

UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO
UNIDAD DE ESTUDIOS A DISTANCIA
MODALIDAD SEMIPRESENCIAL
CARRERA SISTEMAS

TRABAJO INVESTIGATIVO

**“SOFTWARE PARA EL CALCULO DE
LAS DISTANCIAS DE CIUDAD A CIUDAD”**

AUTORES:

Cristóbal Patricio Rojas Fondevila
Patricio Fernando Fuentes Silva

ASESORA:

Ing. Lucrecia Llerena Guevara Msc.

Quevedo – Los Ríos – Ecuador

2010

UNIVERSIDAD TECNICA ESTATAL DE QUEVEDO
UNIDAD DE ESTUDIOS A DISTANCIA
MODALIDAD SEMIPRESENCIAL
CARRERA SISTEMAS

**“SOFTWARE PARA EL CALCULO DE
LAS DISTANCIAS DE CIUDAD A CIUDAD”**

TRABAJO INVESTIGATIVO

Presentado al Honorable Técnico Académico Administrativo de la Unidad de Estudios a Distancia como requisito previo para la obtención del título de

TÉCNICO PROGRAMADOR

MIEMBROS DEL TRIBUNAL

PRESIDENTE DEL TRIBUNAL

Lcdo. Ramiro Rivera García Msc.

MIEMBRO DEL TRIBUNAL

Ab. Miriam Cárdenas Zea Msc.

MIEMBRO DEL TRIBUNAL

Ing. Stalin Carreño Sandoya

ASESORA

Ing. Lucrecia Llerena Guevara Msc.

Quevedo – Los Ríos – Ecuador

2010

La responsabilidad por la investigación, resultados, conclusiones y recomendaciones pertenecen exclusivamente a los autores de este informe.

Patricio Fernando Fuentes Silva

Cristóbal Patricio Rojas Fondevila

CERTIFICACIÓN

Ing. Msc. Lucrecia Llerena Guevara, Asesora certifico: Que los señores Patricio Fernando Fuentes Silva y Cristóbal Patricio Rojas Fondevila realizaron el Trabajo Investigativo titulado: **“SOFTWARE PARA EL CÁLCULO DE LAS DISTANCIAS DE CIUDAD A CIUDAD”**, bajo mi dirección, habiendo cumplido con la disposición reglamentaria establecida para el efecto.

Ing. Lucrecia Llerena Guevara Msc.

ASESORA

AGRADECIMIENTO

Queremos expresar nuestros agradecimientos a las siguientes personas:

Dr. Roque Vivas Moreira Msc. Rector de la UTEQ.

Ing. Guadalupe Murillo de Luna Msc. Vicerrectora de la UTEQ.

Econ. Roger Yela Burgos Msc. Directora de la UED.

Ing. Nancy Rodríguez Gavilanes Msc. Subdirectora de la UED,

Ing. Lucrecia Llerena Guevara Msc. Asesora.

Quienes forman parte de la Institución Superior, que gentilmente abre las puertas a todos quienes buscan realizar una meta más de superación en su vida estudiantil, hasta llegar a ser profesionales.

A todos quienes de una u otra forma contribuyeron para el desarrollo, elaboración y culminación de la presente investigación.

Patricio Fernando Fuentes Silva
Cristóbal Patricio Rojas Fondevila

ÍNDICE

Caratula

Certificación

Agradecimiento

Índice.

I.	INTRODUCCIÓN.....	8
1.1.	Justificación	9
1.3.	Objetivos	11
1.3.1	General	11
1.3.2	Específicos	11
II.	REVISIÓN DE LITERATURA	13
2.2.1.	¿Para Qué Necesitamos Bases de Datos?	16
2.2.2.	¿Qué es una Base de Datos?.....	17
2.2.3	¿Por Qué no Simplemente Usar Excel?	18
2.2.4.	Organizando y Filtrando para Ubicar Datos en Excel.....	19
2.2.5.	Ingresar Datos en Excel	20
2.2.6.	Ingresar Datos en Access	21
2.2.7.	Datos de una Tabla de Información de Access.....	22
2.2.8.	Diagrama de Relación de Tablas en Access.....	23
III.	MATERIALES Y MÉTODOS.....	25
3.6.	Técnicas e instrumentos de evaluación	27
IV.	RESULTADOS Y DISCUSIÓN	28
4.4.1.	Opción Altas de Ciudades:	31
4.4.2.	Opción “Actualizar Distancias”	32
4.4.3.	Opción Salir:	33
4.5.1.	Tablas:.....	34
4.5.2.	Consultas:	34
V.	CONCLUSIONES.....	35
VI.	RECOMENDACIONES	36

VII.	FUENTES DEL SISTEMA.....	37
7.1.	Menu Inicial:.....	37
VIII.	BIBLIOGRAFÍA	51
IX.	ANEXOS.....	52

I. INTRODUCCIÓN

El presente trabajo tiene como objetivo principal, conseguir la acreditación como Técnico programador de sistemas informáticos, en la Universidad Técnica Estatal de Quevedo. El mismo que ha sido desarrollado sobre la base de una investigación sobre las tendencias del futuro de la programación de sistemas informáticos y, bajo la hipótesis del desarrollo del sistema con el mínimo de codificación posible, tender a “cero código”.

El manejo de computadores tiene la misión de hacer más fácil el desempeño de las actividades que antes se hacían en forma manual. Al nacimiento del Estándar ANSI del lenguaje C y hoy el VISUAL BASIC 2008, Los motivos que han llevado a Microsoft al desarrollo de VB han sido tanto tecnológicos como estratégicos.

Respecto a las motivaciones tecnológicas, la necesidad de poner a disposición del programador una plataforma de desarrollo con plena potencia para abarcar los requerimientos de las nuevas aplicaciones que están a punto de llegar, y que no soporte incómodos lastres derivados de antiguos modelos de programación, ha desembocado en una tecnología totalmente nueva, que no arrastra pesadas incompatibilidades, pero que sin embargo, permite la ejecución de componentes basados en el anterior modelo de programación. Esto es VB, una nueva arquitectura para el futuro del desarrollo de aplicaciones, y no, como en un principio pudiera pensarse, una operación más de marketing, que proporciona las herramientas ya conocidas con algunas remodelaciones.

A parte de ello pensemos lo importante que ha sido la utilidad que proporciona el manejo del computador, que no solamente ahora nos guía por una ruta sino que podemos crear aplicaciones que nos den la distancia de una ciudad a otra la cual nos puede ser de mucha utilidad, ya que solo en

internet se puede acceder a estos datos pero no a un programa que se lo puede tener en el computador personal o quizá en el teléfono pero de menor tamaño.

1.1. Justificación

Se ha considerado que es muy importante aprender un lenguaje de programación, que ayude a comprender que la computadora ha sido creada para facilitar las actividades del hombre: en su trabajo, en el hogar y hasta para recreación y entretenimiento.

La elaboración de este trabajo en el que se trata del desarrollo del proyecto de la creación de un software que calcule la distancia de ciudad a ciudad.

Se justifica que la razón primordial de efectuar este software es practicar lo que se ha aprendido no sólo en el último módulo de Programación Estructurada, sino también en los anteriores y la realización de una base de datos en Access la cual se interrelacione con Visual Basic 2008 es por eso que todos los registros de las distancias y las ciudades se las realizo en este programa junto con Visual Basic 2008.

Frecuentemente, cuando se viaja, lo primero que la gente se pregunta es cuánto tiempo le tomará trasladarse de un lugar a otro. Esa unidad de medición aunque no es la más lógica, es la más usada. Sin embargo, para conseguir un tiempo aproximado, es indispensable conocer la distancia o la dimensión de la ruta. Es por eso que, por ser los integrantes de este grupo de trabajo, residentes de ciudades lejanas a Quevedo, complementado con el deseo de conocer y entrenarse en afrontar los retos y esfuerzos que requieren la obtención de un título académico se optó por emprender el proyecto que se describe en el presente documento, como es la creación de un software que

calcule la distancia entre dos ciudades.

