

004 X 600
L.

004
software
L216
Microcomputa

004.678

004

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA



**UNIVERSIDAD TÉCNICA
PARTICULAR DE LOJA**
La Universidad Católica de Loja

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

MEMORIA DE TESIS

Aplicación Web para determinar la trayectoria óptima en enlaces de
Microonda

LÍNEAS DE INVESTIGACIÓN

Telecomunicaciones y Sistemas de Información Geográfica

AUTOR

Julio Nivardo Guarderas Maldonado

DIRECTOR

Ing. Daniel Alejandro Guamán

LOJA-ECUADOR

2005

Certificación

Loja, Febrero del 2006

Ing. Daniel Alejandro Guamán
DOCENTE INVESTIGADOR DE LA ESCUELA DE CIENCIAS DE LA
COMPUTACIÓN

CERTIFICA:

Que en calidad de director, he procedido a coordinar, revisar y verificar la elaboración de la tesis de grado *Aplicación Web para determinar la trayectoria óptima en enlaces de Microonda*; presentada por el Sr. Julio Nivardo Guarderas Maldonado, de la Escuela de Ciencias de la Computación.

Por lo tanto, se autoriza su presentación, sustentación y defensa.



Ing. Daniel Guamán C.
DIRECTOR DE TESIS

Cesión de Derechos

Yo, Julio Nivardo Guarderas Maldonado, declaro conocer y aceptar la disposición del Art.67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja, que en su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de las investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través o con el apoyo financiero, académico o institucional (operativo) de la Universidad."



Julio Nivardo Guarderas Maldonado

TESISTA

Dedicatoria

Con todo mi amor, dedico este trabajo investigativo especialmente a mi madre Gina Maldonado, mi hermana Darling Guarderas, mis abuelitos Julio Maldonado y Cristina Apolo y sobre todo lo dedico a mis tíos: Jhonson, María, Ximena, Betty y Enrique Maldonado Apolo, quienes han constituido mi mayor motivación durante toda mi carrera universitaria y han luchando contra toda vicisitud anteponiendo mis necesidades a las suyas; a todos ellos por haberme guiado y apoyado en todo momento de mi vida, sin esperar nada a cambio.

Julio N. Guarderas M.

Agradecimiento

Muy sinceramente quiero agradecer a mi amigo y director de tesis Ing. Daniel Guamán quien ha contribuido arduamente y ha trabajado hombro a hombro conmigo en la realización de este proyecto.

A todos los compañeros del departamento de SIG y GESE que siempre estuvieron prestos a poner de parte para que este trabajo llegue a su feliz término.

A todos mis amigos que de una u otra forma colaboraron para que se lleve adelante este proyecto, quiero decirles que les estaré siempre agradecido y les llevaré en mis mejores recuerdos y pensamientos.

Julio N. Guarderas M.

Introducción

El estudio de radioenlaces requiere una alta precisión en los cálculos y a su vez un tedioso proceso de evaluación de todas las posibles rutas entre dos puntos conocidos. Este proceso puede tomar a un profesional entre 10 y 15 días, ya que se debe tomar en cuenta aspectos como irregularidad del terreno, área del terreno, altura de las antenas repetidoras; parámetros climatológicos como temperatura, humedad, etc. Además, otro inconveniente inherente a éste es la no disponibilidad de una herramienta flexible a cualquier orografía que facilite a los técnicos realizar estos cálculos.

Una de las necesidades del usuario es poder acceder a la aplicación desde cualquier parte del mundo a cualquier instante y cargando su propia cartografía digital tomando en consideración un formato establecido para nombrar y ubicar los respectivos valores a las geofiguras (objetos puntuales, objetos lineales).

Cuando se habla de implementar aplicaciones, se tiene diferentes opciones como aplicaciones de escritorio (que funcionan únicamente para las personas que se encuentran frente al ordenador sobre el cual está el software instalado), aplicaciones de consola (cuyo uso es complejo y limitado debido a que su entorno de trabajo es modo texto), y, aplicaciones Web (en las cuales los usuarios pueden estar geográficamente distantes del lugar donde se encuentra la aplicación y recibir todas las utilidades y facilidades de uso de una aplicación Windows común).

La principal ventaja de las aplicaciones Web es la posibilidad de compartir experiencias, tecnología, soluciones desarrolladas, e ideas.

La memoria de tesis esta conformada por cinco capítulos, incluyendo las conclusiones y recomendaciones.

Capítulo 1: Introducción a los Sistemas de Información Geográfica

Capítulo 2: Problemática actual y selección de la herramienta de desarrollo.

Capítulo 3: Diseño de la aplicación.

Capítulo 4: Pruebas de verificación y validación.

Capítulo 5: Conclusiones y recomendaciones.

**INTRODUCCIÓN A LOS
SISTEMAS DE INFORMACIÓN
GEOGRÁFICA**

1. 1 ¿Qué es un Sistema de Información Geográfica?

“Un Sistema de Información Geográfica (*SIG*) se define como una herramienta de software que permite almacenar, recuperar, analizar y desplegar información geográfica¹”.

Un SIG tiene la capacidad de efectuar una gestión completa sobre datos referenciados geográficamente (coordenadas geográficas reales), mostrados sobre geofiguras o mapas, datos alfanuméricos o descriptivos que se asocian a esos mapas para formar una base de datos integrada.

1.1.1 Elementos de un SIG. Un SIG está formado por cinco componentes que cumplen una función propia e interactúan entre sí: hardware, software, datos, personas y métodos.

1.1.1.1 Hardware. Son todos los equipos físicos que operan con un SIG como son ordenadores, periféricos de entrada y salida de datos, dispositivos de almacenamiento, navegador GPS, GPS diferencial, equipos de digitalización de mapas, etc. Actualmente estos equipos se han adaptado a diversos tipos de arquitecturas tales como cliente – servidor, Web y de escritorio.

1.1.1.2 Software. Proporciona las herramientas y funciones necesarias para almacenar, analizar y desplegar la información geográfica, para ello se necesitan elementos principales de software como herramientas para la entrada y manipulación de información geográfica, sistema de administración de base de datos (Data Base Management System), herramientas que soportan consultas, análisis y visualización de elementos geofiguras, interface gráfica de usuario.

¹ BOSQUE SENDRA Joaquín, Sistemas de Información Geográfica, Segunda edición corregida, Ediciones Rialp S.A, Madrid, 2000

1.1.1.3 Datos. Se refiere al elemento principal para lograr una correcta información. Es decir, una vez conocido el objeto del modelo del mundo real, se identifican las propiedades que lo forman, por ejemplo, los atributos que se refieren a los elementos descriptivos y el tipo de geometría como el elemento espacial.

1.1.1.4 Personas. Son las que se encargan de administrar el sistema así como de desarrollar un proyecto basado en el mundo real, entre los que se involucran analistas, desarrolladores, administradores, programadores y usuarios.

1.1.1.5 Métodos. Son los planes de un buen diseño y las normas por parte de la empresa, las cuales son modelos y prácticas de operación de cada organización.

1.2 Clasificación de las funciones de un SIG

Un SIG es, entre otras cosas, un programa de ordenador con unas capacidades específicas que se pueden resumir en algunos subsistemas o componentes lógicos. Las funciones del SIG se muestran en *el Esquema 1.1*.

1.2.1 Funciones para la entrada de información. Son los procedimientos que permiten convertir la información geográfica del formato analógico (habitual en el mundo real en forma de mapas) al formato digital que puede manejar el ordenador.

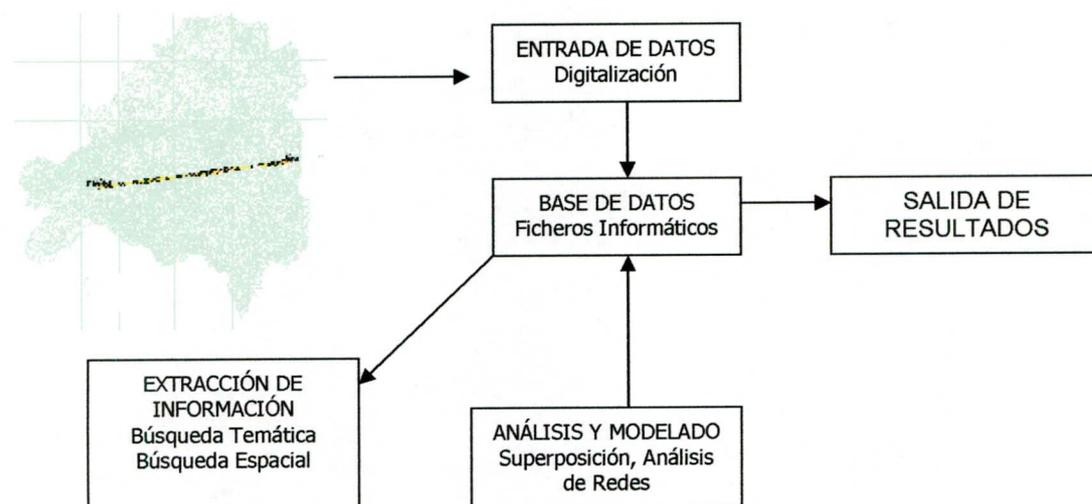
Esta conversión se debe realizar manteniendo todas las características iniciales de los datos espaciales (coordenadas Universal Transverse Mercator (UTM)) o longitudinales. Por ello, en esta función se incluyen no sólo los mecanismos de entrada propiamente dichos (digitalización o similares) sino también los procedimientos que permiten eliminar errores o redundancias en la información incorporada al SIG.

Previamente a la entrada de datos en un SIG, la información que se va a utilizar se ha debido reunir y preparar para que sea tratada y convertida al formato digital (proceso de obtención de la información).

1.2.2 Funciones para la salida / representación grafica y cartográfica de la información. Se refiere a las actividades que sirven para mostrar al usuario los propios datos incorporados en la base de datos del SIG y los resultados de las operaciones analíticas realizados sobre ellos. Permiten obtener mapas, figuras, tablas y otro tipo de resultados en diferentes medios: impresos, pantallas gráficas u otros.

1.2.3 Funciones de gestión de la información espacial. Extraen de la base de datos las porciones que interesan en cada momento y es posible reorganizar todos los elementos integrados en ella de diversas maneras.

1.2.4 Funciones analíticas. Son el elemento más importante de un SIG. Facilitan el procesamiento de los datos integrados en él, de modo que sea posible obtener mayor información y con ella mayor conocimiento del que inicialmente se disponía.



Esquema 1.1. Funciones de un SIG

1.3 Medición de la información temática

1.3.1 Variables discretas y continuas. En muchas ocasiones es importante diferenciar entre variables de carácter continuo o discreto. Una variable continua es aquella cuyas modalidades pueden adoptar infinitos valores extraídos de una escala numérica ininterrumpida. Por su parte, una variable discreta tiene modalidades que solo adoptan alguno de los números enteros posibles.

Un ejemplo de la primera es la longitud o latitud geográfica de un lugar. Una variable de tipo discreto puede ser la población de un cantón, por ejemplo 5.876 ó 5.877 habitantes pero no 5.876,5 habitantes.

Otra distinción usual es la que se refiere a las variables fundamentales y derivadas.

1.3.2 Variables fundamentales. Son las generadas directamente por el proceso de medición sin necesidad de emplear ninguna otra variable relacionada directa o indirectamente con ella.

1.3.3 Variables derivadas. Se obtienen al relacionar mediante alguna operación aritmética o similar dos o más variables fundamentales, medidas independientemente. Los mejores ejemplos son la densidad de población, porcentajes de obreros, porcentajes de jóvenes, índices de precios, índices de nivel de vida, índices de concentración urbana, etc.

1.4 Tipología de mapas en función del nivel de medida de las variables y de las dimensiones geométricas de las unidades de observación

Teniendo como base la *Tabla 1.1*, en la presente aplicación se utilizará para el diseño de mapas, el nivel de medida nominal y la dimensión geométrica por puntos (antenas y pueblos) y líneas (curvas de nivel o cotas).

Tabla 1.1. Cuadro de tipos de mapas en función del nivel de medida de las variables y de las dimensiones geométricas de las unidades de observación

Nivel de medida	Dimensiones geométricas			
	Puntos	Líneas	Áreas	Volúmenes
Nominal	Mapas de puntos. Ej.: punto casa/vértice geodésico	Mapas de redes coloreadas. Línea: mapa de carreteras.	Mapas de áreas coloreadas. Área: mapa político.	Mapas de superficies coloreadas. Mapa de zona con/sin lluvia
Ordinal	Mapa de símbolos. Tipos de ciudades.	Mapas de redes con orden. Mapa de carreteras con tipos de caminos.	Mapa de colores, ordenados. Mapa de ciudades jerarquizadas	Mapa corocromático ordenado. Mapa del orden de importancia de la lluvia.

Tabla 1.1. Cuadro de tipos de mapas en función del nivel de medida de las variables y de las dimensiones geométricas de las unidades de observación. Continuación...

Nivel de medida	Dimensiones geométricas			
	Puntos	Líneas	Áreas	Volúmenes
Cuantitativa: intervalo y razón	Mapa de símbolos graduados. Número de habitantes de cada ciudad	Mapa de flujos. Flujo de tráfico en las carreteras.	Mapa de coropletas. Mapa Político con la densidad de habitantes.	Mapa de isolíneas. Curvas de nivel de la lluvia en mm.

1.5 Búsqueda / recuperación de información de una base de datos geográfica

1.5.1 Búsqueda temática. Consiste en determinar las localizaciones de geofiguras concretas que cumplen alguno de los valores temáticos especificados por el usuario. Existen dos modalidades de búsqueda: recuperación de información mediante especificación simbólica o nominal y recuperación de información mediante condición aritmética y/o lógica referida a los atributos temáticos.

a) Recuperación de información mediante especificación simbólica o nominal

Dado el nombre de uno o varios objetos especiales, se obtienen características temáticas y/o espaciales adicionales a partir de una tabla y/o un mapa.

b) Recuperación de información mediante condición aritmética o lógica referida a los atributos temáticos

Se establece una condición aritmética o lógica que afecta a uno o varios atributos temáticos, todos los objetos espaciales que cumplan dicha condición se extraen y se representan mediante una tabla o un mapa.

1.5.2 Búsqueda espacial. La finalidad general de este tipo de operación es determinar que valor temático aparece en una localización sobre el mapa. También existen varias formas de búsqueda espacial, una de ellas es la recuperación de información mediante especificación de un dominio espacial.

Recuperación de información mediante especificación de un dominio espacial

Se establece un dominio espacial, señalando un par de coordenadas (dominio puntual) o varios pares de coordenadas (dominio lineal o poligonal), y se extraen todos los objetos espaciales que estén dentro: completa o parcialmente.

CAPÍTULO 2

PROBLEMÁTICA ACTUAL Y SELECCIÓN DE LA HERRAMIENTA DE DESARROLLO

2.1 Introducción

La planificación de radio enlaces exige un estudio muy serio de la propagación y confiabilidad del enlace, este requiere una alta precisión en los cálculos y un tedioso proceso de evaluación de todas las posibles rutas entre dos puntos conocidos, todo este proceso puede tomarle a un técnico especializado en el área de las telecomunicaciones o ciencias afines alrededor de 10 a 15 días dependiendo de la irregularidad del terreno y de la cantidad de variables consideradas como son: tamaño del área del terreno para el análisis, altura de las antenas repetidoras, parámetros climatológicos como temperatura, humedad, etc.; además, otro problema inherente a éste es la no disponibilidad de una herramienta flexible para cualquier orografía que les ayude a los técnicos a realizar estos cálculos y a su vez esté disponible desde cualquier parte del mundo a cualquier instante.

Cuando se habla de implementar aplicaciones se tiene diferentes opciones como son: aplicaciones de escritorio que funcionan únicamente para personas que se hallan frente al ordenador que contiene el software instalado; aplicaciones de consola cuyo uso es complejo y limitado; aplicaciones Web en las cuales los usuarios pueden estar geográficamente distantes del lugar donde se encuentra la aplicación y reciben todas las utilidades y facilidades de uso (botones, cuadros de texto, combos, etc.) de una aplicación Windows común.

2.2 Objetivos

2.2.1 Objetivo general

Desarrollar una aplicación Web que facilite la planificación de radio enlaces de microonda para cualquier zona del mundo, en la que el usuario podrá usar un mapa

que ya se encuentre disponible en nuestra página Web o en su defecto deberá cargar su propio mapa con un formato predefinido.

