrastrar y soltar antes, durante y al finalizar la ejecución.

El alumno ubica el objeto en la posición que considera necesaria para el experimento. Esto permite mayor interacción entre el simulador y el alumno para analizar mejor los experimentos físicos y observar los resultados obtenidos.

Para realizar el proceso de simulación, el sistema debe contar con botones para ejecutar, pausar y detener el proceso.

El sistema contiene un menú, el cual permite acceder a la administración de las simulaciones, la ejecución, los efectos visuales, la administración de los objetos y mostrar los reportes del sistema.

El sistema contiene una barra de tareas. La cual contiene iconos y botones de las principales funcionalidades de la aplicación y permite un acceso más rápido a la ejecución de estas funcionalidades.

Mientras se realiza la ejecución de la simulación, el sistema muestra una barra de progreso. Este control permite visualizar el progreso de la ejecución del experimento físico.

El sistema permite realizar gráficos lineales de los objetos involucrados en la simulación. Los gráficos lineales que muestra el sistema son velocidad versus tiempo, posición versus tiempo y fuerza versus tiempo. El alumno puede seleccionar cuales son los objetos que estarán en el diagrama. El sistema muestra los resultados de la simulación como gráfico lineal en un recuadro, dicho gráfico se puede imprimir, guardar como imagen o enviarlo vía correo electrónico.

El sistema es una aplicación Web. De esta manera, no hay necesidad de descargar ni instalar ningún programa, solo basta tener un navegador Web actualizado, no ocupa mucho espacio en disco duro, la conexión en la Web permite obtener la última versión de lo publicado, el consumo de recursos es bajo, es multiplataforma, la disponibilidad a la aplicación es alta, tiene un acceso sencillo y permite compartir datos entre varios usuarios.

El sistema utiliza una base de datos. El uso de esta tecnología permite obtener muchas ventajas, tales como: compartir información entre varios usuarios; mejorar el acceso y la seguridad de los datos; mantiene la integridad, el control de redundancia y la consistencia de los datos; permite la independencia de los datos y administra la concurrencia, además de realizar copias de seguridad.

No es necesario crear un nuevo escenario de simulación para un experimento, solo es necesario variar algunas características de la simulación para tener un escenario diferente.

Durante el proceso de ejecución de la simulación algunas funcionalidades están deshabilitadas. Estas restricciones evitan causar confusión en el alumno durante el proceso de simulación.

2.3 Análisis de la Solución

La presente sección describe la viabilidad del sistema, el análisis técnico y económico y la asignación de funciones.

2.3.1 Viabilidad del Sistema

La viabilidad del sistema se sustenta en atender los requerimientos de interacción entre los actores del proceso de enseñanza y aprendizaje: educador, alumno, conocimiento y computador.

Identificación de los usuarios:

La presente sección identifica los usuarios de la aplicación.

- Los usuarios identificados son los profesores y los alumnos de centros educativos estatales o particulares de nivel escolar secundario o pre - universitario.

- El proyecto está dirigido a los colegios o centros educativos, ya sean estatales o particulares. Los centros educativos cuentan, en su mayoría, con un laboratorio de computadoras, esta infraestructura permite que el simulador de física sea utilizado de manera exitosa.
- El dominio de física, es un campo obligatorio para la educación secundaria y pre-universitaria, esto permite que la difusión del proyecto sea más amplio.

2.3.2 Análisis Técnico y Económico

Aspecto Técnico:

En el aspecto técnico, el desarrollo y el funcionamiento del proyecto se realizarán a través de tecnologías seguras, confiables y compatibles con las máquinas o computadoras que se tienen en un centro educativo.

Las características mínimas que deben tener estas maquinas son:

- Software mínimo: Como sistema operativo se puede utilizar: Windows 2000 o superior, Mac OS X 10.4 o superior ó Linux. Cada sistema operativo debe tener un navegador Web (Recomendado: Firefox 4.0.1 o superior).
- Hardware mínimo: 128 MB RAM (Recomendado: 256 MB RAM o más) y 200 MB de espacio en disco.

Aspecto Económico:

Para el desarrollo del proyecto no se produjeron desembolsos respecto al tema de recursos humanos debido a que el desarrollo del proyecto es responsabilidad total del tesista.

Las herramientas que se utilizarán para su construcción o desarrollo del proyecto no tienen restricción de costos debido a que son herramientas de uso libre.

Se cuenta con el apoyo de la universidad para el uso de diversas tecnologías tales como herramientas de programación, base de datos, programas de diseño, etc. La tabla 2.1 muestra un estimado de los costos del proyecto.

<u>COSTO PERSONAL</u>	
Horas trabajadas aproximadamente	700 horas planificadas.
Días trabajados aproximadamente (1 día tiene 8 horas de trabajo).	87.5 días
Costo por hora de trabajo	S/. 25
Costo por día de trabajo (8 horas):	S/. 200
Costo Total	S/. 17 500

Tabla 2.1: Análisis de la Solución, Costos del Proyecto

2.3.3 Asignar Funciones

Hardware:

El hardware utilizado debe dar soporte y permitir el funcionamiento al sistema operativo, así como a los requerimientos del simulador. Durante el funcionamiento del sistema, el hardware no debe tener fallas. El tiempo de funcionamiento debe ser durante los días en que se utilice el laboratorio de física o al menos durante las horas que dure la sesión en el laboratorio. El alumno podrá interactuar con el sistema a través del teclado y el mouse.

Usuarios:

El profesor o encargado de la capacitación podrá desarrollar ejemplos con el sistema, para que los alumnos entiendan su funcionamiento, por ejemplo, las propiedades de la simulación, el mantenimiento de los cuerpos físicos y los reportes. Con la explicación hecha por el profesor o encargado de la capacitación, cada alumno podrá desarrollar su propio experimento físico a través del sistema. La capacitación de los profesores en el uso del sistema es sencilla, debido a que el sistema cuenta con una interfaz amigable que permite fácilmente interactuar con el alumno.

A la base de datos:

La base de datos utilizada debe dar soporte y permitir el funcionamiento al sistema operativo, así como a los requerimientos del simulador. Durante el funcionamiento del sistema, la base de datos no debe tener fallas. La base de datos debe permitir un almacenamiento continuo de información. Este proceso debe ser durante los días en que se utilice el laboratorio de física o al menos durante las horas que dure la sesión en el laboratorio.

2.4 Definición del Sistema

El sistema se define como un sistema de simulación educativo que permite simular experimentos físicos, en este caso en el tema de la Gravitación Universal.

El sistema permite que el alumno pueda simular sus propios ambientes de experimentación física incorporando en ellos cuerpos en que pueda analizar y experimentar, de esta manera se puede observar cómo se relacionan o se comportan los cuerpos frente a otros con relación a la teoría de Gravitación Universal.

2.5 Descripción de los Casos de Uso

La presente sección describe el diagrama de casos de uso. Tal como se muestra en la figura 2.1.

- Administrar Simulación: El propósito de este caso de uso es registrar, modificar y eliminar un proyecto de simulación. Este caso de uso soluciona el problema que el alumno tenga la necesidad de ir a un laboratorio para realizar el experimento físico. Además el sistema le permite visualizar los experimentos de Gravitación Universal.
- Asignar Objetos: El propósito de este caso de uso es registrar, modificar y eliminar los objetos de una simulación, en este caso de uso se incluye la administración de los objetos que se encuentran en dicha simulación. El alumno podrá colocar o quitar un objeto de la simulación de forma rápida y sencilla. Facilitando el proceso de realización del experimento e ir de una manera directa a su análisis.

- Cargar Simulación: El propósito de este caso de uso es cargar un escenario de la simulación en el sistema. De esta manera, permite administrar las simulaciones que el alumno haya guardado, para realizar su posterior análisis y evitando de esta manera volver a realizar el experimento o modificar el experimento simulado hecho anteriormente.
- Ejecutar Simulación: El propósito de este caso de uso es ejecutar un escenario con diversas opciones durante su ejecución. Este caso de uso permite ejecutar, pausar o detener la simulación en el momento que el alumno lo desee. De esta manera el alumno puede analizar detenidamente el proceso del experimento, evitando volver a realizarlo, rearmando las piezas ó los materiales u cualquier otro objeto involucrado en el experimento, tal como ocurre en los laboratorios del dominio de física.
- Establecer Efectos Visuales: El propósito de este caso de uso es establecer efectos visuales en la simulación de acuerdo a los requerimientos del usuario. Este caso de uso permite que la simulación sea cercana a la realidad, de esta manera su objetivo es el realismo de la simulación del experimento, como si fuera en un laboratorio de un centro educativo.
- Generar Reportes: El propósito de este caso de uso es generar reportes de las simulaciones: velocidad versus tiempo, fuerza versus tiempo y posición versus tiempo. De esta manera el alumno podrá visualizar de diversas maneras la interacción entre los cuerpos frente a la Gravitación Universal, de forma gráfica ó en tabulaciones, facilitando el proceso de aprendizaje o análisis del tema en el dominio de física.

2.5.1 Especificación de Requerimientos de Software (ERS)

La presente sección describe el caso de uso “Administrar Simulación”. Este caso de uso tiene como flujo principal el registrar una simulación, además tiene como flujos alternativos el consultar, modificar y eliminar una simulación y como flujos excepcionales tiene el salir, los datos incorrectos y los datos obligatorios.

Caso de uso: Administrar Simulación	
Descripción:	La finalidad de este caso de uso es realizar una simulación de experimento físico en relación a la Gravitación Universal.

Actores:	Alumno ó Profesor.
Precondición:	No existe precondición.
Flujo Principal: Registrar Simulación	
<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el usuario elige la opción “Simulación”. 2. El usuario elige la opción “Construcción”. 3. El sistema muestra un formulario con las opciones que involucran a la simulación del experimento: “Registrar”, “Modificar”, “Eliminar”, “Cargar” y “Salir”. 4. El usuario selecciona la opción “Registrar”. 5. El sistema muestra un formulario con los siguientes campos: “Título”, “Intervalo de Tiempo (dt)”, “Tiempo de evaluación”, “Número máximo de iteraciones”, “Vista Superior”, “Vista Frontal”, “Vista Lateral”, “Mostrar nombres de los cuerpos” y “Límite del área de visión”. 6. El usuario ingresa los datos solicitados. 7. El usuario selecciona la opción “Guardar”. 8. El sistema muestra un mensaje si desea realizar la operación. 9. El usuario acepta la operación. 10. El sistema verifica la información. 11. El sistema crea un nuevo registro de simulación y guarda su información. 12. El sistema muestra mensaje de éxito. 13. Los pasos 3 al 12 se repiten para cada simulación que se desee registrar. 14. El caso de uso finaliza cuando el usuario elige la opción “Salir”. 	
Post condición:	Se ha registrado los datos de la simulación exitosamente.
Flujo Alternativo 1: Consultar Simulación	
<ol style="list-style-type: none"> 1. El sistema muestra un formulario con los siguientes criterios de búsqueda: nombre de la simulación. 2. El usuario ingresa las opciones de búsqueda que desea y selecciona la opción “Buscar”. 3. El sistema muestra un listado de los registros que coinciden con los criterios indicados en el paso anterior. 	
Flujo Alternativo 2: Modificar Simulación	
<ol style="list-style-type: none"> 1. Desde el flujo alternativo Consultar Simulación, el usuario selecciona uno de los registros del listado mostrado. 2. El usuario selecciona la opción “Modificar”. 	

3. El sistema muestra un formulario con los siguientes campos: “Título”, “Intervalo de Tiempo (dt)”, “Tiempo de evaluación”, “Número máximo de iteraciones”, “Vista Superior”, “Vista Frontal”, “Vista Lateral”, “Mostrar nombres de los cuerpos” y “Límite del área de visión”.
4. El sistema carga los datos de la simulación seleccionada.
5. El usuario ingresa los datos a modificar.
6. El usuario selecciona la opción “Guardar”.
7. El sistema muestra un mensaje si desea realizar la operación.
8. El usuario acepta la operación.
9. El sistema verifica la información.
10. El sistema modifica el registro de la simulación y guarda su información.
11. El sistema muestra mensaje de éxito.
12. Los pasos 1 al 11 se repiten para cada simulación que se desee modificar.
13. El caso de uso finaliza cuando el usuario elige la opción “Salir”.

Flujo Alternativo 3: Eliminar Simulación

1. Desde el flujo alternativo Consultar Simulación, el usuario selecciona uno de los registros del listado mostrado.
2. El usuario selecciona la opción “Eliminar”.
3. El sistema muestra un mensaje si desea realizar la operación.
4. El usuario acepta la operación.
5. El sistema verifica la información.
6. El sistema elimina registro de simulación.
7. El sistema muestra mensaje de éxito.
8. Los pasos 1 al 7 se repiten para cada simulación que se desee eliminar.
9. El caso de uso finaliza cuando el usuario elige la opción “Salir”.

Flujo Excepcional 1: Salir

1. En cualquier parte del flujo principal puede seleccionar “Salir”.
2. El sistema muestra un mensaje de verificación.
 - a. Si el usuario selecciona “Sí” el caso de uso termina.
 - b. Si el usuario selecciona “No” el caso de uso continúa donde se quedó.

Flujo Excepcional 2: Datos Incorrectos

1. Si el sistema detecta que los datos ingresados son incorrectos, muestra un mensaje de error.
2. Una vez que el usuario corrija los datos, el caso de uso continúa con el paso 6 del flujo principal.

Flujo Excepcional 3: Datos Obligatorios

1. Si el sistema detecta que no se han ingresado todos los campos obligatorios del formulario, se muestra el mensaje "Falta completar algunos campos".
2. Una vez que el usuario ingrese los valores en los campos faltantes, el caso de uso continúa con el paso 6 del flujo principal.

2.6 Diagrama de Clases (Análisis)

La presente sección muestra y describe el diagrama de clases de análisis. Tal como se muestra en la figura 2.2, el sistema cuenta con cuatro clases principales que se utilizan para el proceso de simulación.

- Clase Vista: Contiene la información de la vista de la simulación.
- Clase Simulación: Contiene la información propia de la simulación.
- Clase Objeto: Contiene la información propia del objeto que se encuentra en la simulación.
- Clase Vector: Contiene la información de los vectores del objeto (posición y velocidad) con respecto al eje de coordenadas. Eje "X", Eje "Y" y Eje "Z".

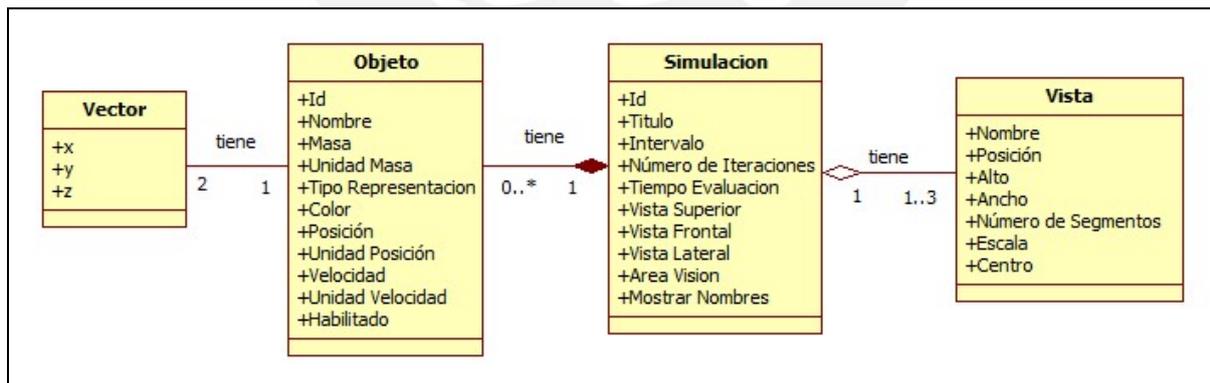


Figura 2.2: Diagrama de Clases de Análisis

3. Diseño

El presente capítulo describe la arquitectura de la solución y el diseño de la interfaz gráfica.