Se justifica la realización de este software para aplicar los conocimientos adquiridos en la creación de una base de datos en Access la cual se interrelacione con Visual Basic 2008.

EcuRutas le brinda la capacidad de evaluar entre varias rutas de viaje y luego analizar el impacto sobre la distancia total a recorrer, con el fin de elegir la más adecuada a las necesidades del usuario del sistema. Por ejemplo, puede evaluar el impacto de cambiar la ruta de desplazamiento entre ciudades y ver las implicaciones del cambio. Esto significa que los usuarios, con ayuda de herramientas complementarias como Excel, puedan analizar las rutas que más le convenga sin tener que adivinar ni aventurar qué ciudades necesitan incluirse en un proyecto de viaje.

Generalmente, cuando se planifica un viaje, lo primero que uno se imagina es cómo llegar al destino esperado y cuantos kilómetros de debe recorrer.

Aunque en nuestro medio, la gente se ha acostumbrado a medir la en función del tiempo que duraría trasladarse de un lugar a otro, técnicamente no es tan real, pues el recorrer una misma distancia entre dos puntos, puede variar el tiempo dependiendo de muchos factores, como flujo del tráfico, velocidad de circulación, paradas intermedias, entre otros. Por tal motivo, la mejor manera de conocer la distancia entre dos ciudades es a través de las medidas de longitud.

Adicionalmente, por la disponibilidad de información fiable en organismos del Estado como la “Dirección Nacional de Control del Tránsito y Seguridad Vial”, han hecho factible el desarrollo de este proyecto.

Algunos de los beneficiarios del proyecto son:

- Turistas que viajan en su vehículo.
- Choferes profesionales.
- Estudiantes
- Personas que desean averiguar las distancias entre ciudades por cultura general.
- Empresa de transporte y logística terrestres.

1.3. Objetivos

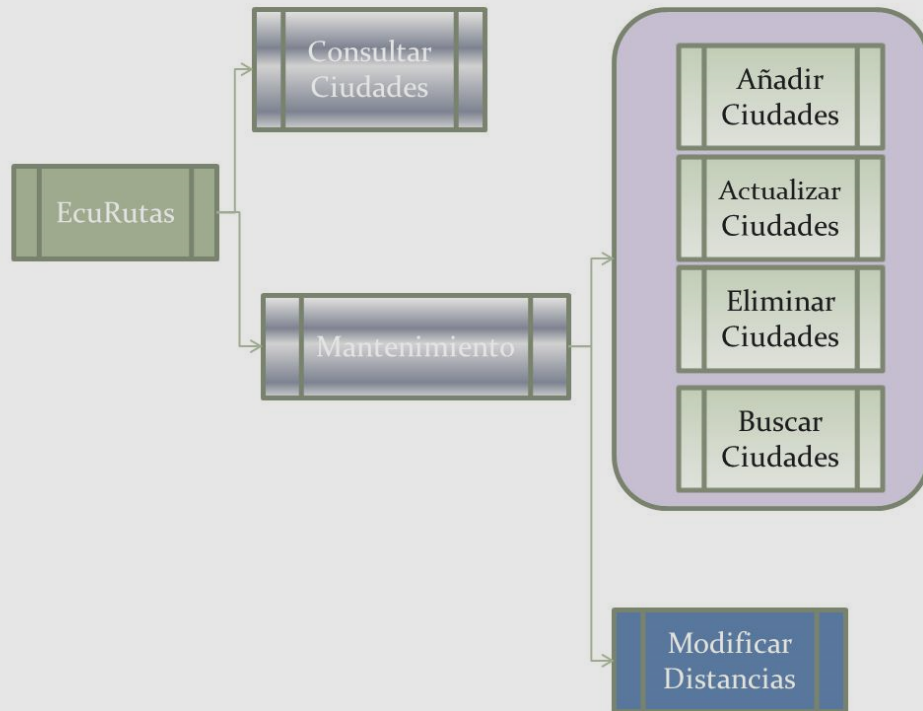
1.3.1 General

Crear un software en lenguaje Visual Basic 2008 y con acceso a base de datos que sirva para calcular la distancia entre dos ciudades.

1.3.2 Específicos

- Elaborar un programa que nos permita consultar las distancias entre ciudades.
- Definir funcionalidades dentro del programa para añadir, modificar, eliminar y buscar ciudades.
- Implementar los controles adecuados para navegar o consultar las ciudades definidas en la base de datos.
- Desarrollar las funcionalidades pertinentes para modificar las distancias entre ciudades.

OBJETIVOS EPECÍFICOS:



II. REVISIÓN DE LITERATURA

2.1. VISUAL BASIC

Rod Stephens (2008). La respuesta es muy sencilla, el Visual Basic nos permite realizar unos buenos Diseños en nuestros programas, además de que el programa que se realice en Visual Basic se podrá correr (ejecutar) en casi todos los sistemas operativos de la Microsoft, y lo mejor, bueno a nuestra opinión, es mucho más fácil de realizar los códigos, ya que tiene una ayuda de auto complementar.

2.1.1. ¿Por qué utilizar la consola de aplicación y no la aplicación de Windows forms?

Rod Stephens (2008). Utilizamos la **Consola de Aplicación** en las primeras practicas para así acostumbrarnos al uso del programa y el lenguaje para luego poder aprender el uso de este, así nos sería más fácil de aprender la lógica que vamos a necesitar para resolver cualquier algoritmo que se nos presente, y además porque casi todos los códigos que usamos en la Consola de Aplicación, lo vamos a usar en Aplicación de Windows. No se uso la Aplicación de Windows para comenzar a programar, Ya que si no se aprende a ver como uno puede capturar y mostrar datos mediante código y no mediante unos controles, al principio nos iba a ser fácil, pero conforme avancemos no nos iba a quedar en claro, y es por eso que uso primero APLICACIÓN DE CONSOLA.

En Visual Studio .Net existe un ambiente de trabajo o **IDE (INTEGRATED DEVELOPMENT ENVIRONMENT = AMBIENTE INTEGRADO DE DESARROLLO)** el cual permite trabajar con el Visual Basic, Visual C#, Visual J# y Visual C++, el que permite diseñar las interfaces gráficas de usuario de

una o más aplicaciones, es decir el que se puede crear formularios que contengan a otros formularios u objetos diversos como los TextBox, ComboBox, etc.

2.1.2. Visual Basic (VB)

Rod Stephens (2008). Nombre que se le da a una colección de tecnologías de Microsoft que permite a la plataforma Windows e Internet trabajar en conjunto. Algunas de sus tecnologías principales son el ActiveX, DHTML y COM), ha sido fundamentalmente, el de la escritura de componentes para su uso por parte de las páginas ASP (ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.) de una aplicación WEB; de hecho, es el lenguaje preferido para el desarrollo de componentes debido a su ya larga tradición como lenguaje sencillo y de fácil manejo.

Microsoft hizo un intento de dotar de un mayor número de características a Visual Basic para que pudiera convertirse en una herramienta de desarrollo integral para Internet; para ello, incorporó las WEB CLASSES, los documentos ActiveX y controles ActiveX, aunque ninguno de ellos obtuvo plena aceptación.

Por un lado, las Web Clases tenían el complejo modelo de programación, mientras que los documentos ActiveX arrojaban unos pobres rendimientos de ejecución. Con respecto a los controles ActiveX, necesitaban de cierto proceso de instalación por parte del servidor, lo que los hacía en muchas situaciones poco operativas. Estas circunstancias han impedido que VB pudiera convertirse en la herramienta de desarrollo para Internet de Microsoft.

Otros factores decisivos que han limitado la plena entrada de VB en la programación web han sido la falta de capacidades multihebra (Podemos definir la programación multihebra o multihilo como un estilo de ejecución en el que se conmuta (Cambiar una cosa por otra) entre distintas partes del código de un mismo programa durante la ejecución), inexistencia de un interfaz de usuario específico para aplicaciones web, falta de herencia y otras características orientadas a objetos, escasa integración con otros lenguajes, deficiente gestión de errores, etc., aspectos todos, solucionados en VB.NET.