2.2.2 Objetivos específicos

- ✚ Crear una aplicación Web que solucione el problema planteado, dicha aplicación se alojará en el servidor del equipo de SIG - UTPL.
- ✚ Permitir a los usuarios utilizar cualquier mapa previamente cargado en el sitio, o en su defecto, permitirle cargar un mapa del lugar sobre el cual se desee hacer los cálculos, cabe señalar que dicho mapa será validado de acuerdo al formato preestablecido.
- ✚ Determinar las tres rutas más óptimas en base a número de saltos y distancia total en el enlace establecido, aplicando algoritmos ya probados y aceptados por profesionales en el ámbito del enlace radial como es el algoritmo de Fresnel.
- ✚ Mostrar el resultado de los diferentes cálculos y consultas como mapas, gráficos y tablas.
 - ✚ Mapas. Los mapas estarán en formato de AutoCAD (.dxf).
 - ✚ Gráficos. Los gráficos permitirán al usuario visualizar mediante un gráfico en 2D el perfil del suelo y la trayectoria de la línea de vista en relación al perfil.
 - ✚ Tablas. La mayoría de los resultados se presentan en este formato y pueden ser exportados a Excel a través del cual pueden ser impresos o almacenados en algún dispositivo.

2.3 Justificación del desarrollo de la aplicación Web

Inicialmente el proyecto denominado *PathAnalysis* surge en base a la necesidad de automatizar el proceso de análisis de radio enlaces, para lo cual fueron necesarios algunos recursos tales como:

- ✚ Topografía digital con la información necesaria para realizar análisis de radioenlaces.
- ✚ Un profesional en el ámbito de las comunicaciones con la capacidad de plantear todo el proceso de análisis de radioenlaces como un algoritmo.
- ✚ Un profesional en el ámbito del desarrollo de software capaz de comprender y entender los requerimientos del profesional en telecomunicaciones y plasmar sus ideas en un programa.

En base a los puntos mencionados anteriormente, se empieza a trabajar de manera conjunta entre los departamentos de SIG y GESE en el desarrollo del proyecto, en el mismo intervienen los profesionales en formación Patricia Ludeña y Julio Guarderas, quienes junto con el apoyo de los Docentes Investigadores de los departamentos antes en mención empiezan a trabajar de manera continua hasta obtener un resultado final, el cual es un software de escritorio denominado *PathAnalisys*, para mayor información acerca del algoritmo utilizado para el programa, características del software, requerimientos, etc., se puede consultar la tesis de grado² ***Desarrollo de un algoritmo e implementación de software para la determinación de confiabilidad de trayectoria en enlaces de Microonda, en las bandas comerciales desde 2 a 12GHz, sobre la base de la cartografía digital en formato dxf.***

Luego que la aplicación fue puesta a prueba por los estudiantes de la Escuela de Electrónica y Telecomunicaciones, se pensó en aprovechar las capacidades del Internet para que los estudiantes ingresen desde cualquier computador dentro del campus universitario y a cualquier instante a realizar tareas relacionadas al cálculo de radioenlaces, las mismas que antes les tomaba entre 10 y 15 días ahora con el software les tomaría no más de 10 minutos, la idea fue analizada y planteada como tema de tesis tomando como base al software *PathAnalisys*, es así como se empieza a elaborar la primera versión de la aplicación Web para el cálculo de radioenlaces de microonda.

Algunas de las diferencias entre las dos aplicaciones, *PathAnalisys* y *WebNetCross* se listan en la *Tabla 2.1*.

² Autor: Ing. Patricia Ludeña, Edición 2005, Universidad Técnica Particular de Loja, Escuela de Electrónica y Telecomunicaciones.

Tabla 2.1. Comparaciones entre WebNetCross y PathAnalysis

Criterio	WebNetCross	PathAnalysis
1. Funcionalidades		
a. Cargar mapas diferentes	SI	SI
b. Visualización de mapas	Especializado, diferente página.	Usa la misma pantalla que para cálculos.
c. Herramientas de visualización (Zoom, Refresh, Pan, etc.)	SI	SI
d. Guardar Imagen en formato JPG y BMP	SI	SI
e. Facilidad de trazo de nuevas rutas.	SI	SI
f. Borrado de acciones en el mapa.	SI	SI
g. Visualización de perfiles	SI	SI
h. Cálculo de las tres mejores rutas.	SI	SI
i. Ayuda(manual)	SI	SI
j. Exportar a Excel	SI	SI
k. Cálculo de confiabilidad	NO	SI
2. Ventajas y desventajas		
<i>Punto de vista del usuario</i>		
a. Validación de usuarios	Registra y valida	No valida al usuario
b. Rapidez en búsquedas dentro del mapa	Búsquedas específicas más rápidas	Busca en todo el mapa más lentas.
c. Número de usuarios	Multiusuario	Monousuario
d. Difusión del programa	Internet, Internacional	Internamente UTPL
e. Acceso a la herramienta	Disponible en Internet.	Solicitar instalador en SIG o GESE.
f. Visualización de las antenas.	Ubica el gráfico de una antena en su respectiva intersección.	No diferencia entre antenas e intersecciones.
g. Complejidad en la instalación	Compleja	Fácil
h. Configuraciones: selección de picos, antenas, etc.	No lo hace	Si lo hace
i. Velocidad en cálculos	Mayor	Normal
j. Interface gráfica	Amigable	Amigable
<i>Punto de vista del programador</i>		
a. Tipo de la aplicación	Web	De escritorio
b. Lenguaje de Programación	C#.net	VB6.0
c. Pruebas de caja negra y caja blanca.	Realizadas pero no documentadas.	Realizadas pero no documentadas.
d. Metodología	Orientada a objetos	Orientada a eventos
e. Arquitectura	En capas	No tiene
f. Control de errores	Trata cada excepción	Aborta del programa.

Tabla 2.1. Comparaciones entre WebNetCross y PathAnalysis. Continuación...

Criterio	WebNetCross	PathAnalysis
g. Estándares de código.	En declaración de variables, controles, etc.	No los tiene
h. Documentación del programa	Documentado y manual de programador	Documentado en partes importantes solamente
i. Validaciones de las entradas	Realizadas	Realizadas
j. Flexibilidad al cambio.	Mayor	Limitadamente.
k. Dependencias de terceras aplicaciones.	Requiere del Framework en el cliente.	No tiene dependencias terceras.
3. Costos de desarrollo y producción		
a. Personal involucrado	1. Julio Guarderas	1. Patricia Ludeña. 2. Julio Guarderas
b. Compra de licencias	No se adquirieron	No se adquirieron
c. Recursos de servidor	Requerido	No requerido
d. Costos de conexión a Internet	Requerido	No requerido

2.4 Funciones y algoritmos más importantes en el desarrollo de la aplicación Web

2.4.1 Función para el algoritmo de Fresnel para evaluar tramos (función Fresnel). El algoritmo de Fresnel es mundialmente conocido en el ámbito de las telecomunicaciones, específicamente en el cálculo de radioenlaces; el algoritmo basa sus cálculos en una matriz de puntos, los cuales representan el perfil del suelo, los mismos que se obtienen a partir de la intersección de la línea de transmisión con las cotas del mapa (curvas de nivel). Esta matriz es enviada a la función Fresnel como parámetro.

Por cada uno de los puntos se evalúa si se encuentra o no dentro de una zona imaginaria llamada **zona de Fresnel**, el ancho de dicha zona está en función del valor de k el cual puede tomar como valores $2/3$ ó $4/3$. Si alguno de los puntos de la ruta evaluada está dentro de dicha zona toda la ruta es descartada, pues se considera obstruida. *Ver Anexo N° 1*

2.4.2 Encontrar todas las posibles rutas entre dos puntos (Dijkstra modificado). Se denomina algoritmo de Dijkstra modificado porque no es exactamente el algoritmo de Dijkstra (da como resultado únicamente el mejor

camino desde un origen hacia un destino dado), el requerimiento del sistema es obtener las tres mejores rutas; por lo tanto, se almacena todas las posibles rutas en un *arraylist* (lista de objetos), de forma que no se pierda ninguna de ellas, luego se aplica un algoritmo de selección de las tres mejores rutas en base a todas las obtenidas, algoritmo que se explica en el *Anexo N° 2*.

2.4.3 Determinar las tres mejores rutas. El objetivo de esta función, es devolver un arreglo bidimensional de todas las rutas ordenadas en base a dos criterios de evaluación de las mejores rutas: el primer criterio es ordenar por *número de repetidoras*, pues mientras menor sea el número de repetidoras mejor es dicha ruta; el siguiente criterio de ordenación es *la distancia total de la ruta*, pues, mientras menor sea la distancia mejor es dicha ruta. *Ver Anexo N° 3*

2.4.4 Armar página wfCalculos. Para armar la página *wfCalculos* hacemos uso de la función *Response.Write (string)*, la cual recibe como parámetro un valor string conteniendo el código html necesario para ver la página, de esta manera el programa arma todo el código html en función del nombre del mapa que se desea cargar, dirección IP del servidor, referencia al control de cálculos y enlace a la página principal. Todos estos datos se arman como una sola página html y es enviada al cliente. *Ver Anexo N° 4*

2.4.5 Cargar mapa al servidor. Uno de los requerimientos de la aplicación Web es cargar al servidor cartografía digital en formato dxf de AutoCAD; al tener los mapas en el servidor el resto de usuarios accederán libremente a los mismos.

A nivel de cliente y de servidor la aplicación valida la extensión, la existencia y los nombres de los mapas, dado que el usuario puede especificar un nombre con el que se guardará el mapa en el servidor. La función que permite cargar dicho mapa en el servidor se denomina *uplTheFileDXF.PostedFile.SaveAs()* que es parte del componente html *HtmlInputFile*. *Ver Anexo N° 5*

2.5 Condiciones de selección de la herramienta para el desarrollo de la aplicación Web

Para poder llevar a cabo el desarrollo del presente proyecto, se analizó varios aspectos relacionados con el lenguaje de programación a utilizar y las herramientas, las cuales se mencionan a continuación:

2.5.1 Elaboración de los mapas

Tabla 2.2. Criterios de selección de la herramienta para la elaboración de mapas

CRITERIOS	ARCVIEW	PC ARC/INFO	AUTOCAD	MAPINFO
Permite la creación de mapas en el formato Puntual	√		√	√
Trabaja con base de datos georeferenciadas	√	√	√	√
Simula el funcionamiento de un SIG	√		√	
Manejador Interno de base de datos	√	√		√
Licencia disponible	√		√	
El resultado es en formato dxf			√	
Formato de exportación de mapa útil para usar con MapObjects.	√	√	√	
Ubicación de puntos sobre cartografía digital georeferenciada			√	

2.5.2 Selección del lenguaje de programación

Tabla 2.3. Criterios de selección del lenguaje de programación para la codificación

CRITERIOS	VISUAL Basic	C# .NET	JAVA	C++
Programación bajo entorno Win32	√	√	√	√
Tutoriales y manuales de programación con MapObjects 2.3	√	√	√	
Programación con MapObjects LT2	√	√	√	√
Fácil acceso a bases de datos	√	√		
Facilita la utilización para crear componentes de interface	√	√		
Dominio y conocimiento avanzado de la herramienta		√		
Programación Web		√	√	

2.5.3 Comparación entre MAPOBJECTS2.3, ASPMAP2.0, SPATIALLY AWARE, ARCGIS

Tabla 2.4. Criterios de selección de la herramienta para trabajar con mapas diseñados

Características/Funcionalidad	MAPOBJECTS 2.3	ASPMAP2.0	SPATIALLY AWARE	ARCGIS
Funciones de navegación por múltiples capas	√			√
Visualización de gran variedad de formatos	√	√		√
Transparencia de imágenes			√	√
Proyección dinámica	√		√	√
Consulta de ficheros en formato .dxf	√			
Soporte para ficheros en formato .dxf	√	√	√	√
Segmentación dinámica	√			
Visualización de ficheros CAD		√		
Etiquetado de entidades	√		√	√
Creación de representaciones temáticas por valores únicos, símbolos graduados	√		√	√
Filtros especiales y alfanuméricos	√		√	√
Selección de entidades mediante sentencias SQL	√		√	√
Procesamiento de conjuntos de entidades seleccionadas	√			
Conexión a bases de datos externas	√	√	√	√
Operaciones geométricas	√			
Bajo costo	√			
Manuales disponibles en Web y biblioteca	√		√	√
Componente apto para usarse con .NET	√			
Versión de evaluación superior a 90 días	√			

2.5.4 Detalle de las herramientas seleccionadas

La herramienta que se utilizará en el presente proyecto es *Visual Studio. Net* básicamente por el IDE avanzado que posee, el cual facilita trabajar con el lenguaje de programación C#.NET (utilizado para desarrollar la aplicación Web); su utilización es legal porque se cuenta con la licencia necesaria conjuntamente con una serie de manuales y documentación en la Web y en la Biblioteca de la Universidad.

Otra de las razones por la cual se usa .Net es porque se puede trabajar conjuntamente con **MapObjects2.3**, dicho componente permite visualizar los mapas en aplicaciones de escritorio, para poder trabajar con este componente en la Web se llevó a cabo una investigación acerca del uso de componentes Windows en Web, para lo cual se recomienda leer el *Anexo N° 8*.

El componente MapObjects2.3 fue de gran utilidad al desarrollar el proyecto, ya que el equipo del SIG - UTPL posee algunos manuales y referencias del mencionado componente, cuyo costo de adquisición no es elevado en relación a otros componentes para visualizar mapas en la Web, además se tuvo buenas experiencias en el desarrollo de algunas aplicaciones de escritorio con dicho componente y sus capacidades para el desarrollo SIG cada día aumentan con el lanzamiento de nuevas versiones las cuales van en beneficio de los desarrolladores.

MapObjects2.3, permite flexibilidad a los desarrolladores en sus proyectos ya que puede utilizarse en diferentes lenguajes de programación tales como: Java, C++, Visual Basic 6.0, Visual Basic.NET, C#.NET, entre otros; el componente está disponible en la Web y junto a él se asocian manuales, ejemplos y referencias con la finalidad de facilitar la programación.

Si desea conocer los precios de algunos otros componentes para visualizar mapas en Web, puede revisar la parte de bibliografía en la cual se incluyen algunos links. *Referencias Web [8], [9] y [10]*.

El motor de base de datos que se utilizará en el presente proyecto es *SQL SERVER* debido a que el servidor que alojará la aplicación Web lo tiene instalado,

además la Universidad Técnica Particular de Loja posee licenciamiento para el mismo; entre otras características, el desempeño en las transacciones con el lenguaje C#.NET es más óptimo y rápido de implementar gracias a la disponibilidad de componentes tales como SQLHelper.

2.6 Diferencias entre aplicaciones Windows y Web

A continuación se pone a consideración algunas de las diferencias entre aplicaciones Windows y Web, las cuales fueron importantes para el desarrollo del presente proyecto.

2.6.1 Instalación

El usuario quiere una instalación y actualización transparente

Windows clásico	Web
El usuario interviene en la instalación y actualización de los programas	No existe noción de programa
La instalación de un programa puede afectar el comportamiento de otros programas	Configuración del navegador (aspecto, seguridad, etc.)
El usuario interviene en la instalación y actualización de los programas	Transparente para el usuario
Problemas con el registro de clases o dll's	

Observación. Esta última característica de las aplicaciones Web ha tenido que ser sacrificada; es decir, el usuario debe realizar una instalación guiada y breve en el cliente, debido a que la aplicación WebNetCross depende de un componente llamado *MapObjects* para realizar la lectura de los mapas, dicho componente se utiliza por las razones expuestas en un ítem anterior. El *utilizar controles Windows en Web* no se lo había realizado anteriormente, este es el motivo por el cual se realizó una ardua investigación sobre el tema, empezando desde su utilización, instalación, aspectos de permisos, etc.

2.6.2 Mecanismos de navegación

Existen dos paradigmas de navegación:

✚ El paradigma “Ir a” envía al usuario a otro contexto

✚ El paradigma “Tráeme” ofrece al usuario información dentro del mismo contexto

Windows clásico	Web
Puede ofrecer ambos paradigmas	Solo ofrece el paradigma “ir a”, y el usuario puede perder la orientación
Algunos botones producen navegación y otros no	Orientado a la navegación
La respuesta es inmediata	Respuesta menos inmediata condicionada por la latencia
Mecanismos de navegación estándar definidos por guías de estilo	Solo algunos (pocos) estándares definidos: back, forward.

