



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL  
DEPARTAMENTO DE COMPUTACIÓN

# **Modelo de Vibración y Simulación de Comportamiento Dinámico de “Riser”, con Interacción Agua, Estructura y Suelo**

Tesis que presenta:

**Julián Edgardo Guzmán Cortés**

Para obtener el grado de

**Maestro en Ciencias en Computación**

Director de la Tesis: **Dr. Sergio Víctor Chapa Vergara**

Codirector de la Tesis: **Dr. Amilcar Meneses Viveros**

México, D. F.

Febrero, 2010



---

## Agradecimientos

A lo largo de este trabajo de investigación y durante mis estudios de maestría, he sido afortunado de contar con el apoyo de diversas personas, a las cuales quisiera agradecer por impulsarme a cumplir una más de mis metas.

En primer lugar, quisiera agradecer a *Dios* por permitirme llegar a este momento tan importante de mi vida profesional.

A *mis padres* que siempre me han apoyado incondicionalmente en mis decisiones y me alientan para seguir adelante. Gracias por su amor, comprensión y apoyo que me brindaron durante este tiempo que estuve lejos de casa.

A *mis abuelitos* que me brindan su cariño y siempre están al pendiente de mis pasos.

A *mis tíos* que siempre me escuchan y están presentes cuando necesito su ayuda. Gracias por sus consejos que me han ayudado a tomar las mejores decisiones.

A todos *mis viejos y nuevos amigos* que conocí en el CIVESTAV. Gracias por compartir tantas aventuras y desveladas de estudio durante este tiempo.

A *Pamela Sánchez* por apoyarme y tolerarme en los momentos difíciles. Gracias por permitirme compartir contigo tantos momentos y por hacer que la estancia en la escuela, lejos de mi familia, fuera menos difícil.

A mi asesor, el *Dr. Sergio V. Chapa Vergara* le agradezco la oportunidad que me ofreció para trabajar con él. Gracias por todo el apoyo y su ayuda para poder realizar este trabajo de tesis. He aprendido mucho bajo su asesoría.

Al *Dr. Amilcar Meses Viveros* le agradezco su ayuda y el tiempo que me brindo durante el desarrollo de esta tesis. Gracias por la paciencia que tuvo para explicarme los temas que me eran desconocidos, pero sobretodo, gracias por su amistad.

A mis sinodales: *Dr. José Matías Alvarado* y *Dr. Oliver Schütze* gracias por su consejos, comentarios y sugerencias que me ayudaron a mejorar mi trabajo de investigación.

Quisiera agradecer a todos los doctores del departamento de computación, por todas sus enseñanza, consejos y por compartir sus conocimientos conmigo.

A *Sofía Reza* por brindarme su amistad y su ayuda para realizar todos los tramites que se requieren durante estancia de la maestría. Gracias por su trato tan gentil y por hacernos sentir como en casa.

Finalmente quisiera agradecer al *CINVESTAV* por permitirme realizar mis estudios de maestría en la institución y al *Consejo Nacional de Ciencia y Tecnología* (CONACyT) por el apoyo económico que me permitió solventar los gastos durante mis estudios.





# Contenido

Índice de Figuras	IX
Índice de Tablas	XII
Índice de Algoritmos	XIV
Resumen	1
Abstract	3
<b>1 Introducción</b>	<b>5</b>
<b>2 Antecedentes</b>	<b>9</b>
2.1 El petróleo en aguas profundas . . . . .	9
2.1.1 Importancia del petróleo . . . . .	9
2.1.2 El petróleo en México . . . . .	12
2.1.3 Exploración y explotación de petróleo en aguas profundas . . . . .	13
2.1.4 Retos y dificultades en la exploración y explotación en aguas profundas . . . . .	16
2.2 Desarrollo de campos petroleros en alta mar . . . . .	17
2.2.1 Sistema submarino . . . . .	17
2.2.2 Líneas de flujo, tuberías y sistemas <i>riser</i> . . . . .	19
2.2.3 Plataformas fijas y flotantes . . . . .	21
2.2.4 Sistema de procesamiento superior . . . . .	22
2.3 Modelos oceanográficos . . . . .	22
2.3.1 Base de datos oceanográfica . . . . .	23
2.3.2 Método geostrófico . . . . .	23
2.4 Simulación . . . . .	25
2.4.1 El proceso de simulación por computadoras . . . . .	26
2.5 Vibración . . . . .	27
2.5.1 Ejemplos de estructuras afectadas por el fenómeno de vibración . . . . .	28
2.6 Trabajos relacionados . . . . .	30
2.7 Resumen . . . . .	31

<b>3</b>	<b>Modelo del sistema</b>	<b>33</b>
3.1	Modelo físico del sistema . . . . .	34
3.2	Movimiento Armónico Simple (MAS) . . . . .	35
3.2.1	Sistema de un grado de libertad . . . . .	37
3.2.2	Sistema de varios grados de libertad . . . . .	37
3.3	Modelo matemático . . . . .	38
3.3.1	Modelo matemático con interacción a primeros vecinos . . . . .	39
3.3.2	Modelo matemático con interacción a segundos vecinos . . . . .	42
3.4	Resumen . . . . .	45
<b>4</b>	<b>Diseño del sistema</b>	<b>47</b>
4.1	Diseño Conceptual . . . . .	48
4.2	Modelo del <i>riser</i> (cadenas de vibración) . . . . .	49
4.2.1	Problema de valores propios . . . . .	50
4.2.2	Análisis de Fourier . . . . .	51
4.3	Modelo de corrientes basado en Monte Carlo . . . . .	52
4.3.1	Método Monte Carlo . . . . .	53
4.3.2	Distribución normal . . . . .	55
4.4	Visualización . . . . .	56
4.5	Resumen . . . . .	57
<b>5</b>	<b>Desarrollo e implementación del sistema</b>	<b>59</b>
5.1	Arquitectura general . . . . .	59
5.1.1	Plataforma del subsistema de cómputo de datos . . . . .	60
5.1.2	Plataforma del subsistema de visualización . . . . .	60
5.2	Desarrollo del subsistema de cómputo de datos . . . . .	61
5.2.1	Proceso para obtener las matrices de movimiento . . . . .	62
5.2.2	Proceso para obtener los valores y vectores propios . . . . .	63
5.2.3	Proceso para obtener modos normales de vibración . . . . .	64
5.2.4	Proceso para dividir el <i>riser</i> en segmentos . . . . .	65
5.2.5	Proceso para generar frecuencias aleatorias . . . . .	67
5.2.6	Proceso para calcular los modos aleatorios de vibración . . . . .	70
5.3	Desarrollo del subsistema para la interfaz gráfica . . . . .	72
5.3.1	Módulo de gráficas de resultados . . . . .	74
5.3.2	Módulo de animación de partículas 2D . . . . .	75
5.3.3	Módulo de visualización interactiva 3D . . . . .	77
5.4	Resumen . . . . .	78
<b>6</b>	<b>Validación y pruebas del simulador</b>	<b>79</b>
6.1	Descripción general del uso del simulador . . . . .	79
6.1.1	Aplicación de cómputo de datos . . . . .	80
6.1.2	Aplicación de visualización . . . . .	83
6.2	Diseño de la pruebas . . . . .	87

---

6.3	Resultados de las pruebas . . . . .	88
6.3.1	Prueba con 8 segmentos . . . . .	89
6.3.2	Prueba con 16 segmentos . . . . .	91
6.3.3	Prueba con 32 segmentos . . . . .	94
6.4	Comparación de resultados con otros modelos . . . . .	96
6.5	Resumen . . . . .	99
<b>7</b>	<b>Resultados, discusión, y conclusiones</b>	<b>101</b>
7.1	Resultados . . . . .	101
7.2	Discusión . . . . .	102
7.3	Conclusiones . . . . .	104
7.4	Trabajo futuro . . . . .	105
	<b>Bibliografía</b>	<b>106</b>
<b>A</b>	<b>Instalación del sistema</b>	<b>111</b>
A.1	Requerimientos de Instalación . . . . .	111
A.1.1	Requerimientos para el subsistema de Cómputo de Datos . . . . .	111
A.1.2	Requerimientos para el subsistema de Visualización . . . . .	112
A.2	Tutorial de compilación . . . . .	112
A.2.1	Subsistema de Cómputo de Datos . . . . .	112
A.2.2	Subsistema de Visualización . . . . .	113



# Lista de Figuras

2.1	Proceso de formación del petróleo . . . . .	10
2.2	Pozo de alquitrán . . . . .	10
2.3	Principales usos del petróleo . . . . .	11
2.4	Regiones de aguas profundas . . . . .	14
2.5	Superficie de las Aguas Profundas en el Golfo de México . . . . .	15
2.6	Procesos para desarrollar un proyecto en Aguas Profundas . . . . .	16
2.7	Elementos para el desarrollo de campos en alta mar . . . . .	18
2.8	Clasificación de sistemas submarinos . . . . .	19
2.9	Línea de flujo, tuberías y sistemas <i>riser</i> . . . . .	20
2.10	Tipos de sistema de <i>riser</i> para aguas profundas . . . . .	21
2.11	Tipos de plataformas que se utilizan en los campos petroleros . . . . .	22
2.12	Puente Tacoma . . . . .	28
2.13	Puente del Milenio, Londres. . . . .	28
2.14	Hotel Burj Al Arab, Dubai. . . . .	29
2.15	Tipos de <i>riser</i> utilizados en los campos de aguas profundas. . . . .	30
3.1	Representación del modelo físico. Tubería sumergida en agua dividida en $n$ partículas . . . . .	34
3.2	Movimiento del <i>riser</i> provocado por las corrientes marinas . . . . .	35
3.3	Gráfica de Movimiento Armónico Simple . . . . .	36
3.4	Sistema de un grado de libertad . . . . .	37
3.5	Transformación del modelo físico al modelo matemático . . . . .	38
3.6	Cadena de vibración con interacción a primeros vecinos . . . . .	39
3.7	Cadena de vibración con interacción a primeros vecinos con partículas imaginarias en los extremos . . . . .	40
3.8	Ecuaciones de movimiento con interacción a primeros vecinos . . . . .	41
3.9	Cadena de vibración con interacción a segundos vecinos . . . . .	43
3.10	Cadena de vibración con interacción a segundos vecinos con partículas imaginarias en los extremos . . . . .	44
4.1	División del simulador en dos subsistemas . . . . .	48
4.2	Módulos que componen el subsistema de cómputo de datos . . . . .	48
4.3	Módulos que componen el subsistema de Visualización . . . . .	49

4.4	Representación general del sistema . . . . .	49
4.5	División del <i>riser</i> en partículas y segmentos . . . . .	53
4.6	Generación de frecuencias aleatorias para cada partícula, con diferentes condiciones en cada segmento . . . . .	54
4.7	Campa de la distribución normal. . . . .	55
4.8	Distribución normal con rangos de $-1$ a $1$ . . . . .	56
5.1	Arquitectura general del sistema . . . . .	60
5.2	Diagrama de flujo de datos del Subsistema de Cómputo de Datos . . . . .	61
5.3	Asignación de partículas por segmento . . . . .	67
5.4	Cálculo de la superposición de frecuencias aleatorias . . . . .	70
5.5	Diagrama de clases del sistema de visualización . . . . .	73
5.6	Procesos del módulo de gráficas de resultados . . . . .	74
5.7	Simulación del módulo de gráficas . . . . .	75
5.8	Interfaz para generar las gráficas de las partículas . . . . .	75
5.9	Procesos del módulo de animación de partículas . . . . .	76
5.10	Animación de partículas . . . . .	77
5.11	Procesos del módulo de visualización interactiva . . . . .	77
5.12	Animación del sistema en 3D . . . . .	78
6.1	Pantalla de inicio de la aplicación de cómputo de datos . . . . .	80
6.2	Pantalla de error en el número de segmentos . . . . .	81
6.3	Pantalla de la creación del nombre del directorio . . . . .	81
6.4	Pantalla de matrices de movimiento con sus valores propios y vector propio . . . . .	82
6.5	Pantalla del cálculo de modos normales de vibración . . . . .	82
6.6	Pantalla de los datos de media y varianza de los segmentos . . . . .	83
6.7	Pantalla de inicio del subsistema de visualización . . . . .	84
6.8	Menús del subsistema de visualización . . . . .	84
6.9	Barra de herramientas del subsistema de visualización . . . . .	85
6.10	Pantalla para abrir archivo. . . . .	85
6.11	Pantalla de inicio de simulación . . . . .	86
6.12	Pantalla del proceso de simulación. . . . .	86
6.13	Pantalla del sistema que muestra la gráfica de los datos. . . . .	87
6.14	Distribución de partículas en el <i>riser</i> para las pruebas . . . . .	88
6.15	Archivos que se generaron de la prueba de 8 segmentos . . . . .	89
6.16	Resultado de la visualización de los modos aleatorios para primeros vecinos . . . . .	89
6.17	Resultado de la visualización de los modos aleatorios para segundos vecinos . . . . .	90
6.18	Resultado de la visualización de la superposición para primeros vecinos . . . . .	90
6.19	Resultado de la visualización de la superposición para segundos vecinos . . . . .	91
6.20	Archivos que se generaron de la prueba de 16 segmentos . . . . .	92
6.21	Visualización de los modos aleatorios con 16 segmentos para primeros vecinos . . . . .	92
6.22	Visualización de los modos aleatorios con 16 segmentos para segundos vecinos . . . . .	92
6.23	Visualización de la superposición con 16 segmentos para primeros vecinos . . . . .	93

---

6.24	Visualización de la superposición con 16 segmentos para segundos vecinos . .	93
6.25	Archivos que se generaron de la prueba de 32 segmentos . . . . .	94
6.26	Visualización de los modos aleatorios con 32 segmento para primeros vecinos .	95
6.27	Visualización de los modos aleatorios con 32 segmentos para segundos vecinos	95
6.28	Visualización de la superposición con 32 segmentos para primeros vecinos . .	96
6.29	Visualización de la superposición con 32 segmentos para segundos vecinos . .	96
6.30	Gráfica de comparación de frecuencias naturales del <i>riser</i> . . . . .	99





# Lista de Tablas

2.1	Principales países productores de petróleo en aguas profundas . . . . .	14
6.1	Datos de entrada para la ejecución de la prueba de 8 segmentos . . . . .	89
6.2	Resultados de los modos aleatorios de vibración. <i>Riser</i> con 8 segmentos. . . . .	90
6.3	Resultados de la superposición de los movimientos. <i>Riser</i> con 8 segmentos . . . . .	91
6.4	Datos de entrada para la ejecución de la prueba de 16 segmentos . . . . .	91
6.5	Resultados de los modos aleatorios de vibración para la prueba de 16 segmentos . . . . .	93
6.6	Resultados de la superposición de los movimientos. <i>Riser</i> con 16 segmentos . . . . .	94
6.7	Datos de entrada para la ejecución de la prueba de 32 segmentos . . . . .	94
6.8	Resultados de los modos aleatorios de vibración. <i>Riser</i> con 32 segmentos . . . . .	95
6.9	Resultados de la superposición de movimientos. <i>Riser</i> con 32 segmentos . . . . .	96
6.10	Parámetros del <i>riser</i> de ejemplo . . . . .	97
6.11	Parámetros del ejemplo para nuestro modelo . . . . .	98
6.12	Comparación de resultados con otros modelos . . . . .	98



# Lista de Algoritmos

1	Genera Matriz de Movimiento . . . . .	63
2	Modos Normales de Vibración . . . . .	65
3	División de Segmentos . . . . .	66
4	Frecuencias aleatorias con Monte Carlo . . . . .	68
5	Movimientos Aleatorios de Vibración . . . . .	71



## Resumen

Actualmente existen diversas formas y fuentes proveedoras de energía, y el petróleo es la más importante. Este energético proporciona el mayor porcentaje de energía que se consume en el mundo y se utiliza como materia prima para fabricar una gran variedad de productos de consumo cotidiano. Históricamente el crecimiento económico de los países industrializados depende directamente del abastecimiento de petróleo que tengan. Por esta razón, los principales países productores de petróleo se han dado a la tarea de realizar la exploración y explotación de petróleo en aguas profundas. Esto representa una gran oportunidad para nuestro país, debido al potencial de petróleo que podría existir en aguas profundas del Golfo de México.

Llevar a cabo el proceso de exploración y explotación de petróleo en aguas profundas no es sencillo, ya que se enfrentan retos y dificultades diferentes a los que se presentan al trabajar en aguas someras. Uno de los problemas más importantes, es que las estructuras que se utilizan para la extracción del petróleo sean capaces de resistir la interacción de las corrientes marinas.

En esta investigación se presenta el diseño y desarrollo de un simulador que ayuda a realizar un análisis de las vibraciones que se presentan en los *riser*, provocadas por la fuerza de las corrientes marinas. Este simulador permite construir un modelo representativo del sistema para conocer o predecir el comportamiento dinámico de las estructuras en diversas situaciones a partir de bases teóricas. Los resultados obtenidos nos ayudan a conocer los puntos críticos de la estructura y permiten tomar decisiones al momento de su construcción para hacerlas más flexibles y resistentes.

El desarrollo del simulador esta basado en un modelo matemático de cadenas de vibración. Además, cuenta con un modelo estocástico de corrientes basado en el método Monte Carlo que nos permite analizar situaciones complejas y que son difíciles de reproducir físicamente. El sistema se compone de dos subsistemas: el subsistema de cómputo de datos que se encarga de utilizar los modelos desarrollados para obtener los resultados del comportamiento del sistema. Por su parte, el subsistema de visualización utiliza los datos obtenidos para generar la simulación gráfica que nos permita interpretar más fácilmente los resultados.

Finalmente el simulador fue sometido a diversas pruebas variando las condiciones iniciales del sistema. De esta manera, se verificó que tiene una precisión adecuada y se garantiza que los resultados estan apegados al modelo matemático.



## Abstract

Nowadays there are various sources and forms of energy suppliers, and oil is the most important one. Oil provides the highest percentage of energy consumed in the world and is used as a raw material to manufacture a wide variety of daily consumed products. Historically the economical growth of industrialized countries depends directly on its oil supply. For this reason, the major oil producers countries have been dealing with the task of performing the exploration and exploitation of oil in deepwater. This represents a great opportunity for our country, based on the possibility of oil existence in deepwater of Gulf of Mexico.

The process of exploration and exploitation of oil in deepwater is not simple, mostly because different conditions founded in contrast to the usual one when working in shallow water. One of the most important problems is, the characteristics of resistance related to the structures used for oil extraction against the interaction of ocean currents.

This research introduces the desing and development of a simulator that helps to analyze the vibrations in the *riser*, caused by the forces of ocean current. This simulator allows the construction of a representative model of the system in order to know or predic the dynamic behavior of the structures in diverse situations from theoretical bases. The results help us to know the critical points of the structure allowing us to take decisions at their construction time to make them more flexible and resistant

The development of the simulator is based on a mathematical model of strings vibration. Moreover, it has a stochastic model of currents based on the Monte Calo method, whole allowing us to analyze complex situations which are difficult to reproduce physically. The system consist of two subsystems: The computing subsystem which is responsible for the use of the models developed to get the system behavior results. Moreover, The display subsystem uses the data to generate graphical simulation allowing us more easily interpret the results.

Finally, the simulator was subject to several test varying the initial conditions of the system. Thus, we verified an adequate precision of the simulator ensuring that the results are related to the mathematical model.





# Capítulo 1

## Introducción

No existe en el mundo nada más  
poderoso que una idea a la que le ha  
llegado su tiempo

---

Victor Hugo

**E**n nuestros días, el petróleo se ha convertido en el principal energético de la humanidad. Actualmente su uso se ha extendido y para satisfacer la demanda, su producción va en aumento. Debido a que este energético es un recurso no renovable, algunos países han buscado alternativas para tener una mayor producción de petróleo. Se ha descubierto que las mayores reservas petrolíferas se encuentran en zonas conocidas como aguas profundas.

Las aguas profundas representan una gran oportunidad para nuestro país. En México las aguas profundas se encuentran ubicadas en las aguas territoriales del Golfo de México. En esta zona a partir de información geológica y geofísica, PEMEX ha identificado la existencia de estructuras geológicas capaces de contener hidrocarburos.

Sin embargo, el reto de explorar y explotar yacimientos de petróleo en aguas profundas no es sencillo, ya que es totalmente diferente a realizarlo en tierra o cerca de las costas (aguas someras) como actualmente se hace en nuestro país. Este es un proceso largo, complicado y muy costoso; ya que se tiene que contar con el financiamiento, la infraestructura, y los conocimientos necesarios para llevarlo a cabo. Trabajar en estas condiciones lleva a las empresas exploradoras a enfrentarse a nuevos retos como son: las corrientes marinas, los cambios de temperatura entre la superficie y el lecho marino, las presiones anormales al momento de la perforación, la geometría de los yacimientos, la operación remota a través de robots y desarrollar nueva tecnología que a países pioneros en el proceso de explotación de yacimientos en aguas profundas (Estados Unidos, Brasil, Noruega, etc.), les ha tomado varios años de desarrollo.

En nuestro país, el lapso del tiempo de estudio que toma decidir excavar un nuevo pozo es de entre 12 a 18 meses aproximadamente, lo que repercute en los costos de operación del proyecto [1]. Durante este periodo se hacen diversos estudios, pruebas y

simulaciones sobre cada una de las partes que conforman el proyecto de explorar y explotar yacimientos de crudo. El tener un control o una predicción sobre las situaciones de riesgo, agilizar este proceso y disminuir el tiempo para la puesta en producción de los campos petroleros, permite aumentar considerablemente las utilidades.

En esta investigación abordamos el problema de poder predecir el comportamiento dinámico de los *riser* (estructuras mecánicas que se utilizan para la explotación de petróleo en aguas profundas), cuyas estructuras se ven afectadas directamente por las corrientes marinas.

Las diversas condiciones de fuerzas que las corrientes marinas ejercen sobre las estructuras producen un movimiento continuo, lo que ocasiona que los *riser* presenten vibraciones a lo largo de su cuerpo de metal. Poder predecir esta clase de comportamiento es de gran importancia, ya que si la frecuencia del movimiento provocado por las corrientes se encuentra cerca de la frecuencia natural de la estructura, se puede caer en Resonancia [2]. Este fenómeno produce desgastes y deformaciones que provocan daños en la estructura, que en el peor de los casos pueden ocasionar accidentes.

Cuando se trabaja con sistemas en los cuales se tienen condiciones que pueden llegar a ser impredecibles (como es el caso de las corrientes marinas); se tiene un alto índice de riesgos. Actualmente se cuenta con herramientas como simuladores y modelos matemáticos de vibración que nos pueden auxiliar en el estudio del comportamiento de las estructuras mecánicas. Sin embargo, este tipo de tecnologías necesarias para la exploración y explotación de yacimientos en aguas profundas actualmente son muy costosas, debido a que son escasas y su disponibilidad en los mercados es limitada.

El desarrollo de simuladores estocásticos que utilicen métodos numéricos y nuevos algoritmos de computación, son de las herramientas principales que pueden ayudar a solucionar diversos problemas en el proceso de exploración y explotación de petróleo en aguas profundas, específicamente predecir el comportamiento dinámico de los *riser* en operación, provocado por las fuertes corrientes marinas.

El objetivo principal de este trabajo de investigación es presentar el diseño y desarrollo de un simulador de vibraciones con un modelo estocástico de corrientes, que nos ayude a predecir el comportamiento dinámico de los *riser*, tomando en consideración la interacción con su ambiente (Agua, Estructura y Suelo). Para su desarrollo se siguieron los siguientes pasos:

**Modelo físico del sistema:** El modelo físico del sistema se presenta como una tubería de metal sumergida en el mar, sobre la cual la fuerza de las corrientes marinas producen vibraciones. Esta tubería se divide en partículas de masa equidistantes sobre las cuales se analiza la fuerza que ejercen las corrientes marinas. El sistema se representa como una cadena de vibración unidimensional de  $n$  grados de libertad formada por una serie de partículas y unidas por medio de resortes.

**Modelo Matemático:** El modelo físico se transformó a un modelo matemático de valores propios, basado en Movimiento Armónico Simple homogéneo MAS [3]. Este modelo permite obtener las matrices de movimiento para la interacción de partículas de masa del *riser* con interacción a primeros y segundos vecinos. MAS sirve como base para nuestro simulador y nos proporciona confiabilidad en nuestros resultados [4]. La

solución del modelo matemático se obtiene de forma numérica. Para ello, se usan bibliotecas computacionales que nos ofrecen soluciones con diversos métodos como el de Kramer, el método de Jacobi, algoritmo QR, etc. Particularmente se utilizó la biblioteca IMSL (FORTRAN Numerical Library) la cual, es una biblioteca computacional con algoritmos matemáticos y estadísticos para lenguaje FORTRAN. La biblioteca recibe como entrada la matriz de movimiento de nuestro modelo y nos ofrece como resultado las frecuencias (valores propios) y las Amplitudes (vectores propios) correspondientes.

**Generación de Frecuencias Aleatorias:** Para la reproducción de fenómenos naturales (como las corrientes marinas), es necesario la simulación de números aleatorios. De esta manera, el simulador cuenta con la implementación de un método numérico no determinístico conocido como Monte Carlo (basado en una distribución normal), el cual nos ayuda a generar las condiciones de fuerzas de excitación y frecuencias que afectan a las estructuras mecánicas. Gracias a esto, vamos a poder analizar situaciones que son muy costosas y difíciles de producir físicamente.

**Análisis de coeficientes de Fourier:** Para obtener el comportamiento general de nuestro sistema, calculamos la superposición de las ondas que se obtuvieron como resultado de nuestro modelo matemático en cada uno de los nodos que conforman el *riser*.

**Desarrollo del sistema:** El sistema se encarga de realizar el cálculo de las vibraciones en la estructura para posteriormente desplegarlos en una simulación gráfica. El desarrollo del sistema se encuentra dividido en 2 módulos principales:

1. Módulo de cómputo de datos: El módulo principal se encarga de recibir los valores de entrada para nuestro simulador (longitud del *riser*, masa, elasticidad, número de partículas, etc). Con estos valores se generan las matrices de movimiento correspondientes a los modelos de primeros y segundos vecinos. Además, el módulo se encarga de generar las frecuencias aleatorias (por medio del método de Monte Carlo) y realizar los cálculos correspondientes para obtener el comportamiento de nuestro sistema. Los resultados se almacenan en archivos que posteriormente son utilizados en el módulo de la Interfaz Gráfica para desplegar la simulación. Esta aplicación fué desarrollada en lenguaje FORTRAN (F90) y funciona en plataformas tipo Unix.
2. Interfaz Gráfica: Este módulo es una aplicación totalmente independiente a la del módulo principal, es decir, que la representación gráfica de los resultados no se realiza en tiempo real. Esta aplicación hace uso de los datos generados por el módulo principal y despliega los resultados en base a los datos contenidos en ellos. La interfaz gráfica es amigable para el usuario, además permite que los resultados se puedan interpretar fácil y rápidamente. La interfaz gráfica se encuentra dividida en tres secciones: La primer sección es una simulación que nos indica el comportamiento individual de cada uno de los nodos que conforman el sistema. La segunda sección nos genera las gráficas correspondientes a cada uno de los nodos por separado. Y la tercer sección nos ofrece una simulación en 3D de nuestro sistema. El desarrollo de la interfaz gráfica está escrita en lenguaje C++

con ayuda de la biblioteca QT para el desarrollo de la interfaz gráfica, además se usa OpenGL para la simulación 3D.

Para verificar los resultados y la buena precisión del simulador, se realizaron diversas pruebas variando las condiciones iniciales del sistema. Estos resultados nos permitieron determinar que nuestro simulador se encuentra apegado a su modelo matemático y que su margen de error es despreciable. Sin embargo, para comprobar totalmente nuestro simulador es necesario llevar a cabo una prueba física, la cual es conocida como "*Prueba de Tanque*"; desafortunadamente este tipo de pruebas requieren de equipo especializado y son demasiado costosas, por lo que esta prueba queda fuera del alcance del trabajo de investigación.

El presente documento se encuentra organizado de tal forma que el lector pueda seguir cada uno de los pasos que fueron necesarios para el diseño y desarrollo de este simulador. La información se encuentra organizada en 7 capítulos que se describen a continuación.

En el Capítulo 2: "*Antecedentes*" se ofrece un panorama general sobre la situación actual de la exploración y explotación de petróleo en aguas profundas, así como las técnicas y tecnologías que se usan actualmente para realizar ese proceso. Además, se describe la importancia de la simulación en este tipo de sistemas y se presentan algunos trabajos relacionados.

El Capítulo 3: "*Modelo del sistema*" describe la representación del modelo físico del sistema. Se construye el modelo matemático con interacción a primeros y segundos vecinos para finalmente obtener nuestras matrices de movimiento.

En el Capítulo 4: "*Diseño del simulador*" se describen los principales módulos que forman el sistema y las metodologías que se utilizaron para desarrollar cada uno de ellos.

En el Capítulo 5: "*Desarrollo e implementación*" se muestra la arquitectura general de nuestro simulador y se detallan los módulos de programación que se realizaron durante su desarrollo.

En el Capítulo 6: "*Validación y pruebas*" se define la selección de pruebas que se realizaron para validar los resultados de nuestro simulador y se hace una comparación de resultados con otros modelos.

En el Capítulo 7: "*Resultados, discusión y conclusiones*" se describen los resultados obtenidos durante este trabajo de investigación y se hace una discusión final haciendo referencia a los resultados que se obtuvieron durante la etapa de pruebas. Además, se describen las conclusiones a las que se llegaron y se presenta una lista sobre los trabajos futuros para mejorar el simulador.

Finalmente, el documento presenta un anexo en donde se detallan los requerimientos necesarios para instalar y compilar los subsistemas que forman el simulador.

# Capítulo 2

## Antecedentes

Vale más saber alguna cosa de todo,  
que saberlo todo de una sola cosa.

---

Blaise Pascal

Para desarrollar este trabajo de tesis, se realizó una investigación sobre los antecedentes, y el proceso general de exploración y explotación de petróleo.

En este capítulo se abordan temas referentes a la importancia del petróleo en el mundo y especialmente en México. Además, se analizan las técnicas actuales que se utilizan para explorar y explotar yacimientos de petróleo en aguas profundas y los problemas a los que se enfrentan las empresas al realizar este proceso.

Por otra parte, en este capítulo se describe el concepto de simulación por computadora y la forma en como nos apoya para enfrentar problemas como las vibraciones que se presentan en los *riser*. Finalmente se mencionan algunos desarrollos de software que se utilizan en la industria petrolera para administrar y monitorear los sistemas *riser*.

### 2.1 El petróleo en aguas profundas

#### 2.1.1 Importancia del petróleo

El petróleo es una forma de energía química potencial [5]. Es líquido de origen natural compuesto por una mezcla de hidrocarburos. La formación del petróleo se debe al desarrollo de rocas sedimentarias depositadas en ambientes marinos o próximos al mar, y es el resultado de un proceso de descomposición de organismos de origen vegetal y animal, que en tiempos remotos quedaron incorporados en esos depósitos (Fig. 2.1 <sup>1</sup>) [6].

---

<sup>1</sup>Imagen tomada de sitio <http://www.telefonica.net/web2/marcillaot/>

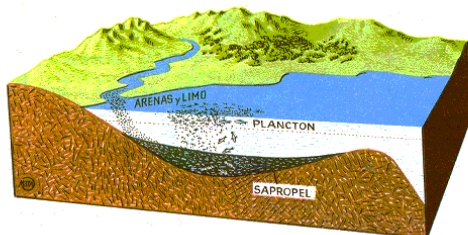


Figura 2.1: Proceso de formación del petróleo

El nombre de petróleo proviene del término griego que significa “Aceite de Roca” (Petro: Roca y Oleum: aceite) y se utilizó por primera vez en el tratado *De Natura Fossilium* por el mineralogista Geor Bauer (conocido también como Georgius Agricola)[7].

No se sabe exactamente cuándo fue utilizado por primera vez el petróleo por la humanidad, sin embargo, las primeras civilizaciones supieron reconocer y aprovechar la utilidad de éste energético. Se tiene conocimiento que se usaba el asfalto (una forma muy viscosa de petróleo) para la construcción de barcos en los años 6000 a. de C. y en la construcción de las pirámides de Egipto alrededor del año 3000 a. de C. [8]. Inclusive, en algunos relatos bíblicos se hace mención al uso de brea o asfalto durante la construcción de la torre de babel y por Noé para la construcción del arca.

En un principio las únicas fuentes de petróleo que se conocían eran los pozos de alquitrán (Fig. 2.2 <sup>2</sup>), sin embargo estos no eran muy productivos. Debido al aumento de la demanda del petróleo, se tuvieron que buscar nuevas fuentes proveedoras de dicho combustible. En el año de 1859 el coronel Edwin L. Drake realizó el descubrimiento y la perforación del primer pozo para extraer petróleo cerca de Titusville, Pensilvania[5]; siendo el inicio de la industria petrolera en el mundo.



Figura 2.2: Pozo de alquitrán

A partir de los descubrimientos de la década de 1850, se comenzaron a construir numerosas refinerías para convertir el petróleo en queroseno y aceites lubricantes que se

<sup>2</sup>Imagen tomada de <http://blog.webosaurs.com/2009/10/page/2/>

utilizaban en lámparas y en las máquinas de la revolución industrial. No obstante, la importancia del petróleo para la humanidad comenzó cuando esta fuente de energía reemplazó al carbón como principal combustible para las máquinas que se utilizaban durante esa época, esto debido principalmente a que el petróleo es una fuente de energía más versátil y potente que el carbón.

Con la aparición de los primeros automóviles, gracias a la invención del motor de gasolina (1885) y el motor de diesel (1892) [8], aumentó la demanda de este tipo de combustibles y propició que esta industria fuera avanzando cada vez más, hasta convertirse en un elemento primordial para el desarrollo económico e industrial.

Actualmente, el petróleo es la fuente de energía más importante y se ha convertido en una de las materias primas de mayor importancia para la humanidad. Es el recurso que proporciona el mayor porcentaje del total de la energía que se consume en todo el mundo. Alrededor del 40% del total del suministro de energía a nivel mundial es suministrada por el petróleo [9]. Por su parte el transporte depende de un 90% del petróleo (ya sea transporte terrestre, marítimo o aéreo). Históricamente el consumo mundial de petróleo ha seguido muy de cerca la trayectoria del crecimiento del producto interno bruto de muchos países. Por lo que el crecimiento económico de los países industrializados va acompañado por un consumo creciente de petróleo. Se estima que la demanda de petróleo seguirá creciendo más rápidamente en los países con mayor desarrollo. Por lo tanto, la economía y el estilo de vida de las sociedades industrializadas, dependen en gran medida del suministro suficiente de petróleo que se tenga así como de su precio [10].