3.1 Arquitectura de la Solución

La presente sección muestra la descripción de la arquitectura, así como los diagramas de arquitectura, de paquetes, de clases de diseño, de componentes, de despliegue, de secuencia, además del modelo de base de datos y finalmente el diccionario de datos.

3.1.1 Descripción

Arquitectura de 3 capas: Las aplicaciones compactas causan gran cantidad de problemas de integración en sistemas de software complejos. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad e integración. Para solucionar estos problemas, la arquitectura de la aplicación se divide en tres capas:

- Una capa de acceso a base de datos. Para acceder a la base de datos.
- Una capa para centralizar la lógica de negocio (modelo).

- Una capa de presentación ó interfaz gráfica. La cual facilita al usuario el uso del sistema.

La separación entre las capas permite obtener una arquitectura con ciertas ventajas como:

- Centralización de las transacciones, las cuales serían responsabilidad del modelo.
- Centralización de las conexiones a base de datos.
- Reutilización de las capas.
- Facilidad en la estandarización. Ver en [Sommerville 2006].

Justificación:

La arquitectura de tres capas permite que la capa de interfaz gráfica del usuario o del estudiante, carezca de toda lógica del negocio, en este caso de las funcionalidades de la simulación. La capa de interfaz gráfica, solo ofrece la funcionalidad de visión y de petición de los datos para generar la simulación, esto permite que el proceso que genera la simulación se encuentre en la capa de lógica de negocio. La capa de acceso a base de datos se encarga de realizar el proceso de almacenamiento de las propiedades de la simulación y sus objetos en la base de datos. Ver en [Sommerville 2006].

El uso de una base de datos en el proyecto de fin de carrera ofrece el almacenamiento de los datos de la simulación. Esto permite realizar cualquier tipo de consulta sobre sus datos. Además separa la descripción de los datos y la aplicación, lo que simplifica el mantenimiento de la aplicación.

Se eligió utilizar una base de datos, el cual es una librería. La librería es un solo archivo binario en el servidor que utiliza la funcionalidad de llamadas simples a subrutinas y funciones. Las llamadas a funciones son más eficientes que la comunicación entre procesos de una base de datos y la aplicación. Mantiene los datos consistentes ya que cuando se realiza una transacción se bloquea todo el archivo de base de datos. Además para proteger los datos del archivo se puede realizar copias de seguridad para luego restaurarlas.

3.1.2 Diagrama de Arquitectura

La presente sección muestra el diagrama de arquitectura del proyecto de fin de carrera, tal como se muestra en la figura 3.1.

El servidor almacena la base de datos y la aplicación Web. La aplicación Web y la base de datos se comunican a través de las librerías de conexión proporcionadas por el proveedor de base de datos y referenciadas por la aplicación.

La aplicación está dividida en tres capas: Capa de interfaz gráfica, capa de lógica de negocio y capa de acceso a base de datos. Existe un conjunto de clases como entidades de negocio. Este componente agrupa clases que permiten contener datos y operaciones de las entidades de la aplicación, tales como la simulación, el objeto, el vector y la vista.

La capa de interfaz gráfica, de lógica de negocio y de acceso a base de datos referencian al componente de entidades, ya que este componente permite almacenar información de las entidades de la aplicación, de esta manera a través de ella, se puede transportar la información entre las tres capas.

Las capas se comunican a través de referencias a los paquetes. Cada paquete representa una capa lógica y agrupa a las clases. Cada clase contiene atributos y métodos que permite dar funcionalidad a cada uno de las capas de la aplicación.

Para acceder a la aplicación Web, el cliente debe contar con un navegador Web con acceso a internet para poder visualizar la aplicación. La comunicación se realiza a través del protocolo http.

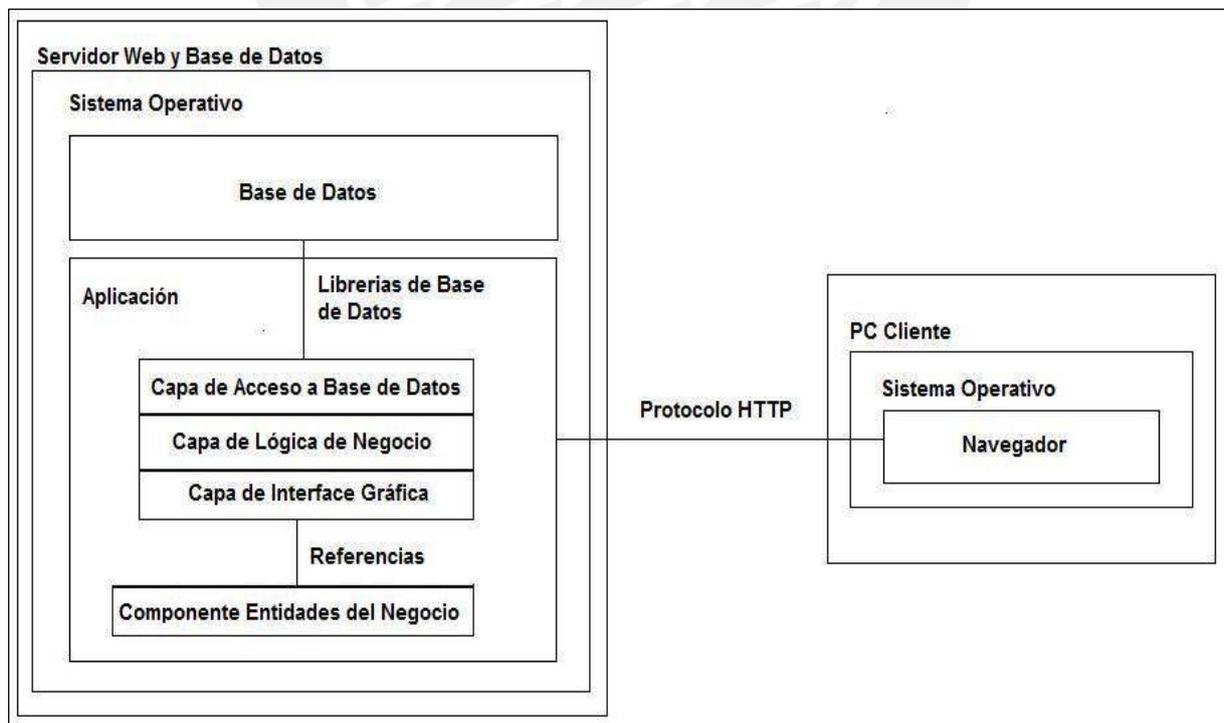


Figura 3.1: Diagrama de Arquitectura.

3.1.3 Diagrama de Paquetes

La presente sección muestra y describe el diagrama de paquetes, tal como se muestra en la figura 3.2. El sistema cuenta con cuatro paquetes que almacenan las clases que se utilizan para el proceso de simulación.

- Paquete de Entidades de Negocio: Es el paquete que agrupa las clases que representan las entidades del negocio en el sistema. En el paquete se encuentran la clase vector, la clase vista, la clase simulación y la clase objeto.
- Paquete de Acceso a Base de Datos: Es el paquete que agrupa las clases que representan el acceso a la base de datos en el sistema. En el paquete se encuentran la clase conexión, la clase de acceso a base de datos de la simulación y del objeto.
- Paquete de Lógica de Negocios: Es el paquete que agrupa las clases que representa la lógica de negocio en el sistema. En el paquete se encuentran la clase de lógica de negocio de la simulación, del objeto y de utilitarios, la cual permite realizar conversiones y obtener valores constantes.
- Paquete de Interfaz Gráfica: Es el paquete que agrupa las clases que representan la interfaz gráfica en el sistema. Entre las principales clases se encuentran la clase Applet que contiene las funcionalidades del applet, la clase Buscar Simulación que contiene las funcionalidades para administrar los datos de la simulación y la clase Canvas que permite dibujar y generar el movimiento de los objetos de la simulación.

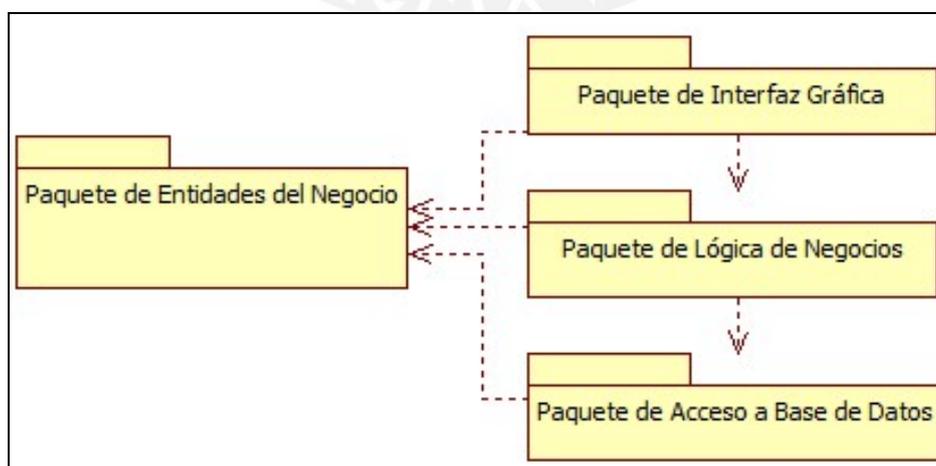


Figura 3.2: Diagrama de Paquetes

3.1.4 Diagrama de Clases de Diseño

La presente sección muestra y describe el diagrama de las clases de diseño que se utilizan en la capa de interfaz gráfica, de lógica de negocios y de acceso a base de datos del proyecto de fin de carrera. Además del componente de entidades del negocio.

Capa de Interfaz Gráfica:

La presente sección muestra y describe el diagrama de las clases de diseño que se utilizan en la capa de interfaz gráfica. Tal como se muestran en las figuras 3.3 y 3.4. A continuación se describen las clases de diseño de la capa de interfaz gráfica.

- Applet: Es la interfaz que permite administrar las funcionalidades del simulador y muestra la simulación del experimento físico.
- Panel del Objeto: Es el componente que permite visualizar la información del objeto durante la ejecución de la simulación.
- Canvas: Es el componente que permite dibujar las vistas y los objetos de la simulación.
- Mouse Listener Panel Objeto: Es el componente que permite administrar las funcionalidades del mouse en el panel de información del objeto del applet.
- Mouse Listener Canvas: Es el componente que permite administrar las funcionalidades del mouse en el canvas del applet.
- Thread Applet: Es el componente que permite realizar una ejecución simultánea de varios objetos en el applet.
- Menú Popup Applet: Es el componente que permite visualizar un menú de acciones sobre el objeto con los eventos del mouse.
- Buscar Simulación: Es la interfaz que permite la administración de las simulaciones. Contiene las opciones de consultar, registrar, modificar, eliminar y cargar una simulación.

- Mantener Simulación: Es la interfaz que permite el registro y la modificación de una simulación.
- Mantener Objeto: Es la interfaz que permite el registro y la modificación de un objeto.
- Eliminar Objeto: Es la interfaz que permite la eliminación de un objeto.
- Panel de Representación: Es el componente que permite visualizar la representación del objeto.
- Reporte Fuerza: Es la interfaz que permite visualizar el reporte de las fuerzas que actúan sobre el objeto con respecto al tiempo.
- Reporte Posición: Es la interfaz que permite visualizar el reporte de las posiciones de un objeto con respecto al tiempo.
- Reporte Velocidad: Es la interfaz que permite visualizar el reporte de las velocidades de un objeto con respecto al tiempo.
- Mail: Es la interfaz que permite enviar un correo electrónico con el gráfico del reporte y otros archivos.
- Acerca De: Es la interfaz que permite visualizar el nombre del producto y el autor.

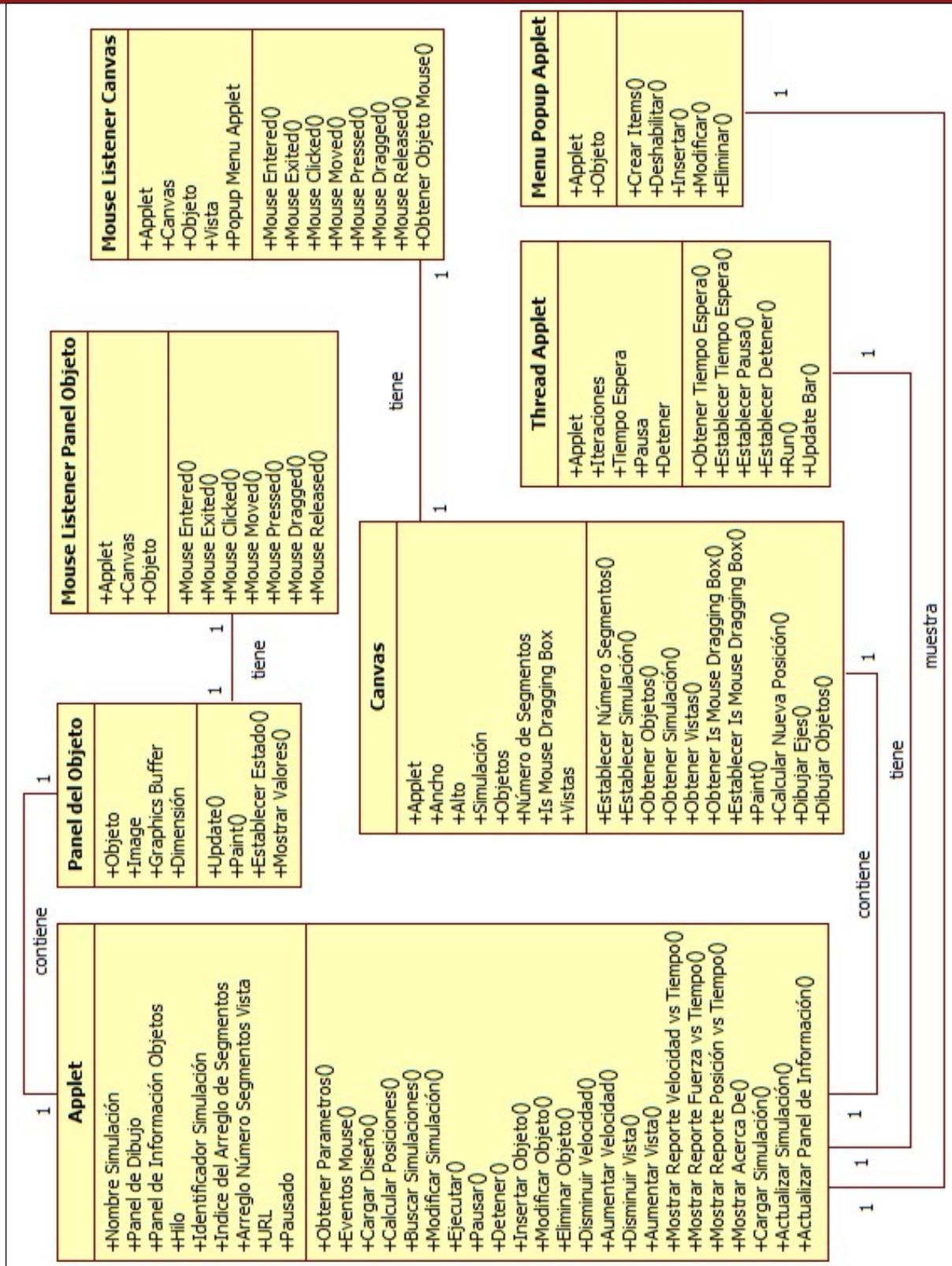


Figura 3.3: Diagrama de Clases de Diseño de la Capa de Interfaz 1.

Capa de Lógica de Negocio:

La presente sección muestra y describe el diagrama de las clases de diseño que se utilizan en la capa de lógica de negocios. Tal como se muestra en la figura 3.5. A continuación se describen las clases de diseño de la capa de lógica de negocios.