2.6.3 Interacción

Windows clásico	Web
Diferentes técnicas estándar de interacción: <ul style="list-style-type: none"> ✚ double-click ✚ drag-and-drop ✚ copy-paste, etc 	Más orientadas a un solo clic
Diferentes técnicas estándar de interacción	En la mayoría de los casos repercute en un cambio de contexto

2.6.4 Estilo visual

Windows clásico	Web
Estilo predefinido por el sistema	La Web ha sido siempre diseño y creatividad
Pocas veces personalizado por la aplicación	Limitado por el ancho de banda y la plataforma cliente (navegador)
	En algunos casos se da una alternativa menos atractiva pero más universal
	En las aplicaciones Web se vuelve al interface clásico (no tan rico como el actual), pero aprovechando las características de diseño de la Web

2.6.5 Ayuda

Windows clásico	Web
Sistema de ayuda con mecanismos estándares.	No existe un sistema estándar.
Menú de ayuda	Normalmente integrada en la página
Tecla F1	Las páginas se describen por sí mismas

2.6.6 Capacidad y eficacia de las tareas

Windows clásico	Web
Alto nivel de programación	Lenguaje Script más restrictivo
Limitado por los recursos de la máquina	Limitado por: <ul style="list-style-type: none">✚ La capacidad del navegador✚ Acceso a los recursos✚ Latencia, seguridad, privacidad, etc.
	La Web siempre conlleva una reducción de la eficiencia

2.6.7 Seguridad y fiabilidad

Windows clásico	Web
Ambos pueden ser más controlados	Entorno más abierto y por tanto más vulnerable.
Usuarios y fuentes suelen ser de confianza.	Si el acceso es externo será menos controlable y por tanto menos fiable.
La seguridad no ha sido un factor a tener en cuenta	La seguridad es un factor muy a tener en cuenta.
El medio está más controlado	El usuario quiere un sistema seguro con mínima intervención para conseguirla.
	Son necesarios mecanismos adicionales que ofrezcan alta disponibilidad, escalabilidad, backup, etc.

2.6.8 Desinstalación

Windows clásico	Web
La desinstalación deja residuos	No se aplica en el cliente
Puede afectar a otras aplicaciones (ficheros compartidos)	Las actualizaciones se realizan en el servidor y se aplican en el cliente de forma automática

La desinstalación del cliente de WebNetCross desafortunadamente se lo lleva de manera similar a como se realiza la instalación; debe ser efectuada por el usuario de manera guiada y breve. Está a disposición un manual para guiar al usuario tanto en la instalación como en la desinstalación.

Para que el proceso se desarrolle de manera automática hay un archivo .bat que ejecuta todas las sentencias necesarias para que la aplicación funcione sin problema.

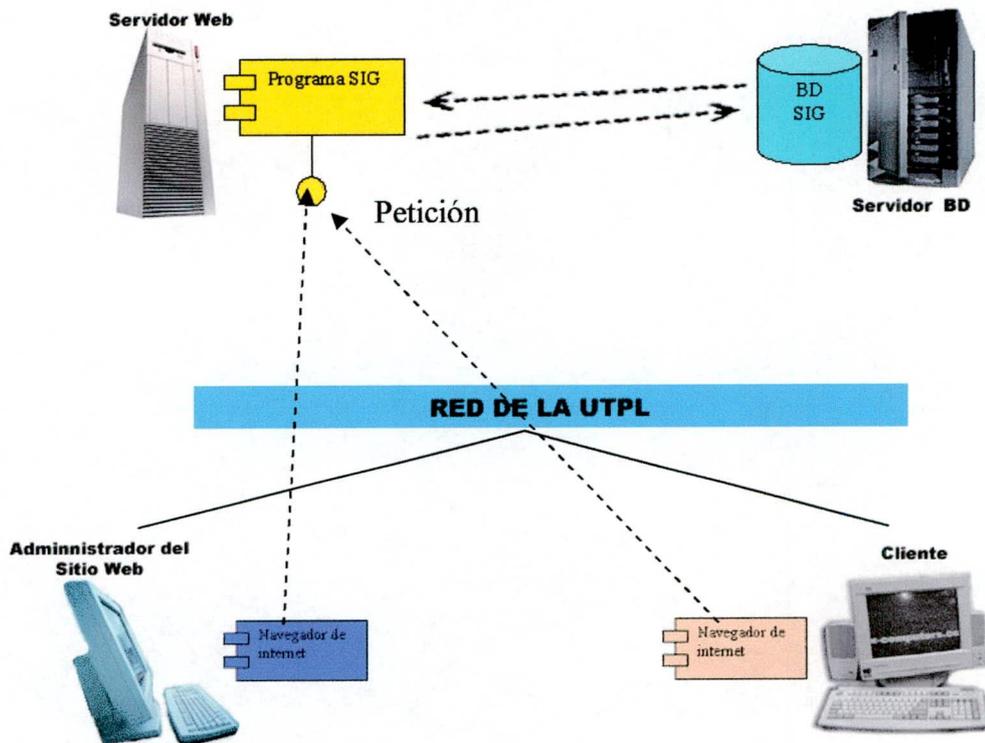
Aunque podrían ocurrir ciertos problemas si el usuario tiene instalado otras versiones de los programas requeridos, por esto en el manual se recomienda desinstalar primero dichos programas.

CAPÍTULO 3

DISEÑO DE LA APLICACIÓN

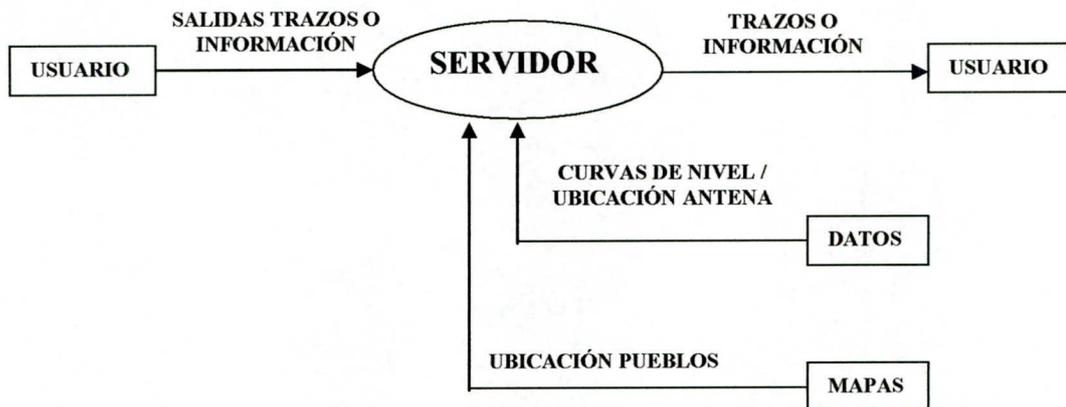
3.1 Diseño físico

La estructura del diagrama de distribución del sistema diseñado para dar solución al problema planteado se indica en el *Esquema 3.1*.

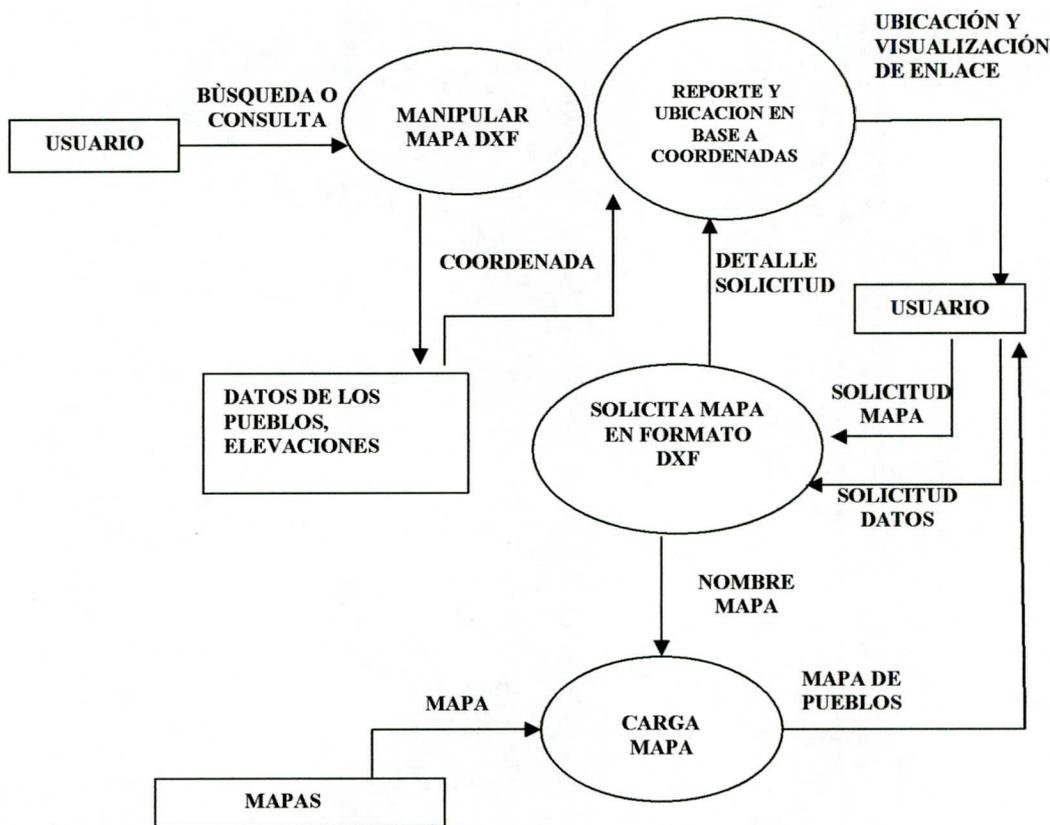


Esquema 3.1. Diagrama de distribución de componentes

A la aplicación Web se la ha podido representar en 2 niveles, los cuales se los muestran en los esquemas 3.2. y 3.3.



Esquema 3.2. Diseño de la aplicación Web Nivel 1



Esquema 3.3. Diseño de la aplicación Web Nivel 2

3.2 Diseño de salidas

El diseño de salidas se refiere a los resultados e información generados por la aplicación Web, la salida se utiliza para denotar cualquier información producida por la aplicación ya sea impresa o en pantalla.

Con el diseño de salidas se puede realizar lo siguiente: identificar la salida específica necesaria para satisfacer los requerimientos de información, seleccionar los métodos de presentación de la información y crear los documentos, reportes que contienen la información producida.

En la aplicación se han identificado dos tipos de salida: presentación de datos sobre mapas georeferenciados y presentación de resultados sobre tablas, gráficos u hojas de Excel que pueden ser almacenados en su computador.

3.3 Consultas temáticas

Las consultas temáticas consisten en presentar al usuario los diferentes objetos existentes en el mapa (pueblos, elevaciones, etc.), mediante consultas SQL transparentes para él.

La aplicación Web contiene cuatro consultas temáticas:

- ✚ Pueblos digitalizados sobre el mapa.
- ✚ Elevaciones inauguradas dentro de un área determinada.
- ✚ Cotas que se intersecan con una línea de corte.
- ✚ Una determinada elevación inaugurada o un determinado pueblo.

3.4 Consideraciones de topografía digital

La topografía digital no es más que la determinación de puntos georeferenciados (con coordenadas X e Y), obtenidos de mediciones a través de GPS's y/o fotos satelitales, con ubicación precisa de información específica de la zona (elevaciones, poblados, hidrografía, sembríos, etc.). Cabe señalar que puede preverse una amplia gama de utilización de las bases de datos topográficos y también que puede identificarse una gran variedad de información sobre superficies del terreno.

Las bases de datos topográficos digitales establecidas para predicciones de la propagación deben contener información relacionada con el tipo de predicción que va a llevarse a cabo. En frecuencias superiores a unos 30 MHz, se necesita normalmente información sobre la altura del terreno y la superficie del mismo. Para realizar predicciones de la propagación detalladas en frecuencias superiores a unos

1 000 MHz, especialmente en zonas urbanas, además de la información relativa a la altura del terreno se necesita generalmente conocer el emplazamiento, el tamaño y la orientación de las edificaciones

3.5 Formato del mapa topográfico digital

Para delimitar el rango de influencia del sistema, se utiliza justamente la topografía digital, esta cartografía utilizada se encuentra en formato *dxf*, por considerárselo de uso común a nivel mundial, en el caso de la cartografía inicial que es la de la provincia de Loja se utilizan curvas de nivel cada 200 metros, que tiene puntos georeferenciados y elevaciones denominadas *antenas inauguradas*, son cerros que en la actualidad tienen instalada una base de comunicaciones (antena de transmisión); para establecer los puntos de conexión se han referenciado también las poblaciones (capital cantonal y parroquias) a lo largo del territorio provincial; pese a contar por defecto con esta cartografía la aplicación permite trabajar con cualquier base digital y realizar las mismas operaciones para cualquier zona, basta con poner toda esta información en el formato adecuado el cual está detallado en la *Tabla 3.1*.

3.6 Coordenadas

En lo que respecta a coordenadas se ha considerado las coordenadas UTM para realizar toda la topografía, en base a los siguientes criterios:

- ✚ La región específica es relativamente de área corta por tanto las dimensiones reales no tendrán una distorsión notable al convertir los datos de un sistema de coordenadas a otro.
- ✚ Los perfiles se extraerán en pequeñas extensiones del terreno, es decir, serán trayectos relativamente cortos en los cuales una línea recta en el espacio se intercepta con una cota que representa el nivel del suelo, por tanto los datos obtenidos permitirán ser utilizados en situaciones reales.
- ✚ Las coordenadas UTM tienen una directa correspondencia a las unidades utilizadas en los cálculos.

- Actualmente, las organizaciones de regulación en telecomunicaciones exigen la presentación de datos geográficos en coordenadas UTM.

3.7 Geofiguras encontradas en la topografía

La herramienta interpreta cada una de las capas del mapa digital acorde con el formato general de leyendas, es así que podemos encontrar poblaciones, antenas, elevaciones y curvas de nivel o cotas.

3.7.1 Formato general de leyendas para la topografía digital

La topografía digital debe adecuarse a un formato legible por la aplicación, lo logra a través de sufijos en cada una de las capas para poder ser identificadas, éstas se indican en la *Tabla 3.1*.

Tabla 3.1. Formato de prefijos para topografía digital

Layer	Prefijo	Ejemplo
Población	pu-nombre	pu-Loja
Elevación inaugurada	el-nombre	el-Villonaco
Elevación(pico)	pi-nombre	pi-Picota
Cotas (m)	altura	2000

3.9 Programación por capas

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles lógicos y en caso de algún cambio sólo se ataca al nivel lógico requerido sin tener que revisar entre código mezclado.

Además permite distribuir el trabajo de creación de una aplicación por niveles lógicos, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles lógicos, simplemente es necesario conocer la *API* que existe entre niveles lógicos.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multicapas o programación por capas. En dichas arquitecturas a cada nivel lógico se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

3.9.1 Capas o niveles

3.9.1.1 La capa de presentación, está formada por los componentes de interface de usuario (IU) y los componentes de proceso de IU

Los componentes de IU pueden ser vistos como la parte con la cual interactúa el usuario, es lo que únicamente él visualiza (ventanas o páginas Web); captura la información necesaria para realizar los cálculos y presenta los resultados a éste; es posible que se aplique un mínimo de proceso realizando un filtrado previo para comprobar que no hay errores en la información ingresada (validaciones). Esta capa se comunica únicamente con la capa de negocio. Los componentes de proceso de IU podrían asociarse a clases que encapsulan la lógica de navegación y control de eventos de la interface.

3.9.1.2 La capa de negocios, encapsula la lógica de negocios

Los servicios de esta capa son encapsulados en tres tipos de componentes: Las entidades empresariales, que representan objetos que van a ser manejados o consumidos por toda la aplicación, estos podrían ser un modelo de objetos, xml, datasets con tipo, estructuras de datos, que permitan representar objetos que han sido identificados durante el modelamiento. Los otros tipos de objetos son los componentes empresariales que contienen lógica de negocio, en algunos casos al usar COM+ son los objetos raíz que inician las transacciones, es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

3.9.1.3 La capa de acceso a datos, contiene clases que interactúan con la base de datos

Estas clases surgen como una necesidad de mantener la cohesión o clases altamente especializadas que ayuden a reducir la dependencia entre las clases y capas, es donde se manejan los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Aquí podemos encontrar también una clase (SQLHelper) con métodos estáticos que permiten uniformizar las operaciones de acceso a datos a través de un único conjunto de métodos.