Aunque el petróleo es la principal fuente de energía, también se utiliza como materia prima en la fabricación una gran variedad de productos, entre los que encontramos: plásticos, pinturas, fertilizantes, detergentes, cosméticos, insecticidas e inclusive algunas grasas y complementos alimenticios (Fig. 2.3 <sup>3</sup>).

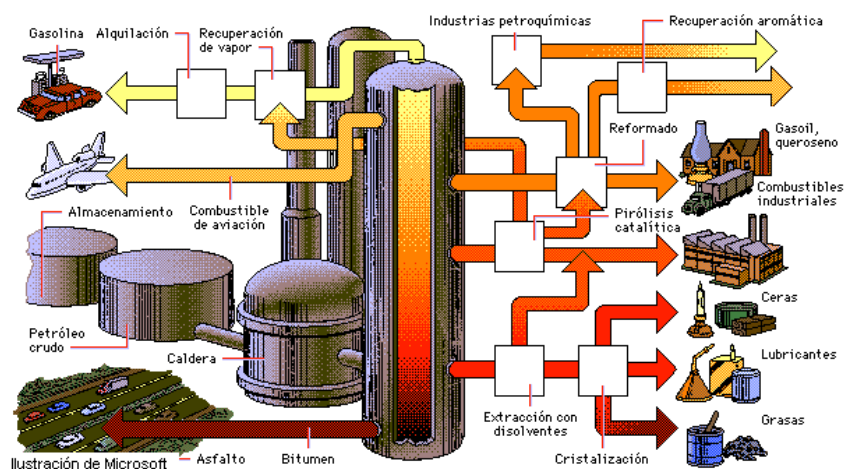


Figura 2.3: Principales usos del petróleo

<sup>3</sup>Imagen tomada del sitio <http://atinachile.bligoo.com/content/view/22576>

Debido a que este energético es un recurso no renovable y a la dificultad de sustituir el petróleo por otras fuentes fósiles de energía, algunos países han buscado alternativas para tener una mayor producción de petróleo. Se ha descubierto que las mayores reservas petrolíferas se encuentran en zonas conocidas como *aguas profundas*, hasta las que hay que descender para poder extraer el petróleo.[11]

### 2.1.2 El petróleo en México

En México se tiene conocimiento del petróleo desde la época prehispánica en la que los indígenas le daban diversos usos a los depósitos de "*chapotli*" en los cuales el petróleo acumulado era utilizado con fines medicinales y para la elaboración de figurillas de arcilla, impermeabilizante, material de construcción, etc. [12].

A pesar de los abundantes depósitos de betún negro (conocidos como chapoteras) en la zona del Golfo de México, durante el periodo de la época colonial los españoles no le dieron un uso práctico ya que cuando se presentaba en la superficie de la tierra era visto como un elemento nocivo para las actividades agrícolas y ganaderas [13].

Por otra parte, la historia comercial comienza hasta 1863 cuando el cura Manuel Gil y Sáenz descubrió un pozo superficial llamado "*Mina de Petróleo de San Fernando*", en el estado de Tabasco. Durante esa época el Emperador Maximiliano intentó promover las actividades petroleras mediante el otorgamiento de concesiones para la explotación de depósitos naturales.

Sin embargo, la etapa de producción de petróleo a gran escala en México, comienza a partir de 1901 con el inicio de las operaciones de la empresa *Mexican Petroleum* [12]. En el mismo año el Ing. Ezequiel Ordóñez descubre un yacimiento petrolero en el campo de el Ébano en San Luis Potosí que fue llamado "*El Pez*". De esta manera inicia de forma organizada la exploración petrolera en México [14]. En un principio no se tuvo mucho éxito en la perforación de varios pozos. Fue hasta el año de 1904 cuando la empresa *Mexican Petroleum* tuvo el primer gran pozo productivo.

Para 1920 había alrededor de 80 compañías productoras de petróleo en el país y 17 compañías exportadoras, en las cuales el mayor porcentaje de su capital era angloamericano. Para fomentar la inversión de capital nacional en la industria del petróleo, el gobierno creó *Petróleos de México*. El 18 de marzo, el presidente Lázaro Cárdenas decreta la disponibilidad de México para indemnizar a las compañías petroleras sus inversiones. De esta manera se crea *Petróleos Mexicanos (PEMEX)* como organismo encargado de la administración y explotación de los hidrocarburos [1].

El petróleo ha sido un factor determinante en México ya que ha influido directamente en el desarrollo de su economía, las finanzas públicas, la industria, el desarrollo tecnológico, la balanza comercial y en las relaciones exteriores que mantiene con otros países, principalmente con Estados Unidos [14].

Actualmente la actividad de exploración en México se ha localizado en dos regiones [15]:

1. Exploración en tierra: Esta exploración se realizaba a principios del siglo pasado en lugares conocidos como chapoteras.



2. Exploración en aguas someras: Esta etapa de exploración se hizo a finales del siglo pasado en la región de Campeche, y en las costas de Veracruz y Tamaulipas.

Actualmente PEMEX ha descubierto la oportunidad de realizar la exploración y explotación de yacimientos de petróleo en aguas profundas. Las aguas profundas representan una gran oportunidad para el país, principalmente por el volumen de petróleo que se estima podría existir.

En el golfo de México se han descubierto la existencia de estructuras geológicas capaces de contener hidrocarburos. Los trabajos de exploración realizados por Petróleos Mexicanos han permitido obtener información geológica y geofísica diversa para estimar el potencial e identificar las principales cuencas petroleras que pueden existir en esa región.

### 2.1.3 Exploración y explotación de petróleo en aguas profundas

El concepto de ingeniería *offshore* o mar adentro, surge en el siglo pasado (en la década de lo 60's) en el continente europeo. Sin embargo, por los elevados costos que se requerían para generarla, no tuvo un gran desarrollo. En la década de los 70's debido al conflicto bélico árabe-israelí, queda al descubierto la importancia que tiene el mercado del petróleo en el desarrollo económico en el mundo [16]. Esta situación provocó el aumento de precio del petróleo a nivel mundial, con lo cual se pudieron sufragar los altos costos de la tecnología requerida para explorar y explotar nuevos yacimientos ubicados en el fondo del mar. De esta forma la ingeniería *offshore* se desarrolló exitosamente, principalmente por la participación y las capacidades de los centros de investigación y la asociación entre universidades y empresas.

A mediados de la década de los 80's, la actividad de exploración de campos de desarrollo se desplazó a las aguas profundas [17]. Principalmente porque en esta época se descubrieron grandes cantidades de petróleo en el Golfo de México; lo que generó la necesidad de nuevas estrategias de exploración y aplicar nuevas tecnologías e infraestructura más especializada.

Según el *Minerals Management Service* (MMS) de Estados Unidos, las aguas profundas significan distancias mayores a 305 mts. (aproximadamente 1000 pies) hacia el fondo del mar[18]. En nuestro país, la exploración y explotación de petróleo en aguas profundas se refiere a regiones ubicadas en tirantes de agua mayores a 500 metros (distancia entre la superficie y el lecho marino, figura 2.4<sup>4</sup>)[1].

---

<sup>4</sup>Imagen tomada del sitio <http://www.pemex.com> [1]

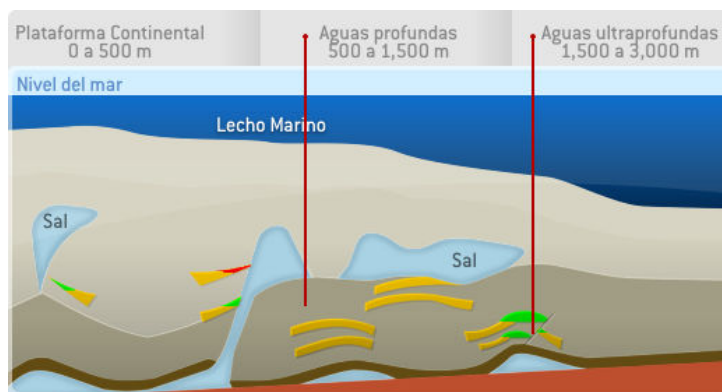


Figura 2.4: Regiones de aguas profundas

Hoy en día, el desarrollo de yacimientos en aguas profundas está dejando atrás el desarrollo de yacimientos en tierra y en aguas poco profundas como se venía haciendo en muchos países hace algunos años. Las principales razones de esto son [18]:

- Las fuentes de petróleo en tierra o en aguas someras son muy limitadas.
- Las reservas que se tienen en aguas profundas son mucho más productivas (Aproximadamente 20 veces más en petróleo y 8 veces más en gas)
- Es necesario realizar una mayor inversión en investigación, pero se tienen más retribuciones.
- Se mejora y se desarrolla nueva tecnología.

Debido al potencial de petróleo que se estima puede haber en aguas profundas, los países se han dado a la tarea de buscar nuevas fuentes proveedoras en esos campos. La tabla 2.1 muestra los principales países productores de petróleo en aguas profundas [1].

Principales Países
1.- Brasil
2.- Nigeria
3.- Estados Unidos
4.- Angola
5.- Egipto
6.- Gran Bretaña
7.- Noruega
8.- India
9.- Filipinas
10.- Guinea Ecuatorial

Tabla 2.1: Principales países productores de petróleo en aguas profundas

En México las aguas profundas se encuentran en una extensión de 575 mil kilómetros cuadrados ubicada en las aguas territoriales del Golfo de México (fig 2.5<sup>5</sup>). En esta zona, PEMEX ha identificado la existencia de estructuras geológicas que pudiesen contener hidrocarburos. Esto representa una gran oportunidad para el país, debido a la gran cantidad de petróleo que puede existir en esta zona de aguas profundas. Sin embargo, algunos de estos yacimientos son transfronterizos, es decir, que geográficamente se localizan en las fronteras de dos o más países. Varios de estos campos se encuentran localizados en la frontera con Estados Unidos en donde ya se tiene un tiempo de exploración de 15 años y que en los últimos 10 años perforan alrededor de 100 pozos por año [1]. Por esta razón, es urgente tomar medidas para la exploración de dichos campos ya que de no hacerlo se producirá un efecto “popote” y se verían afectados los campos que están de lado mexicano.



Figura 2.5: Superficie de las Aguas Profundas en el Golfo de México

Para el MMS de Estados Unidos, el Golfo de México se ha convertido en su mayor proveedor de gas y petróleo. El desarrollo de recursos ha convertido al Golfo de México en el principal punto de exploración y producción de gas y petróleo a nivel mundial. Con la obtención de petróleo en aguas profundas del Golfo de México, el gobierno americano espera que la producción se duplique para la próxima década, es decir, que para el 2011 se obtenga casi el 80% de la producción total [19].

Para el gobierno de los Estados Unidos, las aguas profundas representan un papel fundamental en el futuro de la energía de esa nación. Actualmente cuentan con un gran inventario de contratos de arrendamiento disponibles para la industria de exploración. Esto genera que la tecnología siga avanzando, a fin de facilitar las actividades en las situaciones extremas de las aguas profundas. Todos estos factores, aseguran que las aguas profundas del Golfo de México sigan siendo uno de los principales campos de desarrollo en el mundo [20].

<sup>5</sup>Imagen tomada del sitio <http://www.pemex.com> [1]

### 2.1.4 Retos y dificultades en la exploración y explotación en aguas profundas

El reto de explorar y explotar yacimientos de petróleo en aguas profundas no es sencillo, ya que es totalmente diferente a realizarlo en tierra o cerca de las costas como actualmente se hace en México. Estudios realizados por PEMEX han determinado que el tiempo que llevaría realizar un proyecto en aguas profundas es de aproximadamente 8 a 10 años, ya que se tiene que pasar por diversos procesos (Fig. 2.6<sup>6</sup>) [1].



Figura 2.6: Procesos para desarrollar un proyecto en Aguas Profundas

Cuando se explotan yacimientos en aguas profundas las empresas se enfrentan a riesgos geológicos, de operación y financieros. Estos riesgos producen grandes retos y dificultades que las empresas tienen que solucionar [1]:

1. **Corrientes marinas:** Estas fuerzas externas provocan el movimiento de las estructuras que producen vibraciones en las tuberías y desgastan los equipos de perforación.
2. **Cambios de temperatura:** Se presentan diferentes cambios de temperatura entre la superficie, el lecho marino y las zonas perforadas; que pueden complicar el bombeo del fluido. Además, las bajas temperaturas que se pueden presentar implican el uso de diversas técnicas y metodologías para que los fluidos puedan circular por las tuberías [17].
3. **Problemas de perforación:** Cuando se atraviesan formaciones someras, se pueden presentar flujos de agua de alta presión, flujos de gas y presiones anormales.
4. **Instalaciones submarina:** Después de la perforación, el pozo debe ser controlado por un sistema de producción. Así que se instala una estructura submarina en el fondo mar para su control. Esto requiere la aplicación de nuevas tecnologías, así como un alto grado de automatización.
5. **Operación remota:** Se requiere de la implementación de tecnología robótica que permita operar la instalación submarina desde el fondo del mar.

<sup>6</sup>Imagen tomada del sitio <http://www.pemex.com> [1]

6. **Altos costos:** Los campos son desarrollados con un menor número de pozos que en aguas someras, debido a que la inversión que se requiere en cada uno de ellos es muy elevada.
7. **Geometría de yacimientos:** En aguas profundas los yacimientos podrían ser diferente a los de aguas someras, lo que dificulta su exploración y explotación.
8. **Infraestructura:** Es necesario renovar toda la infraestructura con la que se cuenta actualmente y utilizar nuevas metodologías y tecnologías para llevar a cabo el desarrollo del proyecto.

Para poder trabajar bajo estas condiciones y enfrentar los retos que se presentan, es necesario utilizar nuevas tecnologías, las cuales a países como Estados Unidos, Brasil y Noruega les ha tomado alrededor de 15 años desarrollar. Adicionalmente también se requiere contar con los conocimientos y las habilidades necesarias para poder seleccionar correctamente las tecnologías, utilizarlas adecuadamente y resolver cualquier contratiempo que se pueda presentar.

## 2.2 Desarrollo de campos petroleros en alta mar

Como se ha mencionado anteriormente, el desarrollo de campos petrolíferos es un proceso complicado y muy costoso que requiere de la implementación de tecnología de vanguardia. Actualmente el desarrollo de yacimientos en alta mar esta formado por cuatro elementos principales [18](Fig. 2.7<sup>7</sup>)

1. Sistema Submarino.
2. Líneas de flujo, tuberías y riser.
3. Plataformas fijas o flotantes.
4. Sistema de procesamiento superior.

### 2.2.1 Sistema submarino

El sistema submarino es necesario para controlar el sistema de producción una vez que se ha completado el proceso de perforación. En campos desarrollados en tierra o en aguas someras se implementa un sistema llamado "*dry completion*" (finalización seca), que va desde la cabeza del pozo y se extiende hasta la parte superior de la plataforma. Sin embargo, en aguas profundas no es tan sencillo instalar este tipo de sistemas, por lo que se instala una estructura submarina que controla la cabeza del pozo en el fondo marino llama "*wet completion*" (Finalización húmeda) [17].

El sistema submarino se puede dividir en tres partes principales [18]:

---

<sup>7</sup>Imagen tomada del libro *Introduction to Offshore Pipelines and Risers* [18]

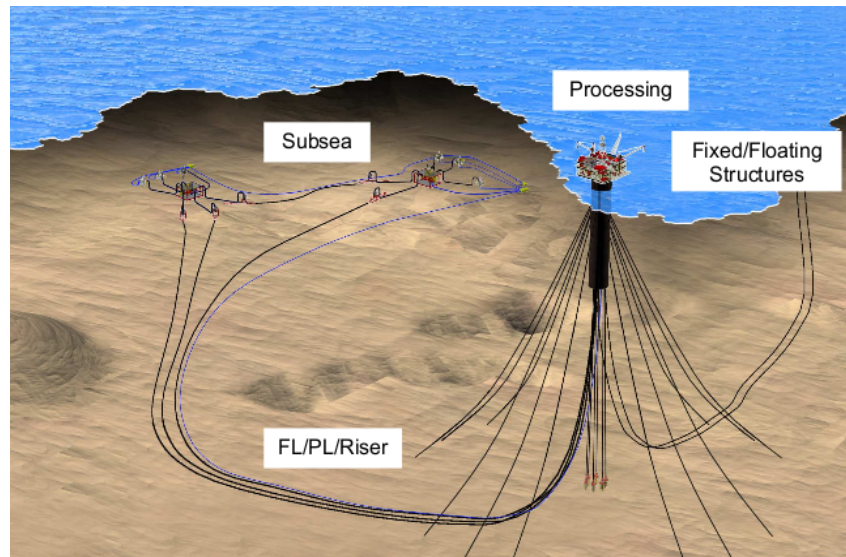


Figura 2.7: Elementos para el desarrollo de campos en alta mar

1. Estructura de la cabeza del pozo (árbol de navidad): Es una estructura de la parte superior del armazón de perforación, que se utiliza para montar un panel de control de válvulas, la cual tiene una forma de pino.
2. Sistema de control: Es un sistema para el control de bobinas, válvulas y monitorear la presión, temperatura y la posición de los sensores que se montan en sistema submarino.
3. Sistema de conexión: Se utiliza para realizar puentes y terminación de tuberías.

Los sistemas submarinos se clasifican por el número de pozos y los métodos de conexión de la siguiente manera [17] (Fig. 2.8<sup>8</sup>):

- *Single Well Tieback*: Es un pozo submarino que está conectado a la plataforma flotante de producción por una línea de flujo simple o dual.
- *Daisy Chain Tieback*: Pozos submarinos individuales que son unidos por una línea de flujo simple que forman un ciclo desde el pozo principal a los demás pozos.
- *Cluster Well Manifold*: Un colector es empleado en varios pozos submarinos individuales agrupados, generalmente en un centro de perforación común.
- *Multi-well Template*: Proporciona una estructura que facilita la operación de perforación desde un solo punto. Los puntos de producción se alcanzan mediante perforación direccional y todo el árbol de pozos está localizado sobre la estructura lo que permite un fácil acceso.

<sup>8</sup>Imagen tomada del artículo *Design and Installation of Deepwater Petroleum Pipelines* [17]



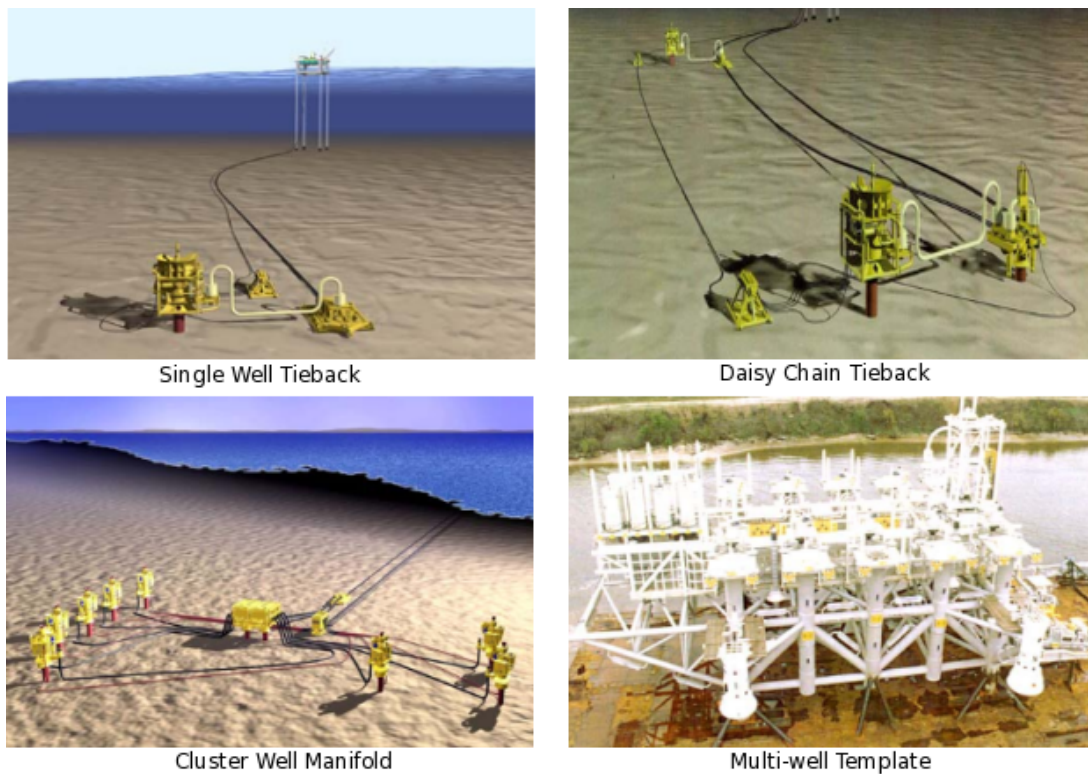


Figura 2.8: Clasificación de sistemas submarinos

## 2.2.2 Líneas de flujo, tuberías y sistemas *riser*

Las líneas de flujo también se denominan línea de producción o de importación y se encargan de transportar fluidos sin procesar (petróleo crudo o gas). Generalmente la transmisión de fluido puede llevar algunos sólidos como parafina y arena (Fig. 2.9<sup>9</sup>) [18].

Por su parte, el sistema de tuberías transporta petróleo y gas procesado, por lo que primero se realiza un proceso para separar el gas, el agua y algunos sólidos que vienen mezclados con el petróleo al momento de su extracción. En general el tamaño de las tuberías es mayor a las líneas de flujo, además presentan bajas temperaturas y baja presión. [18] (Fig. 2.9).

Un sistema *riser* generalmente es un tubo el cual conecta la tubería submarina al sistema de la plataforma flotante. En el desarrollo del campo de aguas profundas el *riser* puede estar hecho por una tubería rígida o flexible. El *riser* permite el acceso desde el fondo del mar a las instalaciones de la plataforma para el desarrollo de estaciones submarinas, tuberías para importar y exportar productos, y una plataforma de servicios (líneas eléctricas y de control) [21] (Fig. 2.9).

<sup>9</sup>Imagen tomada del libro *Introduction to Offshore Pipelines and Risers* [18]

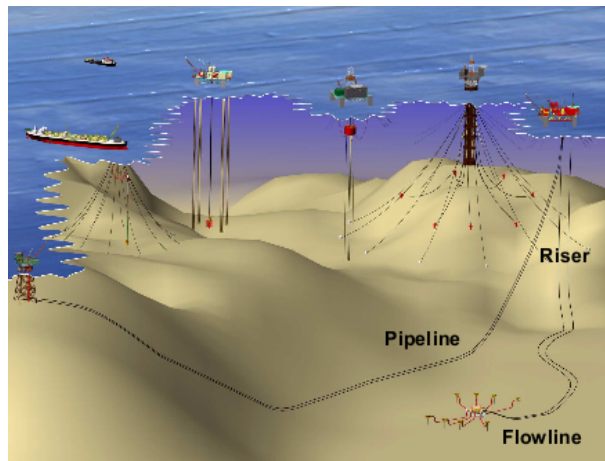


Figura 2.9: Línea de flujo, tuberías y sistemas *riser*

A continuación se describen los sistemas de *riser* que se tienen actualmente para aguas profundas, así como las características de cada uno de ellos [17] (Fig. 2.10<sup>10</sup>):

- *Flexible Riser*: Puede tener gran longitud, es robusto con respecto a la dinámica y tiene un bajo costo de instalación.
- *Steel Catenary Riser (SCR)*: Es un *riser* de acero que puede tener grandes diámetros y resiste altas temperaturas.
- *Top Tension Riser (TTR)*: *Riser* de alta tensión que puede contener largas trayectorias.
- *Tension Leg Riser (TLR)*: Este *riser* es la combinación de SCR y un sistema de *riser* flexible. Es considerado como una de las mejores opciones en cuanto al costo efectivo. Este tipo de sistemas pueden tener un buen aislamiento y menor carga en los buques.

<sup>10</sup>Imagen tomada del artículo *Design and Installation of Deepwater Petroleum Pipelines* [17]



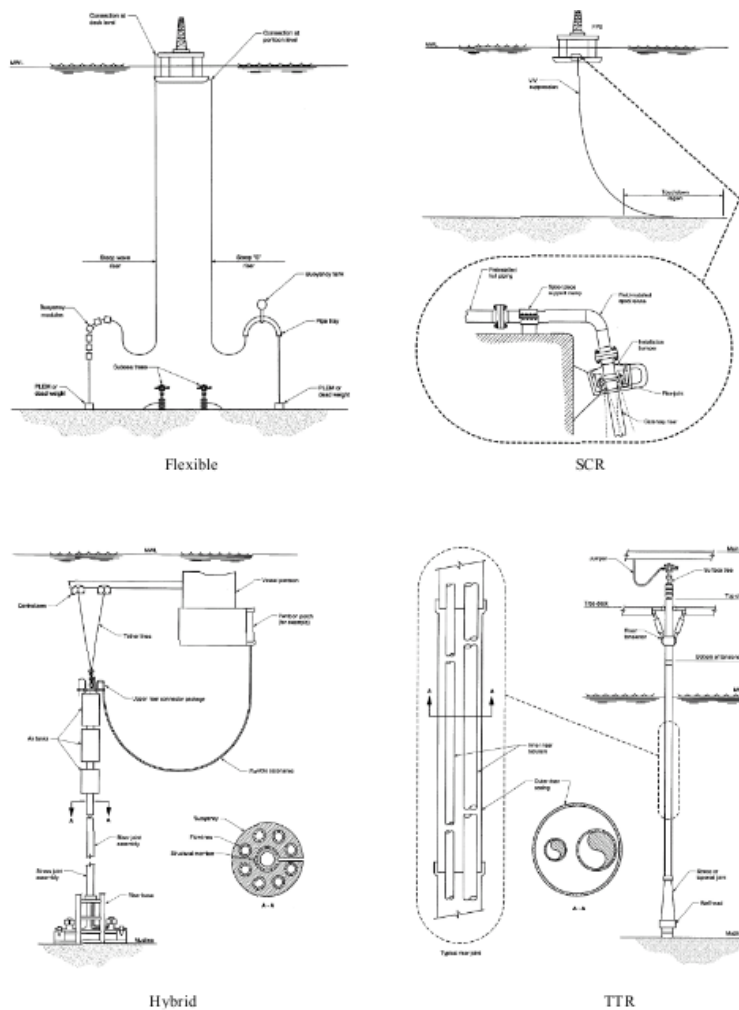


Figura 2.10: Tipos de sistema de *riser* para aguas profundas

La selección del sistema de *riser* adecuado depende de diversos factores como: la profundidad del agua, el medio ambiente, el mantenimiento de la estación, la producción de fluidos, la instalación del sistema, entre otros.

### 2.2.3 Plataformas fijas y flotantes

Para realizar el proceso de extracción de petróleo, es necesario contar con una estructura en la superficie del agua que se encargue de recibir los fluidos que se obtengan del yacimiento. La selección de este tipo de superficies depende de la profundidad del agua, el ambiente, el mantenimiento de la estación, el proceso de fluido, el equipo de procesamiento, ect. [18]

Actualmente existen dos tipos de plataformas (Fig. 2.11 <sup>11</sup>):

<sup>11</sup>Imagen tomada del libro *Introduction to Offshore Pipelines and Risers* [18]

- *Plataformas Fijas*: Este tipo de estructuras están fijas en el fondo del mar y se pueden utilizar cuando la profundidad es poca.
- *Plataformas Flotantes*: Este tipo de estructuras se encuentran sujetas por cables o cadenas en el fondo del mar y son utilizadas cuando la profundidad es muy grande.

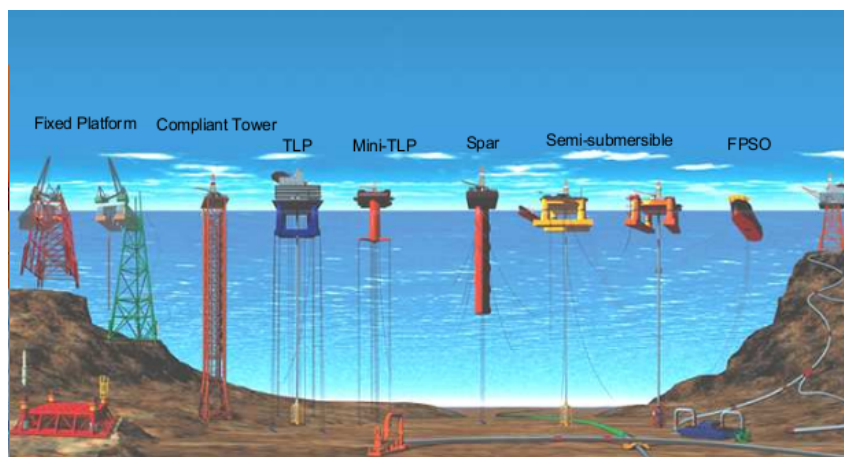


Figura 2.11: Tipos de plataformas que se utilizan en los campos petroleros

Los sistemas de producción flotantes son considerados como la solución más práctica y comercialmente más atractiva para el desarrollo de campos en aguas profundas [21].

#### 2.2.4 Sistema de procesamiento superior

Antes de enviar el crudo a las refinerías, es tratado por las instalaciones que se encuentran en las plataformas que se instalan sobre la superficie marina (fijas y flotantes). El diseño de este tipo de instalaciones tiene que ser compacta debido al límite de espacio y peso que se tiene en las plataformas [18].

Los requisitos del diseño de los sistemas de preprocesamiento superior, dependerán directamente de las condiciones en las que se está desarrollando el pozo, y los planes para su desarrollo futuro.

### 2.3 Modelos oceanográficos

Los modelos oceanográficos permiten entender la estructura de los movimientos del océano para ser capaces de predecir su comportamiento en un estado futuro. A través de ciencias básicas como la física, la química, la biología, la geología y las matemáticas; la *oceanografía* se encarga del estudio de los procesos que ocurren en los océanos y se

divide en cuatro ramas: biológica, física, química y geológica. Actualmente la investigación oceanográfica ha incorporado diversos modelos numéricos para describir la hidrografía de las masas de agua y el comportamiento del movimiento del océano.

Para el desarrollo de modelos oceanográficos se utiliza la oceanografía física, la cual se encarga de estudiar la distribución de propiedades del agua tales como: la temperatura, salinidad, densidad, oxígeno, ect. Además, se encarga de estudiar el movimiento del océano en respuesta a las fuerzas que actúan sobre él (corrientes, mareas y oleaje). La oceanografía física está dividida a su vez en dos subramas: oceanografía descriptiva y oceanografía dinámica.

La obtención de los datos oceanográficos para la construcción de los modelos, provienen de diferentes fuentes como: buques oceanográficos, aviones, satélites y sensores posicionados estratégicamente para recolectar información y analizarla posteriormente. Las propiedades del agua más importantes para el estudio de dinámica de fluidos son la presión, la temperatura y la salinidad.

El estudio del océano es un fenómeno físico complejo que involucra el espacio y el tiempo, que manifiesta características de mesoescala como giros y frentes. El estudio de la evolución, deformación e interacción de estas características dinámicas y su modelado es la esencia de la investigación en la circulación oceánica.

### 2.3.1 Base de datos oceanográfica

Este tipo de base de datos se encuentra conformada con la información recabada durante el desarrollo de campañas oceanográfica. La información se obtiene a través de las mediciones realizadas por las diversas estaciones hidrográficas ubicadas en el Golfo de México, principalmente con dos medios:

1. A través de una sonda hidrográfica *CTD* (*Conductivity, Temperature y Depth*) que registra datos continuos físicos y químicos desde la superficie del mar hasta el suelo del océano en diferentes posiciones geográficas del Golfo de México.
2. Recolectando muestras de agua de mar a diferentes profundidades, las cuales son analizadas posteriormente a bordo del buque oceanográfico para determinar sus concentraciones de oxígeno, salinidad y micronutrientes disueltos. Los resultados obtenidos se utilizan para calibrar los datos del CTD.

Todos los datos crudos obtenidos y toda la información que resulta del procesamiento numérico y gráfico que se aplica de los datos originales, conforman una base de datos oceanográfica, que por sus características en nuestro país sería de gran importancia y utilidad.

### 2.3.2 Método geostrófico

El método geostrófico es uno de los más utilizados en oceanografía dinámica para el cálculo de corrientes y flujos oceánicos. En este método los efectos de fricción, junto

con la fuerza de gravedad del sol y la luna son despreciables, por lo que el balance se da solamente entre las fuerzas horizontales de presión y la fuerza de Coriolis.

El método geostrófico requiere la utilización de tareas de procesamiento científico para realizar el cálculo de las propiedades que se describen a continuación.

*Topografía dinámica:* Para el cálculo de la altura dinámica de cada estación se utilizan datos de conductividad, temperatura y presión registrados por el CTD, tomando como niveles de referencia de movimiento cero profundidades de 100, 250, 500, 1000, 1500, 2000 y 3000 mts.

*Campo de velocidad geostrófico:* El cálculo de velocidad geostrófica de planta, relativo a superficies de movimiento cero, se calcula con base en las anomalías geopotenciales correspondientes a cada estación hidrográfica. A partir de esta distribución irregular de alturas dinámicas, se calculan nuevos valores mediante un algoritmo de interpolación bidimensional. Al realizar esta interpolación dentro de una malla regular definida, se obtiene una distribución uniforme de alturas dinámicas dentro del área de estudio. Con base a esta distribución uniforme de anomalías geopotenciales, para cada nodo de la malla se calculan los componentes norte-sur y este-oeste de la velocidad geostrófica utilizando la ecuación 2.1:

$$V = \frac{D_a - D_b}{2w \operatorname{sen} \phi} \quad (2.1)$$

donde  $V$  es la velocidad geostrófica,  $D_a - D_b$  es la distancia geopotencial,  $w$  la velocidad angular de la tierra y  $\phi$  la latitud media entre nodos adyacentes.

*Secciones verticales de velocidad:* Se obtienen de las alturas dinámicas de cada uno de los transectos latitudinales y meridionales. Para cada una de las estaciones hidrográficas se calculan los perfiles de velocidad de la superficie a diferentes profundidades utilizando la ecuación 2.1. Posteriormente se ajustan los datos mediante un producto tensorial de interpolación bidimensional para graficarlos y representar el campo de velocidad de los transectos latitudinales y meridionales, en donde las velocidades positivas tienen dirección hacia el norte y el este, y las velocidades negativas hacia el sur y el oeste

*Transporte geostrófico horizontal* para todos los transectos (latitudinales y meridionales), se estima a partir de las velocidades geostróficas calculadas. El transporte geostrófico se define en la ecuación 2.2

$$Q = V \times A \quad (2.2)$$

donde  $V$  es la velocidad y  $A$  representa el área perpendicular o normal al flujo.