2.1.3. Interfaces de ventana. Formularios y controles

Rod Stephens (2008). Es un hecho palpable el que la programación para Internet, ha ganado en los últimos tiempos una importante cuota de desarrollo, en detrimento (**destrucción**) de las aplicaciones basadas en Windows. Sin embargo, todavía existe un importante conjunto de programas que deberán seguir funcionando en Windows y que tendrán que migrarse a la plataforma .NET.

Para este sector del desarrollo, .NET Framework proporciona una arquitectura renovada, en lo que a la programación de aplicaciones Windows se refiere, a los nuevos formularios y controles de Windows.

Un formulario Windows representa la conocida ventana, que se utiliza en las aplicaciones ejecutadas bajo alguno de los sistemas operativos de la familia Windows: Windows95/98, NT, ME, 2000, XP, Vista, etc.

Un control, por otra parte es aquel elemento situado dentro de una ventana o formulario y que permite al usuario de la aplicación Windows interactuar con la misma para introducir datos o recuperar información.

Dentro de .NET, las ventanas clásicas Windows, reciben la denominación de

Windows Forms, o WinForms, para diferenciarlas de los formularios Web o WebForms, que son los que se ejecutan en páginas ASP.NET.

La clase form:

Esta clase contiene todos los miembros para la creación y manipulación de formularios.

Tras instanciar un objeto de Form, mediante la configuración de las adecuadas propiedades, podemos crear formularios estándar, de diálogo, de interfaz múltiple o MDI, con diferentes bordes, etc.}

2.2. BASE DE DATOS

2.2.1. ¿Para Qué Necesitamos Bases de Datos?

Pantigoso Silva(2008). Access 2007 es un programa que permite **crear** y **administrar** bases de datos. Una **base de datos** es un lugar donde se **guarda información** relacionada con un tema específico. Según el uso que le va a dar a la información se definirá si es necesaria una base de datos de Access o una herramienta diferente para crear y administrar los datos.



Gráfico 1 Administrador de Tablas

2.2.2. ¿Qué es una Base de Datos?

Una **base de datos** permite guardar información relacionada con un tema específico de una manera organizada. Además de guardar datos, también puede, **clasificar**, **extraer**, y **resumir** información relacionada con los datos. Uno de los programas que le permite hacer esto es **Microsoft Office Access 2007**, el cual es un programa para crear y administrar bases de datos.

IdOrigen	CdOrigen	IdDestino	CdDestino	Distancia
22	Aloag	9	Latacunga	56
22	Aloag	4	Cuenca	418
22	Aloag	6	Guaranda	189
22	Aloag	30	Manta	347
22	Aloag	13	Nueva Loja	283
22	Aloag	37	Pto. Putumayo	467
22	Aloag	24	Baños	138
22	Aloag	22	Aloag	0
22	Aloag	1	Ambato	97
22	Aloag	29	Tena	207
22	Aloag	27	Rumichaca	288
22	Aloag	16	Puyo	199
22	Aloag	10	Loja	625
22	Aloag	18	Riobamba	150
22	Aloag	29	Muisne	342
22	Aloag	34	Quevedo	207
22	Aloag	26	Baeza	122
22	Aloag	14	Portoviejo	314
22	Aloag	31	Otavalo	134
22	Aloag	21	Zamora	683
22	Aloag	32	Salinas	547
22	Aloag	25	Bahía de Carac	292
22	Aloag	2	Azogues	377
22	Aloag	3	Babahoyo	312
22	Aloag	33	San Lorenzo	331
22	Aloag	15	Pto. Fco. De O.	317
22	Aloag	38	Pto. Morona	541
22	Aloag	8	Ibarra	157
22	Aloag	7	Guayaquil	375
22	Aloag	23	Sto. Domingo	98
22	Aloag	12	Machala	513
22	Aloag	17	Quito	34

Gráfico 2 Visualización de una tabla

2.2.3 ¿Por Qué no Simplemente Usar Excel?

Pantigoso Silva (2008). Hay muchos tipos de datos que necesitará archivar y administrar: texto y números, por ejemplo. Dependiendo de lo que quiera de sus datos, nosotros querremos o no usar una base de datos. Usted podrá usar un programa para hacer hojas de cálculo como **Microsoft Excel**. ¿Cómo saber cuáles datos pueden ser administrados adecuadamente con Excel y cuáles datos de verdad necesitan **Access** para administrarlos eficientemente? Todo depende en la cantidad de datos que necesitamos administrar y para que necesitamos los datos. Respondamos esto mirando un escenario como una librería o como el trabajo de distancia a distancia.

Si nosotros trabajamos en el negocio de los libros, necesitaremos llevar registro de sus clientes y sus órdenes.

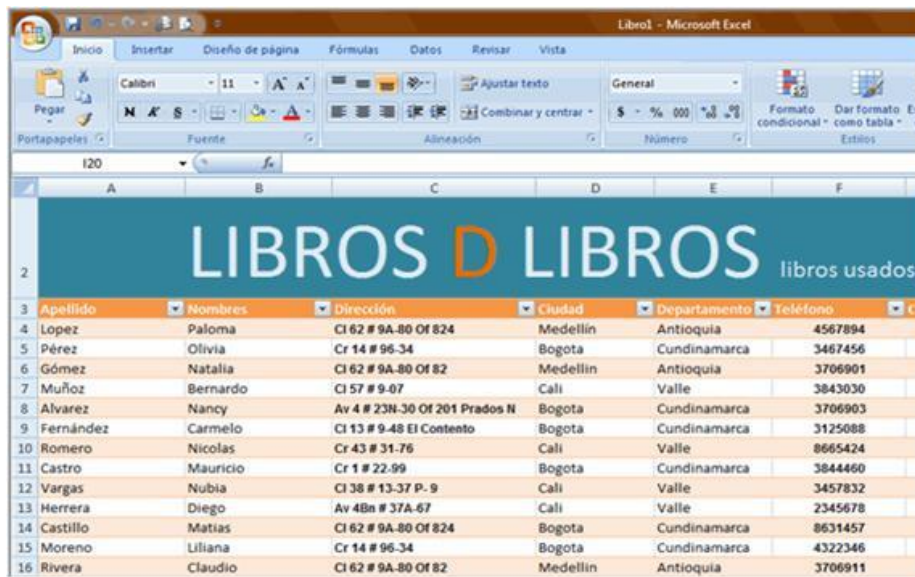
Podría usar **Microsoft Excel** para guardar y administrar este tipo de datos; sin embargo, Excel es un **programa para hojas de cálculo** tradicionalmente usado para administrar **información numérica**, como sumar todas las compras de un cliente. Mientras este puede realizar un buen trabajo a la hora de guardar algunos tipos de bases de datos basadas en texto, como la información de contacto de sus clientes en realidad Excel no fue creado para hacer eso.

Los siguientes ejemplos nos mostrarán por qué una base de datos de Access puede ser una mejor opción para el negocio de los libros o cualquier información que necesite almacenar en orden.

2.2.4. Organizando y Filtrando para Ubicar Datos en Excel

Pantigoso Silva (2008). En Excel, podemos guardar sus datos en una hoja de cálculo y de esta manera enviar correos a toda la lista, o encontrar clientes específicos para enviar correos. Hasta podrá filtrar la información de los clientes para desplegar todos los clientes que viven en un lugar específico, como en la siguiente imagen. Adicionalmente, usted puede ordenar los datos de una manera específica.