- BLSimulacion: Es la clase que permite la administración de las simulaciones.
- BLObjeto: Es la clase que permite la administración de los objetos.
- BLUtil: Es la clase que permite dar funcionalidad general al sistema.

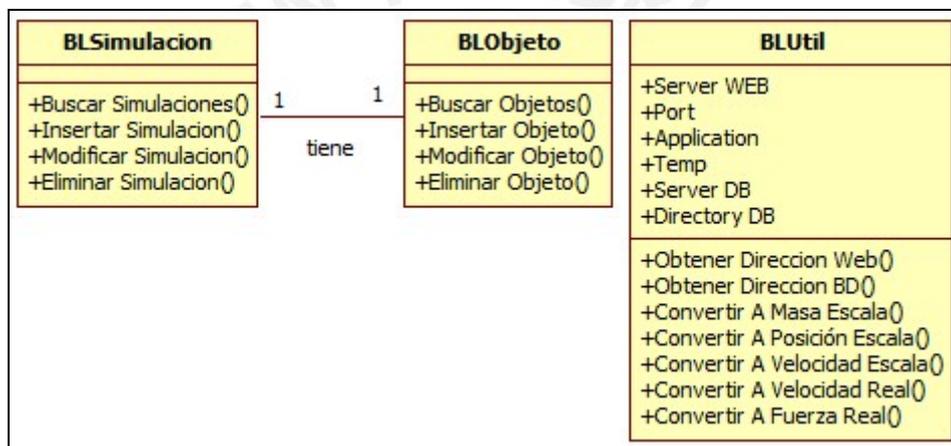


Figura 3.5: Diagrama de Clases de Diseño de la Capa de Lógica de Negocios.

Componente de Entidades de Negocio:

La presente sección muestra y describe el diagrama de las clases de diseño que se utilizan en el componente de entidades de negocio. Tal como se muestra en la figura 3.6. A continuación se describen las clases de diseño del componente de entidades de negocios.

- BESimulacion: Es la clase que representa y contiene la información de cada simulación.
- BEObjeto: Es la clase que representa y contiene la información de cada objeto de la simulación, además permite calcular su posición, su velocidad y las fuerzas que actúan sobre él.

- **BEVista:** Es la clase que representa y contiene la información de una vista.
- **BEVector:** Es la clase que representa y contiene la información de un vector.

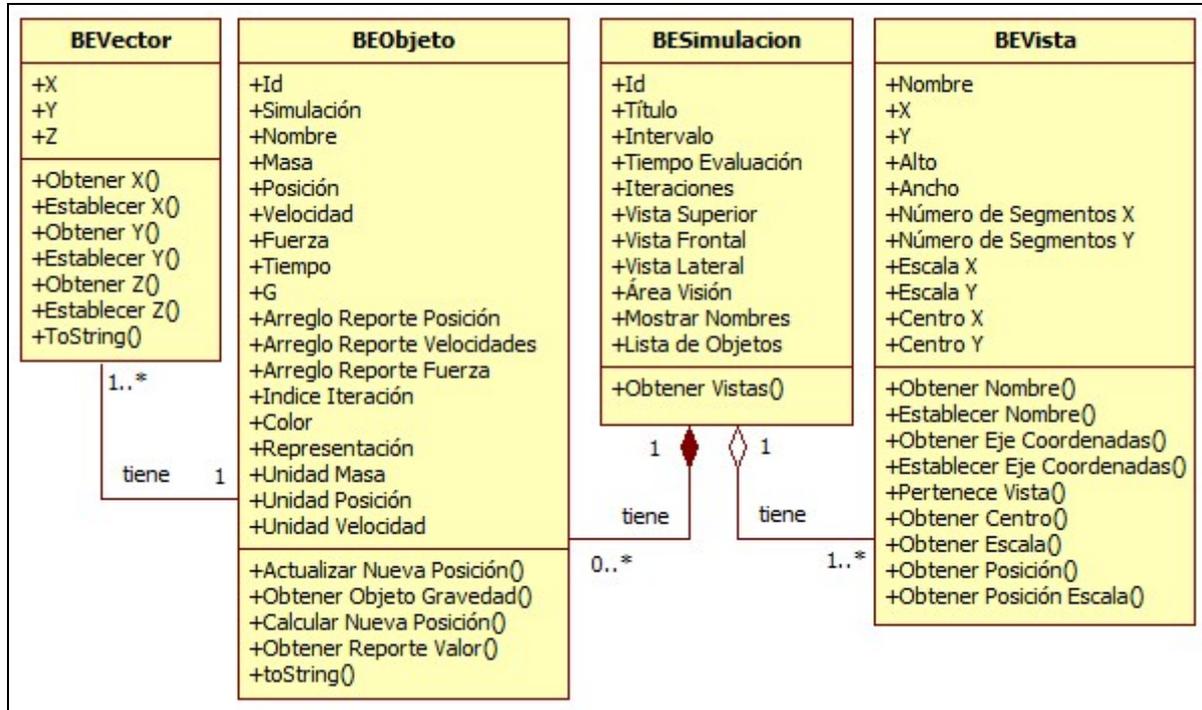


Figura 3.6: Diagrama de Clases de Diseño Componente de Entidades de Negocio.

Capa de Acceso a Base de Datos:

La presente sección muestra y describe el diagrama de las clases de diseño que se utilizan en la capa de acceso a base de datos. Tal como se muestra en la figura 3.7. A continuación se describen las clases de diseño de la capa de acceso a base de datos.

- **BDSimulación:** Es la clase que permite almacenar y obtener información de las simulaciones de la base de datos.
- **BDObjeto:** Es la clase que permite almacenar y obtener información de los objetos de la simulación de la base de datos.
- **BDConexion:** Es la clase que permite la conexión con la base de datos.



Figura 3.7: Diagrama de Clases de Diseño de la Capa de Acceso a Base de Datos.

3.1.5 Diagrama de Componentes

El presente diagrama describe cinco componentes que se utilizan en el proyecto de fin de carrera. Tal como se muestra en la figura 3.8.

- **Interfaz Gráfica:** Es el componente que describe las interfaces y los controles gráficos que el usuario, en este caso el alumno ó profesor, utiliza para realizar la simulación. Este componente depende de la lógica de negocio y las entidades del negocio. Se enlaza con la capa de presentación ó interfaz gráfica de la arquitectura de tres capas.
- **Lógica de Negocio:** Es el componente que describe los procesos que permiten la simulación del experimento físico. Este componente depende de las entidades del negocio y el acceso a base de datos. Se enlaza con la capa del modelo o lógica del negocio de la arquitectura de tres capas.
- **Entidades del Negocio:** Es el componente que describe las entidades que utiliza el sistema, tales como objeto, simulación, vector y vista. Este componente es referenciado por las tres capas de la arquitectura. Es un componente añadido a la arquitectura.
- **Acceso a Base de Datos:** Es el componente que describe los procesos de acceso a la base de datos. Este componente depende de la base de datos. Se enlaza con la capa de acceso a la base de datos de la arquitectura de tres capas.
- **Base de Datos:** Es el componente que describe los elementos físicos y lógicos de la base de datos. Este componente se encuentra en la base de datos y es proporcionada por el proveedor.

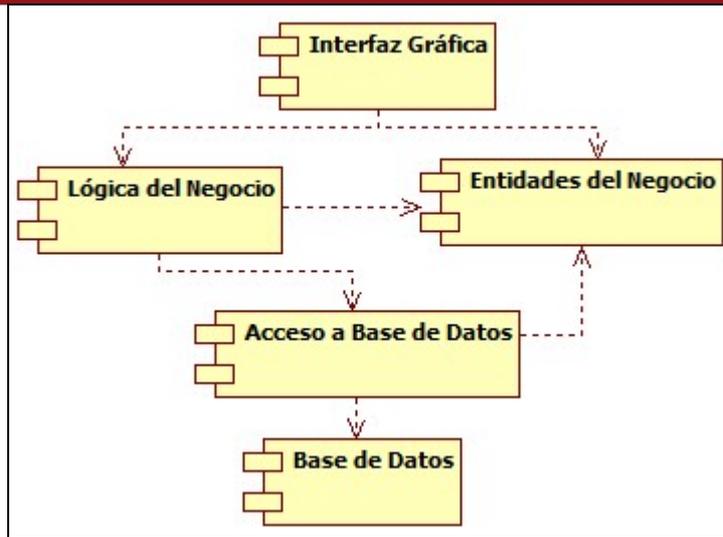


Figura 3.8: Diagrama de Componentes.

3.1.6 Diagrama de Despliegue

El presente diagrama describe los dos nodos que se utilizan en el proyecto de fin de carrera, tal como se muestra en la figura 3.9.

- PC Cliente: Es el nodo que representa el dispositivo físico que utiliza el alumno para la elaboración de la simulación, es decir la computadora del alumno.
- Servidor Web y de Base de Datos: Es el nodo que representa el dispositivo físico que permite el almacenamiento de la aplicación Web y de la base de datos.

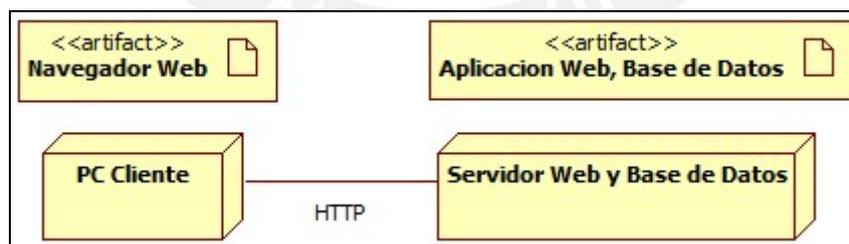


Figura 3.9: Diagrama de Despliegue.

3.1.7 Diagrama de Secuencia

La presente sección muestra el diagrama de interacción entre el usuario y la aplicación a través del tiempo. El diagrama de secuencia muestra el caso de uso “Generar Simulación” en el flujo principal “Registrar Simulación”.

Caso de Uso Generar Simulación

Tal como se muestra en la figura 3.10, el diagrama de secuencia permite observar cómo se registra una simulación.

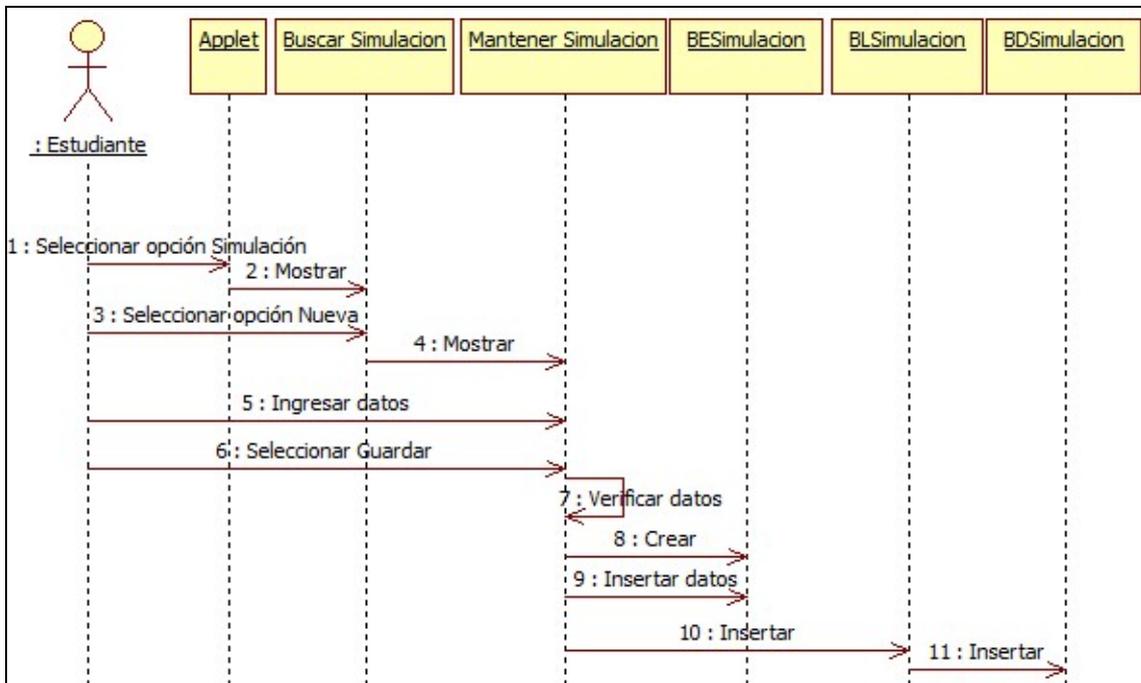


Figura 3.10: Diagrama de Secuencia – Generar Simulación.

3.1.8 Modelo de Base de Datos

La figura 3.11 muestra el modelo de la base de datos del proyecto de fin de carrera. Este presenta dos tablas:

- **Simulación:** Contiene la información de la simulación. Título, intervalo, número de iteraciones, tiempo de evaluación, vista superior, vista frontal, vista lateral, mostrar los nombres de los objetos y área de visión.
- **Objeto:** Contiene la información del objeto. Nombre, masa, posición, velocidad, representación, unidades de medición, composición del color, estado de habilitado y el identificador de la simulación a la que pertenece.

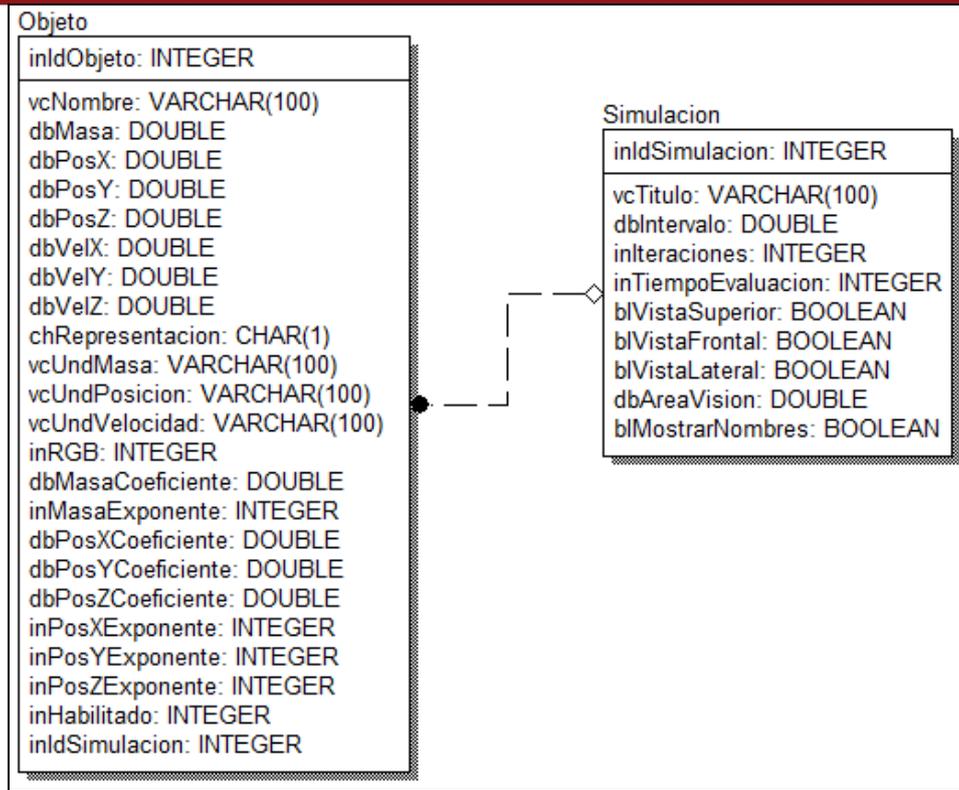


Figura 3.11: Diagrama de Base de Datos.

A pesar de que el sistema utiliza dos tablas, la justificación de utilizar una base de datos se encuentra detallada en el capítulo 4, en la sección 4.1.4.