Para determinar el transporte, se toma la malla definida para las interpolaciones de velocidad y se calcula el transporte para cada rectángulo de la misma mediante la ecuación 2.3

$$T_l = \sum_i \sum_j \left( \frac{v_{i,j} + v_{i,j+1} + v_{i+1,j} + v_{i+1,j+1}}{4} \right) \Delta x \Delta y \quad (2.3)$$

donde  $T_l$  es el transporte horizontal del transecto 1,  $v_{i,j}$  es la velocidad horizontal del nodo  $(i, j)$ ,  $\Delta x$  es la longitud horizontal de un rectángulo de la malla y  $\Delta y$  la longitud vertical de un rectángulo de la malla.

*Transporte vertical:* las velocidades verticales se estiman del balance entre la divergencia horizontal y el gradiente vertical. La distribución se calcula utilizando superficies separadas por intervalos de profundidad. Cada una de las superficies se representa en una malla compuesta de rectángulos. En cada uno de los nodos se hace una interpolación de la velocidad vertical y se utilizan los valores de velocidad de los cuatro vértices para obtener una velocidad vertical promedio, la cual se multiplica por el área del rectángulo para obtener el transporte vertical asociado al rectángulo. Finalmente, se calcula el transporte total para cada superficie sumando los transportes de todos los rectángulos. El transporte positivo indica flujo hacia arriba y el negativo flujo hacia abajo.

## 2.4 Simulación

La simulación se puede ver como una técnica que nos permite construir un modelo representativo de la realidad, con el cual se pueden realizar experimentos para analizar su comportamiento en diversas situaciones. Por su parte, la simulación por computadora se puede definir como una técnica numérica para conducir experimentos en una computadora digital, los cuales requieren ciertos tipos de modelos lógicos y matemáticos, que describen el comportamiento de un sistema (o algún componente de este) en periodos extensos de tiempo real [22].

Una simulación tiene los siguientes propósitos:

- Conocer el funcionamiento de un sistema. La observación detallada de lo que se está simulando permite un mejor conocimiento, proporcionando información para mejorarlo.
- Ayuda a identificar problemas.
- Permite experimentar con nuevas situaciones y generar políticas de decisión.
- Experimentar en un tiempo reducido. La simulación permite, en algunos segundos, analizar procesos que llevaría meses o años desarrollarlo en tiempo real.
- Reducción de los procesos analíticos de investigación y desarrollo convencional.
- Permite estudiar situaciones complejas.
- Para educación y entrenamiento.
- Para tener una visión global del sistema.

Por su parte, la simulación por computadora requiere de un modelo que pueda ser desarrollado por medio de un programa en computadora. La simulación por computadora se encuentra dentro del campo de la *Computación Científica*, y nos ofrece la oportunidad de estudiar y entender el comportamiento de fenómenos naturales [4].

En la actualidad se han desarrollado un gran número de simuladores, los cuales tienen diversos objetivos (simuladores de vuelo, juegos militares, simuladores de economía, fenómenos físicos, ect.). El fundamento principal para usar herramientas de simulación es adquirir conocimientos que permitan la predicción del futuro.

### 2.4.1 El proceso de simulación por computadoras

Para llevar a cabo el desarrollo de experimentos de simulación en computadoras, es necesario utilizar diversas técnicas como: estadística matemática, análisis numérico, programación, ect. El proceso de experimentos de simulación en computadora está formado por las siguientes etapas [22]:

1. *Formulación del problema*: Al igual que en otras áreas de la computación científica, se tiene que comenzar con la formulación de un problema y definir claramente los objetivos de la investigación que se va a desarrollar.
2. *Recolección y procesamiento de datos*: Esta etapa consiste en captar de la realidad, los hechos y la información disponible que pueda ser procesada posteriormente. En muchas ocasiones, es necesario realizar una conversión de la información obtenida, a datos específicos que puedan ser interpretados y procesados de manera más eficiente.
3. *Formulación del modelo matemático*: Esta etapa consiste en construir un modelo para representar el fenómeno que se va a simular. El modelo nos va a proporcionar descripciones o predicciones sobre comportamiento del sistema dado.
4. *Estimación de parámetros a partir de los datos reales*: Una vez que se ha formado el modelo matemático, es necesario estimar los valores de los parámetros del modelo y posteriormente probar su significado estadístico.
5. *Evaluación del modelo*: En este paso se va a probar la exactitud de nuestro modelo para poder tener un juicio inicial de nuestro sistema.
6. *Desarrollo del programa para computadora*: Para llevar a cabo esta etapa, se debe de representar el modelo mediante un algoritmo y posteriormente elegir el lenguaje en el cual se va a desarrollar la aplicación. El desarrollo del programa generalmente debe permitir especificar las condiciones iniciales del sistema, procesar los datos y generar reportes de salida gráficos para presentarlos a los usuario.
7. *Validación*: Para poder validar la aplicación, es necesario hacer un proceso de comparación de resultados. El problema de validar modelos de simulación es

uno de los procesos más complicados ya que implica diferentes complejidades (teórica, práctica, estadística, etc.).

8. *Diseño de experimentos*: Después de estar satisfechos con la validación de nuestro modelo y de la aplicación desarrollada, podemos dirigir de manera eficaz los experimentos. En esta fase vamos a poder seleccionar las combinaciones de factores que intervienen en el proceso (condiciones iniciales, datos aleatorios, etc.) y seleccionar el orden en el que se deben llevar a cabo las simulaciones.
9. *Análisis de los datos simulados*: Finalmente se requiere hacer un análisis de los datos generados por la computadora y darles una interpretación.

El proceso de simulación por computadora nos ayuda a estudiar fenómenos naturales que resultan muy difíciles o inclusive imposibles de reproducir en la realidad. Este medio nos permite resolver conjeturas y tener una predicción sobre el comportamiento que pudiera tener el sistema en condiciones especiales [4].

## 2.5 Vibración

En general los sistemas de ingeniería que poseen masa y elasticidad pueden presentar movimientos relativos. Si este movimiento se repite después de un determinado intervalo de tiempo el movimiento se conoce como *vibración* [3]. La vibración se puede ver como una forma de energía disipada. La vibración puede presentarse en dos formas:

- **Vibración Libre**: Es el movimiento cuando el sistema se desplaza de su posición de equilibrio. En este movimiento interviene la elasticidad de la estructura, la fricción y el amortiguamiento.
- **Vibración Forzada**: Se presenta cuando actúan fuerzas externas sobre el sistema durante su movimiento. Las fuerzas son generalmente de tipo  $F(t) = F_0 \sin \omega t$  o  $F(t) = F_0 \cos \omega t$ .

En la actualidad una gran variedad de aplicaciones derivadas de estructuras mecánicas, sistemas acústicos, circuitos eléctricos, fluidos mecánicos, etc. requieren un análisis de vibraciones en sus estructuras. Cuando una estructura es excitada por fuerzas externas en las cuales las frecuencias se encuentran cerca de sus frecuencias naturales, la vibración en la estructura se amplifica y el sistema se vuelve inestable. Este fenómeno se conoce como "*Resonancia*" [2]. Dependiendo de los sistemas o de la aplicación, el fenómeno de resonancia puede ser deseado o no deseado. En las estructuras mecánicas las vibraciones pueden provocar desgastes que pueden producir catástrofes. Por otra parte, en muchas aplicaciones de ingeniería eléctrica como al sintonizar una estación de radio, el interés está en encontrar una amplitud de resonancia tan grande como sea posible.

### 2.5.1 Ejemplos de estructuras afectadas por el fenómeno de vibración

En la actualidad el fenómeno de vibración y resonancia es muy estudiado sobre todo en la construcción de grandes estructuras. Una de las estructuras donde comúnmente se presenta el fenómeno de la vibración son los puentes, en los cuales si no se hace un cálculo adecuado de la frecuencia natural pueden llegar a colapsarse. Uno de los casos más importantes fue el del puente de Tacoma (Fig. 2.12 <sup>12</sup>) que es un puente colgante de 1600 mts. de longitud. En el año de 1940 la estructura tuvo un dramático colapso debido al viento.

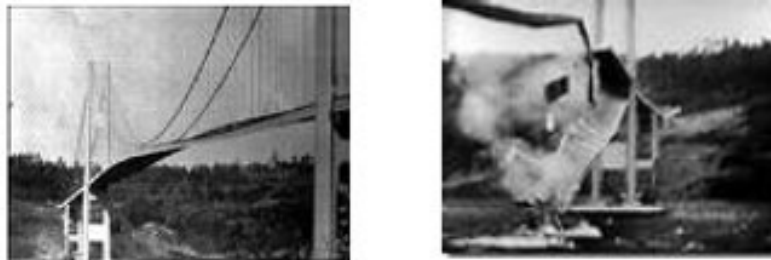


Figura 2.12: Puente Tacoma

Uno de los casos más recientes en este tipo de estructuras fue el que se dió en Junio del 2000 en el Puente del Milenio, el cual tiene una longitud de 320 mts. y se encuentra localizado sobre el río Támesis en Londres (Fig. 2.13 <sup>13</sup>). Esta estructura en el día de su inauguración comenzó a tambalearse de forma alarmante debido al paso de cientos de personas que habían asistido a dicha inauguración. Dos días después el puente fue cerrado y posteriormente fue equipado con amortiguadores para evitar que se tambaleara nuevamente.



Figura 2.13: Puente del Milenio, Londres.

Otro tipo de estructuras que están expuestas a este tipo de fenómenos de vibración son los grandes edificios. En este tipo de estructuras también es necesario implementar

<sup>12</sup>Imagen tomada del sitio <http://puentetacoma.blogspot.com/>

<sup>13</sup>Imagen tomada del artículo *The Quadratic Eigenvalue Problem* [2]



métodos numéricos complejos para evitar o amortiguar las excitaciones a las que están expuestas, principalmente al viento. Uno de los casos más recientes se dio en el Hotel *Burj Al Arab* en Dubai, considerado el hotel más alto del mundo (Fig. 2.14<sup>14</sup>). Debido a que ésta construcción fue inspirada en una embarcación de vela y a su cercanía con la costa, al momento de hacer pruebas de viento simulado se presentó un fenómeno el cual, bajo ciertas condiciones de viento soplando sobre los vértices afilados de la estructura, se podían crear minitornados que producían peligrosas vibraciones que sacudían el edificio y podían llegar a destruirlo. Para solucionar este problema recurrieron a un peso colgante llamado amortiguador de masa sincronizado que hace que cuando el viento sople, se mezan 5 toneladas de peso en lugar de la estructura, amortiguando las vibraciones [23].



Figura 2.14: Hotel Burj Al Arab, Dubai.

Durante el proceso de exploración y explotación de petróleo en aguas profundas, también se presenta el fenómeno de vibración. Este fenómeno se produce principalmente por la fuerza que ejercen las corrientes marinas sobre las estructuras que se utilizan en la extracción del petróleo (Fig. 2.15<sup>15</sup>). Para evitar accidentes debido a las vibraciones, es necesario tener un modelo que nos permita predecir el comportamiento de los *riser* y determinar la resistencia que necesita para soportar la interacción de las corrientes.

<sup>14</sup>Imagen tomada del artículo *Burj Al Arab. Una superestructura de 7 estrellas* [23]

<sup>15</sup>Imagen tomada del *Deepwater Reference Book* [21]

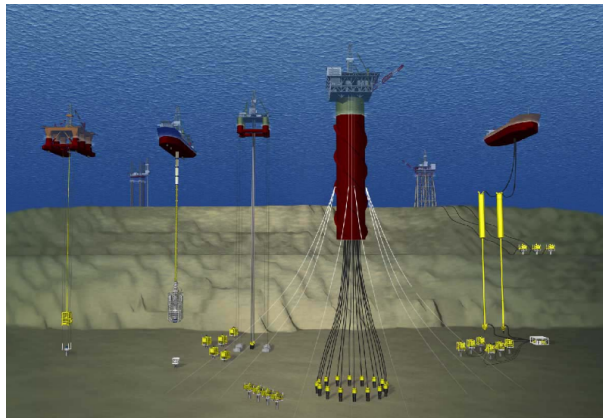


Figura 2.15: Tipos de *riser* utilizados en los campos de aguas profundas.

## 2.6 Trabajos relacionados

En la industria petrolera no se cuenta con mucha información referente al software que se utiliza para la exploración y explotación de petróleo en aguas profundas, ya que este es desarrollado por empresas privadas las cuales lo venden a costos muy altos y no proporcionan al público mucha información respecto a ellos. A continuación se describen algunos desarrollos de software utilizados para administrar y monitorear los sistemas de *riser*:

- *Flexcom 3D*: Desarrollado por MCS Internacional. Este es un sistema dinámico no lineal que utiliza el método del elemento finito para analizar el comportamiento de una amplia gama de estructuras marítimas. Su capacidad de análisis incluye la instalación y la operación de las estructuras en condiciones extremas. Esta herramienta es utilizada por las empresas petroleras y fabricantes de equipo para el análisis de tuberías flexibles [21] [24].
- *Visual Orcaflex*: Desarrollado por Orcina. Es un software para el análisis dinámico de sistemas marinos de alta mar. Esta aplicación nos ofrece un modelado en 3D, no lineal. Permite realizar un análisis de fatiga, de tensión, flexión y torsión. Además permite realizar un análisis de la maniobra de buques (velocidad de avance y giro). [21] [25].
- *Optima*: Desarrollado por MCS Internacional. Es un sistema para administración de herramientas, planeación y monitoreo de perforaciones o finalización y redireccionamiento de *riser* [26].
- *Shear 7*: Desarrollado por el MIT. El Shear 7 es una herramienta de modelado para la predicción de vibraciones inducida por vórtices (VIV). El programa se encarga de la predicción de vibraciones en estructuras marítimas que son sometidas a la variación de corrientes. El programa permite calcular las respuestas de las

estructuras, las medidas de fuerza de arrastre y la predicción de daños por fatiga [21] [27].

- *BOAST*: Es una herramienta de simulación tridimensional trifásica para yacimientos del tipo Black Oil (petróleo negro). Este simulador es de código abierto, se encuentra programado en FORTRAN y fue desarrollado por el Laboratorio Nacional de Tecnología Energética (NETL), el cual se encuentra asociado al Departamento de Energía de los Estados Unidos (DOE). Este simulador puede resultar un poco deficiente, sobretodo a nivel de solución de sistemas de ecuaciones. Posteriormente a la versión inicial, salió una versión mejorada *BOAST-NFR*. Las mejoras fundamentales de esta versión es que puede ser ejecutada bajo Windows, los datos pueden ser cargados en hojas de cálculo de excel y el código se ejecuta en subrutinas programadas en Visual Basic.

## 2.7 Resumen

En este capítulo se dió un panorama general sobre el proceso y los conocimientos técnicos que se requieren para la comprensión de este proyecto de investigación. En la primera sección analizamos los antecedentes del petróleo y las nuevas fuentes proveedoras que se han descubierto. Además, se describieron los principales retos y problemas a los que se enfrenta las empresas durante el proceso de exploración y explotación de petróleo en aguas profundas.

En la segunda sección se analizaron los cuatro elementos principales que se requieren para el desarrollo de un campo de petróleo en alta mar. Se describieron las funciones de cada uno de estos elementos, la metodología y técnicas requeridas para su desarrollo e implementación.

En la tercer sección se describe en que consiste un modelo oceanográfico y la importancia que tiene en el análisis del comportamiento del movimiento de los océanos.

En la cuarta sección describimos el concepto de simulación y su aplicación en el desarrollo de aplicaciones computacionales para el estudio de distintos fenómenos. Se mencionaron las ventajas de la implementación de simuladores. Y finalmente se analizaron las partes principales y el proceso general para la construcción de un simulador.

En la quinta sección del capítulo se vió el fenómeno de vibración y las dos formas en las que se puede presentar. Se analizó el concepto de resonancia y la forma en como afecta a las estructuras mecánicas. También se vieron algunos ejemplos de estructuras donde se presentan estos fenómenos y la forma en como se solucionaron.

Finalmente en la última sección se presenta una investigación sobre simuladores que existen actualmente en la industria petrolera y el diseño de algunos softwares específicos para los sistemas *riser*.



# Capítulo 3

## Modelo del sistema

Todo aquello que un hombre puede imaginar, otros lo podran realizar.

---

Julio Verne

**E**l modelo de un sistema, es una estructura que permite conocer o deducir su comportamiento a partir de bases teóricas, y que tiene una gran semejanza al sistema real que se está analizando [28].

Como se comentó en el Capítulo 1, la elección de un modelo adecuado para representar el sistema, es uno de los pasos más importantes en el proceso de la simulación. La elección del sistema debe estar en función de los objetivos que se plantearon en el inicio de la investigación y de la precisión que se requiera, para que los resultados obtenidos que se obtengan estén apegados a la realidad.

Para formular un modelo, es necesario realizar un proceso de observación del fenómeno que se está estudiando que nos permita hacer una representación física del sistema y formular una o más hipótesis relativas a su funcionamiento. La abstracción de esta representación física nos va a permitir la formulación de un modelo matemático que describa su comportamiento. Además nos va a permitir desarrollar modelos a escala para poderlo someter a pruebas.

El modelo matemático nos va a ofrecer un soporte confiable para nuestra simulación [4], ya que la validación de los resultados obtenidos dependen directamente del modelo matemático que se formuló.

Una de las consideraciones más importantes que se deben de tomar en cuenta para la formulación de un modelo matemático es su complejidad. De esto depende la dificultad de programación y el tiempo de cómputo que requieran para resolverlos [22].

En este capítulo describiremos la representación física de nuestro sistema y los pasos que se realizaron para la construcción del modelo matemático que nos va a permitir predecir su comportamiento.

### 3.1 Modelo físico del sistema

Nuestro modelo físico representa un *riser*, que es una tubería de metal que se encuentra sumergida en el océano y que está conectada a la plataforma flotante y una estación submarina (ver sección 2.2.2). Sobre esta tubería, las corrientes marinas generan vibraciones que producen fatigas y desgastes sobre la estructura. El propósito de nuestro modelo es predecir el comportamiento de estas estructuras en condiciones extremas, producidas principalmente por las corrientes marinas.

Por ello, el sistema se puede representar como una tubería de metal sumergida en agua, sobre la cual se presentan fuerzas externas que producen vibraciones en la estructura metálica. Para analizar las vibraciones a lo largo de la estructura, el *riser* se dividió en varios puntos (partículas), sobre los cuales se analiza la fuerza que ejercen las corrientes marinas y se determina que tanto afectan estas fuerzas a la estructura en un punto determinado. Estas corrientes marinas se pueden presentar con una fuerza y/o dirección diferente en cada una de las partículas (Fig 3.1). Consideramos que la división de las partículas de masa en la que se dividirá el sistema será homogénea, de tal forma que cada uno de los puntos se encontrarán equidistantes uno de otro.

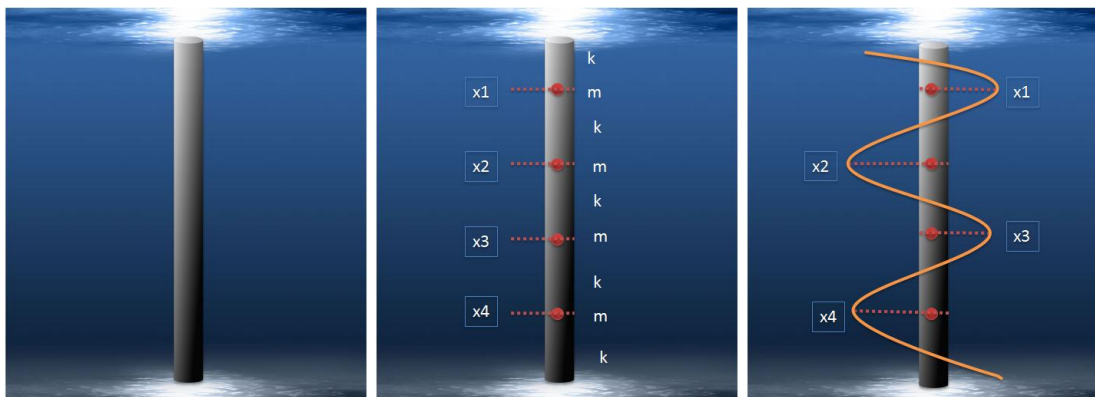


Figura 3.1: Representación del modelo físico. Tubería sumergida en agua dividida en  $n$  partículas

Este sistema se puede representar como una cadena de vibración monoatómica unidimensional que está formada por un conjunto de partículas (con masas iguales  $m$ ) que son equidistantes, y unidas por medio de resortes (con una constante elástica  $k$ ) (Fig. 3.2). Cuando se hace vibrar una cadena conformada de esta forma, los diversos movimientos que pueda tener respecto a un sistema de coordenadas cartesianas (en

este caso cada una de nuestras partículas), definen los grados de libertad de nuestro sistema [28].

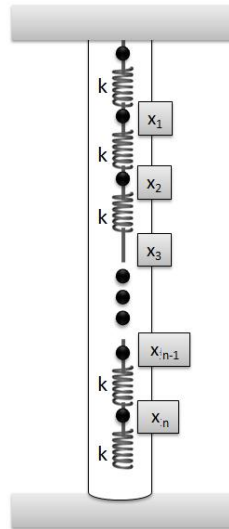


Figura 3.2: Movimiento del riser provocado por las corrientes marinas

De esta manera obtuvimos un sistema con  $n$  grados de libertad y en consecuencia tendremos  $n$  modos normales de vibración, en el cual las masas son iguales para todas sus partículas  $m_0 = m_1 = m_2 \cdots = m_{n-1} = m_n = m$  y de igual forma los resortes también tienen el mismo valor de elasticidad  $k_0 = k_1 = k_2 \cdots = k_{n-1} = k_n = k$ .

En cada uno de los modos normales de vibración, todas las partículas de la cadena oscilarán con la misma frecuencia y fase. Por lo que cualquier vibración en nuestro sistema será el resultado de una superposición de sus modos normales de vibración, es decir, el comportamiento del sistema estará en función de sus modos de vibración.

## 3.2 Movimiento Armónico Simple (MAS)

El *Movimiento Armónico Simple (MAS)* es la forma más sencilla de un movimiento periódico. Una partícula presenta un MAS cuando tiene un movimiento rectilíneo en el cual su aceleración es siempre proporcional a la distancia a un punto fijo de la trayectoria [3], es decir, cuando su posición esta en función del tiempo de una senoide. Si se presenta un movimiento periódico de vibración, sea simple o complejo (como el que se representan en nuestro modelo físico), puede considerarse formado por un MAS ó un número de MAS de diferentes amplitudes y frecuencias por medio de una serie de Fourier.

La ecuación diferencial que representa al MAS esta dada por la ecuación 3.1

$$m\ddot{x} + c\dot{x} + kx = F(t) \quad (3.1)$$

donde  $m$  representa la masa,  $c$  el amortiguamiento y  $k$  es el coeficiente de rigidez. La solución general de este tipo de sistemas para cuando  $c = 0$  esta dada por la siguiente ecuación:

$$x = A \operatorname{sen}(\omega t + \phi) \quad (3.2)$$

donde  $A$  representa la amplitud del movimiento,  $\omega$  es la frecuencia angular,  $\phi$  es la fase inicial y  $t$  representa el tiempo. Debido a que los valores de la función seno van de  $+1$  a  $-1$  el movimiento se realiza entre una región que va desde  $+A$  hasta  $-A$  (fig. 3.3) [29].

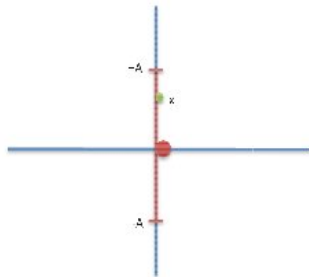


Figura 3.3: Gráfica de Movimiento Armónico Simple

Si derivamos al ecuación 3.2 con respecto al tiempo ( $t$ ), podemos obtener la velocidad a la que se desplaza la partícula [30].

$$v = \frac{dx}{dt} = -\omega A \cos(\omega t + \phi) \quad (3.3)$$

Si derivamos nuevamente la ecuación 3.3 con respecto al tiempo (la doble derivada de 3.2), obtenemos la aceleración de la partícula [30].

$$a = \frac{dv}{dt} = -\omega^2 A \operatorname{sen}(\omega t + \phi) \quad (3.4)$$

donde:

$$\omega^2 = \frac{k}{m} \implies \omega = \sqrt{\frac{k}{m}} \quad (3.5)$$

Puesto que la función *seno* es periódica, el movimiento se repite después de un tiempo  $2\pi/\omega$ , lo que representa el periodo  $T$  del movimiento [30]

$$T = \frac{2\pi}{\omega} = \frac{2\pi}{\sqrt{k/m}} \quad (3.6)$$

A partir de la ecuación 3.6 podemos concluir que a mayor masa, más lenta será la oscilación (mayor periodo). Y si el resorte es más “blando” ( $k$  grande), también se tendrá una oscilación más lenta.

Por el contrario, pequeñas masas y resortes “duros” (menor  $K$ ) darán como resultados oscilaciones rápidas (de alta frecuencia).



### 3.2.1 Sistema de un grado de libertad

Este tipo de sistemas están restringidos de tal forma que solo pueden vibrar de una manera, es decir, que solo se necesita una coordenada independiente para determinar por completo la localización geométrica de la masa del sistema en el espacio [3].

Si se tuviera un sistema de masa resorte (como el que se muestra en la figura 3.4) en cual estuviese restringido para moverse horizontalmente, se dice que el sistema solo posee un grado de libertad.

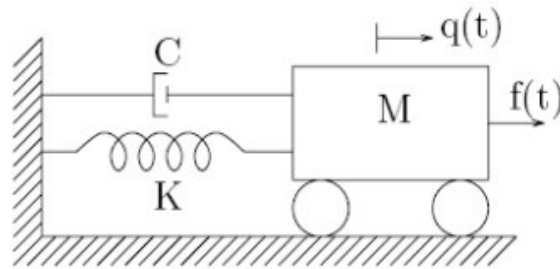


Figura 3.4: Sistema de un grado de libertad

### 3.2.2 Sistema de varios grados de libertad

Cuando en un sistema (como el de nuestro modelo) se necesitan  $n$  coordenadas independientes para determinar las posiciones de las masas de un sistema, el sistema se denomina como de  $n$  grados de libertad. El movimiento de un sistema con  $n$  grados de libertad se puede representar por  $n$  ecuaciones diferenciales [3].

Las ecuaciones diferenciales de un sistema de  $n$  masas se pueden expresar de manera general en forma de matrices de la siguiente manera:

$$\begin{aligned} m_{11}\ddot{x}_1 + m_{12}\ddot{x}_2 + \cdots + m_{1n}\ddot{x}_n + k_{11}x_1 + k_{12}x_2 + \cdots + k_{1n}x_n &= 0 \\ m_{21}\ddot{x}_1 + m_{22}\ddot{x}_2 + \cdots + m_{2n}\ddot{x}_n + k_{21}x_1 + k_{22}x_2 + \cdots + k_{2n}x_n &= 0 \\ &\vdots \\ m_{n1}\ddot{x}_1 + m_{n2}\ddot{x}_2 + \cdots + m_{nn}\ddot{x}_n + k_{n1}x_1 + k_{n2}x_2 + \cdots + k_{nn}x_n &= 0 \end{aligned}$$

utilizando una notación matricial la podemos representar de la siguiente manera:

$$\begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \vdots \\ \ddot{x}_n \end{bmatrix} + \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$[M] \{\ddot{x}\} + [K] \{x\} = 0 \quad (3.7)$$

donde  $[M]$  es la matriz de la masa de las partículas y  $[K]$  es la matriz de rigidez de los resortes.

El uso de matrices en el análisis de este tipo de sistemas de vibración, nos ayuda a simplificar el trabajo y a comprender el procedimiento usado en la solución. Para este tipo de sistemas debido al gran número de posibilidades que existen, encontrar las frecuencias naturales y evaluar las respuestas, generalmente requiere mucho más trabajo matemático [3].

### 3.3 Modelo matemático

Una vez que se ha descrito la representación de nuestro modelo físico, el siguiente paso consiste en transformarlo a un modelo matemático. Como se comentó anteriormente, el modelo del sistema se puede considerar como una cadena monoatómica lineal, formada con  $n$  partículas ( con  $n$  grados de libertad) homogéneas equidistantes, con sus extremos fijos (Fig. 3.2).

Este tipo de sistemas presentan un movimiento periódico y se componen de movimientos armónicos de diferentes amplitudes y frecuencias que se pueden representar en una matriz diagonal (Matriz  $\lambda$ ) (Fig. 3.5).

En el desarrollo de este trabajo de tesis se propone el desarrollo de dos modelos:

1. Modelo con Interacción a primeros vecinos
2. Modelo con Interacción a segundos vecinos

los cuales se describirán en las siguientes secciones.

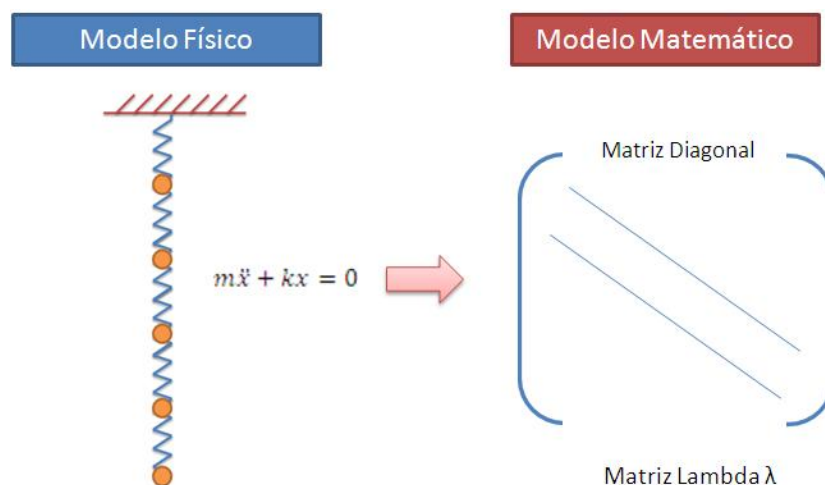


Figura 3.5: Transformación del modelo físico al modelo matemático

### 3.3.1 Modelo matemático con interacción a primeros vecinos

En una cadena de vibración como la de nuestro caso de estudio, las partículas que componen el sistema y que están unidas por resortes elásticos, se ven afectadas por el movimiento de sus partículas vecinas superior e inferior; es decir, que la partícula  $i$ -ésima solo tiene una interacción con la partícula  $i - 1$  (partícula superior) y con la partícula  $i + 1$  (partícula inferior), lo que se conoce como *Interacción a Primeros Vecinos* (Fig. 3.6) [31].

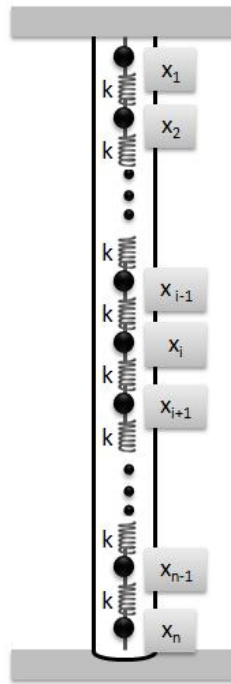


Figura 3.6: Cadena de vibración con interacción a primeros vecinos

La fuerza resultante de una partícula de la cadena se puede obtener mediante la ecuación de la ley de Hooke (ecuación 3.8)

$$f_i = \sum_{j=1}^n k_{ij} (x_{i+j} + x_{i-j} - 2x_i) \quad (3.8)$$

donde  $f_i$  es la fuerza resultante de la partícula  $i$ ,  $x_l$  es el desplazamiento que tiene la partícula  $l$  y  $k_{ij}$  es el resorte que liga a las partículas  $i$  y  $j$  [31].

Tomando la ecuación 3.8 y si consideramos que solo se tiene una interacción a primeros vecinos obtenemos la ecuación 3.9

$$f_i = k (x_{i+j} + x_{i-j} - 2x_i). \quad (3.9)$$

Usando la segunda ley de Newton [32], encontramos que la fuerza actuando sobre la partícula  $i$ -ésima del sistema esta dada por:

$$\sum_{i=1}^n f_i = m_i \ddot{x}_i. \quad (3.10)$$

Al igualar la ecuación 3.10 con la ecuación 3.9 obtenemos:

$$m_i \ddot{x}_i = k(x_{i+1} + x_{i-1} - 2x_i) \quad (3.11)$$

Sin embargo, debido a que en nuestro sistema tenemos fronteras fijas, la ecuación 3.11 tiene restricciones para las partículas de los extremos superior e inferior. La partícula 1 del sistema no cuenta con un vecino superior y la partícula  $n$  no cuenta con un vecino inferior, por lo que se tomaron dos partículas imaginarias  $0$  y  $n + 1$  las cuales tendrán un desplazamiento con valor de  $0$  (Fig. 3.7).