Datos Filtrados para Desplegar los Registros



	A	B	C	D	E	F
	LIBROS D LIBROS libros usados					
	Apellido	Nombres	Dirección	Ciudad	Departamento	Teléfono
3	Lopez	Paloma	CI 62 # 9A-80 Of 824	Medellin	Antioquia	4567894
4	Pérez	Olivia	Cr 14 # 96-34	Bogota	Cundinamarca	3467456
5	Gómez	Natalia	CI 62 # 9A-80 Of 82	Medellin	Antioquia	3706901
6	Muñoz	Bernardo	CI 57 # 9-07	Cali	Valle	3843030
7	Alvarez	Nancy	Av 4 # 23N-30 Of 201 Prados N	Bogota	Cundinamarca	3706903
8	Fernández	Carmelo	CI 13 # 9-48 El Contento	Bogota	Cundinamarca	3125088
9	Romero	Nicolas	Cr 43 # 31-76	Cali	Valle	8665424
10	Castro	Mauricio	Cr 1 # 22-99	Bogota	Cundinamarca	3844460
11	Vargas	Nubia	CI 38 # 13-37 P- 9	Cali	Valle	3457832
12	Herrera	Diego	Av 48n # 37A-67	Cali	Valle	2345678
13	Castillo	Matias	CI 62 # 9A-80 Of 824	Bogota	Cundinamarca	8631457
14	Moreno	Liliana	Cr 14 # 96-34	Bogota	Cundinamarca	4322346
15	Rivera	Claudio	CI 62 # 9A-80 Of 82	Medellin	Antioquia	3706911

Gráfico 3 Ejemplo de utilización de filtros.

Sin embargo, si se desea ver resultados muy específicos, como cuantas órdenes realizó un cliente al año, **Excel no es tan eficiente como Access** para proveerle esta clase de información.

2.2.5. Ingresar Datos en Excel

Si usted usa una hoja de cálculo de Excel para seguir sus órdenes, cada vez que un cliente pide una orden, usted tendrá que insertar una nueva fila de información en la hoja de cálculo. Esto incluirá los nombres y direcciones de los clientes. Si ese cliente pide más de una orden a la vez, esa información tendrá que ingresarse cada vez. Su hoja de cálculo va a tener información **redundante o repetida**.

Apellido	Nombres	Dirección	Ciudad	Departamento	Teléfono	Celular	Títulos
Berna	Tina	Cr 43 # 31-76	Cali	Valle	8665424	3155678977	LA REINA MUY DE CERCA
Berna	Tina	Cr 43 # 31-76	Cali	Valle	8665424	3155678977	CREPUSCULO
Berna	Tina	Cr 43 # 31-76	Cali	Valle	8665424	3155678977	NUEVA LUNA
Berna	Tina	Cr 43 # 31-76	Cali	Valle	8665424	3155678977	ECLIPSE
Berna	Tina	Cr 43 # 31-76	Cali	Valle	8665424	3155678977	LECTURAS DE MI MISMO
Herrera	Diego	Av 48n # 37A-67	Cali	Valle	2345678	3124567889	LA SONRISA DEL FLAMENCO
Granados	Maria Helena	Cl 57 # 9-07	Medellin	Antioquia	3706912	4345345368	LA REINA MUY DE CERCA
Granados	Maria Helena	Cl 57 # 9-07	Medellin	Antioquia	3706912	4345345368	CREPUSCULO
Granados	Maria Helena	Cl 57 # 9-07	Medellin	Antioquia	3706912	4345345368	NUEVA LUNA
Granados	Maria Helena	Cl 57 # 9-07	Medellin	Antioquia	3706912	4345345368	ECLIPSE
Granados	Maria Helena	Cl 57 # 9-07	Medellin	Antioquia	3706912	4345345368	LECTURAS DE MI MISMO
Lopez	Paloma	Cl 62 # 9A-80 Of 824	Medellin	Antioquia	4567894	3155678977	LECTURAS DE MI MISMO
Pérez	Olivia	Cr 14 # 96-34	Bogota	Cundinamarca	3467456	4345345368	POR TIERRA MAR Y AIRE
Gómez	Natalia	Cl 62 # 9A-80 Of 82	Medellin	Antioquia	3706901	4353453453	LA SENDA DEL CHAMAN

Gráfico 4 Ejemplo de limitaciones en el uso del Excel

Esta es la limitación del programa de hojas de cálculo como **Excel** porque es una sola línea.

2.2.6. Ingresar Datos en Access

Pantigoso Silva (2008). **Microsoft Access** está diseñado específicamente para ayudarle a administrar información como los datos descritos en el escenario de distancias. Si estábamos usando una base de datos de Access en vez de un programa de hojas de cálculo, sólo se tendrá que ingresar el nombre del cliente, su dirección de residencia y el número de teléfono una sola vez la primera vez que hagan un ingreso.

Esa información de la provincia se podría ingresar en una **tabla** de Access diseñada para contener la información básica de las provincias. Una tabla es una lista de información relacionada en columnas y filas. En una tabla, cada **FILA** es llamada **REGISTRO** y cada **COLUMNA** es llamada un **CAMPO**. Una tabla de Access en **VISTA** la hoja de datos se ve similar a una hoja de cálculo de **EXCEL**, como puede ver en la imagen de abajo.

IdOrigen	CdOrigen	IdDestino	CdDestino	Distancia
22	Aloag	9	Latacunga	56
22	Aloag	4	Cuenca	418
22	Aloag	6	Guaranda	189
22	Aloag	30	Manta	347
22	Aloag	13	Nueva Loja	283
22	Aloag	37	Pto. Putumayc	467
22	Aloag	24	Baños	138
22	Aloag	22	Aloag	0
22	Aloag	1	Ambato	97
22	Aloag	19	Tena	207
22	Aloag	27	Rumichaca	288
22	Aloag	16	Puyo	199
22	Aloag	10	Loja	625
22	Aloag	18	Riobamba	150
22	Aloag	39	Muisne	342

Gráfico 5 Ejemplo de la organización de una tabla

2.2.7. Datos de una Tabla de Información de Access

Adicional a la tabla con información de las distancias como sea el caso, nosotros probablemente también queramos una tabla con información de los productos para vender, etc. y una tercera tabla para tener datos relacionados con órdenes específicas de algunos clientes o mas distancias ya que las carreteras siempre aumentan o desvían entonces las distancias cambian. Todas estas tablas estarán vinculadas, para ayudarnos a sacar el mayor provecho de sus datos.

Access es llamado programa de **administración de bases de datos relacionadas**, porque **LAS TABLAS ESTÁN VINCULADAS**, o **RELACIONADAS**, como lo puede ver en la siguiente imagen.

En este ejemplo, las tablas de información y distancias, orígenes y nombres están relacionadas por el ID de las **Ciudades** y el ID de las **Distancias**.

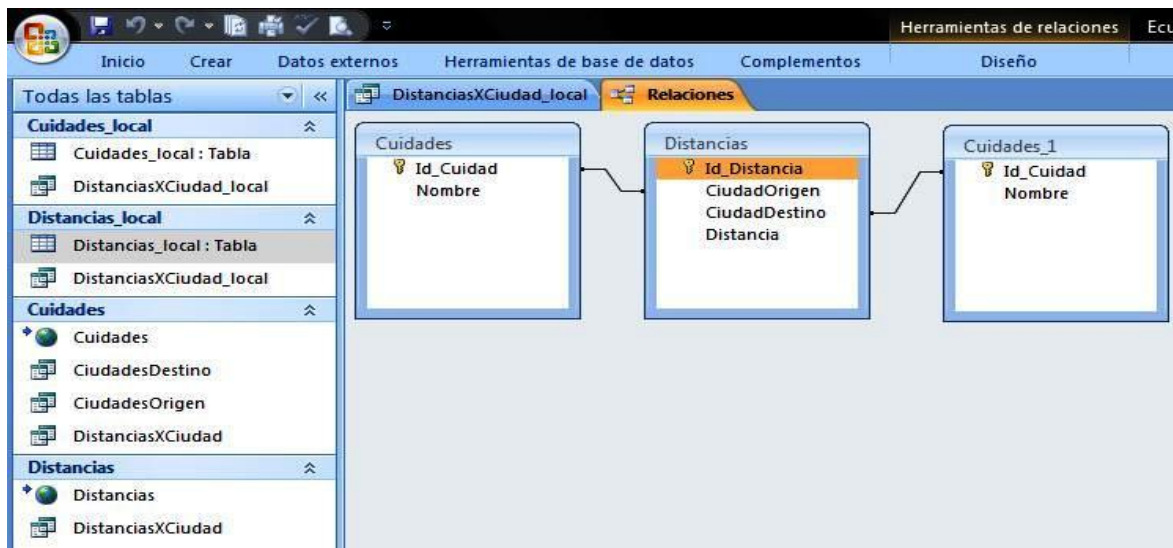


Gráfico 6: Mapa entidad Relación

2.2.8. Diagrama de Relación de Tablas en Access

Ahora, supongamos que queremos identificar las distancias de las Provincias. Con Access, esto es posible porque puede buscar y coleccionar información de varias tablas al mismo tiempo.

