

UNIVERSIDAD TECNICA DE AMBATO

FACULTAD DE INGENIERIA EN SISTEMAS

**CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES E
INFORMATICOS**

TEMA:

“Desarrollo de una aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA”

Proyecto de Pasantía de Grado, presentado como requisito previo a la obtención del Título de Ingeniero en Sistemas Computacionales e Informáticos.

Autor:

Ricardo Rodrigo Rivera Acurio

Tutor:

Ing. Teresa Freire

Ambato – Ecuador

2006

APROBACIÓN DEL TUTOR

En calidad de Tutor del trabajo de investigación sobre el tema:

“DESARROLLO DE UNA APLICACIÓN PARA LA GESTIÓN DE SERVICIOS EN LA VENTA DE TANQUEROS EN EL MÓDULO COMERCIAL DE EMAPA”, desarrollado por el señor Ricardo Rodrigo Rivera Acurio, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, de la Universidad Técnica de Ambato, considero que dicho informe investigativo, reúne los requisitos y méritos suficientes para ser sometidos a la evaluación del tribunal de grado, que el Honorable Consejo Directivo designe, para su correspondiente estudio y calificación.

Ambato, Noviembre 2006

Atentamente,

Ing. Teresa Freire

AUTORIA

El presente trabajo de investigación “**DESARROLLO DE UNA APLICACIÓN PARA LA GESTIÓN DE SERVICIOS EN LA VENTA DE TANQUEROS EN EL MÓDULO COMERCIAL DE EMAPA**”, es absolutamente original, autentico y personal, en tal virtud, el contenido efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Noviembre 2006

Ricardo Rivera

DEDICATORIA

Quiero dedicar este trabajo, a mi Esposa e Hija, razones principales de mi vida; a mis Padres y hermanas quienes han sido mi fuerza necesaria para cada día

AGRADECIMIENTO

Mi sincero agradecimiento a todas aquellas personas que han intervenido en este proceso, especialmente a mis Padres y hermanas que siempre fueron el apoyo necesario en todo momento, a mi Esposa e Hija que han sido la fuerza para llevar a cabo mi objetivo, a mis Amigos y Maestros con los que pude compartir una parte de mi vida.

INDICE

PORTADA.....	i
APROBACION DEL TUTOR.....	ii
AUTORIA.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	iv
INDICE.....	vi
RESUMEN EJECUTIVO.....	ix
INTRODUCCION.....	x

CAPITULO I

1.1. TEMA.....	1
1.2. PLANTEAMIENTO DEL PROBLEMA.....	1
1.3. JUSTIFICACIÓN.....	4
1.4. OBJETIVOS.....	4

CAPITULO II

2.1. ANTECEDENTES.....	6
2.2. BASE LEGAL.....	6
2.3. CATEGORIZACIONES CONCEPTUALES.....	8
2.4. HIPÓTESIS.....	29
2.5. SEÑALAMIENTO DE LAS VARIABLES DE HIPÓTESIS.....	29

CAPITULO III

3.1. ENFOQUE.....	31
3.2. MODALIDAD BÁSICA DE LA INVESTIGACIÓN.....	31
3.3. NIVEL O TIPO DE LA INVESTIGACIÓN.....	32
3.4. POBLACIÓN Y MUESTRA.....	32
3.5. RECOLECCIÓN DE INFORMACIÓN.....	32
3.6. PROCESAMIENTO Y ANÁLISIS.....	33

CAPITULO IV

4.1. ANALISIS DEL SISTEMA.....	34
4.2. DIAGRAMA DE FLUJOS DE DATOS.....	40
4.3. DICCIONARIO DE DATOS.....	44
4.4. DISEÑO DE LA BASE DE DATOS.....	60
4.5. DISEÑO DE INTERFACES.....	62

CAPITULO V

5.1. CODIFICACIÓN.....	73
5.2. PRUEBAS.....	139

CAPITULO VI

6.1. INICIO DEL FUNCIONAMIENTO DEL SISTEMA.....	141
6.2. ADIESTRAMIENTO DEL PERSONAL.....	141

CAPITULO VII

7.1. CONCLUSIONES.....	142
7.2. RECOMENDACIONES.....	142
BIBLIOGRAFIA.....	144
ANEXOS.....	145

RESUMEN EJECUTIVO

La Empresa Municipal de Agua Potable y Alcantarillado de Ambato – EMAPA empeñada en brindar un mejor servicio a la colectividad, ha visto la necesidad de mejorar la venta de agua en tanqueros; este proceso se lo realiza con un sistema obsoleto por lo que se ve obligado a realizar apuntes en papeles, lo que conlleva a pérdida de tiempo y desperdicio de recursos.

La empresa colaboró dando la información necesaria para la realización de la aplicación, además de tener la mejor disponibilidad para las mejoras que se realizaron.

El desarrollo de una nueva aplicación que maneje de una mejor manera el proceso de venta de agua en tanqueros, ha sido imprescindible para la empresa, ésta se desarrollo utilizando la Tecnología ASP .NET de Microsoft, y los datos se los almacena en SQL Server 2000, además se la integró al sistema general que maneja la empresa, con lo que se ha logrado que la venta se realice de una manera mas rápida y organizada.

Cabe indicar que se utilizaron herramientas adicionales como Macromedia para el desarrollo y diseño de entorno WEB, Power Designer para el manejo de Base de Datos, entre los principales.

Este sistema permite a los clientes de la EMAPA realizar su pago en cualquiera de las ventanillas que la empresa posee o con los diferentes convenios que posee con otras instituciones.

Se logró cumplir con todos los objetivos planteados, y en el tiempo que fue establecido.

INTRODUCCION

Los avances de la tecnología en los últimos años han sido muy significativos por lo que las organizaciones en todo el mundo se han visto obligadas a implantar nuevos y mejores sistemas informáticos que puedan cumplir con los requerimientos actuales.

La Empresa Municipal de Agua Potable y Alcantarillado de Ambato – EMAPA, es una institución que busca el progreso, por lo que para brindar un mejor servicio a la comunidad ambateña día a día busca mejorar internamente mediante la implantación de nuevos sistemas, capacitación a sus empleados compra de nuevos equipos, etc.

EMAPA cuenta con un sistema que maneja gran parte de los procesos que se realizan en la misma, aunque dicho sistema no se encuentra en la capacidad de abarcar ciertos módulos, los mismos que por ser de gran importancia para la empresa necesitan ser desarrollados, los mismos que servirán para dar una mejora significativa a las necesidades de la empresa.

CAPITULO I

1.1. TEMA

“Desarrollo de una aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA”

1.2. PLANTEAMIENTO DEL PROBLEMA

1.2.1. Contextualización

En la actualidad los avances en la informática han hecho que las grandes empresas se vean obligadas a actualizarse para de este modo cumplir con la demanda de mercado, y de esta manera poder satisfacer las exigencias de una sociedad cada vez más competitiva.

Se puede observar el gran esfuerzo que día a día las organizaciones de todo el mundo realizan en sus actividades diarias, lo que hace imprescindible la implantación de nuevos y mejores sistemas que cumplan completamente las necesidades de las mismas; y de esta manera poder generar mayores beneficios además de brindar un mejor y más rápido servicio a sus clientes.

En el Ecuador los sistemas informáticos cada vez van ganando terreno dentro de las empresas que buscan mejorar sus réditos económicos además de brindar un mejor servicio a sus clientes, que cada vez exigen mejores prestaciones por parte de las mismas.

La Empresa Municipal de Agua Potable y Alcantarillado de Ambato cuenta con un sistema de gestión de varios servicios que presta a la ciudadanía, aunque todavía no es suficiente para abarcar todos los procesos que ésta realiza; en la presente investigación se abarca un módulo que aun no ha sido cubierto, siendo el mismo de gran importancia para la misma.

Los servicios como la venta de tanqueros en el módulo comercial, son de una gran importancia para la Empresa Municipal de Agua Potable y Alcantarillado de Ambato, ya que son actividades que se realizan diariamente.

La presente investigación se centrará en el estudio y desarrollo de una aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de la Empresa Municipal de Agua Potable y Alcantarillado de Ambato.

1.2.2. Análisis Crítico

La falta de un módulo actualizado que controle la venta de tanqueros en la Empresa Municipal de Agua Potable y Alcantarillado de Ambato, ha hecho que se siga trabajando con un sistema obsoleto, mismo que ya no se encuentra en la posibilidad de manejar toda la información necesaria que se requiere para este fin, lo que conlleva al desperdicio de recursos, pérdida de información, falta de confiabilidad; ocasionando falta de organización en las actividades institucionales.

1.2.3 Prognosis

Si no se corrige a tiempo este problema con la implantación de un sistema que cubra todas las necesidades del módulo comercial se puede llegar a crisis económicas y organizacionales, manipulación incorrecta de la

información, pérdida de competitividad y otros problemas como la falta de confiabilidad en la institución. Por lo que, se ha visto la necesidad de desarrollar el proyecto de gestión de servicios para la venta de tanqueros en el módulo comercial de la Empresa Municipal de Agua Potable y Alcantarillado de Ambato, con el fin de utilizar de una forma eficiente los recursos informáticos y minimizar de esta manera la pérdida de confidencialidad de la información.

1.2.4. Formulación del Problema

¿Qué beneficios brindará a la Empresa Municipal de Agua Potable y Alcantarillado de Ambato el desarrollo de una aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial?

1.2.5. Preguntas Directrices

¿Cómo se lleva a cabo el proceso actual de gestión de servicios en la venta de tanqueros en el módulo comercial de la Empresa Municipal de Agua Potable y Alcantarillado de Ambato?

¿Cuáles serán las ventajas que tendrá la automatización del sistema de gestión de servicios en la venta de tanqueros?

1.2.6. Delimitación del Problema

El problema se enfoca en la manera en que se viene llevando el registro y control sobre el proceso de gestión de servicios para la venta de tanqueros, el mismo que se ocasiona por llevar un sistema que cumpla con las necesidades del módulo comercial de la Empresa Municipal de Agua Potable y Alcantarillado de la ciudad de Ambato, ya que actualmente no se emite un buen control sobre la información que se maneja.

1.3. JUSTIFICACIÓN

La automatización de los procesos, es un factor de vital importancia en la Empresa Municipal de Agua Potable y Alcantarillado, ya que mediante dicha automatización se pretende brindar un mejor servicio a la comunidad.

El desarrollo e implementación de este sistema está destinado a cambiar la forma en la que se maneja la venta de tanqueros de agua a la ciudadanía, para que se pueda agilizar este servicio, además de contar con una gran confiabilidad en los datos ingresados ya que todo el proceso será automatizado.

Los recursos de la institución se manejarán de una mejor manera una vez que el nuevo sistema se encuentre en funcionamiento, porque el mismo permitirá al encargado de su manejo, tener información a la mano sin desperdicio de tiempo y sin la molestia de tener información en papeles; por ende se dará una mejora económica a la institución.

Este trabajo será un aporte técnico para la institución ya que permitirá que este actualizada y sea competitiva. Además servirá como aporte científico para futuras implementaciones. Se ha conseguido un total apoyo a la idea de implementar un sistema automatizado que maneje la venta de tanqueros.

1.4. OBJETIVOS

1.4.1. General

Desarrollar una aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de la Empresa Municipal de Agua

Potable y Alcantarillado de Ambato, utilizando la tecnología de ASP.NET, para optimizar tiempo y recursos.

1.4.2. Específicos

- Analizar la manera en que se realiza actualmente el proceso de venta de tanqueros en la Empresa Municipal de Agua Potable y Alcantarillado de Ambato.
- Identificar los inconvenientes que tiene actualmente el módulo comercial de la Empresa Municipal de Agua Potable y Alcantarillado, en lo referente al trámite de venta de tanqueros.
- Investigar los beneficios y utilidades de la tecnología ASP. NET y utilizarlos en el desarrollo de la aplicación.
- Disminuir el tiempo de servicio en la venta de tanqueros y los recursos utilizados, en el módulo comercial de la Empresa Municipal de Agua Potable y Alcantarillado.

CAPITULO II

MARCO TEORICO

2.1. ANTECEDENTES

Revisados los archivos de la Facultad de Ingeniería en Sistemas de la Universidad Técnica de Ambato; en lo que concierne a la venta de tanqueros de agua, no se han encontrado trabajos relacionados con este tema, por lo que no se puede tomar bases de indagaciones anteriores.

2.2. BASE LEGAL

EMAPA es una empresa pública con domicilio en la ciudad de Ambato, misma que se rige por la Ley de Régimen Municipal, por la presente ordenanza, por ordenanzas del Ilustre Municipio de Ambato entre las que se pueden enumerar:

- Acta de Constitución de la Empresa, fines y objetivos
- Ley de Régimen Municipal
- Ley de Servicio Civil y Carrera Administrativa
- Ley Orgánica de Administración Financiera y Control
- Ley de Bienes del Sector Público
- Ley del Control de Gasto Público
- Código de Trabajo

Además, se rige por sus propios reglamentos internos, mismos que se enumeran a continuación:

- Reglamentos de Representación y Residencia.
- Reglamentos de Procedimientos de Proyectos Hidráulicos y Sanitarios.
- Reglamentos del Funcionamiento del Directorio.
- Reglamentos de Remate y Baja de Bienes.
- Reglamento de Interconexiones.
- Reglamento de Adquisiciones.
- Reglamento de Fondo de Caja Chica.

Es importante hacer referencia a un tema muy importante como lo es la propiedad intelectual, que tiene que ver con las creaciones de la mente: las invenciones, las obras literarias y artísticas, los símbolos, los nombres, las imágenes y los dibujos, modelos utilizados en el comercio, etc.

En general, desde un punto de vista jurídico, pueden distinguirse dos clases de derechos inherentes al derecho de autor:

- **DERECHOS MORALES:** Son derechos inalienables e intransmisibles que engloban principalmente el derecho a la paternidad de la obra (ser reconocido autor de una obra), el derecho a la integridad de la misma (impedir cualquier deformación, modificación, alteración o atentado contra ella) y el derecho a decidir en qué forma se difundirá la obra. Del mismo modo, la ley española, ha decidido ampliar dichos derechos intransmisibles a otros de menor importancia, como el derecho a retirar la obra del mercado, o el derecho a acceder al ejemplar único o raro de la misma.

- **DERECHOS PATRIMONIALES.** Son los derechos de explotación sobre la obra, generalmente con contenido económico. Se incluye aquí los derechos de reproducción, distribución, comunicación pública, puesta a disposición y transformación, aunque ésta clasificación no es números clausus ya que la práctica del sector incluye otros como el derecho de sincronización.

La Empresa Municipal de Agua Potable y Alcantarillado consciente de los derechos de propiedad intelectual, cuenta con las licencias de las herramientas de desarrollo de programas entre las cuales podemos numerar:

1. Visual Studio 6.0
2. SQL Server 2000 Edición Standard

Además cabe indicar que se encuentran por adquirir las siguientes licencias.

1. Visual Studio .NET
2. Macromedia

2.3. CATEGORIZACIONES CONCEPTUALES

2.3.1. Software

También conocido como programática o equipamiento lógico es el conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Se trata del conjunto de instrucciones que permite la utilización del ordenador o computador. El software es la parte intangible de la computadora, es decir programas, aplicaciones, etc.

2.3.1.1. Definición

Probablemente la definición más formal de software es la atribuida a la IEEE en su estándar 729: la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

Bajo esta definición el concepto de software va más allá de los programas de cómputo en sus distintas formas: código fuente, binario o código ejecutable, además de su documentación. Es decir, el software es todo lo intangible.

2.3.1.2. Tipología

El software se clasifica en dos categorías:

- **Software de sistema.** Consistente en todo aquel software cuyo propósito es facilitar la ejecución de otro software. Entran en esta categoría:
 - Sistemas operativos.
 - Compiladores.
 - Gestores de bases de datos.

- **Software de aplicación.** Consistente en aquel software que automatiza un sistema de información, es decir, con relevancia para un fin concreto. Entran en esta categoría:
 - Procesadores de texto.
 - Hojas de cálculo.
 - Aplicaciones Web.

2.3.1.3. Formas

El software adopta varias formas en distintos momentos de su ciclo de vida:

- **Código fuente:** escrito por programadores. Contiene el conjunto de instrucciones destinadas a la computadora.
- **Código objeto:** resultado del uso de un compilador sobre el código fuente. Consiste en una traducción de éste último. El código objeto no es directamente inteligible por el ser humano, pero tampoco es directamente entendible por la computadora. Se trata de una representación intermedia del código fuente.
- **Código ejecutable:** resultado de enlazar uno o varios fragmentos de código objeto. Constituye un archivo binario con un formato tal que el sistema operativo es capaz de cargarlo en la memoria de un ordenador, y proceder a su ejecución. El código ejecutable es directamente inteligible por la computadora.

2.3.2. Bases de datos

Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

2.3.2.1 Tipos de bases de datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

Según la variabilidad de los datos almacenados

- ✓ **Bases de datos estáticas.** Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- ✓ **Bases de datos dinámicas.** Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

Según el contenido

✓ **Bases de datos bibliográficas.**

Solo contienen un surrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo.

✓ **Bases de datos numéricas**

Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio.

✓ **Bases de datos de texto completo**

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

✓ **Directorios**

Un ejemplo son las guías telefónicas en formato electrónico.

✓ **Banco de imágenes, audio, video, multimedia, etc.**

Como su nombre lo indica, almacenan información en distintos formatos.

✓ **Bases de datos o "bibliotecas" de información Biológica**

Son bases de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas.

Se pueden considerar en varios subtipos:

- Aquellas que almacenan secuencias de nucleótidos o proteínas.
- Las bases de datos de rutas metabólicas
- Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas
- Bases de datos clínicas
- Bases de datos bibliográficas (biológicas)

2.3.2.2. Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

✓ **Bases de datos jerárquicas**

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

✓ **Bases de datos de red**

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

✓ **Bases de datos relacionales**

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y *campos* (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años '80 (1980-1989) la aparición de dBASE produjo una revolución en los lenguajes de programación y sistemas de administración de datos. Aunque nunca debe olvidarse que dBase no utilizaba SQL como lenguaje base para su gestión.

✓ **Bases de datos orientadas a objetos**

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros).

La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

Se está trabajando en **SQL3**, que es el estándar de SQL92 ampliado, que soportará los nuevos conceptos orientados a objetos y mantendría compatibilidad con SQL92.

✓ **Bases de datos documentales**

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesauro es un sistema de índices optimizado para este tipo de bases de datos.

✓ **Base de datos deductivas**

Un sistema de **base de datos deductivas**, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos.

También las bases de datos deductivas son llamadas base de datos lógicas, a raíz de que se basan en lógica matemática.

✓ **Gestión de bases de datos distribuida**

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etc.

2.3.3. Sistemas de Control

La palabra control ha sido utilizada con varios y diferentes sentidos.

Control como función coercitiva y restrictiva, para inhibir o impedir conductas indeseables, como llegar con atraso al trabajo o a clases, hacer escándalos, etcétera.

Control como verificación de alguna cosa, para apreciar si está correcto, como verificar pruebas o notas.

Control como comparación con algún estándar de referencia como pensar una mercadería en otra balanza, comparar notas de alumnos, etcétera.

Control como función administrativa, esto es, como la cuarta etapa del proceso administrativo.

Constituye la cuarta y última etapa del proceso administrativo. Este tiende a asegurar que las cosas se hagan de acuerdo con las

expectativas o conforme fue planeado, organizado y dirigido, señalando las fallas y errores con el fin de repararlos y evitar que se repitan.

Establecimiento de Estándares: Un estándar puede ser definido como una unidad de medida que sirve como modelo, guía o patrón con base en la cual se efectúa el control.

Los estándares son criterios establecidos contra los cuales pueden medirse los resultados, representan la expresión de las metas de planeación de la empresa o departamento en términos tales que el logro real de los deberes asignados pueda medirse contra ellos.

Los estándares pueden ser físicos y representar cantidades de productos, unidades de servicio, horas-hombre, velocidad, volumen de rechazo, etc., o pueden estipularse en términos monetarios como costos, ingresos o inversiones; u otros términos de medición.

Medición de resultados: Si el control se fija adecuadamente y si existen medios disponibles para determinar exactamente que están haciendo los subordinados, la comparación del desempeño real con lo esperado es fácil. Pero hay actividades en las que es difícil establecer estándares de control por lo que se dificulta la medición.

Corrección: Si como resultado de la medición se detectan desviaciones, corregir inmediatamente esas desviaciones y establecer nuevos planes y procedimientos para que no se vuelvan a presentar.

Retroalimentación: Una vez corregidas las desviaciones, reprogramar el proceso de control con la información obtenida causante del desvío.

Existen cuatro factores que deben ser considerados al aplicar el proceso de control.

- Cantidad
- Tiempo
- Costo
- Calidad

Los tres primeros son de carácter cuantitativo y el último es eminentemente cualitativo.

El factor cantidad se aplica a actividades en la que el volumen es importante.

A través del factor tiempo se controlan las fechas programadas.

El costo es utilizado como un indicador de la eficiencia administrativa, ya que por medio de él se determinan las erogaciones de ciertas actividades.

La calidad se refiere a las especificaciones que debe reunir un cierto producto o ciertas funciones de la empresa.

Establece medidas para corregir las actividades, de tal forma que se alcancen los planes exitosamente.

Se aplica a todo: a las cosas, a las personas y a los actos.

Determina y analiza rápidamente las causas que pueden originar desviaciones para que no vuelvan a presentarse en el futuro.

Localiza los sectores responsables de la administración, desde el momento en que se establecen medidas correctivas.

Proporciona información acerca de la situación de la ejecución de los planes, sirviendo como fundamento al reiniciarse el proceso de la planeación.

Reduce costos y ahorra tiempo al evitar errores.

Su aplicación incide directamente en la racionalización de la administración y consecuentemente, en el logro de la productividad de todos los recursos de la empresa.

2.3.4. Empresa Pública

Se llama empresa pública o empresa estatal a toda aquella que es propiedad del Estado, sea este nacional, municipal o de cualquier otro estrato administrativo. La gran mayoría de los estados del mundo controla empresas con diferentes fines, que pueden ser proveer servicios públicos, incentivar la producción del país o generar empleo, por nombrar algunos.

Teóricamente una empresa pública busca la rentabilidad social sobre la rentabilidad económica típica de las empresas privadas.

Con la aparición del Estado de bienestar en Europa y el populismo en América Latina luego de la Segunda Guerra Mundial, se hizo muy común la estatización o nacionalización de empresas privadas que pasaron a la órbita pública; hasta mediados de la década del '70 del siglo XX hubo en casi todos los países del mundo gran cantidad

de empresas públicas en el marco del denominado Estado empresario.

En aquellos países con un sistema económico socialista y una economía planificada como lo fue la Unión Soviética y lo es Cuba, el Estado controla la totalidad de la economía o gran parte de ella, por lo que prácticamente cualquier empresa existente es estatal.

No obstante, así como se dice que las empresas públicas buscan el bien de la sociedad, son blanco de críticas por parte de la ortodoxia económica y la doctrina neoliberal, que las describe como ineficientes por naturaleza, a causa de la burocracia administrativa y la falta de estímulo de los trabajadores, planteando además que la libre competencia mejora la productividad.

Con el auge de esta serie de teorías a fines de los años '70 y su posterior hegemonía en los '80 y '90, muchas antiguas empresas del estado fueron privatizadas y convertidas en empresas privadas, dando lugar al llamado libre mercado.