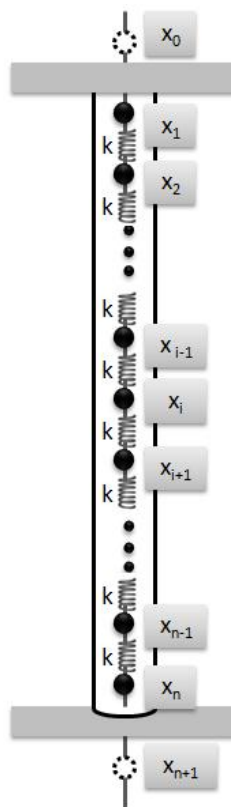


Figura 3.7: Cadena de vibración con interacción a primeros vecinos con partículas imaginarias en los extremos

Debido a lo anterior se obtuvieron tres ecuaciones de movimiento diferentes, una para la partícula  $x_1$  que representa la frontera superior, posteriormente la partícula

$x_i$  que representa todas las partículas del centro y finalmente la partícula  $x_n$  que representa la frontera inferior del sistema (Fig. 3.8). De esta manera las ecuaciones de movimiento que obtuvimos quedaron de la siguiente manera:

$$\begin{aligned} m_1 \ddot{x}_1 &= -2kx_1 + kx_2 \\ m_2 \ddot{x}_2 &= kx_1 - 2kx_2 + kx_3 \\ m_3 \ddot{x}_3 &= kx_2 - 2kx_3 + kx_4 \\ m_i \ddot{x}_i &= kx_{i-1} - 2kx_i + kx_{i+1} \\ m_{n-1} \ddot{x}_{n-1} &= kx_{n-2} - 2kx_{n-1} + kx_n \\ m_n \ddot{x}_n &= kx_{n-1} - 2kx_n \end{aligned}$$

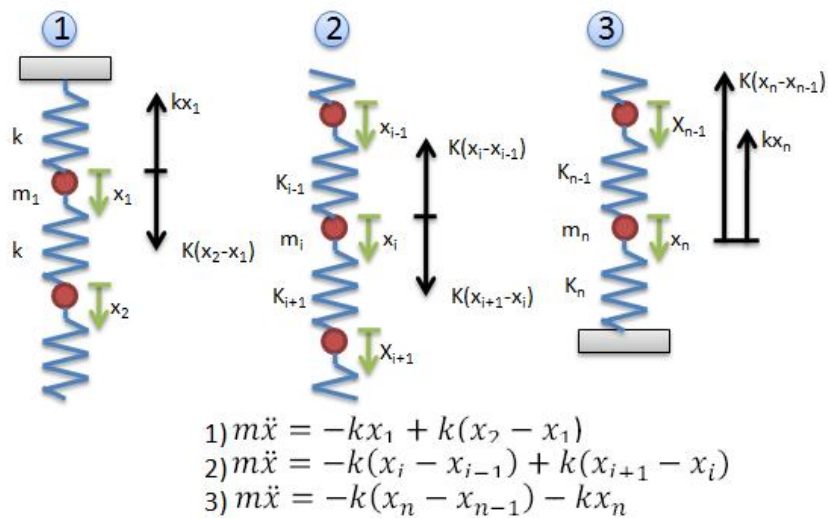


Figura 3.8: Ecuaciones de movimiento con interacción a primeros vecinos

El sistema de ecuaciones obtenido, representan el número de MAS por los que está compuesto el sistema, entonces fue necesario encontrar una solución del tipo:

$$x_i = A_i \text{sen}(\omega t) \tag{3.12}$$

en donde al obtener la segunda derivada de la ecuación 3.12 con respecto al tiempo ( $t$ ) tenemos:

$$\ddot{x}_i = -\omega^2 A_i \text{sen}(\omega t) . \tag{3.13}$$

Las ecuaciones de movimiento producen un sistema de ecuaciones diferenciales. Este sistema se puede representar en forma matricial reemplazando las ecuaciones 3.12 y 3.13 en las ecuaciones de movimiento de nuestro sistema. Si se simplifica obtenemos el siguiente sistema de ecuaciones homogéneo:

$$\begin{pmatrix} \frac{2k}{m} & -\frac{k}{m} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{k}{m} & \frac{2k}{m} & -\frac{k}{m} & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{k}{m} & \frac{2k}{m} & -\frac{k}{m} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & -\frac{k}{m} & \frac{2k}{m} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{pmatrix} = \omega^2 \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{pmatrix} \quad (3.14)$$

De esta forma, si por ejemplo se tuviera un sistema con 4 grados de libertad obtendríamos el siguiente modelo matemático con interacción a primeros vecinos:

$$\begin{pmatrix} \frac{2k}{m} & -\frac{k}{m} & 0 & 0 \\ -\frac{k}{m} & \frac{2k}{m} & -\frac{k}{m} & 0 \\ 0 & -\frac{k}{m} & \frac{2k}{m} & -\frac{k}{m} \\ 0 & 0 & -\frac{k}{m} & \frac{2k}{m} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix} = \omega^2 \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix}$$

Este sistema se puede expresar de forma

$$MX = \omega^2 X \quad (3.15)$$

Una forma de resolver este sistema de ecuaciones diferenciales de segundo orden, es obtener los valores propios diagonalizando la matriz  $M$  [28]. Donde  $\omega^2$  son los valores propios de la matriz cuadrada y  $X$  el vector propio de los coeficientes  $A_1, A_2 \dots A_n$  que representan las amplitudes de los modos normales de vibración.

### 3.3.2 Modelo matemático con interacción a segundos vecinos

Se propuso un segundo modelo en el cual, el movimiento de las partículas del sistema son afectadas por sus dos partículas vecinas superiores y sus dos partículas vecinas inferiores; es decir, que la partícula  $i$ -ésima del sistema tiene interacción con las partículas  $i-2, i-1$  superior y con  $i+1, i+2$  inferior, lo que se conoce como *Interacción a Segundos Vecinos* (Fig. 3.9) [28].

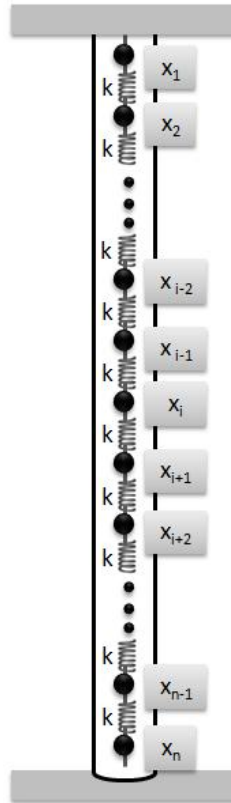


Figura 3.9: Cadena de vibración con interacción a segundos vecinos

Al igual que en el modelo anterior, en este modelo también se aplica la ley de Hooke, por lo que tomando en consideración las ecuaciones 3.8 y 3.10 obtenemos la ecuación general de nuestro sistema para una interacción a segundos vecinos.

$$f_i = m_i \ddot{x}_i = k(x_{i+1} + x_{i-1} - 2x_i) + k'(x_{i+2} + x_{i-2} - 2x_i) \quad (3.16)$$

Si tomamos en consideración que la constante elástica  $k$  es igual para todos los resortes que ligan a las partículas del sistema  $k = k'$ . De la ecuación 3.17 obtenemos:

$$m_i \ddot{x}_i = kx_{i-2} + kx_{i-1} - 4kx_i + kx_{i+1} + kx_{i+2} \quad (3.17)$$

La ecuación (3.17) presenta restricciones para las partículas 1, 2,  $n-1$  y  $n$ , ya que estas no cuentan con todas sus partículas vecinas (2 vecinas superiores y 2 inferiores). Para afrontar estas restricciones, igual que en el modelo de primeros vecinos se propusieron agregar partículas imaginarias en los extremos con un valor de desplazamiento 0 (Fig. 3.10).

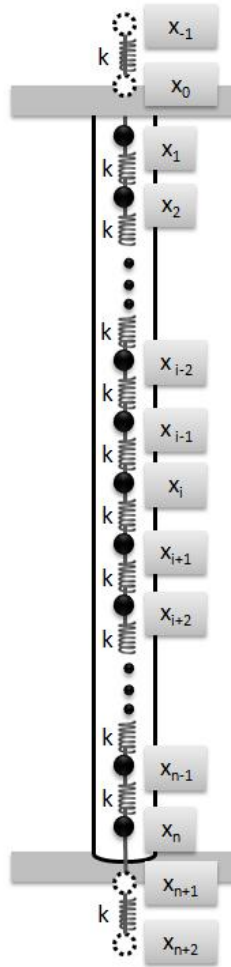


Figura 3.10: Cadena de vibración con interacción a segundos vecinos con partículas imaginarias en los extremos

Tomando en consideración lo anterior, se obtuvieron 5 ecuaciones distintas para este modelo. La primer ecuación es para la partícula  $x_1$  que representa la frontera superior y que no cuenta con sus dos vecinas superiores. La segunda ecuación es para la partícula  $x_2$  la cual sólo cuenta con una partícula vecina superior. La tercera ecuación representa al movimiento de todas las partículas del centro del sistema  $x_i$  y que cuenta con sus dos vecinos superiores e inferiores. La cuarta ecuación es para la partícula  $x_{n-1}$  que es la penúltima del sistema y que sólo cuenta con una partícula vecina inferior. Por último la ecuación para la partícula  $x_n$  que representa la frontera inferior del sistema y que no cuenta con sus dos vecinas inferiores. Así las ecuaciones que se obtienen son:

$$\begin{aligned}
 m_1 \ddot{x}_1 &= -4kx_1 + kx_2 + kx_3 \\
 m_2 \ddot{x}_2 &= kx_1 - 4kx_2 + kx_3 + kx_4 \\
 m_i \ddot{x}_i &= kx_{i-2} + kx_{i-1} - 4kx_i + kx_{i+1} + kx_{i+2} \\
 m_{n-1} \ddot{x}_{n-1} &= kx_{n-3} + kx_{n-2} - 4kx_{n-1} + kx_n \\
 m_n \ddot{x}_n &= kx_{n-2} + kx_{n-1} - 4kx_n
 \end{aligned}$$



De la misma forma que en el modelo de primeros vecinos, estas ecuaciones de movimiento también generan un sistema de ecuaciones diferenciales. Al aplicar la solución general 3.12 y 3.13 obtenemos el siguiente sistema de ecuaciones representado en forma matricial:

$$\begin{pmatrix} \frac{4k}{m} & -\frac{k}{m} & -\frac{k}{m} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -\frac{k}{m} & \frac{4k}{m} & -\frac{k}{m} & -\frac{k}{m} & 0 & \cdots & 0 & 0 & 0 & 0 \\ -\frac{k}{m} & -\frac{k}{m} & \frac{4k}{m} & -\frac{k}{m} & -\frac{k}{m} & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -\frac{k}{m} & -\frac{k}{m} & \frac{4k}{m} & -\frac{k}{m} \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -\frac{k}{m} & -\frac{k}{m} & \frac{4k}{m} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_{n-1} \\ A_n \end{pmatrix} = \omega^2 \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_{n-1} \\ A_n \end{pmatrix} \quad (3.18)$$

Un ejemplo de este modelo para un sistema con 5 partículas quedaría representado de la siguiente manera:

$$\begin{pmatrix} \frac{4k}{m} & -\frac{k}{m} & -\frac{k}{m} & 0 & 0 \\ -\frac{k}{m} & \frac{4k}{m} & -\frac{k}{m} & -\frac{k}{m} & 0 \\ -\frac{k}{m} & -\frac{k}{m} & \frac{4k}{m} & -\frac{k}{m} & -\frac{k}{m} \\ 0 & -\frac{k}{m} & -\frac{k}{m} & \frac{4k}{m} & -\frac{k}{m} \\ 0 & 0 & -\frac{k}{m} & -\frac{k}{m} & \frac{4k}{m} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{pmatrix} = \omega^2 \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{pmatrix}$$

Este sistema de ecuaciones diferenciales correspondiente al modelo matemático con interacción a segundos vecinos, también puede resolverse obteniendo los valores propios y el vector propio correspondientes que representan las frecuencias y las amplitudes de los modos normales de vibración respectivamente.

### 3.4 Resumen

En este capítulo se detallan los pasos para la construcción del modelo del sistema. En la primer sección se describió la forma en como fue representado el modelo físico del sistema. El modelo que se obtuvo se trató como un sistema de modos normales de vibración con  $n$  grados de libertad.

En la segunda sección se describe la forma más sencilla del movimiento periódico, el Movimiento Armónico Simple (MAS). Se analizó la relación del MAS con nuestro modelo físico y se explicó la ecuación diferencial que representa a este tipo de movimiento. Además se presentó la solución general para los sistemas que están conformados por un MAS o un número de MAS.

Finalmente en la tercer sección se presentó la transformación del modelo físico del sistema al modelo matemático. Se desarrollaron los modelos matemáticos correspondientes a la interacción de primeros y segundos vecinos. Con estos modelos se obtuvo un sistema homogéneo de ecuaciones diferenciales para cada uno de ellos.



# Capítulo 4

## Diseño del sistema

La duda es la madre de la invención.

---

Galileo Galilei

El objetivo de este trabajo de investigación es el diseño y construcción de un sistema que nos permita predecir el comportamiento de las estructuras que se utilizan para la extracción de petróleo en aguas profundas. En los capítulos anteriores describimos el planteamiento del problema, los objetivos planteados, los antecedentes del sistema y finalmente el desarrollo de nuestro modelo físico y matemático.

En este capítulo planteamos el diseño de nuestro sistema, el cual nos proporciona una idea general de la funcionalidad de éste a través de sus componentes [33].

Generalmente los sistemas de simulación se descomponen en subsistemas y estos a su vez en módulos, los cuales proporcionan algún conjunto de servicios. El proceso de diseño que se lleva a cabo en este capítulo, nos ayudará a identificar los subsistemas y módulos del sistema, así como la comunicación que existe entre ellos.

El objetivo de este capítulo es dar a conocer el diseño de los diferentes subsistemas y módulos por los que está compuesto nuestro sistema. En cada uno de estos módulos nos enfrentamos a diversos problemas y para resolverlos fue necesario usar diversas herramientas y métodos numéricos, los cuales se describen en las siguientes secciones.

Un diseño modular como el que se usa, reduce la complejidad del sistema, facilita los cambios y la detección de errores en los módulos. Además, el mantenimiento de los módulos es más sencillo ya que atacan problemas específicos. Este diseño en módulos da como resultado una implementación más sencilla [34].

## 4.1 Diseño Conceptual

Al analizar los requerimientos del sistema descritos en nuestro planteamiento del problema y los objetivos de la investigación, se propuso dividir el simulador en dos subsistemas (Fig. 4.1)

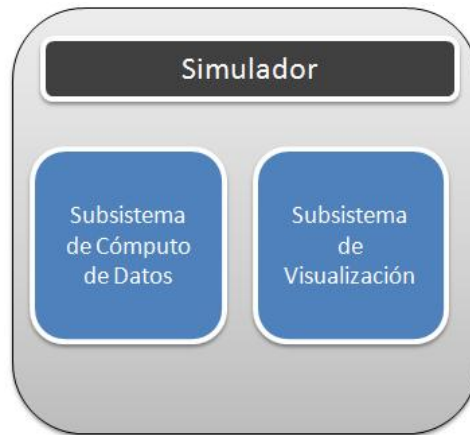


Figura 4.1: División del simulador en dos subsistemas

**Subsistema para el cómputo de datos:** El primer subsistema es el encargado de resolver el modelo matemático del *riser* propuesto en el capítulo anterior. Además, genera las condiciones de corrientes aleatorias a las que será sometido a prueba el sistema. Finalmente crea archivos con los resultados de las pruebas, que nos permiten determinar el comportamiento dinámico del *riser* en las condiciones de corrientes generadas. El subsistema de cómputo de datos está formado por los siguientes módulos (Fig. 4.2):

1. Módulo del modelo matemático del *riser*.
2. Módulo del modelo hidrodinámico de corrientes

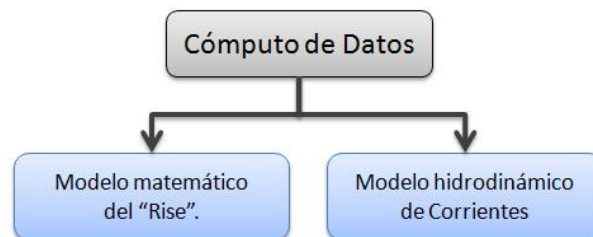


Figura 4.2: Módulos que componen el subsistema de cómputo de datos

**Subsistema de Visualización de resultados:** Es el encargado de tomar los resultados generados por el primer subsistema para convertirlos en imágenes y animaciones,

que permitan representar de una forma sencilla el comportamiento del sistema. Este subsistema está conformado por los siguientes módulos (Fig. 4.3):

1. Gráfica de Resultados.
2. Animación de partículas 2D.
3. Visualización Interactiva 3D.

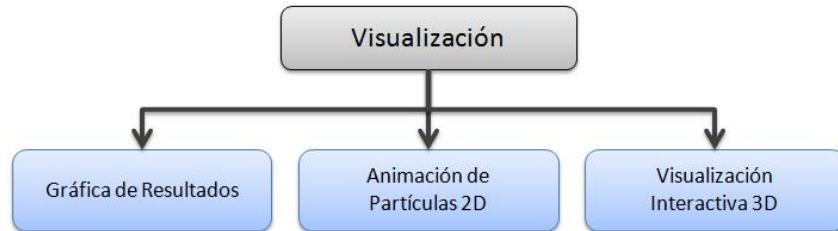


Figura 4.3: Módulos que componen el subsistema de Visualización

En resumen el diseño general del sistema está estructurado como se muestra en la figura 4.4.

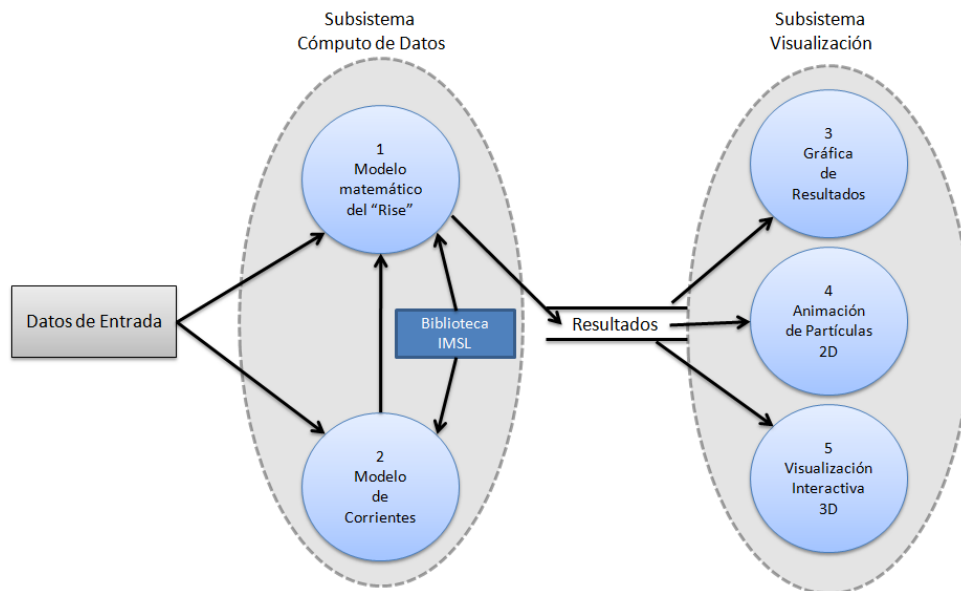


Figura 4.4: Representación general del sistema

## 4.2 Modelo del *riser* (cadenas de vibración)

Como vimos en el Capítulo 3, el modelo de nuestro sistema es tratado como una cadena de vibración monoatómica, formada por partículas de masa equidistantes unidas por

resortes. De este modelo matemático se obtuvieron dos matrices de movimiento, una para las partículas con interacción a primeros vecinos (Ecuación 3.14) y la segunda para partículas con interacción a segundos vecinos (Ecuación 3.18).

Las matrices de movimiento para los dos modelos matemáticos que se desarrollaron, nos generan un sistema de ecuaciones diferenciales de segundo orden. Para resolver este sistema de ecuaciones, se obtienen el vector y los valores propios correspondientes al sistema, que representan las frecuencias y las amplitudes de las partículas respectivamente.

### 4.2.1 Problema de valores propios

El problema de calcular valores propios ( $\lambda$ ) se puede presentar en diversos campos de ingeniería y las matemáticas, como son: ecuaciones diferenciales, estabilidad de sistemas lineales, sistemas eléctricos, ect. Los valores y vectores propios se definen formalmente de la siguiente manera:

*Si  $A$  es una matriz de  $n \times n$ . Un vector  $v$  distinto de cero es un **vector propio** de  $A$  si para cierto escalar de  $\lambda$ ,*

$$Av = \lambda v \quad (4.1)$$

*donde  $\lambda$  es un **valor propio** de  $A$  asociado con el vector  $v$ .*

Como se definió en el capítulo 3, el sistema de ecuaciones homogéneo que se obtuvo para los modelos de primeros y segundos vecinos se puede representar de la forma:

$$MX = \lambda X \quad (4.2)$$

La ecuación 4.2 se puede representar como una *Ecuación diferencial para los Modos Normales de Vibración* [31]

$$\ddot{X} = \lambda X \quad (4.3)$$

donde  $\lambda$  representa las frecuencias naturales o de resonancia del sistema.

Para resolver el sistema (Ecuación 4.2 y 4.3) es necesario determinar las frecuencias naturales o de resonancia y los modos normales de vibración. Esto implica encontrar los valores propios ( $\lambda$ ) y los vectores propios ( $X$ ) de la matriz de movimiento ( $M$ ).

Debido que nuestro sistema está representado por una cadena de vibración homogénea, es posible la resolución del problema por medios analíticos. De otra forma, sería necesario utilizar métodos numéricos para obtener espectros de frecuencia y de esta manera obtener los modos normales de vibración.[31].

Las soluciones obtenidas de las matrices de movimiento (3.14 y 3.18), son llamadas vibración libre o frecuencias de resonancia del sistema (valores propios  $\lambda_1, \lambda_2, \dots, \lambda_n$ ) y dan la variación del movimiento de las partículas del sistema por unidad de tiempo. Por su parte los elementos del vector propio ( $X$ ) se denominan coordenadas normales del sistema. Cada una de estas coordenadas corresponde a una vibración del sistema con una sola frecuencia. Por lo tanto al conjunto de los componentes de la oscilación

que conforman el vector propio se denominan *Modos Normales de Vibración*[32]. Particularmente en nuestro sistema todas las partículas oscilan con la misma frecuencia y fase, por lo que la vibración total de nuestro sistema será el resultado de la suma de los modos normales (superposición de los modos de vibración).

### 4.2.2 Análisis de Fourier

En diversos problemas de análisis de ingeniería, es necesario trabajar con funciones periódicas, es decir, funciones que se repiten cada cierto intervalo de tiempo y donde este intervalo es conocido como *periodo*. Las funciones trigonométricas son funciones periódicas se utilizan comúnmente y son relativamente fáciles de trabajar ya que tienen la propiedad de ortogonalidad. Debido a estas ventajas, es recomendable expandir la función periódica en funciones trigonométricas, donde esa expansión se conoce como *serie de Fourier* [35]. El análisis de Fourier es la clave para el análisis de la señales de vibración [36].

Debido a que las ecuaciones obtenidas para nuestro sistema son ecuaciones lineales, podemos aplicar el *principio de superposición* que nos dice que la suma de las soluciones obtenidas es también una solución. Sin embargo debido a que nuestro sistema esta compuesto de  $n$  modos vibración y  $n$  frecuencias naturales, se debe de considerar la suma de  $n$  soluciones armónicas. Esto se obtiene mediante una solución de la *Serie de Fourier* [37].

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} [a_n \cos(\omega_n t) + b_n \sin(\omega_n t)] \quad (4.4)$$

donde  $\omega$  se conoce como la frecuencia,  $a_n(n = 1, 2, \dots)$  y  $b_n(n = 1, 2, \dots)$  son conocidos como los *coeficientes de Fourier* de la ecuación  $f(x)$

De esta manera, por medio de una *serie de Fourier*, se puede determinar que un movimiento periódico de vibración esta formado por un número de MAS de diferentes amplitudes y frecuencias.

Bajo ciertas circunstancias las series de Fourier se pueden simplificar. Esto se da cuando se tiene una función periódica par (Ecuación 4.5) y otra impar (Ecuación 4.6) que satisfagan ciertas relaciones de simetría, entonces la serie de Fourier puede no contener cualquiera de los dos términos [35].

$$f(t) = f(-t) \quad (4.5)$$

$$f(t) = -f(-t) \quad (4.6)$$

Tomando en consideración lo anterior para nuestro modelo, las funciones periódicas las podemos expandir a una *serie de senos de Fourier* (Ecuación 4.7).

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} [b_n \sin(\omega_n t)] \quad (4.7)$$

De esta manera, la ecuación 4.7 se va aplicar a nuestro modelo para obtener el comportamiento general del sistema con sus frecuencias naturales de resonancia y con las frecuencias aleatorias generadas con nuestro modelo de corrientes basada en el método de Monte Carlo (el cual se describirá más adelante).

### 4.3 Modelo de corrientes basado en Monte Carlo

Debido a que el sistema se ve afectado por un fenómeno natural (en nuestro caso las corrientes marinas), es necesario el desarrollo de un modelo estocástico de corrientes, el cual es independiente al modelo del *riser*. Este modelo genera condiciones de fuerza y frecuencias aleatorias de corrientes que afectan al sistema, y se utiliza para probar que el modelo matemático que se desarrolló funciona correctamente en diversas situaciones.

Para poder desarrollar este tipo de modelos estocásticos, es necesario la generación de números aleatorios que permitan la representación de fenómenos como el de las corrientes marinas, ya que para este tipo de sistemas es poco práctico poder examinar todos los casos o condiciones posibles.

Antes de comenzar a generar las frecuencias aleatorias, fue necesario acondicionar nuestro modelo físico de tal forma, que nos permitiera producir diversas condiciones de corrientes en diferentes segmentos del *riser*.

Para llevar a cabo lo anterior, además de la división en partículas, el *riser* se dividió en varios segmentos. Esta división nos va a permitir determinar condiciones específicas para la generación de frecuencias en un segmento determinado (Fig. 4.5). Lo anterior nos va a permitir simular condiciones de corrientes en dos sentidos diferentes.



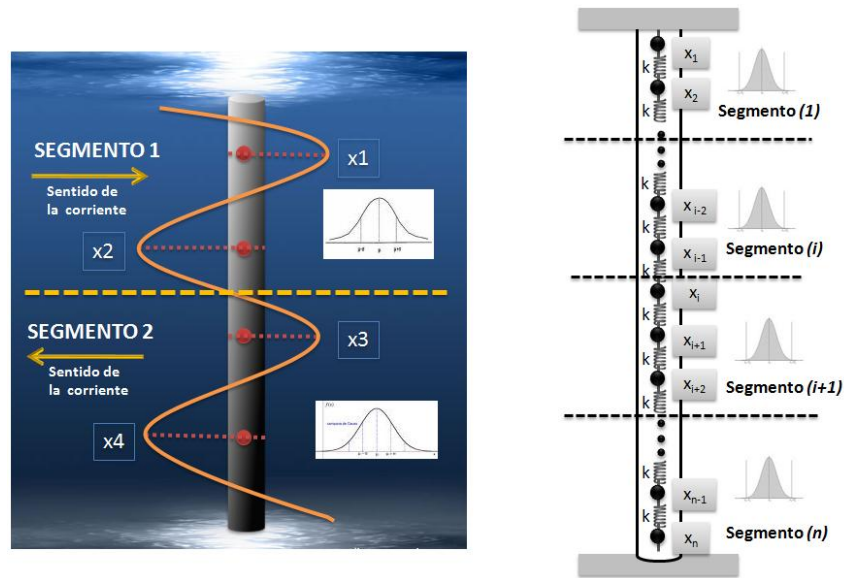


Figura 4.5: División del *riser* en partículas y segmentos

Una vez que se hicieron las modificaciones a nuestro modelo físico, se diseñó el proceso encargado de generar las frecuencias aleatorias, a las cuales será sometido nuestro sistema para analizar su comportamiento.

Para la generación de los números aleatorios utilizamos un método numérico no determinístico conocido como *Método Monte Carlo*, con el cual logramos simular las variables aleatorias que representan la fuerza y frecuencias de excitación del sistema.

### 4.3.1 Método Monte Carlo

El método *Monte Carlo* es utilizado en diferente áreas de la ciencia (física, matemáticas, biología, etc.), sin embargo en un inicio este método no tuvo mucho auge debido a la gran cantidad de cálculos que se necesitan realizar para resolver diversos problemas. Fue hasta después de la invención de la computadora que tuvo una gran impacto, ya que estas herramientas permiten realizar un gran número de cálculos en un experimento numérico y se tiene una gran fiabilidad en sus resultados. Con ayuda de las computadoras la solución de un problema o experimento numérico en el que se implemente el método Monte Carlo, es mucho más rápido [38].

La función principal del método *Monte Carlo* consiste en la generación de números aleatorios. Para generarlos, este método requiere utilizar una distribución de probabilidad, con la cual se pueden crear muestras con un comportamiento determinado [38].

La implementación de este método, fue necesario para generar el desalajo de cargas de frecuencia para cada una de las partículas que componen nuestra cadena de vibración.

Para cada una de las partículas de masa del *riser* es necesario generar  $n$  veces un mismo vector que contenga  $m$  números aleatorios, es decir, que para cada partícula se necesitan  $m$  números aleatorios (frecuencias) y cada uno de estos números se deben generar  $n$  veces (Fig. 4.6). Cada uno de estos números del vector representan las frecuencias que afectan a cada una de las partículas en una unidad de tiempo.

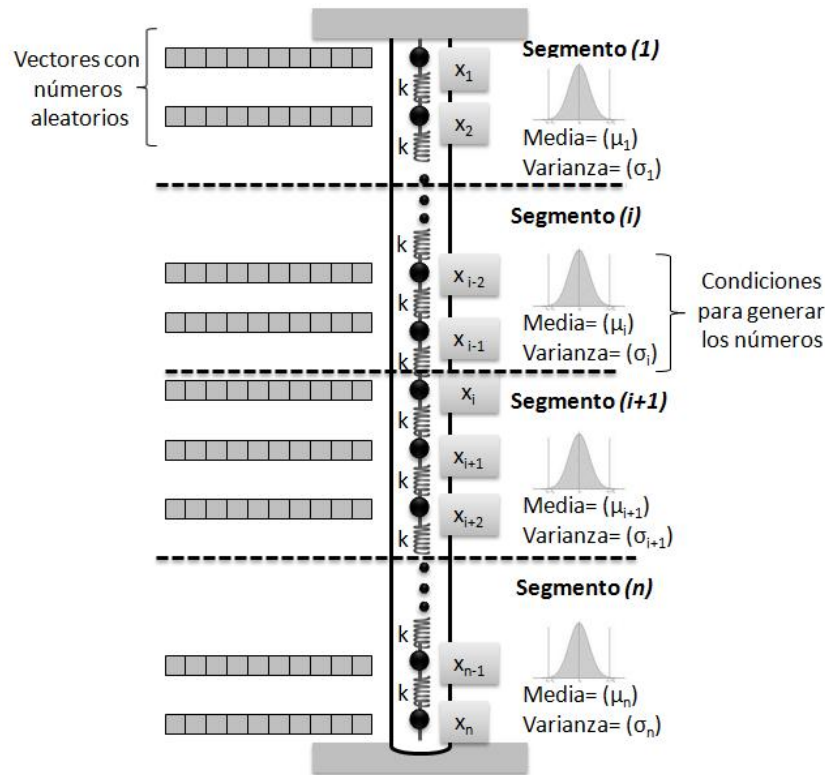


Figura 4.6: Generación de frecuencias aleatorias para cada partícula, con diferentes condiciones en cada segmento

La generación de cada uno de estos números aleatorios se realiza en base a una *Distribución Normal*, en la cual los valores de media ( $\mu$ ) y varianza ( $\sigma^2$ ) determinan las condiciones para generar cada uno de estos números aleatorios (esto se describirá más a detalle en la siguiente sección). Es importante mencionar que cada uno de los segmentos en los cuales se dividió nuestro sistema, deberá tener diferentes valores de media y varianza con lo cual representaremos diferentes condiciones de corrientes.

Cabe mencionar que nuestro modelo hidrodinámico de corrientes debe estar basado en los *modos normales de vibración* de nuestro sistema. Por esta razón, las frecuencias aleatorias para cada una de nuestras partículas se obtiene del producto del vector de números aleatorios por la frecuencia normal de vibración de cada partícula; es decir, el vector de cada partícula se multiplica por su valor propio correspondiente (ecuación 4.8).

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_i \\ \vdots \\ \omega_m \end{pmatrix} = \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_i \\ \vdots \\ A_m \end{pmatrix} \omega \quad (4.8)$$

La aproximación de resultados reales en el método *Monte Carlo*, depende de diversos factores como lo es la exactitud de los valores, el número de experimentos realizados (en ese caso, el tamaño del vector de números aleatorios) y las varianzas de la muestra [38].

### 4.3.2 Distribución normal

La distribución *Normal* o *Gaussiana*, es la distribución de probabilidad que más se utiliza para modelar fenómenos naturales. Esta distribución cuenta con una gráfica en forma de campana, la cual se denomina *curva normal* y es simétrica respecto a un parámetro [39] (Fig. 4.7).

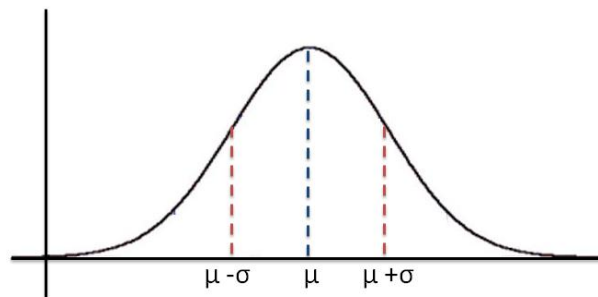


Figura 4.7: Campa de la distribución normal.

Una variable aleatoria normal tiene una ecuación matemática que depende de dos parámetros: media ( $\mu$ ) y desviación estándar ( $\sigma$ ).  $n(x : \mu, \sigma)$  y su función de densidad esta dada por:

$$n(x : \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(1/2)[(x-\mu)/\sigma]^2} \quad (4.9)$$

Estos dos parámetros determinan la forma como se comporta la gráfica de la curva.

Esta distribución de probabilidad nos permite describir una gran cantidad de fenómenos naturales, industriales, económicos y de investigación. Cuando este tipo de fenómenos son estudiados, las muestras que se toman de ellos tienden hacia una distribución normal, conforme se aumenta el tamaño de las muestras [38].

Con esta distribución de probabilidad vamos a generar los números aleatorios que se requieren para implementar el método *Monte Carlo*. Como se comentó en la sección

anterior, estos números aleatorios se deben generar para cada una de las partículas que conforman el sistema, en donde los valores de media y varianza deben ser diferentes para cada uno de los segmentos en los que se dividió el sistema (Fig. 4.6).

Para la generar estos números aleatorios, primero es necesario definir el rango en el cual deben estar los valores generados. Para el caso de nuestro sistema, los valores deberán estar en un rango de  $-1 < x < 1$  (Fig. 4.8). Esto será una restricción al momento de solicitar los valores de  $\mu$  y  $\sigma$ .

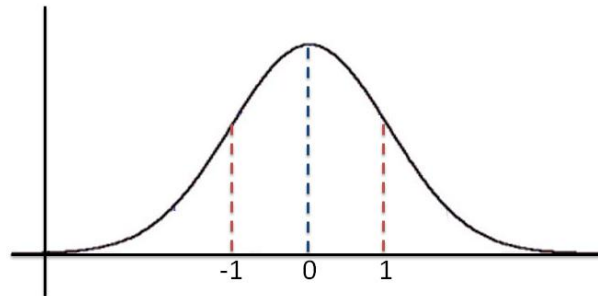


Figura 4.8: Distribución normal con rangos de  $-1$  a  $1$ .

## 4.4 Visualización

Una vez que se tuvo el diseño de los módulos del primer subsistema, se diseñó el segundo subsistema, el cual permite la visualización de los resultados obtenidos por los módulos anteriores.

La *visualización científica* nos permite transformar la información y los datos generados por los módulos anteriores en imágenes y animaciones, que permitan expresar y comunicar de una manera más sencilla el comportamiento de nuestro sistema. La simulación del fenómeno que estamos desarrollando nos va a producir una gran cantidad de información numérica, por lo que resulta muy complicado observar e interpretar los resultados. Afortunadamente, en la actualidad se cuenta con un gran poder de cómputo y se han desarrollado técnicas que nos facilitan la interpretación visual de la información. Estas técnicas se conocen como *visualización científica* [40].

