



# UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS

## Título

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal”*

### Autor

Luis Leandro Hernández Fleitas

### Tutores

Lic. Daciel Alberto Olivera Cortina  
Ing. Eduardo Manuel Macías Sotolongo

Ciudad de La Habana, Junio del 2009.

“Año del 50 Aniversario del Triunfo de la Revolución Cubana”

## Declaración de Autoría

Declaro que Luis Leandro Hernández Fleitas es el único autor del trabajo titulado: *“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal”* y autorizo a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma.

Para que así conste se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del Autor

Luis Leandro Hernández Fleitas

---

Firma del Tutor

Ing. Eduardo Manuel Macías Sotolongo

---

Firma del Tutor

Lic. Daciel Alberto Olivera Cortina

## DEDICATORIA

*Dedico este trabajo de diploma a toda mi familia que ha sido siempre una guía firme en mi formación como persona, como profesional. Y en especial:*

*A mi mamá por tantas noches de desvelo, de cuidado, de amor.*

*A mi papá por ser el espejo donde me miro, la meta a superar, el soporte para afrontar las dificultades.*

*A mi hermano para que tenga un ejemplo más por el cual seguir y trazar sus propias metas.*

*A mi abuelita Irma, a Mima, a Pipo por ser los mejores abuelitos del mundo, preocupados, dedicados; por ser su nieto de la universidad.*

*A mis bisabuelos Noevia y Perico quienes estarían muy orgullosos de este momento y la vida no les alcanzo para vivirlo, pero que siempre los llevo en mi corazón.*

*A mis tías y tíos, por estar pendientes de mí en cada momento, por aconsejarme y darme ánimo en los momentos oportunos.*

*A mi novia que se ha ganado un lugar especial en mi corazón y en mi vida.*

## **AGRADECIMIENTOS**

A mis padres y a mi hermano por ayudarme en todo momento con sus consejos y apoyo a salir de los momentos difíciles. A mi familia en general que se ha preocupado tanto por mi formación como profesional.

A mis tutores con los que he aprendido que no existe nada imposible, sino pequeños obstáculos que nos dificultan el camino en espera de que los superemos.

A Daciel por adentrarme en un mundo tan interesante como el de las matemáticas aplicadas.

A todo el claustro de profesores que de una forma u otra haya influido en mi preparación como profesional y que me han enseñado a transitar en el camino por la búsqueda del conocimiento.

A al grupo de Comunidades Virtuales del proyecto GIDI (Laboratorio 4), por los años compartidos y por su gran ayuda en la realización de este trabajo.

A Yamila, mi novia, por estar a mi lado en los momentos más irritables, en los más duros, en los más difíciles, como cuando se estaba llenando, por tercera vez, la base de datos del sistema. Y también por todos los momentos lindos que pasamos juntos, que dan fuerzas y energía para continuar.

A Yonnys por las tantas veces que me brindó su ayuda y sus conocimientos cuando no había más nadie a quien recurrir.

A Yusibel por ser de gran ayuda en la realización una tarea titánica, “El documento”.

Le agradezco a mis compañeros y amigos que han compartido conmigo tantos momentos durante estos años y a todas aquellas personas que de alguna forma ha hecho posible la realización de este trabajo, en especial a las chicas (Karina, Yulianis, Grettel) del apto 44 202 que se la pasaban mandándome a hacer la tesis.

A todos muchísimas Gracias

## RESUMEN

En este Trabajo de Diploma se presenta la información resultante de la investigación y desarrollo de una aplicación web, que utilizando algoritmos matemáticos, brinda trayectorias óptimas a seguir por automóviles y personal. Permitiendo dar solución al proceso de trasladarse de un lugar a otro, que se lleva a cabo diariamente en la Universidad de las Ciencias Informáticas. Este sistema aportará a la institución, grandes ahorros de recursos, como combustibles y tiempo, beneficiando así el desempeño de muchas de las actividades asociadas a esta, directa o indirectamente.

Se hizo una búsqueda y estudio de diferentes algoritmos existentes para tratar los problemas de optimización de redes, en la rama de la investigación de operaciones. Se definen las herramientas empleadas en el desarrollo, las características del sistema y de las tareas realizadas para el diseño e implementación, así como la realización de pruebas que validan la calidad del producto, contando con un análisis estadístico de encuesta donde se refleja la aceptación por parte de los usuarios, y de estudios para determinar la factibilidad del proyecto.

En la implementación del sistema se usó como lenguaje de desarrollo PHP, gestor de base de datos MySQL, modelando el sistema con la metodología Extreme Programming. Empleando diferentes algoritmos matemáticos para el cálculo de distancias, consumos de combustible y conversión de métricas, utilizando Dijkstra para determinar la ruta óptima.

## ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>FUNDAMENTACIÓN TEÓRICA .....</b>	<b>6</b>
1.1 INTRODUCCIÓN .....	6
1.2 LA INVESTIGACIÓN DE OPERACIONES .....	6
1.2.1 Problemas de optimización.....	9
1.2.2 Problemas de optimización en la universidad.....	12
1.3 METODOLOGÍAS DE DESARROLLO PROPUESTA .....	12
1.3.1 Programación Extrema .....	13
1.3.2 Metodología de desarrollo Scrum.....	15
1.3.3 Proceso Racional Unificado y El Lenguaje Unificado de Modelado .....	17
1.3.4 Justificación de la propuesta para la Metodología Programación Extrema .....	18
1.4 TECNOLOGÍAS PROPUESTAS .....	20
1.4.1 Lenguaje HTML.....	20
1.4.2 Lenguaje JavaScript.....	21
1.4.3 Lenguaje PHP.....	21
1.4.4 Servidor Web Apache .....	22
1.5 HERRAMIENTAS PROPUESTAS .....	24
1.5.1 Gestor de Bases de Datos MySQL.....	24
1.5.2 Gestor de Bases de Datos PostgreSQL.....	25
1.5.3 MySQL, gestor de base de datos definido.....	27
1.5.4 Zend Studio.....	27
1.6 CONCLUSIONES .....	28
<b>CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>29</b>
2.1 INTRODUCCIÓN .....	29
2.2 DESCRIPCIÓN .....	29
2.2.1 Situación Actual.....	30
2.2.2 Objeto de automatización.....	31
2.3 CARACTERÍSTICAS DE LA PROPUESTA DE SISTEMA .....	31
2.4 PERSONAS RELACIONADAS CON EL SISTEMA .....	32
2.5 RELACIÓN DE LOS REQUERIMIENTOS .....	32
2.5.1 Requisitos funcionales del sistema.....	32
2.5.2 Requisitos no funcionales del sistema .....	33
2.6 FASE DE EXPLORACIÓN .....	35
2.6.1 Historias de Usuario .....	35
2.7 FASE DE PLANIFICACIÓN .....	37
2.7.1 Plan de iteraciones.....	37
2.7.2. Plan de duración de las iteraciones .....	38
2.7.3. Plan de entregas.....	39
2.8 CONCLUSIONES .....	40
<b>DISEÑO, IMPLEMENTACIÓN Y PRUEBAS.....</b>	<b>41</b>
3.1 INTRODUCCIÓN .....	41
3.2 FASE DE DISEÑO DEL SISTEMA .....	41
3.3 FASE DE IMPLEMENTACIÓN DEL SISTEMA.....	43

3.3.1 Interacciones.....	44
3.3.2 Iteración 1 .....	44
3.3.3 Iteración 2 .....	47
3.3.4 Iteración 3 .....	50
3.4 FASE DE PRUEBAS .....	53
3.4.1 Pruebas Unitarias.....	53
3.4.1.1 Pruebas de Unidad para el módulo Construir Grafo .....	53
3.4.1.2 Pruebas de Unidad para el módulo Ruta Mínima.....	56
3.4.2 Pruebas de Aceptación.....	59
3.4.2.1 Análisis estadístico para la encuesta.....	69
3.5 CONCLUSIONES .....	73
<b>ESTUDIOS DE FACTIBILIDAD.....</b>	<b>74</b>
4.1 INTRODUCCIÓN .....	74
4.2 COCOMO II.....	74
4.3 CARACTERÍSTICAS DEL PROYECTO .....	75
4.3.1 Entradas externas.....	75
4.3.2 Salidas externas .....	76
4.3.3 Consultas Externas.....	76
4.3.4 Archivos Lógicos Internos .....	77
4.3.5 Archivos de Interfaz Externos .....	77
4.3.6 Puntos de función desajustados .....	78
4.4 CÁLCULO DE INSTRUCCIONES FUENTES, ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO .....	79
4.4.1 Cálculo del esfuerzo nominal.....	79
4.4.2 Cálculo del esfuerzo ajustado .....	81
4.5 RESULTADOS .....	82
4.5.1. Análisis de costos y beneficios .....	82
4.6 CONCLUSIONES .....	83
<b>CONCLUSIONES GENERALES .....</b>	<b>84</b>
<b>RECOMENDACIONES .....</b>	<b>85</b>
<b>BIBLIOGRAFÍA.....</b>	<b>86</b>
<b>ANEXOS .....</b>	<b>89</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>93</b>

# INTRODUCCIÓN

En el decursar de la historia de la humanidad uno de los principales retos que se han evidenciado es la necesidad de minimizar el gasto de tiempo y de recursos.

Los modelos matemáticos y las técnicas de programación matemática nacieron para dar respuesta a la necesidad de mejorar los procesos productivos y se han venido aplicando mayoritariamente a la organización y distribución de los recursos físicos.

No obstante, hace varias décadas, se ha podido comprobar los excelentes resultados que estas técnicas aportan cuando son utilizadas para mejorar la eficiencia de los recursos humanos, en particular, dicha aplicación es interesante en lugares donde el potencial humano constituye el factor más importante y por lo tanto el coste más relevante.

La investigación de operaciones, una de las ramas de la matemática aplicada, se ha visto inmersa en la búsqueda de soluciones para lograr este objetivo, mediante el uso de modelos matemáticos, estadísticas y algoritmos. Frecuentemente lleva a cabo el estudio de complejos sistemas reales, con la finalidad de optimizar o mejorar el funcionamiento del mismo. Permitiendo el análisis en la obtención de datos para apoyar la toma de decisiones.

Entre las distintas aplicaciones se encuentra la asignación de recursos, donde su característica esencial es la limitante o escases de los mismos, como: materias primas, presupuesto, horas-hombre, etc. y la asignación de estos recursos limitados de manera óptima, minimizando costos, tiempo, desperdicios, maximizando las ganancias, la producción etc. Aunque en la práctica pueda o no aplicarse la solución óptima propuesta por la aplicación o el método usado; por otras políticas del tomador de decisiones o intereses de la organización donde se realice el estudio, pretendiendo lograr un trabajo más eficiente.

El avance de las tecnologías ha permitido que muchos de los problemas tratados por la Investigación de Operaciones tiempo atrás, que traían consigo un gasto adicional de recursos, pudiendo no ser del todo



## Introducción

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

factibles los resultados obtenidos; sean ahora, simulados con diferentes aplicaciones para determinar la solución, sin un costo adicional.

En la Universidad de las Ciencias Informáticas (UCI) se ha experimentado un crecimiento acelerado en sus áreas provocado por un notable incremento del personal profesional y estudiantil que habita dentro de la misma, esto determina el aumento de la complejidad de muchos de los procesos para la logística y funcionamiento, constituyendo una problemática trasladarse de un lugar a otro en el menor tiempo posible.

A consecuencia del crecimiento que ha experimentado la universidad desde sus inicios y a la sucesiva expansión de sus áreas es casi imposible que al hacer un recorrido evitemos pensar en: ¿Cuál será la trayectoria más corta? ¿Por qué lugar puedo llegar más rápido? ¿Cuál es la ruta que ahorre más combustible? Estas interrogantes han llevado a pensar en un modo para darle solución a estas preguntas que diariamente se hacen los peatones y choferes de la UCI. A pesar de que en el centro se han automatizado o informatizado muchos procesos no existe un sistema informático de apoyo a la toma de decisiones que ayude a responder estas interrogantes.

Como resultado de la situación actual, los peatones y autos pueden estar desperdiciando tiempo innecesario o combustible. El monto total de este gasto o el impacto que está provocando no se ha cuantificado en un estudio, sin embargo el mismo pudiera ser considerable y una buena fuente de ahorro si no se incurriera en ellos.

Por ejemplo, si un estudiante al dirigirse al laboratorio de su proyecto, no elige el camino más corto, puede provocar que se tarde 1 minuto o varios de más, en llegar a su destino. Un proyecto productivo está constituido por cierta cantidad de estudiantes, si se toma como promedio que tiene 30 estudiantes, entonces, en un solo día se estará perdiendo 30 minutos de trabajo. Si de un mes se trabajan 24 días en las labores productivas, con estas condiciones se estará dejando de aprovechar 720 minutos, lo cual puede afectar el rendimiento del proyecto o el cumplimiento de la fecha de entrega de las tareas planificadas.

## Introducción

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

Caso similar sucede con los vehículos de automotor existentes en la universidad, que diariamente transitan por sus áreas sin tener en cuenta el ahorro de combustible que podría ocasionar el seguir por una trayectoria más óptima. Por ejemplo un automóvil realiza un recorrido que le lleve a consumir 0.1 Litros de combustible, más de lo que pudiera consumir realizando una trayectoria óptima; si en un día realiza esta ruta al menos 3 veces estará gastando 0.3 Litros. En un mes laboral donde se realice esta única trayectoria durante 24 días, estará consumiendo 7.2 Litros de más. La UCI cuenta con un amplio parque automotor, si se toma un promedio de 50 autos, entonces estarán realizando un gasto considerable de 360 Litros de combustible al mes.

Los ejemplos anteriores evidencian el gasto y derroche innecesario de los recursos con los que cuenta la universidad. Por lo que en la forma actual en que se desarrollan pueden no estar a tono con la política de ahorro de nuestro país y nuestra universidad.

Por todo lo expuesto anteriormente el **problema a resolver** quedaría definido en la siguiente interrogante: ¿Cómo brindar trayectorias óptimas a seguir por automóviles y personal en la Universidad de las Ciencias Informáticas?

Teniendo como **objeto de estudio** los algoritmos y métodos para la optimización de trayectorias.

El **campo de acción** está enfocado en los procesos de cálculo de trayectorias óptimas para la circulación de automóviles y personal dentro de la Universidad de las Ciencias Informáticas.

El **Objetivo General** de este trabajo consiste en: Desarrollar una aplicación que haga uso de algoritmos matemáticos para el cálculo de rutas óptimas para la circulación de automóviles y personal en la UCI.

### **Objetivos Específicos:**

- Investigar y analizar el estado del arte y principales conceptos sobre los algoritmos matemáticos para optimizar problemas de redes.

## Introducción

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

- Determinar el camino más corto de un punto a otro, trazando un recorrido óptimo para los peatones y autos.
- Mostrar gráficamente sobre un mapa de la UCI el recorrido calculado.

La **Idea a defender** es: Con una aplicación que brinde trayectorias óptimas para la circulación de automóviles y personal, se ahorrará tiempo y combustible, contribuyendo al ahorro en la UCI.

Para dar cumplimiento a los objetivos específicos de la investigación se trazaron una serie de **Tareas**:

- **Investigar sobre el estado del arte.**

1. Revisión bibliográfica del tema.
2. Existencia de otras aplicaciones o soluciones similares.

- **Elaborar el diseño teórico de la investigación.**

1. Definición de la situación problemática, problema, objetivos, novedad científica, aportes teóricos y prácticos.

- **Elaborar la propuesta de solución.**

1. Análisis y diseño de la aplicación.
2. Definición de la arquitectura de software a usar.
3. Definición de la plataforma de desarrollo de soporte para la aplicación.
4. Elaboración de un prototipo funcional.
5. Elaborar y aplicar una estrategia de prueba para el prototipo funcional.
6. Documentar el proceso de desarrollo y prueba.

- **Estructurar el documento tesis**

1. Definición de los Capítulos, Epígrafes y Anexos.

Luego de la realización de las tareas antes mencionadas se espera tener estos **posibles resultados**:

- Una funcionalidad que permita a cualquier usuario, dado el mapa real de la universidad, determinar cuál es el mejor camino a tomar, partiendo desde un punto A a un punto B de llegada. Esta funcionalidad podrá aplicarse para los automóviles y peatones de la UCI.
- Una funcionalidad que permita la adaptabilidad del sistema a nuevas condiciones en la infraestructura de la UCI.
- Información persistente en Bases de Datos de la redes de caminos y edificaciones de la universidad, estando disponible para la solución a nuevos planteamientos de esta índole.
- Un material de apoyo para la asignatura de Investigación de Operaciones, permitiendo mostrar de manera práctica la utilización de los algoritmos matemáticos impartidos en clases.

