



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

---

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
CAMPUS ARAGON

## "MIGRACION DE SOFTWARE"

2007

T E S I S  
QUE PARA OBTENER EL TITULO DE:  
INGENIERO EN COMPUTACION  
P R E S E N T A :  
**LUIS HERNANDEZ PEREZ**

ASESOR:  
ING. VICTOR VELASCO VEGA



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## \*\* MIGRACIÓN DE SOFTWARE \*\*

### INDICE

TEMA	Página
<b>INTRODUCCION</b>	
<b>1. ANTECEDENTES</b>	
➤ 1.1 Concepto de Migración .....	06
➤ 1.2 Paradigmas de la migración.....	07
○ 1.2.1 Cultura Organizacional	
○ 1.2.2 Información y cultura corporativa	
➤ 1.3 Justificación .....	12
○ 1.3.1 Indicadores de cambio	
○ 1.3.2 Agentes de cambio	
○ 1.3.3 Implementación del cambio	
<b>2. ANÁLISIS DE REQUERIMIENTOS PARA LA MIGRACIÓN</b>	
➤ 2.1 Estudio de factibilidad .....	20
○ 2.1.1 Evaluación de proyectos	
○ 2.1.2 Técnico	
○ 2.1.3 Operativo	
○ 2.1.4 Económico	
○ 2.1.5 Legal	
➤ 2.2 Evaluación de Alternativas .....	31
➤ 2.3 Tipos de migración .....	34
○ 2.3.1 Migración Directa	
○ 2.3.2 Migración en paralelo	
○ 2.3.3 Migración modular	
○ 2.3.4 Migración en fases	
<b>3. METODOLOGÍA PARA LA MIGRACIÓN</b>	
➤ 3.1 Situación actual .....	39
➤ 3.2 Análisis del sistema actual .....	40
➤ 3.3 Estándares de programación .....	56
➤ 3.4 Metodología para el desarrollo de sistemas .....	73
➤ 3.5 Ejemplo practico .....	76
➤ 3.6 Documentación .....	149
<b>CONCLUSIONES</b>	
<b>BIBLIOGRAFÍA</b>	
<b>ANEXOS (PROGRAMAS FUENTE CLIPPER-VISUAL BASIC)</b>	

## **OBJETIVOS DE ESTE PROYECTO:**

- Desarrollar bajo una metodología una migración de un sistema de computo, mostrando paso a paso al lector su realización, con el apoyo de herramientas, métodos, y procedimientos.
- Proporcionar al lector una visión general de los elementos que se deben considerar para realizar una migración de software, apoyándolo con los elementos de juicio necesarios para tomar una decisión en la realización de una migración.
- Mostar, a través de un estudio de costo-beneficio los indicadores económicos, para observar si es factible la realización de una migración de sistemas.
- Proporcionar algunas herramientas de modelado, que faciliten el proceso en una migración de software, así como la comunicación en todo momento del proyecto entre analista-usuario, esto es; con apoyo de modelos en papel del futuro sistema, que coadyuven a realizar representaciones abstractas de lo que al final será el sistema.
- Proporcionar herramientas de modelado que permitan administrar el avance del proyecto, como son los diagramas de Pert y Diagramas de Gantt, permitiendo así realizar en cualquier momento adecuaciones en los avances del proyecto.
- Mostrar al lector la tendencia actual de utilizar ambientes gráficos en la realización de los sistemas de computo, realizando en este proyecto ejemplos comparativos de las características de las plataformas gráficas (Windows) y no gráficas (Dos).
- Mostrar, la importancia de la estandarización en el desarrollo de los sistemas de computo, así como la documentación de los mismos.

## **INTRODUCCIÓN:**

En estos días donde la revolución tecnológica sigue creciendo a pasos agigantados y la tecnología existente es modificada constantemente, tanto en hardware como en software, es de suma importancia mantener actualizada la documentación, estándares de programación y políticas de diseño de un sistema de computo.

Por lo anterior las Empresas han realizado cambios trascendentales en sus procesos y han tenido que actualizarse constantemente para no rezagarse con respecto al mercado, debido a la tendencia de cambiar a ambientes mas amigables (gráficos), por lo que en algunos casos se ha tenido que migrar, tanto en su plataforma, como de sistemas.

La migración es una alternativa de solución para actualizar un sistema, debido a que mantiene la información y el procedimiento, pero no debe olvidarse que trae consigo reacciones en su entorno, recuerde que cuando existe un cambio, siempre existe resistencia, tiempo, dinero y esfuerzo.

Debido a que en muchas Organizaciones hoy en día, carecen de esta información o bien, esta incompleta, cuando tienen la necesidad de realizar mantenimiento, migrar o bien hacer una reingeniería del sistema, se enfrentan a una infinidad de problemas, consecuencia de la inexistencia de la documentación antes mencionada. También sucede que la información existente se realizo en el tiempo en que se diseño originalmente el sistema y que ahora es insuficiente o inadecuada para la nueva tecnología de diseño.

Por lo anterior es necesario tener en cuenta que es importante estandarizar y documentar cualquier desarrollo, para evitar problemas futuros. El presente proyecto nos muestra algunos elementos y técnicas que se deben tomar en cuenta para realizar una migración de un sistema de computo, así como una metodología de cómo hacerlo. Se procura en todo momento hacer en donde es posible un comparativo paso a paso del sistema anterior y del nuevo sistema.

Este proyecto esta dividido en tres capítulos, en el capítulo uno se presenta una breve introducción del concepto de migración así como la justificación para realizarla.

El siguiente capítulo nos presenta como realizar un pequeño estudio de factibilidad para determinar si es conveniente hacer o no una migración, alternativas de construcción y finalmente los tipos de migración existentes.

En el último capítulo se enfoca a un ejemplo real, desde el análisis de su situación real y documentación existente, se introduce al lector en como aplicar estándares de programación al nuevo desarrollo, posteriormente se presenta una metodología con un ejemplo real de una migración de un sistema y finalmente se hace mención de la documentación mínima que debe contemplar cualquier proyecto de sistemas, esto no implica que cada una de las organizaciones existentes pueda complementar o mejorar la que en este punto, esta sugerida.

Ahora, si usted quisiera realizar una migración, solamente tendría que seguir la metodología mencionada en el punto 3.4 de este proyecto y apoyarse en el ejemplo práctico, y un buen desarrollador para realizar el sistema. Este ejemplo específico se realizó con una metodología denominada en cascada ("waterfall"), aunque también existen algunas otras metodologías como son el modelo incremental, espiral, de agrupamiento, fuente, remolino, pinball, entre otras.

La metodología denominada clásica o en cascada es una de las más antiguas, pero también una de las más completas, establece una serie de fases, donde al final de cada una de ellas se obtiene una serie de productos (documentos, diagramas, programas), con los cuales se puede evaluar cada una de ellas; y así poder realizar las adecuaciones pertinentes antes de iniciar con la siguiente fase.

# **CAPITULO I**

# **ANTECEDENTES**



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo I "Antecedentes"**

#### **Evolución del Software**

En las últimas décadas el hardware y el software han evolucionado a pasos agigantados, en los primeros años, el software se veía como un complemento, la programación se realizaba sin procedimientos, y la documentación no existía. La programación se realizaba virtualmente sin ninguna planificación y generalmente se realizaba para sistemas a la medida, por lo que solo era ejecutado y depurado por una sola persona.

La segunda era de los sistemas se extiende desde la mitad de la década de los sesenta y hasta finales de los setenta. La multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre-máquina, el software se desarrollaba para computadoras grandes y mini computadoras, crecía el número de sistemas informáticos e iniciaron las bibliotecas de software. Las casas desarrolladoras producían programas con decenas de miles de sentencias fuente, las cuales tenían que ser corregidas cuando se detectaba alguna falla. En esta época el mantenimiento de software comenzó a absorber recursos (horas-hombre) en una medida alarmante.

La tercera era se caracterizó por la llegada de un sinnúmero de microprocesadores, los cuales se utilizaron para un extenso grupo de productos inteligentes desde automóviles, hornos de microondas, robótica, equipos industriales de diagnósticos, pero ninguno más importante que las computadoras personales

La cuarta era es la era de la evolución de los sistemas informáticos, se aleja de las computadoras individuales y sus programas aislados, para el impacto colectivo de las computadoras y su software, evolucionando a potentes máquinas personales controladas con sistemas operativos sofisticados, interconectadas entre sí (redes locales y globales) y acompañadas por aplicaciones de software avanzado. Las arquitecturas informáticas han cambiado de entornos centralizados de grandes computadoras a entornos descentralizados cliente/servidor.





## MIGRACIÓN DE SOFTWARE

### Capítulo I "Antecedentes"

#### Evolución del Software

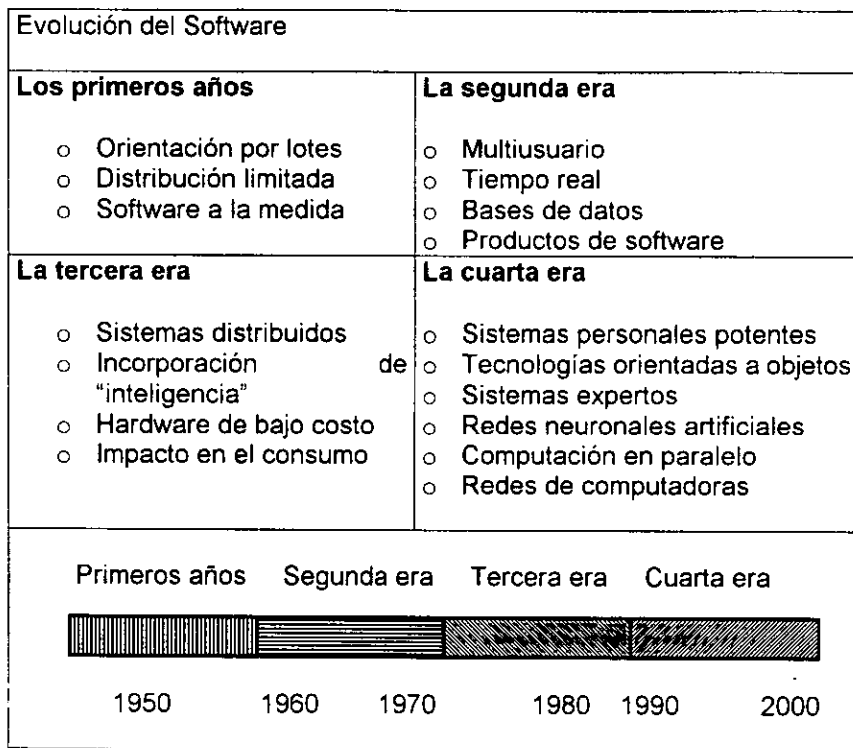


Figura 1.1

Con los avances tecnológicos sobre todo el hardware, las capacidades para realizar un sistema, son mejores, se pueden realizar aplicaciones mucho más poderosas. Por lo tanto si queremos estar a la vanguardia y en competencia con otras empresas, tenemos que utilizar las herramientas actuales del mercado, además debemos de adaptarnos a estos cambios y estar en una constante actualización, de lo contrario quedaremos sin oportunidad de realizar un servicio rápido y oportuno dependiendo del cual se la razón de ser de la empresa.

Un ejemplo de la evolución es el sistema operativo MS-DOS con respecto a Windows. Con MS-DOS se tenía que conocer comandos específicos para realizar una operación (copiar, eliminar, mover, crear archivos), mientras que con Windows se puede realizar cualquier de las operaciones antes mencionadas, sin necesidad de conocer ninguno de esos comandos; ya que con el ambiente operativo gráfico y el apoyo del "mouse" se realizan estas acciones en una forma más fácil y rápida.



# MIGRACIÓN DE SOFTWARE

## Capítulo I "Antecedentes"

### Evolución del Software

A partir de la década de los setenta, los sistemas se multiplicaron y el desarrollo del software se hizo más complejo, por lo que fue necesario realizar enfoques sistemáticos para su desarrollo y mantenimiento, ahora el software es una empresa altamente competitiva y es la cuna de la economía en el mundo y las grandes empresas, como Microsoft que invierte billones de dólares para su desarrollo. A medida que esta cuarta generación progresa surgen nuevas tecnologías y una de las más sobresalientes es la programación orientada a objetos.

Los lenguajes de las tres primeras generaciones se denominaban lenguajes procedimentales y los de la cuarta no procedimentales, a continuación el siguiente cuadro nos muestra su evolución.

**Primera generación:** Contiene los lenguajes de muy bajo nivel es decir los de código binario y los llamados ensambladores en los que cada instrucción se ha sustituido por un código alfanumérico para evitar confusiones con las cifras. En la actualidad se utiliza muy poco y solo en caso de software que necesite mucha potencia (juegos) o acceso directo a los recursos de la máquina (drivers) se usan.

**Segunda generación:** En este primer escalón encontramos lenguajes clásicos como el C y otros similares capaces de estar muy cercanos a la máquina y a la vez ofrecer un interfaz basado en palabras, mucho más sencillo que los binarios. A pesar de su antigüedad, C se mantiene al día de hoy muy vigente y es uno de los lenguajes más usados debido sobretodo a su potencia.

**Tercera generación:** La lejanía de los números binarios ya es muy acusada en este punto encontrándose lenguajes de muy alto nivel como basic, pascal, ada, lisp, lenguajes orientados a objetos como C++, Java, lenguajes de bases de datos como Clipper, etc. Todos ellos son muy habituales hoy en día aunque nunca se suelen usar solos ya que vienen acompañados de otras herramientas de cuarta generación como veremos a continuación.

**Cuarta generación:** Existe un doble razonamiento que justifica la subida de nivel del anterior escalón a este. Por un lado la información, y con ella los datos han tomado una especial preponderancia sobre las acciones en el campo de la programación. Además la potencia de los procesadores se ha disparado vertiginosamente y hemos pasado del viejo interfaz de texto a potentes interfaz gráficos donde todo es mucho más vistoso, intuitivo y fácil de realizar. Basándose en estas premisas las herramientas de cuarta generación nos resuelven un problema indicándonos la solución. Es decir, si necesitamos diseñar un programa para disponer un archivo en pantalla, basta que cojamos una de estas herramientas visuales, y utilizando sus herramientas dibujemos la solución y será ella misma quien genere el código que antes debíamos programar nosotros. Esto no es aún totalmente cierto puesto que en muchos casos debemos introducir nosotros código que en la mayoría de los casos será en alguno de los lenguajes pertenecientes a la tercera generación. Entre las herramientas podemos incluir casi todas las visuales actuales, Visual C++, Visual Basic, Delphi, Visual Java, Visual Café, C++ Builder, Watcom C++, Visual Interdev, etc. En los últimos tiempos existe una nueva corriente de lenguajes de programación entre los que se engloban los lenguajes de etiquetas diseñados para la web junto con múltiples extensiones y lenguajes de script (sucesión de ordenes para formar otra más compleja).



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo I "Antecedentes"**

#### **Evolución del Software**

Hoy en día un detalle muy importante que afecta a todos los lenguajes son las librerías. en las cuales radica mucha de la riqueza de un lenguaje, estas son conjuntos de código ya realizado que podemos reutilizar en nuestros programas y que nos ahorran mucho esfuerzo en la programación. En muchos casos la decisión de escoger uno u otro lenguaje reside en las librerías ya que en general cualquier operación se puede programar de forma más o menos complicada con cualquier lenguaje siendo el tiempo de desarrollo el parámetro determinante en este caso



#### 1.1 CONCEPTO DE MIGRACION

El concepto de "migración" en forma social significa desplazamiento geográfico de individuos o grupos, generalmente por causas económicas o sociales. En este caso el individuo se desplaza a otro lugar (el individuo sigue siendo el mismo – la esencia es la misma-) debido a que en el sistema en que se encuentra ya no satisface sus necesidades, por lo que cambia de sistema esperando encontrar algo mejor. Si aplicamos este concepto en nuestra área informática hablamos de "migración de software", donde existe un desplazamiento de un sistema, de un software de aplicación a otro.

La migración en el área de sistemas es un problema real y muy común, que se presenta en diferentes grados, en la mayoría de las empresas de nuestro país.

En la migración cuando se hace un cambio (evolución) del sistema las operaciones se harán de la misma forma internamente, sólo tendrá una mayor facilidad de entendimiento para los usuarios ya que con la nueva tecnología se pueden hacer sistemas fácilmente operables, lo que traerá consigo un tiempo de respuesta menor y aunque finalmente la información es constante, es decir, sigue siendo la misma la forma como se obtenga deberá de ser mejor que con el anterior sistema.

La migración de los sistemas de información es en general el proceso de trasladar aplicaciones, datos y software de un sistema a otro.

Un problema que puede limitar la migración de un sistema es el aspecto del costo, normalmente las unidades de informática en las organizaciones se ven presionadas por reducciones presupuestales en muchas formas: limitaciones en las compras, recorte de personal, etc.

Otro gran problema es que muchas de las organizaciones han instalado una variedad de plataformas y en cada una de ellas se maneja la información con características muy específicas. El reto de los sistemas abiertos y con ello de los estándares, está en hacer que estas diferentes plataformas trabajen de manera conjunta y esto se puede lograr con una adecuada metodología de migración e integración de los sistemas.

Los estándares de programación permiten mejorar la compatibilidad, dejan mayor libertad a los usuarios para elegir lo que más convenga a sus necesidades tanto de hardware como de software, permiten explotar nuevas tecnologías y generan un ambiente de competencia, tanto en precio como en desempeño en el mercado.

El concepto de migración de software es nuevo por lo que puede causar aceptación en algunas personas o miedo y desconfianza en otras.



## 1.2 PARADIGMAS DE LA MIGRACIÓN

El significado es paradigma en latín y paradeigma en Griego, en su origen significaba un ejemplo ilustrativo, en particular "enunciado modelo", que mostraba todas las inflexiones de una palabra. En el libro *The structure of Scientific Revolutions*, el historiador Thomas Kuhn describía un paradigma como un conjunto de teorías, estándares y métodos que juntos representan un medio de organización del conocimiento; es decir, un medio de visualizar el mundo.

La Programación orientada a objetos (POO) se suele conocer como un nuevo paradigma de programación. Otros paradigmas conocidos son: el paradigma de la programación imperativa (tales como Pascal o C), el paradigma de la programación lógica (PROLOG) y el paradigma de la programación Funcional (Lips).

Jenkins y Glasgow observan que la mayoría de los programadores trabajan en un lenguaje y utilizan solo un estilo de programación. Ellos programan en un paradigma forzado por el lenguaje que utilizan. Con frecuencia no se enfrentan a métodos alternativos de solución de un problema, y por consiguiente tienen la dificultad de ver la ventaja de elegir un estilo más apropiado al problema a manejar.

En la actualidad el cambio a un nuevo paradigma, implica miedo a lo nuevo y a lo desconocido. Cuando tenemos que abandonar aquello que nos fue útil y a tener que trabajar con algo diferente el usuario comienza a temer, ya que tiene que adaptarse y aprender a manejar la herramienta que le es proporcionada por la empresa, pero esto tiene solución ya que el sistema que se desarrolle deberá ser entendible amigable y con ambientes gráficos que motiven al usuario a manejarlo; pero siempre debemos de tener presentes que la migración de viejos a nuevos sistemas, la información es constante, sólo la forma de operar será diferente y mejor.

Para entender mejor el proceso de la migración y todo lo que la rodea, es necesario conocer y entender a la organización a fin de cosiderar todos los elementos que estén involucrados en el proceso de la migración, evitando así fallas y retrasos en la conclusión del proceso.

A continuación se definirá de forma muy general los componentes de la organización que son los elementos internos y externos que influyen en el proceso de cambio.



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo I "Antecedentes"**

#### **Paradigmas de la Migración**

Los componentes esenciales de una organización pueden verse en función del área de trabajo, la cultura, la base de sus activos y los interesados y afectados. Para que una organización funcione sin obstáculos, estos componentes deben estar enfocados hacia los mismos objetivos y estar sincronizados entre sí. La información es el ingrediente clave que permite a una organización lograr y mantener un estado de unidad y armonía.

La organización está formada por personas que se unen para lograr un objetivo en común: crear y ofrecer un producto o servicio. Por ejemplo, un fabricante convierte la materia prima en producto terminado, un banco proporciona servicios financieros y un hospital ofrece servicios médicos. El trabajo para lograr los objetivos de la organización se divide entre las personas de acuerdo a sus habilidades y los objetivos de sus tareas, y luego se unen para lograr una coordinación general. El trabajo incluye actividades físicas y mentales, y en algunos casos una combinación entre ambas.

#### **1.2.1 Cultura de la Organización**

Una definición breve de la cultura organizacional o corporativa "es la forma en que hacemos las cosas". Para bien o para mal es el aglutinante social que mantiene unida a la organización o que, en algunos casos, la desintegra. Es el ambiente diario observado y sentido por quienes trabajan en él. Es el aprendizaje acumulado de la organización según se ve reflejado en las promociones, castigos y recompensas y decisiones. Es la forma en que las personas han aprendido a comportarse en la organización, en particular la cultura corporativa no está escrita en ningún lugar; no se encuentra en el manual de procedimientos; es más bien la forma en que las personas se comportan realmente y lo que realmente pasa dentro de la organización.

La cultura corporativa da a cada organización su personalidad, su singularidad y su significado. Su poder es tremendo y con frecuencia absoluto. Algunas culturas "aceptan el cambio" y motivan la innovación y la aceptación de riesgos. Otras culturas detestan el cambio y la incertidumbre y no desean que nadie mueva el barco, intente ideas no probadas o haga algo diferente. Una cultura realmente enferma es aquella en que la organización está en problemas y la cultura no permite que la organización obtenga ayuda o sea ayudada. Dichas organizaciones, están de hecho enfermas y moribundas.

¿Qué tiene que ver entonces la cultura organizacional con los Sistemas de Información?



## MIGRACIÓN DE SOFTWARE

### Capítulo I "Antecedentes"

#### Paradigmas de la Migración

Muchas culturas tienen una mentalidad de barrera debido a la paranoia o debido a ciertos temores, la cultura siente la necesidad de protegerse a sí misma. Nadie desea el negativismo y sin embargo lo practican entre sí de todas maneras. A decir verdad, muchos de los temores en la cultura provienen de las computadoras, la tecnología informática y los cambios propuestos en el Sistema de Información.

Cualquier cambio en el Sistema de Información va a forzar a la gente a comportarse de una manera diferente, a comportarse de forma que viola claramente las normas culturales. No importa con qué ahínco el personal de sistemas o de alta gerencia afirme los beneficios que todo mundo va a obtener con un nuevo sistema de información, este no será aceptado en forma automática: Al tratar de poner en práctica dichos cambios en el sistema, generalmente provocará una fuerte resistencia. La cultura puede decir que está acostumbrada a producir la información de esta forma y que no desea ningún cambio ni ninguna nueva tecnología informática. Adicionalmente, un nuevo sistema de información puede requerir que se comparta la información entre grupos y es muy posible que estos grupos no confíen entre sí y por lo tanto se nieguen a compartir la información. Algunos incluso pueden sentir que la información se utilizará en contra de ellos. Además, algunos gerentes quizá no desean perder la excusa por sus pobres decisiones. El nuevo sistema de información los dejará desarmados para poder decir, "bueno, si no tenía la información ¿cómo iba a tomar la decisión correcta"? No desean un sistema que los haga responsables.

Los diseñadores de Sistemas de Información no pueden alterar de manera significativa las normas culturales dentro de la organización, pero sí pueden hacer algo para que el sistema de información sea más aceptable para la cultura y hacer que este de su parte. El mejor consejo para lograr la aceptación de este, es hacer que los usuarios de el sistema se involucren en su desarrollo y de esta forma se desarrollen sistemas que funcionen como la gente lo desea y no que la gente funciones como el sistema desea.

No se debe forzar a las personas a que se conviertan en conocedores de computación; las computadoras son las que se deben convertir en conocedoras del ser humano. Los sistemas demasiado amables con el usuario deberían de ser fáciles de usar, que no se viertan como una amenaza y sean un verdadero compañero en el trabajo que le resuelva sus problemas. Por la misma razón, a la gente no se le debe engañar haciéndole creer que el trabajar con el sistema va a ser algo fácil, deben estar concientes de la curva de aprendizaje que encontrarán. Si la gente se involucra en el desarrollo de sistemas y sabe lo que puede esperar, entonces no incurrirán "los choques tecnológicos"



## MIGRACIÓN DE SOFTWARE

### Capítulo I "Antecedentes"

#### Paradigmas de la Migración

Quienes desarrollan sistemas de información tendrán que ser quienes calmen estos temores si se quiere que los sistemas tengan éxito. La tecnología informática, los modelos y los sistemas de información, todos ellos son racionales y sistemáticos, basados en suposiciones sólidas a cerca de los deberes de los trabajadores y acerca de las decisiones que toman las personas, la información que deseen y como la desean. Las personas sin embargo no siempre son racionales y lógicas, los diseñadores de sistemas de información deben de ir más allá de la tecnología y considerar aspectos sociales, psicológicos y antropológicos de la organización y sus relaciones con el sistema de información. Si los diseñadores no cuentan con el apoyo de la cultura y su aceptación colectiva, se enfrentarán a un muro de piedra e incluso el mejor sistema de información tecnológicamente hablando fracasaría.

#### 1.2.2 La Información y la Cultura Corporativa.

A medida que las nuevas herramientas de comunicaciones por computadoras logran entrar en las organizaciones cambian, no solo los patrones de flujo de trabajo, sino también las estructuras básicas y las actitudes. Las conexiones de microcomputadoras a computadoras grandes han descentralizado el control de poder de la información, y como resultado de ello, han transformado la forma en que los gerentes y trabajadores piensan en sus funciones. Desde el nivel más alto hasta el nivel más bajo, es posible, por lo menos en teoría, que todo mundo tenga de esta forma acceso a la misma información.

Las comunicaciones por computadora están cambiando, la forma en que se ve el conocimiento y lo que se puede hacer con él. La cultura corporativa está cambiando como resultado de esto. Los ejecutivos pueden preparar su propia información sin tener que esperar a que alguien más genere sus reportes o tener que arreglárselas con la interpretación de los hechos que hace otra persona

La cultura corporativa se ha desplazado aún más con computadoras portátiles y el acceso remoto a las computadoras grandes (LANs). El cambio es sutil pero significativo. El trabajo se puede hacer en cualquier parte.

Peter Dahow, director de servicios de Administración para la Ciudad de Fort Collins, Colorado, dice que el compromiso de la ciudad para la telecomunicación y la conectividad general, produce mejores actitudes de los empleados. Quienes tienen acceso a la información mediante la tecnología de las comunicaciones pueden ver con mayor claridad y por ellos mismos lo que está sucediendo mientras hay mayor comunicación vertical, también se comparte más la información lateral, y las cosas suceden más rápidamente.





## **MIGRACIÓN DE SOFTWARE**

### **Capítulo I "Antecedentes"**

#### **Paradigmas de la Migración**

Ya antes se maneja a grandes rasgos como y cuales son los factores que motivan a ese cambio y como afecta y reaccionan las diferentes partes de la organización.

La estrategia detrás de cualquier cambio por grande o pequeño que sea, debe de ser supervisada, y no se debe caer en el error de creer que una renovación en las actividades tecnológicas traerá consigo como resultado implícito el control. Uno de los principales interrogantes a los que se encuentran las personas encargados del control de cambio de un organización, es sin duda, ¿Cómo hacer que la nueva tecnología se adapte a la vieja tecnología sin tener que desperdiciar los avances que se tengan?. La cuestión anterior, tiene infinidad de respuestas.

Desde el punto de vista administrativo, el cambio es continuo, es decir, es un suceso que se va dando por sí mismo, de acuerdo a las necesidades y circunstancias a los que la organización se enfrenta, sin embargo existen ocasiones que los que el cambio se hace más palpable, debido que se hace tan necesario que algunos procesos se deben detener por un lapso de tiempo dentro de la organización para poder ajustar la tecnología a las nuevas necesidades, posteriormente se continúa con el mismo movimiento interno y externo de la organización.

Existen factores externos que influyen en el cambio, entre los cuales se puede mencionar, la acción competitiva, las necesidades de los clientes el comportamiento legal del medio ambiente y las tácticas actuales de la organización. Entre los factores internos se tienen: las estrategias de negocios, las prioridades de la Administración la cultura y la estructura de la organización; además de los factores externos e internos existe otro factor muy importante, que es la capacidad y la habilidad de que deben de caracterizar a los elementos humanos encargados del control de cambio.

La formulación del cambio consiste, en las tácticas de control de la nueva tecnología, así como la difusión de ésta dentro de la organización, el cual es el resultado de la investigación a lo largo de un cúmulo de situaciones de cambio tecnológico dentro de diferentes organizaciones.



### 1.3 JUSTIFICACIÓN

El software de una computadora es la llave para la productividad, ya que con su correcta explotación una computadora puede ser un arma poderosa para el desarrollo de múltiples actividades.

Todos los sistemas tienen un ciclo de vida, y en algún momento determinado se tendrán que cambiar o actualizar, ya que estos se vuelven obsoletos. Cuando el software que utilizamos ya no da los resultados esperados, el tiempo de respuesta no es el adecuado, el usuario no se identifica con el sistema (la interfaz no es amigable), o el sistema ha crecido o inclusive ha cambiado, etc., tal vez se debería realizarse un análisis para saber si sólo se da mantenimiento, o es tiempo de "migrar".

Creatividad, innovación y cambio son términos usados comúnmente para realizar una migración, y se definen de la siguiente manera:

**Creatividad.** La generación de una idea

**Innovación.** Modificación de un producto servicio o proceso

**Cambio.** Alteración de las políticas de la organización, estructural, actitudes humanas, así como el comportamiento debido al aumento en el nivel de desempeño. En una implementación ideal del cambio los resultados son evaluados con la retroalimentación, la cual es obtenida con todos aquellos elementos que son afectados por esta situación sus reacciones son requeridas, sus puntos de vista son tomados en cuenta, sus sugerencias son evaluadas y de esta manera se tiene la percepción global de los resultados de la innovación tecnológica.

#### 1.3.1. Indicadores de Cambio.

Dentro de la Organización el mismo movimiento y la necesidad de ella tienden hacer cambiar consciente y inconscientemente a la gente y con ello las actividades de esta, pero existen algunos parámetros que se deben de tomar en cuenta para ir midiendo paulatinamente la necesidad de cambio así como el trato de este, tomando como referencia un lapso de tiempo; a continuación se presentan algunos elementos que se deben de tomarse en cuenta.



## MIGRACIÓN DE SOFTWARE

### Capítulo I "Antecedentes"

#### Justificación

1. Oportunidad de la Información. Las empresas que tienen un proceso de cambio sano, tratar a la información como una ventaja competitiva, esto hace que la información sea un arma de crecimiento.
2. Dirección. Tratar a cada uno de los recursos humanos como una fuente de creatividad. Cuando la gente es tratada con respeto y haciéndolas sentir parte importante de la organización se incrementa la motivación así como la productividad de esta.
3. Factores de Relaciones efectivas. Las presiones a la que la gente trabaja provoca en ella niveles de estrés, este es diminuto con los lapsos efectivos que entre los miembros de la organización surgen.
4. Auto evaluación. Las personas encargada del control de cambio deben de ser lo suficientemente objetivas para juzgar los errores de la propia organización y de su mismo trabajo.
5. Trabajo en Equipo, Confianza, Políticas y Poder. El trabajo en grupo en una manera de interacción y retroalimentación entre los miembros de la empresa, esto cuando a las políticas y lineamientos establecidos genera un clima de confianza y grado de autoridad y responsabilidad suficientemente estable como para lograr el cambio.
6. Estabilización de situación de cambio de la empresa. Dirigir de una manera eficiente y adecuar el ambiente de las áreas involucradas en el cambio, para el logro del cambio.
7. Atención y Actitud. Es muy importante la atención y la actitud de la gente de la organización para con el trabajo y es deber de las personas encargadas del control de cambio elevar los grados de estos dos factores en la organización.

#### 1.3.2 Agentes de Cambio.

Existen dos principales tipos de agentes de cambio, estos inciden en la empresa porque ellos dictan el cambio a seguir en el proceso de renovación de la tecnología; estos son:

- Cambio de la gente
- Cambio del entorno organizacional



## MIGRACIÓN DE SOFTWARE

### Capítulo I "Antecedentes"

#### Justificación

En donde se trata de cambiar a la gente de la organización, se debe pensar en elevar la motivación, así como la actitud de esta para con el trabajo, o en su defecto lo que se trata es de cambiar las circunstancias que rodean a la organización. Para en un momento determinando, saber si es mejor cambiar la actitud de la gente o cambiar las circunstancias en que la organización trabaja.

Es necesario tomar en cuenta los siguientes puntos en la organización.

- La estructura formal de la empresa
- El trabajo que con la persona realiza
- Las líneas de comunicación
- Motivación Organizacional
- Tecnología
- Políticas
- Procedimientos

#### 1.3.3. Implementación del Cambio.

Se puede efectuar cambios en cualquier ambiente organizacional. El nivel y forma de aceptación, así como la estrategia de implementación puede variar, pero aún así, el cambio es posible.

Uno de los problemas al implementar cambios, son las técnicas utilizadas, pues consisten, en manipulación y forzamiento. Utilizando estas técnicas el problema puede resolverse momentáneamente, pero a largo plazo puede tener ciertas implicaciones como consecuencia. El riesgo de la manipulación, es que si la gente se da cuenta de esta, antes que se complete el cambio, puede sabotear el mismo.

El uso de la fuerza implica altos riesgos su uso inicial afectará hasta la finalización del cambio, así como a cambios subsecuentes. La fuerza invita al sabotaje y pone en defensiva a la gente.

La mayoría de los cambios organizacionales, encuentran alguna forma de resistencia humana, la cual puede ser o no tomada en cuenta dependiendo de la confianza con las personas. Un bajo nivel de confianza puede producir una reacción no favorable. Sin embargo, un alto nivel de confianza, puede facilitar la implementación, de hecho a mayor nivel de confianza mayor aceptación al cambio.

Algunos elementos que se deben considerar son:



## MIGRACIÓN DE SOFTWARE

### Capítulo I "Antecedentes"

#### Justificación

1. Tener un diagnóstico organizacional de la facilidad del cambio
    - i. Nivel de confianza
    - ii. Nivel de comunicación
    - iii. Cambio
    - iv. Actitud alrededor del cambio
  2. Estrategia de implementación que tome en cuenta
    - i. Recursos Internos
    - ii. Recursos Externos
    - iii. Combinación de Ambos
  3. Tomar en cuenta el vehículo de implementación
    - i. Orden ejecutivo
    - ii. Entrenamiento de Instrucción
    - iii. Comunicación
  4. Considerar el grado de participación de los administradores en el diseño y la ejecución de la implementa
  5. Considerar la facilidad y el soporte que darán los empleados durante el proceso inicial.
  6. Causa del cambio.
    - i. Rivalidad
    - ii. Comportamiento competitivo
    - iii. Logro de status
  7. Definir si los cambios agregan nuevas responsabilidades al staff que provoquen carga de trabajo.
  8. Definir la ambigüedad que provocara el cambio
  9. Identificar los resistores de cambio
  10. Determinar la causa de que por la que se resistan al cambio.
-



## MIGRACIÓN DE SOFTWARE

### Capítulo I "Antecedentes"

#### Justificación

11. Considerar la urgencia de cambio
12. Incluir la estructura conceptual del cambio:
  - i. Accesos alternos
  - ii. Estrategias alternas
  - iii. Herramientas alternas
  - iv. Calendarización tentativa
13. Definir la necesidad para que el cambio se claramente articulado y comunicado junto con los beneficios esperados por la organización y los efectos individuales.
14. Elaborar un estudio organizacional que muestre los resultados del cambio.
15. Definir los aspectos solicitados por el personal:
  - i. Su reacción al cambio propuesto
  - ii. Sus perspectivas y percepciones
  - iii. Sus sugerencias
  - iv. Sus compromisos.



# MIGRACIÓN DE SOFTWARE

## Capítulo I "Antecedentes"

### Justificación

El siguiente cuadro muestra en forma general las ventajas y desventajas de las plataformas utilizadas durante una migración.

SOFTWARE ORIGEN	
VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> <li>➤ Menor requerimiento tecnológico (memoria, espacio en disco, procesador, etc. )</li> <li>➤ Menores costos de operación y adquisición.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Lenguaje obsoleto</li> <li>➤ Tiempos de respuesta lentos</li> <li>➤ Operación compleja (no amigable)</li> <li>➤ Laborioso en la construcción de pantallas y reportes (ambiente gráfico)</li> <li>➤ Lenguaje no portable</li> <li>➤ Sin ayuda en línea</li> </ul>
<b>M I G R A C I Ó N</b>	
SOFTWARE DESTINO	
VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> <li>➤ Fácil manejo de objetos</li> <li>➤ Tiempos de respuesta rápidos</li> <li>➤ Fácil operación (amigable)</li> <li>➤ Fácil manejo de errores</li> <li>➤ Ayuda en línea</li> <li>➤ Lenguaje portable</li> </ul>	<ul style="list-style-type: none"> <li>➤ Mayor requerimiento tecnológico (memoria, espacio en disco, procesador, etc.)</li> <li>➤ Mayores costos de operación y adquisición.</li> </ul>

Figura 1.1

Si alguna de estas causas no cumple en el software existente, no implica que hay que migrar forzosamente.

## **CAPITULO II**

# **ANÁLISIS DE REQUERIMIENTOS PARA LA MIGRACIÓN**





## **MIGRACIÓN DE SOFTWARE**

### **Capítulo II "Análisis de Requerimientos para la Migración"**

#### **Introducción**

Si se ha propuesto efectuar una migración de un sistema, es necesario llevar a cabo un estudio de factibilidad a fin de establecer una base que permita decidir sobre la posibilidad y conveniencia de utilizar un nuevo sistema de computación, aprovechando la tecnología informática.

Los beneficios que justifiquen el desarrollo de un sistema pueden ser el ahorro en los costos de operación, la reducción del tiempo de proceso de un sistema, mayor exactitud, mejor servicio, una mejoría en los procedimientos de control, mayor confiabilidad y seguridad.

La realización del estudio requiere de dos grupos, uno de los cuales deberá fungir como coordinador del estudio y el segundo como ejecutor del mismo:

Grupo coordinador. Es el mecanismo mediante el cual se hace participar a los funcionarios de la dependencia en las decisiones que habrá de tomarse a lo largo del estudio. Es conveniente que este grupo esté integrado por funcionarios de alto nivel, entre los cuales estén los responsables de las unidades de: Programación, Organización y Métodos e informática. Así mismo, el comité deberá ser presidido por un funcionario de alto nivel jerárquico, de manera que pueda garantizar el acceso necesario a las distintas áreas involucradas ya que de ello depende en gran medida el éxito del estudio.

Grupo Técnico. Deberá estar constituido por diferentes especialistas en disciplinas, tales como: el análisis de sistemas, procesamiento de datos, investigación de requerimientos de información, métodos e instrumentos de captación, técnicas de diseminación y flujos de información, etc. Es recomendable que el responsable de la unidad de informática o en su defecto el de la organización y métodos, dirijan el grupo técnico y que ambos participen en el comité de decisiones.



## 2.1 ESTUDIO DE FACTIBILIDAD.

Un estudio de factibilidad permitirá decidir sobre la posibilidad de modificar total o parcialmente los procedimientos totales de captación, tratamiento y uso de la información. Por lo anterior dichos resultados deberán comprender los beneficios e implicaciones en términos cualitativos y cuantitativos, en el corto, mediano y largo plazo con respecto a su eficiencia, funcionalidad, capacidad de desarrollo, flexibilidad y costo.

Se puede decir que el estudio de factibilidad es en esencia un instrumento de planeación y control en el desarrollo y aplicación de la informática en cada una de las empresas y sirve para recopilar datos relevantes sobre el desarrollo de un proyecto y en base a ello, tomar la mejor decisión, si procede su estudio, desarrollo o implementación.

El Objetivo de un estudio de Factibilidad es, auxiliar a una organización a lograr sus objetivos y cubrir las metas con los recursos actuales. La investigación de la factibilidad consiste en determinar si el proyecto es útil para que la empresa logre sus metas.

En la empresa se cuentan con una serie de objetivos que determinan la posibilidad de factibilidad de un proyecto sin ser limitativos. Estos objetivos podrían ser los siguientes:

- Reducción de errores y mayor precisión en los procesos.
- Reducción de costos mediante la optimización o eliminación de recursos no necesarios.
- Integración de todas las áreas y subsistemas de la empresa.
- Actualización y mejoramiento de los servicios a clientes y usuarios.
- Aceleración en la recopilación de los datos.
- Reducción en el tiempo de procesamiento y ejecución de tareas.
- Automatización óptima de procesamientos manuales.

Un estudio de factibilidad requiere ser presentado con todas las posibles ventajas para la empresa u organización. Pero sin descuidar ninguno de los elementos necesarios para que el proyecto funcione. Para esto dentro de los estudios de factibilidad se complementa con la presentación de dos pasos en la presentación del estudio:

- Requisitos óptimos
- Requisitos mínimos.



El primero se refiere a presentar un estudio con los requisitos óptimos que el proyecto requiera, estos elementos deberán ser los necesarios para que las actividades y resultados del proyecto sean obtenidos con la máxima eficacia.

El segundo paso consiste en un estudio de requisitos mínimos, el cual cubre los requisitos mínimos necesarios para realizar el proyecto y que además debe cumplir las metas y objetivos del mismo, este paso procura hacer uso de los recursos disponibles en la empresa para minimizar cualquier gasto o adquisición adicional.

Un estudio de factibilidad debe presentar claramente los gastos y beneficios que acarreará la puesta en marcha del proyecto.

### 2.1.1 Evaluación de Proyectos

Para evaluar un sistema, debemos considerar las siguientes preguntas:

¿Es la corrección necesaria o simplemente una recomendación?

¿Puede derivarse algún beneficio del cambio? Si es así, ¿Cuáles son ellos y, pueden ser evaluados en dinero, recursos, etc.?

¿Cuál será el costo de la implementación del cambio?

¿Afectará el cambio al costo del sistema? Si es así, ¿en qué dirección y por cuánto?

¿Afectará el cambio a otros elementos del sistema o interconexiones con otros subsistemas? Si es así, ¿Serán necesarios otros cambios? De ser así, ¿son estos cambios posibles? Y si lo son, ¿cuál será su costo?

¿Pesarán los beneficios esperados del cambio más que los gastos que acarrea su puesta en ejecución?

¿Están los recursos (mano de obra, dinero, equipo) disponibles para efectuar el cambio?

Así mismo deben ser evaluados los sistemas con mucho detalle, para lo cual se debe revisar si existen realmente sistemas entrelazados como un todo, o bien si existen programas aislados. La estrategia de desarrollo deberá establecer las nuevas aplicaciones y recursos que proporcionará el área de informática y la arquitectura en que estarán fundamentados.

¿Qué aplicaciones serán desarrolladas y cuando?

¿Qué tipos de archivos se desarrollarán y cuando?

¿Qué lenguajes se utilizarán y en qué software?

¿Qué bases de datos serán desarrolladas y cuándo?

¿Qué tecnología será utilizada y cuando se implementará?



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo II "Análisis de Requerimientos para la Migración"**

#### **Estudio de Factibilidad**

¿Cuántos recursos se requerirán aproximadamente?

Estos recursos (hardware, software y comunicaciones) deberán ser compatibles con la estrategia de la arquitectura de la tecnología, conque se cuenta actualmente.

Para identificar los problemas de los sistemas, primero debemos detectar los síntomas, los cuales son un reflejo del área problemática y después de analizar los síntomas podremos definir y detectar causas y efectos.

Los sistemas debemos evaluarlos de acuerdo con el ciclo de vida que normalmente siguen: requerimientos de usuarios, estudio de factibilidad, diseño general, análisis, diseño lógico, diseño físico, pruebas, implementación, evaluación, modificaciones, instalación y mejoras. Y se vuelve nuevamente al ciclo inicial, el cual a su vez debe comenzar con el de factibilidad.

Normalmente, el criterio final de una evaluación económica para un proyecto de software descansa en la determinación de su "rentabilidad" para el organismo.

Los problemas principales a los que se enfrenta un evaluador son el determinar los valores de las variables que intervendrán en su análisis con un aceptable grado de confiabilidad. Para lograr lo anterior se deberá crear un "modelo" que sea un reflejo lo más cercano posible a la realidad.

Las evaluaciones deberán incluir estimados de costos de operación desglosados en: mano de obra, materiales, costos indirectos, inversiones requeridas, periodo de producción y vida del proyecto.

Con la determinación de las partidas mencionadas establecerán una plataforma de análisis económico básica que permita definir: los beneficios, la recuperación de trabajo horas-hombre, costos de operación y de administración, las depreciaciones, inversiones y capital de trabajo y finalmente los resultados (rentabilidad).

Se deberá evaluar la configuración actual del sistema, tomando en consideración las aplicaciones y el nivel de uso, evaluar el grado de eficiencia con el cual el sistema operativo satisface las necesidades de la instalación y revisar las políticas seguidas por la unidad de informática en la conservación de su programoteca.



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo II "Análisis de Requerimientos para la Migración"**

#### **Estudio de Factibilidad**

Esta sección está orientada a:

- Evaluar posibles cambios en el hardware a fin de nivelar el sistema de cómputo con la carga de trabajo actual o de comparar la capacidad instalada con los planes de desarrollo a mediano y largo plazo.
- Evaluar las posibilidades de modificar el equipo para reducir el costo o bien el tiempo de proceso.
- Evaluar la utilización de los diferentes dispositivos periféricos.

El Estudio de la Factibilidad se apoya en cuatro aspectos básicos:

- Técnico
- Operativo
- Económico
- Legal

#### **2.1.2 Técnico.**

En el análisis técnico se deben de considerar las siguientes preguntas: ¿Qué tecnologías se requieren para conseguir la funcionalidad y rendimiento del sistema? ¿Hay nuevos materiales, métodos, algoritmos, o procesos, que se requieren y cual es el riesgo de desarrollo? ¿Cómo afectará al costo estos elementos de la tecnología?

En este punto será conveniente definir las aplicaciones que migraran a otra plataforma, pues ello será determinante si se llega al paso de definir la adquisición y/o cambio de hardware y software.

#### **Factibilidad Técnica**

El estudio de la factibilidad Técnica es un estudio de la funcionalidad, rendimiento y restricciones que pueden afectar la posibilidad para realizar un proyecto aceptable. La factibilidad técnica es frecuentemente el área más difícil de evaluar, debido a que los objetivos, funciones y rendimiento son de alguna manera difusos.

Ésta se refiere a los recursos necesarios como herramientas, conocimientos, habilidades, experiencia, etc., que son necesarios para efectuar las actividades o procesos que se requiere en el proyecto , generalmente nos referimos a



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo II "Análisis de Requerimientos para la Migración"**

#### **Estudio de Factibilidad**

elementos tangibles (medibles). El proyecto debe de considerar si los recursos técnicos actuales son suficientes o deben complementarse.

El área técnica se puede dividir en dos secciones: equipo y sistemas. Por equipo se entiende simplemente la unidad de procesamiento, los dispositivos adyacentes, el equipo de comunicación de datos y otros elementos asociados. El sistema incluye los métodos y técnicas, así como los programas de computación y los modos de operación. Así pues, con factibilidad técnica, el analista determina simplemente si el diseño preliminar se puede desarrollar e implantar con tecnología existente.

Las consideraciones que se asocian normalmente a la factibilidad Técnica comprenden:

Riesgos de desarrollo. ¿Se puede diseñar la nueva arquitectura de computo de tal forma que las funciones y rendimiento necesarios se consignan dentro de las restricciones determinadas en el análisis?

Disponibilidad de Recursos ¿Hay personas capacitadas para desarrollar el proyecto? ¿Están disponibles para el proyecto otros recursos necesarios?, Hardware, Software, por ejemplo:

- Capacidad de memoria interna
- Unidades de memoria
- Unidades de entrada y salida
- Comunicación de datos
- Número y tipo de canales
- Software necesario para el sistema
- Características especiales.

El nivel de tecnología empleado en un proyecto de programación se refleja en el lenguaje utilizado, la computadora y las herramientas de programación utilizadas. El uso de un lenguaje de alto nivel, en vez de un ensamblador, aumenta la productividad, seguridad, aspectos de autodocumentación y la facilidad de modificar los programas. Los lenguajes modernos de programación, brindan características adicionales, como son el manejo de datos, la compilación independiente, el manejo de interrupciones, así como los mecanismos de concurrencia.

La familiaridad, estabilidad y facilidad de acceso al hardware y al software influyen en la productividad del programador y, por ende, en el costo del proyecto. La productividad se afectará si los programadores tienen que aprender a usar un



nuevo ambiente de programación como parte del proceso de desarrollo del producto, o si se planea adquirir una computadora paralelamente con los programas, o si los programadores tienen un acceso restringido a la máquina.

Las prácticas modernas de programación comprenden el uso de análisis y diseño sistemático, de notaciones estructuradas de diseño, recorridos e inspecciones, de la programación estructurada, de pruebas sistemáticas y del desarrollo de una biblioteca de programas básicos. Las herramientas de programación van desde las herramientas más elementales como ensambladores y depuradores, hasta compiladores y ligadores, incluyendo editores interactivos de texto y sistemas de base de datos, así como también lenguajes procesadores de diseño, analizadores de requisitos de programación, ambientes totales de desarrollo que incluyen herramientas de manejo de configuraciones y verificación automática. El uso de prácticas modernas y herramientas de desarrollo, pueden reducir el esfuerzo de diseño.

Hoy en día, la auditoria de software es una herramienta de control que está apoyando la realización de los sistemas en forma ordenada y sistemática.

### **2.1.3 Operativo.**

La factibilidad operacional se refiere a todos aquellos recursos donde interviene algún tipo de actividad (procesos), depende de los recursos humanos que participan durante la operación del proyecto. Durante esta etapa se identifican todas aquellas actividades que son necesarias para lograr el objetivo, se evalúa y se determina todo lo necesario para realizar el proyecto.

Este aspecto significa que el sistema podrá realizar las funciones señaladas, en el medio actual, con el mismo personal y con los procedimientos existentes. De no ser así, si se requieren cambios, el analista debe señalar dichos cambios e indicar las probabilidades de que puedan lograrse con éxito. En realidad, el aspecto operativo es un problema de relaciones humanas, ya que el diseño del sistema propuesto operativamente nunca tendrá éxito a menos que se reconozca el problema que plantean las personas.

Para esto es necesario consultar a los usuarios del sistema para ver si la solución propuesta satisface sus necesidades y si se puede encajar en la operación del sistema actual.



#### 2.1.4 Económico

La estimación de costos de un producto de programación es una de las más difíciles y erráticas tareas, debido a que no es fácil hacer estimaciones exactas durante la fase de planeación de un desarrollo debido a la gran cantidad de factores desconocidos en ese momento, sin embargo, la práctica normal en los contratos implica un firme compromiso monetario como parte del estudio de factibilidad. Lo anterior, aunado a la naturaleza competitiva de este negocio, es un factor que contribuye a los retrasos de entrega y sobregiro en presupuesto tan comunes en los proyectos de programación.

Reconociendo este problema, algunas organizaciones utilizan una serie de estimadores de costos; se prepara un estudio preliminar durante la fase de planeación y se presenta en la revisión de la factibilidad del proyecto.

El informe de costo-beneficio es un instrumento de gestión para uso y guía del desarrollo del sistema, sirve para evaluar su rentabilidad, las inversiones que precisa y ordena por prioridades los objetivos, basándose en el aprovechamiento de los desarrollos parciales, marca los costos de desarrollo del proyecto comparando los benéficos tangibles (medibles directamente en dinero) e intangibles del sistema.

El reporte de costo-beneficio facilita información básica para:

- Decidir la oportunidad de la continuación del desarrollo del sistema al final del estudio de la factibilidad.
- Ajuste o confirmación de prioridades al final de la fase general de análisis y diseño del sistema.
- Contribuir a la toma de decisiones cruciales en el curso del proyecto.
- Evaluar el nuevo sistema tras la puesta en marcha, entregas al usuario y también en la evaluación post-entrega.

Este análisis es complicado por los criterios que varían según las características del problema, el tamaño del proyecto y la recuperación esperada de la inversión como parte del plan estratégico de una organización. Además, muchos de los beneficios derivados de los sistemas basados en computadoras son intangibles (por ejemplo, mayor satisfacción del cliente o mejorar en las tomas de decisiones basadas en datos de ventas preanalizados) y puede ser difícil realizar comparaciones cuantitativas.





## **MIGRACIÓN DE SOFTWARE**

### **Capítulo II "Análisis de Requerimientos para la Migración"**

#### **Estudio de Factibilidad**

Los beneficios de un nuevo sistema se determinan siempre comparando el modo existente de operación contra el sistema proyectado. Dicha comparación se puede hacer en términos absolutos o relativos.

#### **Costos Previos**

- Costo de Consultoría
- Costos reales de compra o alquiler del equipo
- Costos de Acondicionamiento del lugar destinado al equipo
- Costos de Capital
- Costos del personal y directivos encargados
- Costos de Puesta a Punto
- Costos de Instalaciones del equipo de Comunicaciones
- Costos del personal para la Puesta a Punto
- Costo de las actividades de búsqueda y Contratación del personal
- Costos de la Administración requerida para dirigir las actividades de Puesta a Punto.

#### **Costos relativos al proyecto**

- Costo de compra de software de aplicación
- Costos de modificaciones del software para ajustarse a los sistemas locales.
- Costos de personal, generales, etc. Del desarrollo interno de aplicaciones
- Costos de formación del personal en el uso de aplicación
- Costos de procedimientos de recolección de datos y de instalación de datos
- Costos de administración de desarrollo

Hasta aquí se han señalado los costos no recurrentes es decir, aquellos que serán erogados por una sola vez.

#### **Costos continuos**

- Costos del mantenimiento del sistema
- Costos de alquiler
- Costos de depreciación del hardware
- Costos del personal involucrado en las actividades de administración y planeación del proyecto.

Una vez obtenido el costo de la plataforma actual y el costo de la plataforma propuesta, si la diferencia es positiva, existe beneficio, negativa representa el costo adicional, esta es la comparación que puede usarse en términos absoluto.



## MIGRACIÓN DE SOFTWARE

### Capítulo II "Análisis de Requerimientos para la Migración"

#### Estudio de Factibilidad

#### Factibilidad Económica.

La factibilidad económica es una evaluación del costo del desarrollo frente al beneficio final producido por la realización del proyecto.

La justificación económica comprende un amplio rango de aspectos entre los que se encuentran el análisis de costo-beneficio, estrategias de ingresos a largo plazo, impactos en otros productos o entes de beneficios y el costo de los recursos que se necesitan para desarrollo.

- Tiempo de analista
- Costo de capacitación.
- Costo del horas-hombre
- Costo del tiempo del proyecto
- Costo de desarrollo / adquisición

Generalmente la factibilidad económica es el elemento más importante ya que a través de él se solventan las demás carencias de otros recursos, es el más difícil de conseguir y requiere de actividades adicionales cuando no se tiene.

El estudio debe determinar si con los recursos asignados se pueden alcanzar los objetivos del proyecto. Además de ver si merece la pena seguir con el proyecto, o si los beneficios obtenidos del nuevo sistema no compensan los costos (en cuyo caso, el proyecto se terminara).

Con frecuencia a este aspecto de la factibilidad se le llama análisis de costo/efectividad, e implica la comparación del costo con la efectividad de la recomendación.

ESTIMACIÓN DE COSTOS	EVALUACIÓN DE EFECTIVIDAD
Costos del equipo	Beneficios directos. Son un resultado directo del sistema recomendado.
Costos de personal	Incluyen la reducción de errores, la disminución de personal, etc.
Costos de desarrollo	Beneficios indirectos. No provienen necesaria y automáticamente del



## MIGRACIÓN DE SOFTWARE

### Capítulo II "Análisis de Requerimientos para la Migración"

#### Estudio de Factibilidad

	sistema y son difíciles de medir salvo con ayuda de la probabilidad. Entre esos beneficios figuran: una mayor eficiencia, mejores decisiones, más utilidades, mejor servicio a los clientes, etc.
--	---

Finalmente para investigar el costo de un sistema se debe considerar, con una exactitud razonable, el costo de los programas, el uso de los equipos, (compilaciones, programas, pruebas, paralelos), tiempo, personal y operación.

#### **Evaluación económica.**

Los sistemas de información deberán ser verificados y cuidados para que las actividades que se realicen en forma sistemática y completa no perjudiquen a la organización y afecten su economía. Hay que tratar de obtener el mayor beneficio con el equipo disponible e invertir en equipos adicionales sólo cuando esté plenamente justificada la inversión por los beneficios que se obtendrán.

La evaluación económica es la actividad que se encarga de obtener el costo de una aplicación y de cuantificar los beneficios esperados con el objetivo de justificar o no su desarrollo, o comprobar que la aplicación se desarrolló según lo presupuestado.

En el impacto medimos cómo una aplicación de sistemas de información a contribuido o mejorado la eficiencia en el área donde se usa. Así mismo la evaluación después de su implementación es crítica para conocer cómo el sistema opera y dónde pueden necesitarse cambios.

La evaluación económica es importante, puesto que el capital de la organización no es gratuito, debiéndose cuantificar los beneficios y los costos del sistema en términos monetarios para estar en condiciones de justificar o no su desarrollo e implementación.

Cuando la aplicación sea realizada se buscará obtener el costo real contra el beneficio real para comprobar o determinar el porqué de la diferencia con lo presupuestado y/o la calidad de aplicación.



### **2.1.5 LEGAL**

La factibilidad legal es una determinación de cualquier infracción, violación e ilegalidad que pudiera resultar del desarrollo del proyecto. La factibilidad legal comprende un amplio rango de aspectos que incluyen contratos de responsabilidad, infracciones, licencias y otros detalles frecuentemente desconocidos por el personal técnico.



## 2.2 EVALUACIÓN DE ALTERNATIVAS

Una vez que se han respondido preguntas asociadas al análisis técnico, se consideran diferentes alternativas de solución. El estudio de las alternativas es una evaluación de aproximaciones alternativas al desarrollo del proyecto.

El proceso global de evaluación de configuraciones alternativas de la arquitectura de computo, por ejemplo, consiste en calificar cada alternativa en conjunto de "parámetros de evaluación" (criterios de compromiso) que han sido ordenados por su importancia. En general, los parámetros de información se determinan considerando los factores económicos.

### Selección de las plataformas de hardware y software

- Capacidad del hardware y su rendimiento
- Rendimiento del Software
- Soporte del proveedor
- Su disponibilidad del Software y su Aplicación
- Costos
- Flexibilidad de la configuración y capacidad de expansión
- Compatibilidad de los sistemas actuales
- Documentación disponible

### Selección de la plataforma del hardware

Entre las diferentes áreas de evaluación de un equipo de computo, se encuentra en primer término, la evaluación del rendimiento (performance) del equipo.

#### Aspectos en la Evaluación de los Equipos

El problema de selección de equipo de computo, si bien puede plantearse exclusivamente en términos de rendimiento, dejaría sin evaluar otros aspectos, que son tan importantes, como el mismo, y contemplan al sistema como un todo, con el interés de una evaluación óptima del sistema a fin de llevar a cabo una mejor selección.

Existen dos tipos de limitaciones: Obligatorias y Deseables. Si bien es común que las limitaciones se determinen como obligatorias, muchas de ellas podrían ser consideradas como deseables, después de un análisis detallado de las mismas.



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo II "Análisis de Requerimientos para la Migración"**

#### **Evaluación de Alternativas**

Para hacer más claras las categorías se dice que los elementos OBLIGATORIOS son todos aquellos elementos esenciales para las necesidades del comprador, por consiguiente el vendedor deberá cumplir con los requerimientos obligatorios, antes de que el cliente lo pueda considerar como prospecto. Los requerimientos deseables son aquellos que permiten al comprador mejor implantación de la solución. Por lo tanto, si un vendedor en particular no cumple con algunos de estos requisitos podrá ser considerado como un prospecto válido, aún cuando la falta de elementos pueda representar, tal vez alguna penalización en la evaluación general.

La mayoría de las condiciones consideradas como limitaciones al sistema, pueden catalogarse dentro de los siguientes aspectos.

1. Costo de Equipo
2. Fecha de entrega
3. Capacidad para manejo de Aplicaciones
4. Compatibilidad
5. Soporte del Proveedor
6. Confiabilidad
7. Requerimientos de Entrada / Salida
8. Otros

#### **Selección de plataforma de software**

La realización de Sistema de Computo se encamina a la migración de Sistemas de ambientes centralizados a ambientes distribuidos. Es por ello que la selección de la plataforma de software a la migración de los sistemas es una decisión fundamental para el éxito del plan de migración.

Por ejemplo, seleccionar un Sistema Manejador de Base de Datos (DDBMS) implica evaluar diferentes productos, no solo DDBMS sino también, Diccionario de Datos, Paquetes, de Consulta, reportadores y Hardware y Software, preferiblemente al mismo tiempo, esto es, elegir solamente el DDBMS pensar que después se seleccionaran las demás herramientas es una situación que pueden acarrear problemas, pues así se reduce el universo de requerimientos.

No se debe olvidar la adquisición (Si es que aún no existe en la organización) de herramientas CASE (Computer aided Software Engineering), que permiten elevar la productividad y mejorar la eficiencia en los desarrollos.



#### Metodología general para la selección del software

1. Identificar las necesidades de la organización
  - Necesidades Presentes y Futuras
  - Determinación de la primera aplicación que migrará
  - Priorización de las necesidades
  
2. Primera eliminación de alternativas
  - Eliminar los productos que no serán considerados en el proceso de evaluación.
  
3. Evaluación de Proveedores
  - Entrar en contacto con los Proveedores
  - Evaluar el tiempo de respuesta a la solicitud de cotización
  - Llevar a cabo presentaciones con los Proveedores
  
4. Evaluación de Productos
  - Preparar una solicitud de Cotización
  - Hablar y reunirse con otros usuarios que estén usando el producto.
  
5. Recomendación.
  - Hacer una recomendación que incluya los puntos negativos y positivos de cada producto y proveedor participante en el proceso.



### 2.3 TIPOS DE MIGRACIÓN

El tipo de migración será determinado por los siguientes factores dentro de cualquier empresa:

- El giro de la misma
- La necesidad del cambio
- Factores externos
- Factores internos

Basadas en esto pueden existir varios tipos de migración, todos de acuerdo a la situación particular de cada empresa. Estos pueden ir desde un cambio de aplicación a otra 100% compatible o hasta algo más complejo, como lo sería una migración de equipo y aplicación a otros no compatibles.

Independientemente del tipo de migración que se realice, de su magnitud y de la empresa que se trate, siempre estarán dentro de estos, 3 tipos de migración.

- De Equipo.  
Se hace el cambio de aplicación e información a otro equipo.
- De Aplicación.  
Se cambia de una aplicación a otra, permaneciendo la información y el equipo como constantes.
- De Equipo – Aplicación.  
Se realiza el cambio de aplicación y equipo permaneciendo como única constante la información, solo algunas veces se le realizan ajustes a esta, para que la otra aplicación u equipo las tome.

Las migraciones más realizadas en este momento, son las migraciones de equipo-aplicación, siguiéndole las de equipo y por último las de aplicación.

En los tipos de migración de "Aplicación" y de "Equipo-Aplicación", las aplicaciones pueden ser migradas desde cuatro enfoques diferentes, a continuación se da una descripción de ellos.





#### 2.3.1 Migración directa

Una migración directa es la implementación del nuevo sistema y la discontinuación inmediata del anterior, este enfoque para la migración es significativa cuando:

- El sistema anterior se considera totalmente sin valor.
- El sistema nuevo es muy pequeño o muy sencillo y
- El Diseño del nuevo sistema es radicalmente diferente al sistema anterior y no tiene sentido de comparación entre los sistemas.

Ventajas:

- a) Relativamente barato
- b) Los usuarios no tienen posibilidad de utilizar el sistema antiguo.

Desventajas:

- a) Alto riesgo de fracaso.
- b) Retrasos prolongados si suceden errores.
- c) Los usuarios pueden resentir el estar siendo forzados al uso de un sistema no familiar sin alternativas.
- d) No hay una forma adecuada para comparar los resultados.

Cuando se va a emplear este tipo de migración, la actividad de pruebas de sistemas asume mayor importancia y este cambio sólo puede ser exitoso si previamente se han realizado extensas pruebas.

#### 2.3.2 Migración en Paralelo.

Es un enfoque en el que tanto el viejo como el nuevo sistema operan simultáneamente durante cierto tiempo. Es totalmente opuesta a la migración directa. Las salidas de cada sistema se comparan y las diferencias se reconcilian.

Ventajas:

- a) Proporciona un alto grado de protección, para la organización ante un fracaso de un nuevo sistema.
- b) Hay una forma adecuada para comparar los resultados.
- c) Existe mayor seguridad al comparar los resultados.



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo II "Análisis de Requerimientos para la Migración"**

#### **Tipos de Migración**

##### **Desventajas:**

- a) Costos asociados con la duplicidad de instalaciones y personal, para mantener los sistemas duales.
- b) Tendencia en utilizar el sistema anterior.

No obstante, debido a la muchas dificultades experimentadas por las organizaciones en el pasado al implementar un nuevo sistema, este enfoque para la migración ha ganado una amplia popularidad. Cuando el proceso de migración de un sistema incluya operaciones en paralelo, el analista deberá realizar operaciones periódicas con el personal de operación y usuarios respecto al desempeño del nuevo sistema y designar una fecha razonable para su aceptación y discontinuar el sistema anterior.

### **2.3.3 Migración Modular**

La migración modular en ocasiones denominada "enfoque piloto", se refiere a la implementación por partes de un sistema en la organización.

##### **Ventajas:**

- a) El riesgo del fracaso del sistema está localizado
- b) Los problemas identificados en el sistema pueden probarse y corregirse antes de intentar una mayor implementación
- c) Se puede capacitar a otro personal de operaciones en un ambiente en línea, antes de implementar el sistema en sus instalaciones.
- d) Los usuarios se familiarizan con cada módulo con forme llega a ser operacional.

##### **Desventajas:**

- a) El período de migración para la organización puede ser extremadamente largo.
- b) No es factible para todo sistema y organización.
- c) Se debe poner especial atención a las interfases, para que los módulos que están siendo construidos trabajen, de hecho, como un sistema.



#### **2.3.4 Migración en Fases.**

El enfoque en fases es similar al enfoque modular, sin embargo, difiere en que es el mismo sistema el que se segmenta y no la organización.

Ventajas:

- a) Minimizar la rapidez de cambio en una organización dada y los recursos de procesamiento de datos se pueden adquirir gradualmente durante un espacio largo de tiempo.
- b) Permite a los usuarios involucrarse con el sistema gradualmente.
- c) Posibilidad de detección y recuperación de errores sin mucho tiempo perdido.

Desventajas:

- a) Los costos incurridos al desarrollar interfases temporales con los sistemas anteriores.
- b) Su aplicación limitada genera una atmósfera de "desmoralización" en la organización al "nunca completar un sistema".

## **CAPITULO III**

# **METODOLOGÍA PARA LA MIGRACIÓN.**



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

A medida de que las computadoras se hacen más baratas, las Empresas se encuentran con que pueden obtener muchas ventajas de la automatización. Si una Empresa no tiene todavía una computadora, las herramientas de análisis estructurado pueden ser útiles para analizar un sistema manual existente, para entenderlo con claridad antes de tomar la decisión sobre si se debe utilizar una computadora en la Empresa y decidir que partes se deberán automatizar.

Se debe puntualizar, que la metodología, ha sido desarrollada en primer término para situaciones en donde ya existe un sistema computarizado, vinculado con sistemas manuales, procedimientos administrativos y quizás con otros sistemas computarizados y se trata de efectuar algunas mejoras a este conjunto de sistemas.

Las preguntas que deberán contestarse en un estudio inicial son:

- ¿Qué desventajas el sistema en la situación actual?
- ¿Qué mejoras son posibles?
- ¿Qué áreas serán afectadas por el desarrollo o la migración del sistema?

### 3.1 ESTUDIO INICIAL

En cualquier empresa es usual tener un sin número de requerimientos por parte de los diferentes usuarios, a fin de mejorar el servicio de procesamiento de datos. Mientras que algunos de estos requerimientos pueden lograrse mediante una mejor operación, como puede ser el mejorar el tiempo de respuesta, o utilizar una facilidad de consulta existente y otros pueden lograrse perfeccionando los sistemas existentes, como puede ser proveer un nuevo informe a partir de datos ya existentes. Muchos otros implican el desarrollo de un nuevo sistema o la compra de una aplicación que cubra las necesidades de la empresa.

Muchas empresas encuentran que la demanda de nuevos sistemas es varias veces superior a sus posibilidades de construirlo. Y a medida que se implementan más y más sistemas, la tarea de mejorarlos adecuadamente ocupa más tiempo del personal disponible, en algunos de los casos es del 50% al 60% del equipo humano de desarrollo. En consecuencia, los sistemas que se elijan para su construcción o compra, deberán ser aquellos que representen el máximo beneficio para la mayoría de los usuarios de la Empresa. El estudio inicial a veces denominado "Evaluación de requerimientos", es en parte un proceso de filtro para suprimir los requerimientos de desarrollo que no van a ser de suficiente utilidad y



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

puedan hacerse en forma relativamente rápida y barata, ya que un estudio involucra personal y tiempos de los que se está escaso en la mayoría de los casos.

El analista deberá estudiar los requerimientos y reunirse con los diferentes usuarios para obtener los antecedentes de la situación y comenzar a analizar el valor probable del nuevo sistema. Deberá hacerse preguntas como ¿puede indicar la cantidad de ingresos que se están perdiendo a consecuencia de las deficiencias del sistema actual?, ¿Qué costos se podrían evitar mediante un mejor sistema?, ¿Es posible hacer una estimación en dinero sobre el servicio mejorado que se podría dar con un mejor sistema?. También deberá evaluar los siguientes puntos: satisfacción al operar el sistema por parte de los usuarios, tiempos de respuesta, reportes, facilidades y todos los comentarios inherentes al sistema a evaluar, también verificar la documentación existente, como pudieran ser manuales técnicos, manuales de usuario, diagramas de flujo, diagramas de entidad relación, ect., finalmente con toda esta información podremos realizar una valoración general de la forma en como y que realiza dicho sistema y así poder tener un panorama de lo que se tendrá que emigrar.

Cuando se defina la realización de un sistema, se deberán estipular todos los aspectos legales pertinentes, como son fecha de entrega y penalidades de mora. Los usuarios deberán ser invitados a definir en términos muy generales los beneficios intangibles que ven surgir en el sistema mejorado e identificar otras áreas de la Empresa que se verán beneficiados. Si es necesario, los usuarios de esas otras áreas deberán ser entrevistados para obtener una información mas precisa del sistema.

Para este ejemplo específico se migrará un sistema utilizando la metodología denominada clásica o en cascada (waterfall model). El sistema a migrar está realizado en lenguaje Clipper y bases de datos (dbf) y será migrado a una plataforma de Visual Basic y base de datos (access).

El sistema irá mostrando paso a paso la metodología realizada para el desarrollo (migración) del sistema utilizando la información existente del sistema original. Donde sea posible se realizarán ejemplos comparativos en cada una de las fases del desarrollo, tanto de la versión anterior como de la versión actual, también se incluirán algunos ejemplos en el tema de estándares de programación.

### 3.2 FASE DE ANÁLISIS

La función del sistema a migrar, es proporcionar recursos económicos, transporte, hospedaje, alimentación, etc. al personal de una empresa, a partir de este



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

momento a este sistema lo denominaremos "**Sistema de Viáticos**", el cual tiene normas, procedimientos de cálculo y cuotas que se aplican a cada uno de los solicitantes de viáticos y que serán explicados posteriormente.

La documentación existente del sistema original es la siguiente:

- I. Manual de procedimientos para el otorgamiento de viáticos (forma manual).
- II. Modelo lógico del sistema.
- III. Programas fuente.
- IV. Documentación de las bases de datos.

Para poder realizar la migración se tuvo que analizar cada una de las partes antes mencionadas que a continuación se anexan y que son los **elementos básicos** para la realización de ésta.

#### I. Manual de Procedimientos Viáticos y Gastos de Viaje

##### CONTENIDO

1. MARCO LEGAL
2. ALCANCE
3. OBJETIVO
4. VIÁTICOS
  - Tarifa de Viáticos
  - Clasificación de Zonas
5. GASTOS DE VIAJE
6. SOLICITUD DE ANTICIPO PARA GASTOS DE VIAJE Y TRANSPORTACION EN COMISION.
7. COMPROBACION DE GASTOS

#### 1. MARCO LEGAL

Con fundamento en el Artículo 4 del Reglamento de la Ley de Presupuesto, Contabilidad y Gasto Público y de la Norma de Comisiones, Viáticos Nacionales, Viáticos Internacionales y Pasajes para las Dependencias y Entidades de la Administración Pública Federal, emitida por la Secretaría de Hacienda y Crédito Público, se expide el siguiente manual .

Se pretende contar con un adecuado medio de control para normar el procedimiento de asignación, ministración y comprobación de viáticos y gastos de viaje del personal.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

#### 2. ALCANCE :

De aplicación exclusiva para el organismo.

#### 3. OBJETIVO :

Establecer los lineamientos para la asignación y aplicación de gastos, originados en comisiones de trabajo, los cuales se registran en las cuentas correspondientes del Catálogo Contable.

#### 4. VIÁTICOS :

- 4.1 Cantidad que se otorga al personal para cubrir sus gastos de hospedaje, alimentación, transporte local, tintorería, lavandería, y cualquier otro gasto similar o conexo a éstos durante el tiempo que dure la comisión.

Para mandos medios y superiores la tarifa incluye los taxis en comisiones mayores a 24 horas, del domicilio-aeropuerto y aeropuerto-lugar de trabajo en la salida y los correspondientes al regreso, comprobables en todos los casos los desplazamientos a partir del aeropuerto. Para el resto del personal estos taxis se pagarán por separado.

- 4.2 Los viáticos se calcularán por cada día o fracción que dure la comisión, salvo las menores a 24 horas que serán del 100%.

- 4.3 En el caso de personal que viaje con un funcionario de mayor jerarquía en el desempeño total de la comisión, se les asignará la misma cuota, a excepción de los chóferes, quienes en todos los casos percibirán tarifa del resto del personal.

- 4.4 Para la asignación de los viáticos parciales se tomará como base la hora de salida y retorno al lugar de origen, aplicándose a la cuota diaria el porcentaje que corresponda, en base a lo siguiente:

SI SALE DE :

SI LLEGA DE:

00:01 HRS.	A	10:00 HRS.	100 %	00:01 HRS.	A	11:00 HRS.	30 %
10:01 HRS.	A	16:00 HRS.	70 %	11:01 HRS.	A	20:00 HRS.	70 %
16:01 HRS.	A	00:00 HRS.	30 %	20:01 HRS.	A	00:00 HRS.	100 %

- 4.5 Las cuotas establecidas para viáticos autorizadas por el Comité Técnico son:





## MIGRACIÓN DE SOFTWARE

Capítulo III "Metodología para la migración"

Situación actual

### TARIFA DE VIÁTICOS NACIONALES POR UN TIEMPO MENOR DE 24 HORAS ( CUOTA DIARIA EN PESOS)

NIVEL JERARQUICO	Z O N A			
	I	II	III	IV
Director General, Director de Área, Contralor Interno y Subdirector de Área.	193.00	211.00	229.00	246.00
Gerente, Subgerente, Gerentes Regionales y Subgerente Regional.	160.00	171.00	182.00	193.00
Resto del Personal.	126.00	137.00	149.00	160.00

### TARIFA DE VIÁTICOS NACIONALES POR UN TIEMPO MAYOR DE 24 HORAS ( CUOTA DIARIA EN PESOS)

NIVEL JERARQUICO	Z O N A			
	I	II	III	IV
Director General, Director de Área, Contralor Interno y Subdirector de Área.	710.00	850.00	906.00	1,178.00
Gerente, Subgerente, Gerentes Regionales y Subgerente Regional.	525.00	637.00	706.00	889.00
Resto del Personal.	401.00	487.00	561.00	656.00

### " CLASIFICACION DE ESTADOS Y MUNICIPIOS PARA LA ASIGNACION DE VIÁTICOS "

ENTIDAD FEDERATIVA.	ZONA I.	ZONA II.	ZONA III.	ZONA IV.
Aguascalientes		Todo el Estado		
B. C.				Todo el Estado



## MIGRACION DE SOFTWARE

### Capitulo III "Metodología para la migración"

#### Situación actual

				Estado
B. C. S.			Resto del Estado	Los Cabos, Loreto, Nopolo, Mulege
Campeche		Resto del Estado	Cd. del Carmen	
Coahuila		Resto del Estado	Piedras Negras, Cd. Acuña	
Colima	Resto del Estado			Manzanillo
Chiapas		Resto del Estado	Tuxtla Gutiérrez	Tapachula
Chihuahua		Resto del Estado		Cd. Juárez
Distrito Federal			Todo el D.F.	
Durango		Todo el Estado		
Guanajuato		Resto del Estado	Guanajuato, León, San Miguel Allende	
..... etc.				

Cuando se utilice vehículo, se pagará el consumo de gasolina por kilómetro. La Gerencia de Presupuesto y Contabilidad, definirá el número de kilómetros entre el lugar de adscripción y el de comisión; dicha cantidad se dividirá entre 5 (número de kilómetros promedios recorridos por litro de gasolina), el resultado se multiplicará por el precio vigente del litro de gasolina.

Para el cálculo de la cantidad a pagar por vehículo utilizado se aplicará como base la siguiente tabla de distancias por carretera o en su defecto las establecidas en la Tabla de Distancias emitida por la Guía Roji vigente y en los casos en que las comisiones no sean exactamente a la ciudad de referencia se consignarán las distancias de la bitácora del vehículo.



## MIGRACIÓN DE SOFTWARE

Capítulo III "Metodología para la migración"

Situación actual

DE MÉXICO A:	KMS.	DE MÉXICO A:	KMS.
Acapulco, Gro.	411	Mérida, Yuc.	1,458
Aguascalientes, Ags.	504	Mexicali, B. C. N.	2,705
Angangueo, Mich.	200	Monterrey, N.L.	989
Barra De Navidad, Jal.	910	Morelia, Mich.	242
Campeche, Camp.	1,295	Nogales, Son.	2,283
Cd. Altamirano, Gro.	289	Nueva Rosita, Coah..	1,218
Cd. Cuauhtemoc, Chih.	1,355	Nuevo Laredo, Tamps.	1,187
Cd. Juárez, Chih.	1,820	Oaxaca, Oax.	488
Cd. Oregón, Son.	1,783	Orizaba, Ver.	280
Cd. Victoria, Tamps.	710	Parral, Chih.	1,361
Coatzacoalcos, Ver.	759	Pachuca, Hgo.	92
Colima, Col.	688	Piedras Negras, Coah.	1,302
Concepción Del Oro, Zac.	904	Poza Rica, Ver.	297
Concordia, Sin.	1,220	Puebla, Pue.	125
Culiacán, Sin.	1,430	Puerto Juárez, Quintana Roo.	1,834
Cuernavaca, Mor.	85	Querétaro, Qro.	215
Chihuahua, Chih.	1,445	Sabinas, Coah..	1,205
Chilpancingo, Gro.	282	Saltillo, Coah.	869
..... etc.			

### 5. GASTOS DE VIAJE

Comprende las erogaciones realizadas en cumplimiento de una comisión con base a las tarifas señaladas en este Manual.

#### 5.1 Gastos de Hospedaje:

Comprende los gastos erogados por concepto de habitación; así como, los impuestos correspondientes, y solo se pagarán contra la presentación del comprobante original del establecimiento que prestó el servicio, el cuál deberá reunir los requisitos fiscales y legales establecidos.

#### 5.2 Gastos de Alimentación:

Los alimentos se comprobarán con las notas de consumo las cuales deberán reunir los requisitos fiscales y legales establecidos.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

- 5.3 Gastos por Taxis:  
Son los gastos efectuados por los servicios de automóviles de alquiler para el traslado a la terminal de transportación, de ésta al hotel del lugar de la comisión y, viceversa .
- 5.4 Gastos de Transportación:  
Son los gastos realizados por concepto de pasajes foráneos, cualquiera que sea el medio de transporte a utilizar.
- 5.5 Gastos Extraordinarios:  
Son las erogaciones que el comisionado se ve obligado a realizar en el desempeño de sus actividades y que corresponden a situaciones imprevistas.

#### 6. SOLICITUD DE ANTICIPO

- 6.1 Tramitación de Anticipo:  
Se efectuará por medio del formato "**Solicitud de Anticipo para gastos de viaje y transportación en comisión**" que el sistema computarizado contempla, con un mínimo de **2 días hábiles** previo a la salida programada.
- 6.2 Bases para su Cálculo:  
Los anticipos se calcularán de acuerdo al número de días que durará la comisión.
- 6.3 Negativas:  
No se tramitarán anticipos a personas que no hayan comprobado las comisiones anteriores.  
No se tramitarán viáticos que incluyan días inhábiles, salvo días festivos que estén intercalados dentro del periodo que abarque la comisión y que la permanencia sea más económica que el efectuar su regreso, previa autorización y justificación del funcionario responsable que comisiona.

#### 7. COMPROBACION DE GASTOS

##### APLICACIÓN DE TARIFA EN LA COMPROBACION

La comprobación de Gastos se efectuará con base en los siguientes lineamientos:

Al presentar la "**Comprobación de Gastos de Viaje en Comisión**", se deberá anotar por separado el I.V.A. que se pagó. El informe contendrá un breve resumen de las actividades realizadas y de las conclusiones y/o resultados obtenidos.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

- 7.1 La comprobación deberá presentarse en un plazo máximo de **3 días hábiles**, contados a partir de la fecha de retorno, utilizando el formato **"Comprobación de Gastos de Viaje en Comisión "** que el sistema computarizado contempla, en el mismo tipo de moneda en la que se haya entregado el anticipo y referirse a una sola comisión, anexando copia de la **"Solicitud de Anticipo para Gastos de Viaje y Transportación en Comisión"** y los comprobantes de gastos originales sin tachaduras, enmendaduras o cualquier otra alteración.

**La presentación extemporánea será causa del descuento vía nómina.**

- 7.2 Comisiones a dos o más zonas.  
Para determinar la tarifa de viáticos, se tomará en cuenta el lugar donde se pernocte.
- 7.3 La comprobación no deberá exceder el monto que conforme al día y hora de salida y llegada corresponda, el excedente de un día podrá compensarse con otro siempre y cuando todos los comprobantes de alimentación y hospedaje cumplan los requisitos fiscales.