La tabla con **Información de las Distancias** contiene información acerca de los estados, y el **Orden de las tablas** incluye información acerca de cuáles rutas fueron más cortas. Necesitaremos información de la **tabla** para identificar la distancia más cercana y la ruta a una ciudad específica.

Nosotros podríamos mirar la información en estas tablas separadamente para responder su pregunta: "¿Cual es la distancia más corta de ciudad a ciudad?". En la tabla **Información**, podremos ver todas las distancias de una ciudad en particular y en la tabla de Provincias, usted puede ver todas las provincias. **El verdadero poder de Access es poder vincular y extraer información** de varias tablas para responder preguntas específicas e incluso puede ligarse con otros programas como VISUAL BASIC y Excel que son de Microsoft, ya

que el presente trabajo está realizado a que la base de datos realizada en **ACCESS** sea ejecutada a través de **VB** y que este nos permita aumentar la base de datos, reformar, eliminar o actualizar información o como llamamos en nuestro trabajo ALTAS DE CIUDADES y ACTUALIZACIÓN DE DISTANCIA aunque con solo ACCESS se lo podría hacer pero con fines didácticos y aprendizaje se realizó este trabajo vinculando los dos programas.

Como podemos ver en la siguiente imagen, los resultados de sus preguntas específicas, o consultas son desplegados.



IdOrigen	CdOrigen	IdDestino	CdDestino	Distancia
22	Aloag	9	Latacunga	56
22	Aloag	4	Cuenca	418
22	Aloag	6	Guaranda	189
22	Aloag	30	Manta	347
22	Aloag	13	Nueva Loja	283
22	Aloag	37	Pto. Putumayc	467
22	Aloag	24	Baños	138
22	Aloag	22	Aloag	0
22	Aloag	1	Ambato	97
22	Aloag	19	Tena	207
22	Aloag	27	Rumichaca	288
22	Aloag	16	Puyo	199
22	Aloag	10	Loja	625
22	Aloag	18	Riobamba	150
22	Aloag	39	Muisne	342
22	Aloag	34	Quevedo	207

Gráfico 7 Vista lógica de una tabla

III. MATERIALES Y MÉTODOS

2.2. Localización y duración de la investigación

El presente trabajo fue realizado en la ciudad de Quito, la investigación estuvo enfocada principalmente en dos áreas y tuvo una duración de 1 mes y medio:

2.3. Materiales y equipos

Los materiales y equipos utilizados, fueron pocos y principalmente son los siguientes:

MATERIALES	CANTIDAD
Resma de papel	1
Lapiceros	2
Equipos	2
Laptops	
Impresora	1
Memory Flash	1
Software	
Microsoft Visual Basic 2008 (licencia legal gratuita)	
Microsoft Access 2008	

2.4. Tipo de investigación

El trabajo realizado encaja en la “INVESTIGACIÓN APLICADA O TECNOLÓGICA” pues estuvo enfocada a encontrar solución y conocimientos científicos para resolver problemas específicos como la Metodología más adecuada para acceder y manipular datos en **Visual Basic .Net**, así como encontrar la información adecuada para la alimentación de la base de datos.

2.5. Métodos

El método que más se aproxima al utilizado, podríamos definirlo como “Deductivo”, partimos de una teoría, la aplicamos en el sistema y regresamos a la investigación de la teoría para corregir errores o desvíos de los resultados esperados.

2.6. Fuentes

3.1. Primarios

Las fuentes primarias que se seleccionaron para el proyecto fueron por el método de búsqueda por referencias en el Internet y, también, en libros especializados como son los almanaques.

Como fuentes más idóneas, criterio nuestro se seleccionaron, el libro Almanaque 2009, así como la información publicada por la “Dirección Nacional de Control del Tránsito y Seguridad Vial” en su página web.

3.2. Secundarios

- Libro de Access 2007 de Jaime Pantigoso Silva (grupo editorial megabyte)
- Documentación de las tutorías.
- msdn.com

Igual que en el caso de las fuentes primarias, se utilizaron libros, documentos que se disponían al momento de la investigación, así como librerías especializadas en el Internet en lo concerniente al desarrollo y programación de sistemas informáticos.

3.6. Técnicas e instrumentos de evaluación

La técnica de evaluación fue la de ensayo de prueba – error. Conforme se presentaban los problemas se corregían y se volvía a probar el sistema cíclicamente hasta conseguir el objetivo deseado.

IV. RESULTADOS Y DISCUSIÓN

4.1. Descripción Funcional

Elaborar un sistema informático que permita conocer la distancia en kilómetros entre dos ciudades.

El sistema debe permitir añadir, modificar y eliminar ciudades, así como ingresar las distancias correspondientes entre una a varias ciudades predefinidas.

El sistema presenta una pantalla de presentación inicial con un menú de opciones, el cual permite navegar sobre la aplicación y conseguir los objetivos propuestos en la descripción funcional.

4.2. Menú Inicial

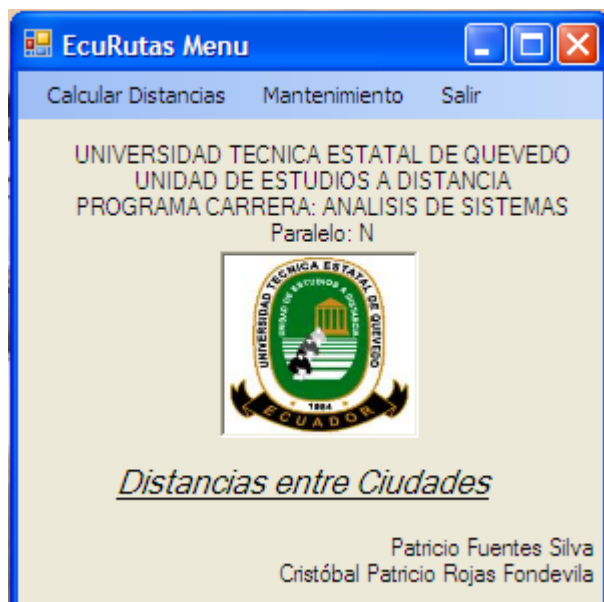


Gráfico 8 Menú Inicial

El menú presenta las siguientes opciones:

- Calcular Distancias:
- Mantenimiento y
- Salir.

En la opción Mantenimiento tiene dos opciones adicionales:

- Altas de Ciudades y
- Actualizar Distancias.



Gráfico 9 Selección de Opciones

Adicionalmente, por ser el objetivo principal del sistema, es posible acceder directamente al cálculo de distancias presionando la leyenda “**Distancias entre Ciudades**”

4.3. Opción Calcular Distancias

El usuario presiona la opción del menú “**Calcular Distancias**” o presiona la leyenda “*Distancia entre Ciudades*” durante la instancia en que el puntero se transforma en una mano.