Todas estas capas pueden residir en un único ordenador (no sería lo normal), si bien lo más usual es que haya una multitud de ordenadores donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos, y otra serie de ordenadores sobre los cuales corre la base de datos.

Así que en una arquitectura de tres niveles, los términos capas o niveles no significan lo mismo ni son similares.

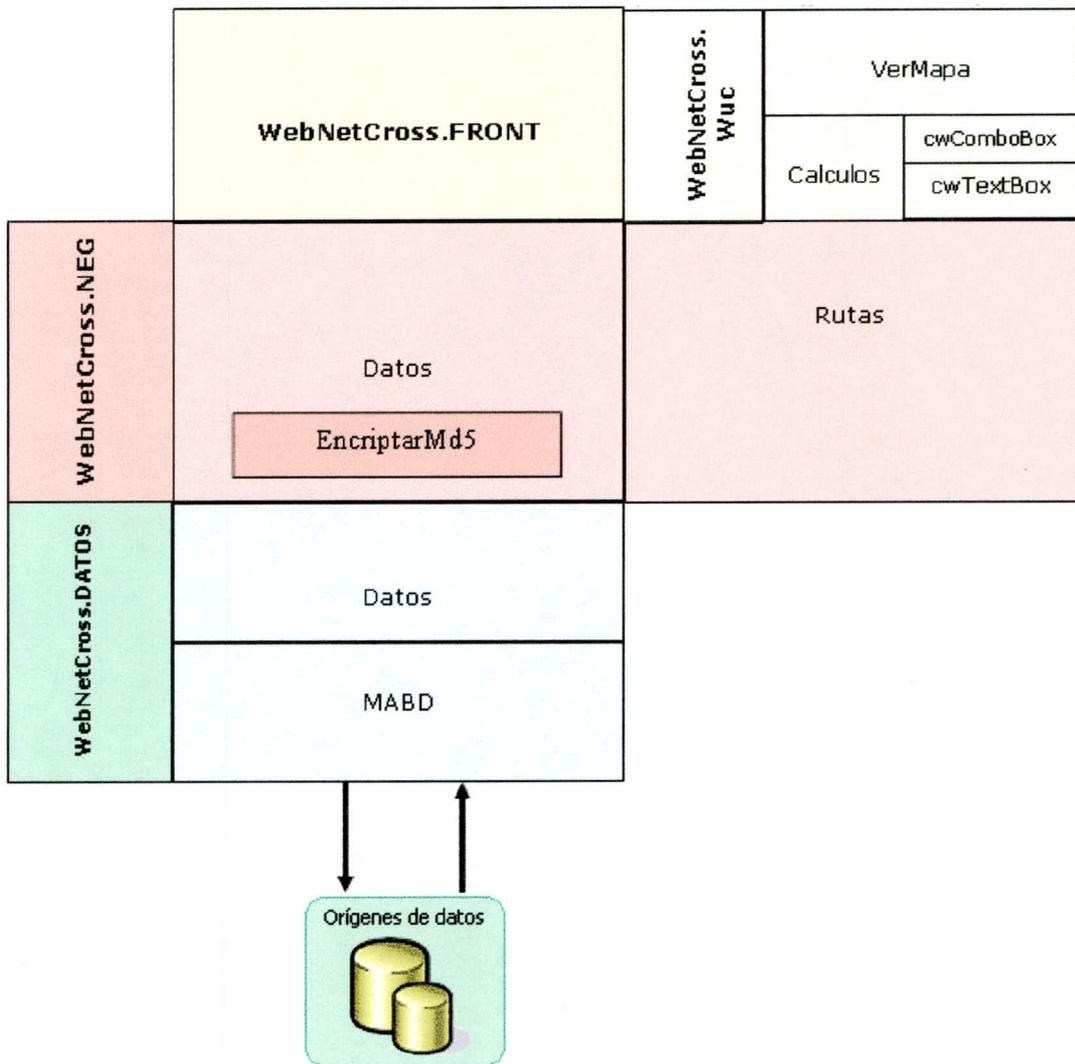
El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico:

Presentación/ Lógica de negocio/ Datos.

El término nivel, corresponde a la forma como las capas lógicas, se encuentran distribuidos de forma física. Ejemplo:

- ✚ Una solución de tres capas (presentación, lógica, datos): que residen en un solo ordenador (presentación + lógica + datos). Se dice, que la arquitectura de la solución es de tres capas y un nivel.
- ✚ Una solución de tres capas (presentación, lógica, datos): que residen en dos ordenadores (presentación + lógica, lógica + datos). Se dice, que la arquitectura de la solución es de tres capas y dos niveles.
- ✚ Una solución de tres capas (presentación, lógica, datos): que residen en tres ordenadores (presentación, lógica, datos). La arquitectura que la define es: solución de tres capas y tres niveles.

La arquitectura de la aplicación WebNetCross está representada en el *Esquema 3.4*.



Esquema 3.4. Arquitectura utilizada en la aplicación WebNetCross

Una explicación breve acerca de las 3 capas principales de la aplicación se detalla a continuación:

1. **La capa de presentación.** Está representada por las páginas Web que están en el proyecto tipo Web llamado *WebNetCross.FRONT*, aquí se muestra los resultados de los cálculos, se visualiza los mapas, se cargan los mapas al servidor, etc., es en sí el frontal con el que interactúa el usuario. Esta capa a su vez utiliza una capa llamada *WebNetCross.Wuc* donde se encuentran dos proyectos de tipo *WindowsUserControls* llamados *Calculos* y *VerMapa* que son los que permiten realizar los cálculos y visualizar el

mapa en una página Web respectivamente, el proyecto *Calculos* hace uso de dos componentes llamados *cwComboBox* y *cwTextBox* que sirven para realizar un ingreso de datos validados, en el caso de *cwComboBox* permite mediante un *ComboBox* seleccionar los nombres de las poblaciones que únicamente están disponibles y el componente *cwTextBox* filtra el ingreso de sólo números (enteros o decimales).

2. **La capa de Negocio**, que interactúa con la capa de presentación por dos frentes bien diferenciados, por un lado sirve de enlace con la capa de datos (*WebNetCross.NEG.Datos*) para permitir la validación, autenticación, registro y cambio de password de los usuarios, además es donde se encriptan los datos con el algoritmo MD5 (programado en lenguaje J#.NET), por otro lado *WebNetCross.NEG.Rutas* que al igual que el anterior es un proyecto de tipo Librería de clases las mismas que dan soporte al componente de *Calculos* para la selección de las mejores rutas.
3. **La capa de datos**, esta capa contiene un proyecto de tipo librería de clases denominado *WebNetCross.DATOS.Datos*, es el encargado de recuperar los datos desde el repositorio o a su vez de enviarlos hacia él, para hacer esto de forma transparente utiliza una clase con métodos estáticos denominada SQLHELPER ubicada dentro de una librería de clases llamada Microsoft Data Access Application Block puesta a disposición por la misma empresa Microsoft para los programadores .NET. Cabe indicar que la librería de clases *WebNetCross.DATOS.Datos* contiene una clase llamada *Acceso* que es una clase padre donde se encuentran atributos y métodos genéricos para interacción con la base de datos, los mismos que son heredados por la clase *Usuarios*.

Las pruebas son una parte importante en el proceso de desarrollo de una aplicación, pues, es en esta etapa donde se detecta y corrige todo tipo de error, lo cual permite asegurar la calidad de una aplicación.

4.1 Validación

Es el proceso de evaluación inicial de la aplicación con el objetivo de determinar si satisfará las condiciones del usuario final.

4.1.1 Validación de WebNetCross

- ✚ Como base para el desarrollo de la aplicación se tomó en cuenta el software de escritorio denominado *PathAnalysis*. Ver referencia bibliográfica [1]. La nueva aplicación Web debería tener todas las funcionalidades de éste excepto el módulo de *Confiabilidad*.
- ✚ A inicios del desarrollo de la aplicación, se llevó a cabo una ardua investigación para decidirse por un componente para la lectura de los mapas en Web, finalmente se eligió trabajar con *MapObjects2.3* el mismo que funciona sólo en aplicaciones de escritorio pues no es un componente Web, pero la investigación tuvo sus frutos cuando se encontró la forma de hacerlo trabajar en una página Web al igual que trabaja en Windows. Ver Anexo N° 8.
- ✚ Se puso en marcha la aplicación usando la cartografía digital de la provincia de Loja; se probó inicialmente con este mapa porque se conoce empíricamente que: "lo que funciona cartográficamente para Loja funciona para todo el mundo".

- ✚ Se analizó los mapas para identificar los requerimientos, formatos³ y normas que debían seguir todos los mapas con los que se quisiera que trabaje el sistema, estos requerimientos se detallan en el manual de usuario de la aplicación.

Errores que se detectaron luego del proceso de validación:

- ✚ La cartografía estaba mal digitalizada, algunos pueblos faltaban y las ubicaciones eran erróneas.
- ✚ El cliente no tenía instalado los requerimientos necesarios para el uso de WebNetCross.
- ✚ El computador en el que se instaló la aplicación Web tenía ya instalado el servidor de aplicaciones Web Apache que trabaja en el puerto 80 lo que ocasionó problemas al momento de ejecutar el IIS (Internet Information Server) que es el servidor Web requerido por la aplicación WebNetCross.
- ✚ El algoritmo de validación de las rutas inicialmente no consideraba los cambios determinados por el tipo de transmisión (digital o analógica) y los diferentes rangos de frecuencias.

Cómo se corrigieron los errores:

- ✚ La cartografía digital se la mejoró conjuntamente con el Ing. José Tamay y la Ing. Patricia Ludeña quienes laboran en el SIG y GESE respectivamente, con la finalidad de ubicar correctamente los puntos de los pueblos y de las antenas, aspectos importantes para la aplicación.
- ✚ Para resolver el problema de los requerimientos en el cliente, existe un instalador disponible en la sección de *Downloads* de la aplicación, el cual se ejecuta e instala todo lo necesario de forma transparente para el usuario.

³ Referirse al subcapítulo 3.7.1 de esta tesis.

- ✚ Para resolver el conflicto entre los servidores, se investigó acerca de la configuración del IIS y de las aplicaciones para que puedan trabajar en otro puerto, el puerto elegido fue 8088.
- ✚ En lo referente al algoritmo de validación de las rutas se pidió asesoría a la Ing. Patricia Ludeña, despejando así dudas y reprogramando el algoritmo para que se considere lo obviado anteriormente.

4.2 Verificación

Es el proceso de evaluación de una aplicación o de uno de sus componentes durante o al final del proceso de desarrollo para determinar si satisface los requisitos marcados por el usuario.

El propósito de las pruebas de verificación de una aplicación es suministrar una valoración sobre cada uno de los módulos o proceso que permite tanto al usuario y al programador identificar fallas, falta de controles y sus consecuencias.

4.2.1 Verificación de WebNetCross

En este punto se tomó en cuenta los siguientes aspectos:

- ✚ La verificación involucró el uso de la aplicación en un ambiente real, confrontando los resultados esperados por los usuarios con los obtenidos por el programa, para esto se creó una versión *Beta* que fue subida al servidor y a la que se accedió desde varios computadores en calidad de clientes.
- ✚ Se seleccionó un grupo de personas para trabajar con la aplicación Web, en un ambiente real con situaciones reales, estas personas fueron miembros del departamento del SIG y GESE y estudiantes de la escuela de Electrónica y Telecomunicaciones, luego de estas pruebas se formuló una encuesta a cada persona. *Ver Anexo 7*
- ✚ Para llevar a cabo el proceso de verificación el desarrollador no estuvo presente, esto permitió que los usuarios no se limiten de hacer sugerencias para mejorar la aplicación.

- Los usuarios estuvieron advertidos que estaban usando una aplicación de SIG el mismo que podía tener sus fallas que debían corregirse; por lo tanto, todas sus sugerencias fueron aceptadas y corregidas debidamente.

4.3 Resultados de la aplicación Web, una visión general

WEBNETCROSS es una herramienta básica de diseño para estudiantes y profesionales en la rama de telecomunicaciones, específicamente en radioenlaces de microondas, la aplicación permite realizar un análisis de radioenlaces para cualquier zona o mapa del mundo la cual está digitalizada en base a autocad.

WEBNETCROSS permite determinar rutas óptimas de comunicación en microondas y realizar el análisis de *vanos* tomando como base las dimensiones y la banda de frecuencia.

La selección de rutas se realiza por enlace, es decir, la trayectoria obtenida será una sucesión de vanos y perfiles parciales; para validar las trayectorias se utiliza la metodología de despejamiento de zona de Fresnel, y limitantes de distancia.

WEBNETCROSS utiliza topografía digital para realizar las lecturas de elevaciones, distancias y localización de lugares; por tanto es importante seleccionar mapas de alta confiabilidad para trabajar en el programa.

A continuación se expone brevemente aspectos importantes de la aplicación WebNetCross:

1. Para comenzar, debe acceder a la página inicial a través del link <http://172.16.3.96:8088/sigwebnetcross>; requerirá ingresar su login y password o registrarse como un nuevo usuario. Ver *Figura 4.1*.