El Manual de Procedimientos antes referido nos proporcionó el procedimiento de cálculo para proporcionar viáticos con base a su destino, fecha y hora de salida, fecha y hora de llegada y su nivel jerárquico



# MIGRACIÓN DE SOFTWARE

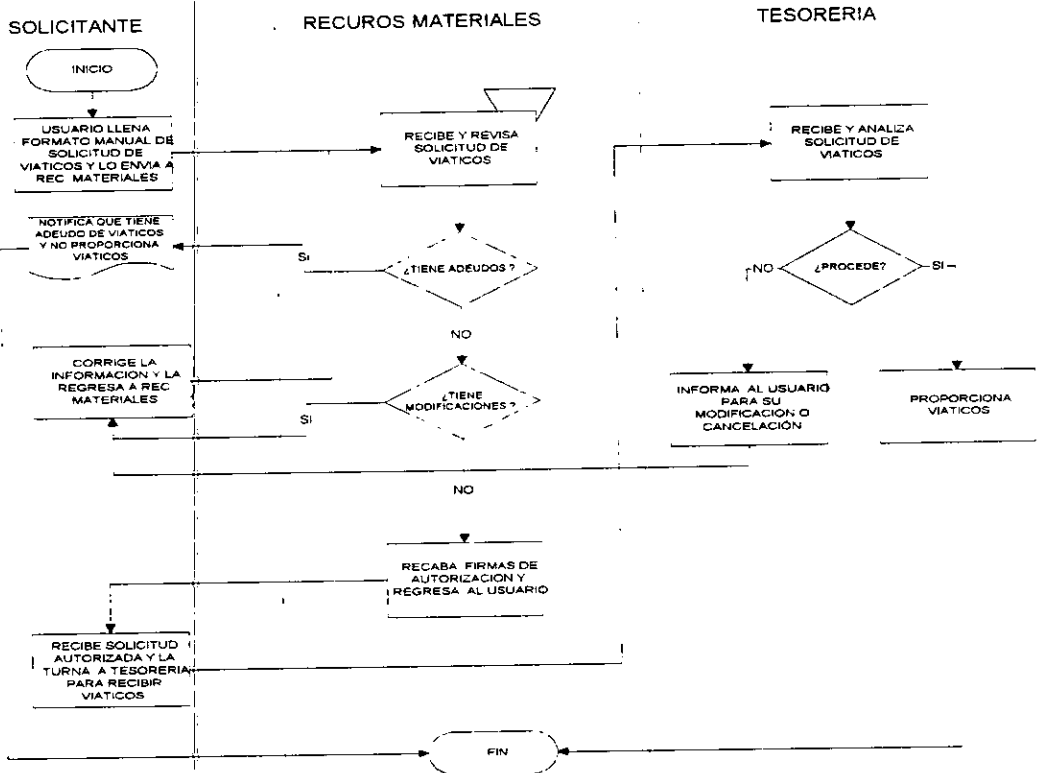
## Capítulo III "Metodología para la migración"

### Situación actual

## II. Modelo lógico

### Modelo lógico actual

DIAGRAMA DE FLUJO DE LA SOLICITUD DE GASTOS DE VIAJE Y VIATICOS (FORMA MANUAL)



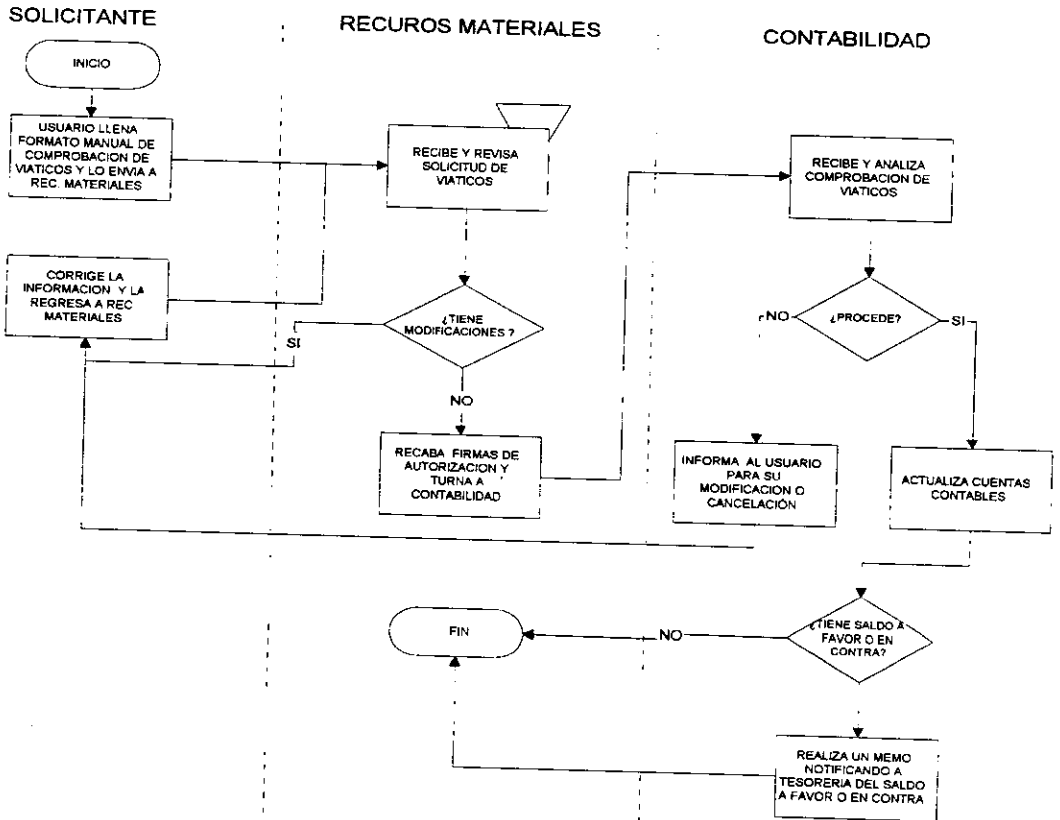


# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Situación actual

DIAGRAMA DE FLUJO DE COMPROBACION DE GASTOS DE VIAJE Y VIATICOS (FORMA MANUAL)





## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

### III. Programas fuente.

La siguiente tabla nos muestra los programas fuente del sistema anterior.

Módulo	Nombre del Programa	Función específica
General	locks	Procedimientos generales
Anticipos	Via000	Anticipos de viáticos
Anticipos	Via001	Baja de anticipos
Comprobación	Via200	Captura de comprobación
Reportes	Via301	Anticipos de gastos
Reportes	Via302	Solicitud de transportación aérea
Reportes	Via303	Comprobación de gastos
Reportes	Via304	Devolución de viáticos
Reportes	Via305	Reporte individual
Reportes	Via306	Reporte general
Catálogos	Via401	Catálogos de destino
Catálogos	Via402	Catálogos de tarifas
Catálogos	Via403	Catálogos de empleados.
Catálogos	Via404	Catálogos de puestos
Catálogos	Via405	Catálogos de áreas.
Utilerías	Via500	Parámetros
Utilerías	Via501	Reindexado
Utilerías	Via502	Parámetros
Estadísticos	Via600	Estadísticos
Menú Principal	Viáticos	Pantalla principal





## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

Como información se anexa parte del programa fuente principal del sistema anterior.

```
//-----  
// Sistema: VIÁTICOS, Proceso de Viáticos      -  
// Autor: *****                            -  
//-----  
*--- Definición de Medio Ambiente -----  
#include "Inkey.ch"  
  
#define cBACKCOLOR 3 // blue on black  
#define mBACKCOLOR 7 // white on black  
****  
  
// background pattern for 1.50 stuff  
STATIC pattern := "oooooooo+++++++222222222+++++++"  
IF !ISCOLOR()  
    BCOLOR=mBACKCOLOR  
ELSE  
    BCOLOR=cBACKCOLOR  
ENDIF  
cls( bcolor, pattern)  
SAVE SCREE TO BEFORE  
SETHANDLES(50)  
IF ISCOLOR()  
    COLOR3='W/N,N/W'  
    COLOR2='B/BG,BG/B'  
    COLOR1='B/W,W/B'  
    COLOR4='W+/B+,N/W'  
    COLOR5='N/W,W+/B+'  
ELSE  
    COLOR3='W/N,N/W'  
    COLOR2='N/W,W/N'  
    COLOR1='N/W,W/N'  
    COLOR4='W/N,N/W,,,N/W'  
    COLOR5='N/W,W/N,,,W/N'  
ENDIF  
  
SET EXCLUSIVE OFF  
SET PROCEDURE TO LOCKS  
SET STATUS OFF  
SET SAFETY OFF  
SET DATE BRIT
```



**MIGRACIÓN DE SOFTWARE**  
**Capítulo III "Metodología para la migración"**  
**Situación actual**

```
SET DELETED ON
SET TALK OFF
SET CONFIRM OFF
set console off
set score off
set wrap on
*----- Pantalla principal -----
store 1 to aa,bb,cc,dd
store 0 to nw
public p_Ayuda := '00000', p_Date, p_Barra

p_Date := Date()
sigue = .t.
set color to b/w
* win(0,0,2,79,1,0) && abre la pantalla del centro hacia afuera
@ 0,0 to 2,79 double
set color to b/w
@ 0,21 say ' Organización X '
@ 1,1 say 'Gerencia de Sistemas, VIÁTICOS-1.0 '
@ 1,48 say 'Gerencia de Recursos Materiales'
@ 3,0 to 24,79 double
//----- About-----

... más código ...

//----- abre archivos-----

más código...

*****
* Función que Activa el Menu *
*****

.... más código ....

//-----
// Menu31 - Función que activa reportes
//-----

//-----
// Menu41 - Función que activa catálogos
//-----
```



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

```
más código...
//-----
// Menu51 - Función que activa utilerías
//-----

... más código...

//-----
// Abre todos los archivos de un jalón
//-----

... más código ...
//-----FIN-----
```

Documentación de las bases de datos

Directorio de C:\viáticoslanterior

NOMBRE	TAMAÑO	FECHA	HORA	TIPO
VIATICOS	106.988	06/11/00	7:10	VIATICOS.DBF
ADN01	95.190	08/04/98	12:34	adn01.dbf
ADN26	487	04/09/96	10:28	ADN26.DBF
ADN26DES	395	24/07/95	18:07	ADN26DES.DBF
COMPROBA	2.184	01/03/96	11:50	COMPROBA.DBF
DDU	13.859	04/09/96	14:27	DDU.DBF
DESTINOS	3.922	09/02/96	13:01	DESTINOS.DBF
FGCENFE	1.422	18/07/95	18:01	FGCENFE.DBF
MAT1	29.216	14/04/98	5:26	MAT1.DBF
RH1	29.454	14/04/98	5:27	rh1.dbf
RH1BAK	29.097	11/04/97	13:20	rh1bak.dbf
RH2	14.352	22/05/97	12:55	RH2.DBF
RH3	5.793	26/03/96	11:27	RH3.DBF
TARIFAS	3.196	25/09/97	11:28	TARIFAS.DBF
TEMP	395	22/05/97	12:54	TEMP.DBF

15 archivos      335.950 bytes



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Situación actual

Como referencia se anexa la base de datos principal

display structure

Structure for database: C:\VIÁTICOS\ANTERIOR\VIÁTICOS.DBF

Number of data records: 147

Date of last update : 11/06/00

Field	Field Name	Type	Width	Dec
1	NUMEMP	Numeric	6	
2	NUMANT	Numeric	6	
3	FANTICI	Date	8	
4	NUMDES	Numeric	3	
5	COMISION	Character	40	
6	DESTINO	Character	40	
7	MOTIVO	Character	40	
8	PLANSAL	Date	8	
9	PLANLLE	Date	8	
10	PLANSALT	Character	8	
11	PLANLLET	Character	8	
12	REALSAL	Date	8	
13	REALLLE	Date	8	
14	REALSALT	Character	8	
15	REALLLET	Character	8	
16	PDIAS	Numeric	8	2
17	RDIAS	Numeric	8	2
18	CUOTAD	Numeric	8	2
19	CUOTAUSD	Numeric	8	2
20	VIÁTICOS	Numeric	8	2
21	HOSPEDAJE	Numeric	8	2
22	TICA	Numeric	8	4
23	GASOLINA	Numeric	8	2
24	GVIAJE	Numeric	8	2
25	ANTICIPO	Numeric	8	2
26	GTRANS	Numeric	8	2
27	GTAXI	Numeric	8	2
28	ACOMPRO	Numeric	8	2
29	COMPRO	Numeric	8	2
30	SFAVOR	Numeric	8	2
31	SCARGO	Numeric	8	2
32	FCOMPRO	Date	8	



## MIGRACIÓN DE SOFTWARE

### Capitulo III "Metodología para la migración"

#### Situación actual

33	NUMRECIBO	Numeric	3	
34	STATUS	Character	1	
35	AVIACIA	Character	40	
36	AVIADEST	Character	40	
37	AVIANVS	Character	40	
38	AVIANVR	Character	40	
39	AVIABOL	Character	40	
40	AVIAFHS	Character	40	
41	AVIAFHR	Character	40	
42	AVIAOBS	Character	40	
43	NOMHON	Character	30	
44	PUESTHON	Character	20	

\*\* Total \*\*

718



### 3.3 ESTANDARES DE PROGRAMACIÓN

Cuando se desarrolla software propietario se debe estar seguro de que se entenderá con facilidad el código y lo que realiza cada una de las instrucciones, tanto en el momento de la realización como después de cierto tiempo, así cualquier persona ajena al momento de la realización, podrá realizar alguna modificación o mantenimiento.

Al integrar un grupo de trabajo y hacer desarrollos de Software en forma paralela, dando a cada uno de los integrantes un módulo que desarrollar, surge otro problema al momento de conjuntar éstos debido a que cada módulo fue realizado de acuerdo a los hábitos individuales de programación, quedando un programa difícil de entender y con problemas de eficiencia en la operación y en el mantenimiento.

En virtud de lo anterior surge la necesidad de desarrollar software de acuerdo a estándares; ya que con ello se asegura que el sistema desarrollado sea fácil de entender, sencillo para proporcionar mantenimiento y lo más importante obtener la máxima eficiencia en la utilización de los recursos disponibles.

#### 3.3.1 Lenguaje de programación utilizado

En la actualidad existe gran diversidad de lenguajes de programación, los cuales tienen características propias, como pueden ser el nombramiento de sus objetos, funciones, instrucciones, reportes, comentarios, ect., por lo antes mencionado sería muy complejo tratar de estandarizarlos en conjunto, en este caso específico mostraremos el lenguaje anterior comparando con el lenguaje al que se migró la aplicación, esto no implica que los demás lenguajes existentes no puedan estandarizarse en el lugar donde se utilicen.

En nuestro ejemplo se hará referencia como lenguaje origen a Clipper y como lenguaje destino "Visual Basic" de Microsoft®, el cual es un lenguaje visual que combina características de Programación estructurada con programación Orientada a Objetos.



### 3.3.2. Edición de código

**FORMA DE ESCRIBIR EL CÓDIGO.** Cuando se escribe código en Visual Basic, es importante que éste se escriba con letra minúscula, ya que Visual Basic contiene implícitamente un reconocimiento de funciones y de sintaxis del código y lo convierte en mayúsculas y minúsculas al momento de cambiar de renglón, esto ayuda de sobremanera al programador para ver si no cometió un error de sintaxis en la escritura o si la función que está escribiendo es reconocida por el programa o si las variables fueron declaradas anteriormente. (Esta característica no aplica en el lenguaje clipper )

**SANGRÍAS.** Es necesario el uso de sangrías para diferenciar las partes de código, tales como loops y saltos condicionales. Para sangrar una selección de código, se debe usar la tecla TAB

Ejemplo:

CODIGO CLIPPER	CODIGO VISUAL BASIC
<pre>if LastKey() = K_ESC   SugNumAnt = 0   go top   Keyboard chr(K_ENTER) else   l_Valido = .t. endif</pre>	<pre>If imgVerde.Visible = True Then   imgVerde.Visible = False   imgAmari.Visible = True Else   imgRojo.Visible = False   imgVerde.Visible = True End If.</pre>

**GUIÓN.** El guión ( \_ ) se debe usar para hacer de una línea de código, múltiples líneas cuando ésta es muy grande. Esto hace al código más fácil de leer ya que el código se encuentra contenido totalmente dentro de la ventana.

El guión se pone dejando un espacio al final de la línea y puede ser usado dentro de una cadena de texto.

Ejemplo:

CODIGO CLIPPER	CODIGO VISUAL BASIC
<pre>//-----Definición de Teclas----- set key K_F1 to Reg_existe</pre>	<pre>Public Psrep_area As String</pre>



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

set key K_F3 to Elimina_Registro set key K_F4 to Next_Pant	Public Psrep_clie As String
set key K_F5 to Modo_Edicion	Public Psrep_proy As String

**COMENTARIOS.** Agregar documentación y comentarios al código lo hace más fácil y además ayuda a determinar lo que el código realiza. Esto ayuda a entender el código si se necesita regresar a él después de un tiempo, así los comentarios deberán ser puestos antes de su propia línea o al final de una línea de código.

Es imprescindible colocar comentarios para los siguientes códigos:

1. Cualquier tipo de cálculo
2. Búsquedas
3. Actualizaciones
4. Validaciones

Ejemplo:

CODIGO CLIPPER	CODIGO VISUAL BASIC
*este asterisco es para hacer *comentarios en clipper	'este apóstrofe es para hacer 'comentarios en Visual Basic

Se deben poner comentarios después de la definición del procedimiento o función para indicar que es lo que hace, y también agregar comentarios después de la línea de código que realiza una acción. Esto mantiene líneas individuales más cortas y dentro de la ventana de código.

### 3.3.3 Nombre de los objetos

**CONVENCIONES PARA NOMBRAR OBJETOS.** Así como cada objeto es creado, Visual Basic le asigna un nombre por default, tal como *Form1*, *Command2*, y *Text3*, etc., deberá cambiar la propiedad **Name** de cada control a la hora de crearlo en la caja de propiedades. El nombre de un objeto deberá determinarse por las tres primeras letras del prefijo del tipo de objeto con minúsculas, seguida por una descripción de la característica de éste comenzando con una letra mayúscula. La siguiente tabla muestra el prefijo a usar con los diferentes objetos creados así como un ejemplo del nombre que debe estar asociado.





# MIGRACION DE SOFTWARE

## Capitulo III "Metodología para la migración"

### Estándares de Programación

OBJETO	PREFIJO	EJEMPLO
Form *	frm	FrmAbrirArch
Check box	chk	ChkSoloLectura
Combo box	cbo	CboInglés
Command button	cmd	CmdCancel
CommandDialog	cdg	
CrystalReport	cr	
Data	dat	DatBiblio
DBCombo	dbc	
DBList	dbl	
DBGrid	dbg	DbgPrices
Directory list box	dir	DirFuente
Drive list box	drv	DrvTar
File list box	fil	FilFuente
Frame	fra	FraLenguaje
Grid	grd	GrdPrec
Horizontal scroll bar	hsb	HsbVolumen
Image	img	ImgIcono
Label	lbl	LblMensaje
Line	lin	LinVertical
List box	lst	LstPolicyCodes
MaskedTextBox	meb	
Menú	mnu	MnuAbrirArch
MSRDC	rdc	
OLE	ole	oleObjeto1
Option button	opt	OptFrancés
Picture box	pic	PicEspacDisco
RichTextBox	rtb	
Shape	shp	ShpCircle
SSCheck	ssch	
SSCommand	scmcd	
SSFrame	ssf	
SSOption	sso	
SSPanel	ssp	
SSTab	sst	
SSRibbon	srib	
StatusBar	stb	
TabStrip	tab	
Text box	txt	TxtGetText
Timer	tmr	TmrAlarm
ToolBar	tlb	
TreeView	tvw	



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

Vertical scroll bar	vsb	VsbRate
---------------------	-----	---------

El cambio de conceptos por parte del usuario, sólo implicará cambiar el nombre de la forma y comentar en el código el cambio de conceptos. (No aplica a clipper versión no orientada a objetos).

**CÓDIGO DE LOS OBJETOS.** Es importante cambiar el nombre de cualquier control que se haya creado antes de escribir cualquier código, porque esta asociado con el nombre del objeto. si el nombre del objeto se cambia después de que el código fue escrito, éste perdería su asociación con el control. En el caso de que un objeto se borre, su código asociado pasará al objeto General en el evento Declarations por lo que debe de ser borrado también este código para evitar el dejar código no usable dentro de este evento. (No aplica a clipper versión no orientada a objetos).

El siguiente ejemplo muestra el código para una "Command Button" llamado cmdOK.

```
Private Sub cmdOK_Click()  
    Static SiLogonAttempts As Integer  
    Const MAX_LOGON_ATTEMPTS As Integer = 3  
    SiLogonAttempts = SiLogonAttempts + 1  
End Sub
```

#### 3.3.4 Nombres de Variables

**DECLARACION DE OPTION EXPLICIT.** Dentro del Objeto General en el evento Declarations, debe de ir siempre el comando **OPTION EXPLICIT**, el cual es un comando que obliga a Visual Basic que cualquier variable que use sea declarada antes, esto evitará el utilizar variables no definidas ya que las tomaría como tipo Variant y el uso de memoria de estas es muy alto, también evita errores de transcripción o de cambio del nombre de una variable, por lo cual es necesario que en la declaración de configuración de Visual Basic marcar la opción Requiere Variable Declaration (requiere declaraciones de variables) dentro del menú Tools, Options Environment, con lo cual al iniciar un proyecto, y cargar cualquier nueva forma, automáticamente aparecerá la sentencia **OPTION EXPLICIT**.

Visual Basic tiene la ventaja de que si se declara explícitamente una variable con letras mayúsculas y minúsculas, automáticamente cuando se escribe ésta, es capitalizada de acuerdo a la forma como se escribió, esto sirve para confirmar al momento de escribir el código que la variable fue declarada y que no se esta



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

cometiendo un error de escritura. (No aplica a clipper versión no orientada a objetos).

Ejemplo:

En el objeto General en la opción de Declaration, se declara la variable PiTotal que es una variable pública de tipo integer. Al momento de escribir el código, se teclea pitotal, al dar enter o cambiar de renglón Visual Basic cambia automáticamente a PiTotal, con lo que puede el programador estar seguro de que la variable fue declarada anteriormente y está escrita en forma correcta ya que se capitalizo.

**REGLAS PARA NOMBRAR VARIABLES.** El alcance de una variable dentro de un lenguaje, se define por la forma y el lugar en donde se declara, el cual puede ser público, local, estático, etc.; asimismo existen varios tipos de variables como son de string, numéricas, variantes, etc., cada una de ellas tiene un alcance de vida y un consumo de memoria diferente. Para el buen entendimiento del programa se distinguirá el alcance y el tipo de variable con el simple hecho de leer su nombre, por lo que deberá seguir la siguiente regla:

- a) Debe comenzar con una letra mayúscula, la cual será una abreviatura que represente el alcance

ALCANCE	PREFIJO
Pública	P
Module/Form	M
Local	L
Estática	S

- b) El siguiente carácter debe ser una abreviación en minúscula, representando el tipo de dato al que pertenece la variable:

TIPO DE DATO	PREFIJO
Byte	B



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

Boolean	F
Integer	i
Long	l
Single	s
Double	dbl
Currency	C
Date	Dt
Object	use el prefijo del objeto (frm, txt, etc.)
String	Str
Variant	V

c) No puede contener símbolos especiales. tales como ., %, &!, #, o \$.

d) No debe ser mayor de 255 caracteres en Visual Basic y en el caso de clipper no debe ser mayor de 8 caracteres.

e) Se debe asignar siempre que sea posible un tipo determinado de variable, ya que si no se declara la toma como variant y la variable variant consume más recursos de memoria que cualquier otro tipo. esta característica no aplica en lenguaje clipper.

Ejemplos:

#### EJEMPLO CLIPPER

```
store "Nombre empleado" to nom_empl  
store 0 to V_SUB  
store mes to V_mes
```

#### EJEMPLO VISUAL BASIC

```
Public PfFinal As Boolean (Variable Publica tipo boolean)  
Dim STotal As Single (Variable estática tipo singel)  
Dim MstrNombre As String (Variable de Módulo tipo string)
```



### 3.3.5 Aspectos Generales.

**NOMBRES DE LAS FUNCIONES.** Es de vital importancia el nombrar las funciones, de acuerdo a las actividades que realizan y sus características. Debemos tener presente que se cuenta con dos categorías de función:

1. Las predefinidas.- Son las secciones de código encargadas de realizar las operaciones necesarias para la correcta ejecución de un evento.
2. Las definidas por el usuario.- Este tipo de funciones son propuestas por el programador de acuerdo con sus propias necesidades, es decir, el programador es libre de usar una función que realice una actividad en especial, por ejemplo la validación de una clave, una fecha, etc., ya que con el uso de esta función podrá ahorrar líneas de código en otro momento, siendo innecesario volver a escribir dicho código, para realizar la misma acción bastará con hacer una llamada a la función creada anteriormente; esto se logra mencionando el nombre de dicha función. Para asignar un nombre a estas funciones se debe usar la siguiente sintaxis:

F+TD+Nombre

donde:

**F** se usará para distinguir una función de un procedimiento

**TD** debe ser una abreviación en minúscula, representando el tipo de dato que retornará la función.

**nombre** será el nombre que se desee dar a la función.

El nombre de la función debe ser descriptivo, es decir, debe proporcionar información importante como es: para qué sirve o para qué se utiliza dentro del sistema, así cualquier persona que desee utilizar la función y no conozca su funcionamiento entenderá más fácilmente la razón de su existencia, ejemplo:

<b>EJEMPLO CLIPPER</b>
function Reg_existe (Función que verifica si existe un registro)
<b>EJEMPLO VISUAL BASIC</b>
<b>Fvalid_clave (Función que valida la clave de acceso de un usuario)</b>



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

No olvide:

- que una función a diferencia de un procedimiento retorna un valor a la parte del código donde fue llamada, y
- comentar acerca de lo que hace la función, los valores que regresa y parámetros que recibe.

**NOMBRES DE LOS PROCEDIMIENTOS.** Al igual que el nombre de las funciones, es importante definir el nombre de los procedimientos de acuerdo a las características de cada uno de ellos. Para dar nombre a un procedimiento se utilizará la siguiente sintaxis:

Pnombre

donde

**P** Se utiliza para diferenciar un procedimiento de una función

**nombre** será el nombre que se desee dar a la función.

El nombre del procedimiento debe ser descriptivo, es decir, debe proporcionar información importante como: para que sirve o para que se utiliza dentro del sistema para cualquier persona que desee utilizar un procedimiento y no conozca su funcionamiento, esto se logra usando nombres que sean lo más adecuados posibles para describir la función de un procedimiento. Por ejemplo:

<b>EJEMPLO CLIPPER</b>
set procedure to locks
* donde locks es un programa de procedimientos.
<b>EJEMPLO VISUAL BASIC</b>
Pcalculo_totales (procedimiento encargado de calcular valores totales)

Es importante recordar que un procedimiento **NO** regresa ningún valor. Se deben colocar comentarios indicando que es lo que hace el procedimiento, que parámetros recibe y que resultado genera.

**NOMBRES DE LOS ARCHIVOS.** Para obtener un buen control sobre los archivos que integren el sistema que se este desarrollando tenemos que considerar una notación especial para dar un nombre a cada uno de ellos, esto con la finalidad de



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

contar con nombres sugestivos: es decir, nombres que nos proporcionen información clave que permita identificar el tipo de archivo (formas, clases y módulos BAS) de que se trata. Para lograr este objetivo, se deberá cumplir con una nomenclatura especial para nombrar a cada archivo, dicha nomenclatura deberá contar con la siguiente estructura.

<b>TIPO DE ARCHIVO CLIPPER</b>
Viáticos.prg
<b>TIPO DE ARCHIVO VISUAL BASIC</b>
Viáticos.bas
Viáticos.frm
Viáticos.cls

Se tiene que escribir el tipo de archivo con minúsculas e iniciar el nombre con mayúscula y el resto con minúsculas, entre el tipo y nombre de archivo no debe ir ningún espacio en blanco, además es importante recordar que para nombrar un archivo no se deben usar caracteres especiales como: #,\$,%,&,!,:,;, etc.

NOTA: Recordar que Visual Basic no tiene un límite de caracteres para dar un nombre de archivo. Para clipper se limita a una longitud de 8 caracteres.

**CONSIDERACIONES ESPECIALES.** Con la inminente finalización del presente siglo, es necesario tomar algunas providencias en cuanto a los formatos de fecha, para evitar los errores de cálculos que puedan ser originados por esta razón. Empleando el siguiente formato se tendrá especial cuidado en la definición de las fechas:

**dd/mm/aaaa**

donde:

**dd** es el día

**mm** es el mes

**aaaa** es el año utilizando los cuatro dígitos.

**MENSAJES AL USUARIO.** Muchas son las ocasiones en que el sistema necesita dar alguna información al usuario acerca de los datos que han ingresado o de las operaciones que están ejecutándose, así como también requiera la confirmación de cierta acción, en estos casos se usa la función MsgBox de VisualBasic para establecer el diálogo con el usuario.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

#### *MsgBox mensaje, botones, encabezado*

Se han identificado cuatro tipos de mensajes, los cuales se han estandarizado y deberán ser empleados en todas las aplicaciones que se desarrollen. A continuación se muestra una tabla en donde se especifica:

1. Tipo de error,
2. Constantes que identifican: el estilo del ícono a desplegar, el número y tipo de botones
3. Encabezado,
4. Casos en que se deberán emplear
5. Mensajes

Tipo	Botones	Ícono	Encabezado	Casos de uso	Mensajes
Error	VbOK VbCritical		Error	Cuando la aplicación genere un error, sin importar la causa	Generado por la aplicación
Precaución	VbOK VbExclamation		- Datos incompleto - Datos inválidos - Login - Precaución	Cuando la aplicación requiera de alguna acción extra por el usuario	Libre
Informativos	VbOK VbInformation		Información	Cuando la aplicación le haga saber al usuario el status de la operación realizada	Libre
Confirmación	VbYesNo VbQuestion		Confirmación	Cuando la aplicación requiera la autorización del usuario para seguir ejecutando cierta acción	Libre

Nota: El mensaje tendrá como máximo 3 líneas y 40 caracteres en cada una de ellas. Para separar una línea de otra se empleará el vbCrLf

Mensajes al usuario en clipper. Todos los mensajes de información, advertencia y error, se despliegan al calce de la pantalla en uso. Ejemplo:





**MIGRACIÓN DE SOFTWARE**  
Capítulo III "Metodología para la migración"  
Estándares de Programación



**TRATAMIENTO DE ERRORES.** Los Sistemas realizados en Visual Basic, deberán de contener forzosamente un tratamiento especial para los errores, ya que si no se cuenta con una forma de atraparlos, cuando un error ocurra en la ejecución, se desplegará un mensaje y la aplicación finalizará, aunado a esto es necesario guardar los errores generados en un archivo para poder depurar alguna aplicación y encontrar el tipo de error generado y el módulo, sin depender directamente del usuario del sistema.

La forma de capturar errores es por medio del comando On Error, el cual es la sentencia que establece un manejador de errores, si un error ocurre al tiempo de ejecución, el programa transferirá el control hasta la etiqueta indicada por la sentencia On Error. (No aplica a clipper versión no orientada a objetos).

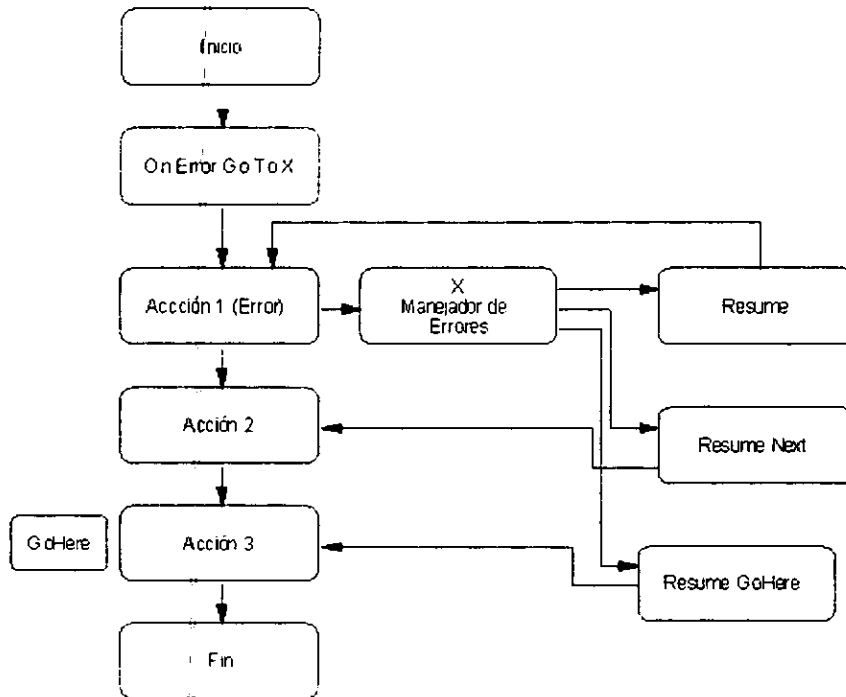
El siguiente diagrama muestra la forma en que el manejador de errores trabaja.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación



**Nombres de los Reportes.** Para el correcto control de las formas y de los reportes, se utilizará la siguiente convención:

1. El nombre de la forma debe iniciar con frm y tener a continuación un campo descriptivo del reporte de que se trata.
2. El nombre del reporte debe iniciar con el prefijo REP y a continuación el número de reporte, marcado con tres dígitos, ejemplo REP001.\*\*\*, este número de reporte debe ser incluido necesariamente en el encabezado del reporte.

**Formato del Reporte.** Los reportes realizados deberán forzosamente incluir una serie de datos en el encabezado, cuerpo del reporte y pie de página, para poder asegurar que cumpla con los aspectos de Control Interno y puedan ser fácilmente auditados. Estos campos de ninguna manera deberán ser excluidos y deben tener como mínimo:



Encabezado:

- o Logotipo (si aplica)
- o Departamento que produjo el reporte
- o Nombre del Reporte
- o Nombre de la Empresa
- o Nombre del Sistema que lo produjo
- o Fecha de Emisión del Reporte
- o Título descriptivo del contenido del reporte
- o Hora de producción
- o Rango de datos incluidos y período de proceso cubierto
- o Línea de división (si se utiliza encabezado de campos)
- o Encabezado de campos (si aplica)
- o Línea de división

Cuerpo del Reporte

- o Número de Registros impresos (si aplica)
- o Marca de fin de trabajo (Especificando si finalizo o fue cancelado por el usuario)

Pie de página

- o Línea de división
  - o Marca de clasificación de seguridad/confidencialidad y forma de destrucción
- Nivel de Confidencialidad
- |     |       |  |
|-----|-------|--|
| *   | Bajo  | Puede reciclarse el papel              |
| **  | Medio | No puede reciclarse el papel           |
| *** | Alto  | Debe ser destruido por equipo especial |
- o Número de página consecutivo

**FORMA DE DOCUMENTACION DE SISTEMAS.** Para la correcta documentación y facilidad de entendimiento de las formas realizadas en Visual Basic, éstas deberán llevar forzosamente un encabezado con información en el Objeto General en el procedimiento Declarations y antes de la sentencia Option Explicit conteniendo los siguientes datos

1. Nombre de la Forma y breve descripción de ésta
2. Autor de la forma
3. Tablas utilizadas, marcando entre paréntesis (R) si es de lectura (W) si es de



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Estándares de Programación

escritura o una (RW) si es de ambas, asimismo se debe de marcar el origen de los datos (Informix o Access).

4. Formas que llama (Si llama formas)
5. Reportes que llama (Si llama Reportes)
6. Llamada por (indica la forma que mando llamar a esta)
7. Lista de las 10 últimas modificaciones, incluyendo el nombre, fecha y un breve comentario

Es importante escribir una línea de separación en el 1er. Renglón, así como entre el campo 2 y 3 y al final de los comentarios.

A continuación se muestra un ejemplo de un encabezado de programa

<b>CLIPPER</b>	
//	----- // destinos // desarrollado por el Ing. *** // Programa que actualiza la tabla de destinos //-----
<b>VISUAL BASIC</b>	
'FRMMERDET.FRM	3.1. ASPECTOS GENERALES DE MERCADO (detalle)
'Autor:	
'Tablas:	Mercado (RW) Access MerExporta (RW) Access MerNacional (RW) Access CatProductos (R) Access CatEstados (R) Access
'Llamada por:	frmMerGral.frm
'Modificada por:	(26-03-1998)

**Módulo de Control de Accesos.** Los Sistemas desarrollados en cualquier lenguaje y Empresa, deben de cumplir necesariamente con los objetivos de seguridad planteados para ese organismo y son los siguientes:

- Restricción del acceso y uso de datos y programas sólo a personas autorizadas.
- Protección de datos y registros vitales de revelación no autorizada y prevención de lectura o copiado no procedente.



## **MIGRACION DE SOFTWARE**

Capitulo III "Metodologia para la migración"

Estándares de Programación

- Protección de datos y programas contra la modificación o actualización no autorizada.
- Dotación de pistas de auditoria a los sistemas desarrollados dentro de la Empresa, que permitan monitorear el buen aprovechamiento de éstos.

Para el cumplimiento de estos objetivos, se desarrolló un módulo de control de accesos estándar que forzosamente debe de ser utilizado en todos los sistemas desarrollados por el Área de Sistemas. El módulo de control de acceso tiene como finalidad realizar las siguientes funciones:

1. No permitir la entrada a usuarios que no cuenten o no conozcan la clave de acceso.
2. Poner una marca (incidencia) de uso del Sistema.
3. Poner una marca de inhabilitación cuando se intentó ingresar tres veces consecutivas sin conocer el password de acceso.
4. Poder dar de alta nuevos usuarios.
5. Poder dar de baja usuarios.
6. Poder retirar incidencias a usuarios.
7. Generar una bitácora de accesos al Sistema y tener forma de poder checar ésta.
8. Generar una bitácora de incidencias y tener forma de poder checar ésta.
9. Poder checar archivo de errores locales.

**Claves de acceso.** El módulo de control de acceso, se basa en clave

### **Tipos de Usuario.**

Para efectos del módulo de control de accesos, tomaremos dos tipos de usuario:

**Normal.-** Usuario Normal del Sistema, que no tiene acceso al módulo de Control de Accesos

**Responsable.-** Usuario responsable del Sistema que se encarga de administrar el módulo de control de accesos, proporcionando claves, derechos de acceso, retirando incidencias, etc.

Las claves de acceso, las cuales son un conjunto de números, caracteres o combinación de ambos, que se asignan a un usuario o recurso de un sistema de computo.

Es importante cumplir con las siguientes reglas para la seguridad y control de las claves de acceso:



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo III "Metodología para la migración"**

#### **Estándares de Programación**

- Encriptar la clave de acceso durante su almacenamiento y transmisión a través de la red de comunicaciones.
- Manejar un rango de longitud de la clave de acceso entre 1 y 8 caracteres.
- Proteger en la pantalla la digitación de la clave de acceso.
- Prohibir el uso del sistema a usuarios sin clave de acceso.



## MIGRACIÓN DE SOFTWARE

Capítulo III "Metodología para la migración"

Metodología para el desarrollo (migración) de sistemas

### 3.4 METODOLOGIA PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN.

Por metodología se entiende "un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo de un sistema de información.

#### CICLO DE VIDA

El ciclo de vida es el conjunto de fases por las que debe pasar un proyecto de desarrollo de un sistema de información, desde su concepción inicial hasta que el sistema deja de utilizarse o se transforma en otro.

El ciclo de vida más completo es el denominado clásico o en cascada (waterfall model). Este ciclo establece una serie de fases, al finalizar las cuales se obtiene una serie de productos (documentos, diagramas, programas) que permite evaluar lo realizado hasta ese momento y continuar con la fase siguiente o modificar algunos aspectos de las fases anteriores (proceso de retroalimentación).

A continuación se indica someramente el objetivo de cada una de ellas:

**Planificación Estratégica:** No se considera estrictamente como una fase del ciclo de vida, ya que su existencia es opcional (no es necesaria si se desarrolla un sistema aislado) y no afecta a un solo sistema. Su objetivo es adecuar los objetivos estratégicos de la Empresa (de usuario) y la información necesaria para soportar dichos grandes objetivos. Para tal labor se hace necesaria una metodología de planificación de sistemas que abarque a toda la Empresa y exige considerar una serie de conceptos que desbordan el marco específico de una Metodología de Desarrollo de Sistemas. El producto final que se habrá generado en este punto será la definición de los sistemas de información que se deben de desarrollar para satisfacer los objetivos estratégicos de la Empresa. Por tanto, la parte de este documento que haga referencia al sistema concreto que se vaya a desarrollar, puede ser el punto de partida para la siguiente fase: Análisis del Sistema.

#### a) Fase de Análisis:

El objetivo de esta fase es el estudio de las necesidades de información que debe satisfacer el sistema a desarrollar, elaborando una serie de especificaciones formales que describan la funcionalidad del mismo y que permitan abordar con garantías la siguiente fase (Diseño). Esta fase de Análisis se puede estructurar en



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Metodología para el desarrollo (migración) de sistemas

dos subfases claramente diferenciadas, en las que se lleven a cabo tales actividades:

**Análisis de Requisitos del Sistema:** Se trata de establecer el alcance, los objetivos y requisitos del sistema, examinando las posibles alternativas que podrían solucionar las necesidades del usuario (cliente) y recomendar una de ellas.

**Especificación Funcional del Sistema (conocida tradicionalmente como "Análisis Funcional"):** Una vez aceptado formalmente el documento anterior por las partes (organización de desarrollo-organización de usuario), se elabora un conjunto de especificaciones formales que describen la funcionalidad del sistema, estableciendo los subsistema en que se descompondrá, definiendo los datos que utilizará y las interfases del usuario. También se planificarán las pruebas que deberá superar el sistema, se estimará la relación costo/beneficio para comprobar si interesa su construcción y se establecerán los plazos de entrega del sistema.

#### **b) Fase de Diseño (conocida tradicionalmente como "Análisis Orgánico"):**

El objetivo de esta fase es obtener un conjunto de especificaciones que contemplarán los aspectos físicos del sistema, considerando las características tecnológicas del entorno específico en el que se implantará, que constituirán el punto de partida para la construcción del sistema.

#### **c) Fase de Construcción:**

El propósito de esta fase es, a partir de las especificaciones de diseño, la obtención del sistema completamente construido y probado, listo para ser implantado en la Empresa del usuario. También durante esta fase se desarrollará el conjunto de procedimiento y se llevará a cabo la formación necesaria que permitirá, tanto al personal del área de usuario final como al personal de área de explotación o proceso de datos (si existe), la utilización óptima del sistema.

#### **d) Fase de Implantación:**

El objetivo de esta última fase es la puesta en servicio del sistema construido y conseguir la aceptación final por parte de los usuarios del mismo, para lo cual se tratará de hacer ver a éstos, mediante demostraciones formales (pruebas de aceptación), que el sistema cumple todos los objetivos y requisitos para los que fue desarrollado. En esta fase e incluye la ejecución del mantenimiento del sistema, con lo que su duración se prolongará hasta que el sistema deje de utilizarse o sea sustituido por otro.





## MIGRACIÓN DE SOFTWARE

Capítulo III "Metodología para la migración"

Metodología para el desarrollo (migración) de sistemas

ANÁLISIS DE REQUISITOS DEL SISTEMA		DISEÑO	CONSTRUCCION	IMPLANTACION
REQUISITOS DEL SISTEMA	ESPECIFICACIÓN FUNCIONAL			
<ul style="list-style-type: none"><li>• Establecimiento del ámbito y alcance del proyecto.</li><li>• Definición de objetivos y requisitos</li><li>• Diseño del modelo lógico actual</li><li>• Estudio de alternativas de construcción</li></ul>	<ul style="list-style-type: none"><li>• Realización del Modelo de Funciones del Sistema</li><li>• Realización del Modelo de datos del Sistema</li><li>• Diseño de Interfases de usuarios y prototipos</li><li>• Establecimiento de especificaciones no Funcionales</li><li>• Establecimiento de Especificaciones de Entrega</li></ul>	<ul style="list-style-type: none"><li>• Diseño de la arquitectura de Funciones del Sistema</li><li>• Diseño de la estructura de Datos del Sistema</li><li>• Definición del Entorno Tecnológico del Sistema</li><li>• Diseño del Plan de Pruebas del Sistema</li><li>• Diseño del Plan de Construcción e Implantación</li></ul>	<ul style="list-style-type: none"><li>• Construcción de las Bases de Datos o Ficheros</li><li>• Construcción de las Unidades de Tratamiento y Pruebas Unitarias</li><li>• Realización de Pruebas de Integración y Globales</li><li>• Elaboración de los Procedimientos de Explotación y Formación</li><li>• Elaboración de los Procedimientos de Usuario y Formación</li></ul>	<ul style="list-style-type: none"><li>• Revisión del Plan de Implantación del Sistema</li><li>• Instalación del Sistema</li><li>• Realización de las Pruebas de Aceptación</li><li>• Transición al nuevo Sistema</li></ul>



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

### 3.5 EJEMPLO PRACTICO

A continuación se mostrará cómo se aplicaron las cuatro fases del ciclo de vida clásico, en la migración del sistema de viáticos.

#### 3.5.1 Fase de Análisis:

##### *1 Análisis de Requisitos del Sistema*

El estudio desarrollado en esta fase permitirá apreciar los puntos fuertes y débiles del sistema actual, identificar uno o varios escenarios de modificación, definir los objetivos y elegir una determinada trayectoria para obtenerlos.

El desarrollo de la etapa de Análisis de Requisitos del Sistema se estructuró en las siguientes etapas o subactividades:

- 3.5.1.1.1 Establecimiento del ámbito y alcance del proyecto.
- 3.5.1.1.2 Definición de objetivos y requisitos
- 3.5.1.1.3 Diseño del modelo lógico actual
- 3.5.1.1.4 Estudio de alternativas de construcción

---

##### **3.5.1.1.1 Establecimiento del ámbito y alcance del proyecto**

Inicialmente el proyecto del "Sistema de Viáticos" se diseñara para operarse en oficinas centrales del Organismo y será de aplicación exclusiva para el personal de la institución.

##### Técnicas Utilizadas

En esta primera actividad, la obtención de información, se realizó a través de entrevistas y reuniones con los responsables usuarios de las áreas afectadas.

---

##### **3.5.1.1.2 Definición de Requisitos**

Se necesita tener la información histórica en un Servidor Central y minimizar los tiempos de respuesta en la autorización, por que se propone se realicen las



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

autorizaciones con firmas electrónicas, se realicen los depósitos de viáticos en las cuentas personales de los usuarios y se realicen las afectaciones contables en línea.

Para la realización del sistema se respetaron las normas y políticas para el otorgamiento de viáticos del Manual de procedimientos de viáticos y gastos de viaje.

En el siguiente cuadro se muestran las características mínimas que tienen cada uno de los sistemas desarrollados en los dos lenguajes referidos con anterioridad:

CLIPPER	VENTAJAS
<ul style="list-style-type: none"><li>• Conexión al servidor Central (red)</li><li>• Espacio en disco de 5 Mb Pathway (conexión)</li><li>• Espacio en disco 500 Kb para Viáticos (ejecutable)</li><li>• Espacio en disco para la base de datos 300 Kb</li><li>• MS-DOS Versión 5.0 (Ambiente no gráfico)</li><li>• Estar dado de alta como usuario en la red</li></ul>	X X X
VISUAL BASIC	
<ul style="list-style-type: none"><li>• Conexión al servidor Central (red)</li><li>• Espacio en disco de 25 Mb para conexión ODBC</li><li>• Espacio en disco de 8 Mb para Viáticos (ejecutable)</li><li>• Espacio en disco para la base de datos 500 Mb</li><li>• Windows 95 (Ambiente gráfico)</li><li>• Estar dado de alta como usuario en la red</li></ul>	X

Como se puede apreciar en la tabla anterior los recursos de cada una de las plataformas son extremadamente diferentes, debido a que la programación orientada a objetos (Visual Basic) consume muchos de ellos.

#### Técnicas Utilizadas:

Entrevistas con los responsables de las áreas de usuario afectadas para aspectos generales y con los usuarios para obtener información detallada.



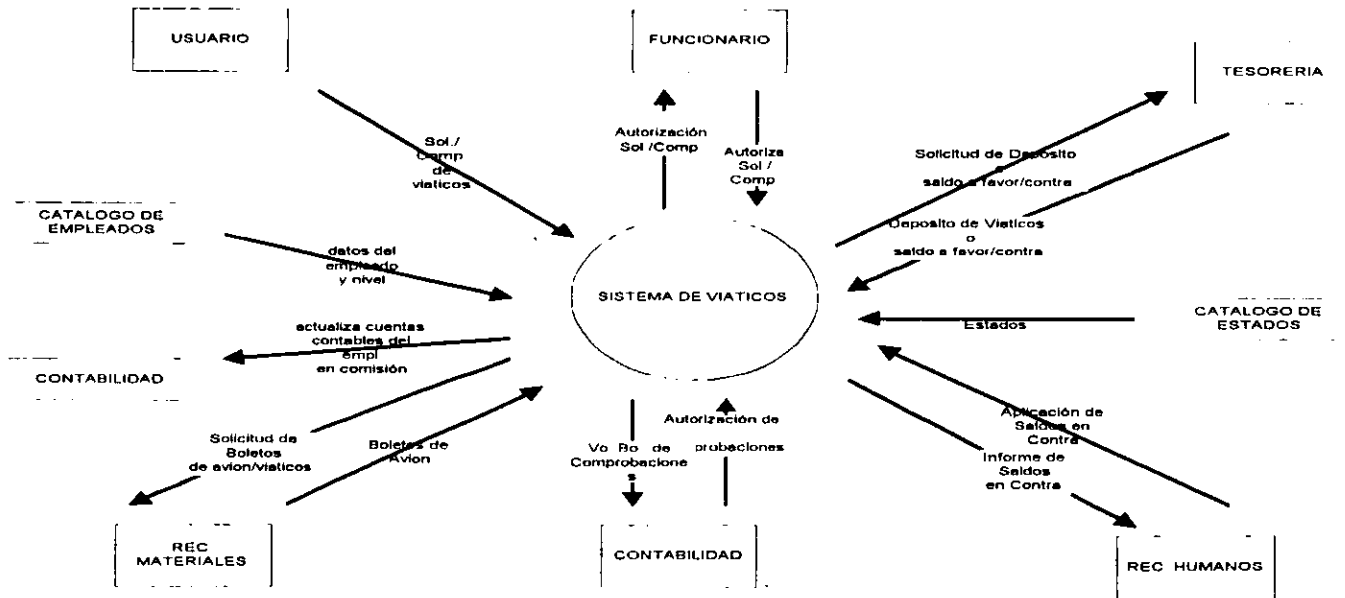
# MIGRACION DE SOFTWARE

## Capitulo III "Metodología para la migración"

### Ejemplo práctico

### 1.3 Diseño del modelo Lógico Actual

DIAGRAMA DE CONTEXTO DEL SISTEMA DE VIATICOS



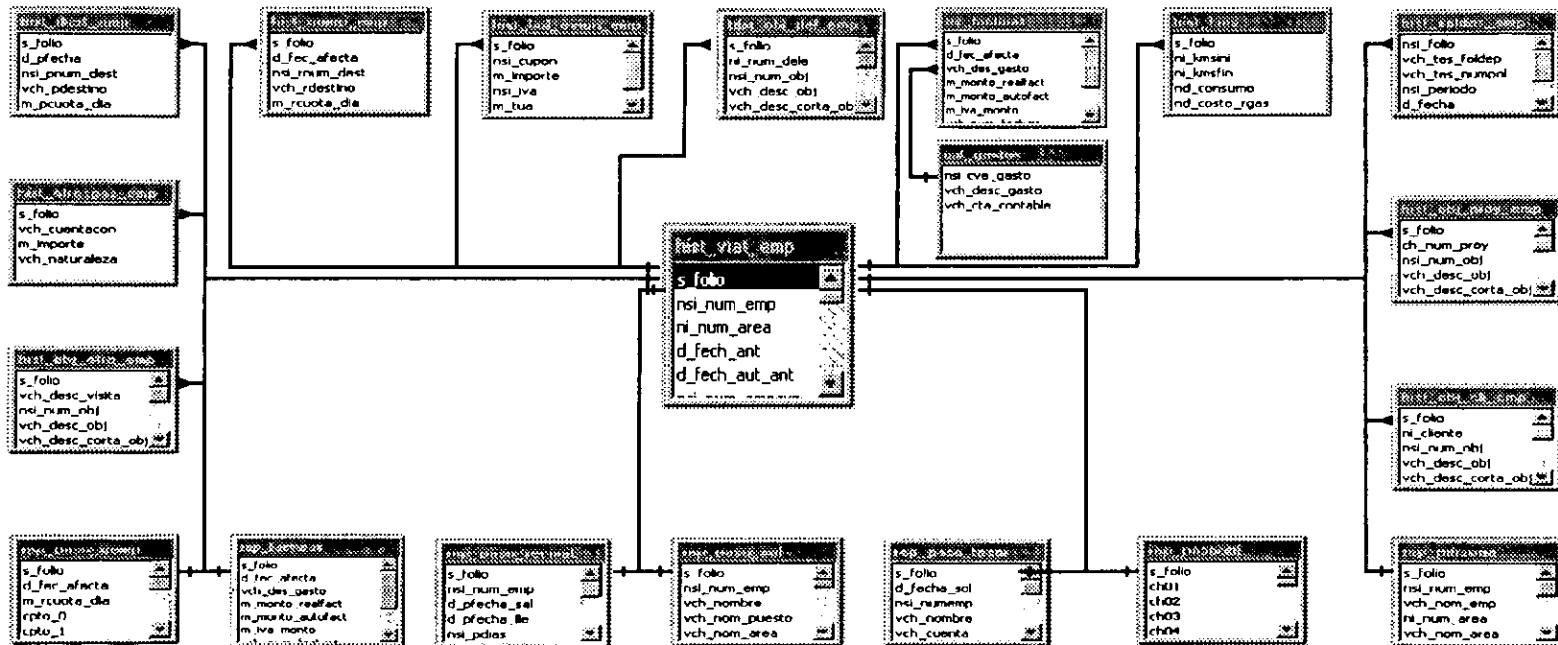


# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

Relaciones existentes en el Sistema de Viaticos  
En la Base de Datos de Viaticos  
Lunes, 15 de Mayo del 2000





#### **Técnicas utilizadas**

Para la construcción del modelo de funciones se utilizará la técnica estructurada denominada Diagrama de Flujo de Datos (DFD), que se completará con un Diagrama de Descomposición Funcional (DDF), mientras que para el modelo de datos se utilizará la técnica de Diagrama de Entidad Relación (DER).

---

#### **3.5.1.1.4 Estudio de Alternativas de Construcción**

Captando que el área de Recursos Materiales es el área que revisa y recopila las firmas de autorización para el otorgamiento de Viáticos, es conveniente sistematizar este proceso y responsabilizar a cada área de realizar su trabajo.

Para agilizar la recopilación de firmas y participación del área responsable se realizará un módulo al área participante y así tenga un sistema de monitoreo en el que pueda autorizar vía firma electrónica lo que le corresponda.

Cada una de las áreas tendrá un módulo en el cual tendrá que realizar su solicitud o comprobación de viáticos, autorizar solicitudes y/o comprobaciones vía firma electrónica, proporcionar Viáticos o saldo a favor, proporcionar boleto de avión, realizar descuentos (saldos en contra) y realizar afectaciones contables.

El almacenamiento de información podrá realizarse en la base de datos de Informix en el servidor de "Unix" y otra alternativa podría ser la realización de la base de datos en ambiente de Access en el servidor de "Windows NT". Finalmente se optó por manejar bases de datos realizadas en Access, debido que este manejador de base de datos viene integrado con el software de Office 97, mientras que el manejador de base de datos de Informix tiene un costo superior a los recursos económicos de la Empresa.

#### **Técnicas utilizadas**

Para la comparación de los costos de desarrollo del Sistema y los beneficios esperados de su implantación, se aplicó la técnica de análisis coste-beneficio.

Para seleccionar una alternativa serán necesarias entrevistas y reuniones con el Comité de Informática.

**Participación de Usuarios**

La siguiente matriz muestra la participación de los usuarios en las diferentes actividades de esta fase:

	11	12	13	14
Comité de Informática	X			X
Jefe del Proyecto	X	X	X	X
Analista	O	X	X	X
Programador				
Gerente del Área involucrada	X	X	X	X
Usuario Final		X	X	X

---



### ***II Especificación Funcional del Sistema***

En esta etapa los usuarios finales del Sistema tendrán una participación activa para garantizar que el sistema responde a sus necesidades en todos los aspectos.

El desarrollo de la etapa de Especificación Funcional del sistema en las siguientes subetapas o actividades:

- 3.5.1.2.1 Realización del Modelo de Funciones del Sistema
- 3.5.1.2.2 Realización del Modelo de datos del Sistema
- 3.5.1.2.3 Diseño de Interfases de usuarios y prototipos
- 3.5.1.2.4 Establecimiento de especificaciones no Funcionales
- 3.5.1.2.5 Establecimiento de Especificaciones de Entrega

En cada una de las actividades de esta etapa de Especificación Funcional del Sistema, se obtendrá una serie de productos que se concentrarán en el Documento de Especificación Funcional del Sistema (DEFS) y en el Documento de Pruebas del Sistema (DPS), los cuales serán el producto final en esta etapa, utilizando los recursos definidos en la matriz de Participación de Usuarios

---



**3.5.1.2.1 Realización del Modelo de las Funciones del Sistema**

Con esta actividad se pretende obtener el modelo de funciones del Sistema aplicando la técnica del Análisis Estructurado:

**DESCRIPCIÓN DEL PROCEDIMIENTO DE SOLICITUD DE GASTOS DE VIAJE Y VIÁTICOS**

AREA RESPONSABLE	ACT. No.	DESCRIPCION
Solicitante	1	Realiza la solicitud de viáticos y transportación proporcionando numero de empleado al modulo de solicitud del sistema de viáticos.
Sistema de viáticos	2	Recibe la solicitud de viáticos y revisa con el numero de empleado alguna posible deuda en el sistema de viáticos.
Sistema de viáticos	3	Si el empleado que solicita Viáticos tiene algún adeudo de alguna comisión anterior, le informa al empleado que no podrá solicitar viáticos hasta que no liquide sus adeudos y no le permite continuar con la solicitud, en caso contrario continua.
Solicitante	4	Captura todos los datos que le solicite el sistema de viáticos, corrige la información si en necesario.
Solicitante	5	Imprime solicitud de viáticos no autorizada.
Solicitante	6	¿Requiere modificaciones? Si la respuesta es positiva se regresa a la actividad no. 4
Solicitante	7	Si no requiere modificaciones la solicitud de viáticos, solicita el Vo. Bo. al funcionario que corresponda.
Funcionario	8	Recibe la(s) solicitud(es) de viáticos en el modulo de autorización del sistema de viáticos y analiza la solicitud de viáticos.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

Funcionario	9	¿Procede la solicitud de viáticos? Si la respuesta es positiva autoriza la solicitud de viáticos e informa al usuario. Si la respuesta es negativa informa al usuario para su modificación o cancelación.
Sistema de viáticos	10	En cuanto el funcionario autorice la solicitud de viáticos, el solicitante no podrá hacer ya ninguna modificación en esa solicitud y automáticamente aparecerá en los módulos de tesorería y el de Recursos Materiales, el monto de viáticos y el boleto de avión respectivamente.
Solicitante	11	Imprime solicitud de viáticos autorizada por el funcionario.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

		automáticamente aparecerá en el modulo de contabilidad la comprobación para visto bueno de esa área en el modulo de contabilidad del sistema de viáticos.
Solicitante	10	Imprime comprobación de viáticos autorizada por el funcionario, pero faltara el visto bueno del área de contabilidad.

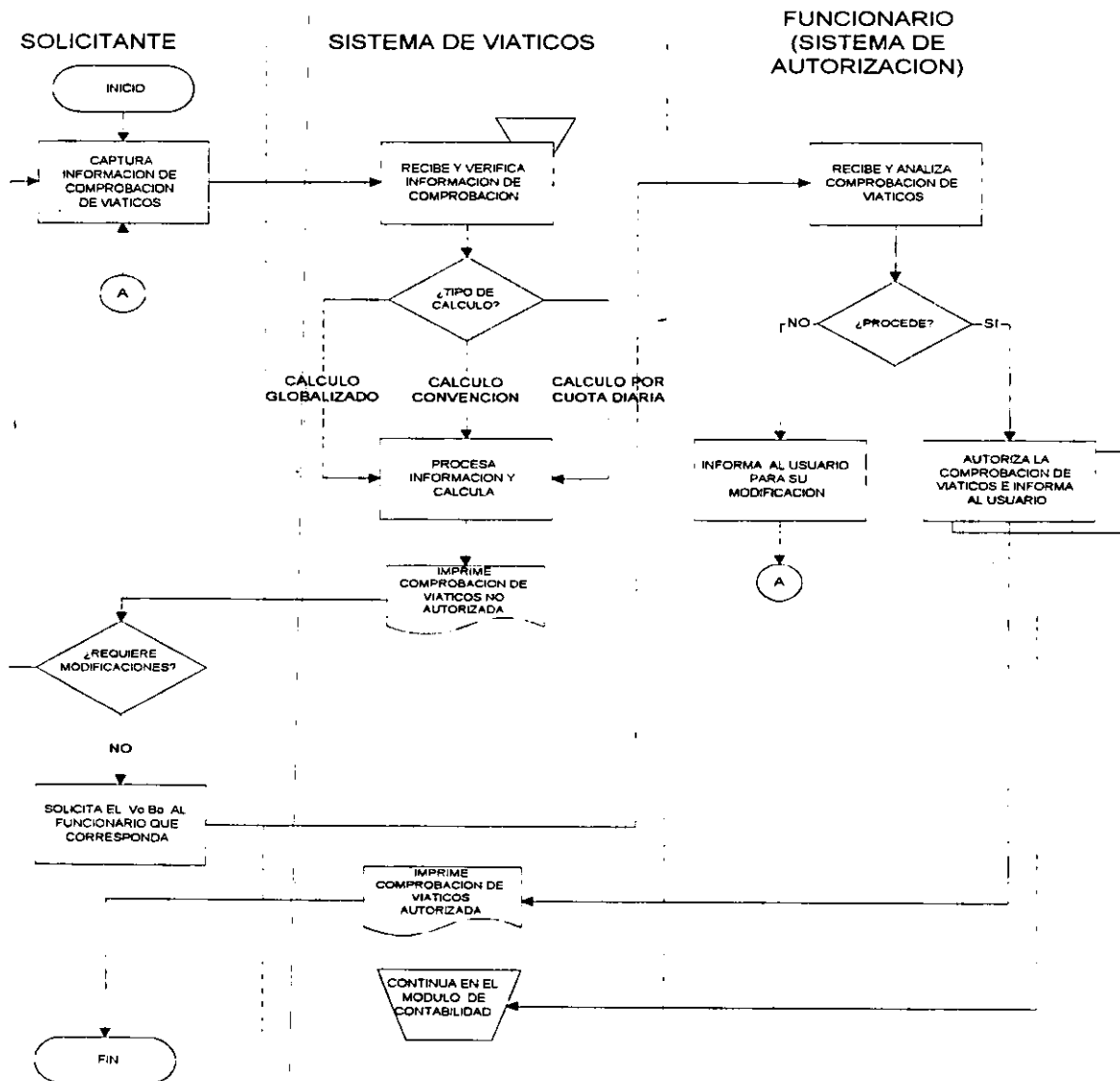


# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

#### PROCEDIMIENTO DEL MODULO DE LA COMPROBACION DE GASTOS DE VIAJE Y VIATICOS





#### **Técnicas Utilizadas:**

Para el desarrollo del modelo de funciones se utilizarán técnicas de análisis estructurado como son los Diagramas de Flujo de Datos y los Diagramas de Descomposición Funcional. También se utilizaran técnicas de este tipo de análisis para la descripción de las funciones elementales o primitivas, como son el pseudocódigo, los diagramas de acción y las tablas de decisión.

---

#### **3.5.1.2.2. Realización del Modelo de Datos del Sistema**

El producto principal que se debe obtener son los diagramas correspondientes al modelo, donde se representen las entidades y relaciones entre ellas. Estos diagramas deben complementarse con un catálogo donde se describan los objetos que aparecen en los mismos y otros aspectos asociados, como las reglas de integridad, estimación de volumen, frecuencia de uso, tiempo medio de almacenamiento y, en general, cualquier información que pueda ser de utilidad para las siguientes fases del ciclo de vida.

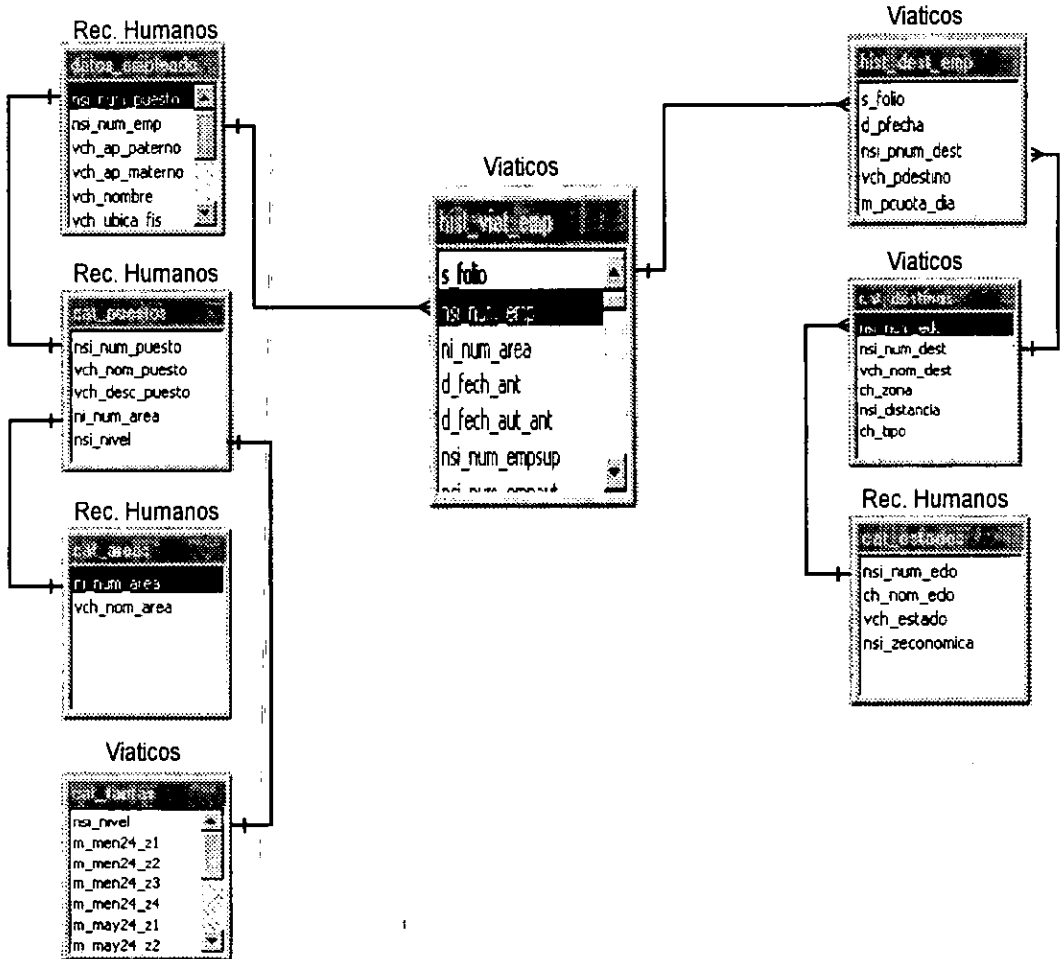


# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

Relaciones existentes en el Sistema de Viaticos  
Entre las Bases de Datos de Viaticos y Recursos Humanos  
Lunes, 15 de Mayo del 2000



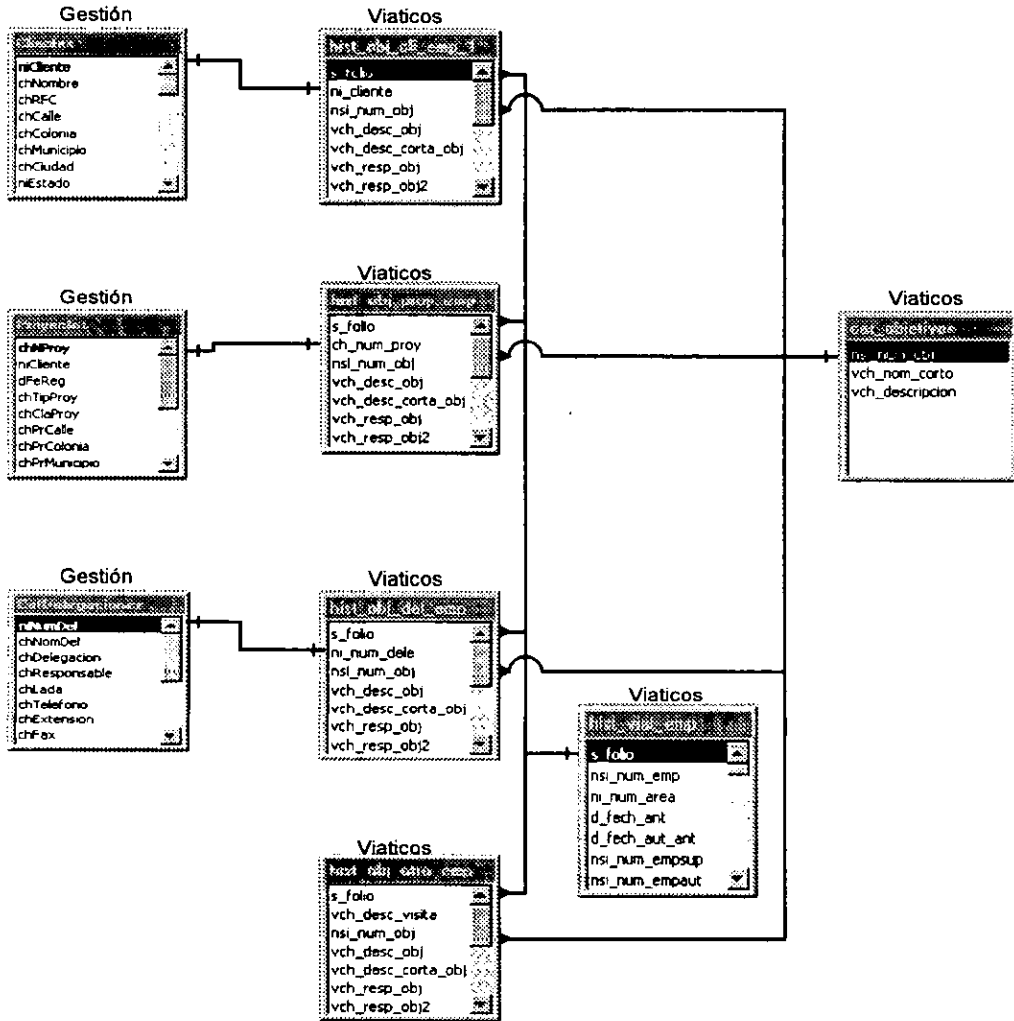


# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

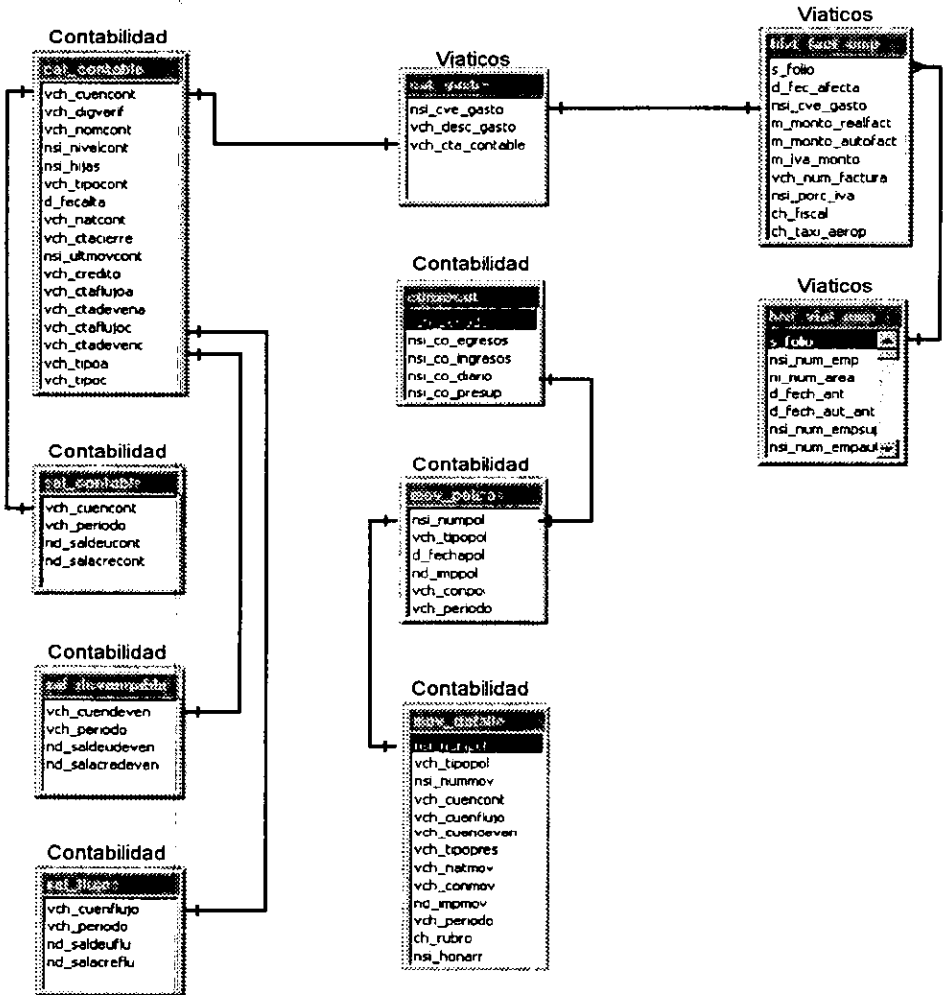
### Ejemplo práctico

**Relaciones existentes en el Sistema de Viáticos**  
**Entre las Bases de Datos de Viaticos y Gestión**  
**Lunes, 15 de Mayo del 2000**





**Relaciones existentes en el Sistema de Viáticos y Contabilidad**  
**Entre las Bases de Datos de Viáticos y Contabilidad**  
**Lunes, 15 de Mayo del 2000**







## MIGRACIÓN DE SOFTWARE

Capítulo III "Metodología para la migración"

Ejemplo práctico

### **Técnicas Utilizadas:**

La técnica del modelado conceptual más utilizada es la denominada análisis o modelo entidad-relación, que contempla también la normalización del modelo mediante las técnicas de transformación a primera y sucesivas formas normales.

---



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

#### 3.5.1.2.3. Diseño de interfaces de Usuarios y prototipos

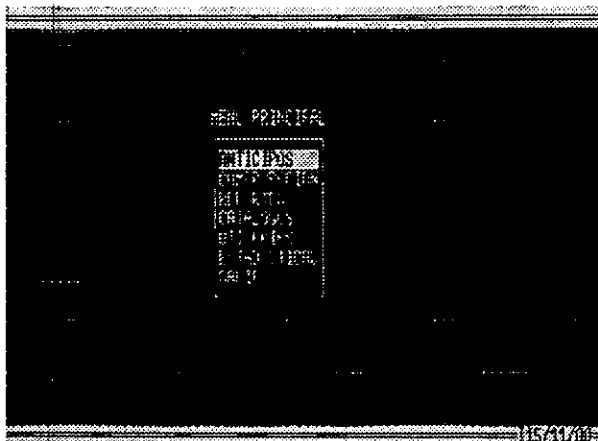
- Formato de las pantallas: Estos formatos consistirán en una descripción sobre papel de las pantallas del sistema.
- Diagrama conversacional: Representación jerárquica de las pantallas del sistema, que permite comprobar la posible navegación que se puede establecer a través de ellas. En este diagrama se pueden identificar los grupos de usuarios que utilicen las pantallas, así como los diálogos críticos para el correcto funcionamiento del sistema por ser utilizados con gran frecuencia, por implicar a una gran cantidad de entidades o atributos de datos, por contener opciones complejas por estar asociados a funciones importantes del sistema, por suponer la ejecución de un procesamiento por lotes (batch), etc.
- Formatos de los informes generados por el sistema

#### DISEÑO DE PANTALLAS

##### Pantalla Inicial

Las siguientes figuras nos muestran la pantalla inicial del sistema de viáticos en su forma original y el prototipo para el nuevo sistema, en ésta última se observa la utilización de objetos

#### CLIPPER



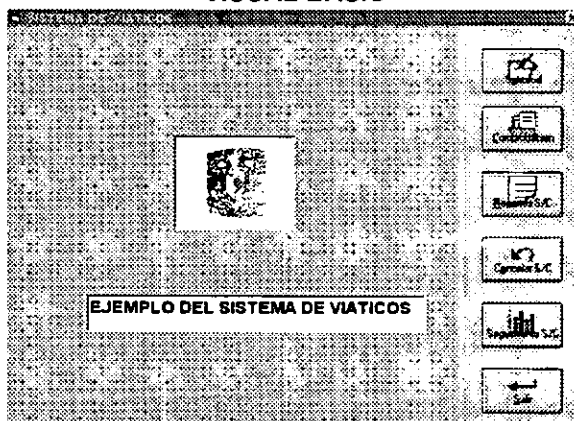


## MIGRACIÓN DE SOFTWARE

Capítulo III "Metodología para la migración"

Ejemplo práctico

### VISUAL BASIC



a continuación se presenta la documentación de la pantalla de inicio.

### SOLICITUD

Nombre de la forma:

FRMSOLICITUD.FRM - SOLICITUD DE VIÁTICOS

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Viáticos	Consecutivo	RDO/Execute	sql
Viáticos	Cat_destinos	RDO/Execute	
Viáticos	Cat_tarifas	RDO/Execute	
Viáticos	Hist_dest_emp	RDO/Execute	sql
Viáticos	Cat_objetivos	RDO/Execute	
Viáticos	Hist_obj_cli_emp	RDO/Execute	sql
Viáticos	Hist_obj_del_emp	RDO/Execute	sql
Viáticos	Hist_obj_otro_emp	RDO/Execute	sql
Viáticos	Hist_obj_proy_emp	RDO/Execute	sql
Viáticos	Hist_saldos_emp	RDO/Execute	sql
Rec. Humanos	Datos_empleado	RDO/Execute	
Rec. Humanos	Cat_areas	RDO/Execute	
Rec. Humanos	Cat_estados	RDO/Execute	
Rec. Humanos	Cat_puestos	RDO/Execute	
Gestión	Cientes	RDO/Execute	



# MIGRACIÓN DE SOFTWARE

## Capitulo III "Metodología para la migración"

### Ejemplo práctico

Gestión	Proyectos	RDO/Execute	
Gestión	CatDelegaciones	RDO/Execute	



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

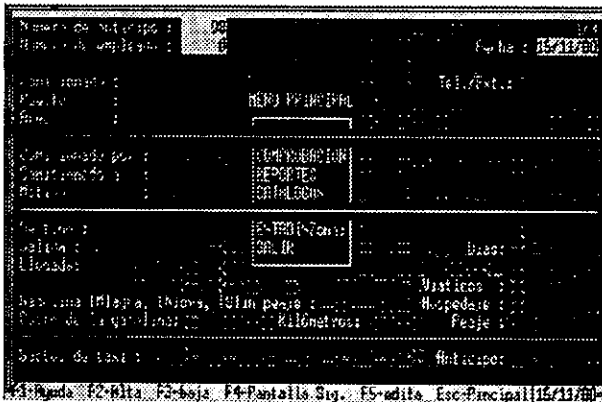
Nombre del fólder: Fólder Datos del Empleado

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Viáticos	Consecutivo	RDO/Execute	sql
Viáticos	Hist_saldos_emp	RDO/Execute	sql
Rec. Humanos	Datos_empleado	RDO/Execute	
Rec. Humanos	Cat_areas	RDO/Execute	
Rec. Humanos	Cat_estados	RDO/Execute	
Rec. -Humanos	Cat_puestos	RDO/Execute	

Fólder de Datos del Empleado.

CLIPPER





## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

#### VISUAL BASIC

#### **Consideraciones:**

- El folio es un numero consecutivo que asigna el sistema
- Solo deberá teclearse el numero de empleado
- Los demás campos de este fólder se llenaran automáticamente (Datos Empleado)
- El campo de saldo se calcula de la tabla de hist\_saldos\_emp cuando el empleado tenga saldo en contra
- Este sstab tiene el indice "0" y su nombre es "Datos Empleado", no tiene datos referenciados



# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

Nombre del fólдер: Fólдер Datos de la Comisión

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Rec. Humanos	Datos_ empleado	RDO/Execute	
Rec. Humanos	Cat_areas	RDO/Execute	
Rec. Humanos	Cat_estados	RDO/Execute	
Rec. Humanos	Cat_puestos	RDO/Execute	

Fólдер de Datos de la Comisión.

CLIPPER

The screenshot shows a CLIPPER application window with a menu bar at the top containing: Archivo, F2-Arriba, F3-Abajo, F4-Pantalla, F5-Exp, F6-Editar, Esc, F10-Principal, and F15-Cl. The main area of the window contains a form with the following fields and labels:

- Nombre de empresa: [ ]
- Fecha: 15/01/00
- Destino: [ ]
- Fecha de salida: [ ]
- Fecha de llegada: [ ]
- Viáticos: [ ]
- Hostedate: [ ]
- Costo de la gasolina: [ ]
- Kilómetros: [ ]
- Costo de taxi: [ ]
- Anticipo: [ ]

Below the form, there are several menu items: F2-Arriba, F3-Abajo, F4-Pantalla, F5-Exp, F6-Editar, Esc, F10-Principal, and F15-Cl.



# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

#### VISUAL BASIC

#### Consideraciones:

- Para seleccionar el nombre de la persona que autoriza seleccionarlo del combo, el cual se carga en el evento cmdfind de folder "datos del empleado" con los siguientes campos vch\_ap\_paterno, vch\_ap\_materno y vch\_nombre de la base de datos de recursos humanos y de Nivel Subdirector hasta Director General, ordenados alfabéticamente
- De igual forma se realizara para Acompañado por un Superior, seleccionando el Check Box correspondiente el cual se carga en el evento cmdfind de folder "datos del empleado" con los siguientes campos vch\_ap\_paterno, vch\_ap\_materno y vch\_nombre de la base de datos de recursos humanos y de Nivel Gerente hasta Director General, ordenados alfabéticamente.
- La relación de personas que Autorizan será de nivel Subdirector hacia arriba y las del Superior que Acompañan de Gerente hacia arriba. (Catalogo de Empleados)
- Este sstab tiene el indice "1" y su nombre es "Datos Comisión", no tiene datos referenciados





## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

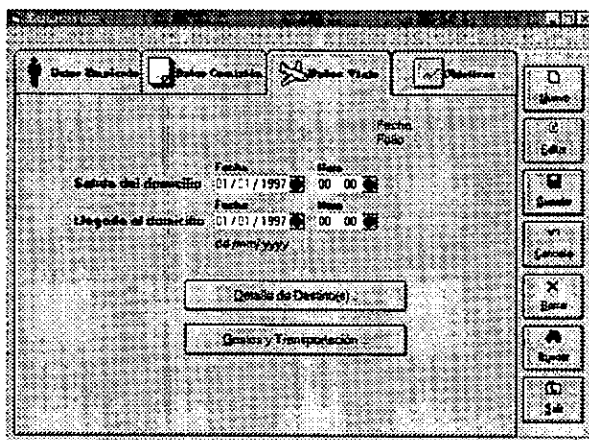
Nombre del fólder: Fólder Datos del Viaje

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql

Fólder de Datos del Viaje

VISUAL BASIC



#### Consideraciones:

- Seleccionar con los Spin boton las fechas y horas
- Las fechas y hora de Solicitud de Viáticos no podrán ser anteriores a la actual
- La fecha y hora de llegada deberá ser mayor a la fecha y hora de salida.
- Este sstab tiene el indice "2" y su nombre es "Datos Viaje", no tiene datos referenciados y se enlaza con dos formas (detalle de destinos y gastos de transportación)



# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

Nombre de la forma: Destinos

FRMDESTINOS.FRM - SOLICITUD DE VIATICOS

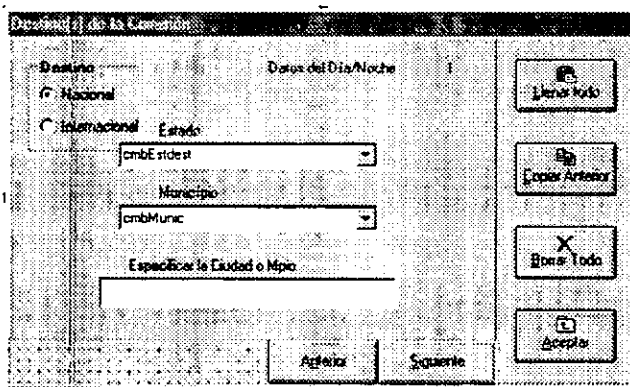
Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	Sql
Viáticos	Cat_destinos	RDO/Execute	
Viáticos	Cat_tarifas	RDO/Execute	
Viáticos	Hist_dest_emp	RDO/Execute	Sql

### CLIPPER



### VISUAL BASIC







# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

#### VISUAL BASIC

#### Consideraciones:

- Para llenar la parte correspondiente a avión, autobús o automóvil deberá seleccionarse el check botón de la izquierda correspondiente y así poder introducir los datos correspondientes. ( los check boton son independientes entre si)
- Los campos de importe son longitud (7,2)
- El campo de kilometraje es un Small Integer ( -32,767 a 32,767 )
- El campo de observaciones es de 200 caracteres.
- Los campos de itinerarios son de 100 caracteres

Nombre del folder : Folder de Objetivos

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Viáticos	Cat_objetivos	RDO/Execute	
Viáticos	Hist_obj_cli_emp	RDO/Execute	sql
Viáticos	Hist_obj_del_emp	RDO/Execute	sql
Viáticos	Hist_obj_otro_emp	RDO/Execute	sql
Viáticos	Hist_obj_proy_emp	RDO/Execute	sql
Gestión	Cientes	RDO/Execute	
Gestión	Proyectos	RDO/Execute	
Gestión	CatDelegaciones	RDO/Execute	



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

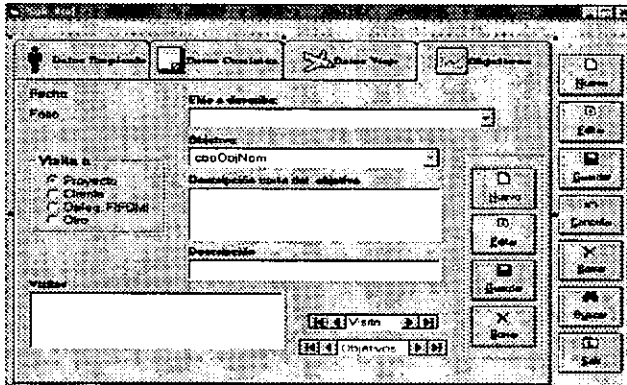
#### Ejemplo práctico

#### Fólder de Objetivos

#### CLIPPER

El fólder de objetivos es una de las mejoras realizadas al sistema, por lo tanto no existe en clipper.