Debido a que el modelo del sistema depende del tiempo, es importante el desarrollo de animaciones y visualizaciones interactivas que permitan realizar un análisis más completo de la información. Una buena visualización nos permite explicar a personas que no son expertas en el tema lo que está sucediendo en el fenómeno simulado. Sin embargo, es importante mencionar que las imágenes y animaciones que se generen, dependen directamente de los algoritmos de extracción de datos, por esto es necesario estar completamente seguros de su correcto funcionamiento.

Para poder interpretar los resultados de nuestro sistema se diseñaron tres módulos de visualización:

1. **Gráfica de Resultados:** El primer módulo de visualización consiste en el uso

de gráficas en dos dimensiones  $(x, y)$ , en el cual se van a representar los datos generados por cada partícula del sistema. Este módulo nos permite comparar el comportamiento y tendencias de todas las partículas.

2. **Animación de Partículas 2D:** Este módulo consiste en una animación en 2D en el cual se podrá observar el comportamiento de cada partícula en cada unidad de tiempo.
3. **Visualización Interactiva 3D:** El tercer módulo consiste una animación 3D en el cual vamos a poder observar el comportamiento de todo el sistema en conjunto. Esta visualización nos permite cambiar el punto de vista del sistema mediante rotaciones (de los ejes  $x, y$  y  $z$ ) y acercamientos.

## 4.5 Resumen

En este capítulo se presentó el proceso general de nuestro simulador. En la primera sección se describieron los subsistemas y módulos por los cuales esta conformado nuestro simulador. Se analizó la interacción entre los procesos y se desarrolló un diagrama general.

En la segunda sección se analizó el modelo del *riser* y los fundamentos teóricos que se utilizaron para resolver cada una de las partes del modelo.

En la tercera sección se describe el desarrollo del modelo hidrodinámico de corrientes en donde se describió como se lleva a cabo la implementación del método Monte Carlo para la generación de las frecuencias aleatorias de las corrientes marinas.

Finalmente, en la cuarta sección se describe el subsistema de visualización y los módulos por los que está conformado (Gáfica de Resultados, Animación de Partículas 2D y Visualización Interactiva 3D).



# Capítulo 5

## Desarrollo e implementación del sistema

Actuar es fácil, pensar es difícil;  
actuar según se piensa es aún más  
difícil.

---

Johann Wolfgang Goethe

**E**n el capítulo anterior tratamos el diseño de nuestro sistema, en el cual se describen los módulos que lo conforman y la interacción que hay entre ellos. Este diseño es la base para el desarrollo de nuestro sistema. Seguir estas especificaciones de diseño reduce el riesgo de errores durante la etapa de desarrollo.

Como se describe en el Capítulo 4, el sistema que se desarrolló está formado de dos subsistemas que se encargan de realizar los cálculos para obtener los modos de vibración del sistema y posteriormente desplegarlos en una animación gráfica.

En este capítulo se presenta la arquitectura general de nuestro sistema. Además, se describen los algoritmos de programación que se implementaron para desarrollar cada uno de los módulos que conforman el sistema. Cada uno de los módulos que se desarrollaron utilizan diversas herramientas de hardware y software las cuales también se describen en este capítulo.

### 5.1 Arquitectura general

Una vez que se tuvo el diseño de nuestro sistema y que se tuvieron definidos los subsistemas y cada uno de los módulos que los conformaban, el siguiente paso consistió en definir la arquitectura para nuestro sistema. Para esto se realizó un análisis para

definir la plataforma, el lenguaje y las bibliotecas necesarias para poder llevar a cabo la programación de los subsistemas de nuestro simulador. La arquitectura general de nuestro simulador esta definida como se muestra en la figura 5.1.

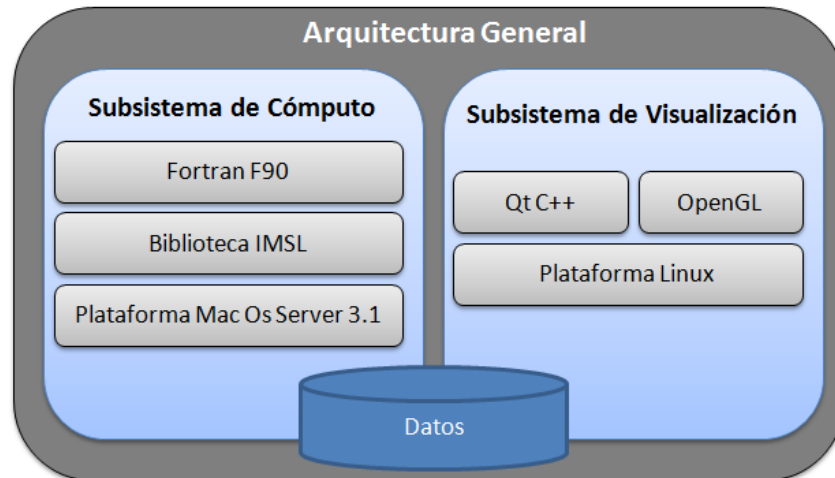


Figura 5.1: Arquitectura general del sistema

En la figura 5.1 se presentan los componentes de la arquitectura para cada subsistema. Estos componentes de la arquitectura se describen a continuación.

### 5.1.1 Plataforma del subsistema de cómputo de datos

Debido a que la solución del modelo matemático y el modelo de corrientes requieren un gran poder de cómputo, el subsistema de cómputo de datos se desarrolló en un servidor el cual cuenta con un procesador Quad Core de 8 núcleos. El servidor trabaja con un sistema operativo *Mac OS Server 3.1*. Este sistema operativo es usado en servidores de la familia *Macintosh* y es comercializado por Apple.

Por su parte el desarrollo de esta aplicación se realizó en lenguaje *Fortan F90*. Este lenguaje nos permite tener una interacción con la *biblioteca IMSL*, la cual proporciona rutinas que permitan resolver el modelo matemático e implementar el método Monte Carlo para generar las frecuencias aleatorias de corrientes.

### 5.1.2 Plataforma del subsistema de visualización

El desarrollo de este subsistema se llevó a cabo en una computadora con menor poder de cómputo. La computadora que se utilizó para su desarrollo cuenta con un procesador AMD con 1GB en RAM, con sistema operativo *Linux*.

Para el desarrollo de esta aplicación de visualización, se utilizaron diversas herramientas de software libre. En primer lugar el sistema se diseñó para trabajar en plataformas basadas en *Linux*. El lenguaje base para el desarrollo de esta aplicación fue



C++ con ayuda de la biblioteca *QT* versión 4.3.4, la cual nos permite crear aplicaciones con una interfaz gráfica. Una de las ventajas de trabajar con este lenguaje y esta biblioteca, es su disponibilidad para varias plataformas como Mac OS, Windows, sistemas Unix, Linux, inclusive para algunos sistemas operativos para dispositivos móviles.

Otra de las herramientas que se utilizaron para desarrollar esta aplicación, fue la implementación de *OpenGL* (*Open Graphics Library*), la cual es una interfaz de software que nos permite hacer uso eficiente de las tarjetas aceleradoras de gráficos. Esta interfaz se compone de más de 250 funciones que permiten programar gráficos en dos o tres dimensiones, los cuales se deben de crear a partir de primitivas geométricas como puntos, líneas y polígonos.

## 5.2 Desarrollo del subsistema de cómputo de datos

Para llevar a cabo el desarrollo del subsistema de Cómputo de Datos, se elaboró un diagrama de flujo de datos (*DFD*) para identificar los procesos que conforman a los dos módulos generales de este subsistema y la relación que existe entre ellos (Fig. 5.2).

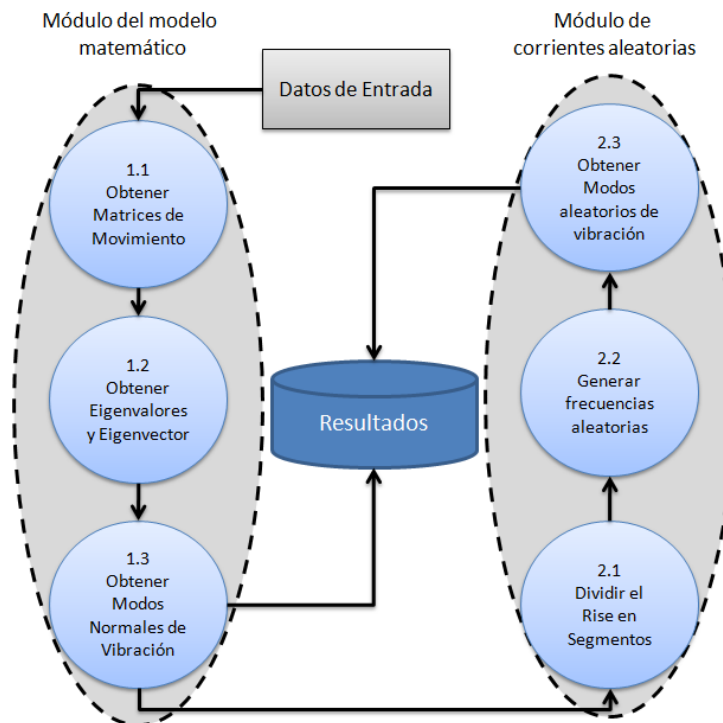


Figura 5.2: Diagrama de flujo de datos del Subsistema de Cómputo de Datos

En la figura 5.2 podemos observar que para comenzar la ejecución del sistema se requieren de algunos datos iniciales, los cuales serán proporcionados por el usuario. Los datos iniciales determinan las condiciones generales para ejecutar la simulación. Los datos de entrada que se requieren son:

- *Número de partículas ( $n$ )*: Define el número de partículas en las que se va a dividir el *riser* y sobre las cuales se realizará el análisis de vibraciones.
- *Masa de las partículas ( $m$ )*: Especifica la masa que va a tener cada una de las partículas del sistema. Debido a que nuestro sistema es homogéneo, solo es necesario definir la masa de una de las partículas.
- *Constante de elasticidad ( $k$ )*: Define la constante de elasticidad de los resortes que unen a las partículas de masa de nuestra cadena de vibración.
- *Longitud del riser ( $Long$ )*: Con este dato vamos a establecer la longitud que tendrá nuestro *riser* y realizaremos la distribución de las partículas.
- *Numero de Segmentos ( $NumSeg$ )*: Especifica el número de segmentos en los que vamos a dividir el *riser*, en donde cada uno de estos tendrán diferentes condiciones para generar las frecuencias de corrientes aleatorias.

Con estos datos iniciales podremos comenzar a ejecutar los procesos del subsistema, los cuales generan una serie de archivos que contienen los resultados de la simulación.

En los procesos que se muestran en el DFD de la figura 5.2, se contemplan las rutinas para generar los cálculos de los dos modelos matemáticos que se desarrollaron (primeros y segundos vecinos). El proceso para el desarrollo de cada uno de estos módulos se describen en las siguientes secciones.

### 5.2.1 Proceso para obtener las matrices de movimiento

Las matrices de movimiento son el resultado de las ecuaciones obtenidas por los modelos matemáticos (primeros y segundos vecinos) que se desarrollaron en el Capítulo 3. A través de ellas podemos obtener los modos normales de vibración del sistema con las condiciones iniciales que se definen previamente.

Para obtener estas matrices de movimiento, es necesario contar con diversos datos iniciales como son: el número de partículas, la masa de las partículas y constante de elasticidad de los resortes. Con estos datos se desarrolló el proceso que se encarga de generar las matrices de movimiento para los modelos de primeros y segundos vecinos (Ecuaciones 3.14 y 3.18).

Para almacenar los datos de nuestras matrices de movimiento, se necesitó de dos matrices cuadradas cuyo tamaño está definido por el número de partículas en las que se divide el *riser*. Posteriormente, se desarrolló un algoritmo para llenar éstas matrices con los datos correspondientes a los modelos de primeros y segundos vecinos obtenidos en el Capítulo 3.

Este algoritmo se encarga de recorrer todos los elementos de la matriz de movimiento. Cuando el algoritmo identifica que está en un elemento de la diagonal principal asigna el valor correspondiente (dependiendo si el modelo es de primeros o segundos vecinos). Además, se encarga de asignar a los elementos vecinos de la diagonal principal el valor correspondiente a un vecino. Finalmente a los elementos que

no pertenecen a la diagonal principal y que no son vecinos de estos elementos se les asigna un valor de 0.

---

### Algoritmo 1 Genera Matriz de Movimiento

---

**Require:** Número de partículas  $n$ , Masa  $m$ , Elasticidad  $k$ , Matriz  $Matrix[n][n]$

```

1: for  $i = 0$  to  $n$  do
2:   {Se recorren las filas de la matriz}
3:   for  $j = 0$  to  $n$  do
4:     {Se recorren las columnas de la matriz}
5:     if  $i = j$  then
6:       {Si es un elemento de la diagonal principal}
7:        $Matrix[i][j] \leftarrow (2 * k/m)$  {Para matriz de Primeros Vecinos}
8:        $Matrix[i][j] \leftarrow (4 * k/m)$  {Para matriz de Segundos Vecinos}
9:     else if  $j = (i - 1) \vee j = (i + 1) \vee j = (i - 2) \vee j = (i + 2)$  then
10:      {Si es un elemento vecino}
11:       $Matrix[i][j] \leftarrow -(k/m)$ 
12:    else
13:      {Si es cualquier otro elemento}
14:       $Matrix[i][j] \leftarrow 0$ 
15:    end if
16:  end for

```

---

## 5.2.2 Proceso para obtener los valores y vectores propios

Una vez que se tuvieron las matrices de movimiento correspondientes a los dos modelos matemáticos que se desarrollaron, el siguiente paso fue resolver el sistema de ecuaciones. Como se describió en el Capítulo 4, este proceso se realiza obteniendo los valores propios y el vector propio correspondientes a cada uno de los sistemas de ecuaciones desarrollados.

Para obtener los valores y vectores propios correspondientes se usó una función matemática de la biblioteca IMSL:

*CALL EVCRG(N, MATIRX, LDA, EVAL, EVEC, LDVEC)*

*Esta función de la biblioteca IMSL se encarga de calcular todos los valores propios y el vector propio de una matriz con datos de tipo real.*

Los parámetros que recibe la función *EVCRG* son:

- $N$ : Nos indica el orden de la matriz (Dato de entrada).

- *MATRIX*: Es un arreglo bidimensional de punto flotante que contiene los datos de la matriz (Dato de entrada).
- *LDA*: Valor de tipo entero que contiene la dimensión de la matriz tal como se hizo en su declaración (Dato de entrada).
- *EVAL*: Arreglo de tamaño  $N$  que contienen los valores propios obtenidos de la matriz (Datos de salida).
- *EVEC*: Contiene la matriz de vectores propios obtenidos (Datos de Salida).
- *LDVEC*: Valor de tipo entero que contiene la dimensión de la matriz de vectores propios *EVEC* (Datos de entrada).

La función se encarga en primer lugar de balancear la matriz utilizando una transformación ortogonal. Para el cálculo de los valores y vectores propios se utiliza el *algoritmo QR*, que consiste en obtener el producto de una matriz ortogonal por una matriz triangular superior (ecuación 5.1) [41].

$$MATRIX = QR \quad (5.1)$$

donde  $Q$  representa la matriz ortogonal y  $R$  es la matriz triangular superior.

### 5.2.3 Proceso para obtener modos normales de vibración

Después de realizar el cálculo de los valores y vectores propios correspondientes a las matrices de movimiento de los modelos que se desarrollaron, el siguiente proceso consistió en obtener los modos normales de vibración y la superposición de los movimientos del sistema por cada unidad de tiempo.

Para esto fue necesario implementar un algoritmo, el cual ejecuta un ciclo que representa el tiempo. En cada iteración se recorre el arreglo de valores y vectores propios para calcular los modos normales de vibración del sistema aplicando la ecuación 5.2.

$$x_i = A_i \sin(\omega t) \quad (5.2)$$

En este mismo ciclo se calcula la superposición de estos modos normales para cada partícula. Para realizar este cálculo se implemento la ecuación 5.3.

$$x_i = \sum_{n=1}^{\infty} [A_i \sin(\omega_n t)] \quad (5.3)$$

Para almacenar los resultados de los cálculos, el algoritmo requiere de una matriz para los modos normales de vibración y un vector para almacenar los resultados de la sumatoria. Sin embargo, al final de nuestro algoritmo estos resultados son almacenados en archivos los cuales se utilizan en los módulos de visualización para hacer la representación grafica. Este proceso queda representado en el algoritmo 2.

**Algoritmo 2** Modos Normales de Vibración

---

**Require:** Matriz de vectores propios para primeros y segundos vecinos  $EVECM1$ ,  $EVECM2$ . Vector de valores propios para primeros y segundos vecinos  $EVALM1$ ,  $EVALM2$ . Número de partículas  $n$ . Límite de Tiempo  $lim$ .

- 1: {Se declaran las matrices para almacenar los modos normales de vibración}
- 2: **real**  $ResultM1[n][n]$ ,  $ResultM2[n][n]$
- 3: {Se declaran los vectores para almacenar la superposición de las vibraciones}
- 4: **real**  $SumaM1[n]$ ,  $SumaM2[n]$
- 5: **for**  $t = 0$  to  $lim$  **do**
- 6:   {Se recorren todos los valores propios}
- 7:   **for**  $i = 0$  to  $n$  **do**
- 8:     {Se inicializa en cero el valor de la sumatoria para la superposición}
- 9:      $SumaM1[i] \leftarrow 0$
- 10:     $SumaM2[i] \leftarrow 0$
- 11:    {Se recorre todos los vectores propios}
- 12:    **for**  $j = 0$  to  $n$  **do**
- 13:     {Se calculan los modos normales de vibracion de cada partícula}
- 14:      $ResultM1[i][j] \leftarrow EVECM1[i][j] * \sin(EVALM1[i] * t)$
- 15:      $ResultM2[i][j] \leftarrow EVECM2[i][j] * \sin(EVALM2[i] * t)$
- 16:     {Se calcula la superposición de los modos normales}
- 17:      $SumaM1[i] \leftarrow SumaM1[i] + EVECM1[i][j] * \sin(EVALM1[i] * t)$
- 18:      $SumaM2[i] \leftarrow SumaM2[i] + EVECM2[i][j] * \sin(EVALM2[i] * t)$
- 19:    **end for**
- 20:    {Se almacenan los resultados en archivos}
- 21:     $ModNormVec1.txt \leftarrow ResultM1$
- 22:     $ModNormVec2.txt \leftarrow ResultM2$
- 23:    {Se almacenan los resultados de la superposición}
- 24:     $SumaNormVec1.txt \leftarrow SumaM1$
- 25:     $SumaNormVec2.txt \leftarrow SumaM2$
- 26:    **end for**
- 27: **end for**

---

**5.2.4 Proceso para dividir el riser en segmentos**

Como se describió en los Capítulos 3 y 4, para realizar el cálculo de los modos aleatorios de vibración, primeramente se requería dividir el riser en varios segmentos, los cuales van a determinar las condiciones para generar las frecuencias aleatorias para cada una de las partículas de dichos segmentos.

Para llevar a cabo la división de segmentos se implementó un algoritmo en el cual obtiene una relación del número de partículas por segmento de tal forma que cada segmento tenga un número de partículas equivalente o aproximado. El proceso se describe en el algoritmo 3.

**Algoritmo 3** División de Segmentos

---

**Require:** Vector para asignar segmentos *AsignaSeg*. Número de partículas *n*. Número de segmentos *NumSeg*

- 1: {Variables para almacenar el número de partículas por segmento y restantes}
- 2: integer *NodosSeg*, *NodosRestan*, *iNodo* = 1
- 3: {Se obtiene el número de nodos por segmento}
- 4:  $NodosSeg \leftarrow n / NumSeg$
- 5: {Obtenemos los nodos que restan de la división}
- 6:  $NodosRestan \leftarrow n \text{ MODULO } NumSeg$
- 7: {Se recorren los segmentos para asignarles sus nodos correspondientes}
- 8: **for** *iSeg* = 1 to *NumSeg* **do**
- 9:   {Asignamos los nodos que le tocan al segmento}
- 10:   **for** *j* = 1 to *NodosSeg* **do**
- 11:      $AsignaSeg[iNodo] \leftarrow iSeg$
- 12:      $iNodo \leftarrow iNodo + 1$
- 13:   **end for**
- 14:   {Si el segmento es menor al número de nodos restantes}
- 15:   **if** *iSeg* <= *NodosRestan* **then**
- 16:     {Asignamos un nodo mas a este segmento}
- 17:      $AsignaSeg[iNodo] \leftarrow iSeg$
- 18:      $iNodo \leftarrow iNodo + 1$
- 19:   **end if**
- 20: **end for**

---

El algoritmo requiere como datos de entrada el número de partículas en las que se dividió el *riser*, el número de segmentos que se requiere tener y por último un vector donde se tendrá una relación que indica a que segmento pertenece cada una de las partículas. El proceso consiste en dividir el número de partículas entre el número de segmentos, para asignar un número de partículas equivalentes. Posteriormente, las partículas restantes de la división son asignadas entre los primeros segmentos, esto implica que en algunos casos no se tenga el mismo número de partículas en cada segmento. La asignación de las partículas se lleva a cabo en un arreglo de datos de tipo entero, en donde el índice del vector indica el número de la partícula y el contenido hace referencia al segmento al que pertenece. La asignación se puede observar en la figura 5.3

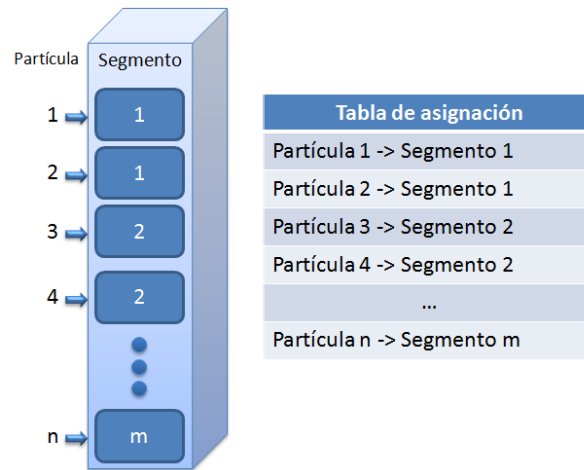


Figura 5.3: Asignación de partículas por segmento

### 5.2.5 Proceso para generar frecuencias aleatorias

Una vez que se tuvo la división del sistema en segmentos y que se realizó la asignación de las partículas a cada uno de ellos, el siguiente proceso consistió en implementar el *Método Montecarlo* para generar las frecuencias de corriente aleatorias para cada una de las partículas.

Como se describió en el capítulo 4, para generar las frecuencias aleatorias mediante el *Método Montecarlo*, fue necesario utilizar la *distribución Normal*. Esta distribución de probabilidad nos va a permitir determinar las condiciones (media  $\mu$  y varianza  $\sigma$ ) para generar los números aleatorios en cada segmento.

Para llevar a cabo este proceso se desarrolló un algoritmo el cual hace uso de las funciones de la *Biblioteca IMSL*, lo que nos permite generar números aleatorios basados en una distribución normal con valores de media  $\mu$  y varianza  $\sigma$  distintos.

El algoritmo que se implementó (algoritmo 4), requiere como entradas el número de segmentos, el número de partículas, el vector de asignación de partículas con segmentos y los vectores de valores propios correspondientes a los dos modelos (primeros y segundos vecinos). Lo primero que se tiene que definir es el tamaño del vector de frecuencias para cada una de las partículas y el número de repeticiones que se tiene hacer para generarlo.

El proceso que realiza el algoritmo se encarga de recorrer los segmentos en los que se dividió el *riser*. Para cada uno de los segmentos, se producen o se solicitan (según se seleccione) los datos de  $\mu$  y  $\sigma$  que serán utilizados para generar las frecuencias aleatorias. Posteriormente, se recorren cada una de las partículas que fueron asignadas al segmento para generar su vector de frecuencias. Cada vector de frecuencias es generado un número de veces determinado (en este caso 1000 veces) antes de ser asignado a la partícula. Finalmente, las frecuencias generadas deben estar basadas en los *modos normales de vibración* del sistema, por lo que el vector de números aleatorios generado

se multiplica por el valor propio obtenido para la partícula y se almacena en una matriz de frecuencias aleatorias.

---

**Algoritmo 4** Frecuencias aleatorias con Monte Carlo

---

**Require:** Matriz de frecuencias aleatorias  $MatrixRandom1, MatrixRandom2$ .

Número de segmentos  $NumSeg$ . Vector de asignación de segmento  $AsignaSeg$ .

Número de partículas  $n$ . Vector de valores propios  $EVALM1, EVALM2$

```

1: {Variables para definir el número de frecuencias y repeticiones}
2: integer  $NumFrec = 100, NumRep = 1000$ 
3: {Variables para la media, varianza y el vector para almacenar las frecuencias}
4: real  $xm, xstd, VecRandom[NumFrec]$ 
5: CALL  $RNSET(0)$  {Fijamos la semilla (función IMSL)}
6: {Recorremos todos los segmentos para generar los números aleatorios}
7: for  $i = 1$  to  $NumSeg$  do
8:   if  $Genera == 1$  then
9:     CALL  $GeneraMediaVarianza(xm, xstd)$  {Genera media y varianza}
10:  else
11:     $xm \leftarrow$  Leer valor,  $xstd \leftarrow$  Leer valor
12:  end if
13:  {Mientras el nodos pertenezca al segmento se generan las frecuencias}
14:  while  $AsignaSeg[iNodo] == i$  do
15:    {Generamos el vector de frecuencias varias veces para esa nodo}
16:    for  $j = 1$  to  $NumRep$  do
17:      {Generamos el vector aleatorio con la distribución normal (función IMSL)}
18:      CALL  $RNNOR(NumFrec, VecRandom)$ 
19:      {Establecemos la desviación estándar (función IMSL)}
20:      CALL  $SSCAL(NumFrec, xstd, VecRandom, 1)$ 
21:      {Establecemos la media de la distribución (función IMSL)}
22:      CALL  $SADD(NumFrec, xm, VecRandom, 1)$ 
23:    end for
24:    {Multiplicamos el vector por el valor propio correspondiente al nodo}
25:    for  $j = 1$  to  $NumFrec$  do
26:       $MatrixRandom1[iNodo][j] \leftarrow VecRandom[j] * EVALM1[i]$ 
27:       $MatrixRandom2[iNodo][j] \leftarrow VecRandom[j] * EVALM2[i]$ 
28:    end for
29:     $iNodo \leftarrow iNodo + 1$  {Incrementamos el índice de los nodos}
30:  end while
31: end for

```

---

Como podemos observar en el algoritmo 4 se implementaron diversas funciones de la biblioteca IMSL, las cuales nos ayudaron principalmente a que la generación de los números aleatorios estuvieran basados en un *distribución Normal*. Las funciones que se utilizaron en este algoritmo se describen a continuación:



*Función RNSET(0)*

*Inicializa la semilla para generar los números aleatorios. Recibe como parámetro [42]:*

- *ISEED=0: Indica que se utiliza el reloj del sistema para generar la semilla (Dato de entrada).*

*Función RNNOR(NumFrec, VecRandom)*

*Esta función se encarga de generar números pseudoaleatorios con una distribución normal estándar. Los parámetros que recibe son [42]:*

- *NumFrec: Cantidad de números aleatorios que se desea generar (Dato de entrada).*
- *VecRandom: Vector de longitud Numfrec que contiene los números generados (Datos de Salida).*

*Función SSCAL(NumFrec, XSTD, VecRandom, 1)*

*Esta función se encarga de establecer la desviación estandar que se utiliza en la distribución. Los parámetros que recibe son [42]:*

- *NumFrec: Cantidad de números aleatorios que se van a generar.*
- *XSTD: Desviación que se utiliza en la distribución*
- *VecRandom: Vector de longitud Numfrec que contiene los números generados.*

*Función SADD(NumFrec, XM, VecRandom, 1)*

*Esta función se encarga de establecer la desviación estándar que se utiliza en la distribución. Los parámetros que recibe son [42]:*

- *NumFrec: Cantidad de números aleatorios que se van a generar.*
- *XM: Valor de la media que se va a utilizar*
- *VecRandom: Vector de longitud Numfrec que contiene los números generados.*

### 5.2.6 Proceso para calcular los modos aleatorios de vibración

Después de que se tuvieron generadas las frecuencias aleatorias para cada una de las partículas, el siguiente paso fue desarrollar un proceso que se encargara de calcular los movimientos aleatorios de vibración de cada partícula y posteriormente obtener la superposición de estos movimientos del sistema por unidad de tiempo.

Para calcular los movimientos aleatorios de vibración de cada partícula, fue necesario aplicar la ecuación de movimiento 5.2 para cada uno de los valores del vector de frecuencias generado y posteriormente obtener el promedio de los resultados (ecuación 5.4).

$$x_i = \frac{\sum_{k=1}^{nFrec} A_i \sin(\omega_k t)}{nFrec} \quad (5.4)$$

donde  $nFrec$  representa el tamaño del vector de frecuencias para cada partícula,  $A_i$  es el vector propio correspondiente a la partícula,  $\omega_k$  es la frecuencia aleatoria que se esta tratando y  $t$  representa el tiempo.

Por su parte, para calcular la superposición de estos movimientos fue necesario desarrollar un proceso mucho más complicado. Este cálculo consiste en obtener la sumatoria de los promedios de aplicar la ecuación de movimiento 5.2, con todos los vectores de las partículas que pertenecen a su mismo segmento (ecuación 5.5). Lo anterior se puede observar más claramente en la figura 5.4.

$$x_i = \sum_{l=1}^{nParSeg} \frac{\sum_{k=1}^{nFrec} A_i \sin(\omega_k t)}{nFrec} \quad (5.5)$$

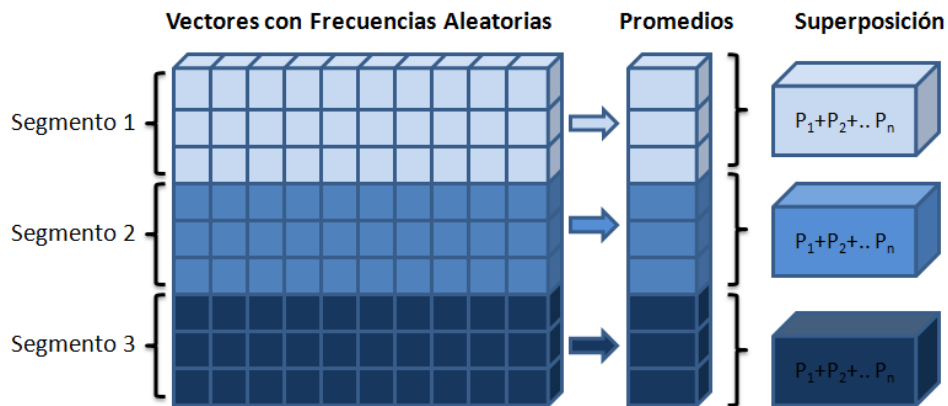


Figura 5.4: Cálculo de la superposición de frecuencias aleatorias

donde  $nParSeg$  representa el número de partículas en el segmento,  $NumFrec$  es el tamaño del vector de frecuencias,  $A_i$  es el vector propio de la partícula,  $\omega_k$  es la frecuencia aleatoria actual y  $t$  el tiempo.

**Algoritmo 5** Movimientos Aleatorios de Vibración

**Require:** Vector de valores propios  $EV_{ALM1}$ ,  $EV_{ALM2}$ . Matriz de vectores propios  $EVE_{CM1}$ ,  $EVE_{CM2}$ . Número de partículas  $n$ . Límite de Tiempo  $lim$ . Número de Segmentos  $NumSeg$

```

1: {Vectores para almacenar los resultados del movimiento y superposición}
2: real  $ResultM1[n]$ ,  $ResultM2[n]$ ,  $SumaM1[n]$ ,  $SumaM2[n]$ 
3: {Hacemos la división de segmentos y generamos las frecuencias aleatorias}
4: CALL  $AsigaSegmentos(AsignaSeg, n, NumSeg)$ 
5: CALL  $GeneraFrecuenciasMC(NumFrec, Matrix1, Matrix2, NumSeg, AsignaSeg, n, EV_{ALM1}, EV_{ALM2})$ 
6: for  $t = 0$  to  $lim$  do
7:   {Se recorren todas las partículas del sistema}
8:   for  $i = 1$  to  $n$  do
9:      $ResultM1[i] \leftarrow 0$ ,  $ResultM2[i] \leftarrow 0$ 
10:    {Se recorren las frecuencias generadas para esa partícula}
11:    for  $j = 1$  to  $NumFrec$  do
12:      {Obtenemos el movimiento de la partícula con la frecuencia aleatoria}
13:       $ResultM1[i] \leftarrow ResultM1[i] + EVE_{CM1}[i][j] * \sin(Matrix1[i][j] * t)$ 
14:       $ResultM2[i] \leftarrow ResultM2[i] + EVE_{CM2}[i][j] * \sin(Matrix2[i][j] * t)$ 
15:    end for
16:    {Obtenemos el promedio de los movimientos de la partícula}
17:     $ResultM1[i] \leftarrow ResultM1[i]/Numfrec$ ,  $ResultM2[i] \leftarrow ResultM2[i]/Numfrec$ 
18:    {Proceso para calcular la superposición por segmento}
19:     $SumaM1[i] \leftarrow 0$ ,  $SumaM2[i] \leftarrow 0$ ,  $iAsig \leftarrow 1$ 
20:    {Recorremos el vector de asignaciones.}
21:    while  $AsignaSeg[i] \leq AsignaSeg[iAsig] \wedge iAsig \leq n$  do
22:      {Si la partición es igual a la partición de la partícula actual}
23:      if  $AsignaSeg[i] \leq AsignaSeg[iAsig]$  then
24:         $SumaT1 \leftarrow 0$ ,  $SumaT2 \leftarrow 0$  {Variables temporales para promedios}
25:        {Se recorren las frecuencias generadas}
26:        for  $j = 1$  to  $NumFrec$  do
27:          {Obtenemos el movimiento de la partícula con la frecuencia aleatoria}
28:           $SumaT1 \leftarrow SumaT1 + EVE_{CM1}[i][j] * \sin(Matrix1[iAsig][j] * t)$ 
29:           $SumaT2 \leftarrow SumaT2 + EVE_{CM2}[i][j] * \sin(Matrix2[iAsig][j] * t)$ 
30:        end for
31:        {Obtenemos la suma de los promedios}
32:         $SumaM1[i] \leftarrow SumaM1[i] + (SumaT1 NumFrec)$ 
33:         $SumaM2[i] \leftarrow SumaM2[i] + (SumaT2 NumFrec)$ 
34:        {Se almacenan los resultados en archivos}
35:         $MovAleaVec1.txt \leftarrow ResultM1$ ,  $MovAleaVec2.txt \leftarrow ResultM2$ 
36:         $SumaAleaVec1.txt \leftarrow SumaM1$ ,  $SumaAleaVec2.txt \leftarrow SumaM2$ 
37:      end if
38:    end while
39:  end for
40: end for

```

Como podemos observar el algoritmo 5, se encarga de realizar los cálculos necesarios para obtener los movimientos aleatorios y la superposición de estos movimiento para cada ua de las partículas del sistema. Los resultados de los cálculos obtenidos al ejecutar este algoritmo son almacenados en archivos de texto, en los cuales los archivos *MovAleaVec1.txt* y *MovAleaVec2.txt* corresponden a los resultados de los movimientos aleatorios de cada partícula para el modelo de primeros y segundos vecinos respectivamente. Por otra parte los archivos *SumaAleaVec1.txt* y *SumaAleaVec2.txt* corresponden al cálculo de la superposición de los movimientos para el modelo de primeros y segundos vecinos.