Latinoamérica y Europa del Este (después de la desarticulación de la Unión Soviética, fueron un campo de pruebas para las privatizaciones, promovidas por el FMI y la OMC, entre otros organismos internacionales. Sin embargo, hubo diferencias en el grado de privatización de la economía entre los diferentes países.

La mayoría de los países Europeos conserva muchas empresas del estado, como la RENFE española o la British Railways británica, esta última con la particularidad de haber sido privatizada y luego reestatizada.

Por razones estructurales, la mayoría de los países africanos retuvieron también las empresas públicas. En los Estados Unidos de América las empresas públicas siempre fueron pocas, por lo que el avance del neoliberalismo como doctrina económica no implicó muchos cambios.

En la actualidad, sobre todo en países latinoamericanos, se está viviendo una leve vuelta del Estado a la economía. Se pueden citar como ejemplos la reestatización de la petrolera venezolana PDVSA y la empresa argentina Correo Argentino, descendiente de la antigua ENCoTel.

2.3.5. ASP.NET

ASP.NET es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

- **Mejor rendimiento.** ASP.NET es un código de Common Language Runtime compilado que se ejecuta en el servidor. A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.
- **Compatibilidad con herramientas de primer nivel.** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de

Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.

- **Eficacia y flexibilidad.** Debido a que ASP.NET se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad de Common Language Runtime garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.
- **Simplicidad.** ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios. Por ejemplo, el marco de trabajo de página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación, y controlar eventos en un sencillo modelo de procesamiento de formularios de tipo Visual Basic. Además, Common Language Runtime simplifica la programación, con servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados.
- **Facilidad de uso.** ASP.NET emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web.

Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Una aplicación ASP.NET Framework se implementa en un servidor sencillamente mediante la copia de los archivos necesarios al servidor. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.

- **Escalabilidad y disponibilidad.** ASP.NET se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.
- **Posibilidad de personalización y extensibilidad.** ASP.NET presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado. La implementación de la autenticación personalizada o de los servicios de estado nunca ha sido más fácil.
- **Seguridad.** Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

2.3.6. Desarrollo De Aplicaciones Web

2.3.6.1. Plan de Trabajo

El plan de trabajo general utilizado está enfocado en 4 fases, las cuales se definen como:

Fase de Levantamiento de información

- ✓ Plan de trabajo
- ✓ Levantamiento de información con los usuarios funcionales
- ✓ Documentación funcional
- ✓ Restricciones
- ✓ Seguridad

Fase de Diseño

- ✓ Investigación
- ✓ Diseño funcional
- ✓ Elaboración de propuesta
- ✓ Evaluación de factibilidad
- ✓ Elaboración de prototipos
- ✓ Diseño gráfico de la aplicación

Fase de Desarrollo

- ✓ Desarrollo de la aplicación o portal por secciones

Fase de Documentación

- ✓ Manual de usuario
- ✓ Ayuda en línea
- ✓ Manual del sistema

Pruebas e Implementación

- ✓ Configuración del ambiente de producción
- ✓ Implantación del sistema
- ✓ Pruebas funcionales
- ✓ Aseguramiento de Calidad
- ✓ Soporte Post Producción

2.3.6.2. Metodología de Desarrollo

- ✓ Elaboración de un Plan de Trabajo
- ✓ Levantamiento de información
- ✓ Análisis de la solución
- ✓ Diseño de la solución
- ✓ Evaluación de factibilidad
- ✓ Elaboración de prototipo
- ✓ Desarrollo de la solución
- ✓ Implementación de la solución
- ✓ Documentación
- ✓ Pruebas funcionales
- ✓ Aseguramiento de la Calidad (*Principios de Nilsen*)
- ✓ Puesta en Producción
- ✓ Soporte Post Producción

2.3.6.3. Ambientes de Trabajo

Existen 3 ambientes de trabajo:

✓ ***Desarrollo:***

Este es el ambiente de trabajo para los desarrolladores (grupo de tecnología) del sistema y es donde se lleva a cabo la programación, prueba y depuración de las páginas y programas.

✓ ***Administradores y pruebas:***

Este es el ambiente de mantenimiento del sistema. Aquí se montan los programas de los administradores, además de una copia funcional del sistema completo para ver los cambios que van realizando los administradores. En este ambiente los analistas pueden mantener, modificar y visualizar los cambios que realizan sin afectar el ambiente de producción (o ambiente final).

✓ ***Producción:***

Esta es el área de publicación de las páginas, donde los usuarios finales van a acceder al sistema. Los administradores no van a poder ser accedidos por este medio, solo se van a poder acceder por el ambiente de pruebas y administradores, de esta forma se tienen niveles de seguridad mayor para los mismos. Los cambios de las páginas solo van a llegar a este ambiente cuando sean aprobados por los editores.

2.3.7. Gestión de Servicios en la Venta de Tanqueros en el Módulo Comercial de La Empresa de Agua Potable y Alcantarillado.

EMAPA es una empresa que brinda servicios a la comunidad entre ellos se contempla la venta de agua en tanqueros, actividad que es de gran importancia ya que se la realiza diariamente.

Este trámite se lo realiza en el módulo comercial de la Empresa, para lo que se cuenta con un sistema obsoleto que no se encuentra en la capacidad de cubrir todas las necesidades requeridas.

Los ciudadanos que requieren este servicio, tienen que acercarse al módulo comercial, indicar sus datos, el número de tanqueros que necesitan, la dirección de entrega y otros datos; luego tiene que cancelar el valor de la factura en una ventanilla y tomar su comprobante.

Este proceso necesita un nuevo estudio y análisis para poder integrar al sistema principal que maneja la empresa un módulo actualizado que maneje de una manera eficaz dicho trámite, a más de optimizar el recurso tiempo y garantizar la integridad de la información; con lo que se podrá brindar un mejor servicio a la comunidad.

2.4. HIPÓTESIS

El desarrollo de una aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de la Empresa Municipal de Agua Potable y Alcantarillado de Ambato, utilizando tecnología ASP.NET, optimizara el tiempo y los recursos utilizados en dicho proceso.

2.5. SEÑALAMIENTO DE LAS VARIABLES DE HIPÓTESIS

2.5.1. Variable Independiente: El desarrollo de una aplicación para la gestión de servicios en la venta de tanqueros.

2.5.2. Variable Dependiente: El tiempo y los recursos utilizados en el proceso de venta de tanqueros

CAPITULO III

METODOLOGIA

3.1. ENFOQUE

La presente investigación se encuentra dentro de un enfoque cualitativo porque:

- Busca la comprensión de fenómenos sociales.
- Orientado al descubrimiento de hipótesis
- Es Holístico

Además está dentro de un enfoque cuantitativo, porque:

- Orientado a la comprobación de hipótesis.
- Pone énfasis en el resultado.
- Asume una realidad estable.

3.2. MODALIDAD BÁSICA DE LA INVESTIGACIÓN

La investigación se centrara dentro de los siguientes ámbitos:

- Investigación bibliográfica; para obtener información que servirá para la fundamentación teórica,

- Investigación experimental para realizar pruebas de comprobación el funcionamiento del sistema que se esta implementando,
- Investigación aplicada, porque busca la aplicación o utilización de los conocimientos que se adquieren; e,
- Investigación de campo, para obtener la información primaria.

3.3. NIVEL O TIPO DE LA INVESTIGACIÓN

Para realizar el presente trabajo se partirá de un nivel de investigación exploratorio para llegar a un nivel explicativo hasta comprobar la hipótesis.

3.4. POBLACIÓN Y MUESTRA

La población a investigarse será de 12 personas que laboran en el Módulo Comercial, por lo que no se determina muestra ya que el universo es muy pequeño.

3.5. RECOLECCIÓN DE INFORMACIÓN

Las técnicas y los instrumentos a aplicarse para la recolección y registro de información son:

- la entrevista para conocer detalles del sistema existente y de cómo se maneja el proceso de venta de tanqueros; utilizando un cuestionario de preguntas base.
- la observación que permitirá analizar de manera directa diferentes aspectos del manejo en sí de la venta de tanqueros; esto se

complementará con cuaderno de apuntes en el que se registrará todo lo que sea necesario en este proceso.

- La encuesta, la misma que servirá para obtener información relacionada al manejo del sistema de venta de tanqueros; el cuestionario servirá como instrumento para recoger datos.

3.6. PROCESAMIENTO Y ANÁLISIS

Para el análisis y procesamiento de la información se seguirán los siguientes pasos:

- Revisión y análisis de la información recogida
- Tabulaciones y gráficos
- Manejo de información.
- Conclusiones

CAPITULO IV

DESARROLLO DEL SISTEMA

4.1. ANALISIS DEL SISTEMA

4.1.1. Estudio De La Situación Actual

El proceso de venta de tanqueros actualmente se lo realiza dentro del departamento comercial, mediante un software en el que se registra los datos de la persona que requiere el servicio, se ingresa la fecha de entrega y el tipo de tanquero; luego de lo cual se imprime un recibo que es entregado al usuario para que con el mismo realice el correspondiente pago en cajas, recibe una factura con la cual debe volver al departamento comercial presentar el comprobante de pago, para que finalmente se autorice el despacho del tanquero de agua.

4.1.2. Estudio De Factibilidades

4.1.2.1. Factibilidad Económica

Los costos que involucran el desarrollo de la aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, se obtuvieron mediante el proceso de estimación de costos que se detalla a continuación.

COSTO DEL SISTEMA ACTUAL

Cantidad	Personal	Dólares Mensual
1	Administrador	400.00
1	Secretaria	300.00
Gasto Personal (GP)		700.00

Cantidad	Equipo de Computo	Valor	Depreciación (1.67%)
1	Servidor	4000 USD	66.80
1	Computador	700 USD	11.69
Gasto Equipo de Cómputo (GE)			78.49

Materiales Directos	Dólares Mensual
Materiales de Oficina	100.00
Gasto Materiales Directos (GMD)	100.00

Varios	Dólares Mensual
Mantenimiento de Computadores	10.00
Útiles de Limpieza	10.00
Gasto Varios (GV)	20.00

COSTO DEL SISTEMA ACTUAL	
GASTOS	Dólares Mensual
Gasto Personal (GP)	700.00
Gasto Equipo de Cómputo (GE)	78.49
Gasto Materiales Directos (GMD)	100.00
Gasto Varios (GV)	20.00
SUBTOTAL	898.49
Gastos Indirectos (0.67%)(SUBTOTAL)	5.69
TOTAL	904.19

SON NOVECIENTOS CUATRO DOLARES CON DIECINUEVE CENTAVOS AMERICANOS

COSTO DEL SISTEMA NUEVO

Cantidad	Personal	Dólares Mensual
1	Administrador	400.00
1	Secretaria	300.00
Gasto Personal (GP)		700.00

Cantidad	Equipo de Computo	Valor	Depreciación (1.67%)
1	Servidor	4000 USD	66.80
1	Computador	700 USD	11.69
Gasto Equipo de Cómputo (GE)			78.49

Materiales Directos	Dólares Mensual
Materiales de Oficina	50.00
Gasto Materiales Directos (GMD)	50.00

Varios	Dólares Mensual
Mantenimiento de Computadores	10.00
Útiles de Limpieza	10.00
Gasto Varios (GV)	20.00

COSTO DEL SISTEMA ACTUAL	
GASTOS	Dólares Mensual
Gasto Personal (GP)	700.00
Gasto Equipo de Cómputo (GE)	78.49
Gasto Materiales Directos (GMD)	50.00
Gasto Varios (GV)	20.00
SUBTOTAL	848.49
Gastos Indirectos (0.67%)(SUBTOTAL)	5.68
TOTAL	854.17

SON OCHOCIENTOS CINCUENTA Y CUATRO DOLARES CON DIECISIETE CENTAVOS AMERICANOS

BENEFICIOS

La variación en cuanto al costo del sistema actual y del sistema nuevo es CINCUENTA DOLARES CON DOS CENTAVOS AMERICANOS, la cantidad de ahorro en dólares no es muy considerable, pero el verdadero beneficio se encuentra en la facilidad de manejo de la aplicación y facilidad de cobro de facturas que pueden ser pagadas en cualquier agencia de EMAPA, y no solo en la matriz como se lo venia manejando

4.1.2.2. Factibilidad Técnica

Software

Mediante la siguiente comparación podemos hacer la elección más idónea en cuanto se refiere al motor de base de datos y lenguaje de programación a utilizarse en el desarrollo del presente sistema.

Base de Datos

Parámetros	Access	SQL server	Oracle
Menor Costo	X		
Conocimientos	X	X	X
Mayor Seguridad		X	X
Menor Espacio en Disco	X		
Mayor Capacidad de Almacenamiento			X
Control de Concurrencia		X	X
Licencias	X	X	
Preferencia de la empresa		X	

Lenguaje de Programación

Parámetros	Delphi	Visual Studio .NET
Conocimientos	X	X
Compatibilidad con B.D.	X	X
Licencias		X
Preferencia de la empresa		X

Realizando la comparación hemos optado por Visual Studio .NET; específicamente se trabajara con ASP .NET con el entorno de Visual Basic .NET como Lenguaje de Programación y SQL Server 2000 como motor de Base de Datos.

Hardware

Servidor de Base de Datos

- Servidor Proliant Hp DL380 G4
- Doble Procesador XEON 3.6 GHZ
- 1MB Caché
- 2GB de Memoria RAM

Maquina Cliente

- Intel P4
- Procesador de 3,2
- Disco Duro de 80GB
- 512Mb de memoria RAM

4.1.2.3. Factibilidad Operacional

La Empresa Municipal de Agua Potable y Alcantarillado de Ambato esta dispuesta a brindar una amplia apertura informativa con respecto a las actividades que se realizan en la misma.

La persona designada a prestarnos el servicio de información, es el Ingeniero Fabián Poveda Jefe del centro de cómputo de la empresa, además del apoyo de los otros miembros del departamento

4.1.3. Parámetros de Desarrollo

Equipo Humano

- Investigador: Ricardo Rodrigo Rivera Acurio
- Tutor : Ing. Teresa Freire

Equipo Hardware

La maquina en las que desarrollará el proyecto tienen las siguientes características:

- Procesador Mobile AMD Sempron 3000+
- 512 MB de Memoria RAM.
- Disco Duro de 60 GB
- Unidad de CD-ROM
- Tarjeta de Red
- 3 Puertos USB

Software:

ASP .NET

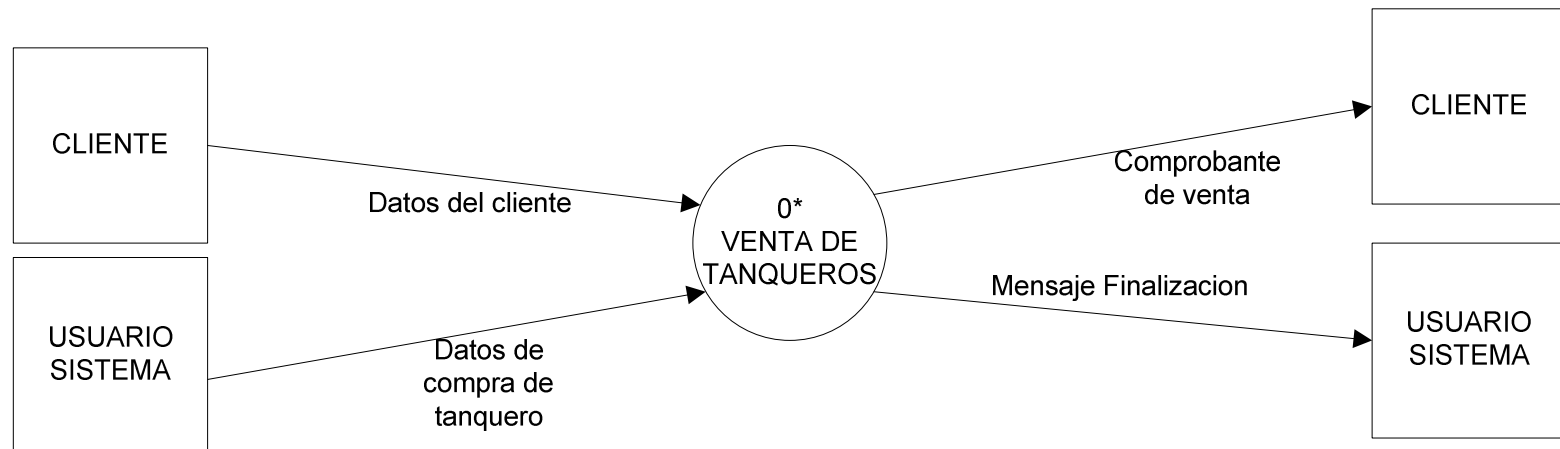
VISUAL BASIC .NET

SQL SERVER 2000

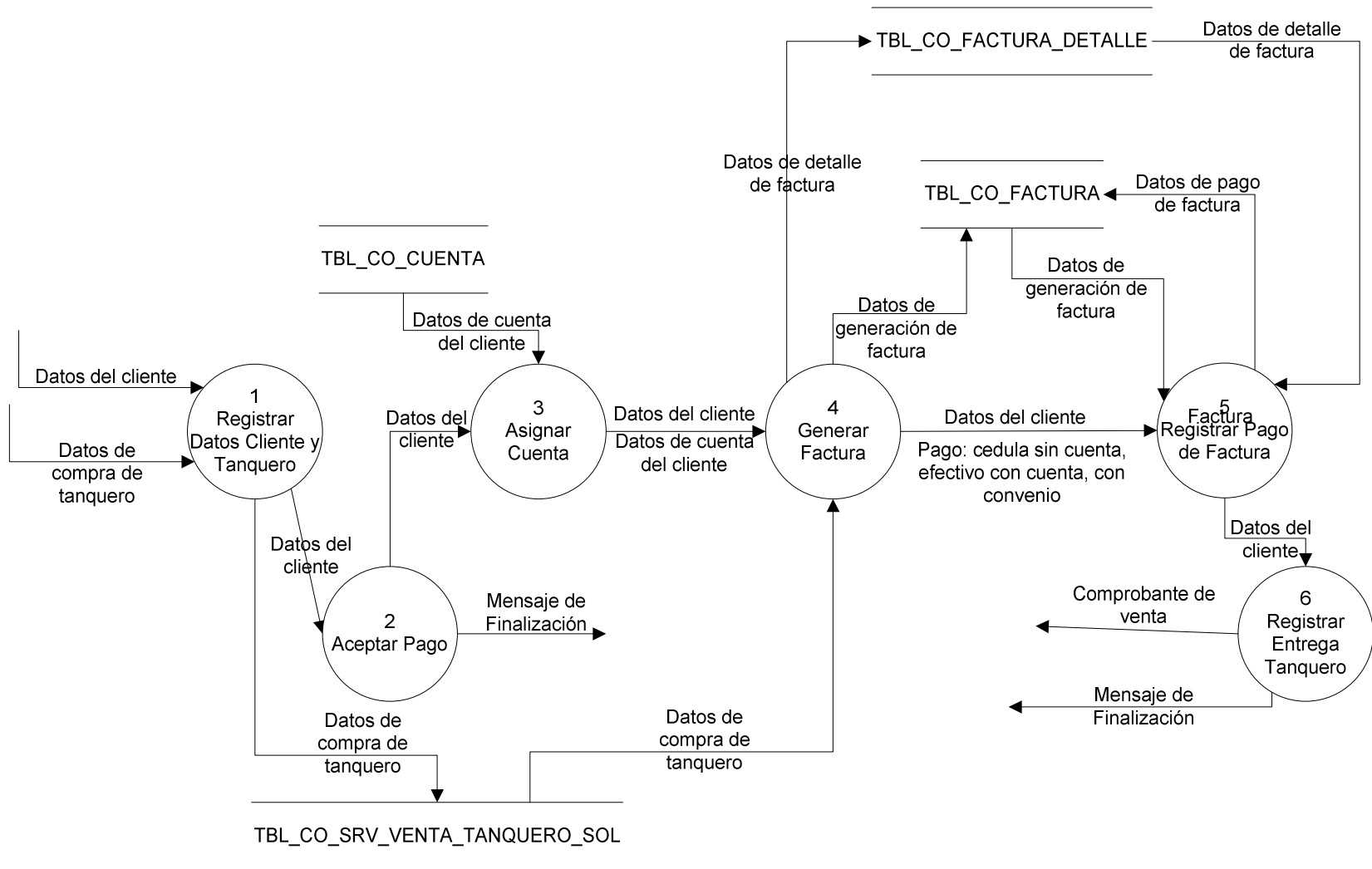
4.2. DIAGRAMA DE FLUJOS DE DATOS

PROCESO DE VENTA DE TANQUEROS

NIVEL 0

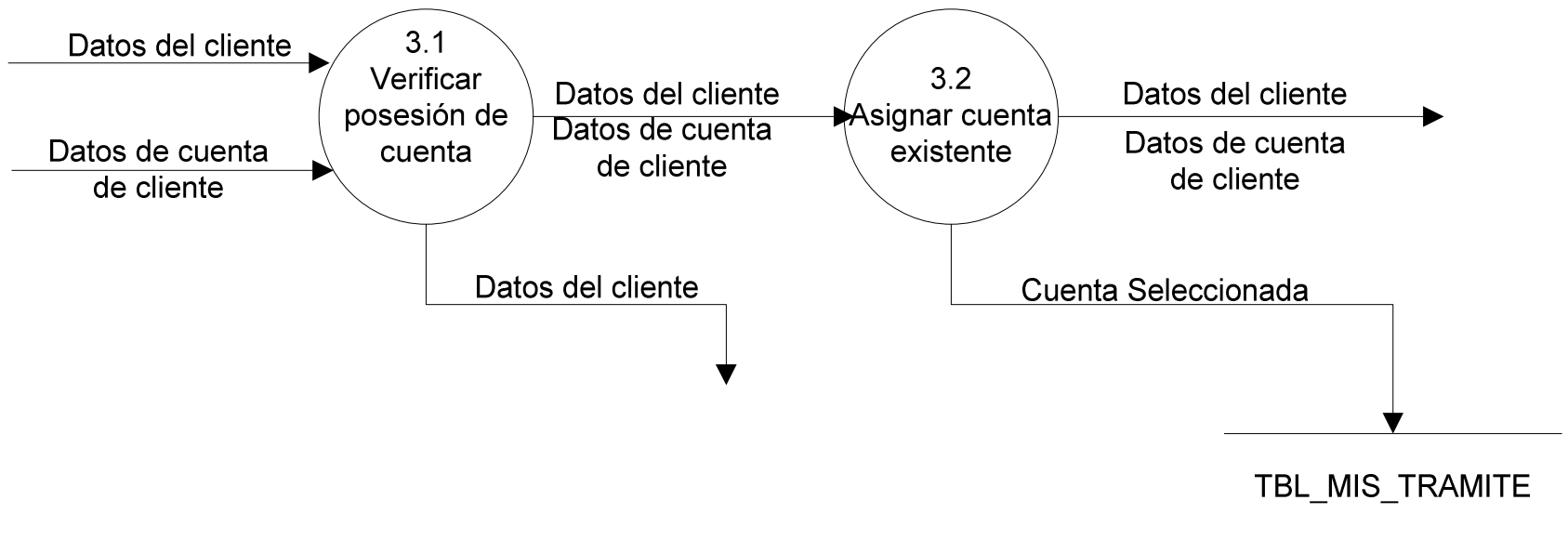


NIVEL 1

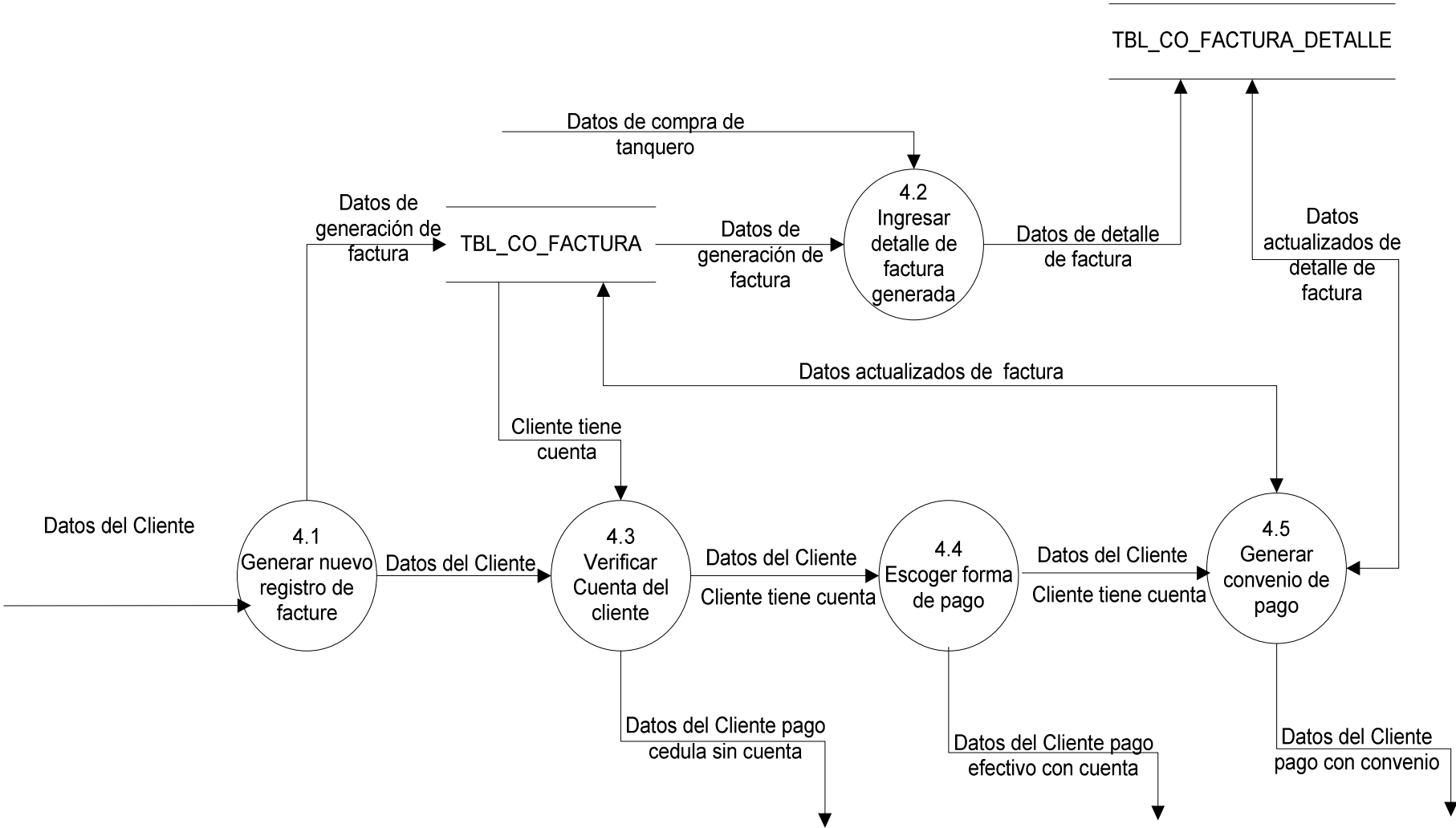


NIVEL 2

ASIGNAR CUENTA



GENERAR FACTURA



4.3. DICCIONARIO DE DATOS

ENTIDADES

Entidad	CLIENTE
Descripción	Persona que llega a solicitar el servicio.
Flujos Proporcionados	Datos de Cliente.
Flujos de Entrada	Comprobante de venta.

Entidad	USUARIO SISTEMA
Descripción	Persona que esta a cargo de realizar la venta de tanqueros en el departamento comercial.
Flujos Proporcionados	Datos de Compra de Tanquero
Flujos de Entrada	Mensaje de Finalización

PROCESOS

Proceso	VENTA DE TANQUEROS (0)
Descripción	Tener un mejor control en las actividades que se realizan dentro del departamento comercial en lo referente a venta de tanqueros.
Entradas	Datos de Cliente, Datos de Compra de Tanquero.
Salidas	Comprobante de venta, Mensaje de Finalización.
Resumen	Manejar de mejor manera la venta de tanqueros.

ESTRUCTURA DE DATOS

Estructura	DATOS CLIENTE
Descripción	Contiene información de los clientes que solicitan el

	servicio.
Origen	Cliente
Destino	Venta de Tanqueros
Estructura	Cedula Cliente, Nombres Cliente, Apellidos Cliente

Estructura	DATOS DE COMPRA DE TANQUERO
Descripción	Contiene información referente a la venta de tanquero
Origen	Usuario sistema
Destino	Venta de Tanqueros
Estructura	Metros Cúbicos, Numero de Tanqueros, Tipo de tanquero, Dirección de Entrega, Fecha de Entrega, Tipo de Transporte, Descuento

ELEMENTOS DATO

Elemento	CEDULA CLIENTE
Descripción	Contiene el numero de cedula de identidad o ruc del cliente que solicitan el servicio.
Tipo	Varchar
Longitud	15
Alias	
Rango Valores	
Limite Valores	

Elemento	NOMBRES CLIENTE
Descripción	Contiene los nombres del cliente que solicitan el servicio.
Tipo	Varchar
Longitud	30
Alias	
Rango Valores	
Limite Valores	

Elemento	APELLIDOS CLIENTE
Descripción	Contiene los apellidos del cliente que solicitan el servicio.
Tipo	Varchar
Longitud	30
Alias	
Rango Valores	
Limite Valores	

Elemento	METROS CUBICOS
Descripción	Contiene la cantidad de agua solicitada dada en metros cúbicos.
Tipo	Float
Longitud	8
Alias	
Rango Valores	
Limite Valores	

Elemento	NUMERO DE TANQUEROS
Descripción	Contiene el número de tanqueros solicitado por el cliente.
Tipo	Integer

Longitud	4
Alias	
Rango Valores	
Limite Valores	

Elemento	TIPO DE TANQUERO
Descripción	Contiene información referente al tipo de compra del tanquero.
Tipo	Numeric
Longitud	9
Alias	
Rango Valores	
Limite Valores	

Elemento	DIRECCIÓN DE ENTREGA
Descripción	Contiene la información del lugar en el que se debe entregar el tanquero de agua.
Tipo	Varchar
Longitud	255
Alias	
Rango Valores	
Limite Valores	

Elemento	FECHA DE ENTREGA
Descripción	Contiene la fecha en la que se debe entregar el tanquero de agua.
Tipo	Datetime
Longitud	8
Alias	
Rango Valores	
Limite Valores	

Elemento	TIPO DE TRANSPORTE
Descripción	Contiene la información del tipo de transporte que realizara la entrega.
Tipo	Smallint
Longitud	2
Alias	
Rango Valores	
Limite Valores	

Elemento	DESCUENTO
Descripción	Contiene el descuento que se le puede otorgar al cliente.
Tipo	Varchar
Longitud	3
Alias	
Rango Valores	
Limite Valores	

NIVEL 1

PROCESOS

Proceso	REGISTRAR DATOS CLIENTE Y TANQUERO (1)
Descripción	Guarda información referente al cliente y al pedido del tanquero de agua que realiza.
Entradas	Datos de Cliente, Datos de Compra de Tanquero.
Salidas	Datos de Cliente, Datos de Compra de Tanquero.
Resumen	Guarda información del cliente y el pedido del tanquero

Proceso	ACEPTAR PAGO(2)
Descripción	Registrar que el cliente se encuentra de acuerdo con efectuar el respectivo pago por el servicio solicitado.
Entradas	Datos de Cliente
Salidas	Datos de Cliente, Mensaje de Finalización.
Resumen	Registrar que el cliente esta de acuerdo con el pago.

Proceso	ASIGNAR CUENTA (3)
Descripción	Registra el numero de cuenta al que se cargara el valor de la factura en caso de poseerla, si el cliente no tiene cuenta puede seguir el tramite pero debe cancelar en efectivo.
Entradas	Datos de Cliente, Datos de Cuenta del Cliente.
Salidas	Datos de Cliente, Datos de Cuenta del Cliente.
Resumen	Registrar la cuenta a la que se realizara el valor del pedido.

Proceso	GENERAR FACTURA (4)
Descripción	Realiza los respectivos cálculos por el pedido y presenta en forma detallada la cantidad a pagarse.
Entradas	Datos de Cliente, Datos de Cuenta del Cliente, Datos de Compra de Tanquero.
Salidas	Datos de Detalle de Factura, Datos de Generación de Factura, Datos del cliente Pago: cedula sin cuenta, efectivo con cuenta, con convenio
Resumen	Realiza los cálculos para el valor a pagar.

Proceso	REGISTRAR PAGO FACTURA (5)
Descripción	Guardar datos del pago de la factura del servicio prestado.
Entradas	Datos del cliente Pago: cedula sin cuenta, efectivo

	con cuenta, con convenio; Datos de Generación de Factura, Datos de Detalle de Factura.
Salidas	Datos de pago de Factura, Datos de Cliente
Resumen	Guardar los datos del pago de la factura.

Proceso	REGISTRAR ENTREGA TANQUERO (6)
Descripción	Guardar los datos referentes a la entrega del tanquero de agua.
Entradas	Datos de Cliente.
Salidas	Comprobante de venta, Mensaje de Finalización.
Resumen	Guardar datos de la entrega del tanquero.

ESTRUCTURA DE DATOS

Estructura	DATOS DE DETALLE DE FACTURA
Descripción	Contiene información detallada de los valores que se generan por la respectiva factura.
Origen	Generar Factura
Destino	Registrar Pago de Factura
Estructura	Numero de factura, Valor Total, Cantidad, Descripción, Valor Unitario.

Estructura	DATOS DE GENERACION DE FACTURA
Descripción	Contiene información de la factura que se genero por la venta del servicio.
Origen	Generar Factura
Destino	Registrar Pago de Factura

Estructura	Numero de Factura, Valor Total, Numero de Cuenta, Fecha de Facturación, Cedula de Cliente.
-------------------	--

Estructura	DATOS DE PAGO DE FACTURA
Descripción	Contiene información del pago de la respectiva factura.
Origen	Registrar Pago de Factura
Destino	Almacén TBL_CO_FACTURA
Estructura	Recaudador, Estado, Pagada o No Pagada, Factura Contabilizada.

Estructura	DATOS DE CUENTA DE CLIENTE
Descripción	Contiene información de la o las cuentas que posea el cliente.
Origen	Almacén TBL_CO_CUENTA
Destino	Generar Factura
Estructura	Cedula Cliente, Numero de cuenta.

ELEMENTOS DATO

Elemento	NUMERO DE FACTURA
Descripción	Contiene el número único de factura a pagarse por el servicio otorgado.
Tipo	Bigint

Longitud	8
Alias	
Rango Valores	
Limite Valores	

Elemento	VALOR TOTAL
Descripción	Contiene el Valor Total a cancelarse por el servicio.
Tipo	Float
Longitud	8
Alias	
Rango Valores	
Limite Valores	

Elemento	CANTIDAD
Descripción	Contiene la cantidad en metros cúbicos a entregarse al cliente.
Tipo	Float
Longitud	8
Alias	
Rango Valores	
Limite Valores	

Elemento	DESCRIPCIÓN
Descripción	Contiene una breve descripción del servicio que se esta dando al cliente.
Tipo	Varchar
Longitud	50
Alias	
Rango Valores	
Limite Valores	

Elemento	VALOR UNITARIO.
Descripción	Contiene el valor de la unidad del servicio solicitado por el cliente.
Tipo	Float
Longitud	8
Alias	
Rango Valores	
Limite Valores	

Elemento	NUMERO DE CUENTA
Descripción	Contiene el número de cuenta que posee el cliente para el respectivo trámite que realiza.
Tipo	Numeric
Longitud	9
Alias	
Rango Valores	
Limite Valores	

Elemento	FECHA DE FACTURACIÓN
Descripción	Contiene la fecha en la que se genero la factura.
Tipo	Smalldate
Longitud	4
Alias	
Rango Valores	
Limite Valores	

Elemento	RECAUDADOR
Descripción	Contiene la información referente a la persona que realiza el cobro de la factura.
Tipo	Varchar
Longitud	50

Alias	
Rango Valores	
Limite Valores	

Elemento	ESTADO
Descripción	Contiene el valor del estado en la que se encuentra la factura.
Tipo	Varchar
Longitud	2
Alias	
Rango Valores	
Limite Valores	

Elemento	PAGADA O NO PAGADA
Descripción	Contiene la información de si la factura ya ha sido pagada o no.
Tipo	Bit
Longitud	1
Alias	
Rango Valores	
Limite Valores	

Elemento	FACTURA CONTABILIZADA
Descripción	Nos indica si la factura ya ha sido o no contabilizada para el respectivo cierre de caja.
Tipo	Bit
Longitud	1
Alias	
Rango Valores	
Limite Valores	

ALMACEN DE DATOS

Almacén	TBL_CO_CUENTA
Descripción	Guarda información referente a las cuentas que poseen los clientes
Flujo Entrada	
Flujo Proporcionado	Datos de Cuenta de Cliente
Descripción de Datos	Numero de Cuenta Cedula de Cliente
Acceso	Diario

Almacén	TBL_CO_SRV_VENTA_TANQUERO_SOL
Descripción	Guarda información referente al pedido que realiza el cliente a la empresa en lo concerniente a venta de tanqueros de agua.
Flujo Entrada	Datos de compra de tanquero
Flujo Proporcionado	Datos de compra de tanquero
Descripción de Datos	Metros Cúbicos, Numero de Tanqueros, Tipo de tanquero, Dirección de Entrega, Fecha de Entrega, Tipo de Transporte, Descuento
Acceso	Diario

Almacén	TBL_CO_FACTURA
Descripción	Guarda información referente a todas las facturas generadas en la empresa por los diferentes servicios que presta.
Flujo Entrada	Datos de generación de factura, datos de pago de factura

Flujo Proporcionado	Datos de generación de factura
Descripción de Datos	Numero de Factura, Valor Total, Numero de Cuenta, Fecha de Facturación, Cedula de Cliente, Recaudador, Estado, Pagada o No Pagada, Factura Contabilizada.
Acceso	Diario

Almacén	TBL_CO_FACTURA_DETALLE
Descripción	Guarda información referente al detalle de todas las facturas que se han generado en la empresa por los diferentes servicios que presta.
Flujo Entrada	Datos de detalle de factura
Flujo Proporcionado	Datos de detalle de factura
Descripción de Datos	Numero de factura, Valor Total, Cantidad, Descripción, Valor Unitario.
Acceso	Diario

NIVEL 2

PROCESOS

Proceso	VERIFICAR POSESION DE CUENTA (3.1)
Descripción	Verifica si el cliente tiene o no cuentas asignar para

	la realización de trámites.
Entradas	Datos de Cliente, Datos de Cuenta de Cliente.
Salidas	Datos de Cliente, Datos de Cuenta de Cliente.
Resumen	Verifica si el cliente tiene cuenta

Proceso	ASIGNAR CUENTA EXISTENTE (3.2)
Descripción	Asigna un numero de cuenta en caso de poseer una o varias cuenta, a la cual deber'a cargarse el respectivo valor a cancelar.
Entradas	Datos de Cliente, Datos de Cuenta de Cliente.
Salidas	Datos de Cliente, Datos de Cuenta de Cliente, Cuenta Seleccionada.
Resumen	Asigna un número de cuanta para pago de factura.

Proceso	GENERAR NUEVO REGISTRO DE FACTURA (4.1)
Descripción	Genera un nuevo número de factura en el que se asignaran los datos referentes a la misma.
Entradas	Datos de Cliente, Datos de Cuenta de Cliente.
Salidas	Datos de Cliente, Datos de Generación de Factura.
Resumen	Genera un nuevo número de factura.

Proceso	INGRESAR DETALLE DE FACTURA GENERADA (4.2)
Descripción	Registra la información detallada correspondiente al número de factura generado.
Entradas	Datos de Generación de Factura, Datos de compra de Tanquero
Salidas	Datos de Detalle de Factura
Resumen	Registra el detalle de la factura

Proceso	VERIFICAR CUENTA DEL CLIENTE (4.3)
Descripción	Verifica si el cliente tiene o no una cuenta asignada para el respectivo tramite.
Entradas	Datos de Cliente, Cliente Tiene Cuenta.
Salidas	Datos de Cliente Pago en Efectivo sin Cuenta, Cliente Tiene Cuenta.
Resumen	Verifica si el cliente tiene asignada cuenta.

Proceso	ESCOGER FORMA DE PAGO (4.4)
Descripción	Registra la forma de pago escogida por el cliente para cancelar el valor de la factura.
Entradas	Datos de Cliente, Cliente Tiene Cuenta.
Salidas	Datos de Cliente Pago en Efectivo, Cliente Tiene Cuenta.
Resumen	Registra la forma de pago escogida por el cliente

Proceso	GENERAR CONVENIO DE PAGO (4.5)
Descripción	Genera el convenio a los meses que haya escogido el cliente para la cancelación de la factura.
Entradas	Datos de Cliente, Cliente Tiene Cuenta, Datos detalle de Factura
Salidas	Datos de Cliente Pago con Convenio
Resumen	Guarda información del cliente y el pedido del tanquero

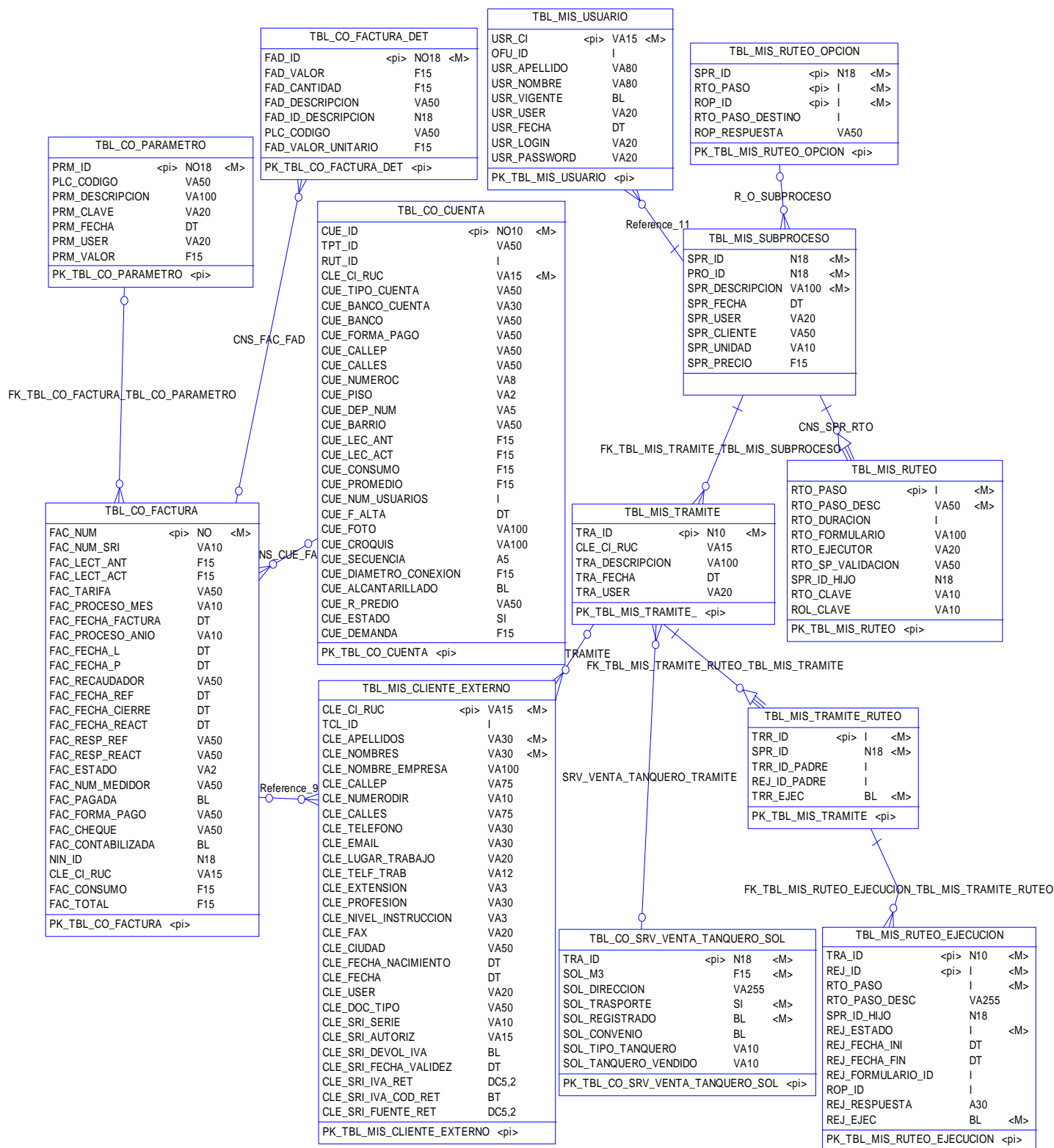
ALMACEN DE DATOS

Almacén	TBL_MIS_TRAMITE
Descripción	Guarda información referente al tramite que esta realizando la empresa de acuerdo al servicio solicitado por el cliente

Flujo Entrada	Cuenta Seleccionada
Flujo Proporcionado	
Descripción de Datos	Numero de Cuenta Numero de Tramite
Acceso	Diario

4.4. DISEÑO DE LA BASE DE DATOS

4.4.1. Diseño Lógico



4.5. DISEÑO DE INTERFACES

4.5.1. Diseño de Entradas

SOLICITUD DE VENTA DE AGUA EN TANQUERO

Tipos de Entrada

Mouse para selección de datos

Teclado para ingreso de información

Características del Diseño de Entrada

Validaciones por registro

Metros Cúbicos

Fecha de Entrega

Verificación de registros

Verificar que se ingresen solamente números en metros cúbicos,

Verificar que la fecha ingresada no sea una fecha anterior a la actual

Verificar que el número de tanqueros vendidos en el día y/o en el mes, no excedan a límite establecido

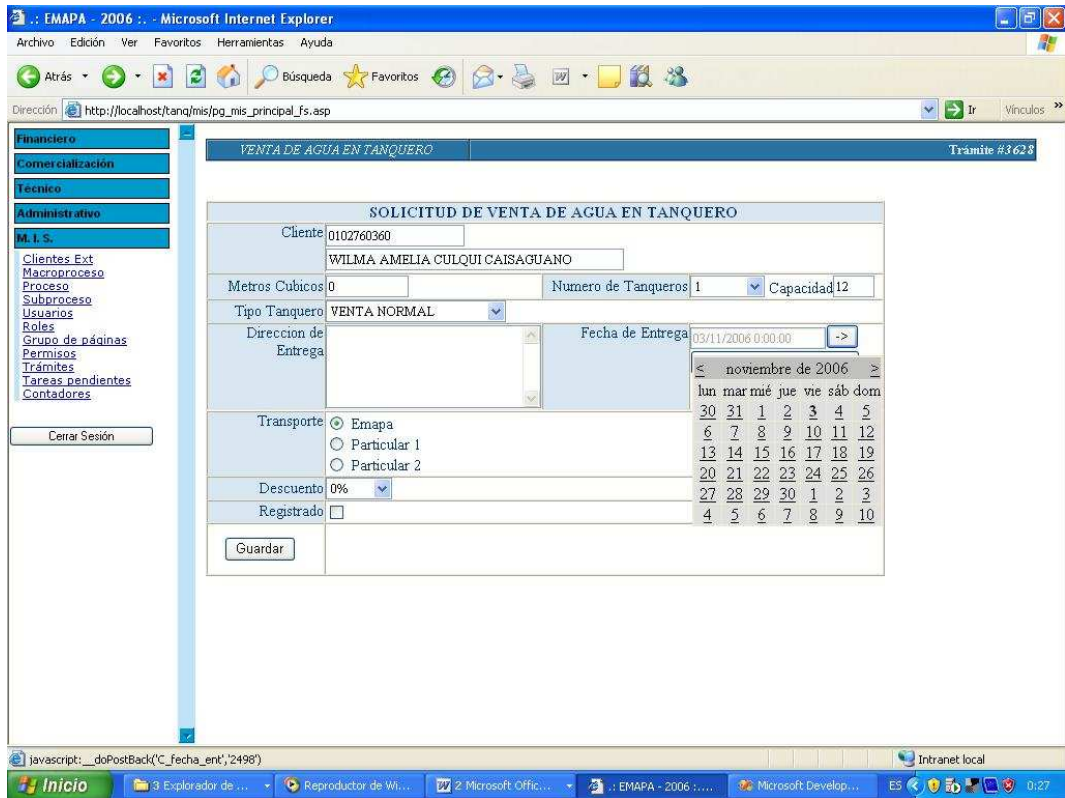
Uso de Colores

Blanco para identificar el título,

Azul para identificar etiqueta,

Negro para identificar cajas de texto, botones y drop down list box

Disposición de Información



Diseño de Mensajes

Mensajes de Error: Se producirán cuando:

La fecha ingresada sea anterior a la fecha actual,

El número de turnos diarios de tanqueros exceda al límite establecido

Objetivo

Proporcionar una ventana de fácil manejo para que el usuario del sistema pueda ingresar sin problemas un nuevo registro de venta de tanqueros.