#### VISUAL BASIC



#### **Consideraciones:**

- Los Proyectos, Clientes y/o Delegaciones serán los registrados en la base de datos de Gestión.(Proyectos, Clientes y Cat\_delegaciones), se cargaran los datos en el evento clic del option boton "del frame "fraTipVisita"
- Utilizar los opción botón para seleccionar que tipo de visita se realizara
- Podrá seleccionar "n" proyectos, Clientes y/o delegaciones.
- Para seleccionar el objetivo, seleccionarlo del combo (Catalogo de objetivos)
- La longitud de la descripción corta del objetivo es de 255 caracteres
- Este sstab tiene el índice "3" y su nombre es "Objetivos", no tiene datos referenciados



# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

#### COMPROBACION

Nombre de la forma:

FRMCOMPRO.FRM - COMPROBACIÓN DE VIÁTICOS

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Viáticos	Cat_objetivos	RDO/Execute	
Viáticos	Hist_obj_cli_emp	RDO/Execute	sql
Viáticos	Hist_obj_del_emp	RDO/Execute	sql
Viáticos	Hist_obj_otro_emp	RDO/Execute	sql
Viáticos	Hist_obj_proy_emp	RDO/Execute	sql
Viáticos	Cat_destinos	RDO/Execute	
Viáticos	Hist_dest_emp	RDO/Execute	sql
Viáticos	Cat_tarifas	RDO/Execute	
Viáticos	Cat_gastos	RDO/Execute	
Viáticos	Hist_comp_emp	RDO/Execute	
Viáticos	Hist_fact_emp	RDO/Execute	
Viáticos	Hist_otrosgas_emp	RDO/Execute	
Viáticos	Hist_kms	RDO/Execute	
Rec. Humanos	Datos_empleado	RDO/Execute	
Rec. Humanos	Cat_areas	RDO/Execute	
Rec. Humanos	Cat_estados	RDO/Execute	
Rec. Humanos	Cat_puestos	RDO/Execute	
Gestión	Cientes	RDO/Execute	
Gestión	Proyectos	RDO/Execute	
Gestión	CatDelegaciones	RDO/Execute	

Nombre del fólдер : Fólдер Datos Generales

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Rec. Humanos	Datos_empleado	RDO/Execute	
Rec. Humanos	Cat_areas	RDO/Execute	
Rec. Humanos	Cat_estados	RDO/Execute	
Rec. Humanos	Cat_puestos	RDO/Execute	



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico



#### Fólder de Datos Generales

##### CLIPPER

Esta pantalla no aparecía en el sistema anterior, solamente se ligaba por el número de folio.

##### VISUAL BASIC

#### **Consideraciones:**

- El folio será el asignado en la solicitud del sistema (histórico de Viáticos) los datos se cargan automáticamente de la base de datos de viáticos (hist\_viat\_emp) según el folio tecleado y sus datos generales de la base de datos de recursos humanos.
- Solo deberá teclearse el folio
- Este sstab tiene el índice "1" y su nombre es "Datos Generales", no tiene datos referenciados

Nombre del fólder: Fólder Objetivos

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Viáticos	Cat_objetivos	RDO/Execute	
Viáticos	Hist_obj_cli_emp	RDO/Execute	sql
Viáticos	Hist_obj_del_emp	RDO/Execute	sql
Viáticos	Hist_obj_otro_emp	RDO/Execute	sql
Viáticos	Hist_obj_proy_emp	RDO/Execute	sql



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

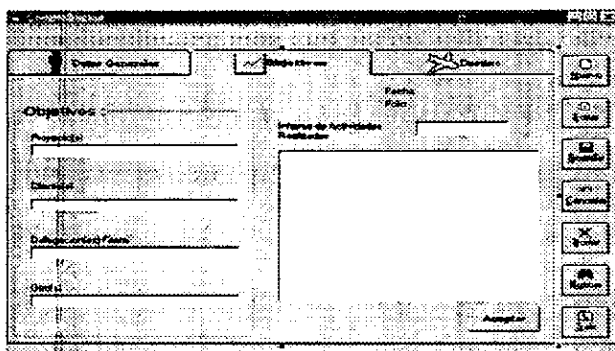
Gestión	Clientes	RDO/Execute	
Gestión	Proyectos	RDO/Execute	
Gestión	CatDelegaciones	RDO/Execute	

#### Fólder de Objetivos

#### CLIPPER

El fólder de objetivos es una de las mejoras realizadas al sistema, por lo tanto no existe en clipper.

#### VISUAL BASIC



#### **Consideraciones:**

- La fecha y número de folio (aparecen automáticamente).
  - El campo de respuesta de objetivos es de dos cuartillas para cada respuesta de objetivos y para responder un objetivo específico deberá seleccionarlo primeramente del recuadro izquierdo (actualizan las siguientes tablas hist\_obj\_cli\_emp y/o hist\_obj\_del\_emp y/o hist\_obj\_otros\_emp y/o hist\_obj\_del\_emp)
  - Este sstab tiene el índice "2" y su nombre es "Objetivos", no tiene datos referenciados





## MIGRACIÓN DE SOFTWARE

### Capitulo III "Metodología para la migración"

#### Ejemplo práctico

Nombre del fólder : Fólder Destino

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	sql
Viáticos	Cat_destinos	RDO/Execute	
Viáticos	Hist_dest_emp	RDO/Execute	sql

### Fólder de Destino

#### CLIPPER

Esta pantalla no existe en clipper.

#### VISUAL BASIC

#### Consideraciones:

- Seleccionar con los Spin boton las fechas y horas reales de la comisión
  - La fecha y hora real de llegada deberá ser mayor a la fecha y hora real de salida.
  - Este sstab tiene el índice "2" y su nombre es "Destino", no tiene datos referenciados pero enlaza la forma desglose de facturas



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

Nombre de la forma: General de facturas

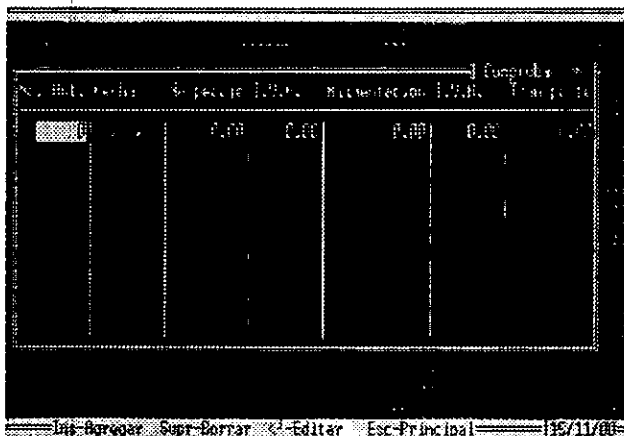
FRMFACTURA.FRM - COMPROBACION DE VIATICOS

Nombre de las tablas que utiliza:

Base de Datos	Tablas que Utiliza	Lectura	Escritura
Viáticos	Hist_viat_emp	RDO/Execute	
Viáticos	Cat_destinos	RDO/Execute	
Viáticos	Cat_tarifas	RDO/Execute	
Viáticos	Cat_gastos	RDO/Execute	
Viáticos	Hist_comp_emp	RDO/Execute	
Viáticos	Hist_fact_emp	RDO/Execute	
Viáticos	Hist_dest_emp	RDO/Execute	
Viáticos	Hist_otrosgas_emp	RDO/Execute	
Viáticos	Hist_kms	RDO/Execute	
Rec. Humanos	Datos_empleado	RDO/Execute	
Rec. Humanos	Cat_estados	RDO/Execute	

Desglose de Facturas

CLIPPER





## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

## VISUAL BASIC

### Consideraciones:

- Para Seleccionar el Estado, seleccionarlo del combo, este se carga en el evento load, solamente se considera localidades nacionales con los siguientes campos vch\_estado de la tabla catálogo de estados de la base de datos de recursos humanos, ordenados alfabéticamente
- La localidad se refiere a los Municipios del estado seleccionado, éste se carga en el evento clic del objeto seleccionar estado, con los siguientes campos nsi\_num\_dest, cat\_destinos.vch\_nom\_dest de la tabla catálogo de destinos de la base de datos de viáticos, ordenados alfabéticamente
- Los conceptos se refiere a los que se utilizan en la comprobación de viáticos estos se cargan en el evento load, con los siguientes campos nsi\_cve\_gasto, vch\_desc\_gasto de la tabla catalogo de gastos de la base de datos de viáticos
- Los campos de importe, iva y precio de gasolina son longitud (7,2)
- El campo de kilometraje es un Small Integer (-32,767 a 32,767)
- El campo de factura es de 15 caracteres.
- Los botones de anterior y siguiente permitirán navegar en los días de la comisión, en el caso de que sea de un solo día, no aparecerán.
- Los campos de kilometraje inicial, final y precio de la gasolina solo aparecerán cuando se compruebe algún concepto de gasolina



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

## DISEÑO DE REPORTES

Formatos de los informes generados por el sistema

Modulo : Solicitud Comprobación de Viáticos y Tesorería

Reporte : Solicitud de Anticipo para gastos de viaje y transportación

Nombre : rep001.rpt

Herramienta : Cristal Report Ver. 6.0

Descripción: Este reporte contiene la información inherente de la solicitud de un anticipo de viáticos.

Formulas

Nombre	Codigo	Descripción
AUTO PROPIO	if {hist_viat_emp.vch_itin_auto}="" then " " else if {hist_viat_emp.ch_auto_recmat}="" then "PROPIO" else " "	Esta formula verifica si se utiliza automovil, si es correcto y observa que no es de Recursos materiales implica que el auto es propio
AUTORIZADA	if IsNull ( {hist_viat_emp.d_fech_aut_ant}) then 'NO AUTORIZADA' else 'AUTORIZADA'	Verifica el campo de autorizacion de la solicitud, si esta lleno pone "Autorizada", de lo contrario pone "No Autorizada"
FE_HO_RE	{hist_viat_emp.d_pfecha_lle}	Pone la fecha propuesta de regreso de la comisión
TOTAL_VIAT	{rep_paso_sol.m_pvia_alimen} + (({hist_viat_emp.nsi_km} * {hist_viat_emp.m_costo_gas} / 5) + {rep_paso_sol.m_pvia_peaje} + {hist_viat_emp.m_pvia_legal} + {rep_paso_sol.m_pvia_foto} + {rep_paso_sol.m_pvia_otros} + {rep_paso_sol.m_pvia_bus}	Calcula el monto total de la comision de los diferentes conceptos
... otras ...		
Sentencia SQL	SELECT rep_paso_sol.s_folio, rep_paso_sol.nsi_num_emp, rep_paso_sol.vch_nombre,	



# MIGRACION DE SOFTWARE

## Capitulo III "Metodología para la migración"

### Ejemplo práctico

	rep_paso_sol.vch_nom_puesto, rep_paso_sol.vch_nom_area, rep_paso_sol.vch_num_ext, rep_paso_sol.vch_nom_comisiona, rep_paso_sol.vch_nom_supaco, rep_paso_sol.vch_destino, rep_paso_sol.objetivo_1, rep_paso_sol.objetivo_2, rep_paso_sol.objetivo_3, rep_paso_sol.objetivo_4, rep_paso_sol.m_pvia_alimen, rep_paso_sol.m_pvia_bus, rep_paso_sol.m_pvia_peaje, rep_paso_sol.m_pvia_foto, rep_paso_sol.m_pvia_otros, rep_paso_sol.vch_itin_avion FROM informix.rep_paso_sol ,bolivos.hist_viat_emp WHERE rep_paso_sol.s_folio=bolivos.hist_viat_emp.s_folio
Tablas que utiliza	hist_viat_emp rep_paso_sol
Llamado por	FRMREPORTES.FRM

**SOLICITUD DE ANTICIPA PARA GASTOS DE VIAJE Y TRANSPORTACION EN COMISION** No. 640

Comisionado: VIDAL VAZQUEZ JUAN IGNACIO  
 PUESTO: JEFE DE AREA  
 AREA: DIRECCION GENERAL  
 UNIDAD ORGANIZACIONAL: FELIX GAMIZ HORACIO  
 NOMBRE: FELIX GAMIZ HORACIO  
 OBJETIVO: REPOSICION DE GASTOS

Fecha: 17 de Agosto de 2001  
 No. DE NOMINA: 112

TRANSPORTO:	FECHA Y HORA DE SALIDA:	DEPARTO:	10:00
PLANEADO:	FECHA Y HORA LLEGADA:	DEPARTO:	20:00
DESTINO:	CHURRASACA		
FECHA DE SALIDA Y LLEGADA (AVION):			
VEHICULO:			

ANEXOS: GASTOS EN VIAJE EN COMISION  
 IMPORTE QUE LO ACOMPAÑA

RECIBI DE LA CATA (MR. REINVENTOS) DE FUNDOS MINERALS LA CANTIDAD DE \$

PER CONCEPTO DE ANTICIPA DE:	1 DIA DE VIAJE PARA GASTOS (TRAMO EN RE)	DE GASTOS:	\$	0.00
VIAJE EN:	17.00	PERMANENCIA:	\$	0.00
TRAVEL:	20.00	ALIMENTOS:	\$	0.00
GAS DE:	200 KMS. APPROX. A S 4.00	OTROS:	\$	0.00
	170.00	TOTAL A RECONSTRUIR:	\$	0.00

NOTA: CON PREVIOS PAGOS DE GASTOS EN VIAJE EN COMISION, EL CANTO DE ANTICIPA DEBE SER DE \$0.00. EN CASO DE NO PAGAR EL CANTO DE ANTICIPA DEBE SER DE \$0.00. EN CASO DE NO PAGAR EL CANTO DE ANTICIPA DEBE SER DE \$0.00. EN CASO DE NO PAGAR EL CANTO DE ANTICIPA DEBE SER DE \$0.00.

COMISIONADO:	AUTORIZACION:
VIDAL VAZQUEZ JUAN IGNACIO	FELIX GAMIZ HORACIO
NOMBRE:	NOMBRE:
	SOLICITUD AUTORIZADA DE:
PERSONA QUE RECIBE EL IMPORTE EN:	
FECHA:	



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

Modulo : Solicitud Comprobación de Viáticos y Tesorería

Reporte : Comprobación de gastos de viaje y transportación

Nombre : rep002.rpt

Herramienta : Cristal Report Ver. 6.0

Descripción: Este reporte contiene la información inherente de la comprobación de una comisión de viáticos.

#### Formulas

Nombre	Código	Descripción
ALIMEN	{rep_paso_comp.cpto_2} + {rep_paso_comp.cpto_3}	Suma el total de Concepto de Alimentación (Cpto 2 y 3)
AUTORIZADA	if IsNull ( {hist_viat_emp.d_fec_aut_comp}) then 'NO AUTORIZADA' else 'AUTORIZADA'	Verifica el campo de autorización de la comprobación, si esta vacío pone "No Autorizado" de lo contrario "Autorizada"
CALCULO_G LOBAL	if {hist_viat_emp.ch_cal_global}="S" then "Si" else "N"	Verifica el campo de calculo Global, si esta lleno pone "Si", de lo Contrario "N"
... otras...		

Sentencia SQL	<pre> SELECT   hist_viat_emp.ch_recibi_dinyba, hist_viat_emp.ch_cal_global,   hist_viat_emp.d_fecha_sal, hist_viat_emp.d_fecha_lle,   hist_viat_emp.ch_rhora_sal, hist_viat_emp.ch_rhora_lle,   hist_viat_emp.nsi_rdias, hist_viat_emp.m_pvia_alimen,   hist_viat_emp.m_pvia_hosp, hist_viat_emp.m_pvia_bus,   rep_paso_comp.cpto_2, rep_paso_comp.cpto_3,   rep_paso_comp.cpto_4, rep_paso_comp.cpto_5,   rep_paso_comp.cpto_6, rep_paso_comp.cpto_7,   rep_paso_comp.cpto_8, rep_paso_comp.cpto_9,   rep_paso_comp.cpto_10, rep_paso_comp.cpto_11,   rep_paso_comp.cpto_12, rep_paso_comp.cpto_13 FROM   informix.rep_paso_sol, bolivos.hist_viat_emp,   informix.rep_paso_comp WHERE   bolivos.hist_viat_emp.s_folio= informix.rep_paso_sol.s_folio ORDER BY   rep_paso_comp.s_folio ASC,   rep_paso_comp.d_fec_afecta ASC </pre>
Tablas que	hist_viat_emp



# MIGRACION DE SOFTWARE

## Capitulo III "Metodologia para la migración"

### Ejemplo práctico

utiliza	rep_paso_sol rep_paso_comp
Llamado por	FRMREPORTES.FRM

**COMPROBACION DE GASTOS DE VIAJE EN COMISION** No. 630

**COMISIONADO** VIDAL VAZQUEZ JUAN IGNACIO **MÉDICO** 11 P. 20 de Mayo de 2001  
**PUESTO** CHOFER S.P.S. **TÍT. VIO EXT** 155  
**AREA** DIRECCION GENERAL **Nº DE NOMINA** 3044

**HA SIDO COMISIONADO POR** FELIX GAMEZ HORACIO  
**DISTINO INICIAL** NOROLLOS NOROLLOS  
**OBJETIVO(S) DE LA COMISION** REPARACION MANTENIMIENTO OTROS

**FECHA Y HORA DE SALIDA** 04/04/00 10:00 **CUOTA DIARIA \$** 112  
**FECHA Y HORA LLEGADA** 04/04/00 20:00 **NUMERO DE DIAS** 1  
**RECIBI EFECTIVO Y/O BOLETO DE AVION** SI **CALCULO GLOBALIZADO** N

PLATA	RESERVA	ALIMENTACION	TRANSPORTE	TELÉFONO	REPARACION	DESCANSO	OTROS	TOTAL
0.00	0.00	0.00	239.11	0.00	0.00	0.00	0.00	239.11
<b>RENTOTAL</b>	<b>0.00</b>	<b>0.00</b>	<b>239.11</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>239.11</b>
<b>I.V.A.</b>	<b>0.00</b>	<b>0.00</b>	<b>7.87</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>7.87</b>
<b>TOTAL</b>	<b>0.00</b>	<b>0.00</b>	<b>246.98</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>246.98</b>

**VIDAL VAZQUEZ JUAN IGNACIO**  
**NOMBRE DEL COMISIONADO**

**FELIX GAMEZ HORACIO**  
**NOMBRE DEL FUNCIONARIO (O.T. COMISION)**

**RECIBI EFECTIVO** 239.11  
**BOLETO AVION** 0.00  
**SALDO** 246.98

**COMPROBACION AUTORIZADA** 11/04/00



## MIGRACIÓN DE SOFTWARE

Capítulo III "Metodología para la migración"

### Ejemplo práctico

Modulo : Solicitud Comprobación de Viáticos y Tesorería

Reporte : Informe de la Comisión

Nombre : rep003.rpt

Herramienta : Cristal Report Ver. 6.0

Descripción: Este reporte contiene la información inherente a la respuesta de objetivos de una comisión de viáticos.

#### Formulas

Nombre	Código	Descripción
OBJETIVOS	Trim ((rep_paso_sol.objetivo_1))+" "+Trim ((rep_paso_sol.objetivo_2))+ " " +Trim ((rep_paso_sol.objetivo_3))+" "+Trim ((rep_paso_sol.objetivo_4))	Realiza un string de los objetivos 1,2,3 y 4
Sentencia SQL	SELECT rep_pasoobj.ch01, rep_pasoobj.ch02, rep_pasoobj.ch03, rep_pasoobj.ch04, rep_pasoobj.ch05, rep_pasoobj.ch06, rep_pasoobj.ch07, rep_pasoobj.ch08, rep_pasoobj.ch09, rep_pasoobj.ch10, rep_paso_sol.s_folio, rep_paso_sol.nsi_num_emp, rep_paso_sol.vch_nombre, rep_paso_sol.vch_nom_puesto, rep_paso_sol.vch_nom_area, rep_paso_sol.vch_num_ext, rep_paso_sol.vch_nom_comisiona, rep_paso_sol.vch_destino, rep_paso_sol.objetivo_1, rep_paso_sol.objetivo_2, rep_paso_sol.objetivo_3, rep_paso_sol.objetivo_4 FROM rep_paso_sol,rep_pasoobj,hist_viat_emp WHERE rep_paso_sol.s_folio=rep_pasoobj.s_folio and rep_paso_sol.s_folio=hist_viat_emp.s_folio	
Tablas que utiliza	hist_viat_emp rep_paso_sol Rep_pasoobj	
Llamado por	FRMREPORTES.FRM	





# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

INFORME DE LA COMISION		No. 520	
	Polifuncionario de Personal Admisión		
NOMBRE	VIDAL VAZQUEZ JUAN IGNACIO	Edad	26 de Mayo de 7
CARGO	ES ROBERTO	Sexo	masculino
AREA	DIRECCION GENERAL	No. DE NOMINA	31004
DISTRITO DE COMISION	QUERETARO		
OBJETIVO(S) DE LA COMISION	REVISAR Y RECOMENDAR LA MIGRACION DE SOFTWARE		
PRINCIPALES ACTIVIDADES REALIZADAS Y/O RECOMENDACIONES			
LLEVAR A CABO LAS ACTIVIDADES DE REVISAR Y RECOMENDAR LA MIGRACION DE SOFTWARE AL GOBIERNO DEL ESTADO DEL CUERPO PARA INTEGRACION DEL SISTEMA.			
De ser, bajo protesta de decir verdad, que los datos contenidos en este informe son verdaderos y exactos como conocimiento de los funcionarios que se aplicaron en caso contrario.			
FELIX GARCIA HERRERA		VIDAL VAZQUEZ JUAN IGNACIO	
FUNCIONARIO QUE AUTORIZA LA COMISION		COMISIONADO	

### Técnicas Utilizadas:

La información necesaria para especificar las interfases de usuario (pantallas, diálogos, informes), además de obtenerse del documento de requisitos, se conseguirá de los usuarios por medio de entrevistas. Además, se podrán desarrollar prototipos utilizando alguna técnica de prototipado de pantallas y diálogos (herramientas CASE)



#### 3.5.1.2.4 Establecimiento de especificaciones no funcionales

Como resultado de esta actividad se obtendrá la lista de requisitos del sistema actualizada con los requisitos no funcionales como son para nuestro ejemplo, los relacionados con la seguridad y el rendimiento.

Los requisitos de seguridad serán aquellos necesarios para prevenir la pérdida de información y asegurar su integridad. Habrá que tener en cuenta las diferentes causas posibles de estos problemas, que pueden darse en tres niveles diferentes: físico (hardware), lógico (software) y nivel de operación.

Entre las medidas de seguridad que habrá de adoptar, se incluyen las siguientes: Protección de acceso, Contingencias, Tratamiento de errores, Seguridad del Almacenamiento.

En cuanto a los requisitos de rendimiento, estos incluirán los tiempos de respuesta requeridos, los volúmenes de transacciones, las circunstancias especiales de explotación como aquellos periodos de alto volumen de transacciones.

#### Técnicas Utilizadas:

Para la estimación de tiempos de respuesta y volúmenes de transacciones se requerirá la asistencia de los usuarios, por lo que serán necesarias entrevistas con los implicados.

---

#### 3.5.1.2.5 Establecimiento de especificaciones de entrega

Como resultado de esta actividad se obtendrán el plan de desarrollo y entrega del nuevo sistema, así como el plan de pruebas de aceptación.

#### Técnicas Utilizadas:

Para la elaboración del plan de pruebas de aceptación se realizarán entrevistas con los usuarios para asegurar que éstos, que han de decidir la aceptación final del sistema, están de acuerdo con dicho plan.

---



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

En cuanto a la planificación del desarrollo y entrega, se pueden utilizar técnicas de gestión de sistemas (PERT, CPM, diagramas de Gantt) como ayuda en la planificación del desarrollo y entrega del sistema, especialmente útiles se este se ha dividido en subsistemas que deben ser entregados de forma progresiva.

---

### Participación de Usuarios

La siguiente matriz muestra la participación de los usuarios en las diferentes actividades de esta fase:

	2.1	2.2	2.3	2.4	2.5
Comité de Informática					
Jefe del Proyecto				X	X
Analista			X	X	X
Programador			O		
Gerente del Área Involucrada					
Usuario Final			X	X	X

---



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo III "Metodología para la migración"**

#### **Ejemplo práctico**

### **3.5.2 Fase de Diseño**

La segunda fase del ciclo de vida es la del diseño del Sistema a desarrollar, en esta etapa se va a definir como se va a hacer el Sistema.

#### **OBJETIVOS DEL DISEÑO**

Los objetivos fundamentales de esta fase son los siguientes:

- Obtención de la estructura modular de funciones o arquitectura del sistema, definiendo componentes software (programas, subprogramas) o unidades de tratamiento, cuyo funcionamiento será descrito de forma estructurada. El punto de partida será el modelo de funciones generado en la fase de análisis.
- Obtención de la estructura de datos del Sistema o modelo de datos físico.
- Especificación del entorno en que funcionará el Sistema, incluyendo la organización del área de usuario final y de explotación y de la infraestructura que permitirá el correcto uso del sistema, Se deberán identificar los componentes tecnológicos necesarios para servir de soporte al sistema: hardware (computadoras, dispositivos de almacenamiento), software (sistema operativo, sistema de gestión de base de datos), comunicaciones (protocolos, controladores), entorno de explotación (herramientas de control).
- Establecimiento de las prueba necesarias para validar el sistema diseñado, que permitirán comprobar que éste satisface las especificaciones del la fase de análisis. Se trata en definitiva de completar el plan de pruebas que se comenzó a elaborarse al final de la fase de análisis.

La fase del diseño del Sistema, se puede estructurar en las siguientes etapas.

- 3.5.2.1.1 Diseño de la arquitectura de Funciones del Sistema
- 3.5.2.1.2 Diseño de la estructura de Datos del Sistema
- 3.5.2.1.3 Definición del Entorno Tecnológico del Sistema
- 3.5.2.1.4 Diseño del Plan de Pruebas del Sistema
- 3.5.2.1.5 Diseño del Plan de Construcción e Implantación



# MIGRACIÓN DE SOFTWARE

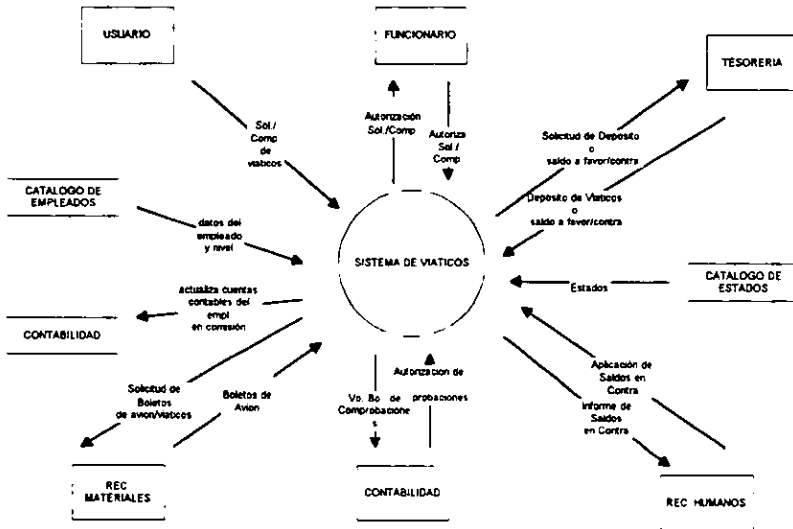
## Capítulo III "Metodología para la migración"

### Ejemplo práctico

#### 3.5.2.1.1 Diseño de la Arquitectura de Funciones del Sistema

Estructura modular del Sistema (que incluye la interfase entre módulos y con otros sistemas).

DIAGRAMA DE CONTEXTO DEL SISTEMA DE VIATICOS





# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

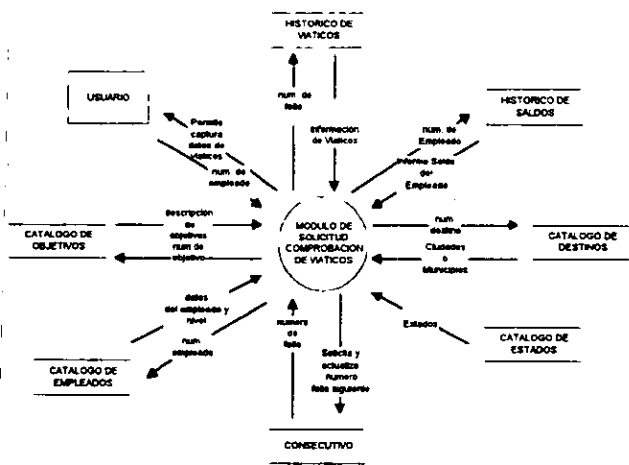
Representación gráfica con las entradas y salidas de datos

### Arquitectura Física (Nivel 0)



### Nivel (1)

DIAGRAMA DE FLUJO DE DATOS DE LA SOLICITUD DEL SISTEMA DE VIATICOS





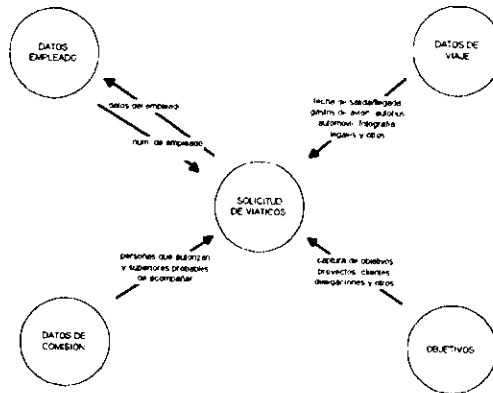
# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

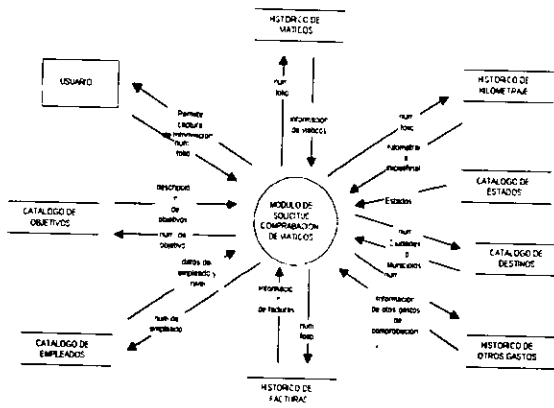
Nivel (2)

MODULO DE SOLICITUD DE VIATICOS



Nivel (1)

DIAGRAMA DE FLUJO DE DATOS DE LA COMPROBACION DEL SISTEMA DE VIATICOS



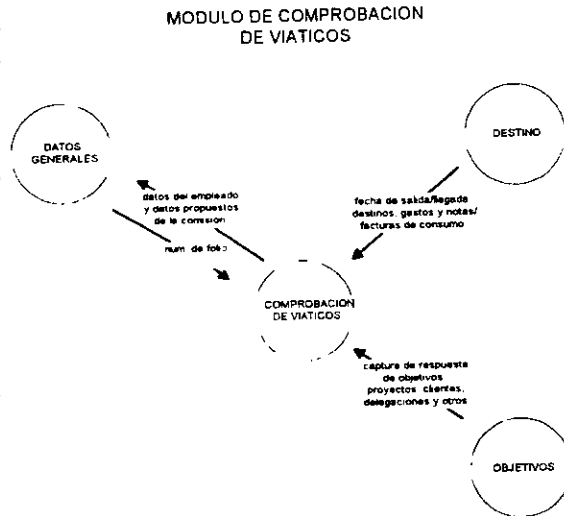


# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

Nivel (2)



### Técnicas Utilizadas

Para la obtención de la estructura modular del Sistema, se aplicarán las técnicas de diseño estructurado denominadas "análisis de transformación" y "análisis de transacción", que permiten derivar dicha estructura a partir de los diagramas de flujo de datos que se obtuvieron en la Fase de Análisis.

Representación gráfica con las entradas y salidas de datos de la unidad.





## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

#### 3.5.2.1.2 Diseño de la Estructura de Datos del Sistema

Modelo lógico de los datos manejados por las funciones mecanizables del sistema

Base de Datos: viatdesa

#### Tabla : Catalogo de Destinos

Nombre: cat\_destinos

Descripción: Contiene los municipios y ciudades de cada estado de la republica y se relaciona con el Catalogo de Estados de la Base de Datos de Recurso Humanos y ambos catálogos ( no se pueden borrar).

CAMPO	TIPO	NULO	DESCRIPCIÓN
nsi_num_edo	smallint	No	Número del estado
nsi_num_dest	smallint	No	Número del destino
vch_nom_dest	varchar(30,1)	No	Nombre del destino
ch_zona	char(1)	No	Zona Económica (valores 1-4) y según la zona toma el monto de la cuota de viáticos
nsi_distancia	Smallint	No	Distancia del D. F. al destino
ch_tipo	char(1)	No	Tipo de destino (Nal/Inal.)

#### Índices de la Tabla

Nombre	Campos	Origen	Primario	Foraneo
lx129_1	nsi_num_edo	Rec. Humanos	XX	
lx129_2	nsi_num_dest	Viáticos		XX

#### Archivos del sistema que la utilizan

Tipo de Acceso	Archivo
R	Formas: frmdestinos, frmfactura, frmcancela, frmseguimiento, frm_pant_aut_sol, frm_pant_aut_comp, Reportes: rep001, rep002, rep003, rep004, rep005, rep006, rep007
W	Tabla no modificable, de nos ser por intrucciones del responsable del sistema



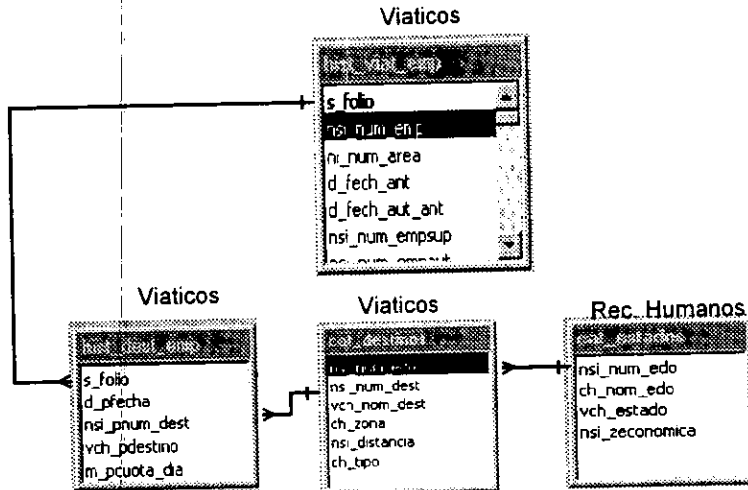
# MIGRACIÓN DE SOFTWARE

## Capítulo III "Metodología para la migración"

### Ejemplo práctico

#### Relaciones

Tabla	Tipo
Cat_estados	Uno a Varios



Base de Datos: viatdesa

Tabla : Catalogo de Gastos

Nombre: cat\_gastos

Descripción: Contiene los gastos permitidos en la comisión de viáticos, así como las cuentas contables correspondientes a afectar y se relaciona con la base de datos de contabilidad para su afectación. Cuando exista modificación en las cuentas contables, se deberá actualizar dicho catalogo (no se debe borrar)

CAMPO	TIPO	NULO	DESCRIPCIÓN
nsi_cve_gasto	smallint	No	Clave del gasto
vch_desc_gasto	varchar(15,1)	No	Descripción del gasto
vch_cta_contable	varchar(40,1)	No	Cuenta Contable del gasto



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

#### Índices de la Tabla

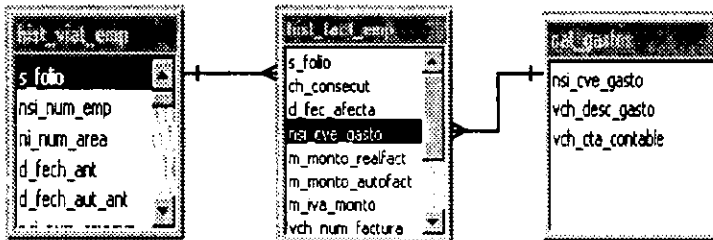
Nombre	Campos	Origen	Primario	Foraneo
lx161_1	nsi_cve_gasto	Viáticos	XX	

#### Archivos del sistema que la utilizan

Tipo de Acceso	Archivo
R	Formas: frmfactura, frm_pant_aut_sol, frm_pant_aut_comp, frmcvrm Reportes:
W	Tabla no modificable, de nos ser por instrucciones del responsable del sistema

#### Relaciones

Tabla	Tipo
Hist_fact_emp	Uno a Varios



Base de Datos: viatdesa

Tabla : Catalogo de Tarifas de Viáticos y catalogo de Tarifas de Viáticos nuevas

Nombre: cat\_tarifas, cat\_tarifas\_nva

Descripción: Contiene las cuotas autorizadas por zona, tiempo de duración de la comisión y nivel, cada vez que exista cambio de tarifas, cargarlas en la tabla "cat\_tarifas\_nva" y en el programa de solicitud/comprobación poner la fecha a partir de cuando toma efecto (no se deben borrar)



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

CAMPO	TIPO	NULO	DESCRIPCIÓN
<b>Nsi_nivel</b>	smallint	No	Nivel del empleado
m_men24_z1	money(7,2)	No	Cuota menor a 24 hrs. zona 1
m_men24_z2	money(7,2)	No	Cuota menor a 24 hrs. zona 2
m_men24_z3	money(7,2)	No	Cuota menor a 24 hrs. zona 3
m_men24_z4	money(7,2)	No	Cuota menor a 24 hrs. zona 4
m_may24_z1	money(7,2)	No	Cuota mayor a 24 hrs. zona 1
m_may24_z2	money(7,2)	No	Cuota mayor a 24 hrs. Zona 2
m_may24_z3	money(7,2)	No	Cuota mayor a 24 hrs. Zona 3
m_may24_z4	money(7,2)	No	Cuota mayor a 24 hrs. Zona 4
m_int_z1	money(7,2)	No	Cuota Internacional zona 1
m_int_z2	money(7,2)	No	Cuota Internacional zona 2
m_int_z3	money(7,2)	No	Cuota Internacional zona 3

#### Índices de la Tabla

Nombre	Campos	Origen	Primario	Foraneo
lx104_1	nsi_nivel	Rec. Humanos		XX

#### Archivos del sistema que la utilizan

Tipo de Acceso	Archivo
R	Formas: frmsolicitud, frmcompro, frmfactura Reportes:
W	Tabla no modificable, de nos ser por intrucciones del reponsable del sistema

#### Relaciones

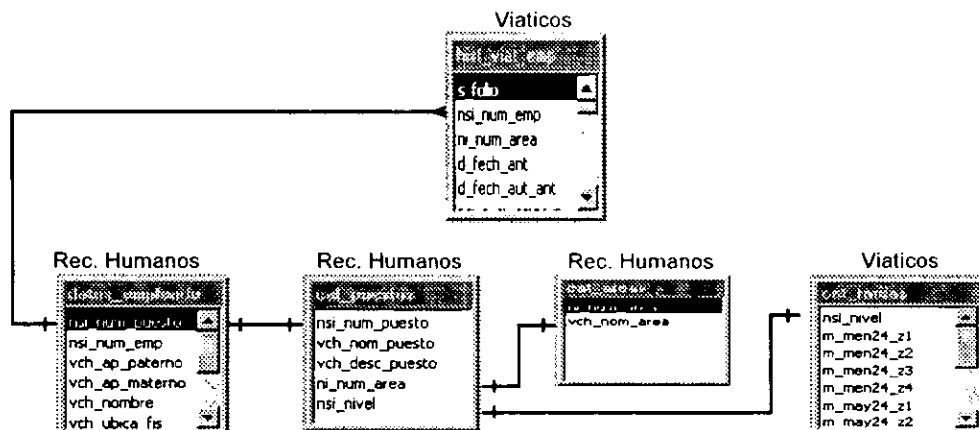
Tabla	Tipo
Cat_puestos	Uno a Varios



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico



Base de Datos: viatdesa

Tabla : Histórico de boletos de avión

Nombre: hist\_bol\_avion\_emp

Descripción: Contiene todos los cupones del boleto de avión del fólío/solicitud de viáticos que solicito boleto de avión (no se deben borrar)

CAMPO	TIPO	NULO	DESCRIPCIÓN
s_folio	integer	No	Folio de viáticos
nsi_cupon	integer	No	Núm. de cupón del avión
m_importe	money(10,2)	Si	Importe de cupón del avión
nsi_iva	smallint	Si	Iva del cupón del avión
m_tua	money(10,2)	Si	Tua del cupón del avión
m_total	money(10,2)	Si	Total del cupón del avión

Índices de la Tabla

Nombre	Campos	Origen	Primario	Foraneo

Archivos del sistema que la utilizan

Tipo de Acceso	Archivo
R	Formas: frmsolicitud, frmcompro, frmcap_cupon Reportes:
W	Formas: frmcap_cupon:



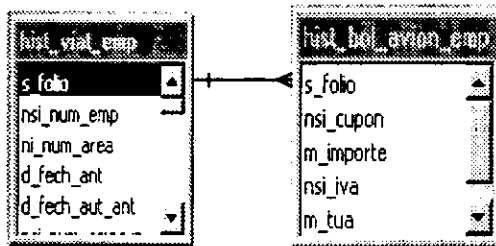
## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

#### Relaciones

Tabla	Tipo
Hist_viat_emp	Uno a Varios



Base de Datos: viatdesa

Tabla : Histórico de comprobaciones del empleado

Nombre: hist\_comp\_emp

Descripción: Contiene día a día, el lugar y la cuota autorizada de la comprobación de una comisión (esta información se borrara cuando se actualice la información)

CAMPO	TIPO	NULO	DESCRIPCIÓN
<b>s_folio</b>	integer	Si	Folio de viáticos
<b>d_fec_afecta</b>	date	Si	Fecha del destino
<b>nsi_rnum_dest</b>	smallint	Si	Número real del destino
<b>vch_rdestino</b>	varchar(40,0)	Si	Nombre real del destino
<b>m_rcuota_dia</b>	money(9,2)	Si	Cuota real diaria

#### Índices de la Tabla

Nombre	Campos	Origen	Primario	Foraneo
ix105_1	s_folio	Viáticos	XX	
ix105_1	D_fec_afecta	Viáticos	XX	
106_96	Nsi_rnum_dest	Viáticos		XX

#### Archivos del sistema que la utilizan

Tipo de Acceso	Archivo
R	Formas: frmcompro Reportes: rep001, rep002, rep007
W	Formas: frmcompro



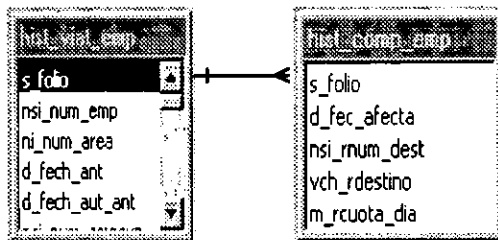
## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

#### Relaciones

Tabla	Tipo
Hist_viat_emp	Uno a Varios



Base de Datos: viatdesa

Tabla : Histórico de destinos del empleado

Nombre: hist\_dest\_emp

Descripción: Contiene día a día, el lugar y la cuota autorizada de la solicitud de una comisión (esta información se borra cuando se actualice la información)

CAMPO	TIPO	NULO	DESCRIPCIÓN
s_folio	integer	No	Folio de viáticos
d_pfecha	date	No	Fecha del destino
nsi_pnum_dest	smallint	No	Número propuesto de destino
vch_pdestino	varchar(40,0)	Si	Nombre propuesto del destino
m_pcuota_dia	money(7,2)	No	Cuota propuesta diaria

#### Índices de la Tabla

Nombre	Campos	Origen	Primario	Foraneo
ix106_1	s_folio	Viáticos	XX	
ix106_2	D_pfecha	Viáticos	XX	
103_95	Nsi_pnum_dest	Viáticos		XX



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

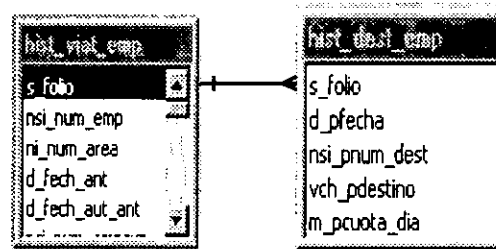
#### Ejemplo práctico

Archivos del sistema que la utilizan

Tipo de Acceso	Archivo
R	Formas: frmsolicitud, frmdestinos, frmfactura Reportes: rep001, rep002, rep007
W	Formas: frmdestinos, frmfactura

Relaciones

Tabla	Tipo
Hist_viat_emp	Uno a Varios







## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

Base de Datos: viatdesa

Tabla : Histórico de viáticos del empleado (Tabla Principal)

Nombre: hist\_viat\_emp

Descripción: Contiene Todas las solicitudes de viáticos y su seguimiento correspondiente (esta información se borrara cuando se actualice la información)

CAMPO	TIPO	NULO	DESCRIPCIÓN
s_folio	smallint	No	Folio de viáticos (llave principal)
nsi_num_emp	smallint	No	Número de empleado
ni_num_area	integer	Si	Número de área
d_fech_ant	date	No	Fecha de solicitud del anticipo
d_fech_aut_ant	date	Si	Fecha de autorización del anticipo
nsi_num_empsup	smallint	Si	Número de empleado superior que acompaña
nsi_num_empaut	smallint	No	Número de empleado que autoriza
d_pfecha_sal	date	No	Fecha propuesta de salida
d_pfecha_lle	date	No	Fecha propuesta de llegada
ch_phora_sal	char(5)	No	Hora propuesta de salida
ch_phora_lle	char(5)	No	Hora propuesta de llegada
... más campos . ...			
d_fec_cancela	date	Si	Fecha de cancelación de la solicitud de Viáticos
vch_tes_foldep	varchar(15, 1)	Si	Número de folio de deposito de tesorería
vch_tes_numpol	varchar(15, 1)	Si	Número de póliza de deposito de tesorería
d_tes_fechdep	date	Si	Fecha de deposito de tesorería
m_tes_imp	money(10,2 )	Si	Importe del deposito de tesorería

Índices de la Tabla

Nombre	Campos	Origen	Primario	Foraneo
Ix151_1	S_folio	Viáticos	XX	



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

Archivos del sistema que la utilizan

<b>Tipo de Acceso</b>	<b>Archivo</b>
R	Formas: frmsolicitud, frmdestinos, frmgastos, frmcompro, frmfactura, frmreportes, frmcancela, frmseguimiento, frmpant_aut_sol, frmpant_aut_comp, frmobjetivos, frmno_tesor, frmcaptura_so, frmsolicitud, frmcap_sald_favor, frmcap_sal_contra, frmautoriza, frmsof_ba, frmcap_cupon, frmsaldos, frmcap_sal_contra, frmcap_salvariable, frmcyrm Reportes: rep001, rep002, rep003, rep004, rep005, rep006, rep007
W	Formas: frmsolicitud, frmcompro, frmcancela, frmpant_aut_sol, frmpant_aut_comp, frmobjetivos, frmcaptura_so, frmcap_sald_favor, frmcap_sal_contra, frmautoriza, frmsof_ba, frmsaldos, frmcap_sal_contra, frmcap_salvariable

Relaciones

<b>Tabla</b>	<b>Tipo</b>
Hist_dest_emp	Uno a Varios
Hist_comp_emp	Uno a varios
Hist_bol_avion_emp	Uno a Varios
... más relaciones ...	
Rep_paso_recmat	Uno a Varios
Rep_paso_tesor	Uno a Varios
Rep_pasoobj	Uno a Varios
Rep_persona	Uno a Varios
Datos_empleado	Uno a Varios

**Técnicas utilizadas**

- Diagramas de estructuras de datos
- Reglas de conversión de un modelo lógico a físico



#### 3.5.2.1.3 Definición del Entorno Tecnológico del Sistema

Al final de esta actividad se habrán definido el entorno de funcionamiento del sistema que incluye:

- La organización del área de usuario final. Se deberá definir a que personas del área usuaria, tendrán acceso al sistema, así como los permisos a cada uno de los módulos existentes, esto es con el fin de deslindar responsabilidades.
- La infraestructura necesaria. Para que el sistema no tenga problemas de operación las características mínimas que deberán tener los equipos son:

Conexión al servidor Central (red)

Espacio en disco de 33 Mb (25 Mb ODBC y 8 Mb para los ejecutables del sistema Viáticos)

Windows 95

Estar dado de alta como usuario en la red

- La configuración de la red de comunicaciones (si es necesaria). Para garantizar el funcionamiento del sistema se deberá instalar en cada uno de los equipos donde se instalara el (los) módulos del Sistema de Viáticos el ODBC de conexión y realizar las siguientes conexiones:

Nombre de base de Datos	Nombre de la Conexión	Modulo que lo necesita	Tipo de Conexión de ODBC
Viatdesa	Viáticos	Todos	Access
Rechum	Crudo	Todos	Access
Gestion	Gestion	Todos	Access
Con0012000	Con0012000v	Contabilidad	Access

- Los procedimientos de operación de las unidades de tratamiento (programas) del sistema son los siguientes, se muestra el código y ambiente de programación para Clipper y Visual Basic. Como se ha realizado anteriormente no se está poniendo en su totalidad las tablas, los reportes y en este caso el código por su gran cantidad de instrucciones.





## MIGRACIÓN DE SOFTWARE

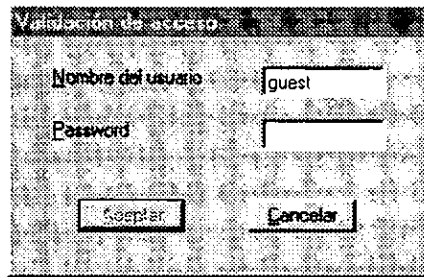
### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

- Los procedimientos de seguridad y control.

#### Control de Acceso

La parte inicial del programa esta compuesta de un Nombre de usuario y *password* (Contraseña) para permitarnos entrar y hacer uso del *Sistema de Viáticos* por cada empleado. esto se hace por razones de seguridad.



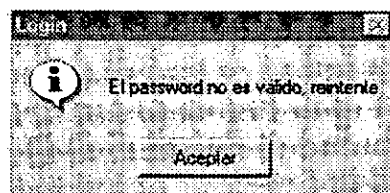
En cualquier momento podemos decidir no entrar al sistema oprimiendo el botón de comando <<CANCELAR>> o utilizando la combinación de teclas <<ALT+C>>.

El botón de comando <<ACEPTAR>> continuará inhabilitado hasta el momento en que se introduzca el *password* del usuario en la caja de texto correspondiente.

Una vez teclado el *password* podrá acceder al botón de comando <<ACEPTAR>> por medio de:

- Con el puntero del *mouse* haciendo un *clic* en el objeto
- Utilizando la combinación de teclas <<ALT+A>>
- Pulsando la tecla <<ENTER>> sobre el objeto

En caso de que el *password* introducido sea incorrecto, utilizará una caja de mensaje sugiriendo ingresar de nuevo la contraseña:



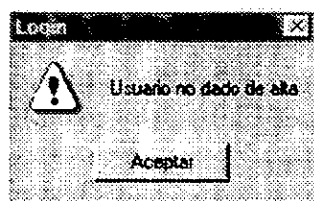


## MIGRACIÓN DE SOFTWARE

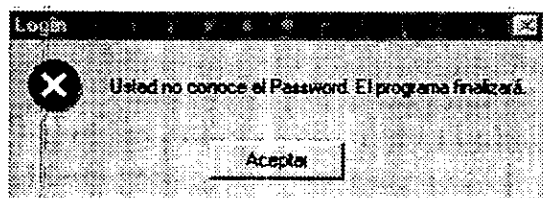
### Capítulo III "Metodología para la migración"

#### Ejemplo práctico

Puede suceder que se requiera cambiar el *nombre de usuario* que aparece por defecto en su respectiva caja de texto; el mismo con el cual inicio la sesión de red en Windows. Si el nombre de usuario no resulta ser válido o no pertenece al grupo de usuarios del sistema, aparecerá la siguiente caja de mensaje:



Si usted alcanza tres intentos al introducir el *password*, y éste no es correcto, el sistema por seguridad se cerrará:



#### Técnicas utilizadas

Para la descripción de las organizaciones de las áreas de usuario y explotación, se puede utilizar alguna técnica de representación gráfica (organigramas) que ayude a comprender mejor estas estructuras.

---

#### 3.5.2.1.4 Diseño del Plan de Pruebas del Sistema

Al finalizar esta actividad se habrá obtenido el plan de pruebas que será seguido durante la posterior fase de construcción del sistema. Este plan incluye la identificación de los responsables de las pruebas, los criterios de

---



determinación y finalización de los casos de prueba y una descripción del entorno técnico necesario.

**Técnicas utilizadas**

Para el diseño de los casos de prueba y de los datos de prueba (juegos de ensayo) se pueden utilizar técnicas habituales de prueba de "caja negra" (participaciones de equivalencia, análisis de valores límite, valores típicos de error, valores imposibles) que ignoran la estructura interna de los componentes software y se centran en su funcionalidad externa.

---

**3.5.2.1.5 Diseño del plan de construcción e implantación**

Al final de esta etapa, se obtiene el plan de construcción e implantación del Sistema.

**Técnicas utilizadas**

Se pueden utilizar técnicas de estimación empírica para determinar los tiempos y costos de construcción del sistema.

---

**Participación de Usuarios.**

La siguiente matriz muestra la participación de los usuarios en las diferentes actividades de esta fase:

	3.1	3.2	3.3	3.4	3.5
Comité de Informática					X
Jefe del Proyecto		Y	X	X	X
Analista		Y	X	X	
Programador					
Gerente del Área Involucrada			Y		
Usuario Final			X		



### 3.5.3 Fase de Construcción

La tercera fase del ciclo de vida es la Construcción del Sistema, donde se llevará cabo la elaboración y prueba del Sistema

#### OBJETIVOS DE LA CONSTRUCCION.

El objetivo fundamental de esta fase es la obtención del sistema completamente construido y probado, listo para ser implantado en la organización. También durante esta fase se desarrollará el conjunto de instrucciones y se llevará a cabo la formación necesaria que permitirá, tanto al personal del área como al usuario final la utilización óptima del sistema.

La fase de construcción del sistema se puede estructurar en las siguientes etapas:

- 3.5.3.1.1 Construcción de las Bases de Datos o Ficheros
  - 3.5.3.1.2 Construcción de las Unidades de Tratamiento y Pruebas Unitarias
  - 3.5.3.1.3 Realización de Pruebas de Integración y Globales
  - 3.5.3.1.4 Elaboración de los Procedimientos de Explotación y Formación
  - 3.5.3.1.5 Elaboración de los Procedimientos de Usuario y Formación
- 

#### 3.5.3.1.1 Construcción de las Bases de Datos o Ficheros

En esta etapa se obtienen las Bases de Datos Físicas (o ficheros convencionales) necesarias para realizar la construcción y las pruebas (con una copia para construcción y otra para pruebas), junto con la documentación correspondiente (reglas de integridad, técnicas de compresión utilizada, aspectos de almacenamiento como tamaños y tipos de registros, etc.) que incluirá también la descripción de cualquier incidencia ocurrida durante el proceso de creación.

#### Técnicas Utilizadas

La construcción se realizará aplicando las técnicas de construcción particulares del SGBD utilizado (Access)

---





### 3.5.3.1.2 Construcción de las Unidades de Tratamiento y Pruebas Unitarias

Productos a obtener

Una vez finalizada esta etapa, se habrán generado los siguientes productos:

- Revisiones de las especificaciones de las unidades de tratamiento (por ejemplo, mensajes añadidos, modificación de las pantallas e informes, etc.).
- Código fuente, sin errores, de cada unidad de tratamiento, perfectamente estructurado y documentado mediante comentarios y nombres representativos para los objetos utilizados por la misma (variables, subprogramas).
- Código ejecutable, sin errores, de cada unidad de tratamiento.
- Informes de las pruebas unitarias, en los que se detallarán :
  - Los datos de ensayo de la prueba.
  - Los resultados esperados y obtenidos.
  - Fecha y hora de realización.
  - Personal participante en la preparación y ejecución.
  - Incidencias ocurridas y acciones que se llevarán a cabo.
  - Causas de los fallos detectados.

#### Técnicas Utilizadas

Para la realización de cada unidad de tratamiento se utilizarán técnicas de programación estructurada, que deberán ser soportadas por el lenguaje de programación, el cual deberá ofrecer la posibilidad de utilizar estructuras de control y de datos, y de permitir la división modular de programas y subprogramas.

Si la complejidad de alguna unidad de tratamiento así lo requiere, se utilizará la técnica de diseño estructurado denominada "Diagrama de Estructura Modular".

En cuanto a las pruebas unitarias, se aplicarán tanto técnicas de caja negra, para validar el funcionamiento de las unidades de tratamiento, como de caja blanca, para examinar su estructura interna.



#### 3.5.3.1.3 Realización de Pruebas de Integración y Globales

El principal objetivo de esta etapa es la obtención del sistema (o aplicación informática) definitivo, libre de errores y ajustado su funcionamiento a los requisitos establecidos.

Además del sistema correcto, se deben obtener los siguientes documentos sobre los resultados de las pruebas:

- o Informe de las pruebas de integración, que incluirá los problemas de integración detectados y su corrección.
- o Informe de las pruebas globales, con los problemas detectados y corregidos, así como la notificación de problemas que no han podido ser resueltos.

Otros productos que se obtiene como consecuencia de las pruebas son los documentos generados en las fases anteriores de análisis y diseño afectados por alguna corrección debido a los problemas detectados. En este sentido, además de los errores de codificación que pueden detectarse en ambos tipos de pruebas y que sólo afectarán a los listados de los programas fuente, los errores detectados en integración suelen afectar al diseño técnico del sistema ("Documento de Diseño Técnico"), mientras que los de las pruebas globales, los más graves, afectarán al análisis ("Documento de Requisitos" o "Documento de Especificación Funcional") y en consecuencia, también al diseño, al poder precisarse una nueva redefinición de las especificaciones, al detectarse que la funcionalidad no se ajusta a los requisitos de los usuarios o que se especificó incorrectamente la solución.

#### Técnicas Utilizadas

Para la realización de las pruebas de integración y globales, se utilizarán técnicas de "prueba de caja negra".



#### 3.5.3.1.4 Elaboración de los Procedimientos de Explotación y Formación

El producto que se obtiene en esta etapa es el manual de explotación del sistema, con los siguientes contenidos:

- Descripción general del conjunto de actividades de explotación posibles: arranque diario, procesos periódicos, procesos bajo petición, procesos de mantenimiento, copias de respaldo y manipulación de errores.
- Descripción de cada una de las cadenas de tratamiento correspondientes a las actividades anteriores, mediante un diagrama, un listado, la descripción de cada paso y las unidades de tratamiento incluidas en la cadena, el espacio en disco necesario, y el tiempo medio de ejecución.
- Instrucciones para la ejecución de cada cadena, indicando las condiciones de ejecución, las entradas y salidas, los recursos que utiliza (discos, cintas, papel, etc.).
- Descripción de las bases de datos y/o ficheros convencionales.
- Instrucciones para la ejecución de cadenas relacionadas con la gestión de las bases de datos o ficheros (copias de seguridad, reorganización,...).
- Anexos con los gráficos, listados, pantallas, informes, etc. que se consideren necesarios.

#### Técnicas Utilizadas

La estructura del manual de explotación (así como las técnicas utilizadas en los diferentes apartados, por ejemplo de diagramación) responderá a los estándares existentes para este rubro. La formación del personal de explotación se realizará a través de sesiones en las que se discutirá sobre el manual, del que dispondrán cada uno de los asistentes de una copia. Se utilizarán los recursos didácticos precisos (pizarrón, transparencias, computadora, etc.,...) para ayudar comprender todos los procesos de explotación del sistema.

---



#### 3.5.3.1.5 Elaboración de los Procedimientos de Usuario y Formación

Durante esta etapa se obtendrá el manual de usuario del sistema, cuyos contenidos básicos serán los siguientes:

- Descripción general del sistema, con entradas y salidas de información.
- Convenciones a tener en cuenta en los diferentes menús de la aplicación: teclas de función, uso de mandatos directos, etc.
- Identificación de las funciones generales del sistema o subsistemas del mismo.
- Para cada subsistema, los diálogos que se establecerán con el usuario, con la representación gráfica de los diferentes menús que pueden aparecer y la explicación de la función de cada opción.
- Si existen cadenas de procesamiento por lotes, descripción de cadenas.
- Formatos de reportes generados por la aplicación.
- Instrucciones generales de uso de los recursos técnicos.

#### Técnicas Utilizadas

Se aplicarán las normas de estilo estándar establecidas para la elaboración de manuales de usuario, que especifiquen su estructura (títulos, secciones, márgenes, numeración de páginas,...) y establezcan recomendaciones sobre contenidos (texto, gráficos con los menús de opciones junto con la descripción de cada opción, ejemplos de reportes, ejemplos de pantallas de entrada o salida de datos, glosario de términos, etc.). También es importante tener en cuenta las recomendaciones de control de versiones en manuales y de seguridad, confidencialidad, archivo y circulación de documentos.

En la capacitación de usuarios se utilizarán los recursos (medios audiovisuales, transparencias,...) necesarios para atraer la atención de los asistentes a las sesiones y ayudar al entendimiento. Serán sesiones de diálogo, en las que los asistentes podrán participar de forma activa, aceptando sugerencias y preguntas de los mismos. Como ejemplo, un guión para los cursos podría ser:

- Introducción. Exposición breve de los objetivos del curso.
- Índice de los temas a tratar
- Visión general de los procesos y funciones del sistema.



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo III "Metodología para la migración"**

#### **Ejemplo práctico**

- Descripción de ventajas y beneficios aportados por el sistema.
- Descripción secuencial y detallada de cada actividad del sistema.
- Ayudas disponibles en el sistema.
- Ejercicios y sesiones prácticas.
- Conclusiones

---

### **Documentación Técnica de Programación**

Se trata de la documentación estrictamente de construcción, ya que contiene el código fuente perfectamente documentado de los programas que constituyen el sistema junto con la especificación de la estructura modular de los que por su complejidad así lo requieren.

<b>DOCUMENTACION TECNICA DE PROGRAMACIÓN</b>
--

- |  |
|--|
| <ol style="list-style-type: none"><li>1. Estructura modular del código de cada Unidad de Tratamiento.</li><li>2. Listado con el código fuente.</li></ol> |
|--|



### 3.5.4 Fase de Implantación

La última fase del ciclo de vida es la Implantación del Sistema donde se llevará a cabo la puesta en servicio del Sistema construido.

#### OBJETIVOS DE LA IMPLANTACION.

El objetivo de esta fase es la puesta en servicio del sistema construido y conseguir su aceptación final por parte de los usuarios del mismo, para lo cual se tratará de hacer ver a estos, mediante una demostración formal, que el sistema cumple todos los objetivos y requisitos para los que fue desarrollado y que fueron detallados en la fase de análisis (Análisis de Requisitos del Sistema).

La fase de implantación se puede estructurar en las siguientes etapas:

- 3.5.4.1.1 Revisión del Plan de Implantación del Sistema
- 3.5.4.1.2 Instalación del Sistema
- 3.5.4.1.3 Realización de las Pruebas de Aceptación
- 3.5.4.1.4 Transición al nuevo Sistema

Participantes en la fase de implantación	5.1	5.2	5.3	5.4	5.5
Comité de Informática	X	X	X		X
Jefe del Proyecto	X	X	X		X
Analista		X	X	X	X
Programador			X	O	X
Gerente del Área Involucrada	X	X	X	X	X
Usuarios finales			X	X	X

En esta fase, se obtiene el contenido final del Documento de Pruebas del Sistema.

#### 3.5.4.1.1 Revisión del Plan de Implantación del Sistema

Al final de esta etapa se obtiene el Plan de Implantación definitivo, que incluirá la planificación de:



## **MIGRACIÓN DE SOFTWARE**

### **Capítulo III "Metodología para la migración"**

#### **Ejemplo práctico**

- Los procedimientos de instalación del sistema en el entorno de explotación, incluyendo la instalación de productos estándar adquiridos externamente. Se establecerán los planes de trabajo y horarios.
- Las pruebas de aceptación, donde se determinen las características de las pruebas (personas afectadas, juego de datos de ensayo, etc.) y la posibilidad de crear un entorno especial para realizarlas.
- La transición de sistema antiguo al nuevo, estableciendo:
  - \* el procedimiento que se aplicará: inmediato a través de un proyecto piloto o mediante un período provisional de ejecución en paralelo, directo o modular.
  - \* planes de contingencia, estableciendo el mecanismo de retorno al sistema antiguo en el caso de problemas graves imprevistos.
  - \* los procedimientos de conversión de datos, estimando la necesidad de personal adicional para tal labor (grabadores de datos), los criterios de finalización de la conversión, el mantenimiento de los ficheros de conversión hasta el arranque definitivo del sistema.
- El mantenimiento del sistema, estableciendo el mecanismo de seguimiento de la ejecución del sistema, que consistirá básicamente en el establecimiento de una línea abierta de consultas de usuario.
- Técnicas Utilizadas:
- Se puede aplicar alguna técnica, si se estima oportuno, de planificación de sistemas, como el PERT.

---

#### **3.5.4.1.2 Instalación del Sistema**

Se instalarán los componentes informáticos del sistema (software) necesarios para la explotación y facilidades necesarias para los procesos.

##### **Técnicas utilizadas:**

Se aplicarán las convenciones o normas establecidas por la organización referidas al procedimiento de instalación.

---

#### **3.5.4.1.3 Realización de las Pruebas de Aceptación**

Al final de esta etapa se obtendrá un informe de las pruebas de aceptación, que será incluido en el Documento de Pruebas del Sistema, en el que se reflejarán aspectos tales como la fecha de ejecución de la prueba y personal



implicado, los juegos de datos utilizados, los resultados esperados y obtenidos, las incidencias detectadas, sus causas y acciones correctivas. Además del anterior documento, si fuera necesario, se revisarán y modificarán todos los generados en las fases anteriores para corregir las posibles desviaciones que sobre los requisitos de usuario han sido detectadas. Finalmente, también se deberá obtener un documento de aceptación del sistema por parte del usuario, con la firma de éste, al que se le entregará formalmente entonces la documentación definitiva del proyecto (manuales del usuario, de explotación, documento de análisis de requisitos y documento de especificación funcional). Si como se ha considerado a lo largo del texto, el desarrollo del sistema se hizo "a la medida" de las necesidades del usuario, y no como la adaptación de una aplicación estándar ofrecida por la organización de desarrollo, entonces se entregarán también los ficheros con el código fuente de los programas y toda la documentación de diseño y pruebas, que permitan su futuro mantenimiento por parte del departamento de explotación.

#### **Técnicas utilizadas:**

Se podrán aplicar las mismas técnicas de "prueba de caja negra" que se utilizaron las pruebas globales del sistema.

Para la elaboración del informe de las pruebas, se utilizarán las reglas y formatos establecidas por la organización.

---

#### **3.5.4.1.4 Transición al nuevo Sistema**

Al finalizar la transición, una vez eliminado el sistema antiguo, se dispondrá del nuevo sistema en explotación, que puede ya ser utilizado regularmente, con lo que deberá obtener la firma final que asegure que el personal usuario, de explotación y la dirección de usuario consideran correctamente implantado el nuevo sistema.

#### **Técnicas utilizadas:**

Podría considerarse como tal la estrategia de transición elegida.

---





### 3.6 DOCUMENTACION

La documentación que se recomienda que un sistema debe tener, es la que a continuación se relaciona, esto no impide que se pueda enriquecer de acuerdo al criterio o normas de cada organización.

1. Diagrama de Contexto y Diagrama de Flujo de Datos.
2. Documento de Análisis y Documento de Diseño.
3. Manual de usuario y carta de aprobación.
4. Manual técnico con los siguientes apartados:
  - Carta de aprobación.
  - Documentación de formularios incluyendo para cada uno:
    - Nombre de las conexiones que utiliza.
    - Nombre de las tablas y tipo de acceso.
    - Consideraciones de operación (observaciones y restricciones).
  - Diagrama Entidad-Relación.
  - Documentación de las Bases de Datos  
incluyendo por cada tabla.
    - Nombre, tipo, longitud y descripción de los campos.
    - Nombre de los índices y campos que lo conforman.
    - Nombre de los formularios que utilizan y tipo de acceso
    - Nombre y tipo de relaciones con otras tablas.
    - Esquema de relación (donde el Diagrama de E-R no es claro)
    - En caso de utilizar solo una parte de las tablas de otra base de datos, se debe indicar el tipo de acceso y los campos requeridos para los procesos, además en los formularios donde son requeridas.
  - Documentación de los reportes utilizados incluyendo por cada uno.
    - Nombre y descripción del archivo
    - Nombre y descripción de las formulas
    - Información técnica de la herramienta con que se construyó el reporte.
- Todos los documentos deberán entregarse impresos y en formato Word.
- Las carátulas deberán estar firmadas por el Responsable del proyecto, Gerente de Sistemas y en caso de ser necesario por el Usuario responsable.



## MIGRACIÓN DE SOFTWARE

### Capítulo III "Metodología para la migración"

#### Documentación

#### Archivos del proyecto:

- Formularios y Módulos. estandarizados en cuanto a:
  - Nombre de los objetos.
  - Nombre de las variables. constantes, procedimientos y funciones.
  - Comentarios de cabecera y definición de procedimientos.
  - Manejo de errores.
  - Mensajes de sistema
- Archivos de ayuda.
  - hlp, hpj, rtf y doc
- Reportes.
- Archivos adicionales (imágenes, archivos de recursos, archivos de texto, etc.).

## CONCLUSIONES.

La expectativa final de haber realizado este proyecto, es que el lector tenga un panorama general para realizar una migración bajo una metodología y también que se de cuenta de la importancia de estandarizar y documentar un sistemas de computo. El objetivo final de la realización de este trabajo, fue el de tratar de compendiar el mayor número de técnicas y herramientas útiles que sirven de apoyo en la realización de una migración.

La importancia de realizar una migración es que se pueden reutilizar algunas de las partes realizadas con anterioridad, mejorando la calidad de software, no solo en cuanto al ambiente donde se ha de trabajar, sino en las herramientas y documentación final. Gran parte del éxito de realizar una migración, depende de realizar un buen análisis del sistema anterior, ya que con ello nos permitirá entender perfectamente el sistema y así poder corregir los errores existentes.

Es importante mencionar que existen muchas otras herramientas de modelado con las que el desarrollador se puede apoyar en cada una de las fases del desarrollo y que cada Organismo defina y decida las que ha de utilizar, finalmente es dejarlo lo mejor documentado posible para que este sea un sistema robusto y con el menor numero de problemas, ya sea para la fase de mantenimiento, migración o reingeniería.

El analista de sistemas deberá construir los modelos con las herramientas, con la finalidad de enfatizar en ciertas actividades críticas del sistema que le permitan tener una mejor comunicación con el usuario, generalmente estas herramientas deberán ser graficas.

Una de las conclusiones mas importantes, es la tendencia que se tiene a migrar a nuevas y mas amigables plataformas de Sistemas Operativos, esto es; se esta desplazando la utilización del sistema operativo "DOS" a ambientes gráficos de "Windows". En este ejemplo se muestra perfectamente las características de cada uno, los

ambientes gráficos necesitan equipos mas potentes tanto en capacidad de memoria RAM, así como mayor capacidad de almacenamiento en sus discos.

Los objetivos planteados inicialmente, como algoritmos, metodologías, herramientas de modelado, etc. se cumplieron con el desarrollo en cada uno de las fases de los diferentes capítulos.

Puede ser posible que con el paso de algunos años, este proyecto, solo sirva cómo referencia, debido al gran avance tecnológico existente, tanto en hardware como en software y debido a esto se innoven nuevas técnicas para la realización de una migración.

Finalmente, espero que la información aquí contenida sea de utilidad, no sólo para una persona que quiera realizar una migración, sino de apoyo formativo a todas las personas que tengan una carrera afín con la computación.

## **BIBLIOGRAFÍA:**

1. Echenique José Antonio  
**Auditoria en Informática**  
Ed. Mc. Graw Hill.
2. S. Pressman Roger  
**Ingeniería de Software**  
Ed. Mc. Graw Hill.
3. Fairley Richard  
**Ingeniería de Software**  
Ed. Mc. Graw Hill.
4. Kendall & Kendall  
**Análisis y Diseño de Sistemas**  
Ed. Prentice Hall
5. W. Hartman, H. Mattites & A. Proeme  
**Manual de los Sistemas de Información**  
Ed. Paraninfo
6. Yourdon Edward  
**Análisis Estructurado Moderno**  
Ed. Prentice may
- 7- G. Piattini Mario, Calvo Manzano José  
**Análisis y diseño detallado de aplicaciones informáticas de gestión**  
Ed. Ra-ma



MIGRACIÓN DE SOFTWARE

Anexos

Programas fuertes (Clipper)

# ANEXOS



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

#### PROGRAMAS FUENTE DE LA VESIÓN EN CLIPPER

```
//
// Sistema: VIATICOS, Proceso de viaticos
// Autor: Ing. Luis Hernández Pérez
//-----
*--- Definición de Medio Ambiente -----
#include "inkey.ch"
#define cBACKCOLOR 3 // blue on black
#define mBACKCOLOR 7 // white on black
****
// background pattern for 1.50 stuff
STATIC pattern := "*****+*****+*****+*****+*****"
IF !ISCOLOR()
  BCOLOR=mBACKCOLOR
ELSE
  BCOLOR=cBACKCOLOR
ENDIF
cls( bcolor, pattern)
SAVE SCREE TO BEFORE
SETHANDLES(50)
IF ISCOLOR()
  COLOR3='W/N,N/W'
  COLOR2='B/BG,BG/B'
  COLOR1='B/W,W/B'
  COLOR4='W+/B+,N/W'
  COLOR5='N/W,W+/B+'
ELSE
  COLOR3='W/N,N/W'
  COLOR2='N/W,W/N'
  COLOR1='N/W,W/N'
  COLOR4='W/N,N/W,...N/W'
  COLOR5='N/W,W/N,...W/N'
ENDIF
SET EXCLUSIVE OFF
SET PROCEDURE TO LOCKS
SET STATUS OFF
SET SAFETY OFF
SET DATE BRIT
SET DELETED ON
SET TALK OFF
SET CONFIRM OFF
set console off
set score off
set wrap on
*----- Pantalla principal -----
store 1 to aa,bb,cc,dd
store 0 to nw
public p_Ayuda := '00000', p_Date, p_Barra
p_Date := Date()
sigue = .t.
set color to b/w
* win(0,0,2,79,1,0) && abre la pantalla del centro hacia afuera
@ 0,0 to 2,79 double
set color to b/w
@ 0,21 say ' Nombre de la Institución '
@ 1,1 say 'Gerencia de Sistemas, VIATICOS-1.0'
@ 1,48 say 'Area responsable'
@ 3,0 to 24,79 double
//----- About -----
save screen to pantalla
@ 3,0 to 24,79 double
@ 9,24 to 18,58 double
cent('Sistema de Control de Viaticos',10)
cent('Version 1.0',11)
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Clipper)

```
cent('Nombre de la Institución',13)
cent('Gerencia de Sistemas',14)
cent('Ing. Luis Herenández Pérez',15)
set color to +w/b
cent(' OK ',17)
inkey(0)
set color to w/b
@ 9,30 clear to 16,45
restore screen from pantalla
//----- Solicita la fecha de trabajo-----
SET CONFIRM ON
win(20,19,22,61,1,1)
@ 20,19 TO 22,60
@ 21,20 say 'Fecha de trabajo (dia/mes/año):' get p_Date
read
cwin(20,19,22,61)
set color to b/w
@ 24,69 say '~'+DTC(p_Date)
set color to w/b
* p_Barra = saveScreen(24,0,24,79)
SET CONFIRM OFF
//----- abre archivos-----
lin22 = savescreen(22,0,22,79)
cent('Favor de esperar, apertura de archivos en proceso ...',22)
AbreArchivos()
RelaArchivos()
restscreen (22,0,22,79,lin22)
* do borr23
.....
* Funcion que Activa el Menu
*
store 0 to vop,vop3,vop4,vop5
do while aa=1
@ 8,32 say "MENU PRINCIPAL"
@ 9,32 TO 17,45
@ 10,33 PROMPT "ANTICIPOS "
@ 11,33 PROMPT "COMPROBACION"
@ 12,33 PROMPT "REPORTES "
@ 13,33 PROMPT "CATALOGOS "
@ 14,33 PROMPT "UTILERIAS "
@ 15,33 PROMPT "ESTADISTICAS"
@ 16,33 PROMPT "SALIR "
menu to vop
do case
case vop = 1
do via000
case vop = 2
do via200
case vop = 3
Menu31()
case vop = 4
Menu41()
case vop=5
Menu51()
case vop = 6
do via600
case vop = 7
clear
exit
endcase
if vop>2.and.vop<6
cvop=str(vop,1)
do case
case vop3 > 0
cvop3=str(vop3,1)
proceso='via'+cvop+'0'+cvop3
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
case vop4 > 0
  cvop4=str(vop4,1)
  proceso='via'+cvop+'0'+cvop4
case vop5 > 0
  cvop5=str(vop5,1)
  proceso='via'+cvop+'0'+cvop5
endcase
endif
sum=vop3+vop4+vop5
if sum=0
  restore screen from pantalla
  loop
endif
do &proceso
store 0 to vop,vop3,vop4,vop5
restore screen from pantalla
enddo
aa=1
close all
return
```

```
//-----
// Menu31 - Función que activa reportes
```

```
//-----
function Menu31
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
save screen to pantalla
do while bb=1
  @ 13,52 say 'REPORTES'
  @ 14,45 to 21,66
  @ 15,46 prompt 'Autorización'
  @ 16,46 prompt 'Solic. Trans. Area'
  @ 17,46 prompt 'Comprobación'
  @ 18,46 prompt 'Recibo de Devolución'
  @ 19,46 prompt 'Historial'
  @ 20,46 prompt 'Listado General'
  menu to vop3
  bb=2
enddo
bb=1
return vop3
```

```
//-----
// Menu41 - Función que activa catalogos
```

```
//-----
function Menu41
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
save screen to pantalla
do while cc=1
  @ 13,46 say 'CATALOGOS'
  @ 14,45 to 20,55
  @ 15,46 prompt 'Destinos'
  @ 16,46 prompt 'Tarifas'
  @ 17,46 prompt 'Plantilla'
  @ 18,46 prompt 'Puestos'
  @ 19,46 prompt 'Areas'
  menu to vop4
  cc=2
enddo
cc=1
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
return vop4
//-----
// Menu51 - Función que activa utilerías
//-----
function Menu51
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
save screen to pantalla
do while dd=1
  @ 13,47 say 'UTILERIAS'
  @ 14,45 to 17,56
  @ 15,46 prompt 'Reindexado'
  @ 16,46 prompt 'Par metros'
  menu to vop5
  dd=2
enddo
dd=1
return vop5
//-----
// Abre todos los archivos de un jalón
//-----
function AbreArchivos
sele 1
do while !net_use('viaticos','VIA',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to via001,via002,via003
sele 2
do while !net_use('destinos','DES',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to des001,des002,des003
sele 3
do while !net_use('comproba','COM',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to com001,com002
sele 4
do while !net_use('rh1','RH1',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to rh1001,rh1002
sele 5
do while !net_use('rh2','RH2',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to rh2001
sele 6
do while !net_use('rh3','RH3',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to rh3001
sele 7
do while !net_use('tarifas','TAR',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to tar001
sele 8
do while !net_use('fgcnefe','ENF',.t,.5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
set index to enf001,enf002
sele 9
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
do while !net_use('adn26',PARAM,.t.,5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
sele 10
do while !net_use('adn26des',DESC,.t.,5)
  mensa('Archivo en uso exclusivo, presione cualquier tecla ...')
enddo
retu
//-----
function RelaArchivos
  sele 8
  set order to 1
  sele 2
  set order to 2
  set rela to de_nument into ENF
  sele 3
  set order to 1
  sele 4
  set order to 1
  set rela to puesto into RH2
  set rela to numare into RH3 additive
  set rela to nivel into TAR additive
  sele 1
  set rela to numdes into DES
  set rela to numant into COM additive
  set rela to numemp into RH1 additive
return
//-----
//-----
// Program : VIA000.prg
// : altas, bajas y cambios
// Dated : July 1995
// Developed by Ing. Luis Hernández Pérez
//-----
#ifdef K_ESC
#include "InKey.ch"
#endif
#include "DbEdit.ch"
#define MAX_PANT 3
public Teclaso, l_NoPant := '1', Reg_Borrado := .f.
public Edit_Mode := ''
nom=space(30)
store 0 to nivhon,niv
set color to b/w
@ 24,1 say 'F1-Ayuda F2-Alta F3-baja F4-Pantalla Sig F5-edita Esc-Pincipar'
set color to w/b
set delete off
//-----abre archivos -----
RelaArchivos()
sele 1
set order to 2
set key K_F2 to Inserta_Registro
//-----Define Pantalla-----
store 0 to teclaso
if reccount() = 0
  cent('No se encuentran registros, presione [F2] para agregar',23)
  wait
  if reccount() = 0
    set key K_F2 to
    return
  endif
endif
* do borr23
//-----Definición de Teclasos-----
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
set key K_F1 to Reg_existe
set key K_F3 to Elimina_Registro
set key K_F4 to Next_Pant
set key K_F5 to Modo_Edicion

-----
store 1 to Saltos
SELE 1
go top
getnumant=0
do while .t.