Para el desarrollo de las tareas de la investigación se tienen en cuenta algunos métodos tradicionales investigativos. A continuación se mencionarán cada uno de ellos y de qué forma se ponen de manifiesto en la investigación.

**Análítico-Sintético.** Se utiliza este método centrándose en el análisis de las teorías, documentos, permitiendo la selección de los elementos más importantes de manera que se elabore correctamente la información.

**Análisis Histórico-Lógico.** Este método se utiliza con el objetivo de constatar teóricamente la evolución de la programación Web, de los algoritmos y problemas matemáticos de optimación y cubrimiento.

**Inductivo-Deductivo.** Se utiliza este método con el objetivo de llegar a conocimientos generalizadores que permitan determinar cual es el algoritmo que permitirá obtener resultados eficientes.

**Métodos Empíricos (Matemáticos y Medición):** Este método se utiliza en la observación, cálculos, mediciones y en la investigación relacionada con este trabajo de diploma y además para la selección de los programas necesarios para el desarrollo de la aplicación Web.

# CAPÍTULO 1

## FUNDAMENTACIÓN TEÓRICA

### **1.1 Introducción**

Este capítulo muestra una visión de los aspectos relacionados con la disciplina de Investigación de Operaciones y los métodos de solución a problemas de optimización de recursos, más específicamente en los modelos y algoritmos de solución de problemas de redes, haciendo énfasis en los de camino extremal.

Se definen los principales conceptos para comprender mejor la utilización de herramientas y metodologías empleadas para darle solución al problema planteado.

### **1.2 La Investigación de Operaciones**

La disciplina de investigación de operaciones permite a los profesionales del área de sistemas, el estudio y planteamiento de soluciones a problemas complejos con ayuda de herramientas de software, que en la práctica no serían posibles de resolver de manera adecuada, en términos de exactitud, tiempo y costo. Por eso es la importancia de la aplicación de estos conocimientos en las áreas de los servicios.

Churchman, Ackoff y Arnoff definen a la investigación de operaciones como la aplicación, por grupos interdisciplinarios, del método científico a problemas relacionados con el control de las organizaciones o sistemas (hombre-máquina), a fin de que se produzcan soluciones que sirvan mejor a los objetivos de la organización.

La actitud diferencial de la disciplina consiste en desarrollar un modelo científico del sistema tal que incorpore valoraciones de factores como el azar y el riesgo, mediante el cual se predigan y comparen los resultados de decisiones, estrategias o controles alternativos. El propósito es el de ayudar a la gerencia a determinar científicamente sus políticas y acciones.

## Capítulo 1: Fundamentación Teórica.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

Esta tuvo sus orígenes en la Segunda Guerra Mundial. La necesidad de abastecer los ejércitos necesitó la participación de grupos interdisciplinarios para resolver problemas que sólo eran competencia de militares.

El empleo se extendió hacia el campo Administrativo e Ingeniería y hoy se considera que la Investigación de Operaciones es la utilización científica de un enfoque planeado de grupos interdisciplinarios con el fin de representar las relaciones funcionales como modelos matemáticos para suministrar información cuantitativa en la toma de decisiones y búsqueda de nuevas aplicaciones.

A continuación se hace referencia a una tabla que muestra algunos casos reales de organizaciones que han hecho uso de la Investigación Operativa y las ganancias y/o ahorros conseguidos a raíz de ello (Ver Anexo 1).

Algunos de los modelos para la representación de los problemas que aplica son:

- Modelo de programación.
- Modelo de administración.
- Modelos Físicos, Analógicos, Simbólicos.
- Modelos de decisión.
- Modelos de asignación, transporte.
- Modelos de Simulación.

Estructuras del modelo matemático que se aplican son:

- Variables de decisión.
- Parámetros y constantes.

## Capítulo 1: Fundamentación Teórica.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

- Función de optimización: Máx. / Min.
- Ecuaciones de restricción.

Elementos que componen el modelo matemático:

- **Variables de decisión:** Representan las actividades que compiten por los recursos.
- **Coefficiente económico:** Representa el objetivo a optimizar.
- **Función objetivo:** Función lineal que representa en la variable Z el valor óptimo. El que puede ser maximizar o minimizar.
- **Restricciones:** Relaciones lógico matemáticas lineales que representan limitaciones que se oponen al objetivo que se quiere alcanzar.

### DEFINICIÓN DE VARIABLES Y PARÁMETROS

n decisiones → Variables de decisión  $x_1, x_2, \dots, x_n$

Medida del desempeño conjunto → Función Objetivo (F. O.)

Ejemplo:  $Z = F(x_1, x_2, \dots, x_n) = 5x_1 + 7x_2 + \dots + 20x_n$ .

Conjunto de limitaciones → Restricciones: Como ecuaciones y desigualdades.

Ejemplo:  $5x_1 + 7x_4 \leq 10$

Coefficientes y los lados derechos → Parámetros

Algunos de los modelos matemáticos propuestos a darle solución en este trabajo son los relacionados con los problemas transporte y redes lineales.

### **1.2.1 Problemas de optimización**

Existen diferentes algoritmos para resolver los problemas de camino extremal entre ellos se encuentra el Simplex. El método Simplex es un procedimiento iterativo que permite ir mejorando la solución inicial en cada paso. El proceso concluye cuando no es posible seguir mejorando más dicha solución.

Partiendo del valor de la función objetivo en un vértice cualquiera, el método consiste en buscar sucesivamente otro vértice que mejore al anterior. La búsqueda se hace siempre a través de los lados del polígono (o de las aristas del poliedro, si el número de variables es mayor) que es donde se encuentra el rango de las soluciones. Como el número de vértices (y de aristas) es finito, siempre se podrá encontrar la solución.

El método Simplex se basa en la siguiente propiedad: si la función objetivo,  $F ( )$ , no toma su valor máximo en el vértice A, entonces hay una arista que parte de A, a lo largo de la cual  $F ( )$  aumenta.

Deberá tenerse en cuenta que este método sólo trabaja para restricciones que tengan un tipo de desigualdad " $\leq$ " y coeficientes independientes mayores o iguales a 0, y habrá que estandarizar las mismas para el algoritmo. En caso de que después de éste proceso, aparezcan (o no varíen) restricciones del tipo " $\geq$ " o " $=$ " habrá que emplear otros métodos que son variantes o extensiones del método Simplex.

Ejemplo de solución de un problema por el método Simplex, ver Anexo 2

El problema de la ruta más corta se puede resolver utilizando programación lineal sin embargo, debido a que el método Simplex es de complejidad exponencial, se prefiere utilizar algoritmos que aprovechen la estructura en red que se tiene para estos problemas, como por ejemplo el algoritmo de Dijkstra o el de Bellman-Ford.

Una red de comunicaciones involucra un conjunto de nodos conectados mediante arcos, que transfiere datos, vehículos, etc. desde determinados nodos origen a otros nodos destino. La forma más común para



seleccionar la trayectoria (o ruta) de dichos vehículos, se basa en la formulación de la ruta más corta. En particular a cada arco se le asigna un escalar positivo el cual se puede ver como su longitud.

Un algoritmo de trayectoria más corta, rutea cada vehículo a lo largo de la trayectoria de longitud mínima (ruta más corta) entre los nodos origen y destino. Hay varias formas posibles de seleccionar la longitud de los enlaces.

La forma más simple es que cada enlace tenga una longitud unitaria, en cuyo caso, la trayectoria más corta es simplemente una trayectoria con el menor número de enlaces. De una manera más general, la longitud de un enlace puede depender de su capacidad de transmisión.

Otra forma para determinar esta trayectoria es asignar al enlace un peso, que sería el valor de la distancia entre los nodos conectados, en cuyo caso, el costo de una trayectoria sería la suma de los costos de las vías (aristas) que la componen. La trayectoria más corta sería la representada por aquella cuyo costo es el mínimo.

El algoritmo que se estudiará es el Algoritmo de Dijkstra para solucionar el problema de los caminos más cortos de un vértice, que denominamos el origen, a los restantes vértices del grafo. Este algoritmo es un algoritmo de tipo greedy. También conocidos como algoritmos ávidos, glotones o voraces.

Estos algoritmos trabajan por etapas, tomando en cada una de ellas la solución mejor (óptimo local) sin considerar las consecuencias futuras. El óptimo encontrado en una etapa puede posteriormente ser modificado si surge una solución mejor.

A continuación se especifica Algoritmo de Dijkstra en pseudo-código:

### **Inicialización**

- Sea  $V$  un conjunto de vértices de un grafo.
- Sea  $C$  una matriz de costos de las aristas del grafo, donde en  $C [u, v]$  se almacena el costo de la arista entre  $u$  y  $v$ .

## Capítulo 1: Fundamentación Teórica.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

- Sea  $S$  un conjunto que contendrá los vértices para los cuales ya se tiene determinado el camino mínimo.
- Sea  $D$  un arreglo unidimensional tal que  $D[v]$  es el costo del camino mínimo del vértice origen al vértice  $v$ .
- Sea  $P$  un arreglo unidimensional tal que  $P[v]$  es el vértice predecesor de  $v$  en el camino mínimo que se tiene construido.
- Sea  $V$  inicial el vértice origen. Recordar que el Algoritmo Dijkstra determina los caminos mínimos que existen partiendo de un vértice origen al resto de los vértices.

**Paso 1.**  $S \leftarrow \{V_{inicial}\}$       *//Inicialmente  $S$  contendrá el vértice origen*

**Paso 2.** Para cada  $v \in V, v \neq V_{inicial}$ , **hacer**

**2.1.**  $D[v] \leftarrow C[V_{inicial}, v]$       *//Inicialmente el costo del camino mínimo de  $V_{inicial}$  a  $v$  es lo contenido en la matriz de costos*

**2.2.**  $P[v] \leftarrow V_{inicial}$       *//Inicialmente, el predecesor de  $v$  en el camino mínimo construido hasta el momento es  $V_{inicial}$ .*

**Paso 3.** Mientras  $(V - S \neq \emptyset)$  **hacer** *//Mientras existan vértices para los cuales no se ha determinado el camino mínimo.*

**3.1.** Elegir un vértice  $w \in (V - S)$  tal que  $D[w]$  sea el mínimo.

**3.2.**  $S \leftarrow S \cup \{w\}$       *//Se agrega  $w$  al conjunto  $S$ , pues ya se //tiene el camino mínimo hacia  $w$*

**3.3.** Para cada  $v \in (V - S)$  **hacer**

**3.3.1.**  $D[v] \leftarrow \min (D[v], D[w] + C [w, v])$  //Se escoge, entre el camino mínimo hacia  $v$  que se tiene hasta el momento, y el camino hacia  $v$  pasando por  $w$  mediante su camino mínimo, el de menor costo.

**3.3.2.** Si  $\min (D[v], D[w] + C [w, v]) = D[w] + C [w, v]$  entonces  $P[v] \leftarrow w$   
//Si se escoge ir por  $w$  entonces el predecesor de  $v$  por el momento es  $w$ .

#### **Paso 4. Fin**

### **1.2.2 Problemas de optimización en la universidad**

En la UCI existen proyectos que trabajan con gráficos, los cuales para desarrollar el producto final, realizan un gran análisis matemático y lo llevan a la práctica. Estos análisis utilizan algoritmos matemáticos para la optimización de los recursos de la máquina, desde los más sencillos hasta los más complejos para poder cargar la mayor cantidad de gráficos posibles sin afectar el rendimiento la computadora.

Estos proyectos por lo general pertenecen a la facultad 5, debido a que ese es el perfil que desarrolla. Ya la universidad ha obtenido productos que se han basado en gran parte en los algoritmos matemáticos de optimización, como es el ejemplo del simulador SIMPRO. Pero todos estos proyectos solo se han desarrollado con aplicaciones de escritorio, y enfocados principalmente al ahorro de los recursos de una estación de trabajo, pero hasta ahora no se ha creado una aplicación que contribuya a la optimización de recursos y esfuerzos de los usuarios que interactúen con el sistema.

### **1.3 Metodologías de desarrollo propuesta**

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se aplica una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de

proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo. (Pressman, 1999)

Cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí si toma sentido en basar su desarrollo en una metodología, y empezar a buscar cual sería la más apropiada para nuestro caso. Lo cierto es que muchas veces no se encuentra la más adecuada y se termina por hacer o diseñar la propia metodología, algo que por supuesto no esta mal, siempre y cuando cumpla con el objetivo (JACOBSON 1992).

Muchas veces se realiza el diseño del software de manera rígida, con los requerimientos que el cliente solicitó, de tal manera que cuando el cliente en la etapa final (etapa de prueba), solicita un cambio se hace muy difícil realizarlo, pues de hacerlo, se alteran muchas cosas que no se había previsto, y es justo éste, uno de los factores que ocasiona un atraso en el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido. Obviamente para evitar estos incidentes se debe haber llegado a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique al desarrollo del mismo.

Por experiencia, muchas veces los usuarios finales, se dan cuenta de las cosas que dejaron de mencionar, recién en la etapa final del proyecto, pese a que se les mostró un prototipo del software en la etapa inicial del proyecto. Los proyectos en problemas son los que salen del presupuesto, tienen importantes retrasos, o simplemente no cumplen con las expectativas del cliente.

Para dar una idea qué metodología se utilizará y cual se adapta más al proyecto, se hará mención a tres de las que se consideran las más importantes, tal como: RUP, XP y Scrum.

### **1.3.1 Programación Extrema**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizada para proyectos de corto plazo, corto de equipo y cuyo plazo de entrega era ayer. La metodología consiste en

## Capítulo 1: Fundamentación Teórica.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP, la metodología se basa en:

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- Re fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantivo del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierten en miembro del equipo.

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

(Sanchez, 2004)

### **1.3.2 Metodología de desarrollo Scrum**

Scrum es un proceso ágil para desarrollar software que fue aplicado por primera vez por Ken Schwaber y Jeff Sutherland, quienes lo documentaron en detalle en el libro *Agile Software Development with Scrum*. Esta metodología centra su atención en las actividades de Gerencia y no especifica prácticas de Ingeniería. Fomenta el surgimiento de equipos auto dirigidos, cooperativos y aplica inspecciones frecuentes como mecanismo de control.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Scrum surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. El mismo adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

En Scrum se realizan entregas parciales del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales.

El ciclo de vida definido por Scrum es incremental iterativo y se caracteriza por ser muy adaptable.

### **Principales características**

- Equipos auto dirigidos.
- Utiliza reglas para crear un entorno ágil de administración de proyectos.
- No prescribe prácticas específicas de ingeniería.
- Los requerimientos se capturan como ítems de la lista Product Backlog.
- El producto se construye en una serie de Sprints de un mes de duración.

### **Principales elementos de la metodología**

- Herramientas
- Product Backlog (lista de todos los cambios requeridos sobre un producto.)
- Sprint Backlog( Es el punto de entrada de cada Sprint)
- Prácticas
- Sprints(Procedimiento de adaptación de las cambiantes variables del entorno)
- Sprint Planning Meeting( Reunión organizada por el Scrum Master)
- Daily Meetings(Reuniones diarias)
- Sprint Review Meeting (Reunión informal que tiene como regla que su preparación no puede tomar más de 2 horas.)
- Design Review Meeting(Se comunica el diseño a los interesados para revisar el cumplimiento de los ítems especificados en el Product Backlog)
- Stabilization Sprints( En estos Sprints el equipo se concentra en encontrar defectos, no en agregar funcionalidad)
- Meta Scrums(Grupos de trabajos)
- Roles y responsabilidades
- Scrum Master(Rol de administración)
- Product Owner(Responsable del proyecto, administra, controla y comunica la Backlog List)

- Scrum Team (Equipo del proyecto que tiene la autoridad para decidir como organizarse para cumplir con los objetivos de un Sprint.)
- Customer (El cliente participa en las tareas que involucran la lista Product Backlog.)
- Management (Responsable de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el proyecto)

### 1.3.3 Proceso Racional Unificado y El Lenguaje Unificado de Modelado

RUP o Proceso Racional Unificado (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, forman la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El Proceso Unificado de Desarrollo (RUP) describe como aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (Jacobson, 2000)

RUP no es una metodología con pasos firmes y establecidos, sino un conjunto de metodologías adaptables al contexto o al proyecto que se quiera desarrollar, pues aunque se recomienda para proyectos grandes, este se puede adaptar a proyectos pequeños, sobre todos a estos donde sea necesario almacenar toda la bibliografía y artefactos generados durante el proceso de desarrollo.