ACEPTAR PAGO

Tipos de Entrada

Mouse para selección de datos

Características del Diseño de Entrada

Validaciones por registro

Ninguna

Verificación de registros

Ninguna

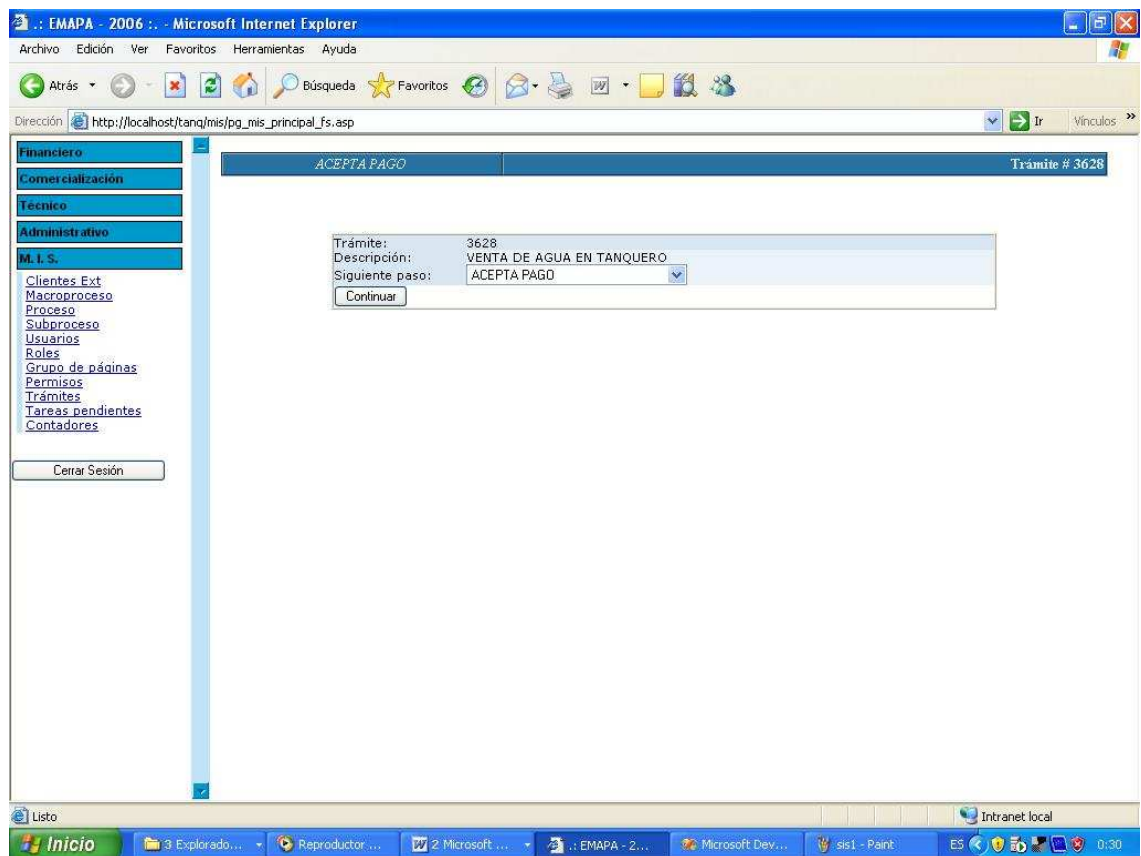
Uso de Colores

Blanco para identificar el título,

Azul para identificar etiqueta,

Negro para identificar cajas de texto, botones y drop down list box

Disposición de Información



Diseño de Mensajes

Ninguno

Objetivo

Aceptar el pago del respectivo tramite

ASIGNAR CUENTA

Tipos de Entrada

Mouse para selección de datos

Características del Diseño de Entrada

Validaciones por registro

Cuenta

Verificación de registros

Verificar que el usuario tenga asignado un número de cuenta

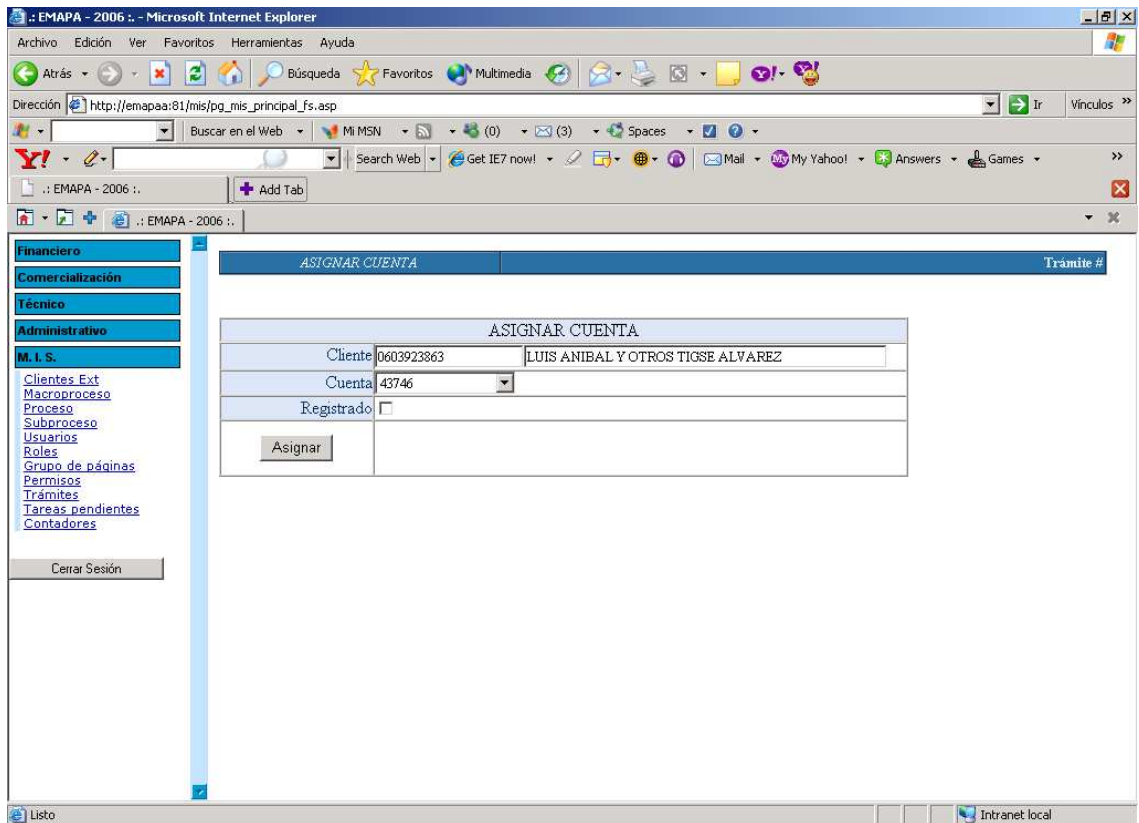
Uso de Colores

Blanco para identificar el título,

Azul para identificar etiqueta,

Negro para identificar cajas de texto, botones y drop down list box

Disposición de Información



Diseño de Mensajes

Mensajes de Información: Se producirán cuando:
El Usuario no tenga una cuenta asignada.

Objetivo

Asignar una cuenta en caso de existir al trámite que se realiza

FACTURA

Tipos de Entrada

Mouse para selección de datos

Características del Diseño de Entrada

Validaciones por registro

Forma de Pago

Verificación de registros

Verificar si la forma de pago es efectivo o crédito

Uso de Colores

Blanco para identificar el título,

Azul para identificar etiqueta,

Negro para identificar cajas de texto, botones y drop down list box

Disposición de Información

The screenshot shows a Microsoft Internet Explorer browser window displaying a web application. The address bar shows the URL: `http://emapaa:81/mis/pg_mis_principal_fs.asp`. The page title is "EMAPA - 2006".

The interface is divided into a left sidebar and a main content area. The sidebar contains a navigation menu with the following items:

- Financiero
- Comercialización
- Técnico
- Administrativo
- M. I. S.
- Clientes Ext
- Macroproceso
- Proceso
- Subproceso
- Usuarios
- Roles
- Grupo de páginas
- Permisos
- Trámites
- Tareas pendientes
- Contadores

The main content area is titled "VENTA DE AGUA EN TANQUERO" and "Trámite #4202". It contains a form titled "FACTURA" with the following fields:

Numero de Factura	1388435	Fecha Facturación	07/11/2006 9:21:00
Cliente Externo	0603923863	LUIS ANIBAL Y OTROS TIGSE ALVAREZ	
CI/RUC			
Pagada	<input type="checkbox"/>		
Forma de Pago	Efectivo		

Below the form is a "Guardar" button. Underneath is a section titled "DETALLE DE FACTURA" and "Trámite # 4202", which contains a table with the following data:

Id	Servicio	Cantidad	Valor Unitario	IVA	Valor Total
4721008	Venta de agua en tanquero	24	0.45		10.8
4721009	Transporte en tanquero	2	10		20

The status bar at the bottom of the browser window shows "Listo" and "Intranet local".

Diseño de Mensajes

Ninguno

Objetivo

Presentar y registrar de una forma clara los valores a cancelar en la factura y su forma de pago

CONVENIO FACTURA

Tipos de Entrada

Mouse para selección de datos

Teclado para ingreso de información

Características del Diseño de Entrada

Validaciones por registro

Ninguna

Verificación de registros

Ninguna

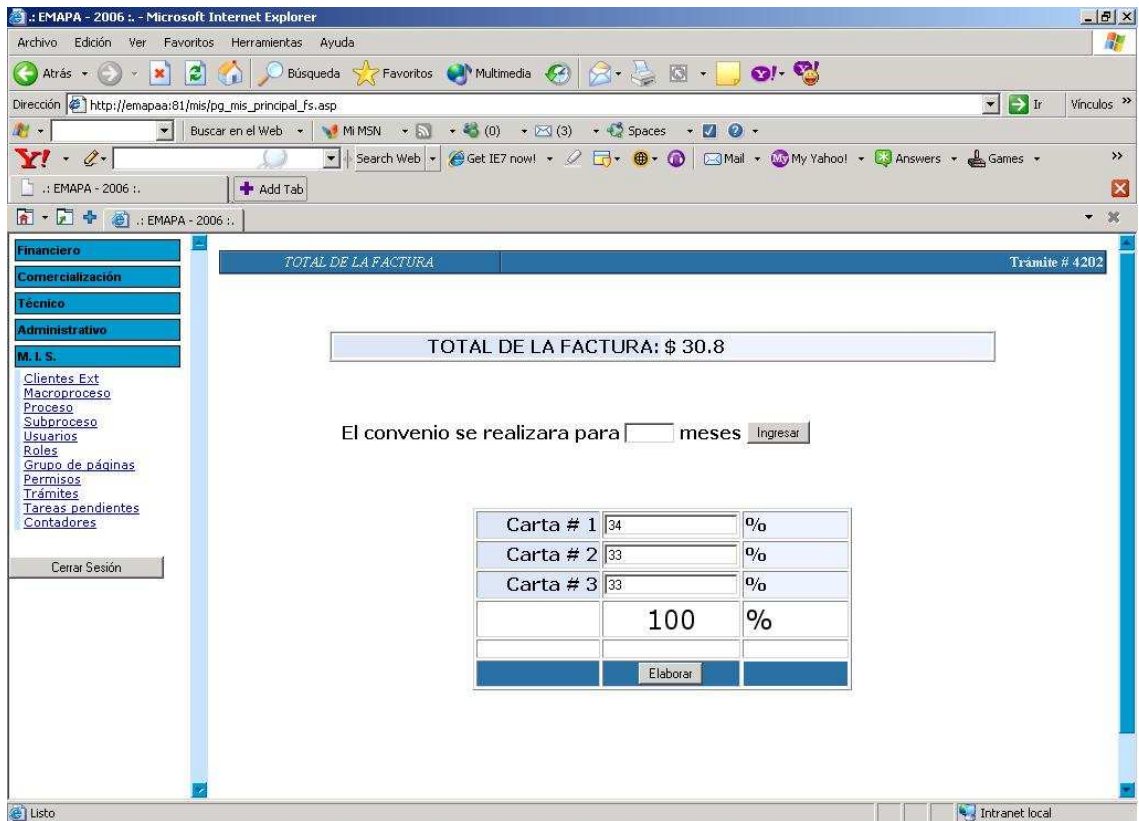
Uso de Colores

Blanco para identificar el título,

Azul para identificar etiqueta,

Negro para identificar cajas de texto, botones y drop down list box

Disposición de Información



Diseño de Mensajes

Mensajes de Información: Se producirán cuando:

El usuario elabore el convenio.

Objetivo

Realizar un convenio de pago entre la empresa y el cliente

ENTREGA DE AGUA

Tipos de Entrada

Mouse para selección de datos

Características del Diseño de Entrada

Validaciones por registro

Ninguna

Verificación de registros

Ninguna

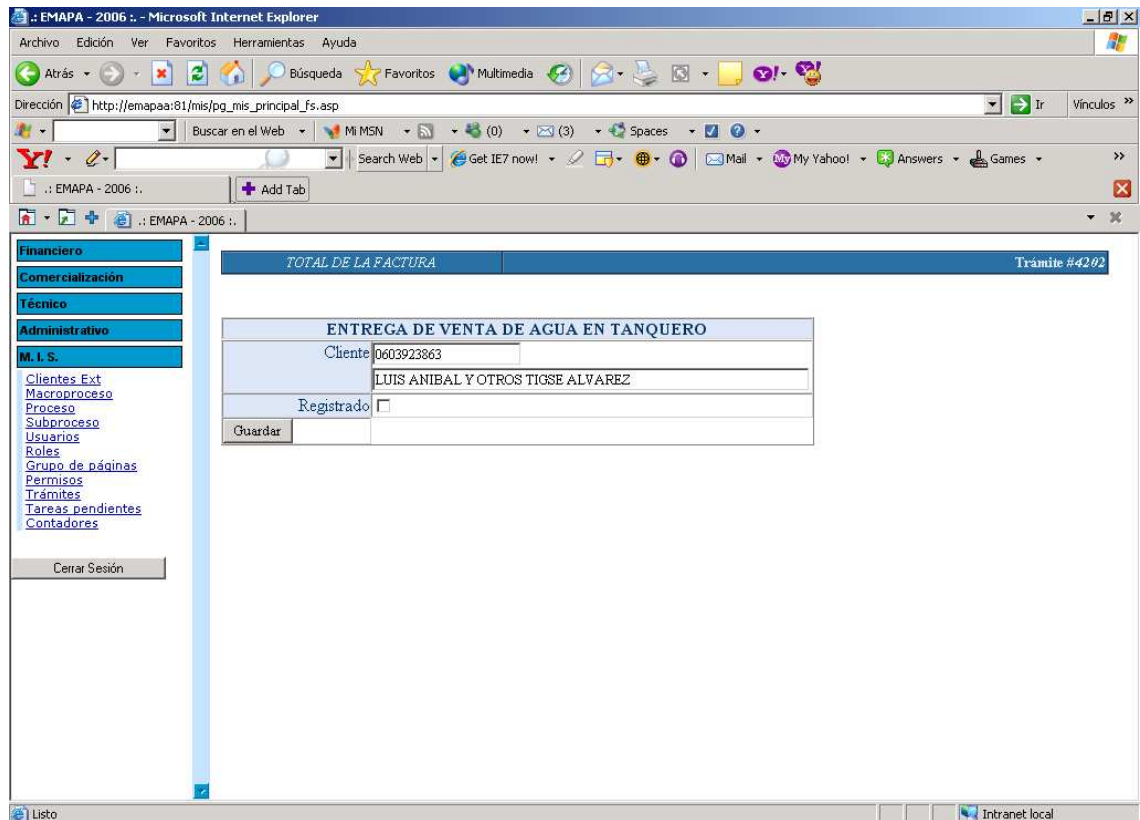
Uso de Colores

Blanco para identificar el título,

Azul para identificar etiqueta,

Negro para identificar cajas de texto y botones

Disposición de Información



Diseño de Mensajes

Ninguno

Objetivo

Registrar la entrega de agua al cliente

4.5.2. Diseño de Salidas

Características del Diseño de salida

Tipos de Letras

Para todo el documento se utilizara el tipo de letra Times New Roman (Occidental)

Tamaños

Para todo el documento el tamaño de la letra será de 10

Colores

Para los títulos el color de la letra será blanco y para el resto del documento será de color azul

Negritas

Se utilizara negrita para los títulos

Detalles

EMAPA - 2006 :: Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://emapaa:81/mis/pg_mis_principal_fs.asp

Financiero

Comercialización

Técnico

Administrativo

M. I. S.

Cientes Ext

Macroproceso

Proceso

Subproceso

Usuarios

Roles

Grupo de páginas

Permisos

Trámites

Tareas pendientes

Contadores

Cerrar Sesión

VENTA DE AGUA EN TANQUERO

REPORTE DE VENTAS

Mes: Enero Año: 2004 Ver

FECHA DE ENTREGA	No.	DESCRIPCION	No.	TRAMITE CEDULA	NOMBRES	APELLIDOS	DIRECCION	DESCRIPCION	No.
		DONADO	2		CÉSAR POMPILLO Y SRA.	CHIRIBOGA ASTUDILLO		TECHO PROPIO	3
Nov 6 2006	8	TECHO PROPIO	3	4227	0100480524				
Nov 7 2006	7	VENTA NORMAL	6						
Nov 8 2006	2			4229	1100555083	LUZ VICTORIA	SANDOVAL SANDOVAL	VENTA NORMAL	3
Nov 9 2006	5			4236	1100555083	LUZ VICTORIA	SANDOVAL SANDOVAL	VENTA NORMAL	1
Nov 10 2006	3								
Nov 13 2006	1			4237	0502226038	FLOR CARMELINA	VASQUEZ TARCO	VENTA NORMAL	1
Nov 14 2006	2								
Nov 15 2006	3			4238	0502226038	FLOR CARMELINA	VASQUEZ TARCO	DONADO	1
Nov 16 2006	3								
Nov 17 2006	11			4240	0182001022277	CARLOS RIGOBERTO	FAICAN LANDI	VENTA NORMAL	1
Nov 18 2006	3								
Nov 21 2006	1			4243	0182001022277	CARLOS RIGOBERTO	FAICAN LANDI	DONADO	1
Nov 23 2006	2								
Nov 24 2006	1								
Nov 25 2006	5								

Intranet local

CAPITULO V

IMPLEMENTACION Y PRUEBAS

5.1. CODIFICACIÓN

A continuación detallaremos los procesos más importantes que se utilizaron para la implementación de la aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA.