-----Cambio de Proyecto-----
  I_NumAnt = NumAnt
  @ 4,2 SAY "Número de anticipo :".
  @ 4,23 get I_NumAnt pict '999999' vaLID I_NumAnt > 0
  read
  if LastKey() = K_ESC .or. LastKey() = K_PGDN .or. LastKey() = K_PGUP
    set key K_F3 to
    keyboard chr(LastKey())
    inkey()
  endif
  if I_NumAnt != NumAnt
    RegActual = recno()
    go top
    seek I_NumAnt
    if !found()
      mensa("Anticipo No. '+str(I_NumAnt,6)+' no existe, " desea crearlo ? (s/n) ...")
      if upper(chr(LastKey())) = 'S'
        do Inserta_Registro with I_NumAnt
      else
        goto RegActual
      loop
    endif
    set color to b/w
    @ 4,23 say I_NumAnt
    set color to w/b
  endif
endif

-----Fin cambio de perfil-----
set color to +w/b
@23,70 say if(delete(),'Borrar,' )
set color to w/b
//----- No Pantalla dependiendo de I_NoPant -----
if LastKey() != K_ESC .AND. LastKey() != K_PGDN .AND. LastKey() != K_PGUP
set key K_F3 to Elimina_Registro
do Screen&I_NoPant
if LastKey() = K_ESC
keyboard chr(K_F9)
inkey()
endif
dbcommitall()
endif

//-----
Edit_Mode = ''
teclado = lastkey()
do case
case teclado = K_ESC && Sale
exit
endcase
enddo
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
if Reg_Borrado
do via001
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
endif
set delete on
RestScreen(24,0,24,79,p_Barra)
return
```

#### \* Teclaso [F3] Elimina Registro

```
procedure Elimina_Registro
local l_CvePerfil
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
Reg_Borrado = .t.
do while !rec_lock(5)
  mensa("Registro en uso exclusivo, presione cualquier tecla .")
enddo
if delete()
  recall
else
  delete
endif
set color to +w/b
@23,70 say if(delete()),'Borrar,'
set color to w/b
set key K_F1 to Reg_Existe
set key K_F2 to Inserta_Registro
set key K_F3 to Elimina_Registro
set key K_F4 to Next_Pant
set key K_F5 to Modo_Edicion
return
```

#### \* Teclaso [F2] Inserta Registro

```
procedure Inserta_Registro
parameter SugNumAnt
local l_valido := .t.

clear gets
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
if valtype(SugNumAnt) != 'N'
  win(9,9,11,35,1,0)
  SugNumAnt := 0
  do while l_Valido
    l_Valido = .f.
    @ 4,2 say 'NEmero de anticipo : ' get SugNumAnt pict '999999' valid SugNumAnt > 0
    read
    if LastKey() = K_ESC
      Keyboard chr(K_ENTER)
    endif
    go top
    seek SugNumAnt
    if found()
      mensa("Anticipo "+str(SugNumAnt,6)+" ya existe, Esc - Salir ...")
      if LastKey() = K_ESC
        SugNumAnt = 0
        go top
        Keyboard chr(K_ENTER)
      else
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes: (Clipper)

```

        l_Valido = t.
    endif
    else
    go top
    endif
    enddo
    cwin(9,9,11,35)
endif
//----- Anexa reg.-----
if SugNumAnt > 0
do while !add_rec()
    cent('Archivo en uso exclusivo, favor de esperar ...',23)
    enddo
    repl a->NumAnt with SugNumAnt,a->fantic with p_Date
// repl a->plansal with p_Date, planlle with p_Date
    repl a->plansal with p_Date
// repl a->realsal with p_Date, reallle with p_Date
    repl a->realsal with p_date
    repl a->fcompro with p_Date
    repl a->realsalt with time(), realllet with time()
endif
//----- Fin anexa registro -----
set key K_F1 to Reg_existe
set key K_F2 to Inserta_Registro
set key K_F3 to Elimina_Registro
set key K_F4 to Next_Pant
set key K_F5 to Modo_Edicion
l_NumAnt = NumAnt
if SugNumAnt > 0
    Modo_Edicion()
endif
return
.....
* Teclaso [F4] Siguiete Pantalla
*
procedure Next_Pant
clear gets
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
@ 6,1 clear to 21,78
l_NoPant := if(l_NoPant = str(_MAX_PANT,1),'1',str(val(l_NoPant)+1,1))
set key K_F1 to Reg_existe
set key K_F2 to Inserta_Registro
set key K_F3 to Elimina_Registro
set key K_F4 to Next_Pant
set key K_F5 to Modo_Edicion
keyboard chr(K_ENTER)
Edit_Mode = ''
return
.....
* Teclaso [F5] Modo de Edición
*
procedure Modo_Edicion
clear gets
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
Edit_Mode = if(Edit_Mode='READ','/','READ')
set key K_F1 to Reg_existe

```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
set key K_F2 to Inserta_Registro
set key K_F3 to Elimina_Registro
set key K_F4 to Next_Pant
set key K_F5 to Modo_Edicion
// keyboard chr(K_ENTER)
return
-----
//
// [ F1 ] Teclas
// Procedimiento de búsqueda de registro existente
// Reg_existe
//-----
function Reg_existe
local oTbDB := TBrowseDB(11,18,16,60)
local getnumant := 0
clear gets
set key K_F1 to
set key K_F2 to
set key K_F3 to
set key K_F4 to
set key K_F5 to
win(10,17,17,61,1,0)
//-----
oTbDB:addColumn(TBColumnNew("NEmero Ant.",{||via->numant}))
oTbDB:addColumn(TBColumnNew("Nombre del Empleado",{||alltrim(rh1->nombre)+' '+alltrim(rh1->paterno)+' '+alltrim(rh1->materno)}))
*if numemp=999999
* oTbDB:addColumn(TBColumnNew("Nombre del Empleado",{||via->nomhon}))
*endif
oTbDB:colorSpec := 'w/b,w+/b,w*/b,b/w,b/w'
oTbDB:colsep := '|'
oTbDB:headsep := '-'
//oTbDB:freeze := 1
//-----Navegación-----
go top
do while LastKey() != K_ESC
while !oTbDB:stabilize()
enddo
inkey(0)
do case
case LastKey() = K_UP
oTbDB:up()
case LastKey() = K_DOWN
oTbDB:down()
case LastKey() = K_PGDN
oTbDB:pageDown()
case LastKey() = K_PGUP
oTbDB:pageUp()
case LastKey() = K_ENTER
i_NumAnt := via->numant
exit
endcase
enddo
cwin(10,17,17,61)
set key K_F1 to Reg_existe
set key K_F2 to Inserta_Registro
set key K_F3 to Elimina_Registro
set key K_F4 to Next_Pant
set key K_F5 to Modo_Edicion
retu getNumAnt
-----
//
// Pantalla 1
// Datos Generales
// Developed by Ing. Luis Hernández Pérez
//-----
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
procedure Screen1
if Edit_Mode != 'READ'
  @ 6,1 clear to 22,78
endif
@ 10, 2 TO 10,77
@ 14, 2 TO 14,77
@ 21, 2 TO 21,77
@ 4,75 SAY "1/3"
@ 5, 2 SAY "Nºmero de empleado :."
@ 5,62 SAY "Fecha :."
@ 7, 2 SAY "Comisionado ."
@ 7,57 SAY "Tel./Ext.:"
@ 8, 2 SAY "Puesto :."
@ 9, 2 SAY "Area :."
@ 11, 2 SAY "Comisionado por :."
@ 13, 2 SAY "Motivo :."
@ 15, 2 SAY "Destino :."
@ 15,40 SAY "Zona:."
@ 19, 2 SAY "Gasolina [M]agna, [N]ova, [S]in peaje :."
@ 20,36 SAY "Kilómetros:."
@ 20, 2 SAY "Costo de la gasolina:."
@ 16, 2 SAY "Salida :."
@ 17, 2 SAY "Llegada:."
@ 16,61 SAY "Dias:."
@ 12, 2 SAY "Comisionado a :."
@ 22,57 SAY "Anticipo:."
@ 22, 2 SAY "Gastos de taxi :."
@ 18,55 SAY "Viaticos :."
@ 19,55 SAY "Hospedaje :."
@ 20,55 SAY " Peaje :."
//-----Empleado-----
@ 5,23 GET numemp
@ 5,70 GET fantici
if Edit_Mode = 'READ'
  READ
  @ 7,18 clear to 7,51
  if updated().and.edit_mode='READ'
    if numemp=999999
      @ 4,36 say 'H O N O R A R I O S'
      @ 6,49 say 'NIVEL'
      @ 7,17 say 'get nom'
      @ 7,50 say 'get niv pict '9'
      read
      sele 7
      loca for nivhon=niv
      sele 1
    else
  // if updated().and.Edit_Mode='READ'
    sele 4
    seek VIA->NumEmp
    sele 5
    seek RH1->puesto
    sele 6
    seek RH1->numare
    sele 7
    seek RH1->nivel
    sele 1
  endif
endif
if numemp=999999
  replace via->nomhon with nom
  replace via->puesthon with 'H O N O R A R I O S'
  nom=space(30)
endif
endif
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
* @ 7,18 clear to 7,51
if numemp<>999999
  @ 7,18 SAY left(alltrim(rh1->nombre)+' '+alltrim(RH1->paterno)+' '+alltrim(rh1->materno),53)
  @ 7,68 SAY rh1->telex
  @ 8,18 SAY rh2->Nompues
  @ 9,18 SAY rh3->NomArea
else
  @ 7,18 say via->nomhon
  @ 8,18 say via->puesthon
endif
//-----Quien comisiona-----
@ 11,20 GET comision pict '@!'
@ 12,20 get destino pict '@!'
@ 13,20 GET motivo pict '@!'
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP or Tcla=K_F4 or Tcla=K_F2.or. Tcla=K_F5) .and. Edit_Mode = 'READ'
  READ
endif
//-----Destino-----
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP or Tcla=K_F4 or Tcla=K_F2.or. Tcla=K_F5) .and. Edit_Mode = 'READ'
  locNumEst = GetEstado(enf->ef_nument,des->de_tipo)
  if lastkey()=K_ESC
    RETURN
  ENDIF
  if LastKey() != K_ESC
    locNumDes = GetDestino(locNumEst)
    if LastKey()=K_ESC
      keyboard chr(K_F9)
      inkey()
    endif
    if locNumDes > 0
      repl via->numdes with locNumDes
      repl gViaje with gasolina * 0.05 * des->de_kms
      lZona = des->de_zona
      if des->de_tipo = 'I'
        repl CuotaUSD with tar->int_z&lZona
        repl tica with param->USD
      else
        repl CuotaD with tar->May_z&lZona
      endif
      sele 8
      seek des->de_nument
      sele 1
    else
      sele 2
      seek via->numdes
      sele 1
    endif
    else
      keyboard chr(K_F9)
      inkey()
    endif
  endif
  @ 15,12 clear to 15,39
  @ 15,12 SAY left(alltrim(DES->de_NomDes)+' '+alltrim(enf->ef_nomcor),27)
  @ 15,45 SAY iif(des->de_zona='4','IV',iif(des->de_zona='2','II',iif(des->de_zona='3','III','I')))
  if des->de_tipo = 'I'
    @ 15,53 SAY "Cuota USD ."
    @ 15,67 GET cuotaUSD pict '$99,999.99'
    @ 16,25 SAY "Tipo C mbio ."
    @ 16,39 GET tica pict '99,999.99'
  else
    @ 15,53 SAY "Cuota diaria:"
    @ 15,67 GET cuotad pict '$99,999.99'
  endif
endif
```



# MIGRACION DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ 16,11 GET plansal
@ 17,11 GET planlle
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP.or. Tcla=K_F4.or. Tcla=K_F2.or. Tcla=K_F5) .and. Edit_Mode = 'READ'
  read
  * if updated()
    ICalcdias = planlle-plansal
    repl pdias with ICalcdias+1
    repl reallie with pianlle
    if pdias = 1 .and. numemp <> 999999
      repl CuotaD with tar->Men_Z&IZona
      @ 15,67 get CuotaD pict '$99,999.99'
    endif
  * endif
endif
PeajeMNS = iif(Gasolina=0,'S','N')
@ 16,67 GET pdias pict ' 99,999.99'
@ 19,42 get PeajeMNS pict '!' valid PeajeMNS $ 'MNS'
Tcla = LastKey()
if !(Tcla=K_ESC.or.Tcla=K_PGDN.or.Tcla=K_PGUP.or.Tcla=K_F4.or.Tcla=K_F2.or.Tcla=K_F5).and.Edit_Mode = 'READ'
  read
  if PeajeMNS $ 'MN' .and. updated()
    repl gasolina with iif(PeajeMNS='M',param->magna,param->nova)
    repl gViaje with gasolina * 0.05 * des->de_kms
  endif
endif
// LHP
ICuotaD = iif(des->de_tipo = 'I',cuotaUSD*tica,cuotaD)
repl viaticos with pdias*ICuotaD*param->viaticos/100, hospedaje with (pdias-1)*ICuotaD*param->hospedaje/100
if pdias=1
  repl viaticos with ICuotaD, hospedaje with 0
endif
@ 18,67 GET viaticos pict '$99,999.99'
@ 19,67 GET hospedaje pict '$99,999.99'
@ 20,24 get gasolina
@ 20,48 SAY des->de_kms
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP.or. Tcla=K_F4.or. Tcla=K_F2.or. Tcla=K_F5) .and. Edit_Mode = 'READ'
  read
  if updated()
    repl gViaje with gasolina * 0.05 * des->de_kms
  endif
endif
@ 20,67 get gViaje pict '$99,999.99'
@ 22,19 GET gTaxi pict '$99,999.99'
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP.or. Tcla=K_F4.or. Tcla=K_F2.or. Tcla=K_F5) .and. Edit_Mode = 'READ'
  read
  repl anticipo with gViaje + viaticos + hospedaje + gTaxi
  @ 22,67 GET anticipo pict '$99,999.99'
  Tcla = LastKey()
  if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP.or. Tcla=K_F4.or. Tcla=K_F2.or. Tcla=K_F5) .and. Edit_Mode = 'READ'
    read
  endif
  clear gets
  if LastKey() = K_ESC
    keyboard chr(K_F9)
  inkey()
endif
Return
//-----
// Pantalla 3
// Datos Personales
// Developed by Ing. Luis Hernandez Pérez
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
//-----
procedure Screen3
if Edit_Mode != 'READ'
  @ 6,1 clear to 22,78
endif
@ 10, 2 TO 10,77
@ 14,23 TO 14,37
@ 17,23 TO 17,37
@ 4,75 SAY "3/3"
@ 5, 2 SAY "Número de empleado ."
@ 5,62 SAY "Fecha ."
@ 7, 2 SAY "Comisionado ."
@ 7,57 SAY "Tel./Ext.:"
@ 8, 2 SAY "Puesto ."
@ 9, 2 SAY "Area ."
@ 5,23 say numemp
@ 5,70 say fantici
@ 7,18 SAY left(altrim(rh1->nombre)+' '+altrim(RH1->paterno)+' '+altrim(rh1->materno),53)
@ 7,68 SAY rh1->telext
@ 8,18 SAY rh2->Nompues
@ 9,18 SAY rh3->NomArea
@ 11,42 SAY "Salida real :."
@ 12,42 SAY "Llegada real:"
@ 13,02 SAY "Dias :."
@ 16,02 SAY "Gastos de transporte :."
@ 16,42 SAY "Fecha comprobación :."
@ 17,42 SAY "Gastos comprobados :."
@ 18,02 SAY "Gastos a comprobar :."
@ 18,42 SAY "Saldo a favor :."
@ 19,42 SAY "Saldo en contra :."
@ 21,02 SAY "Número de recibo :."
@ 22, 2 say 'Observaciones'
if des->de_tipo = 'I'
  @ 11,02 SAY "Cuota USD :."
  @ 11,18 say cuotaUSD pict '$99,999.99'
  @ 12,02 SAY "Tipo C mbio :."
  @ 12,18 say tica pict '99,999.99'
else
  @ 11,02 SAY "Cuota diaria:"
  @ 11,25 say cuotad pict '$99,999.99'
endif
@ 11,57 GET realsal
@ 11,66 GET realsalt pict '##.##.##'
@ 12,57 GET reallie
@ 12,66 GET realliet pict '##.##.##'
Tcla = LastKey()
if Edit_Mode = 'READ'
  read
  * if updated()
    ICalcDias = reallie-realsal+1
    ILlegadaT := val(left(realsalt,2))+val(right(left(realsalt,5),2))/60
    do case
      case ILlegadaT > 10 .and. ILlegadaT <= 16 .and. ICalcDias > 1
        ICalcDias--
        ICalcDias+=,7
      case ILlegadaT > 16 .and. ICalcDias > 1
        ICalcDias--
        ICalcDias+=,3
    endcase
    ILlegadaT := val(left(realliet,2))+val(right(left(realliet,5),2))/60
    do case
      case ILlegadaT > 11 .and. ILlegadaT <= 20 .and. ICalcDias > 1
        ICalcDias--
        ICalcDias+=,7
      case ILlegadaT <= 11 .and. ICalcDias > 1
        ICalcDias--

```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes: (Clipper)

```
        ICalcDias+=.3
    endcase
    repl rdias with ICalcDias
*   endif
endif
@ 13,25 GET rdias pict ' 99,999.99'
@ 16,25 get gtrans                pict '$99,999.99'
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP.or. Tcla=K_F4.or. Tcla=K_F2.or Tcla=K_F5) .and. Edit_Mode = 'READ'
    read
    repl acompro with rdias*iif(des->de_tipo='I',cuotaUSD*tica,cuotaD)+gViaje+gTaxi+gTrans
endif
@ 15,25 say rdias*iif(des->de_tipo='I',cuotaUSD*tica,cuotaD) pict '$99,999.99'
@ 18,25 get acompro                pict '$99,999.99'
@ 16,65 get fcompro
@ 17,65 get compro                pict '$99,999.99'
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP.or. Tcla=K_F4.or. Tcla=K_F2.or Tcla=K_F5) .and. Edit_Mode = 'READ'
    read
endif
if acompro > compro
    repl scargo with acompro-compro,sfavor with 0
else
    repl sfavor with compro-acompro,scargo with 0
endif
@ 18,65 say sfavor                pict '$99,999.99'
@ 19,65 say scargo                pict '$99,999.99'
@ 21,25 get numrecibo            pict ' 999'
@ 22,25 get aviaobs pict '@!'
Tcla = LastKey()
if !(Tcla=K_ESC.or. Tcla=K_PGDN.or. Tcla=K_PGUP.or. Tcla=K_F4.or. Tcla=K_F2.or. Tcla=K_F5) .and. Edit_Mode = 'READ'
    read
endif
clear gets
if LastKey() = K_ESC
    keyboard chr(K_F9)
    inkey()
endif
return
//-----
// Pantalla 2
// Datos Personales
// Developed by Ing. Luis Hernandez P,rez
//-----
procedure Screen2
if Edit_Mode != 'READ'
    @ 6,1 clear to 22,78
endif
@ 10, 2 TO 10,77
@ 4,75 SAY "2/3"
@ 5, 2 SAY "NEmero de empleado ."
@ 5,62 SAY "Fecha ."
@ 7, 2 SAY "Comisionado ."
@ 7,57 SAY "Tel./Ext.:"
@ 8, 2 SAY "Puesto ."
@ 9, 2 SAY "Area ."
@ 5,23 say numemp
@ 5,70 say fantici
@ 7,18 SAY left(altrim(rh1->nombre)+' '+altrim(RH1->paterno)+' '+altrim(rh1->materno),53)
@ 7,68 SAY rh1->telex
@ 8,18 SAY rh2->NomPues
@ 9,18 SAY rh3->NomArea
@ 14, 2 say 'CIA de Aviación .'
@ 15, 2 say 'Destino .'
@ 16, 2 say 'No. de Boleto .'
@ 17, 2 say 'No. de Vuelo de Salida .'
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Clipper)

```
* Programa.....: via001.prg
* .....: Elimina Anticipos de bases de datos
// Developed by Luis Hernández Pérez
*
*
local No_Borrados
set delete off
count for delete() to No_Borrados
do while !(chr>LastKey())$sSnN'
  mensa("Se encontraron '+str>No_Borrados,3)+' anticipos dados de baja. Continua? (s/n)")
enddo
do case
  case chr>LastKey())$nN'
    do while !(chr>LastKey())$sSnN'
      mensa(" Desea recuperar los anticipos dados de baja ? (s/n)")
    enddo
    if chr>LastKey())$sS'
      recall all
    endif
  case chr>LastKey())$sS'
    go top
    do while !eof()
//---- Bajas en diferentes bases de datos -----
      if Delete()
        iNumAnt = NumAnt
        sele 3
        cent("Dando de baja anticipos en la comprobación ...",23)
        delete all for Numant = iNumant
      endif
      sele 1
      skip
    enddo
    pack
    sele 3
    pack
  endcase
do borr23
return
```

```
//-----
// Tarifas
//
// Developed by Ing. Luis Hernández Pérez
//-----
#ifdef K_ESC
#include "InKey.ch"
#endif

local lbBorrado := .f., Texto := " Anticipos "
local oTbDB := TBrowseDB(7,8,19,74)

set delete on
set cursor off
//-----Define pantalla-----
win(6,5,20,75,2,0)
@ 6,60 say Texto

set color to n/w
@ 24,0 say '<U>-Seleccionar Esc-Salir'
set color to w/n

//-----Relaciona archivos-----
sele 1
//-----Define TBrowse para Altas-----
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
oTbDB:addColumn(TBColumnNew('No. Ant.' ,{||via->numant}))
oTbDB:addColumn(TBColumnNew('Alfa' ,{||via->fanticl}))
oTbDB:addColumn(TBColumnNew('No. Emp' ,{||via->numemp}))
oTbDB:addColumn(TBColumnNew('Nombre del Empleado'+''+alltrim(rh1->paterno)+''+alltrim(rh1->nombre)+space(40),40)))
oTbDB:addColumn(TBColumnNew('Destino' ,{||des->de_nomdes}))
oTbDB:addColumn(TBColumnNew('Anticipo' ,{||via->anticipo}))
oTbDB:addColumn(TBColumnNew('Gastos de viaje' ,{||via->gviaje}))
oTbDB:addColumn(TBColumnNew('A comprobar' ,{||via->acompro}))
oTbDB:addColumn(TBColumnNew('Comprobados' ,{||via->compro}))
oTbDB:addColumn(TBColumnNew('Fecha' ,{||via->fcompro}))
oTbDB:colorSpec := 'w/n,n/w,w*/n,n/w,n/w'
oTbDB:colsep := '*'
oTbDB:headsep := 'A'
// oTbDB:freeze := 1
//-----Navegación-----
CurRec = Recno()
do while LastKey() != K_ESC
  while !oTbDB:stabilize()
    enddo
    inkey(0)
    do case
//      oTbDB[2]:refreshAll()
      case LastKey() = K_UP
        oTbDB:up()
      case LastKey() = K_DOWN
        oTbDB:down()
      case LastKey() = K_RIGHT
        oTbDB:right()
      case LastKey() = K_LEFT
        oTbDB:left()
      case LastKey() = K_PGDN
        oTbDB:pageDown()
      case LastKey() = K_PGUP
        oTbDB:pageUp()
      case LastKey() = K_ENTER
        CurRec = Recno()
        keyboard chr(K_ESC)
        inkey()
    endcase
  enddo
  goto CurRec
  l_NumAnt := via->NumAnt
  set cursor on
  set delete off
  cwin(6,5,20,75)
  if LastKey() = K_ESC
    keyboard chr(K_F9)
    inkey()
  endif
  sele 8
  set order to 1
  sele 2
  set order to 2
  sele 3
  set order to 1
  sele 4
  set order to 1
  sele 1
  return
//-----
// Tarifas
// Program : VIA200.prg
// Developed by Ing. Luis Hernandez Pérez
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Clipper)

```
//-----
#ifdef K_ESC
#include "InKey.ch"
#endif

local lbBorrado = .f., Texto := " Comprobación"
local oTbDB := TBrowseDB(7,6,19,74)

set delete on
set cursor off
//-----Define pantalla-----
win(6,5,20,75,2,0)
@ 6,60 say Texto
set color to b/w
// @ 24,10 say 'F2-Reporte Ins-Agregar Supr-Borrar <U-Editar Esc-Principal'
@ 24,10 say 'Ins-Agregar Supr-Borrar <U-Editar Esc-Principal'
set color to w/b

//-----Relaciona archivos-----
sete 3
//-----Define TBrowse para Altas-----
oTbDB:addColumn(TBColumnNew("No. Ant.",{com->numant}))
oTbDB:addColumn(TBColumnNew("Fecha",{com->actividad}))
oTbDB:addColumn(TBColumnNew("Hospedaje",{com->hospedaje}))
oTbDB:addColumn(TBColumnNew("I.V.A.",{com->ivahosp}))
oTbDB:addColumn(TBColumnNew("Alimentación",{com->alimentac}))
oTbDB:addColumn(TBColumnNew("I.V.A.",{com->ivaalimen}))
oTbDB:addColumn(TBColumnNew("Transporte",{com->transport}))
oTbDB:addColumn(TBColumnNew("I.V.A.",{com->ivatran}))
oTbDB:addColumn(TBColumnNew("Tel.fono",{com->telefono}))
oTbDB:addColumn(TBColumnNew("I.V.A.",{com->ivatele}))
oTbDB:addColumn(TBColumnNew("Material Foto.",{com->fotografi}))
oTbDB:addColumn(TBColumnNew("I.V.A.",{com->ivafoto}))
oTbDB:addColumn(TBColumnNew("Gastos Legales",{com->glegales}))
oTbDB:addColumn(TBColumnNew("I.V.A.",{com->ivaLegal}))
oTbDB:addColumn(TBColumnNew("Otros Gastos",{com->otros}))
oTbDB:addColumn(TBColumnNew("I.V.A.",{com->ivaotro}))
oTbDB:colorSpec := 'w/n,n/w,w/n,n,w/n,w/n'
oTbDB:colsep := ' '
oTbDB:headsep := 'A'
oTbDB:freeze := 1
//-----Navegación-----
set filter to com->NumAnt = via->NumAnt
go top
do while LastKey() != K_ESC
while !oTbDB:stabilize()
enddo
inkey(0)
do case
// oTbDB[2]:refreshAll()
case LastKey() = K_UP
oTbDB:up()
case LastKey() = K_DOWN
oTbDB:down()
case LastKey() = K_RIGHT
oTbDB:right()
case LastKey() = K_LEFT
oTbDB:left()
case LastKey() = K_PGDN
oTbDB:pageDown()
case LastKey() = K_PGUP
oTbDB:pageUp()
case LastKey() = K_INS
dbappend()
repl com->numant with via->NumAnt
oTbDB:configure()

```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
case LastKey() = K_DEL
  if delete()
    recall
  else
    delete
  endif
  oTbDB:configure()
  lbBorrado = .t.
case LastKey() = K_ENTER
  set cursor on
  if oTbDB.colPos > 1
    cCampo = FieldName(oTbDB.colPos)
    nPosicion = oTbDB.colPos
    oColumna = oTbDB.getColumn(nPosicion)
    @ oTbDB:nTop + oTbDB:rowPos + 1, col() get &cCampo pict '@!'
    read
    if LastKey() = K_ESC
      keyboard chr(K_F9)
      inkey()
    endif
    oTbDB:setColumn(nPosicion,oColumna)
    oTbDB:configure()
  endif
  set cursor off
case LastKey() = K_F2
  // RepDestino()
endcase
enddo
LocTot := LocIva := 0
go top
do while !eof()
  LocTot += com->hospedaje+com->alimentac+com->transport+com->telefono +
    com->fotografi+com->glegales +com->otros
  LocIva += com->ivahosp +com->ivaalimen+com->ivatran +com->ivatele +
    com->ivafoto +com->ivaLega +com->ivaotro
  skip
enddo
repl com->Total with LocTot
repl com->ivatot with LocIva
repl via->Compro with LocTot + LocIva

set filter to
if lbBorrado
  @ 23,10 say 'Favor de esperar, se estan borrando los registros marcados .'
  pack
  do borr23
endif
set cursor on
set delete off
cwin(6,5,20,75)
if LastKey() = K_ESC
  keyboard chr(K_F9)
  inkey()
endif
sele 8
set order to 1
sele 2
set order to 2
sele 3
set order to 1
sele 4
set order to 1
sele 1
return
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuertes (Clipper)

```
//-----
//- Impresión de formato de Autorización
//- Program VIA301.prg
//- Developed by Ing. Luis Hernandez Perez
//-----
local lMeses := {'Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio',
                'Agosto','Septiembre','Octubre','Noviembre','Diciembre'}

lLine=1
DO CHIMPRES
if LastKey() != 27
!COPY VIATICOS.PRN LPT1: /B >nul // ----- Impresión del logotipo -----
set devi to print
@ lLine .0 say chr(18) // ----- Letrotas -----
@ lLine .40 say '
@ lLine .35 SAY 'SOLICITUD DE ANTICIPO PARA GASTOS DE'
@ lLine .35 SAY 'SOLICITUD DE ANTICIPO PARA GASTOS DE'
@ lLine +1.36 say 'VIAJE Y TRANSPORTACION EN COMISION'
@ lLine +1.36 say 'VIAJE Y TRANSPORTACION EN COMISION'
ViaTitulo()
lLine = 14
@ lLine+0 .1 say 'SOLICITUD DE BOLETO DE : AVION ____ AUTOBUS ____'
if via->aviacia <> '
@ lLine+0 .35 say 'XX'
else
@ lLine+0 .46 say 'XX'
endif
@ lLine+1 .1 SAY 'CIA. AEREA : '+via->AVIACIA
@ lLine+1 .24 SAY REPLICATE('_',53)
@ lLine+2 .1 SAY 'DESTINO : '+via->AVIADEST
@ lLine+2 .24 SAY REPLICATE('_',53)
@ lLine+3 .1 SAY 'FECHA Y HORA DE SALIDA '+via->AVIAFHS
@ lLine+3 .24 SAY REPLICATE('_',30)
@ lLine+3 .55 SAY 'NO.DE VUELO '+via->AVIANVS
@ lLine+3 .68 SAY REPLICATE('_',09)
@ lLine+4 .1 SAY 'FECHA Y HORA DE RETORNO '+via->AVIAFHR
@ lLine+4 .24 SAY REPLICATE('_',30)
@ lLine+4 .55 SAY 'NO.DE VUELO '+via->AVIANVR
@ lLine+4 .68 SAY REPLICATE('_',09)
@ lLine+5 .1 SAY 'OBSERVACIONES '+via->AVIAOBS
@ lLine+5 .24 SAY REPLICATE('_',53)
@ lLine+6 .1 say 'VEHICULO : PROPIO ____ DEL FIFOMI ____'
@ lLine+7 .0 say replicate('A',77)
lLine = 22
@ lLine .1 say 'ANTICIPO PARA GASTOS DE VIAJE EN COMISION : '
@ lLine+1 .1 say 'RECIBI DE LA CAJA DEL NONBRE DE LA INSTITUCIÓN'
@ lLine+2 .1 say ' LA CANTIDAD DE $ '
@ lLine+2.60 say transform(via->Anticipo,###,###,###)
@ lLine+2.60 say transform(via->Anticipo,###,###,###)
@ lLine+2.58 say replicate('_',19)
@ lLine+3 .1 say ('+Letras(via->Anticipo)+')
@ lLine+3 .1 say replicate('_',76)
@ lLine+4 .1 say 'POR CONCEPTO DE ANTICIPO PARA GASTOS COMO SIGUE:'
lLine = 28
@ lLine .1 say 'VIATICOS DE '+alltrim(TRANSFORM(VIA->PDias,###,###,###))+ ' DIAS A RAZON DE $ '
+alltrim(TRANSFORM(VIA->VIATICOS,###,###,###))+ ' POR DIA'
@ lLine .12 say replicate('_',8)
@ lLine .35 say replicate('_',8)
@ lLine .67 say '$'+TRANSFORM(VIA->VIATICOS/via->PDias,#,###,###)
@ lLine .69 say replicate('_',8)
@ ++lLine .1 say 'HOSPEDAJE A RAZON DE $ '+alltrim(TRANSFORM(VIA->HOSPEDAJE,#,###,###))
@ lLine .23 say replicate('_',9)
@ lLine.32 say 'POR NOCHE'
@ lLine.67 say '$'+TRANSFORM(VIA->hospedaje,#,###,###)
@ lLine.69 say replicate('_',8)
if via->gViaje> 0
@ ++lLine .1 say 'GASTOS DE KILOMETRAJE '+alltrim(TRANSFORM(DES->DE_KMS,###,###))+ ' KMS. APROX. A RAZON DE
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
$+alltrim(TRANSFORM(GASOLINA*0.05,#.##))+ ' POR KM'
  @ ILine,23 say replicate('_',5)
  @ ILine,67 say '$+TRANSFORM(VIA->gViaje.#.###.##)'
  @ ILine,69 say replicate('_',8)
*
end:f
  @ ILine+1,45 say 'CANTIDAD A ENTREGAR $'
  @ ILine+1,69 say transform(via->Anticipo,#.###.##)
  @ ILine+1,69 say replicate('_',8)
  @ ILine+1,70 say ''
  @ ILine+2,0 say replicate('Á',77)
Iline = 33
  @ ILine , 1 say chr(15) // -----Letritas -----
  @ ILine , 1 say 'NOTA: CON OBJETO DE TRAMITAR OPORTUNAMENTO EL (LOS) BOLETO (S) SOLICITADO (S), ESTE
FORMATO DEBERA PRESENTARSE EN LA GERENCIA DE'
  @ ILine+1,1 say 'AREA RESPONSABLE CON 3 (TRES) DIAS HABLES DE ANTICIPACION COMO MINIMO. LOS COSTOS
QUE SE ORIGINEN POR CANCELACIONES Y/O CAMBIOS'
  @ ILine+2,1 say 'DE HORARIO NO INPUTABLES A LA ENTIDAD. SE LE COBRARAN AL COMISIONADO. LOS CASOS
IMPUTABLES A LA ENTIDAD, DEBERAN SER AUTORIZADOS'
  @ ILine+3,1 say 'POR EL DIRECTOR DEL AREA CORRESPONDIENTE'
  @ ILine+4,1 say 'EN BASE AL REGLAMENTO DE VIATICOS Y GASTOS DE VIAJE EN EL PUNTO 10.1, LA COMPROBACION
DEBERA PRESENTARSE EN UN PLAZO MAXIMO DE 5'
  @ ILine+5,1 say '(CINCO) DIAS HABLES, CONTADOS A PARTIR DE LA FECHA DE RETORNO, DE LO CONTRARIO,
AUTORIZO SE ME DESCUENTE POR NOMINA EL IMPORTE'
  @ ILine+6,1 say 'DEL ANTICIPO.'
  @ ILine+7,63 say 'Vo. Bo. AUTORIZACION'
  @ ILine+8,1 say 'COMISIONADO DE LA PERSONA QUE COMISIONA DE ANTICIPO Y
TRANSPORTACION'
  @ ILine+10 ,5 say replicate('-',29)
  @ ILine+10 ,52 say replicate('-',31)
  @ ILine+10 ,92 say replicate('-',28)
  @ ILine+11,7 say alltrim(rh1->Nombre)+' +alltrim(rh1->Paterno)+' +alltrim(rh1->Materno)
  @ ILine+11 ,55 say via->comision
  @ ILine+11,92 say param->Autoriza
  @ ILine+12,1 say chr(18) // ----- Letrotas -----
  @ ILine+12,0 say replicate('Á',77)
  @ ILine+13,0 say chr(15) // ----- Letritas -----
  @ ILine+14,63 say 'Vo. Bo.'
  @ ILine+15,55 say 'PARA VIAJES AL EXTRANJERO'
  @ ILine+17,55 say '
  @ ILine+18,55 say 'DIRECCION GENERAL'
  @ ILine+20,1 say 'PERSONA QUE RECIBE EL BOLETO DE AVION'
  @ ILine+22,1 say 'BOLETO No.
  @ ILine+22,92 say '
  @ ILine+23,92 say 'NOMBRE Y FIRMA'
  @ ILine+24,0 say chr(18)
  @ ILine+24,0 say replicate('Á',77)
RETU
eject
set devi to screen
end:f
*do borr23
return

//-----
//- Impresión de formato de Autorización
//- Program VIA302.prg
//- Developed by Ing. Luis Hernandez Perez
//-----
local lMeses := {'Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio',
'Agosto','Septiembre','Octubre','Noviembre','Diciembre'}

DO CHIMPRE
if LastKey() != 27
!COPY VIATICOS.PRN LPT1: /B >nul // ----- Impresión del logotipo -----
set devi to print
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ prow(),0 say chr(18) // -----Letrotas-----
@ prow(),40 say '
@ prow(),44 SAY 'SOLICITUD DE TRANSPORTACION AEREA'
@ prow(),44 SAY 'SOLICITUD DE TRANSPORTACION AEREA'
ViaTitulo()
!Line = 14
@ !Line+ , 1 say 'SOLICITA UN BOLETO DE AVION CON LAS SIGUIENTES CARACTERISTICAS:'
@ !Line+1 , 1 SAY 'CIA. DE AVIACION '+VIA->AVIACIA
@ !Line+1 ,22 SAY REPLICATE('_',55)
@ !Line+2 , 1 SAY 'DESTINO '+VIA->AVIADEST
@ !Line+2 ,22 SAY REPLICATE('_',55)
@ !Line+3 , 1 SAY 'NO. VUELO SALIDA '+VIA->AVIANVS
@ !Line+3 ,22 SAY REPLICATE('_',55)
@ !Line+4 , 1 SAY 'FECHA Y HORA SALIDA '+VIA->AVIAFHS
@ !Line+4 ,22 SAY REPLICATE('_',55)
@ !Line+5 , 1 SAY 'NO. VUELO DE RETORNO '+VIA->AVIANVR
@ !Line+5 ,22 SAY REPLICATE('_',55)
@ !Line+6 , 1 SAY 'FECHA Y HORA RETORNO '+VIA->AVIAFHR
@ !Line+6 ,22 SAY REPLICATE('_',55)
@ !Line+7 , 1 SAY 'OBSERVACIONES '+VIA->AVIAOBS
@ !Line+7 ,22 SAY REPLICATE('_',55)
@ !Line+ 9, 0 say replicate('Á',77)
!Line = 26
@ !Line , 1 say 'COMISIONADO'
@ !Line , 32 SAY 'Vo. Bo.'
@ !Line , 53 SAY 'AUTORIZACION'
@ !Line+4 , 1 say replicate('Á',24)
@ !Line+4 , 26 say replicate('Á',24)
@ !Line+4 , 52 say replicate('Á',24)
@ !Line+4 , 0 say chr(15) // -----Letritas-----
@ !Line+5 , 1 SAY alltrim(rh1->Nombre)+' alltrim(rh1->Paterno)+' alltrim(rh1->Materno)
@ !Line+5 , 45 SAY VIA->COMISION
@ !Line+5 , 90 SAY PARAM->AUTORIZA
@ !Line+5 , 0 say chr(18) // -----Letrotas-----
@ !Line+7 , 0 say replicate('Á',77)
!Line = 35
@ !Line , 1 say 'CON OBJETO DE TRAMITAR OPORTUNAMENTE LOS BOLETOS SOLICITADOS, ESTA FORMA'
@ !LINE+1 , 1 SAY 'DEBERA PRESENTARSE EN LA AREA RESPONSABLE CON 3(TRES) DIAS'
@ !Line+2 , 1 SAY 'HABILES DE ANTICIPACION COMO MINIMO, ASI MISMO INDICAMOS QUE TODO CARGO POR'
@ !Line+3 , 1 SAY 'CANCELACION EXTEMPORANEA MOTIVADA POR EL SOLICITANTE, SERA POR CUENTA DE EL.'
@ !Line+6 , 36 SAY 'PERSONA QUE LO RECIBE'
@ !Line+11 , 1 SAY 'RECIBI BOLETO No.'
@ !Line+11 , 19 SAY replicate('Á',10)+' '+replicate('Á',20)+' '+replicate('Á',20)
@ !Line+12,36 SAY 'NOMBRE'
@ !Line+12,58 SAY 'FIRMA'
eject
set devi to screen
endif
do borr23
return

//-----
//- Impresión de formato de Comprobación
//- Program : VIA303.prg
//- Developed by Ing. Luis Hernández Pérez
//-----
local !Meses := {Enero,'Febrero','Marzo','Abril','Mayo','Junio','Julio',
'Agosto','Septiembre','Octubre','Noviembre','Diciembre'}
local !Totosp := !Totalimen := !tototran := !Tototele := !TotFoto := ;
!Totlega := !TotOtro := 0
local !Tivosp := !Tivalimen := !tivtran := !Tivtele := !TivFoto := ;
!Tivlega := !TivOtro := 0
DO CHIMPRE
if LastKey() != 27
!COPY VIATICOS.PRN LPT1: /B >nul // ----- Impresión del logotipo -----
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
set devi to print
@ prow(),0 say chr(18) // -----Letrotas-----
@ prow(),40 say '
@ prow(),34 SAY 'COMPROBACION DE GASTOS DE VIAJE EN COMISION'
@ prow(),34 SAY 'COMPROBACION DE GASTOS DE VIAJE EN COMISION'
ViaTitulo()
!Line = 15
@ !Line ,1 say 'SALIDA '+dtoc(via->realSal)+' '+VIA->realsalt
@ !Line ,35 say 'CUOTA DIARIA $'+TRANSFORM(via->cuotaD,###,###,##)
@ !Line+ 2,1 say 'RETORNO '+dtoc(via->realLLE)+' '+via->realllet
@ !Line+ 2,35 say 'PROPORCION DE DIAS $'+TRANSFORM(via->cuotaD,###,###,##)
@ !Line+ 3,7 say '(dja/mes/año) hora'
@ !Line+ 4,0 say replicate('Á',77)
@ prow(),0 say chr(15) // -----Letritas-----
!Line = 20
@ !Line, 1 say 'FECHA'
@ !Line, 10 say ' HOSPEDAJE'
@ !Line, 25 say ' ALIMENTACION'
@ !Line, 40 say ' TRANSPORTE'
@ !Line, 55 say ' TELEFONO'
@ !Line, 70 say ' MAT.FOTO'
@ !Line, 85 say 'GASTOS LEGALES'
@ !Line,100 say ' OTROS GASTOS'
@ !Line,115 say ' TOTAL'
@ !Line+1,0 say replicate('Á',132)
sele 3
set order to 2
set filter to numant = via->numant
!Line += 2
go top
do while !eof()
@ !Line,1 say com->actividad
Ponlo(!Line,com->hospedaje,com->alimentac,com->transport,com->telefono,,
com->fotografi,com->glegales,com->otros)
!Totohosp += com->hospedaje
!Totalimen += com->alimentac
!Tototran += com->transport
!Tototele += com->telefono
!TotFoto += com->fotografi
!Totlega += com->glegales
!TotOtro += com->otros
!tivhosp += com->ivahosp
!tivalimen += com->ivaalimen
!tivtran += com->ivatran
!tivtele += com->ivatele
!tivFoto += com->ivafoto
!tivlega += com->ivalega
!tivOtro += com->ivaotro

skip
!Line++
enddo
!Line += 1
@ !Line,1 say 'SUBTOTAL'
Ponlo(!Line,!Totohosp,!Totalimen,!Tototran,!Tototele,,
!totfoto,!Totlega,!Tototro)
!Line++
@ !Line,1 say 'I.V.A.'
Ponlo(!Line,!Tivhosp,!Tivalimen,!Tivtran,!Tivtele,,
!tivfoto,!Tivlega,!Tivotro)
!Line++
@ !Line,1 say 'TOTAL'
Ponlo(!Line,!Tivhosp+!Totohosp,!Tivalimen+!Totalimen,!Tivtran+!Tototran,,
!Tivtele+!Tototele,!tivfoto +!Totfoto ,!Tivlega+!Totlega,,
!Tivotro+!TotOtro)

set filter to
set order to 1
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
sele 1
@ prow(),0 say chr(18) // -----Letrotas-----
!Line := 35
@ !Line , 1 say 'COMISIONADO'
@ !Line ,42 say 'ANTICIPOS.'
@ !Line+1,42 say ' EFECTIVO '$+transform(via->Acompro-via->Gtrans,###,###,##)
@ !Line+2,42 say ' TRANSPORTE '$+transform(via->gTrans,###,###,##)
@ !Line+3,63 say '$'+transform(via->acompro,###,###,##)
@ !Line+3,65 say replicate('_',10)
@ !Line+4, 1 say replicate('_',35)
@ !Line+4,42 say 'SALDO A '+if(via->acompro < via->compro,'FAVOR','CONTRA')
@ !Line+4,63 say '$'+transform(if(via->acompro < via->compro,via->sFAVOR,via->scargo),###,###,##)
@ !Line+4,63 say '$'+transform(if(via->acompro < via->compro,via->sFAVOR,via->scargo),###,###,##)
@ !Line+5, 1 say alltrim(rh1->Nombre)+' +alltrim(rh1->Paterno)+' +alltrim(rh1->Materno)
!Line := 43
@ !Line , 1 say 'FUNCIONARIO QUE COMISIONO'
@ !Line ,42 SAY 'Vo. Bo. VIAJES AL EXTRANJERO'
@ !Line+4, 1 say replicate('_',35)
@ !Line+4,42 say replicate('_',35)
@ !Line+5, 1 say via->comision
@ !Line+5,42 say param->Director
!Line := 51
@ !Line , 1 say 'REVISY VERIFICA'
@ !Line ,42 SAY 'AUTORIZA'
@ !Line+4, 1 say replicate('_',35)
@ !Line+4,42 say replicate('_',35)
@ !Line+5, 1 say param->Revisa
@ !Line+5,42 say param->Autoriza

eject
set devi to screen
endif
do borr23
return
//-----
Function Ponto
Parameter !Li,hos,ali,tra,tel,fot,gle,otr
@ !Li, 10 say hos pict '999,999,999.99'
@ !Li, 25 say ali pict '999,999,999.99'
@ !Li, 40 say tra pict '999,999,999.99'
@ !Li, 55 say tel pict '999,999,999.99'
@ !Li, 70 say fot pict '999,999,999.99'
@ !Li, 85 say gle pict '999,999,999.99'
@ !Li,100 say otr pict '999,999,999.99'
@ !Li,115 say hos+ali+tra+tel+fot+gle+otr pict '999,999,999.99'
return

//-----
//- Impresión de Recibo
//- Program : VIA304.prg
//- Developed by Ing. Luis Hernández Pérez
//-----
local !Meses := ('Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio',
'Agosto','Septiembre','Octubre','Noviembre','Diciembre')
DO CHIMPRE
if LastKey() != 27
set devi to print
!COPY VIATICOS.PRN LPT1: /B >nul // ----- Impresión del logotipo -----
@ prow() , 0 say chr(18) // -----Letrotas-----
@ prow() ,40 say '
@ prow() ,32 SAY 'SOLICITUD DE ENTERO A CAJA POR DEVOLUCION'
@ prow() ,32 SAY 'SOLICITUD DE ENTERO A CAJA POR DEVOLUCION'
@ prow()+1,32 SAY 'DE SOBRENTE DE GASTOS DE VIAJE'
@ prow() ,32 SAY 'DE SOBRENTE DE GASTOS DE VIAJE'
!Line = prow()+1
@ !Line++, 1 say str(day(p_Date),2)+' de '+!Meses[month(p_Date)]+' de '+str(year(p_Date),4)
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ ILine 1 say 'U'+replicate('A',12)+'A'+replicate('A',58)+'
@ ILine+1 1 say ' ' No. DE ' ' + ' ' N O M B R E '
@ ILine+1 .73 say ' '
@ ILine+2 1 say ' ' MATRICULA ' '
@ ILine+2 .73 say ' '
@ ILine+3 1 say ' ' +str(via->NumEmp,12)+' ' +alltrim(rh1->Nombre)+' ' +alltrim(rh1->Paterno)+' ' +alltrim(rh1->Materno)
@ ILine+3 1 say ' ' +str(via->NumEmp,12)+' ' +alltrim(rh1->Nombre)+' ' +alltrim(rh1->Paterno)+' ' +alltrim(rh1->Materno)
@ ILine+3 .73 say ' '
@ ILine+4 1 say 'E'+replicate('I',12)+'I'+replicate('I',58)+'µ'
@ ILine+5 1 say ' ' +space(71)+' ' //Blanco---
@ ILine+6 1 say ' ' CANTIDAD $ '+transform(sFavor,999,999.99)
@ ILine+6 .12 say replicate('_',58)
@ ILine+6 .73 say ' '
@ ILine+7 1 say ' ' +space(71)+' '
@ ILine+8 1 say ' '
@ ILine+8 .3 say alltrim(Letras(via->sFavor))
@ ILine+8 .3 say alltrim(Letras(via->sFavor))
@ ILine+8 .3 say replicate('_',67)
@ ILine+8 .73 say ' '
@ ILine+9 1 say ' ' +space(71)+' ' //Blanco---
@ ILine+10 1 say ' ' +space(71)+' ' //Blanco---
@ ILine+11 1 say ' ' VIAJE DEL '
@ ILine+11 .13 say str(day(via->RealSal),2)+' de ' +IMeses[month(via->RealSal)]+' de ' +str(year(via->RealSal),4)
@ ILine+11 .45 say str(day(via->Realle),2)+' de ' +IMeses[month(via->Realle)]+' de ' +str(year(via->Realle),4)
@ ILine+11 .13 say replicate('_',25)
@ ILine+11 .42 say 'AL '+replicate('_',25)
@ ILine+11 .73 say ' '
@ ILine+12 1 say ' ' +space(71)+' ' //Blanco---
@ ILine+13 1 say ' ' LUGAR DE VISITA '
@ ILine+13 .19 say via->DEstino
@ ILine+13 .19 say replicate('_',51)
@ ILine+13 .73 say ' '
@ ILine+14 1 say 'E'+replicate('I',21)+'N'+replicate('I',49)+'µ'
@ ILine+15 1 say ' ' SUBGERENCIA DE '
@ ILine+15 .73 say ' '
@ ILine+16 1 say ' ' SERVICIOS '
@ ILine+16 .73 say ' '
@ ILine+17 1 say ' ' ADMINISTRATIVOS '
@ ILine+17 .73 say ' '
@ ILine+18 1 say 'A'+replicate('A',21)+'A'+replicate('A',49)+'U'
ILine += 22
@ ILine 1 say ' '
eject
set devi to screen
endif
do borr23
return

//-----
//- Reporte Individual
//- Program VIA305.prg
//- Developed by Ing. Luis Hernández Pérez
//-----
do chimpre
if LastKey() != 27
set devi to print
@ 0,0 say chr(15) //-----Letritas-----
CurRec := RecNo()
CurEmp := via->NumEmp
set filter to via->numemp=CurEmp
go top
do while !eof()
@ 1, 1 say 'NOMBRE DE LA INSTITUCIÓN'
@ 2, 1 say 'AREA RESPONSABLE'
@ 3, 1 say 'NOMBRE : '+alltrim(rh1->Nombre)+' '+alltrim(rh1->Paterno)+' '+alltrim(rh1->Materno)
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes. (Clipper)

```
@ 3, 1 say 'NOMBRE : '+alltrim(rh1->Nombre)+' '+alltrim(rh1->Paterno)+' '+alltrim(rh1->Materno)
@ 3,194 say 'Fecha :'+dtoc(p_Date)
@ 4, 0 say replicate('-',220)
@ 5, 1 say 'Destino'
@ 5, 42 say 'Salida'
@ 5, 54 say 'Regreso'
@ 5, 66 say ' Dias'
@ 5, 79 say ' Cuota'
@ 5, 92 say ' Viaticos'
@ 5,105 say ' Hospedaje'
@ 5,157 say ' Anticipo'
@ 5,118 say ' G.Viaje'
@ 5,131 say ' G. Trans.'
@ 5,144 say ' Total'
@ 5,170 say 'Saldo Favor'
@ 5,183 say 'Saldo Cargo'
@ 5,196 say 'Comprobaci6n'
@ 5,209 say 'No. Recibo'
@ 6, 0 say replicate('-', 220)
lLine = 7
do while !eof() .and. lLine < 55
  @ lLine . 1 say via->Destino
  @ lLine . 42 say via->pLANSal
  @ lLine . 54 say via->PlanLle
  @ lLine . 66 say via->pDias pict '9999,999.99'
  @ lLine . 79 say via->CuotaD pict '9999,999.99'
  @ lLine . 92 say via->Viaticos pict '9999,999.99'
  @ lLine .105 say via->Hospedaje pict '9999,999.99'
  @ lLine .118 say via->Anticipo pict '9999,999.99'
  @ lLine .131 say via->gViaje+;
    via->gTaxi pict '9999,999.99'
  @ lLine .144 say via->gTrans pict '9999,999.99'
  @ lLine .157 say via->Anticipo+;
    via->gViaje+;
      via->gTaxi+;
        via->gTrans pict '9999,999.99'
  @ lLine .170 say via->sfavor pict '9999,999.99'
  @ lLine .183 say via->scargo pict '9999,999.99'
  @ lLine .196 say via->fcompro
  @ lLine .212 say via->NumRecibo
  skip
  lLine++
enddo
enddo
@ lLine.0 say ''
set filter to
goto CurRec
set devi to screen
endif
do borr23
return

//-----
//- Reporte General
//- Program : VIA306.prg
//- Developed by Ing. Luis Hernandez Perez
//-----
do chimpre
if LastKey() != 27
set devi to print
@ 0,0 say chr(15) //-----Letritas-----
CurRec := RecNo()
go top
do while !eof()
  @ 1, 1 say 'NOMBRE DE LA INSTITUCI6N'
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ 2, 1 say 'AREA RESPONSABLE'
@ 3, 1 say 'LISTADO GENERAL DE ANTICIPOS'
@ 3,194 say 'Fecha :'+dtoc(p_Date)
@ 4, 0 say replicate('-',210)
@ 5, 1 say 'No. Anticipo'
@ 5, 14 say 'Nombre'
@ 5, 55 say 'Puesto'
@ 5, 95 say 'Destino'
@ 5,135 say 'Salida'
@ 5,147 say 'Regreso'
@ 5,159 say 'Importe'
@ 5,170 say 'Observaciones'
@ 6, 0 say replicate('-',210)
!Line = 7'
do while !eof() .and. !Line < 55
  @ !Line ,1 say numant
  if via->numemp<>999999
    @ !Line ,14 say alltrim(rh1->Nombre)+' '+alltrim(rh1->Paterno)+' '+alltrim(rh1->Materno)
    @ !Line , 55 say rh2->Nompuas
  else
    @ !Line ,14 say via->nomhon
    @ !Line , 55 say 'H O N O R A R I O S'
  endif
  @ !Line , 95 say via->Destino
  @ !Line ,135 say via->REalSal
  @ !Line ,147 say via->Realle
  @ !Line ,159 say via->Acompro
  @ !Line ,170 say via->aviaobs
  skip
  !Line++
enddo
enddo
@ !Line,0 say ''
goto CurRec
set devi to screen
endif
do borr23
return
```

```
//-----
// Destinos
//- Program : VIA401.prg
//- Developed by Ing. Luis Hernández Pérez
//-----
#ifndef K_ESC
#include "InKey.ch"
#endif
local oTbDB := TBrowseDB(7,6,19,74)
local lbBorrado := .f., Texto := ' Destinos '
set delete on
set cursor off
//-----Define pantalla-----
win(6,5,20,75,2,0)
@ 6,60 say Texto

set color to n/w
@ 24,0 say 'F2-Reporte Ins-Agregar Supr-Borrar <Ú-Editar Esc-Salir'
set color to win
//-----Relaciona archivos-----
sele 8
set order to 1
sele 2
set order to 1
go bottom
!LastNumDes := de NumDes
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
lLastNumDes++
set order to 1
//-----Define TBrowse para Atlas-----
oTbDB:addColumn(TBColumnNew('Ent. Federativa',{||enf->ef_noment}))
oTbDB:addColumn(TBColumnNew('Destino',{||des->de_nomdes}))
oTbDB:addColumn(TBColumnNew('Zona',{||des->de_zona}))
oTbDB:addColumn(TBColumnNew('Dist (Kms)',{||des->de_kms}))
oTbDB:colorSpec := 'w/n,n/w,w*/n,n/w,n/w
oTbDB:colsep := ' '
oTbDB:headsep := 'Ä'
// oTbDB:freeze := 1
//-----Navegación-----
sele 2
CurRec = REcNo()
go top
do while LastKey() != K_ESC
  while !oTbDB:stabilize()
  enddo
  inkey(0)
  do case
  // oTbDB[2]:refreshAll()
  case LastKey() = K_UP
    oTbDB:up()
  case LastKey() = K_DOWN
    oTbDB:down()
  case LastKey() = K_RIGHT
    oTbDB:right()
  case LastKey() = K_LEFT
    oTbDB:left()
  case LastKey() = K_PGDN
    oTbDB:pageDown()
  case LastKey() = K_PGUP
    oTbDB:pageUp()
  case LastKey() = K_INS
    set cursor on
    AddDestino()
    oTbDB:configure()
    set cursor off
  case LastKey() = K_DEL
    if delete()
      recall
    else
      delete
    endif
    oTbDB:configure()
    lbBorrado = .t.
  case LastKey() = K_ENTER
    set cursor on
    cCampo = if(oTbDB:colPos > 1,FieldName(oTbDB:colPos+1),FieldName(oTbDB:colPos))
    nPosicion = oTbDB:colpos
    oColumna = oTbDB:getColumn(nPosicion)
    if oTbDB:colPos > 1
      if oTbDB:colPos = 3
        @ oTbDB:nTop + oTbDB:rowPos + 1, col() get &cCampo pict '#' valid &cCampo>='1'.and. &cCampo<=if(de_tipo='1','3','4')
        else
          @ oTbDB:nTop + oTbDB:rowPos + 1, col() get &cCampo pict '@!'
        endif
      read
      else
        if &cCampo > 0
          INumEst := GetEstado(&cCampo,de_tipo)
          if INumEst > 0
            repl &cCampo with INumEst
          else
            keyboard chr(K_PGDN)
            inkey()
          endif
        else
          keyboard chr(K_PGDN)
          inkey()
        endif
      endif
    endif
  enddo
enddo
inkey()
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes. (Clipper)

```
endif
endif

endif
oTbDB:setColumn(nPoscion,oColumna)
oTbDB:configure()
set cursor off
case LastKey() = K_F2
  RepDestino('N')
  RepDestino('I')
endcase
enddo
goto CurRec
if lbBorrado
  @ 23,10 say 'Favor de esperar, se estan borrando los registros marcados..'
  sele 2
  pack
  do borr23
endif
set cursor on
set delete off
cwin(6,5,20,75)
if LastKey() = K_ESC
  keyboard chr(K_F9)
  inkey()
endif
sele 8
set order to 1
sele 2
set order to 2
sele 3
set order to 1
sele 4
set order to 1
sele 1
return
Function RepDestino
parameter INacion
local aZonas := {{0,0,0,0}}
do chimpre
if LastKey() = K_ESC
do borr23
return
endif
set devi to print
@ 0,0 say chr(15) //-----letras pequenas-----
set order to 1
set filter to de_tipo = INacion
go top
laLine := lMaxLine := 0
do while !eof()
  @ 1,1 say 'NOMBRE DE LA INSTITUCIÓN'
  @ 2,1 SAY 'AREA RESPONSABLE'
  if INacion = 'N'
    @ 3,1 SAY 'CLASIFICACION DE ESTADOS Y MUNICIPIOS PARA ASIGNACION DE CUOTAS DE VIATICOS'
  else
    @ 3,1 SAY 'CLASIFICACION DE PAISES PARA ASIGNACION DE CUOTAS DE VIATICOS'
  endif
  @ 3,117 say 'Fecha :'+dtoc(p_Date)
  @ 4, 0 SAY replicate('-',150)
  @ 5, 1 say 'ENTIDAD'
  @ 5, 27 say 'ZONA I'
  @ 5, 57 say 'ZONA II'
  @ 5, 87 say 'ZONA III'
  @ 5,117 say 'ZONA IV'
  @ 6, 0 SAY replicate('-',150)
  lLine = 7
enddo
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
do while !eof() .and. lLine < 60
  @ ++lLine,1 say enf->ef_noment
  // if val(enf->ef_zona) > 0 .and. lMaxLine = 0
  // aadd(aZonas[val(enf->ef_zona)],RESTO DEL ESTADO)
  // endif
  //-----llenada de arreglo-----
  lNumEnt := de_nument
  do while !eof() .and. de_nument = lNumEnt .and. lMaxLine = 0
    aadd(aZonas[val(des->de_zona)],des->de_NomDes)
    skip
  enddo

  //-----Impresión de arreglo-----
  lMaxLine := max(len(aZonas[1]),max(max(len(aZonas[2]),len(aZonas[3]),len(aZonas[4])))
  do while laLine < lMaxLine .AND. lLine < 60
    laLine++
    for i:=1 to 4
      if len(aZonas[i]) >= laLine
        @ lLine,30*(i-1)+27 say aZonas[i][laLine]
      endif
    next
    lLine++
    if lMaxLine = laLine
      lMaxLine := laLine := 0
      asize(aZonas[1],0)
      asize(aZonas[2],0)
      asize(aZonas[3],0)
      asize(aZonas[4],0)
    endif
  enddo

  //-----
  enddo
enddo
@ lLine+2,0 say ''
set devi to screen
do borr23
set filter to
set order to 3
return
Function AddDestino
local lde_NumEnt := 0
win(7,2,14.55,1,0)
do while .t
  lde_NomDes = space(30)
  lde_zona := '1'
  lde_kms := 0
  lde_Tipo := 'N'
  @ 8,5 say 'Destino : ' get lde_NomDes pict '@!'
  read
  if LastKey() = K_ESC .or. lde_NomDes = ''
    keyboard chr(K_F9)
    inkey()
    exit
  endif
  // seek alltrim(lde_NomDes)
  // if found()
  // mensa('Destino ya existente, intente otra vez ...')
  // loop
  // endif
  @ 9,5 say '(I)nternacional'
  @ 10,5 say '(N)acional : ' get lde_Tipo pict '!'
  read
  if lde_Tipo = 'N'
    @ 11,5 say 'Estado : //23
    lde_NumEnt := GetEstado(0,lde_Tipo)
    @ 11,23 say enf->ef_noment
    @ 12,5 say 'Zona (1,2,3,4) : ' get lde_zona pict '#' valid lde_zona>='1' .and. lde_zona<='4'
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ 13,5 say 'Distancia (Kms) :' get lde_kms pict '9999' valid lde_kms >= 0
read
else
  @ 12,5 say 'Zona (1,2,3) :' get lde_zona pict '9' valid lde_zona >= '1' .and. lde_zona <= '3'
  read
endif
dbappend()
repl de_NumDes with lLastNumDes, de_nument with lde_NumEnt
repl de_nomdes with lde_nomdes, de_tipo with lde_tipo
repl de_Zona with lde_Zona, de_Kms with lde_Kms
lLastNumDes++
exit
enddo
cwin(7,2,14,55)
return
//-----
//
// Developed by Ing. Luis Herandez P,rez
//-----
function GetEstado
parameter INumEst,ITipo
local oTbDB := TBrowseDB(11,23,16,48), SelEstado := 0
win(10,22,17,49,1,0)
sele 8
set order to 2
do case
  case ITipo = 'I'
    set filter to ef_nument > 90
  case ITipo = 'N'
    set filter to ef_nument < 90
endcase
//-----Define TBrowse para Altas-----
oTbDB:addColumn(TBColumnNew('Entidad Federativa',({|enf->ef_noment}))
oTbDB:colorSpec := 'w/n,w*/n,w*/n,n/n,n/n'
oTbDB:colsep := '*'
oTbDB:headsep := 'Ã'
oTbDB:freeze := 1
//-----Navegación-----
sele 8
go top
if INumEst > 0
  locate for ef_nument = INumEst
endif
do while LastKey() != K_ESC .and. SelEstado = 0
  while !oTbDB:stabilize()
  enddo
  inkey(0)
  do case
    case LastKey() = K_UP
      oTbDB:up()
    case LastKey() = K_DOWN
      oTbDB:down()
    case LastKey() = K_PGDN
      oTbDB:pageDown()
    case LastKey() = K_PGUP
      oTbDB:pageUp()
    case LastKey() = K_ENTER
      SelEstado := enf->ef_nument
  endcase
enddo
set filter to
cwin(10,22,17,49)
set order to 1
sele 2
return SelEstado
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
//-----  
// Tarifas  
//- Program : VIA402.prg  
//- Developed by Ing. Luis Hernández Pérez  
//-----  
#ifndef K_ESC  
#include "InKey.ch"  
#endif  
  
local lbBorrado := .f., Texto := " Tarifas "  
local oTbDB := TBrowseDB(7,6,19,74)  
set delete on  
set cursor off  
//-----Define pantalla-----  
win(6,5,20,75,2,0)  
@ 6,60 say Texto  
  
set color to n/w  
@ 24,0 say 'F2-Reporte Ins-Agregar Supr-Borrar <Ú-Editar Esc-Salir'  
set color to w/n  
//-----Relaciona archivos-----  
sele 7  
//-----Define TBrowse para Altas-----  
oTbDB:addColumn(TBColumnNew('Nivel' ,({|tar->nivel}))  
oTbDB:addColumn(TBColumnNew('Descripcion' ,({|tar->categoria}))  
oTbDB:addColumn(TBColumnNew('Zona I minimo' ,({|tar->men_z1}))  
oTbDB:addColumn(TBColumnNew('Zona II minimo' ,({|tar->men_z2}))  
oTbDB:addColumn(TBColumnNew('Zona III minimo' ,({|tar->men_z3}))  
oTbDB:addColumn(TBColumnNew('Zona IV minimo' ,({|tar->men_z4}))  
oTbDB:addColumn(TBColumnNew('Zona I' ,({|tar->may_z1}))  
oTbDB:addColumn(TBColumnNew('Zona II' ,({|tar->may_z2}))  
oTbDB:addColumn(TBColumnNew('Zona III' ,({|tar->may_z3}))  
oTbDB:addColumn(TBColumnNew('Zona IV' ,({|tar->may_z4}))  
oTbDB:addColumn(TBColumnNew('Zona I Internacional' ,({|tar->int_z1}))  
oTbDB:addColumn(TBColumnNew('Zona II Internacional' ,({|tar->int_z2}))  
oTbDB:addColumn(TBColumnNew('Zona III Internacional' ,({|tar->int_z3}))  
oTbDB:colorSpec := 'w/n,n/w,w*/n,n/w,n/w'  
oTbDB:colsep := '*'  
oTbDB:headsep := 'Á'  
// oTbDB:freeze := 1  
//-----Navegación-----  
CurRec = RecNo()  
go top  
do while LastKey() != K_ESC  
while !oTbDB:stabilize()  
enddo  
inkey(0)  
do case  
// oTbDB[2]:refreshAll()  
case LastKey() = K_UP  
oTbDB:up()  
case LastKey() = K_DOWN  
oTbDB:down()  
case LastKey() = K_RIGHT  
oTbDB:right()  
case LastKey() = K_LEFT  
oTbDB:left()  
case LastKey() = K_PGDN  
oTbDB:pageDown()  
case LastKey() = K_PGUP  
oTbDB:pageUp()  
case LastKey() = K_INS  
set cursor on  
win(7,2,10,55,1,0)  
LocNivel := 0
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ 8,5 say 'Nivel .' get LocNivel pict '9999'
read
if LastKey() = K_ESC .or. LocNivel = 0
  keyboard chr(K_F9)
  inkey()
endif
seek LocNivel
if found()
  mensa('Nivel ya existente, presione cualquier tecla .')
else
  dbappend()
  repl nivel with LocNivel
endif
cwin(7,2,10,55)
oTbDB:configure()
set cursor off
case LastKey() = K_DEL
  if delete()
    recall
  else
    delete
  endif
  oTbDB:configure()
  lbBorrado = .t.
case LastKey() = K_ENTER
  set cursor on
  cCampo = fieldName(oTbDB:colPos)
  nPosicion = oTbDB:colpos
  oColumna = oTbDB:getColumn(nPosicion)
  @ oTbDB:nTop + oTbDB:rowPos + 1, col() get &cCampo pict '@'
  read
  if LastKey() = K_ESC
    keyboard chr(K_F9)
    inkey()
  endif
  oTbDB:setColumn(nPosicion,oColumna)
  oTbDB:configure()
  set cursor off
case LastKey() = K_F2
  RepTarifas(1)
  RepTarifas(2)
  RepTarifas(3)
endcase
enddo
goto CurRec
if lbBorrado
  @ 23,10 say 'Favor de esperar, se estan borrando los registros marcados...'
  pack
  do borr23
endif
set cursor on
set delete off
cwin(6,5,20,75)
if LastKey() = K_ESC
  keyboard chr(K_F9)
  inkey()
endif
sele 8
set order to 1
sele 2
set order to 2
sele 3
set order to 1
sele 4
set order to 1
sele 1
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
return
//-----
//- Reporte
//-----
Function RepTarifas
PARAMETER locTabla

do chimpre
if LastKey() = K_ESC
do borr23
return
endif
set devi to print
@ 0,0 say chr(18) //-----letras GRANDOTOTAS -----
GO TOP
do while !eof()
@ 1,1 say 'NONBRE DE LA INSTITUCIÓN'
@ 2,1 SAY 'AREA RESPONSABLE'
do case
case locTabla = 1
@ 3,1 say 'TARIFA DE VIATICOS NACIONALES'
@ 4,1 SAY 'PARA PERSONAL COMISIONADO POR UN TIEMPO MENOR DE 24 HORAS'
@ 5,1 SAY '(CUOTA DIARIA EN NUEVOS PESOS)'
case locTabla = 2
@ 3,1 say 'TARIFA DE VIATICOS NACIONALES'
@ 4,1 SAY 'PARA PERSONAL COMISIONADO POR UN TIEMPO MAYOR A 24 HORAS'
@ 5,1 SAY '(CUOTA DIARIA EN NUEVOS PESOS)'
case locTabla = 3
@ 3,1 say 'TARIFA DE VIATICOS INTERNACIONALES'
@ 5,1 SAY '(CUOTA DIARIA EN DOLARES)'
endcase
@ 5,60 say 'Fecha : '+dtoc(p_Date)
@ 6,0 say replicate('-',78)
@ 7,1 say 'NIVEL JERARQUICO'
@ 7,37 SAY ' NIVEL I'
@ 7,47 SAY ' NIVEL II'
@ 7,57 SAY ' NIVEL III'
@ 7,67 SAY 'IF(locTabla=3,',' NIVEL IV)'
@ 8,0 say replicate('-',78)
lLine =9
do while !eof() .and. lLine < 60
@ lLine,1 say tar->Categoria
do case
case locTabla = 1
for i=1 to 4
ci = str(i,1)
@ lLine,(i-1)*10+37 say men_Z&ci pict '999,999.99'
next
case locTabla = 2
for i=1 to 4
ci = str(i,1)
@ lLine,(i-1)*10+37 say may_Z&ci pict '999,999.99'
next
case locTabla = 3
for i=1 to 3
ci = str(i,1)
@ lLine,(i-1)*10+37 say int_Z&ci pict '999,999.99'
next
endcase
lLine++
skip
enddo
enddo
@ ++lLine,0 say replicate('-',78)
@ lLine+3,1 say 'LAS LLAMADAS TELEFONICAS PERSONALES ESTAN INCLUIDAS EN LA TARIFA GLOBAL'
@ lLine+4,1 say ''
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
set dev to screen
do borr23
return

//-----
// Plantilla
//- Program . VIA403.prg
//- Developed by Ing. Luis Hernández Pérez
//-----
#ifndef K_ESC
#include "InKey.ch"
#endif
local lbBorrado := .f., Texto := ' Plantilla '
local oTbDB := TBrowseDB(7,6,19,74)
set delete on
set cursor off
//-----Define pantalla-----
win(6,5,20,75,2,0)
@ 6,60 say Texto
set color to n/w
@ 24,0 say 'F2-Reporte Ins-Agregar Supr-Borrar <U-Editar Esc-Salir'
set color to w/n
//-----Relaciona archivos-----
sele 4
//-----Define TBrowse para Atlas-----
oTbDB:addColumn(TBColumnNew('No. Emp.' ,{||rh1->numemp}))
oTbDB:addColumn(TBColumnNew('Apellido Paterno',{||rh1->paterno}))
oTbDB:addColumn(TBColumnNew('Apellido Materno',{||rh1->materno}))
oTbDB:addColumn(TBColumnNew('Nombre' ,{||rh1->nombre}))
oTbDB:addColumn(TBColumnNew('No. Area' ,{||rh1->numare}))
oTbDB:addColumn(TBColumnNew('No. Puesto' ,{||rh1->puesto}))
oTbDB:addColumn(TBColumnNew('No. Nivel' ,{||rh1->nivel}))
oTbDB:addColumn(TBColumnNew('Tel.fono/Ext' ,{||rh1->telex}))
oTbDB:colorSpec := 'w/n,n/w,w/n,n,w/n,w'
oTbDB:colsep := ' '
oTbDB:headsep := 'A'
// oTbDB:freeze := 1
//-----Navegación-----
CurRec = REcNo()
go top
do while LastKey() != K_ESC
while !oTbDB:stabilize()
enddo
inkey(0)
do case
// oTbDB[2]:refreshAll()
case LastKey() = K_UP
oTbDB:up()
case LastKey() = K_DOWN
oTbDB:down()
case LastKey() = K_RIGHT
oTbDB:right()
case LastKey() = K_LEFT
oTbDB:left()
case LastKey() = K_PGDN
oTbDB:pageDown()
case LastKey() = K_PGUP
oTbDB:pageUp()
case LastKey() = K_INS
set cursor on
win(7,2,10,55,1,0)
LocNivel := 0
@ 8,5 say 'Número de Empleado : ' get LocNivel pict '999999'
read
if LastKey() = K_ESC .or. LocNivel = 0
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Clipper)

```
        keyboard chr(K_F9)
        inkey()
    endif
    seek LocNivel
    if found()
        mensa("Emplado ya existente, presione cualquier tecla ...")
    else
        dbappend()
    #
        append blank
        repl numemp with LocNivel
    endif
    cwin(7,2,10,55)
    oTbDB:configure()
    set cursor off
    case LastKey() = K_DEL
        if delete()
            recall
        else
            delete
        endif
        oTbDB:configure()
        lbBorrado = .t.
    case LastKey() = K_ENTER
        set cursor on
        cCampo = FieldName(oTbDB:colPos+1)
        altd()
        nPosicion = oTbDB:colpos
        oColumna = oTbDB:getColumn(nPosicion)
        @ oTbDB:nTop + oTbDB:rowPos+1, col() get &cCampo
    ###
        read
        if LastKey() = K_ESC
            keyboard chr(K_F9)
            inkey()
        endif
        oTbDB:setColumn(nPosicion,oColumna)
        oTbDB:configure()
        set cursor off
    case LastKey() = K_F2
        RepPlantilla()
    endcase
enddo
goto CurRec
if lbBorrado
    @ 23,10 say "Favor de esperar, se estan borrando los registros marcados..."
    pack
    do borr23
endif
set cursor on
set delete off
cwin(6,5,20,75)
if LastKey() = K_ESC
    keyboard chr(K_F9)
    inkey()
endif
sele 8
set order to 1
sele 2
set order to 2
sele 3
set order to 1
sele 4
set order to 1
sele 1
return
//-----
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
// Reporte de plantilla
//
Function RepPlantilla
do chimpre
if LastKey() = K_ESC
do borr23
return
endif
set devi to print
@ 0,0 say chr(15) //-----letras peques-----
set order to 2
GO TOP
do while !eof()
@ 1,1 say 'NOMBRE DE LA INSTITUCIÓN'
@ 2,1 say 'AREA RESPONSABLE'
@ 3,1 say 'REPORTE DE PLANTILLA DE PERSONAL'
@ 3,90 say 'Fecha : '+dtoc(p_Date)
@ 4,0 say replicate('-',106)
@ 5,1 say 'No. Emp.'
@ 5,10 say 'Nombre'
@ 5,70 say 'Area'
@ 5,78 say 'Puesto'
@ 5,85 say 'Nivel'
@ 5,91 say 'Tel.fono/Ext.'
@ 6,0 say replicate('-',106)
lLine = 7
do while !eof() .and. lLine < 60
@ lLine, 1 say rh1->numemp
@ lLine,10 say alltrim(rh1->paterno)+' '+alltrim(rh1->materno)+' '+alltrim(rh1->Nombre)
@ lLine,70 say rh1->numare
@ lLine,78 say rh1->puesto
@ lLine,85 say rh1->nivel
@ lLine,91 say rh1->telex
lLine++
skip
enddo
@ lLine+1,0 say ''
set devi to screen
do borr23
set order to 1
return
//-----
// Puestos
// Program VIA404.prg
// Developed by Ing. Luis Hernández Pérez
//-----
#ifdef K_ESC
#include "InKey.ch"
#endif
local lBorrado := .f., Texto := " Puestos "
local oTbDB := TBrowseDB(7,6,19,74)
set delete on
set cursor off
//-----Define pantalla-----
win(6,5,20,75,2,0)
@ 6,60 say Texto
set color to n/w
@ 24,0 say 'F2-Reporte Ins-Agregar Supr-Borrar <Ú-Editar Esc-Salir'
set color to w/n
//-----Relaciona archivos-----
sele 5
//-----Define TBrowse para Altas-----
oTbDB::addColumn(TBColumnNew('No. Puesto' ,{(lRh2->puesto)}))
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Clipper)

```
oTbDB:addColumn(TBColumnNew('Nombre del Puesto' ,{|rh2->nompues}))
oTbDB:colorSpec := 'w/n,n/w,w'/n,n/w,n/w
oTbDB:colsep := ' '
oTbDB:headsep := 'A'
// oTbDB:freeze := 1
//-----Navegación-----
CurRec = REcNo()
go top
do while LastKey() != K_ESC
  while !oTbDB:stabilize()
    enddo
  inkey(0)
  do case
//    oTbDB[2]:refreshAll()
    case LastKey() = K_UP
      oTbDB:up()
    case LastKey() = K_DOWN
      oTbDB:down()
    case LastKey() = K_RIGHT
      oTbDB:right()
    case LastKey() = K_LEFT
      oTbDB:left()
    case LastKey() = K_PGDN
      oTbDB:pageDown()
    case LastKey() = K_PGUP
      oTbDB:pageUp()
    case LastKey() = K_INS
      set cursor on
      win(7,2,10,55,1,0)
      LocNivel := 0
      @ 8,5 say 'No. Puesto .' get LocNivel pict '9999'
      read
      if LastKey() = K_ESC .or. LocNivel = 0
        keyboard chr(K_F9)
        inkey()
      endif
      seek LocNivel
      if found()
        mensa('Nivel ya existente, presione cualquier tecla ...')
      else
        dbappend()
        repl puesto with LocNivel
      endif
      cwin(7,2,10,55)
      oTbDB:configure()
      set cursor off
    case LastKey() = K_DEL
      if delete()
        recall
      else
        delete
      endif
      oTbDB:configure()
      lbBorrado = .t.
    case LastKey() = K_ENTER
      set cursor on
      cCampo = fieldName(oTbDB:colPos+1)
      nPosicion = oTbDB:colpos
      oColumna = oTbDB:getColumn(nPosicion)
      @ oTbDB:nTop + oTbDB:rowPos + 1, col() get &cCampo pict '@!'
      read
      if LastKey() = K_ESC
        keyboard chr(K_F9)
        inkey()
      endif
      oTbDB:setColumn(nPosicion,oColumna)
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes. (Clipper)

```
oTbDB:configure()
set cursor off
case LastKey() = K_F2
  RepPuestos()
endcase
enddo
goto CurRec
if lbBorrado
  @ 23,10 say 'Favor de esperar, se estan borrando los registros marcados...'
  pack
  do borr23
endif
set cursor on
set delete off
cwin(6,5,20,75)
if LastKey() = K_ESC
  keyboard chr(K_F9)
  inkey()
endif
sele 8
set order to 1
sele 2
set order to 2
sele 3
set order to 1
sele 4
set order to 1
sele 1
return
//-----
//- Reporte de Puestos
//-----
Function RepPuestos
do chimpre
if LastKey() = K_ESC
do borr23
return
endif
set devi to print
@ 0,0 say chr(18) //-----letras GRANDOTOTAS-----
GO TOP
do while !eof()
  @ 1,1 say 'NONBRÉ DE LA INSTITUCIÓN'
  @ 2,1 SAY 'AREA RESPONSABLE'
  @ 3,1 say 'REPORTE DE PUESTOS'
  @ 3,55 say 'Fecha :'+dtoc(p_Date)
  @ 4,0 say replicate('-',72)
  @ 5,1 SAY 'No. Puesto'
  @ 5,12 say 'Descripción'
  @ 6,0 say replicate('-',72)
  lLine = 7
do while !eof() .and, lLine < 60
  @ lLine, 1 say rh2->puesto
  @ lLine,12 say rh2->nompues
  lLine++
  skip
enddo
enddo
@ lLine+1,0 say ''
set devi to screen
do borr23
return
//-----
// Areas
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
//- Program : VIA405.prg
//- Developed by Ing. Luis Hernández Pérez
//-----
#ifndef K_ESC
#include "InKey.ch"
#endif
local lbBorrado := .f., Texto := " Areas "
local oTbDB := TBrowseDB(7,6,19,74)
set delete on
set cursor off
//-----Define pantalla-----
win(6,5,20,75,2,0)
@ 6,60 say Texto
set color to n/w
@ 24,0 say 'F2-Reporte Ins-Agregar Supr-Borrar <Ú-Editar Esc-Salir'
set color to w/n
//-----Relaciona archivos-----
sele 6
//-----Define TBrowse para Altas-----
oTbDB:addColumn(TBColumnNew('No. Area',{||rh3->numare}))
oTbDB:addColumn(TBColumnNew('Nombre del Area',{||rh3->nomarea}))
oTbDB:colorSpec := 'w/n,n/w,w*/n,n/w,n/w'
oTbDB:colsep := ' '
oTbDB:headsep := 'À'
// oTbDB:freeze := 1
//-----Navegación-----
CurRec = RecNo()
go top
do while LastKey() != K_ESC
while !oTbDB:stabilize()
enddo
inkey(0)
do case
// oTbDB[2]:refreshAll()
case LastKey() = K_UP
oTbDB:up()
case LastKey() = K_DOWN
oTbDB:down()
case LastKey() = K_RIGHT
oTbDB:right()
case LastKey() = K_LEFT
oTbDB:left()
case LastKey() = K_PGDN
oTbDB:pageDown()
case LastKey() = K_PGUP
oTbDB:pageUp()
case LastKey() = K_INS
set cursor on
win(7,2,10,55,1,0)
LocNivel := 0
@ 8,5 say 'No. Puesto ' get LocNivel pict '9999'
read
if LastKey() = K_ESC .or. LocNivel = 0
keyboard chr(K_F9)
inkey()
endif
seek LocNivel
if found()
mensa('Nivel ya existente, presione cualquier tecla ...')
else
dbappend()
repl puesto with LocNivel
endif
cwin(7,2,10,55)
oTbDB:configure()
set cursor off
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
case LastKey() = K_DEL
  if delete()
    recall
  else
    delete
  endif
  oTbDB:configure()
  lbBorrado = .t.
case LastKey() = K_ENTER
  set cursor on
  cCampo = FieldName(oTbDB:colPos)
  nPosicion = oTbDB:colPos
  oColumna = oTbDB:getColumn(nPosicion)
  @ oTbDB:nTop + oTbDB:rowPos + 1, col() get &cCampo pict '@'
  read
  if LastKey() = K_ESC
    keyboard chr(K_F9)
    inkey()
  endif
  oTbDB:setColumn(nPosicion,oColumna)
  oTbDB:configure()
  set cursor off
case LastKey() = K_F2
  RepAreas()
endcase
enddo
goto CurRec
if lbBorrado
  @ 23,10 say 'Favor de esperar, se estan borrando los registros marcados...'
  pack
  do borr23
endif
set cursor on
set delete off
cwin(6,5,20,75)
if LastKey() = K_ESC
  keyboard chr(K_F9)
  inkey()
endif
sele 8
set order to 1
sele 2
set order to 2
sele 3
set order to 1
sele 4
set order to 1
sele 1
return
//-----
// Reporte de Areas
//-----
function RepAreas
do chimpre
if LastKey() = K_ESC
do borr23
return
endif
set devi to print
@ 0,0 say chr(18) //-----letras GRANDOTOTAS-----
GO TOP
do while !eof()
  @ 1,1 say 'NONBRE DE LA INSTITUCIÓN'
  @ 2,1 say 'AREA RESPONSABLE'
  @ 3,1 say 'REPORTE DE AREAS'
  @ 3,55 say 'Fecha : *dtoc(p_Date)
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ 4,0 say replicate('-',72)
@ 5,1 SAY 'No. Area'
@ 5.12 say 'Descripción'
@ 6,0 say replicate('-',72)
Iline = 7
do while eof() .and. Iline < 60
  @ Iline, 1 say rh3->numAre
    @ Iline,12 say rh3->NomArea
  Iline++
  skip
enddo
enddo
@ Iline+1,0 say ''
set devi to screen
do borr23
return
```

```
//-----
// Program : VIA500.prg
// Developed by Ing. Luis Hernández Pérez
// Since Feb 95
//-----
#ifdef K_ESC
#include "inkey.ch"
#endif
set cursor off
//-----Seleciona Archivos-----
CurWork := Select()
sele 9
P := Select()
sele 10
D := Select()
NoParam = LastRec()
//-----Define pantalla-----
win(6,4,16,75,2,0)
@ 6,60 say 'Par metros'
@ 8,43 to 8,70
set color to n/w
@ 24,0 say '<Ú-Editar Esc-Salir'
set color to win
//-----Define TBrowse -----
oTbDB := TBrowseDB(7,5,15,46)
Select(D)
oTbDB:addColumn(TBColumnNew('Concepto',{||ParamDesc}))
oTbDB:colorSpec := 'win,n/w,n/w,n/w,n/w'
oTbDB:colsep := '~'
oTbDB:headsep := 'A'
//-----navegación-----
do while LastKey() != K_ESC
  while !oTbDB:stabilize()
  enddo
  ParamField()
  inkey(0)
  do case
    case LastKey() = K_UP
      @ 9,48 clear to 15,74
      oTbDB:up()
    case LastKey() = K_DOWN
      @ 9,48 clear to 19,74
      oTbDB:down()
    case LastKey() = K_PGDN
      @ 9,48 clear to 15,74
      oTbDB:pageDown()
      oTbDB:Stable := .f.
    case LastKey() = K_PGUP
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ 9,48 clear to 15,74
oTbDB:pageUp()
oTbDB:Stable := .f.
case LastKey() = K_ENTER
  nPosicion = oTbDB:colpos
  oColumna = oTbDB:getColumn(nPosicion)
  l_NoRec := REcno()
  select(P)
  l_FieldName := FieldName(l_NoRec)
  select(D)
  set cursor on
  do case
    case vatype(Param->&l_FieldName) = 'C'
      @ oTbDB:nTop + oTbDB:rowPos + 1, 48 get Param->&l_FieldName pict '@!S25'
    case vatype(Param->&l_FieldName) = 'N'
      @ oTbDB:nTop + oTbDB:rowPos + 1, 48 get Param->&l_FieldName pict '999,999,999.99999999'
    otherwise
      @ oTbDB:nTop + oTbDB:rowPos + 1, 48 get Param->&l_FieldName
  endcase
  read
  set cursor off
  oTbDB:setColumn(nPosicion,oColumna)
  oTbDB:configure()
  oTbDB:Stable := .f.
endcase
enddo
select(P)
dbcommit() // Actualiza la base de datos -----
Select(D)
dbcommit() // Actualiza la base de datos -----
cwin(6,4,16,75)
Select(CurWork)
set cursor on
return