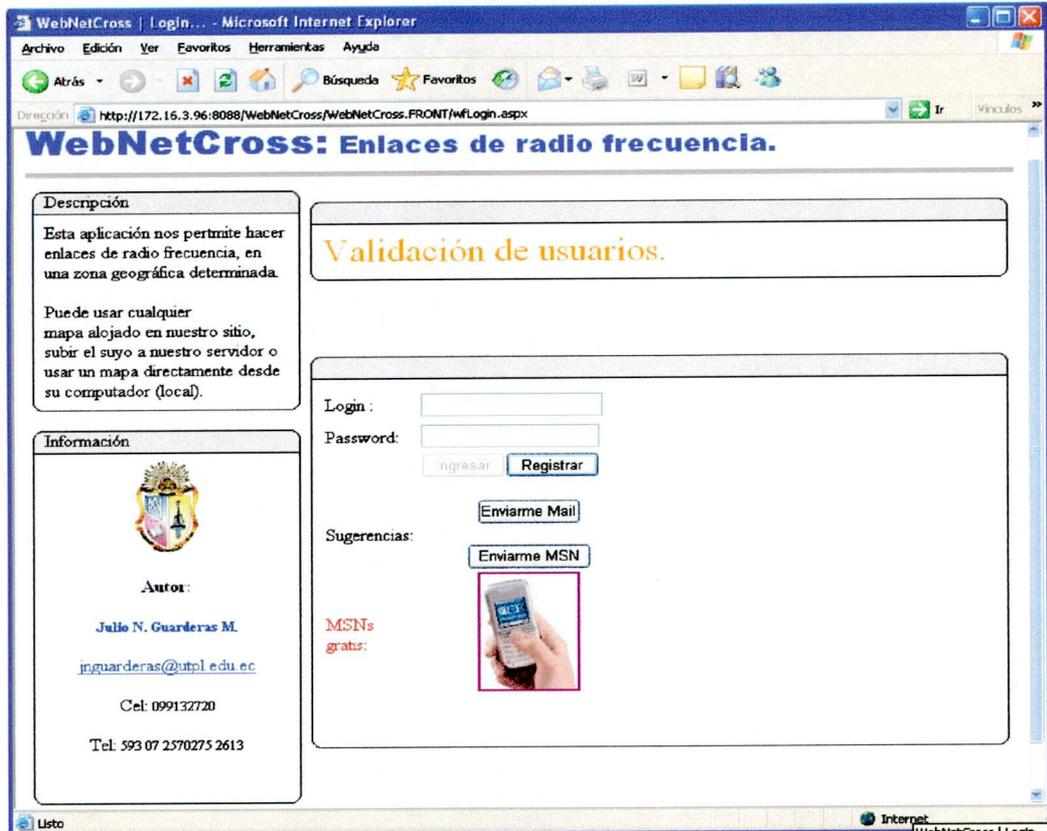


Figura 4.1. Página donde se validan los usuarios

2. Una vez que ha registrado el ingreso, se accede a la página principal, tal como se muestra en al *Figura 4.2.*

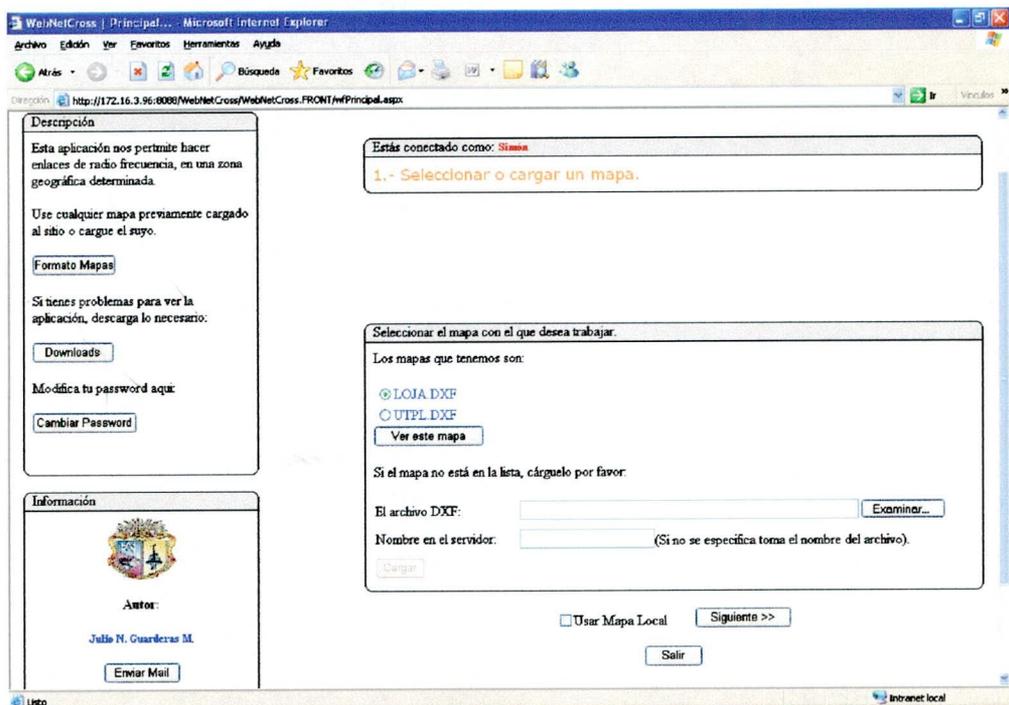


Figura 4.2. Página principal de WebNetCross.

3. Una vez que se encuentra en esta página se puede empezar a trabajar directamente con la aplicación.
4. La forma de cargar un mapa al servidor se indica en las letras rojas, como se muestra en al *Figura 4.3*.

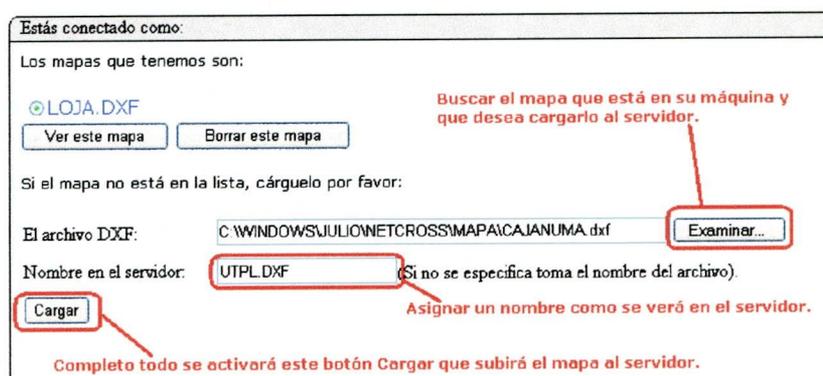


Figura 4.3. Sección para cargar los mapas.

5. Es posible visualizar un mapa que ya esté cargado en el servidor, tal como se muestra en la *Figura 4.4*, haciendo clic en *Ver este mapa*.

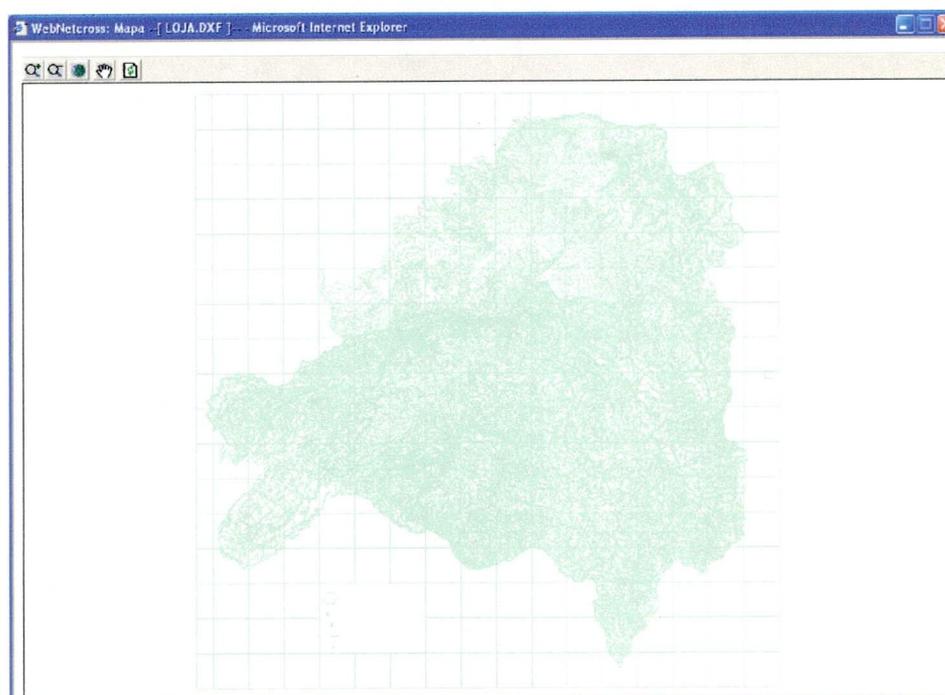


Figura 4.4. Página de visualización de los mapas.

Las opciones para visualizar de mejor manera el mapa son las siguientes.



1. Expandir



2. Reducir



3. Tamaño normal



4. Mover



5. Refrescar el mapa.

6. Una vez dentro de la aplicación, en la ventana principal se selecciona el mapa de entre la lista de mapas disponibles, luego se presiona el botón siguiente para comenzar a realizar los cálculos de radioenlaces. *Ver Figura 4.5.*

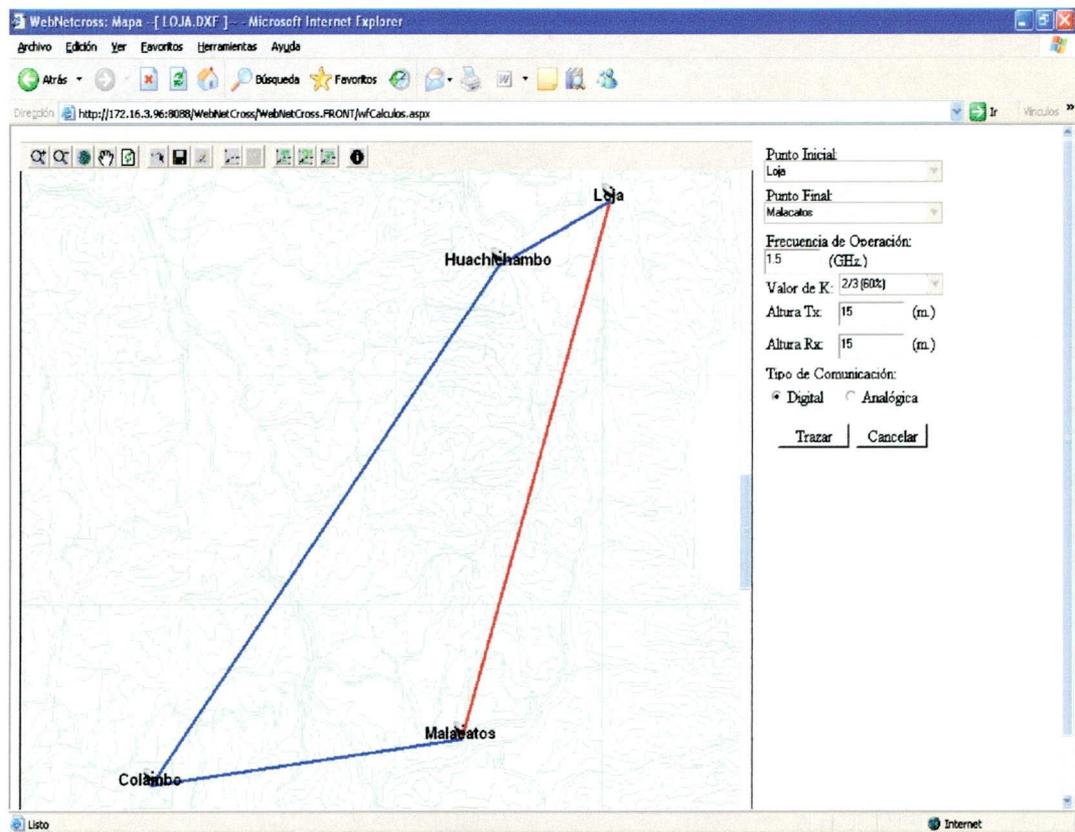


Figura 4.5. Página de cálculos de radioenlaces.

7. Como se puede observar en la imagen, la aplicación Web permite:



1. Abrir (Cuando selecciona cargar mapa localmente)



2. Expandir



3. Reducir



4. Tamaño normal



5. Mover



6. Refrescar



7. Trazar ruta



8. Guardar mapa como imagen bmp o jpg



9. Borrar



10. Perfiles



11. Cálculo de las rutas óptimas.



12. Primera mejor ruta



13. Segunda mejor ruta



14. Tercera mejor ruta



15. Información.

En la *Figura 4.5* se visualiza el resultado final de haber evaluado la ruta Loja – Malacatos, la misma que se encuentra obstruida (color rojo), por lo tanto, se ha calculado la ruta más óptima (color azul) que sugiere se envíe la señal de Loja hacia Huachichambo, de Huachichambo hacia Colambo y finalmente de Colambo se podrá enviar la señal a Malacatos.

Como un requerimiento del usuario fue que se visualice no solo la mejor ruta sino las tres mejores rutas, por razones técnicas o económicas podrían tener que elegirse la segunda o tercera, esto si permite la aplicación.

Podemos visualizar también como está el perfil de cada tramo para cada una de las rutas presionando en los botones respectivos. Lo que se visualizará es lo que muestra la *Figura 4.6*:

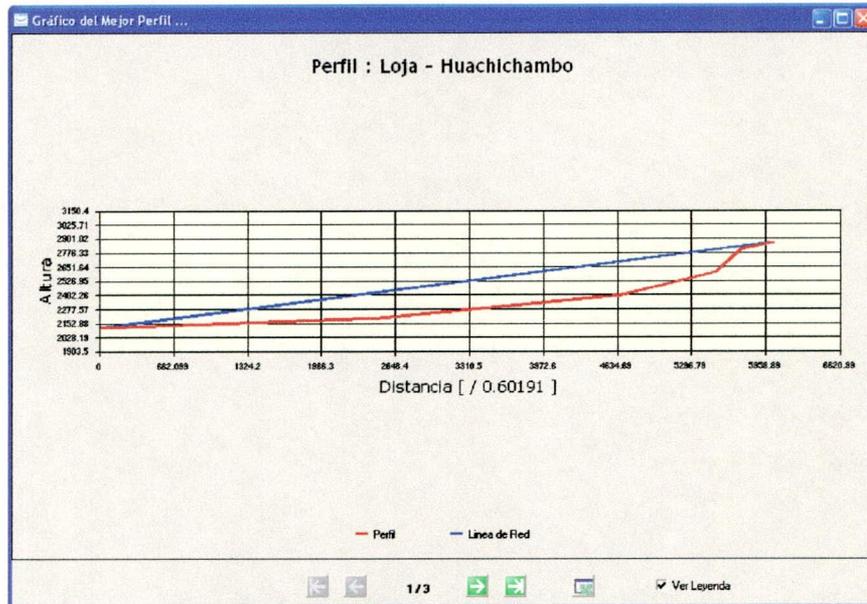


Figura 4.6. Perfil de los tramos.

La línea azul es la línea de vista o de red y la roja es el perfil del suelo, el cual es calculado en base a coordenadas georeferenciadas del mapa y de las alturas de cada cota con las que se intercepta la línea de transmisión.

Finalmente el resultado de los perfiles puede exportarse a Excel, tal como se muestra en la *Figura 4.7*:

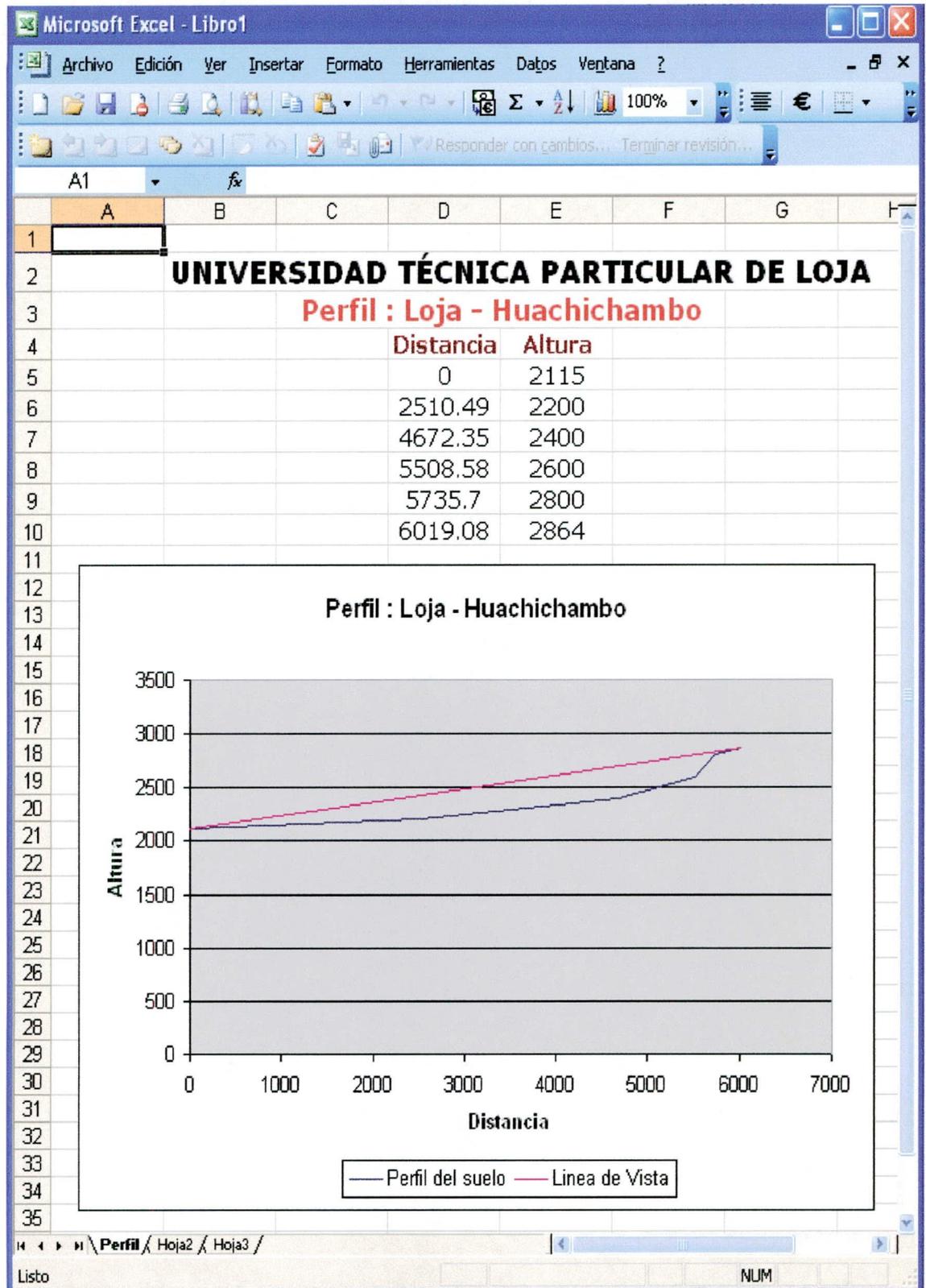


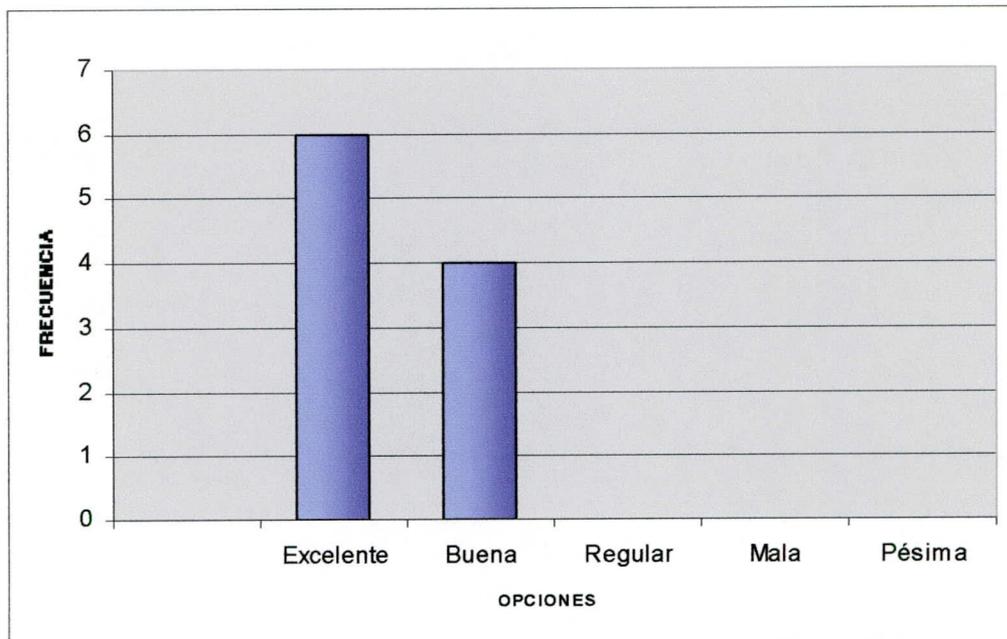
Figura 4.7. Datos exportados a Excel.