CLASES

Clase utilizada para la cadena de conexión

```
Imports System.Data.SqlClient
Public Class Clase_inicio
    Public Shared conexion As String =
        System.Configuration.ConfigurationSettings.AppSettings("Cadena")
End Class
```

Clase utilizada para la ejecución de consultas SQL

```
Imports System.Data.SqlClient
Public Class Clase_sql
    Private con As SqlConnection
    Private cmdSql As SqlCommand
    Private msg As String

    #Region "Selecciones"
```

```
Public Function getItemstramite(ByVal tra_id As String, ByVal conexion As String) As DataSet
```

```
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select * FROM TBL_MIS_TRAMITE WHERE
    TRA_ID =@tra_id"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@tra_id",
    SqlDbType.VarChar)).Value = tra_id
    Dim da1 As New SqlDataAdapter(cmdSql)
    Dim ds1 As New DataSet
    Try
        con.Open()
        da1.Fill(ds1, "datos_tramite")
        Return ds1
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Function
```

```
Public Function getItemparametro(ByVal conexion As String) As DataSet
```

```
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select * FROM TBL_CO_PARAMETRO WHERE
    PRM_CLAVE ='VENTA TANQUERO'"
    Dim cmdSql As New SqlCommand(sql, con)
    Dim da1 As New SqlDataAdapter(cmdSql)
    Dim ds1 As New DataSet
    Try
        con.Open()
        da1.Fill(ds1, "datos_parametro")
        Return ds1
    End Try
End Function
```

```

Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

```

```

Public Function getItemsparametro(ByVal tipo As String, ByVal conexion As
String) As DataSet

```

```

    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select * FROM TBL_CO_PARAMETRO WHERE
PRM_DESCRIPCION =@tipo"
    Dim cmdSql As New SqlCommand(sql, con)
    Dim da1 As New SqlDataAdapter(cmdSql)
    cmdSql.Parameters.Add(New SqlParameter("@tipo",
SqlDbType.VarChar)).Value = tipo
    Dim ds1 As New DataSet
    Try
        con.Open()
        da1.Fill(ds1, "datos_parametro")
        Return ds1
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Function

```

```

Public Function getItemsruteo(ByVal tra_id As String, ByVal rej_id As String,
ByVal conexion As String) As DataSet
    Dim con As New SqlConnection(conexion)

```

```

Dim sql As String = "select * FROM TBL_MIS_RUTEO_EJECUCION
WHERE TRA_ID =@tra_id and rej_id=@rej_id"
Dim cmdSql As New SqlCommand(sql, con)
cmdSql.Parameters.Add(New SqlParameter("@tra_id",
SqlDbType.VarChar)).Value = tra_id
cmdSql.Parameters.Add(New SqlParameter("@rej_id",
SqlDbType.VarChar)).Value = rej_id
Dim da1 As New SqlDataAdapter(cmdSql)
Dim ds1 As New DataSet
Try
    con.Open()
    da1.Fill(ds1, "datos_ruteo")
    Return ds1
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

```

```

Public Function getItemstanquero(ByVal conexion As String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select * from
TBL_CO_SRV_VENTA_TANQUERO_SOL"
    Dim cmdSql As New SqlCommand(sql, con)
    Dim da1 As New SqlDataAdapter(cmdSql)
    Dim ds1 As New DataSet
    Try
        con.Open()
        da1.Fill(ds1, "datos_tanquero")
        Return ds1
    Catch ex As SqlClient.SqlException

```

```

        msg = ex.Message
    Finally
        con.Close()
    End Try
End Function

```

```

Public Function getItemstanqueromes(ByVal conexion As String, ByVal mes
As String) As DataSet

```

```

    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select sol_fecha_ent as 'FECHA DE ENTREGA' ,
sol_direccion AS 'DIRECCION', sol_tipo_tanquero AS 'TIPO',
sol_num_tanquero AS 'CANTIDAD' from
TBL_CO_SRV_VENTA_TANQUERO_SOL where
month(sol_fecha_ent) = @mes"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@mes",
SqlDbType.VarChar)).Value = mes
    Dim da1 As New SqlDataAdapter(cmdSql)
    Dim ds1 As New DataSet
    Try
        con.Open()
        da1.Fill(ds1, "datos_tanqueromes")
    Return ds1
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Function

```

```

Public Function getItemstanquerodia(ByVal conexion As String, ByVal fecha
As String) As DataSet

```

```

Dim con As New SqlConnection(conexion)
Dim sql As String = "select sol_fecha_ent as 'FECHA DE ENTREGA' ,
sol_direccion AS 'DIRECCION', sol_tipo_tanquero AS 'TIPO',
sol_num_tanquero AS 'CANTIDAD' from
TBL_CO_SRV_VENTA_TANQUERO_SOL where
month(sol_fecha_ent) = @fecha"
Dim cmdSql As New SqlCommand(sql, con)
cmdSql.Parameters.Add(New SqlParameter("@fecha",
SqlDbType.VarChar)).Value = fecha
Dim da1 As New SqlDataAdapter(cmdSql)
Dim ds1 As New DataSet
Try
    con.Open()
    da1.Fill(ds1, "datos_tanqueromes")
Return ds1
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

Public Function getItemstanquero_consulta(ByVal conexion As String, ByVal
fecha As String) As DataSet
Dim con As New SqlConnection(conexion)
Dim sql As String = "SELECT
dbo.TBL_CO_PARAMETRO.PRM_DESCRIPCION,
SUM(dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL.SOL_NUM_TA
NQUERO) AS CANTIDAD FROM
dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL INNER JOIN
dbo.TBL_CO_PARAMETRO ON
dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL.SOL_TIPO_TANQUE

```



```

RO = dbo.TBL_CO_PARAMETRO.PRM_ID WHERE
(dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL.SOL_FECHA_ENT =
@fecha)GROUP BY
dbo.TBL_CO_PARAMETRO.PRM_DESCRIPCION"
Dim cmdSql As New SqlCommand(sql, con)
cmdSql.Parameters.Add(New SqlParameter("@fecha",
SqlDbType.VarChar)).Value = fecha
Dim da1 As New SqlDataAdapter(cmdSql)
Dim ds1 As New DataSet
Try
    con.Open()
    da1.Fill(ds1, "datos_tanqueromes")
    Return ds1
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

```

```

Public Function getItemstanquero_consulta_diaria(ByVal conexion As String,
ByVal fecha As String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "SELECT
dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL.TRA_ID,
dbo.TBL_MIS_TRAMITE.CLE_CI_RUC,
dbo.TBL_MIS_CLIENTE_EXTERNO.CLE_NOMBRES,dbo.TBL_MIS_
CLIENTE_EXTERNO.CLE_APELLIDOS,
dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL.SOL_DIRECCION,dbo
.TBL_CO_SRV_VENTA_TANQUERO_SOL.SOL_NUM_TANQUERO(
) FROM dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL INNER JOIN
dbo.TBL_MIS_TRAMITE ON

```

```

dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL.TRA_ID =
dbo.TBL_MIS_TRAMITE.TRA_ID INNER JOIN
dbo.TBL_MIS_CLIENTE_EXTERNO ON
dbo.TBL_MIS_TRAMITE.CLE_CI_RUC =
dbo.TBL_MIS_CLIENTE_EXTERNO.CLE_CI_RUC WHERE
dbo.TBL_CO_SRV_VENTA_TANQUERO_SOL.SOL_FECHA_ENT =
@fecha"
Dim cmdSql As New SqlCommand(sql, con)
cmdSql.Parameters.Add(New SqlParameter("@fecha",
SqlDbType.VarChar)).Value = fecha
Dim da1 As New SqlDataAdapter(cmdSql)
Dim ds1 As New DataSet
Try
    con.Open()
    da1.Fill(ds1, "datos_tanqueromes")
    Return ds1
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

Public Function getItemstanqueromes_reporte(ByVal conexion As String,
ByVal mes As String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select CAST(SOL_FECHA_ENT AS
VARCHAR(12)) as 'FECHA DE ENTREGA', sum(sol_num_tanquero)
AS 'CANTIDAD', month(sol_fecha_ent) as 'MES' from
TBL_CO_SRV_VENTA_TANQUERO_SOL where
month(sol_fecha_ent) = @mes GROUP BY sol_fecha_ent"
    Dim cmdSql As New SqlCommand(sql, con)

```

```

cmdSql.Parameters.Add(New SqlParameter("@mes",
SqlDbType.VarChar)).Value = mes
Dim da1 As New SqlDataAdapter(cmdSql)
Dim ds1 As New DataSet
Try
    con.Open()
    da1.Fill(ds1, "datos_tanqueromes")
    Return ds1
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

Public Function getItemstanquero(ByVal tra_id As String, ByVal conexion As
String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select * from
TBL_CO_SRV_VENTA_TANQUERO_SOL WHERE TRA_ID
=@tra_id"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@tra_id",
SqlDbType.VarChar)).Value = tra_id
    Dim da1 As New SqlDataAdapter(cmdSql)
    Dim ds1 As New DataSet
    Try
        con.Open()
        da1.Fill(ds1, "datos_tanquero")
        Return ds1
    Catch ex As SqlClient.SqlException
        msg = ex.Message

```

```

    Finally
        con.Close()
    End Try
End Function

Public Function getItemsUsuario(ByVal tra_id As String, ByVal conexion As
String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select TBL_MIS_CLIENTE_EXTERNO.* FROM
TBL_MIS_CLIENTE_EXTERNO INNER JOIN TBL_MIS_TRAMITE
ON TBL_MIS_CLIENTE_EXTERNO.CLE_CI_RUC =
TBL_MIS_TRAMITE.CLE_CI_RUC WHERE
TBL_MIS_TRAMITE.TRA_ID =@tra_id"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@tra_id",
SqlDbType.VarChar)).Value = tra_id
    Dim da As New SqlDataAdapter(cmdSql)
    Dim ds As New DataSet
    Try
        con.Open()
        da.Fill(ds, "datos")
        Return ds
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Function

Public Function getItemsFactura(ByVal fac_num As String, ByVal conexion
As String) As DataSet
    Dim con As New SqlConnection(conexion)

```

```

Dim sql As String = "select * FROM TBL_CO_FACTURA WHERE
fac_num =@fac_num"
Dim cmdSql As New SqlCommand(sql, con)
cmdSql.Parameters.Add(New SqlParameter("@fac_num",
SqlDbType.VarChar)).Value = fac_num
Dim da As New SqlDataAdapter(cmdSql)
Dim ds As New DataSet
Try
    con.Open()
    da.Fill(ds, "datos_factura")
    Return ds
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

```

```

Public Function getItemsCuenta(ByVal cue_id As String, ByVal conexion As
String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select * FROM TBL_CO_CUENTA WHERE
fac_num =@fac_num"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@cue_id",
SqlDbType.VarChar)).Value = cue_id
    Dim da As New SqlDataAdapter(cmdSql)
    Dim ds As New DataSet
    Try
        con.Open()
        da.Fill(ds, "datos_cuenta")
        Return ds
    
```

```

Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

Public Function getItemsCuenta_Cedula(ByVal cle_ci_ruc As String, ByVal
conexion As String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select cue_id FROM TBL_CO_CUENTA WHERE
cle_ci_ruc =@cle_ci_ruc"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@cle_ci_ruc",
SqlDbType.VarChar)).Value = cle_ci_ruc
    Dim da As New SqlDataAdapter(cmdSql)
    Dim ds As New DataSet
    Try
        con.Open()
        da.Fill(ds, "datos_cuenta")
        Return ds
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Function

Public Function getItemsFacturaDet(ByVal fac_num As String, ByVal
conexion As String) As DataSet
    Dim con As New SqlConnection(conexion)

```

```

Dim sql As String = "select * FROM TBL_CO_FACTURA_DET
WHERE FAC_NUM =@fac_num and (FAD_DESCRIPCION='Venta de
agua en tanquero' or FAD_DESCRIPCION='Venta de agua en tanquero
con 50% descuento' or FAD_DESCRIPCION='Venta de agua en tanquero
con 100% descuento')"
Dim cmdSql As New SqlCommand(sql, con)
cmdSql.Parameters.Add(New SqlParameter("@fac_num",
SqlDbType.VarChar)).Value = fac_num
Dim da1 As New SqlDataAdapter(cmdSql)
Dim ds1 As New DataSet
Try
    con.Open()
    da1.Fill(ds1, "datos_fac")
    Return ds1
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

```

```

Public Function getItemsFactura_Det(ByVal fac_num As String, ByVal
conexion As String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "select * FROM TBL_CO_FACTURA_DET
WHERE FAC_NUM =@fac_num and (FAD_DESCRIPCION<>'Venta
de agua en tanquero')and (FAD_DESCRIPCION<>'Venta de agua en
tanquero 50% descuento') and (FAD_DESCRIPCION<>'Venta de agua en
tanquero 100% descuento')"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@fac_num",
SqlDbType.VarChar)).Value = fac_num

```

```

Dim da1 As New SqlDataAdapter(cmdSql)
Dim ds1 As New DataSet
Try
    con.Open()
    da1.Fill(ds1, "datos_fac")
    Return ds1
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Function

Public Function getItemsFactura_Num(ByVal fac_num As String, ByVal
conexion As String) As DataSet
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "SELECT TOP 1 @FAC_NUM=FAC_NUM FROM
TBL_CO_FACTURA ORDER BY FAC_NUM DESC"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@fac_num",
SqlDbType.VarChar)).Value = fac_num
    Dim da1 As New SqlDataAdapter(cmdSql)
    Dim ds1 As New DataSet
    Try
        con.Open()
        da1.Fill(ds1, "datos_fac")
        Return ds1
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try

```


End Function

#End Region

#Region "Actualizaciones"

```
Public Sub Update_tbl_co_srv_venta_tanquero_sol(ByVal tra_id As String,  
ByVal conexion As String, ByVal sol_m3 As String, ByVal sol_direccion As  
String, ByVal sol_descuento As String, ByVal sol_transporte As String, ByVal  
sol_registrado As String, ByVal sol_fecha_ent As String, ByVal  
sol_num_tanquero As String, ByVal sol_tipo_tanquero As String)
```

```
    Dim con As New SqlConnection(conexion)
```

```
    Dim sql As String = "UPDATE
```

```
TBL_CO_SRV_VENTA_TANQUERO_SOL SET SOL_M3 =@sol_m3,
```

```
SOL_DIRECCION =@sol_direccion, SOL_TRASPORTE
```

```
=@sol_transporte, SOL_REGISTRADO =@sol_registrado
```

```
,SOL_DESCUENTO=@sol_descuento,SOL_FECHA_ENT=@sol_fecha_
```

```
ent, SOL_NUM_TANQUERO=@sol_num_tanquero,
```

```
sol_tipo_tanquero=@sol_tipo_tanquero WHERE TRA_ID = @tra_id "
```

```
Dim cmdSql As New SqlCommand(sql, con)
```

```
cmdSql.Parameters.Add(New SqlParameter("@tra_id",
```

```
SqlDbType.VarChar)).Value = tra_id
```

```
cmdSql.Parameters.Add(New SqlParameter("@sol_m3",
```

```
SqlDbType.VarChar)).Value = sol_m3
```

```
cmdSql.Parameters.Add(New SqlParameter("@sol_direccion",
```

```
SqlDbType.VarChar)).Value = sol_direccion
```

```
cmdSql.Parameters.Add(New SqlParameter("@sol_transporte",
```

```
SqlDbType.VarChar)).Value = sol_transporte
```

```
cmdSql.Parameters.Add(New SqlParameter("@sol_registrado",
```

```
SqlDbType.VarChar)).Value = sol_registrado
```

```
cmdSql.Parameters.Add(New SqlParameter("@sol_descuento",
```

```
SqlDbType.VarChar)).Value = sol_descuento
```

```

cmdSql.Parameters.Add(New SqlParameter("@sol_fecha_ent",
SqlDbType.VarChar)).Value = sol_fecha_ent
cmdSql.Parameters.Add(New SqlParameter("@sol_num_tanquero",
SqlDbType.VarChar)).Value = sol_num_tanquero
cmdSql.Parameters.Add(New SqlParameter("@sol_tipo_tanquero",
SqlDbType.VarChar)).Value = sol_tipo_tanquero
Dim da As New SqlDataAdapter(cmdSql)
Dim ds As New DataSet
Try
    con.Open()
    cmdSql.ExecuteNonQuery()
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Sub

Public Sub Update_tbl_mis_tramite(ByVal tra_id As String, ByVal conexion
As String, ByVal cue_id As String)
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "UPDATE TBL_MIS_TRAMITE SET CUE_ID
=@cue_id WHERE TRA_ID = @tra_id "
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@tra_id",
SqlDbType.VarChar)).Value = tra_id
    cmdSql.Parameters.Add(New SqlParameter("@cue_id",
SqlDbType.VarChar)).Value = cue_id
    Dim da As New SqlDataAdapter(cmdSql)
    Dim ds As New DataSet
    Try
        con.Open()

```

```

        cmdSql.ExecuteNonQuery()
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Sub

Public Sub Update_tbl_co_factura(ByVal fac_num As String, ByVal conexion
As String, ByVal cue_id As String)
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "UPDATE TBL_CO_FACTURA SET CUE_ID
=@cue_id WHERE FAC_NUM = @fac_num"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@fac_num",
SqlDbType.VarChar)).Value = fac_num
    cmdSql.Parameters.Add(New SqlParameter("@cue_id",
SqlDbType.VarChar)).Value = cue_id
    Dim da As New SqlDataAdapter(cmdSql)
    Dim ds As New DataSet
    Try
        con.Open()
        cmdSql.ExecuteNonQuery()
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Sub

Public Sub Update_tbl_co_fac(ByVal fac_num As String, ByVal conexion As
String, ByVal fecha As String)

```

```

Dim con As New SqlConnection(conexion)
Dim sql As String = "UPDATE TBL_CO_FACTURA SET
FAC_FECHA_FACT=@fecha WHERE FAC_NUM = @FAC_NUM"
Dim cmdSql As New SqlCommand(sql, con)
cmdSql.Parameters.Add(New SqlParameter("@fac_num",
SqlDbType.VarChar)).Value = fac_num
cmdSql.Parameters.Add(New SqlParameter("@fecha",
SqlDbType.VarChar)).Value = fecha
Dim da As New SqlDataAdapter(cmdSql)
Dim ds As New DataSet
Try
    con.Open()
    cmdSql.ExecuteNonQuery()
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Sub

Public Sub Update_tbl_mis_ruteo(ByVal fac_num As String, ByVal tra_id As
String, ByVal rej_id As String, ByVal conexion As String)
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "UPDATE TBL_MIS_RUTEO SET
RTO_FORMULARIO =@fac_num WHERE TRA_ID = @TRA_ID AND
REJ_ID = @REJ_ID"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@fac_num",
SqlDbType.VarChar)).Value = fac_num
    cmdSql.Parameters.Add(New SqlParameter("@tra_id",
SqlDbType.VarChar)).Value = tra_id

```

```

cmdSql.Parameters.Add(New SqlParameter("@rej_id",
SqlDbType.VarChar)).Value = rej_id
Dim da As New SqlDataAdapter(cmdSql)
Dim ds As New DataSet
Try
    con.Open()
    cmdSql.ExecuteNonQuery()
Catch ex As SqlClient.SqlException
    msg = ex.Message
Finally
    con.Close()
End Try
End Sub

Public Sub Update_tbl_mis_ruteo(ByVal tra_id As String, ByVal rej_id As
String, ByVal conexion As String)
    Dim con As New SqlConnection(conexion)
    Dim sql As String = "UPDATE TBL_MIS_RUTEO_EJECUCION SET
rop_id =1 WHERE TRA_ID = @TRA_ID AND REJ_ID = @REJ_ID"
    Dim cmdSql As New SqlCommand(sql, con)
    cmdSql.Parameters.Add(New SqlParameter("@tra_id",
SqlDbType.VarChar)).Value = tra_id
    cmdSql.Parameters.Add(New SqlParameter("@rej_id",
SqlDbType.VarChar)).Value = rej_id
    Dim da As New SqlDataAdapter(cmdSql)
    Dim ds As New DataSet
    Try
        con.Open()
        cmdSql.ExecuteNonQuery()
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally

```

```

        con.Close()
    End Try
End Sub

#End Region

#Region "Procedimientos"

Public Sub EJECUTAR_SP_MIS_RUTEO_EJECUCION_VALIDAR(ByVal
tra_id As String, ByVal rej_id As String, ByVal conexion As String)
    Dim con As New SqlConnection(conexion)
    Dim cmdSql As New
SqlCommand("SP_MIS_RUTEO_EJECUCION_VALIDAR", con)
    cmdSql.CommandType = CommandType.StoredProcedure
    cmdSql.CommandText = "SP_MIS_RUTEO_EJECUCION_VALIDAR"
    cmdSql.Parameters.Add(New SqlParameter("@tra_id",
SqlDbType.VarChar)).Value = tra_id
    cmdSql.Parameters.Add(New SqlParameter("@rej_id",
SqlDbType.VarChar)).Value = rej_id
    Dim da As New SqlDataAdapter(cmdSql)
    Dim ds As New DataSet
    Try
        con.Open()
        cmdSql.ExecuteNonQuery()
    Catch ex As SqlClient.SqlException
        msg = ex.Message
    Finally
        con.Close()
    End Try
End Sub

```

```
Public Sub EJECUTAR_SP_CO_FACTURA_INSUMO_CAB_TANQ(ByVal  
cle_ci_ruc As String, ByVal cue_id As String, ByVal tra_id As String, ByVal  
rej_id As String, ByVal conexion As String)
```

```
    Dim con As New SqlConnection(conexion)
```

```
    Dim cmdSql As New
```

```
        SqlCommand("SP_CO_FACTURA_INSUMO_CAB_TANQ", con)
```

```
        cmdSql.CommandType = CommandType.StoredProcedure
```

```
        cmdSql.CommandText = "SP_CO_FACTURA_INSUMO_CAB_TANQ"
```

```
        cmdSql.Parameters.Add(New SqlParameter("@cle_ci_ruc",
```

```
            SqlDbType.VarChar)).Value = cle_ci_ruc
```

```
        cmdSql.Parameters.Add(New SqlParameter("@cue_id",
```

```
            SqlDbType.VarChar)).Value = cue_id
```

```
        cmdSql.Parameters.Add(New SqlParameter("@tra_id",
```

```
            SqlDbType.VarChar)).Value = tra_id
```

```
        cmdSql.Parameters.Add(New SqlParameter("@rej_id",
```

```
            SqlDbType.VarChar)).Value = rej_id
```

```
    Dim da As New SqlDataAdapter(cmdSql)
```

```
    Dim ds As New DataSet
```

```
    Try
```

```
        con.Open()
```

```
        cmdSql.ExecuteNonQuery()
```

```
    Catch ex As SqlClient.SqlException
```

```
        msg = ex.Message
```

```
    Finally
```

```
        con.Close()
```

```
    End Try
```

```
End Sub
```

```
#End Region
```

```
End Class
```

Código Principal

Código Visual Basic .NET

Solicitud de Venta de Agua en Tanquero

```
Imports System.Data.SqlClient
Public Class pg_co_srv_venta_tanquero_sol_nueva
    Inherits System.Web.UI.Page

    #Region " Código generado por el Diseñador de Web Forms "
```

El Diseñador de Web Forms requiere esta llamada.

```
<System.Diagnostics.DebuggerStepThrough(>
Private Sub InitializeComponent()

    End Sub
Protected WithEvents txtCedula As
System.Web.UI.WebControls.TextBox
Protected WithEvents lbl_num_tramite As
System.Web.UI.WebControls.Label
Protected WithEvents txt_nombre_cliente As
System.Web.UI.WebControls.TextBox
Protected WithEvents txt_metros_cubicos As
System.Web.UI.WebControls.TextBox
Protected WithEvents txt_dir_entrega As
System.Web.UI.WebControls.TextBox
Protected WithEvents Button1 As System.Web.UI.WebControls.Button
Protected WithEvents rb_emapa As
System.Web.UI.WebControls.RadioButton
Protected WithEvents ddl_descuento As
System.Web.UI.WebControls.DropDownList
```


Protected WithEvents cb_registrado As
System.Web.UI.WebControls.CheckBox

Protected WithEvents btn_Guardar As
System.Web.UI.WebControls.Button

Protected WithEvents C_fecha_ent As
System.Web.UI.WebControls.Calendar

Protected WithEvents txt_fecha_ent As
System.Web.UI.WebControls.TextBox

Protected WithEvents Label1 As System.Web.UI.WebControls.Label

Protected WithEvents ddl_Num_tanqueros As
System.Web.UI.WebControls.DropDownList

Protected WithEvents txt_capacidad As
System.Web.UI.WebControls.TextBox

Protected WithEvents rb_particular1 As
System.Web.UI.WebControls.RadioButton

Protected WithEvents rb_particular2 As
System.Web.UI.WebControls.RadioButton

Protected WithEvents ddl_tipo_tanquero As
System.Web.UI.WebControls.DropDownList

Protected WithEvents Button3 As System.Web.UI.WebControls.Button

'NOTA: el Diseñador de Web Forms necesita la siguiente declaración del
marcador de posición.

'No se debe eliminar o mover.

Private designerPlaceholderDeclaration As System.Object

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles MyBase.Init

'CODEGEN: el Diseñador de Web Forms requiere esta llamada de
método

'No la modifique con el editor de código.