Function ParamField()
local Inicio := Recno()-oTbDB:rowPos+1,;
  Line := oTbDB:nTop + 2
for i = Inicio to Inicio + 6
  Select(P)
  l_FieldName := FieldName(i)
  Select(D)
  @ Line,46 say ':'
  do case
    case vatype(Param->&l_FieldName) = 'C'
      @ Line, 48 say if(len(Param->&l_FieldName)>25,left(Param->&l_FieldName,25),Param->&l_FieldName)
    case vatype(Param->&l_FieldName) = 'N'
      @ Line, 48 say Param->&l_FieldName pict '999,999,999.99999999'
    otherwise
      @ Line, 48 say Param->&l_FieldName
  endcase
  Line++
next
RETURN

//-----
// Reindexa archivos
///- Program : VIA501.prg
///- Developed by Ing. Luis Hernández Pérez
//-----
clear gets
cent('Favor de esperar, reindexando archivos ...',23)
sele 2
reindex
sele 3
```



```
reindex
sele 4
reindex
sele 5
reindex
sele 6
reindex
sele 7
reindex
sele 8
reindex
sele 1
reindex
do borr23
Edit_Mode := "!"
return
```

```
//-----
// Program   : VIA502.prg
// Developed by Ing. Luis Hernández Pérez
// Since Feb 95
//-----
#ifndef K_ESC
#include "inkey.ch"
#endif
set cursor off
//-----Seleciona Archivos-----
CurWork := Select()
sele 9
P := Select()
sele 10
D := Select()
NoParam = LastRec()
//-----Define pantalla-----
win(6,4,16,75,2,0)
@ 6,60 say 'Par metros'
@ 8,43 to 8,74
set color to n/w
@ 24,0 say '<U>-Editar  Esc-Salir'
set color to w/n
//-----Define TBrowse-----
oTbDB := TBrowseDB(7,5,15,46)
Select(D)
oTbDB:addColumn(TBColumnNew('Concepto', (||ParamDesc)))
oTbDB:colorSpec := 'w/n,n/w,n/w,n/w,n/w'
oTbDB:colsep := ""
oTbDB:headsep := 'A'
//-----navegación-----
do while LastKey() != K_ESC
while !oTbDB:stabilize()
enddo
ParamField()
inkey(0)
do case
case LastKey() = K_UP
@ 9,48 clear to 15,74
oTbDB:up()
case LastKey() = K_DOWN
@ 9,48 clear to 15,74
oTbDB:down()
case LastKey() = K_PGDN
@ 9,48 clear to 15,74
oTbDB:pageDown()
oTbDB:Stable := .f.
case LastKey() = K_PGUP
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
@ 9,48 clear to 15,74
oTbDB:pageUp()
oTbDB:Stable = .f.
case LastKey() = K_ENTER:
  nPosicion = oTbDB:colpos
  oColumna = oTbDB:getColumn(nPosicion)
  l_NoRec := REcno()
  select(P)
  l_FieldName := FieldName(l_NoRec)
  select(D)
  set cursor on
do case
  case valtype(Param->l_FieldName) = 'C'
    @ oTbDB:nTop + oTbDB:rowPos + 1, 48 get Param->l_FieldName pict '@!S25'
  case valtype(Param->l_FieldName) = 'N'
    @ oTbDB:nTop + oTbDB:rowPos + 1, 48 get Param->l_FieldName pict '999,999,999.99999999'
  otherwise
    @ oTbDB:nTop + oTbDB:rowPos + 1, 48 get Param->l_FieldName
endcase
read
set cursor off
oTbDB:setColumn(nPosicion,oColumna)
oTbDB:configure()
oTbDB:Stable = .f.
endcase
enddo
select(P)
dbcommit() // Actualiza la base de datos -----
Select(D)
dbcommit() // Actualiza la base de datos -----
cwin(6,4,16,75)
Select(CurWork)
set cursor on
if LastKey()=K_ESC
  keyboard chr(K_F9)
  inkey()
endif
return

Function ParamField()
local Inicio := Recno()-oTbDB:rowPos+1,
  Line := oTbDB:nTop + 2
for i = Inicio to Inicio + 6
  Select(P)
  l_FieldName := FieldName(i)
  Select(D)
  @ Line,46 say ':'
  do case
    case valtype(Param->l_FieldName) = 'C'
      @ Line, 48 say if(len(Param->l_FieldName)>25,left(Param->l_FieldName,25),Param->l_FieldName)
    case valtype(Param->l_FieldName) = 'N'
      @ Line, 48 say Param->l_FieldName pict '999,999,999.99999999'
    otherwise
      @ Line, 48 say Param->l_FieldName
  endcase
  Line++
next
RETURN
```

```
*****
* Programa: Via600.prg
* Sistema: Viaticos, Estadísticos de Viaticos
* Autor : Ing Luis Hernández Pérez
*****
```

```
#include "Inkey.ch"
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Clipper)

```
td=1
do while .not. eof()
  do case
    case rh2->pue<5
      td=td+1
    endcase
  enddo
inkey(0)
retu
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas Fuentes (Visual Basic)

#### PROGRAMAS FUENTE DE LA VERSIÓN EN VISUAL BASIC

\*frmSolicitud Forma de Solicitud de viaticos del Modulo  
de Viaticos  
\*Autor: Luis Hernández Pérez

\*Tablas:

\*Base de Datos Tablas que Utiliza Lectura Escritura

\*Viaticos Hist\_viat\_emp sql sql

\*Viaticos Cat\_destinos sql

\*Viaticos Cat\_objetivos sql

\*Viaticos Cat\_tarifas sql

\*Viaticos Consecutivo sql sql

\*Viaticos Hist\_bol\_avion\_emp sql sql

\*Viaticos Hist\_dest\_emp sql sql

\*Viaticos Hist\_obj\_cli\_emp sql sql

\*Viaticos Hist\_obj\_del\_emp sql sql

\*Viaticos Hist\_obj\_otro\_emp sql sql

\*Viaticos Hist\_obj\_proy\_emp sql sql

\*Rec. Humanos Datos\_empleado sql

\*Rec. Humanos Cat\_areas sql

\*Rec. Humanos Cat\_estados sql

\*Rec. Humanos Cat\_puestos sql

\*Gestión Clientes sql

\*Gestión Proyectos sql

\*Gestión CatDelegaciones sql

\*Forma(s) que Llama: frmmain, frmdestinos, frmgastos

\*Llamada por: frmmain

\*Modificada por:

\*crea tipo de variable para guardar

\*dinámicamente los objetivos

Dim Isql As String

Dim IObjNoC As Byte 'variable de identificacion Nuevo o Cambio Objeto

Dim lmin As Byte 'variable de identificacion de textboxes fecha

Dim lMin1 As Byte 'variable de identificacion de textboxes hora

Dim IObjrv(20) As ListaObj 'arreglo para la lista de objetivos

Public pRegEmp As rdoResultset 'variable para el query de empleados

Public pRegEmpAut As rdoResultset 'variable para autorizadores

Public pRegEmpSup As rdoResultset 'variable para superiores

Public pRegTari As rdoResultset 'variable para la tabla de tarifas

Public pRegViat As rdoResultset 'variable para el historico de viaticos

Public pintNivel As Integer

Public pintNivelSup As Integer

Public pstrfol As String

Public pstrfecha As String

Private Function FDaFecha(fechaOld As Date) As String

Dim Ianio As String

Dim lmes As String

Dim ldia As String

lanio = Format(DatePart("yyyy", fechaOld), "0000")

lmes = Format(DatePart("m", fechaOld), "00")

ldia = Format(DatePart("d", fechaOld), "00")

FDaFecha = lmes & "/" & ldia & "/" & Ianio

End Function

Private Function FVeriFechas() As Boolean

Dim lFechAux As Date

Dim lFecAct As Date

FVeriFechas = True

Pfechas = txtDiaS.Text + "/" + txtMesS.Text + "/" + txtAnioS.Text

Pfechal = lstDiaL.Text + "/" + txtMesLL.Text + "/" + txtAnioLL.Text

pHoraS = TimeValue(txtHoraS.Text + ":" + txtMinS.Text)

pHoratL = TimeValue(txtHoraL.Text + ":" + txtMinLL.Text)

pNumDias = Pfechal - Pfechas

lFecAct = Date

If (pNumDias < 0) Or ((Pfechas - lFecAct) < 0) Or (Pfechal = Pfechas And



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
(pHoraL - pHoraS <= 0)) Or (Pfechas = IfecAct And (pHoraS - Time) <= 0) _
Then
MsgBox "Periodo de salida incorrecto!!", vbExclamation, "Solicitud"
pNumDias = 0
FVerifFechas = False
Else
If pNumDias >= 5 Then
MsgBox "su periodo de salida es mayor a cinco dias", vbExclamation, "Solicitud"
End If
For IfechAux = Pfechas To Pfechal
If Weekday(IfechAux) = _
vbSunday Or Weekday(IfechAux) = vbSaturday Then
MsgBox "su periodo de salida incluye un fin de semana", vbExclamation, "Solicitud"
pNumDias = -1
Exit For
End If
Next
pNumDias = pNumDias + 1
End If
MsgBox "dias " & pNumDias, vbExclamation, "Solicitud"
End Function
Private Function FVerifSol() As Boolean
FVerifSol = True
If btnEmpAut = "" Or PintDias = 0 Or _
btDiaS = "" Or txtMesS = "" Or txtAnioS = "" Or tstDiaL = "" Or txtMesLI = "" _
Or txtAnioLI = "" Or txtHoraS = "" Or txtMinS = "" Or txtHoraLI = "" Or _
txtMinLI = "" Or lstObj.List(0) = "" Or _
(chkSuper = 1 And btnEmpSup = "") Then
FVerifSol = False
MsgBox "No ha ingresado todos los datos requeridos!!!", vbExclamation, "Solicitud"
End If
End Function
Sub PLimpiaCampos()
datos generales
btFolio.Text = ""
lblFAntici.Caption = ""
btNombre.Text = ""
btpuesto.Text = ""
btArea.Text = ""
btTel.Text = ""
btSal.Text = ""
proyecto
btNEmpAut.Text = ""
cboNomAut.Text = ""
btAreaAut.Text = ""
btPuestAut.Text = ""
btnEmpSup.Text = ""
cmbNomSup.Text = ""
btAreaSup.Text = ""
btPuestSup.Text = ""
cboVisita.Text = ""
btVisita.Text = ""
cboObjNom.Text = ""
btDesObj.Text = ""
End Sub
Sub ButtonNavigateMode()
Define que botones estaran habilitados cuando este por elegirse
nueva solicitud o editar solicitud
SstSol.Enabled = False
cmdSave.Enabled = False
cmdCancel.Enabled = False
cmdAdd.Enabled = True
cmdEdit.Enabled = True
cmdDelete.Enabled = False
cmdFind.Enabled = False
cmdClose.Enabled = True
End Sub
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas Fuentes (Visual Basic)

```
' cmdMovePrevious.Enabled = False
' cmdMoveNext.Enabled = False
End Sub
Sub PModoEdicion()
' habilita solo cancelar
  cmdSave.Enabled = False
  cmdCancel.Enabled = True
  cmdAdd.Enabled = False
  cmdEdit.Enabled = False
  cmdDelete.Enabled = False
  cmdFind.Enabled = True
  cmdClose.Enabled = False
' cmdMovePrevious.Enabled = False
' cmdMoveNext.Enabled = False
  SstSol.Enabled = True
End Sub
Public Function fVerifNumero(txt As TextBox, msg As String) As Boolean
' verifica que el numero de empleado sea un valor numerico
If txt.Text = "" Then 'si no ha introducido ningun dato
  MsgBox ("Debe Introducir " + msg), vbExclamation, "Solicitud"
  fVerifNumero = False
  txt.SetFocus
Elseif Not (IsNumeric(txt.Text)) Then 'si ha introducido un valor numerico
  MsgBox ("Debe Introducir un Número"), vbExclamation, "Solicitud"
  fVerifNumero = False
  txt.Text = ""
  txt.SetFocus
Else
  fVerifNumero = True
End If
End Function
Sub pHabilFolder()
' procedimiento que habilita solo al folder de datos generales
' y lo define como tab actual
  SstSol.TabEnabled(0) = True
  SstSol.TabEnabled(1) = False
  SstSol.TabEnabled(2) = False
  SstSol.TabEnabled(3) = False
  SstSol.Tab = 0
  If MfAppend Then
    txtNEmp.Enabled = True
    txtNEmp.SetFocus
  Else
    txtNEmp.Enabled = False
  End If
  cmdFind.Enabled = True
End Sub
Sub pInicia()
Dim lcad As String
Dim lRegDest As rdoResultset
Dim lrsrhEst As rdoResultset
Dim lRegEdo As rdoResultset
Dim lfecha As Date
Dim lhora As String
Dim lintNItem As Integer
Dim lRegEmpAut As rdoResultset
Dim lRegEmpSup As rdoResultset
Dim lRegHistDest As rdoResultset
Dim lIndex As Integer
If MfAppend Then
' inicializa variables y textboxes con fecha
' y hora actual del sistema para el registro
' de salida y llegada
' inicializa variables y botones para el
' control de objetivos
  chkSuper.Value = False
End If
End Sub
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
'datos de salida al destino
btDiaS.Text = Format(Day(Date), "00")
btMesS.Text = Format(Month(Date), "00")
btAñoS.Text = Format(Year(Date), "00")
btHoraS.Text = Format(Hour(Time), "00")
btMinS.Text = Format(Minute(Time), "00")
'datos de llegada al destino
tstDiaL.Text = Format(Day(Date), "00")
btMesLi.Text = Format(Month(Date), "00")
btAñoLi.Text = Format(Year(Date), "00")
btHoraLi.Text = Format(Hour(Time), "00")
btMinLi.Text = Format(Minute(Time), "00")
'inicializa variables para identificar el
'textbox al cual se le hará el cambio con
'el spinbutton
lmin = 1
lMin1 = 1
'inicializa botones de objetivos
lstObj.Enabled = False
lstObj.Clear
cmdNuevObj.Enabled = True
cmdBorraObj.Enabled = False
cmdCambObj.Enabled = False
cmdRegObj.Enabled = False
fraTipVisita.Enabled = False
cboVisita.Enabled = False
btVisita.Enabled = False
cboObjNom.Enabled = False
btDesObj.Enabled = False
lObjNoC = 0
' otras variables
pNumDias = 0
PintDias = 0
'gastos y transportacion
'pcosto_gas = 0
pKm = 0
pBus = 0
pPeaje = 0
pFoto = 0
Psciaarea = ""
pLegales = 0
pOtros = 0
pltinAvion = ""
pltinBus = ""
pltinCarro = ""
pFechaAvion = ""
pHoraAvion = ""
pObs = ""
pRecMat = False
Else
'inicializa variables y textboxes
'inicializa variables y botones para el
'control de objetivos
'recuerde que cmdEmp inicia los datos generales
'datos de autorizador y superior
btNEmpAut.Text = pRegViat("nsi_num_empaut")
'Dim IRegEmpAut As rdoResultset
'Dim IRegEmpSup As rdoResultset
'Dim IRegHistDest As rdoResultset
lsqj = "select distinct datos_employado.nsi_num_emp,datos_employado.vch_ap_paterno," + _
"datos_employado.vch_ap_materno, datos_employado.vch_nombre,cat_puestos.vch_nom_puesto," + _
"cat_areas.vch_nom_area,datos_personales.ni_num_area from datos_employado," + _
"datos_personales, cat_puestos, cat_areas where datos_personales.ni_num_area = cat_areas.ni_num_area" + _
"and datos_employado.nsi_num_puesto=cat_puestos.nsi_num_puesto and datos_personales.nsi_num_emp" + _
"=datos_employado.nsi_num_emp and datos_employado.nsi_num_emp=" & btNEmpAut
'Set IRegEmpAut = pDbRechum.OpenRecordset(lsqj, dbOpenDynaset, dbSQLPassThrough)
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Set IRegEmpAut = bdrdorh.OpenResultset(Name:=!sql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
cboNomAut.Text = _
    Trim(IRegEmpAut("vch_ap_paterno")) +
    " " + Trim(IRegEmpAut("vch_ap_materno")) + _
    " " + Trim(IRegEmpAut("vch_nombre"))
btPuestAut.Text = IRegEmpAut("vch_nom_puesto")
btAreaAut.Text = IRegEmpAut("vch_nom_area")
IRegEmpAut.Close
If pRegViat("nsi_num_empsup") = 0 Then
    chkSuper.Value = False
Else
    chkSuper.Value = 1
    btnEmpSup.Text = pRegViat("nsi_num_empsup")
    !sql = "select distinct datos_empleado.nsi_num_emp,datos_empleado.vch_ap_paterno," + _
        "datos_empleado.vch_ap_materno, datos_empleado.vch_nombre,cat_puestos.vch_nom_puesto," + _
        "cat_puestos.nsi_nivel,cat_areas.vch_nom_area,datos_personales.ni_num_area from datos_empleado," + _
        "datos_personales, cat_puestos, cat_areas where datos_personales.ni_num_area = cat_areas.ni_num_area" + _
        "and datos_empleado.nsi_num_puesto=cat_puestos.nsi_num_puesto and datos_personales.nsi_num_emp" + _
        "=datos_empleado.nsi_num_emp and datos_empleado.nsi_num_emp=" & btnEmpSup
    Set IRegEmpSup = pDbRechum.OpenRecordset(!sql, dbOpenDynaset, dbSQLPassThrough)
    Set IRegEmpSup = bdrdorh.OpenResultset(Name:=!sql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
    'cmbNomSup.Text =
    If IsNull(IRegEmpSup!\vch_ap_materno) Then
        cmbNomSup.Text = _
            Trim(IRegEmpSup("vch_ap_paterno")) + _
            " " + Trim(IRegEmpSup("vch_nombre"))
    Else
        cmbNomSup.Text = _
            Trim(IRegEmpSup("vch_ap_paterno")) +
            " " + Trim(IRegEmpSup("vch_ap_materno")) + _
            " " + Trim(IRegEmpSup("vch_nombre"))
    End If
    btPuestSup.Text = IRegEmpSup("vch_nom_puesto")
    btAreaSup.Text = IRegEmpSup("vch_nom_area")
    pintNivelSup = IRegEmpSup("nsi_nivel")
    IRegEmpSup.Close
End If
'datos de destino
pNumDias = pRegViat!nsi_pdias
PintDias = pNumDias
!fecha = pRegViat("d_pfecha_sal")
btDiaS.Text = Format(DatePart("d", !fecha), "00")
btMesS.Text = Format(DatePart("m", !fecha), "00")
btAnioS.Text = Format(DatePart("yyyy", !fecha), "00")
!hora = pRegViat("ch_phora_sal")
btHoraS.Text = Format(DatePart("h", !hora), "00")
btMinS.Text = Format(DatePart("n", !hora), "00")
!fecha = pRegViat("d_pfecha_lle")
tstDiaL.Text = Format(DatePart("d", !fecha), "00")
btMesLl.Text = Format(DatePart("m", !fecha), "00")
btAnioLl.Text = Format(DatePart("yyyy", !fecha), "00")
!hora = pRegViat("ch_phora_lle")
btHoraLl.Text = Format(DatePart("h", !hora), "00")
btMinLl.Text = Format(DatePart("n", !hora), "00")
If pNumDias = 1 Then
    ReDim pDest(1)
Else
    ReDim pDest(pNumDias - 1)
End If
!sql = "select nsi_pnum_dest,vch_pdestino from hist_dest_emp where" + _
"s_folio=" & txtFolio
Set IRegHistDest = pDbViat.OpenRecordset(!sql, dbOpenDynaset)
Set IRegHistDest = bdrdoviat.OpenResultset(Name:=!sql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Index = 0
While Not IRegHistDest.EOF
    Isql = "select nsi_num_edo,vch_nom_dest,ch_zona,ch_tipo " +
    "from cat_destinos where cat_destinos.nsi_num_dest=" & IRegHistDest!nsi_pnum_dest
    'Set IRegDest = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
    Set IRegDest = bdrdoviat.OpenResultset(Name:=Isql, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
    'rrr
    Isql = "select cat_estados.vch_estado from cat_estados "
    Isql = Isql & "where cat_estados.nsi_num_edo="
    Isql = Isql & IRegDest!nsi_num_edo
    Set IRegEdo = bdrdorh.OpenResultset(Name:=Isql, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
    'Set IRegEdo = pDbRechum.OpenRecordset(Isql, dbOpenDynaset)
    IRegDest.MoveFirst
    pintEdo = IRegDest!nsi_num_edo
    With pDest(IIndex)
        .INDest = IRegHistDest!nsi_pnum_dest
        .JDest = IRegDest!vch_nom_dest
        .IZona = IRegDest!ch_zona
        .ITipo = IRegDest!ch_tipo
        .INEdo = pintEdo
        .JEdo = IRegEdo!vch_estado
    End With
    If IsNull(IRegHistDest!vch_pdestino) Then
        pDest(IIndex).IDescDest = ""
    Else
        pDest(IIndex).IDescDest = IRegHistDest!vch_pdestino
    End If
    IRegDest.Close
    IRegEdo.Close
    IRegHistDest.MoveNext
    IIndex = IIndex + 1
Wend
IRegHistDest.Close
'gastos y transportacion
pFoto = pRegViat!m_pvia_foto
Psciaerea = pRegViat!vch_pvia_ciarea
pLegales = pRegViat!m_pvia_legal
If IsNull(pRegViat!m_costo_gas) Then
    pLtrGas = 0
Else
    pLtrGas = pRegViat!m_costo_gas
End If
pOtros = pRegViat!m_pvia_otros
If IsNull(pRegViat!vch_via_obsrv) Then
    pObs = ""
Else
    pObs = pRegViat!vch_via_obsrv
End If
If IsNull(pRegViat!vch_itin_avion) Then
    pItinAvion = ""
    pFechaAvion = ""
    pHoraAvion = ""
Else
    pItinAvion = pRegViat!vch_itin_avion
    pFechaAvion = pRegViat!vch_fec_avion_sal
    pHoraAvion = pRegViat!vch_hora_avion_sal
End If
If IsNull(pRegViat!vch_itin_bus) Then
    pItinBus = ""
    pBus = 0
Else
    pItinBus = pRegViat!vch_itin_bus
    pBus = pRegViat!m_pvia_bus
End If
```



## MIGRACION DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
If IsNull(pRegViat!vch_ritin_auto) Then
    pItinCarro = ""
    pKm = 0
    pcosto_gas = 0
    pPeaje = 0
    pRecMat = False
Else
    pItinCarro = pRegViat!vch_ritin_auto
    pKm = pRegViat!nsi_km
    pcosto_gas = pRegViat!nsi_costo_gas
    pPeaje = pRegViat!m_pvvia_peaje
    If pRegViat!ch_auto_recmat = "S" Then llevara auto de INSTITUCION
        pRecMat = True
    Else
        pRecMat = False
    End If
End If
' inicializa variables para identificar el
' textbox al cual se le hara el cambio con
' el spinbutton
iMin = 1
iMin1 = 1
' objetivos
Dim Lcont As Byte
Dim iNItems As Integer
Dim IRegVisitas As rdoResultset
' Dim IRegNombre As Recordset
Dim IRegObj As rdoResultset
IstObj.Clear
Isql = "select * from hist_obj_proy_emp where s_folio=" & txtFolio
'Set IRegVisitas = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
Set IRegVisitas = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
Lcont = 0
While Not IRegVisitas.EOF
    With IObjetiv(Lcont)
        .ITipo = "P"
        .IstrCve = IRegVisitas!ch_num_proy
        .INObj = IRegVisitas!nsi_num_obj
        .IDesc_corta_obj = IRegVisitas!vch_desc_corta_obj ' lhp
    End With
    If IsNull(IRegVisitas!vch_desc_obj) Then
        IObjetiv(Lcont).IDescObj = ""
    Else
        IObjetiv(Lcont).IDescObj = IRegVisitas!vch_desc_obj
    End If
    Isql = "select vch_nom_corto from cat_objetivos " + _
        "where nsi_num_obj=" & IObjetiv(Lcont).INObj
    'Set IRegObj = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
    Set IRegObj = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
    IObjetiv(Lcont).IDObj = IRegObj!vch_nom_corto
    IObjetiv(Lcont).IDesc = IObjetiv(Lcont).IstrCve
    IRegVisitas.MoveNext
    Lcont = Lcont + 1
Wend
Isql = "select * from hist_obj_cli_emp where s_folio=" & txtFolio
'Set IRegVisitas = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
Set IRegVisitas = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
Lcont = 0
While Not IRegVisitas.EOF
    With IObjetiv(Lcont)
        .ITipo = "C"
        .IintCve = IRegVisitas!ni_cliente
        .INObj = IRegVisitas!nsi_num_obj
    End With
End If
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
.IDesc_corta_obj = IRegVisitas!vch_desc_corta_obj ' lhp
End With
If IsNull(IRegVisitas!vch_desc_obj) Then
  IObjetiv(Lcont).IDescObj = ""
Else
  IObjetiv(Lcont).IDescObj = IRegVisitas!vch_desc_obj
End If
Isql = "select vch_nom_corto from cat_objetivos " + _
      "where nsi_num_obj=" & IObjetiv(Lcont).INObj
Set IRegObj = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
Set IRegObj = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
IOjetiv(Lcont).IDObj = IRegObj!vch_nom_corto
Icad = "select chNombre from Clientes where niCliente=" & _
      IObjetiv(Lcont).IintCve
Set rsViat01 = bdrdogest.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
Isql = "select chNombre from Clientes where niCliente=" & _
      IObjetiv(Lcont).IintCve
Set IRegNombre = pDbGestion.OpenRecordset(Isql, dbOpenDynaset)
IOjetiv(Lcont).IDesc = rsViat01!chnombre
IRegNombre.Close
IOjetiv(Lcont).IDesc = IOjetiv(Lcont).IstrCve
Lcont = Lcont + 1
IRegVisitas.MoveNext
Wend
Isql = "select * from hist_obj_del_emp where s_folio=" & btFolio
Set IRegVisitas = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
Set IRegVisitas = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
Icont = 0
While Not IRegVisitas.EOF
  With IOjetiv(Lcont)
    .ITipo = "D"
    .IintCve = IRegVisitas!ni_num_dele
    .INObj = IRegVisitas!nsi_num_obj
    .IDesc_corta_obj = IRegVisitas!vch_desc_corta_obj ' lhp
  End With
  If IsNull(IRegVisitas!vch_desc_obj) Then
    IObjetiv(Lcont).IDescObj = ""
  Else
    IObjetiv(Lcont).IDescObj = IRegVisitas!vch_desc_obj
  End If
  Isql = "select vch_nom_corto from cat_objetivos " + _
        "where nsi_num_obj=" & IOjetiv(Lcont).INObj
  Set IRegObj = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
  Set IRegObj = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
  IOjetiv(Lcont).IDObj = IRegObj!vch_nom_corto
  Icad = "select chDelegacion from CatDelegaciones where niNumDel=" & _
        IOjetiv(Lcont).IintCve
  Icad = "select chNombre from Clientes where niCliente=" & _
        IOjetiv(Lcont).IintCve

  Set rsViat01 = bdrdogest.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

  Isql = "select chDelegacion from CatDelegaciones where niNumDel=" & _
        IOjetiv(Lcont).IintCve
  Set IRegNombre = pDbGestion.OpenRecordset(Isql, dbOpenDynaset)
  IOjetiv(Lcont).IDesc = rsViat01!chDelegacion
  IRegNombre.Close
  IOjetiv(Lcont).IDesc = IOjetiv(Lcont).IstrCve
  Lcont = Lcont + 1
  IRegVisitas.MoveNext
Wend
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Isql = "select * from hist_obj_otro_emp where s_folio=" & txtFolio
'Set IRegVisitas = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
Set IRegVisitas = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
While Not IRegVisitas.EOF
    With IObjetiv(Lcont)
        .ITipo = "O"
        .IDesc = IRegVisitas!vch_desc_visita
        .INObj = IRegVisitas!nsi_num_obj
        .IDesc_corta_obj = IRegVisitas!vch_desc_corta_obj ' lhp
    End With
    If IsNull(IRegVisitas!vch_desc_obj) Then
        IObjetiv(Lcont).IDescObj = ""
    Else
        IObjetiv(Lcont).IDescObj = IRegVisitas!vch_desc_obj
    End If
    Isql = "select vch_nom_corto from cat_objetivos " + _
        "where nsi_num_obj=" & IObjetiv(Lcont).INObj
    'Set IRegObj = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
    Set IRegObj = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
    IObjetiv(Lcont).IDObj = IRegObj!vch_nom_corto
    Lcont = Lcont + 1
    IRegVisitas.MoveNext
Wend
INItems = pRegViat!nsi_tot_proy - 1
For jintNItem = 0 To INItems
    IstObj.AddItem IObjetiv(jintNItem).IDesc
Next
IstObj.Enabled = True
cmdNuevObj.Enabled = True
cmdBorraObj.Enabled = False
cmdCambObj.Enabled = False
cmdRegObj.Enabled = False
fraTipVisita.Enabled = False
cboVisita.Enabled = False
txtVisita.Enabled = False
cboObjNom.Enabled = False
txtDesObj.Enabled = False
IObjNoC = 0
IRegVisitas.Close
IRegObj.Close
End If
End Sub
Public Sub pLenaSup()
' llena combo de superiores que pueden acompañar
Dim IstrNombre As String
Isql = "select distinct datos_empleado.nsi_num_emp,datos_empleado.vch_ap_paterno," + _
"datos_empleado.vch_ap_materno,datos_empleado.vch_nombre,cat_puestos.vch_nom_puesto," + _
"cat_puestos.nsi_nivel,cat_areas.vch_nom_area,datos_personales.ni_num_area from datos_empleado," + _
"datos_personales, cat_puestos, cat_areas where datos_personales.ni_num_area = cat_areas.ni_num_area" + _
"and datos_empleado.nsi_num_puesto=cat_puestos.nsi_num_puesto and datos_personales.nsi_num_emp" + _
"=datos_empleado.nsi_num_emp and cat_puestos.nsi_nivel<60 and cat_puestos.nsi_nivel<" & pintNivel & " order by " + _
"datos_empleado.vch_ap_paterno,datos_empleado.vch_ap_materno,datos_empleado.vch_nombre"
Set pRegEmpSup = bdrdorh.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
cmbNomSup.Clear
While Not pRegEmpSup.EOF
    If pRegEmpSup!vch_ap_paterno <> "" Then
        IstrNombre = Trim(pRegEmpSup!vch_ap_paterno)
        If pRegEmpSup!vch_ap_materno <> "" Then
            IstrNombre = IstrNombre + " " + Trim(pRegEmpSup!vch_ap_materno)
        End If
        IstrNombre = IstrNombre + " " + Trim(pRegEmpSup!vch_nombre)
    Else
        If pRegEmpSup!vch_ap_materno <> "" Then
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
    IstrNombre = Trim(pRegEmpSup!vch_ap_materno) + " " + Trim(pRegEmpSup!vch_nombre)
Else
    IstrNombre = Trim(pRegEmpSup!vch_nombre)
End If
End If
cmbNomSup.AddItem IstrNombre
pRegEmpSup.MoveNext
Wend
End Sub
Public Sub PllenaAut()
' llena combo de superiores que pueden autorizar
Dim IstrNombre As String
    pintNivel = 10 'se puso para autorizacion del directores
    Isql = "select distinct datos_empleado.nsi_num_emp,datos_empleado.vch_ap_paterno," + _
"datos_empleado.vch_ap_materno, datos_empleado vch_nombre,cat_puestos.vch_nom_puesto," + _
"cat_puestos.nsi_nivel,cat_areas.vch_nom_area,datos_personales.ni_num_area from datos_empleado," + _
"datos_personales, cat_puestos, cat_areas where datos_personales.ni_num_area = cat_areas.ni_num_area " + _
"and datos_empleado.nsi_num_puesto=cat_puestos.nsi_num_puesto and datos_personales.nsi_num_emp" + _
"=datos_empleado.nsi_num_emp and cat_puestos.nsi_nivel<=30 order by
"datos_empleado.vch_ap_paterno,datos_empleado.vch_ap_materno,datos_empleado.vch_nombre"
    Isql = "select distinct datos_empleado.nsi_num_emp,datos_empleado.vch_ap_paterno," + _
"datos_empleado.vch_nombre,cat_puestos.vch_nom_puesto," + _
"cat_puestos.nsi_nivel,cat_areas.vch_nom_area,datos_personales.ni_num_area from datos_empleado," + _
"datos_personales, cat_puestos, cat_areas where datos_personales.ni_num_area = cat_areas.ni_num_area " + _
"and datos_empleado.nsi_num_puesto=cat_puestos.nsi_num_puesto and datos_personales.nsi_num_emp" + _
"=datos_empleado.nsi_num_emp and cat_puestos.nsi_nivel<=30 and cat_puestos.nsi_nivel" + _
"<" & pintNivel & " order by" + _
datos_empleado.vch_ap_paterno,datos_empleado.vch_ap_materno,datos_empleado.vch_nombre"
    cboNomAut.Clear
    Set pRegEmpAut = pDbRechum.OpenRecordset(Isql, dbOpenDynaset, dbSQLPassThrough)
    Set pRegEmpAut = bdrdorh.OpenResultset(Name:=Isql, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
    While Not pRegEmpAut.EOF
        If pRegEmpAut!vch_ap_paterno <> "" Then
            IstrNombre = Trim(pRegEmpAut!vch_ap_paterno)
            If pRegEmpAut!vch_ap_materno <> "" Then
                IstrNombre = IstrNombre + " " + Trim(pRegEmpAut!vch_ap_materno)
            End If
            IstrNombre = IstrNombre + " " + Trim(pRegEmpAut!vch_nombre)
        Else
            If pRegEmpAut!vch_ap_materno <> "" Then
                IstrNombre = Trim(pRegEmpAut!vch_ap_materno) + " " + Trim(pRegEmpAut!vch_nombre)
            Else
                IstrNombre = Trim(pRegEmpAut!vch_nombre)
            End If
        End If
        cboNomAut.AddItem IstrNombre
        pRegEmpAut.MoveNext
    Wend
End Sub
Private Sub cboNomAut_Click()
    pRegEmpAut.MoveFirst
    pRegEmpAut.Move cboNomAut.ListIndex
    If cboNomAut.ListIndex = -1 Then
        MsgBox "Debes seleccionar una Persona", vbExclamation, "Solicitud"
    Else
        txtNempAut.Text = pRegEmpAut!nsi_num_emp
        txtPuestAut.Text = pRegEmpAut!vch_nom_puesto
        txtAreaAut.Text = pRegEmpAut!vch_nom_area
    End If
End Sub
Private Sub cboNomAut_KeyDown(KeyCode As Integer, Shift As Integer)
    MsgBox "ENTRE EN,KEYDOWN"
End Sub
Private Sub cboObjNom_Click(Area As Integer)
    If cboObjNom.Text Like "OTROS" Then
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
lblDescObj.Visible = True
txtDesObj.Visible = True
Else
  lblDescObj.Visible = False
  txtDesObj.Visible = False
End If
txtDesObj.Text = ""
End Sub
Private Sub chkSuper_Click()
'habilita el frame de Superior para el llenado de datos
'si se da un click en este checkbox que lo habilite
If chkSuper.Value Then
  ssfSuper.Enabled = True
  cmbNomSup.SetFocus
  pbolSwitcSup = True
Else
  ssfSuper.Enabled = False
  pbolSwitcSup = False
End If
btnEmpSup.Text = ""
cmbNomSup.Text = ""
txtAreaSup.Text = ""
txtPuestSup.Text = ""
End Sub
Private Sub chkSuper_KeyPress(KeyAscii As Integer)
'habilita el frame de Supervisor para el llenado de datos
'si se presiona una tecla en este checkbox que lo habilite
If chkSuper.Value Then
  ssfSuper.Enabled = True
Else
  ssfSuper.Enabled = False
End If
End Sub
Private Sub cmbNomSup_Click()
Dim lcad As String 'variable para asignar el criterio de busqueda
pRegEmpSup.MoveFirst
pRegEmpSup.Move cmbNomSup.ListIndex
If cmbNomSup.ListIndex = -1 Then
  MsgBox "Debes seleccionar una Persona", vbExclamation, "Solicitud"
Else
  btnEmpSup.Text = pRegEmpSup!nsi_num_emp
  txtPuestSup.Text = pRegEmpSup!vch_nom_puesto
  txtAreaSup.Text = pRegEmpSup!vch_nom_area
  pintNivelSup = pRegEmpSup!nsi_nivel
End If
End Sub
Private Sub cmdAdd_Click()
Dim lsqlQ As String
Dim lintUFolio As Integer
Screen.MousePointer = 11
MfAppend = True
pStatus = "0" 'Esta bandera define que la solicitud aun no esta autorizada
Call PModoEdicion
Dim lbarray As Byte 'variable para el ciclo de etiquetas fecha y folio
'ciclo para actualizar fecha y numero de folio en cada uno de los folders
'segun esto siempre sera fecha actual
lcad = "select * from consecut"

Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
lintUFolio = rsviat01!nsi_cofolio
txtFolio.Text = lintUFolio + 1 'ultimo registro de s_folio?
txtFolio.Enabled = False
lblFAntici.Caption = Format(Date, "dd/mm/yyyy") & "Str(Date)
For lbarray = 1 To 3
  lblFol(lbarray).Caption = pstrfol + txtFolio.Text
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
    IbfFecha(Ibarray).Caption = pstrfecha + Format(Date, "dd/mm/yyyy")
    Format(Date, "dd/mm/yyyy")
Next
'pRegViat.Close
Call pInicio 'procedimiento para inicializar variables y botones
Call pHabilFolder ' procedimiento para solo habilitar folder de datos
Screen.MousePointer = vbNormal
End Sub
Private Sub cmdBorraObj_Click()
Dim IBorra As Byte 'variable para identificar si realmente se borra
If IstObj.ListCount <> 0 Then 'verifica que la lista no este vacia
    IBorra = _
    MsgBox("Esta seguro de borrar este objetivo?", _
vbYesNo, "Borra Objetivo") 'verifica si desea borrar el objetivo
    'si se va a borrar el objetivo
    If IBorra = vbYes Then
        Dim IConta As Integer
        Dim IIndice As Integer
        IIndice = IstObj.ListIndex 'asigna el numero de objetivo que se va a borrar
        IConta = IstObj.ListCount 'asigna el numero total de objetivos
        'si es el unico
        If IIndice + 1 = IConta Then
            With IObjetiv(IIndice)
                .IIntCve = 0
                .IstrCve = ""
                .ITipo = ""
                .INObj = 0
                .IDObj = ""
                .IDesc = ""
                .IDescObj = ""
                .IDesc_corta_obj = ""
            End With
            IstObj.RemoveItem IIndice
        Else
            IstObj.RemoveItem IIndice
            Do 'reordena el arreglo de objetivos
                IObjetiv(IIndice) = _
                IObjetiv(IIndice + 1)
                IIndice = IIndice + 1
            Loop Until IIndice = IConta
            With IObjetiv(IIndice)
                .IIntCve = 0
                .IstrCve = ""
                .ITipo = ""
                .INObj = 0
                .IDObj = ""
                .IDesc = ""
                .IDescObj = ""
                .IDesc_corta_obj = ""
            End With
        End If
        cmdNuevObj.Enabled = True
        cmdBorraObj.Enabled = False
        cmdCambObj.Enabled = False
    End If
Else 'la lista esta vacia
    MsgBox "No hay objetivos que borrar", vbExclamation, "Solicitud"
    cmdNuevObj.Enabled = True
    cmdBorraObj.Enabled = False
    cmdCambObj.Enabled = False
End If
End Sub
Private Sub cmdCambObj_Click()
If IstObj.ListCount <> 0 Then 'verifica que la lista no este vacia
    IObjNoC = 2 'asigna este valor para identificar que la accione es cambio
    'habilita solo los controles para cambiar el objetivo y registrarlos
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
lstObj.Enabled = False
cmdNuevObj.Enabled = False
cmdCambObj.Enabled = False
cmdBorraObj.Enabled = False
cmdRegObj.Enabled = True
fraTipVisita.Enabled = True
cboVisita.Enabled = True
btVisita.Enabled = True
cboObjNom.Enabled = True
btDesObj.Enabled = True
If optOtro Then
    btVisita.SetFocus
Else
    cboVisita.SetFocus
End If
Else 'lista vacia
MsgBox "No hay objetivos que cambiar", vbExclamation, "Solicitud"
cmdNuevObj.Enabled = True
cmdBorraObj.Enabled = False
cmdCambObj.Enabled = False
End If
End Sub
Private Sub cmdCancel_Click()
    Screen.MousePointer = vbHourglass
    Call BtnonNavigateMode
    Call PLimpiaCampos
    ReDim pDest(1)
    lstObj.Clear
    ssfBotones.Refresh
    Screen.MousePointer = vbNormal
End Sub
Private Sub cmdClose_Click()
    Screen.MousePointer = vbHourglass
    Unload Me
    frmMain.Show
    Screen.MousePointer = vbNormal
End Sub
Private Sub pProcDest()
    Dim IstrZon As String
    Dim IstrCampZon As String
    Dim lIndx As Integer
    Dim iRegParam As rdoResultset
    If pbolSwitcSup Then
        pintNivelViat = pintNivelSup 'esto no debe ocurrir con choferes
    Else
        pintNivelViat = pintNivel
    End If
    lIndx = 0
    pAlimnt = 0
    phospedj = 0
    If pNumDias > 1 Then
        While lIndx <= pNumDias - 2
            Select Case pDest(lIndx).iZona
                Case "1"
                    IstrZon = "_z1"
                Case "2"
                    IstrZon = "_z2"
                Case "3"
                    IstrZon = "_z3"
                Case "4"
                    IstrZon = "_z4"
            End Select
            Select Case pDest(lIndx).ITipo
                Case "N"
                    IstrCampZon = "m_may24" + IstrZon
                Case "I"
            End Select
        End While
    End If
End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
IstrCampZon = "m_int" + IstrZon
End Select
' Fecha a partir de la que se maneja nueva tarifa en ( cat_tarifas_nva )
' poner nuevas tarifas en esta tabla
' cuando no se tengan problemas igualar las dos tablas (cat_tarifas) ya
If Pfechas > Pfecha_nvatar Then
    Isql = "select " + IstrCampZon + _
        " from cat_tarifas_nva where nsi_nivel<=" & _
        pintNivelViat
Else
    Isql = "select " + IstrCampZon + _
        " from cat_tarifas where nsi_nivel<=" & _
        pintNivelViat
End If

Set pRegtari = bdrdoviat.OpenResultSet(Name =Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
pRegtari.MoveLast
pDest(IIndx).ICuota = pRegtari(IstrCampZon)
'es la cuota diaria que le corresponde. Valor en tabla tarifas
'con el cambio de reglamento se dejo palimnt con el monto de viaticos y phospedaj=0
Select Case IIndx
Case 0
    If pHoraS > TimeValue("00:00") And pHoraS <= TimeValue("10:00") Then
        pAlimnt = pDest(IIndx).ICuota
    ElseIf pHoraS > TimeValue("10:00") And pHoraS <= TimeValue("16:00") Then
        pAlimnt = pDest(IIndx).ICuota * 0.7
    Else
        pAlimnt = pDest(IIndx).ICuota * 0.3
    End If
Case Else
    pAlimnt = pAlimnt + pDest(IIndx).ICuota
End Select
phospedj = 0
' phospedj = phospedj + pDest(IIndx).ICuota * 0.6
IIndx = IIndx + 1
Wend
IIndx = IIndx - 1
If pHoraL > TimeValue("00:00") And pHoraL <= TimeValue("11:00") Then
    pAlimnt = pAlimnt + pDest(IIndx).ICuota * 0.3

ElseIf pHoraL > TimeValue("11:00") And pHoraL <= TimeValue("20:00") Then
    pAlimnt = pAlimnt + pDest(IIndx).ICuota * 0.7
Else
    pAlimnt = pAlimnt + pDest(IIndx).ICuota
End If
Else
    Select Case pDest(0).IZona
    Case "1"
        IstrZon = "_z1"
    Case "2"
        IstrZon = "_z2"
    Case "3"
        IstrZon = "_z3"
    Case "4"
        IstrZon = "_z4"
    End Select
    Select Case pDest(0).ITipo
    Case "N"
        IstrCampZon = "m_men24" + IstrZon
    Case "I"
        IstrCampZon = "m_int" + IstrZon
    End Select
' Fecha a partir de la que se maneja nueva tarifa en ( cat_tarifas_nva )
' poner nuevas tarifas en esta tabla
' cuando no se tengan problemas igualar las dos tablas (cat_tarifas) ya
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
If Pfecha > Pfecha_nvatar Then
    Isql = "select " + IstrCampZon + _
        " from cat_tarifas_nva where nsi_nivel<=" & _
        pintNivelViat
Else
    Isql = "select " + IstrCampZon + _
        " from cat_tarifas where nsi_nivel<=" & _
        pintNivelViat
End If
'Set pRegtari = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
Set pRegtari = bdrdoivat.OpenResultset(Name:=Isql, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
pRegtari.MoveLast
pDest(0).ICuota = pRegtari(IstrCampZon)
pAlimnt = pDest(0).ICuota
phospedj = 0
End If
pdblCostAuto = (pKm / 5 * pLtrGas)
End Sub
Private Sub cmdDest_Click()
Screen.MousePointer = vbHourglass
If FVeriFechas Then
    frmDestinos.Show
    frmSolicitud.Enabled = False
Else
    Screen.MousePointer = vbNormal
End If
End Sub
Private Sub cmdEdit_Click()
Dim IsqlQ As String
Screen.MousePointer = vbHourglass
MfAppend = False
Call PModoEdicion 'solo boton cancelar
IsqlQ = "select s_folio,d_fech_aut_ant from hist_viat_emp"
'Set pRegViat = pDbViat.OpenRecordset(IsqlQ, dbOpenDynaset)
Set pRegViat = bdrdoivat.OpenResultset(Name:=IsqlQ, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
txtFolio.Enabled = True
txtFolio.SetFocus
Call pHabilFolder ' procedimiento para solo habilitar folder de datos
Screen.MousePointer = vbNormal
End Sub
Private Sub cmdFind_Click()
Dim lcad As String 'variable para asignar el criterio de busqueda
Screen.MousePointer = vbHourglass
Pl_flag_saldo = False
If MfAppend Then 'si nueva solicitud
    If (VerifNumero(txtNEmp, "su número de empleado") Then
        If Pnum_emp = txtNEmp Or Ptipo_usr = "R" Then
            Isql = "select distinct datos_empleado.nsi_num_emp,datos_empleado.vch_ap_paterno," & _
                "datos_empleado.vch_ap_materno," & _
                "datos_empleado.vch_nombre,datos_empleado.vch_ext_tel,cat_puestos.vch_nom_puesto," & _
                "cat_puestos.nsi_nivel,cat_areas.vch_nom_area,datos_personales.ni_num_area from datos_empleado," & _
                "datos_personales, cat_puestos, cat_areas where datos_personales.ni_num_area = cat_areas.ni_num_area" & _
                "and datos_empleado.nsi_num_puesto=cat_puestos.nsi_num_puesto and datos_personales.nsi_num_emp" & _
                "=datos_empleado.nsi_num_emp and datos_empleado.nsi_num_emp=" & txtNEmp
            Set pRegEmp = bdrdorh.OpenResultset(Name:=Isql, _
                Type:=rdOpenDynamic, LockType:=rdConcurValues)
            If pRegEmp.EOF Then 'no lo encuentra
                If Val(txtNEmp) > 9000 And Val(txtnenmp) < 10000 Then
                    MsgBox "Es un empleado por Honorarios", vbExclamation, "Solicitud"
                    lcad = "select distinct cat_honorarios.ch_nombre,cat_honorarios.nsi_nivel, cat_puestos.vch_nom_puesto," & _
                        "cat_areas.vch_nom_area" & _
                        "from cat_honorarios, cat_puestos,cat_areas where cat_honorarios.nsi_num_puesto = cat_puestos.nsi_num_puesto" & _
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
"and cat_honorarios.ni_num_area = cat_areas.ni_num_area and cat_honorarios.nsi_numhono = " & btNEmp - 9000
'lcad = "select distinct cat_honorarios.ch_nombre,cat_honorarios.nsi_nivel, cat_areas.vch_nom_area" + _
"from cat_honorarios, cat_areas where cat_honorarios.ni_num_area = cat_areas.ni_num_area and
cat_honorarios.nsi_numhono = " & btNEmp - 9000

Set rsrh01 = bdrdorh.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsrh01.EOF Then
'debe tener area, puesto y nivel si no aunque exista no lo lee
MsgBox "Número de empleado inexistente", vbExclamation, "Solicitud"
Call pHabilFolder
btNEmp.Text = ""
btNombre.Text = ""
btArea.Text = ""
btpuesto.Text = ""
bttel.Text = ""
btSal.Text = ""
Else
'hp Pnum_area = rsrh01!ni_num_area
btNombre.Text = rsrh01!ch_nombre
'btpuesto.Text = rsrh01!vch_nom_puesto
'btpuesto.Text = "HONORARIOS"
btArea.Text = rsrh01!vch_nom_area
pintNivel = rsrh01!nsi_nivel
Call pLlenaAut
Call pLlenaSup
'habilita los demas floders
Call pcheqa_saldo
If Pt_flag_saldo = False Then
SstSol.TabEnabled(1) = True
SstSol.TabEnabled(2) = True
SstSol.TabEnabled(3) = True
cmdSave.Enabled = True
btNEmp.Enabled = False
End If
End If
Else
MsgBox "Número de empleado inexistente", vbExclamation, "Solicitud"
Call pHabilFolder
btNEmp.Text = ""
btNombre.Text = ""
btArea.Text = ""
btpuesto.Text = ""
bttel.Text = ""
btSal.Text = ""
End If
Else 'si lo encuentra, jala los datos de la tabla
Pnum_area = pRegEmp!ni_num_area
If IsNull(Trim(pRegEmp("vch_ap_materno"))) Then
btNombre.Text =
Trim(pRegEmp("vch_ap_paterno")) +
"" + Trim(pRegEmp("vch_nombre"))
Else
btNombre.Text =
Trim(pRegEmp("vch_ap_paterno")) +
"" + Trim(pRegEmp("vch_ap_materno")) +
"" + Trim(pRegEmp("vch_nombre"))
End If
btpuesto.Text = pRegEmp("vch_nom_puesto")
btArea.Text = pRegEmp("vch_nom_area")
If IsNull(pRegEmp!vch_ext_tel) Then
bttel.Text = ""
Else
bttel.Text = pRegEmp("vch_ext_tel")
End If
pintNivel = pRegEmp("nsi_nivel")
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes. (Visual Basic)

```
'llena el combo de Superior que autoriza y del que acompaña
Call pLlenaAut
Call pLlenaSup
'habilita los demás floders
Call pCheca_saldo
If Pt_flag_saldo = False Then
    SstSol.TabEnabled(1) = True
    SstSol.TabEnabled(2) = True
    SstSol.TabEnabled(3) = True
    cmdSave.Enabled = True
    btnEmp.Enabled = False
    cmdFind.Enabled = False
End If
End If
Eise
MsgBox "El Pasword no corresponde al empleado", vbExclamation, "Solicitud"
btnfolio.SetFocus
btnEmp.Text = ""
btnNombre.Text = ""
btnArea.Text = ""
btnpuesto.Text = ""
btnTel.Text = ""
btnSal.Text = ""
Call pHabilFolder
End If
Eise
btnEmp.Text = ""
btnNombre.Text = ""
btnArea.Text = ""
btnpuesto.Text = ""
btnTel.Text = ""
btnSal.Text = ""
Call pHabilFolder
End If
Eise ' Si Editar solicitud
If !VerifNumero(btnFolio, "el número de folio") Then
    "yyy lcad = "s_folio = " & btnfolio.Text 'asigna el valor del textbox al criterio de sort
    lcad = "select * from hist_viat_emp where s_folio=" & btnFolio.Text
    "yyy pRegViat.findFirst lcad 'busca el criterio"
    Set pRegViat = bdrdoviat.OpenResultSet(Name:=lcad, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
    'If pRegViat.NoMatch Then 'no lo encuentra
    If pRegViat.EOF Then 'no lo encuentra
        MsgBox "Número de folio inexistente", vbExclamation, "Solicitud"
        btnFolio.SetFocus
    ElseIf Not IsNull(pRegViat!d_fech_aut_ant) Then
        MsgBox "La solicitud no puede editarse ya fue autorizada", vbExclamation, "Solitud"
        'la solicitud ya ha sido autorizada y no puede editarse
        btnFolio.SetFocus
    Else
        'si lo encuentra y aun no esta autorizado, jala los datos de la tabla
        lsqQ = " select * from hist_viat_emp where s_folio=" & btnFolio
        Set pRegViat = bdrdoviat.OpenResultSet(Name:=lsqQ, _
        Type:=rdOpenDynamic, LockType:=rdConcurValues)
        'valida que la solicitud sea de la persona de pasword de acceso
        If Pnum_emp = pRegViat!nsi_num_emp Or Ptipo_usr = "R" Then
            'datos empleado
            btnEmp.Text = pRegViat("nsi_num_emp")
            lblFAntici = pRegViat("d_fech_ant")
            Dim lbarry As Byte
            For lbarry = 1 To 3
                LblFol(lbarry).Caption = pstrfol + btnFolio.Text
                lblFecha(lbarry).Caption = pstrfecha + lblFAntici
            Next
        End If
    End If
    If Val(btnEmp) > 9000 And Val(btnemp) < 10000 Then 'Honorarios
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
MsgBox "Es un empleado por Honorarios", vbExclamation, "Solicitud"
icad = "select distinct cat_honorarios.ch_nombre,cat_honorarios.nsi_nivel, cat_puestos.vch_nom_puesto,
cat_areas.vch_nom_area " +
"from cat_honorarios, cat_puestos,cat_areas where cat_honorarios.nsi_num_puesto = cat_puestos.nsi_num_puesto "
+
"and cat_honorarios.ni_num_area = cat_areas.ni_num_area and cat_honorarios.nsi_numhono = " & txtNEmp - 9000
Set rsrh01 = bdrdorh.OpenResultSet(Name:=icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsrh01.EOF Then
'debe tener area, puesto y nivel si no aunque exista no lo lee
MsgBox "Número de empleado inexistente", vbExclamation, "Solicitud"
Call pHabilFolder
txtNEmp.Text = ""
txtNombre.Text = ""
txtArea.Text = ""
txtpuesto.Text = ""
txttel.Text = ""
txtSal.Text = ""
Else
txtNombre.Text = rsrh01!ch_nombre
txtpuesto.Text = rsrh01!vch_nom_puesto
'txtpuesto.Text = "Honorarios"
txtArea.Text = rsrh01!vch_nom_area
pintNivel = rsrh01!nsi_nivel
End If
Else 'empleado normal
Isq1 = "select distinct datos_empleado.nsi_num_emp,datos_empleado.vch_ext_tel,datos_empleado.vch_ap_paterno,"
+
"datos_empleado.vch_ap_materno, datos_empleado.vch_nombre,cat_puestos.vch_nom_puesto," +
"cat_puestos.nsi_nivel,cat_areas.vch_nom_area,datos_personales.ni_num_area from datos_empleado," +
"datos_personales, cat_puestos, cat_areas where datos_personales.ni_num_area = cat_areas.ni_num_area " +
"and datos_empleado.nsi_num_puesto=cat_puestos.nsi_num_puesto and datos_personales.nsi_num_emp" +
"=datos_empleado.nsi_num_emp and datos_empleado.nsi_num_emp=" & txtNEmp
Set pRegEmp = pDbRechum.OpenRecordset(Isq1, dbOpenDynaset, dbSQLPassThrough)
Set pRegEmp = bdrdorh.OpenResultSet(Name:=Isq1, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
pRegEmp.MoveFirst
If IsNull(Trim(pRegEmp("vch_ap_materno"))) Then
txtNombre.Text = _
Trim(pRegEmp("vch_ap_paterno")) + _
"" + Trim(pRegEmp("vch_nombre"))
Else
txtNombre.Text = _
Trim(pRegEmp("vch_ap_paterno")) + _
"" + Trim(pRegEmp("vch_ap_materno")) + _
"" + Trim(pRegEmp("vch_nombre"))
End If
txtpuesto.Text = pRegEmp("vch_nom_puesto")
txtArea.Text = pRegEmp("vch_nom_area")
If IsNull(pRegEmp!vch_ext_tel) Then
txttel = ""
Else
txttel.Text = pRegEmp("vch_ext_tel")
End If
pintNivel = pRegEmp("nsi_nivel")
End If
Call PllenaAut
Call pLlenaSup
Call pinicia
'habilita los demas folders
SetSol.TabEnabled(1) = True
SetSol.TabEnabled(2) = True
SetSol.TabEnabled(3) = True
cmdSave.Enabled = True
txtNEmp.Enabled = False
txtFolio.Enabled = False
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuente (Visual Basic)

```
cmdFind.Enabled = False
Else
MsgBox "El Pasword no corresponde al empleado", vbExclamation, "Solicitud"
btNEmp.Text = ""
btNombre.Text = ""
btArea.Text = ""
btPuesto.Text = ""
btTel.Text = ""
btSal.Text = ""
Call pHabilFolder
btFolio.SetFocus
End If
End If
Else
btFolio.SetFocus
End If
End If
Screen.MousePointer = vbNormal
End Sub
Private Sub cmdGastos_Click()
Screen.MousePointer = vbHourglass
Pfechas = btDiaS.Text + "/" + btMesS.Text + "/" + btAnioS.Text
pHoraS = TimeValue(btHoraS.Text + ":" + btMinS.Text)
frmGastos.Show
End Sub
Private Sub cmdNuevObj_Click()
btVisita = ""
cboVisita.Text = ""
cboObjNom.Text = ""
btDesObj.Text = ""
iObjNoC = 1 'asigna el valor para saber que se va a introducir un
'objetivo nuevo
'habilita spñp el boton de aceptar y los textboxes
lstObj.Enabled = False
cmdNuevObj.Enabled = False
cmdBorraObj.Enabled = False
cmdCambObj.Enabled = False
cmdRegObj.Enabled = True
fraTipVisita.Enabled = True
cboVisita.Enabled = True
btVisita.Enabled = True
cboObjNom.Enabled = True
btDesObj.Enabled = True
If optOtro Then
btVisita.SetFocus
Else
cboVisita.SetFocus
End If
End Sub
Private Sub cmdRegObj_Click()
If Trim(cboVisita.Text) = "" And Trim(btVisita) = "" Then 'si no puso el proyecto
MsgBox "Debe especificar su visita", vbExclamation, "Solicitud"
Elseif Trim(cboObjNom.Text) = "" Then 'si no puso el nombre del objetivo
MsgBox "Debe nombrar un objetivo", vbExclamation, "Solicitud"
Elseif cboObjNom.Text Like "OTROS" And Trim(btDesObj.Text) = "" Then 'si no puso la descripcion
MsgBox "Debe dar una descripción de su objetivo", vbExclamation, "Solicitud"
Elseif btDesc_obj.Text = "" Then 'so no puso la descripción corta del obj
MsgBox "Debe escribir una descripción corta del objetivo", vbExclamation, "Solicitud"
Elseif iObjNoC = 1 Then 'si es nuevo
'llena datos en una localidad nueva del arreglo
If optProy Then
With iObjetiv(lstObj.ListCount)
.Tipo = "P" 'VISITA A PROYECTO
.IstrCve = cboVisita.BoundText
.IDesc = cboVisita.Text
.IDesc_corta_obj = btDesc_obj.Text 'lhp en los 4
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
End With
Elseif optCli Then
  With IObjetivo(lstObj.ListCount)
    .ITipo = "C" 'VISITA A CLIENTE
    .IntCve = Val(cboVisita.BoundText)
    .IDesc = cboVisita.Text
    .IDesc_corta_obj = txtdesc_obj.Text
  End With
Elseif optDele Then
  With IObjetivo(lstObj.ListCount)
    .ITipo = "D" 'VISITA A DELEG
    .IntCve = Val(cboVisita.BoundText)
    .IDesc = cboVisita.Text
    .IDesc_corta_obj = txtdesc_obj.Text
  End With
Else
  With IObjetivo(lstObj.ListCount)
    .ITipo = "O" 'OTRA VISITA
    .IstrCve = cboVisita.BoundText
    .IDesc = txtVisita.Text
    .IDesc_corta_obj = txtdesc_obj.Text
  End With
End If
IObjetivo(lstObj.ListCount).INObj = _
  cboObjNom.BoundText
IObjetivo(lstObj.ListCount).IDObj = _
  cboObjNom.Text
IObjetivo(lstObj.ListCount).IDescObj = _
  txtDesObj.Text
lstObj.AddItem IObjetivo(lstObj.ListCount).IDesc 'inseta nombre a la listbox
'habilita boton de nuevo
lstObj.Enabled = True
'If lstObj.ListCount < 4 Then 'solo permitir 4 objetivos
  cmdNuevObj.Enabled = True
'Else
  cmdNuevObj.Enabled = False
'End If
cmdBorraObj.Enabled = False
cmdCambObj.Enabled = False
cmdRegObj.Enabled = False
fraTipVisita.Enabled = False
cboVisita.Enabled = False
txtVisita.Enabled = False
cboObjNom.Enabled = False
txtDesObj.Enabled = False
IObjNoC = 0 'reinicializa la variable
Elseif IObjNoC = 2 Then 'si es un cambio
  'reemplaza datos en el arreglo
  If optProy Then
    With IObjetivo(lstObj.ListIndex)
      .ITipo = "P" 'VISITA A PROYECTO
      .IstrCve = cboVisita.BoundText
      .IDesc = cboVisita.Text
      .IDesc_corta_obj = txtdesc_obj.Text 'lhp en los 4
    End With
  Elseif optCli Then
    With IObjetivo(lstObj.ListIndex)
      .ITipo = "C" 'VISITA A CLIENTE
      .IntCve = Val(cboVisita.BoundText)
      .IDesc = cboVisita.Text
      .IDesc_corta_obj = txtdesc_obj.Text
    End With
  Elseif optDele Then
    With IObjetivo(lstObj.ListIndex)
      .ITipo = "D" 'VISITA A DELEG
      .IntCve = Val(cboVisita.BoundText)
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
.IDesc = cboVisita.Text
IDesc_corta_obj = txtdesc_obj.Text
End With
Else
With IObjetiv(IstObj.ListIndex)
.ITipo = "O" 'OTRA VISITA
.IstrCve = cboVisita.BoundsText
.IDesc = txtVisita.Text
IDesc_corta_obj = txtdesc_obj.Text
End With
End If
IOjetiv(IstObj.ListIndex).INObj = _
cboObjNom.BoundsText
IOjetiv(IstObj.ListIndex).IDObj = _
cboObjNom.Text
IOjetiv(IstObj.ListIndex).IDescObj = _
txtDesObj.Text
'reemplaza nombre en la lista
IstObj.List(IstObj.ListIndex) =
IOjetiv(IstObj.ListIndex).IDesc
'habilita boton de nuevo
IstObj.Enabled = True
cmdNuevoObj.Enabled = True
cmdBorraObj.Enabled = False
cmdCambObj.Enabled = False
cmdRegObj.Enabled = False
fraTipVisita.Enabled = False
cboVisita.Enabled = False
txtVisita.Enabled = False
cboObjNom.Enabled = False
txtDesObj.Enabled = False
IObjNoC = 0
End If
End Sub
Private Sub cmdSave_Click()
Dim Istrq As String
Dim Istrf As String
Dim Lcont As Byte
Dim Ihora As String
Dim Iminu As String
Dim IFechaSal As String
Dim IFechaLle As String
Dim IFechAnt As String
Screen.MousePointer = vbHourglass
If Not FVerifSol Then
Screen.MousePointer = vbNormal
Exit Sub
End If
If FVeriFechas Then
If PintDias <> pNumDias Then
MsgBox "Verifique el detalle de destinos!!", vbExclamation, "Solicitud"
Screen.MousePointer = vbNormal
Exit Sub
End If
Call pProcDest
IFechaSal = FDaFecha(Pfechas)
IFechaLle = FDaFecha(Pfechal)
Ihora = Format(DatePart("h", pHoraS), "00")
Iminu = Format(DatePart("n", pHoraS), "00")
PhoraSal = Ihora & ":" & Iminu
Ihora = Format(DatePart("h", pHoraL), "00")
Iminu = Format(DatePart("n", pHoraL), "00")
PhoraLle = Ihora & ":" & Iminu
If M(Append Then ' nueva solicitud
" actualiza consecutivo (consecut)
Icad = "select * from consecut"
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Set rsviat01 = bdrdoviat.OpenResultSet(Name =lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
lintUFolio = rsviat01!nsi_cofolio
btFolio.Text = lintUFolio + 1 'ultimo registro de s_folio?

Istrq = "(" & btFolio & ")"
Istrf = "(nsi_cofolio)"
bdrdoviat.Execute "update consecut set" & Istrf & "=" & Istrq
IFechAnt = FDaFecha(lblFAntici)
Istrq = "(" & btFolio & "," & btNEmp & "," & IFechAnt & ","
If btnEmpSup = "" Then 'guarda 0 si ningun superior acompaña
    Istrq = Istrq & " 0,"
Else
    Istrq = Istrq & btnEmpSup & ","
End If
Istrq = Istrq & btNEmpAut & "," & IFechaSal & "," & IFechaLle & "," & _
"" & PhoraSal & "," & PhoraLle & "," & pNumDias & "," & _
pAlimnt & "," & phospedj & ","
If pKm <> 0 Then
    Istrq = Istrq & pLtrGas & ","
Else
    Istrq = Istrq & "null,"
Istrq = Istrq & "0,"
End If
'meter costo gas y validarlo arriba
Istrq = Istrq & pBus & "," & pPeaje & "," & pdblCostAuto & ","
& pFoto & "," & pLegales & "," & pOtros & "," & pKm & "," & _
IstObj.ListCount & ","
If pObs = "" Then
    Istrq = Istrq & "null,"
Else
    Istrq = Istrq & "" & pObs & ","
End If
If pItnAvion = "" Then
    Istrq = Istrq & "null,"
Else
    Istrq = Istrq & "" & pItnAvion & ","
End If
If pFechaAvion = "" Then
    Istrq = Istrq & "null,"
Else
    Istrq = Istrq & "" & pFechaAvion & ","
End If
If pHoraAvion = "" Then
    Istrq = Istrq & "null,"
Else
    Istrq = Istrq & "" & pHoraAvion & ","
End If
If pItnBus = "" Then
    Istrq = Istrq & "null,"
Else
    Istrq = Istrq & "" & pItnBus & ","
End If
If pItnCarro = "" Then
    Istrq = Istrq & "null,null,"
ElseIf pRecMat Then
    Istrq = Istrq & "" & pItnCarro & ",";'S'
Else
    Istrq = Istrq & "" & pItnCarro & ",";'N'
End If
If Psciaerea = "" Then
    Psciaerea = "null"
End If

Istrq = Istrq & Pnum_area & "," & Psciaerea & ",";0)"
```



# MIGRACION DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Istrf = "(s_folio,nsi_num_emp,d_fech_ant," + _  
"nsi_num_empsup,nsi_num_empaut,d_pfecha_sal," + _  
"d_pfecha_lle,ch_phora_sal,ch_phora_lle,nsi_pdias," + _  
"m_pvia_alimen,m_pvia_hosp,m_costo_gas,m_pvia_bus,m_pvia_peaje," + _  
"m_pvia_gas,m_pvia_foto,m_pvia_legal,m_pvia_otros,nsi_km," + _  
"nsi_tot_proy,vch_via_obsrv,vch_itin_avion,vch_fec_avion_sal," + _  
"vch_hora_avion_sal,vch_itin_bus,vch_itin_auto,ch_auto_recmat,ni_num_area,vch_pvia_ciarea,m_saldo_afect)"  
pDbViat.Execute "insert into hist_viat_emp " & Istrf & " values " & Istrq  
  
bdrdoviat.Execute "insert into hist_viat_emp " & Istrf & " values " & Istrq  
Lcont = 0  
If pNumDias = 1 Then  
Istrq = txtFolio & "," & FDaFecha(Pfechas + Lcont) & "," & pDest(Lcont).INDest & ","  
If pDest(Lcont).IDescDest = "" Then  
Istrq = Istrq & "null,"  
Else  
Istrq = Istrq & "" & pDest(Lcont).IDescDest & ""  
End If  
Istrq = Istrq & pDest(Lcont).ICuota  
bdrdoviat.Execute "insert into hist_dest_emp values(" & Istrq & ")"  
Else  
While Lcont <= pNumDias - 2  
Istrq = txtFolio & "," & FDaFecha(Pfechas + Lcont) & "," & pDest(Lcont).INDest & ","  
If pDest(Lcont).IDescDest = "" Then  
Istrq = Istrq & "null,"  
Else  
Istrq = Istrq & "" & pDest(Lcont).IDescDest & ""  
End If  
Istrq = Istrq & pDest(Lcont).ICuota  
bdrdoviat.Execute "insert into hist_dest_emp values(" & Istrq & ")"  
Istrq = "(" & txtfolcom & "," & lfecha & "," & rsviat01:m_rcuota_dia & ",0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)"  
Istrf = "(s_folio,d_fec_afecta,  
m_rcuota_dia,cpto_0,cpto_1,cpto_2,cpto_3,cpto_4,cpto_5,cpto_6,cpto_7,cpto_8,cpto_9,cpto_10,cpto_11,cpto_12,cpto_13)"  
bdrdoviat.Execute "insert into rep_paso_comp " & Istrf & " values " & Istrq  
Lcont = Lcont + 1  
Wend  
End If  
Lcont = 0  
While Lcont < lstObj.ListCount  
Select Case lObjetiv(Lcont).ITipo  
Case "P"  
Istrf = "(s_folio,ch_num_proy,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"  
Istrq = "(" & txtFolio & "," & lObjetiv(Lcont).IstrCve & _  
" " & lObjetiv(Lcont).lNObj & " "  
If lObjetiv(Lcont).IDescObj = "" Then  
Istrq = Istrq & "null"  
Else  
Istrq = Istrq & "" & lObjetiv(Lcont).IDescObj & ""  
End If  
Istrq = Istrq & "," & lObjetiv(Lcont).IDesc_corta_obj & ")" ' lhp  
  
bdrdoviat.Execute "insert into hist_obj_proy_emp " & Istrf & _  
"values " & Istrq  
Case "C"  
Istrf = "(s_folio,ni_cliente,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"  
Istrq = "(" & txtFolio & "," & lObjetiv(Lcont).IintCve & _  
" " & lObjetiv(Lcont).lNObj & " "  
If lObjetiv(Lcont).IDescObj = "" Then  
Istrq = Istrq & "null"  
Else  
Istrq = Istrq & "" & lObjetiv(Lcont).IDescObj & ""  
End If  
Istrq = Istrq & "," & lObjetiv(Lcont).IDesc_corta_obj & ")" ' lhp  
  
bdrdoviat.Execute "insert into hist_obj_cli_emp " & Istrf & _  
"values " & Istrq
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Case "D"
    Istrq = "(s_folio,ni_num_dele,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"
    Istrq = "(" & btFolio & "," & IObjetiv(Lcont) lintCve & _
        "," & IObjetiv(Lcont).INObj & ","
    If IObjetiv(Lcont).IDescObj = "" Then
        Istrq = Istrq & "null"
    Else
        Istrq = Istrq & "" & IObjetiv(Lcont).IDescObj & ""
    End If
    Istrq = Istrq & "," & IObjetiv(Lcont).IDesc_corta_obj & ")" ' lhp

    bdrdoviat.Execute "insert into hist_obj_del_emp " & Istrq & _
        "values " & Istrq

Case "O"
    Istrq = "(s_folio,vch_desc_visita,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"
    Istrq = "(" & btFolio & "," & IObjetiv(Lcont).IDesc & _
        "," & IObjetiv(Lcont).INObj & ","
    If IObjetiv(Lcont).IDescObj = "" Then
        Istrq = Istrq & "null"
    Else
        Istrq = Istrq & "" & IObjetiv(Lcont).IDescObj & ""
    End If
    Istrq = Istrq & "," & IObjetiv(Lcont).IDesc_corta_obj & ")" ' lhp

    bdrdoviat.Execute "insert into hist_obj_otro_emp " & Istrq & _
        "values " & Istrq

End Select
Lcont = Lcont + 1
Wend

MsgBox "Solicitud Grabada con el Folio : " & btFolio.Text, vbInformation, "Solicitud"
DBEngine.Workspaces(0).CommitTrans
Else 'modo de edición
    Istrq = Istrq & "nsi_num_empsup="
    If btnEmpSup = "" Then
        Istrq = Istrq & " 0,"
    Else
        Istrq = Istrq & btnEmpSup & ","
    End If
    Istrq = Istrq & "nsi_num_empaut=" & btnNEmpAut & ","
    Istrq = Istrq & "d_pfecha_sal=" & IFechaSal & "," & d_pfecha_lle=" & _
        "" & IFechaLle & "," & ch_phora_sal=" & PhoraSal & "," & ch_phora_lle=" & _
        "" & PhoraLle & "," & nsi_pdias=" & pNumDias & _
        "" & m_pvia_alimen=" & pAlimnt & "," & m_pvia_hosp=" & phosped) & "," & _
        "m_costo_gas="
    If pKm <> 0 Then
        Istrq = Istrq & pLtrGas & ","
    Else
        Istrq = Istrq & "0," 'lhp decia null
    End If
    Istrq = Istrq & "m_pvia_bus=" & pBus & "," & m_pvia_peaje=" & pPeaje & _
        "" & m_pvia_gas=" & pdblCostAuto & "" & m_pvia_foto=" & pFoto & _
        "" & m_pvia_legal=" & pLegales & "" & m_pvia_otros=" & pOtros & _
        "" & nsi_km=" & pKm & "" & nsi_tot_proy=" & IstObj.ListCount & ""
    If pObs = "" Then
        Istrq = Istrq & "vch_via_obsv=null,"
    Else
        Istrq = Istrq & "vch_via_obsv=" & pObs & ""
    End If
    If pItinAvion = "" Then
        Istrq = Istrq & "vch_itin_avion=null,"
    Else
        Istrq = Istrq & "vch_itin_avion=" & pItinAvion & ""
    End If
    If pFechaAvion = "" Then
        Istrq = Istrq & "vch_fec_avion_sal=null,"
    End If
End If
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Else
    Istrq = Istrq & "vch_fec_avion_sal=" & pFechaAvion & ","
End If
If pHoraAvion = "" Then
    Istrq = Istrq & "vch_hora_avion_sal=null,"
Else
    Istrq = Istrq & "vch_hora_avion_sal=" & pHoraAvion & ","
End If
If pItinBus = "" Then
    Istrq = Istrq & "vch_itin_bus=null,"
Else
    Istrq = Istrq & "vch_itin_bus=" & pItinBus & ","
End If
If pItinCarro = "" Then
    Istrq = Istrq & "vch_itin_auto=null,ch_auto_recmat=null"
ElseIf pRecMat Then
    Istrq = Istrq & "vch_itin_auto=" & pItinCarro & "," + _
        "ch_auto_recmat='S'"
Else
    Istrq = Istrq & "vch_itin_auto=" & pItinCarro & "," + _
        "ch_auto_recmat='N'"
End If
bdrdoviat.Execute "update hist_viat_emp set " & Istrq & " where s_folio = " & txtFolio.Text
bdrdoviat.Execute "delete from hist_dest_emp where s_folio=" & txtFolio & dbSQLPassThrough
bdrdoviat.Execute "delete from hist_obj_proy_emp where s_folio=" & txtFolio & dbSQLPassThrough
bdrdoviat.Execute "delete from hist_obj_cii_emp where s_folio=" & txtFolio & dbSQLPassThrough
bdrdoviat.Execute "delete from hist_obj_del_emp where s_folio=" & txtFolio & dbSQLPassThrough
bdrdoviat.Execute "delete from hist_obj_otro_emp where s_folio=" & txtFolio & dbSQLPassThrough
Lcont = 0
If pNumDias = 1 Then
    Istrq = txtFolio & "," & FDaFecha(Pfechas + Lcont) & "," & pDest(Lcont).INDest & ","
    If pDest(Lcont).IDescDest = "" Then
        Istrq = Istrq & "null,"
    Else
        Istrq = Istrq & "" & pDest(Lcont).IDescDest & ","
    End If
    Istrq = Istrq & pDest(Lcont).ICuota
    bdrdoviat.Execute "insert into hist_dest_emp values(" & Istrq & ")"
Else
    While Lcont <= pNumDias - 2
        Istrq = txtFolio & "," & FDaFecha(Pfechas + Lcont) & "," & pDest(Lcont).INDest & ","
        If pDest(Lcont).IDescDest = "" Then
            Istrq = Istrq & "null,"
        Else
            Istrq = Istrq & "" & pDest(Lcont).IDescDest & ","
        End If
        Istrq = Istrq & pDest(Lcont).ICuota

        bdrdoviat.Execute "insert into hist_dest_emp values(" & Istrq & ")"
        Lcont = Lcont + 1
    Wend
End If
Lcont = 0
While Lcont < IstObj.ListCount
    Select Case IObjetiv(Lcont).ITipo
    Case "P"
        Istrf = "(s_folio,ch_num_proy,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"
        Istrq = "(" & txtFolio & "," & IObjetiv(Lcont).IstrCve & _
            "" & IObjetiv(Lcont).INObj & ","
        If IObjetiv(Lcont).IDescObj = "" Then
            Istrq = Istrq & "null"
        Else
            Istrq = Istrq & "" & IObjetiv(Lcont).IDescObj & ""
        End If
        Istrq = Istrq & "" & IObjetiv(Lcont).IDesc_corta_obj & ")"
        bdrdoviat.Execute "insert into hist_obj_proy_emp " & Istrf &
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
"values " & Istrq
Case "C"
Istrf = "(s_folio,ni_cliente,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"
Istrq = "(" & btFolio & "," & IOjetiv(Lcont).lntCve & _
" " & IOjetiv(Lcont).lNObj & ","
If IOjetiv(Lcont).lDescObj = "" Then
Istrq = Istrq & "null"
Else
Istrq = Istrq & "" & IOjetiv(Lcont).lDescObj & ""
End If
Istrq = Istrq & "," & IOjetiv(Lcont).lDesc_corta_obj & ")" ' lhp

bdrdoviat.Execute "insert into hist_obj_cli_emp " & Istrf & _
"values " & Istrq
Case "D"
Istrf = "(s_folio,ni_num_dele,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"
Istrq = "(" & btFolio & "," & IOjetiv(Lcont).lntCve & _
" " & IOjetiv(Lcont).lNObj & ","
If IOjetiv(Lcont).lDescObj = "" Then
Istrq = Istrq & "null"
Else
Istrq = Istrq & "" & IOjetiv(Lcont).lDescObj & ""
End If
Istrq = Istrq & "," & IOjetiv(Lcont).lDesc_corta_obj & ")" ' lhp

bdrdoviat.Execute "insert into hist_obj_del_emp " & Istrf & _
"values " & Istrq
Case "O"
Istrf = "(s_folio,vch_desc_visita,nsi_num_obj,vch_desc_obj,vch_desc_corta_obj)"
Istrq = "(" & btFolio & "," & IOjetiv(Lcont).lDesc & _
" " & IOjetiv(Lcont).lNObj & ","
If IOjetiv(Lcont).lDescObj = "" Then
Istrq = Istrq & "null"
Else
Istrq = Istrq & "" & IOjetiv(Lcont).lDescObj & ""
End If
Istrq = Istrq & "," & IOjetiv(Lcont).lDesc_corta_obj & ")"
bdrdoviat.Execute "insert into hist_obj_otro_emp " & Istrf & _
"values " & Istrq
End Select
Lcont = Lcont + 1
Wend
MsgBox "Edición/Solicitud grabada con éxito", vbInformation, "Solicudt"
'DBEngine.Workspaces(0).CommitTrans
End If
ButtonNavigateMode
PLimpiaCampos
IstObj.Clear
ReDim pDest(1)
End If
Screen.MousePointer = vbNormal
Exit Sub
HandleError:
Screen.MousePointer = vbNormal
MsgBox Err.Number & "." & Err.Description
Exit Sub
End Sub
Private Sub Form_Load()
On Error GoTo ManError
pstrfol = LblFol(1).Caption
pstrfecha = lblFecha(1).Caption
'definicion de sentencia sql para proyectos, clientes, delegaciones y objetivos
Call ButtonNavigateMode
pbolSwitcSup = False
pLtrGas = 2.3 'litro de gas? lhp
pObs = ""
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Screen.MousePointer = vbNormal
Centrar Me
Exit Sub
ManError
Screen.MousePointer = 0
If rdoErrors.Count > 0 Then
    Dim lerror As rdoError
    For Each lerror In rdoErrors
        MsgBox F_RdoError(lerror), vbCritical, "Error"
        Call P_GuardaError(Me.Name, lerror.Number, F_RdoError(lerror))
    Next
    rdoErrors.Clear
Else
    MsgBox F_ManError(Err.Number), vbCritical, "Error"
    Call P_GuardaError(Me.Name, Err.Number, Err.Description)
End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
    'pdrhnm.Close
    frmMain.Show
End Sub
Private Sub lstObj_Click()
    'actualiza los datos seleccionados a las textboxes
    Select Case lObjetiv(lstObj.ListIndex).ITipo
        Case "P"
            optProy.Value = True
            cboVisita.Text = lObjetiv(lstObj.ListIndex).IDesc
            txtdesc_obj.Text = lObjetiv(lstObj.ListIndex).IDesc_corta_obj ' lhp
        Case "C"
            optCli.Value = True
            cboVisita.Text = lObjetiv(lstObj.ListIndex).IDesc
            txtdesc_obj.Text = lObjetiv(lstObj.ListIndex).IDesc_corta_obj ' lhp
        Case "D"
            optDele.Value = True
            cboVisita.Text = lObjetiv(lstObj.ListIndex).IDesc
            txtdesc_obj.Text = lObjetiv(lstObj.ListIndex).IDesc_corta_obj ' lhp
        Case "O"
            optOtro.Value = True
            txtVisita = lObjetiv(lstObj.ListIndex).IDesc
            txtdesc_obj.Text = lObjetiv(lstObj.ListIndex).IDesc_corta_obj ' lhp
    End Select
    cboObjNom.Text = _
        lObjetiv(lstObj.ListIndex).IDObj
    If lObjetiv(lstObj.ListIndex).IDescObj <> "" Then
        lblDescObj.Visible = True
        txtDesObj.Visible = True
        txtDesObj.Text = lObjetiv(lstObj.ListIndex).IDescObj
    Else
        lblDescObj.Visible = False
        txtDesObj.Visible = False
        txtDesObj.Text = ""
    End If
    'habilita boton de cambiar y borrar
    cmdCambObj.Enabled = True
    cmdBorraObj.Enabled = True
End Sub
Private Sub optCli_Click()
    Screen.MousePointer = vbHourglass
    If optCli.Value Then
        datVisita.RecordSource = "Clientes"
        datVisita.Refresh
        With cboVisita
            .Visible = True
            .Text = ""
            .ListField = "chNombre"
            .BoundColumn = "niCliente"
        End With
    End If
End Sub
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
End With
    btnVisita.Visible = False
    btnVisita = ""
End If
Screen.MousePointer = vbNormal
End Sub
Private Sub optDele_Click()
    Screen.MousePointer = vbHourglass
    If optDele.Value Then
        datVisita.RecordSource = "CatDelegaciones"
        datVisita.Refresh
        With cboVisita
            .Visible = True
            Text = ""
            .ListField = "chDelegacion"
            .BoundColumn = "niNumDel"
        End With
        btnVisita.Visible = False
        btnVisita = ""
    End If
    Screen.MousePointer = vbNormal
End Sub
Private Sub optOtro_Click()
    Screen.MousePointer = vbHourglass
    If optOtro.Value Then
        'datVisita.RecordSource = ""
        'datVisita.Refresh
        With cboVisita
            .Visible = False
            Text = ""
        End With
        btnVisita.Visible = True
        btnVisita = ""
    End If
    Screen.MousePointer = vbNormal
End Sub
Private Sub optProy_Click()
    Screen.MousePointer = vbHourglass
    If optProy.Value Then
        With datVisita
            .RecordSource = "Proyectos"
            .Refresh
        End With
        With cboVisita
            .Visible = True
            Text = ""
            .ListField = "chNproy"
            .BoundColumn = "chNProy"
        End With
        btnVisita = ""
        btnVisita.Visible = False
    End If
    Screen.MousePointer = vbNormal
End Sub
Private Sub spbfechas_SpinDown()
    If lmin = 1 Then 'si son dias
        If Val(btnDiaS.Text) <= 1 Then 'si llega al limite
            btnDiaS.Text = "31"
        Else
            btnDiaS.Text = Format(Val(btnDiaS.Text) - 1, "00")
        End If
    ElseIf lmin = 2 Then 'si son meses
        If Val(btnMesS.Text) <= 1 Then 'si llega al limites
            btnMesS.Text = "12"
        Else
            btnMesS.Text = Format(Val(btnMesS.Text) - 1, "00")
        End If
    End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
End If
Else 'si son años
    txtAnioS.Text = Format(Val(txtAnioS.Text) - 1, "00")
End If
End Sub
Private Sub spbfechas_SpinUp()
If lmin = 1 Then 'si son dias
    If Val(txtDiaS.Text) >= 31 Then 'si llega al limite
        txtDiaS.Text = "01"
    Else
        txtDiaS.Text = Format(Val(txtDiaS.Text) + 1, "00")
    End If
ElseIf lmin = 2 Then 'si son meses
    If Val(txtMesS.Text) >= 12 Then 'si llega al limite
        txtMesS.Text = "01"
    Else
        txtMesS.Text = Format(Val(txtMesS.Text) + 1, "00")
    End If
Else 'si son años
    txtAnioS.Text = Format(Val(txtAnioS.Text) + 1, "00")
End If
End Sub
Private Sub spnFechaL_SpinDown()
If lmin = 1 Then 'si son dias
    If Val(txtDiaL.Text) <= 1 Then 'si llega al limite
        txtDiaL.Text = "31"
    Else
        txtDiaL.Text = Format(Val(txtDiaL.Text) - 1, "00")
    End If
ElseIf lmin = 2 Then 'si son meses
    If Val(txtMesL.Text) <= 1 Then 'si llega al limite
        txtMesL.Text = "12"
    Else
        txtMesL.Text = Format(Val(txtMesL.Text) - 1, "00")
    End If
Else 'si son años
    txtAnioL.Text = Format(Val(txtAnioL.Text) - 1, "00")
End If
End Sub
Private Sub spnFechaL_SpinUp()
If lmin = 1 Then 'si son dias
    If Val(txtDiaL.Text) >= 31 Then 'si llega al limite
        txtDiaL.Text = "01"
    Else
        txtDiaL.Text = Format(Val(txtDiaL.Text) + 1, "00")
    End If
ElseIf lmin = 2 Then 'si son meses
    If Val(txtMesL.Text) >= 12 Then 'si llega al limite
        txtMesL.Text = "01"
    Else
        txtMesL.Text = Format(Val(txtMesL.Text) + 1, "00")
    End If
Else 'si son años
    txtAnioL.Text = Format(Val(txtAnioL.Text) + 1, "00")
End If
End Sub
Private Sub spnHoraL_SpinDown()
If lmin1 = 1 Then 'si son horas
    If Val(txtHoraL.Text) <= 0 Then 'si llega al limite
        txtHoraL.Text = "23"
    Else
        txtHoraL.Text = Format(Val(txtHoraL.Text) - 1, "00")
    End If
Else 'si son minutos
    If Val(txtMinL.Text) <= 0 Then 'si llega al limite
        txtMinL.Text = "59"
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Else
    txtMinLI.Text = Format(Val(txtMinLI.Text) - 1, "00")
End If
End Sub
Private Sub spnHoraL_SpinUp()
If IMin1 = 1 Then 'si son horas
    If Val(txtHoraLI.Text) >= 23 Then 'si llega al limite
        txtHoraLI.Text = "00"
    Else
        txtHoraLI.Text = Format(Val(txtHoraLI.Text) + 1, "00")
    End If
Else 'si son minutos
    If Val(txtMinLI.Text) >= 59 Then 'si llega al limite
        txtMinLI.Text = "00"
    Else
        txtMinLI.Text = Format(Val(txtMinLI.Text) + 1, "00")
    End If
End If
End Sub
Private Sub spnHoraS_SpinDown()
If IMin1 = 1 Then 'si son horas
    If Val(txtHoraS.Text) <= 0 Then 'si llega al limite
        txtHoraS.Text = "23"
    Else
        txtHoraS.Text = Format(Val(txtHoraS.Text) - 1, "00")
    End If
Else 'si son minutos
    If Val(txtMinS.Text) <= 0 Then 'si llega al limite
        txtMinS.Text = "59"
    Else
        txtMinS.Text = Format(Val(txtMinS.Text) - 1, "00")
    End If
End If
End Sub
Private Sub spnHoraS_SpinUp()
If IMin1 = 1 Then 'si son horas
    If Val(txtHoraS.Text) >= 23 Then 'si llega al limite
        txtHoraS.Text = "00"
    Else
        txtHoraS.Text = Format(Val(txtHoraS.Text) + 1, "00")
    End If
Else 'si son minutos
    If Val(txtMinS.Text) >= 59 Then 'si llega al limite
        txtMinS.Text = "00"
    Else
        txtMinS.Text = Format(Val(txtMinS.Text) + 1, "00")
    End If
End If
End Sub
Private Sub tstDiaL_GotFocus()
    Imin = 1 'asigna valor para saber que son dias
End Sub
Private Sub txtAnioLI_GotFocus()
    Imin = 3 'asigna valor para saber que son años
End Sub
Private Sub txtAnioS_GotFocus()
    Imin = 3 'asigna valor para saber que son años
End Sub
Private Sub txtdesc_obj_Change()
    If MfAppend = False Then
        IObjetivo(IstObj.ListIndex).IDesc_corta_obj = txtdesc_obj.Text ' End If
End Sub
Private Sub txtdesc_obj_KeyDown(KeyAscii As Integer)
    Dim car As String * 1
    car = UCase(Chr(KeyAscii))
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
KeyAscii = Asc(car)
If KeyAscii = 13 Or KeyAscii = 34 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
    Beep
End If
End Sub
Private Sub txtDesObj_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub
Private Sub txtDiaS_GotFocus()
    lMin = 1 'asigna valor para saber que son dias
End Sub
Private Sub txtfolio_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    cmdFind.Value = True
End If
End Sub
Private Sub txtHoraLI_GotFocus()
    lMin1 = 1 'asigna valor para saber que son horas
End Sub
Private Sub txtHoraS_GotFocus()
    lMin1 = 1 'asigna valor para saber que son horas
End Sub
Private Sub txtMesLI_GotFocus()
    lMin = 2 'asigna valor para saber que son meses
End Sub
Private Sub txtMesS_GotFocus()
    lMin = 2 'asigna valor para saber que son meses
End Sub
Private Sub txtMinLI_GotFocus()
    lMin1 = 2 'asigna valor para saber que son minutos
End Sub
Private Sub txtMinS_GotFocus()
    lMin1 = 2 'asigna valor para saber que son minutos
End Sub
Private Sub txtNEmp_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
    cmdFind.Value = True
End If
End Sub
Public Sub Pcheca_saldo()
    Dim Lb_cont As Byte
    Screen.MousePointer = 11
    lcad = "select * from hist_viat_emp where ch_vb_conta is not null " +
    "and vch_logcancela is null and m_saldo > 0 and nsi_num_emp=" & txtNEmp
    Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lcad, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
    If rsviat01.EOF Then
        ' No tiene adeudos esta persona
    Else
        rsviat01.MoveFirst
        'a = rsviat01.RowCount
        While Not rsviat01.EOF
            lcad = "select * from hist_saldos_emp where nsi_folio" +
            "= " & rsviat01!s_folio & " And ch_tipo=" & "CT" & " or " +
            " nsi_folio=" & rsviat01!s_folio & " and ch_tipo=" & "CN"
            Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
            Type:=rdOpenDynamic, LockType:=rdConcurValues)
            'a = rsviat02.RowCount
            If rsviat02.EOF Then
```

**ESTA TESIS NO SALE  
DE LA BIBLIOTECA**



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
no ha realizado ningun pago
Else
    rsviat02.MoveFirst
    Ps_suma_m2 = 0
    While Not rsviat02.EOF
        Ps_suma_m2 = Ps_suma_m2 + rsviat02!m_importe
        rsviat02.MoveNext
    Wend
    Format(Ln_cuota, "###,##0.00")
    If Format(Ps_suma_m2, "###,##0.00") < Format(Abs(rsviat01!m_saldo), "###,##0.00") Then
        MsgBox "Este empleado tiene un adeudo y no puede sacar Viaticos", vbExclamation, "Solicitud"
        Pt_flag_saldo = True
    End If
End If
rsviat01.MoveNext
Lb_cont = lbcont + 1
Wend
End If
Screen.MousePointer = 1
End Sub
Private Sub btnVisita_KeyPress(KeyAscii As Integer)
    Dim car As String * 1
    car = UCase(Chr(KeyAscii))
    KeyAscii = Asc(car)
End Sub
```