Los archivos de resultados que genera este algoritmo se utilizan para generar las gráficas y animaciones del subsistema de visualización que se describe en las siguientes secciones.

### 5.3 Desarrollo del subsistema para la interfaz gráfica

Como se observa en el diagrama general del sistema (Fig 4.4), se necesitó desarrollar un subsistema independiente al de cómputo de datos, el cual permitiera hacer una visualización gráfica en forma de imágenes y animaciones de los resultados obtenidos por el subsistema anterior.

Este subsistema utiliza los archivos generados por el subsistema de *cómputo de datos* para poder desplegar en pantalla los resultados gráficamente, de tal forma que el usuario pueda entender e interpretar el comportamiento del sistema de una manera más sencilla.

El *subsistema de visualización* consiste en una aplicación con una interfaz gráfica dividida en tres segmentos, que están desarrollados en el lenguaje de programación C++ con ayuda de QT. Los segmentos en los que se divide la interfaz gráfica representan cada uno de los módulo que conforman el subsistema de visualización descritos en el Capítulo 4.

El desarrollo de este subsistema se llevó a cabo en un esquema orientado a objetos, por lo que los módulos que lo conforman están compuestos por clases, las cuales se encargan de generar las imágenes y animaciones que se despliegan en pantalla.

Para describir mejor este subsistema, se desarrolló un diagrama general de las clases que conforman cada uno de los módulos(Fig. 5.5).

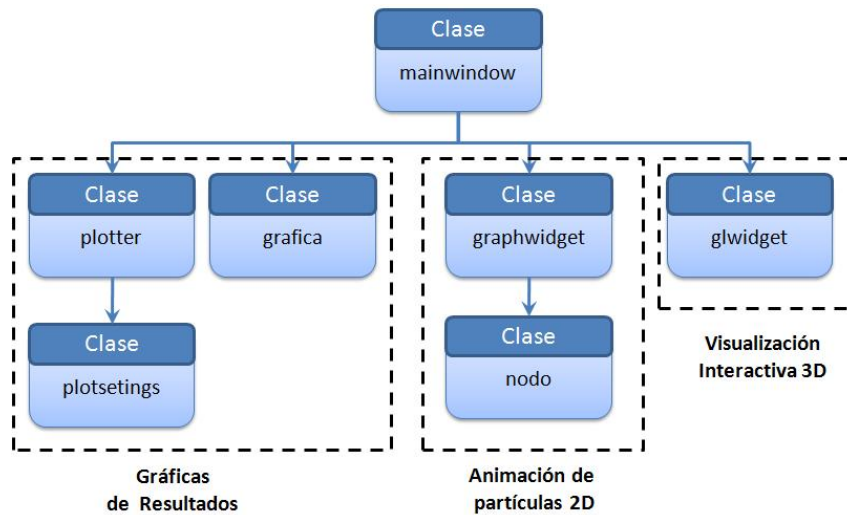


Figura 5.5: Diagrama de clases del sistema de visualización

La clase principal denominada *mainwindow* se encarga de generar la interfaz principal de nuestro sistema y la divide en segmentos para la visualización de cada uno de los módulos. Además, se encarga de establecer los diferentes eventos que se pueden ejecutar en la simulación.

- *Abrir archivo:* Permite seleccionar un archivo de resultados y generar la simulación con sus datos.
- *Salir:* Finaliza la aplicación.
- *Ejecutar:* Esta opción se habilita cuando se abre un archivo y nos permite comenzar la simulación.
- *Pausar:* Cuando una animación está en proceso, este evento permite detenerla temporalmente para poder continuar posteriormente con su ejecución.
- *Siguiente:* Esta opción nos permite ver el comportamiento del sistema paso a paso, siempre y cuando la simulación se encuentre en pausa.
- *Detener:* Detiene la simulación.
- *Reinicia:* Permite volver a comenzar la simulación del archivo que se abrió en un inicio.
- *Graficar:* Genera una gráfica con los datos del archivo que se tiene abierto.

Cuando se abre un archivo, esta clase se encarga de obtener los datos de la cabecera del archivo, con los cuales verifica que éste sea un archivo válido para la aplicación. Posteriormente, obtiene la información adicional que proporciona la cabecera del archivo

como el número de partículas, la longitud del *riser*, el número de segmentos, el límite de tiempo, la masa y la constante de elasticidad para las partículas y resortes.

Una vez que se pudo abrir el archivo de resultados y que se obtuvieron los datos de la cabecera, la clase *mainwindow* se encarga de dividir la interfaz principal en tres regiones y genera instancias de las clases *grafica*, *graphwidget* y *glwidget* para inicializar la animación en cada uno de los segmentos.

Cuando se selecciona el evento para ejecutar la simulación, se crea un *timer* en el cual cada determinado intervalo de tiempo se lee una línea de datos del archivo. Estos datos son enviados a las instancias de las clases que se crearon para que puedan ser desplegados en cada uno de los módulos de visualización. El proceso para los diversos módulos de visualización y las clases que se implementaron se describen en la siguientes secciones.

### 5.3.1 Módulo de gráficas de resultados

Este módulo de visualización se encarga de ir graficando los resultados de cada una de las partículas del sistema que son leídos del archivo por los clase *mainwindow*.

Para el desarrollo de este módulo se implementó una clase denominada *grafica*, la cual cuenta con tres procesos principales (Fig. 5.6):

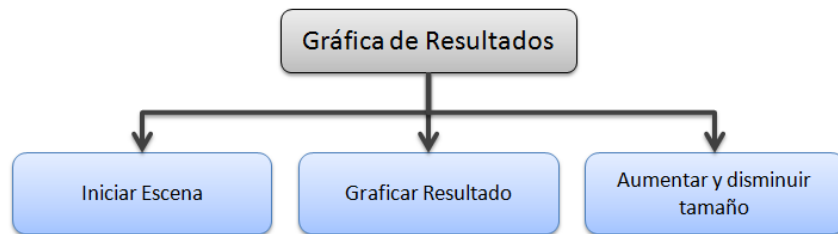


Figura 5.6: Procesos del módulo de gráficas de resultados

1. **Inicializar Escena:** Este proceso se ejecuta en el momento de abrir un archivo de resultados y generar una instancia de la clase *grafica*. En este proceso la clase se encarga de construir toda la escena donde se van generar las gráficas. A partir del número de nodos, la clase asigna un espacio para cada una de las partículas del sistema en donde se van poder graficar los resultados leídos del archivo.
2. **Graficar resultado:** Cada que se cumple el intervalo de tiempo del *timer* en la clase *mainwindow*, se realiza una lectura al archivo de datos. Los datos leídos son almacenados en una cadena y enviados a una función de la clase *grafica*. Esta clase se encarga de tomar la cadena de resultados y dividirla de manera que a cada partícula corresponda un resultado. Cada uno de estos resultados representa un punto en la gráfica que se genera para cada partícula, por lo que después de hacer la división, se manda pintar un punto en el espacio asignado para cada gráfica (Fig. 5.7).

3. **Aumentar y disminuir tamaño:** Debido a la cantidad de gráficas que se tienen que generar y al límite del espacio que se tiene asignado para cada ellas, la clase *grafica* cuenta con rutinas que permiten aumentar o disminuir el tamaño de la escena en general. Para esto se implementó una función que captura eventos del teclado, de tal forma que cuando se presiona la *tecla +* aumenta de tamaño los componentes de la escena. Por el contrario cuando se selecciona la *tecla -* los elementos disminuyen su tamaño.

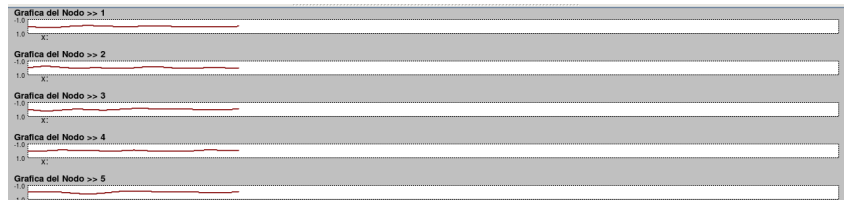


Figura 5.7: Simulación del módulo de gráficas

Además el módulo de gráficas anterior, la aplicación cuenta con un módulo adicional que nos permite generar las gráficas de todas las partículas de forma inmediata. Para desarrollar este proceso se necesitó implementar dos clases adicionales *plotter* y *plotsettings*. Esta opción se ejecuta cuando se selecciona el *evento graficar* de la interfaz principal. El proceso se encarga de realizar una lectura al archivo de resultados independiente a la lectura de la simulación para graficar todos los resultados de las partículas (Fig 5.8).

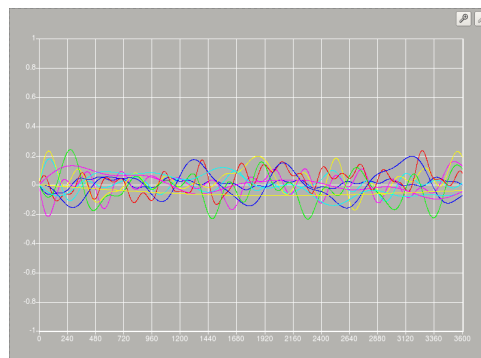


Figura 5.8: Interfaz para generar las gráficas de las partículas

### 5.3.2 Módulo de animación de partículas 2D

Además de las gráficas, se requeriría un módulo que nos permitiera observar el comportamiento independiente de cada una de las partículas del sistema. Para cumplir con este requerimiento, se desarrolló un módulo que genera una animación en 2D y

nos permite analizar el movimiento independiente de las partículas basándonos en el archivo de resultados leído.

Este módulo consta de dos clases principales *graphwidget* y *nodo*. Al igual que en el módulo anterior, este también cuenta con tres procesos principales (Fig. 5.9), los cuales se describen a continuación:

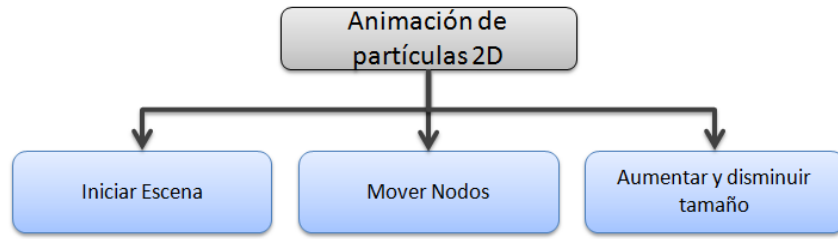


Figura 5.9: Procesos del módulo de animación de partículas

1. **Inicializar Escena:** Este proceso se ejecuta cuando se abre un archivo de resultados y se genera una instancia de la clase *graphwidget*. El proceso se encarga de construir la escena y crear los elementos necesario para realizar la animación de las partículas. Tomando como base el número de partículas del sistema, la clase crea un arreglo de nodos. Cada uno de estos nodos representan una instancia de la clase *nodo*, lo que nos va a permitir darles movimiento de manera independiente.
2. **Mover Nodos:** Al igual que en módulo anterior, cuando se cumple el intervalo de tiempo del *timer*, se envían los datos leídos a una función de la clase *graphwidget*, la cual se encarga de separar los resultados por partículas. Con estos datos se actualiza la posición de los nodos en la escena. Esto se lleva a cabo en intervalos de tiempo definidos por el *timer*, lo que da una perspectiva de movimiento en las partículas (Fig. 5.10).
3. **Aumentar y disminuir tamaño:** En este módulo también fue necesario implementar una rutina que permita interactuar con el tamaño de los elemento de la escena. Para esto se implemetó una función que captura los eventos del teclado y que nos permite aumentar (*tecla +*) y disminuir (*tecla -*) el tamaño de los componente de la escena.



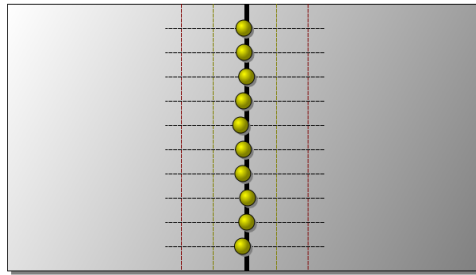


Figura 5.10: Animación del partículas

### 5.3.3 Módulo de visualización interactiva 3D

Este módulo nos va permitir analizar el funcionamiento general del sistema. Para llevar a cabo desarrollo de este módulo se implementaron rutinas del lenguaje *OpenGL*. Estas rutinas nos va permitir generar un modelo en 3D de nuestro sistema y actualizarlo con los datos del archivo de resultados.

El módulo esta conformado por una clase denominada *glwidget*, encargada de construir el modelo del sistema y generar la animación. Los procesos que conforman este módulo se muestran en la figura 5.11:

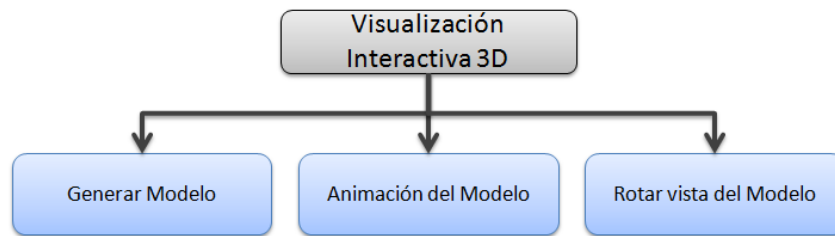


Figura 5.11: Procesos del módulo de visualización interactiva

1. **Generar modelo:** Este proceso inicia cuando la clase *mainwindow* crea una instancia de la clase *glwidget*. El proceso se encarga de generar la escena y construir un modelo que representa nuestro sistema. El modelo que se construye esta dividido en partes, las cuales van a representar las partículas del sistema.
2. **Animación del Modelo:** Al igual que en los módulos anteriores, la animación de nuestro modelo se realiza en cada intervalo de tiempo del *timer*. Los resultados obtenidos del archivo para cada partícula, representan la posición de los nodos por los que se conforma el sistema. En cada intervalo de tiempo estos nodos van actualizando sus posiciones, lo que produce el movimiento de la estructura (Fig. 5.12).
3. **Rotar Vista del Modelo:** Este proceso nos permite ver el comportamiento del sistema desde distintos ángulos. Para llevar a cabo esto se implementaron fun-

ciones que nos permiten mover el modelo del sistema en sus distintos ejes ( $x$ ,  $y$  y  $z$ ). Para llevar a cabo esta rotación, se implementó una función que detecta los eventos del mouse y conforme a su movimiento hace rotar el modelo. Además, se implementaron tres barras de rotación que también nos permiten realizar este tipo de movimientos en los tres ejes.

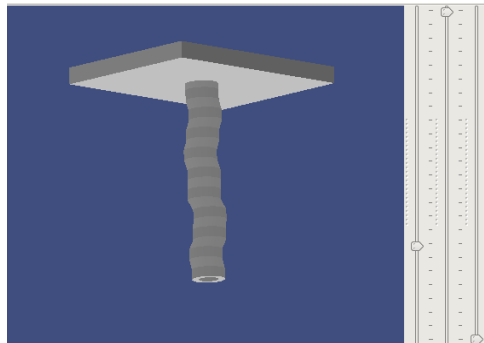


Figura 5.12: Aimación del sistema en 3D

## 5.4 Resumen

En este capítulo se describió el proceso de como se llevó a cabo el desarrollo del simulador. En la primer sección se describió la arquitectura general de nuestro sistema. Se detallaron los lenguajes de programación que se utilizaron durante el desarrollo del sistema y se describieron las librerías adicionales que se utilizaron.

En la segunda sección se describieron los módulos y los diversos algoritmos que se implementaron para desarrollar el subsistema de *cómputo de datos*.

Finalmente en la última sección, se describió el proceso y las clases que se implementaron para llevar a cabo el desarrollo del subsistema del Visualización.

# Capítulo 6

## Validación y pruebas del simulador

Todos somos muy ignorantes. Lo que ocurre es que no todos ignoramos las mismas cosas.

---

Albert Einstein

La validación y pruebas de un proceso de simulación por computadora, nos permite comprobar los resultados obtenidos y analizar que tan exactas son las predicciones del comportamiento del sistema.

El objetivo de este capítulo es presentar una descripción general de las pruebas que se realizaron del sistema y comprobar su correcta funcionalidad.

En esta capítulo se describe la forma de operar los dos subsistemas que conforman el simulador y de esta forma obtener los resultados correctos. Además, se presenta el diseño de las pruebas que se llevaron a cabo y los resultados obtenidos.

### 6.1 Descripción general del uso del simulador

Como se describió en los Capítulos 3 y 4 (diseño y desarrollo), nuestro sistema está formado por dos subsistemas. El primero se encarga realizar el cálculo de las vibraciones y almacena los resultados en archivos de texto. Por su parte, el segundo subsistema se encarga de visualizar los resultados generados por medio de gráficas y animaciones.

Estas dos aplicaciones son independientes ya que se pueden estar ejecutando en estaciones de trabajo distintas e inclusive pueden estar realizando pruebas totalmente diferentes. Sin embargo, la aplicación encargada de la visualización depende de los resultados que genera la aplicación de cómputo de datos. Por lo cual, para llevar a cabo las pruebas se requiere ejecutar primero la aplicación de *cómputo de datos* y posteriormente

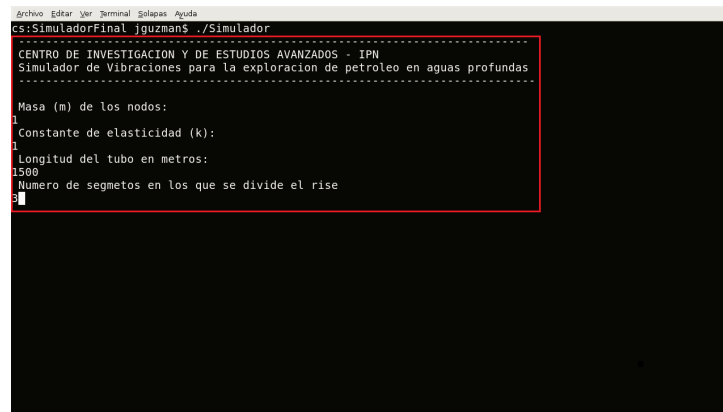
la aplicación de *visualización*. La forma de operar estos dos subsistema se describe en las siguientes secciones.

### 6.1.1 Aplicación de cómputo de datos

Para comenzar a utilizar esta aplicación es necesario contar con el archivo binario ejecutable correspondiente a la aplicación (*SistComputo*). Este programa puede ejecutarse desde línea de comandos mediante la siguiente instrucción:

```
./SistComputo
```

Una vez que se haya ejecutado este comando, en la pantalla de la consola aparecerá un mensaje (Fig. 6.1) el cual nos indica que la aplicación inició correctamente. Posteriormente, la aplicación solicitará los datos de las condiciones iniciales para la simulación. Los datos que se solicitan son el número de nodos, la masa de los nodos ( $n$ ), la constante de elasticidad ( $k$ ), la longitud ( $Long$ ) y el número de segmentos ( $NumSeg$ ).



```
cs:SimuladorFinal jguzman$ ./Simulador
-----
CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS - IPN
Simulador de Vibraciones para la exploracion de petroleo en aguas profundas
-----
Masa (m) de los nodos:
1
Constante de elasticidad (k):
1
Longitud del tubo en metros:
1500
Numero de segmetos en los que se divide el rise
3
```

Figura 6.1: Pantalla de inicio de la aplicación de cómputo de datos

Si el número de segmentos que se proporciona es mayor al número de partículas del sistema, la aplicación desplegará un mensaje de error como el que se muestra en la figura 6.2 y volverá a pedir el número de segmentos hasta que se proporcione un número correcto.

```
Archivo Editar Ver Terminal Salidas Ayuda
cs:SimuladorFinal jguzman$ ./Simulador
-----
CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS - IPN
Simulador de Vibraciones para la exploracion de petroleo en aguas profundas
-----
Masa (m) de los nodos:
1
Constante de elasticidad (k):
1
Longitud del tubo en metros:
1500
Numero de segmentos en los que se divide el rise
10
[**ERRR**]El numero de segmento no puede ser mayor al Numero de nodos
Numero de segmentos en los que se divide el rise
3
3
```

Figura 6.2: Pantalla de error en el número de segmentos

Después de proporcionar los datos iniciales, se genera un directorio en el cual se almacenarán los archivos de resultados. El nombre de este directorio esta conformado por la fecha y la hora en que se ejecutó la aplicación (Fig. 6.3).

```
Archivo Editar Ver Terminal Salidas Ayuda
cs:SimuladorFinal jguzman$ 1
-bash: 1: command not found
cs:SimuladorFinal jguzman$ clear
cs:SimuladorFinal jguzman$ ./Simulador
-----
CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS - IPN
Simulador de Vibraciones para la exploracion de petroleo en aguas profundas
-----
Masa (m) de los nodos:
1
Constante de elasticidad (k):
1
Longitud del tubo en metros:
1500
Numero de segmentos en los que se divide el rise
10
[**ERRR**]El numero de segmento no puede ser mayor al Numero de nodos
Numero de segmentos en los que se divide el rise
3
-----
Generando la cabecera del archivo-> C\INVESTAV-IPN 3 1500 3 3600 1.00000 1.00000
-----
Creando directorio-> 15-11-2009 01-18-21
-----
2.00000
1.00000
```

Figura 6.3: Pantalla de la creación del nombre del directorio

Posteriormente la aplicación se encarga de generar las matrices de movimiento para los dos modelos y obtiene los valores propios y el vector propio correspondientes a estas matrices (Fig. 6.4).

```

Archivo Editar Ver Terminal Solapas Ayuda
-1.00000
4.00000

**** Eigenvalores (EVAL) ****
      1      2      3
( 3.414, 0.000) ( 2.000, 0.000) ( 0.586, 0.000)

**** Eigenvector (EVEC) ****
      1      2      3
1 (-0.5000, 0.0000) ( 0.7071, 0.0000) ( 0.5000, 0.0000)
2 ( 0.7071, 0.0000) ( 0.0000, 0.0000) ( 0.7071, 0.0000)
3 (-0.5000, 0.0000) (-0.7071, 0.0000) ( 0.5000, 0.0000)

**** Eigenvalores (EVAL) ****
      1      2      3
( 5.000, 0.000) ( 5.000, 0.000) ( 2.000, 0.000)

**** Eigenvector (EVEC) ****
      1      2      3
1 ( 0.8165, 0.0000) (-0.4201, 0.0000) ( 0.5774, 0.0000)
2 (-0.4082, 0.0000) (-0.3962, 0.0000) ( 0.5774, 0.0000)
3 (-0.4082, 0.0000) ( 0.8164, 0.0000) ( 0.5774, 0.0000)

-----
Calculando Modos Normales de vibracion
-----

```

Figura 6.4: Pantalla de matrices de movimiento con sus valores propios y vector propio

Con estos datos el sistema comienza a realizar el cálculo de los modos normales de vibración de las partículas y la superposición de los movimientos (Fig. 6.5). Los resultados obtenidos son almacenados en cuatro archivos:

1. *ModNormVec1.txt*: Contiene los resultados de los modos normales de vibración del modelo de primeros vecinos.
2. *ModNormVec2.txt*: Contiene los resultados de los modos normales de vibración del modelo de segundos vecinos.
3. *SumaNormVec1.txt*: Contiene los resultados de la superposición de los modos normales de vibración del modelo de primeros vecinos
4. *SumaNormVec2.txt*: Contiene los resultados de la superposición de los modos normales de vibración del modelo de segundos vecinos

```

Archivo Editar Ver Terminal Solapas Ayuda

**** Eigenvalores (EVAL) ****
      1      2      3
( 3.414, 0.000) ( 2.000, 0.000) ( 0.586, 0.000)

**** Eigenvector (EVEC) ****
      1      2      3
1 (-0.5000, 0.0000) ( 0.7071, 0.0000) ( 0.5000, 0.0000)
2 ( 0.7071, 0.0000) ( 0.0000, 0.0000) ( 0.7071, 0.0000)
3 (-0.5000, 0.0000) (-0.7071, 0.0000) ( 0.5000, 0.0000)

**** Eigenvalores (EVAL) ****
      1      2      3
( 5.000, 0.000) ( 5.000, 0.000) ( 2.000, 0.000)

**** Eigenvector (EVEC) ****
      1      2      3
1 ( 0.8165, 0.0000) (-0.4201, 0.0000) ( 0.5774, 0.0000)
2 (-0.4082, 0.0000) (-0.3962, 0.0000) ( 0.5774, 0.0000)
3 (-0.4082, 0.0000) ( 0.8164, 0.0000) ( 0.5774, 0.0000)

-----
Calculando Modos Normales de vibracion
-----
Calculando Modos Aleatorios de vibracion
-----

```

Figura 6.5: Pantalla del cálculo de modos normales de vibración

Una vez obtenidos los resultados de los modos normales de vibración, el sistema realiza la división de segmento del sistema y genera la matriz de frecuencias aleatorias. Para esto el sistema pregunta si los valores de media  $\mu$  y varianza  $\sigma$  para cada uno de los segmentos se van a generar o se van a proporcionar por el usuario (Fig. 6.6).

```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
4 ( 0.2972, 0.0000) ( 0.3807, 0.0000)
5 ( 0.1121, 0.0000) ( 0.4118, 0.0000)
6 (-0.1121, 0.0000) ( 0.4118, 0.0000)
7 (-0.2972, 0.0000) ( 0.3807, 0.0000)
8 (-0.4893, 0.0000) ( 0.3252, 0.0000)
9 (-0.3786, 0.0000) ( 0.2364, 0.0000)
10 (-0.2972, 0.0000) ( 0.1544, 0.0000)

-----
Calculando Modos Normales de vibracion
-----
Calculando Modos Aleatorios de vibracion
-----
Como se van proporcionar los valores de media y varianza:
1- Aleatorias
2- Manual
Selecciona una opcion:
1
Valor de la Media 1 : 8.436054E-02
Valor de la Varianza 1 : 0.568414

Valor de la Media 2 : -0.890116
Valor de la Varianza 2 : 0.372307

Valor de la Media 3 : 0.893889
Valor de la Varianza 3 : 0.165109

cs:SimuladorFinal jguzman$

```

Figura 6.6: Pantalla de los datos de media y varianza de los segmentos

Finalmente la aplicación realiza el cálculo de los modos aleatorios de vibración de las partículas y la superposición de estos movimientos. Los resultados obtenidos se almacenan en otros cuatro archivos:

1. *MovAleaVec1.txt*: Contiene los resultados de los modos aleatorios de vibración del modelo de primeros vecinos.
2. *ModAleaVec2.txt*: Contiene los resultados de los modos aleatorios de vibración del modelo de segundos vecinos.
3. *SumaAleaVec1.txt*: Contiene los resultados de la superposición de los modos aleatorios de vibración del modelos de primeros vecinos
4. *SumaAleaVec2.txt*: Contiene los resultados de la superposición de los modos aleatorios de vibración del modelos de segundos vecinos

Para evitar que el usuario este proporcionando los datos necesarios para ejecutar la aplicación, se puede crear un archivo (*DatosEntrada.txt*) que contenga todos estos datos de entrada y ejecutar la aplicación de la siguiente manera:

```
./SistComputo < DatosEntrada.txt
```

## 6.1.2 Aplicación de visualización

Para comenzar a utilizar la aplicación de visualización, se requiere del archivo ejecutable *SistVisualizacion*. Este programa puede ejecutarse desde la interfaz gráfica haciendo

doble click sobre el icono del programa o bien desde línea de comandos mediante la instrucción:

*./SistVisualizacion*

Al inicio se desplegará en pantalla una interfaz que contienen un menú principal y una barra de herramientas con diversas opciones para realizar la simulación de un archivo de resultados (Fig. 6.7).

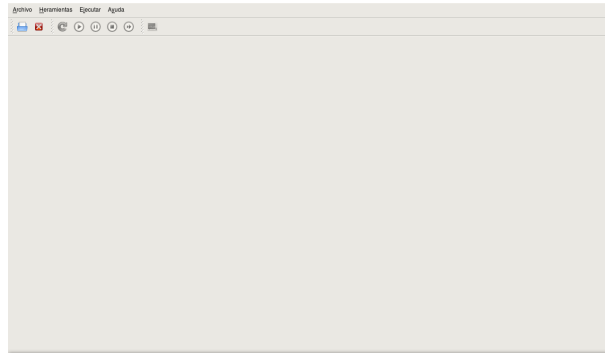


Figura 6.7: Pantalla de inicio del subsistema de visualización

El menú principal está compuesto por cuatro submenús (Fig. 6.8) que contienen las siguientes opciones:

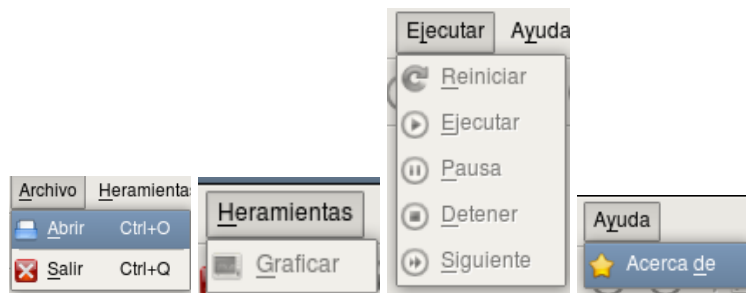


Figura 6.8: Menús del subsistema de visualización

### 1. *Archivo*

- (a) *Abrir*: Abre un archivo para visualizar sus resultados.
- (b) *Salir*: Cierra la aplicación.

### 2. *Herramientas*

- (a) *Graficar*: Grafica los datos del archivo en una ventana independiente a la interfaz principal.



### 3. Ejecutar

- (a) *Reiniciar*: Reinicia la simulación.
- (b) *Ejecutar*: Comienza la simulación.
- (c) *Pausa*: Detiene momentáneamente la simulación
- (d) *Detener*: Detiene totalmente la simulación.
- (e) *Siguiente*: Genera paso a paso la simulación

### 4. Ayuda

- (a) *Acerca de*: Contiene información sobre la versión del simulador.

Por su parte la barra de herramientas se compone de las siguientes secciones (Fig. 6.9):



Figura 6.9: Barra de herramientas del subsistema de visualización

1. *Sección Archivo*: Contiene las mismas opciones que el menú archivo.
2. *Sección Ejecutar*: Contiene las opciones del menú ejecutar
3. *Sección Herramientas*: Contiene las opciones del menú herramientas.

Para comenzar con la simulación, es necesario seleccionar la opción *Abrir archivo*. Al seleccionar esta opción, se despliega una ventana (Fig. 6.10) en la que se puede elegir alguno de los archivos generados por el subsistema de cómputo de datos.

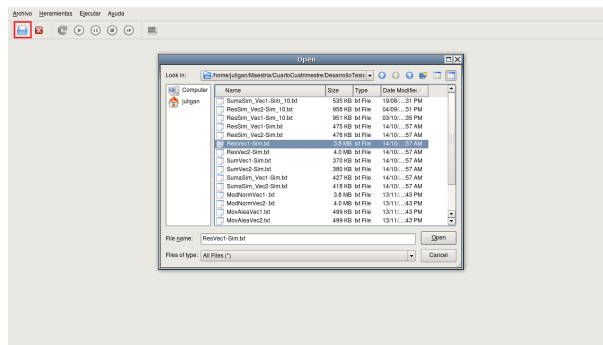


Figura 6.10: Pantalla para abrir archivo.

Al abrir un archivo de datos, la interfaz se divide en tres segmentos en los que se generan las escenas correspondientes para hacer la visualización de los datos (Fig. 6.11) y se habilitan las diversas opciones del menú para manejar la simulación.

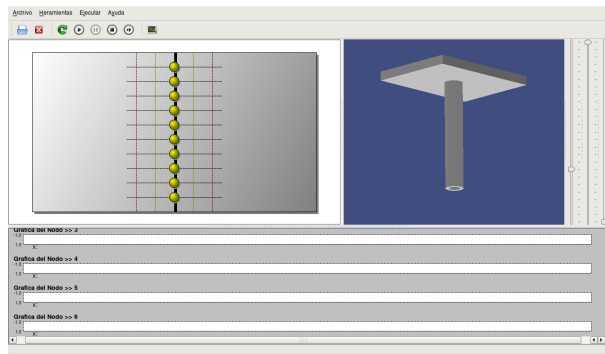


Figura 6.11: Pantalla de inicio de simulación

Posteriormente se puede seleccionar la opción *ejecutar* para comenzar a generar las gráficas y animaciones con los datos del archivo (Fig. 6.12). Si se desea ejecutar la simulación paso a paso, es necesario seleccionar la opción *pausar* para detener momentáneamente la simulación y después seleccionar la opción *siguiente*.

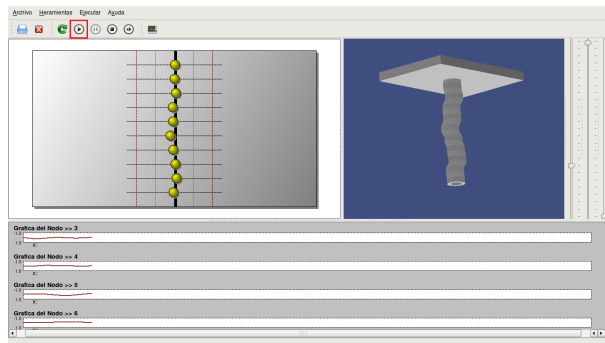


Figura 6.12: Pantalla del proceso de simulación.

Para obtener la gráfica completa de todos los datos del archivos, se debe seleccionar la opción *graficar*, la cual despliega una ventana con la gráfica generada (Fig. 6.13). Este proceso se puede ejecutar en cualquier momento sin necesidad de detener la simulación principal.