InitializeComponent()

End Sub

#End Region

```
Dim pr_clave As String
Dim pr_rej_id As String
Dim pr_tra_id As Decimal
Dim c As New Clase_inicio
Dim sql As New Clase_sql
Dim dr As DataRow
Dim num_tanquero As Integer
Dim MES As String
```

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
    pr_clave = Request.QueryString("pr_clave")
    pr_tra_id = Request.QueryString("pr_tra_id")
    pr_rej_id = Request.QueryString("pr_rej_id")
    If Not Page.IsPostBack Then
        Me.txt_capacidad.Attributes.Add("onkeypress", "solonumeros() ;")
        Dim i As Integer
        For i = 1 To 30
            ddl_Num_tanqueros.Items.Add(i.ToString)
        Next
        Dim ds_usuario As DataSet = sql.getItemsUsuario(pr_tra_id, c.conexion)
        Dim ds_tramite As DataSet = sql.getItemstramite(pr_tra_id, c.conexion)
        Dim ds_tanquero As DataSet = sql.getItemstanquero(pr_tra_id,
        c.conexion)
        Dim ds_parametro As DataSet = sql.getItemsparametro(c.conexion)
        Dim convenio As String
        Dim transporte As String
        Dim tipo_t As String
```

```

For Each dr In ds_usuario.Tables("datos").Rows
    Me.txtCedula.Text = dr.Item("cle_ci_ruc").ToString
    Me.txt_nombre_cliente.Text = dr.Item("cle_nombres").ToString &
    " " & dr.Item("cle_apellidos").ToString
Next
For Each dr In ds_parametro.Tables("datos_parametro").Rows
    Me.ddl_tipo_tanquero.Items.Add(dr.Item("prm_descripcion").ToString)
Next
For Each dr In ds_tanquero.Tables("datos_tanquero").Rows
    If IsDBNull(dr.Item("sol_m3")) = False Then
        Me.txt_metros_cubicos.Text = dr.Item("sol_m3").ToString
    End If
    Me.txt_dir_entrega.Text = dr.Item("sol_direccion").ToString
    Me.txt_fecha_ent.Text = dr.Item("sol_fecha_ent").ToString
    If IsDBNull(dr.Item("sol_descuento")) = False Then
        Me.ddl_descuento.SelectedIndex = CInt(dr.Item("sol_descuento"))
    End If
    If IsDBNull(dr.Item("sol_num_tanquero")) = False Then
        Me.ddl_Num_tanqueros.SelectedValue =
        dr.Item("sol_num_tanquero").ToString
    End If
    If IsDBNull(dr.Item("sol_tipo_tanquero")) = False Then
        tipo_t = dr.Item("sol_tipo_tanquero").ToString
    End If
    If Me.txt_fecha_ent.Text = "" Then
        Me.txt_fecha_ent.Text =
        Me.C_fecha_ent.SelectedDate.Today.ToString
    End If
    If dr.Item("sol_trasporte").ToString = "1" Then
        Me.rb_emapa.Checked = True
    Else
        If dr.Item("sol_trasporte").ToString = "2" Then

```

```

        Me.rb_particular1.Checked = True
    Else
        Me.rb_particular2.Checked = True
    End If
End If
Next
For Each dr In ds_parametro.Tables("datos_parametro").Rows
    If tipo_t = dr.Item("prm_id").ToString Then
        Me.ddl_tipo_tanquero.SelectedValue =
            dr.Item("prm_descripcion").ToString
    End If
Next
For Each dr In ds_tramite.Tables("datos_tramite").Rows
    Me.lbl_num_tramite.Text = dr.Item("tra_id").ToString
Next
Dim estado As String
Dim ds_ruteo As DataSet = sql.getItemsruteo(pr_tra_id, pr_rej_id,
c.conexion)
For Each dr In ds_ruteo.Tables("datos_ruteo").Rows
    estado = dr.Item("rej_estado").ToString
    If estado = "1" Then
        Me.cb_registrado.Checked = True
        Me.cb_registrado.Enabled = False
        Me.btn_Guardar.Enabled = False
    End If
Next
End If
End Sub

```

```

Private Sub btn_Guardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_Guardar.Click

```

```

    Dim ds_tanquero As DataSet = sql.getItemstanquero(c.conexion)

```

```

Dim ds_parametro As DataSet = sql.getItemsparametro(c.conexion)
Dim ds_tanqueromes As DataSet = sql.getItemstanqueromes(c.conexion,
Me.C_fecha_ent.TodaysDate.Month.ToString)
Dim ds_parametrotipo As DataSet =
sql.getItemsparametro(Me.ddl_tipo_tanquero.SelectedValue, c.conexion)
Dim num_techo As Integer
Dim num_techo_prm As Integer
Dim tipo_tanquero As String
Dim sol_transporte As Integer
Dim sol_registrado As Integer
Dim sol_convenio As Integer
sol_registrado = 1
If Me.rb_emapa.Checked Then
    sol_transporte = 1
Else
    If Me.rb_particular1.Checked Then
        sol_transporte = 2
    Else
        sol_transporte = 3
    End If
End If
For Each dr In ds_tanquero.Tables("datos_tanquero").Rows
    If Me.txt_fecha_ent.Text = dr.Item("sol_fecha_ent").ToString And
dr.Item("sol_trasporte").ToString = sol_transporte Then
        Me.num_tanquero = CInt(Me.dr.Item("sol_num_tanquero").ToString)
        + Me.num_tanquero
    End If
Next
Me.num_tanquero = Me.ddl_Num_tanqueros.SelectedValue +
Me.num_tanquero
For Each dr In ds_tanquero.Tables("datos_tanquero").Rows

```

```

If Me.txt_fecha_ent.Text = dr.Item("sol_fecha_ent").ToString And
dr.Item("sol_tipo_tanquero").ToString = "234" Then
    num_techo = CInt(Me.dr.Item("sol_num_tanquero").ToString) +
    num_techo
End If
Next
For Each dr In ds_parametro.Tables("datos_parametro").Rows
    If Me.ddl_tipo_tanquero.SelectedValue =
dr.Item("prm_descripcion").ToString Then
        num_techo_prm = CInt(dr.Item("prm_valor").ToString)
    End If
Next
num_techo = Me.ddl_Num_tanqueros.SelectedValue + num_techo
Dim num_techo_mes As Integer
For Each dr In ds_tanqueromes.Tables("datos_tanqueromes").Rows
    If Me.ddl_tipo_tanquero.SelectedValue = dr.Item("TIPO").ToString Then
        num_techo_mes = CInt(Me.dr.Item("CANTIDAD").ToString) +
        num_techo_mes
    End If
Next
num_techo_mes = Me.ddl_Num_tanqueros.SelectedValue +
num_techo_mes

If (Me.num_tanquero > 5 And sol_transporte <> 2) Or (num_techo >
num_techo_prm) Or num_techo_mes > 60 Then
    Label1.Text = "Ud. ha excedido el cupo diario/mensual de venta"
Else
    If Me.txt_fecha_ent.Text <> "" Or Me.txt_fecha_ent.Text <> "Fecha
Incorrecta" Then
        If Me.cb_registrado.Checked Then
            For Each dr In ds_parametrotipo.Tables("datos_parametro").Rows
                tipo_tanquero = dr.Item("prm_id").ToString
            Next
        End If
    End If
End If

```

```

Next
    sql.Update_tbl_co_srv_venta_tanquero_sol(pr_tra_id, c.conexion,
    Me.txt_metros_cubicos.Text, Me.txt_dir_entrega.Text,
    Me.ddl_descuento.SelectedItem.Value, sol_transporte,
    sol_registrado, Me.txt_fecha_ent.Text,
    Me.ddl_Num_tanqueros.SelectedValue, tipo_tanquero)
    sql.EJECUTAR_SP_MIS_RUTEO_EJECUCION_VALIDAR(pr_t
    ra_id, pr_rej_id, c.conexion)
    Dim ds_tramite As DataSet = sql.getItemstramite(pr_tra_id,
    c.conexion)
    Dim cedula As String
    For Each dr In ds_tramite.Tables("datos_tramite").Rows
        cedula = dr.Item("cle_ci_ruc").ToString
    Next
    Me.Response.Redirect("../mis/pg_mis_tramite.asp")
Else
    Me.Response.Redirect("../mis/pg_mis_tramite.asp")
End If
End If
Me.Response.Redirect("../mis/pg_mis_tramite.asp")
End If
End Sub

```

```

Private Sub C_fecha_ent_SelectionChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles C_fecha_ent.SelectionChanged
    Dim year As Integer
    Dim mes As Integer
    Dim dia As Integer
    year = (CInt(C_fecha_ent.SelectedDate.Year) - CInt(Date.Today.Year))
    mes = (CInt(C_fecha_ent.SelectedDate.Month) - CInt(Date.Today.Month))
    dia = (CInt(C_fecha_ent.SelectedDate.Day) - CInt(Date.Today.Day))
    If year > 0 Then

```

```

txt_fecha_ent.Text = C_fecha_ent.SelectedDate.ToString
Else
  If year = 0 And mes > 0 Then
    txt_fecha_ent.Text = C_fecha_ent.SelectedDate.ToString
  Else
    If year = 0 And mes = 0 And dia >= 0 Then
      txt_fecha_ent.Text = C_fecha_ent.SelectedDate.ToString
    Else
      Me.txt_fecha_ent.Text = "Fecha Incorrecta"
      'txt_fecha_ent.Text = ""
    End If
  End If
End If
C_fecha_ent.Visible = False
End Sub

Private Sub ddl_Num_tanqueros_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ddl_Num_tanqueros.SelectedIndexChanged
  Me.txt_metros_cubicos.Text = Me.ddl_Num_tanqueros.SelectedValue *
  Me.txt_capacidad.Text
End Sub

Public Function IsDBNull(ByVal dbvalue) As Boolean
  Return dbvalue Is DBNull.Value
End Function

Private Sub txt_capacidad_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txt_capacidad.TextChanged
  Me.txt_metros_cubicos.Text = Me.ddl_Num_tanqueros.SelectedValue *
  Me.txt_capacidad.Text
End Sub

```


End Class

Asignar Cuenta

```
Public Class pg_co_asignar_cuenta_nueva
```

```
    Inherits System.Web.UI.Page
```

```
#Region " Código generado por el Diseñador de Web Forms "
```

```
    'El Diseñador de Web Forms requiere esta llamada.
```

```
    <System.Diagnostics.DebuggerStepThrough(>
```

```
    Private Sub InitializeComponent()
```

```
End Sub
```

```
    Protected WithEvents lbl_num_tramite As
```

```
        System.Web.UI.WebControls.Label
```

```
    Protected WithEvents txtCedula As
```

```
        System.Web.UI.WebControls.TextBox
```

```
    Protected WithEvents btn_asignar As
```

```
        System.Web.UI.WebControls.Button
```

```
    Protected WithEvents ddl_cuenta As
```

```
        System.Web.UI.WebControls.DropDownList
```

```
    Protected WithEvents lbl_Sin_Cuenta As
```

```
        System.Web.UI.WebControls.Label
```

```
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
```

```
    Protected WithEvents txt_nombre_cliente As
```

```
        System.Web.UI.WebControls.TextBox
```

```
    Protected WithEvents cb_registrado As
```

```
        System.Web.UI.WebControls.CheckBox
```

```
    Protected WithEvents txtCuenta As
```

```
        System.Web.UI.WebControls.TextBox
```

'NOTA: el Diseñador de Web Forms necesita la siguiente declaración del marcador de posición.

'No se debe eliminar o mover.

```
Private designerPlaceholderDeclaration As System.Object
```

```
Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
```

'CODEGEN: el Diseñador de Web Forms requiere esta llamada de método

'No la modifique con el editor de código.

```
InitializeComponent()
```

```
End Sub
```

```
#End Region
```

```
Dim cadena As New Clase_inicio
```

```
Dim sql As New Clase_sql
```

```
Dim pr_clave As String
```

```
Dim pr_rej_id As String
```

```
Dim pr_tra_id As Decimal
```

```
Dim cuenta As String
```

```
Dim dr As DataRow
```

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

'Introducir aquí el código de usuario para inicializar la página

```
pr_clave = Request.QueryString("pr_clave")
```

```
pr_tra_id = Request.QueryString("pr_tra_id")
```

```
pr_rej_id = Request.QueryString("pr_rej_id")
```

```
If Not Page.IsPostBack Then
```

```
Dim ds_tramite As DataSet = sql.getItemstramite(pr_tra_id, cadena.conexion)
```

```

Dim ds_usuario As DataSet = sql.getItemsUsuario(pr_tra_id, c
adena.conexion)

Dim cedula As String

For Each dr In ds_tramite.Tables("datos_tramite").Rows
    Me.txtCedula.Text = dr.Item("cle_ci_ruc").ToString
    Me.txtCuenta.Text = dr.Item("cue_id").ToString
Next

For Each dr In ds_usuario.Tables("datos").Rows
    Me.txt_nombre_cliente.Text = dr.Item("cle_nombres").ToString &
    " " & dr.Item("cle_apellidos").ToString
Next

cedula = Me.txtCedula.Text

Dim ds_cuenta As DataSet = sql.getItemsCuenta_Cedula(cedula,
cadena.conexion)

If Me.txtCuenta.Text <> "" Then
    Me.txtCuenta.Visible = True
    Me.ddl_cuenta.Visible = False
Else
    For Each dr In ds_cuenta.Tables("datos_cuenta").Rows
        Me.ddl_cuenta.Items.Add(dr.Item("cue_id").ToString)
    Next
    If Me.ddl_cuenta.Items.Count() = 0 Then
        Me.ddl_cuenta.Visible = False
        Me.lbl_Sin_Cuenta.Visible = True
    End If
End If

Dim estado As String

Dim ds_ruteo As DataSet = sql.getItemsruteo(pr_tra_id, pr_rej_id,
cadena.conexion)

For Each dr In ds_ruteo.Tables("datos_ruteo").Rows
    estado = dr.Item("rej_estado").ToString
    If estado = "1" Then

```

```

        Me.cb_registrado.Checked = True
        Me.cb_registrado.Enabled = False
        Me.btn_asignar.Enabled = False
    End If
Next
End If
End Sub

Private Sub btn_asignar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_asignar.Click
    If Me.cb_registrado.Checked = True Then
        cuenta = Me.ddl_cuenta.SelectedValue.ToString
        sql.Update_tbl_mis_tramite(pr_tra_id, cadena.conexion, cuenta)
        sql.EJECUTAR_SP_MIS_RUTEO_EJECUCION_VALIDAR(pr_tra_id,
        pr_rej_id, cadena.conexion)
    End If
    Me.Response.Redirect("../mis/pg_mis_tramite.asp")
End Sub

End Class

```

Factura

```

Public Class pg_co_factura_pre_fs_prueba
    Inherits System.Web.UI.Page
    Dim cadena As New Clase_inicio
    Protected WithEvents txt_num_fac As
System.Web.UI.WebControls.TextBox
    Protected WithEvents txt_fecha_fac As
System.Web.UI.WebControls.TextBox
    Protected WithEvents txt_cle_ci_ruc As
System.Web.UI.WebControls.TextBox

```

```

Protected WithEvents ddl_forma_pago As
System.Web.UI.WebControls.DropDownList
Dim sql As New Clase_sql
Dim pr_clave As String
Dim pr_rej_id As String
Protected WithEvents txt_nombres As
System.Web.UI.WebControls.TextBox
Protected WithEvents cb_pagada As
System.Web.UI.WebControls.CheckBox
Dim pr_tra_id As Decimal
#Region " Código generado por el Diseñador de Web Forms "

```

'El Diseñador de Web Forms requiere esta llamada.

```

<System.Diagnostics.DebuggerStepThrough(>
Private Sub InitializeComponent()

```

```

End Sub

```

```

Protected WithEvents btnGuardar As
System.Web.UI.WebControls.Button
Protected WithEvents lbl_num_tramite As
System.Web.UI.WebControls.Label

```

'NOTA: el Diseñador de Web Forms necesita la siguiente declaración del marcador de posición.

'No se debe eliminar o mover.

```

Private designerPlaceholderDeclaration As System.Object
Dim cuenta As String

```

```

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Init

```

'CODEGEN: el Diseñador de Web Forms requiere esta llamada de método

'No la modifique con el editor de código.

InitializeComponent()

End Sub

#End Region

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

'Introducir aquí el código de usuario para inicializar la página

pr_clave = Request.QueryString("pr_clave")

pr_tra_id = Request.QueryString("pr_tra_id")

pr_rej_id = Request.QueryString("pr_rej_id")

If Not Page.IsPostBack Then

Dim ds_tramite As DataSet = sql.getItemstramite(pr_tra_id, cadena.conexion)

Dim ds_factura As DataSet = sql.getItemsFactura(pr_clave, cadena.conexion)

Dim ds_usuario As DataSet = sql.getItemsUsuario(pr_tra_id, cadena.conexion)

Dim dr As DataRow

Dim pagada As String

For Each dr In ds_tramite.Tables("datos_tramite").Rows

lbl_num_tramite.Text = dr.Item("tra_id").ToString

Next

For Each dr In ds_factura.Tables("datos_factura").Rows

txt_num_fac.Text = dr.Item("fac_num").ToString

txt_cle_ci_ruc.Text = dr.Item("cle_ci_ruc").ToString

txt_fecha_fac.Text = dr.Item("fac_fecha_factura").ToString

cuenta = dr.Item("cue_id").ToString

pagada = dr.Item("fac_pagada").ToString

Next

If pagada = True Then

```

        Me.cb_pagada.Checked = True
    End If
    For Each dr In ds_usuario.Tables("datos").Rows
        Me.txt_nombres.Text = dr.Item("cle_nombres").ToString &
            " " & dr.Item("cle_apellidos").ToString
    Next
    If cuenta <> "" Then
        ddl_forma_pago.Enabled = True
    Else
        ddl_forma_pago.Enabled = False
    End If
    Dim estado As String
    Dim ds_ruteo As DataSet = sql.getItemsruteo(pr_tra_id, pr_rej_id,
        cadena.conexion)
    For Each dr In ds_ruteo.Tables("datos_ruteo").Rows
        estado = dr.Item("rej_estado").ToString
        If estado = "1" Then
            Me.btnGuardar.Enabled = False
        End If
    Next
End If
End Sub

```

```

Private Sub btnGuardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGuardar.Click

```

```

    If Me.ddl_forma_pago.SelectedValue = "Credito" Then
        sql.Update_tbl_mis_ruteo(pr_tra_id, pr_rej_id, cadena.conexion)
        Me.script("../co/pg_co_factura_convenio.asp?pr_fac_num=" +
            pr_clave.ToString + "&pr_tra_id=" + pr_tra_id.ToString)
    
```

```

Else

```

```

If Me.cb_pagada.Checked = True Then
    sql.Update_tbl_mis_ruteo(pr_tra_id, pr_rej_id, cadena.conexion)
    sql.EJECUTAR_SP_MIS_RUTEO_EJECUCION_VALIDAR(pr_t
ra_id, pr_rej_id, cadena.conexion)
    Me.script("../mis/pg_mis_tramite.asp")
Else
    Me.script("../mis/pg_mis_tramite.asp")
End If
End If
End Sub

```

```

Private Sub script(ByVal url As String)
    System.Web.HttpContext.Current.Response.Write("<script>" & vbCrLf)
    System.Web.HttpContext.Current.Response.Write("parent.location= """"
& url & """" & vbCrLf)
    System.Web.HttpContext.Current.Response.Write("</script>")
End Sub

```

End Class

```

Public Class pg_co_factura_pre_det_nueva
    Inherits System.Web.UI.Page
    Dim cadena As New Clase_inicio
    Dim sql As New Clase_sql
    Dim pr_clave As String
    Dim pr_rej_id As String
    Dim pr_tra_id As Decimal
    Dim cuenta As String
    Protected WithEvents txt_servicio1 As
System.Web.UI.WebControls.TextBox
    Protected WithEvents txt_cantidad1 As
System.Web.UI.WebControls.TextBox

```



```

Protected WithEvents txt_vu1 As System.Web.UI.WebControls.TextBox
Protected WithEvents txt_vt1 As System.Web.UI.WebControls.TextBox
Protected WithEvents txt_servicio2 As
System.Web.UI.WebControls.TextBox
Protected WithEvents txt_cantidad2 As
System.Web.UI.WebControls.TextBox
Protected WithEvents txt_vu2 As System.Web.UI.WebControls.TextBox
Protected WithEvents txt_vt2 As System.Web.UI.WebControls.TextBox
Protected WithEvents lbl_tramite As System.Web.UI.WebControls.Label
Protected WithEvents lbl_id As System.Web.UI.WebControls.Label
Protected WithEvents lbl_id2 As System.Web.UI.WebControls.Label
Protected WithEvents txt_iva1 As System.Web.UI.WebControls.TextBox
Protected WithEvents txt_iva2 As System.Web.UI.WebControls.TextBox
Dim dr As DataRow

```

#Region " Código generado por el Diseñador de Web Forms "

'El Diseñador de Web Forms requiere esta llamada.

```

<System.Diagnostics.DebuggerStepThrough(>
Private Sub InitializeComponent()

```

```

End Sub

```

'NOTA: el Diseñador de Web Forms necesita la siguiente declaración del marcador de posición.

'No se debe eliminar o mover.

```

Private designerPlaceholderDeclaration As System.Object

```

```

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Init

```

'CODEGEN: el Diseñador de Web Forms requiere esta llamada de método

'No la modifique con el editor de código.

```
InitializeComponent()
```

```
End Sub
```

```
#End Region
```

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
pr_clave = Request.QueryString("pr_clave")
```

```
pr_tra_id = Request.QueryString("pr_tra_id")
```

```
pr_rej_id = Request.QueryString("pr_rej_id")
```

```
If Not Page.IsPostBack Then
```

```
Dim ds_facturadet As DataSet = sql.getItemsFacturaDet(pr_clave,  
cadena.conexion)
```

```
Dim ds_factura_det As DataSet = sql.getItemsFactura_Det(pr_clave,  
cadena.conexion)
```

```
Dim cedula As String
```

```
lbl_tramite.Text = pr_tra_id
```

```
For Each dr In ds_facturadet.Tables("datos_fac").Rows
```

```
Me.lbl_id.Text = dr.Item("fad_id").ToString
```

```
Me.txt_servicio1.Text = dr.Item("fad_descripcion").ToString
```

```
Me.txt_cantidad1.Text = dr.Item("fad_cantidad").ToString
```

```
Me.txt_vu1.Text = dr.Item("fad_valor_unitario").ToString
```

```
Me.txt_iva1.Text = dr.Item("fad_iva").ToString
```

```
Me.txt_vt1.Text = dr.Item("fad_valor").ToString
```

```
Next
```

```
For Each dr In ds_factura_det.Tables("datos_fac").Rows
```

```
Me.lbl_id2.Text = dr.Item("fad_id").ToString
```

```
Me.txt_servicio2.Text = dr.Item("fad_descripcion").ToString
```

```
Me.txt_cantidad2.Text = dr.Item("fad_cantidad").ToString
```

```
Me.txt_vu2.Text = dr.Item("fad_valor_unitario").ToString
```

```
Me.txt_iva2.Text = dr.Item("fad_iva").ToString
```

```
Me.txt_vt2.Text = dr.Item("fad_valor").ToString
```

```
Next
End If
End Sub
```

```
End Class
```

Entrega de agua

```
Public Class pg_co_srv_venta_tanquero_ent_nueva
    Inherits System.Web.UI.Page
    Dim cadena As New Clase_inicio
    Dim sql As New Clase_sql
    Dim pr_clave As String
    Dim pr_rej_id As String
    Protected WithEvents txt_nombres As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents btn_guardar As
    System.Web.UI.WebControls.Button
    Protected WithEvents cb_registrado As
    System.Web.UI.WebControls.CheckBox
    Dim pr_tra_id As Decimal

    #Region " Código generado por el Diseñador de Web Forms "
```

'El Diseñador de Web Forms requiere esta llamada.

```
<System.Diagnostics.DebuggerStepThrough(>
Private Sub InitializeComponent()
```

```
End Sub
```

```
Protected WithEvents lbl_num_tramite As
System.Web.UI.WebControls.Label
```

```
Protected WithEvents txtCedula As  
System.Web.UI.WebControls.TextBox
```

'NOTA: el Diseñador de Web Forms necesita la siguiente declaración del
marcador de posición.