---

\*frm\_factxfolio    Forma de Reportes de facturas por folio  
                  de Sol/Comp de Viaticos  
\*Autor:            Luis Hernández Pérez

---

\*Tablas  
\*Base de Datos    Tablas que Utiliza    Lectura    Escritura  
\*Viaticos        Hist\_fact\_emp        sql        sql  
\*Viaticos        Hist\_viat\_emp        sql  
\*Viaticos        rep\_facturas        sql        sql

---

\*Forma(s) que Llama: frmreportes, frm\_factxfolio  
\*Llamada por:        frmreportes

---

```
Private Sub cmd_foliofact_Click()
If bt_foliofact = "" Then
    MsgBox ("Debe Instoducir un Folio"), vbExclamation, "Reportes"
    bt_foliofact.SetFocus
Else
    lcad = "select * from hist_fact_emp where " + _
"s_folio=" & bt_foliofact & " order by d_fec_afecta,nsi_cve_gasto"

Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select nsi_num_emp from hist_viat_emp where " + _
"s_folio=" & bt_foliofact

Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

Screen.MousePointer = 1
If Pnum_emp = rsviat02!nsi_num_emp Or Ptipo_usr = "R" Then
    If rsviat01.EOF Then
        MsgBox ("No existen Facturas para este Folio"), vbExclamation, "Reportes"
    Else
        Screen.MousePointer = 1
        rsviat01.MoveFirst
        bdrdoviat.Execute "delete from rep_facturas"
        Do Until rsviat01.EOF
            lcad = "select vch_desc_gasto from cat_gastos where " +
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
"nsi_cve_gasto" & rsviat01!nsi_cve_gasto
Set rsviat02 = bdrdoviat.OpenResultSet(Name:=!cad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

Istrq = "(" & txt_foliofact & "," & rsviat01!d_fec_afecta & "," & rsviat02!vch_desc_gasto & _
"," & rsviat01!m_monto_realfact & "," & rsviat01!m_monto_autofact & "," & rsviat01!m_iva_monto & _
"," & rsviat01!vch_num_factura & "," & rsviat01!ch_fiscal & "," & rsviat01!ch_taxi_aerop & ")
Istrf = "(s_folio,d_fec_afecta,vch_des_gasto,m_monto_realfact,m_monto_autofact,
m_iva_monto,vch_num_factura,ch_fiscal,ch_taxi_aerop)"
bdrdoviat.Execute "insert into rep_facturas " & Istrf & " values " & Istrq
rsviat01.MoveNext
Loop
If opt_pant.Value = True Then
    cr_factura.Destination = 0
Else
    cr_factura.Destination = 1
End If
cr_factura.Connect = "DSN = viaticos;UID = " & PstrUsuario & ";PWD = " & Ppassword & ";DSQ = viaticos"
lpath = App.Path & "\rep009 rpt"
cr_factura.ReportFileName = lpath
Screen.MousePointer = 11
Pier_cr = cr_factura.PrintReport
MsgBox ("Tu Reporte fue impreso sin problemas"), vbExclamation, "Reportes"
End If
Else
    MsgBox "El Folio no corresponde al Empleado", vbExclamation, "Reportes"
    Screen.MousePointer = 1
End If
End If
End Sub
Private Sub cmd_return_Click()
    Unload Me
End Sub
Private Sub Form_Load()
    Call CenterForm(Me)
End Sub
Private Sub txt_foliofact_KeyPress(KeyAscii As Integer)
    If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
        Else
            MsgBox ("Introducir Numeros"), vbExclamation, "Comprobación"
            KeyAscii = 0
        End If
    End Sub
End Sub
-----
frmcancela      Forma de Cancelacion de Solicitud de viaticos
                (solamente si no ha depositado tesoreria)
'Autor:         Luis Hernández Pérez
-----
Tablas:
Base de Datos  Tablas que Utiliza  Lectura Escritura
-----
Viaticos      Hist_viat_emp      sql      sql
Viaticos      Cat_destinos      sql
Rec. Humanos  Datos_empleado    sql
Forma(s) que Llama: frmmain
Llamada por:  frmmain
Modificada por:
Private Sub cmdAceptar_Click()
    If btnnum_folio = "" Then
        MsgBox "Falta numero de folio por introducir", vbExclamation, "Tresoreria"
        btnnum_folio = ""
        btnnum_folio.SetFocus
    Else
        Screen.MousePointer = 11
        !cad = "select * from hist_viat_emp where m_tes_imp is null " + _
        "and vch_logcancela is null and d_fech_aut_ant is null and s_folio=" & btnnum_folio & ""
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
lcad = "select * from hist_viat_emp where m_mes_imp is null" + _
"and vch_logcancela is null and s_folio=" & txtnum_folio & ""
Set rsviat01 = bdrdoviat.OpenResultSet(Name=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsviat01.EOF Then
    Screen.MousePointer = 1
    MsgBox "No se puede Cancelar este folio porque no existe, ya fue Autorizado o Tiene un Deposito", vbExclamation,
"Tesoreria"
    MsgBox "No se puede Cancelar este folio porque no existe o Tiene un Deposito", vbExclamation, "Tesoreria"
    txtnum_folio.SetFocus
Else
    If Pnum_emp = rsviat01!nsi_num_emp Or Ptipo_usr = "R" Then
        txtnum_emp = rsviat01!nsi_num_emp
        txtfecha_sol = rsviat01!d_fech_ant
        txtdestino = rsviat01!nsi_num_emp
        Call llena_nombre
        Call llena_destino
        txtNombre = rsviat01!nsi_num_emp
        cmdcancelar.Visible = True
        cmdcancelar.SetFocus
    Else
        Screen.MousePointer = 1
        MsgBox "Usted no esta Autorizado para Cancelar Solicitudes o no es de Usted este folio", vbExclamation,
"Comprobación"
    End If
End If
Screen.MousePointer = 1
End If
End Sub
Private Sub cmdCancelar_Click()
bdrdoviat.Execute "update hist_viat_emp set vch_logcancela='S' where s_folio=" & rsviat01!s_folio
MsgBox "Cancelacion Realizada con Exito", vbExclamation, "Comprobación"
End Sub
Private Sub cmdreturn_Click()
Unload Me
frmMain.Show
End Sub
Private Sub Form_Load()
txtnum_emp.Enabled = False
txtnombre.Enabled = False
txtfecha_sol.Enabled = False
txtdestino.Enabled = False
cmdcancelar.Visible = False
End Sub
Public Sub llena_nombre()
If rsviat01!nsi_num_emp > 9000 And rsviat01!nsi_num_emp < 10000 Then
    MsgBox "Es un empleado por Honorarios", vbExclamation, "Comprobación"
    lcad = "select distinct cat_honorarios.ch_nombre,cat_honorarios.nsi_nivel, cat_puestos.vch_nom_puesto,
cat_areas.vch_nom_area" + _
"from cat_honorarios, cat_puestos,cat_areas where cat_honorarios.nsi_num_puesto = cat_puestos.nsi_num_puesto" + _
"and cat_honorarios.ni_num_area = cat_areas.ni_num_area and cat_honorarios.nsi_numhono = " & rsviat01!nsi_num_emp -
9000
    Set rsrh01 = bdrdoh.OpenResultSet(Name=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
    If rsrh01.EOF Then
        "debe tener area, puesto y nivel si no aunque exista no lo lee
        MsgBox "Número de empleado inexistente", vbExclamation, "Comprobación"
        txtnombre.Text = ""
    Else
        txtnombre.Text = rsrh01!ch_nombre
    End If
Else ' Empleado normal
    lcad = "select distinct datos_empleado.nsi_num_puesto,datos_empleado.vch_ap_paterno," + _
"datos_empleado.vch_ap_materno,datos_empleado.vch_nombre,datos_empleado.vch_ext_tel," + _
"cat_puestos.vch_nom_puesto,datos_personales.ni_num_area,cat_areas.vch_nom_area," + _
"cat_puestos.nsi_nivel" + _
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
"from datos_empleado, cat_puestos, datos_personales, cat_areas where " + _  
"cat_puestos.nsi_num_puesto = datos_empleado.nsi_num_puesto and " + _  
"cat_areas.ni_num_area = datos_personales.ni_num_area and " + _  
"datos_personales.nsi_num_emp = datos_empleado.nsi_num_emp and " + _  
"datos_empleado.nsi_num_emp = " & rsviat01!nsi_num_emp  
Screen.MousePointer = 11  
Set rsrh01 = bdrdorh.OpenResultSet(Name:=lacad, _  
Type:=rdOpenDynamic, LockType:=rdConcurValues)  
Screen.MousePointer = 1  
nivelemp = rsrh01!nsi_nivel  
If IsNull(Trim(rsrh01!vch_ap_materno)) Then  
    txtnombre.Text = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_nombre"))  
Else  
    txtnombre.Text = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_ap_materno")) + " " +  
Trim(rsrh01("vch_nombre"))  
End If  
End If  
End Sub  
Public Sub llena_destino()  
    lcad = "select * from hist_dest_emp where " + _  
"s_folio = " & rsviat01!s_folio  
Screen.MousePointer = 11  
Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lacad, _  
Type:=rdOpenDynamic, LockType:=rdConcurValues)  
If Not rsviat01.EOF Then  
    lcad = "select * from cat_destinos where " + _  
"nsi_num_dest = " & rsviat01!nsi_pnum_dest  
Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lacad, _  
Type:=rdOpenDynamic, LockType:=rdConcurValues)  
Screen.MousePointer = 1  
txtdestino = rsviat02!vch_nom_dest  
End If  
End Sub
```

frmcompro      Forma de Comprobación de viaticos del Modulo  
de Sol/Comp de Viaticos  
Autor:          Luis Hernández Pérez

#### Tablas:

Base de Datos    Tablas que Utiliza    Lectura    Escritura

'Viáticos	Hist_viat_emp	sql	sql
'Viáticos	Cat_destinos	sql	
'Viáticos	Cat_objetivos	sql	
'Viáticos	Cat_tarifas	sql	
'Viáticos	Cat_gastos	sql	
'Viáticos	Hist_bol_avion_emp	sql	sql
'Viáticos	Hist_comp_emp	sql	sql
'Viáticos	Hist_fact_emp	sql	sql
'Viáticos	Hist_dest_emp	sql	sql
'Viáticos	Hist_otrosgas_emp	sql	sql
'Viáticos	Hist_kms	sql	sql
'Viáticos	Hist_saldos_emp	sql	sql
'Viáticos	Hist_obj_cli_emp	sql	sql
'Viáticos	Hist_obj_del_emp	sql	sql
'Viáticos	Hist_obj_otros_emp	sql	sql
'Viáticos	Hist_obj_proy_emp	sql	sql
'Rec. Humanos	Datos_empleado	sql	
'Rec. Humanos	Cat_areas	sql	
'Rec. Humanos	Cat_estados	sql	
'Rec. Humanos	Cat_puestos	sql	
'Gestión Clientes		sql	
'Gestión Proyectos		sql	
'Gestión CatDelegaciones		sql	
'Forma(s) que Llama:	frmmain, frmfactura		
'Llamada por:	frmmain		
'Modificada por:			





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Dim lmincom As Byte
Dim lminfcom As Byte
Dim lsq1 As String
Dim cancelar As Byte
Dim lcont2 As Byte
Dim Piva_Hotel As Single
Dim Piva_impHotel As Single
Dim Piva_Alimen As Single
Dim Pivaprop_Alimen As Single
Dim Piva_Avion As Single
Dim Piva_Tua As Single
Dim Piva_Bus As Single
Dim Piva_pPaje As Single
Dim Piva_Gas As Single
Dim Piva_Taxi As Single
Dim Piva_Foto As Single
Dim Piva_Telefono As Single
Dim Piva_Legal As Single
Dim Piva_Otros As Single
Private Sub cmdaceptaobj_Click()
    robjproy = True
    robjclie = True
    robjdele = True
    robjotro = True
    Dim Lcont As Byte
    Screen.MousePointer = 11
    If nrsviatobj01 > 0 Then
        For Lcont = 0 To nrsviatobj01 - 1
            If arroyecto(Lcont) = "" Then
                MsgBox "Falta contestar Objetivos", vbExclamation, "Comprobación"
                robjproy = False
            End If
        Next
    End If
    If nrsviatobj02 > 0 Then
        For Lcont = 0 To nrsviatobj02 - 1
            If arrcliente(Lcont) = "" Then
                MsgBox "Falta contestar Objetivos", vbExclamation, "Comprobación"
                robjclie = False
            End If
        Next
    End If
    If nrsviatobj03 > 0 Then
        For Lcont = 0 To nrsviatobj03 - 1
            If ardelegacion(Lcont) = "" Then
                MsgBox "Falta contestar Objetivos", vbExclamation, "Comprobación"
                robjdele = False
            End If
        Next
    End If
    If nrsviatobj04 > 0 Then
        For Lcont = 0 To nrsviatobj04 - 1
            If arrotro(Lcont) = "" Then
                MsgBox "Falta contestar Objetivos", vbExclamation, "Comprobación"
                robjotro = False
            End If
        Next
    End If
    Screen.MousePointer = 1
End Sub
Private Sub cmdAdd_Click()
    Dim rsrh01 As rdoResultset 'conexion a un record set (tabla)
    Dim rsrh02 As rdoResultset
    Dim rsviat01 As rdoResultset
    Dim comfolio As Integer
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Dim larray As Byte
Dim fecha As Date
frmvalida_dinero.Show vbModal
Screen.MousePointer = 11
txtFolio = ""
Agrega = True
Call pmodoedicioncom 'Habilita solo modo cancelar
Call piniaciacom
Call PHabilFoldercom 'habilita folder de datos
Screen.MousePointer = 1
End Sub
Private Sub cmdCancel_Click()
Screen.MousePointer = 11
ReDim pfactura(1)
Call buttonnavigatemodecom
Call PLimpiaCamposcom
Screen.MousePointer = 1
Unload Me
frmMain.Show
End Sub
Private Sub cmdClose_Click()
Screen.MousePointer = 11
Unload Me
frmMain.Show
Screen.MousePointer = 1
End Sub
Private Sub cmddesgfact_Click()
Dim Lcont As Byte
Dim lhora As String
Dim lminu As String
Dim lFechaSal As String
Dim lFechaLle As String
If Not FVerifcom Then 'veffica fechas y horas llenas
Screen.MousePointer = 1
Exit Sub
End If
If FVerifFechas Then 'valida fechas correctas, fines de semana
Screen.MousePointer = 11
lFechaSal = FDaFechacom(Pfechas)
lFechaLle = FDaFechacom(Pfechal)
lFecha_masuno = Format(lFechaSal)
lhora = Format(DatePart("h", pHoraS), "00")
lminu = Format(DatePart("n", pHoraS), "00")
PhoraSal = lhora & ":" & lminu
lhora = Format(DatePart("h", pHoraL), "00")
lminu = Format(DatePart("n", pHoraL), "00")
PhoraLle = lhora & ":" & lminu
Pfolio = txtFolio
lFecha_hoy = Format(DatePart("m", Date), "00") + "/" + Format(DatePart("d", Date), "00") + "/" + Format(DatePart("yyyy",
Date), "0000")
lFechAnt = FDaFechacom(lblFAntici)
lstrq = "(" & "" & lFechaSal & "" & lFechaLle & "" & PhoraSal & "" & PhoraLle & "" & pNumDias & "" & Ptaxi & "" &
lFecha_hoy & "" & Psdinero & "" & PsGlobal & "" & PsConven & "")
lstrf =
"(d_rfecha_sal,d_rfecha_lle,ch_rhora_sal,ch_rhora_lle,nsi_rdias,m_comp_taxi,d_fec_comp,ch_recibi_dinyba,ch_cal_global,ch_con
vencion)"
bdrdroviat.Execute "update hist_viat_emp set" & lstrf & "=" & lstrq + _
" where s_folio=" & txtFolio
frmfactura.Show
frmcompro.Enabled = False
Else
Screen.MousePointer = 1
End If
Screen.MousePointer = 1
End Sub
Private Sub cmdEdit_Click()
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Dim rsrh01 As rdoResultset 'conexion a un record set (tabla)
Dim rsrh02 As rdoResultset
Dim rsviat01 As rdoResultset
Dim comfolio As Integer
Dim lbarray As Byte
Dim fecha As Date
frmvalida_dinero.Show vbModal
Screen.MousePointer = 11
btFolio = ""
Agrega = False
Call pmodoedicioncom 'Habilita solo modo cancelar
Call piniaciacom
Call PHabilFoldercom 'habilita folder de datos
Screen.MousePointer = 1
End Sub
Private Sub cmdFind_Click()
Dim lbarray As Byte
Dim Lsblanco As String
Lsblanco = ""
If fVerinumcom(btFolio, "el número de folio") Then
'instruccion sql abre tabla
Screen.MousePointer = 11
'icad = "select s_folio,nsi_num_emp,nsi_num_empsup,d_pfecha_sal,d_fech_ant,d_fech_aut_ant,d_fec_aut_comp from
hist_viat_emp where vch_logcancela is null and s_folio = " & btfolio.Text
'icad = "select * from hist_viat_emp where vch_logcancela is null and ch_vb_conta is null and s_folio = " & btFolio.Text

Set rsviat01 = bdrdoviat.OpenResultset(Name:=icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If Not rsviat01.EOF Then
Pfechas = Format(rsviat01!d_pfecha_sal, "mm/dd/yyyy")
Pempsup = rsviat01!nsi_num_empsup
End If
'icad = "select s_folio from hist_comp_emp where s_folio = " & btFolio.Text
Set rsviat02 = bdrdoviat.OpenResultset(Name:=icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
Screen.MousePointer = 1

If rsviat01.EOF Then 'no lo encuentra
MsgBox "Número de folio inexistente, cancelado o con V B de Contabilidad", vbExclamation, "Comprobación"
btFolio.Text = ""
btFolio.SetFocus
ElseIf IsNull(rsviat01!d_fech_aut_ant) Then
MsgBox "No ha sido autorizada la solicitud", vbExclamation, "Comprobación"
btFolio.SetFocus
ElseIf Not IsNull(rsviat01!d_fec_aut_comp) Then
MsgBox "La comprobacion no puede editarse, ya fue autorizada", vbExclamation, "Comprobación"
'la comprobacion ya ha sido autorizada y no puede editarse
btfolio.SetFocus
Else 'si lo encuentra jala los datos de la tabla
If Pnum_emp = rsviat01!nsi_num_emp Or Ptipo_usr = "R" Then
If IsNull(rsviat01!d_fec_aut_comp) Or Ptipo_usr = "R" Then
If (rsviat02.EOF And Agrega) Or (rsviat02.RowCount > 0 And Agrega = False) Then ' implica que no tiene comprobacion
btNEmp.Text = rsviat01("nsi_num_emp")
lblFAntici.Caption = Format(rsviat01!d_fech_ant, "dd/mm/yyyy")
For lbarray = 1 To 2
LblFol(lbarray).Caption = pstrfol + btFolio.Text
lblFecha(lbarray).Caption = pstrfecha + lblFAntici
num_area = cat_areas.ni_num_area and cat_honorarios.nsi_numhono = " & rsviat01!nsi_num_emp - 9000
Set rsrh01 = bdrdorh.OpenResultset(Name:=icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsrh01.EOF Then
'debe tener area, puesto y nivel si no aunque exista no lo lee
MsgBox "Número de empleado inexistente", vbExclamation, "Comprobación"
btNEmp.Text = ""
btNombre.Text = ""
btArea.Text = ""
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
btPuesto.Text = ""
btTel.Text = ""
btSal.Text = ""

Else
    btNombre.Text = rsrh01!ch_nombre
    btPuesto.Text = rsrh01!vch_nom_puesto
    btArea.Text = rsrh01!vch_nom_area
    'pintNivel = rsrh01!nsi_nivel 'hp
    nivelemp = rsrh01!nsi_nivel
End If

Else ' Empleado normal
Pfolio = btFolio
Icad = "select distinct datos_empleado.nsi_num_puesto,datos_empleado.vch_ap_paterno," +
"datos_empleado.vch_ap_materno,datos_empleado.vch_nombre,datos_empleado vch_ext_tel," +
"cat_puestos.vch_nom_puesto,datos_personales.ni_num_area,cat_areas vch_nom_area," +
"cat_puestos.nsi_nivel " +
"from datos_empleado, cat_puestos, datos_personales,cat_areas where " +
"cat_puestos.nsi_num_puesto = datos_empleado.nsi_num_puesto and " +
"cat_areas.ni_num_area = datos_personales.ni_num_area and " +
"datos_personales.nsi_num_emp = datos_empleado.nsi_num_emp and " +
"datos_empleado.nsi_num_emp = " & rsrvat01!nsi_num_emp
Screen.MousePointer = 11
Set rsrh01 = bdrdorh.OpenResultset(Name=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
Screen.MousePointer = 1
nivelemp = rsrh01!nsi_nivel

If IsNull(Trim(rsrh01!vch_ap_materno)) Then
    btNombre.Text = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_nombre"))
Else
    btNombre.Text = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_ap_materno")) + " " +
Trim(rsrh01("vch_nombre"))
End If
If IsNull(rsrh01("vch_ext_tel")) Then
    btTel.Text = ""
Else
    btTel.Text = rsrh01("vch_ext_tel")
End If
btPuesto.Text = rsrh01("vch_nom_puesto")
btArea.Text = rsrh01("vch_nom_area")
End If
Sstcom.TabEnabled(1) = True 'habilita los demas floders
Sstcom.TabEnabled(2) = True
cmdSave.Enabled = True
btNEmp.Enabled = False
btFolio.Enabled = False
cmdFind.Enabled = False
Call Pllenadestino
Call Pllenaobjetivo
If Agrega = False Then
    ' Call Pllenafact_dia quitar si no causa problemas
    ' hoy 30 de nov. de 1999 lo quite porque regresaba el monto de facturas
    ' bdrdoivat.Execute "update hist_fact_emp set
(m_monto_autofact,m_iva_monto)=(m_monto_realfact,m_monto_realfact*nsi_porcentaje/100) where s_folio=" & btfolio
End If
Else
    MsgBox "Si esta comprobacion ya la realizo para modificarla editela o si no a intoducido ningun dato iniciela como
nueva", vbExclamation, "Comprobación"
    'Unload Me
    'frmcompro.Show
    btFolio.SetFocus
    Call buttonnavigatemodecom
    Screen.MousePointer = 1
    imincorn = 1
    imin1com = 1
End If
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Else
    MsgBox "La comprobacion no puede editarse, ya fue autorizada", vbExclamation, "Comprobación"
    btFolio.SetFocus
End If
Else
    MsgBox "El Pasword no corresponde al Empleado/Folio", vbExclamation, "Comprobación"
    btFolio.SetFocus
End If
End If
Else
    btFolio.SetFocus
End If
End Sub
Private Sub cmdSave_Click()
    'On Error GoTo ManError
    Dim Lcont As Byte
    Dim lhora As String
    Dim lminu As String
    Dim lFechaSal As String
    Dim lFechaLle As String
    Screen.MousePointer = 11
    If Not FVerifcom Then 'verifica fechas y horas llenas
        Screen.MousePointer = vbNormal
        Exit Sub
    End If
    If FVeriFechas Then 'valida fechas correctas, fines de semana
        If PintDias <> pNumDias Then
            MsgBox "Verifique el detalle de destinos!!!!", vbExclamation, "Comprobación"
            Screen.MousePointer = vbNormal
            Exit Sub
        End If
        lFechaSal = FDaFechacom(lPfechas)
        lFechaLle = FDaFechacom(lPfechal)
        lFecha_masuno = Format(lFechaSal)
        lhora = Format(DatePart("h", pHoraS), "00")
        lminu = Format(DatePart("n", pHoraS), "00")
        PhoraSal = lhora & ":" & lminu
        lhora = Format(DatePart("h", pHoraL), "00")
        lminu = Format(DatePart("n", pHoraL), "00")
        PhoraLle = lhora & ":" & lminu

        lFecha_hoy = Format(DatePart("m", Date), "00") + "/" + Format(DatePart("d", Date), "00") + "/" + Format(DatePart("yyyy", Date), "00")
        lFechAnt = FDaFechacom(lblFAntici)
        lstrq = "(" & "" & lFechaSal & "" & lFechaLle & "" & PhoraSal & "" & PhoraLle & "" & pNumDias & "" & Ptaxi & "" & lFecha_hoy & "" & PsidInero & "" & PsGlobal & "" & PsConven & "")
        lstrf = "(d_rfecha_sal,d_rfecha_lle,ch_rhora_sal,ch_rhora_lle,nsi_rdias,m_comp_taxi,d_fec_comp,ch_recibi_dinyba,ch_cal_global,ch_convencion)"
        bdrdoviat.Execute "update hist_viat_emp set" & lstrf & "=" & lstrq + _
            " where s_folio=" & btFolio
        'borra las fechas que no correspondan
        bdrdoviat.Execute "delete from hist_fact_emp where s_folio=" & btFolio & "and d_fec_afecta <" & lFechaSal & ""
        dbSQLPassThrough
        bdrdoviat.Execute "delete from hist_fact_emp where s_folio=" & btFolio & "and d_fec_afecta >" & lFechaLle & ""
        dbSQLPassThrough
        bdrdoviat.Execute "delete from hist_comp_emp where s_folio=" & btFolio & "and d_fec_afecta <" & lFechaSal & ""
        dbSQLPassThrough
        bdrdoviat.Execute "delete from hist_comp_emp where s_folio=" & btFolio & "and d_fec_afecta >" & lFechaLle & ""
        dbSQLPassThrough
        'llena la parte de respuesta de objetivos
        If nrsviatobj01 > 0 Then
            Call Pobj_proy
        End If
        If nrsviatobj02 > 0 Then
            Call Pobj_clic
        End If
    End Sub
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes. (Visual Basic)

```
End If
If nrsviatobj03 > 0 Then
    Call Pobj_dele
End If
If nrsviatobj04 > 0 Then
    Call Pobj_otro
End If
IFechaSal = FDaFechacom(Pfechas)
' Actualiza el monto autorizado de facturas hotel y otros gastos día a día
If PsGlobal = "S" And pNumDias > 1 And PsConven = "N" Then
    'Calculo Globalizado
    MsgBox "Tu Calculo se realizara con cuota Globalizada", vbExclamation, "Comprobación"
    Call Pcuota_global
ElseIf PsGlobal = "N" And PsConven = "N" Then
    'Calculo Normal Cuota Diaria
    MsgBox "Tu Calculo se realizara con cuota Diaria", vbExclamation, "Comprobación"
    Call Pcuota_diaria
ElseIf PsConven = "S" Then
    'Calculo para Convenciones y Eventos Mineros
    MsgBox "Tu Calculo se realizara para Convención y/o Evento Minero", vbExclamation, "Comprobación"
    Call Pcuota_conven
Else
    MsgBox "El procedimiento de calculo no es el adecuado, cancele e inicie nuevamente", vbExclamation, "Comprobación"
    Screen.MousePointer = 1
End If
End If
Exit Sub
ManError:
Screen.MousePointer = 0
If rdoErrors.Count > 0 Then
    Dim lerror As rdoError
    For Each lerror In rdoErrors
        MsgBox F_RdoError(lerror), vbCritical, "Error"
        Call P_GuardaError(Me.Name, lerror.Number, F_RdoError(lerror))
    Next
    rdoErrors.Clear
Else
    MsgBox F_ManError(Err.Number), vbCritical, "Error"
    Call P_GuardaError(Me.Name, Err.Number, Err.Description)
End If
End Sub
Private Sub Form_Load()
    On Error GoTo ManError
    Call buttonnavigatemodecom
    Screen.MousePointer = 1
    lmincom = 1
    lmin1com = 1
    Centrar Me
Exit Sub
ManError:
Screen.MousePointer = 0
If rdoErrors.Count > 0 Then
    Dim lerror As rdoError
    For Each lerror In rdoErrors
        MsgBox F_RdoError(lerror), vbCritical, "Error"
        Call P_GuardaError(Me.Name, lerror.Number, F_RdoError(lerror))
    Next
    rdoErrors.Clear
Else
    MsgBox F_ManError(Err.Number), vbCritical, "Error"
    Call P_GuardaError(Me.Name, Err.Number, Err.Description)
End If
End Sub
Sub PLimpiaCamposcom()
    ' datos generales
    txtFolio.Text = ""
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
txtNombre.Text = ""
btpuesto.Text = ""
txtArea.Text = ""
bttel.Text = ""
destino
btestado.Text = ""
Ttxtsupaut.Text = ""
Ttxtsupaco.Text = ""
objetivos
lstproyecto.Text = ""
lstcliente.Text = ""
lstdelegacion.Text = ""
lstotro.Text = ""
btresobj.Text = ""
btcveobj = ""
End Sub
Public Sub controlbotones()
' Define que botones estaran habilitados cuando este por elegirse
' nueva solicitud o editar solicitud
Sstcom.Enabled = False
cmdSave.Enabled = False
cmdCancel.Enabled = False
cmdAdd.Enabled = True
cmdEdit.Enabled = True
cmdDelete.Enabled = False
cmdFind.Enabled = False
cmdClose.Enabled = True
End Sub

Public Sub iniciacom()
Dim lcad As String
Dim IRegDest As rdoResultset
Dim IRegEdo As rdoResultset
Dim lfecha As Date
Dim lhora As String
Dim lintNitem As Integer
Dim IRegEmpSup As rdoResultset
Dim IRegHistDest As rdoResultset
Dim IRegEmpAut As rdoResultset
Dim lindex As Integer
If Agrega Then
'inicializa variables y textboxes con fecha
'y hora actual del sistema para el registro
'de salida y llegada
'inicializa variables y botones para el
'control de objetivos
'datos de salida al destino
txtDiaS.Text = Format(Day(Date), "00")
txtMess.Text = Format(Month(Date), "00")
txtAnioS.Text = Format(Year(Date), "0000")
txtHoraS.Text = Format(Hour(Time), "00")
txtMinS.Text = Format(Minute(Time), "00")
'datos de llegada al destino
txtDiaLl.Text = Format(Day(Date), "00")
txtmesLl.Text = Format(Month(Date), "00")
txtAnioLl.Text = Format(Year(Date), "0000")
txthoraLl.Text = Format(Hour(Time), "00")
txtMinLl.Text = Format(Minute(Time), "00")
'inicializa variables para identificar el
'textbox al cual se le hara el cambio con
'el spinbutton
lmin = 1
lmin1 = 1
lObjNoC = 0
' otras variables
pNumDias = 0
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
PintDias = 0
nivelsup = 0
nvelemp = 0
'gastos y transportacion
pKm = 0
'pcosto_gas = 0
pBus = 0
pPeaje = 0
pFoto = 0
Psciaerea = ""
pLegales = 0
pOtros = 0
pltnAvion = ""
pltnBus = ""
pltnCarro = ""
pFechaAvion = ""
pHoraAvion = ""
pObs = ""
pRecMat = False
Else
'inicializa variables y textboxes con fecha
'y hora actual del sistema para el registro
'de salida y llegada
'inicializa variables y botones para el
'control de objetivos
'datos de salida al destino
btDiaS.Text = Format(Day(Date), "00")
btMess.Text = Format(Month(Date), "00")
btAnioS.Text = Format(Year(Date), "0000")
btHoraS.Text = Format(Hour(Time), "00")
btMinS T
```

```
Text = Format(Minute(Time), "00")
'datos de llegada al destino
btDiaLI.Text = Format(Day(Date), "00")
btmesLI.Text = Format(Month(Date), "00")
btAnoLI.Text = Format(Year(Date), "0000")
bthoraLI.Text = Format(Hour(Time), "00")
btMinLI.Text = Format(Minute(Time), "00")
'inicializa variables para identificar el
'textbox al cual se le hara el cambio con
'el spinbutton
lmin = 1
lMin1 = 1
lObjNoC = 0
' otras variables
pNumDias = 0
PintDias = 0
nivelsup = 0
nvelemp = 0
'gastos y transportacion
pKm = 0
'pcosto_gas = 0
pBus = 0
pPeaje = 0
pFoto = 0
Psciaerea = ""
pLegales = 0
pOtros = 0
pltnAvion = ""
pltnBus = ""
pltnCarro = ""
pFechaAvion = ""
pHoraAvion = ""
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
pObs = ""
pRecMat = False
End If
End Sub
Public Sub PHabilFoldercom()
    ' procedimiento que habilita solo al folder de datos generales
    ' y lo define como tab actual
    Sstcom.TabEnabled(0) = True
    Sstcom.TabEnabled(1) = False
    Sstcom.TabEnabled(2) = False
    Sstcom.Tab = 0
    txtFolio.Enabled = True
    txtFolio.SetFocus
End Sub
Public Sub pmdoedicioncom()
    ' habilita solo cancelar
    cmdSave.Enabled = False
    cmdCancel.Enabled = True
    cmdAdd.Enabled = False
    cmdEdit.Enabled = False
    cmdDelete.Enabled = False
    cmdFind.Enabled = True
    cmdClose.Enabled = True
    Sstcom.Enabled = True
End Sub
Public Sub buttonnavigatemodecom()
    ' Define que botones estaran habilitados cuando este por elegirse
    ' nueva solicitud o editar solicitud
    Sstcom.Enabled = False
    cmdSave.Enabled = False
    cmdCancel.Enabled = False
    cmdAdd.Enabled = True
    cmdEdit.Enabled = True
    cmdDelete.Enabled = False
    cmdFind.Enabled = False
    cmdClose.Enabled = True
End Sub
Public Sub Pcuota_diaria()
    ' Paga la Cuota Diaria debido a que no tiene todas las notas de Hotel
    ' y/o alguna nota de Alimentacion no cumple con Requisitos Fiscales o
    ' Comision < a 24 Hrs.
    Dim Lcont As Byte
    Dim Lni As Byte

    Dim lanio As String
    Dim lmes As String
    Dim l dias As String
    Dim Ls_fecha_bus As String
    Dim Lscuota As Single
    Dim Lntot_Hotel As Single
    Dim Lntot_Alimento As Single
    Dim Lmonto_Hotel As Single
    Dim Lmonto_Alimen As Single
    Dim Lmonto_gas As Single
    Dim Lntot_gas As Single
    Dim Lbflag_Hotel As Boolean
    Dim Lbflag_Alimen As Boolean
    Dim Lbflag_gas As Boolean
    Dim Lbflag_50 As Boolean
    Dim Lbflag_taxi As Boolean
    Dim Lmonto_Taxi As Single
    Dim Lntot_taxi As Single
    Dim Lsfactor As Single
    Dim Ls_cuotas_str As String
    Dim Ls_destinos_str As String
    bdrdoviat.Execute "update hist_fact_emp set
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
(m_monto_autofact,m_va_monto)=(m_monto_realfact,m_monto_realfact/100*nsi_porc_iva) where s_folio=" & Pfolio
' Selecciona todos los dias con comprobación
lcad = "select * from hist_comp_emp where s_folio=" & Pfolio & " order by d_fec_afecta"
Set rsviat01 = bdrdoviat.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
'Calcula el monto autorizado de Combustible
lcad = " select * from hist_kms where s_folio=" & Pfolio
Set rsviat03 = bdrdoviat.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsviat03.EOF Then
    Pns_consumo = 0
Else
    Pns_consumo = rsviat03!nd_consumo
End If
rsviat01.MoveFirst
Lntot_gas = 0
Lbflag_gas = False
'Barre dia por dia
For Lcont = 0 To rsviat01.RowCount - 1
    ' Call Pcheca_iva
    Lbflag_Hotel = False
    Lbflag_Alimen = False
    Lbflag_taxi = False
    lanio = Format(DatePart("yyyy", (rsviat01!d_fec_afecta)), "0000")
    lmes = Format(DatePart("m", (rsviat01!d_fec_afecta)), "00")
    ldia = Format(DatePart("d", (rsviat01!d_fec_afecta)), "00")
    Ls_fecha_bus = lmes & "/" & ldia & "/" & lanio
    ' Selecciona Todas las facturas correspondientes a ese día
    lcad = "select * from hist_fact_emp where s_folio=" & Pfolio & "and " +
    " d_fec_afecta=" & Ls_fecha_bus & " order by nsi_cve_gasto, ch_fiscal"
    Set rsviat02 = bdrdoviat.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
    ' calculo del porcentaje a pagar segun dia y hora
    If pNumDias > 1 Then
        If Lcont = 0 Then 'dia de salida
            If TimeValue(PhoraSal) > TimeValue("00:00") And TimeValue(PhoraSal) <= TimeValue("10:00") Then
                Lsfactor = 1
            ElseIf TimeValue(PhoraSal) > TimeValue("10:00") And TimeValue(PhoraSal) <= TimeValue("16:00") Then
                Lsfactor = 0.7
            Else
                Lsfactor = 0.3
            End If
            Lscuota = rsviat01!m_rcuota_dia * Lsfactor
        ElseIf Lcont > 0 And Lcont <= pNumDias - 2 Then 'dias intermedios
            Lscuota = rsviat01!m_rcuota_dia
        Else ' Ultimo dia
            If TimeValue(PhoraLle) > TimeValue("00:00") And TimeValue(PhoraLle) <= TimeValue("11:00") Then
                Lsfactor = 0.3
            ElseIf TimeValue(PhoraLle) > TimeValue("11:00") And TimeValue(PhoraLle) <= TimeValue("20:00") Then
                Lsfactor = 0.7
            Else
                Lsfactor = 1
            End If
            Lscuota = rsviat01!m_rcuota_dia * Lsfactor
        End If
    Else ' Cuota Diaria Menor a 24 Hrs
        Lscuota = rsviat01!m_rcuota_dia
    End If
    'Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Format(rsviat01!m_rcuota_dia * Lsfactor, "###,###.00"))) + "/"
    Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Lscuota)) + "/"
    Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_rdestino) + "/"
    If pNumDias = 1 Then
        Ls_cuotas_str = Trim(Str(Format(rsviat01!m_rcuota_dia, "###,###.00"))) + "/"
        Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_rdestino) + "/"
    End If
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
' Recalcula cuotas de los conceptos que rebasan la cuota día a día
Lntot_Hotel = 0
Lntot_Alimento = 0
Lntot_taxi = 0
Lbflag_50 = True

'Barre nota por nota cada día
For Lni = 1 To rsviat02.RowCount
  Select Case Val(rsviat02!nsi_cve_gasto)
    Case 0, 1 ' Monto Hotel e Impuesto Hotel
      If Lcont < pNumDias - 1 And pNumDias > 1 Then 'no ultimo día
        If Lbflag_Hotel = True Then ' Exedio la cuota diaria
          Lmonto_Hotel = 0
        Else
          Lmonto_Hotel = Val(rsviat02!m_monto_realfact)
          If Lntot_Hotel + Lmonto_Hotel + Lmonto_Hotel * rsviat02!nsi_porc_iva / 100 > Lscuota Then
            MsgBox "El monto del hotel excede la cuota del día numero " & Lcont + 1, vbExclamation, "Comprobación"
            If rsviat02!nsi_porc_iva > 0 Then
              Lmonto_Hotel = (Lscuota - Lntot_Hotel) / (1 + rsviat02!nsi_porc_iva / 100)
            Else
              Lmonto_Hotel = Lscuota - Lntot_Hotel
            End If
            Lbflag_Hotel = True 'puse la bandera para indicar que exedio la cuota
            If Lmonto_Hotel < 0 Then
              Lmonto_Hotel = 0
            End If
          End If
        End If
        Istrq = "(" & Lmonto_Hotel & ", " & Lmonto_Hotel * rsviat02!nsi_porc_iva / 100 & ")"
        Istrf = "(m_monto_autofact,m_iva_monto)"

        bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
          " where s_folio=" & txFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
          " and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
          " and ch_consecut=" & rsviat02!ch_consecut & ""
        Lntot_Hotel = Lntot_Hotel + Lmonto_Hotel + Lmonto_Hotel * rsviat02!nsi_porc_iva / 100
      End If

    Case 2, 3 ' Alimentacion
      'Si no comprueba hotel la tarifa es al 50 %
      If pNumDias > 0 And Lntot_Hotel = 0 And (Lcont - (pNumDias - 1)) < 0 And Lbflag_50 = True Then
        Lscuota = Lscuota / 2
        Lbflag_50 = False
      End If

      'Si la Primera nota de alimentacion no cumple Req. Fiscales
      If rsviat02!ch_fiscal = "N" And Lbflag_50 = True Then
        If Lntot_Hotel >= Lscuota / 2 Then
          'No hay problema lo restante es por lo menos el 50% de
          'la cuota para alimentacion y taxi si corresponde
        Else 'cuota de Hotel menor al 50%
          'Se da el 50% de la cuota y se pone en cero hotel para
          'que no afecte calculos de abajo
          Lscuota = Lscuota / 2
          Lbflag_50 = False
          Lntot_Hotel = 0
        End If
      End If
    End If
    If Lbflag_Hotel = True Or Lbflag_Alimen = True Then ' Exedio la cuota diaria
      Lmonto_Alimen = 0
    Else
      Lmonto_Alimen = Val(rsviat02!m_monto_realfact)
      If Lntot_Hotel + Lntot_Alimento + Lmonto_Alimen + Lmonto_Alimen * rsviat02!nsi_porc_iva / 100 > Lscuota Then
        MsgBox "El monto de Hotel mas Alimentación excede la cuota del día numero " & Lcont + 1, vbExclamation,
          "Comprobación"
        If rsviat02!nsi_porc_iva > 0 Then
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Lmonto_Alimen = (Lscuota - Lntot_Hotel - Lntot_Alimento) / (1 + rsviat02!nsi_porcentaje_iva / 100)
Else
  Lmonto_Alimen = Lscuota - Lntot_Hotel - Lntot_Alimento
End If
Lbflag_Alimen = True
If Lmonto_Alimen < 0 Then
  Lmonto_Alimen = 0
End If
End If
End If
Istrq = "(" & Lmonto_Alimen & "," & Lmonto_Alimen * rsviat02!nsi_porcentaje_iva / 100 & ")"
Istrf = "(m_monto_autofact,m_iva_monto)"
bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
" where s_folio=" & btFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
" and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
" and ch_consecut=" & rsviat02!ch_consecut & ""

Lntot_Alimento = Lntot_Alimento + Lmonto_Alimen + Lmonto_Alimen * rsviat02!nsi_porcentaje_iva / 100

Case 4, 5, 6, 7, 10, 11, 12, 13 ' Conceptos que se pagan totalmente
Case 8 ' Gasolina
If Lbflag_gas = True Then ' Exedio la cuota de Gasolina
  Lmonto_gas = 0
Else
  Lmonto_gas = Val(rsviat02!m_monto_realfact)
  If Lntot_gas + Lmonto_gas + Lmonto_gas * rsviat02!nsi_porcentaje_iva / 100 > Pns_consumo Then
    MsgBox "El monto de Combustible excede la cuota Autorizada", vbExclamation, "Comprobación"
    If rsviat02!nsi_porcentaje_iva > 0 Then
      Lmonto_gas = (Pns_consumo - Lntot_gas) / (1 + rsviat02!nsi_porcentaje_iva / 100)
    Else
      Lmonto_gas = Pns_consumo - Lntot_gas
    End If
    Lbflag_gas = True
    If Lmonto_gas < 0 Then
      Lmonto_gas = 0
    End If
  End If
End If
Istrq = "(" & Lmonto_gas & "," & Lmonto_gas * rsviat02!nsi_porcentaje_iva / 100 & ")"
Istrf = "(m_monto_autofact,m_iva_monto)"

bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
" where s_folio=" & btFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
" and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
" and ch_consecut=" & rsviat02!ch_consecut & ""

Lntot_gas = Lntot_gas + Lmonto_gas + Lmonto_gas * rsviat02!nsi_porcentaje_iva / 100

Case 9 And rsviat02!ch_taxi_aerop = "S" ' Taxi
If pNumDias = 1 Then ' Unico dia
  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
ElseIf Lcont = 0 Or pNumDias = Lcont + 1 Then ' Primero/Ultimo dia se incluye en tarifa diaria
  If Val(nivelemp) < 60 Or Val(Pempsup) > 0 Then ' se incluye taxi a mandos medios y superiores
    If Lbflag_Hotel = True Or Lbflag_Alimen = True Or Lbflag_taxi = True Then ' Exedio la cuota diaria
      Lmonto_Taxi = 0
    Else
      ' Taxi de Casa/Apto/Central
      ' If rsviat02!ch_taxi_aerop = "S" Then

      Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
      If (Lntot_Hotel + Lntot_Alimento + Lmonto_Taxi + Lmonto_Taxi * nsi_porcentaje_iva / 100) > Lscuota Then
        MsgBox "El monto de Hotel mas Alimentación mas Taxi excede la cuota del dia numero " & Lcont + 1,
vbExclamation, "Comprobación"
        If rsviat02!nsi_porcentaje_iva > 0 Then
          Lmonto_Taxi = (Lscuota - Lntot_Hotel - Lntot_Alimento) / (1 + rsviat02!nsi_porcentaje_iva / 100)
        Else

```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Lmonto_Taxi = Lscuota - (Lntot_Hotel + Lntot_Alimento)
End If
Lbflag_taxi = True
If Lmonto_Taxi < 0 Then
    Lmonto_Taxi = 0
End If
End If
'aqui debe afectar el taxi como suma
Lntot_taxi = Lntot_taxi + Lmonto_Taxi + Lmonto_Taxi * rsviat02!nsi_porcentaje_iva / 100
End If
End If
Else ' El taxi es libre para mandos bajos
    Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
End If
Else 'días intermedios
    Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
End If
Istrq = "(" & Lmonto_Taxi & "," & Lmonto_Taxi * rsviat02!nsi_porcentaje_iva / 100 & ")"
Istrf = "(m_monto_autofact,m_iva_monto)"

bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
" where s_folio=" & txtFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
" and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
" and ch_consecut=" & rsviat02!ch_consecut & ""

Case Else
End Select
rsviat02.MoveNext
Next
Lfecha_masuno = lFechaSal
Call fecha_mas_uno
lFechaSal = lFecha_regreso
rsviat01.MoveNext
Next
Istrq = "(vch_cuotas_dias,vch_destinos)"
Istrf = "(" & Ls_cuotas_str & "," & Ls_destinos_str & ")"

bdrdoviat.Execute "update hist_viat_emp set" & Istrq & "=" & Istrf & " where s_folio = " & Pfolio
MsgBox "Comprobacion grabada con exito", vbInformation, "Comprobación"
rdoEngine.rdoEnvironments(0).CommitTrans
buttonnavigatemodecom
ReDim pfactura(1)
Screen.MousePointer = 1
End Sub
Public Sub Pcuota_conven()
' Paga la Cuota Para Directores en Convenciones
' o para los que nos Directores en solo Convenciones
' es el mismo calculo que para cuota_globalizada solo que se elimina el
' Hospedaje (libre) al acumulado y la cuota se reduce al 50%
Dim Lcont As Byte
Dim Lni As Byte
Dim lanio As String
Dim lmes As String
Dim ldiar As String
Dim Ls_fecha_bus As String
Dim Lscuota As Single
Dim Lntot_Hotel As Single
Dim Lntot_Alimento As Single
Dim Lmonto_Hotel As Single
Dim Lmonto_Alimen As Single
Dim Lmonto_gas As Single
Dim Lntot_gas As Single
Dim Lbflag_Hotel As Boolean
Dim Lbflag_Alimen As Boolean
Dim Lbflag_gas As Boolean
Dim Lbflag_taxi As Boolean
Dim Lmonto_Taxi As Single
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Dim Lntot_taxi As Single
Dim Lsf

actor As Single
Dim Lns_Totcuotas As Single
Dim Lntot_Global As Single
Dim Ls_cuotas_str As String
Dim Ls_destinos_str As String

If pNumDias <= 1 Then
    MsgBox "Para Calculo de Convencion debe ser mayor a 24 Hrs", vbExclamation, "Comprobación"
    Screen.MousePointer = 1
Else
    bdrdoviat.Execute "update hist_fact_emp set
(m_monto_autofact,m_iva_monto)=(m_monto_realfact,m_monto_realfact/100*nsi_porc_iva) where s_folio=" & Pfolio
' Selecciona los dias con comprobación para sumar todas las cuotas
lcad = "select * from hist_comp_emp where s_folio =" & Pfolio & " order by d_fec_afecta"
Set rsviat01 = bdrdoviat.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

'Suma todas las cuotas en la comision dia por dia
rsviat01.MoveFirst
Lns_Totcuotas = 0
If rsviat01.RowCount = 1 Then ' Comision < a 24 Hrs
    Lns_Totcuotas = rsviat01!m_rcuota_dia
Else ' Comision > a 24 Hrs
    For Lcont = 0 To rsviat01.RowCount - 1
        If Lcont = 0 Then 'Primer dia
            If TimeValue(PhoraSal) > TimeValue("00:00") And TimeValue(PhoraSal) <= TimeValue("10:00") Then
                Lsfactor = 1
            ElseIf TimeValue(PhoraSal) > TimeValue("10:00") And TimeValue(PhoraSal) <= TimeValue("16:00") Then
                Lsfactor = 0.7
            Else
                Lsfactor = 0.3
            End If
            Lns_Totcuotas = Lns_Totcuotas + rsviat01!m_rcuota_dia * Lsfactor
            Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Format(rsviat01!m_rcuota_dia * Lsfactor / 2, "###,##0.00"))) + "/"
            Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_rdestino) + "/"
        ElseIf Lcont - (rsviat01.RowCount - 1) = 0 Then 'Ultimo Dia
            If TimeValue(PhoraLle) > TimeValue("00:00") And TimeValue(PhoraLle) <= TimeValue("11:00") Then
                Lsfactor = 0.3
            ElseIf TimeValue(PhoraLle) > TimeValue("11:00") And TimeValue(PhoraLle) <= TimeValue("20:00") Then
                Lsfactor = 0.7
            Else
                Lsfactor = 1
            End If
            Lns_Totcuotas = Lns_Totcuotas + rsviat01!m_rcuota_dia * Lsfactor
            Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Format(rsviat01!m_rcuota_dia * Lsfactor / 2, "###,##0.00"))) + "/"
            Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_rdestino) + "/"
        Else 'Dias Intermedios
            Lns_Totcuotas = Lns_Totcuotas + rsviat01!m_rcuota_dia
            Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Format(rsviat01!m_rcuota_dia / 2, "###,##0.00"))) + "/"
            Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_rdestino) + "/"
        End If

        rsviat01.MoveNext
    Next
End If

lstrq = "(vch_cuotas_dias,vch_destinos)"
lstrf = "(" & Ls_cuotas_str & "," & Ls_destinos_str & ")"
bdrdoviat.Execute "update hist_viat_emp set" & lstrq & "=" & lstrf & " where s_folio =" & Pfolio

' Cuota Globalizada al 50%
Lns_Totcuotas = Lns_Totcuotas / 2
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
'Calcula el monto autorizado de Combustible
lcad = "select * from hist_krms where s_folio=" & Pfolio
Set rsviat03 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsviat03.EOF Then
    Pns_consumo = 0
Else
    Pns_consumo = rsviat03!nd_consumo
End If
Lntot_gas = 0
Lbflag_gas = False
Lntot_Global = 0
Lbflag_Hotel = False
Lbflag_Alimen = False
Lbflag_taxi = False

'Barre dia por dia la comprobacion
rsviat01.MoveFirst
For Lcont = 0 To rsviat01.RowCount - 1
    lanio = Format(DatePart("yyyy", (rsviat01!d_fec_afecta)), "0000")
    lmes = Format(DatePart("m", (rsviat01!d_fec_afecta)), "00")
    ldia = Format(DatePart("d", (rsviat01!d_fec_afecta)), "00")
    Ls_fecha_bus = lmes & "/" & ldia & "/" & lanio

' Selecciona Todas las facturas correspondientes a ese dia
lcad = "select * from hist_fact_emp where s_folio=" & Pfolio & "and " + _
"d_fec_afecta=" & Ls_fecha_bus & " order by nsi_cve_gasto, ch_fiscal"

Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

'Barre nota por nota cada dia
rsviat02.MoveFirst
For Lni = 1 To rsviat02.RowCount
    Select Case Val(rsviat02!nsi_cve_gasto)

    Case 2, 3 ' Alimentacion
        MsgBox "2 o 3"
        If Lbflag_Hotel = True Or Lbflag_Alimen = True Then ' Exedio la cuota diaria
            Lmonto_Alimen = 0
        Else
            Lmonto_Alimen = Val(rsviat02!m_monto_realfact)
            If Lntot_Global + Lmonto_Alimen + Lmonto_Alimen * rsviat02!nsi_porcentaje_iva / 100 > Lns_Totcuotas Then
                MsgBox "El monto de Alimentación excede la cuota del dia numero " & Lcont + 1, vbExclamation, "Comprobación"
                If rsviat02!nsi_porcentaje_iva > 0 Then
                    Lmonto_Alimen = (Lns_Totcuotas - Lntot_Global) / (1 + rsviat02!nsi_porcentaje_iva / 100)
                Else
                    Lmonto_Alimen = Lns_Totcuotas - Lntot_Global
                End If
                Lbflag_Alimen = True
                If Lmonto_Alimen < 0 Then
                    Lmonto_Alimen = 0
                End If
            End If
        End If
        Istrq = "(" & Lmonto_Alimen & "," & Lmonto_Alimen * rsviat02!nsi_porcentaje_iva / 100 & ")"
        Istrf = "(m_monto_autofact,m_iva_monto)"

        bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
        " where s_folio=" & btFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
        " and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
        " and ch_consecut=" & rsviat02!ch_consecut & ""

        Lntot_Global = Lntot_Global + Lmonto_Alimen + Lmonto_Alimen * rsviat02!nsi_porcentaje_iva / 100

    Case 0, 1, 4, 5, 6, 7, 10, 11, 12, 13 ' Conceptos que se pagan totalmente
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
"MsgBox "4, 5, 6, 7, 10, 11, 12, 13"

Case 8 ' Gasolina
'MsgBox "B"
If Lbflag_gas = True Then ' Exedio la cuota de Gasolina
  Lmonto_gas = 0
Else
  Lmonto_gas = Val(rsviat02!m_monto_realfact)
  If Lntot_gas + Lmonto_gas + Lmonto_gas * rsviat02!nsi_porcentaje_iva / 100 > Pns_consumo Then
    MsgBox "El monto de Combustible excede la cuota Autorizada", vbExclamation, "Comprobación"
    If rsviat02!nsi_porcentaje_iva > 0 Then
      Lmonto_gas = (Pns_consumo - Lntot_gas) / (1 + rsviat02!nsi_porcentaje_iva / 100)
    Else
      Lmonto_gas = Pns_consumo - Lntot_gas
    End If
    Lbflag_gas = True
    If Lmonto_gas < 0 Then
      Lmonto_gas = 0
    End If
  End If
End If
End If
Lstrq = "(" & Lmonto_gas & "," & Lmonto_gas * rsviat02!nsi_porcentaje_iva / 100 & ")"
Lstrf = "(m_monto_autofact,m_iva_monto)"

bdrdoviat.Execute "update hist_fact_emp set" & Lstrf & "=" & Lstrq + _
" where s_folio=" & txtFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
" and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
" and ch_consecut=" & rsviat02!ch_consecut & ""

Lntot_gas = Lntot_gas + Lmonto_gas + Lmonto_gas * rsviat02!nsi_porcentaje_iva / 100

Case 9 ' Taxi
'MsgBox "9"
If pNumDias = 1 Then ' Unico dia
  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
ElseIf Lcont = 0 Or pNumDias = Lcont + 1 Then ' Primero/Ultimo dia se incluye en tarifa diaria
  If Val(nivelemp) < 60 Or Val(Permpsup) > 0 Then ' se incluye taxi a mandos medios y superiores
    If Lbflag_Hotel = True Or Lbflag_Alimen = True Or Lbflag_taxi = True Then ' Exedio la cuota diaria
      Lmonto_Taxi = 0
    Else
      'Taxi de Casa/Apto/Central
      'If rsviat02!ch_taxi_aerop = "S" Then
        Lmonto_Taxi = Val(rsviat02!m_monto_realfact)

      If (Lntot_Global + Lmonto_Taxi + Lmonto_Taxi * nsi_porcentaje_iva / 100) > Lns_Totcuotas Then
        MsgBox "El monto de Alimentación mas Taxi excede la cuota del dia numero " & Lcont + 1, vbExclamation,
"Comprobación"
        If rsviat02!nsi_porcentaje_iva > 0 Then
          Lmonto_Taxi = (Lns_Totcuotas - Lntot_Global) / (1 + rsviat02!nsi_porcentaje_iva / 100)
        Else
          Lmonto_Taxi = Lns_Totcuotas - Lntot_Global
        End If
        Lbflag_taxi = True
        If Lmonto_Taxi < 0 Then
          Lmonto_Taxi = 0
        End If
      End If
      'aqui debe afectar el taxi como suma
      Lntot_Global = Lntot_Global + Lmonto_Taxi + Lmonto_Taxi * rsviat02!nsi_porcentaje_iva / 100
    End If
  End If
Else ' El taxi es libre para mandos bajos
  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
End If
Else 'días intermedios
  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
End If
Istrq = "(" & Lmonto_Taxi & "," & Lmonto_Taxi * rsviat02!nsi_porcentaje / 100 & ")"
Istrf = "(m_monto_autofact,m_iva_monto)"

bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq +
" where s_folio=" & btFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto &
" and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact &
" and ch_consecut=" & rsviat02!ch_consecut & ""

Case Else
End Select
rsviat02.MoveNext
Next
Lfecha_masuno = IFechaSal
Call fecha_mas_uno
IFechaSal = Lfecha_regreso
rsviat01.MoveNext
Next
MsgBox "Comprobación grabada con éxito", vbInformation, "Comprobación"
buttonnavigatemodecom
ReDim pfactura(1)
Screen.MousePointer = 1
End If
End Sub

Public Sub Pcuota_global()
' Paga la Cuota Globalizada Autorizada el 2 de julio de 1999 y debe
' cumplir que presente todas las notas de hotel y todas las notas de
' Alimentación cumplan con Requisitos Fiscales y
' Comision > a 24 Hrs.
Dim Lcont As Byte
Dim Lni As Byte
Dim lanio As String
Dim lmes As String
Dim ldiad As String
Dim Ls_fecha_bus As String
Dim Lscuota As Single
Dim Lntot_Global
Dim Lmonto_Hotel As Single
Dim Lmonto_Alimen As Single
Dim Lmonto_gas As Single
Dim Lntot_gas As Single
Dim Lbflag_Hotel As Boolean
Dim Lbflag_Alimen As Boolean
Dim Lbflag_gas As Boolean
Dim Lmonto_Taxi As Single
Dim Lbflag_Rf As Boolean
Dim Ls_cuotas_str As String
Dim Ls_destinos_str As String
Dim Lsfactor As Single
Dim Lns_Totcuotas As Single

Lbflag_Rf = False
Lbflag_Hotel = False
Lbflag_Alimen = False
Lbflag_taxi = False
Lbflag_gas = False
Lntot_gas = 0
Lntot_Global = 0

'Valida que todas las facturas tengan requisitos fiscales
lcad = "select * from hist_fact_emp where s_folio =" & Pfolio
Set rsviat02 = bdrdoviat.OpenResultSet(Name=lcad,
Type:=rdOpenDynamic, LockType:=rdConcurValues)
rsviat02.MoveFirst
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuente (Visual Basic)

```
For Lcont = 0 To rsviat02.RowCount - 1
    If rsviat02!ch_fiscal = "N" And rsviat02!nsi_cve_gasto < 4 Then
        MsgBox "Alguna nota no cumple con Req. Fiscales debe comprobar por la cuota no Globalizada" & Lcont + 1,
        vbExclamation, "Comprobación"
        bdrdoviat.Execute "update hist_fact_emp set m_monto_autofact=0 where s_folio=" & Pfolio
        Lbflag_Rf = True
    End If
    rsviat02.MoveNext
Next

'Solamente entra a este calculo Comision > 24 Hrs. y Notas con R/F
If pNumDias > 1 And Lbflag_Rf = False Then
    bdrdoviat.Execute "update hist_fact_emp set
(m_monto_autofact,m_iva_monto)=(m_monto_realfact,m_monto_realfact/100*nsi_porc_iva) where s_folio=" & Pfolio
    'Selecciona todos los dias con comprobación
    lcad = "select * from hist_comp_emp where s_folio=" & Pfolio & " order by d_fec_afecta"
    Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lcad, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)

'Suma todas las cuotas en la comision dia por dia
rsviat01.MoveFirst
Lns_Totcuotas = 0
If rsviat01.RowCount = 1 Then 'Comisión < a 24 Hrs
    Lns_Totcuotas = rsviat01!m_rcuota_dia
Else 'Comision > a 24 Hrs
    For Lcont = 0 To rsviat01.RowCount - 1
        If Lcont = 0 Then 'Primer dia
            If TimeValue(PhoraSal) > TimeValue("00:00") And TimeValue(PhoraSal) <= TimeValue("10:00") Then
                Lsfactor = 1
            ElseIf TimeValue(PhoraSal) > TimeValue("10:00") And TimeValue(PhoraSal) <= TimeValue("16:00") Then
                Lsfactor = 0.7
            Else
                Lsfactor = 0.3
            End If
            Lns_Totcuotas = Lns_Totcuotas + rsviat01!m_rcuota_dia * Lsfactor
            Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Format(rsviat01!m_rcuota_dia * Lsfactor, "###.##00.00"))) + "/"
            Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_destino) + "/"
        ElseIf Lcont - (rsviat01.RowCount - 1) = 0 Then 'Ultimo Dia
            If TimeValue(PhoraLle) > TimeValue("00:00") And TimeValue(PhoraLle) <= TimeValue("11:00") Then
                Lsfactor = 0.3
            ElseIf TimeValue(PhoraLle) > TimeValue("11:00") And TimeValue(PhoraLle) <= TimeValue("20:00") Then
                Lsfactor = 0.7
            Else
                Lsfactor = 1
            End If
            Lns_Totcuotas = Lns_Totcuotas + rsviat01!m_rcuota_dia * Lsfactor
            Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Format(rsviat01!m_rcuota_dia * Lsfactor, "###.##00.00"))) + "/"
            Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_destino) + "/"
        Else 'Dias Intermedios
            Lns_Totcuotas = Lns_Totcuotas + rsviat01!m_rcuota_dia
            Ls_cuotas_str = Ls_cuotas_str + Trim(Str(Format(rsviat01!m_rcuota_dia, "###.##00.00"))) + "/"
            Ls_destinos_str = Ls_destinos_str + Trim(rsviat01!vch_destino) + "/"
        End If
        rsviat01.MoveNext
    Next
End If

Istrq = "(vch_cuotas_dias,vch_destinos)"
Istrf = "(" & Ls_cuotas_str & "," & Ls_destinos_str & ")"
bdrdoviat.Execute "update hist_viat_emp set" & Istrq & "=" & Istrf & " where s_folio =" & Pfolio

'Calcula el monto autorizado de Combustible
lcad = "select * from hist_kms where s_folio=" & Pfolio
Set rsviat03 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
If rsviat03.EOF Then
    Pns_consumo = 0
Else
    Pns_consumo = rsviat03.nd_consumo
End If

' Barre comisión dia por dia

rsviat01.MoveFirst
For Lcont = 0 To rsviat01.RowCount - 1
    lanio = Format(DatePart("yyyy"), (rsviat01!d_fec_afecta)), "0000"
    lmes = Format(DatePart("m"), (rsviat01!d_fec_afecta)), "00"
    ldia = Format(DatePart("d"), (rsviat01!d_fec_afecta)), "00"
    Ls_fecha_bus = lmes & "/" & ldia & "/" & lanio

' Selecciona Todas las facturas correspondientes a ese día
lcad = "select * from hist_fact_emp where s_folio =" & Pfolio & "and " + _
"d_fec_afecta=" & Ls_fecha_bus & " order by d_fec_afecta,nsi_cve_gasto,ch_fiscal"

Set rsviat02 = bdrdoviat.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

' Revisa que la suma de las notas de Hotel, Alimentacion y taxi
' no rebasen la cuota global, si la rebasan pondra todas las sig.
' notas de estos rubros en cero

'Barre Factura por Factura

For Lni = 1 To rsviat02.RowCount
    Select Case Val(rsviat02!nsi_cve_gasto)

    Case 0, 1 ' Monto Hotel e impuesto Hotel

        If Lcont < pNumDias - 1 And pNumDias > 1 Then 'no ultimo dia
            If Lbflag_Hotel = True Then ' Exedio la cuota global
                Lmonto_Hotel = 0
            Else
                Lmonto_Hotel = Val(rsviat02!m_monto_realfact)
                If Lntot_Global + Lmonto_Hotel + Lmonto_Hotel * rsviat02!nsi_porcentaje_iva / 100 > Lns_Totcuotas Then
                    MsgBox "El monto del hotel excede la cuota del dia numero " & Lcont + 1, vbExclamation, "Comprobación"
                    If rsviat02!nsi_porcentaje_iva > 0 Then
                        Lmonto_Hotel = (Lns_Totcuotas - Lntot_Global) / (1 + rsviat02!nsi_porcentaje_iva / 100)
                    Else
                        Lmonto_Hotel = Lns_Totcuotas - Lntot_Global
                    End If
                    Lbflag_Hotel = True 'puse la bandera para indicar que exedio la cuota
                    If Lmonto_Hotel < 0 Then
                        Lmonto_Hotel = 0
                    End If
                End If
            End If
            Istrq = "(" & Lmonto_Hotel & "," & Lmonto_Hotel * rsviat02!nsi_porcentaje_iva / 100 & ")"
            Istrf = "(m_monto_autofact,m_iva_monto)"

            bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
            " where s_folio=" & txfolio & "and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
            "and d_fec_afecta=" & Ls_fecha_bus & "and m_monto_realfact=" & rsviat02!m_monto_realfact & _
            "and ch_consecut=" & rsviat02!ch_consecut & ""

            Lntot_Global = Lntot_Global + Lmonto_Hotel + Lmonto_Hotel * rsviat02!nsi_porcentaje_iva / 100
        End If

    Case 2, 3 ' Alimentacion

        If Lbflag_Hotel = True Or Lbflag_Alimen = True Then ' Exedio la cuota diaria
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Lmonto_Alimen = 0
Else
  Lmonto_Alimen = Val(rsviat02!m_monto_realfact)
  If Lntot_Global + Lmonto_Alimen + Lmonto_Alimen * rsviat02!nsi_porcentaje / 100 > Lns_Totcuotas Then
    MsgBox "El monto de Hotel mas Alimentación excede la cuota del dia numero " & Lcont + 1, vbExclamation,
    "Comprobación"
    If rsviat02!nsi_porcentaje > 0 Then
      Lmonto_Alimen = (Lns_Totcuotas - Lntot_Global) / (1 + rsviat02!nsi_porcentaje / 100)
    Else
      Lmonto_Alimen = Lns_Totcuotas - Lntot_Global
    End If
    Lbflag_Alimen = True
    If Lmonto_Alimen < 0 Then
      Lmonto_Alimen = 0
    End If
  End If
  Lntot_Global = Lntot_Global + Lmonto_Alimen + Lmonto_Alimen * rsviat02!nsi_porcentaje / 100

  Istrq = "(" & Lmonto_Alimen & "," & Lmonto_Alimen * rsviat02!nsi_porcentaje / 100 & ")"
  Istrf = "(m_monto_autofact,m_iva_monto)"

  bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
  " where s_folio=" & txtFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
  " and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
  " and ch_consecut=" & rsviat02!ch_consecut & ""

  Lntot_Global = Lntot_Global + Lmonto_Alimen + Lmonto_Alimen * rsviat02!nsi_porcentaje / 100

Case 4, 5, 6, 7, 10, 11, 12, 13 ' Conceptos que se pagan totalmente

Case 8 ' Gasolina

If Lbflag_gas = True Then ' Exedio la cuota de Gasolina
  Lmonto_gas = 0
Else
  Lmonto_gas = Val(rsviat02!m_monto_realfact)
  If Lntot_gas + Lmonto_gas + Lmonto_gas * rsviat02!nsi_porcentaje / 100 > Pns_consumo Then
    MsgBox "El monto de Combustible excede la cuota Autorizada", vbExclamation, "Comprobación"
    If rsviat02!nsi_porcentaje > 0 Then
      Lmonto_gas = (Pns_consumo - Lntot_gas) / (1 + rsviat02!nsi_porcentaje / 100)
    Else
      Lmonto_gas = Pns_consumo - Lntot_gas
    End If
    Lbflag_gas = True
    If Lmonto_gas < 0 Then
      Lmonto_gas = 0
    End If
  End If
  Lntot_gas = Lntot_gas + Lmonto_gas + Lmonto_gas * rsviat02!nsi_porcentaje / 100

  Istrq = "(" & Lmonto_gas & "," & Lmonto_gas * rsviat02!nsi_porcentaje / 100 & ")"
  Istrf = "(m_monto_autofact,m_iva_monto)"

  bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
  " where s_folio=" & txtFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
  " and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
  " and ch_consecut=" & rsviat02!ch_consecut & ""

  Lntot_gas = Lntot_gas + Lmonto_gas + Lmonto_gas * rsviat02!nsi_porcentaje / 100

Case 9 And rsviat02!ch_taxi_aerop = "S" ' Taxi

  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
  If pNumDias = 1 Then ' Unico dia
    Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
  Elseif Lcont = 0 Or pNumDias = Lcont + 1 Then ' Primero/Ultimo dia se incluye en tarifa diaria
    If Val(nivelemp) < 60 Or Val(Pempsup) > 0 Then ' se incluye taxi a mandos medios y superiores
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
If Lbflag_Hotel = True Or Lbflag_Alimen = True Or Lbflag_taxi = True Then ' Exedio la cuota Global
  Lmonto_Taxi = 0
Else
  'Taxi de Casa/Apto/Central
  'If rsviat02!ch_taxi_aerop = "S" Then
  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
  If (Lntot_Global + Lmonto_Taxi + Lmonto_Taxi * nsi_porc_iva / 100) > Lns_Totcuotas Then
    MsgBox "El monto de Hotel mas Alimentación mas Taxi excede la cuota del día numero " & Lcont + 1,
vbExclamation, "Comprobación"
    If rsviat02!nsi_porc_iva > 0 Then
      Lmonto_Taxi = (Lns_Totcuotas - Lntot_Global) / (1 + rsviat02!nsi_porc_iva / 100)
    Else
      Lmonto_Taxi = Lns_Totcuotas - Lntot_Global
    End If
    Lbflag_taxi = True
    If Lmonto_Taxi < 0 Then
      Lmonto_Taxi = 0
    End If
  End If
  'aquí es donde afecta el taxi
  Lntot_Global = Lntot_Global + Lmonto_Taxi + Lmonto_Taxi * rsviat02!nsi_porc_iva / 100
'End If
End If
Else ' El taxi es libre para mandos bajos
  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
End If
Else 'días intermedios
  Lmonto_Taxi = Val(rsviat02!m_monto_realfact)
End If
Istrq = "(" & Lmonto_Taxi & "," & Lmonto_Taxi * rsviat02!nsi_porc_iva / 100 & ")"
Istrf = "(m_monto_autofact,m_iva_monto)"

bdrdoviat.Execute "update hist_fact_emp set" & Istrf & "=" & Istrq + _
" where s_folio=" & txtFolio & " and nsi_cve_gasto=" & rsviat02!nsi_cve_gasto & _
" and d_fec_afecta=" & Ls_fecha_bus & " and m_monto_realfact=" & rsviat02!m_monto_realfact & _
" and ch_consecut=" & rsviat02!ch_consecut & ""

Case Else
End Select
rsviat02.MoveNext
Next
Lfecha_masuno = IFechaSal
Call fecha_mas_uno
IFechaSal = Lfecha_regreso
Call Pcerogastos_dia
rsviat01.MoveNext
Next
MsgBox "Comprobacion grabada con exito", vbInformation, "Comprobación"
buttonnavigatemodecom
ReDim pfactura(1)
Screen.MousePointer = 1
Else
  MsgBox "Esta Comprobacion no es Mayor a 24 Hrs. o Alguna Nota no cumple con Requisitos Fiscales, debes volver a
recalcularla porque se pondra todo en cero", vbInformation, "Comprobación"
  Screen.MousePointer = 1
End If
End Sub

Public Sub Pobj_proy()
  For Lcont = 0 To nrsviatobj01 - 1
    If Len(arrproyecto(Lcont)) <= 255 Then
      Istrq = "(" & "" & Left(arrproyecto(Lcont), 255) & "," & "", null, null, null, null, null, null, null, null)
    ElseIf Len(arrproyecto(Lcont)) <= 510 Then
      Istrq = "(" & "" & Left(arrproyecto(Lcont), 255) & "," & Mid(arrproyecto(Lcont), 256, 255) &
"", null, null, null, null, null, null, null)
    ElseIf Len(arrproyecto(Lcont)) <= 765 Then
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
" & Mid(arrdelegacion(Lcont), 1021, 255) & " & Mid(arrdelegacion(Lcont), 1276, 255) & ".null,null,null,null)"
Elseif Len(arrdelegacion(Lcont)) <= 1785 Then
Istrq = "(" & "" & Left(arrdelegacion(Lcont), 255) & "" & Mid(arrdelegacion(Lcont), 256, 255) & _
"" & Mid(arrdelegacion(Lcont), 511, 255) & "" & Mid(arrdelegacion(Lcont), 766, 255) & _
"" & Mid(arrdelegacion(Lcont), 1021, 255) & "" & Mid(arrdelegacion(Lcont), 1276, 255) & _
"" & Mid(arrdelegacion(Lcont), 1531, 255) & ".null,null,null)"
Elseif Len(arrdelegacion(Lcont)) <= 2040 Then
Istrq = "(" & "" & Left(arrdelegacion(Lcont), 255) & "" & Mid(arrdelegacion(Lcont), 256, 255) & _
"" & Mid(arrdelegacion(Lcont), 511, 255) & "" & Mid(arrdelegacion(Lcont), 766, 255) & _
"" & Mid(arrdelegacion(Lcont), 1021, 255) & "" & Mid(arrdelegacion(Lcont), 1276, 255) & _
"" & Mid(arrdelegacion(Lcont), 1531, 255) & "" & Mid(arrdelegacion(Lcont), 1786, 255) & ".null,null)"
Elseif Len(arrdelegacion(Lcont)) <= 2295 Then
Istrq = "(" & "" & Left(arrdelegacion(Lcont), 255) & "" & Mid(arrdelegacion(Lcont), 256, 255) & _
"" & Mid(arrdelegacion(Lcont), 511, 255) & "" & Mid(arrdelegacion(Lcont), 766, 255) & _
"" & Mid(arrdelegacion(Lcont), 1021, 255) & "" & Mid(arrdelegacion(Lcont), 1276, 255) & _
"" & Mid(arrdelegacion(Lcont), 1531, 255) & "" & Mid(arrdelegacion(Lcont), 1786, 255) & _
"" & Mid(arrdelegacion(Lcont), 2041, 255) & ".null)"
Elseif Len(arrdelegacion(Lcont)) > 2295 Then
Istrq = "(" & "" & Left(arrdelegacion(Lcont), 255) & "" & Mid(arrdelegacion(Lcont), 256, 255) & _
"" & Mid(arrdelegacion(Lcont), 511, 255) & "" & Mid(arrdelegacion(Lcont), 766, 255) & _
"" & Mid(arrdelegacion(Lcont), 1021, 255) & "" & Mid(arrdelegacion(Lcont), 1276, 255) & _
"" & Mid(arrdelegacion(Lcont), 1531, 255) & "" & Mid(arrdelegacion(Lcont), 1786, 255) & _
"" & Mid(arrdelegacion(Lcont), 2041, 255) & "" & Mid(arrdelegacion(Lcont), 2296, 255) & ")"
End If
Istrf = "(vch_resp_obj,vch_resp_obj2,vch_resp_obj3,vch_resp_obj4," & _
"vch_resp_obj5,vch_resp_obj6,vch_resp_obj7,vch_resp_obj8,vch_resp_obj9,vch_resp_obj10)"

bdrvdiat.Execute "update hist_obj_del_emp set" & Istrf & "=" & Istrq + _
" where nsi_num_obj =" & Val(arrdelobj(Lcont)) & " and s_folio=" & bdfolio
Next
End Sub
Public Sub Pobj_clie()
For Lcont = 0 To nrsviatobj02 - 1
If Len(arrcliente(Lcont)) <= 255 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & ".null,null,null,null,null,null,null,null,null)"
Elseif Len(arrcliente(Lcont)) <= 510 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & _
".null,null,null,null,null,null,null)"
Elseif Len(arrcliente(Lcont)) <= 765 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & "" & Mid(arrcliente(Lcont), 511,
255) & ".null,null,null,null,null,null,null)"
Elseif Len(arrcliente(Lcont)) <= 1020 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & "" & Mid(arrcliente(Lcont), 511,
255) & "" & Mid(arrcliente(Lcont), 766, 255) & ".null,null,null,null,null,null)"
Elseif Len(arrcliente(Lcont)) <= 1275 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & _
"" & Mid(arrcliente(Lcont), 511, 255) & "" & Mid(arrcliente(Lcont), 766, 255) & _
"" & Mid(arrcliente(Lcont), 1021, 255) & ".null,null,null,null,null)"
Elseif Len(arrcliente(Lcont)) <= 1530 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & _
"" & Mid(arrcliente(Lcont), 511, 255) & "" & Mid(arrcliente(Lcont), 766, 255) & _
"" & Mid(arrcliente(Lcont), 1021, 255) & "" & Mid(arrcliente(Lcont), 1276, 255) & ".null,null,null,null)"
Elseif Len(arrcliente(Lcont)) <= 1785 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & _
"" & Mid(arrcliente(Lcont), 511, 255) & "" & Mid(arrcliente(Lcont), 766, 255) & _
"" & Mid(arrcliente(Lcont), 1021, 255) & "" & Mid(arrcliente(Lcont), 1276, 255) & _
"" & Mid(arrcliente(Lcont), 1531, 255) & ".null,null,null)"
Elseif Len(arrcliente(Lcont)) <= 2040 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & _
"" & Mid(arrcliente(Lcont), 511, 255) & "" & Mid(arrcliente(Lcont), 766, 255) & _
"" & Mid(arrcliente(Lcont), 1021, 255) & "" & Mid(arrcliente(Lcont), 1276, 255) & _
"" & Mid(arrcliente(Lcont), 1531, 255) & "" & Mid(arrcliente(Lcont), 1786, 255) & ".null,null)"
Elseif Len(arrcliente(Lcont)) <= 2295 Then
Istrq = "(" & "" & Left(arrcliente(Lcont), 255) & "" & Mid(arrcliente(Lcont), 256, 255) & _
"" & Mid(arrcliente(Lcont), 511, 255) & "" & Mid(arrcliente(Lcont), 766, 255) & _
"" & Mid(arrcliente(Lcont), 1021, 255) & "" & Mid(arrcliente(Lcont), 1276, 255) & _
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
" " & Mid(arrrciente(Lcont), 1531, 255) & " " & Mid(arrrciente(Lcont), 1786, 255) & _
" " & Mid(arrrciente(Lcont), 2041, 255) & " ,null)"
Elsif Len(arrrciente(Lcont)) > 2295 Then
Istrq = "(" & " " & Left(arrrciente(Lcont), 255) & " " & Mid(arrrciente(Lcont), 256, 255) & _
" " & Mid(arrrciente(Lcont), 511, 255) & " " & Mid(arrrciente(Lcont), 766, 255) & _
" " & Mid(arrrciente(Lcont), 1021, 255) & " " & Mid(arrrciente(Lcont), 1276, 255) & _
" " & Mid(arrrciente(Lcont), 1531, 255) & " " & Mid(arrrciente(Lcont), 1786, 255) & _
" " & Mid(arrrciente(Lcont), 2041, 255) & " " & Mid(arrrciente(Lcont), 2296, 255) & " )"
End If
Istrf = "(vch_resp_obj,vch_resp_obj2,vch_resp_obj3,vch_resp_obj4," +
"vch_resp_obj5,vch_resp_obj6,vch_resp_obj7,vch_resp_obj8,vch_resp_obj9,vch_resp_obj10)"

bdrdoviat.Execute "update hist_obj_cli_emp set" & Istrf & "=" & Istrq + _
" where nsi_num_obj =" & Val(arrrclobj(Lcont)) & " and s_folio=" & txtFolio
Next
End Sub
Public Sub Pobj_otro()
For Lcont = 0 To nrsviatobj04 - 1
If Len(arrotro(Lcont)) <= 255 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " ,null,null,null,null,null,null,null,null)"
Elsif Len(arrotro(Lcont)) <= 510 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & " ,null,null,null,null,null,null)"
Elsif Len(arrotro(Lcont)) <= 765 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & " " & Mid(arrotro(Lcont), 511, 255) & _
" ,null,null,null,null,null)"
Elsif Len(arrotro(Lcont)) <= 1020 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & " " & Mid(arrotro(Lcont), 511, 255) & " " & _
& Mid(arrotro(Lcont), 766, 255) & " ,null,null,null,null,null)"
Elsif Len(arrotro(Lcont)) <= 1275 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & _
" " & Mid(arrotro(Lcont), 511, 255) & " " & Mid(arrotro(Lcont), 766, 255) & _
" " & Mid(arrotro(Lcont), 1021, 255) & " ,null,null,null,null)"
Elsif Len(arrotro(Lcont)) <= 1530 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & _
" " & Mid(arrotro(Lcont), 511, 255) & " " & Mid(arrotro(Lcont), 766, 255) & _
" " & Mid(arrotro(Lcont), 1021, 255) & " " & Mid(arrotro(Lcont), 1276, 255) & " ,null,null,null,null)"
Elsif Len(arrotro(Lcont)) <= 1785 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & _
" " & Mid(arrotro(Lcont), 511, 255) & " " & Mid(arrotro(Lcont), 766, 255) & _
" " & Mid(arrotro(Lcont), 1021, 255) & " " & Mid(arrotro(Lcont), 1276, 255) & _
" " & Mid(arrotro(Lcont), 1531, 255) & " ,null,null,null)"
Elsif Len(arrotro(Lcont)) <= 2040 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & _
" " & Mid(arrotro(Lcont), 511, 255) & " " & Mid(arrotro(Lcont), 766, 255) & _
" " & Mid(arrotro(Lcont), 1021, 255) & " " & Mid(arrotro(Lcont), 1276, 255) & _
" " & Mid(arrotro(Lcont), 1531, 255) & " " & Mid(arrotro(Lcont), 1786, 255) & " ,null,null)"
Elsif Len(arrotro(Lcont)) <= 2295 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & _
" " & Mid(arrotro(Lcont), 511, 255) & " " & Mid(arrotro(Lcont), 766, 255) & _
" " & Mid(arrotro(Lcont), 1021, 255) & " " & Mid(arrotro(Lcont), 1276, 255) & _
" " & Mid(arrotro(Lcont), 1531, 255) & " " & Mid(arrotro(Lcont), 1786, 255) & _
" " & Mid(arrotro(Lcont), 2041, 255) & " ,null)"
Elsif Len(arrotro(Lcont)) > 2295 Then
Istrq = "(" & " " & Left(arrotro(Lcont), 255) & " " & Mid(arrotro(Lcont), 256, 255) & _
" " & Mid(arrotro(Lcont), 511, 255) & " " & Mid(arrotro(Lcont), 766, 255) & _
" " & Mid(arrotro(Lcont), 1021, 255) & " " & Mid(arrotro(Lcont), 1276, 255) & _
" " & Mid(arrotro(Lcont), 1531, 255) & " " & Mid(arrotro(Lcont), 1786, 255) & _
" " & Mid(arrotro(Lcont), 2041, 255) & " " & Mid(arrotro(Lcont), 2296, 255) & " )"
End If
Istrf = "(vch_resp_obj,vch_resp_obj2,vch_resp_obj3,vch_resp_obj4," +
"vch_resp_obj5,vch_resp_obj6,vch_resp_obj7,vch_resp_obj8,vch_resp_obj9,vch_resp_obj10)"
'hp truena
bdrdoviat.Execute "update hist_obj_otro_emp set" & Istrf & "=" & Istrq + _
" where nsi_num_obj =" & Val(arrotroobj(Lcont)) & " and s_folio=" & txtFolio
Next
End Sub
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes. (Visual Basic)