4.4 Resultados de la evaluación

En base a la encuesta realizada para la evaluación de la presente aplicación se obtuvieron los siguientes resultados.

a. ¿Cómo califica el diseño de la interface de usuario?

OPCIÓN	FRECUENCIA
Excelente	6
Bueno	4
Regular	0
Malo	0
Pésimo	0

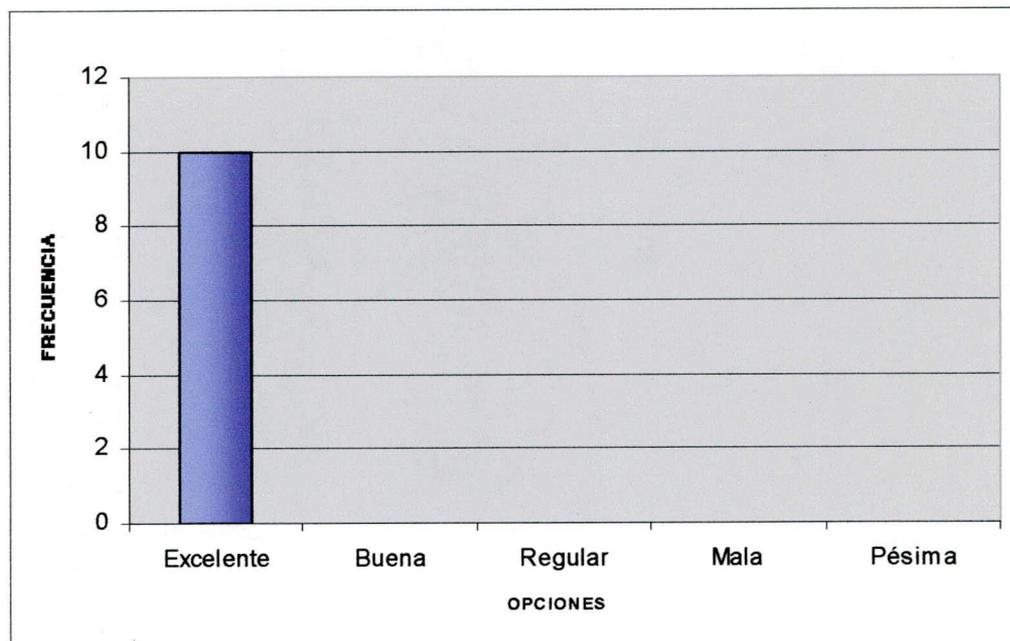


Conclusión

Según las encuestas, se puede notar que los criterios están divididos en cuanto a la calificación de la interface de usuario, entre excelente y buena, lo que hace pensar que en versiones posteriores se debe mejorar la apariencia porque la aplicación debe superar las expectativas del usuario; por tanto, en el grupo de SIG ya se ha designado a una persona para mejorar la interface.

b. ¿Cómo describe usted a la concatenación de procedimientos (navegación)?

OPCIÓN	FRECUENCIA
Excelente	10
Buena	0
Regular	0
Mala	0
Pésima	0

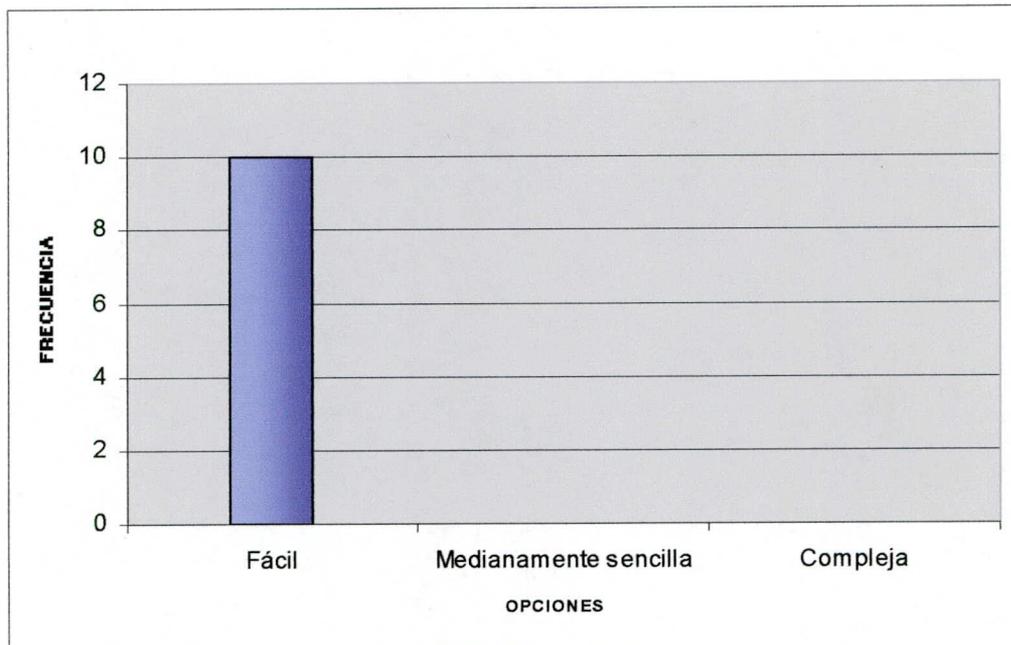


Conclusión:

En lo que se refiere a la navegación por el sitio, se puede apreciar que es del agrado del usuario, dado que a todos les parece excelente, además el tiempo que se usó para probar todas las posibilidades de navegación para facilitar las cosas al usuario ha sido exitoso.

c. ¿Cómo considera usted el ingreso de los datos requeridos para realizar los cálculos del enlace?

OPCIÓN	FRECUENCIA
Fácil	10
Medianamente sencilla	0
Compleja	0

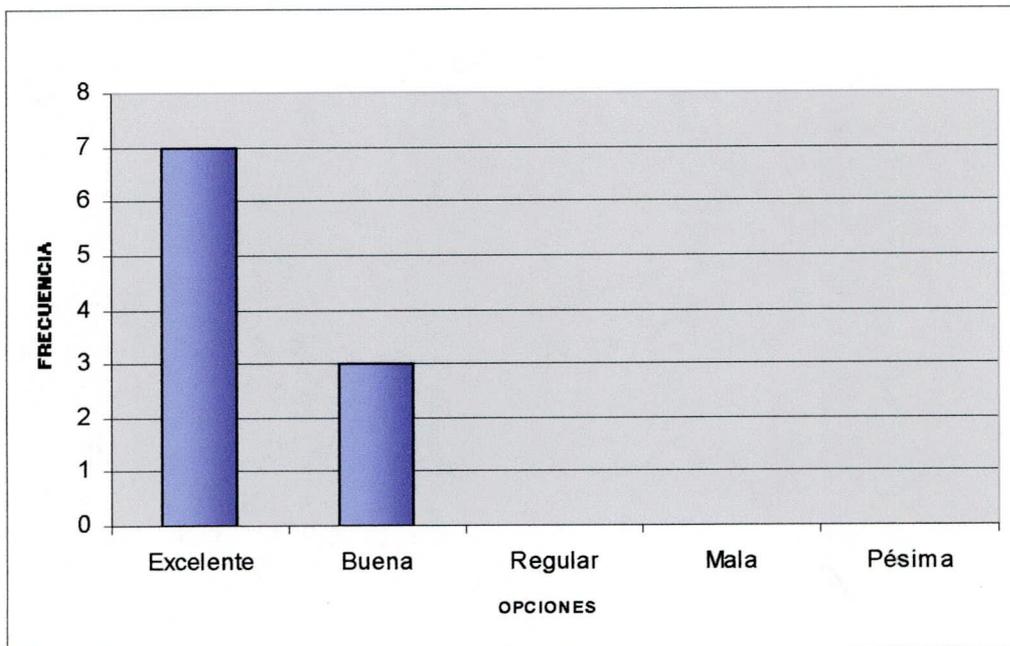


Conclusión:

El criterio utilizado al momento de programar era que el sistema debe ser hecho de manera tal que el usuario no tenga dificultades para saber lo que debe ingresar al sistema; es decir, que las entradas sean fácilmente comprendidas, que sepan con exactitud lo que se estaba requiriendo, todo esto se traduce en que el 100% de usuarios mencionó que era fácil de comprender.

d. ¿En qué medida los resultados obtenidos satisfacen sus expectativas?

OPCIÓN	FRECUENCIA
Excelente	7
Buena	3
Regular	0
Mala	0
Pésima	0

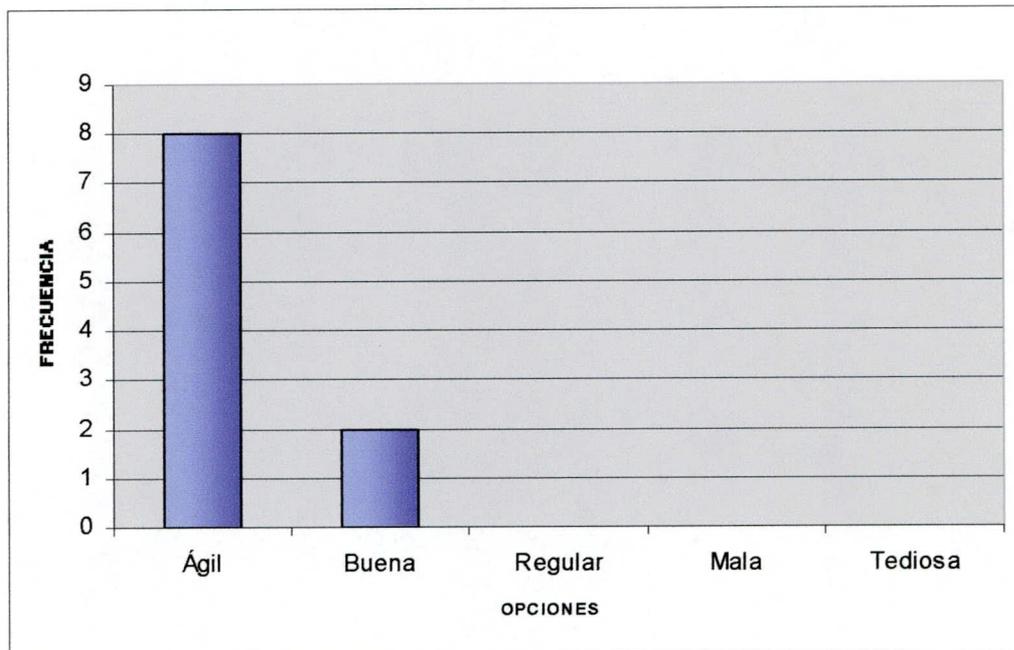


Conclusión:

Este es un punto crítico en la aplicación, pues los resultados deben ser de satisfacción completa de los usuarios, aunque el error que se cometió fue el no haber explicado de antemano que la confiabilidad no se incluía en esta versión por eso el 30% de los usuarios evaluaron que no era excelente sino únicamente buena, aunque se cumplió con las expectativas.

e. ¿Cómo califica la manipulación hacia nuevas rutas probables de enlace?

OPCIÓN	FRECUENCIA
Ágil	8
Buena	2
Regular	0
Mala	0
Tediosa	0

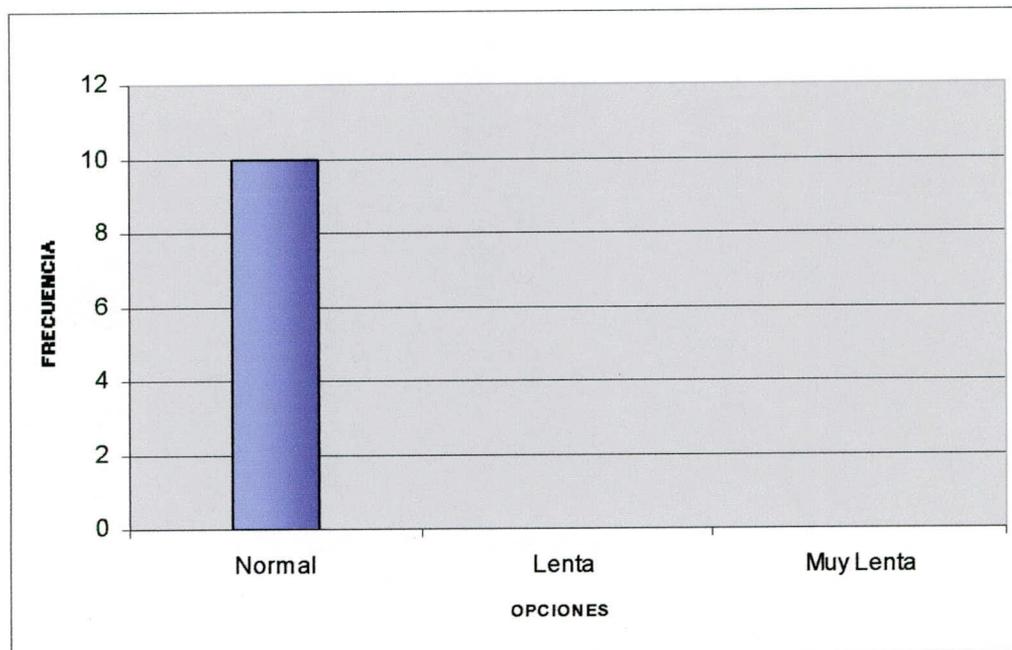


Conclusión:

Lo que se trata de conocer al formular esta pregunta es saber si el usuario considera que el trazar una nueva ruta no le daría conflictos con las anteriores, es decir, que se debía encerrar todo y que los nuevos cálculos sean fáciles de realizar. En esta parte el 80% mencionó que es ágil aunque un 20% dijo que solo era buena.

f. ¿Cómo evalúa el tiempo de respuesta al usar la aplicación?