Está basado en componentes. Utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP, no obstante, los verdaderos aspectos definitorios del proceso unificado se resumen en tres características claves: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

RUP provee a cada miembro del equipo de las guías de proceso, plantillas y mentores de herramientas necesarios para que el equipo completo. Divide el proceso de desarrollo en 4 fases el desarrollo del software, la fase de Inicio o Conceptualización, cuyo objetivo es determinar la visión del proyecto. La fase



de Elaboración, en esta etapa el objetivo es determinar la arquitectura óptima. La siguiente fase es Construcción, donde el objetivo es llegar a obtener la capacidad operacional del software y por último la fase de Transición, el objetivo de la misma es llegar a obtener versiones ejecutables del producto. Cada una de estas fases es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. El proceso de desarrollo de software se encuentra dividido en disciplinas o Flujos de trabajos, entre los que se encuentran: Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue, y las de soporte Gestión de Configuración, Ambiente y Administración de Proyecto.

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. (Larman, 2002) Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios. (Larman, 2002)

El modelo gráfico de UML tiene un vocabulario en el que se identifican los elementos que son las abstracciones que constituyen los bloques básicos de construcción; las relaciones, que son las que unen los elementos y los diagramas, que constituyen la representación gráfica de un conjunto de elementos, visualizando un sistema desde diferentes perspectivas.

### **1.3.4 Justificación de la propuesta para la Metodología Programación Extrema**

Para lograr un proyecto con calidad se debe seleccionar la metodología que más se ajuste al sistema a desarrollar, hoy en día este proceso es complejo y difícil ya que existe un amplio grupo de metodologías, porque a las tradicionales que existían se les unen un grupo de metodologías ágiles las cuales son muy apropiadas para desarrollar un software con calidad.

Para la realización de este sistema se cuenta con muy poco tiempo, y con un equipo de desarrollo reducido. Por este motivo no es conveniente utilizar una metodología de desarrollo pesada como RUP, pues solo en la generación de toda la documentación y artefactos que propone esta metodología se consumiría todo el tiempo disponible.

Para la realización de este sistema sería conveniente utilizar una metodología ágil, con la que se pueda lograr un producto eficiente y en el tiempo requerido, atendiendo a las condiciones específicas de este trabajo. Dentro de este grupo se pueden definir dos como las más utilizadas por múltiples desarrolladores del mundo: Programación Extrema (XP) y Scrum.

Los métodos Ágiles, tales como XP son estrategias de desarrollo de software que promueven prácticas que son adaptativas en vez de predictivas, centradas en el usuario o en los equipos, iterativas, orientadas hacia prestaciones y hacia la entrega, de comunicación intensiva, y que requieren que el negocio se involucre en forma directa.

Scrum es una metodología ágil para gestionar proyectos de software, es de desarrollo muy simple, que requiere trabajo duro, porque la gestión no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. Scrum contiene un modo de desarrollo de carácter adaptable, orientado a las personas antes que a los procesos y emplea desarrollo ágil: iterativo e incremental.

XP tiene su foco principal en el código producido, independientemente del aspecto tiempo. En el comienzo de un proyecto, la atención se centra en el producto básico, y más tarde en las características. Además da la posibilidad de que con pocos programadores se puede desarrollar software con requerimientos cambiantes y de alto riesgo.

Por lo tanto para el desarrollo del sistema se utilizará XP que es más centrada en las prácticas de programación, y se adapta perfectamente a las características de este proyecto.

## **1.4 Tecnologías propuestas**

En el desarrollo de la aplicación se utilizarán un grupo de herramientas y tecnologías, junto con las cuales se asegurará de la calidad y entrega en tiempo del producto.

### **1.4.1 Lenguaje HTML**

Desde el surgimiento de Internet se han publicado sitios Web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios Web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales).

#### **Ventajas:**

- Sencillo, que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas Web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

#### **Desventajas:**

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

### **1.4.2 Lenguaje JavaScript**

Es un lenguaje interpretado, no requiere compilación, utilizado principalmente en páginas Web. La mayoría de los navegadores en sus últimas versiones interpretan el código JavaScript.

El código JavaScript puede ser integrado dentro de las páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (Modelo de Objetos del Documento).

**Ventajas:**

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.

**Desventajas:**

- Código visible por cualquier usuario.
- El código debe descargarse completamente.

### **1.4.3 Lenguaje PHP**

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor. PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo una base de datos. El código PHP es ejecutado en el servidor y el resultado es enviado al navegador.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Las principales características de PHP son: rapidez; facilidad de aprendizaje; soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta. PHP es el líder mundial de la programación Web de forma dinámica. (linuxcentro.net, 2007)

Existen otras tecnologías para el desarrollo de página Web las cuales se podían utilizar en algún momento como son JSP, ASP, ASP.NET. En el caso de JavaServer Pages (JSP) es una tecnología Java la cual permite desarrollar páginas Web dinámicas, pero tiene como inconveniente el bajo enfoque de cara al usuario donde da mucho trabajo para presentar páginas Web al navegador, así como su excesiva complejidad. Active Server Pages (ASP) se han desechado rápidamente ya que es una tecnología propietaria que al igual que ASP.NET es comercializada por Microsoft. Por estas características y por lo antes explicados se utilizará como lenguaje de programación PHP.

### **1.4.4 Servidor Web Apache**

Apache es el servidor Web hecho por excelencia, su fácil configuración, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Se mencionarán una serie de ventajas y características que demuestran porque la utilización de este servidor:

- Corre en una multitud de Sistemas Operativos (es multiplataforma), lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierta. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Dándole una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este.
- Apache trabaja con gran cantidad de lenguajes como: Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para crear páginas dinámicas.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad (Runeson & Greberg, 2005) en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

(Ciberaula, 2006)

Otro servidor que se utiliza bastante, como servidor Web es IIS (Internet Information Server), pero el hecho de que es propietario y solo trabaja para sistemas operativos Microsoft Windows, no es la herramienta indicada.

## **1.5 Herramientas Propuestas**

### **1.5.1 Gestor de Bases de Datos MySQL**

MySQL es un sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la Licencia Publica General (en inglés GPL, General Public License).

Es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL, esto significa que también todos pueden contribuir con ideas, elementos, mejoras o sugerir optimizaciones. Y así es que este ha pasado de ser una pequeña base de datos a una completa herramienta. Su rápido desarrollo se debe en gran medida a la contribución de mucha gente al proyecto, así como la dedicación de su equipo. MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. Utiliza el lenguaje de consulta estructurado (SQL). Es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos.

Antes se consideraba como la opción ideal de sitios Web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. Es una base de datos robusta que se la puede comparar con una base de datos comercial, es incluso más veloz en el procesamiento de las transacciones y dispone de un sistema de permisos elegante y potente, y ahora, además, además dispone de procedimientos de almacenado, triggers y vistas. Es rápido, y una solución accesible para administrar

correctamente los datos de una empresa, y, como ocurre con la mayor parte de las comunidades de código abierto, se puede encontrar una gran cantidad de ayuda en la Web. Son muchas las razones para escoger a MySQL como una solución de misión crítica para la administración de datos:

- Costo: Es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- Velocidad: Es mucho más rápido que la mayoría de sus rivales.
- Funcionalidad: Dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI21, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- Portabilidad: Se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- Facilidad de uso: Resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

### 1.5.2 Gestor de Bases de Datos PostgreSQL

PostgreSQL está considerado como uno de los gestores de bases de datos de código abierto más avanzado. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como Oracle (es básicamente una herramienta cliente/servidor para la gestión de bases de datos. Se considera como uno de los sistemas de bases de datos más completos destacando su soporte de transacciones, estabilidad, escalabilidad, además es multiplataforma). PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, optimización de consultas y herencia. PostgreSQL tiene soporte para lenguajes procedurales internos (son aquellos en los cuales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos con el fin de calcular el resultado deseado y están fundamentados en la utilización de variables para almacenar valores y en la realización de operaciones con los datos almacenados), incluyendo un lenguaje nativo



denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. (Quiñones, 2007)

### **Las características principales de PostgreSQL son:**

- Atomicidad (Indivisible) es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- Consistencia es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- Aislamiento es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- Durabilidad es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.
- Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc.
- Soporte de todas las características de una base de datos profesional (triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.)
- Soporte de protocolo de comunicación encriptado por SSL
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales, minería de datos, etc.

Otros gestores de base de datos existentes son: Microsoft SQL Server y Oracle los cuales se podían usar ya que son estables en el desarrollo de las aplicaciones. Pero al ser propietario, deja de ser una vía al desarrollo, aunque se puede señalar de Oracle que no es que tan ligero como MySQL y PostgreSQL consume más recursos y no nos brinda tantas facilidades a la hora de trabajarlo con PHP como estos.

### 1.5.3 MySQL, gestor de base de datos definido

Después de haberse realizado un estudio de los gestores de base de datos más usados en el mundo, que son libres o están bajo licencia GNU/GPL, se ha decidido usar MySQL debido a que se ajusta más a los requerimientos de la aplicación, como agilidad y rapidez en la ejecución de consultas, siendo la característica esencial para una ejecución eficiente del sistema y el gestor por excelencia de las aplicaciones Web. A diferencia del gestor PostgreSQL que es mucho más lento y consume muchos recursos del servidor en la transferencia y manejo de los datos.

### 1.5.4 Zend Studio

Es un editor Web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. Este editor es multiplataforma y su idioma para trabajar es inglés.

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.(Desarrolladores Web, 2003).

Zend Studio no tiene muchos programas que le hagan competencia ya que es líder en el desarrollo de aplicaciones Web dinámicas en PHP. Un programa que es muy similar es el Adobe DreamWeaver pero este es más bien para diseño, teniendo en cuenta que es propietario y no esta disponible para GNU/Linux,

## Capítulo 1: Fundamentación Teórica.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

además tiene un completamiento de script de PHP muy pobre comparado con el del Zend Studio. Otro programa que trabaja con PHP es el Aptana que se especializa en Java Script.

### 1.6 Conclusiones

Durante el desarrollo de este capítulo se ha fundamentado las bases, herramientas y metodologías de soporte para la creación de un sistema web que de solución a los objetivos planteados en este trabajo de diploma.

# CAPÍTULO 2

## CARACTERÍSTICAS DEL SISTEMA

### 2.1 Introducción

El presente capítulo tiene como objetivo hacer una valoración de las características principales del sistema a desarrollar, prestando gran atención a la situación problemática que dio origen al mismo. Se detallan las necesidades de los usuarios, describiéndose las funcionalidades que serán objeto de automatización. Se presentará una propuesta del software a implementar, especificando detalladamente los requerimientos funcionales y no funcionales. Por último se llevarán a cabo las fases de exploración y planificación propuestas por la metodología de desarrollo a utilizar para la realización del sistema. Mostrando además las historias de los usuarios, así como una estimación del esfuerzo por cada una.

### 2.2 Descripción

Cuando las personas en la universidad se disponen a trasladarse de un lugar a otro, realizan el recorrido que individualmente consideran que es el más corto, lo que puede ser cierto o no. Destacando la necesidad de implementar una herramienta que permita determinar el camino mínimo de un punto a otro, logrando así grandes ahorros de esfuerzos y recursos, dando por resuelto la problemática existente al inicio del desarrollo de este trabajo.

Veamos el siguiente ejemplo donde se pone en práctica los elementos referidos:

Si una persona que trabaja en un proyecto pierde 0,5 minutos en ir de su edificio a su proyecto por no utilizar la ruta mas corta en este solo viaje no parece nada.

Sin embargo si hace 4 veces en el día el mismo recorrido perdió 2 min, si esto lo repite 5 días de la semana pierde 10 min, y al mes perdió 40 min. ¿Aun le parece poco?

## Capítulo 2: Características del Sistema.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

Bueno este mismo compañero no está solo en su proyecto, supongamos que son 30 integrantes del proyecto, si los 30 perdieran 0,5 min en un solo viaje, al mes estarían perdiendo 1200 minutos. Saque usted la cuenta de cuanto se perdería en un año en todos los proyectos.

Veamos ahora otro ejemplo donde se pone de manifiesto el ahorro de combustible, recurso que escasea y cada vez es más costoso.

Un automóvil en un recorrido del complejo comedor 3 al docente 6 consume alrededor de 0.123 Litros de combustible para una autonomía de 8 Km/L utilizando la vía más corta.

Supongamos ahora que en su recorrido no utiliza esta vía sino otra que le lleva a consumir 0.20 Litros de combustible para una autonomía de 8 Km/L ósea un ligero aumento de 0,077 Litros. En un día normal deberá realizar este recorrido como mínimo 3 veces suponiendo que fue a desayunar, almorzar y a comer. Teniendo en cuenta que solamente el chofer fue hasta el comedor y luego regresó al docente, el consumo resultante para estos tres viajes de ida y vuelta es de 1.2 Litros.

Si el chofer realiza esta trayectoria en los 5 días de la semana entonces estaría consumiendo 6 Litros, resultado un total de 24 Litros de combustible consumido para un mes, solamente en esta trayectoria.

Sin embargo si el chofer optara por recorrer esta distancia por la trayectoria más corta, el consumo sería de 14.76 Litros para un mes con las mismas características anteriores. Dando como resultado un ahorro de 9.24 Litros de combustible solamente para un auto y considerando que solo hace este único recorrido.

### 2.2.1 Situación Actual

Actualmente el apoyo a la toma de decisiones en el proceso de circulación no es realizada por ningún sistema. Esto provoca que se realice de forma individual, basándose en el conocimiento empírico

adquirido con la experiencia de la vida, el cual, la mayoría de las veces lleva a seguir una trayectoria sin estar seguros de que ésta sea la más corta.

Situación que no está de acorde con las políticas de ahorro del país, ya que las personas están perdiendo un tiempo valioso por no conocer la trayectoria indicada, tiempo que se puede traducir en eficiencia y un mejor desempeño de la persona. Por otra parte está el gasto de combustible que se está generando por no realizar la trayectoria correcta. Siendo uno de los mayores problemas a los que actualmente se está enfrentando el país.

### **2.2.2 Objeto de automatización**

Durante el ciclo de desarrollo de esta actividad varios son los procesos que deben ser automatizados, de lo contrario autos y peatones consumirían tiempos y esfuerzos innecesarios. Por lo que en la forma actual en que se desarrollan no están a tono con la política de ahorro y optimización de recursos de nuestro país.

La automatización del proceso de determinación de la ruta mínima es importante pues se contará con un sistema que realizando cálculos confiables sobre toda una red de caminos, será capaz de retornar una ruta óptima, una vez seleccionado los puntos de inicio y destino.

### **2.3 Características de la propuesta de sistema**

Se propone la implementación de un sistema que sirva de soporte a los usuarios para la toma de decisiones en la obtención de rutas óptimas. Contará con un módulo de administración que permitirá gestionar (crear, modificar, eliminar) los nodos y aristas de la red de caminos del mapa de la universidad.

El sistema será una aplicación Web que por sus características podrá ser integrada con cualquier portal existente en la universidad. Para la fase de pruebas este será integrado con el sistema de gestión de contenidos (CMS) Joomla, el cual permitirá el acceso de la comunidad de usuarios y aprovechando sus características, se podrá evaluar la aceptación de la aplicación.

## **2.4 Personas relacionadas con el sistema**

Las personas relacionadas con el sistema serán aquellas que se relacionen con cualquiera de los procesos que lleva a cabo el sistema y obtienen un resultado de valor con el mismo.

Tabla 2.1 Personas relacionadas con el sistema.

<b>Personas relacionadas con el sistema.</b>	<b>Justificación</b>
Administrador	Es la persona con todos los permisos necesarios para acceder a la información disponible en las bases de datos de la aplicación, para llevar un control sobre la misma y el sistema en general.
Usuario	Es la persona que interactúa con la aplicación Web para hacer consultas.

## **2.5 Relación de los Requerimientos**

### **2.5.1 Requisitos funcionales del sistema**

Los Requerimientos Funcionales serán las acciones que realizará el usuario haciendo uso de la aplicación, así como las acciones que realizara el sistema y que no serán producto de la acción directa de los usuarios. El sistema debe ser capaz de:

#### **R1: Gestión de Nodos**

- R1.1 Mostrar formulario de creación de un nuevo nodo.
- R1.2 Mostrar formulario eliminar nodo existente.
- R1.3 Obtener directamente del navegador las coordenadas para un nodo.
- R1.4 Validar los datos introducidos.
- R1.5 Mostrar tablas con los nodos existentes.

R1.6 Permitir conectar nodos (Crear la arista).

R1.7 Permitir actualizar nodos transitables.

### **R2: Obtención de Camino Mínimo.**

R2.1 Mostrar el formulario de selección de destinos.