```
Private Sub Pllenafact_dia()
' para llenar facturas por dia en edicion
Dim IsqQ As String
Dim Isql As String
Dim Idias As Byte
Dim Ianio As String
Dim lmes As String
Dim ldia As String
Dim lbarray As Byte
Dim Lfecha_regreso As String

IsqQ = " select s_folio,d_fec_afecta,nsi_mum_dest,vch_rdestino,m_rcuota_dia " + _
"from hist_comp_emp where s_folio=" & txtFolio.Text
Screen.MousePointer = 11
Set rsviat01 = bdrdoviat.OpenResultSet(Name:=IsqQ, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
Screen.MousePointer = 1

If rsviat01.EOF Then 'no lo encuentra
MsgBox "no existen facturas para este comision", vbExclamation, "Comprobación"
Else
Idias = rsviat01.RowCount
End If
' once por numero de conceptos
rsviat01.MoveFirst
ReDim pfactura(11 * Idias)

For lbarray = 0 To Idias - 1

Isql = " select nsi_num_edo,nsi_num_dest,vch_nom_dest from cat_destinos " + _
"where nsi_num_dest=" & rsviat01!nsi_mum_dest
Set rsviatdest = bdrdoviat.OpenResultSet(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

Isql = " select nsi_num_edo,vch_estado from cat_estados " + _
"where nsi_num_edo=" & rsviatdest!nsi_num_edo
Set rsrh01 = bdrdorh.OpenResultSet(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

Ianio = Format(DatePart("yyy", rsviat01!d_fec_afecta), "0000")
lmes = Format(DatePart("m", rsviat01!d_fec_afecta), "00")
ldia = Format(DatePart("d", rsviat01!d_fec_afecta), "00")
Lfecha_regreso = lmes & "/" & ldia & "/" & Ianio

IsqQ = " select s_folio,d_fec_afecta,nsi_cve_gasto,m_monto_autofact,m_iva_monto, " + _
"vch_num_factura,nsi_porc_iva from hist_fact_emp " + _
"where s_folio=" & rsviat01!s_folio & " and " & "d_fec_afecta =" & Lfecha_regreso & ""
Set rsviatdia01 = bdrdoviat.OpenResultSet(Name:=IsqQ, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

IsqQ = " select s_folio,d_fec_afecta,nsi_mum_dest, " + _
"m_rcuota_dia from hist_comp_emp " + _
"where s_folio=" & rsviat01!s_folio & " and " & "d_fec_afecta =" & Lfecha_regreso & ""
Set rsviatmonto = bdrdoviat.OpenResultSet(Name:=IsqQ, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

pfactura(lbarray).DDest = rsrh01!vch_estado
pfactura(lbarray).Dedo = rsviatdest!vch_nom_dest
pfactura(lbarray).Cuota = rsviatmonto!m_rcuota_dia
pfactura(lbarray).NDest = rsviatmonto!nsi_mum_dest

For Li = 1 To rsviatdia01.RowCount
MsgBox Li
If rsviatdia01!nsi_cve_gasto = 0 Then
pfactura(lbarray).nHotel = rsviatdia01!m_monto_autofact
pfactura(lbarray).nivaHotel = rsviatdia01!nsi_porc_iva
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas Fuentes (Visual Basic)

```
    pfactura(lbarray).nfactHotel = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 1 Then
    pfactura(lbarray).nimpHotel = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaimpHotel = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactimpHotel = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 2 Then
    pfactura(lbarray).nAlimen = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaAlimen = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactAlimen = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 3 Then
    pfactura(lbarray).nprop_Alimen = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaprop_Alimen = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactprop_Alimen = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 4 Then
    pfactura(lbarray).nAvion = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaAvion = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactAvion = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 5 Then
    pfactura(lbarray).nTua = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaTua = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactTua = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 6 Then
    pfactura(lbarray).nBus = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaBus = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactBus = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 7 Then
    pfactura(lbarray).nPeaje = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaPeaje = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactPeaje = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 8 Then
    pfactura(lbarray).nGas = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaGas = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactGas = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 9 Then
    pfactura(lbarray).nTaxi = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaTaxi = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactTaxi = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 10 Then
    pfactura(lbarray).nFoto = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaFoto = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactFoto = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 11 Then
    pfactura(lbarray).nTelefono = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaTelefono = rsviatdia01!nsi_porc_iva
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
    pfactura(lbarray).nfactTelefono = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 12 Then
    pfactura(lbarray).nLegal = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaLegal = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactLegal = rsviatdia01!vch_num_factura
End If

If rsviatdia01!nsi_cve_gasto = 13 Then
    pfactura(lbarray).nOtros = rsviatdia01!m_monto_autofact
    pfactura(lbarray).nivaOtros = rsviatdia01!nsi_porc_iva
    pfactura(lbarray).nfactOtros = rsviatdia01!vch_num_factura
End If
rsviatdia01.MoveNext
Next
rsviat01.MoveNext
Next
Screen.MousePointer = 1
End Sub

Private Function FDaFechacom(fechaOld As Date) As String
    Dim Ianio As String
    Dim lmes As String
    Dim ldia As String
    Ianio = Format(DatePart("yyyy", fechaOld), "0000")
    lmes = Format(DatePart("m", fechaOld), "00")
    ldia = Format(DatePart("d", fechaOld), "00")
    FDaFechacom = lmes & "-" & ldia & "-" & Ianio
End Function

Private Sub pProcDestcom()

Dim lstrZon As String
Dim lstrCampZon As String
Dim lIndx As Integer
If Str(nivelsup) = "" Then
    pintNvelViat = nivelemp 'esto no debe ocurrir con choferes
Else
    pintNvelViat = nivelsup
End If

lIndx = 0
pAlimnt = 0
phospedj = 0
If pNumDias > 1 Then
    Select Case lIndx
        Case 0
            If pHoraS > TimeValue("00:00") And pHoraS <= TimeValue("10:00") Then
                pAlimnt = pfactura(lIndx).Cuota
            ElseIf pHoraS > TimeValue("10:00") And pHoraS <= TimeValue("16:00") Then
                pAlimnt = pfactura(lIndx).Cuota * 0.7
            Else
                pAlimnt = pfactura(lIndx).Cuota * 0.3
            End If
        Case Else
            pAlimnt = pAlimnt + pfactura(lIndx).Cuota
        End Select
    phospedj = 0
    phospedj = phospedj + pfactura(lIndx).Cuota * 0.6
    lIndx = lIndx + 1
    lIndx = lIndx - 1
    If pHoraL > TimeValue("00:00") And pHoraL <= TimeValue("11:00") Then
        pAlimnt = pAlimnt + pfactura(lIndx).Cuota * 0.3
    ElseIf pHoraL > TimeValue("11:00") And pHoraL <= TimeValue("20:00") Then
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
pAlimnt = pAlimnt + pfactura(Indx).Cuota * 0.7
Else
pAlimnt = pAlimnt + pfactura(Indx).Cuota
End If
Else
phospedj = 0
End If
' Luis para costo de gas
pdblCostAuto = (pKm / 5 * pLtrGas)
rsviat01.Close
End Sub

Private Function FVerifcom() As Boolean
FVerifcom = True
If txtFolio = "" Or _
txtDiaS = "" Or txtMess = "" Or txtAnioS = "" Or txtDiaLI = "" Or txtmesLI = "" _
Or txtAnioLI = "" Or txtHoraS = "" Or txtMinS = "" Or txthoraLI = "" Or _
txtMinLI = "" Then
FVerifcom = False
MsgBox "No ha ingresado todos los datos requeridos!!!", vbExclamation, "Comprobación"
End If
End Function

Private Sub Form_Unload(Cancel As Integer)
frmMain.Show
End Sub

Private Sub Istcliente_Click()
If Istcliente.ListIndex <> -1 Then
If nrsviatobj02 > 0 Then
idobjetivo = 2
btresobj.Text = arrcliente(Istcliente.ListIndex)
btresobj.SetFocus
btceobj.Text = "CLIENTE "
End If
End If
End Sub

Private Sub Istdelegacion_Click()
If Istdelegacion.ListIndex <> -1 Then
If nrsviatobj03 > 0 Then
idobjetivo = 3
btresobj.Text = arrdelegacion(Istdelegacion.ListIndex)
btresobj.SetFocus
btceobj.Text = "DELEGACION"
End If
End If
End Sub

Private Sub Istotro_Click()
If Istotro.ListIndex <> -1 Then
If nrsviatobj04 > 0 Then
idobjetivo = 4
btresobj.Text = arrotro(Istotro.ListIndex)
btresobj.SetFocus
btceobj.Text = "OTRO "
End If
End If
End Sub

Private Sub Istproyecto_Click()
If Istproyecto.ListIndex <> -1 Then
If nrsviatobj01 > 0 Then
idobjetivo = 1
btresobj.Text = arrproyecto(Istproyecto.ListIndex)
btresobj.SetFocus
End If
End If
End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
        btcveobj.Text = "PROYECTO "
    End If
End If
End Sub

Private Sub spnFechaLI_SpinDown()
If lmincom = 1 Then 'si son dias
    If Val(txtDiaLI.Text) <= 1 Then 'si llega al limite
        txtDiaLI.Text = "31"
    Else
        txtDiaLI.Text = Format(Val(txtDiaLI.Text) - 1, "00")
    End If
ElseIf lmincom = 2 Then 'si son meses
    If Val(txtmesLI.Text) <= 1 Then 'si llega al limites
        txtmesLI.Text = "12"
    Else
        txtmesLI.Text = Format(Val(txtmesLI.Text) - 1, "00")
    End If
Else 'si son años
    txtAnioLI.Text = Format(Val(txtAnioLI.Text) - 1, "00")
End If
End Sub

Private Sub spnFechaLI_SpinUp()
If lmincom = 1 Then 'si son dias
    If Val(txtDiaLI.Text) >= 31 Then 'si llega al limite
        txtDiaLI.Text = "01"
    Else
        txtDiaLI.Text = Format(Val(txtDiaLI.Text) + 1, "00")
    End If
ElseIf lmincom = 2 Then 'si son meses
    If Val(txtmesLI.Text) >= 12 Then 'si llega al limite
        txtmesLI.Text = "01"
    Else
        txtmesLI.Text = Format(Val(txtmesLI.Text) + 1, "00")
    End If
Else 'si son años
    txtAnioLI.Text = Format(Val(txtAnioLI.Text) + 1, "00")
End If
End Sub

Private Sub spnfechas_SpinDown()
If lmincom = 1 Then 'si son dias
    If Val(txtDiaS.Text) <= 1 Then 'si llega al limite
        txtDiaS.Text = "31"
    Else
        txtDiaS.Text = Format(Val(txtDiaS.Text) - 1, "00")
    End If
ElseIf lmincom = 2 Then 'si son meses
    If Val(txtMess.Text) <= 1 Then 'si llega al limites
        txtMess.Text = "12"
    Else
        txtMess.Text = Format(Val(txtMess.Text) - 1, "00")
    End If
Else 'si son años
    txtAnioS.Text = Format(Val(txtAnioS.Text) - 1, "00")
End If
End Sub

Private Sub spnfechas_SpinUp()
If lmincom = 1 Then 'si son dias
    If Val(txtDiaS.Text) >= 31 Then 'si llega al limite
        txtDiaS.Text = "01"
    Else
        txtDiaS.Text = Format(Val(txtDiaS.Text) + 1, "00")
    End If
End If
End Sub
```



## MIGRACIÓN DE SOFTWARE

### Anexos

### Programas fuentes (Visual Basic)

```
End If
ElseIf lmincom = 2 Then 'si son meses
  If Val(txtMess.Text) >= 12 Then 'si llega al limite
    txtMess.Text = "01"
  Else
    txtMess.Text = Format(Val(txtMess.Text) + 1, "00")
  End If
Else 'si son años
  txtAnioS.Text = Format(Val(txtAnioS.Text) + 1, "00")
End If
End Sub

Private Sub spnHoraLI_SpinDown()
  If lmin1com = 1 Then 'si son horas
    If Val(txthoraLI.Text) <= 0 Then 'si llega al limite
      txthoraLI.Text = "23"
    Else
      txthoraLI.Text = Format(Val(txthoraLI.Text) - 1, "00")
    End If
  Else 'si son minutos
    If Val(txtMinLI.Text) <= 0 Then 'si llega al limite
      txtMinLI.Text = "59"
    Else
      txtMinLI.Text = Format(Val(txtMinLI.Text) - 1, "00")
    End If
  End If
End Sub

Private Sub spnHoraLI_SpinUp()
  If lmin1com = 1 Then 'si son horas
    If Val(txthoraLI.Text) >= 23 Then 'si llega al limite
      txthoraLI.Text = "00"
    Else
      txthoraLI.Text = Format(Val(txthoraLI.Text) + 1, "00")
    End If
  Else 'si son minutos
    If Val(txtMinLI.Text) >= 59 Then 'si llega al limite
      txtMinLI.Text = "00"
    Else
      txtMinLI.Text = Format(Val(txtMinLI.Text) + 1, "00")
    End If
  End If
End Sub

Private Sub spnHoraS_SpinDown()
  If lmin1com = 1 Then 'si son horas
    If Val(txtHoraS.Text) <= 0 Then 'si llega al limite
      txtHoraS.Text = "23"
    Else
      txtHoraS.Text = Format(Val(txtHoraS.Text) - 1, "00")
    End If
  Else 'si son minutos
    If Val(txtMinS.Text) <= 0 Then 'si llega al limite
      txtMinS.Text = "59"
    Else
      txtMinS.Text = Format(Val(txtMinS.Text) - 1, "00")
    End If
  End If
End Sub

Private Sub spnHoraS_SpinUp()
  If lmin1com = 1 Then 'si son horas
    If Val(txtHoraS.Text) >= 23 Then 'si llega al limite
      txtHoraS.Text = "00"
    Else
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
btHoraS.Text = Format(Val(btHoraS.Text) + 1, "00")
End If
Else 'si son minutos
If Val(btMinS.Text) >= 59 Then 'si llega al limite
btMinS.Text = "00"
Else
btMinS.Text = Format(Val(btMinS.Text) + 1, "00")
End If
End If
End Sub

Private Sub btAnioLI_GotFocus()
Imincom = 3
End Sub

Private Sub btAnioS_GotFocus()
Imincom = 3
End Sub

Private Sub btDestPart_KeyPress(KeyAscii As Integer)
If KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
End Sub

Private Sub btDiall_GotFocus()
Imincom = 1
End Sub

Private Sub btDiaS_GotFocus()
Imincom = 1
End Sub

Private Sub btfolio_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
cmdFind.Value = True
End If
End Sub

Public Function fVerinumcom(bt As TextBox, msg As String) As Boolean
' verifica que el numero de empleado sea un valor numerico
If bt.Text = "" Then 'si no ha introducido ningun dato
MsgBox ("Debe introducir " + msg), vbExclamation, "Comprobación"
fVerinumcom = False
bt.SetFocus
ElseIf Not (IsNumeric(bt.Text)) Then 'si ha introducido un valor numerico
MsgBox ("Debe introducir un Número"), vbExclamation, "Comprobación"
fVerinumcom = False
bt.Text = ""
bt.SetFocus
Else
fVerinumcom = True
End If
End Function

Public Sub Pllenadestino()
Dim lfechas As Date
Dim lfechall As Date
Dim lhoras As String
Dim lmins As String
Dim lhorall As String
Dim lminll As String
Screen.MousePointer = 11
If Agrega Then
lcad = "select distinct hist_viat_emp.s_folio,hist_viat_emp.nsi_num_emp," +
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
"hist_viat_emp.ch_phora_sal, hist_viat_emp.ch_phora_ile, hist_viat_emp.m_comp_taxi," + _
"hist_viat_emp.d_pfecha_sal, hist_viat_emp.d_pfecha_ile," + _
"hist_viat_emp.ch_phora_ile, hist_viat_emp.ch_phora_sal," + _
"hist_viat_emp.nsi_num_empsup, hist_viat_emp.nsi_num_empaut, hist_viat_emp.vch_itin_auto, hist_viat_emp.m_tipo_camb,
" +
"hist_viat_emp.vch_itin_avion, hist_viat_emp.vch_itin_bus, hist_dest_emp.nsi_pnum_dest, cat_destinos.vch_nom_dest" + _
"from hist_viat_emp, hist_dest_emp, cat_destinos where " + _
"hist_dest_emp.s_folio = hist_viat_emp.s_folio and " + _
"cat_destinos.nsi_num_dest = hist_dest_emp.nsi_pnum_dest and " + _
"hist_viat_emp.s_folio = " & txtFolio.Text

Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
txtestado.Text = rsviat01("vch_nom_dest")
lfechas = rsviat01("d_pfecha_sal")
lfechall = rsviat01("d_pfecha_ile")
lhoras = rsviat01("ch_phora_sal")
lhorall = rsviat01("ch_phora_ile")
Else
lcad = "select distinct hist_viat_emp.s_folio, hist_viat_emp.nsi_num_emp," + _
"hist_viat_emp.ch_rhora_sal, hist_viat_emp.ch_rhora_ile, hist_viat_emp.m_comp_taxi," + _
"hist_viat_emp.d_rfecha_sal, hist_viat_emp.d_rfecha_ile," + _
"hist_viat_emp.ch_rhora_ile, hist_viat_emp.ch_rhora_sal," + _
"hist_viat_emp.nsi_num_empsup, hist_viat_emp.nsi_num_empaut, hist_viat_emp.vch_itin_auto, hist_viat_emp.m_tipo_camb,
" +
"hist_viat_emp.vch_itin_avion, hist_viat_emp.vch_itin_bus, hist_dest_emp.nsi_pnum_dest, cat_destinos.vch_nom_dest" + _
"from hist_viat_emp, hist_dest_emp, cat_destinos where " + _
"hist_dest_emp.s_folio = hist_viat_emp.s_folio and " + _
"cat_destinos.nsi_num_dest = hist_dest_emp.nsi_pnum_dest and " + _
"hist_viat_emp.s_folio = " & txtFolio.Text

Screen.MousePointer = 11
Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
txtestado.Text = rsviat01("vch_nom_dest")
lfechas = rsviat01("d_rfecha_sal")
lfechall = rsviat01("d_rfecha_ile")
lhoras = rsviat01("ch_rhora_sal")
lhorall = rsviat01("ch_rhora_ile")

If Agregado = False Then
    Ptaxi = rsviat01!m_comp_taxi
End If
If IsNull(rsviat01!m_comp_taxi) Then
    Ptaxi = 0
Else
    Ptaxi = rsviat01!m_comp_taxi
End If

End If
txtDiaS.Text = Format(DatePart("d", lfechas), "00")
txtMess.Text = Format(DatePart("m", lfechas), "00")
txtAnioS.Text = Format(DatePart("yyyy", lfechas), "00")
txtDiaLI.Text = Format(DatePart("d", lfechall), "00")
txtmesLI.Text = Format(DatePart("m", lfechall), "00")
txtAnioLI.Text = Format(DatePart("yyyy", lfechall), "00")
txtHoraS.Text = Format(DatePart("h", lhoras), "00")
txtMinS.Text = Format(DatePart("n", lhoras), "00")
txthoraLI.Text = Format(DatePart("h", lhorall), "00")
txtMinLI.Text = Format(DatePart("n", lhorall), "00")

lcad = "select distinct datos_empleado.vch_ap_paterno, datos_empleado.vch_ap_materno," + _
"datos_empleado.vch_nombre from datos_empleado where " + _
"datos_empleado.nsi_num_emp = " & rsviat01("nsi_num_empaut")
Set rsrh01 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Txtsupaut.Text = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_ap_materno")) + " " + Trim(rsrh01("vch_nombre"))

lcad = "select distinct datos_empleado.vch_ap_paterno, datos_empleado.vch_ap_materno, " + _
"datos_empleado.vch_nombre, datos_empleado.nsi_num_puesto, cat_puestos.nsi_nivel " + _
"from datos_empleado, cat_puestos where " + _
"cat_puestos.nsi_num_puesto = datos_empleado.nsi_num_puesto and " + _
"datos_empleado.nsi_num_emp = " & rsrvat01("nsi_num_empsup")
Set rsrh01 = bdrdorh.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
nempsup = rsrvat01!nsi_num_empsup

If rsrh01.EOF Then
    Txtsupaco.Text = ""
Else
    If IsNull(rsrh01!vch_ap_materno) Then
        Txtsupaco.Text = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_nombre"))
    Else
        Txtsupaco.Text = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_ap_materno")) + " " +
Trim(rsrh01("vch_nombre"))
    End If
    nivelsup = rsrh01!nsi_nivel
End If
If IsNull(rsrvat01("vch_itin_avion")) Then
    Chkavion.Value = Unchecked
Else
    Chkavion.Value = Checked
End If
If IsNull(rsrvat01("vch_itin_bus")) Then
    Chkbus.Value = Unchecked
Else
    Chkbus.Value = Checked
End If
If IsNull(rsrvat01("vch_itin_auto")) Then
    chkcarro.Value = Unchecked
Else
    chkcarro.Value = Checked
End If
Screen.MousePointer = 1
End Sub

Private Sub btHoraLI_GotFocus()
    lmin1com = 1
End Sub

Private Sub btHoraS_GotFocus()
    lmin1com = 1
End Sub

Private Sub txtMesLI_GotFocus()
    lmincom = 2
End Sub

Private Sub txtMesS_GotFocus()
    lmincom = 2
End Sub

Private Sub btMinLI_GotFocus()
    lmin1com = 2
End Sub

Private Sub btMinS_GotFocus()
    lmin1com = 2
End Sub
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Private Function FVeriFechas() As Boolean
Dim IFechAux As Date
Dim IfecAct As Date

FVeriFechas = True
Pfechas = txtDiaS.Text + "/" + txtMess.Text + "/" + txtAnioS.Text
Pfechal = txtDiaLI.Text + "/" + txtmesLI.Text + "/" + txtAnioLI.Text
pHoraS = TimeValue(txtHoraS.Text + ":" + txtMinS.Text)
pHoraL = TimeValue(txtHoraLI.Text + ":" + txtMinLI.Text)
pNumDias = (Pfechal - Pfechas)
IfecAct = Date
If (pNumDias < 0) Or (pFechaLI = Pfechas And pHoraLI - pHoraS <= 0) _
Or (Pfechas = IfecAct And (pHoraS - Time) <= 0) Then
MsgBox "Periodo de salida incorrecto!!", vbExclamation, "Comprobación"
pNumDias = 0
FVeriFechas = False
Else
If pNumDias >= 5 Then
'!hp MsgBox "su periodo de salida es mayor a cinco dias"
End If
For IFechAux = Pfechas To Pfechal
If Weekday(IFechAux) = _
vbSunday Or Weekday(IFechAux) = vbSaturday Then
' MsgBox "su periodo de salida incluye un fin de semana"

Exit For
End If
Next
pNumDias = pNumDias + 1
End If
MsgBox "dias: " & pNumDias, vbInformation, "Comprobación"
End Function

Public Sub Pllenaobjetivo()
Dim rsgest01 As rdoResultset
Screen.MousePointer = 11
Icad = "select distinct hist_obj_proy_emp.s_folio,hist_obj_proy_emp.ch_num_proy," + _
"hist_obj_proy_emp.nsi_num_obj, hist_obj_proy_emp.vch_desc_obj," + _
"hist_obj_proy_emp.vch_resp_obj,cat_objetivos.vch_nom_corto," + _
"hist_obj_proy_emp.vch_resp_obj2,hist_obj_proy_emp.vch_resp_obj3,hist_obj_proy_emp.vch_resp_obj4," + _
"hist_obj_proy_emp.vch_resp_obj5,hist_obj_proy_emp.vch_resp_obj6,hist_obj_proy_emp.vch_resp_obj7," + _
"hist_obj_proy_emp.vch_resp_obj8,hist_obj_proy_emp.vch_resp_obj9,hist_obj_proy_emp.vch_resp_obj10" + _
"from hist_obj_proy_emp,cat_objetivos where " + _
"hist_obj_proy_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_proy_emp.s_folio = " & txtFolio.Text
Set rsviatobj01 = bdrdoviat.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

ReDim arrproyobj(3)
ReDim arrclobj(3)
ReDim arrdelobj(3)
ReDim arrtroobj(3)

If rsviatobj01.EOF Then
!stproyecto.Enabled = False
Else
Lcont = 0
rsviatobj01.MoveFirst

While Not rsviatobj01.EOF
!stproyecto.AddItem Trim(rsviatobj01("ch_num_proy")) + "/" + rsviatobj01("vch_nom_corto")
!stproyecto.AddItem rsgest01("chdescorta") + "/" + rsviatobj01("vch_nom_corto")
arrproyobj(Lcont) = rsviatobj01("nsi_num_obj")

rsviatobj01.MoveNext
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Lcont = Lcont + 1
Wend
rsviatobj01.MoveFirst
End If

Icad = "select distinct hist_obj_cli_emp.s_folio, hist_obj_cli_emp.ni_cliente, " + _
"hist_obj_cli_emp.nsi_num_obj, hist_obj_cli_emp.vch_desc_obj, " + _
"hist_obj_cli_emp.vch_resp_obj, cat_objetivos.vch_nom_corto, " + _
"hist_obj_cli_emp.vch_resp_obj2, hist_obj_cli_emp.vch_resp_obj3, hist_obj_cli_emp.vch_resp_obj4, " + _
"hist_obj_cli_emp.vch_resp_obj5, hist_obj_cli_emp.vch_resp_obj6, hist_obj_cli_emp.vch_resp_obj7, " + _
"hist_obj_cli_emp.vch_resp_obj8, hist_obj_cli_emp.vch_resp_obj9, hist_obj_cli_emp.vch_resp_obj10 " + _
"from hist_obj_cli_emp, cat_objetivos where " + _
"hist_obj_cli_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_cli_emp.s_folio = " & txtFolio.Text
Set rsviatobj02 = bdrdoviat.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsviatobj02.EOF Then
    Istcliente.Enabled = False
Else
    Lcont = 0
    rsviatobj02.MoveFirst

    Icad = "select chnombre from clientes where nicliente = " & rsviatobj02("ni_cliente")
    Set rsgest02 = bdrdgest.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
    'viernes
    While Not rsviatobj02.EOF
        Istcliente.AddItem Str(rsviatobj02("ni_cliente")) + " " + rsviatobj02("vch_nom_corto")
        Istcliente.AddItem Trim(rsgest02("chnombre")) + "/" + rsviatobj02("vch_nom_corto")
        arrcliobj(Lcont) = rsviatobj02("nsi_num_obj")
        rsviatobj02.MoveNext
        Lcont = Lcont + 1
    Wend
    rsviatobj02.MoveFirst
End If

Icad = "select distinct hist_obj_del_emp.s_folio, hist_obj_del_emp.ni_num_dele, " + _
"hist_obj_del_emp.nsi_num_obj, hist_obj_del_emp.vch_desc_obj, " + _
"hist_obj_del_emp.vch_resp_obj, cat_objetivos.vch_nom_corto, " + _
"hist_obj_del_emp.vch_resp_obj2, hist_obj_del_emp.vch_resp_obj3, hist_obj_del_emp.vch_resp_obj4, " + _
"hist_obj_del_emp.vch_resp_obj5, hist_obj_del_emp.vch_resp_obj6, hist_obj_del_emp.vch_resp_obj7, " + _
"hist_obj_del_emp.vch_resp_obj8, hist_obj_del_emp.vch_resp_obj9, hist_obj_del_emp.vch_resp_obj10 " + _
"from hist_obj_del_emp, cat_objetivos where " + _
"hist_obj_del_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_del_emp.s_folio = " & txtFolio.Text
Set rsviatobj03 = bdrdoviat.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsviatobj03.EOF Then
    Istdelegacion.Enabled = False
Else
    Lcont = 0
    rsviatobj03.MoveFirst
    Icad = "select chdelegacion from catdelegaciones where ninumdel = " & rsviatobj03("ni_num_dele")
    Set rsgest03 = bdrdgest.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

    While Not rsviatobj03.EOF
        Istdelegacion.AddItem Str(rsviatobj03("ni_num_dele")) + " " + rsviatobj03("vch_nom_corto")
        Istdelegacion.AddItem Trim(rsgest03("chdelegacion")) + "/" + rsviatobj03("vch_nom_corto")
        arrdelobj(Lcont) = rsviatobj03("nsi_num_obj")
        rsviatobj03.MoveNext
        Lcont = Lcont + 1
    Wend
    rsviatobj03.MoveFirst
End If
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
lcad = "select distinct hist_obj_otro_emp.s_folio, hist_obj_otro_emp.vch_desc_visita, " + _
"hist_obj_otro_emp.nsi_num_obj, hist_obj_otro_emp.vch_resp_obj, cat_objetivos.vch_nom_corto, " + _
"hist_obj_otro_emp.vch_resp_obj2, hist_obj_otro_emp.vch_resp_obj3, hist_obj_otro_emp.vch_resp_obj4, " + _
"hist_obj_otro_emp.vch_resp_obj5, hist_obj_otro_emp.vch_resp_obj6, hist_obj_otro_emp.vch_resp_obj7, " + _
"hist_obj_otro_emp.vch_resp_obj8, hist_obj_otro_emp.vch_resp_obj9, hist_obj_otro_emp.vch_resp_obj10 " + _
"from hist_obj_otro_emp, cat_objetivos where " + _
"hist_obj_otro_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_otro_emp.s_folio = " & txtFolio.Text
Set rsviatobj04 = bdrdoviat.OpenResultset(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsviatobj04.EOF Then
    Istotro.Enabled = False
Else
    Lcont = 0
    rsviatobj04.MoveFirst
    While Not rsviatobj04.EOF
        Istotro.AddItem Trim(rsviatobj04("vch_desc_visita")) + "/" + rsviatobj04("vch_nom_corto")
        arrotroobj(Lcont) = rsviatobj04("nsi_num_obj")
        rsviatobj04.MoveNext
        Lcont = Lcont + 1
    Wend
    rsviatobj04.MoveFirst
End If

nrsviatobj01 = rsviatobj01.RowCount
nrsviatobj02 = rsviatobj02.RowCount
nrsviatobj03 = rsviatobj03.RowCount
nrsviatobj04 = rsviatobj04.RowCount

ReDim arproyecto(nrsviatobj01)
ReDim arrcliente(nrsviatobj02)
ReDim arrdelegacion(nrsviatobj03)
ReDim arrotro(nrsviatobj04)

If nrsviatobj01 > 0 Then
    For Lcont = 0 To nrsviatobj01 - 1
        If IsNull(rsviatobj01!vch_resp_obj) Then
            arproyecto(Lcont) = ""
        Else
            Call llenaproy
            arproyecto(Lcont) = Psc1 + Psc2 + Psc3 + Psc4 + Psc5 + Psc6 + Psc7 + Psc8 + Psc9 + Psc10
            arproyobj(Lcont) = rsviatobj01!nsi_num_obj
        End If
        rsviatobj01.MoveNext
    Next
End If

If nrsviatobj02 > 0 Then
    For Lcont = 0 To nrsviatobj02 - 1
        If IsNull(rsviatobj02!vch_resp_obj) Then
            arrcliente(Lcont) = ""
        Else
            Call llenaccli
            arrcliente(Lcont) = Psc1 + Psc2 + Psc3 + Psc4 + Psc5 + Psc6 + Psc7 + Psc8 + Psc9 + Psc10
            arrcliobj(Lcont) = rsviatobj02!nsi_num_obj
        End If
        rsviatobj02.MoveNext
    Next
End If

If nrsviatobj03 > 0 Then
    For Lcont = 0 To nrsviatobj03 - 1
        If IsNull(rsviatobj03!vch_resp_obj) Then
            arrdelegacion(Lcont) = ""
        Else
            Call llenadele
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
arrdelegacion(Lcont) = Psc1 + Psc2 + Psc3 + Psc4 + Psc5 + Psc6 + Psc7 + Psc8 + Psc9 + Psc10
arrdelobj(Lcont) = rsviatobj03!nsi_num_obj
End If
rsviatobj03.MoveNext
Next
End If

If nrsviatobj04 > 0 Then
For Lcont = 0 To nrsviatobj04 - 1
If IsNull(rsviatobj04!vch_resp_obj) Then
arrotro(Lcont) = ""
Else
Call llenaotro
arrotro(Lcont) = Psc1 + Psc2 + Psc3 + Psc4 + Psc5 + Psc6 + Psc7 + Psc8 + Psc9 + Psc10
arrotroobj(Lcont) = rsviatobj04!nsi_num_obj
End If
rsviatobj04.MoveNext
Next
End If
Screen.MousePointer = 1
End Sub

Private Sub btproyecto_Click()
'actualiza los datos seleccionaos a las listas

If lstproyecto.ListIndex <> "" Then
Ibiresproy.Visible = True
btproyecto.Visible = True
btproyecto.Text = lstproyecto.ListIndex
Else
IbDescObj.Visible = False
btDesObj.Visible = False
txtDesObj.Text = ""
End If
End Sub

Private Sub btresobj_Change()
If lstotro.ListIndex <> -1 Or lstdelegacion.ListIndex <> -1 Or istcliente.ListIndex <> -1 Or lstproyecto.ListIndex <> -1 Then
Select Case idobjetivo
Case 1
arrproyecto(lstproyecto.ListIndex) = btresobj.Text
Case 2
arrcliente(istcliente.ListIndex) = btresobj.Text
Case 3
arrdelegacion(lstdelegacion.ListIndex) = btresobj.Text
Case 4
If cmdCancel = -1 Then
Else
arrotro(lstotro.ListIndex) = btresobj.Text
End If
End Select
Else
MsgBox "Debes Seleccionar un Objetivo del recuadro izquierdo", vbExclamation, "Comprobación"
btresobj = ""
End If
End Sub

Public Sub fecha_mas_uno()
Dim Ianio As String
Dim lmes As String
Dim ldia As String
Lfecha_masuno = Lfecha_masuno + 1
Ianio = Format(DatePart("yyyy", Lfecha_masuno), "0000")
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Imes = Format(DatePart("m", Lfecha_masuno), "00")
Idia = Format(DatePart("d", Lfecha_masuno), "00")
Lfecha_regreso = Imes & "/" & Idia & "/" & Ianio
End Sub

Private Sub Pcheca_iva()
' chequea el iva de todos los conceptos dia a dia

If pfactura(Lcont2).nivaHotel = 0 Then
    Piva_Hotel = 0
Else
    Piva_Hotel = pfactura(Lcont2).nivaHotel * 0.01
End If

If pfactura(Lcont2).nivaImpHotel = 0 Then
    Piva_impHotel = 0
Else
    Piva_impHotel = pfactura(Lcont2).nivaImpHotel * 0.01
End If

If pfactura(Lcont2).nivaAlimen = 0 Then
    Piva_Alimen = 0
Else
    Piva_Alimen = pfactura(Lcont2).nivaAlimen * 0.01
End If

If pfactura(Lcont2).nivaprop_Alimen = 0 Then
    Pivaprop_Alimen = 0
Else
    Pivaprop_Alimen = pfactura(Lcont2).nivaprop_Alimen * 0.01
End If

If pfactura(Lcont2).nivaAvion = 0 Then
    Piva_Avion = 0
Else
    Piva_Avion = pfactura(Lcont2).nivaAvion * 0.01
End If

If pfactura(Lcont2).nivaTua = 0 Then
    Piva_Tua = 0
Else
    Piva_Tua = pfactura(Lcont2).nivaTua * 0.01
End If

If pfactura(Lcont2).nivaBus = 0 Then
    Piva_Bus = 0
Else
    Piva_Bus = pfactura(Lcont2).nivaBus * 0.01
End If

If pfactura(Lcont2).nivaPeaje = 0 Then
    Piva_peaje = 0
Else
    Piva_peaje = pfactura(Lcont2).nivaPeaje * 0.01
End If

If pfactura(Lcont2).nivaGas = 0 Then
    Piva_Gas = 0
Else
    Piva_Gas = pfactura(Lcont2).nivaGas * 0.01
End If

If pfactura(Lcont2).nivaTaxi = 0 Then
    Piva_Taxi = 0
Else
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Piva_Taxi = pfactura(Lcont2).nivaTaxi * 0.01
End If

If pfactura(Lcont2).nivaFoto = 0 Then
    Piva_Foto = 0
Else
    Piva_Foto = pfactura(Lcont2).nivaFoto * 0.01
End If

If pfactura(Lcont2).nivaTelefono = 0 Then
    Piva_Telefono = 0
Else
    Piva_Telefono = pfactura(Lcont2).nivaTelefono * 0.01
End If

If pfactura(Lcont2).nivaLegal = 0 Then
    Piva_Legal = 0
Else
    Piva_Legal = pfactura(Lcont2).nivaLegal * 0.01
End If

If pfactura(Lcont2).nivaOtros = 0 Then
    Piva_Otros = 0
Else
    Piva_Otros = pfactura(Lcont2).nivaOtros * 0.01
End If
Lcont2 = Lcont2 + 1
End Sub
```

```
Private Sub Pcerogastos_dia()
    Piva_Hotel = 0
    Piva_impHotel = 0
    Piva_Alimen = 0
    Pivaprop_Alimen = 0
    Piva_Avion = 0
    Piva_Tua = 0
    Piva_Bus = 0
    Piva_peaje = 0
    Piva_Gas = 0
    Piva_Taxi = 0
    Piva_Foto = 0
    Piva_Telefono = 0
    Piva_Legal = 0
    Piva_Otros = 0
End Sub

Private Sub btresobj_KeyPress(KeyAscii As Integer)
    Dim car As String * 1
    car = UCase(Chr(KeyAscii))
    KeyAscii = Asc(car)
    'If (KeyAscii <= 90 And KeyAscii >= 65) Or (KeyAscii <= 122 And KeyAscii >= 97) Or (KeyAscii <= 57 And KeyAscii >= 48) Or
    KeyAscii = 46 Or KeyAscii = 44 Or KeyAscii = 32 Or KeyAscii = 58 Or KeyAscii = 59 Then
        ' Caracteres Validos
    'Else
        MsgBox ("Símbolo incorrecto"), vbExclamation, "Comprobación"
        KeyAscii = 0
    'End If

    If KeyAscii = 13 Or KeyAscii = 34 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
        KeyAscii = 0
        Beep
    End If
End Sub

Public Sub pllenaotro()
```



```
Psc1 = ""
Psc2 = ""
Psc3 = ""
Psc4 = ""
Psc5 = ""
Psc6 = ""
Psc7 = ""
Psc8 = ""
Psc9 = ""
Psc10 = ""

If Not IsNull(rsviatobj04!vch_resp_obj) Then
    Psc1 = rsviatobj04!vch_resp_obj
End If

If Not IsNull(rsviatobj04!vch_resp_obj2) Then
    Psc2 = rsviatobj04!vch_resp_obj2
End If

If Not IsNull(rsviatobj04!vch_resp_obj3) Then
    Psc3 = rsviatobj04!vch_resp_obj3
End If

If Not IsNull(rsviatobj04!vch_resp_obj4) Then
    Psc4 = rsviatobj04!vch_resp_obj4
End If

If Not IsNull(rsviatobj04!vch_resp_obj5) Then
    Psc5 = rsviatobj04!vch_resp_obj5
End If

If Not IsNull(rsviatobj04!vch_resp_obj6) Then
    Psc6 = rsviatobj04!vch_resp_obj6
End If

If Not IsNull(rsviatobj04!vch_resp_obj7) Then
    Psc7 = rsviatobj04!vch_resp_obj7
End If

If Not IsNull(rsviatobj04!vch_resp_obj8) Then
    Psc8 = rsviatobj04!vch_resp_obj8
End If

If Not IsNull(rsviatobj04!vch_resp_obj9) Then
    Psc9 = rsviatobj04!vch_resp_obj9
End If

If Not IsNull(rsviatobj04!vch_resp_obj10) Then
    Psc10 = rsviatobj04!vch_resp_obj10
End If

End Sub
Public Sub pitenacii()
    Psc1 = ""
    Psc2 = ""
    Psc3 = ""
    Psc4 = ""
    Psc5 = ""
    Psc6 = ""
    Psc7 = ""
    Psc8 = ""
    Psc9 = ""
    Psc10 = ""

    If Not IsNull(rsviatobj02!vch_resp_obj) Then
        Psc1 = rsviatobj02!vch_resp_obj
    End If
End Sub
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
End If

If Not IsNull(rsviatobj02!vch_resp_obj2) Then
    Psc2 = rsviatobj02!vch_resp_obj2
End If

If Not IsNull(rsviatobj02!vch_resp_obj3) Then
    Psc3 = rsviatobj02!vch_resp_obj3
End If

If Not IsNull(rsviatobj02!vch_resp_obj4) Then
    Psc4 = rsviatobj02!vch_resp_obj4
End If

If Not IsNull(rsviatobj02!vch_resp_obj5) Then
    Psc5 = rsviatobj02!vch_resp_obj5
End If

If Not IsNull(rsviatobj02!vch_resp_obj6) Then
    Psc6 = rsviatobj02!vch_resp_obj6
End If

If Not IsNull(rsviatobj02!vch_resp_obj7) Then
    Psc7 = rsviatobj02!vch_resp_obj7
End If

If Not IsNull(rsviatobj02!vch_resp_obj8) Then
    Psc8 = rsviatobj02!vch_resp_obj8
End If

If Not IsNull(rsviatobj02!vch_resp_obj9) Then
    Psc9 = rsviatobj02!vch_resp_obj9
End If

If Not IsNull(rsviatobj02!vch_resp_obj10) Then
    Psc10 = rsviatobj02!vch_resp_obj10
End If

End Sub

Public Sub pllenaproy()
    Psc1 = ""
    Psc2 = ""
    Psc3 = ""
    Psc4 = ""
    Psc5 = ""
    Psc6 = ""
    Psc7 = ""
    Psc8 = ""
    Psc9 = ""
    Psc10 = ""

    If Not IsNull(rsviatobj01!vch_resp_obj1) Then
        Psc1 = rsviatobj01!vch_resp_obj1
    End If

    If Not IsNull(rsviatobj01!vch_resp_obj2) Then
        Psc2 = rsviatobj01!vch_resp_obj2
    End If

    If Not IsNull(rsviatobj01!vch_resp_obj3) Then
        Psc3 = rsviatobj01!vch_resp_obj3
    End If

    If Not IsNull(rsviatobj01!vch_resp_obj4) Then
        Psc4 = rsviatobj01!vch_resp_obj4
    End If
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
End If

If Not IsNull(rsviatobj01!vch_resp_obj5) Then
    Psc5 = rsviatobj01!vch_resp_obj5
End If

If Not IsNull(rsviatobj01!vch_resp_obj6) Then
    Psc6 = rsviatobj01!vch_resp_obj6
End If

If Not IsNull(rsviatobj01!vch_resp_obj7) Then
    Psc7 = rsviatobj01!vch_resp_obj7
End If

If Not IsNull(rsviatobj01!vch_resp_obj8) Then
    Psc8 = rsviatobj01!vch_resp_obj8
End If

If Not IsNull(rsviatobj01!vch_resp_obj9) Then
    Psc9 = rsviatobj01!vch_resp_obj9
End If

If Not IsNull(rsviatobj01!vch_resp_obj10) Then
    Psc10 = rsviatobj01!vch_resp_obj10
End If

End Sub

Public Sub llenadele()
    Psc1 = ""
    Psc2 = ""
    Psc3 = ""
    Psc4 = ""
    Psc5 = ""
    Psc6 = ""
    Psc7 = ""
    Psc8 = ""
    Psc9 = ""
    Psc10 = ""

    If Not IsNull(rsviatobj03!vch_resp_obj) Then
        Psc1 = rsviatobj03!vch_resp_obj
    End If

    If Not IsNull(rsviatobj03!vch_resp_obj2) Then
        Psc2 = rsviatobj03!vch_resp_obj2
    End If

    If Not IsNull(rsviatobj03!vch_resp_obj3) Then
        Psc3 = rsviatobj03!vch_resp_obj3
    End If

    If Not IsNull(rsviatobj03!vch_resp_obj4) Then
        Psc4 = rsviatobj03!vch_resp_obj4
    End If

    If Not IsNull(rsviatobj03!vch_resp_obj5) Then
        Psc5 = rsviatobj03!vch_resp_obj5
    End If

    If Not IsNull(rsviatobj03!vch_resp_obj6) Then
        Psc6 = rsviatobj03!vch_resp_obj6
    End If
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
If Not IsNull(rsviatobj03!vch_resp_obj7) Then  
    Psc7 = rsviatobj03!vch_resp_obj7  
End If
```

```
If Not IsNull(rsviatobj03!vch_resp_obj8) Then  
    Psc8 = rsviatobj03!vch_resp_obj8  
End If
```

```
If Not IsNull(rsviatobj03!vch_resp_obj9) Then  
    Psc9 = rsviatobj03!vch_resp_obj9  
End If
```

```
If Not IsNull(rsviatobj03!vch_resp_obj10) Then  
    Psc10 = rsviatobj03!vch_resp_obj10  
End If
```

End Sub

'frmdestinos.frm Forma de captura de Estado/municipio del Modulo  
de Viaticos

'Autor: Luis Hernández Pérez

'Tablas:

'Base de Datos Tablas que Utiliza Lectura Escritura

'Viaticos Hist\_viat\_emp sql sql

'Viaticos Cat\_destinos sql

'Viaticos Hist\_dest\_emp sql sql

'Rec. Humanos Cat\_estados sql

'Forma que Llama: frmsolicitud

'Llamada por: frmsolicitud

'Modificada por:

Dim Isql As String

Dim IstrDest As String

Dim IstrReg As String

Dim MRegEst As rdoResultset

Dim MRegCont As rdoResultset

Dim MRegDest As rdoResultset

Dim intIndice As Integer

Dim iUIIndice As Integer

Private Sub PDatos()

If pDest(intIndice).INEdo <= 40 Then

optNacio.Value = True

IstrReg = "MRegEst"

Else

optInter.Value = True

IstrReg = "MRegCont"

End If

cmbEstdest.Text = pDest(intIndice).IDEdo

cmbMunic.Text = pDest(intIndice).IDDest

cmbMunic.Enabled = False

If pDest(intIndice).IDescDest <> "" Then

btDestPart.Visible = True

lbiDestPart.Visible = True

Else

btDestPart.Visible = False

lbiDestPart.Visible = False

End If



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
btDestPart = pDest(intIndice).IDescDest
lblDia(1) = intIndice + 1
'bfFecha = pFechaS
End Sub

Private Sub PGuarda()
With pDest(intIndice)
    .INDest = IRegDest.nsi_num_dest
    .IDDest = cmbMunic.Text
    .IDescDest = btDestPart
    .INEdo = pintEdo
    .IDEdo = cmbEstdest.Text
End With
IRegDest.Close
End Sub

Private Sub PLInicia()
Dim IRegDest As rdoResultset
intIndice = 0 'indice del arreglo que contendra los destinos
If MfAppend And pDest(intIndice).INDest = 0 Then
    Call pLlenaEst ' inicializa combo de estados
    'MRegEst.MoveFirst
    IstrDest = "cat_estados"
    IstrReg = "MRegEst"
    cmbMunic.Enabled = False
    pintEdo = 0
    pintDest = 0
    cmbMunic.Text = ""
    cmbEstdest.Text = ""
    btDestPart = ""
Else
    pintEdo = pDest(intIndice).INEdo
    pintDest = pDest(intIndice).INDest
    If pintEdo <= 40 Then
        Call pLlenaEst
        IstrDest = "cat_estados"
    End If

    Call PDatos
End If
End Sub

Private Function FVerifTextos() As Boolean
FVerifTextos = True
If cmbEstdest.Text = "" Or cmbMunic.Text = "" Or _
(cmbMunic.Text Like "RESTO DEL ESTADO" And btDestPart = "") Then
    MsgBox "Datos de destino faltantes!!!", vbExclamation, "Destinos"
    FVerifTextos = False
End If
End Function

Private Sub cmbestdest_Click()
Screen.MousePointer = 11
If IstrReg = "MRegEst" Then
    MRegEst.MoveFirst
    MRegEst.Move cmbEstdest.ListIndex
    If cmbEstdest.ListIndex = -1 Then
        MsgBox "Debes seleccionar una Opcion", vbExclamation, "Destinos"
        Screen.MousePointer = 1
    Else
        pintEdo = MRegEst.nsi_num_edo
    End If
    Isql = "SELECT cat_destinos.nsi_num_dest,cat_destinos.vch_nom_dest FROM cat_destinos " + _
        "where cat_destinos.nsi_num_edo = " & pintEdo
Else
    MRegCont.MoveFirst
    MRegCont.Move cmbEstdest.ListIndex
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
pintEdo = MRegCont!nsi_num_edo
Isql = "SELECT cat_destinos.nsi_num_dest,cat_destinos vch_nom_dest FROM cat_destinos " + _
"where cat_destinos.nsi_num_edo=" & pintEdo
End If
If cmbEstdest.ListIndex = -1 Then
    MsgBox "Debes seleccionar una Persona"
    Screen.MousePointer = 1
Else
    pDest(intIndice).INEdo = pintEdo
    Call pLlenaDest
    cmbMunic.Enabled = True
    Screen.MousePointer = vbNormal
End If
End Sub

Private Sub cmbmunic_Click()
    "falta
    Dim IRegDest As rdoResultset 'Recordset
    If InStr(cmbMunic.Text, "RESTO DEL ESTADO") <> 0 Then
        btDestPart.Visible = True
        lblDestPart.Visible = True
        btDestPart.SetFocus
    Else
        btDestPart.Visible = False
        lblDestPart.Visible = False
    End If
    btDestPart.Text = ""
    Isql = "select nsi_num_dest,ch_zona,ch_tipo from cat_destinos where nsi_num_edo=" & _
    pintEdo & " and vch_nom_dest like " & cmbMunic.Text & ""
    'Set IRegDest = pDbViat.OpenRecordset(Isql, dbOpenDynaset)
    Set IRegDest = bdrdoivat.OpenResultset(Name:=Isql, _
    Type:=rdOpenDynamic, LockType:=rdConcurValues)
    If cmbMunic.ListIndex = -1 Then
        MsgBox "Debes seleccionar una Opcion", vbExclamation, "Destinos"
    Else
        pintDest = IRegDest!nsi_num_dest

        With pDest(intIndice)
            .INDest = pintDest
            .ITipo = IRegDest!ch_tipo
            .IZona = IRegDest!ch_zona
        End With
        IRegDest.Close
    End If
End Sub

Private Sub cmdAnterior_Click()
    Screen.MousePointer = vbHourglass
    PbFlag = False
    If FVerifTextos Then
        Call PGuarda
        intIndice = intIndice - 1
        Call PDatos
        If intIndice = 0 Then
            cmdAnterior.Enabled = False
            cmdLlenar.Enabled = True
            cmdRepetir.Enabled = False
        Else
            cmdLlenar.Enabled = False
            cmdRepetir.Enabled = True
        End If
        cmdSig.Enabled = True
    End If
    Screen.MousePointer = vbNormal
End Sub
```



```
Private Sub cmdBorrar_Click()  
Screen.MousePointer = vbNormal  
intIndice = 0  
Select Case pNumDias  
    Case 1  
        ReDim pDest(1)  
    Case Else  
        ReDim pDest(pNumDias - 1)  
End Select  
If pNumDias = 1 Or pNumDias = 2 Then  
    cmdAnterior.Enabled = False  
    cmdSig.Enabled = False  
    cmdLlenar.Enabled = False  
Else  
    cmdAnterior.Enabled = False  
    cmdSig.Enabled = True  
    cmdLlenar.Enabled = True  
End If  
cmdAceptar.Enabled = True  
cmdRepetir.Enabled = False  
cmdBorrar.Enabled = True  
Call PDatos  
Screen.MousePointer = vbNormal  
  
End Sub  
  
Private Sub cmdLlenar_Click()  
Dim lIndex As Integer  
Screen.MousePointer = vbHourglass  
  
If FVerifTextos Then  
    Call PGuarda  
    For lIndex = 1 To IUtIndice - 1  
        pDest(lIndex) = pDest(0)  
    Next  
End If  
Screen.MousePointer = vbNormal  
  
End Sub  
  
Private Sub cmdRepetir_Click()  
pDest(intIndice) = pDest(intIndice - 1)  
Call PDatos  
End Sub  
  
Private Sub cmdSig_Click()  
PbFlag = True  
Screen.MousePointer = vbHourglass  
If FVerifTextos Then  
    Call PGuarda  
    intIndice = intIndice + 1  
    Call PDatos  
    If intIndice = IUtIndice - 1 Then  
        cmdSig.Enabled = False  
    End If  
    cmdRepetir.Enabled = True  
    cmdLlenar.Enabled = False  
    cmdAnterior.Enabled = True  
End If  
Screen.MousePointer = vbNormal  
End Sub  
  
Private Sub optInter_Click()  
Screen.MousePointer = vbHourglass  
If optInter.Value Then
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes. (Visual Basic)

```
Call pLlenaCont
IstrDest = "cat_estados" 'catcont
IstrReg = "MRegCont"
cmbMunic.Text = ""
cmbEstdest.Text = ""
cmbMunic.Enabled = False
'optCarro.Enabled = False
'optBus.Enabled = False
'optAvion.Value = True
End If
Screen.MousePointer = vbNormal
End Sub
Private Sub optNacio_Click()
Screen.MousePointer = vbHourglass
If optNacio.Value Then
Call pLlenaEst
IstrDest = "cat_estados"
IstrReg = "MRegEst"
cmbMunic.Text = ""
cmbEstdest.Text = ""
cmbMunic.Enabled = False
'optCarro.Enabled = True
'optBus.Enabled = True
End If
Screen.MousePointer = vbNormal
End Sub
Private Sub txtDestPart_KeyPress(KeyAscii As Integer)
If KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub

Private Sub pLlenaDest()
'MsgBox Isql
'Set MRegDest = pDbViat.OpenRecordset(Isql, dbOpenSnapshot, dbSQLPassThrough)
Set MRegDest = bdrdoviat.OpenResultSet(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
cmbMunic.Clear
MRegDest.MoveFirst
'cmbMunic.text = mRegDest!vch_nom_dest
Do Until MRegDest.EOF
cmbMunic.AddItem MRegDest!vch_nom_dest
MRegDest.MoveNext
Loop
MRegDest.Close
End Sub

Private Sub pLlenaCont()
cmbEstdest.Clear
MRegCont.MoveFirst
'cmbEstdest.text = MRegCont!vch_estado
Do Until MRegCont.EOF
cmbEstdest.AddItem MRegCont!vch_estado
MRegCont.MoveNext
Loop
End Sub
Private Sub pLlenaEst()
'inicializa el combo de catalogo de estados cmbEstdest
cmbEstdest.Clear
MRegEst.MoveFirst
'cmbEstdest.text = MRegEst!vch_estado
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Do Until MRegEst.EOF
    cmbEstdest.AddItem MRegEst!vch_estado
    MRegEst.MoveNext
Loop
End Sub

Private Sub cmdAceptar_Click()
    Dim lflag As Boolean
    Dim lindex As Integer
    lflag = False
    If FVerifTextos Then
        Call PGuarda
        'el siguiente codigo busca en el arreglo por si algun destino no
        'no fue llenado por el usuario, en ese caso le pregunta si desea
        'llenar los destinos faltantes con los datos del primer destino
        For lindex = 0 To lUltIndice - 1
            If pDest(lindex).INDest = 0 Then
                If lflag Then
                    pDest(lindex) = pDest(0)
                Else
                    If MsgBox("llenar datos faltantes " + _
                        "con primer destino?", vbYesNo) = vbNo Then
                        Exit Sub
                    Else
                        lflag = True
                        pDest(lindex) = pDest(0)
                    End If
                End If
            End If
        Next
        PintDias = pNumDias
        Unload Me
        frmSolicitud.Enabled = True
    End If
    Screen.MousePointer = vbNormal
End Sub

Private Sub Form_Load()
    On Error GoTo ManError
    'definicion de sentencia sql para estados y continentes
    lsql = "select nsi_num_edo,vch_estado from cat_estados where nsi_num_edo<=40"
    Set MRegEst = bdrdorh.OpenResultset(Name:=lsql, _
        Type:=rdOpenDynamic, LockType:=rdConcurValues)

    lsql = "select nsi_num_edo,vch_estado from cat_estados where nsi_num_edo>=90"
    Set MRegCont = bdrdorh.OpenResultset(Name:=lsql, _
        Type:=rdOpenDynamic, LockType:=rdConcurValues)
    MRegCont.MoveFirst
    frmIntern.Visible = False
    'Call ButtonNavigateMode
    'if pNumDias <> pintDias Then
        Select Case pNumDias
            Case 1
                lUltIndice = 1
                'ReDim Preserve pDest(1)
            Case Else
                lUltIndice = pNumDias - 1
                'ReDim Preserve pDest(pNumDias - 1)
            End Select
    If pNumDias <> PintDias Then
        ReDim Preserve pDest(lUltIndice)
    End If
    Call PLInicia

    If pNumDias = 1 Or pNumDias = 2 Then
        cmdAnterior.Enabled = False
    End If
End Sub
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
cmdSig.Enabled = False
cmdLlenar.Enabled = False
Else
cmdAnterior.Enabled = False
cmdSig.Enabled = True
cmdLlenar.Enabled = True
End If
cmdAceptar.Enabled = True
cmdRepetir.Enabled = False
cmdBorrar.Enabled = True
Screen.MousePointer = vbNormal
Centrar Me
Exit Sub
ManError:
Screen.MousePointer = 0
if rdoErrors.Count > 0 Then
Dim lerror As rdoError
For Each lerror In rdoErrors
MsgBox F_RdoError(lerror), vbCritical, "Error"
Call P_GuardaError(Me.Name, lerror.Number, F_RdoError(lerror))
Next
rdoErrors.Clear
Else
MsgBox F_ManError(Err.Number), vbCritical, "Error"
Call P_GuardaError(Me.Name, Err.Number, Err.Description)
End If
End Sub
```

---

\*frmfactura Forma de Captura de Facturas del Modulo  
de Comprobación de Viaticos  
\*Autor: Luis Hernández Pérez

---

\*Tablas:  
\*Base de Datos Tablas que Utiliza Lectura Escritura

*Viáticos	Hist_viat_emp	sql	sql
*Viáticos	Cat_destinos	sql	
*Viáticos	Cat_tarifas	sql	
*Viáticos	Cat_gastos	sql	
*Viáticos	Hist_comp_emp	sql	sql
*Viáticos	Hist_fact_emp	sql	sql
*Viáticos	Hist_otrosgas_emp	sql	sql
*Viáticos	Hist_kms	sql	sql

*Rec. Humanos	Datos_empleado	sql	
*Rec. Humanos	Cat_areas	sql	
*Rec. Humanos	Cat_estados	sql	
*Rec. Humanos	Cat_puestos	sql	

\*Forma(s) que Llama: frmcompro  
\*Llamada por: frmcompro  
\*Modificada por:

\*mmm

```
Private Sub Pnumera_fact()
' Numera las facturas de la comisión
Dim Lcont As Byte
Dim Lstr_Fact As String
Dim Ls_taxi As String
Dim Ls_fecha_Fact As String
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Set rsviat01 = bdrdoviat.OpenResultSet(Name =IsqI, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
'rsviat01.Clear
rsviat01.MoveFirst
Do Until rsviat01.EOF
    cmbMunic.AddItem rsviat01!\vch_nom_dest
    rsviat01.MoveNext
Loop
rsviat01.Close
End Sub
Private Sub botones()
    cmdAdd.Enabled = True
    cmdEdit.Enabled = False
    cmdSave.Enabled = True
    cmdCancel.Enabled = True
    cmdDelete.Enabled = False
    cmdFind.Enabled = False
    cmdClose.Enabled = True
End Sub

Public Sub fecha_mas_uno()
    Dim lanio As String
    Dim lmes As String
    Dim ldia As String
    Dim Lfec_paso As Date
    Dim Lfecha_masuno As String

    lanio = Format(Pfecha1, "yyyy")
    lmes = Mid(Pfecha1, 1, 2)
    ldia = Mid(Pfecha1, 4, 2)
    Lfecha_regreso = ldia & "/" & lmes & "/" & lanio
    Lfec_paso = Lfecha_regreso
    Lfec_paso = Lfec_paso + 1
    Lfecha_masuno = Mid(Lfec_paso, 4, 2) & "/" & Mid(Lfec_paso, 1, 2) & "/" & Format(Lfec_paso, "yyyy")
    Pfecha1 = Lfecha_masuno
    Putltime_dia = Lfecha_masuno
    'txtfecha.Text = Lfec_paso
End Sub

Public Sub fecha_menos_uno()
    Dim lanio As String
    Dim lmes As String
    Dim ldia As String
    Dim Lfec_paso As Date
    Dim Lfecha_menosuno As String
    lanio = Format(Pfecha1, "yyyy")
    lmes = Mid(Pfecha1, 1, 2)
    ldia = Mid(Pfecha1, 4, 2)
    Lfecha_regreso = ldia & "/" & lmes & "/" & lanio
    Lfec_paso = Lfecha_regreso
    Lfec_paso = Lfec_paso - 1
    Lfecha_menosuno = Mid(Lfec_paso, 4, 2) & "/" & Mid(Lfec_paso, 1, 2) & "/" & Format(Lfec_paso, "yyyy")
    Pfecha1 = Lfecha_menosuno
    'txtfecha.Text = Lfec_paso
End Sub

Private Sub Pllena_istfact()
    'llena el Ist del importe de facturas
    Dim Lsgasto As String
    'lhp Pfecha1 = Format(Pfecha1, "mm/dd/yyyy")
    lcad = "select * from hist_fact_emp where s_folio =" & Pfolio & " and " + _
    " d_fec_afecta=" & Pfecha1 & " order by hsi_cve_gasto"
    Set rsviat01 = bdrdoviat.OpenResultSet(Name=:lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
ReDim Arrimporate(rsviat01.RowCount)
ReDim Arrsecfact(rsviat01.RowCount)
Istfacturas.Clear
If rsviat01.EOF Then
    'Istfacturas.Enabled = False
Else
    Lcont = 0
    rsviat01.MoveFirst
    While Not rsviat01.EOF
        If Val(rsviat01!nsi_cve_gasto) = 8 Then
            lcad = "select * from hist_kms where s_folio =" & Pfolio
            Set rsviat04 = bdrdoviat.OpenResultSet(Name:=lcad, _
                Type:=rdOpenDynamic, LockType:=rdConcurValues)

            If rsviat04.EOF Then
                MsgBox "Debes cargar el Kilometraje Inicial/Final", vbExclamation, "Facturas"

            Else
                btkms1 = rsviat04!ni_kmsini
                btkms2 = rsviat04!ni_kmsfin
                btgas = rsviat04!nd_costo_rgas
            End If
            End If
            lcad = "select * from cat_gastos where nsi_cve_gasto =" & rsviat01!nsi_cve_gasto
            Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
                Type:=rdOpenDynamic, LockType:=rdConcurValues)
            Lsgasto = rsviat02!vch_desc_gasto

            Istfacturas.AddItem rsviat01!ch_fiscal & " / " & Format(rsviat01!m_monto_realfact, "#0.00") & " / " & rsviat01!nsi_porc_iva
            & " / " & Lsgasto & " / " & rsviat01!ch_taxi_aerop
            ' hoy hoy
            Arrimporate(Lcont) = rsviat01!m_monto_realfact
            Arrsecfact(Lcont) = rsviat01!ch_consecut
            rsviat01.MoveNext
            Lcont = Lcont + 1
        Wend
    End If
End Sub
Private Sub Pllena_dest()
    'llena el Ist del concepto de otros
    Dim lanio As String
    Dim lmes As String
    Dim ldia As String
    If Agrega Then
        lcad = "select * from hist_dest_emp where s_folio =" & Pfolio & "and " + _
            " d_pfecha=" & Pfecha1 & ""
        Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
            Type:=rdOpenDynamic, LockType:=rdConcurValues)
    Else
        lcad = "select * from hist_comp_emp where s_folio =" & Pfolio & "and " + _
            " d_fec_afecta=" & Pfecha1 & ""
        Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
            Type:=rdOpenDynamic, LockType:=rdConcurValues)
    End If
    If rsviat02.EOF Then
        'Istfacturas.Enabled = False
    Else
        If Agrega Then
            Ps_numdest = rsviat02!nsi_pnum_dest ' hoy
        Else
            Ps_numdest = rsviat02!nsi_rnum_dest ' hoy
        End If
    End If
    If rsviat01.EOF Then
    Else
        rsviat01.MoveLast
    End If
End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
End If

lcad = "select * from cat_destinos where nsi_num_dest =" & Ps_numdest
Set rsrvat03 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select vch_estado from cat_estados where nsi_num_edo =" & rsrvat03!nsi_num_edo
Set rsrh01 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

cmbEstdest.Text = rsrh01!vch_estado
cmbMunic.Text = rsrvat03!vch_nom_dest
If Agrega Then
    txtcuota_dia = rsrvat02!m_pcuota_dia
Else
    txtcuota_dia = rsrvat02!m_rcuota_dia
End If

txtfecha = Format(Pfecha1, "dd/mm/yyyy")
End If
lanio = Format(Pfecha1, "yyyy")
lmes = Mid(Pfecha1, 1, 2)
ldia = Mid(Pfecha1, 4, 2)
txtfecha.Text = ldia & "/" & lmes & "/" & lanio
End Sub

Private Sub valida_factura()
If Val(txtimporte) = 0 Then
    MsgBox "Debes cargar el monto de la Factura", vbExclamation, "Facturas"
    txtimporte.SetFocus
ElseIf cmbcpto.ListIndex = -1 Then
    MsgBox "Debes seleccionar un Concepto", vbExclamation, "Facturas"
    txtimporte.SetFocus
Else
    Ptflag_factura = True
End If
End Sub

Private Sub cmbcpto_Click()
If cmbcpto.ListIndex = -1 Then
    MsgBox "Debes seleccionar un Concepto", vbExclamation, "Facturas"
Else
    If cmbcpto.ListIndex = 9 Then
        fra_taxi_aerop.Visible = True
    Else
        fra_taxi_aerop.Visible = False
    End If
End If
End Sub

Private Sub cmbestdest_Click()
If cmbEstdest.ListIndex = -1 Then
    MsgBox "Debes seleccionar un Estado", vbExclamation, "Facturas"
    Screen.MousePointer = 1
Else
    Screen.MousePointer = 11
    'If IstrReg = "MRegEst" Then
    rsrhEst.MoveFirst
    rsrhEst.Move cmbEstdest.ListIndex
    pintEdo = rsrhEst!nsi_num_edo

    Isql = "SELECT * FROM cat_destinos * _
    "where cat_destinos.nsi_num_edo =" & pintEdo
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
cmbMunic.Clear
Call pLenaDest ` Municipios
cmbMunic.Enabled = True
Screen.MousePointer = vbNormal
End If
End Sub

Private Sub cmbmunic_Click()
Dim lRegDest As rdoResultset
Dim Ln_cuota As Single
Dim Ls_tipo As String
If cmbMunic.ListIndex = -1 Then
MsgBox "Debes seleccionar una Ciudad", vbExclamation, "Facturas"
Else
If InStr(cmbMunic.Text, "RESTO DEL ESTADO") <> 0 Then
txtDestPart.Visible = True
lbldestino.Visible = True
txtDestPart.SetFocus
Else
txtDestPart.Visible = False
lbldestino.Visible = False
End If
End If
txtDestPart.Text = ""

Isql = "select * from cat_destinos where nsi_num_edo=" & _
pintEdo & " and vch_norm_dest like " & cmbMunic.Text & ""
Set rsviat01 = bdrdoviat.OpenResultset(Name:=Isql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

If cmbEstdest.ListIndex = -1 Then
MsgBox "Debes seleccionar una Opcion", vbExclamation, "Comprobación"
Screen.MousePointer = 1
Else

Ls_tipo = rsviat01!ch_tipo
Ps_numdest = rsviat01!nsi_num_dest

If pNumDias > 1 Then
Select Case Ls_tipo
Case "N"
IstrCampZon = "m_may24" + "_" + "z" + rsviat01!ch_zona
Case "I"
IstrCampZon = "m_intl" + "_" + "z" + rsviat01!ch_zona
End Select
If nivelsup > 0 Then
nivelcom = nivelsup
Else
nivelcom = nivelemp
End If

' Fecha a partir de la que se maneja nueva tarifa en ( cat_tarifas_nva )
' poner nuevas tarifas en esta tabla, cuando no se tengan
' problemas igualar las dos tablas (cat_tarifas) comp/nva/edic/>1 dia

If Pfechas > Pfecha_nvatar Then
Isql = "select " + IstrCampZon + _
" from cat_tarifas_nva where nsi_nivel<=" & _
nivelcom
Else
Isql = "select " + IstrCampZon + _
" from cat_tarifas where nsi_nivel<=" & _
nivelcom
End If
End If
```



# MIGRACION DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
'hoysql = "select " + IstrCampZon + _  
'hoysql " from cat_tarifas where nsi_nivel <=" & _  
'hoys nivelcom  
  
Set rsviat01 = bdrdoviat.OpenResultset(Name:=Isq, _  
Type:=rdOpenDynamic, LockType:=rdConcurValues)  
  
rsviat01.MoveLast  
Ln_cuota = rsviat01(0)  
  
Else 'Comision menores a 24 hrs  
Select Case Ls_tipo  
Case "N"  
IstrCampZon = "m_men24" + "_"z" + rsviat01!ch_zona  
Case "I"  
IstrCampZon = "m_int" + "_"z" + rsviat01!ch_zona  
End Select  
If nivelsup > 0 Then  
nivelcom = nivelsup  
Else  
nivelcom = nivelemp  
End If  
  
' Fecha a partir de la que se maneja nueva tarifa en ( cat_tarifas_nva )  
' poner nuevas tarifas en esta tabla  
' cuando no se tengan problemas igualar las dos tablas (cat_tarifas) << a dia/nva/dic.  
  