OPCIÓN	FRECUENCIA
Normal	10
Lenta	0
Muy Lenta	0

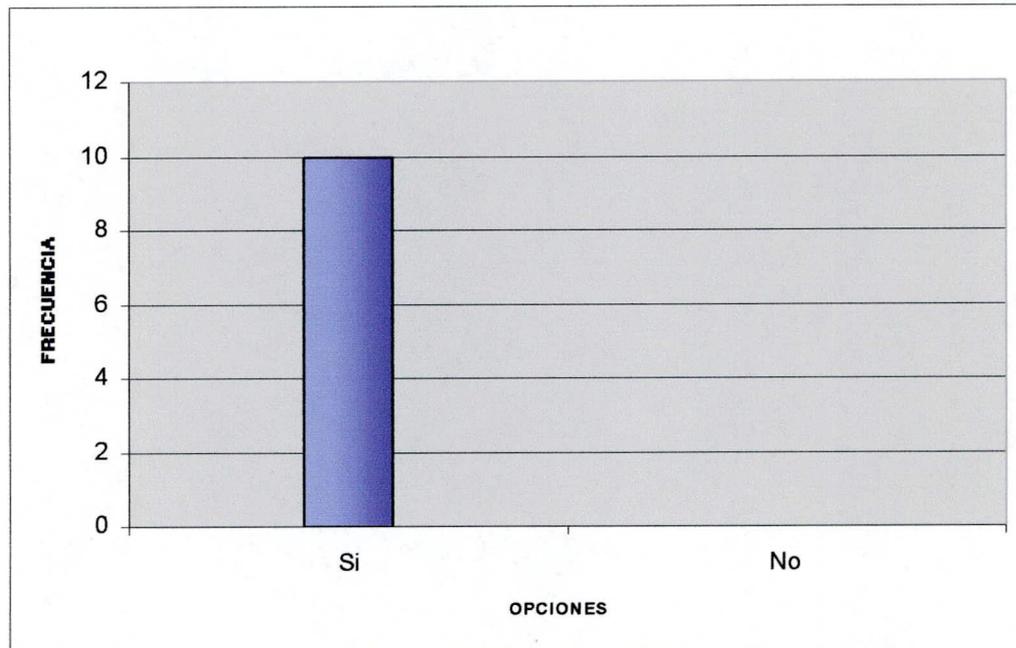


Conclusión:

Cuando se inició el proyecto lo que más se temía era precisamente la velocidad de la aplicación en Web porque de por sí una aplicación Web está limitada en su velocidad por infinitos factores como son: comunicación de datos por la red, número de accesos simultáneos al servidor, velocidad de procesamiento de página en el cliente, etc., pero lo que se hizo es optimizar al máximo cada procedimiento, resultado que se traduce en que un 100% de los usuarios hayan considerado que la velocidad de la aplicación es normal.

g. ¿Los resultados del enlace se exportan correctamente a Excel?

OPCIÓN	FRECUENCIA
SI	10
NO	0



Conclusión:

Una de las funcionalidades más importantes de la aplicación es la capacidad de poder exportar los datos a Excel, permitiendo que el usuario pueda utilizar estos resultados para futuros cálculos o en su defecto para su impresión y toma de decisiones.

5.1. CONCLUSIONES

- ✚ Son numerosas las ventajas que presenta un sistema informatizado e integrado de planificación de radioenlaces dado que reduce significativamente el consumo de recursos tanto humanos como materiales.
- ✚ Es muy ventajosa la multidisciplinariedad en un departamento, porque agiliza notablemente el reconocimiento de las variables y formatos necesarios en el desarrollo de grandes proyectos en el que intervengan profesionales de distintas ramas del conocimiento.
- ✚ Uno de los parámetros más importantes al momento de evaluar una aplicación Web es su rapidez, por tanto, se debe tomar muy en cuenta factores tales como: velocidad de conexión, optimización de los algoritmos, tiempo de respuesta del servidor, concurrencia de usuarios y características del equipo cliente.
- ✚ Es posible utilizar controles Windows en páginas Web aspx, creando un *WindowsUserControl* que es un componente personalizado (similar a los applets), pero con el inconveniente de que el usuario debe tener instalado el Framework en su computador para poder observar el mismo.
- ✚ El desarrollo de un sistema con una arquitectura en capas permite tener el código debidamente organizado en función de lo que cumpla cada capa, de esta manera reconocer rápidamente el punto del problema y resolverlo ágilmente.

- ✚ El formato utilizado para la cartografía es .dxf de AutoCAD por considerarlo de mayor uso a nivel mundial, facilita el intercambio de datos con investigadores extranjeros y garantiza que la aplicación WebNetCross funcione con cualquier cartografía.
- ✚ Para elaborar una aplicación SIG se debe tener un modelo de la situación real, que describa los objetos (pueblos, antenas, curvas de nivel) los mismos que pueden ser puntos, líneas o polígonos.
- ✚ Al utilizar *datos referenciados* podemos diseñar mapas o figuras, los cuales constan de coordenadas geográficas, datos alfanuméricos o descriptivos, que se asocian entre sí para formar una base de datos integrada y de fácil acceso, las cuales dan una perspectiva real del terreno.

5.2 RECOMENDACIONES

- ✚ Al momento de desarrollar un sistema se aconseja utilizar una arquitectura en capas, con la finalidad de facilitar su comprensión global y resolver los problemas rápidamente.
- ✚ Para hacer validaciones en el cliente se recomienda utilizar el lenguaje de programación JavaScript porque así se tiene control absoluto de todo lo que se ejecuta en el cliente, lo que se recomienda es limitar al máximo el uso de los validadores de Visual Studio.Net porque todo lo hacen transparentemente, dificultándonos tareas relacionadas a la validación.
- ✚ Para el desarrollo de todo sistema se aconseja hacer un riguroso análisis de requerimientos para cumplir los objetivos y evitar futuros inconvenientes con el usuario.
- ✚ Usar en lo posible estándares de programación tanto para nombres de variables, de funciones, de controles, etc., esto facilitará el entendimiento de su código fuente por otras personas.

- ✚ Para la elaboración de los mapas a utilizarse en el desarrollo de una herramienta similar a la presentada, coordinar con el responsable del equipo de SIG de la UTPL, para trabajar en forma conjunta en la ejecución de proyectos.

- ✚ En futuros trabajos de investigación relacionados con el tema de radioenlaces debe trabajarse en forma directa y coordinada con la(s) persona(s) responsable(s) de proporcionar la información.

- ✚ Se recomienda que la topografía utilizada sea totalmente actualizada y cuente con un 100 % de fiabilidad en sus datos; puesto que el software trabaja con los datos del mapa, no valida sus valores.

- ✚ Es imperioso que nuevas herramientas de este tipo sean realizadas por un grupo multidisciplinario, esencialmente personal de Electrónica y Telecomunicaciones y de Ciencias de la Computación.

- ✚ Es aconsejable que la topografía utilizada sea descrita mínimo en escala 1:50000 y las curvas de nivel denotadas cada 200 m para poder tener una buena resolución.

BIBLIOGRAFÍA

[1]. **Tesis:** “Desarrollo de un algoritmo e implementación de software para la determinación de confiabilidad de trayectoria en enlaces de Microonda, en las bandas comerciales desde 2 a 12 GHz, sobre la base de la cartografía digital en formato dxf.”, **Programa:** PathAnalysis, **Autor:** Patricia Ludeña.

[2]. BOSQUE SENDRA Joaquín, Sistemas de Información Geográfica, Segunda edición corregida, Ediciones Rialp S.A, Madrid.

REFERENCIAS WEB

[1]. www.elguille.com: Página de la cual obtuvimos algunos ejemplos cruciales para el desarrollo de esta tesis, por ejemplo, “Como usar controles Windows en Web”.

[2]. www.asp.net: Página dedicada a la programación con C#.NET.

[3]. www.getec.etsit.upm.es: Página que explica sobre el ciclo de vida del sistema para reconocer el más acorde a nuestras circunstancias.

[4]. msdn.microsoft.com: MSDN Online de .NET, aquí hallamos algunos ejemplos resueltos y ayudas con la descripción de algunas librerías del lenguaje C#.NET.

[5]. www.esri.com: Página Oficial de ESRI la cual posee ejemplos y utilidades para MapObjects – Windows Edition, podemos descargar las versiones de MapObjects.

[6]. www.programacionfacil.com: Conceptos básicos del desarrollo de aplicaciones Web con C#.Net.

[7]. www.portalvb.com: Página con ejemplos y código de C#.NET, posee un foro para realizar preguntas relacionadas a la programación con dicho lenguaje.

[8]. http://store.esri.com/esri/showprod.cfm?SID=2&Category_ID=121, Página donde se encuentra información y precio del software ArcGis.

[9]. <http://www.xtras.net/vendors/spatiallyaware/>: Página donde se encuentra información y precio del componente Spatiallyaware.

[10]. <http://www.vdstech.com/aspmap.htm>: Página donde se encuentra información y precio del componente AspMap.

[11].

http://www.archives.nysed.gov/a/nysaservices/ns_mgr_active_gisguides.html:
Guía para el desarrollo de aplicaciones SIG

[12]. www.gisdevelopment.net/technology: Papers, Aplicaciones y Tecnologías de Aplicaciones SIG.

[13]. www.microsoft.com: Página Web general de Microsoft, permite obtener algunos instaladores, revisar actualizaciones y bajarse algunos ejemplos útiles para la aplicación WebNetCross.

Anexo N° 1: Función Fresnel para evaluar los tramos

```
public static bool Fresnel(double[,] matrizPuntos)
{
    int totalPuntos = matrizPuntos.GetLength(0);
    double d1,d2, Rf1, valFresnel, H0,M, IncAltura;
    double distanciaTotal = Math.Abs(matrizPuntos[totalPuntos-1,0] - matrizPuntos[0,0]);
    bool bienM = true;
    double H1 = matrizPuntos[0, 1];
    double H2 = matrizPuntos[totalPuntos -1 , 1];
    int i=0;
    while(i < totalPuntos && bienM)
    {
        //d1 = el valor del punto en X
        d1 = matrizPuntos[i,0];
        //d2 = el total de la distancia menos el valor de d1
        d2 = Math.Abs(distanciaTotal - d1);
        // Calculo del radio de la primera zona de Fresnel
        Rf1 = Math.Sqrt((double)(longitudOnda * d1 * d2) / distanciaTotal);
        IncAltura = (double)((distanciaTotal * d1) - Math.Pow(d1,2)) / (2 * dblvalK * Radio);
        H0 = matrizPuntos[i, 1] + IncAltura;
        valFresnel = Math.Round(((double)((H1 * d2) + (d1 * H2)) / distanciaTotal) - H0 - ((double)(0.0784 * d1 * d2) / (dblvalK * 1000000)),5);
        if(valFresnel < 0)
            bienM = false;
        else
        {
            M = valFresnel - Rf1;
            //De acuerdo al valor de k el valor de M puede ser un porcentaje determinado de Rf1
            if(M < 0 && Rf1 != 0)
                bienM = ValidarM((double)(valFresnel * 100) / Rf1);
        }
        i++;
    }
    return bienM;
}
```

Anexo N° 2: Encontrar todas las posibles rutas entre dos puntos (Dijkstra modificado)

```
private static void findPaths(int origen, int destino, string rutaTotal)
{
    string tc;
    tc = rutaTotal;
    //Si el origen es diferente al destino
    if (origen != matrizUnos.GetLength(0) - 1)
    {
        for(int i=0; i < matrizUnos.GetLength(0); i++)
        {
            //Si es posible llegar del origen hacia el destino
            //la matriz debe tener 1
            if(matrizUnos[origen, i] == 1)
            {
                //Si no hay coincidencia, es decir si el nombre del pueblo no está aun en la ruta.
                if(tc.IndexOf(NombrePuntos[i].ToString()) < 0)
                    findPaths(i, destino, tc + "|" + NombrePuntos[i].ToString());
            }
        }
    }
    if((rutaTotal.Length + 1) > NombrePuntos[NombrePuntos.Count-1].ToString().Length)
    {
        //Agregando la ruta a la lista de rutas encontradas
        if(rutaTotal.Substring(rutaTotal.Length -
            NombrePuntos[NombrePuntos.Count-1].ToString().Length).CompareTo(NombrePuntos[NombrePuntos.Count-1]) == 0)
            listGoodPaths.Add(rutaTotal);
    }
}
```

Anexo N° 3: Determinar las tres mejores rutas.

```
private string[,] determinarLas3Mejores(ArrayList todasRutas)
{
    string[,] rutaSepTodas = new string [todasRutas.Count,NombrePuntos.Count + 2];
    int i,j;
    for(i=0; i < todasRutas.Count; i++ )
    {
        string[] rutaSeparada = todasRutas[i].ToString().Split('|');
        rutaSepTodas[i, NombrePuntos.Count] = rutaSeparada.GetLength(0).ToString();
        rutaSepTodas[i, NombrePuntos.Count + 1] = distanciaDeRuta(rutaSeparada);
        for(j = 0; j < rutaSeparada.GetLength(0); j++)
            rutaSepTodas[i, j] = rutaSeparada[j];
    }
    //Ordenar por repetidoras
    int h;
    string auxRepe;
    for(i = todasRutas.Count; i > 1; i--)
    {
        for(j = 1; j < i; j++)
        {
            if (Convert.ToInt16(rutaSepTodas[j - 1, NombrePuntos.Count]) >
                Convert.ToInt16(rutaSepTodas[j, NombrePuntos.Count]))
            {
                for(h = 0; h < (NombrePuntos.Count+2); h++)
                {
                    auxRepe = rutaSepTodas[j - 1, h];
                    rutaSepTodas[j - 1, h] = rutaSepTodas[j, h];
                    rutaSepTodas[j, h] = auxRepe;
                }
            }
        }
    }
    //Ordenar por distancias
    for(i = todasRutas.Count; i > 1; i--)
    {
```

```
for(j = 1; j < l;j++)
{
    if(rutaSepTodas[j - 1, NombrePuntos.Count] == rutaSepTodas[j, NombrePuntos.Count] &&
        Convert.ToDouble(rutaSepTodas[j - 1, NombrePuntos.Count + 1]) >
        Convert.ToDouble(rutaSepTodas[j, NombrePuntos.Count+1]))
    {
        for(h = 0; h < (NombrePuntos.Count+2); h++)
        {
            auxRepe = rutaSepTodas[j - 1, h];
            rutaSepTodas[j - 1, h] = rutaSepTodas[j, h];
            rutaSepTodas[j, h] = auxRepe;
        }
    }
}
return rutaSepTodas;
}
```