3.1.9 Diccionario de Datos

La presente sección muestra las tablas utilizadas en el proyecto de fin de carrera.

- **Tabla Simulación:** Es la tabla que contiene la información que se necesita para generar la simulación. Esta tabla cuenta con los siguientes campos: Identificador, título, intervalo, tiempo de evaluación, número de iteraciones, las vistas que se utilizan, la opción si se requiere mostrar los nombres de los objetos durante la simulación y finalmente el área de visión.
- **Tabla Objeto:** Es la tabla que contiene la información del objeto tales como: Identificador, nombre, masa, posición, velocidad, tipo de representación, composición del color, el estado de habilitado y el identificador de la simulación. Además la posición y la velocidad se almacenan en sus tres ejes de coordenadas “X”, “Y” y “Z”, con sus respectivas unidades de medidas. Los valores reales de la masa y la posición se almacena en notación científica: Coeficiente y Exponente.

3.2 Diseño de Interfaz Gráfica

La presente sección muestra los estándares de interfaz, la pantalla principal, la pantalla secundaria y finalmente los pasos para ejecutar una simulación y mostrar sus resultados.

3.2.1 Estándares de Interfaz

La presente sección describe los estándares de interfaz del proyecto de fin de carrera. Tal como se muestra en la figura 3.12, el sistema se visualiza a través de una página Web.

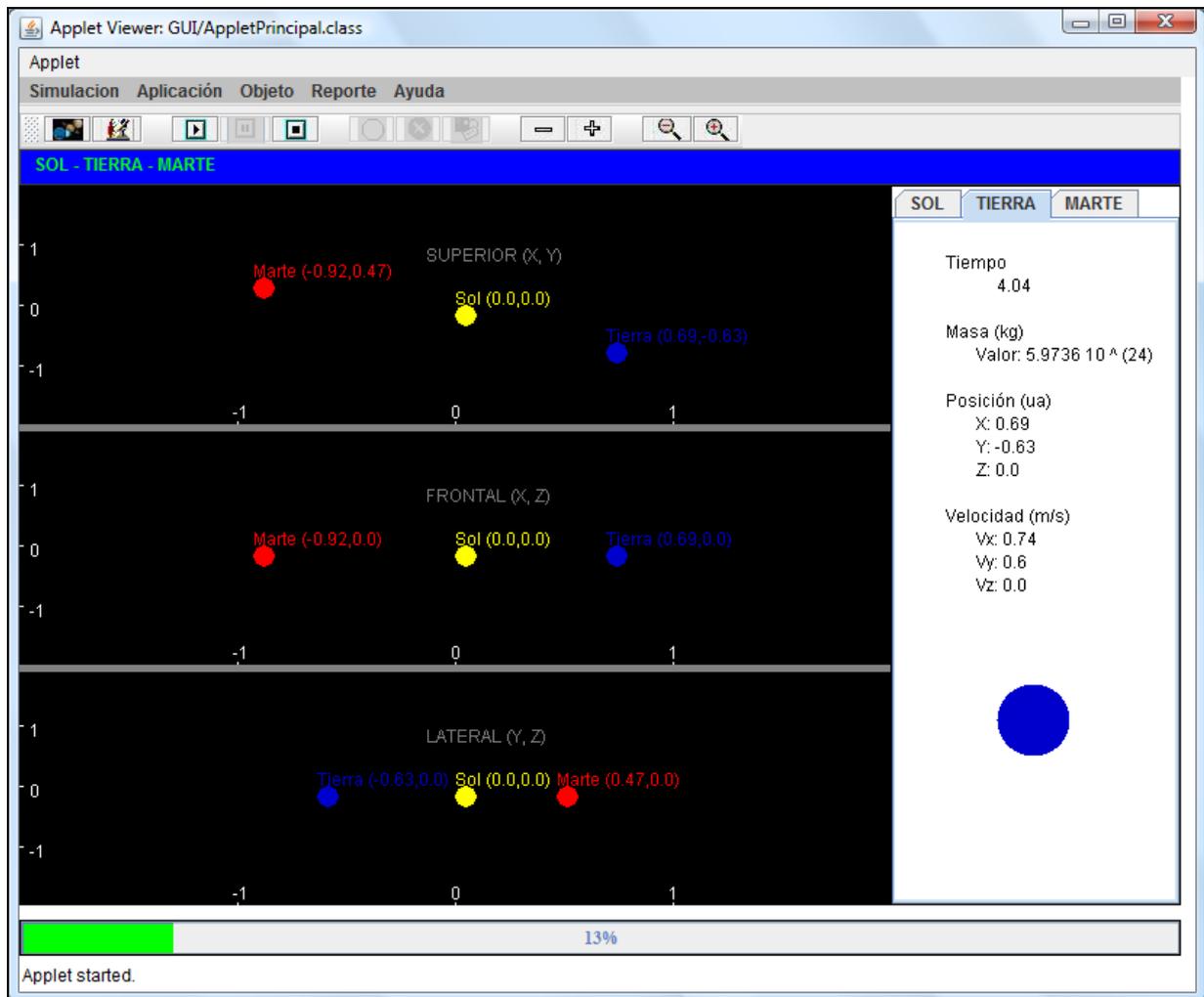


Figura 3.12: Estándar de interfaz, Pagina Web del Simulador.

La página Web contiene un menú que permite visualizar de manera ordenada las funcionalidades del simulador además de una barra de herramientas que ofrece, a través de iconos, un acceso más rápido a las principales funcionalidades del sistema. El menú y la barra de herramientas se encuentran en la parte superior de la pantalla. Ambos controles contienen opciones que al ser seleccionados llaman a las ventanas. Todas las ventanas del simulador son modales, es decir, se mantiene el foco en la ventana del sistema hasta que se tome una determinada acción sobre ella.

Las ventanas del simulador son de los siguientes tipos: Ventana de búsqueda de simulación, ventana de registrar o modificar simulación, ventana de registrar o modificar objetos, ventana de eliminar objeto y ventana de reportes.

- Ventana de búsqueda de simulación: En esta ventana se cuenta con las siguientes secciones: Título de la ventana, el cual describe la operación de la ventana y una barra de herramientas que permite registrar, modificar, eliminar, cargar una simulación o salir de la ventana (parte superior). Esta ventana incluye el área de búsqueda, para localizar una simulación específica en la base de datos según los criterios de búsqueda (parte central). Finalmente el área de resultado que permite mostrar las simulaciones encontradas por la búsqueda (parte inferior). Tal como se muestra en la figura 3.13.

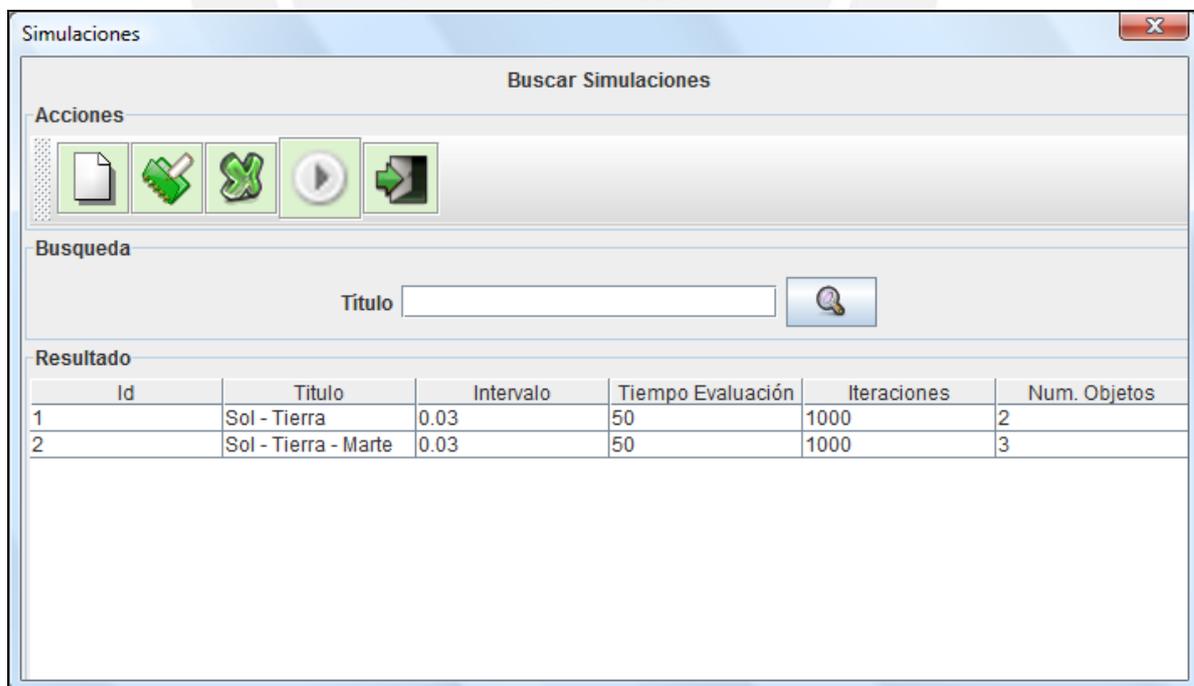


Figura 3.13: Estándar de Interfaz, Ventana de Búsqueda de Simulación.

- Ventana de registrar o modificar simulación: En esta ventana se cuenta con las siguientes secciones: Título de la ventana, el cual describe la operación de la ventana y una barra de herramientas que permite guardar una simulación ó cancelar la operación (parte superior). Asimismo en el título se muestra el nombre de la simulación para identificarlo.

Esta ventana además muestra el área de los datos generales, estos datos son necesarios para el modelo matemático, el cual realiza el proceso de simulación (parte central).

Finalmente el área de las vistas, donde se seleccionan los parámetros de visualización que se realizará durante la ejecución de la simulación (parte inferior), tal como se muestra en la figura 3.14.

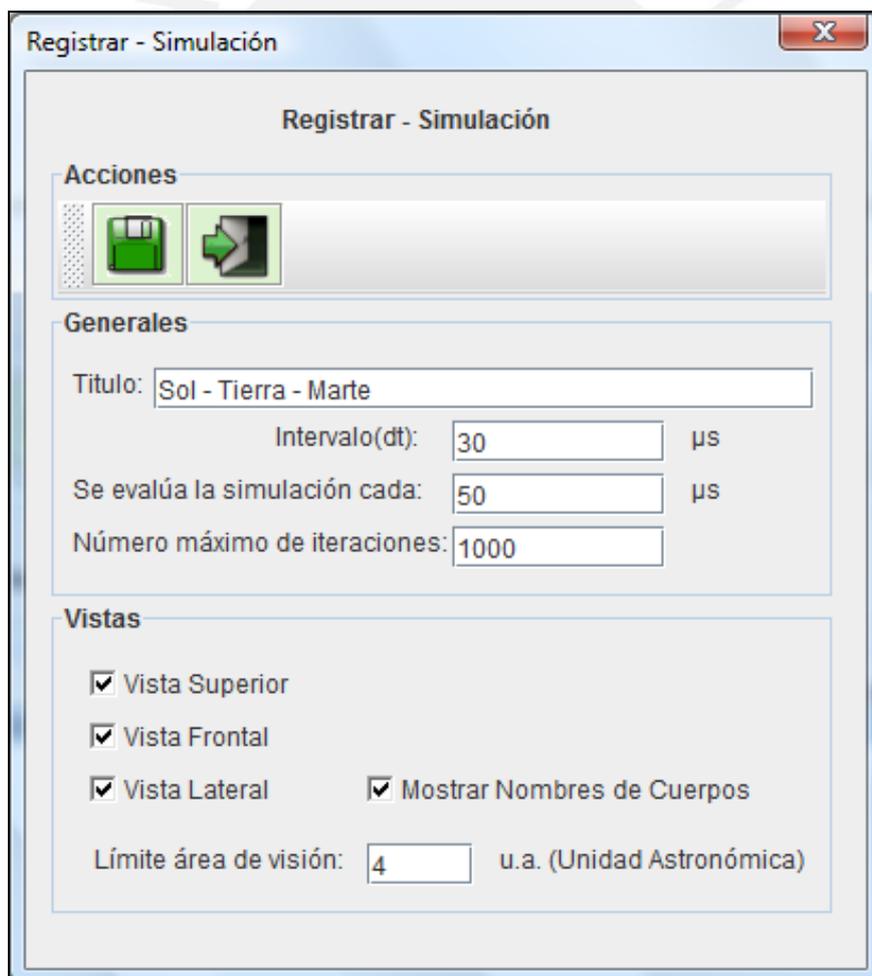


Figura 3.14: Estándar de Interfaz, Ventana de Registrar ó Modificar Simulación

- Ventana de registrar o modificar objetos: En esta ventana se cuenta con las siguientes secciones: Título de la ventana, el cual describe la operación de la ventana y una barra de herramientas que permita guardar un objeto ó cancelar la operación (parte superior). Asimismo en el título se muestra el nombre del objeto para identificarlo.

Esta ventana además muestra el área de la masa del objeto, cuyo valor se registra en connotación científica y el área de la representación del objeto, el cual muestra cómo se visualizará el objeto en la simulación (parte central).

Finalmente se muestra el área de ingreso de la posición inicial y la velocidad inicial del objeto, cuyos valores se registran en connotación científica en sus tres ejes (parte inferior), tal como se muestra en la figura 3.15.

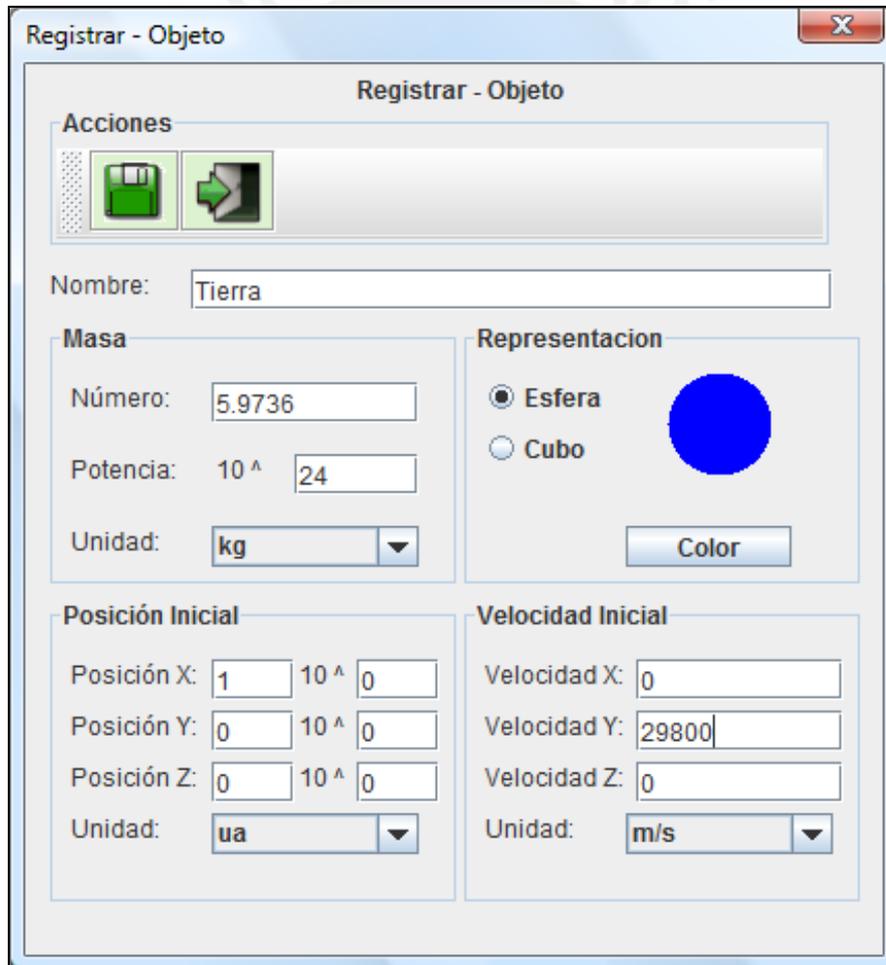


Figura 3.15: Estándar de Interfaz, Ventana de Registrar ó Modificar Objetos

- Ventana eliminar objeto: En esta ventana se cuenta con las siguientes secciones: El título de la ventana, el cual describe la operación de la ventana. Una barra de herramientas que permite eliminar un objeto o cancelar la operación (parte superior).