'No se debe eliminar o mover.

```
Private designerPlaceholderDeclaration As System.Object
```

```
Private Sub Page_Init(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Init
```

'CODEGEN: el Diseñador de Web Forms requiere esta llamada de método

'No la modifique con el editor de código.

```
InitializeComponent()
```

```
End Sub
```

```
#End Region
```

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
pr_clave = Request.QueryString("pr_clave")
```

```
pr_tra_id = Request.QueryString("pr_tra_id")
```

```
pr_rej_id = Request.QueryString("pr_rej_id")
```

'Introducir aquí el código de usuario para inicializar la página

```
If Not Page.IsPostBack Then
```

```
Dim ds As DataSet = sql.getItemstramite(pr_tra_id, cadena.conexion)
```

```
Dim dr As DataRow
```

```
For Each dr In ds.Tables("datos_tramite").Rows
```

```
lbl_num_tramite.Text = dr.Item("tra_id").ToString
```

```
Next
```

```
Dim ds_usuario As DataSet = sql.getItemsUsuario(pr_tra_id,  
cadena.conexion)
```

```
For Each dr In ds_usuario.Tables("datos").Rows
```

```

        Me.txtCedula.Text = dr.Item("cle_ci_ruc").ToString
        Me.txt_nombres.Text = dr.Item("cle_nombres").ToString & " " &
        dr.Item("cle_apellidos").ToString
    Next
    Dim estado As String
    Dim ds_ruteo As DataSet = sql.getItemsruteo(pr_tra_id, pr_rej_id,
cadena.conexion)
    For Each dr In ds_ruteo.Tables("datos_ruteo").Rows
        estado = dr.Item("rej_estado").ToString
        If estado = "1" Then
            Me.cb_registrado.Checked = True
            Me.cb_registrado.Enabled = False
            Me.btn_guardar.Enabled = False
        End If
    Next
End If
End Sub

Private Sub btn_guardar_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_guardar.Click
    If Me.cb_registrado.Checked = True Then
        sql.EJECUTAR_SP_MIS_RUTEO_EJECUCION_VALIDAR(pr_t
ra_id, pr_rej_id, cadena.conexion)
        Me.Response.Redirect("../mis/pg_mis_tramite.asp")
    End If
End Sub
End Class

```

Código Html

Solicitud de Venta de Agua en Tanquero

```
<%@ Page Language="vb" AutoEventWireup="false"
Codebehind="pg_co_srv_venta_tanquero_sol_nueva.aspx.vb"
Inherits="sisnuevo.pg_co_srv_venta_tanquero_sol_nueva"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
  <HEAD>
    <title>pg_co_srv_venta_tanquero_sol_nueva</title>
    <meta content="Microsoft Visual Studio .NET 7.1"
name="GENERATOR">
    <meta content="Visual Basic .NET 7.1"
name="CODE_LANGUAGE">
    <meta content="JavaScript"
name="vs_defaultClientScript">
    <meta
content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
  </HEAD>
  <body MS_POSITIONING="GridLayout">
    <form id="Form1" method="post" runat="server">
      <script language="javascript">
function solonumeros()
{
var tecla = window.event.keyCode;
if (tecla < 48 || tecla > 57)
{
window.event.keyCode=0;
}
}
      </script>
    </form>
  </body>
</HTML>
```

```

</script>
<table height="15" cellSpacing="0" cellPadding="1" width="100%"
bgColor="#2872a3" summary="Encabezado"
borderColorLight="#003366" border="1">
<tr>
<td align="center" width="250">
<DIV style="DISPLAY: inline; FONT-WEIGHT: bold; FONT-
SIZE: 9pt; WIDTH: 206px; COLOR: #ebf2f8; FONT-STYLE:
italic; FONT-FAMILY: 'Times New Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">VENTA DE AGUA EN
TANQUERO</DIV>
</td>
<td align="right" height="20">
<DIV style="DISPLAY: inline; FONT-WEIGHT: bold; FONT-
SIZE: 10pt; WIDTH: 184px; COLOR: #d8e6f1; FONT-FAMILY:
'Times New Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">Trámite #<asp:label
id="lbl_num_tramite" Font-Italic="True" ForeColor="#EBF2F8"
Font-Bold="True" Font-Size="10pt"

Runat="server"></asp:label>
</DIV>
</td>
</tr>
</table>
<br>
<asp:calendar id="C_fecha_ent" style="Z-INDEX: 101; LEFT: 480px;
POSITION: absolute; TOP: 224px" runat="server" CellPadding="0"
BackColor="#E0E0E0" BorderWidth="2px" BorderStyle="Solid"
ShowGridLines="True" Height="152px" Width="184px">
<TodayDayStyle Font-Bold="True"></TodayDayStyle>
</asp:calendar><br>

```

```
<table style="WIDTH: 656px; HEIGHT: 362px" height="362"
cellSpacing="0" cellPadding="0" border="1">
```

```
<TBODY>
```

```
<tr>
```

```
<td align="center" bgColor="#d8e6f1" colSpan="4">
```

```
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
372px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 16px" align="right"
ms_positioning="FlowLayout"><b>SOLICITUD DE VENTA DE
AGUA EN TANQUERO</b></DIV>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="WIDTH: 102px" vAlign="top" align="right"
bgColor="#d8e6f1">
```

```
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
112px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Cliente</DIV>
```

```
</td>
```

```
<td colSpan="3"><asp:textbox id="txtCedula" runat="server"
Font-Size="10pt" ReadOnly="True" Font-Names="Times New
Roman"></asp:textbox><br>
```

```
<asp:textbox id="txt_nombre_cliente" runat="server" Font-
Size="10pt" Width="288px" ReadOnly="True"Font-
Names="Times New Roman"></asp:textbox></td>
```

```
</tr>
```

```
<tr>
```

```
<td style="WIDTH: 102px; HEIGHT: 25px" align="right"
width="102" bgColor="#d8e6f1">
```



```

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
112px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Metros Cubicos</DIV>
</td>
<td style="WIDTH: 190px; HEIGHT: 25px"
width="190"><asp:textbox id="txt_metros_cubicos"
runat="server" Font-Size="10pt" Width="48px" ReadOnly="True"
Font-Names="Times New Roman">12</asp:textbox>
</td>
<td style="HEIGHT: 25px" align="right" width="100"
bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
140px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Numero de Tanqueros</DIV>
</td>
<td style="HEIGHT: 25px" width="200">
<asp:dropdownlist id="ddl_Num_tanqueros" runat="server" Font-
Size="10pt" Width="72px" Font-Names="Times New Roman"
AutoPostBack="True"></asp:dropdownlist>
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 62px;
COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="left"
ms_positioning="FlowLayout">Capacidad</DIV>
<asp:textbox id="txt_capacidad" runat="server" Font-Size="10pt"
Width="40px" Font-Names="Times New Roman"
AutoPostBack="True">12</asp:textbox>
</td>
</tr>
<tr>

```

```

<td style="WIDTH: 102px; HEIGHT: 20px" align="right"
width="102" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
112px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Tipo Tanquero</DIV>
</td>
<td style="HEIGHT: 20px" colSpan="3">
<asp:dropdownlist id="ddl_tipo_tanquero" runat="server" Font-
Size="10pt" Width="174px" Font-Names="Times New Roman"
AutoPostBack="True"></asp:dropdownlist>
</td>
</tr>
<tr>
<td style="WIDTH: 102px; HEIGHT: 57px" vAlign="top"
align="right" width="102" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
112px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Direccion de Entrega</DIV>
</td>
<td style="WIDTH: 190px; HEIGHT: 57px" colSpan="1">
<asp:textbox id="txt_dir_entrega" runat="server" Font-Size="10pt"
Height="80px" Width="208px" Font-Names="Times New Roman"
TextMode="MultiLine"></asp:textbox>
</td>
<td style="HEIGHT: 57px" vAlign="top" align="right"
width="100" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
140px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Fecha de Entrega</DIV>

```

```

<td style="HEIGHT: 57px" vAlign="top">
<P><asp:textbox id="txt_fecha_ent" runat="server" Font-
Size="10pt" Width="131px" Font-Names="Times New Roman"
Enabled="False"></asp:textbox><asp:button id="Button1"
runat="server" Width="32px" Text="->"></asp:button></P>
<P><asp:button id="Button3" runat="server" Width="164px"
Text="Verificar Disponibilidad"></asp:button></P>
</TD>
<tr>
<td style="WIDTH: 102px" vAlign="top" align="right"
width="102" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
112px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT:
17px"align="right"ms_positioning="FlowLayout">Transporte</DI
V>
</td>
<td style="WIDTH: 199px" colSpan="3"><asp:radiobutton
id="rb_emapa" runat="server" GroupName="Rb_Transporte"
Checked="True"></asp:radiobutton>
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 64px;
COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="left"
ms_positioning="FlowLayout">Emapa</DIV>
<br>
<asp:radiobutton id="rb_particular1" runat="server"
GroupName="Rb_Transporte"></asp:radiobutton>
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 72px;
COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px" align="left"
ms_positioning="FlowLayout">Particular 1</DIV>
<br>

```

```

<asp:radiobutton      id="rb_particular2"      runat="server"
GroupName="Rb_Transporte"></asp:radiobutton>
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 72px;
COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT: 17px"      align="left"
ms_positioning="FlowLayout">Particular 2</DIV>
</td>
</tr>
<tr>
<td style="WIDTH: 102px; HEIGHT: 13px" align="right"
width="102" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
112px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT:
17px"align="right"ms_positioning="FlowLayout">Descuento</DI
V>
</td>
<td style="HEIGHT: 13px"  colSpan="3"><asp:dropdownlist
id="ddl_descuento"      runat="server"      Font-Size="10pt"
Width="64px" Font-Names="Times New Roman">
<asp:ListItem Value="0" Selected="True">0%</asp:ListItem>
<asp:ListItem Value="1">50%</asp:ListItem>
<asp:ListItem Value="2">100%</asp:ListItem>
</asp:dropdownlist></td>
</tr>
<tr>
<td style="WIDTH: 102px" align="right" width="102"
bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
112px; COLOR: #003366; FONT-FAMILY: 'Times New Roman';
HEIGHT:

```

```

17px"align="right"ms_positioning="FlowLayout">Registrado</DI
V>
</td>
<td colspan="3"><asp:checkbox id="cb_registrado"
runat="server"></asp:checkbox></td>
</tr>
<tr>
<td style="WIDTH: 102px" align="center" colspan="1"
height="50"><asp:button id="btn_Guardar" runat="server"
Text="Guardar"></asp:button></td>
<td colspan="3">
<DIV><asp:label id="Label1" runat="server" Font-Italic="True"
ForeColor="Red" Font-Bold="True" Font-Size="11pt"
Height="8px" Width="520px" Font-Names="Times New
Roman"></asp:label>
</DIV>
</td>
</tr>
</TBODY>
</table>
</form>
</TR></TBODY></TABLE></FORM>
</body>
</HTML>

```

Asignar Cuenta

```

<%@ Page Language="vb" AutoEventWireup="false"
Codebehind="pg_co_asignar_cuenta_nueva.aspx.vb"
Inherits="sisnuevo.pg_co_asignar_cuenta_nueva"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">

```

```

<HTML>
<HEAD>
    <title>pg_co_asignar_cuenta_nueva</title>
    <meta content="Microsoft Visual Studio .NET 7.1"
name="GENERATOR">
    <meta content="Visual Basic .NET 7.1"
name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content=http://schemas.microsoft.com/intellisense/ie5
name="vs_targetSchema">
</HEAD>
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">
<table height="15" cellSpacing="0" cellPadding="1" width="100%"
bgColor="#2872a3" summary="Encabezado"
borderColorLight="#003366" border="1">
    <tr>
        <td align="center" width="250">
            <DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 9pt; WIDTH: 206px; COLOR: #ebf2f8;
FONT-STYLE: italic; FONT-FAMILY: 'Times New
Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">ASIGNAR
CUENTA</DIV>
        </td>
        <td align="right" height="20">
            <DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 10pt; WIDTH: 184px; COLOR: #d8e6f1;
FONT-FAMILY: 'Times New Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">Trámite #<asp:label
id="lbl_num_tramite" Runat="server" Font-Size="10pt"

```

```

        Font-Bold="True"      ForeColor="#EBF2F8"      Font-
        Italic="True"></asp:label>
    </DIV>
</td>
</tr>
</table>
<br>
<br>
<table style="WIDTH: 624px; HEIGHT: 74px" height="74"
cellSpacing="0" cellPadding="0" border="1">
    <tr>
        <td align="center" bgColor="#d8e6f1" colSpan="4"><DIV
        style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
        372px; COLOR: #003366; FONT-FAMILY: 'Times New
        Roman'; HEIGHT: 16px" align="center"
        ms_positioning="FlowLayout"><b>ASIGNAR
        CUENTA</b>
        </DIV>
        </td>
        <tr>
            <td vAlign="top" align="right" bgColor="#d8e6f1">
            <DIV style="DISPLAY: inline; FONT-SIZE: 12pt;
            WIDTH: 132px; COLOR: #003366; FONT-FAMILY:
            'Times New Roman'; HEIGHT: 19px"
            ms_positioning="FlowLayout">Cliente</DIV>
            </td>
            <td colspan="3"><asp:textbox id="txtCedula"
            runat="server" ReadOnly="True" Font-Size="10pt" Font-
            Names="Times New Roman"></asp:textbox><asp:textbox
            id="txt_nombre_cliente" runat="server" Width="329px"

```

```

        ReadOnly="True" Font-Size="10pt" Font-Names="Times
        New Roman"></asp:textbox></td>
</tr>
<tr>
    <td vAlign="top" align="right" bgColor="#d8e6f1"
    style="HEIGHT: 20px">
        <DIV style="DISPLAY: inline; FONT-SIZE: 12pt;
        WIDTH: 132px; COLOR: #003366; FONT-FAMILY:
        'Times New Roman'; HEIGHT: 19px"
        ms_positioning="FlowLayout">Cuenta</DIV>
    </td>
    <td colspan="3" style="HEIGHT:
    20px"><asp:dropdownlist id="ddl_cuenta" runat="server"
    Width="128px" Font-Size="10pt" Font-Names="Times
    New Roman"></asp:dropdownlist>
    <asp:TextBox id="txtCuenta" runat="server"
    Visible="False" Width="128px" ReadOnly="True" Font-
    Size="10pt" Font-Names="Times New
    Roman"></asp:TextBox>
    <asp:label id="lbl_Sin_Cuenta" runat="server"
    Width="112px" Visible="False" ForeColor="Navy" Font-
    Size="10pt" Font-Names="Times New Roman">Cliente sin
    Cuenta</asp:label>
    </td>
</tr>
<tr>
    <td vAlign="top" align="right" bgColor="#d8e6f1">
        <DIV style="DISPLAY: inline; FONT-SIZE: 12pt;
        WIDTH: 132px; COLOR: #003366; FONT-FAMILY:
        'Times New Roman'; HEIGHT: 19px"
        ms_positioning="FlowLayout">Registrado</DIV>
    </td>

```



```

        <td colspan="3"><asp:checkbox id="cb_registrado"
runat="server"></asp:checkbox></td>
    </tr>
    <tr>
        <td align="center" colspan="1" height="50"><asp:button
id="btn_asignar"runat="server" Text="Asignar">
</asp:button></td>
        <td colspan="3"><asp:label id="Label1" runat="server"
Width="416px"></asp:label></td>
    </tr>
</table>
</form>
</body>
</HTML>

```

Factura

```

<% @ Page Language="vb" AutoEventWireup="false"
Codebehind="pg_co_factura_pre_nueva.aspx.vb"
Inherits="sisnuevo.pg_co_factura_pre_fs_prueba"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<HEAD>
    <title>pg_co_factura_pre_prueba</title>
    <meta content="Microsoft Visual Studio .NET 7.1"
name="GENERATOR">
    <meta content="Visual Basic .NET 7.1"
name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">

```

```

</HEAD>
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">
<table height="15" cellSpacing="0" cellPadding="1" width="100%"
bgColor="#2872a3" summary="Encabezado"
borderColorLight="#003366" border="1">
  <tr>
    <td align="center" width="250">
      <DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 9pt; WIDTH: 198px; COLOR: #ebf2f8;
FONT-STYLE: italic; FONT-FAMILY: 'Times New
Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">VENTA DE AGUA EN
TANQUERO</DIV>
    </td>
    <td align="right" height="20">
      <DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 10pt; WIDTH: 184px; COLOR: #d8e6f1;
FONT-FAMILY: 'Times New Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">Trámite #<asp:label
id="lbl_num_tramite" Runat="server" Font-Size="10pt"
Font-Bold="True" ForeColor="#EBF2F8" Font-
Italic="True"></asp:label>
      </DIV>
    </td>
  </tr>
</table>
<br>
<br>
<table cellSpacing="0" cellPadding="0" width="50%" border="1">
  <tr>
    <td align="center" bgColor="#d8e6f1" colspan="6">

```

```

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 112px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px" align="center"
ms_positioning="FlowLayout"><b>FACTURA</DIV>
</B></td>
</tr>
<tr>
<td style="WIDTH: 125px; HEIGHT: 35px" vAlign="top"
align="right" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 125px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Numero de Factura</DIV>
</td>
<td style="HEIGHT: 35px"><asp:textbox
id="txt_num_fac" runat="server" Width="144px"
ReadOnly="True" Font-Size="10pt" Font-Names="Times
New Roman"></asp:textbox></td>
<td style="HEIGHT: 35px" vAlign="top" align="right"
bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 112px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px" align="right"
ms_positioning="FlowLayout">Fecha Facturacion</DIV>
</td>
<td style="HEIGHT: 35px" colSpan="3"><asp:textbox
id="txt_fecha_fac" runat="server" Width="133px"
ReadOnly="True" Font-Size="10pt" Font-Names="Times
New Roman"></asp:textbox></td>
</tr>
<tr>

```

```

<td style="WIDTH: 125px" align="right" width="125"
bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 125px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px"
ms_positioning="FlowLayout">Cliente Externo
CI/RUC</DIV>
</td>
<td width="50"><asp:textbox id="txt_cle_ci_ruc"
runat="server" Width="144px" ReadOnly="True" Font-
Size="10pt" Font-Names="Times New
Roman"></asp:textbox></td>
<td align="right" width="150" colSpan="4"><asp:textbox
id="txt_nombres" runat="server" Width="368px"
ReadOnly="True" Font-Size="10pt" Font-Names="Times
New Roman"></asp:textbox></td>
</tr>
<tr>
<td style="WIDTH: 125px" vAlign="top" align="right"
width="125" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 112px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px"
ms_positioning="FlowLayout">Pagada</DIV>
</td>
<td colSpan="5"><asp:checkbox id="cb_pagada"
runat="server" Enabled="False"></asp:checkbox></td>
</tr>
<tr>
<td style="WIDTH: 125px; HEIGHT: 5px" vAlign="top"
align="right" width="125" bgColor="#d8e6f1">

```

```

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 112px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px"
ms_positioning="FlowLayout">Forma de Pago</DIV>
</td>
<td style="HEIGHT: 5px" colSpan="5"><asp:dropdownlist
id="ddl_forma_pago" runat="server" AutoPostBack="True"
Font-Size="10pt" Font-Names="Times New Roman">
<asp:ListItem Value="Efectivo"Selected="True">Efectivo
</asp:ListItem>
<asp:ListItem Value="Credito">Credito</asp:ListItem>
</asp:dropdownlist></td>
</tr>
<tr>
<td colSpan="6" height="50"><asp:button id="btnGuardar"
runat="server" Text="Guardar"></asp:button></td>
</tr>
</table>
</form>
</body>
</HTML>

```

```

<% @ Page Language="vb" AutoEventWireup="false"
Codebehind="pg_co_factura_pre_det_nueva.aspx.vb"
Inherits="sisnuevo.pg_co_factura_pre_det_nueva"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<HEAD>
<title>pg_co_factura_pre_det_nueva</title>

```

```

<meta content="Microsoft Visual Studio .NET 7.1"
name="GENERATOR">
<meta content="Visual Basic .NET 7.1"
name="CODE_LANGUAGE">
<meta content="JavaScript" name="vs_defaultClientScript">
<meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
</HEAD>
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">
<table height="15" cellSpacing="0" cellPadding="1" width="100%"
bgColor="#2872a3" summary="Encabezado" borderColorLight=
"#003366" border="1">
<tr>
<td align="center" width="250">
<DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 9pt; WIDTH: 198px; COLOR: #ebf2f8;
FONT-STYLE: italic; FONT-FAMILY: 'Times New
Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">DETALLE DE
FACTURA</DIV>
</td>
<td align="right" height="20">
<DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 10pt; WIDTH: 184px; COLOR: #d8e6f1;
FONT-FAMILY: 'Times New Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">Trámite #
<asp:label id="lbl_tramite"
runat="server"></asp:label></DIV>
</td>
</tr>
</table>

```


<table cellSpacing="0" cellPadding="0" width="50%" border="1">

<tr>

<td align="center" bgColor="#d8e6f1"></td>

<td bgColor="#d8e6f1">

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 48px; COLOR: #003366; FONT-FAMILY: 'Times New Roman'; HEIGHT: 17px" align="center" ms_positioning="FlowLayout">Id</DIV>

</td>

<td style="WIDTH: 237px" bgColor="#d8e6f1">

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 252px; COLOR: #003366; FONT-FAMILY: 'Times New Roman'; HEIGHT: 17px" align="center" ms_positioning="FlowLayout">Servicio</DIV>

</td>

<td bgColor="#d8e6f1">

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 88px; COLOR: #003366; FONT-FAMILY: 'Times New Roman'; HEIGHT: 17px" align="center" ms_positioning="FlowLayout">Cantidad</DIV>

</td>

<td bgColor="#d8e6f1">

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH: 88px; COLOR: #003366; FONT-FAMILY: 'Times New Roman'; HEIGHT: 17px" align="center" ms_positioning="FlowLayout">Valor Unitario</DIV>

</td>

<td bgColor="#d8e6f1">