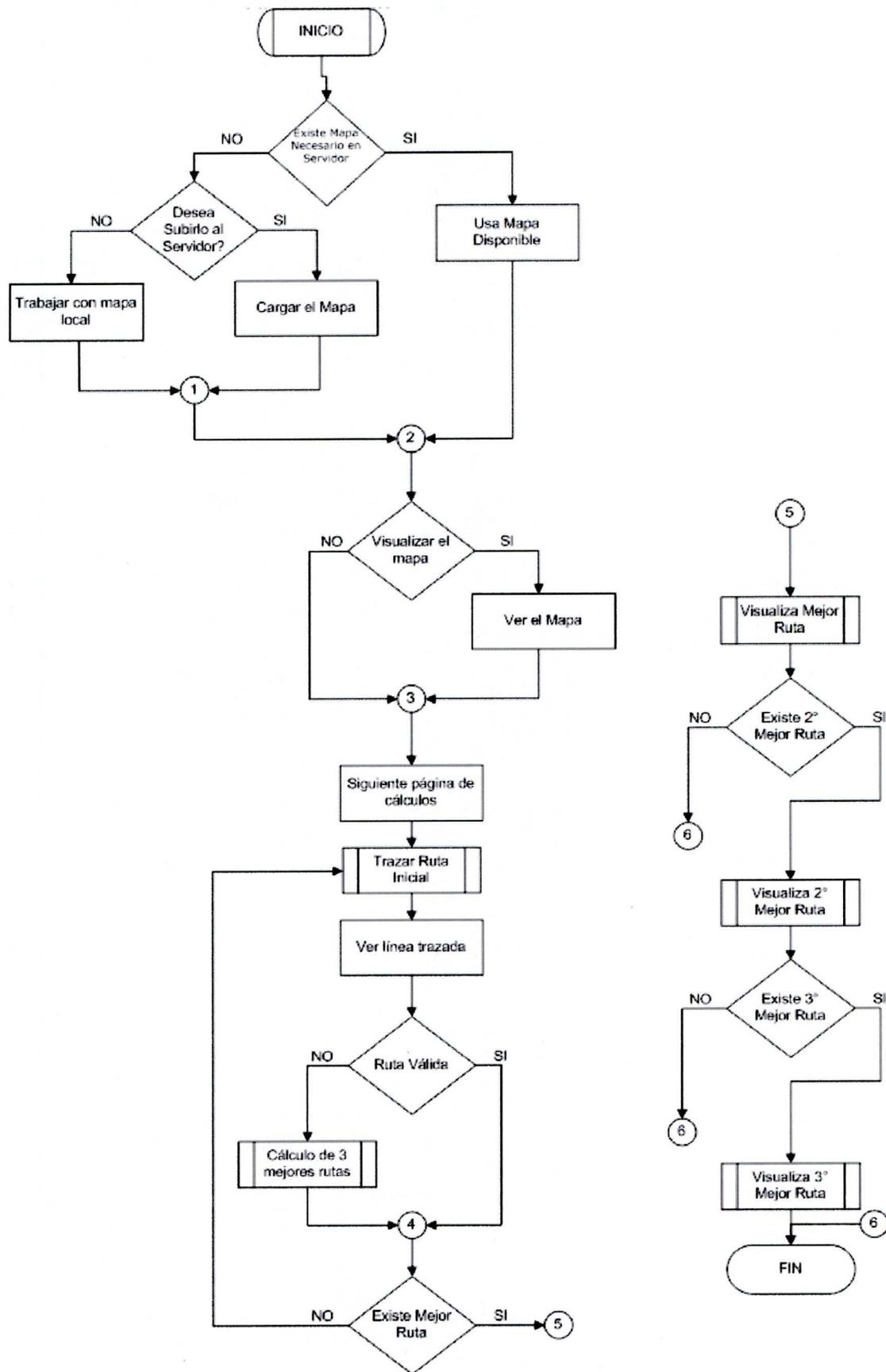
Anexo N° 4: Armar la página wfCalculos

```
private string armarPagina()
{
    if(Convert.ToBoolean(Session["mapaLocal"]))
        Session["mapaSeleccionado"] = " ---[ CARGAR UN MAPA LOCAL ]--- ";
    string ipServerWEB = ConfigurationSettings.AppSettings["Servidor_WEB"];
    string puertoWebSer = ConfigurationSettings.AppSettings["Servidor_Puerto"];
    string paginaGenerada = "";
    paginaGenerada = "<HTML><HEAD><title>WebNetcross: Mapa --[ " + Session["mapaSeleccionado"].ToString() + " ]--</title></HEAD>";
    paginaGenerada += "<BODY> ";
    paginaGenerada += "<P align=\"center\">";
    paginaGenerada += "<a href=\"wfPrincipal.aspx\"><img src=\"./Imagenes/Home.jpg\" border=\"0\"></a>";
    paginaGenerada += "<BR><center>Volver a la página principal</center>";
    paginaGenerada += "<OBJECT id=\"objMapa\" height=\"700\" width=\"1000\"";
    paginaGenerada += " classid=\"http://\" + ipServerWEB + puertoWebSer
        +"/WebNetCross/wucCalculos/calculos.dll#WebNetCross.Wuc.Calculos.wucCalculos\">";
    paginaGenerada += "<param name=\"nombreMapaCalculos\" value=\"\" + Session["mapaSeleccionado"].ToString() + "\">";
    paginaGenerada += "<param name=\"ipServidorWeb\" value=\"\" + ipServerWEB + "\">";
    paginaGenerada += "<param name=\"mapaLocal\" value=\"\" + Convert.ToBoolean(Session["mapaLocal"]).ToString() + "\"></OBJECT>";
    paginaGenerada += "</BODY></HTML>";
    return paginaGenerada;
}
```

Anexo N° 5: Cargar el mapa al servidor

```
private void btnUploadTheFile_Click(object sender, System.EventArgs e)
{
    txtOutput.Visible = false;
    string strFileNameOnServer = txtServername.Value;
    string strBaseLocation = Server.MapPath(@"../Mapas/");
    if (uplTheFileDXF.PostedFile != null && (uplTheFileDXF.PostedFile.ContentLength > 0))
    {
        if (strFileNameOnServer == "")
            strFileNameOnServer = mngCadenas.dividirRuta(uplTheFileDXF.Value, @"\");
        if (!System.IO.File.Exists(strBaseLocation+strFileNameOnServer))
        {
            if (mngCadenas.dividirRuta(uplTheFileDXF.Value, @".").ToUpper() == "DXF")
            {
                try
                {
                    uplTheFileDXF.PostedFile.SaveAs(strBaseLocation+strFileNameOnServer);
                    txtOutput.Text = "Archivo: <font color=\"blue\"><b>" + strFileNameOnServer+"</b></font> satisfactoriamente cargado.";
                    txtServername.Value = "";
                    CargarNombresMapas();
                }
                catch (Exception ex)
                {
                    mngArchivos.borrarArchivos(strBaseLocation, strFileNameOnServer);
                    txtOutput.Text = "Error al Guardar <font color=\"blue\"> <b>" + strFileNameOnServer+"</b></font>";
                }
            }
            else
            {
                txtOutput.Text = "Por favor, verificar que la extensión del mapa sea dxf.<b>";
            }
        }
        else
        {
            txtOutput.Text = "Por favor, ponga otro nombre <font color=\"blue\"> <b>[\" + strFileNameOnServer + "\"]</b></font> ya existe.<b>";
        }
    }
    else
    {
        txtOutput.Text = "Por favor, revise el <font color=\"blue\"> <b> archivo DXF es inválido.</b></font><b>";
    }
    txtOutput.Visible = true;
}
```

Anexo N° 6: Flujograma para el desarrollo del proyecto



Anexo N° 7: Encuesta para validación de la Aplicación Web

Estimado Usuario:

La presente encuesta tiene como finalidad determinar el nivel de aceptación de la aplicación Web que permite determinar la trayectoria óptima en enlaces de Microonda, para lo cual rogamos se digne elegir el criterio que más se ajuste al ítem correspondiente.

1. ¿Cómo califica el diseño de la interface de usuario?
 - a. Excelente
 - b. Buena
 - c. Regular
 - d. Mala
 - e. Pésima

2. ¿Cómo describe usted a la concatenación de procedimientos (navegación)?
 - a. Excelente
 - b. Buena
 - c. Regular
 - d. Mala
 - e. Pésima

3. ¿Cómo considera usted el ingreso de los datos requeridos para realizar los cálculos del enlace?
 - a. Fácil
 - b. Medianamente sencilla
 - c. Compleja

4. ¿En qué medida los resultados obtenidos satisfacen sus expectativas?
 - a. Excelente
 - b. Buena
 - c. Regular
 - d. Mala
 - e. Pésima

5. ¿Cómo califica la manipulación hacia nuevas rutas probables de enlace?
 - a. Ágil
 - b. Buena
 - c. Regular
 - d. Mala
 - e. Tedios

6. ¿Cómo evalúa el tiempo de respuesta al usar la aplicación?
 - a. Normal
 - b. Lenta
 - c. Muy Lenta

7. ¿Los resultados del enlace se exportan correctamente a Excel?
 - a. Si
 - b. No

Anexo N° 8: Cómo usar componentes Windows en Web ⁴

Primeramente creamos un control Web, una vez creado y asignado el nombre que deseemos a nuestro control creamos un atributo nuevo que en mi caso le llamaré ParamPasado y que será de tipo string.

```
private string pasado;  
public string ParamPasado  
{  
    get{ return pasado;}  
    set{this.pasado = value ;}  
}
```

Con esto ya hemos creado el atributo que vamos a inicializar, ahora veremos como inicializarlo desde un tag de Html. Como sabéis usábamos object para incluir el control en la Web, pues bien entre el inicio y el cierre incluimos lo siguiente.

```
<param name="ParamPasado" value="esto es el parametro pasado en el tag param">
```

Bien, con esto ya podemos inicializar los controles y que tal y como comente antes nos facilitará mucho la reutilización de los controles en diversas páginas Web. Vamos entonces con el plato fuerte, tomarlo con calma. Nuestra tarea es conseguir que un evento soltado por nuestro control sea capturado por un script de una página Web, por lo menos eso fue lo que dije yo cuando me observé en la situación de tener que crear un control que necesitaba sincronizarse con una página en Html. Bueno después de leer bastante de la SDK de .NET Framework y diversas consultas, entre ellas al Guille y a Erich Bühler (<http://www.vblibros.com/>), la solución que he encontrado es exponer el Control Windows como un objeto COM declarando los eventos y miembros públicos del control y definiendo la interface adecuada. Para ello necesitaba, como sabréis los que tengáis algo de idea de COM es un identificador único para eso objeto, esto no es problema puesto que Visual Studio nos proporciona esta utilidad, para crear uno podéis ejecutar Guidgen en el símbolo del sistema que viene con el Visual Studio, o directamente en el Visual Studio dirigiéndonos a Herramientas -> crear Guid, nos saldrá algo como lo que se muestra en la *Figura 8.1*:

⁴ <http://www.elguille.info/colabora/puntoNET/unaiWebControl2parte.htm>

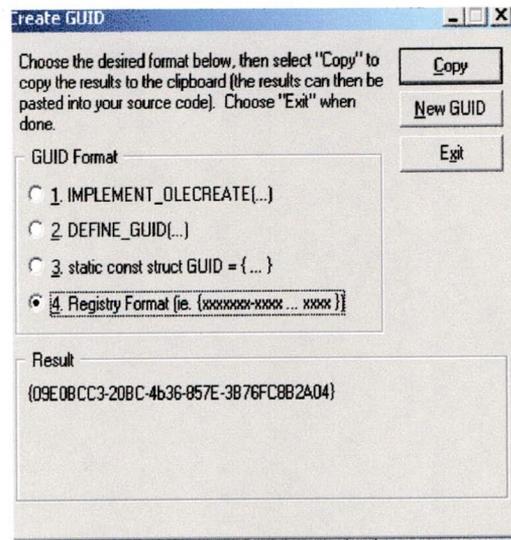


Figura 8.1. Ventana de la aplicación Create GUID

Seleccionamos la opción 4 y hacemos click en Copy, ¿fácil no?, bueno pues con esto hemos creado un identificador único en el mundo que servirá para identificar a nuestro objeto. Bueno ya tenemos nuestro guid, guardarlo en un documento de texto para usarlo más tarde ok?

Para crear un evento primero necesitamos crear un delegado y después asociar al evento este delegado, que básicamente es un puntero a una función. El código para crear el evento es el siguiente.

```
public delegate void TerminarHandler();  
public event TerminarHandler Terminado;
```

Ahora que hemos visto como crear el Guid del control y como crear un evento estamos en disposición de crear definitivamente el control. Escribimos el código y después pasamos a comentarlo.

```

using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Windows.Forms;
using System.Runtime.InteropServices;
namespace controltest2
{
    /// Descripción breve de UserControl1.
    [ClassInterface(ClassInterfaceType.None), ComSourceInterfaces(typeof(IQCDCOMEvents))]
    public class micontrol2 : System.Windows.Forms.UserControl, IQCDCOMIncoming // esto no aparece en el código pero lo pongo para explicar alguna cosa
    {
        private System.Windows.Forms.Button bt1;
        /// Variable del diseñador requerida.
        private System.ComponentModel.Container components = null;
        public delegate void FinHandler();
        public event FinHandler OnFin;
        private string pasado;
        public string ParamPasado
        {
            get{return this.pasado;}
            set{this.pasado = value;}
        }
        public micontrol2()
        {
            // Llamada requerida por el Diseñador de formularios Windows.Forms.
            InitializeComponent();
            // TODO: agregar cualquier inicialización después de llamar a InitForm
        }
        ///
        /// Limpiar los recursos que se estén utilizando.
        ///
        protected override void Dispose( bool disposing )
        {
            if( disposing )
            {
                if( components != null )
                components.Dispose();
            }
            base.Dispose( disposing );
        }
    }
    #region Component Designer generated code

```

```

///
/// Método necesario para admitir el Diseñador, no se puede modificar
/// el contenido del método con el editor de código.
///
private void InitializeComponent()
{
this.bt1 = new System.Windows.Forms.Button();
this.SuspendLayout();
// bt1
this.bt1.Location = new System.Drawing.Point(24, 64);
this.bt1.Name = "bt1";
this.bt1.TabIndex = 0;
this.bt1.Text = "Aceptar";
this.bt1.Click += new System.EventHandler(this.bt1_Click);
// micontrol2
this.Controls.AddRange(new System.Windows.Forms.Control[] {
this.bt1});
this.Name = "micontrol2";
this.ResumeLayout(false);
}
#endregion
private void bt1_Click(object sender, System.EventArgs e)
{
bt1.BackColor = System.Drawing.Color.Blue;
bt1.Text = ParamPasado.ToString();
//Lanzamos el evento, si es lanzado y no se implementa saltará una excepción por lo que para algo serio se debería cachear
OnFin();
}
}
#region exponemos los eventos
[Guid("2BA61129-37D3-4b84-BA47-020F7493A5A3")]
[InterfaceType(ComInterfaceType.InterfaceIsIDispatch)]
public interface IQDCOMEvents
{
[DispId(0x60020000)]
void OnFin();
}
//Necesitamos declarar aqui lo publico de la clase, por lo de el rollo de la especificación de COM
public interface IQDCOMIncoming
{string ParamPasado{get;set;}}
#endregion
}

```

Explicación del código

Como puede verse es necesario incluir el espacio de nombres (namespace) `System.Runtime.InteropServices` para exponer el control como COM. Usamos `ClassInterface` para especificar que deseamos exponer el control como COM, se le pasa como parámetro la interface `IQDCOMEvents` para decirle cuales son los eventos que pueden ser tratados en la Web. La clase del control se puede derivar de `IQDCOMIncoming` para que así mismo podamos decirle que atributos o métodos pueden ser llamados desde `jscrip` o `vbscrip`, esto no lo he hecho en el control pero bueno aquí lo explico. Una vez creado esto fuera del cierre de la clase se crea la interface que he mencionado y aquí es donde se usa lo que habíamos aprendido del GUID, para la interface `IQDCOMEvents`, si se crean varios eventos a tratar deben tener un identificador distinto dentro del control, en este caso solo necesitamos que sean distintos dentro del objeto yo uso `0x60020000` y para los siguientes eventos pues `0x60020001` etc.

Fijaros como se declara `ParamPasado` en `IQDCOMIncoming`, ¡ Cuidado los atributos como `ParamPasado` o métodos que deseamos que sean tratados deben declararse como `public` dentro de las clases, esto es importante!!

Contenido

	Pág.
Certificación	ii
Cesión de derechos	iii
Dedicatoria	iv
Agradecimiento	v
Introducción	vi
1. Capítulo 1: Sistemas de Información Geográfica	1
Qué es un SIG?	1
Clasificación de las Funciones de un SIG	2
Medición de la Información Temática.....	3
Tipología de mapas en función del nivel de medida de las variables y de las dimensiones geométricas de las unidades de observación	4
Búsqueda/Recuperación de información de una base de datos Geográfica.....	5
2. Capítulo 2: Problemática actual y selección de la herramienta de desarrollo	7
Introducción	7
Objetivos.....	7
Justificación del desarrollo de la aplicación Web.....	8
Funciones y algoritmos más importantes en el desarrollo de la aplicación Web ..	11
Condiciones de elección de la herramienta para el desarrollo de la aplicación Web	13
Diferencias entre aplicaciones Windows y aplicaciones Web.....	16
3. Capítulo 3: Diseño de la aplicación	20
Diseño físico	20
Diseño de salidas.....	21

Consultas temáticas.....	22
Consideraciones de topografía digital	22
Formato del mapa topográfico digital	23
Coordenas	23
Geofiguras encontradas en la topografía	24
Programación por capas	25
4. Capítulo 4: Pruebas de verificación y validación.....	30
Validación	30
Verificación	32
Resultados de la aplicación Web, una visión general.....	33
Resultados de la evaluación	40
5. Capítulo 5: Conclusiones y recomendaciones	47
Conclusiones	47
Recomendaciones	48
Bibliografía.....	50
Referencias Web.....	50
Anexos.....	52
Función Fresnel para evaluar los tramos	52
Encontrar todas las posibles rutas entre dos puntos(Dijkstra Modificado)	53
Determinar las tres mejores rutas	54
Armar la página <i>wfCalculos</i>	56
Cargar el mapa al servidor.....	57
Flujograma para el desarrollo del proyecto.....	58
Encuestas para validación de la aplicación Web.....	59
Cómo usar componentes Windows en Web.....	60