Incluye además una sección de eliminar, la cual permite seleccionar el objeto que se desea eliminar mostrando su nombre y representación (parte inferior), tal como se muestra en la figura 3.16.



Figura 3.16: Estándar de Interfaz, Ventana de Eliminar Objeto

- Ventanas de reportes: Las ventanas de los reportes tienen el siguiente diseño: Título de ventana, el cual describe la operación del reporte (parte superior).

El diagrama del reporte, el cual muestra los valores de velocidad vs tiempo, fuerza vs tiempo y posición vs tiempo de los objetos (parte central) y finalmente los botones (parte inferior).

Los botones son: El botón “Objetos” que se utiliza para seleccionar los objetos que el diagrama mostrará en la ventana. El botón “Imprimir”, que permite imprimir el diagrama de la ventana. El botón “Guardar Imagen” que permite exportar el diagrama a una imagen y guardarlo en una ruta deseada. El botón “Enviar Mail” que permite enviar el gráfico como archivo adjunto en un mensaje de correo electrónico y el botón “Cerrar” que permite cerrar la ventana del reporte. Esto se muestra en la figura 3.17.

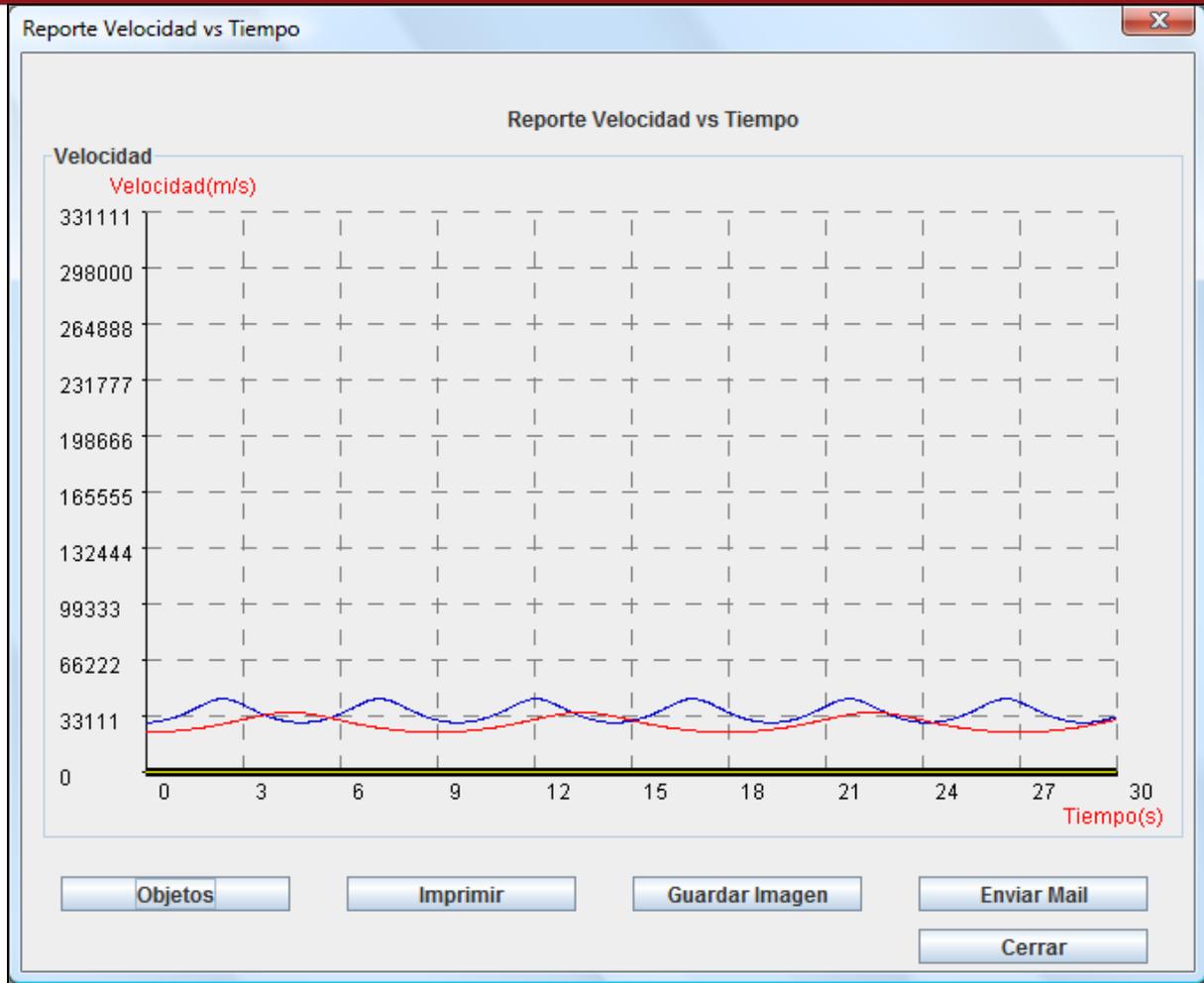


Figura 3.17: Estándar de Interfaz, Ventana de Reportes

Tal como se muestra en la figura 3.12. El sistema muestra el nombre de la simulación que se va a ejecutar, este nombre se encuentra en la parte superior de la página Web. La página Web además tiene un panel de animación el cual contiene tres vistas, para una mejor apreciación de la simulación, estas vistas son utilizadas para mostrar el proceso de simulación.

Las vistas se encuentran en la parte central de la página. La vista superior muestra la parte superior de la simulación. La vista frontal muestra la parte frontal de la simulación. La vista lateral muestra la parte lateral derecha de la simulación. Cada una de las vistas tiene un fondo negro para simular el espacio. Cada vista muestra su nombre para identificarlo y el eje de coordenadas al que pertenece y muestra el número de segmentos en que está dividida.

El sistema contiene un panel de información de los objetos de la simulación, el cual se ubica en la parte derecha del panel de animación. El panel de información contiene pestañas,

cada una de las pestañas muestra la información de cada uno de los objetos involucrados en la simulación. La información es la siguiente: El tiempo transcurrido de la simulación, la masa, la posición y la velocidad en los tres ejes y finalmente la representación del objeto.

El sistema muestra una barra de progreso que se ubica en la parte inferior de la página. Esta barra de progreso se utiliza para mostrar el estado de avance en el tiempo de la simulación, expresado en porcentaje.

3.2.2 Pantalla Principal

La pantalla principal, tal como se muestra en la figura 3.12, muestra un menú que permite encontrar las principales funcionalidades de la simulación del experimento físico. Incluye los siguientes menús:

- Simulación: Menú donde se encuentran las funcionalidades para la construcción de la simulación, cuya opción es Construcción.
- Aplicación: Menú donde se encuentran las funcionalidades ejecutar, detener, pausar una ejecución; aumentar velocidad, disminuir velocidad, alejamiento y acercamiento de los objetos así como las propiedades de visualización de la simulación.
- Objeto: Menú donde se encuentran las funcionalidades de administración de los objetos de una simulación: Registrar, modificar o eliminar un objeto de la simulación.
- Reportes: Menú donde se encuentran las funcionalidades para generar reportes, las cuales son: Velocidad vs Tiempo, Posición vs Tiempo y Fuerza vs Tiempo.
- Ayuda: Contiene información acerca del nombre del producto y el autor.

La pantalla principal muestra una barra de herramientas. En dicha barra se encuentran los íconos, los cuales permiten un acceso más rápido a la funcionalidad del simulador. Tal como se muestra en la tabla 3.1, se muestran y describen las imágenes de los iconos.

<u>TÍTULO</u>	<u>DESCRIPCIÓN</u>	<u>ÍCONO</u>
Simulaciones	Ícono para la búsqueda de las simulaciones.	

<u>TÍTULO</u>	<u>DESCRIPCIÓN</u>	<u>ÍCONO</u>
Modificar Simulación	Ícono para modificar simulación selecciona.	
Ejecutar	Ícono para ejecutar la simulación.	
Pausar	Ícono para pausar la simulación.	
Detener	Ícono para detener la simulación.	
Insertar Objeto	Ícono para insertar un objeto a la simulación.	
Eliminar Objeto	Ícono para eliminar un objeto de la simulación.	
Modificar Objeto	Ícono para modificar un objeto de la simulación.	
Disminuir Velocidad	Ícono para disminuir la velocidad de los objetos.	
Aumentar Velocidad	Ícono para aumentar la velocidad de los objetos.	
Disminuir Vista	Ícono que realiza el efecto de alejamiento de los objetos.	
Aumentar Vista	Ícono que realiza el efecto de acercamiento de los objetos.	

Tabla 3.1: Íconos de la Pantalla Principal

3.2.3 Pantalla Secundaria

La ventana de Simulación, tal como se muestra en la figura 3.14, muestra el título de la ventana, el título describe la acción que se está realizando ya sea registrar o modificar una simulación. La ventana contiene tres secciones, las cuales son: La sección de acciones, la sección de datos generales y la sección de las vistas. La sección de acciones que contiene una barra de herramientas en la parte superior de la ventana, esta barra tiene dos opciones guardar la operación o salir de la ventana. La sección de datos generales que contiene los campos necesarios para que el modelo matemático pueda realizar el proceso de simulación. Dicha sección tiene los siguientes campos:

- Título: Describe la operación (registrar ó modificar) e identifica la simulación.
- Intervalo: Es el diferencial de tiempo que se utiliza para calcular la nueva posición.
- Tiempo de evaluación: Es el tiempo que se demora en refrescar las nuevas posiciones de los objetos en la simulación.
- Número de iteraciones: Es el número de veces que se repetirá la simulación.

La sección de las vistas contiene los campos necesarios que permiten establecer cómo se visualizará la simulación. Dicha sección tiene los siguientes campos:

- Vistas: Permite visualizar la simulación en tres opciones: Superior, lateral y frontal.
- Límite del área de visión: Es el valor del área que abarca la simulación.
- Mostrar nombre de los objetos: Es la opción que permite mostrar los nombres de los objetos durante la ejecución de la simulación.

3.2.4 Pasos para ejecutar una Simulación y mostrar sus resultados

- 1) La pantalla inicial muestra un menú principal, en ella se selecciona la opción “Simulación” y luego la opción “Construcción”, tal como se muestra en la figura 3.18.

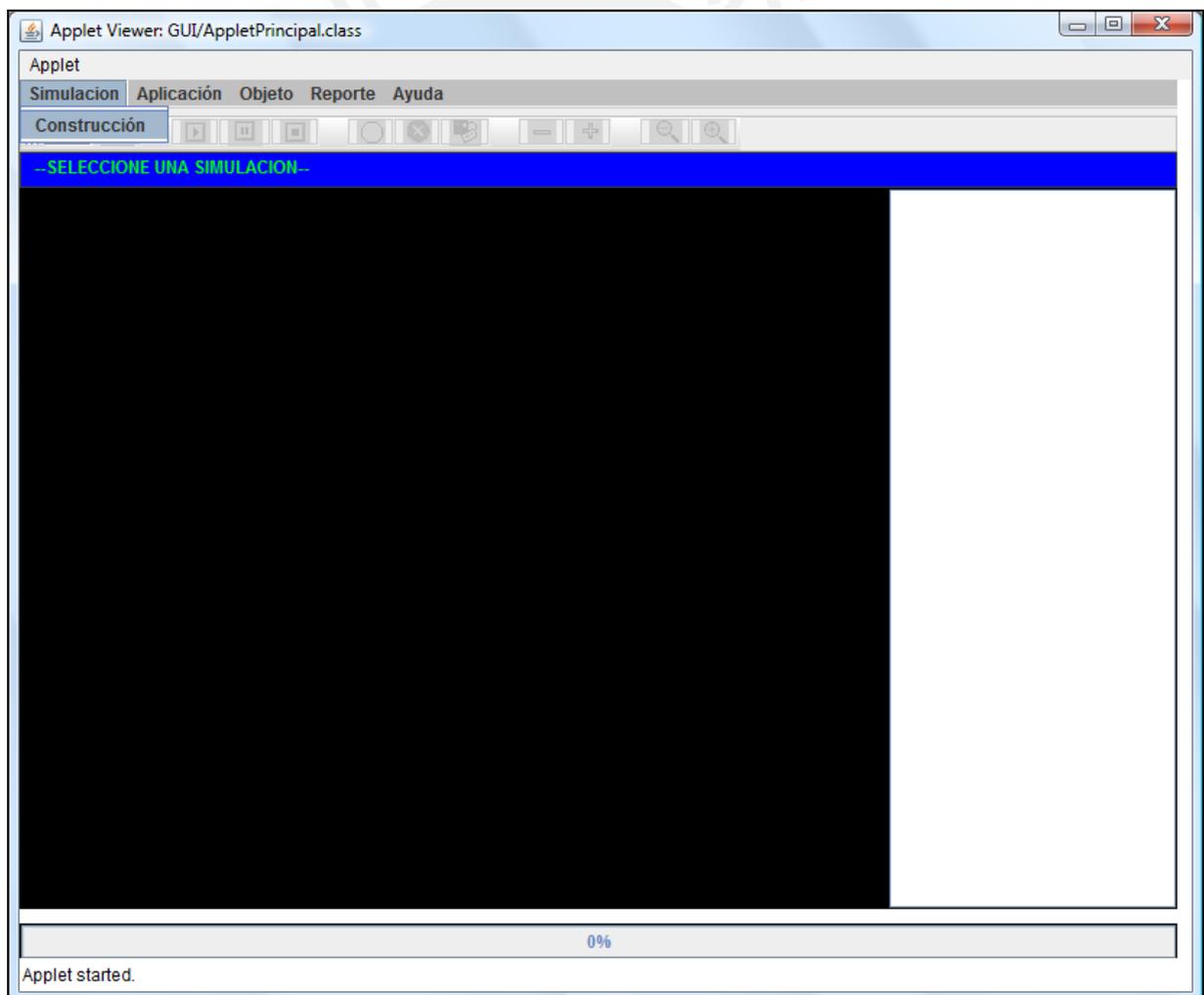


Figura 3.18: Diseño de Interfaz Gráfica, Pantalla Inicial

- 2) Al presionar esta opción se muestra la ventana de “Buscar Simulaciones”, en ella se localiza un botón “Registrar Simulación”, el cual se encuentra en la parte superior izquierda de la ventana y tiene como imagen una hoja en blanco. Este botón permite registrar una nueva simulación, tal como se muestra en la figura 3.13.
- 3) Al seleccionar la opción “Registrar Simulación” permite mostrar la ventana de “Registrar Simulación”, en ella se ingresan los datos, tal como se muestra en la figura 3.14.
- 4) Una vez ingresados los datos de la simulación, se presiona el botón “Guardar”, luego se confirma la operación y se muestra la ventana de “Buscar Simulaciones”. En esta ventana se selecciona la fila de la simulación que se acaba de registrar y se selecciona la opción “Cargar Simulación”, el cual se encuentra en la parte superior de la ventana y se identifica con la imagen “play”, tal como se muestra en la figura 3.13.
- 5) Al seleccionar la opción “Cargar Simulación”, el sistema muestra la pantalla inicial con el nombre de la simulación que se ejecutará. En el menú principal se encuentra la opción “Objeto”, al seleccionarla, se observa la opción “Insertar”, tal como se muestra en la figura 3.19.