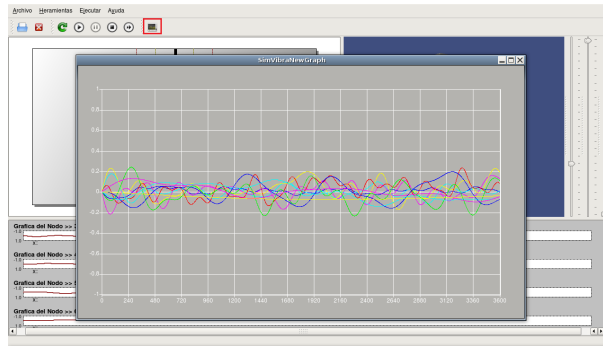


Figura 6.13: Pantalla del sistema que muestra la gráfica de los datos.

Finalmente, para salir de la aplicación se debe seleccionar la opción *salir* o bien seleccionar el botón *cerrar ventana* de la interfaz.

## 6.2 Diseño de la pruebas

Para validar el funcionamiento de nuestro simulador, se diseñaron diversas pruebas con diferentes condiciones. Durante estas pruebas, se verificó que se ejecuten cada uno de los procesos que conforman el sistema y que se generen resultados de acuerdo a las condiciones establecidas.

Los datos principales que fueron considerados para el diseño de las pruebas fueron: la longitud del *riser*, el número de partículas y el número de segmentos en los que se divide el sistema.

Para determinar la longitud del sistema, se tomó como referencia la profundidad máxima que pudiera tener un *riser* en aguas profundas, que como se mencionó en el Capítulo 1 es de 1500 mts (Fig. 2.4). Tomando como referencia esta longitud y debido a que se quería realizar un análisis de las vibraciones en distancias relativamente cortas, se estableció tener una cantidad de 256 partículas, las cuales distribuidas entre los 1500 metros de longitud se tendrá una partícula cada 5.85 mts (Fig. 6.14).

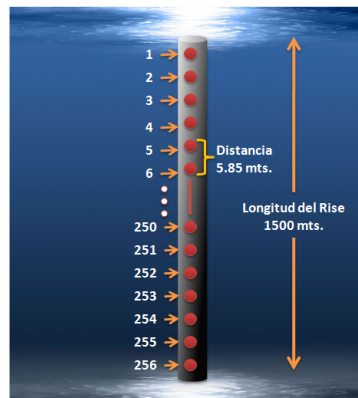


Figura 6.14: Distribución de partículas en el *riser* para las pruebas

Con estas condiciones iniciales se diseñaron tres pruebas para analizar el comportamiento del sistema en diversas condiciones. Para esto el sistema se dividió en segmentos de la siguiente manera:

- Sistema con 32 segmentos y 8 partículas en cada segmento.
- Sistema con 16 segmentos y 16 partículas en cada segmento.
- Sistema con 8 segmentos y 32 partículas en cada segmento.

Para estas pruebas la masa de las partículas y la constante de elasticidad se definieron con un valor de 1. Por su parte los valores de media y varianza para generar las frecuencias de cada segmento, se calcularon de forma aleatoria con la finalidad de generar distintas condiciones de corriente en cada uno de los segmentos.

En las pruebas realizadas, el cálculo de los datos para cada partícula se hizo en un ciclo de 1 a 3600 seg., en donde cada iteración representa una unidad de tiempo. Esto con la finalidad de analizar si el comportamiento de las partículas se repite después de un intervalo de tiempo determinado.

Con estos valores se ejecutó el sistema de cómputo de datos para generar los archivos de resultados correspondientes y posteriormente desplegar estos resultados en el sistema de visualización.

## 6.3 Resultados de las pruebas

En esta sección se mostrarán los resultados que se obtuvieron al realizar las pruebas que se diseñaron en la sección anterior. Cada una de estas pruebas se realizaron por separado y generaron distintos archivos de resultados.

Para agilizar la ejecución de las pruebas, se crearon archivos con los datos de entrada para evitar estar proporcionando los datos a través del teclado.

### 6.3.1 Prueba con 8 segmentos

Para comenzar a realizar los cálculos para esta prueba, se creó un archivo que contiene los datos de entrada para el sistema de cómputo de datos. Los datos para la ejecución de esta prueba se muestran en la tabla 6.1.

Datos de entrada	
Número de partículas $n$	256
Masa de las partículas $m$	1
Constante de elasticidad $k$	1
Longitud del riser $Long$	1500
Número de Segmentos $NumSeg$	8

Tabla 6.1: Datos de entrada para la ejecución de la prueba de 8 segmentos

Posteriormente se ejecutó el sistema con los datos de entrada definidos y se generaron los siguientes archivos (Fig. 6.15).

```

cs:16-11-2009 00-23-41 jguzman$ ls -ltr
total 12969704
-rw-r--r-- 1 jguzman staff 9366945 Nov 16 00:30 SumaNormVec2.txt
-rw-r--r-- 1 jguzman staff 9140567 Nov 16 00:30 SumaNormVec1.txt
-rw-r--r-- 1 jguzman staff 326852072 Nov 16 00:30 ModNormVec2.txt
-rw-r--r-- 1 jguzman staff 3303677064 Nov 16 00:30 ModNormVec1.txt
-rw-r--r-- 1 jguzman staff 11974850 Nov 16 00:33 SumaAleaVec2.txt
-rw-r--r-- 1 jguzman staff 11976274 Nov 16 00:33 SumaAleaVec1.txt
-rw-r--r-- 1 jguzman staff 12908936 Nov 16 00:33 MovAleaVec2.txt
-rw-r--r-- 1 jguzman staff 12912264 Nov 16 00:33 MovAleaVec1.txt
cs:16-11-2009 00-23-41 jguzman$

```

Figura 6.15: Archivos que se generaron de la prueba de 8 segmentos

Una vez que se finalizó el cómputo de los datos, los archivos que contenían los resultados de las frecuencias aleatorias se sometieron al sistema de visualización. El primer archivo que se sometió al sistema de visualización, es el que contiene los resultados de los modos aleatorios de vibración para el modelo de primeros vecinos *MovAleaVec1.txt*. La simulación y la gráfica que se generó en el sistema se muestra en la figura 6.16.

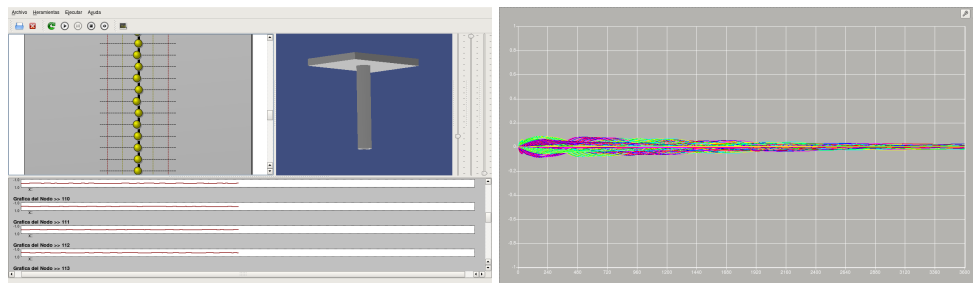


Figura 6.16: Resultado de la visualización de los modos aleatorios para primeros vecinos

El siguiente archivo que se ejecutó en el sistema de visualización, es el que contiene los modos aleatorios de vibración para el modelo de segundos vecinos *MovAleaVec2.txt*. Los resultados obtenidos en el sistema de visualización se muestran en la figura 6.17.

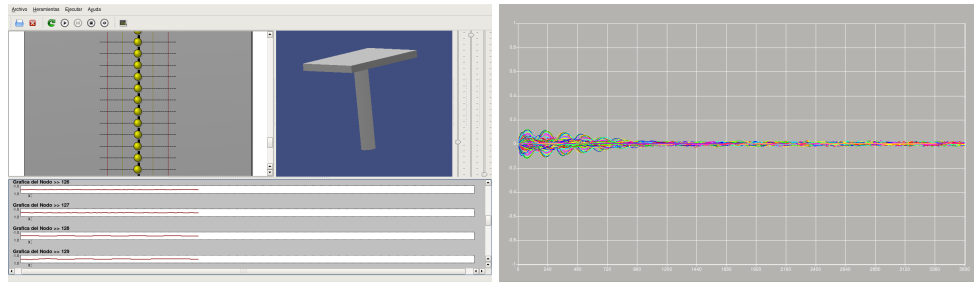


Figura 6.17: Resultado de la visualización de los modos aleatorios para segundos vecinos

En la tabla 6.2 se muestra una comparación de los resultados obtenidos de algunas de las partículas en los tiempos  $t_{100} \rightarrow t_{103}$  para los modelos de primeros y segundos vecinos.

Modos aleatorios de vibración. <i>Riser</i> con 8 segmentos								
$n$	$t_{100}$ (1er Vecinos, 2do Vecinos)		$t_{101}$ (1er Vecinos, 2do Vecinos)		$t_{102}$ (1er Vecinos, 2do Vecinos)		$t_{103}$ (1er Vecinos, 2do Vecinos)	
10	1.907847E-05	-1.992678E-04	-3.929516E-05	-1.699451E-04	-9.593633E-05	-1.479500E-04	-1.504193E-04	-1.325698E-04
20	-1.813446E-03	-1.124881E-03	-1.809459E-03	-1.344840E-03	-1.801484E-03	-1.517096E-03	-1.789948E-03	-1.638451E-03
30	-1.246394E-04	-3.265392E-03	-6.290861E-05	-3.695666E-03	9.108428E-07	-4.033606E-03	6.807496E-05	-4.272896E-03
40	-3.350988E-02	1.990384E-02	-3.241201E-02	1.880272E-02	-3.122393E-02	1.757011E-02	-2.994924E-02	1.621585E-02
50	-4.408304E-02	-9.005854E-03	-4.307781E-02	-8.612059E-03	-4.195805E-02	-8.158773E-03	-4.072702E-02	-7.649585E-03
60	-5.551505E-02	-5.186288E-02	-5.500680E-02	-5.079360E-02	-5.436418E-02	-4.938839E-02	-5.358897E-02	-4.765817E-02
70	-7.175901E-03	1.292170E-02	-7.595513E-03	1.436066E-02	-7.868181E-03	1.554352E-02	-7.990567E-03	1.637010E-02
80	-7.305894E-04	-5.214892E-03	-5.016610E-04	-5.079770E-03	-3.422102E-04	-4.961209E-03	-2.547934E-04	-4.832680E-03
90	-2.965253E-03	-4.334845E-03	-2.882498E-03	-3.916809E-03	-2.827634E-03	-3.518403E-03	-2.796132E-03	-3.159899E-03
100	2.615707E-03	2.836561E-04	2.443388E-03	3.041067E-04	2.267575E-03	3.126351E-04	2.088836E-03	3.094971E-04

Tabla 6.2: Resultados de los modos aleatorios de vibración. *Riser* con 8 segmentos.

Después de visualizar los datos de los modos aleatorios de vibración de los modelos de primeros y segundos vecinos, se procedió a desplegar los resultados de los archivos que contienen la superposición de los movimientos de ambos modelos.

La visualización que se generó para el archivo de superposición del modelo de primeros vecinos (*SumaAleaVec1.txt*) se muestra en la figura 6.18.

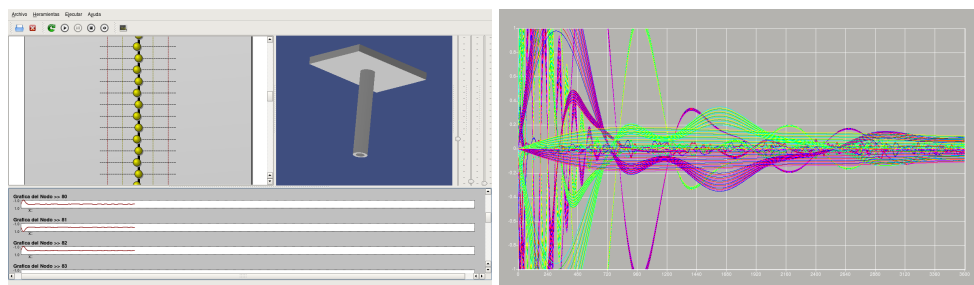


Figura 6.18: Resultado de la visualización de la superposición para primeros vecinos

Finalmente se visualizó el archivo que contiene los resultados de la superposición del modelo de segundos vecinos *SumaAleaVec2.txt* (Fig. 6.19).

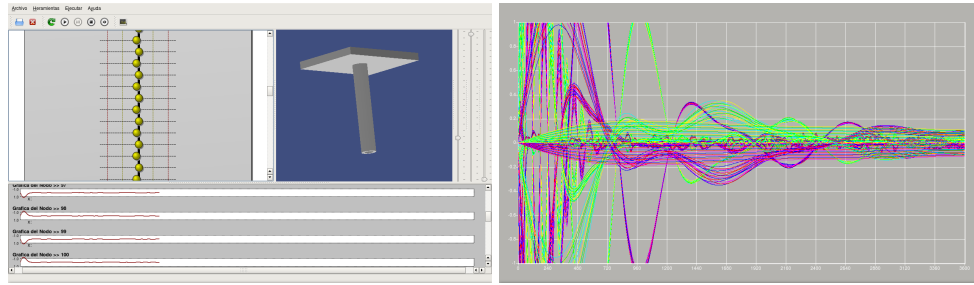


Figura 6.19: Resultado de la visualización de la superposición para segundos vecinos

La comparación de los resultados correspondientes la superposición de los movimientos para esta prueba se muestran en la tabla 6.3.

Superposición de los movimientos. <i>Riser</i> con 8 segmentos								
$n$	$t_{100}$ (1er Vecinos, 2do Vecinos)		$t_{101}$ (1er Vecinos, 2do Vecinos)		$t_{102}$ (1er Vecinos, 2do Vecinos)		$t_{103}$ (1er Vecinos, 2do Vecinos)	
10	-7.063910E-03	-7.742336E-03	-7.019826E-03	-7.797924E-03	-6.907654E-03	-7.791796E-03	-6.726176E-03	-7.721555E-03
20	-1.410334E-02	-1.584999E-02	-1.401532E-02	-1.601859E-02	-1.379137E-02	-1.606765E-02	-1.342904E-02	-1.599171E-02
30	-2.099716E-02	-2.325837E-02	-2.086613E-02	-2.345954E-02	-2.053269E-02	-2.347956E-02	-1.999326E-02	-2.331088E-02
40	-1.14011	-1.11607	-1.11454	-1.09076	-1.08604	-1.06267	-1.05468	-1.03187
50	-1.39728	-1.40756	-1.36595	-1.37611	-1.33101	-1.34100	-1.29258	-1.30233
60	-1.63629	-1.68989	-1.59959	-1.65262	-1.55868	-1.61079	-1.51367	-1.56452
70	-2.510596E-02	-2.021624E-02	-2.362785E-02	-1.984045E-02	-2.202464E-02	-1.929876E-02	-2.030496E-02	-1.853681E-02
80	-2.767916E-02	-2.272061E-02	-2.604954E-02	-2.220884E-02	-2.428201E-02	-2.151776E-02	-2.238609E-02	-2.059305E-02
90	-2.983056E-02	-2.689450E-02	-2.807429E-02	-2.580012E-02	-2.616937E-02	-2.453260E-02	-2.412609E-02	-2.306439E-02
100	-1.368259E-02	-1.443152E-02	-1.591373E-02	-1.661902E-02	-1.826016E-02	-1.890768E-02	-2.073010E-02	-2.130510E-02

Tabla 6.3: Resultados de la superposición de los movimientos. *Riser* con 8 segmentos

### 6.3.2 Prueba con 16 segmentos

Al igual que en la prueba anterior, también se creó un archivo con los datos de entrada, los cuales se muestran la tabla 6.4.

Datos de entrada	
Número de partículas $n$	256
Masa de las partículas $m$	1
Constante de elasticidad $k$	1
Longitud del <i>riser</i> $Long$	1500
Número de Segmentos $Num.Seg$	16

Tabla 6.4: Datos de entrada para la ejecución de la prueba de 16 segmentos

Con estos datos de entrada se ejecutó el proceso de cómputo de datos y se generaron los archivos que se muestran en la figura 6.20.

```

cs:16-11-2009 00-34-49 jguzman$ ls -ltr
total 12972216
-rw-r--r-- 1 jguzman staff 9366945 Nov 16 00:41 SumaNormVec2.txt
-rw-r--r-- 1 jguzman staff 9140567 Nov 16 00:41 SumaNormVec1.txt
-rw-r--r-- 1 jguzman staff 3268520072 Nov 16 00:41 ModNormVec2.txt
-rw-r--r-- 1 jguzman staff 3303677064 Nov 16 00:41 ModNormVec1.txt
-rw-r--r-- 1 jguzman staff 12615500 Nov 16 00:42 SumaAleaVec2.txt
-rw-r--r-- 1 jguzman staff 12616476 Nov 16 00:42 SumaAleaVec1.txt
-rw-r--r-- 1 jguzman staff 12912264 Nov 16 00:42 MovAleaVec2.txt
-rw-r--r-- 1 jguzman staff 12912264 Nov 16 00:42 MovAleaVec1.txt
cs:16-11-2009 00-34-49 jguzman$

```

Figura 6.20: Archivos que se generaron de la prueba de 16 segmentos

Una vez que se generaron los archivos de resultados, se hizo la visualización de los archivos que contienen los modos aleatorios de vibración para los modelos de primeros y segundos vecinos. La visualización correspondiente al archivo *MovAleaVec1.txt* se muestra en la figura 6.21.

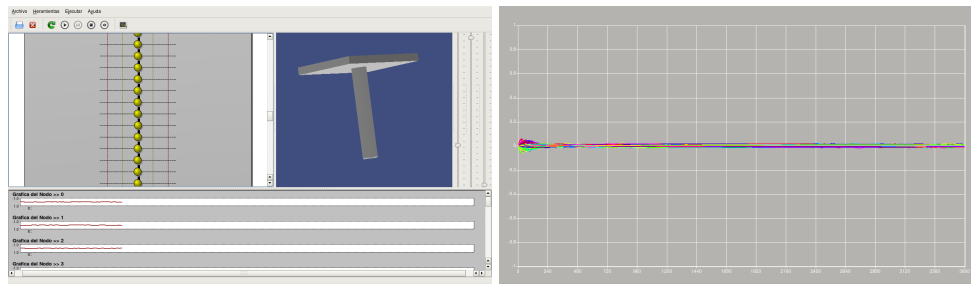


Figura 6.21: Visualización de los modos aleatorios con 16 segmento para primeros vecinos

Por su parte la visualización que se generó para el archivo *MovAleaVec2.txt* se muestra en la figura 6.22.

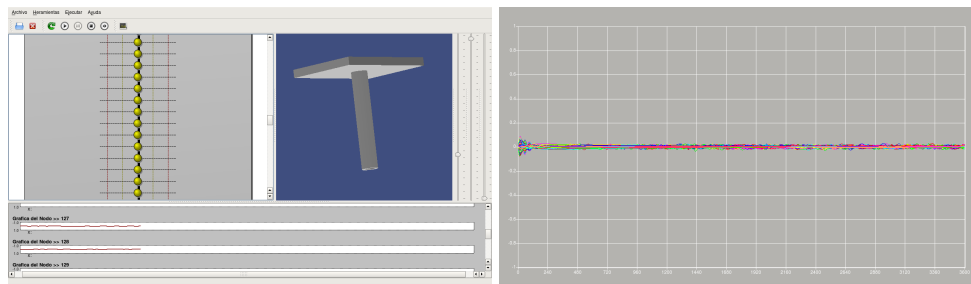


Figura 6.22: Visualización de los modos aleatorios con 16 segmento para segundos vecinos

Por su parte, la tabla 6.5 muestra los resultados obtenidos de los modos aleatorios de vibración que obtuvieron con el sistema para esta prueba.



Modos aleatorios de vibración. Riser con 16 segmentos								
$n$	$t_{100}$ (1er Vecinos, 2do Vecinos)		$t_{101}$ (1er Vecinos, 2do Vecinos)		$t_{102}$ (1er Vecinos, 2do Vecinos)		$t_{103}$ (1er Vecinos, 2do Vecinos)	
10	-3.678883E-04	9.624943E-04	-2.971953E-04	9.215801E-04	-2.171444E-04	8.696534E-04	-1.312430E-04	8.034877E-04
20	-6.906331E-04	4.767009E-03	-7.981448E-04	5.123731E-03	-9.059251E-04	5.431342E-03	-1.011415E-03	5.681378E-03
30	5.285075E-03	5.038357E-04	5.379449E-03	2.732549E-04	5.458830E-03	5.111218E-05	5.521549E-03	-1.565643E-04
40	2.230595E-04	4.435430E-03	7.361718E-05	4.480458E-03	-8.221404E-05	4.500663E-03	-2.413793E-04	4.494882E-03
50	3.338236E-02	3.968179E-03	3.284801E-02	3.895328E-03	3.224033E-02	3.801157E-03	3.156222E-02	3.687246E-03
60	4.311035E-02	2.621725E-02	4.253648E-02	2.529160E-02	4.186080E-02	2.421246E-02	4.108621E-02	2.299237E-02
70	-6.986971E-03	9.441948E-03	-6.866159E-03	9.452593E-03	-6.754513E-03	9.450705E-03	-6.651716E-03	9.434445E-03
80	2.673260E-03	-2.097178E-03	2.977755E-03	-1.881491E-03	3.272270E-03	-1.685780E-03	3.556486E-03	-1.510610E-03
90	-8.699710E-03	5.818764E-03	-8.751728E-03	5.919788E-03	-8.774925E-03	5.904634E-03	-8.770801E-03	5.769524E-03
100	8.079447E-05	-9.115489E-04	-2.119206E-04	-9.772019E-04	-4.824622E-04	-1.046817E-03	-7.311178E-04	-1.119169E-03

Tabla 6.5: Resultados de los modos aleatorios de vibración para la prueba de 16 segmentos

Posteriormente se realizó la visualización de los archivos que contienen los resultados de la superposición de los movimientos. La visualización correspondiente al archivo *SumaAleaVec1.txt* se muestra en la figura 6.23.

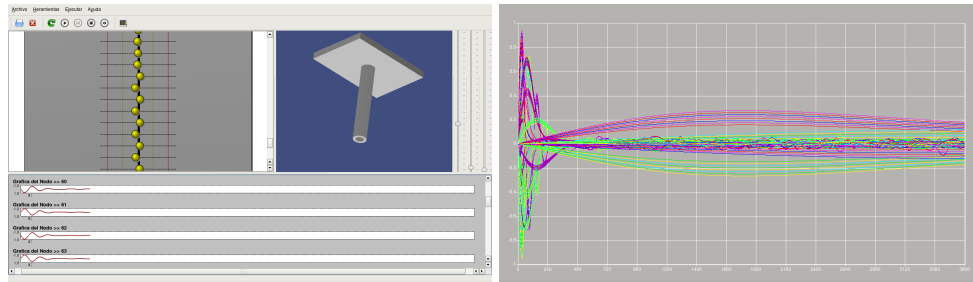


Figura 6.23: Visualización de la superposición con 16 segmentos para primeros vecinos

Por último se realizó la visualización del archivo que contiene los resultados de la superposición correspondiente al modelo de segundos vecinos *SumaAleaVec2.txt* (Fig. 6.24)

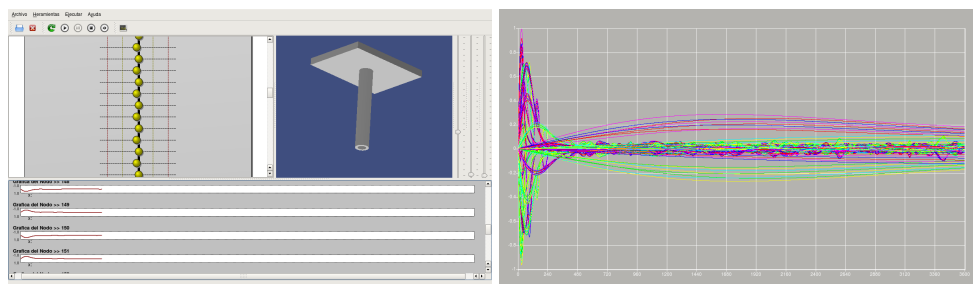


Figura 6.24: Visualización de la superposición con 16 segmentos para segundos vecinos

En la tabla 6.6 se compran los resultados correspondientes a la superposición de los movimientos aleatorios para la prueba con 16 segmentos.

Superposición de los movimientos. <i>Riser</i> con 16 segmentos								
$n$	$t_{100}$ (1er Vecinos, 2do Vecinos)		$t_{101}$ (1er Vecinos, 2do Vecinos)		$t_{102}$ (1er Vecinos, 2do Vecinos)		$t_{103}$ (1er Vecinos, 2do Vecinos)	
10	-2.904481E-03	-2.318145E-03	-2.837077E-03	-2.466894E-03	-2.774331E-03	-2.628331E-03	-2.715826E-03	-2.795956E-03
20	-6.729351E-03	-9.387511E-03	-6.583652E-03	-8.958607E-03	-6.437876E-03	-8.563327E-03	-6.298355E-03	-8.200939E-03
30	-1.001872E-02	-1.345996E-02	-9.801791E-03	-1.287640E-02	-9.584764E-03	-1.233636E-02	-9.377045E-03	-1.184012E-02
40	3.765200E-03	7.211054E-04	4.458917E-03	1.388127E-03	5.127800E-03	2.014992E-03	5.766595E-03	2.595129E-03
50	0.559681	0.564643	0.549596	0.554466	0.538216	0.542966	0.525593	0.530193
60	0.655415	0.681291	0.643604	0.669007	0.630279	0.655050	0.615496	0.639489
70	-4.979879E-02	-5.186686E-02	-4.840570E-02	-5.026108E-02	-4.707154E-02	-4.873393E-02	-4.579417E-02	-4.728381E-02
80	-5.490282E-02	-5.700000E-02	-5.336696E-02	-5.524845E-02	-5.189607E-02	-5.358185E-02	-5.048776E-02	-5.199837E-02
90	-1.402523E-02	-1.247462E-02	-1.408295E-02	-1.162544E-02	-1.404520E-02	-1.073559E-02	-1.391439E-02	-9.815786E-03
100	1.246617E-02	1.139727E-02	1.037795E-02	9.235727E-03	8.338261E-03	7.125000E-03	6.348027E-03	5.066688E-03

Tabla 6.6: Resultados de la superposición de los movimientos. *Riser* con 16 segmentos

### 6.3.3 Prueba con 32 segmentos

La última de las pruebas que se realizó, consiste en un sistema con 32 segmentos y 8 partículas en cada uno de ellos. Los datos de entrada para esta prueba se muestran en la tabla 6.7.

Datos de entrada	
Número de partículas $n$	256
Masa de las partículas $m$	1
Constante de elasticidad $k$	1
Longitud del <i>riser</i> $Long$	1500
Número de Segmentos $Num.Seg$	32

Tabla 6.7: Datos de entrada para la ejecución de la prueba de 32 segmentos

Los archivos de resultados que fueron generados por el sistema de cómputo de datos se muestran en la figura 6.25.

```

cs:16-11-2009 00-43-18 jguzman$ ls -ltr
total 12972448
-rw-r--r-- 1 jguzman staff 9366945 Nov 16 00:50 SumaNormVec2.txt
-rw-r--r-- 1 jguzman staff 9149567 Nov 16 00:50 SumaNormVec1.txt
-rw-r--r-- 1 jguzman staff 326852072 Nov 16 00:50 ModNormVec2.txt
-rw-r--r-- 1 jguzman staff 3303677064 Nov 16 00:50 ModNormVec1.txt
-rw-r--r-- 1 jguzman staff 12676568 Nov 16 00:50 SumaAleaVec2.txt
-rw-r--r-- 1 jguzman staff 12676100 Nov 16 00:50 SumaAleaVec1.txt
-rw-r--r-- 1 jguzman staff 12912228 Nov 16 00:50 MovAleaVec2.txt
-rw-r--r-- 1 jguzman staff 12912264 Nov 16 00:50 MovAleaVec1.txt
cs:16-11-2009 00-43-18 jguzman$

```

Figura 6.25: Archivos que se generaron de la prueba de 32 segmentos

Al igual que en las pruebas anteriores, también se realizaron las visualizaciones de los resultados correspondientes a los modos aleatorios de vibración. La simulación y la gráfica que se generó para el archivo *MovAleaVec1.txt* se muestra en la figura 6.26.

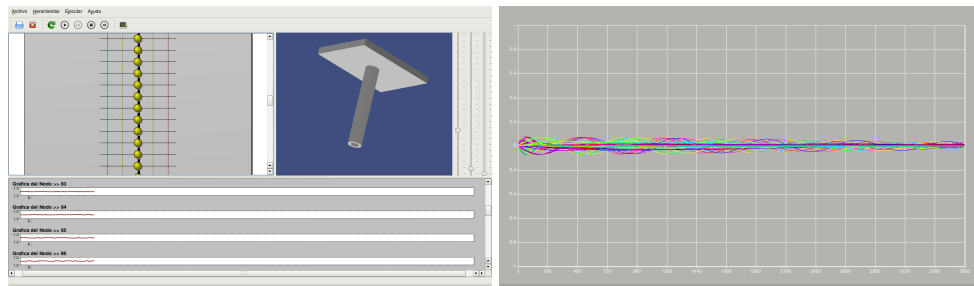


Figura 6.26: Visualización de los modos aleatorios con 32 segmento para primeros vecinos

Posteriormente se hizo la visualización para el archivo que contiene los resultados para los modos aleatorios de vibración del modelo de segundos vecinos *MovAleaVec2.txt* (Fig. 6.27).

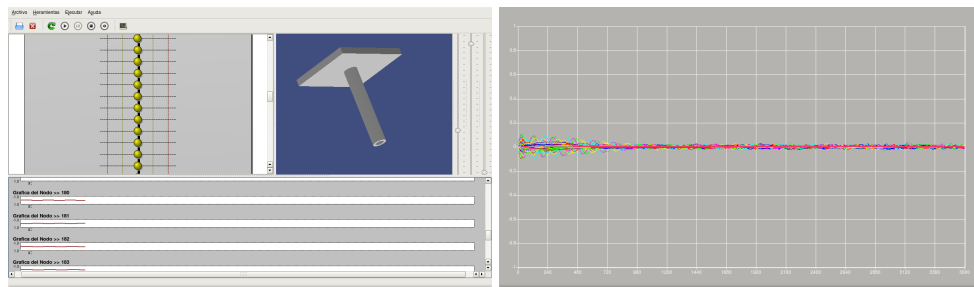


Figura 6.27: Visualización de los modos aleatorios con 32 segmentos para segundos vecinos

La tabla 6.8 muestra una comparación de los movimientos aleatorios de vibración obtenidos para los modelos de primeros y segundos vecinos.

Modos aleatorios de vibración. <i>Riser</i> con 32 segmentos								
$n$	$t_{100}$ (1er Vecinos, 2do Vecinos)		$t_{101}$ (1er Vecinos, 2do Vecinos)		$t_{102}$ (1er Vecinos, 2do Vecinos)		$t_{103}$ (1er Vecinos, 2do Vecinos)	
10	6.876415E-04	-8.850906E-04	6.828061E-04	-8.441567E-04	6.738878E-04	-7.997585E-04	6.608858E-04	-7.524574E-04
20	-4.036903E-04	4.386333E-04	-3.535228E-04	4.923437E-04	-3.069670E-04	5.486439E-04	-2.639329E-04	6.071637E-04
30	-1.502593E-03	4.304219E-03	-1.427628E-03	4.331979E-03	-1.330414E-03	4.311639E-03	-1.213468E-03	4.243405E-03
40	-1.446174E-03	-3.393252E-04	-1.742488E-03	-1.285509E-04	-1.987391E-03	1.928559E-05	-2.176788E-03	9.841991E-05
50	4.417054E-03	9.598906E-05	4.218118E-03	8.527814E-05	3.988178E-03	7.397072E-05	3.730056E-03	6.492515E-05
60	-5.210994E-04	9.901992E-03	-5.576548E-04	9.453511E-03	-5.840030E-04	8.937095E-03	-6.016193E-04	8.358353E-03
70	-1.827724E-04	-4.078187E-03	-5.563378E-05	-4.705289E-03	1.087742E-04	-5.362171E-03	3.090075E-04	-6.031497E-03
80	-1.033875E-02	1.173887E-02	-1.041135E-02	1.198066E-02	-1.043335E-02	1.211056E-02	-1.040436E-02	1.211814E-02
90	-4.143782E-04	2.221534E-03	-3.586058E-04	2.288717E-03	-2.719905E-04	2.301937E-03	-1.560925E-04	2.268540E-03
100	-7.358078E-03	-1.783243E-04	-6.713003E-03	-1.901416E-04	-6.090183E-03	-2.019299E-04	-5.489950E-03	-2.138223E-04

Tabla 6.8: Resultados de los modos aleatorios de vibración. *Riser* con 32 segmentos

Una vez que se tuvieron las visualizaciones de los modos aleatorios de vibración para los modelos de primeros y segundos vecinos, se procedió a generar la visualización de la superposición de los movimientos para ambos modelos. El primer archivo que se ejecutó en el simulador fue el correspondiente a la superposición de los movimientos del modelo de primeros vecinos *SumaAleaVec1.txt*(Fig. 6.28)

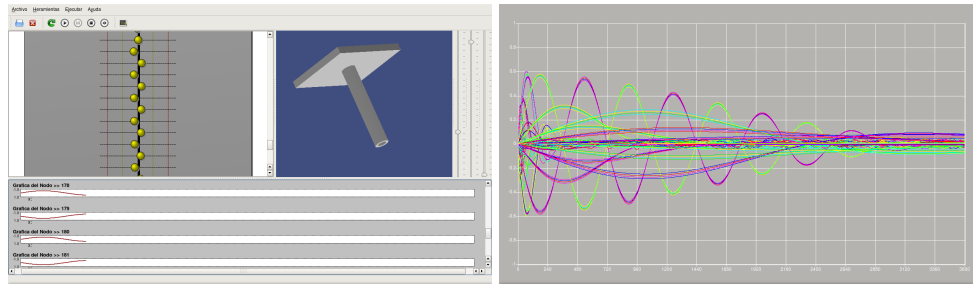


Figura 6.28: Visualización de la superposición con 32 segmentos para primeros vecinos

Finalmente se generó la simulación correspondiente al archivo que contiene los resultados de la superposición de los movimientos correspondientes al modelo de segundos vecinos (Fig. 6.29).