If Pfecha > Pfecha_nvatar Then  
Isq = "select " + IstrCampZon + _  
" from cat_tarifas_nva where nsi_nivel<=" & _  
nivelcom  
Else  
Isq = "select " + IstrCampZon + _  
" from cat_tarifas where nsi_nivel<=" & _  
nivelcom  
End If  
  
' Isq = "select " + IstrCampZon + _  
' " from cat_tarifas where nsi_nivel <=" & _  
' nivelcom  
  
Set rsviat01 = bdrdoviat.OpenResultset(Name:=Isq, _  
Type:=rdOpenDynamic, LockType:=rdConcurValues)  
  
rsviat01.MoveLast  
Ln_cuota = rsviat01(0)  
  
End If  
frataturas.Enabled = True  
btcuota_dia = Format(Ln_cuota, "###,##0.00") 'll  
btcuota_dia = Ln_cuota  
Pn_cuota = Ln_cuota  
Pnum_des_cambio = Ps_numdest  
Pmunic_cambio = cmbMunic.Text  
btimporte.SetFocus  
End If  
End If  
End Sub  
  
Private Sub cmbmunic_LostFocus()  
btcuota_dia = pfactura(intIndice).Cuota  
End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Private Sub cmdAdd_Click()
    txtimporte = ""
    txtiva = ""
    txtfactura = ""
    txtconceptos = "Conceptos"
    txtimporte.SetFocus
End Sub

Private Sub cmdant_Click()
    If Val(txtdia) = 1 Then
        cmdant.Enabled = False
        MsgBox "Primer Dia", vbExclamation, "Facturas"
    Else
        cmdsig.Enabled = True
        Call fecha_menos_uno
        txtdia = Val(txtdia) - 1
        Pcontdias = Val(txtdia)
        txtfecha = Format(Pfecha1, "dd") & "/" & Format(Pfecha1, "mm") & "/" & Format(Pfecha1, "yyyy")
        lstfacturas.Clear
        Call Pllena_istfact
        Call Pllena_dest
        txtimporte = ""
        txtiva = ""
        txtfactura = ""
        txtconceptos = "Conceptos"
        txtimporte.SetFocus
        cmdsig.Enabled = True
        Pflg_ant = True
    End If
End Sub

Private Sub cmdCancel_Click()
    Unload Me
    frmcompro.Enabled = True
End Sub

Private Sub cmdClose_Click()
    If Pflg_kms Then 'Actualiza el archivo de Kilometraje hist_kms
        If btkms1 = "" Then
            MsgBox "Introducir Kilometraje Inicial", vbExclamation, "Facturas"
            btkms1.SetFocus
        ElseIf btkms2 = "" Then
            MsgBox "Introducir Kilometraje Final", vbExclamation, "Facturas"
            btkms2.SetFocus
        ElseIf btgas = "" Then
            MsgBox "Introducir Costo de Cobustible", vbExclamation, "Facturas"
            btgas.SetFocus
        Else
            'hlp bdrdoviat.Execute "delete from hist_comp_emp where s_folio=" & Pfolio
            bdrdoviat.Execute "delete from hist_comp_emp where " +
            "s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & ""
            If btcuota_dia.Text > 0 Then ' Si realizo una captura
                lstreq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & txtcuota_dia.Text & ")"
                lstreq = "(s_folio,d_fec_afecta,nsi_num_dest,vch_rdestino,m_rcuota_dia)"
                bdrdoviat.Execute "insert into hist_comp_emp " & lstreq & " values " & lstreq
            End If
            bdrdoviat.Execute "delete from hist_kms where s_folio=" & Pfolio
            lstreq = "(" & Pfolio & "," & Val(btkms1) & "," & Val(btkms2) & "," & (Val(btkms2) - Val(btkms1)) / 5 * Val(btgas) & "," &
            Val(btgas) & ")"
            lstreq = "(s_folio,ni_kmsini,ni_kmsfin,nd_consumo,nd_costo_rgas)"
            bdrdoviat.Execute "insert into hist_kms " & lstreq & " values " & lstreq
            Pflg_kms = False
        End If
    End If
End Sub
```





## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
PintDias = pNumDias
Unload Me
frmcompro.Enabled = True
Screen.MousePointer = 1
End If
Else ' no hay combustible

' BORRAR HOTEL EN EL ULTIMO DIA

If pNumDias = 1 And Pflg_kms = False Then 'actualiza el Archivos de Comprobaciones hist_comp_emp
bdrdoviat.Execute "delete from hist_fact_emp where " + _
"nsi_cve_gasto=0 and s_folio=" & Pfolio

bdrdoviat.Execute "delete from hist_fact_emp where " + _
"nsi_cve_gasto=1 and s_folio=" & Pfolio

'hp bdrdoviat.Execute "delete from hist_comp_emp where s_folio=" & Pfolio
bdrdoviat.Execute "delete from hist_comp_emp where " + _
"s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & ""

If txtcuota_dia.Text > 0 Then ' Si realizo una captura
Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & txtcuota_dia.Text & ")"
Istrf = "(s_folio,d_fec_afecta,nsi_num_dest,vch_rdestino,m_rcuota_dia)"
bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq
End If

PintDias = pNumDias
Unload Me
frmcompro.Enabled = True
Screen.MousePointer = 1

Else
' Comision mayor a 1 dia se actualiza en el combo de Siguiete dia
End If

If cmbcpto.ListIndex = -1 Then
PintDias = pNumDias
Unload Me
frmcompro.Enabled = True
Screen.MousePointer = 1
Else
PintDias = pNumDias
Unload Me
frmcompro.Enabled = True
Screen.MousePointer = 1
End If
End If
**** 03/feb/2000
' cambio de destino la ultima fecha de la comision
If pNumDias > 1 And Pflg_cam_edo = True And pNumDias = Pcontdias Then

Istrq = "(" & Pnum_des_cambio & "," & Pmunic_cambio & "," & Pn_cuota & ")"
Istrf = "(nsi_num_dest,vch_rdestino,m_rcuota_dia)"

bdrdoviat.Execute "update hist_comp_emp set" & Istrf & "=" & Istrq + _
" where d_fec_afecta=" & Pultimo_dia & " and s_folio=" & Pfolio
' MsgBox "termine"
End If
Pcontdias = 0
Pflg_cam_edo = False

End Sub

Private Sub cmdDelete_Click()
If lstfacturas.ListIndex <> -1 Then
bdrdoviat.Execute "delete from hist_fact_emp where " +
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
"s_folio=" & Pfolio & " and m_monto_realfact=" & Pn_importe & "and " + _  
"d_fec_afecta=" & Pfecha1 & " and ch_consecut=" & Pn_numsectact & ""  
  
bdrdoviat.Execute "update hist_fact_emp set ch_consecut=null where s_folio=" & Pfolio  
Call Pnumera_fact  
Call Pllena_istfact  
Else  
MsgBox "Debes Seleccionar un Importe/Iva/Concepto del recuadro derecho", vbExclamation, "Facturas"  
End If  
End Sub  
  
Private Sub cmdregistra_Click()  
Dim Ls_cpto As String  
Dim Pfecha1 As String  
  
Call valida_factura  
  
If Ptflag_factura Then  
If txtiva = "" Then  
txtiva.Text = 0  
End If  
If txtfactura = "" Then  
txtfactura = "Sin Factura"  
End If  
lstfacturas.AddItem cmbcpto.Text & "/" & txtiva & "/" & btimporte  
Pfecha1 = Format(Pfechas, "mm/dd/yyyy")  
Ls_cpto = Mid(cmbcpto.Text, 1, 2)  
  
lstrq = "(" & Pfolio & "," & Pfecha1 & "," & Val(Ls_cpto) & "," & btimporte & "," & btimporte & "," & (Val(btimporte) * Val(txtiva) /  
100) & "," & txtfactura & "," & txtiva & ")"  
lstrf =  
"(s_folio,d_fec_afecta,nsi_cve_gasto,m_monto_realfact,m_monto_autofact,m_iva_monto,vch_num_factura,nsi_porcentaje)"  
bdrdoviat.Execute "insert into hist_fact_emp " & lstrf & " values " & lstrq  
Call Pllena_istfact  
Else  
btimporte.SetFocus  
End If  
End Sub  
  
Private Sub cmdSave_Click()  
Dim Ls_cpto As String  
Dim Ls_Fiscal As String  
Dim Ls_Taxi_aerop As String  
Dim Lstr_Fact As String  
Call valida_factura  
If Ptflag_factura Then  
If txtiva = "" Then  
txtiva.Text = 0  
End If  
  
If txtfactura = "" Then  
txtfactura = "Sin Factura"  
End If  
Ls_cpto = Mid(cmbcpto.Text, 1, 2)  
  
If Val(Ls_cpto) = 8 Then  
txtkms1.Enabled = True  
txtkms2.Enabled = True  
btgas.Enabled = True  
Ptflag_kms = True  
End If  
  
If optfiscal_si = True Then  
Ls_Fiscal = "S"  
Else  
Ls_Fiscal = "N"
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
End If

If opt_taxi_asi And Val(Ls_cpto) = 9 Then
    Ls_Taxi_aerop = "S"
Else
    Ls_Taxi_aerop = "N"
End If

If (pNumDias = 1 And Val(Ls_cpto) < 2) Or (pNumDias - btdia = 0 And Val(Ls_cpto) < 2) Then
    ' No graba ni el primer, ni el ultimo dia de hotel
    MsgBox "No puede Facturar Hotel en Comisiones menores a 24 HRS o El Ultimo Día", vbExclamation, "Facturas"
    btimporte.SetFocus
Else
    'valida el ultimo numero consecutivo de facturas para dar
    'el siguiente

    lcad = "select * from hist_fact_emp where " + _
        "s_folio=" & Pfolio

    Set rsrvat01 = bdrdoviat.OpenResultset(Name:=lcad, _
        Type:=rdOpenDynamic, LockType:=rdConcurValues)

    If rsrvat01.EOF Then
        Pnum_cons_fact = 1
    Else
        Pnum_cons_fact = 1 + rsrvat01.RowCount
    End If

    If Pnum_cons_fact < 10 Then
        Lstr_Fact = "00" + Trim(Str(Pnum_cons_fact))
    ElseIf Pnum_cons_fact < 100 Then
        Lstr_Fact = "0" + Trim(Str(Pnum_cons_fact))
    Else
        Lstr_Fact = Trim(Str(Pnum_cons_fact))
    End If

    Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Val(Ls_cpto) & "," & btimporte & "," & btimporte & "," & (btimporte * btxiva / 100) & _
        "," & btxiva & "," & btxiva & "," & Ls_Fiscal & "," & Ls_Taxi_aerop & "," & Lstr_Fact & ")"
    Istrf =
        "(s_folio,d_fec_afecta,nsi_cve_gasto,m_monto_realfact,m_monto_autofact,m_iva_monto,vch_num_factura,nsi_porc_iva,ch_fiscal,
        ch_taxi_aerop,ch_consecut)"
    bdrdoviat.Execute "insert into hist_fact_emp " & Istrf & " values " & Istrq

    Call Pllena_Istfact
    fra_taxi_aerop.Visible = False
End If
optfiscal_si.Value = True
Else
    btimporte.SetFocus
End If
End Sub

Private Sub cmdSig_Click()
    If Val(btdia) = pNumDias Then 'ultimo dia
        cmdsig.Enabled = False
        MsgBox "Ultimo Dia", vbExclamation, "Facturas"
    Else
        If Agrega And Ptfilag_ant = False Then 'Nuevo Registro
            Istfacturas.Clear
            If (pNumDias - Val(btdia)) >= 2 And Agrega Then 'Modo nuevo
                'actualiza el Archivos de Comp. hist_comp_emp dia a dia
                bdrdoviat.Execute "delete from hist_comp_emp where " + _
                    "s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & "
                Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & Val(btcuota_dia) & ")"
                Istrf = "(s_folio,d_fec_afecta,nsi_num_dest,vch_destino,m_rcuota_dia)"
                bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq
            End If
        End If
    End If
End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq

cmdant.Enabled = True

Call fecha_mas_uno

btdia = Val(btdia) + 1
Pcontdias = Val(btdia)
Call Pllena_istfact
Call Pllena_dest
'btfecha = Format(Pfecha1, "dd") & "/" & Format(Pfecha1, "mm") & "/" & Format(Pfecha1, "yyyy")
Elseif (pNumDias - Val(btdia)) >= 1 And Agrega Then ' Dia Penultimo y Ultimo
'Penultimo dia
bdrdoviat.Execute "delete from hist_comp_emp where " + _
"s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & ""

Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & txtcuota_dia & ")"
Istrf = "(s_folio,d_fec_afecta,nsi_rnum_dest,vch_rdestino,m_rcuota_dia)"
bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq

'Ultimo dia
Call fecha_mas_uno
' Pfecha1 = Format(Pfecha1, "mm/dd/yyyy")
bdrdoviat.Execute "delete from hist_comp_emp where " + _
"s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & ""

Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & txtcuota_dia & ")"
Istrf = "(s_folio,d_fec_afecta,nsi_rnum_dest,vch_rdestino,m_rcuota_dia)"
bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq
btdia = Val(btdia) + 1
Pcontdias = Val(btdia)
'btfecha = Format(Pfecha1, "dd") & "/" & Format(Pfecha1, "mm") & "/" & Format(Pfecha1, "yyyy")
cmdant.Enabled = True
Call Pllena_istfact
Call Pllena_dest
Else 'Ultimo dia
cmdsig.Enabled = False
cmdant.Enabled = True
MsgBox "Ultimo Dia", vbExclamation, "Facturas"
End If
btimporte = ""
btiva = ""
btfactura = ""
btconceptos = "Conceptos"
btimporte.SetFocus
Elseif Agrega And Ptfag_ant = True Then
btdia = Val(btdia) + 1
Pcontdias = Val(btdia)
Call fecha_mas_uno
Istfacturas.Clear
Call Pllena_istfact
Call Pllena_dest

Ptfag_ant = False 'bandera de regresar al anterior
cmdant.Enabled = True
Else ' Modo Edicion
""
If (pNumDias - Val(btdia)) = 1 Then ' Dia Penultimo y Ultimo
'Penultimo dia
bdrdoviat.Execute "delete from hist_comp_emp where " + _
"s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & ""

Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & txtcuota_dia & ")"
Istrf = "(s_folio,d_fec_afecta,nsi_rnum_dest,vch_rdestino,m_rcuota_dia)"
bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
'Ultimo dia
Call fecha_mas_uno
Istfacturas.Clear
Call Pllena_istfact
Call Pllena_dest
cmdant.Enabled = True

bdrdoviat.Execute "delete from hist_comp_emp where " +
"s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & ""

Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & btcuota_dia & ")"
Istrf = "(s_folio,d_fec_afecta,nsi_num_dest,vch_rdestino,m_rcuota_dia)"
bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq
btdia = Val(btdia) + 1
Pcontdias = Val(btdia)
cmdant.Enabled = True
cmdsig.Enabled = True
Else
btdia = Val(btdia) + 1
Pcontdias = Val(btdia)
bdrdoviat.Execute "delete from hist_comp_emp where " +
"s_folio=" & Pfolio & " and d_fec_afecta=" & Pfecha1 & ""

Istrq = "(" & Pfolio & "," & Pfecha1 & "," & Ps_numdest & "," & cmbEstdest.Text & "," & btcuota_dia & ")"
Istrf = "(s_folio,d_fec_afecta,nsi_num_dest,vch_rdestino,m_rcuota_dia)"
bdrdoviat.Execute "insert into hist_comp_emp " & Istrf & " values " & Istrq

Call fecha_mas_uno
Istfacturas.Clear
Call Pllena_istfact
Call Pllena_dest
cmdant.Enabled = True
End If
End If
End If
End Sub

Private Sub Form_Load()
'On Error GoTo ManError

Ptflag_kms = False
Ptflag_ant = False
Pfecha1 = Format(Pfechas, "mm/dd/yyyy")
btdia.Text = 1
fra_taxi_aerop.Visible = False
Call Pnumera_fact
If pNumDias = 1 Then
cmdant.Visible = False
cmdsig.Visible = False
End If
If Agrega And Pn_acceso = 0 Then ' Captura Nueva
btfcha = Format(Pfechas, "dd/mm/yyyy")
btkms1.Enabled = False
btkms2.Enabled = False
btgas.Enabled = False

Istfacturas.Clear
Ptflag_factura = False
Istfacturas.Enabled = False
Istdest.Enabled = True
Istacmdest.Visible = True
Call botones
' Definicion de sentencia sql para Estados y continentes
Isql = "select nsi_num_edo, vch_estado from cat_estados " +
"where nsi_num_edo<=40"
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Set rsrhEst = bdrdorh.OpenResultSet(Name:=!sql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

Call pLlenaEst ' inicializa combo de Estados
IstrDest = "cat_estados"
IstrReg = "MRegEst"
cmbMunic.Enabled = False
pintEdo = 0
pintDest = 0
cmbMunic.Text = ""
cmbEstdest.Text = ""
txtDestPart = ""

' Definicion de sentencia sql para llenado de gastos
!sql = "select nsi_cve_gasto, vch_desc_gasto from cat_gastos "
Set rsviatgastos = bdrdoviat.OpenResultSet(Name:=!sql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

rsviatgastos.MoveFirst

Do Until rsviatgastos.EOF
    cmbcpto.AddItem rsviatgastos!nsi_cve_gasto & " " & rsviatgastos!vch_desc_gasto
    rsviatgastos.MoveNext
Loop
Screen.MousePointer = 1
Pn_acceso = 1

Elseif Pn_acceso = 1 Or Agrega = False Then 'Si captura y regresa o es Edicion
Screen.MousePointer = 1
!stfacturas.Clear
Call PLlena_!stfact
Call PLlena_dest
' Definicion de sentencia sql para llenado de gastos
!sql = "select nsi_cve_gasto, vch_desc_gasto from cat_gastos "
Set rsviatgastos = bdrdoviat.OpenResultSet(Name:=!sql, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

rsviatgastos.MoveFirst

Do Until rsviatgastos.EOF
    cmbcpto.AddItem rsviatgastos!nsi_cve_gasto & " " & rsviatgastos!vch_desc_gasto
    rsviatgastos.MoveNext
Loop
Screen.MousePointer = 1
Else

End If
Centrar Me

Exit Sub
ManError:
Screen.MousePointer = 0
If rdoErrors.Count > 0 Then
    Dim lerror As rdoError
    For Each lerror In rdoErrors
        MsgBox F_RdoError(lerror), vbCritical, "Error"
        Call P_GuardaError(Me.Name, lerror.Number, F_RdoError(lerror))
    Next
    rdoErrors.Clear
Else
    MsgBox F_ManError(Err.Number), vbCritical, "Error"
    Call P_GuardaError(Me.Name, Err.Number, Err.Description)
End If
End Sub
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Private Sub Istfacturas_Click()
    Dim Ls_valor As String
    If Istfacturas.ListIndex <> -1 Then
        If rsviat01.RowCount > 0 Then
            Pn_importe = Arrimporte(Istfacturas.ListIndex)
            Pn_numsecfact = Arrsecfact(Istfacturas.ListIndex)
            cmdDelete.Enabled = True
            Ls_valor = Arrimporte(Istfacturas.ListIndex)

            If Val(Ls_valor) - Arrimporte(Istfacturas.ListIndex) < 0 Then
                Pn_importe = Pn_importe / 100
            End If
        End If
    End If
End Sub

Private Sub optsi_Click()
    If optsi = True Then
        ' Definicion de sentencia sql para Estados y continentes
        Isql = "select nsi_num_edo, vch_estado from cat_estados " + _
            "where nsi_num_edo <= 40"
        Set rsrEst = bdrdorh.OpenResultset(Name:=Isql, _
            Type:=rdOpenDynamic, LockType:=rdConcurValues)

        Call pLlenaEst ' inicializa combo de Estados
        IstrDest = "cat_estados"
        IstrReg = "MRegEst"
        cmbMunic.Enabled = False
        pintEdo = 0
        pintDest = 0
        cmbMunic.Text = ""
        cmbEstdest.Text = ""
        btDestPart = ""
        Pflag_cam_edo = True
        optno.Value = True
    End If
End Sub

Private Sub SSCommand3_Click()
End Sub

Private Sub SSCommand1_Click()
End Sub

Private Sub btgas_KeyPress(KeyAscii As Integer)
    If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
        Else
            MsgBox ("Introducir Numeros"), vbExclamation, "Comprobación"
            KeyAscii = 0
        End If
    End Sub

Private Sub btimporte_GotFocus()
    If btimporte.Text = "" Then
        btimporte.Text = ""
    Else
        btimporte = Format(btimporte, "###.##0.00")
    End If
End Sub

Private Sub btimporte_KeyPress(KeyAscii As Integer)
    If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Else
    MsgBox ("Introducir Numeros"), vbExclamation, "Comprobación"
    KeyAscii = 0
End If
End Sub

Private Sub btimporte_LostFocus()
    If Val(btimporte) > 10000 Then
        MsgBox "El Importe es muy Alto", vbExclamation, "Comprobación"
        btiva = ""
        btiva.SetFocus
    End If
End Sub

Private Sub btiva_KeyPress(KeyAscii As Integer)
    If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
        Else
            MsgBox ("Introducir Numeros"), vbExclamation, "Comprobación"
            KeyAscii = 0
        End If
    End Sub

Private Sub btiva_LostFocus()
    If Val(btiva) > 15 Then
        MsgBox "El Porcentaje de Iva es muy Alto", vbExclamation, "Comprobación"
        btiva = ""
        btiva.SetFocus
    End If
End Sub

Private Sub btkms1_Change()
    Ptflag_kms = True
End Sub

Private Sub btkms1_KeyPress(KeyAscii As Integer)
    If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
        Else
            MsgBox ("Introducir Numeros"), vbExclamation, "Comprobación"
            KeyAscii = 0
        End If
    End Sub

Private Sub btkms2_Change()
    Ptflag_kms = True
End Sub

Private Sub btkms2_KeyPress(KeyAscii As Integer)
    If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
        Else
            MsgBox ("Introducir Numeros"), vbExclamation, "Comprobación"
            KeyAscii = 0
        End If
    End Sub

Private Sub btkms2_LostFocus()
    If Val(btkms1) > Val(btkms2) Then
        MsgBox "No puede ser mayor el kilometraje final", vbExclamation, "Comprobación"
        btkms2.SetFocus
    End If
End Sub
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

#### FRM GASTOS

'frmgestos.frm Forma de captura otros gastos e itinerarios  
del modulo de Viaticos

'Autor: Luis Hernández Pérez

'Tablas:

'Base de Datos Tablas que Utiliza Lectura Escritura

'Viaticos Hist\_viat\_emp sql sql

'Forma que Llama: frmsolicitud.frm

'Llamada por: frmsolicitud.frm

'Modificada por:

Public Function fVerifNumero(bt As TextBox, msg As String) As Boolean  
'verifica que el numero de empleado sea un valor numerico

If bt.Text = "" Then 'si no ha introducido ningun dato  
MsgBox ("Debe Introducir " + msg), vbExclamation, "Gastos"  
fVerifNumero = False  
bt.SetFocus

Elseif Not (IsNumeric(bt.Text)) Then 'si ha introducido un valor numerico  
MsgBox ("Debe Introducir un Número"), vbExclamation, "Gastos"  
fVerifNumero = False  
bt.Text = ""  
bt.SetFocus

Else  
fVerifNumero = True

End If  
End Function

Private Sub chkAvion\_Click()

If chkAvion = 1 Then  
btItinAvion.Enabled = True  
btFechSal.Enabled = True  
btHoraSal.Enabled = True

btFechSal = pFechas  
'LHP btHoraSal = pHoraS  
btHoraSal = pHoraAvion

Else  
btItinAvion.Enabled = False  
btItinAvion = ""  
btFechSal.Enabled = False  
btFechSal = ""  
btHoraSal.Enabled = False  
btHoraSal = ""

End If  
End Sub

Private Sub chkBus\_Click()

If chkBus = 1 Then  
btItinBus.Enabled = True  
btItinBus.SetFocus  
btGastoBus.Enabled = True

Else  
btItinBus.Enabled = False  
btGastoBus.Enabled = False  
btItinBus = ""



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
    txtGastoBus = Format(0, "###0.00")
End If
End Sub

Private Sub chkCarro_Click()
If chkCarro = 1 Then
    txtInCarro.Enabled = True
    txtInCarro.SetFocus
    txtKm.Enabled = True
    txtcosto_gas.Enabled = True
    txtPeaje.Enabled = True
    chkRecMat.Enabled = True
Else
    txtInCarro.Enabled = False
    txtKm.Enabled = False
    txtcosto_gas.Enabled = False
    txtPeaje.Enabled = False
    chkRecMat.Enabled = False
    txtInCarro = ""
    txtKm = Format(0, "###0.0")
    txtcosto_gas = Format(0, "###0.00")
    txtPeaje = Format(0, "###0.00")
    chkRecMat = 0
End If
End Sub

Private Sub cmdAceptar_Click()
Dim istrNum As String
Screen.MousePointer = 11
istrNum = "un número"
Istrdatos = "no ha ingresado todos los datos requeridos"
If chkAvion = 1 And (txtInAvion = "" Or txtFechSal = "" Or _
    txtHoraSal = "") Then
    MsgBox Istrdatos, vbExclamation, "Gastos"
    Screen.MousePointer = vbNormal
    Exit Sub
End If
If chkBus = 1 Then
    If txtInBus = "" Then
        MsgBox Istrdatos, vbExclamation, "Gastos"
        Screen.MousePointer = vbNormal
        Exit Sub
    ElseIf Not fVerifNumero(txtGastoBus, istrNum) Then
        Screen.MousePointer = vbNormal
        Exit Sub
    End If
End If
If chkCarro = 1 Then
    If txtInCarro = "" Then
        MsgBox Istrdatos, vbExclamation, "Gastos"
        Screen.MousePointer = vbNormal
        Exit Sub
    ElseIf Not fVerifNumero(txtKm, istrNum) Then
        Screen.MousePointer = vbNormal
        Exit Sub
    ElseIf Not fVerifNumero(txtPeaje, istrNum) Then
        Screen.MousePointer = vbNormal
        Exit Sub
    End If
End If
If Not fVerifNumero(txtFoto, istrNum) Then
    Screen.MousePointer = vbNormal
    Exit Sub
ElseIf Not fVerifNumero(txtLegales, istrNum) Then
    Screen.MousePointer = vbNormal
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Exit Sub
ElseIf Not (VerifNumero(txtOtros, lstrNum) Then
Screen.MousePointer = vbNormal
Exit Sub
Else
pcosto_gas = txtcosto_gas
pKm = txtKm
pLtrGas = txtcosto_gas
pBus = txtGastoBus
pPeaje = txtPeaje
pFoto = txtFoto
pLegales = txtLegales
pOtros = txtOtros
pItinAvion = txtItinAvion
pItinBus = txtItinBus
pItinCarro = txtItinCarro
pFechaAvion = txtFechSal
pHoraAvion = txtHoraSal
pObs = txtObs
Psciaerea = txtciaerea

If chkRecMat = 1 Then
pRecMat = True
Else
pRecMat = False
End If
End If
Unload Me
Screen.MousePointer = vbNormal
End Sub

Private Sub Form_Load()
On Error GoTo ManError
Dim lintCero As Integer
lintCero = 0
frmSolicitud.Enabled = False

If pItinAvion = "" Then
chkAvion = 0
txtItinAvion.Enabled = False
txtFechSal.Enabled = False
txtHoraSal.Enabled = False
Else
chkAvion = 1
txtItinAvion.Enabled = True
txtFechSal.Enabled = True
txtHoraSal.Enabled = True

End If
If pItinBus = "" Then
chkBus = 0
txtItinBus.Enabled = False
txtGastoBus.Enabled = False
Else
chkBus = 1
txtItinBus.Enabled = True
txtGastoBus.Enabled = True
End If
If pItinCarro = "" Then
chkCarro = 0
txtItinCarro.Enabled = False
txtKm.Enabled = False
txtcosto_gas.Enabled = False
txtPeaje.Enabled = False
chkRecMat.Enabled = False
Else
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
btItinCarro.Enabled = True
chkCarro = 1
btKm.Enabled = True
btcosto_gas.Enabled = True
btPeaje.Enabled = True
chkRecMat.Enabled = True
End If

btItinAvion = pItinAvion
btFechSal = pFechaAvion
'btHoraSal = pHoraAvion
btItinBus = pItinBus
btItinCarro = pItinCarro
btGastoBus = Format(pBus, "###0.00")
If pRecMat Then 'llevara auto de INSTITUCION
    chkRecMat = 1
Else
    chkRecMat = 0
End If

btcosto_gas = Format(pLtrGas, "###0.00")
btKm = Format(pKm, "###0.0")
btPeaje = Format(pPeaje, "###0.00")
btFoto = Format(pFoto, "###0.00")
btLegales = Format(pLegales, "###0.00")
btOtros = Format(pOtros, "###0.00")
btObs = pObs
Screen.MousePointer = vbNormal

Centrar Me

Exit Sub
ManError:
    Screen.MousePointer = 0
    If rdoErrors.Count > 0 Then
        Dim lerror As rdoError
        For Each lerror In rdoErrors
            MsgBox F_RdoError(lerror), vbCritical, "Error"
            Call P_GuardaError(Me.Name, lerror.Number, F_RdoError(lerror))
        Next
        rdoErrors.Clear
    Else
        MsgBox F_ManError(Err.Number), vbCritical, "Error"
        Call P_GuardaError(Me.Name, Err.Number, Err.Description)
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    frmSolicitud.Enabled = True
End Sub

Private Sub btciaerea_KeyPress(KeyAscii As Integer)
    Dim car As String * 1
    car = UCase(Chr(KeyAscii))
    KeyAscii = Asc(car)
    If KeyAscii = 13 Or KeyAscii = 34 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
        KeyAscii = 0
        Beep
    End If
End Sub

Private Sub btFechSal_KeyPress(KeyAscii As Integer)
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
If KeyAscii = 13 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub

Private Sub btnHoraSal_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub

Private Sub btnItinAvion_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub

Private Sub btnItinBus_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub

Private Sub btnItinCarro_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub

Private Sub txtObs_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Or KeyAscii = 39 Or KeyAscii = 96 Or KeyAscii = 124 Then
    KeyAscii = 0
End If
'39=APOSTROFO, 96=APOSTROFO INVERTIDO, 124=BARRA VERTICAL
Dim car As String * 1
car = UCase(Chr(KeyAscii))
KeyAscii = Asc(car)
End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

#### FRM RREPORTES

```
'frmreportes      Forma de Reportes del Modulo  
                  de Sol/Comp de Viaticos  
'Autor:          Luis Hernández Pérez
```

#### 'Tablas

```
'Base de Datos  Tablas que Utiliza  Lectura Escritura
```

```
'Viáticos      Hist_viat_emp    sql  
'Viáticos      Cat_destinos    sql  
'Viáticos      Cat_objetivos    sql  
'Viáticos      Cat_tarifas     sql  
'Viáticos      Cat_gastos      sql  
'Viáticos      Hist_boi_avion_emp sql  
'Viáticos      Hist_comp_emp    sql  
'Viáticos      Hist_fact_emp    sql  
'Viáticos      Hist_dest_emp    sql  
'Viáticos      Hist_otrosgas_emp sql  
'Viáticos      Hist_kms        sql  
'Viáticos      Hist_saldos_emp sql  
'Viáticos      Hist_obj_cli_emp sql  
'Viáticos      Hist_obj_dei_emp sql  
'Viáticos      Hist_obj_otro_emp sql  
'Viáticos      Hist_obj_proy_emp sql  
'Viáticos      Rep_paso_sol    sql  sql  
'Viáticos      Rep_paso_comp   sql  sql  
'Viáticos      Rep_pasoobj     sql  sql  
'Viáticos      Rep_persona     sql  sql
```

```
'Rec. Humanos  Datos_empleado  sql  
'Rec. Humanos  Cat_areas       sql  
'Rec. Humanos  Cat_estados     sql  
'Rec. Humanos  Cat_puestos     sql
```

```
'Gestión      Clientes        sql  
'Gestión      Proyectos      sql  
'Gestión      CatDelegaciones sql
```

```
'Forma(s) que Llama: frmmain, frmreportes  
'Llamada por:   frmmain  
'Modificada por:
```

```
Private Sub cmd_facxfolio_Click()  
    frm_facxfolio.Show vbModal  
    Screen.MousePointer = vbNormal
```

End Sub

```
Private Sub cmdAcepCom_Click()  
    On Error GoTo ManError  
    Dim Ls_tot_bolavion As Single  
    Dim rsgest01 As rdoResultset  
    Dim rsgest02 As rdoResultset  
    Dim numproy As String  
    Dim numcli, numdele As Long  
    Dim LstrVal As String  
    Dim LniCont  
    Dim v_nombre, v_nomaut, v_nomsup As String  
    Dim obj_1, obj_2, obj_3, obj_4 As String  
    Dim d_obj_1, d_obj_2, d_obj_3, d_obj_4 As String  
    Dim tcamb, valim, vhosp, cgas, vavion, vbus, vpeaje As Integer
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Dim vgas, vfoto, vlegal, votros, vkm As Integer
Dim lavion, fecasal, horasal, ibus, iauto, autorm As String

LstrVal = ""
Pb_imprime = 1
If Val(txtfolcom) > 0 Then
    'llena parte del esqueleto reporte de Rec. Hum
    lcad = "select distinct hist_viat_emp.s_folio,hist_viat_emp.nsi_num_emp, hist_viat_emp.d_pfecha_sal,
hist_viat_emp.d_pfecha_ile, hist_viat_emp.nsi_num_empsup, " +
    "hist_viat_emp.nsi_num_empauf, hist_viat_emp.ch_phora_sal,hist_viat_emp.ch_phora_ile, hist_viat_emp.nsi_num_empsup,
hist_viat_emp.nsi_num_empauf, " +
    "hist_dest_emp.nsi_pnum_dest,cat_destinos.nsi_num_dest,cat_destinos.vch_nom_dest, hist_viat_emp.vch_itin_avion,
hist_viat_emp.vch_fec_avion_sal, " +
    "hist_viat_emp.vch_hora_avion_sal, hist_viat_emp.vch_itin_bus, hist_viat_emp.vch_itin_auto, hist_viat_emp.ch_auto_recreat,
hist_viat_emp.m_tipo_camb, " +
    "hist_viat_emp.m_pvia_alimen, hist_viat_emp.m_pvia_hosp, hist_viat_emp.m_costo_gas, hist_viat_emp.m_pvia_avion,
hist_viat_emp.m_pvia_bus, " +
    "hist_viat_emp.m_pvia_peaje, hist_viat_emp.m_pvia_gas, hist_viat_emp.m_pvia_foto, hist_viat_emp.m_pvia_legal,
hist_viat_emp.m_pvia_otros, hist_viat_emp.nsi_km from " +
    "hist_viat_emp,cat_destinos, hist_dest_emp where " +
    "hist_viat_emp.s_folio = hist_dest_emp.s_folio and " +
    "cat_destinos.nsi_num_dest = hist_dest_emp.nsi_pnum_dest and " +
    "vch_logcancela is null and hist_viat_emp.s_folio = " & txtfolcom.Text

Screen.MousePointer = 11
Set rsviat01 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

If Pnum_emp = rsviat01!nsi_num_emp Or Ptipo_usr = "R" Then
    If rsviat01.EOF Then 'no lo encuentra
        MsgBox "Número de folio inexistente o cancelado", vbExclamation, "Reportes"
        txtfolcom.SetFocus
        Screen.MousePointer = 1
    Else 'si lo encuentra jala los datos de la tabla
        If rsviat01!nsi_num_emp > 9000 And rsviat01!nsi_num_emp < 10000 Then
            MsgBox "Es un empleado por Honorarios", vbExclamation, "Reportes"
            lcad = "select distinct cat_honorarios.ch_nombre,cat_honorarios.nsi_nivel,cat_puestos.vch_nom_puesto,
cat_areas.vch_nom_area " +
            "from cat_honorarios, cat_puestos,cat_areas where cat_honorarios.nsi_num_puesto = cat_puestos.nsi_num_puesto " +
            "and cat_honorarios.ni_num_area = cat_areas.ni_num_area and cat_honorarios.nsi_numhono = " &
rsviat01!nsi_num_emp - 9000
            Set rsrh01 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
            'a = rsrh01.RowCount
        Else
            lcad = "select distinct datos_empleado.nsi_num_puesto,datos_empleado.vch_ap_paterno, " +
            "datos_empleado.vch_ap_materno,datos_empleado.vch_nombre,datos_empleado.vch_ext_tel, " +
            "cat_puestos.vch_nom_puesto,datos_personales.ni_num_area,cat_areas.vch_nom_area, " +
            "cat_puestos.nsi_nivel " +
            "from datos_empleado, cat_puestos, datos_personales,cat_areas where " +
            "cat_puestos.nsi_num_puesto = datos_empleado.nsi_num_puesto and " +
            "cat_areas.ni_num_area = datos_personales.ni_num_area and " +
            "datos_personales.nsi_num_emp = datos_empleado.nsi_num_emp and " +
            "datos_empleado.nsi_num_emp = " & rsviat01("nsi_num_emp")
            Set rsrh01 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
        End If
        lcad = "select vch_ap_paterno, vch_ap_materno,vch_nombre " +
            "from datos_empleado where " +
            "nsi_num_emp = " & rsviat01("nsi_num_empauf")
        Set rsrh02 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

        lcad = "select vch_ap_paterno, vch_ap_materno,vch_nombre " +
            "from datos_empleado where " +
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
"nsi_num_emp = " & rsviat01("nsi_num_empsup")
Set rsrh03 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select distinct hist_obj_proy_emp.s_folio, hist_obj_proy_emp.nsi_num_obj," + _
"hist_obj_proy_emp.ch_num_proy, cat_objetivos.vch_nom_corto " + _
"from hist_obj_proy_emp, cat_objetivos where " + _
"hist_obj_proy_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_proy_emp.s_folio = " & txtfolcom.Text
Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select distinct hist_obj_cli_emp.s_folio, hist_obj_cli_emp.nsi_num_obj," + _
"hist_obj_cli_emp.ni_cliente, cat_objetivos.vch_nom_corto " + _
"from hist_obj_cli_emp, cat_objetivos where " + _
"hist_obj_cli_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_cli_emp.s_folio = " & txtfolcom.Text
Set rsviat03 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select distinct hist_obj_del_emp.s_folio, hist_obj_del_emp.nsi_num_obj," + _
"hist_obj_del_emp.ni_num_dele, cat_objetivos.vch_nom_corto " + _
"from hist_obj_del_emp, cat_objetivos where " + _
"hist_obj_del_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_del_emp.s_folio = " & txtfolcom.Text
Set rsviat04 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select distinct hist_obj_otro_emp.s_folio, hist_obj_otro_emp.nsi_num_obj," + _
"hist_obj_otro_emp.vch_desc_visita, cat_objetivos.vch_nom_corto " + _
"from hist_obj_otro_emp, cat_objetivos where " + _
"hist_obj_otro_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_otro_emp.s_folio = " & txtfolcom.Text
Set rsviat05 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

If rsviat02.EOF Then
    numproy = ""
Else
    numproy = rsviat02("ch_num_proy")
End If

If rsviat03.EOF Then
    numcli = 0
Else
    numcli = rsviat03("ni_cliente")
End If

If rsviat04.EOF Then
    numdele = 0
Else
    numdele = rsviat04("ni_num_dele")
End If
' el rsviat04 es de otros y no trae nada de otra base de datos

lcad = "select chNProy, chDescCorta from Proyectos where " + _
"chNproy = " & numproy & ""
Set rsgest01 = bdrdogest.OpenResultSet(Name:=lcad, _
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)

lcad = "select niCliente, chNombre from Clientes where " + _
"niCliente = " & numcli
Set rsgest02 = bdrdogest.OpenResultSet(Name:=lcad, _
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)

lcad = "select ninumdel, chdelegacion from Catdelegaciones where " +
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
"niNumdel = " & numdele
Set rsgest03 = bdrdogest.OpenResultset(Name =!cad, _
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)

icad = "select * from hist_obj_otro_emp where " + _
"s_folio = " & rsviat01!s_folio
Set rsviat07 = bdrdoviat.OpenResultset(Name =!cad, _
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)

If rsgest01.EOF Then
    d_obj_1 = ""
Else
    d_obj_1 = rsgest01("chDescCorta")
End If

If rsgest02.EOF Then
    d_obj_2 = ""
Else
    d_obj_2 = rsgest02("chNombre")
End If

If rsgest03.EOF Then
    d_obj_3 = ""
Else
    d_obj_3 = rsgest03("chdelegacion")
End If

If rsviat07.EOF Then
    d_obj_4 = ""
Else
    d_obj_4 = rsviat07!vch_desc_visita
End If

If rsrh01.EOF Then
    v_nombre = ""
Else
    If rsviat01!nsi_num_emp > 9000 And rsviat01!nsi_num_emp < 10000 Then
        v_nombre = rsrh01!ch_nombre
    Else
        If IsNull(rsrh01!vch_ap_materno) Then
            v_nombre = Trim(rsrh01("vch_ap_paterno")) + "" + Trim(rsrh01("vch_nombre"))
        Else
            v_nombre = Trim(rsrh01("vch_ap_paterno")) + "" + Trim(rsrh01("vch_ap_materno")) + "" +
Trim(rsrh01("vch_nombre"))
        End If
        v_nombre = Trim(rsrh01("vch_ap_paterno")) + "" + Trim(rsrh01("vch_ap_materno")) + "" +
Trim(rsrh01("vch_nombre"))
    End If
End If

If rsrh02.EOF Then
    v_nomaut = ""
Else
    v_nomaut = Trim(rsrh02("vch_ap_paterno")) + "" + Trim(rsrh02("vch_ap_materno")) + "" + Trim(rsrh02("vch_nombre"))
End If

If rsrh03.EOF Then
    v_nomsup = ""
Else
    If IsNull(rsrh03!vch_ap_materno) Then
        v_nomsup = Trim(rsrh03("vch_ap_paterno")) + "" + Trim(rsrh03("vch_nombre"))
    Else
        v_nomsup = Trim(rsrh03("vch_ap_paterno")) + "" + Trim(rsrh03("vch_ap_materno")) + "" + Trim(rsrh03("vch_nombre"))
    End If
End If
```



```
If rsviat02.EOF Then
    obj_1 = ""
Else
    obj_1 = Trim(numproy) + "/" + Trim(rsviat02("vch_nom_corto"))
End If
If rsviat03.EOF Then
    obj_2 = ""
Else
    obj_2 = Trim(d_obj_2) + "/" + Trim(rsviat03("vch_nom_corto"))
End If
If rsviat04.EOF Then
    obj_3 = ""
Else
    obj_3 = Trim(d_obj_3) + "/" + Trim(rsviat04("vch_nom_corto"))
End If
If rsviat05.EOF Then
    obj_4 = ""
Else
    obj_4 = Trim(rsviat07!vch_desc_visita) + "/" + Trim(rsviat05("vch_nom_corto"))
End If

If IsNull(rsviat01("m_tipo_camb")) Then
    tcamb = 0
Else
    tcamb = rsviat01("m_tipo_camb")
End If
If IsNull(rsviat01("m_pvia_alimen")) Then
    valim = 0
Else
    valim = rsviat01("m_pvia_alimen")
End If
If IsNull(rsviat01("m_pvia_hosp")) Then
    vhosp = 0
Else
    vhosp = rsviat01("m_pvia_hosp")
End If
If IsNull(rsviat01("m_costo_gas")) Then
    cgas = 0
Else
    cgas = rsviat01("m_costo_gas")
End If
If IsNull(rsviat01("m_pvia_avion")) Then
    vavion = 0
Else
    vavion = rsviat01("m_pvia_avion")
End If
If IsNull(rsviat01("m_pvia_bus")) Then
    vbus = 0
Else
    vbus = rsviat01("m_pvia_bus")
End If
If IsNull(rsviat01("m_pvia_peaje")) Then
    vpeaje = 0
Else
    vpeaje = rsviat01("m_pvia_peaje")
End If
If IsNull(rsviat01("m_pvia_gas")) Then
    vgas = 0
Else
    vgas = rsviat01("m_pvia_gas")
End If
If IsNull(rsviat01("m_pvia_foto")) Then
    vfoto = 0
Else
    vfoto = rsviat01("m_pvia_foto")
End If
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
" & vgas & " & vfoto & " & vlegal & " & votros & " & vkm & " & iavion & " & fecasal & " + _
" & horasal & " & ibus & " & iauto & " & autorm & " + _
" & d_obj_1 & " & d_obj_2 & " & d_obj_3 & ""
Else
Istrq = "(" & btffolcom.Text & " & rsviat01("nsi_num_emp") & " & v_nombre & " & rsrh01("vch_nom_puesto") & " &
rsrh01("vch_nom_area") & " & rsrh01("vch_ext_tel") & " & +
" & v_nomaut & " & v_nomsup & " & rsviat01("vch_nom_dest") & " & obj_1 & " & obj_2 & " & obj_3 & " &
obj_4 & "" +
" & tcamb & " & valim & " & vhosp & " & cgas & " & vavion & " & vbus & " & vpeaje & " + _
" & vgas & " & vfoto & " & vlegal & " & votros & " & vkm & " & iavion & " & fecasal & " + _
" & horasal & " & ibus & " & iauto & " & autorm & " + _
" & d_obj_1 & " & d_obj_2 & " & d_obj_3 & ""
End If
Istrf = "(s_folio,nsi_num_emp,vch_nombre,vch_nom_puesto,vch_nom_area,vch_num_ext," + _
"vch_nom_comisiona,vch_nom_supaco,vch_destino,objetivo_1,objetivo_2,objetivo_3,objetivo_4," + _
"m_tipo_camb,m_pvia_alimen,m_pvia_hosp,m_costo_gas,m_pvia_avion,m_pvia_bus,m_pvia_peaje," + _
"m_pvia_gas,m_pvia_foto,m_pvia_legal,m_pvia_otros,nsi_km,vch_itin_avion,vch_fec_avion_sal," + _
"vch_hora_avion_sal,vch_itin_bus,vch_itin_auto,ch_auto_recrmat,descobj_1,descobj_2,descobj_3)"

bdrdoviat.Execute "insert into rep_paso_sol" & Istrf & " values " & Istrq

'bdrdoviat.CommitTrans
'bdrdogest.CommitTrans

Call plena_matriz_gastos

Icad = "select * from hist_bol_avion_emp where s_folio=" & btffolcom
Set rsviat01 = bdrdoviat.OpenResultSet(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

If rsviat01.EOF Then
Ls_tot_bolavion = 0
Istrq = "(m_bol_avion)"
Istrf = "(" & Ls_tot_bolavion & ")"
bdrdoviat.Execute "update hist_viat_emp set" & Istrq & "=" & Istrf & " where s_folio = " & btffolcom.Text
Else
Ls_tot_bolavion = 0
rsviat01.MoveFirst
While Not rsviat01.EOF
Ls_tot_bolavion = Ls_tot_bolavion + rsviat01!m_total
rsviat01.MoveNext
Wend
istrq = "(m_bol_avion)"
Istrf = "(" & Ls_tot_bolavion & ")"
bdrdoviat.Execute "update hist_viat_emp set" & Istrq & "=" & Istrf & " where s_folio = " & btffolcom.Text
End If

If Pb_imprime = 1 Then
Ipath = App.Path & "vrep002.rpt"
crrpviat.ReportFileName = Ipath
crrpviat.WindowTitle = "Comprobación de gastos"
If opt_pantalla = True Then crrpviat.Destination = 0
On Error GoTo crrer_comp
crrpviat.Connect = "DSN = viaticos;UID = " & PstrUsuario & ";PWD = " & Ppassword & ";DSQ = viaticos"

Dim LiForm As Integer
LiForm = 22 ' indica el numero de formulas en el reporte
Call vacia_formulas(crrpviat, LiForm) 'Proceso que limpia formulas
Pierr_cr = crrpviat.PrintReport

' Aqui empieza respuesta de objetivos
PstrSQL = "SELECT vch_resp_obj,vch_resp_obj2,vch_resp_obj3," & _
"vch_resp_obj4,vch_resp_obj5,vch_resp_obj6," & _
"vch_resp_obj7,vch_resp_obj8,vch_resp_obj9,vch_resp_obj10" & _
"FROM hist_obj_cli_emp" &
```



# MIGRACION DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
"WHERE s_folio=" & Trim(txtfolcom)
Set rsObjClic = bdrdoviat.OpenResultSet(PstrSQL, rdOpenForwardOnly, rdConcurReadOnly)

PstrSQL = "SELECT vch_resp_obj,vch_resp_obj2,vch_resp_obj3," & _
"vch_resp_obj4,vch_resp_obj5,vch_resp_obj6," & _
"vch_resp_obj7,vch_resp_obj8,vch_resp_obj9,vch_resp_obj10" & _
"FROM hist_obj_del_emp" & _
"WHERE s_folio=" & Trim(txtfolcom)
Set rsObjDele = bdrdoviat.OpenResultSet(PstrSQL, rdOpenForwardOnly, rdConcurReadOnly)

PstrSQL = "SELECT vch_resp_obj,vch_resp_obj2,vch_resp_obj3," & _
"vch_resp_obj4,vch_resp_obj5,vch_resp_obj6" & _
"vch_resp_obj7,vch_resp_obj8,vch_resp_obj9,vch_resp_obj10" & _
"FROM hist_obj_otro_emp" & _
"WHERE s_folio=" & Trim(txtfolcom)
Set rsObjOtro = bdrdoviat.OpenResultSet(PstrSQL, rdOpenForwardOnly, rdConcurReadOnly)

PstrSQL = "SELECT vch_resp_obj,vch_resp_obj2,vch_resp_obj3," & _
"vch_resp_obj4,vch_resp_obj5,vch_resp_obj6" & _
"vch_resp_obj7,vch_resp_obj8,vch_resp_obj9,vch_resp_obj10" & _
"FROM hist_obj_proy_emp" & _
"WHERE s_folio=" & Trim(txtfolcom)
Set rsObjProy = bdrdoviat.OpenResultSet(PstrSQL, rdOpenForwardOnly, rdConcurReadOnly)

If Not rsObjClic.EOF Then
    If Not IsNull(rsObjClic!vch_resp_obj) Then
        PstrObj = "OBJETIVO/CLIENTE"
        pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj
    End If
    If Not IsNull(rsObjClic!vch_resp_obj2) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj2
    If Not IsNull(rsObjClic!vch_resp_obj3) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj3
    If Not IsNull(rsObjClic!vch_resp_obj4) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj4
    If Not IsNull(rsObjClic!vch_resp_obj5) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj5
    If Not IsNull(rsObjClic!vch_resp_obj6) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj6
    If Not IsNull(rsObjClic!vch_resp_obj7) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj7
    If Not IsNull(rsObjClic!vch_resp_obj8) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj8
    If Not IsNull(rsObjClic!vch_resp_obj9) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj9
    If Not IsNull(rsObjClic!vch_resp_obj10) Then pstrobj_clie = pstrobj_clie & rsObjClic!vch_resp_obj10
End If

If Not rsObjDele.EOF Then
    If Not IsNull(rsObjDele!vch_resp_obj) Then
        PstrObj = PstrObj & "OBJETIVO/DELEGACION"
        pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj
    End If
    If Not IsNull(rsObjDele!vch_resp_obj2) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj2
    If Not IsNull(rsObjDele!vch_resp_obj3) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj3
    If Not IsNull(rsObjDele!vch_resp_obj4) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj4
    If Not IsNull(rsObjDele!vch_resp_obj5) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj5
    If Not IsNull(rsObjDele!vch_resp_obj6) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj6
    If Not IsNull(rsObjDele!vch_resp_obj7) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj7
    If Not IsNull(rsObjDele!vch_resp_obj8) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj8
    If Not IsNull(rsObjDele!vch_resp_obj9) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj9
    If Not IsNull(rsObjDele!vch_resp_obj10) Then pstrobj_dele = pstrobj_dele & rsObjDele!vch_resp_obj10
End If

If Not rsObjOtro.EOF Then
    If Not IsNull(rsObjOtro!vch_resp_obj) Then
        PstrObj = PstrObj & "OBJETIVO/OTRO"
        pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj
    End If
    If Not IsNull(rsObjOtro!vch_resp_obj2) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj2
    If Not IsNull(rsObjOtro!vch_resp_obj3) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj3
    If Not IsNull(rsObjOtro!vch_resp_obj4) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj4
    If Not IsNull(rsObjOtro!vch_resp_obj5) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj5
    If Not IsNull(rsObjOtro!vch_resp_obj6) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj6
    If Not IsNull(rsObjOtro!vch_resp_obj7) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj7
    If Not IsNull(rsObjOtro!vch_resp_obj8) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj8
End If
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
If Not IsNull(rsObjOtro!vch_resp_obj9) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj9
If Not IsNull(rsObjOtro!vch_resp_obj10) Then pstrobj_otro = pstrobj_otro & rsObjOtro!vch_resp_obj10
End If
If Not rsObjProy.EOF Then
If Not IsNull(rsObjProy!vch_resp_obj) Then
' PstrObj = PstrObj & "OBJETIVO/PROYECTO "
pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj
End If
If Not IsNull(rsObjProy!vch_resp_obj2) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj2
If Not IsNull(rsObjProy!vch_resp_obj3) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj3
If Not IsNull(rsObjProy!vch_resp_obj4) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj4
If Not IsNull(rsObjProy!vch_resp_obj5) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj5
If Not IsNull(rsObjProy!vch_resp_obj6) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj6
If Not IsNull(rsObjProy!vch_resp_obj7) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj7
If Not IsNull(rsObjProy!vch_resp_obj8) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj8
If Not IsNull(rsObjProy!vch_resp_obj9) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj9
If Not IsNull(rsObjProy!vch_resp_obj10) Then pstrobj_proy = pstrobj_proy & rsObjProy!vch_resp_obj10 & Chr(13) &
Chr(10)
End If

'repuesta de objetivos de otros
If Len(pstrobj_otro) > 0 Then
PstrSQL = "(s_folio,ch01"
LstrVal = "(" & Trim(btfolcom) & "," & Mid(pstrobj_otro, 1, 95)
For LniCont = 1 To 39
If Len(pstrobj_otro) > ((95 * LniCont) + 1) Then
If LniCont > 8 Then
PstrSQL = PstrSQL & ",ch" & Trim(Str(LniCont + 1))
Else
PstrSQL = PstrSQL & ",ch0" & Trim(Str(LniCont + 1))
End If
LstrVal = LstrVal & "," & Mid(pstrobj_otro, (95 * LniCont) + 1, 95)
End If
Next LniCont
PstrSQL = PstrSQL & ")"
LstrVal = LstrVal & ")"
'bdrdoviat.BeginTrans
bdrdoviat.Execute "INSERT INTO rep_pasooobj " & PstrSQL & " VALUES " & LstrVal
'bdrdoviat.CommitTrans
End If

'respuesta obj. de delegaciones
If Len(pstrobj_dele) > 0 Then
PstrSQL = "(s_folio,ch01"
LstrVal = "(" & Trim(btfolcom) & "," & Mid(pstrobj_dele, 1, 95)
For LniCont = 1 To 39
If Len(pstrobj_dele) > ((95 * LniCont) + 1) Then
If LniCont > 8 Then
PstrSQL = PstrSQL & ",ch" & Trim(Str(LniCont + 1))
Else
PstrSQL = PstrSQL & ",ch0" & Trim(Str(LniCont + 1))
End If
LstrVal = LstrVal & "," & Mid(pstrobj_dele, (95 * LniCont) + 1, 95)
End If
Next LniCont
PstrSQL = PstrSQL & ")"
LstrVal = LstrVal & ")"
'bdrdoviat.BeginTrans
bdrdoviat.Execute "INSERT INTO rep_pasooobj " & PstrSQL & " VALUES " & LstrVal
'bdrdoviat.CommitTrans
End If

'respuesta obj. de clientes
If Len(pstrobj_clie) > 0 Then
PstrSQL = "(s_folio,ch01"
LstrVal = "(" & Trim(btfolcom) & "," & Mid(pstrobj_clie, 1, 95)
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
For LniCont = 1 To 39
  If Len(pstrobj_clie) > ((95 * LniCont) + 1) Then
    If LniCont > 8 Then
      PstrSQL = PstrSQL & ",ch" & Trim(Str(LniCont + 1))
    Else
      PstrSQL = PstrSQL & ",ch0" & Trim(Str(LniCont + 1))
    End If
    LstrVal = LstrVal & "," & Mid(pstrobj_clie (95 * LniCont) + 1, 95)
  End If
Next LniCont
PstrSQL = PstrSQL & ")"
LstrVal = LstrVal & ")"
'bdrdoviat.BeginTrans
bdrdoviat.Execute "INSERT INTO rep_pasobj " & PstrSQL & " VALUES " & LstrVal
'bdrdoviat.CommitTrans
End If

'respuesta obj. de proyectos
If Len(pstrobj_proy) > 0 Then
  PstrSQL = "(s_folio,ch01"
  LstrVal = "(" & Trim(txtfolcom) & " " & Mid(pstrobj_proy 1, 95)
  For LniCont = 1 To 39
    If Len(pstrobj_proy) > ((95 * LniCont) + 1) Then
      If LniCont > 8 Then
        PstrSQL = PstrSQL & ",ch" & Trim(Str(LniCont + 1))
      Else
        PstrSQL = PstrSQL & ",ch0" & Trim(Str(LniCont + 1))
      End If
      LstrVal = LstrVal & "," & Mid(pstrobj_proy, (95 * LniCont) + 1, 95)
    End If
  Next LniCont
  PstrSQL = PstrSQL & ")"
  LstrVal = LstrVal & ")"
  'bdrdoviat.BeginTrans
  bdrdoviat.Execute "INSERT INTO rep_pasobj " & PstrSQL & " VALUES " & LstrVal
  'bdrdoviat.CommitTrans
End If
"Aquí termina reporte de objetivos

LiForm = 50 ' indica el numero de fórmulas en el reporte
Call vacia_formulas(crrepiat, LiForm) 'Proceso que limpia fórmulas

lpath = App.Path & "\rep003.rpt"
crrepiat.ReportFileName = lpath
crrepiat.WindowTitle = "Informe de la comisión"
Pierr_cr = crrepiat.PrintReport

LstrVal = ""
pstrobj_clie = ""
pstrobj_dele = ""
pstrobj_otros = ""
pstrobj_proy = ""

bdrdoviat.Execute "delete from rep_pasobj " & txtfolcom Text

On Error GoTo ManError
txtfolcom.Enabled = False
txtfolcom.Enabled = False
txtfolcom.Text = ""
cmdAcepCom.Enabled = False
cmdAcepSol.Enabled = False
cmdSolicitud.Enabled = True
Screen.MousePointer = 1
rsviat01.Close
rsviat02.Close
rsviat03.Close
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
rsviat04.Close
rsviat05.Close
rsviat06.Close
rsrh01.Close
rsrh02.Close
rsrh03.Close
rsgest01.Close
rsgest02.Close
rsgest03.Close
MsgBox "Tu Reporte fue impreso sin problemas", vbExclamation, "Reportes"
End If
Screen.MousePointer = 1
End If
Else
MsgBox "Tu Pasword no corresponde al Empleado", vbExclamation, "Reportes"
Screen.MousePointer = 1
End If
Else
MsgBox "Tiene que teclear un numero de Folio", vbExclamation, "Reportes"
btfolcom.SetFocus
End If
Exit Sub
crerr_comp:
If Pierr_cr > 0 Then
MsgBox F_crError(Pierr_cr), vbCritical, "Error"
Call P_GuardaError(Me.Name, Err.Number, F_crError(Pierr_cr))
End If

ManError:
Screen.MousePointer = 1
End Sub

Private Sub cmdacepsol_Click()
On Error GoTo ManError
Dim v_nombre, v_nomaut, v_nomsup As String
Dim obj_1, obj_2, obj_3, obj_4 As String
Dim d_obj_1, d_obj_2, d_obj_3, d_obj_4 As String
Dim tcamb, valim, vhosp, cgas, vavion, vbus, vpeaje As Integer
Dim vgas, vfoto, viegal, votros, vkm As Integer
Dim iavion, fecasal, horasal, ibus, iauto, autorm As String
Dim rsgest01 As rdoResultset
Dim rsgest02 As rdoResultset
Dim numproy As String
Dim numcli, numdele As Integer

If Val(btfolso) > 0 Then
' llena parte del esqueleto reporte de Rec Hum.
lcad = "select distinct hist_viat_emp.s_folio,hist_viat_emp.nsi_num_emp, hist_viat_emp.d_pfecha_sal,
hist_viat_emp.d_pfecha_lle, hist_viat_emp.nsi_num_empsup, " + _
"hist_viat_emp.nsi_num_empaut, hist_viat_emp.ch_phora_sal,hist_viat_emp.ch_phora_lle, hist_viat_emp.nsi_num_empsup,
hist_viat_emp.nsi_num_empaut, " + _
"hist_dest_emp.nsi_pnum_dest,cat_destinos.vch_nom_dest, hist_viat_emp.vch_itin_avion,
hist_viat_emp.vch_fec_avion_sal, " + _
"hist_viat_emp.vch_hora_avion_sal, hist_viat_emp.vch_itin_bus, hist_viat_emp.vch_itin_auto, hist_viat_emp.ch_auto_recmat,
hist_viat_emp.m_tipo_camb, " + _
"hist_viat_emp.m_pvia_alimen, hist_viat_emp.m_pvia_hosp, hist_viat_emp.m_costo_gas, hist_viat_emp.m_pvia_avion,
hist_viat_emp.m_pvia_bus, " + _
"hist_viat_emp.m_pvia_peaje, hist_viat_emp.m_pvia_gas, hist_viat_emp.m_pvia_foto, hist_viat_emp.m_pvia_legal,
hist_viat_emp.m_pvia_otros, hist_viat_emp.nsi_km from " + _
"hist_viat_emp.cat_destinos, hist_dest_emp where " + _
"hist_viat_emp.s_folio = hist_dest_emp.s_folio and " + _
"cat_destinos.nsi_num_dest = hist_dest_emp.nsi_pnum_dest and " + _
"vch_logcancela is null and hist_viat_emp.s_folio = " & btfolso.Text

Screen.MousePointer = 11
Set rsviat01 = bdrdoviat.OpenResultset(Name:=lcad,
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Type:=rdOpenDynamic, LockType:=rdConcurValues)

If Pnum_emp = rsviat01!nsi_num_emp Or Ptipo_usr = "R" Then
If rsviat01.EOF Then 'no lo encuentra
MsgBox "Número de folio inexistente o cancelado", vbExclamation, "Reportes"
btfofolso.SetFocus
Screen.MousePointer = 1
Else 'si lo encuentra jala los datos de la tabla
If rsviat01!nsi_num_emp > 9000 And rsviat01!nsi_num_emp < 10000 Then
MsgBox "Es un empleado por Honorarios", vbExclamation, "Reportes"
lcad = "select distinct cat_honorarios.ch_nombre.cat_honorarios.nsi_nivel, cat_puestos.vch_nom_puesto,
cat_areas.vch_nom_area " + _
"from cat_honorarios, cat_puestos,cat_areas where cat_honorarios.nsi_num_puesto = cat_puestos.nsi_num_puesto " + _
"and cat_honorarios.ni_num_area = cat_areas.ni_num_area and cat_honorarios.nsi_numhono = " & rsviat01!nsi_num_emp
-9000
Set rsrh01 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
'a = rsrh01.RowCount
Else
lcad = "select distinct datos_empleado.nsi_num_puesto,datos_empleado.vch_ap_paterno," + _
"datos_empleado.vch_ap_materno,datos_empleado.vch_nombre,datos_empleado.vch_ext_tel," + _
"cat_puestos.vch_nom_puesto,datos_personales.ni_num_area,cat_areas.vch_nom_area," + _
"cat_puestos.nsi_nivel " + _
"from datos_empleado, cat_puestos, datos_personales,cat_areas where " + _
"cat_puestos.nsi_num_puesto = datos_empleado.nsi_num_puesto and " + _
"cat_areas.ni_num_area = datos_personales.ni_num_area and " + _
"datos_personales.nsi_num_emp = datos_empleado.nsi_num_emp and " + _
"datos_empleado.nsi_num_emp =" & rsviat01("nsi_num_emp")
Set rsrh01 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
End If
lcad = "select vch_ap_paterno, vch_ap_materno,vch_nombre " + _
"from datos_empleado where " + _
"nsi_num_emp =" & rsviat01("nsi_num_empaut")
Set rsrh02 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select vch_ap_paterno, vch_ap_materno,vch_nombre " + _
"from datos_empleado where " + _
"nsi_num_emp =" & rsviat01("nsi_num_empsup")
Set rsrh03 = bdrdorh.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select distinct hist_obj_proy_emp.s_folio, hist_obj_proy_emp.nsi_num_obj," + _
"hist_obj_proy_emp.ch_num_proy, cat_objetivos.vch_nom_corto " + _
"from hist_obj_proy_emp, cat_objetivos where " + _
"hist_obj_proy_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_proy_emp.s_folio =" & btfofolso.Text
Set rsviat02 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select distinct hist_obj_cli_emp.s_folio, hist_obj_cli_emp.nsi_num_obj," + _
"hist_obj_cli_emp.ni_cliente, cat_objetivos.vch_nom_corto " + _
"from hist_obj_cli_emp, cat_objetivos where " + _
"hist_obj_cli_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_cli_emp.s_folio =" & btfofolso.Text
Set rsviat03 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

lcad = "select distinct hist_obj_del_emp.s_folio, hist_obj_del_emp.nsi_num_obj," + _
"hist_obj_del_emp.ni_num_dele, cat_objetivos.vch_nom_corto " + _
"from hist_obj_del_emp, cat_objetivos where " + _
"hist_obj_del_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _
"hist_obj_del_emp.s_folio =" & btfofolso.Text
Set rsviat04 = bdrdoviat.OpenResultSet(Name:=lcad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)
```



## MIGRACIÓN DE SOFTWARE

### Anexos

### Programas fuentes (Visual Basic)

```
lcard = "select distinct hist_obj_otro_emp.s_folio, hist_obj_otro_emp.nsi_num_obj, " + _  
"hist_obj_otro_emp.vch_desc_visita, cat_objetivos.vch_nom_corto " + _  
"from hist_obj_otro_emp, cat_objetivos where " + _  
"hist_obj_otro_emp.nsi_num_obj = cat_objetivos.nsi_num_obj and " + _  
"hist_obj_otro_emp.s_folio = " & txtfoliol.Text  
Set rsviat05 = bdrdoviat.OpenResultset(Name:=lcard, _  
Type:=rdOpenDynamic, LockType:=rdConcurValues)  
  
If rsviat02.EOF Then  
    numproy = ""  
Else  
    numproy = rsviat02("ch_num_proy")  
End If  
  
If rsviat03.EOF Then  
    numcli = 0  
Else  
    numcli = rsviat03("ni_cliente")  
End If  
  
If rsviat04.EOF Then  
    numdele = 0  
Else  
    numdele = rsviat04("ni_num_dele")  
End If  
' el rsviat04 es de otros y no trae nada de otra base de datos  
  
lcard = "select chNProy, chDescCorta from Proyectos where " + _  
"chNproy = " & numproy & ""  
Set rsgest01 = bdrdogest.OpenResultset(Name:=lcard, _  
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)  
  
lcard = "select niCliente, chNombre from Clientes where " + _  
"niCliente = " & numcli  
Set rsgest02 = bdrdogest.OpenResultset(Name:=lcard, _  
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)  
  
lcard = "select ninumdel, chdelegacion from Catdelegaciones where " + _  
"niNumdel = " & numdele  
Set rsgest03 = bdrdogest.OpenResultset(Name:=lcard, _  
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)  
  
lcard = "select * from hist_obj_otro_emp where " + _  
"s_folio = " & rsviat01.s_folio  
Set rsviat07 = bdrdoviat.OpenResultset(Name:=lcard, _  
Type:=rdOpenKeyset, LockType:=rdConcurReadOnly)  
  
If rsgest01.EOF Then  
    d_obj_1 = ""  
Else  
    d_obj_1 = Trim(rsgest01("chDescCorta"))  
End If  
  
If rsgest02.EOF Then  
    d_obj_2 = ""  
Else  
    d_obj_2 = rsgest02("chNombre")  
End If  
  
If rsgest03.EOF Then  
    d_obj_3 = ""  
Else  
    d_obj_3 = rsgest03("chdelegacion")  
End If
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
If rsviat07.EOF Then
    d_obj_4 = ""
Else
    d_obj_4 = rsviat07!vch_desc_visita
End If

If rsrh01.EOF Then
    v_nombre = ""
Else
    If rsviat01!nsi_num_emp > 9000 And rsviat01!nsi_num_emp < 10000 Then
        v_nombre = rsrh01!ch_nombre
    Else
        If IsNull(rsrh01!vch_ap_materno) Then
            v_nombre = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_nombre"))
        Else
            v_nombre = Trim(rsrh01("vch_ap_paterno")) + " " + Trim(rsrh01("vch_ap_materno")) + " " +
Trim(rsrh01("vch_nombre"))
        End If
    End If
End If

If rsrh02.EOF Then
    v_nomauf = ""
Else
    v_nomauf = Trim(rsrh02("vch_ap_paterno")) + " " + Trim(rsrh02("vch_ap_materno")) + " " + Trim(rsrh02("vch_nombre"))
End If

If rsrh03.EOF Then
    v_nomsup = ""
Else
    If IsNull(rsrh03!vch_ap_materno) Then
        v_nomsup = Trim(rsrh03("vch_ap_paterno")) + " " + Trim(rsrh03("vch_nombre"))
    Else
        v_nomsup = Trim(rsrh03("vch_ap_paterno")) + " " + Trim(rsrh03("vch_ap_materno")) + " " + Trim(rsrh03("vch_nombre"))
    End If
End If

If rsviat02.EOF Then
    obj_1 = ""
Else
    obj_1 = Trim(numproy) + "/" + Trim(rsviat02("vch_nom_corto"))
End If

If rsviat03.EOF Then
    obj_2 = ""
Else
    obj_2 = Trim(d_obj_2) + "/" + Trim(rsviat03("vch_nom_corto"))
End If

If rsviat04.EOF Then
    obj_3 = ""
Else
    obj_3 = Trim(d_obj_3) + "/" + Trim(rsviat04("vch_nom_corto"))
End If

If rsviat05.EOF Then
    obj_4 = ""
Else
    obj_4 = Trim(rsviat07!vch_desc_visita) + "/" + Trim(rsviat05("vch_nom_corto"))
End If

If IsNull(rsviat01("m_tipo_camb")) Then
    tcamb = 0
Else
    tcamb = rsviat01("m_tipo_camb")
End If

If IsNull(rsviat01("m_pvia_alimen")) Then
    valim = 0
Else
    valim = rsviat01("m_pvia_alimen")
End If
```



```
End If
If IsNull(rsviat01("m_pvia_hosp")) Then
    vhosp = 0
Else
    vhosp = rsviat01("m_pvia_hosp")
End If
If IsNull(rsviat01("m_costo_gas")) Then
    cgas = 0
Else
    cgas = rsviat01("m_costo_gas")
End If
If IsNull(rsviat01("m_pvia_avion")) Then
    vavion = 0
Else
    vavion = rsviat01("m_pvia_avion")
End If
If IsNull(rsviat01("m_pvia_bus")) Then
    vbus = 0
Else
    vbus = rsviat01("m_pvia_bus")
End If
If IsNull(rsviat01("m_pvia_peaje")) Then
    vpeaje = 0
Else
    vpeaje = rsviat01("m_pvia_peaje")
End If
If IsNull(rsviat01("m_pvia_gas")) Then
    vgas = 0
Else
    vgas = rsviat01("m_pvia_gas")
End If
If IsNull(rsviat01("m_pvia_foto")) Then
    vfoto = 0
Else
    vfoto = rsviat01("m_pvia_foto")
End If
If IsNull(rsviat01("m_pvia_legal")) Then
    vlegal = 0
Else
    vlegal = rsviat01("m_pvia_legal")
End If
If IsNull(rsviat01("m_pvia_otros")) Then
    votros = 0
Else
    votros = rsviat01("m_pvia_otros")
End If
If IsNull(rsviat01("nsi_km")) Then
    vkm = 0
Else
    vpkm = rsviat01("nsi_km")
End If

If IsNull(rsviat01("vch_itin_avion")) Then
    iavion = ""
Else
    iavion = rsviat01("vch_itin_avion")
End If
If IsNull(rsviat01("vch_fec_avion_sal")) Then
    fecasal = ""
Else
    fecasal = rsviat01("vch_fec_avion_sal")
End If

If IsNull(rsviat01("vch_hora_avion_sal")) Then
    horasal = ""
Else
```



# MIGRACION DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
horasal = rsviat01("vch_hora_avion_sal")
End If
If IsNull(rsviat01("vch_itin_bus")) Then
ibus = ""
Else
ibus = rsviat01("vch_itin_bus")
End If
If IsNull(rsviat01("vch_itin_auto")) Then
iauto = ""
Else
iauto = rsviat01("vch_itin_auto")
End If
If IsNull(rsviat01("ch_auto_recrmat")) Then
autorm = ""
Else
autorm = rsviat01("ch_auto_recrmat")
End If

'bdrdoviat.BeginTrans

lcard = "select * from rep_paso_sol"
Set rsviat06 = bdrdoviat.OpenResultset(Name:=lcard, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

If rsviat06.EOF Then
MsgBox "No hay datos en reporte", vbExclamation, "Reportes"
Else
bdrdoviat.Execute "delete from rep_paso_sol"
End If

If rsviat01.nsi_num_emp > 9000 And rsviat01.nsi_num_emp < 10000 Then
Istrq = "(" & txtfoliol.Text & "," & rsviat01("nsi_num_emp") & "," & v_nombre & "," & rsrh01("vch_nom_puesto") & "," &
rsrh01("vch_nom_area") & "," & rsviat01("vch_ext_tef") & "," & v_nomaut & "," & v_nomsup & "," & rsviat01("vch_nom_dest") & "," & obj_1 & "," & obj_2 & "," & obj_3 & "," &
obj_4 & "," &
"& tcamb & "," & valim & "," & vhosp & "," & cgas & "," & vavion & "," & vbus & "," & vpeaje & "," &
vagas & "," & vfoto & "," & vlegal & "," & votros & "," & vkm & "," & iavion & "," & fecasal & "," &
horasal & "," & ibus & "," & iauto & "," & autorm & "," &
d_obj_1 & "," & d_obj_2 & "," & d_obj_3 & "," & d_obj_4 & ")")
Else
Istrq = "(" & txtfoliol.Text & "," & rsviat01("nsi_num_emp") & "," & v_nombre & "," & rsrh01("vch_nom_puesto") & "," &
rsrh01("vch_nom_area") & "," & rsviat01("vch_ext_tef") & "," & v_nomaut & "," & v_nomsup & "," & rsviat01("vch_nom_dest") & "," & obj_1 & "," & obj_2 & "," & obj_3 & "," &
obj_4 & "," &
"& tcamb & "," & valim & "," & vhosp & "," & cgas & "," & vavion & "," & vbus & "," & vpeaje & "," &
vagas & "," & vfoto & "," & vlegal & "," & votros & "," & vkm & "," & iavion & "," & fecasal & "," &
horasal & "," & ibus & "," & iauto & "," & autorm & "," &
d_obj_1 & "," & d_obj_2 & "," & d_obj_3 & "," & d_obj_4 & ")")
End If

Istrf = "(s_folio,nsi_num_emp,vch_nombre,vch_nom_puesto,vch_nom_area,vch_num_ext, " +
"vch_nom_comisiona,vch_nom_supaco,vch_destino,objetivo_1,objetivo_2,objetivo_3,objetivo_4, " +
"m_tipo_camb,m_pvvia_alimen,m_pvvia_hosp,m_costo_gas,m_pvvia_avion,m_pvvia_bus,m_pvvia_peaje, " +
"m_pvvia_gas, m_pvvia_foto, m_pvvia_legal, m_pvvia_otros,nsi_km,vch_itin_avion,vch_fec_avion_sal, " +
"vch_hora_avion_sal,vch_itin_bus,vch_itin_auto,ch_auto_recrmat,descobj_1,descobj_2, descobj_3, descobj_4)"

bdrdoviat.Execute "insert into rep_paso_sol " & Istrf & " values " & Istrq

'bdrdoviat.CommitTrans
'bdrdogest.CommitTrans

On Error GoTo crerr_soli
lpath = App.Path & "vrep001.rpt"
crrpvat.ReportFileName = lpath
If opt_pantalla = True Then
crrpvat.Destination = 0
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Else
    crrepiat.Destination = 1
End If

crrepiat.Connect = "DSN = viaticos;UID = " & PstrUsuario & ";PWD = " & Ppassword & ";DSQ = viaticos"

Dim LiForm As Integer
LiForm = 11 ' indica el numero de formulas en el reporte
Call vacia_formulas(crrepiat, LiForm) 'Proceso que limpia formulas

Pierr_cr = crrepiat.PrintReport

On Error GoTo ManError
txtfolSol.Enabled = False
txtfolcom.Enabled = False
txtfolSol.Text = ""
cmdAcepCom.Enabled = False
cmdAcepSol.Enabled = False
cmdComprobacion.Enabled = True
Screen.MousePointer = 1
rsviat01.Close
rsviat02.Close
rsviat03.Close
rsviat04.Close
rsviat05.Close
rsviat06.Close
rsrh01.Close
rsrh02.Close
rsrh03.Close
rsgest01.Close
rsgest02.Close
rsgest03.Close
MsgBox "Tu Reporte fue impreso sin problemas", vbExclamation, "Reportes"
End If
Else
    MsgBox "Tu Password no corresponde al Empleado", vbExclamation, "Reportes"
    Screen.MousePointer = 1
End If
Else
    MsgBox "Tiene que teclear un numero de Folio", vbExclamation, "Reportes"
    txtfolSol.SetFocus
End If
Screen.MousePointer = 1
Exit Sub
crerr_soli
If Pierr_cr > 0 Then
    MsgBox F_crError(Pierr_cr), vbCritical, "Error"
    Call P_GuardaError(Me.Name, Err.Number, F_crError(Pierr_cr))
End If
ManError:
    Screen.MousePointer = 1
End Sub

Private Sub cmdComprobacion_Click()
    txtfolSol.Enabled = False
    txtfolcom.Enabled = True
    txtfolcom.SetFocus
    cmdAcepCom.Enabled = True
    cmdAcepSol.Enabled = False
    cmdSolicitud.Enabled = False
End Sub

Private Sub cmdrepgal_Click()
    frm_reportes.Show
End Sub
```



# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Private Sub cmdrepor_Click()
    Screen.MousePointer = vbHourglass
    frmReportAnt.Show
    frmreportes.Hide
    Screen.MousePointer = vbNormal
End Sub

Private Sub cmdSalir_Click()
    Screen.MousePointer = vbHourglass
    frmMain.Show
    Unload Me
    Screen.MousePointer = vbNormal
End Sub

Private Sub cmdSolicitud_Click()
    btfolcom.Enabled = False
    btfolsol.Enabled = True
    btfolsol.SetFocus
    cmdAcepSol.Enabled = True
    cmdAcepCom.Enabled = False
    cmdComprobacion.Enabled = False
End Sub

Private Sub Form_Load()
    On Error GoTo ManError
    btfolsol.Enabled = False
    btfolcom.Enabled = False
    cmdAcepSol.Enabled = False
    cmdAcepCom.Enabled = False
    If Ptipo_usr = "R" Then
        cmdrepregral.Visible = True
        cmdrepor.Visible = True
    Else
        cmdrepregral.Visible = False
        cmdrepor.Visible = False
    End If
    Centrar Me

Exit Sub
ManError:
    Screen.MousePointer = 0
    If rdoErrors.Count > 0 Then
        Dim lerror As rdoError
        For Each lerror In rdoErrors
            MsgBox F_RdoError(lerror), vbCritical, "Error"
            Call P_GuardaError(Me.Name, lerror.Number, F_RdoError(lerror))
        Next
        rdoErrors.Clear
    Else
        MsgBox F_ManError(Err.Number), vbCritical, "Error"
        Call P_GuardaError(Me.Name, Err.Number, Err.Description)
    End If
End Sub

Private Sub SSCommand1_Click()

End Sub

Private Sub btfolcom_KeyPress(KeyAscii As Integer)
    If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
        Else
            MsgBox "Introducir Numeros", vbExclamation, "Reportes"
            KeyAscii = 0
        End If
    End Sub
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Private Sub txtfolso1_KeyPress(KeyAscii As Integer)
If (KeyAscii <= 57 And KeyAscii >= 48) Or KeyAscii = 46 Or KeyAscii = 8 Then
Else
MsgBox "Introducir Numeros", vbExclamation, "Reportes"
KeyAscii = 0
End If
End Sub
Private Sub llena_matriz_gastos()
' llena la matriz de gastos dia a dia por concepto
Dim Lsproc_00, Lsproc_01, Lsproc_02, Lsproc_03, Lsproc_04, Lsproc_05, Lsproc_06 As Single
Dim Lsiva_00, Lsiva_01, Lsiva_02, Lsiva_03, Lsiva_04, Lsiva_05, Lsiva_06 As Single
Dim LbNumdias As Byte
Dim LbNumcon_dia As Byte
Dim ifecha As String
Dim LbConcepto As Byte
Dim Lsfactor As Single
Dim Lhorasal1 As Date
Dim LhoraLle1 As Date
Dim Lstr_cuotas As String
Dim Lni_c00 As Single
Dim Lni_c01 As Single
Dim Lni_c02 As Single
Dim Lni_c03 As Single
Dim Lni_c04 As Single
Dim Lni_c05 As Single
Dim Lni_c06 As Single
Dim Lni_c07 As Single
Dim Lni_c08 As Single
Dim Lni_c09 As Single
Dim Lni_c10 As Single
Dim Lni_c11 As Single
Dim Lni_c12 As Single
Dim Lni_c13 As Single

Lni_c01 = Lni_c02 = Lni_c03 = Lni_c04 = Lni_c05 = Lni_c06 = Lni_c07 = Lni_c08 = Lni_c09 = Lni_c10 = Lni_c11 = Lni_c12 =
Lni_c13 = 0
Lni_c00 = 0
Lsproc_00 = 0
Lsproc_01 = 0
Lsproc_02 = 0
Lsproc_03 = 0
Lsproc_04 = 0
Lsproc_05 = 0
Lsproc_06 = 0
Lssum_cuotas = 0
Lsiva_00 = 0
Lsiva_01 = 0
Lsiva_02 = 0
Lsiva_03 = 0
Lsiva_04 = 0
Lsiva_05 = 0
Lsiva_06 = 0
LbNumdias = 0
LbNumcon_dia = 0
Ls_cuotas = ""
Screen.MousePointer = 11
Icad = "select s_folio, d_fec_afecta, m_rcuota_dia from hist_comp_emp" + _
"where s_folio = " & txtfolcom.Text
Set rsviat01 = bdrdoviat.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

LbNumdias = rsviat01.RowCount

Icad = "select s_folio, ch_rhora_sal, ch_rhora_lle from hist_viat_emp" + _
"where s_folio = " & txtfolcom.Text
Set rsviat03 = bdrdoviat.OpenResultset(Name:=Icad,
```





# MIGRACIÓN DE SOFTWARE

## Anexos

### Programas fuentes (Visual Basic)

```
Type:=rdOpenDynamic, LockType:=rdConcurValues)
If rsviat01.EOF Then
    MsgBox "No se ha realizado esta comprobación", vbExclamation, "Reportes"
    Pb_imprime = 2
Else
    bdrdoviat.Execute "delete from rep_paso_comp"
    rsviat01.MoveFirst
    'Barre Dia por Dia
    For Lcont = 0 To LbNumdias - 1
        lfecha = Format(rsviat01!d_fec_afecta)
        lanio = Format(DatePart("yyyy", lfecha), "0000")
        lmes = Format(DatePart("m", lfecha), "00")
        ldia = Format(DatePart("d", lfecha), "00")
        lfecha = lmes & "/" & ldia & "/" & lanio

        'Selecciona las facturas de ese dia
        lcad = "select s_folio, d_fec_afecta, nsi_cve_gasto, m_monto_autofact, " + _
            "nsi_porc_iva, m_iva_monto from hist_fact_emp where " + _
            "d_fec_afecta = " & lfecha & " and s_folio = " & txtfolcom Text
        Set rsviat02 = bdrdoviat.OpenResultset(Name=lcad, _
            Type:=rdOpenDynamic, LockType:=rdConcurValues)

        'barre concepto por concepto ese dia
        If rsviat02.EOF Then 'lhp
            'lhp lo modifique el 17/nov/99 porque no actualizaba el monto final porque
            'sacaba si no se cargaba alguna comprobacion
            'Exit Sub
        Else
            rsviat02.MoveFirst
        End If
        LbNumcon_dia = rsviat02.RowCount
        For lcont1 = 0 To rsviat02.RowCount - 1 'Llena el monto p/c cpto
            Select Case rsviat02!nsi_cve_gasto
                Case 0
                    Lni_c00 = Lni_c00 + rsviat02!m_monto_autofact
                Case 1
                    Lni_c01 = Lni_c01 + rsviat02!m_monto_autofact
                Case 2
                    Lni_c02 = Lni_c02 + rsviat02!m_monto_autofact
                Case 3
                    Lni_c03 = Lni_c03 + rsviat02!m_monto_autofact
                Case 4
                    Lni_c04 = Lni_c04 + rsviat02!m_monto_autofact
                Case 5
                    Lni_c05 = Lni_c05 + rsviat02!m_monto_autofact
                Case 6
                    Lni_c06 = Lni_c06 + rsviat02!m_monto_autofact
                Case 7
                    Lni_c07 = Lni_c07 + rsviat02!m_monto_autofact
                Case 8
                    Lni_c08 = Lni_c08 + rsviat02!m_monto_autofact
                Case 9
                    Lni_c09 = Lni_c09 + rsviat02!m_monto_autofact
                Case 10
                    Lni_c10 = Lni_c10 + rsviat02!m_monto_autofact
                Case 11
                    Lni_c11 = Lni_c11 + rsviat02!m_monto_autofact
                Case 12
                    Lni_c12 = Lni_c12 + rsviat02!m_monto_autofact
                Case 13
                    Lni_c13 = Lni_c13 + rsviat02!m_monto_autofact
            End Select
            rsviat02.MoveNext
        Next
        Istrq = "(" & txtfolcom & "," & lfecha & "," & rsviat01!m_rcuota_dia & "," & Lni_c00 & "," & Lni_c01 & "," & Lni_c02 & "," &
            Lni_c03 & "," & Lni_c04 & "," & Lni_c05 & "," & Lni_c06 & "," & Lni_c07 & "," & Lni_c08 & "," & Lni_c09 & "," & Lni_c10 & "," &
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
Lni_c11 & "." & Lni_c12 & "." & Lni_c13 & ")")
Istrf = "(s_folio,d_fec_afecta,
m_rcuota_dia,cpto_0,cpto_1,cpto_2,cpto_3,cpto_4,cpto_5,cpto_6,cpto_7,cpto_8,cpto_9,cpto_10,cpto_11,cpto_12,cpto_13)"
bdrdoviat.Execute "insert into rep_paso_comp " & Istrf & " values " & Istrq
Lni_c01 = 0
Lni_c02 = 0
Lni_c03 = 0
Lni_c04 = 0
Lni_c05 = 0
Lni_c06 = 0
Lni_c07 = 0
Lni_c08 = 0
Lni_c09 = 0
Lni_c10 = 0
Lni_c11 = 0
Lni_c12 = 0
Lni_c13 = 0
Lni_c00 = 0
rsviat01.MoveNext
Next

' Llena el acumulado total por concepto en hist_viat_emp

Icad = "select * from hist_fact_emp " + _
"where s_folio = " & txtfolcom.Text
Set rsviat04 = bdrdoviat.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

Icad = "select s_folio, m_rcuota_dia from rep_paso_comp " + _
"where s_folio = " & txtfolcom.Text
Set rsviat05 = bdrdoviat.OpenResultset(Name:=Icad, _
Type:=rdOpenDynamic, LockType:=rdConcurValues)

If rsviat04.RowCount > 0 Then
rsviat04.MoveFirst
For Lcont = 0 To rsviat04.RowCount - 1

Select Case rsviat04!nsi_cve_gasto
Case 0, 1
Lsproc_00 = Lsproc_00 + rsviat04!m_monto_autofact
Lsiva_00 = Lsiva_00 + rsviat04!m_monto_autofact * rsviat04!nsi_porcentaje_iva * 0.01
Case 2, 3
Lsproc_01 = Lsproc_01 + rsviat04!m_monto_autofact
Lsiva_01 = Lsiva_01 + rsviat04!m_monto_autofact * rsviat04!nsi_porcentaje_iva * 0.01
' Case 6 To 9 para sacar boleto de avion
Case 4 To 9
Lsproc_02 = Lsproc_02 + rsviat04!m_monto_autofact
Lsiva_02 = Lsiva_02 + rsviat04!m_monto_autofact * rsviat04!nsi_porcentaje_iva * 0.01
Case 10
Lsproc_03 = Lsproc_03 + rsviat04!m_monto_autofact
Lsiva_03 = Lsiva_03 + rsviat04!m_monto_autofact * rsviat04!nsi_porcentaje_iva * 0.01
Case 11
Lsproc_04 = Lsproc_04 + rsviat04!m_monto_autofact
Lsiva_04 = Lsiva_04 + rsviat04!m_monto_autofact * rsviat04!nsi_porcentaje_iva * 0.01
Case 12
Lsproc_05 = Lsproc_05 + rsviat04!m_monto_autofact
Lsiva_05 = Lsiva_05 + rsviat04!m_monto_autofact * rsviat04!nsi_porcentaje_iva * 0.01
Case 13
Lsproc_06 = Lsproc_06 + rsviat04!m_monto_autofact
Lsiva_06 = Lsiva_06 + rsviat04!m_monto_autofact * rsviat04!nsi_porcentaje_iva * 0.01
End Select

rsviat04.MoveNext
Next

Istrq =
```



## MIGRACIÓN DE SOFTWARE

### Anexos

#### Programas fuentes (Visual Basic)

```
"(m_comp_hotel,m_comp_iva_hotel,m_comp_alimen,m_comp_iva_alimen,m_comp_transp,m_comp_iva_transp,m_comp_foto,m_
comp_iva_foto." + _
    "m_comp_tel,m_comp_iva_tel,m_comp_legal,m_comp_iva_legal,m_comp_otros,m_comp_iva_otros)"
Istrf = "(" & Lsproc_00 & "," & Lsiva_00 & "," & Lsproc_01 & "," & Lsiva_01 & "," & Lsproc_02 & "," & Lsiva_02 & "," &
Lsproc_03 & "," & Lsiva_03 & _
    "," & Lsproc_04 & "," & Lsiva_04 & "," & Lsproc_05 & "," & Lsiva_05 & "," & Lsproc_06 & "," & Lsiva_06 & ")"

bdrdoviat.Execute "update hist_viat_emp set" & Istrq & "=" & Istrf & " where s_folio =" & btfolcom.Text

    End If
End If
End Sub
```