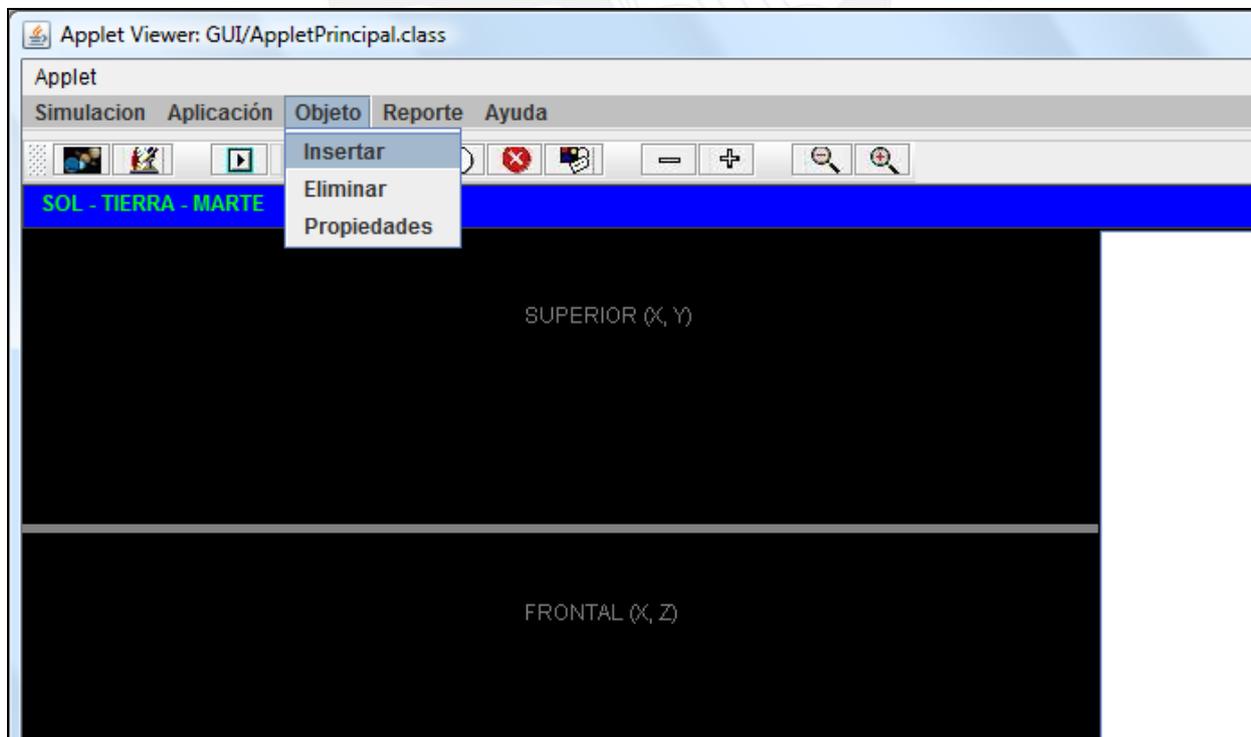


Figura 3.19: Diseño de Interfaz Gráfica, Pantalla Inicial Insertar Objeto.

- 6) Al seleccionar esta opción se muestra la ventana de “Registrar Objeto” que permite ingresar los datos de un nuevo objeto en la simulación, tal como se observa en la figura 3.15.

- 7) Una vez ingresados los datos del objeto, se presiona el botón “Guardar”, luego se confirma la operación y se muestra en pantalla inicial la simulación con el nuevo objeto creado. Se registran los objetos que sean necesarios, luego se presiona el botón “Ejecutar” que permite mostrar el movimiento de los objetos, tal como se muestra en la figura 3.12.

- 8) Al terminar la ejecución de la simulación, se presiona el botón “Reporte” y luego la opción “Fuerza vs Tiempo”, luego se muestra una ventana de “Reporte Fuerza vs Tiempo”, en ella se selecciona los objetos que se desea para el análisis y se muestran los resultados obtenidos, tal como se muestra en la figura 3.17.



4. Construcción

El presente capítulo describe la construcción del sistema, las pruebas al sistema, el algoritmo de simulación y las variables consideradas.

4.1 Construcción del Sistema

La presente sección describe las principales tecnologías empleadas para la construcción del producto del proyecto, tales como el lenguaje de programación java, los applets, los JDBC, el SQLite, el IDE, el JDK, las librerías, las bibliotecas gráficas y los estándares de programación.

4.1.1 Lenguaje de Programación Java

El proyecto se desarrolla en Java ya que este lenguaje de programación es utilizado en los applets, los cuales permiten manipular gráficos en dos dimensiones en las páginas Web.

Java ofrece varias funcionalidades en el entorno Web, tales como usar navegadores que son compatibles con java, dichos navegadores permiten el ahorro a los usuarios en instalar

“plug-in” o programas similares, los cuales que quitan tiempo y espacio en el disco duro. Además java permite poner cualquier clase de elementos multimedia y tiene un alto nivel de interactividad entre el simulador y el alumno.

Dentro de la librería de java existen clases gráficas, tales como AWT y SWING, las cuales permiten crear objetos gráficos configurables, como los círculos y los cuadrados. Estas librerías permiten facilitar la manipulación de los gráficos.

Java es portable, es decir el código compilado de java del simulador es interpretado y solo se requiere implementar un intérprete de java para poder ser utilizado en cualquier computadora.

El lenguaje java soporta la programación con hilos, de esta manera se puede ejecutar diferentes líneas de código al mismo tiempo, tal como el movimiento de cada uno de los objetos en la simulación.

4.1.2 Applets

El proyecto es una aplicación Web y manipula gráficos en dos dimensiones. La tarea de manipulación de gráficos en dos dimensiones en una página Web es dificultoso debido que ante cualquier pedido del cliente, a través de un navegador, estas páginas deben solicitar el código HTML al servidor Web para poder ser visualizadas en el navegador.

El applet es una tecnología que permite manipular los gráficos en dos dimensiones a través de una página Web ya que los applets se ejecutan en la maquina del cliente y no en el servidor, de esta manera se hace más fácil realizar funcionalidades con los gráficos. Esta característica se debe a que el applet es un programa en java, el cual es insertado en una página Web y está almacenado en el servidor Web.

Cuando el cliente desea acceder a la página que contiene el applet, el navegador descarga los siguientes elementos: El applet y la página Web, dicha página contiene el llamado al applet a través del código HTML. El applet una vez descargado se ejecuta en la maquina del cliente.

El applet ofrece funcionalidades como la portabilidad ya que es independiente de la máquina o del sistema operativo que se está utilizando. Además se evita los problemas de actualización y distribución, ya que si se realiza cualquier modificación, esta se puede

almacenar en el servidor Web y luego automáticamente los clientes pueden tener acceso para descargar la última versión del applet.

Los applet pueden ser insertados en cualquier página Web, es decir, puede ser parte de una página Web desarrollada en Java o .NET, solo basta referenciarlo a través del código HTML de la página Web.

4.1.3 Java Data Base Connectivity (JDBC)

Como el proyecto se desarrolla en java, este lenguaje proporciona ventajas en la conexión a base de datos.

Java utiliza JDBC para conectarse a una base de datos. Esta tecnología permite acceder a distintas bases de datos fácilmente, independientemente de la plataforma utilizada.

El manejo de la base de datos es uniforme, es decir, transparente y simple. Solo se requiere un único driver JDBC para poder acceder a la base de datos.

4.1.4 SQLite – Base de Datos Compacta

En el proyecto se utilizó una base de datos compacta, la cual ofrece más beneficios acordes al proyecto.

No se utilizó una base de datos relacional tradicional debido a que posee muchas funcionalidades que no son útiles o necesarias para la aplicación, además que puede ocasionar la sobrecarga en el servidor.

El SQLite implementa un motor de base de datos compacta multiplataforma que no requiere de configuración. El SQLite puede realizar transacciones, los cuales tienen las características siguientes: atomicidad, consistencia, aislamiento y durabilidad. Al principio de cada transacción se bloquea todo el archivo de base de datos. Además, es rápido y permite utilizar el lenguaje estándar SQL. Ver en [SQLite 2011].

SQLite es una pequeña librería, es decir, no es necesario instalarla antes de ser utilizada. Esto evita sobrecargar el servidor, ya que no hay ningún proceso del servidor o cualquier otro que deba iniciar, detener o configurar. La base de datos utiliza la funcionalidad de

SQLite a través de llamadas simples a subrutinas y funciones. Las llamadas a funciones son más eficientes que la comunicación entre procesos.

Además, no hay necesidad de un administrador para crear una instancia de base de datos, administrar los espacios de trabajo o asignar permisos de acceso a los usuarios. El uso de este tipo de base de datos hace que sea más flexible y útil su manejo en el proyecto.

Las bases de datos relacionales tienen una instalación y configuración inicial que a menudo puede ser muy intimidante.

El SQLite combina el motor y la interfaz de base de datos en una sola librería. Los datos son almacenados en un único archivo binario. El SQLite encapsula toda la base de datos en un fichero. Esta característica permite la portabilidad de los datos y solo tiene la restricción del espacio de disco asignado por el administrador del servidor.

Debido a la simplicidad de estas funcionalidades de la base de datos, es una buena solución para el proyecto ya que posee pocos campos de base de datos y la cantidad de registros para guardar es menor. Además el proyecto no tiene un elevado el tráfico de datos y existe poco acceso concurrente a datos.

4.1.5 Entorno de Desarrollo Integrado (IDE)

El proyecto se desarrolla en el IDE NetBeans. El NetBeans es un IDE que permite escribir, compilar, depurar y ejecutar programas escritos en Java. Para el desarrollo del sistema se utiliza la versión 6.8.

Se utiliza este IDE ya que es un producto libre y gratuito, es decir, sin restricciones para su uso. Además su manejo es de fácil uso, contiene mucha documentación de tutoriales y ejemplos en internet.

4.1.6 Java Development Kit (JDK)

El proyecto utiliza el “Java Development Kit” (JDK), el cual provee un conjunto de herramientas para la creación de software en java. Para la construcción del sistema se utiliza la versión 1.6.

Entre los principales programas del JDK se encuentra el Appletviewer, que es un visor de applet que permite generar vistas previas de un applet. Este programa es importante para visualizar el diseño del simulador durante el desarrollo.

4.1.7 Librerías

El proyecto utiliza una librería para la conexión a base de datos. La librería “sqlitejdbc-v056.jar” permite la conexión entre java y la base de datos SQLite. SQLiteJDBC es un driver JDBC para SQLite.

Además el proyecto utiliza una librería para exportar datos a un documento Excel. La librería es “jxl.jar”. Y finalmente las librerías para enviar correos electrónicos. Las cuales son: “dns.jar”, “imap.jar”, “mailapi.jar”, “pop3.jar” y “smtp.jar”. Estas librerías se deben incluir en el desarrollo del proyecto y referenciarlo.

4.1.8 Bibliotecas Gráficas

El proyecto utiliza las siguientes bibliotecas gráficas para Java:

- Java.awt: Contiene todas las clases que permiten crear interfaces gráficas de usuarios y pintar gráficos e imágenes. Por ejemplo, botones, etiquetas, listas, campos de texto, áreas de texto, entre otros.
- Javax.swing: Contiene componentes gráficos tales como cajas de texto, botones, desplegables y tablas.

4.1.9 Estándares de Programación

La presente sección describe los estándares de programación utilizados en el proyecto.

- Declaración de paquetes: Los nombres de los paquetes son en mayúsculas. Por ejemplo: *package GUI.OBJETO*;
- Declaración de clases: Los nombres de las clases son una serie de palabras juntas, sin espacios, las primeras letras se declaran según al paquete al que pertenece. Se separan en cuatro grupos, tal como se muestra en la tabla 4.1.

<u>PAQUETE</u>	<u>PREFIJO</u>
BE (Entidades del negocio)	BE + Nombre de la clase.
BD (Acceso a base de datos)	BL + Nombre de la clase.
BL (Lógica de negocios)	BD + Nombre de la clase.
GUI (Interfaz gráfica)	Si se hereda de otra clase o implementa una interface, las primeras letras es el nombre de la clase o interfaz y a continuación el nombre de la clase. Por ejemplo: <code>public class AppletPrincipal extends Applet{ ... }</code>

Tabla 4.1: Estándar de Programación de Clases

Para el nombre de la clase se utiliza el prefijo mostrado en la tabla 4.1, más el nombre de la clase, el cual tiene la primera letra de cada palabra en mayúsculas y el resto en minúsculas. Por ejemplo: `public class BEObjecto { ... }`

- **Declaración de constantes:** Los nombres de las constantes se declaran en mayúsculas. Por ejemplo: `public final static String URL = "...";`
- **Declaración de atributos:** Los nombres de los atributos son una serie de palabras, sin espacios y tiene el siguiente formato, tal como se muestra en la tabla 4.2 se utilizan los siguientes prefijos.

<u>NOMBRE DEL TIPO</u>	<u>PREFIJO</u>
boolean	bl
char	ch
double	db
float	fl
int	in

<u>NOMBRE DEL TIPO</u>	<u>PREFIJO</u>
string	str
arreglo	arr

Tabla 4.2: Estándar de Programación de Atributos

En las clases del paquete GUI se encuentran clases o controles como atributos.

A continuación se describen los prefijos de las clases como atributos, tal como se muestra en la tabla 4.3.

<u>NOMBRE DE LA CLASE</u>	<u>PREFIJO</u>
Thread	thr
Applet	app
BESimulacion	beSimulacion
BEObjeto	beObjeto
BEVista	beVista
BEVelocidad	beVelocidad
List	lst
Image	img
Graphics	grp
Dimension	dim
Color	co

Tabla 4.3: Estándar de Programación de Clases como Atributos

A continuación se describen los prefijos de los controles como atributos, tal como se muestra en la tabla 4.4.

<u>NOMBRE DEL CONTROL</u>	<u>PREFIJO</u>
JLabel	lbl
Canvas	can
JTabbedPane	tab
JProgressBar	pb
JButton	btn
JMenuItem	mitem
TextField	txt
JTable	tbl
JDesktopPane	dp
JComboBox	cbb
JPanel	pnl
JRadioButton	rdb
JFrame	frm
Checkbox	chb

Tabla 4.4: Estándar de Programación de Controles como Atributos

Para el nombre del atributo se utiliza el prefijo mostrado en la tabla 4.2, 4.3 y 4.4, más el nombre del atributo, el cual tiene la primera letra de cada palabra en mayúsculas y el resto en minúsculas. Por ejemplo: *private int inIdSimulacion;*

- **Declaración de los métodos:** Los nombres de los métodos son una serie de palabras juntas, sin espacios. La primera palabra representa una acción con todas sus letras en minúsculas, el resto de palabras, si las hay, describen sobre la acción realizada. Esta acción se describe con la primera letra de cada una en mayúsculas y el resto en minúsculas. Por ejemplo: *public void buscarSimulaciones() { ... }.*

4.2 Pruebas

La presente sección describe el tipo de pruebas realizadas y el plan de pruebas en el proyecto.

4.2.1 Selección del Tipo de Prueba

El tipo de pruebas utilizadas para el proyecto son las pruebas unitarias al simulador, estas pruebas es una forma de verificar el correcto funcionamiento de un módulo de código, es decir cada capa de la arquitectura del simulador.

De esta manera se puede asegurar que cada una de las capas funcione correctamente por separado. Las capas de la arquitectura del simulador son: Capa de presentación o interfaz gráfica, capa de lógica de negocios y finalmente la capa de acceso a base de datos. Asimismo del componente de entidades de negocio.

Además se realizan las pruebas de sistemas ya que estas permiten evaluar la integración del simulador. De esta manera se verifica el funcionamiento correcto de las interacciones de las interfaces entre los distintos módulos que lo componen o con los que se comunican, como por ejemplo la interacción entre la administración de la simulación, la ejecución de la simulación, el mantenimiento de objetos y los reportes de la simulación.

En las pruebas de sistemas se destaca las pruebas de caja negra. Estas pruebas se realizan desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno, de esta manera se pone énfasis en las interfaces del sistema.