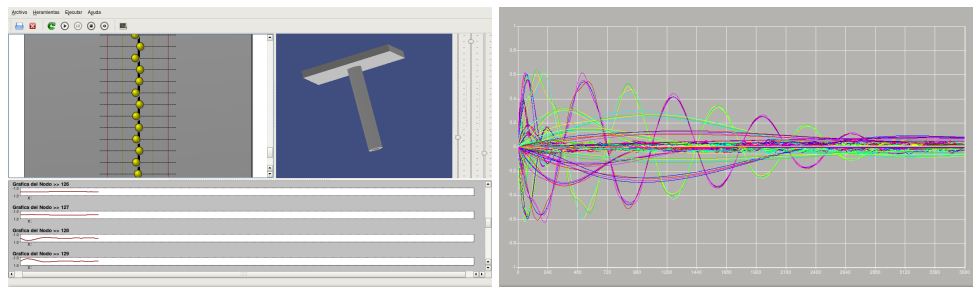


Figura 6.29: Visualización de la superposición con 32 segmentos para segundos vecinos

En la tabla 6.9 se muestran los resultados correspondientes a la superposición de los movimientos obtenidos para la prueba con 32 segmentos.

Superposición de los movimientos. <i>Riser</i> con 32 segmentos								
$n$	$t_{100}$ (1er Vecinos, 2do Vecinos)		$t_{101}$ (1er Vecinos, 2do Vecinos)		$t_{102}$ (1er Vecinos, 2do Vecinos)		$t_{103}$ (1er Vecinos, 2do Vecinos)	
10	-3.002102E-04	-6.024964E-04	-2.033571E-04	-4.529762E-04	-1.048360E-04	-2.958368E-04	-5.790993E-06	-1.334611E-04
20	-2.570995E-03	-2.361001E-03	-2.249602E-03	-2.197622E-03	-1.936035E-03	-2.040577E-03	-1.629300E-03	-1.887964E-03
30	-5.866234E-03	-5.350759E-03	-5.801875E-03	-4.665426E-03	-5.672919E-03	-3.997867E-03	-5.483500E-03	-3.364217E-03
40	-9.734591E-03	-1.007392E-02	-9.905836E-03	-1.003439E-02	-1.004572E-02	-1.002643E-02	-1.014799E-02	-1.004957E-02
50	-8.916042E-04	-4.443963E-04	-1.279588E-03	-9.231197E-04	-1.746433E-03	-1.464311E-03	-2.301221E-03	-2.075825E-03
60	3.514747E-03	-1.472781E-03	3.480793E-03	-1.428056E-03	3.444052E-03	-1.339760E-03	3.403790E-03	-1.207379E-03
70	4.039155E-04	1.159850E-03	5.015741E-04	1.073712E-03	6.165056E-04	1.055959E-03	7.455526E-04	1.110428E-03
80	7.860347E-03	1.959921E-02	6.460517E-03	1.844117E-02	4.917956E-03	1.702852E-02	3.268740E-03	1.538688E-02
90	1.746936E-03	-4.621875E-03	1.785556E-03	-4.204112E-03	1.833161E-03	-3.694265E-03	1.887605E-03	-3.104652E-03
100	-9.515047E-02	-9.466318E-02	-9.029475E-02	-8.982857E-02	-8.556519E-02	-8.512270E-02	-8.096351E-02	-8.054730E-02

Tabla 6.9: Resultados de la superposición de movimientos. *Riser* con 32 segmentos

## 6.4 Comparación de resultados con otros modelos

En la literatura actual, existen investigaciones que utilizan diversos métodos con los cuales se puede hacer un análisis y monitoreo de las vibraciones que sufren los *riser*

por la condiciones marinas. Estos métodos implementan diversos modelos hidrodinámicos de corrientes, los cuales son diferentes y más complejos al que se implementó en este trabajo de investigación.

Debido a las diferencias ente los modelos hidrodinámicos que se implementan para el monitoreo de los *riser*, es muy complicado hacer una comparación directa de los resultados de nuestro modelo con los otros resultados. Otro de los factores que complica la comparación de resultados entre los modelos, es que utilizan diferentes parámetros de entrada para realizar los experimentos y en muchas ocasiones no se cuenta con los datos necesarios para reproducir el mismo experimento con otro modelo. Por su parte, los simuladores que existen actualmente para el monitoreo de los *riser*, son desarrollados por empresas privadas y obtener una licencia de estos para realizar la comparación de resultados sería muy costoso.

Es posible hacer una comparación de las frecuencias normales del sistema de nuestro modelo con algunos otros modelos implementados actualmente. Para esto, se necesitó investigar algunas publicaciones que ofrecieran resultados numéricos de sus pruebas y los parámetros que utilizaron para poder reproducir el experimento.

En los artículos presentados por Senjanovic [43] y Sparks [44], se realiza un análisis de las vibraciones naturales de *riser* con diferente tensión, a través de un método de segmentación. El método que utilizan tiene características similares al nuestro ya que también hacen una segmentación del sistema en nodos, además proporcionan los parámetros que utilizaron para realizar sus pruebas. Esto nos permite reproducir las pruebas que ellos realizaron, principalmente para obtener las frecuencias naturales del *riser*.

Los parámetros que se utilizan [44] y [43] para las pruebas que realizaron se muestran en la tabla 6.10:

Características del <i>riser</i>	
Longitud del <i>riser</i> $L$	2000m
Resistencia de flexión $E$ (Módulo de Young)	$318.6 \cdot 10^6 Nm^2$
Peso Lineal $w$	$3433.5 N/m$
Tensión superior $T_t$	$7.5537 \cdot 10^6 N$
Tensión de fondo $T_b$	$0.6867 \cdot 10^6 N$
Masa del <i>riser</i> $m$	1200Kg/m
Modos Normales $n$	50

Tabla 6.10: Parámetros del *riser* de ejemplo

Para reproducir estas pruebas, de la tabla 6.10 se identificaron los datos que necesita nuestro modelo. Los parámetros que se requieren son: la longitud del *riser* ( $L$ ), el número de partículas( $n$ ), la constante de elasticidad ( $k$ ) y la masa de las partículas ( $m$ ). Para la masa de las partículas, en la tabla de datos se proporciona la cantidad de masa por metro, por lo que fue necesario hacer el cálculo para obtener la masa de cada partícula. Los valores de los parámetros que se utilizaron para reproducir el ejemplo con nuestro modelo se muestran en la tabla 6.11

Parámetros utilizados para nuestro modelo	
Longitud del riser $L$	2000m
Constante de elasticidad $k$ (Módulo de Young)	$318.6 \cdot 10^6 Nm^2$
Masa del riser $m$	2400000Kg
Masa de las partículas $m$	48000Kg
Modos Normales (Partículas) $n$	50

Tabla 6.11: Parámetros del ejemplo para nuestro modelo

Una vez que se reprodujo la prueba, fue necesario hacer la conversión de los resultados, ya que las frecuencias obtenidas por nuestro modelo están en grados y los resultados reportados están en radianes.

La obtención de las frecuencias en nuestro modelo se hace por medio de los valores propios del sistema, el cual se representa con la ecuación 6.1

$$MX = \omega^2 X \quad (6.1)$$

Para obtener la frecuencia natural del riser es necesario obtener el valor de  $\omega$  y hacer la conversión a radianes mediante la ecuación 6.2

$$\omega = \left(\sqrt{\omega^2}\right) \cdot \left(\frac{\pi}{180}\right) \quad (6.2)$$

De esta forma se obtuvieron los resultados para compararlos con los resultados reportados en el artículo [43]. La comparación de los resultados se muestra en la tabla 6.12

Frecuencias naturales del riser $\omega_n$							
$n$	Resultados Literatura				Nuestros Modelos		
	Exacta	Asintótica	Segmentación	FEM	1er Vecinos	2do Vecinos	Promedio
1	0.07973	0.08108	0.08115	0.07983	0.08761	0.19357	0.10050
2	0.16140	0.16217	0.16238	0.16176	0.17506	0.38670	0.20158
3	0.24273	0.24326	0.24399	0.24370	0.26238	0.57847	0.30242
4	0.32395	0.32435	0.32610	0.32602	0.34946	0.76840	0.40305
5	0.40511	0.40544	0.40886	0.40891	0.43623	0.95586	0.50340
10	0.81072	0.81089	0.83553	0.83580	0.86213	1.83570	0.99846
20	1.62170	1.62179	1.77296	1.77331	1.64311	2.87707	1.88499
30	2.43263	2.43268	2.84559	2.84630	2.26944	3.15289	2.66326
40	3.24353	3.24357	4.07467	4.07600	2.68221	3.42449	3.45741
50	4.05443	4.05447	5.47942	5.48210	2.84252	3.55103	4.24400

Tabla 6.12: Comparación de resultados con otros modelos

Como se observa en la gráfica 6.30, los resultados obtenidos por nuestros modelos tienen un comportamiento muy similar a las frecuencias obtenidas con los otros métodos y la diferencia entre ellos es muy pequeña. Sin embargo, se observa que el modelo de

segundos vecinos es el que tienen un comportamiento un poco diferente a los demás, lo que nos puede indicar que la interacción de más partículas vecinas puede influir en el comportamiento del sistema, principalmente en su rigidez y flexibilidad.

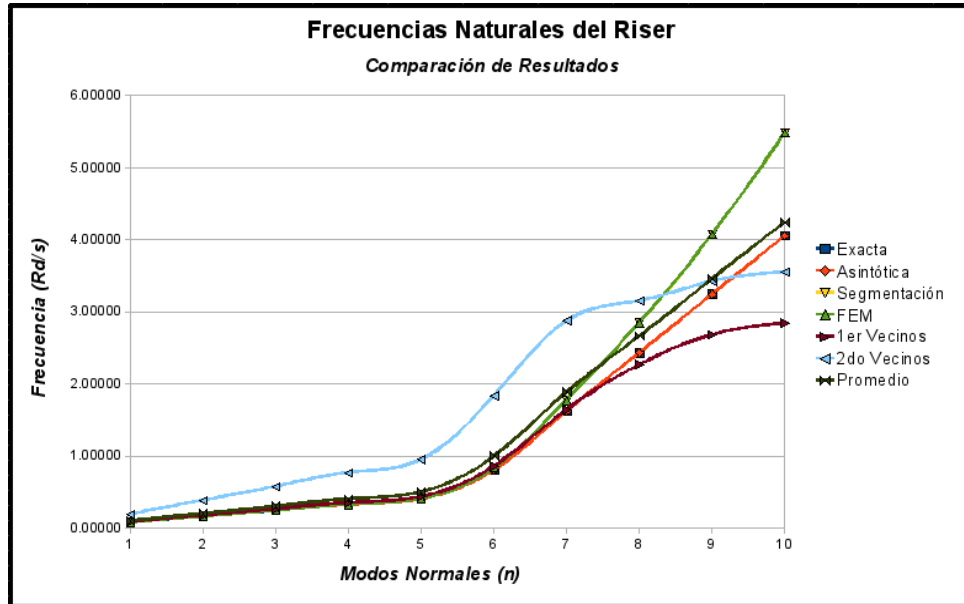


Figura 6.30: Gráfica de comparación de frecuencias naturales del riser

## 6.5 Resumen

En este capítulo se describió la forma en como fue validado nuestro simulador y las pruebas que se hicieron para corroborar su funcionamiento. En la primer sección se describen los pasos a seguir para hacer un uso adecuado de nuestro simulador.

En la segunda sección se describe el diseño de las pruebas que se realizaron con el simulador. Se mostraron los diferentes valores de entrada (masa, elasticidad, longitud de la estructura, etc.) con los que se realizaron las pruebas.

En la tercera sección, se mostraron los resultados obtenidos de cada una de las pruebas realizadas para los modelos de primeros y segundos vecinos.

Finalmente en la cuarta sección se hizo el diseño de una prueba adicional para hacer una comparación de los resultados de nuestro modelo con otros modelos, tomando como base los parámetros de entrada reportados por los otros modelos.





# Capítulo 7

## Resultados, discusión, y conclusiones

La aventura podrá ser loca, pero el  
aventurero ha de ser cuerdo.

---

Gilbert Keith Chesterton

### 7.1 Resultados

En este trabajo de tesis se abordó el problema de las vibraciones que afectan a los *riser* durante el proceso de exploración y explotación de petróleo en aguas profundas. Para ayudar a contrarrestar este problema, se propuso la construcción de un simulador que permita analizar el comportamiento de estas estructuras en diversas condiciones.

Para desarrollar este trabajo de tesis se realizó una investigación para conocer algunos antecedentes del petróleo, así como la importancia y la influencia que tiene éste energético en la economía mundial y especialmente en nuestro país. Además, se dió a conocer el proceso de exploración y explotación de petróleo en aguas profundas y se describieron los diferentes retos que se presentan al realizar este proceso. Específicamente se analizó el problema de las vibraciones que generan las corrientes marinas sobre las tuberías (conocidas como *riser*), que se utilizan para la extracción de petróleo en aguas profundas. Durante esta investigación, también se describieron algunos desarrollos de software que se utilizan en la industria petrolera para controlar y monitorear los sistemas de *riser*.

Con la información anterior, se realizó la construcción del modelo matemático sobre el cual se basa el simulador. Para el desarrollo de este modelo, se hizo una representación física que nos permitió determinar que el sistema se podía tratar como una cadena de vibración monoatómica unidimensional formada por partículas unidas por resortes.

Una vez que se definió la forma en como se podía tratar el sistema, se desarrollaron dos modelos matemáticos: interacción a primeros y segundos vecinos. A partir de estos modelos se obtuvieron dos matrices de movimiento, que nos permiten obtener los modos de vibración del sistema y predecir su comportamiento.

Posteriormente, se desarrolló un modelo estocástico de corrientes basado en el método Monte Carlo. Con este modelo podemos generar diferentes condiciones de corrientes para probar que nuestros modelos matemáticos funcionan adecuadamente.

Con estos modelos, se realizó el desarrollo del simulador. Para esto, se hizo un diseño general del sistema en donde se describen las diversas herramientas y métodos numéricos que necesitaban para su desarrollo.

Como resultado final de este trabajo de tesis, desarrollamos un simulador el cual esta conformado de dos aplicaciones: cómputo de datos y visualización. Las cuales se encargan de realizar los cálculos de los datos y la animación gráfica de los resultados respectivamente.

El simulador se sometió a varias pruebas para comprobar su funcionalidad y analizar el comportamiento del sistema con los dos modelos desarrollados. Los resultados de las pruebas realizadas se pueden observar en la sección 6.3 de este documento.

## 7.2 Discusión

Durante este trabajo de tesis nos enfrentamos a diversos problemas y para resolverlos se utilizaron diversas técnicas, estrategias y herramientas que fueron descritas en este documento.

Para elegir una manera adecuada de tratar nuestro problema, se realizó una investigación para conocer el proceso de exploración y explotación de petróleo en aguas profundas, y analizar cómo afectan las corrientes a las tuberías conocidas como *riser* que se utilizan para este proceso. Con esta información se hizo una representación física del sistema, la cual ayudó a determinar que éste tiene un comportamiento similar al que se presentó en el problema del puente del milenio descrito en la sección 2.4.1 [2]. De esta manera, se decidió tratar el problema como una cadena de vibración y se optó por la construcción de los dos modelos matemáticos que son la base para de nuestro simulador.

En un principio de esta investigación, sólo se tenía contemplado la construcción del modelo matemático, sin embargo, se necesitaba tener un mecanismo adicional para generar diversas condiciones de corrientes que nos permitiera probar nuestro modelo matemático. Por esta razón, se desarrolló un modelo estocástico basado en el método Monte Carlo, que se encargara de generar estas condiciones de corrientes de manera aleatoria.

Debido a que nuestro modelo de corrientes esta basado en el método Monte Carlo, fue necesario elegir una distribución de probabilidad para poder generar datos aleatorios con un cierto comportamiento. A través de una investigación que realizamos, observamos que la distribución más usada para simular fenómenos naturales, como las corrientes marinas, es la distribución normal.

Por otra parte, fue necesario determinar el tamaño de la muestra para cada partícula y el número de veces que se tenían que generar estos datos. Respecto al vector de frecuencias para cada partícula se determinó que fuera de 100 elementos y que cada uno

de estos elementos se debía generar 1000 veces, ya que al realizar pruebas con el sistema se consideró que estas cantidades generan condiciones aleatorias adecuadas para probar el sistema. Sin embargo, para comprobar que estos datos sean correctos y que estén apegados a la realidad, es necesario hacer pruebas físicas y comparar los resultados. Otro aspecto que se tomó en consideración para la generación de frecuencias, es que debían estar en función de nuestro modelo matemático.

Para poder generar diversas condiciones de corriente a lo largo de la estructura, se hizo la división del *riser* en segmentos, la cual consiste en dividir el número de partículas del *riser* en un número de segmentos proporcionado por el usuario, de tal forma que se tenga una distribución uniforme de partículas en todos los segmentos. Esta división, permite determinar los rangos (valores de  $\mu$  y  $\sigma$ ) en los que la distribución de probabilidad va a generar los números aleatorios para cada una de las partículas del segmento. Con esta división de segmentos se van a poder generar condiciones de corriente en 1 dimensión en 2 direcciones distintas.

Con el simulador que se desarrolló se pueden diseñar un gran número de pruebas, variando las condiciones iniciales del sistema (número de partículas, segmentos, longitud de *riser*, etc.). La pruebas que se eligieron para presentar en este documento, muestran el funcionamiento del sistema con la longitud máxima que puede tener un *riser* en aguas profundas. Además, se hizo la división de las partículas en una distancia relativamente corta y se probó con 3 números de segmentos diferentes. Tomado los resultados obtenidos por las pruebas realizadas, se hizo un análisis del comportamiento del sistema en los diversos escenarios.

Respecto a los resultados obtenidos en las pruebas realizadas, en la prueba con 8 segmentos y 256 partículas para los modelos de primeros y segundos vecinos (Fig. 6.16 y 6.17), podemos observar que al calcular los movimientos aleatorios de vibración de las partículas, se presentan movimientos muy pequeños a lo largo de la estructura. Además, se observa que en el modelo de segundos vecinos se tienen más oscilaciones (aunque muy pequeñas) y en el modelo de primeros vecinos las partículas mantienen un desplazamiento estable que se va reduciendo conforme transcurre el tiempo. Sin embargo, al realizar la superposición de estos movimientos (Fig. 6.18 y 6.19), la estructura se ve claramente afectada, principalmente al inicio de la simulación ya que es cuando partículas del sistema tienen un desplazamiento mayor y al realizar la superposición se tienen movimientos considerables. Como podemos observar el comportamiento en los dos modelos muy parecido, esto se debe principalmente a que se utilizaron las mismas condiciones iniciales y los mismos valores de frecuencias aleatorias para realizar los cálculos.

Para el caso de la prueba con 16 segmentos y 256 partículas (Fig. 6.21, 6.22, 6.23 y 6.24), pudimos observar que al realizar el cálculo de los modos aleatorios de vibración y la superposición de estos, la amplitud de los movimientos de la estructura son más pequeños en comparación con la prueba anterior. Esto se debe principalmente a que al tener más segmentos se generan más condiciones diferentes de frecuencias, por lo que el movimiento de las partículas de un segmento se puede ver afectado por el movimiento de las partículas de otro segmento y se reduce su amplitud. Finalmente pudimos

observar que el comportamiento del sistema es muy parecido al de la prueba anterior, donde al realizar la superposición de los movimientos la estructura se ve afectada al inicio de la simulación.

En la prueba con 32 segmento y 256 partículas (Fig. 6.26, 6.27, 6.28 y 6.29), se observó que presentaba una mayor oscilación de las partículas a lo largo de la simulación, lo que puede ser provocado principalmente a los valores de frecuencia que se generaron para ejecutar la simulación.

## 7.3 Conclusiones

En este proyecto de investigación se desarrolló un sistema que permite predecir y analizar el comportamiento dinámico de los *riser* durante el proceso de exploración de petróleo en aguas profundas. El sistema está compuesto por un modelo matemático, un modelo de corrientes y un sistema de visualización de resultados.

Durante la investigación, nos dimos cuenta que desarrollar un simulador con estas características no es una tarea sencilla. Además de requerir conocimientos técnicos y matemáticos, es necesario conocer ampliamente el proceso de exploración de petróleo en aguas profundas para poder cubrir todos los requerimientos del sistema.

Los resultados obtenidos por el sistema nos ayudarán a identificar problemas en las estructuras y determinar puntos críticos que pueden ser susceptibles a accidentes. Además, nos van permitir generar políticas de decisión que ayuden en la construcción de los *riser*, para lograr que sean más flexibles, resistentes y puedan soportar la fuerza de las corrientes marinas.

Uno de los aspectos más importantes durante el desarrollo del simulador, fue la elección y la construcción de nuestro modelo matemático, el cual está basado en cadenas de vibración. Este modelo es la base de nuestro simulador ya que nos permite conocer o predecir el comportamiento del sistema a partir de ciertas condiciones iniciales.

Por otra parte, el modelo estocástico de corrientes que se desarrolló y la división de segmentos, nos permiten experimentar con el sistema en diversas condiciones que pueden ser muy difíciles de generar físicamente. A pesar de que este modelo de corrientes es simple, sirve como base para poder generar nuevos modelos más complicados y que puedan generar condiciones de corrientes diferentes.

Utilizar la biblioteca IMSL nos permitió desarrollar más rápida y adecuadamente el sistema, gracias a que cuenta con rutinas especializadas y optimizadas que nos ayudaron a resolver el modelo matemático y el modelo de corrientes.

Con las pruebas que se realizaron, comprobamos que los resultados obtenidos están apegados a nuestro modelo matemático, por lo que podemos concluir que el sistema está listo para utilizar datos oceanográficos reales. Sin embargo, para poder comprobar totalmente el funcionamiento del simulador es necesario realizar una prueba física conocida como "*Prueba de Tanque*", la cual requiere de equipo especializado y es demasiado costosa.

Debido a la situación actual de la industria petrolera y al interés por buscar nuevas

fuentes proveedoras, se considera que este proyecto surge en un momento adecuado. Principalmente por que en nuestro país, PEMEX está buscando fuentes de energía en aguas profundas y requieren de información, conocimientos y tecnologías como la que se desarrolló en este proyecto.

Por último, las aportaciones que se tuvieron con este proyecto de investigación son:

- El desarrollo de un modelo matemático basado en cadenas de vibración para analizar el comportamiento dinámico de los *riser*.
- El desarrollo de un modelo estocástico de corrientes basado en Monte Carlo.
- Diseño y desarrollo de un sistema que implementa los dos modelos creados y genera los resultados del comportamiento del sistema.
- Diseño y desarrollo de un sistema de visualización para generar imágenes y animaciones a partir de los resultados obtenidos por el sistema anterior.

## 7.4 Trabajo futuro

Algunas recomendaciones que se pueden realizar para mejorar nuestro simulador son:

- Aunque las rutinas de la biblioteca IMSL estan optimizadas para funcionar en plataformas multicore, es necesario realizar un análisis para determinar los algoritmos se pueden paralelizar y reducir el tiempo de procesamiento.
- Someter el sistema a pruebas con datos oceanográficos reales, como los que obtiene CICESE a partir de un estudio que esta realizando en el golfo de México. Además, realizar la *prueba de tanque* que nos permita comprobar totalmente el funcionamiento del simulador.
- Realizar un análisis de los materiales que se utilizan para la construcción de los *riser* y determinar los límites de confianza del sistema.
- Desarrollar un modelo más complejo que nos permita generar corrientes en 2 o más dimensiones y que permita simular situaciones como tener dos o mas *riser* trabajando juntos.
- Mejorar el sistema de visualización agregandole más herramientas que permitan tener opciones adicionales como: graficar resultados por segmento, generar solo un tipo de visualización, etc.
- Mejorar el diseño de la animación 3D o generar animaciones con otras herramientas como el Quicktime reality.



# Bibliografía

- [1] Petróleos Mexicanos. El petróleo. historia, 2009. Disponible en <http://www.pemex.com>. Citada en septiembre 2009.
- [2] Francoise Tisseur and Karl Meerbergen. The quadratic eigenvalue problem. *Society for Industrial and Applied Mathematics*, 43(2):235–286, 2001.
- [3] William W. Seto. *Vibraciones Mecánicas*. 1970.
- [4] Amilcar Meneses Viveros. *Simulación de paquetes de onda y teoría de Weyl*. PhD thesis, Centro de Investigación y de Estudios Avanzados IPN, México, D.F., 2009.
- [5] Shelly Bauman and Rockford MI. *Intermediate Energy Infobook*. NEED, Manassas, VA 20108 PO Box 10101, need edition, 2008-2009.
- [6] Instituto Mexicano del Petróleo. El petróleo, 2009. Disponible en <http://www.imp.mx>. Citada en septiembre 2009.
- [7] Jean A. Bandy Georgius Agricola, Mark Chance Bandy. *De Natura Fossilium*. Dover Publications, 2004.
- [8] Alphonsus Fagan. *An introduction to the petroleum industry*. Government of Newfoundland and Labrador, Department of mines and energy, November 1991.
- [9] Kjell Aleklett Bengt Söderbergh. Canada's oil sands resources and its future impact on global oil supply. *UPPSALA UNIVERSIT, Box 535 SE-751 21 Sweden*, page 5, 2005.
- [10] Energy Watch Group. Crude oil the supply outlook. *Report to the Energy Watch Group, EGW-Series No. 3/2007*, page 18, Octubre 2007.
- [11] Georg Bauer and Rolf Jakoby. A simple prediction model for line of sight coverage in urban scenarios. In *VTC Fall*, pages 1–5. IEEE, 2008.
- [12] Joel Álvarez de la Borda. *Crónica del petróleo en México de 1863 hasta nuestros días*. Archivo histórico de petróleos mexicanos, primera edición edition, 2006.
- [13] Joel Álvarez de la Borda. *Los orígenes de la industria petrolera en México 1900-1925*. Archivo histórico de petróleos mexicanos, primera edición edition, 2005.

- [14] Gerardo Gil Valdivia and Susana Chacón Domínguez. *La crisis del petróleo en México*. Tecnológico de Monterrey, Abril 2008.
- [15] Abelardo Cantú Chapa. *Golfo de México*. *PetroQuiMex*, pages 24–30, Marzo-Abril 2008.
- [16] Amparo García Acevedo. Rumbo a aguas profundas; el próximo reto en educación marítima. *Fidenaútica*, (1):8–12, Enero-Febrero 2008.
- [17] Jaeyoung Lee. Design and installation of deepwater petroleum pipelines. In *The World Congress of Korean and Korean Ethnic Scientists and Engineers, Seoul, Korea, 2002*. Marine Pipeline Engineer.
- [18] Jaeyoung Lee. *Introduction to Offshore Pipelines and Risers*. Primera edición edition, 2009.
- [19] Mineral Management Service. Deep water, where the energy is. *MMS: Securing Ocean Energy And Economic Value For America*, 2004.
- [20] Rita Tuub. Deepwater gulf of mexico: America's offshore energy future. *Pipeline and Gas Journal*, pages 42–46, Octubre 2008.
- [21] J. C. Berger J. C. Cutler. *Deepwater Reference Book*, volume Volumen 1, 2, 3. septiembre 2000.
- [22] Tomas H. Naylor, Josep L. Balintfy, Donald S. Burdick, and Kong Chu. *Técnicas de Simulación en Computadoras*. 1980.
- [23] María Cristina Rojas. *Burj Al Arab. una superestructura de 7 estrellas*. *Metal Actual*, Mayo-Junio 2008.
- [24] MCS. Flexcom, 2009. Disponible en <http://www.mcs.com/site.php?page=15>. Citada en septiembre 2009.
- [25] Orcina. Orcaflex, 2009. Disponible en <http://www.orcina.com/SoftwareProducts/OrcaFlex/index.php>. Citada en septiembre 2009.
- [26] MCS fugro. *OPTIMA. Rise Operations Planning and Management*. Technical report, MCS, 2008.
- [27] Massachusetts Institute of Technology (MIT). Vortex-induced vibration, 2009. Disponible en <http://web.mit.edu/shear7/shear7.html>. Citada en septiembre 2009.
- [28] Erasto Victor Vergara Nava. *Vibraciones de una Red Diatómica con Interacción a Segundos Vecinos*, 1974. Escuela Superior de Física y Matemáticas- IPN.



- [29] Angel Franco García. *Física con ordenador. Curso Interactivo de física en Internet*, 2009. Disponible en <http://www.sc.ehu.es/sbweb/fisica/>. Citada en septiembre 2009.
- [30] David Halliday Robert Resnick. *Física parte I*, volume Volumen I. 1977.
- [31] Isidro Romero Medina. *Vibraciones en Redes Unidimensionales con Interacciones a Primero Vecinos*, 1971. Escuela Superior de Física y Matemáticas- IPN.
- [32] Jesus Ortega Campos. *Estudio de Vibraciones en Cadenas Lineales con Iteracción a Segundos Vecinos*, 1971. Escuela Superior de Física y Matemáticas- IPN.
- [33] Ian Sommerville. *Ingeniería de Software*. Septima edición edition, 2005.
- [34] Roger S. Presssman. *Ingeniería del Software. Un enfoque Práctico*. 5 edición edition, 2002.
- [35] Leonard Meirovitch. *Fundamentals of Vibrations*. 2001.
- [36] Clarence W. de Silva. *VIBRATIONS Fundamentals and Practice*. 2000.
- [37] Mark Cartwright. *FOURIER METHODS for Mathematicians, scientists and engieners*. 1990.
- [38] Fabiola Ortega Robles. *Núcleo de Monte Carlo y Camino Aleatorio en Ambientes de Alto Desempeño*. PhD thesis, Centro de Invetsigación y de Estudios Avanzados IPN, México, D.F., 2008.
- [39] Sharon L. Myers Ronald E. Walpole, Raymond H. Myers. *Probabilidad y Estadística par Ingenieros*. Sexta edición edition, 1999.
- [40] Luis Miguel de la Cruz Salas. Visualización científica. *Revista Digital Universitaria UNAM*, 6(12):1–9, Diciembre 2005.
- [41] Visual Numerics. *IMSL Math/Library*, 1997.
- [42] Visual Numerics. *IMSL Stat/Library*, 1997.
- [43] Sparks C.P. Natural vibration analysis of tensioned risers by segmentation method. *Oil Gas Science and Technology*, 61(5):647–659, 2006.
- [44] Sparks C.P. Transverse modal vibrations of vertical tensioned risers. *Oil Gas Science and Technology*, 57(1):71–86, 2002.



# Apéndice A

## Instalación del sistema

Para obtener éxito en el mundo, hay que parecer loco y ser sabio.

---

Montesquieu

**E**n este anexo se describen los requerimientos necesarios para poder utilizar el simulador. Además, se describe el procedimiento de instalación, así como los pasos para poder compilar el sistema y obtener los archivos ejecutables a partir del código fuente.

### A.1 Requerimientos de Instalación

Para ejecutar el subsistema de cómputo de datos y visualización es necesario contar con los archivos ejecutables correspondientes a cada uno de ellos. Sin embargo, si el usuario requiere compilar nuevamente los códigos, deberán cumplir con ciertos requerimientos para llevar a cabo este proceso.

#### A.1.1 Requerimientos para el subsistema de Cómputo de Datos

Los requerimientos con los que se debe contar para compilar esta aplicación son:

1. Compilador *Absoft Fortran* en su versión 11.5 o superior
2. Tener instalada la biblioteca IMSL Fortran Numerical Library

Esta aplicación se ha desarrollado y probado en una plataforma Mac Os Server 3.1, sin embargo el compilador Absoft y la biblioteca IMSL esta disponibles para diversas plataformas (Windows, Linux, etc.), por lo que la compilación del código no debe de implicar ningún cambio.

### A.1.2 Requerimientos para el subsistema de Visualización

Los requerimientos para poder compilar esta aplicación son:

1. Compilador Ansi C (gcc) version 4.2.4
2. Se deberá contar con las librerías de Qt correspondientes a la versión 4.0
3. Finalmente se debe contar con la biblioteca OpenGL (Open Graphics Library)

Esta aplicación se ha desarrollado y probado en una plataforma Linux Ubuntu 8.10 con kernel 2.6.24

## A.2 Tutorial de compilación

Al igual que en los requerimientos, el proceso de compilación e instalación de cada una de los subsistemas es distinto, por lo que se dará una explicación por separado para cada uno de ellos.

### A.2.1 Subsistema de Cómputo de Datos

Este subsistema se compone de los siguientes archivos:

- *SistComputo.f90*: Contiene las funciones para generar el cálculo de los resultados
- *SistComputo*: Archivo ejecutable
- *compila*: Archivo que contiene el script para compilar el código fuente

Para poder compilar esta aplicación se tiene que hacer desde línea de comando mediante la siguiente instrucción:

```
f90 SistComputo.f90 -o SistComputo -m64 $LINK_f90
```

donde la instrucción *-m64* indica que se esta compilando en un procesador de 64 bits y *\$LINK\_f90* indica que hacemos uso de las rutinas de la biblioteca IMSL.

Esta instrucción se encuentra en un scrip llamado *compila*, por lo que para compilar solo es necesario ejecutar la siguiente instrucción:

```
./compila
```

Esta instrucción se encarga de generar nuestro archivo ejecutable el cual tendrá el nombre del *SistComputo* y para ejecutar este archivo se realiza mediante el siguiente comando:

```
./SistComputo
```

## A.2.2 Subsistema de Visualización

Para poder compilar el sistema de visualización, es necesario tener todos los archivos correspondientes a la aplicación (código fuente, archivos de configuración, etc.) en una carpeta y dentro de ésta un directorio con las imágenes que se utilizan en la interfaz gráfica.

Para comenzar a compilar el proyecto, es necesario generar el archivo de configuración (*.pro*) y el archivo *Makefile* que nos permite generar el archivo ejecutable. Para generar el archivo de configuración debemos posicionarnos en la carpeta donde tenemos los archivos fuentes e introducir el comando:

```
qmake-qt4 -project
```

Esto nos creará un archivo con el nombre de la carpeta y extensión *.pro* (i.e. Simulador.pro). Con este archivo vamos a generar nuestro *Makefile*, pero antes de generarlo debido a que en nuestra aplicación usamos rutinas de la biblioteca *OpenGL*, es necesario agregar una línea al archivo *.pro* indicándole que vamos hacer uso de la rutinas de *OpenGL*. Para esto es necesario abrir el archivo y agregarle la siguiente instrucción al final:

```
QT += opengl
```

Posteriormente podemos generar nuestro archivo *Makefile* mediante la siguiente instrucción:

```
qmake-qt4
```

Finalmete ejecutamos nuestro archivo *Makefile* y obtendremos como resultado el archivo ejecutable para poder utilizar el simulador.

```
./Makefile
```

Si no se ha realizado ninguna modificación al sistema y se requiere compilar nuevamente el código, no es necesario repetir todo el proceso, sólo se tendrá que ejecutar nuevamente el archivo *Makefile* que generamos anteriormente.