R2.2 Validar los datos de la consulta introducidos por el usuario.

R2.3 Mostrar el resultado del cálculo del camino mínimo solicitado por el usuario.

### **2.5.2 Requisitos no funcionales del sistema**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

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

A continuación se exponen características no funcionales del sistema:

#### **Diseño e implementación:**

- Aplicación Web sobre lenguaje de programación PHP 5.2.3.
- Utilización del gestor de Base de Datos MySQL 5.0.45.



## Capítulo 2: Características del Sistema.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

- Utilizar como servidor Web Apache 2.4.

### **Apariencia o interfaz externa:**

- Diseño sencillo y amigable.
- Diseño ajustable a la resolución del monitor.
- Interfaz compatible con los diferentes navegadores Web existentes.

### **Usabilidad:**

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.
- Presenta buena distribución de la información presentada.

### **Funcionalidad:**

- Capacidad de respuesta a la consulta realizada en el menor tiempo posible.

### **Rendimiento:**

- La aplicación debe tener una velocidad de procesamiento óptimo y un buen aprovechamiento de los recursos del servidor.

### **Soporte:**

- El sistema debe ser de fácil instalación, adaptable a numerosas plataformas y de fácil mantenimiento.

### **Portabilidad:**

- Facilidad para adaptarlo a diferentes ambientes sin necesidad de usar otros medios que los previstos. Con facilidad de uso en Windows y Linux como plataforma.

## Exploración y Planificación

### 2.6 Fase de Exploración

La primera de las fases de la metodología Extreme Programming es la de exploración. Durante esta se define el alcance general del proyecto. En esta fase, el cliente precisa lo que necesita mediante la redacción de sencillas “Historias de Usuarios” (UH, del inglés: User Histories), los programadores estiman los tiempos de desarrollo en base a esta información, siendo estas estimaciones primarias, ya que estarán basadas en datos de muy alto nivel. También ocurre la familiarización del equipo de trabajo con las tecnologías y herramientas seleccionadas para la construcción del proyecto.

#### 2.6.1 Historias de Usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del Software, sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas Historias son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar, aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas (Beck, 2000). El contenido de estas debe ser concreto y sencillo (XP, 2006) pero deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas.

Tabla 2.2 Historia de usuario Módulo Construir Grafo.

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Módulo Construir Grafo.
<b>Usuario:</b> Luis Leandro Hernández Fleitas	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 1

## Capítulo 2: Características del Sistema.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

<b>Descripción:</b> El módulo Construir Grafo se utilizará para llenar la base de datos con los nodos y caminos obtenidos a partir de una imagen que representa el mapa del área a tratar. Incluye la creación del ambiente necesario para su ejecución.
<b>Observaciones:</b> Es necesario contar con el mapa previamente marcado con los nodos y aristas a entrar en el sistema. Además debe tener la escala de conversión y un alto nivel de detalles

Tabla 2.3. Historia de usuario Componente Ruta Mínima Peatones.

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre:</b> Componente Ruta Mínima Peatones
<b>Usuario:</b> Luis Leandro Hernández Fleitas	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos Estimados:</b> 4	<b>Iteración Asignada:</b> 2
<b>Descripción:</b> Determina el menor camino que existe entre dos nodos que el usuario selecciona. Incluye la creación del ambiente necesario para su ejecución.	
<b>Observaciones:</b> Es necesario contar con el mapa sobre el cual se mostrarán las rutas con un alto nivel de detalle. El resultado a mostrar depende de que componente esté en ejecución.	

Tabla 2.4 Historia de usuario Componente Ruta Mínima Autos.

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Componente Ruta Mínima Autos.
<b>Usuario:</b> Luis Leandro Hernández Fleitas	
<b>Prioridad en Negocio:</b> Baja	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 3
<b>Descripción:</b> Determina el menor camino que existe entre dos nodos que el usuario selecciona. Incluye la creación del ambiente necesario para su ejecución.	

**Observaciones:** Es necesario contar con el mapa sobre el cual se mostrarán las rutas con un alto nivel de detalle. El resultado a mostrar depende de que componente esté en ejecución.

## **2.7 Fase de planificación**

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Este se expresa utilizado como medida el punto. Un punto se considera como una semana ideal de trabajo donde los miembros del equipo de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción (Beck y Fowler, 2000).

Para la determinación de los puntos estimados se ha tenido en cuenta la experiencia acumulada del programador en la realización de proyectos anteriores, lo que da una medida de cuanto puede demorar en la realización de estas tareas.

Tabla 2.5 Estimación de esfuerzo por Historia de Usuario

<b>Historia de Usuario</b>	<b>Puntos estimados (Semanas)</b>
Módulo Construir Grafo.	3
Componente Ruta Mínima Peatones.	4
Componente Ruta Mínima Autos.	3

### **2.7.1 Plan de iteraciones**

Al identificarse cada una de las historias de usuario y la estimación del esfuerzo dedicado a la realización de cada una de estas se procede a la etapa de implementación del proyecto (Beck y Fowler, 2000). Para el desarrollo del sistema se llevarán a cabo 3 iteraciones como se muestra a continuación:

**Iteración 1:**

En la primera iteración se implementan las HU de alta prioridad en el negocio, desarrollando así el Módulo Construir Grafo donde se realiza todo lo referente a la gestión de los nodos y caminos.

**Iteración 2:**

En la segunda iteración se implementan las HU de mediana prioridad en el negocio, desarrollando así el Componente de Ruta Mínima para Peatones, el cual es el encargado de calcular y devolver el menor camino entre puntos seleccionados por el usuario.

**Iteración 3:**

En la tercera iteración se implementan las HU de baja prioridad en el negocio, se desarrollará el Componente de Ruta Mínima para Autos, reutilizando algoritmos empleados en la realización de la iteración anterior.

**2.7.2. Plan de duración de las iteraciones**

El tiempo total que demora la realización de una iteración es el resultado de la suma de los puntos estimados para cada una de las historias de usuarios que intervienen en esta. Para el caso de este proyecto en cada iteración solo se estará realizando una historia de usuario, por lo que el tiempo de duración de esta coincide con el tiempo de la HU a desarrollar expresado en semanas.

Tabla 2.6 Plan de duración de las iteraciones

<b>Iteración</b>	<b>Orden de la historias de usuario a implementar</b>	<b>Duración total de la iteración</b>
Iteración 1	1- Módulo Construir Grafo.	3 semanas.
Iteración 2	1- Componente Ruta Mínima Peatones.	4 semanas.
Iteración 3	2- Componente Ruta Mínima Autos.	3 semanas.

## Capítulo 2: Características del Sistema.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

### 2.7.3. Plan de entregas

A continuación se presenta el plan de entregas ideado para la fase de implementación. Como producto del mismo se harán releases del sistema al finalizar cada iteración en la fecha aproximada que se indica en la siguiente tabla.

Tabla 2.7 Módulos y HU abarcadas

Módulos	Historias de usuario que abarca
Construir Grafo.	Módulo Construir Grafo.
Ruta Mínima	Componente Ruta Mínima Peatones Componente Ruta Mínima Autos

Tabla 2.8 Plan de entregas

Entregable	Final 1ra Iteración 1ra semana de marzo	Final 2da Iteración 2da semana de abril	Final 3ra Iteración 3ra semana de mayo
UCI Map	v0.1	v0.2	v1.0

Al final de cada una de las iteraciones deberá estar concluida una versión del producto, hasta llegar al final de la 3ra iteración donde el proyecto concluye en su versión 1.0.

## **2.8 Conclusiones**

En este capítulo se ha hecho una descripción de cada uno de los artefactos generados durante las fases de exploración y planificación en el desarrollo del sistema, teniendo en cuenta aspectos importantes como los requisitos funcionales y no funcionales, las tareas realizadas en cada iteración del ciclo de vida del software, definiéndole a cada tarea prioridad, fecha de entrega, riesgo; y se definieron fechas de entrega de cada de los releases del producto.

# CAPÍTULO 3

## DISEÑO, IMPLEMENTACIÓN Y PRUEBAS.

### 3.1 Introducción

La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste.

En el presente capítulo se detallan las tres iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiéndose las tareas generadas por cada historia de usuario, así como las pruebas de aceptación efectuadas sobre el sistema.

### 3.2 Fase de Diseño del Sistema

Tarjetas CRC (Clase, Responsabilidad, Colaboración)

- Usar tarjetas CRC, en el diseño grupal. Ayudan a evitar el enfoque procedimental y destacan la orientación a objetos.
- Cada tarjeta CRC representa un objeto. El nombre de Clase va arriba, las responsabilidades (qué debe hacer) a la izquierda, las clases asistentes (que colaboran) a la derecha.
- No suele ser necesario escribir la tarjeta completa; los participantes se familiarizan rápidamente con el propósito de cada clase.
- En la reunión CRC alguien simula el sistema discutiendo los mensajes intercambiados entre objetos.

En la realización de este sistema se utilizaron las siguientes clases:

En el módulo Construir Grafo:

- Acceso datos.
- Gestionar Nodos



### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

En el módulo Ruta Mínima:

- Acceso datos.
- Autos.
- Caminante.
- RutaMin.

Tabla 3.1 **Clase:** Acceso datos (Construir Grafo).

Responsabilidades	Clases relacionadas
Conectarse a la base de datos	

Tabla 3.2 **Clase:** Gestionar Nodos

Responsabilidades	Clases relacionadas
Crear Nodos	Acceso datos
Eliminar Nodos	Acceso datos
Actualizar Nodos Transitables	Acceso datos
Crear Matriz de caminos	Acceso datos
Conectar Nodos	Acceso datos
Mostrar Tabla de Nodos	Acceso datos

Tabla 3.3 **Clase:** Acceso datos (Ruta Mínima).

Responsabilidades	Clases relacionadas
Conectarse a la base de datos	
Consultar Tablas	
Calcular Longitud a Tablas	
Obtener Matriz de caminos	
Obtener Tabla Nodos	
Obtener Nodos Transitables	
Generar Listas de Datos	

Tabla 3.4 **Clase:** Autos.

Responsabilidades	Clases relacionadas
Obtener camino mínimo.	RutaMin
Calcular posiciones reales.	
Calcular Matriz para Autos.	Acceso datos
Mostrar datos de los nodos del camino.	Acceso datos

Tabla 3.5 **Clase:** Caminante.

Responsabilidades	Clases relacionadas
Obtener camino mínimo.	RutaMin
Obtener mapa de la matriz	RutaMin
Calcular posiciones reales	
Mostrar datos de los nodos del camino	Acceso datos

### 3.3 Fase de Implementación del Sistema

En la fase de implementación del sistema se realizan las iteraciones que no es más que la codificación de las historias de usuarios seleccionadas para ser desarrolladas en cada una de ellas. Al principio de estas se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de la ingeniería, asignando a un grupo de desarrollo (o una persona), responsable de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se detallan cada una de las iteraciones.

### 3.3.1 Interacciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de 6 semanas.

A continuación se muestran cada una de las iteraciones planificadas para el desarrollo del sistema, obteniendo al finalizar de cada una de estas el respectivo producto final.

### 3.3.2 Iteración 1

Tabla 3.6 Historia de Usuario: Iteración 1

Historia de Usuario	Estimación	Real
Módulo Construir Grafo.	3	3
<b>Total</b>	3	3

Tareas de las historias de usuario abordadas en la iteración 1.

Tabla 3.7 Tareas de Ingeniería: Diseñar Base de Datos

Tarea	
<b>Número de tarea:</b> 1	<b>Número de la HU:</b> 1
<b>Nombre de tarea:</b> Diseñar Base de Datos	
<b>Tipo de tarea:</b> Diseño de base de datos	<b>Puntos Estimados:</b> 0.1
<b>Fecha de inicio:</b> 9 de febrero del 2009	<b>Fecha de fin:</b> 10 de febrero del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Se identifican y crean los atributos que definen a los nodos y a la matriz de las aristas.	

### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

Tabla 3.8 Tareas de Ingeniería: Crear Nodos

Tarea	
<b>Número de tarea:</b> 2	<b>Número de la HU:</b> 1
<b>Nombre de tarea:</b> Crear Nodos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha de inicio:</b> 11 de febrero del 2009	<b>Fecha de fin:</b> 17 de febrero del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Obtener las coordenadas X y Y del nodo en el mapa real, llenar el campo del nombre que identificará el nodo. Luego estos datos serán enviados a la Base de Datos	

Tabla 3.9 Tareas de Ingeniería: Eliminar Nodos

Tarea	
<b>Número de tarea:</b> 3	<b>Número de la HU:</b> 1
<b>Nombre de tarea:</b> Eliminar Nodo.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha de inicio:</b> 18 de febrero del 2009	<b>Fecha de fin:</b> 20 de febrero del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Luego de seleccionar el nodo a eliminar se realizará una consulta a la Base de Datos la cual ejecutará dicha opción.	

### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

Tabla 3.10 Tareas de Ingeniería: Mostrar Nodos

Tarea	
<b>Número de tarea:</b> 4	<b>Número de la HU:</b> 1
<b>Nombre de tarea:</b> Mostrar Nodos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha de inicio:</b> 23 de febrero del 2009	<b>Fecha de fin:</b> 24 de febrero del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Mostrar en tablas el resultado de una consulta a la Base de Datos que devuelve todos los nodos existentes.	

Tabla 3.11 Tareas de Ingeniería: Conectar Nodos

Tarea	
<b>Número de tarea:</b> 5	<b>Número de la HU:</b> 1
<b>Nombre de tarea:</b> Conectar Nodos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha de inicio:</b> 25 de febrero del 2009	<b>Fecha de fin:</b> 3 de marzo del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Seleccionar en la aplicación los nodos a conectar. Realizar una consulta que devuelva las coordenadas de los mismos. Realizar el cálculo de distancia con los datos obtenidos. Insertar el resultado en la tabla Matriz de la Base de Datos.	

### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

Tabla 3.12 Tareas de Ingeniería: Crear Nodos Transitables

Tarea	
<b>Número de tarea:</b> 6	<b>Número de la HU:</b> 1
<b>Nombre de tarea:</b> Crear Nodos Transitables	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha de inicio:</b> 4 de marzo del 2009	<b>Fecha de fin:</b> 6 de marzo del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Seleccionar en la aplicación de uno a varios nodos que son transitables para los autos, modificando el estado de este campo en la Base de Datos.	

#### 3.3.3 Iteración 2

Tabla 3.13 Historia de Usuario: Componente Ruta Mínima Peatones

Historia de Usuario	Estimación	Real
Componente Ruta Mínima Peatones	4	4
<b>Total</b>	4	4

### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

Tareas de las historias de usuario abordadas en la iteración 2.

Tabla 3.14 Tareas de Ingeniería: Crear Matriz de Caminos.

Tarea	
<b>Número de tarea:</b> 1	<b>Número de la HU:</b> 2
<b>Nombre de tarea:</b> Crear Matriz de Caminos.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha de inicio:</b> 13 de marzo del 2009	<b>Fecha de fin:</b> 17 de marzo del 2009
<b>Responsable:</b> Luís Leandro Hernández Fleitas.	
<b>Descripción:</b> Se realiza una consulta a la Base de Datos para obtener la matriz de caminos, se procede a la conformación de una matriz en tiempo de ejecución con los datos obtenidos la que se utilizará para realizar los cálculos de distancia.	

Tabla 3.15 Tareas de Ingeniería: Calcular Camino Mínimo para Peatones

Tarea	
<b>Número de tarea:</b> 2	<b>Número de la HU:</b> 2
<b>Nombre de tarea:</b> Calcular Camino Mínimo para Peatones	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha de inicio:</b> 18 de marzo del 2009	<b>Fecha de fin:</b> 30 de marzo del 2009
<b>Responsable:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> El usuario selecciona a partir de un menú de selección, los extremos de la trayectoria a recorrer. Con estos datos y la matriz de caminos, se realizarán cálculos para la obtención de los nodos que intervienen en la menor trayectoria posible y el valor de esta.	

### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

Tabla 3.16 Tareas de Ingeniería: Conversión de Valores

Tarea	
<b>Número de tarea:</b> 3	<b>Número de la HU:</b> 2
<b>Nombre de tarea:</b> Conversión de Valores	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha de inicio:</b> 31 de marzo del 2009	<b>Fecha de fin:</b> 6 de abril del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Luego de obtener el valor de la menor trayectoria posible, se realizan los cálculos pertinentes para la conversión de la distancia obtenida a Metros.	