Estas pruebas son útiles para el simulador debido a que se cuenta con varios módulos que tienen interfaces de petición de datos y se requieren pruebas que evalúen la interacción entre estas interfaces y la interfaz principal de la simulación, que es donde se ejecuta la simulación. La funcionalidad principal del simulador es mostrar a través de una interfaz el proceso de simulación de los objetos.

4.2.2 Plan de Pruebas

A continuación se describen las pruebas unitarias del proyecto.

Capa Lógica del Negocio: Se describen las pruebas unitarias realizadas a la capa lógica de negocio del simulador, tal como se muestra en la tabla 4.5.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Se pasa como parámetro un objeto de la clase BObjeto de una determinada simulación, al método “buscarObjetos” de la clase BObjeto. En los atributos de dicho objeto no tienen ningún valor asignado.	El método devuelve una lista de todos los objetos de la simulación.
Se pasa como parámetro un objeto de la clase BObjeto de una determinada simulación al método “buscarObjetos” de la clase BObjeto. Uno de los atributos de dicho objeto tiene un valor asignado.	El método devuelve una lista de todos los objetos de la simulación que tienen en el mismo atributo el mismo valor.
Ocurre un error inesperado en el método “buscarObjetos” de la clase BObjeto.	El método lanza una excepción, la cual contiene la información de lo que ocurrió. La excepción es lanzada a la capa de interfaz.
Se pasa como parámetro un objeto de la clase BObjeto de una determinada simulación, al método “insertarObjeto” de la clase BObjeto. En los atributos de dicho objeto no tienen ningún valor asignado.	El método registra los valores por defecto y devuelve el número de registros procesados.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Se pasa como parámetro un objeto de la clase BObjeto de una determinada simulación al método “insertarObjeto” de la clase BObjeto. Todos los atributos de dicho objeto tienen un valor asignado.	El método registra los valores asignados y devuelve el número de registros procesados.
Ocurre un error inesperado en el método “insertarObjeto” de la clase BObjeto.	El método lanza una excepción, la cual contiene la información de lo que ocurrió. La excepción es lanzada a la capa de interfaz.

Tabla 4.5: Pruebas Unitarias, Capa Lógica.

Componente de Entidades del Negocio: Se describen las pruebas unitarias realizadas al componente de entidades del negocio del simulador, tal como se muestra en la tabla 4.6.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Se pasa como parámetro una lista de objetos de la clase BObjeto de una determinada simulación al método “obtenerObjetoGravedad” de la clase BObjeto. Esta lista no posee ningún objeto.	El método devuelve un valor nulo.
Se pasa como parámetro una lista de objetos de la clase BObjeto de una determinada simulación al método “obtenerObjetoGravedad” de la clase BObjeto. Esta lista posee varios objetos.	El método devuelve un objeto de la clase BObjeto, el cual es el objeto que ejerce mayor fuerza de gravedad al objeto que lo calcula.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
<p>Se pasa como parámetro una lista de objetos de la clase BObjeto de una determinada simulación al método “obtenerObjetoGravedad” de la clase BObjeto. Esta lista posee varios objetos. Esta lista posee una referencia al mismo objeto que está calculando su objeto gravedad.</p>	<p>El método evita que el objeto que está calculando su objeto gravedad, tome su propia referencia en el cálculo. Es decir, solo se toma en cuenta los demás objetos.</p>
<p>Ocurre un error inesperado en el método “obtenerObjetoGravedad” de la clase BObjeto.</p>	<p>El método lanza una excepción, la cual contiene la información de lo que ocurrió. La excepción es lanzada a la capa que lo referencia.</p>
<p>Se pasa como parámetro una lista de objetos de la clase BObjeto de una determinada simulación al método “calcularNuevaPosicion” de la clase BObjeto. Esta lista posee varios objetos. Se obtiene su objeto gravedad. Su objeto gravedad es nulo.</p>	<p>El método calcula la nueva posición del objeto en la simulación sin tomar en cuenta ningún objeto gravedad.</p>
<p>Se pasa como parámetro una lista de objetos de la clase BObjeto de una determinada simulación al método “calcularNuevaPosicion” de la clase BObjeto. Esta lista posee varios objetos. Se obtiene su objeto gravedad. Su objeto gravedad no es nulo.</p>	<p>El método calcula la nueva posición del objeto en la simulación en base a su objeto gravedad.</p>

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
<p>Se pasa como parámetro una lista de objetos de la clase BObjeto de una determinada simulación al método “calcularNuevaPosicion” de la clase BObjeto. Esta lista posee varios objetos. Esta lista posee una referencia al mismo objeto que está calculando su nueva posición.</p>	<p>El método evita que el objeto que está calculando su nueva posición, tome su propia referencia en el cálculo. Es decir, solo se toma en cuenta los demás objetos.</p>
<p>Ocurre un error inesperado en el método “calcularNuevaPosicion” de la clase BObjeto.</p>	<p>El método lanza una excepción, la cual contiene la información de lo que ocurrió. La excepción es lanzada a la capa que lo referencia.</p>

Tabla 4.6: Pruebas Unitarias, Componente de Entidades del Negocio.

Capa de acceso a la base de datos: Se describen las pruebas unitarias realizadas a la capa de acceso a la base de datos del simulador, tal como se muestra en la tabla 4.7.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
<p>Se pasa como parámetro un objeto de la clase BObjeto de una determinada simulación, al método “buscarObjetos” de la clase BObjeto. En los atributos de dicho objeto no tienen ningún valor asignado.</p>	<p>El método devuelve una lista de todos los objetos de la simulación.</p>
<p>Se pasa como parámetro un objeto de la clase BObjeto de una determinada simulación al método “buscarObjetos” de la clase BObjeto. Uno de los atributos de dicho objeto tiene un valor asignado.</p>	<p>El método devuelve una lista de todos los objetos de la simulación que tienen en el mismo atributo el mismo valor.</p>

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Ocurre un error inesperado en el método “buscarObjetos” de la clase BDObjeto.	El método lanza una excepción, la cual contiene la información de lo que ocurrió. La excepción es lanzada a la capa de lógica de negocio.
Se pasa como parámetro un objeto de la clase BEObjeto de una determinada simulación, al método “insertarObjeto” de la clase BDObjeto. En los atributos de dicho objeto no tienen ningún valor asignado.	El método registra los valores por defecto y devuelve el número de registros procesados.
Se pasa como parámetro un objeto de la clase BEObjeto de una determinada simulación al método “insertarObjeto” de la clase BDObjeto. Todos los atributos de dicho objeto tienen un valor asignado.	El método registra los valores asignados y devuelve el número de registros procesados.
Ocurre un error inesperado en el método “insertarObjeto” de la clase BDObjeto.	El método lanza una excepción, la cual contiene la información de lo que ocurrió. La excepción es lanzada a la capa de lógica de negocio.

Tabla 4.7: Pruebas Unitarias, Capa de Acceso a Base de Datos.

La presente sección describe las pruebas de sistemas. Se utiliza las pruebas de caja negra en los casos de uso del proyecto.

Administrar Simulación: Se describen las pruebas realizadas al caso de uso Administrar Simulación, tal como se muestra en la tabla 4.8.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Intentar asignar un valor decimal a la propiedad de número de iteraciones de la simulación.	El simulador de física debe deshabilitar el ingreso de valores decimales a la propiedad de número de iteraciones de la simulación.
Intentar asignar un valor negativo a la propiedad de intervalo, que es el diferencial de tiempo de la simulación.	El simulador de física debe deshabilitar el ingreso de valores negativos a la propiedad de intervalo de la simulación.

Tabla 4.8: Pruebas de Sistema, Administrar Simulación

Asignar Objetos: Se describen las pruebas realizadas al caso de uso Asignar Objetos, tal como se muestra en la tabla 4.9.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Intentar asignar un valor alfabético a la masa, posición y velocidad de un objeto.	El simulador de física debe deshabilitar el ingreso de valores alfabéticos a la masa, posición y velocidad de un objeto.
Intentar asignar un valor negativo a la masa de un objeto.	El simulador de física debe deshabilitar el ingreso de valores negativos a la masa del objeto.

Tabla 4.9: Pruebas de Sistema, Asignar Objetos

Ejecutar Simulación: Se describen las pruebas realizadas al caso de uso Ejecutar Simulación, tal como se muestra en la tabla 4.10.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
En una simulación en ejecución, tratar de editar las propiedades de un objeto.	El simulador de física debe deshabilitar las opciones que permiten editar las propiedades de un objeto durante la ejecución de la simulación.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Intentar añadir objetos en tiempo de ejecución de la simulación.	El simulador de física debe deshabilitar las opciones que permiten añadir objetos durante la ejecución de la simulación.
En una simulación, el número de iteraciones llega a su máximo.	La simulación debe concluir y mostrar el mensaje respectivo.
En una simulación donde aún no se ha realizado la ejecución y se desea realizar un reporte gráfico de variable versus variable.	El simulador de física debe deshabilitar las opciones que permiten realizar un reporte gráfico de variable versus variable cuando aún no se realiza la simulación.
Tratar de ejecutar una simulación que no tiene ningún objeto.	El simulador de física debe mostrar un mensaje de error al usuario indicando que la simulación no tiene objetos.
Realizar una pausa en la simulación e intentar añadir, editar o eliminar un objeto de la simulación.	El simulador de física debe deshabilitar las opciones que permiten añadir, editar o eliminar los objetos durante la pausa de la simulación.

Tabla 4.10: Pruebas de Sistema, Ejecutar Simulación

Cargar Simulación: Se describen las pruebas realizadas al caso de uso Cargar Simulación, tal como se muestra en la tabla 4.11.

<u>PRUEBA</u>	<u>RESULTADO ESPERADO</u>
Seleccionar una simulación que contiene datos con formato incorrecto.	El simulador de física muestra un mensaje de error indicando que existe un error en los datos de la simulación.

Tabla 4.11: Pruebas de Sistema, Cargar Simulación

4.3 Algoritmo de Simulación

En la presente sección se describen los dos algoritmos de simulación principales utilizados en el proyecto. Los dos algoritmos se presentan como una función y un procedimiento, los cuales son:

- Función “BEObjeto.ObtenerObjetoGravedad”.
- Procedimiento “BEObjeto.CalcularNuevaPosicion”.

Ambos algoritmos se utilizan en un solo proceso, es decir, el procedimiento “BEObjeto.CalcularNuevaPosicion” llama a la función “BEObjeto.ObtenerObjetoGravedad”, para que esta función realice una determinada tarea.

La tarea de la función es darle un valor al procedimiento, este valor obtenido es el objeto que ejerce mayor gravedad sobre el objeto que lo está calculando. El procedimiento lo utiliza para calcular la nueva posición del objeto en este instante de tiempo.

A continuación se describen cada uno de los algoritmos:

- Función “BEObjeto.ObtenerObjetoGravedad”: Esta función permite obtener el “objeto gravedad” de otro objeto. El “objeto gravedad” es el objeto que atrae a otro objeto. Un objeto evalúa a cada uno de los demás objetos de la simulación para determinar en ese instante de tiempo cuál es el objeto que lo atrae. Por ejemplo: En el sistema planetario solar el “objeto gravedad” del planeta Tierra es el Sol.

Pasos del algoritmo:

- Primero, se obtienen la lista de todos los objetos de la simulación.
- Luego, se recorren cada uno de los objetos de la simulación.
- Después, se verifica que el objeto que evalúa no aplique la función en sí mismo.
- Luego, para cada uno de los objetos se calcula la fuerza de gravedad que ejercen sobre el objeto que evalúa.
- Después, se determina cuál es la fuerza de gravedad mayor que se ejerce sobre el objeto que evalúa.
- Finalmente, el objeto que ejerza mas fuerza de gravedad es el “objeto gravedad”.

Pseudocódigo del algoritmo:

```

Función BObjeto.ObtenerObjetoGravedad(ListaObjetos)
  fuerzamax ← 0
  índice ← -1
  para i ← 0 hasta ListaObjetos.NumeroDeObjetos() hacer
    objeto ← ListaObjetos.ObtenerObjeto(i)
    si (nombre <> objeto.ObtenerNombre()) entonces
      dx ← posicionX - objeto.PosicionX()
      dy ← posicionY - objeto.PosicionY()
      dz ← posicionZ - objeto.PosicionZ()
      r ← √(dx2+ dy2+ dz2)
      fuerza ← (G x objeto.ObtenerMasa()) ÷ r2
      si (fuerzamax < fuerza) entonces
        fuerzamax ← fuerza
        índice ← i
      fin si
    fin si
  fin para
  devolver ListaObjetos.ObtenerObjeto (índice)
fin función
  
```

- Procedimiento “BObjeto.CalcularNuevaPosicion”: Este procedimiento permite calcular los valores de la nueva posición del objeto en un instante de tiempo “dt” (diferencial de tiempo). Este procedimiento se base en el método de Runge – Kutta.

Los métodos de Runge - Kutta logran la exactitud del procedimiento de una serie de Taylor sin requerir el cálculo de derivadas superiores. Ver en [Ascher 1998]. Existen muchas variaciones, pero todas se pueden denotar en la forma generalizada de la ecuación, tal como se muestra en la figura 4.1:

$$y'_{i+1} = y_i + \phi(x_i, y_i, h)h$$

Figura 4.1: Método Runge – Kutta, Forma Generalizada.

Donde $\phi(x_i, y_i, h)$ es conocida como función incremento, esta función puede interpretarse como una pendiente representativa sobre el intervalo, tal como se muestra en la figura 4.2.

$$\phi = a_1 k_1 + a_2 k_2 + \dots + a_n k_n$$

Figura 4.2: Método Runge – Kutta, Función Incremento.

Donde las “a” son constantes y las “k”, tal como se muestra en la figura 4.3, son:

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + p_1 h, y_i + q_{11} k_1 h) \\ k_3 &= f(x_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h) \end{aligned}$$

Figura 4.3: Método Runge – Kutta, Cálculo de las k.

Las “k” son relaciones de recurrencia, esto es, “k1” aparece en la ecuación para “k2”, la cual aparece en la ecuación para “k3”, etc.

Como cada “k” es una evaluación funcional, esta recurrencia hace que los métodos Runge-Kutta sean eficientes para la programación. Existen varios tipos de métodos Runge-Kutta al emplear diferentes números de términos en la función incremento como la especificada por “n”.

El más popular de los métodos Runge-Kutta es de cuarto orden, $n = 4$, tal como se muestra en la figura 4.4:

$$\begin{aligned} y_{i+1} &= y_i + \left(\frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \right) h \\ k_1 &= f(x_i, y_i) \\ k_2 &= f\left(x_i + \frac{h}{2}, y_i + k_1 \frac{h}{2}\right) \\ k_3 &= f\left(x_i + \frac{h}{2}, y_i + k_2 \frac{h}{2}\right) \\ k_4 &= f(x_i + h, y_i + k_3 h) \end{aligned}$$

Figura 4.4: Método Runge – Kutta, Ecuación de Cuarto Orden.

Pasos del algoritmo:

- Primero, se obtiene la lista de todos los objetos de la simulación.
- Luego, se obtiene el “objeto gravedad” que interactúa con el objeto que evalúa.
- Después, se aplica los métodos de Runge-Kutta de cuarto orden.
- Finalmente, se establecen los valores de las variables en un instante de tiempo “dt” (diferencial de tiempo).