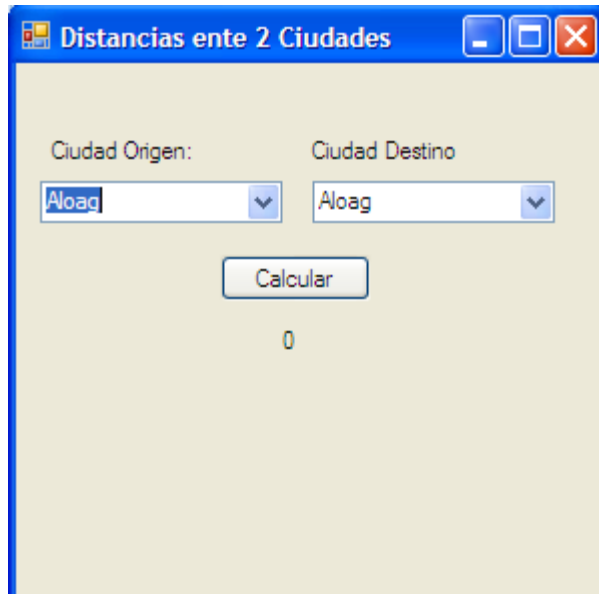


Gráfico 10 Formulario de cálculo de distancias

El sistema muestra dos listas de opciones para que el usuario pueda seleccionar las ciudades de origen y destino requeridas.

El sistema muestra también un botón, en el cual, al presionarlo, el sistema muestra la distancia registrada entre los dos puntos en mención.

Para salir del formulario, el usuario presiona el **ícono X** de la esquina superior derecha.

4.4. Opción Mantenimiento

4.4.1. Opción Altas de Ciudades:

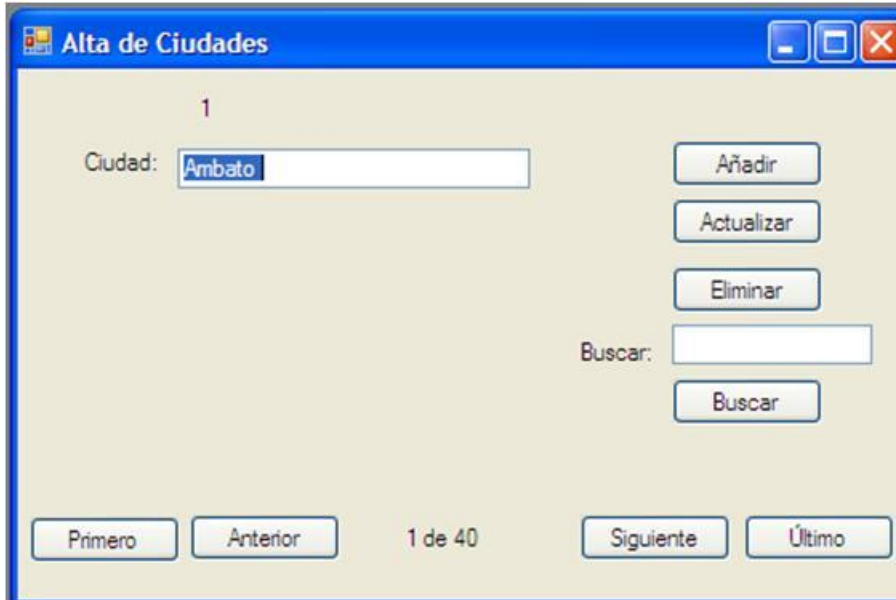


Gráfico 11 Formulario de altas de ciudades.

El Sistema presenta un formulario con una caja de texto que muestra la primera ciudad registrada en la base de datos.

El sistema, adicionalmente, presenta un conjunto de botones con la finalidad que el usuario pueda dar mantenimiento a la base de datos desde el sistema, en lo concerniente a ciudades:

Botón Añadir: Permite el registro de nuevas ciudades.

Botón Actualizar: Modifica los cambios realizados a la ciudad desplegada en la caja de texto “Ciudad”

Botón Eliminar: Borra una ciudad de la base de datos

Botón Buscar: El sistema busca en la base de datos la ciudad que coincida con el texto digitado en el casillero “Buscar”. Si el sistema encuentra una ciudad que coincide exactamente su nombre con el criterio de búsqueda, la despliega en el casillero “Ciudad”, caso contrario emite un mensaje informativo al respecto.

Botones de Navegación:

Primero: El sistema despliega el primer registro de la base de datos.

Anterior: El sistema busca y presenta el registro precedente al que está desplegado en el formulario.

Siguiente: El sistema busca y muestra el registro posterior al que está desplegado en el formulario.

Último: El sistema muestra el último registro de la base de datos.

Adicionalmente el sistema informa la posición del registro visualizado dentro de la base de datos y la cantidad total de ciudades registradas en la base.

Presionado el ícono X, el sistema cierra el formulario.

4.4.2. Opción “Actualizar Distancias”

El usuario presiona la opción del menú “Actualizar Distancias” y el sistema presenta el siguiente formulario:

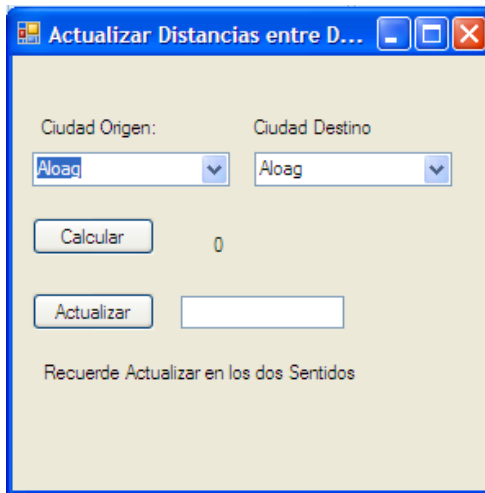


Gráfico 12 Formulario para registrar distancias.

El formulario contiene dos cuerpos de ejecución:

El superior que tiene el mismo comportamiento de la Opción Calcular Distancias y un cuadro de texto adicional para modificar la distancia calculada en caso de ser necesario.

Botón Actualizar: Accede a la base de datos y modifica la información correspondiente con lo digitado en el cuadro de texto adjunto.

Cuando se actualice una distancia, es necesario actualizar en sentido inverso también. Ejemplo: Si se actualiza la distancia entre Quevedo y Quito, también es necesario actualizar la distancia entre Quito y Quevedo. Como son rutas de viaje, no siempre la distancia de ida va a ser la misma que de regreso.

4.4.3. Opción Salir:

Cierra la aplicación y termina el programa.

4.5. Diseño de la Base de Datos

4.5.1. Tablas:

Ciudades	Distancias
Id_Ciudad	Id_Distancia
Nombre	CiudadOrigen
	CiudadDestino
	Distancia
CiudadesTableAdapter	DistanciasTableAdapter
Fill,GetData ()	Fill,GetData ()

Gráfico 13 Tablas del sistema

4.5.2. Consultas:

CiudadesDestino	CiudadesOrigen
Id_Ciudad	Id_Ciudad
Nombre	Nombre
CiudadesDestinoTableAdapter	CiudadesOrigenTableAdapter
Fill,GetData ()	Fill,GetData ()

Gráfico 14 Consultas o vistas lógicas

V. CONCLUSIONES

- El desarrollo de este software ha sido de mucho provecho, pues servirá de base para el entendimiento del resto de conocimientos que involucran la carrera de Ingeniería de Sistemas.
- El software para los usuarios es de gran utilidad pues permitirá tener como referencia la planificación de rutas y desarrollo de presupuesto en viajes.
- Las herramientas de programación son importantes, pero mucho más necesario es saber diseñar las aplicaciones. Tomando palabras ajenas: “No importa en qué lenguaje esté, una vez aprendido la Lógica se puede implementar en cualquier programa, el resto es práctica”.

VI. RECOMENDACIONES

- Para una futura versión de este programa, se puede utilizar la plataforma punto Net.
- Se puede implementar otras utilidades al software como son estadísticas del uso de combustibles, uso de llantas y tiempos de viaje.
- La próxima versión debería permitir diseñar rutas que involucren más de un destino.
- Se recomienda hacer interfaces con sistemas GIS para una mejor visualización de las rutas escogidas.