Tabla 3.17 Tareas de Ingeniería: Mostrar Camino Mínimo para Peatones

Tarea	
<b>Número de tarea:</b> 4	<b>Número de la HU:</b> 2
<b>Nombre de tarea:</b> Mostrar Camino Mínimo para Peatones	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha de inicio:</b> 7 de abril del 2009	<b>Fecha de fin:</b> 17 de abril del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Después de obtener todos los datos de una ruta mínima calculada, se procede a convertir estos datos en algo que el usuario pueda entender, dibujando sobre el mapa el camino a seguir.	



### 3.3.4 Iteración 3

Tabla 3.18 Historia de Usuario: Componente Ruta Mínima Autos

Historia de Usuario	Estimación	Real
Componente Ruta Mínima Autos	3	3
<b>Total</b>	3	3

Tareas de las historias de usuario abordadas en la iteración 3.

Tabla 3.19 Tareas de Ingeniería: Camino Mínimo para Autos

Tarea	
<b>Número de tarea:</b> 1	<b>Número de la HU:</b> 3
<b>Nombre de tarea:</b> Datos para obtener camino mínimo de Autos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha de inicio:</b> 28 de abril del 2009	<b>Fecha de fin:</b> 29 de abril del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Crear un menú para la que el usuario pueda seleccionar el modo de ruta mínima a calcular. El usuario selecciona los extremos de la trayectoria a recorrer. Con la obtención de estos datos comienza el proceso para el cálculo del camino mínimo para autos.	

### Capítulo 3: Diseño, Implementación y Pruebas.

“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”

Tabla 3.20 Tareas de Ingeniería: Filtrar Matriz de camino para Autos

Tarea	
<b>Número de tarea:</b> 2	<b>Número de la HU:</b> 3
<b>Nombre de tarea:</b> Filtrar Matriz de camino para Autos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha de inicio:</b> 30 de abril del 2009	<b>Fecha de fin:</b> 8 de mayo del 2009
<b>Responsables:</b> Luís Leandro Hernández Fleitas.	
<b>Descripción:</b> Se realizará un filtrado de la tabla Nodos obteniendo todos los que son transitable y los nodos extremos seleccionados por el usuario. Con estos datos se realizará un filtrado a la Matriz de camino, anteriormente obtenida de la tabla Matriz en la Base de Datos, para obtener así una nueva matriz donde solo intervengan los datos necesarios para formar una red de caminos la cual pueda ser accesible por los autos.	

Tabla 3.21 Tareas de Ingeniería: Calcular Camino Mínimo para Autos

Tarea	
<b>Número de tarea:</b> 3	<b>Número de la HU:</b> 3
<b>Nombre de tarea:</b> Calcular Camino Mínimo para Autos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha de inicio:</b> 11 de mayo del 2009	<b>Fecha de fin:</b> 14 de mayo del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Con los datos obtenidos del filtrado se realizarán cálculos para la obtención de los nodos que intervienen en la menor trayectoria posible y el valor de esta.	

### Capítulo 3: Diseño, Implementación y Pruebas.

“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”

Tabla 3.22 Tareas de Ingeniería: Conversión de Valores

Tarea	
<b>Número de tarea:</b> 4	<b>Número de la HU:</b> 3
<b>Nombre de tarea:</b> Conversión de Valores	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha de inicio:</b> 15 de mayo del 2009	<b>Fecha de fin:</b> 18 de mayo del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Luego de obtener el valor de la menor trayectoria posible, se realizan los cálculos pertinentes para la conversión de la distancia obtenida a Metros, y en función de esa distancia realizar cálculos del consumo de un auto según su autonomía, mostrando estos datos en el mapa.	

Tabla 3.23 Tareas de Ingeniería: Mostrar Camino Mínimo para Autos

Tarea	
<b>Número de tarea:</b> 5	<b>Número de la HU:</b> 3
<b>Nombre de tarea:</b> Mostrar Camino Mínimo para Autos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha de inicio:</b> 19 de mayo del 2009	<b>Fecha de fin:</b> 22 de mayo del 2009
<b>Responsables:</b> Luis Leandro Hernández Fleitas.	
<b>Descripción:</b> Luego de obtener los nodos que intervienen en la menor trayectoria posible y los valores convertidos, se muestra en el mapa el camino a seguir por los autos tomando solo las calles.	

### 3.4 Fase de Pruebas

#### 3.4.1 Pruebas Unitarias

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. Las pruebas de unidad posibilitan la propiedad colectiva de código, la refactorización, la integración frecuente. El agregado de funcionalidad incluye el agregado de pruebas.

##### 3.4.1.1 Pruebas de Unidad para el módulo Construir Grafo

Tabla 3.24 Pruebas de Unidad para Crear Nodos

Prueba de unidad	
<b>Historia de usuario:</b>	Módulo Construir Grafo.
<b>Componente:</b>	Acceso Datos
<b>Funcionalidad:</b>	Crear Nodos
<b>Objetivos:</b>	Verificar que la información es enviada a almacenar de forma correcta por el método.
<b>Descripción:</b>	Para realizar la prueba, se verifica que las consultas SQL, ejecuten el código requerido para almacenar los valores deseados, que las variables con los datos de entrada sean los correctos.
<b>Resultados esperados:</b>	En caso de haber ocurrido algún error en el proceso, el método devuelve un mensaje informando el error, en caso contrario se realiza un re direccionamiento para volver a usar la funcionalidad si es necesario.
<b>Observaciones:</b>	

Tabla 3.25 Pruebas de Unidad para Llenar Matriz

<b>Prueba de unidad</b>	
<b>Historia de usuario:</b>	Módulo Construir Grafo.
<b>Componente:</b>	Acceso Datos
<b>Funcionalidad:</b>	Llenar Matriz
<b>Objetivos:</b>	<p>Verificar que la información es enviada a almacenar de forma correcta por el método.</p> <p>Verificar cálculos de distancia.</p>
<b>Descripción:</b>	<p>Para realizar la prueba, se verificó que:</p> <p>Se obtuvo de la tabla nodo los valores de las coordenadas de los nodos a conectar. Se realizó efectivamente los cálculos para determinar la distancia entre los nodos seleccionados. Se verifica que las consultas SQL, ejecuten el código requerido para almacenar los valores deseados que las variables con los datos de entrada sean los correctos.</p>
<b>Resultados esperados:</b>	<p>En caso de haber ocurrido algún error en el proceso, el método devuelve un mensaje informando el error, en caso contrario se realiza un re direccionamiento para volver a usar la funcionalidad si es necesario.</p>
<b>Observaciones:</b>	

Tabla 3.26 Pruebas de Unidad para Nodos Transitables

Prueba de unidad	
<b>Historia de usuario:</b>	Módulo Construir Grafo.
<b>Componente:</b>	Acceso Datos
<b>Funcionalidad:</b>	Nodos Transitables
<b>Objetivos:</b>	<p>Verificar llegada al método de los datos deseados.</p> <p>Verificar que la información es enviada a almacenar de forma correcta por el método.</p>
<b>Descripción:</b>	<p>Para realizar la prueba, se verificó que:</p> <p>Llegara al método la lista de los nodos que son transitables.</p> <p>Se verifica que la consultas SQL, ejecuten el código requerido para actualizar los valores deseados que las variables con los datos de entrada sean los correctos.</p>
<b>Resultados esperados:</b>	<p>En caso de haber ocurrido algún error en el proceso, el método devuelve un mensaje informando el error, en caso contrario se realiza un re direccionamiento para volver a usar la funcionalidad si es necesario.</p>
<b>Observaciones:</b>	

### 3.4.1.2 Pruebas de Unidad para el módulo Ruta Mínima

Tabla 3.27 Pruebas de Unidad para getResultats

Prueba de unidad	
<b>Historia de usuario:</b>	Módulo Ruta Mínima.
<b>Componente:</b>	RutaMin
<b>Funcionalidad:</b>	getResultats
<b>Objetivos:</b>	Calcular el menor camino dado dos puntos de extremo y la matriz de nodos adyacentes.
<b>Descripción:</b>	<p>Para realizar la prueba, se verificó que el algoritmo pasa por todos los nodos existentes en la matriz.</p> <p>Verificar los valores de los cálculos intermedios realizados, para comprobar que se va obteniendo el menor camino posible.</p> <p>Verificar que al concluir el algoritmo devuelva la lista de datos.</p>
<b>Resultados esperados:</b>	En caso de haber ocurrido algún error en el proceso de cálculo del menor camino el método devuelve una lista vacía, en caso contrario devuelve un listado con la información del cálculo de la distancia del menor camino posible, además de los nodos implicados y los extremos seleccionados al comentar la operación.
<b>Observaciones:</b>	

Tabla 3.28 Pruebas de Unidad para getMatriz

Prueba de unidad	
<b>Historia de usuario:</b>	Módulo Ruta Mínima.
<b>Componente:</b>	Acceso Datos
<b>Funcionalidad:</b>	getMatriz
<b>Objetivos:</b>	Devolver matriz de nodos adyacentes.
<b>Descripción:</b>	<p>Para realizar la prueba, se verificó que la consulta realizada a la base de datos devolviera todos los nodos y los pesos existentes en la tabla matriz.</p> <p>Se compró que al crear las listas que contienen los datos de la consulta no se modificara el contenido ni las posiciones que originalmente tienen estos datos.</p>
<b>Resultados esperados:</b>	<p>En caso de haber ocurrido algún error en el proceso de obtención de la matriz el método devuelve una lista vacía, en caso contrario devuelve una lista bidimensional con la información de la matriz de caminos.</p>
<b>Observaciones:</b>	



Tabla 3.29 Pruebas de Unidad para getMatriz\_Transitable

Prueba de unidad	
<b>Historia de usuario:</b>	Módulo Ruta Mínima.
<b>Componente:</b>	Acceso Datos
<b>Funcionalidad:</b>	getMatriz_Transitable
<b>Objetivos:</b>	Devolver matriz de adyacencia para los nodos transitables.
<b>Descripción:</b>	<p>Para realizar la prueba, se verificó que la consulta realizada a la base de datos devolviera todos los nodos y los pesos existentes en la tabla matriz.</p> <p>Se compró que al crear las listas que contienen los datos de la consulta no se modificara el contenido ni las posiciones que originalmente tienen estos datos.</p> <p>Luego se verificó que en el filtrado de la matriz original para la matriz de los nodos transitables, no ocurriera ninguna pérdida de datos, y que al correr los elementos de posición para los cubrir espacios vacíos estuvieran en relación con los valores deseados.</p>
<b>Resultados esperados:</b>	En caso de haber ocurrido algún error en el proceso de obtención de la matriz el método devuelve una lista vacía, en caso contrario devuelve una lista bidimensional con la información de la matriz de caminos para los nodos transitables.
<b>Observaciones:</b>	Este método es en especial complejo pues cuando se realiza el filtrado para los nodos transitables, hay que asegurarse que al correr las posiciones quede la matriz cuadrada con los valores correspondientes al filtrado.

Como resultado de la realización de las pruebas de unidad se pudo comprobar el funcionamiento correcto de cada uno de los módulos del sistema, realizándose pruebas sobre las funcionalidades encargadas de las tareas más importantes de la aplicación.

Tabla 3.30 Resultado de las Pruebas de Unidad.

	Cantid ad	F allo	É xito	Cobertura
Pruebas	6		X	100 %
Total	6			

### 3.4.2 Pruebas de Aceptación

Las pruebas de aceptación permiten confirmar que la HU ha sido implementada correctamente al final de cada iteración. Este período de prueba se conoce también como período de caja negra donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable.

La prueba de aceptación es realizada por un grupo de usuarios finales o los clientes del sistema, para asegurarse que el sistema desarrollado cumple sus requisitos. A continuación se representan las pruebas de aceptación realizadas para las HU del sistema:

Tabla 3.31 Pruebas de Aceptación Mostrar Formulario Crear Nodo

Prueba de Aceptación	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Mostrar Formulario Crear Nodo	
<b>Descripción:</b> Prueba de funcionalidad para crear un nuevo nodo.	
<b>Condiciones de ejecución:</b> Debe mostrar el formulario para entrar los datos del nombre y las coordenadas del nodo que se va a crear.	
<b>Entrada / Pasos de Ejecución:</b> Se accede al módulo construir grafo y se muestra la página que contiene el formulario para crear un nuevo nodo.	
<b>Resultado esperado:</b> El formulario es mostrado correctamente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.32 Pruebas de Aceptación Mostrar mapa en la web.

Prueba de Aceptación	
<b>Código:</b> HU1_P2	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Mostrar mapa en la web.	
<b>Descripción:</b> Prueba de funcionalidad para Mostrar el mapa de la UCI en la web.	
<b>Condiciones de ejecución:</b> Debe mostrar en la web el mapa de la UCI del cual se obtendrán las coordenadas.	
<b>Entrada / Pasos de Ejecución:</b> Se accede al módulo construir grafo y se muestra la página que contiene el formulario para crear un nuevo nodo. Al dar clic en el vínculo “Ir a Mapa para Obtener Coordenadas” debe mostrarse en la web la imagen requerida.	
<b>Resultado esperado:</b> El mapa es mostrado correctamente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

Tabla 3.33 Pruebas de Aceptación Obtener coordenadas dentro mapa de la UCI en la web.

Prueba de Aceptación	
<b>Código:</b> HU1_P3	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Obtener coordenadas dentro mapa de la UCI en la web.	
<b>Descripción:</b> Prueba de funcionalidad para obtener las coordenadas de un punto dentro del mapa de la uci en la web.	
<b>Condiciones de ejecución:</b> Debe mostrar en la web el mapa de la UCI del cual se obtendrán las coordenadas.	
<b>Entrada / Pasos de Ejecución:</b> Se accede al módulo construir grafo y se muestra la página que contiene el formulario para crear un nuevo nodo. Al dar clic en el vínculo “Ir a Mapa para Obtener Coordenadas” debe mostrarse en la web la imagen requerida. Luego al dar clic sobre la región seleccionada deberá regresar a la página del formulario llenando los campo de las coordenadas X y Y referentes a la posición obtenida del mapa.	
<b>Resultado esperado:</b> Los campos con las coordenadas X y Y son llenados y mostrados correctamente.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.34 Pruebas de Aceptación Eliminar nodo de la Base de Datos.

Prueba de Aceptación	
<b>Código:</b> HU1_P4	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Eliminar nodo de la Base de Datos.	
<b>Descripción:</b> Prueba de funcionalidad para eliminar un nodo de la Base de Datos seleccionando el nombre en la interfaz visual.	
<b>Condiciones de ejecución:</b> Para poder ejecutar esta acción debe de mostrarse un listado con todos los nodos existentes	

### Capítulo 3: Diseño, Implementación y Pruebas.

“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”

<b>Entrada / Pasos de Ejecución:</b> Se accede al módulo construir grafo y se muestra la página que contiene el formulario para eliminar un nodo. Debe buscar en la lista donde se muestran todos los nodos existentes, el que desee eliminar y dar clic en aceptar. Se enviará para la base de datos la consulta necesaria para eliminar el nodo deseado.
<b>Resultado esperado:</b> Se ha eliminado de la base de datos y de la lista el nodo seleccionado.
<b>Evaluación de la prueba:</b> Prueba satisfactoria.

Tabla 3.35 Pruebas de Aceptación Conectar Nodos.

Prueba de Aceptación	
<b>Código:</b> HU1_P5	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Conectar Nodos	
<b>Descripción:</b> Prueba de funcionalidad para conectar dos nodos. Almacenar en la Base de Datos en la tabla matriz la distancia calculada entre las coordenadas los nodos seleccionados por el nombre en la interfaz visual.	
<b>Condiciones de ejecución:</b> Para poder ejecutar esta acción debe de mostrarse los listados con todos los nodos existentes en la Base de Datos.	
<b>Entrada / Pasos de Ejecución:</b> Se accede al módulo construir grafo, ir al enlace “Conectar nodos”, donde se muestra la página que contiene el formulario para realizar la conexión de los nodos. Debe buscar y seleccionar un nodo dentro de las listas donde se muestran todos y dar clic en conectar. Se enviará para la base de datos la consulta necesaria para insertar en la tabla matriz el cálculo de la distancia entre los nodos seleccionados.	
<b>Resultado esperado:</b> Se muestran las listas con los nombres de los nodos y se han conectado los nodos seleccionados.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.36 Pruebas de Aceptación Crear nodos transitables

Prueba de Aceptación	
<b>Código:</b> HU1_P6	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Crear nodos transitables	
<b>Descripción:</b> Prueba de funcionalidad para crear nodos transitables	
<b>Condiciones de ejecución:</b> Para poder ejecutar esta acción debe de mostrarse un listado con todos los nodos existentes	
<b>Entrada / Pasos de Ejecución:</b> Acceder al formulario para crear los nodos transitables, seleccionar de la lista que se muestra con todos los que de desean convertir en transitables para los autos. Se enviara para la base de datos la consulta necesaria para modificar este valor del nodo.	
<b>Resultado esperado:</b> Se muestran todos los nodos en la lista y se modifica su valor en la base de datos.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.37 Pruebas de Aceptación Mostrar tabla Nodos

Prueba de Aceptación	
<b>Código:</b> HU1_P7	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Mostrar tabla Nodos	
<b>Descripción:</b> Prueba de funcionalidad para mostrar una tabla que contenga todos los nodos que existen en la base de datos.	
<b>Condiciones de ejecución:</b> Para poder ejecutar esta acción deben existir datos en la tabla nodos.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la página donde se muestran todos los nodos insertados hasta el momento.	
<b>Resultado esperado:</b> Se muestra una tabla con todos los nodos existente en la base de datos.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.38 Pruebas de Aceptación Hipervínculos funcionales.