Pseudocódigo del algoritmo:

Procedimiento BObjeto.CalcularNuevaPosicion(ListaObjetos, dt)

objetoGravedad ← obtenerObjetoGravedad(ListaObjetos)

si (objetoGravedad = null) entonces

objetoGravedad ← CrearObjeto()

fin si

si (objetoGravedad.Nombre <> Nombre)

dx ← posicionX – objetoGravedad.PosicionX()

dy ← posicionY – objetoGravedad.PosicionY()

dz ← posicionZ – objetoGravedad.PosicionZ()

x ← dx

y ← dy

z ← dz

vx ← velocidadX

vy ← velocidadY

vz ← velocidadZ

$r \leftarrow \sqrt{(x^2 + y^2 + z^2)}$

Kx1 ← vx x dt

Ky1 ← vy x dt

Kz1 ← vz x dt

$Kvx1 \leftarrow -(G \times objetoGravedad.Masa() \times x) \times dt \div r^3$

$Kvy1 \leftarrow -(G \times objetoGravedad.Masa() \times y) \times dt \div r^3$

$Kvz1 \leftarrow -(G \times objetoGravedad.Masa() \times z) \times dt \div r^3$

x ← dx + Kx1 / 2

y ← dy + Ky1 / 2

$$z \leftarrow dz + Kz1 / 2$$

$$vx \leftarrow \text{velocidadX} + Kvx1 / 2$$

$$vy \leftarrow \text{velocidadY} + Kvy1 / 2$$

$$vz \leftarrow \text{velocidadZ} + Kvz1 / 2$$

$$r \leftarrow \sqrt{(x^2 + y^2 + z^2)}$$

$$Kx2 \leftarrow vx \times dt$$

$$Ky2 \leftarrow vy \times dt$$

$$Kz2 \leftarrow vz \times dt$$

$$Kvx2 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times x) \times dt \div r^3$$

$$Kvy2 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times y) \times dt \div r^3$$

$$Kvz2 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times z) \times dt \div r^3$$

$$x \leftarrow dx + Kx2 / 2$$

$$y \leftarrow dy + Ky2 / 2$$

$$z \leftarrow dz + Kz2 / 2$$

$$vx \leftarrow \text{velocidadX} + Kvx2 / 2$$

$$vy \leftarrow \text{velocidadY} + Kvy2 / 2$$

$$vz \leftarrow \text{velocidadZ} + Kvz2 / 2$$

$$r \leftarrow \sqrt{(x^2 + y^2 + z^2)}$$

$$Kx3 \leftarrow vx \times dt$$

$$Ky3 \leftarrow vy \times dt$$

$$Kz3 \leftarrow vz \times dt$$

$$Kvx3 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times x) \times dt \div r^3$$

$$Kvy3 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times y) \times dt \div r^3$$

$$Kvz3 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times z) \times dt \div r^3$$

$$x \leftarrow dx + Kx3$$

$$y \leftarrow dy + Ky3$$

$$z \leftarrow dz + Kz3$$

$$vx \leftarrow \text{velocidadX} + Kvx3$$

$$vy \leftarrow \text{velocidadY} + Kvy3$$

$$vz \leftarrow \text{velocidadZ} + Kvz3$$

$$r \leftarrow \sqrt{(x^2 + y^2 + z^2)}$$

$$Kx4 \leftarrow vx \times dt$$

$$Ky4 \leftarrow vy \times dt$$

$$Kz4 \leftarrow vz \times dt$$

$$Kvx4 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times x) \times dt \div r^3$$

$$Kvy4 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times y) \times dt \div r^3$$

$$Kvz4 \leftarrow -(G \times \text{objetoGravedad.Masa}() \times z) \times dt \div r^3$$

$$Kposx \leftarrow (Kx1 + 2 \times Kx2 + 2 \times Kx3 + Kx4) \div 6$$

$$Kposy \leftarrow (Ky1 + 2 \times Ky2 + 2 \times Ky3 + Ky4) \div 6$$

$$Kposz \leftarrow (Kz1 + 2 \times Kz2 + 2 \times Kz3 + Kz4) \div 6$$

$$Kvelx \leftarrow (Kvx1 + 2 \times Kvx2 + 2 \times Kvx3 + Kvx4) \div 6$$

$$Kvely \leftarrow (Kvy1 + 2 \times Kvy2 + 2 \times Kvy3 + Kvy4) \div 6$$

$$Kvelz \leftarrow (Kvz1 + 2 \times Kvz2 + 2 \times Kvz3 + Kvz4) \div 6$$

$$\text{posicionX} \leftarrow \text{posicionX} + Kposx$$

$$\text{posicionY} \leftarrow \text{posicionY} + Kposy$$

$$\text{posicionZ} \leftarrow \text{posicionZ} + Kposz$$

$$\text{velocidadX} \leftarrow \text{velocidadX} + Kvelx$$

$$\text{velocidadY} \leftarrow \text{velocidadY} + Kvely$$

$$\text{velocidadZ} \leftarrow \text{velocidadZ} + Kvelz$$

fin si

fin procedimiento

4.4 Variables Consideradas en el Algoritmo:

En el proyecto se incluyen nuevas variables en el algoritmo de simulación que no se consideran en la tesis anterior. Tal como se muestra en la tabla 4.12.

<u>VARIABLES</u>	<u>DESCRIPCIÓN</u>
dt	Es el diferencial de tiempo. Esta variable es utilizada en el algoritmo para determinar en ese instante de tiempo la nueva posición del objeto.
Kx1, Ky1, Kz1	Son los valores de la pendiente al principio del intervalo, el intervalo es el diferencial del espacio, es decir el desplazamiento, en los tres ejes de coordenadas respectivamente (x, y, z).
Kx2, Ky2, Kz2	Son los valores de la pendiente en el punto medio del intervalo, usando Kx1, Ky1 y Kz1 para determinar el valor de la siguiente posición, el intervalo es el diferencial del espacio, es decir el desplazamiento, en los tres ejes de coordenadas respectivamente (x, y, z).

<u>VARIABLES</u>	<u>DESCRIPCIÓN</u>
Kx3, Ky3, Kz3	Son los valores de la pendiente del punto medio del intervalo pero ahora usando Kx2, Ky2 y Kz2 para determinar el valor de la siguiente posición, el intervalo es el diferencial del espacio, es decir el desplazamiento, en los tres ejes de coordenadas respectivamente (x, y, z).
Kx4, Ky4, Kz4	Son los valores de la pendiente al final del intervalo, el intervalo es el diferencial del espacio, es decir el desplazamiento, en los tres ejes de coordenadas respectivamente (x, y, z).
Kvx1, Kvy1, Kvz1	Son los valores de la pendiente al principio del intervalo, el intervalo es el diferencial de la velocidad en los tres ejes de coordenadas respectivamente (x, y, z).
Kvx2, Kvy2, Kvz2	Son los valores de la pendiente en el punto medio del intervalo, usando Kvx1, Kvy1 y Kvz1 para determinar el valor de la siguiente velocidad, el intervalo es el diferencial de la velocidad en los tres ejes de coordenadas respectivamente (x, y, z).
Kvx3, Kvy3, Kvz3	Son los valores de la pendiente del punto medio del intervalo pero ahora usando Kvx2, Kvy2 y Kvz2 para determinar el valor de la siguiente velocidad, el intervalo es el diferencial de la velocidad en los tres ejes de coordenadas respectivamente (x, y, z).
Kvx4, Kvy4, Kvz4	Son los valores de la pendiente al final del intervalo, el intervalo es la diferencial de la velocidad en los tres ejes de coordenadas respectivamente (x, y, z).
Kposx, Kposy, Kposz	Son los valores que se obtienen promediando las cuatro pendientes de cada uno de los tres ejes de la posición. El método de Runge - Kutta asigna mayor peso a las pendientes en el punto medio.
Kvelx, Kvely, Kvelz	Son los valores que se obtienen promediando las cuatro pendientes de cada uno de los tres ejes de la velocidad. El método de Runge - Kutta asigna mayor peso a las pendientes en el punto medio.

Tabla 4.12: Variables en el Algoritmo

5 Observaciones, Conclusiones y Recomendaciones

El presente capítulo describe las observaciones, conclusiones y recomendaciones para trabajos futuros.

5.1 Observaciones

El sistema de simulación permite descubrir y asegurar los conocimientos físicos, además del impulso a aprender, para ello se utiliza el ensayo y error como procedimiento y la prueba de hipótesis.

Un sistema de simulación permite que un escenario donde se realiza la simulación pueda ser reproducido un número ilimitado de veces con las condiciones iniciales, lo cual en un experimento físico real no siempre es factible de realizar.

El sistema de simulación maneja diversos objetos, los cuales interactúan entre sí cuando se realiza la simulación. El sistema ofrece 3 vistas, superior, frontal y lateral. Esto permite una mejor apreciación de la simulación.

El sistema de simulación aporta beneficios al proceso de enseñanza y aprendizaje, mediante el uso de escenarios de simulación reproducibles y de fácil distribución a través de los navegadores Web.

En la actualidad el uso del computador es necesario para diversas tareas, así como para planificar e implementar las actividades educativas. El uso del computador ofrece un valor agregado a los medios educativos tradicionales.

Durante el proceso de desarrollo se empleó la tecnología Java. Esta tecnología tiene dos principales características: La primera característica es que permite diseñar aplicaciones en base a objetos, los cuales son entidades que combinan estado, comportamiento e identidad. La segunda característica es la independencia de la plataforma, es decir permite ejecutar la aplicación en cualquier tipo de sistema operativo.

La programación orientada a objetos permite diseñar las simulaciones y los objetos, además de definir a cada objeto de simulación con un comportamiento complejo propio y con propiedades.

El simulador de física se realizó como una aplicación Web debido a que varios usuarios lo pueden utilizar accediendo a un servidor Web a través de internet mediante un navegador, es decir permite la facilidad para mantener y actualizar el simulador físico sin necesidad de distribuirlo e instalarlo a miles de usuarios.

5.2 Conclusiones

El sistema realiza el proceso de experimentos de física en el tema de la Gravitación Universal.

El sistema de simulación permite al usuario comprobar las leyes del movimiento de los cuerpos celestes de una manera amena complementando el aprendizaje.

El sistema de simulación educativo ofrece un recurso para el proceso de aprendizaje.

El sistema automatiza el proceso de experimentación física, es decir, el usuario puede administrar y generar reportes de un experimento físico de la Gravitación Universal a través de una computadora con acceso a internet y con un navegador Web.

El sistema es robusto ya que si se presenta algún tipo de error en el sistema, se desplegará el mensaje apropiado del error, el cual permite que el sistema termine de manera controlada sin afectar los datos.

El sistema es portable ya que se desarrollo en una tecnología Java, el cual permite la portabilidad. El sistema puede ejecutarse en diferentes plataformas.

Las funcionalidades del sistema son configurables, tanto en la administración de la simulación y la generación de reportes como en la publicación del sistema en el servidor Web.

El sistema genera diversos reportes. Estos reportes muestran todos los datos de los objetos de la simulación en diferentes formas como gráficos o tabulaciones. Entre la información que se muestra de los objetos de la simulación son la fuerza vs tiempo, la velocidad vs tiempo y la posición vs tiempo.

El sistema utiliza una base de datos compacta, el cual permite almacenar los datos del experimento. La base de datos es una librería. La librería es un solo archivo binario almacenado en el servidor.

El sistema de simulación permite a través del almacenamiento de las condiciones iniciales, la reproducción exacta de los escenarios de simulación.

5.3 Recomendaciones Para Trabajos Futuros

Se recomienda que en futuros trabajos de desarrollo relacionados con los sistemas de simulación, estos puedan ser acoplados en un conjunto de simuladores con características similares, para poder obtener un sistema de simulación integrado y completo y que cubra una mayor cantidad de áreas de física o disciplinas relacionadas con ella.

El sistema de simulación puede ser integrado con sensores de captura de datos en tiempo real. Esto permite la comparación con los casos ideales de la simulación y los reales. Además con esta integración se obtiene una solución educativa más completa.

El sistema de simulación puede ser desarrollo para diferentes plataformas tales como Tablet PC, máquinas árcades, videoconsolas, PDAs o teléfonos inteligentes. Su desarrollo para pantallas táctiles es muy útil ya que son muy populares.

El sistema de simulación puede dejar de interactuar con dispositivos externos y permitir simular el experimento mediante las pantallas táctiles o sensores de presión en el dispositivo o el movimiento del propio dispositivo que puede recrear el movimiento de la simulación. Incluso se puede emplear la voz en dispositivos que soportan procesadores de voz.

El sistema de simulación puede hacer uso de otras maneras de proveer la interactividad e información al usuario, tal como el audio. El uso del audio es utilizado en todos los dispositivos. El sistema puede conectarse con dispositivos de reproducción de sonido tales como altavoces y auriculares.

Para mejorar la animación de la simulación se debe desarrollar el sistema con gráficos 3D. La animación es muy importante dentro de los gráficos de la simulación ya que con estas animaciones se intenta resaltar el realismo de los objetos. Se recomienda utilizar diversos APIs (Application Programming Interface) especializados en facilitar los procesos en cada una de las etapas de la generación de gráficos por computadora.

Además se puede utilizar el sonido digital y efectos visuales modernos que se utilizan en las últimas tecnologías en los motores de videojuego y las unidades de procesamiento gráfico.

6 Referencias

- [Newton 1687]
NEWTON, Isaac
1687 *Principios Matemáticos de la Física Natural*. Traducción de Antonio Escohotado. Barcelona: Ediciones Altaya S.A. ISBN: 8448701402.
- [Serway 2000]
SERWAY, Raymond A., BEICHNER, Robert J.
2000 *Physics for Scientists and Engineers*. Quinta Edición. Brooks / Cole.
ISBN: 9780030226540
- [Auris Rodríguez 2001]
AURIS RODRIGUEZ, Rosa Lucero
2001 *Simulador de Física – Oscilaciones*. Tesis de licenciatura en Ciencias e Ingeniería con mención en Ingeniería Informática: Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería.
- [Del Mar Zarate 2001]
DEL MAR ZARATE, Christian
2001 *Simulador de Física – Ley de Gravitación Universal*. Tesis de licenciatura en Ciencias e Ingeniería con mención en Ingeniería Informática: Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería.
- [Prado Lau 2001]
PRADO LAU, Cinthia Virginia
2001 *Simulador de Física – Electrostática*. Tesis de licenciatura en Ciencias e Ingeniería con mención en Ingeniería Informática: Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería.
- [Rodríguez Castillo 2001]
RODRIGUEZ CASTILLO, Hugo Maximiliano
2001 *Simulador de Física – Cantidad de Movimiento y Colisiones*. Tesis de licenciatura en Ciencias e Ingeniería con mención en Ingeniería Informática: Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería.

- [IFPUG 1999]
THE INTERNATIONAL FUNCTION POINT USER GROUP (IFPUG)
1999 *Function Point Counting Practices Manual*. The International Function Point User Group. Release 4.3.1. Consulta: 5 de octubre de 2011.
<http://www.ifpug.org/publications/CPM%204.3.1%20TOC%20Excerpts.pdf>
- [Ascher 1998]
ASCHER, URI M.; PETZOLD, LINDA RUTH
1998 *Computer methods for ordinary differential equations and differential-algebraic equations*. Primera Edición. Philadelphia: SIAM. ISBN: 0898714125.
- [PMI 2009]
PROJECT MANAGEMENT INSTITUTE (PMI)
2009 *Guía de los fundamentos de la dirección de proyectos (Guía del PMBOK)*. Cuarta Edición. Pennsylvania: Project Management Institute, Inc. ISBN: 9781933890722.
- [Chin 2004]
CHIN, Gary
2004 *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*. New York: AMACOM. ISBN: 0814471765.
- [Sommerville 2006]
SOMMERVILLE, IAN
2006 *Software Engineering*. Octava Edición. Madrid: Addison Wesley Iberoamericana. ISBN: 9780321313799.
- [SQLite 2011]
SQLITE
2011 *"About SQLite"*. SQLite. Consulta: 5 de octubre de 2011.
<http://www.sqlite.org/about.html>
- [RAE 2011]
REAL ACADEMIA ESPAÑOLA
2011 *Diccionario de la lengua española*. Real Academia Española. Vigésima Segunda Edición. Consulta: 5 de octubre de 2011.
<http://drae.rae.es>