VII. FUENTES DEL SISTEMA

7.1. Menu Inicial:

```
Imports EcuRutas.DistanciasVista

Public Class Menu

    Private Sub CalcularDistanciasToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CalcularDistanciasToolStripMenuItem.Click
        '    MessageBox.Show("Calcular Distancias en Construcción")
        Dim DistanciasVista As New DistanciasVista()
        DistanciasVista.Show()
    End Sub

    Private Sub AltasDeCiudadesToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AltasDeCiudadesToolStripMenuItem.Click
        '    MessageBox.Show("Altas de Ciudades en Construcción")
        Dim AltasCiudad As New AltasCiudad()
        AltasCiudad.Show()
    End Sub

    Private Sub ActualizarDistanciaToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ActualizarDistanciaToolStripMenuItem.Click
        '    MessageBox.Show("Actualizar Distancias en Construcción")
        Dim DistanciasAct As New DistanciasAct()
        DistanciasAct.Show()
    End Sub

    Private Sub Salir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Salir.Click
```

```
        Close()  
    End Sub  
    Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Label2.Click  
        Dim DistanciasVista As New DistanciasVista()  
        DistanciasVista.Show()  
    End Sub  
End Class
```

7.2. Consulta de distancias entre ciudades

```
Imports System.Data.SqlClient
Imports System.Data
Public Class DistanciasVista
    Protected Const SqlConnectionString As String = _
        "Server=(local);" & _
        "DataBase=;" & _
        "Integrated Security=SSPI"
    Protected connectionString As String = SqlConnectionString
    Private Sub Grilla_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.DistanciasXCiudad' Puede moverla o quitarla según sea
necesario.
Me.DistanciasXCiudadTableAdapter.Fill(Me.EcuRutasDataSet.DistanciasXCiu
dad)
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.CiudadesDestino' Puede moverla o quitarla según sea
necesario.
Me.CiudadesDestinoTableAdapter.Fill(Me.EcuRutasDataSet.CiudadesDestino
)
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.CiudadesOrigen' Puede moverla o quitarla según sea
necesario.
Me.CiudadesOrigenTableAdapter.Fill(Me.EcuRutasDataSet.CiudadesOrigen)
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.Ciudades' Puede moverla o quitarla según sea necesario.
Me.CiudadesTableAdapter.Fill(Me.EcuRutasDataSet.Ciudades)
```

```

'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.Ciudades' Puede moverla o quitarla según sea necesario.
    Me.CiudadesTableAdapter.Fill(Me.EcuRutasDataSet.Ciudades)
End Sub

Private Sub BtCalcular_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtCalcular.Click
    '    Dim _BmBase As BindingManagerBase =
BindingContext(EcuRutasDataSet.Ciudades)
    '

Me.DistanciasXCiudadTableAdapter.Fill(Me.EcuRutasDataSet.DistanciasXCiu
dad)

    Dim _iOrigen As String
    Dim _iDestino As String
    _iDestino = ComboBoxDestino.SelectedValue
    _iOrigen = ComboBoxOrigen.SelectedValue
    Dim _sqlString As String
    '    _sqlString = "SELECT Distancia FROM Distancias WHERE
(CiudadOrigen = 2) AND (CiudadDestino = 3)"
    _sqlString = "SELECT Distancia FROM Distancias WHERE
(CiudadOrigen = " & _iOrigen & ") AND (CiudadDestino = " & _iDestino & ")"
    Dim _connection As New SqlConnection(connectionString)
    Dim _Comando As New SqlCommand(_sqlString, _connection)
    ' crear DataReader
    Dim _DataReader As SqlDataReader
    Try
        _connection.Open()
        _DataReader = _Comando.ExecuteReader()
        While _DataReader.Read()
            LblDistancia.Text = _DataReader("Distancia").ToString & "
Kilómetros"
        End While
    End Try
End Sub

```



```

        _connection.Close()
    Catch ex As Exception
        ' Catch ex As System.Data.ConstraintException
        MessageBox.Show(_sqlString & " " & ex.Message)
    End Try
End Sub
End Class

```

7.3. Mantenimiento de Ciudades

```

Imports System.Data.SqlClient
Imports System.Data
Public Class AltasCiudad
    Protected Const SqlConnectionString As String = _
        "Server=(local);" & _
        "DataBase=;" & _
        "Integrated Security=SSPI"
    Protected connectionString As String = SqlConnectionString
    Private Sub Alta_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtPrimero.Click
        BindingContext(EcuRutasDataSet.Ciudades).Position = 0
        MostrarPosicion()
    End Sub
    Private Sub Altas_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.Ciudades' Puede moverla o quitarla según sea necesario.
        EcuRutasDataSet.Clear()
        Me.CiudadesTableAdapter.Fill(Me.EcuRutasDataSet.Ciudades)
        MostrarPosicion()
    End Sub

```

```

Private Sub MostrarPosicion()
    Dim _BmBase As BindingManagerBase =
BindingContext(EcuRutasDataSet.Ciudades)
    Dim _iTotal As Integer = _BmBase.Count
    Dim _iPos As Integer
    If _iTotal = 0 Then
        LblNoRegistros.Text = "# Registros"
    Else
        _iPos = _BmBase.Position + 1
        LblNoRegistros.Text = _iPos.ToString & " de " & _iTotal.ToString
        MostrarRegistro(_iPos)
    End If
End Sub

Private Sub MostrarRegistro(ByVal _iRegistro As Integer)
    Dim _sqlString As String
    ' _sqlString = "SELECT Id_Ciudad, Nombre FROM Ciudades
WHERE(Id_Ciudad = " & _iRegistro & ")"
    ' _sqlString = "SELECT TOP 1 Id_Ciudad, Nombre FROM Ciudades "
    _sqlString = "SELECT TOP " & _iRegistro & " Id_Ciudad, Nombre FROM
Ciudades "
    Dim _connection As New SqlConnection(connectionString)
    Dim _Comando As New SqlCommand(_sqlString, _connection)
    ' crear DataReader
    Dim _DataReader As SqlDataReader
    Try
        _connection.Open()
        _DataReader = _Comando.ExecuteReader()
        While _DataReader.Read()
            Lbl_Id_Ciudad.Text = _iRegistro.ToString
            FrCiudad.Text = _DataReader("Nombre")
        End While
    End Try
End Sub

```

```

        _connection.Close()
    Catch ex As Exception
        ' Catch ex As System.Data.ConstraintException
        MessageBox.Show(_sqlString & " " & ex.Message)
    End Try
End Sub

Private Sub BtAnterior_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtAnterior.Click
    BindingContext(EcuRutasDataSet.Ciudades).Position -= 1
    MostrarPosicion()
End Sub

Private Sub BtSiguiente_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtSiguiente.Click
    BindingContext(EcuRutasDataSet.Ciudades).Position += 1
    MostrarPosicion()

End Sub

Private Sub BtUltimo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtUltimo.Click
    BindingContext(EcuRutasDataSet.Ciudades).Position = _
BindingContext(EcuRutasDataSet.Ciudades).Count - 1
    MostrarPosicion()
End Sub

Private Sub btAñadir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btAñadir.Click
    Dim _sqlString As String
    Dim _wciudad As String
    _wciudad = "" & FrCiudad.Text & ""
    _sqlString = "INSERT INTO Ciudades ( Nombre )VALUES (" & _wciudad
& ");"
    Dim _connection As New SqlConnection(connectionString)

```

```

Dim _Comando As New SqlCommand(_sqlString, _connection)
Dim _iResultado As Integer
Try
    _connection.Open()
    _iResultado = _Comando.ExecuteNonQuery()
    _connection.Close()
    MessageBox.Show(_iResultado & " Registros añadidos")
Catch ex As Exception
    '    Catch ex As System.Data.ConstraintException
    MessageBox.Show(ex.Message)
End Try
End Sub

Private Sub btBorrar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBorrar.Click
    Dim _xRegistro As String
    Dim _sqlString As String
    _xRegistro = Lbl_Id_Ciudad.Text
    _sqlString = "DELETE FROM Cuidades WHERE (Id_Ciudad = " &
_xRegistro & ")"
    Dim _connection As New SqlConnection(connectionString)
    Dim _Comando As New SqlCommand(_sqlString, _connection)
    Dim _iResultado As Integer
    Try
        _connection.Open()
        _iResultado = _Comando.ExecuteNonQuery()
        _connection.Close()
        MessageBox.Show(_iResultado & " Registros Eliminados")
    Catch ex As Exception
        '    Catch ex As System.Data.ConstraintException
        MessageBox.Show(ex.Message)
    End Try