Prueba de Aceptación	
<b>Código:</b> HU1_P8	<b>Historia de Usuario:</b> Módulo Construir Grafo.
<b>Nombre:</b> Hipervínculos funcionales.	
<b>Descripción:</b> Prueba de funcionalidad para los Hipervínculos que muestren el contenido esperado.	
<b>Condiciones de ejecución:</b> Que se encuentre en ejecución la aplicación.	
<b>Entrada / Pasos de Ejecución:</b> Al seleccionar cada uno de los hipervínculos, este debe mostrar la página esperada por el usuario.	
<b>Resultado esperado:</b> Se muestra la página esperada.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.39 Pruebas de Aceptación Mostrar Interfaz.

Prueba de Aceptación	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Mostrar Interfaz.	
<b>Descripción:</b> Prueba de funcionalidad para la interfaz con que el usuario va a interactuar para determinar el menor camino.	
<b>Condiciones de ejecución:</b> Que se encuentre en ejecución la aplicación, que el usuario acceda al menú correspondiente.	
<b>Entrada / Pasos de Ejecución:</b> Cuando el usuario accede a la opción de camino mínimo se muestra un formulario donde estarán todos los datos necesarios para obtener una ruta.	
<b>Resultado esperado:</b> Se muestra la página esperada.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.40 Pruebas de Aceptación Listado de Nodos

Prueba de Aceptación	
<b>Código:</b> HU2_P2	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Listado de Nodos	
<b>Descripción:</b> Prueba de funcionalidad para los listado donde se muestran todos los nodos	
<b>Condiciones de ejecución:</b> Que se encuentre en ejecución la aplicación, que el usuario acceda al menú correspondiente.	
<b>Entrada / Pasos de Ejecución:</b> Cuando el usuario accede a la opción de camino mínimo se muestra un formulario donde estarán dos listados con los nombres de los nodos existentes.	
<b>Resultado esperado:</b> Se muestran los valores esperados.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.41 Pruebas de Aceptación Modo de Camino a Mostrar

Prueba de Aceptación	
<b>Código:</b> HU2_P3	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Modo de Camino a Mostrar	
<b>Descripción:</b> Prueba de funcionalidad para seleccionar el modo de camino a mostrar.	
<b>Condiciones de ejecución:</b> Que se encuentre en ejecución la aplicación, que el usuario acceda al menú correspondiente. Debe seleccionar los valores extremos del camino a obtener.	
<b>Entrada / Pasos de Ejecución:</b> Cuando el usuario seleccione entre el modo para peatones y el modo para autos, el sistema será capaz de mostrar soluciones diferentes, atendiendo a la opción seleccionada.	
<b>Resultado esperado:</b> Se muestran los valores esperados.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	



Tabla 3.42 Pruebas de Aceptación Mostrar mapa peatones

Prueba de Aceptación	
<b>Código:</b> HU2_P4	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Mostrar mapa peatones	
<b>Descripción:</b> Prueba de funcionalidad para mostrar el mapa del recorrido de los peatones	
<b>Condiciones de ejecución:</b> Que se encuentre en ejecución la aplicación, que el usuario acceda al menú correspondiente. Debe seleccionar los valores extremos del camino a obtener.	
<b>Entrada / Pasos de Ejecución:</b> Cuando el usuario da clic en el botón mostrar camino, se cargará una página donde se muestra el mapa de la UCI y la trayectoria mínima a recorrer por los peatones.	
<b>Resultado esperado:</b> Se muestran los valores esperados.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.43 Pruebas de Aceptación Mostrar mapa autos

Prueba de Aceptación	
<b>Código:</b> HU2_P5	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Mostrar mapa autos	
<b>Descripción:</b> Prueba de funcionalidad para mostrar el mapa del recorrido para los autos	
<b>Condiciones de ejecución:</b> Que se encuentre en ejecución la aplicación, que el usuario acceda al menú correspondiente. Debe seleccionar los valores extremos del camino a obtener.	
<b>Entrada / Pasos de Ejecución:</b> Cuando el usuario da clic en el botón mostrar camino, se cargará una página donde se muestra el mapa de la UCI y la trayectoria mínima a recorrer por los autos.	
<b>Resultado esperado:</b> Se muestran los valores esperados.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.44 Pruebas de Aceptación Mostrar leyenda para peatones

Prueba de Aceptación	
<b>Código:</b> HU2_P6	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Mostrar leyenda para peatones	
<b>Descripción:</b> Prueba de funcionalidad para mostrar la leyenda en el mapa de recorrido para peatones	
<b>Condiciones de ejecución:</b> Haber enviado a mostrar la ruta mínima para peatones.	
<b>Entrada / Pasos de Ejecución:</b> Cuando se muestra el mapa con el recorrido para los peatones, también se mostrará una tabla que contiene el valor de la distancia que interviene en el recorrido y nombres de nodos tomados como referencia.	
<b>Resultado esperado:</b> Se muestran los valores esperados.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Tabla 3.45 Pruebas de Aceptación Mostrar leyenda para autos.

Prueba de Aceptación	
<b>Código:</b> HU2_P7	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Mostrar leyenda para autos	
<b>Descripción:</b> Prueba de funcionalidad para mostrar la leyenda en el mapa de recorrido para los autos.	
<b>Condiciones de ejecución:</b> Haber enviado a mostrar la ruta mínima para peatones.	
<b>Entrada / Pasos de Ejecución:</b> Cuando se muestra el mapa con el recorrido para los autos, también se mostrará una tabla que contiene el valor de la distancia que interviene en el recorrido, los nombres de nodos tomados como referencia y el resultado de los cálculos para diferentes autonomías de los autos.	
<b>Resultado esperado:</b> Se muestran los valores esperados.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

### Capítulo 3: Diseño, Implementación y Pruebas.

“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”

Tabla 3.46 Pruebas de Aceptación Hipervínculos funcionales.

Prueba de Aceptación	
<b>Código:</b> HU2_P8	<b>Historia de Usuario:</b> Módulo Ruta Mínima
<b>Nombre:</b> Hipervínculos funcionales.	
<b>Descripción:</b> Prueba de funcionalidad para los Hipervínculos que muestren el contenido esperado.	
<b>Condiciones de ejecución:</b> Que se encuentre en ejecución la aplicación.	
<b>Entrada / Pasos de Ejecución:</b> Al seleccionar cada uno de los hipervínculos, este debe mostrar la página esperada por el usuario.	
<b>Resultado esperado:</b> Se muestra la página esperada.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

#### Resumen del resultado de las Pruebas de Aceptación.

Como resultado de la realización de las pruebas de aceptación se pudo comprobar el resultado definitivo de la conformidad del cliente con las funcionalidades implementadas por los desarrolladores.

Tabla 3.47 Resultado de las Pruebas de Aceptación

	Cantid ad	F allo	É xito	Cobertura
Probadas	8		X	
Total	8			100 %

### 3.4.2.1 Análisis estadístico para la encuesta

Como parte de las pruebas de aceptación se ha realizado una encuesta: **¿Le ha sido útil la aplicación?** a la comunidad universitaria, para medir la utilidad que aporta la aplicación a esta.

Para este proceso, de manera ideal, se encuestaría a todos los usuarios de la universidad, estudiantes, profesores, trabajadores no docentes, etc. En general todas las personas con una cuenta de usuario, cualquiera de ellos podrían hacer uso de la aplicación, como este proceso ideal no puede ser posible por razones de coste, tiempo, etc. se le hace solo una parte.

Aplicando la estadística y en particular la teoría del muestreo podemos resolver este problema aplicando la encuesta a un número menor de usuarios y que el resultado que se obtenga para este grupo pueda asegurar, con cierto nivel de confianza, que se cumple también para el total de los usuarios potenciales. De esta forma el total posible de usuarios a encuestar sería la población y como se encuesta un número menor pues este número sería el tamaño de muestra.

La teoría del muestreo nos permite calcular un tamaño de muestra y nos brinda además métodos de como escoger esta muestra, de manera que dados los resultados que genere esta muestra, se pueda inferir lo que pasa en la población, todo ello sujeto a una probabilidad y a un error.

En este proceso se va a medir la aceptación o utilidad del sistema y pudiéramos calcularla de la siguiente forma: la proporción de encuestados que aceptan o le ven utilidad al sistema.

Para ello se hizo la siguiente pregunta a cada encuestado ¿Le ha sido útil la aplicación? Cada persona podía escoger solo una de las siguientes respuestas:

Muy útil.

Útil.

Más o menos útil.

No me sirve.

Para nada útil.

### Capítulo 3: Diseño, Implementación y Pruebas.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

Para medir lo que queremos consideremos las personas que respondieron las variantes a y b (“Muy útil” y “Útil”) y veremos la proporción que representan respecto al total y esta sería entonces la variable a medir.

Ahora necesitamos el tamaño de muestra necesario. Como la población es conocida y finita, dado que la variable a medir, sería la proporción  $p$  entonces el tamaño de muestra necesario se calcula de la siguiente forma:

$$n = \frac{Z_{1-\frac{\alpha}{2}}^2 * p * q * N}{e^2(N - 1) + Z_{1-\frac{\alpha}{2}}^2 * p * q}$$

Donde: **N**: Es el tamaño de la población

**e**: Es el error que estamos dispuestos a aceptar (entre un 8 y un 10 %).

**$\alpha$** : Es el de significación siendo  $1 - \alpha$  el grado de confianza.

**$Z_{1-\alpha/2}$** : Es un percentil de la distribución normal. Valor estandarizado en función del grado de confiabilidad de la muestra calculada.

**p**: Es la proporción esperada.

**q**: =  $1 - p$ .

Ahora definamos cada uno de estos valores dado nuestro problema en particular.

Tenemos que la cantidad de usuarios potenciales o que tiene cuenta de usuario en la universidad son aproximadamente 16000 personas.

Luego podemos definir:

$N = 16000$ .

Podemos asumir que estamos dispuestos a aceptar un error del 5%.

Luego:

$e = 0,05$

Para un nivel de confianza del 95%.

Luego:

$$\alpha = 0,05$$

Al calcular el valor de  $Z_{1-\alpha/2}$  en las tablas estadísticas o con el uso del software SPSS (del inglés Statistical Package for the Social Sciences)

Tenemos que:

$$Z_{1-\alpha/2} \approx 1.64$$

En el caso de  $p$  que es la proporción esperada de personas que acepten o hallen entre “Muy útil” y “Útil” el sistema, en este caso no conocemos el valor más apropiado, una forma de saberlo es utilizando un valor de  $p$  de sistemas similares, pero no contamos con este dato. Luego utilizaremos el valor de 0,5 que es el que brinda el valor de  $n$  más grande y con ello se garantiza la calidad del proceso.

Entonces:

$$p = 0,5 \text{ lo que implica que } q = 0,5.$$

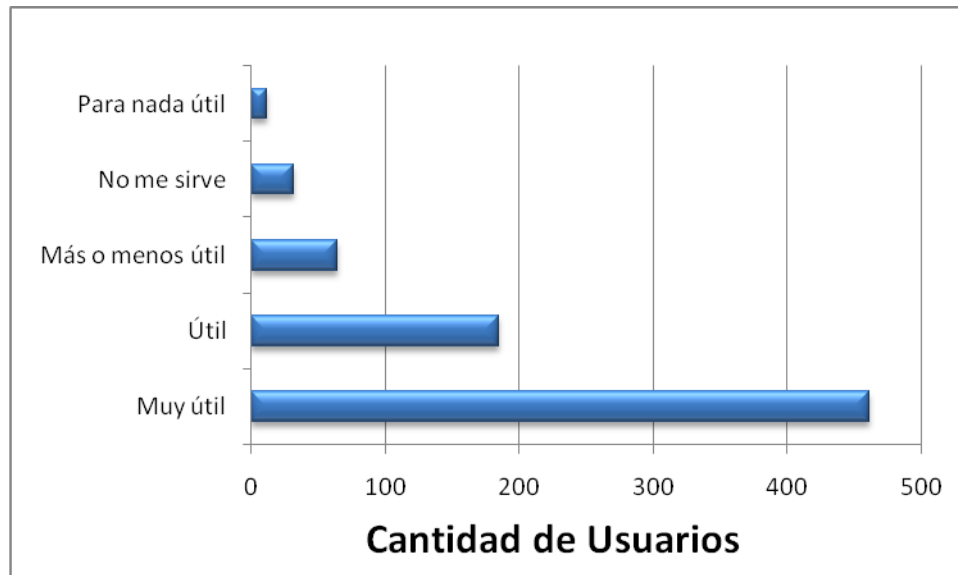
Sustituyendo los valores anteriores en:

$$n = \frac{Z_{1-\frac{\alpha}{2}}^2 * p * q * N}{e^2(N - 1) + Z_{1-\frac{\alpha}{2}}^2 * p * q}$$

Obtenemos que  $n = 266,071$  y lo aproximamos por exceso a  $n = 267$ .

Concluyendo que para obtener una estimación de la proporción de la cantidad de usuarios que aceptan o hallan entre “Muy útil” y “Útil” el sistema, con un nivel de confianza del 95 % y un error del 5 %, se deben encuestar a 267 usuarios.

En la práctica, y gracias a las tecnologías de la información se pudo encuestar de manera aleatoria a 754 usuarios. Obteniéndose los siguientes resultados:



**Diagrama de Barras**

Ahora ya que la variable medida tiene 5 posibles respuestas vamos a convertirla en dos respuestas para poder medir la proporción de personas que aceptan el sistema.

Para ello consideramos que las personas que responden “Muy útil” y “Útil” aceptan el sistema, y el resto no lo acepta. Al contar la cantidad de personas que “Aceptan” tenemos un total de 646 personas (cantidad que respondió “Muy útil” más la cantidad que respondió “Útil”, 461 y 185 respectivamente) contra un total de 754.

Luego la proporción de personas que aceptan el sistema es:

$$p = 646 / 754 = 0.856 \text{ o sea un } 85 \% \text{ de las personas.}$$

Esta proporción es la que se observa en la muestra. Calculemos ahora un intervalo de confianza para la proporción de las personas que aceptan el sistema del total de usuarios, o sea la población.

Al calcular el intervalo de confianza para la proporción en la población usando el software SPSS obtenemos.

[0.82498 ; 0.873194]

O sea que con un 95 % de confianza podemos decir que la proporción de personas que aceptan el sistema está entre el 82 % y el 87%.

Ahora, hagamos una prueba de hipótesis para contrastar la hipótesis de que la proporción de la población es menor que el 85%. Al realizar la prueba en el SPSS obtenemos que:

Para las hipótesis

Hipótesis Nula: proporción = 0,85

Alternativa: proporción < 0,85

Se obtiene un p-Value = 0,524401

Lo que nos indica que con un 95 % de confianza no rechazar la hipótesis nula. O sea que no tenemos evidencia para decir que la proporción de personas que aceptan el sistema sea menor que el 85%.

### **3.5 Conclusiones**

Luego de realizar un estudio de los pasos a seguir para llevar a cabo las respectivas fases de Diseño, Codificación y Pruebas, y los conceptos a tener en cuenta en cada una de ellas, se ha podido comprobar que la metodología proporciona una base sólida para la creación de aplicaciones de alta calidad, fáciles de mantener y que responde en todo momento a los intereses del cliente. Además como parte de las pruebas de aceptación se realizó un análisis estadístico para medir el nivel de utilidad del sistema en la comunidad de usuarios UCI, con lo cual se pudo confirmar que el 85% encuentran la aplicación entre “Muy útil” y “Útil”.



# CAPÍTULO 4

## ESTUDIOS DE FACTIBILIDAD

### 4.1 Introducción

Estudio de factibilidad es el análisis comprensivo de los resultados financieros, económicos y sociales de una inversión. Este evalúa si el equipo disponible puede desarrollar el proyecto propuesto y si tienen las capacidades técnicas requeridas por cada alternativa del diseño que se esté considerando.

El objetivo de este capítulo es realizar un estudio de factibilidad del sistema propuesto, obtener resultados que permitan brindar una ayuda importante para la toma de decisiones, obteniendo así una idea aproximada del costo y el esfuerzo empleados en el desarrollo del mismo.

### 4.2 COCOMO II

El **Modelo Constructivo de Costes** (o **COCOMO**, por su acrónimo del inglés **CO**nstructive **CO**st **MO**del) propuesto y desarrollado por Barry Boehm, es uno de los modelos de estimación de costos mejor documentados y utilizados. Permitiendo estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo software.

Está compuesto por tres modelos que se adaptan tanto a las necesidades de los diferentes sectores descritos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información.

El estudio de factibilidad de este sistema se realizó mediante el modelo de diseño temprano que se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o aspectos específicos del proceso a utilizar. Este nivel de detalle en este modelo es consistente con el nivel

general de información disponible y con el nivel general de estimación detallada que es necesaria en esta etapa, lo que concuerda con el uso de Puntos de Función, para estimar tamaño usa Puntos de Función No Ajustados como métrica de medida y el uso de un número reducido de factores de costo.

### **4.3 Características del proyecto**

El primer paso a llevar a cabo para la estimación del proyecto consiste en la obtención de los Puntos de Función desajustados, los cuales están dados por la suma de cada una de las entradas, las salidas y las consultas externas del sistema, así como los archivos lógicos internos y de interfaz externos. A continuación se muestran cada una de estas características aplicadas al software en cuestión.

#### **4.3.1 Entradas externas**

Se definen como un proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro. En el caso particular de este software se cuenta con 3 entradas externas, especificadas en la siguiente tabla:

Tabla 4.1 Entradas Externas

<b>Nombre de la entrada externa</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de elementos de datos</b>	<b>Clasificación (simple, media y compleja)</b>
Crear Nodo.	0	3	Simple
<b>Total</b>	<b>0</b>	<b>3</b>	

### 4.3.2 Salidas externas

Se definen como un proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados (datos que se calculan a partir de otros, datos cruzan la frontera del sistema desde adentro hacia afuera). Las salidas externas vinculadas al proyecto se describen a continuación.

Tabla 4.2 Salidas Externas

<b>Nombre de la entrada externa</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de elementos de datos</b>	<b>Clasificación (simple, media y compleja)</b>
Reporte de Nodos.	0	1	Simple
Mostrar Mapa con ruta óptima.	0	1	Simple
<b>Total</b>	<b>0</b>	<b>2</b>	

### 4.3.3 Consultas Externas

Se definen como un proceso elemental con componentes de entrada y de salida donde un Actor del sistema rescata datos de uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos).

## Capítulo 4: Estudios de factibilidad.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

Tabla 4.3 Consultas Externas

Nombre de la consulta externa	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
Menú de Selección	0	3	simple
<b>Total</b>		<b>3</b>	

### 4.3.4 Archivos Lógicos Internos

Constituyen un grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de entradas externas. En el proyecto se cuenta con 2 archivos lógicos internos, como se muestra en la siguiente tabla.

Tabla 4.4. Archivos Lógicos Internos

Nombre del fichero interno	Cantidad de records	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
Base de datos	1102	2	compleja
<b>Total</b>		<b>2</b>	

### 4.3.5 Archivos de Interfaz Externos

Son un grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones, es decir, cada Archivo de Interfaz Externo es

#### Capítulo 4: Estudios de factibilidad.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

un Archivo Lógico Interno de otra aplicación. En este caso no contamos con Archivos de interfaz externas, como muestra la siguiente tabla:

Tabla 4.5 Archivos de interfaz externos

Nombre de los archivos lógicos internos	Cantidad de records	Cantidad de elementos de datos	Clasificación (simple, media, compleja)
<b>Total</b>		0	

#### 4.3.6 Puntos de función desajustados

La siguiente tabla muestra la recopilación de las características del sistema anteriormente expuestas, y el peso correspondiente a las mismas según la complejidad, dando como resultado final, los puntos de función sin ajustar pertenecientes al proyecto.

Tabla 4.6 Puntos de función sin ajustar

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entrada Externa	3	3	0	4	0	6	9
Salida Externa	2	4	0	5	0	7	8
Grupos Lógicos de datos internos	0	7	0	10	2	15	30
Grupos Lógicos de datos de interfaz	0	5	0	7	0	10	0
Consulta Externas	3	3	0	4	0	6	9
<b>Total (UFP)</b>							56

#### 4.4 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo

El método COCOMO II consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la Cantidad de Líneas de Código (SLOC, *Source Lines of Code*) estimados para un proyecto, como base para medir tamaño en los modelos de estimación de Diseño Temprano. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo que influyen en el esfuerzo requerido para el desarrollo del software. La meta es obtener un número que caracterice completamente al sistema.

##### 4.4.1 Cálculo del esfuerzo nominal

Se aplica la ecuación de cálculo del esfuerzo nominal:

$$PM_{\text{nominal}} = A * (\text{Size})^E$$

**PM nominal:** es el esfuerzo nominal requerido en hombre/mes.

**A:** Es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. El modelo la calibra con un valor de 2.94.

**Size:** Es el tamaño estimado del software, en Puntos de Función sin ajustar (convertibles a KSLOC). Se calcula el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza PHP (factor de conversión = 53 SLOC/UFP). Entonces:

$$\text{Size} = 53 * 56 = 2968 \text{ SLOC}$$

$$\text{Size} = 3.0 \text{ KSLOC}$$

#### Capítulo 4: Estudios de factibilidad.

“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”

Tabla 4.7. Características.

Características	Valor
Puntos de función desajustados	56
Lenguaje (PHP)	53
Instrucciones fuentes por puntos de función	2968 SLOC

**E:** Es una constante denominada *Factor Escalar*. Se calcula ponderando las variables escalares, mediante la ecuación:

$$E = 0.91 + 0.01 * \Sigma (W_i)$$

Donde las  $W_i$  se muestran en la siguiente tabla:

Tabla 4.8 Factor Escalar ( $W_i$ )

Nombre	Valor	Justificación
PREC	2.24	Existen proyectos similares a nivel mundial y nacional.
FLEC	2.03	Cuenta con alta flexibilidad.
RESL	3.80	Se identificaron algunos de los riesgos críticos.
TEAM	3.29	Existe interacción cooperativa.
PMAT	3.12	Existe una experiencia previa en aplicaciones de este tipo.
<b>Total</b>	<b>14.48</b>	

Luego:  $E = 0.91 + 0.01 * 14.48 = 1.0548$ .

Entonces:

$$PM \text{ nominal} = A * (\text{Size})^E$$

$$PM \text{ nominal} = 2.94 * (3 \text{ KSLOC})^{1.0548}$$

$$PM \text{ nominal} = 9.3673 \text{ hombre/mes.}$$

**4.4.2 Cálculo del esfuerzo ajustado**

Se aplica la ecuación de cálculo del esfuerzo ajustado:

$$PM \text{ ajustado} = PM \text{ nominal} * \Pi (MEi)$$

Tabla 4.9 Multiplicadores de esfuerzo.

Nombre	Valor	Justificación
RUSE	1.07	El nivel de reusabilidad es alto.
RCPX	1.00	Confiabilidad, documentación y volumen de datos moderados.
PDIF	1.00	Uso de memoria y almacenamiento normal, plataforma estable.
PERS	1.00	Capacidad del personal.
SCED	1.00	Se empleó el tiempo planificado para el desarrollo del sistema.
<b>Total</b>	<b>1.07</b>	

Entonces: **PM** ajustado = 9.3673 \* 1.07 = 10.02 hombre/mes.

$$PM \text{ ajustado} = 10.02 \text{ hombre/mes}$$

**4.4.3 Cálculo del tiempo de desarrollo, cantidad de hombres y costo**

Valores calibrados:

$$A = 2.94 \quad B = 0.91 \quad C = 3.67 \quad D = 0.24 \quad E = 1.0548$$

$$F = D + 0.2 * (E - B)$$

$$F = 0.24 + 0.2 * (1.0548 - 0.91) = 0.27$$

Cálculo del tiempo que se necesita para el desarrollo del proyecto.

$$TDEV \text{ (Tiempo de desarrollo)} = C * (PM \text{ ajustado})^F$$



#### Capítulo 4: Estudios de factibilidad.

“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”

---

$$\text{TDEV (Tiempo de desarrollo)} = 3.67 * (10.02)^{0.27} = 6.83 \text{ meses}$$

Cálculo de la cantidad de hombres necesitados para el desarrollo del proyecto.

$$\text{CH (Cantidad de Hombres)} = \text{PM ajustado} / \text{TDVE} = 10.02 / 6.83 = 1.47 \approx 2$$

Cálculo del costo del proyecto

$$\text{Costo} = \text{CH} * \text{PM} * \text{Salario}$$

$$\text{Costo} = 1.51 * 10.02 * 100 = 1513.02 \text{ pesos}$$

### 4.5 Resultados

La siguiente tabla muestra los resultados obtenidos luego de haber realizado el análisis y efectuado los cálculos para determinar el costo y esfuerzo requeridos por el proyecto.

Tabla 4.10. Resultados.

Indicador	Valor
Cantidad de hombres Estimados	2 hombres
Tiempo de desarrollo	6.83 meses
Esfuerzo para cantidad de hombres estimados.	10.02 hombres/mes
Salario medio	\$100
Costo	\$1513.02

#### 4.5.1. Análisis de costos y beneficios

El análisis de costos y beneficios constituye una ayuda importante en la toma de decisiones, ayuda que frecuentemente brinda la información necesaria para determinar si la actividad es deseable, o si, por el contrario viene a constituir un desperdicio.

En términos generales, el análisis de costos y beneficios es un instrumento para desarrollar en forma sistemática una información útil acerca de los efectos deseables e indeseables de los proyectos

Para la realización de este proyecto se ha tenido en cuenta diferentes factores para que no influyan en el aumento del costo, como la utilización de herramientas y plataformas libres que no requieren pagos por concepto de licencias. Pero el desarrollo de todo producto siempre tiene un costo asociado, en el caso de este solo influye el salario del desarrollador, costo que es superado por los beneficios que aportará el sistema una vez puesto en explotación.

#### **4.6 Conclusiones**

En este capítulo se ha realizado un estudio para determinar si es factible desarrollar este producto, haciendo análisis de costos y los beneficios que aportará el sistema. Concluyendo que es viable su desarrollo teniendo en cuenta todos los aspectos analizados anteriormente.

# CONCLUSIONES GENERALES

Después de culminar la investigación se puede concluir que se cumplió el objetivo propuesto de desarrollar una aplicación, utilizando algoritmos matemáticos para gestionar la toma de decisiones realizando cálculos de eficiencia, optimización y ahorro de esfuerzo para la circulación de automóviles y personal en la Universidad de las Ciencias Informáticas.

Para su cumplimiento se desarrolló una aplicación web que cuenta con funcionalidades para brindarle al usuario en pocos segundos el resultado de la trayectoria más corta que debe seguir.

Para lograr una solución que cumpliera con el objetivo planteado, se realizó un análisis de las aplicaciones web que existen en el mundo que aplican soluciones similares a las que se deseaban lograr, donde a partir de ahí se pudo identificar las principales funcionalidades que por sus características pudieron ser implementadas.

También se realizó un análisis sobre algunas librerías para crear gráficos dinámicamente en la web utilizando lenguaje PHP que pudieran servir con el fin que se pretende, de los cuales se pudo concluir que poseen algunas limitaciones, por lo que la más indicada para el uso de este sistema es PHP\_GD2.

De ahí la importancia de la presente investigación, sobre todo por constituir uno de los primeros pasos para el fomento de la creación de aplicaciones que contribuyen a la optimización de recursos de cualquier empresa de nuestro país y en especial de nuestra universidad, tan necesario en esta época que estamos viviendo donde cada vez necesitamos más y tenemos menos disponible.

Además se realizó una estimación del esfuerzo, costo y tiempo de desarrollo del sistema, permitiendo evaluar la factibilidad del mismo.

A modo de prueba se publicó en la red universitaria la aplicación web desarrollada, de manera que permitiera comprobar el funcionamiento de la misma, teniendo como resultado un nivel de aceptación de más del 80% entre los usuarios que la visitaron.

## RECOMENDACIONES

Se le recomienda a la dirección de la universidad publicar la aplicación en la intranet, para que puedan tener acceso a la misma todos los interesados y contribuir al ahorro de recursos, especialmente del combustible utilizado por los automóviles.

A los profesores de la asignatura de Investigación de Operaciones que utilicen este sistema como ejemplo de aplicación dentro de la universidad de los métodos de optimización impartidos en clases.

A futuros desarrolladores de versiones posteriores que incrementen la red de nodos existente en la base de datos, con el objetivo de lograr un mayor nivel de precisión y detalle, en la ubicación de las construcciones existentes y para mejorar de manera gráfica la trayectoria calculada.

Al Departamento Docente Central de Matemática Aplicada proponer nuevas investigaciones que hagan uso de la base de datos resultado de este trabajo, que contiene la red de trayectorias y nodos de la universidad, para resolver otros problemas como el de flujo máximo de personal o autos en casos de catástrofes o defensa del país.

## BIBLIOGRAFÍA

1. Sanchez, I. I. (2004, junio 7). informatizate.net. Retrieved noviembre 4, 2008, from informatizate.net: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
2. Jacobson. 2000. El Proceso Unificado de Desarrollo de software. 2000.
3. Larman, Craig. 2002. UML y Patrones. 2002.
4. Paradigm, V. (2007, marzo 7). Sitios de descargas de Software. Retrieved noviembre 5, 2008, from Sitios de descargas de Software: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_1472\\_0\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_1472_0_p/)
5. linuxcentro.net. (22 de febrero del 2007). linuxcentro.net. Recuperado el 7 de noviembre de 2008, de linuxcentro.net: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>
6. ciberaula. (2006). ciberaula. Retrieved noviembre 10, 2008, from ciberaula: [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/)
7. MySQL. (2008). MySQL. Retrieved noviembre 10, 2008, from MySQL: <http://dev.mysql.com/doc/refman/5.0/es/features.html>
8. DesarrolladoresWeb. (2003, Junio 4). Desarrollo Web. Retrieved Noviembre 11, 2008, from desarrolloweb: <http://www.desarrolloweb.com/articulos/1178.php>
9. Bellini M., D. I. (Septiembre de 2004). INVESTIGACIÓN DE OPERACIONES. Recuperado el 22 de 11 de 2008, de INVESTIGACIÓN DE OPERACIONES: <http://www.investigacion-operaciones.com/operaciones.htm>
10. JavaHispano, A. (11 de Noviembre de 2008). Java Hispano. Obtenido de Java Hispano: [http://www.javahispano.org/contenidos/es/mysql\\_vs\\_postgresql\\_cuando\\_emplear\\_cada\\_una\\_de\\_ellas\\_11/?menuId=MINDS&onlypath=true](http://www.javahispano.org/contenidos/es/mysql_vs_postgresql_cuando_emplear_cada_una_de_ellas_11/?menuId=MINDS&onlypath=true)

## Bibliografía

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

11. Quiñones, E. (2007). Introducción a PostgreSQL. Recuperado el 11 de Noviembre de 2008, de Introducción a PostgreSQL: [http://www.postgresql.org.pe/articles/introduccion\\_a\\_postgresql.pdf](http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf)
12. Sun Microsystems, Inc. (2008). MySQL AB. Recuperado el 21 de 11 de 2008, de MySQL AB: <http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html>
13. Buytaert, D. (2001). Drupal.org. Recuperado el 24 de 11 de 2008, de Drupal.org: <http://drupal.org/press/drupal-6.0/es>
14. Joomla, C. (01 de 2009). Características de Joomla! CMS. Recuperado el 02 de 02 de 2009, de Sitio web de la Comunidad Joomla en Chile: <http://www.joomla.cl/component/content/article/1-pasadas/12-caractericas-de-joomla-cms.html>
15. [1] S.H. Tijs and O.J. Vrieze, 'Game theory of the past decade, the contribution of the Netherlands'. In: Ten years LNMB (Eds. W.K. Klein Haneveld, O.J. Vrieze, and L.C.M. Kallenberg) CWI Tract 122, Stichting Mathematisch Centrum, Amsterdam 1997.
16. [2] J.D. Williams, 'The compleat strategyst'. McGraw Hill Book Comp. Inc., New York, 1994
17. Runeson, P., & Greberg, P. (2005). Extreme Programming and Rational Unified Process. Recuperado el 04 de 02 de 2009, de [http://serg.telecom.lth.se/research/publications/docs/282\\_runeson\\_XP\\_RUP.pdf](http://serg.telecom.lth.se/research/publications/docs/282_runeson_XP_RUP.pdf)
18. Peralta, Adriana (2003). Lb. Metodología Scrum
19. *Monografias.com* S.A. (1997). Recuperado el 17 de 05 de 2009, de *Monografias.com* S.A.: (<http://www.monografias.com/trabajos61/investigacion-operaciones/investigacion-operaciones.shtml>)
20. Abrahamsson, Salo, Ronkainen & Warsta (2002) "Agile software development methods. Review and analysis". [www.info.vtt.fi/pdf/](http://www.info.vtt.fi/pdf/)
21. Sutherland, Jeff (2001) "Inventing and Reinventing SCRUM in Five Companies"
22. Palacio, Juan (2007) Lb. "Flexibilidad con Scrum"
23. Kniberg, Henrik (2007) Lb. "Scrum y XP desde las trincheras".