```

```

End Sub

Private Sub btBuscar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBuscar.Click
    Dim _BmBase As BindingManagerBase =
BindingContext(EcuRutasDataSet.Ciudades)
    Dim _iRegistro As Integer = 0
    Dim _sqlString As String
    ' _sqlString = "SELECT Id_Ciudad, Nombre FROM Ciudades
WHERE Nombre = '" & ctBuscar.Text & "'"
    _sqlString = "SELECT Id_Ciudad, Nombre FROM Ciudades WHERE
Nombre LIKE '" & ctBuscar.Text & "'"
    ' _sqlString = "SELECT Id_Ciudad, Nombre FROM Ciudades
WHERE Nombre LIKE '%" & ctBuscar.Text & "'"
    Dim _connection As New SqlConnection(connectionString)
    Dim _Comando As New SqlCommand(_sqlString, _connection)
    ' crear DataReader
    Dim _DataReader As SqlDataReader
    Lbl_Id_Ciudad.Text = "0"
    FrCiudad.Clear()
    Try
        _connection.Open()
        _DataReader = _Comando.ExecuteReader()
        While _DataReader.Read()
            Lbl_Id_Ciudad.Text = _DataReader.Item(0)
            FrCiudad.Text = _DataReader.Item(1)
            _DataReader.NextResult()
        End While
        If Lbl_Id_Ciudad.Text = "0" Then
            MessageBox.Show("No se encontraron registros coincidentes a " &
ctBuscar.Text)
        End If
    End Try
End Sub

```

```

        _connection.Close()
    Catch ex As Exception
        ' Catch ex As System.Data.ConstraintException
        MessageBox.Show(_sqlString & " " & ex.Message)
    End Try
End Sub

Private Sub Actualizar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Actualizar.Click
    Dim _sqlString As String
    Dim _wciudad As String
    Dim _wld_Ciudad As String
    _wciudad = "" & FrCiudad.Text & ""
    _wld_Ciudad = "" & Lbl_Id_Ciudad.Text & ""
    _sqlString = "UPDATE Cidades SET Nombre = " & _wciudad & "
WHERE (Cidades.Id_Ciudad = " & _wld_Ciudad & ");"
    Dim _connection As New SqlConnection(connectionString)
    Dim _Comando As New SqlCommand(_sqlString, _connection)
    Dim _iResultado As Integer
    Try
        _connection.Open()
        _iResultado = _Comando.ExecuteNonQuery()
        _connection.Close()
        MessageBox.Show(_iResultado & " Registros Actualizados")
    Catch ex As Exception
        ' Catch ex As System.Data.ConstraintException
        MessageBox.Show(ex.Message)
    End Try
End Sub
End Class

```

7.4. Mantenimiento de distancias

```
Imports System.Data.SqlClient
Imports System.Data
Public Class DistanciasAct
    Protected Const SqlConnectionString As String = _
        "Server=(local);" & _
        "DataBase=;" & _
        "Integrated Security=SSPI"
    Protected connectionString As String = SqlConnectionString
    Private Sub DistanciasAct_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.DistanciasXCiudad' Puede moverla o quitarla según sea
necesario.
Me.DistanciasXCiudadTableAdapter.Fill(Me.EcuRutasDataSet.DistanciasXCiu
dad)
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.CiudadesDestino' Puede moverla o quitarla según sea
necesario.

Me.CiudadesDestinoTableAdapter.Fill(Me.EcuRutasDataSet.CiudadesDestino
)
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.CiudadesOrigen' Puede moverla o quitarla según sea
necesario.

Me.CiudadesOrigenTableAdapter.Fill(Me.EcuRutasDataSet.CiudadesOrigen)
        'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.Ciudades' Puede moverla o quitarla según sea necesario.
```

```

Me.CiudadesTableAdapter.Fill(Me.EcuRutasDataSet.Ciudades)
'TODO: esta línea de código carga datos en la tabla
'EcuRutasDataSet.Ciudades' Puede moverla o quitarla según sea necesario.
Me.CiudadesTableAdapter.Fill(Me.EcuRutasDataSet.Ciudades)
End Sub

Private Sub BtCalcular_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtCalcular.Click
    ' Dim _BmBase As BindingManagerBase =
BindingContext(EcuRutasDataSet.Ciudades)
    '

Me.DistanciasXCiudadTableAdapter.Fill(Me.EcuRutasDataSet.DistanciasXCiudad)

Dim _iOrigen As String
Dim _iDestino As String
_iDestino = ComboBoxDestino.SelectedValue
_iOrigen = ComboBoxOrigen.SelectedValue
Dim _sqlString As String
_sqlString = "SELECT Distancia FROM Distancias WHERE
(CiudadOrigen = " & _iOrigen & ") AND (CiudadDestino = " & _iDestino & ")"
Dim _connection As New SqlConnection(connectionString)
Dim _Comando As New SqlCommand(_sqlString, _connection)
' crear DataReader
Dim _DataReader As SqlDataReader
Try
    _connection.Open()
    _DataReader = _Comando.ExecuteReader()
    While _DataReader.Read()
        LblDistancia.Text = _DataReader("Distancia").ToString & "
Kilómetros"
    End While
    _connection.Close()

```



```

Catch ex As Exception
    ' Catch ex As System.Data.ConstraintException
    MessageBox.Show(_sqlString & " " & ex.Message)
End Try
End Sub

Private Sub BtActualizar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtActualizar.Click
    Dim _sqlString As String
    Dim _iOrigen As String = "0"
    Dim _iDestino As String = "0"
    _iDestino = ComboBoxDestino.SelectedValue
    _iOrigen = ComboBoxOrigen.SelectedValue
    If TxtDistancia.Text > "0" And TxtDistancia.Text < "9999" Then
        _iDistancia = TxtDistancia.Text
    End If
    If _iDistancia > 0 And _iDistancia < 1101 Then
        _sqlString = "UPDATE Distancias SET Distancia = " & _iDistancia & "
WHERE (CiudadOrigen = " & _iOrigen & ") AND (CiudadDestino = " &
_iDestino & ")"
        Dim _connection As New SqlConnection(connectionString)
        Dim _Comando As New SqlCommand(_sqlString, _connection)
        Dim _iResultado As Integer
        Try
            _connection.Open()
            _iResultado = _Comando.ExecuteNonQuery()
            _connection.Close()
            MessageBox.Show(_iResultado & " Registros Actualizados")
        Catch ex As Exception
            ' Catch ex As System.Data.ConstraintException
            MessageBox.Show(ex.Message)
        End Try
    End If
End Sub

```

```
Else
    MessageBox.Show("La distancia sólo admite Valores entre 1 y 1100
Kil{ometros}")
End If
End Sub
End Class
```

VIII. BIBLIOGRAFÍA

- Bristol: Almanaque Mundial 2009: Cuadro de distancias
- Dirección Nacional de Control del Tránsito y Seguridad Vial:
http://www.dnctsv.gov.ec/prog_dnt/distancias.php
- LUIS MIGUEL BLANCO: Desarrollo de Software PROGRAMACIÓN EN VISUAL BASIC .NET.
- MSDN: Foros y librerías: <http://msdn.microsoft.com/es-es/default.aspx>
- Pantigoso Silva(2008) Office 2007 paso a paso de Jaime Pantigoso Silva (ACCESS 2007)
- Stephens Rod: (2008) Visual Basic® 2008 Referencia de Programadores
- Universidad Técnica Estatal de Quevedo: Material Didáctico Modulo 9.

IX. ANEXOS

- Disco Compacto
- Presentación del Proyecto
- Fuentes y demás componentes del sistema
- Base de datos EcuRutas.accdb