## Bibliografía

“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”

---

24. Los diferentes lenguajes de programación para la web [07-02-2009]  
<http://websolutionsnic.blogspot.com/2008/09/los-diferentes-lenguajes-de-programacin.html>
25. Problemas de optimización, Método Simplex  
[http://www.phpsimplex.com/teoria\\_metodo\\_simplex.htm](http://www.phpsimplex.com/teoria_metodo_simplex.htm)
26. [www.websolutionsnic.blogspot.com](http://www.websolutionsnic.blogspot.com). *Desarrollo Web en Nicaragua*. [En línea] [Citado el: 07 de Febrero de 2009.] <http://websolutionsnic.blogspot.com/2008/09/los-diferentes-lenguajes-de-programacin.html>.
27. Pressman. (1999). *R. Software Engineering*. (F. Edition, Ed.) USA: A Practitioner's Approach.
28. Beck, K. *Extreme Programming Explained*. s.l. : Addison Wesley, 2000.
29. Beck, K. y Fowler, M. *Planeando en Programación Extrema*. 2000.
30. Davis, D. (2001). *Investigación an administración para la toma de decisiones* (5ta ed.). (I. Thomson, Ed.) Mexico.
31. Maria, U. S. (15 de mayo de 2004). *Fundamentos de Investigación de Operaciones, El Problema de Transporte*.
32. Castillo, E., Conejo, A. J., Pedregal, P., García, R., & Alguacil, N. (20 de febrero de 2002). *Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia*.
33. Davis, D. (2001). *Investigación en administración para la toma de decisiones* (5ta ed.). (I. Thomson, Ed.) Mexico.
34. Figuera, D. S. (Octubre 2002). *Métodos Cuantitativos para la toma de Decisiones*.
35. Maria, U. S. (15 de mayo de 2004). *Fundamentos de Investigación de Operaciones, El Problema de Transporte*.
36. Omaña, Z. *Manual de Investigación de Operaciones*.

## Anexos.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

# ANEXOS

**Anexo 1:** La siguiente tabla muestra algunos casos reales de organizaciones que han hecho uso de la Investigación Operativa y las ganancias y/o ahorros conseguidos a raíz de ello.

Organización	Aplicación	Año	Ahorros anuales
The Netherlands Rijkswaterstaat	Desarrollo de la política nacional de administración del agua, incluyendo mezcla de nuevas instalaciones, procedimientos de operaciones y costeo	1985	\$15 millones
Monsanto Corp.	Optimización de las operaciones de producción para cumplir metas con un costo mínimo	1985	\$2 millones
Weyerhaeuser Co.	Optimización del corte de árboles en productos de madera para maximizar su producción	1986	\$15 millones
Electrobras/CEPAL Brasil	Asignación óptima de recursos hidráulicos y térmicos en el sistema nacional de generación de energía	1986	\$43 millones
United Airlines	Programación de turnos de trabajo en oficinas de reservaciones y aeropuertos para cumplir con las necesidades del cliente a un costo mínimo	1986	\$6 millones
Citgo Petroleum Corp.	Optimización de las operaciones de refinación y de la oferta, distribución y comercialización de productos	1987	\$70 millones
SANTOS, Ltd., Australia	Optimización de inversiones de capital para producir gas natural durante 25 años	1987	\$3 millones
Electric Power Research Institute	Administración de inventarios de petróleo y carbón para el servicio eléctrico con el fin de equilibrar los costos de inventario y los riesgos de faltantes.	1989	\$59 millones
San Francisco Police Department	Optimización de la programación y asignación de oficiales de patrulla con un sistema computarizado	1989	\$11 millones
Texaco Inc.	Optimización de la mezcla de ingredientes disponibles para que los productos de gasolina cumplieran con los requerimientos de ventas y calidad	1989	\$30 millones
IBM	Integración de una red nacional de inventario de refacciones para mejorar el apoyo al servicio	1990	\$20 millones + \$250 millones en menor inventario



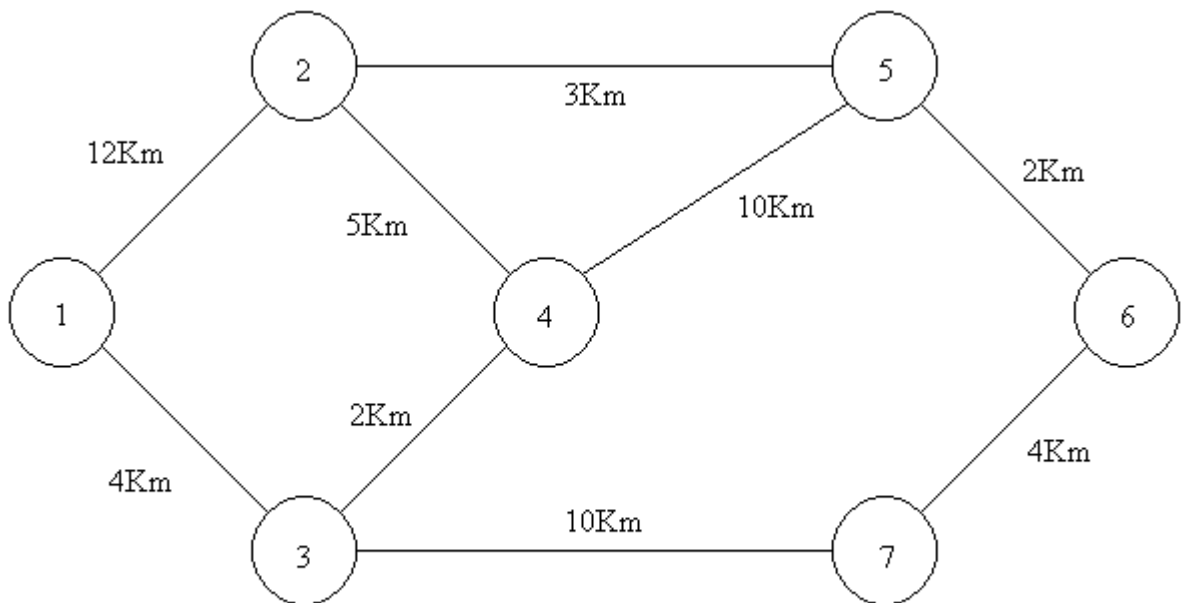
## Anexos.

*"Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal."*

U.S. Military Airlift Command	Rapidez en la coordinación de aviones, tripulación, carga y pasajeros para manejar la evacuación por aire en el proyecto "Tormenta del Desierto" en el Medio Oriente	1992	Victoria
American Airlines	Diseño de un sistema de estructura de precios, sobreventas y coordinación de vuelos para mejorar las utilidades	1992	\$500 millones más de ingresos
Yellow Freight System, Inc.	Optimización del diseño de una red nacional de transporte y la programación de rutas de envío	1992	\$17.3 millones
New Haven Health Dept.	Diseño de un programa efectivo de cambio de agujas para combatir el contagio del SIDA	1993	33% menos contagios
AT&T	Desarrollo de un sistema basado en PC para guiar a los clientes del negocio en el diseño del centro de llamadas	1993	\$750 millones
Delta Airlines	Maximización de ganancias a partir de la asignación de los tipos de aviones en 2.500 vuelos nacionales	1994	\$100 millones
Digital Equipment Corp.	Reestructuración de toda la cadena de proveedores entre proveedores, plantas, centros de distribución, sitios potenciales y áreas de mercado	1995	\$800 millones
China	Selección y programación óptima de proyectos masivos para cumplir con las necesidades futuras de energía del país	1995	\$425 millones
Cuerpo de defensa de Sudáfrica	Rediseño óptimo del tamaño y forma del cuerpo de defensa y su sistema de armas	1997	\$1.100 millones
Procter and Gamble	Rediseño del sistema de producción y distribución norteamericano para reducir costos y mejorar la rapidez de llegada al mercado	1997	\$200 millones
Taco Bell	Programación óptima de empleados para proporcionar el servicio a clientes deseados con un costo mínimo.	1998	\$13 millones
Hewlett-Packard	Rediseño de tamaño y localización de inventarios de seguridad en la línea de producción de impresoras para cumplir metas de producción	1998	\$280 millones de ingreso adicional

## Anexo 2: Solución de un problema por el método Simplex

Una persona tiene que desplazarse a diario de un pueblo 1 a otro 7. Está estudiando cuál es el trayecto más corto usando un mapa de carreteras. Las carreteras y sus distancias están representadas en la figura siguiente:



Mapa de carreteras de los pueblos 1 al 7

Se determinan las variables de decisión, en este caso:

- **X<sub>ij</sub>**: acción de desplazarse del pueblo i al j ( 0 indica que no hay desplazamiento y 1 que sí hay desplazamiento)

Se determinan las restricciones y se expresan como ecuaciones o inecuaciones de las variables de decisión. Dichas restricciones se deducen del balance entre los posibles caminos que parten desde cada pueblo y los que llegan hasta él (obviando los caminos que nos devuelvan al punto de partida y los que provengan del punto de destino):

## Anexos.

*“Sistema de apoyo a la toma de decisiones en la selección de trayectorias óptimas para el recorrido de automóviles y personal.”*

---

- Balance de caminos del pueblo 1:  $X_{12} + X_{13} = 1$
- Balance de caminos del pueblo 2:  $X_{24} + X_{25} - X_{12} - X_{42} - X_{52} = 0$
- Balance de caminos del pueblo 3:  $X_{34} + X_{36} - X_{13} - X_{43} - X_{63} = 0$
- Balance de caminos del pueblo 4:  $X_{42} + X_{43} + X_{45} - X_{24} - X_{34} - X_{54} = 0$
- Balance de caminos del pueblo 5:  $X_{52} + X_{54} + X_{57} - X_{25} - X_{45} = 0$
- Balance de caminos del pueblo 6:  $X_{63} + X_{67} - X_{36} = 0$
- Balance de caminos del pueblo 7:  $-X_{57} - X_{67} = -1$

Se expresan todas las condiciones implícitamente establecidas por la naturaleza de las variables: que no puedan ser negativas, que sean enteras, que solo puedan tomar determinados valores. En este caso las restricciones son que las variables deben ser booleanas (0 no se toma el camino, 1 se toma), y por lo tanto no pueden ser negativas:

- $X_{ij} \geq 0$
- $X_{ij}$  es booleano

Se determina la función objetivo:

Minimizar  $Z = 12 \cdot X_{12} + 4 \cdot X_{13} + 5 \cdot X_{24} + 3 \cdot X_{25} + 2 \cdot X_{34} + 10 \cdot X_{36} + 5 \cdot X_{42} + 2 \cdot X_{43} + 10 \cdot X_{45} + 3 \cdot X_{52} + 10 \cdot X_{54} + 2 \cdot X_{57} + 10 \cdot X_{63} + 4 \cdot X_{67}$

## GLOSARIO DE TÉRMINOS

- **Algoritmo:** un algoritmo (del latín, *dixit algorithmus* y éste a su vez del matemático persa al-Jwarizmi) es una lista bien definida, ordenada y finita de operaciones que se ejecutan en un tiempo finito, consumiendo una cantidad de recurso del dispositivo donde se ejecutan, y permiten hallar la solución a un problema.
- **Algoritmo o Método Simplex:** En la teoría de optimización, descubierto por el matemático norteamericano George Bernard Dantzig en 1947, es una técnica popular para dar soluciones numéricas del problema de la programación lineal. Es un método numérico para optimización de problemas libres multidimensionales, perteneciente a la clase más general de algoritmos de búsqueda. El que permite encontrar una solución óptima en un problema de maximización o minimización, buscando en los vértices del polígono.
- **Algoritmo de Dijkstra:** También llamado **algoritmo de caminos mínimos**, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo dirigido y con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959.
- **Código abierto** (en inglés *open source*) es el término con el que se conoce al software distribuido y desarrollado libremente.
- **Grafo:** Es un conjunto, no vacío, de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas (*arcs* en inglés) que pueden ser orientados o no. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas).
- **GNU/Linux** es el término empleado para referirse al sistema operativo similar a Unix que utiliza como base las herramientas de sistema de GNU y el núcleo Linux. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL de GNU (Licencia Pública General de GNU).

- **Hardware:** corresponde a todas las partes físicas y tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.
- **Hipertextos:** en informática, es el nombre que recibe el texto que en la pantalla de una computadora te conduce a otro texto relacionado. La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos
- **HTML** (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.
- **IDE** es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de usuario (Grafic User Interface siglas en inglés GUI).
- **Interfaz:** parte de un programa que permite el flujo de información entre un usuario y la aplicación, o entre la aplicación y otros programas o periféricos. Esa parte de un programa está constituida por un conjunto de comandos y métodos que permiten estas intercomunicaciones, puede ser del tipo GUI, o línea de comandos.
- **Investigación de Operaciones** o **Investigación Operativa:** es una rama de las Matemáticas consistente en el uso de modelos matemáticos, estadística y algoritmos con objeto de realizar un proceso de toma de decisiones. Frecuentemente, trata del estudio de complejos sistemas reales, con la finalidad de mejorar (u optimizar) el funcionamiento de los mismos. La investigación de operaciones permite el análisis de la toma de decisiones teniendo en cuenta la escasez de recursos, para determinar cómo se puede optimizar un objetivo definido, como la maximización de los beneficios o la minimización de costes.
- **Kent Beck:** Es el creador de la Extreme Programming, desarrollada cuando servía como líder de proyecto en el Chrysler Comprehensive Compensation Project (C3). Beck fue uno de los 17 signatarios originales del Manifiesto Ágil en el 2001. Beck ha sido pionero en el trabajo con patrones de diseño, el desarrollo orientado a pruebas y la aplicación comercial de Smalltalk. Beck

popularizó las tarjetas CRC junto a Ward Cunningham, y creó junto a Erich Gamma el framework de pruebas unitarias JUnit. Tiene una maestría en Ciencias por la Universidad de Oregon.

- **PHP** (*PHP Hypertext Pre-processor*, inicialmente PHP Tools, o, *Personal Home Page Tools*) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.
- **Programación Orientada a Objetos (POO u OOP** según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento.
- **Prueba Unitaria:** en programación, es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.
- **Pruebas de integración** son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las **pruebas unitarias**. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos.
- **Software:** la palabra «software» se refiere al equipamiento lógico o soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema «hardware».
- **SPSS** (*Statistical Package for the Social Sciences*): Es un programa estadístico informático muy usado en las ciencias sociales y las empresas de investigación de mercado. Fue creado en 1968 por Norman H. Nie, C. Hadlai (Tex) Hull y Dale H. Bent. Entre 1969 y 1975 la Universidad de

Chicago por medio de su National Opinión Research Center estuvo a cargo del desarrollo, distribución y venta del programa. A partir de 1975 corresponde a SPSS Inc. Como programa estadístico es muy popular su uso debido a la capacidad de trabajar con bases de datos de gran tamaño.

- **Vértice** (teoría de grafos): un **vértice** o **nodo** es la unidad fundamental de la que están formados los grafos. Un grafo está formado por un conjunto de vértices y un conjunto de aristas o arcos. En este contexto, los vértices son tratados como objetos indivisibles y sin propiedades, aunque puedan tener una estructura adicional dependiendo de la aplicación por la cual se usa el grafo; por ejemplo, una red carretera en un grafo los vértices representan edificaciones o ciudades.
- **Vértice** (geometría): el punto común entre los lados consecutivos de una figura geométrica, o el punto común de los dos lados de un ángulo, o el punto en que concurren tres o más planos, o el punto de una curva en que la encuentra un eje suyo normal a ella.