```

<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 86px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px" align="center"
ms_positioning="FlowLayout">IVA</DIV>
</td>
<td bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 11pt;
WIDTH: 86px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 17px" align="center"
ms_positioning="FlowLayout">Valor Total</DIV>
</td>
</tr>
<tr>
<td align="center" bgColor="#d8e6f1"></td>
<td      bgColor="#d8e6f1"><asp:label      id="lbl_id"
runat="server" Font-Names="Times New Roman" Font-
Size="10pt"></asp:label></td>
<td      style="WIDTH:      237px"
bgColor="#d8e6f1"><asp:textbox      id="txt_servicio1"
runat="server" Width="252px" ReadOnly="True" Font-
Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td bgColor="#d8e6f1"><asp:textbox id="txt_cantidad1"
runat="server" Width="88px" ReadOnly="True" Font-
Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td      bgColor="#d8e6f1"><asp:textbox      id="txt_vu1"
runat="server" Width="88px" ReadOnly="True" Font-
Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td      bgColor="#d8e6f1"><asp:textbox      id="txt_iva1"
runat="server" Width="88px" ReadOnly="True" Font-

```



```

Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td      bgColor="#d8e6f1"><asp:textbox      id="txt_vt1"
runat="server"      Width="88px"      ReadOnly="True"      Font-
Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td align="center" bgColor="#d8e6f1 "></td>
</tr>
<tr>
<td bgColor="#d8e6f1 "></td>
<td      bgColor="#d8e6f1"><asp:label      id="lbl_id2"
runat="server"      Font-Names="Times New Roman"      Font-
Size="10pt"></asp:label></td>
<td      style="WIDTH:      237px"      bgColor="#d8e6f1">
<asp:textbox      id="txt_servicio2"      runat="server"
Width="252px"      ReadOnly="True"      Font-Names="Times
New Roman"      Font-Size="10pt"></asp:textbox></td>
<td      bgColor="#d8e6f1"><asp:textbox      id="txt_cantidad2"
runat="server"      Width="88px"      ReadOnly="True"      Font-
Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td      bgColor="#d8e6f1"><asp:textbox      id="txt_vu2"
runat="server"      Width="88px"      ReadOnly="True"      Font-
Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td bgColor="#d8e6f1">
<asp:TextBox      id="txt_iva2"      runat="server"      Width="88px"
ReadOnly="True"      Font-Names="Times New Roman"
      Font-Size="10pt"></asp:TextBox></td>
<td      bgColor="#d8e6f1"><asp:textbox      id="txt_vt2"
runat="server"      Width="88px"      ReadOnly="True"      Font-

```

```

Names="Times      New      Roman"      Font-
Size="10pt"></asp:textbox></td>
<td bgColor="#d8e6f1"></td>
</tr>
<tr>
<td align="center" colSpan="8" height="1"></td>
</tr>
</table>
</form>
</body>
</HTML>

```

Entrega de agua

```

<%@ Page Language="vb" AutoEventWireup="false"
Codebehind="pg_co_srv_venta_tanquero_ent_nueva.aspx.vb"
Inherits="sisnuevo.pg_co_srv_venta_tanquero_ent_nueva"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<HEAD>
<title>pg_co_srv_venta_tanquero_ent_nueva</title>
<meta content="Microsoft Visual Studio .NET 7.1"
name="GENERATOR">
<meta content="Visual Basic .NET 7.1"
name="CODE_LANGUAGE">
<meta content="JavaScript" name="vs_defaultClientScript">
<meta content=http://schemas.microsoft.com/intellisense/ie5
name="vs_targetSchema">
</HEAD>
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">

```

```
<table height="15" cellSpacing="0" cellPadding="1" width="100%"
bgColor="#2872a3" summary="Encabezado" borderColorLight=
"#003366" border="1">
```

```
<tr>
```

```
<td align="center" width="250">
```

```
<DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 9pt; WIDTH: 180px; COLOR: #ebf2f8;
FONT-STYLE: italic; FONT-FAMILY: 'Times New
Roman'; HEIGHT: 8px"
ms_positioning="FlowLayout">TOTAL DE LA
FACTURA</DIV>
```

```
</td>
```

```
<td align="right" height="20">
```

```
<DIV style="DISPLAY: inline; FONT-WEIGHT: bold;
FONT-SIZE: 10pt; WIDTH: 184px; COLOR: #d8e6f1;
FONT-FAMILY: 'Times New Roman'; HEIGHT: 16px"
ms_positioning="FlowLayout">Trámite #<asp:label
id="lbl_num_tramite" Font-Italic="True"
ForeColor="#EBF2F8" Font-Bold="True" Font-
Size="10pt" Runat="server"></asp:label>
```

```
</DIV>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<br>
```

```
<table style="WIDTH: 537px; HEIGHT: 104px" height="104"
cellSpacing="0" cellPadding="0" border="1">
```

```
<tr>
```

```
<td align="center" bgColor="#d8e6f1" colSpan="4"><DIV
style="DISPLAY: inline; FONT-SIZE: 11pt; WIDTH:
372px; COLOR: #003366; FONT-FAMILY: 'Times New
```

```

Roman';    HEIGHT:    16px"    align="center"
ms_positioning="FlowLayout"><b>ENTREGA    DE
VENTA DE AGUA EN TANQUERO </b>
</DIV>
</td>
</tr>
<tr>
<td vAlign="top" align="right" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 12pt;
WIDTH: 132px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 19px"
ms_positioning="FlowLayout">Cliente</DIV>
</td>
<td colspan="3"><asp:textbox id="txtCedula"
runat="server" ReadOnly="True" Font-Size="10pt" Font-
Names="Times New Roman"></asp:textbox><br>
<asp:textbox id="txt_nombres" runat="server"
Width="396px" ReadOnly="True" Font-Size="10pt" Font-
Names="Times New Roman"></asp:textbox></td>
</tr>
<tr>
<td vAlign="top" align="right" bgColor="#d8e6f1">
<DIV style="DISPLAY: inline; FONT-SIZE: 12pt;
WIDTH: 132px; COLOR: #003366; FONT-FAMILY:
'Times New Roman'; HEIGHT: 19px"
ms_positioning="FlowLayout">Registrado</DIV>
</td>
<td colspan="3"><asp:checkbox id="cb_registrado"
runat="server"></asp:checkbox></td>
</tr>
<tr>

```

```

        <td><asp:button    id="btn_guardar"    runat="server"
        Text="Guardar"  Font-Size="10pt"  Font-Names="Times
        New Roman"></asp:button></td>
    </tr>
</table>
</form>
</body>
</HTML>

```

5.2. PRUEBAS

Para evitar los errores durante el funcionamiento de la aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, hemos realizado controles necesarios a medida que se fue desarrollando:

Evitar que se ingresen letras en campos numéricos

```

function solonumeros()
{
    var tecla = window.event.keyCode;
    if (tecla < 48 || tecla > 57)
    {
        window.event.keyCode=0;
    }
}

```

Evitar que se ingrese una fecha anterior a la actual

```

Private Sub C_fecha_ent_SelectionChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles C_fecha_ent.SelectionChanged

```

```

Dim year As Integer
Dim mes As Integer
Dim dia As Integer
year = (CInt(C_fecha_ent.SelectedDate.Year) - CInt(Date.Today.Year))
mes = (CInt(C_fecha_ent.SelectedDate.Month) - CInt(Date.Today.Month))
dia = (CInt(C_fecha_ent.SelectedDate.Day) - CInt(Date.Today.Day))
If year > 0 Then
    txt_fecha_ent.Text = C_fecha_ent.SelectedDate.ToString
Else
    If year = 0 And mes > 0 Then
        txt_fecha_ent.Text = C_fecha_ent.SelectedDate.ToString
    Else
        If year = 0 And mes = 0 And dia >= 0 Then
            txt_fecha_ent.Text = C_fecha_ent.SelectedDate.ToString
        Else
            Me.txt_fecha_ent.Text = "Fecha Incorrecta"
            'txt_fecha_ent.Text = ""
        End If
    End If
End If

C_fecha_ent.Visible = False
End Sub

```

CAPITULO VI

IMPLANTACION

6.1. INICIO DEL FUNCIONAMIENTO DEL SISTEMA

La aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, se encuentra en pleno funcionamiento, sin encontrarse ninguna novedad en el proceso que maneja.

6.2. ADIESTRAMIENTO DEL PERSONAL

La capacitación se realizó durante la presentación de avances al coordinador empresarial, Ing. Fabián Poveda, además los días previos a la fecha de culminación de aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, se procedió a capacitar a la principal persona encargada del manejo del sistema.

CAPITULO VII

CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

- ✓ La aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, permite al usuario trabajar de una manera confiable y organizada.
- ✓ La aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, es un aporte al desarrollo informático de la Empresa
- ✓ Mediante la aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, se ha logrado reducir el margen de desperdicio de papeles.
- ✓ La aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA, permite atender a los clientes de una manera mas rápida y sencilla

7.2. RECOMENDACIONES

- ✓ Capacitar a los usuarios sobre las modificaciones realizadas al sistema.

- ✓ Mejorar o cambiar las maquinas que no se encuentren en capacidad de manejar el sistema.

- ✓ Deben existir una buena comunicación entre el usuario de la aplicación y la unidad de computo para notificar cualquier punto relativo al sistema

BIBLIOGRAFIA

Libros

SENN, A. James. (1997). Análisis y Diseño de Sistemas de Información. Segunda Edición. Mc. Graw Hill.

KROENKE, David M. (2003). Procesamiento de Base de Datos. Octava Edición. México. Prentice Hall Hispanoamericana S.A.

Internet

<http://www.webestilo.com/aspnet/aspnet01.phtml>

<http://es.gotdotnet.com/quickstart/aspplus/doc/whatisaspx.aspx>

<http://www.wipo.int/about-ip/es/>

http://es.wikipedia.org/wiki/Propiedad_intelectual

<http://www.monografias.com/trabajos5/desof/desof.shtml>

http://es.wikipedia.org/wiki/Base_de_datos

<http://es.wikipedia.org/wiki/ASP.NET>

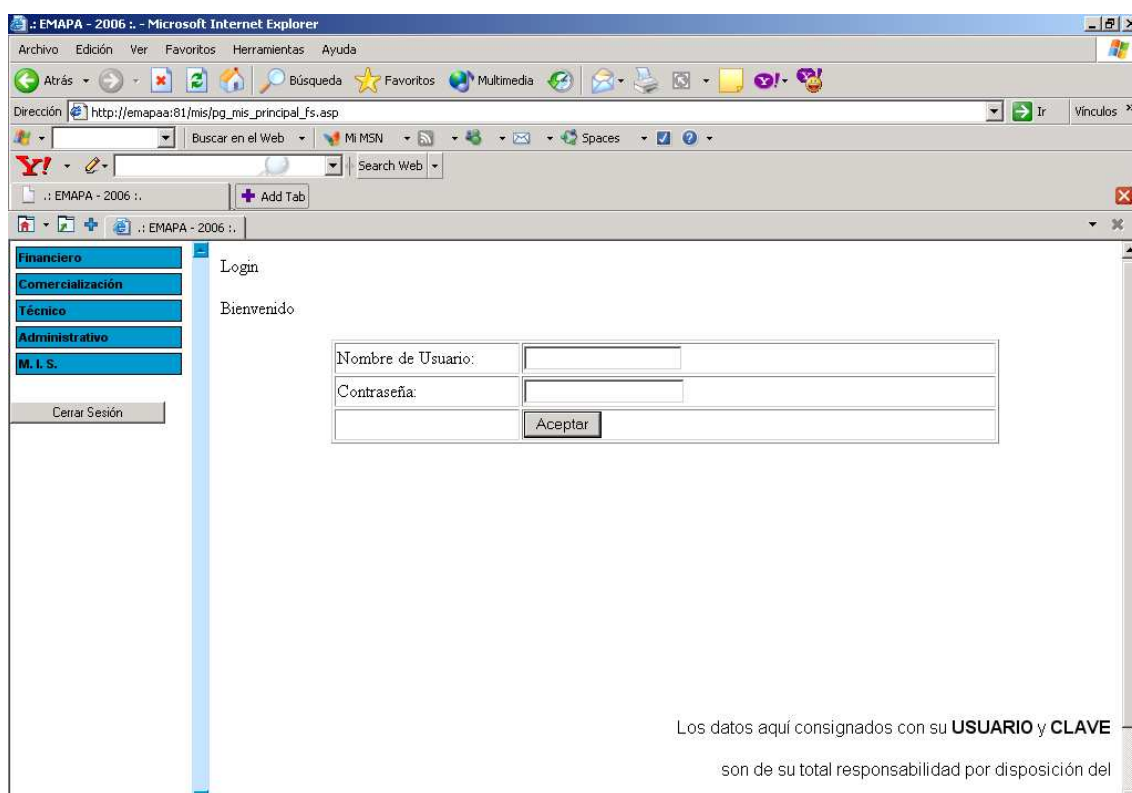
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art169.asp>

ANEXOS

ANEXO 1 MANUAL DE USUARIO

INTRODUCCION

APLICACIÓN PARA LA GESTIÓN DE SERVICIOS EN LA VENTA DE TANQUEROS EN EL MÓDULO COMERCIAL DE EMAPA



La Empresa Municipal de Agua Potable y Alcantarillado de Ambato esta empeñada en día a día mejorar los servicios que presta a la colectividad ambateña; entre ellos se contempla la venta de agua en tanqueros, actividad que es de gran importancia ya que se la realiza diariamente.

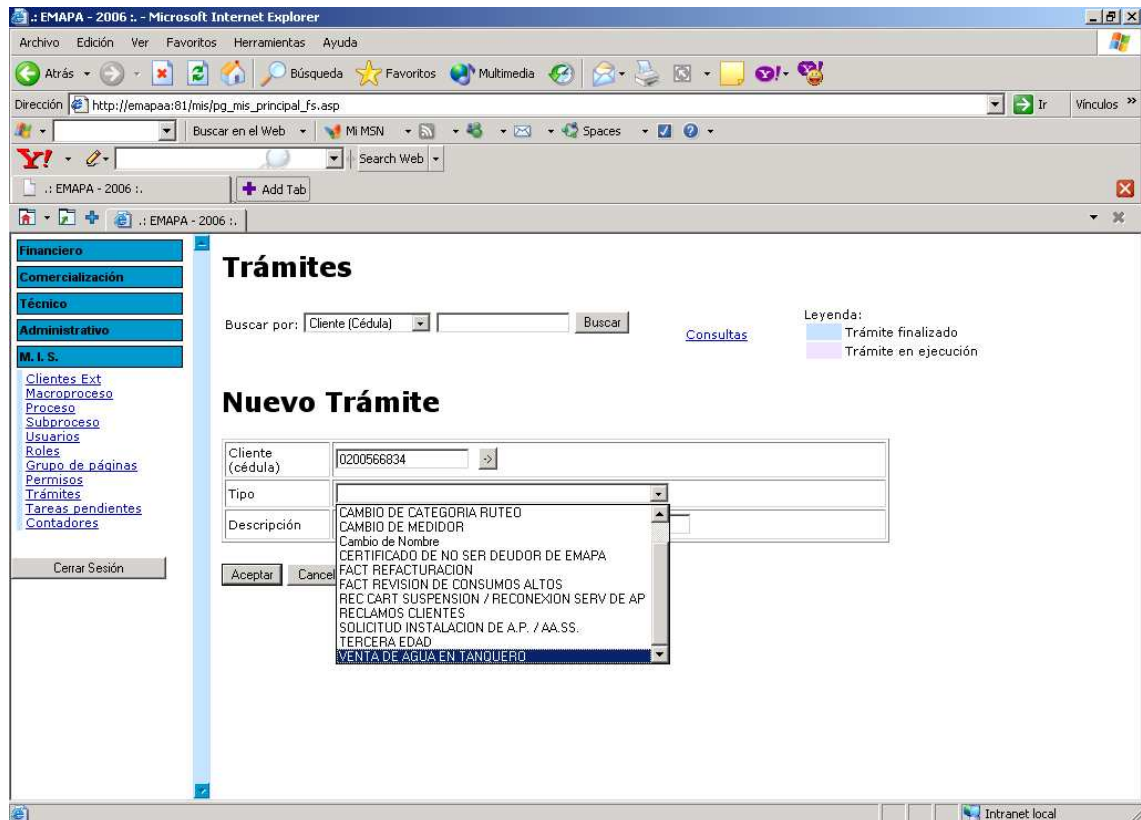
Este trámite se lo realiza en el módulo comercial de la Empresa, donde el cliente se acerca a solicitar el servicio que requiere.

Para facilitar este proceso y disminuir el tiempo que se tarda en dar atención a los clientes se ha visto la necesidad de desarrollar la presente aplicación que ayudara al encargado de registrar todo el tramite, y a su vez brindar un mejor servicio a la comunidad.

El presente documento esta dirigido al usuario de la aplicación como ayuda y soporte en su trabajo.

La aplicación se presenta como paginas Web, que pueden ser visualizadas en el Internet Explorer 6 con Service Pack 1, a continuación describiremos paso a paso la utilización de la aplicación para la gestión de servicios en la venta de tanqueros en el módulo comercial de EMAPA.

1. Ingresar Datos de Usuario y Escoger Tramite



En esta página Ud. debe escoger en el menú izquierdo la sección de trámites, para crear un nuevo registro.

Debe escoger el numero de cedula de un cliente que previamente haya sido registrado, y además, en el combo inferior debe escoger la opción que dice “VENTA DE AGUA EN TANQUERO”, y dar click en el botón aceptar con lo que pasaremos a la siguiente ventana.

2. Solicitud de Venta de Agua en Tanquero

The screenshot shows a web browser window titled "EMAPA - 2006" with the URL "http://localhost/tanq/mis/pg_mis_principal_fs.asp". The page displays a form for "SOLICITUD DE VENTA DE AGUA EN TANQUERO" with the following fields and values:

SOLICITUD DE VENTA DE AGUA EN TANQUERO	
Cliente	0102760360
	WILMA AMELIA CULQUI CAISAGUANO
Metros Cubicos	0
Numero de Tanqueros	1
Capacidad	12
Tipo Tanquero	VENTA NORMAL
Direccion de Entrega	
Fecha de Entrega	03/11/2006 0:00:00
Transporte	<input checked="" type="radio"/> Emapa <input type="radio"/> Particular 1 <input type="radio"/> Particular 2
Descuento	0%
Registrado	<input type="checkbox"/>

A calendar widget is visible, showing the month of November 2006. The date 03/11/2006 is selected. The calendar grid shows days from 30 to 10.

At the bottom of the form, there is a "Guardar" button.

En esta página Ud. debe ingresar todos los datos relacionados con la venta de agua en tanquero como:

Metros cúbicos. Para lo que Ud debe ingresar la capacidad del tanquero y seleccionar el número de tanqueros del pedido, automáticamente se realiza la multiplicación y se obtiene el resultado en el campo Metros Cúbicos

Tipo Tanquero. Este campo nos indica el tipo de venta de tanquero que se quiere realizar puede ser:


Venta Normal

Techo Propio

Contrato Colectivo

Donado

Dirección de Entrega. Que debe ser el lugar al que el cliente desee que le sea entregada el agua.

Fecha de Entrega. Se debe ingresar el día que se va a entregar el banquero de agua para lo que se debe hacer clic sobre el botón  con lo que se desplegará un calendario en el cual podemos seleccionar la fecha de entrega.

Transporte. Indica que transporte será el encargado de llevar el agua hasta su lugar de destino, puede ser:

EMAPA

Particular 1, en caso de ser un pedido grande (20 tanqueros o más)

Particular 2

Descuento. Se debe escoger del combo una opción que puede ser:

0% - Venta sin descuento (Venta Normal)

50% - En caso de que la opción Tipo Tanquero fuera “Techo Propio” o “Contrato Colectivo”

100% - En caso de que la opción Tipo Tanquero Fuera “Donado”

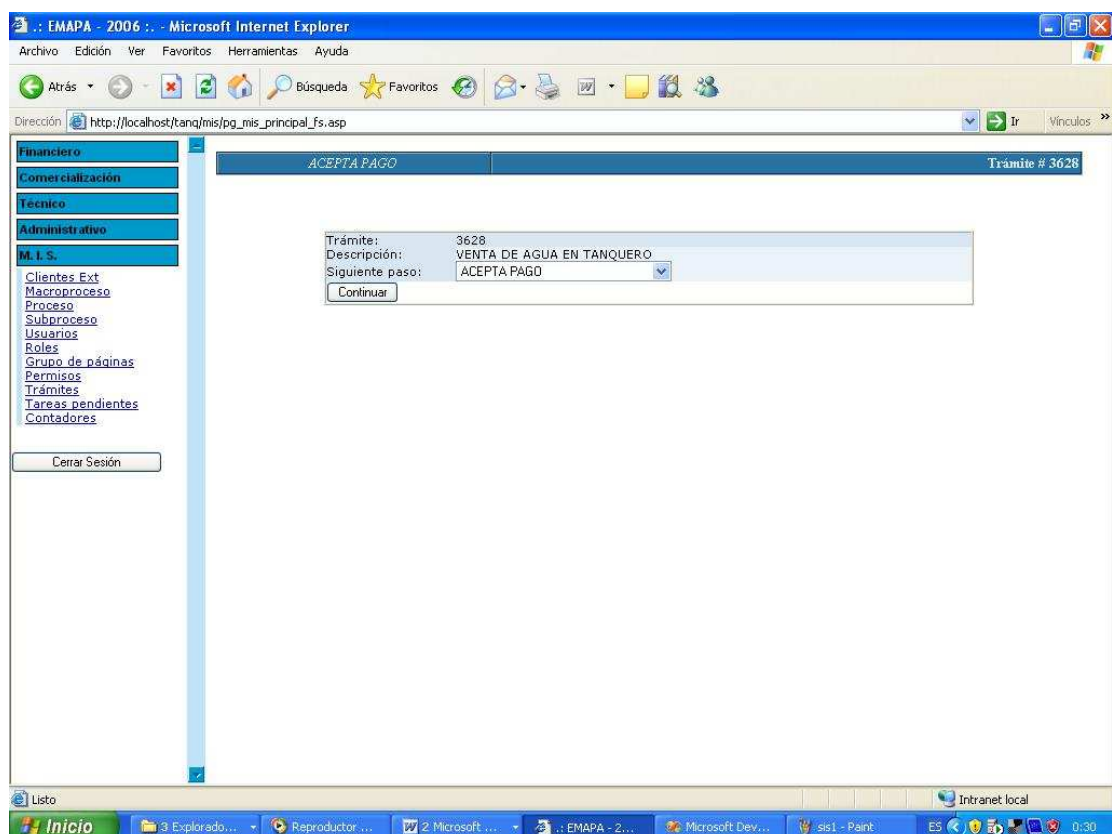
Una vez consignados estos datos y con la seguridad de que se encuentran bien ingresados, se procederá a dar un clic sobre el recuadro de registrado, y luego en el botón guardar,

Si los datos fueron bien ingresados y no se esta cometiendo algún tipo de inconsistencia, la aplicación procederá con siguiente paso

Nota: recuerde que los datos del cliente ya fueron consignados en el paso anterior

3. Aceptación De Pago

En esta pagina se deberá cerciorarse de que el cliente se encuentra de acuerdo con el pago por el servicio que la empresa esta vendiendo, en caso de que se encuentre de acuerdo, se debe escoger la opción “ACEPTA PAGO”, para seguir adelante con el proceso, de lo contrario se tomara la opción “NO ACEPTA PAGO” y el tramite será finalizado.



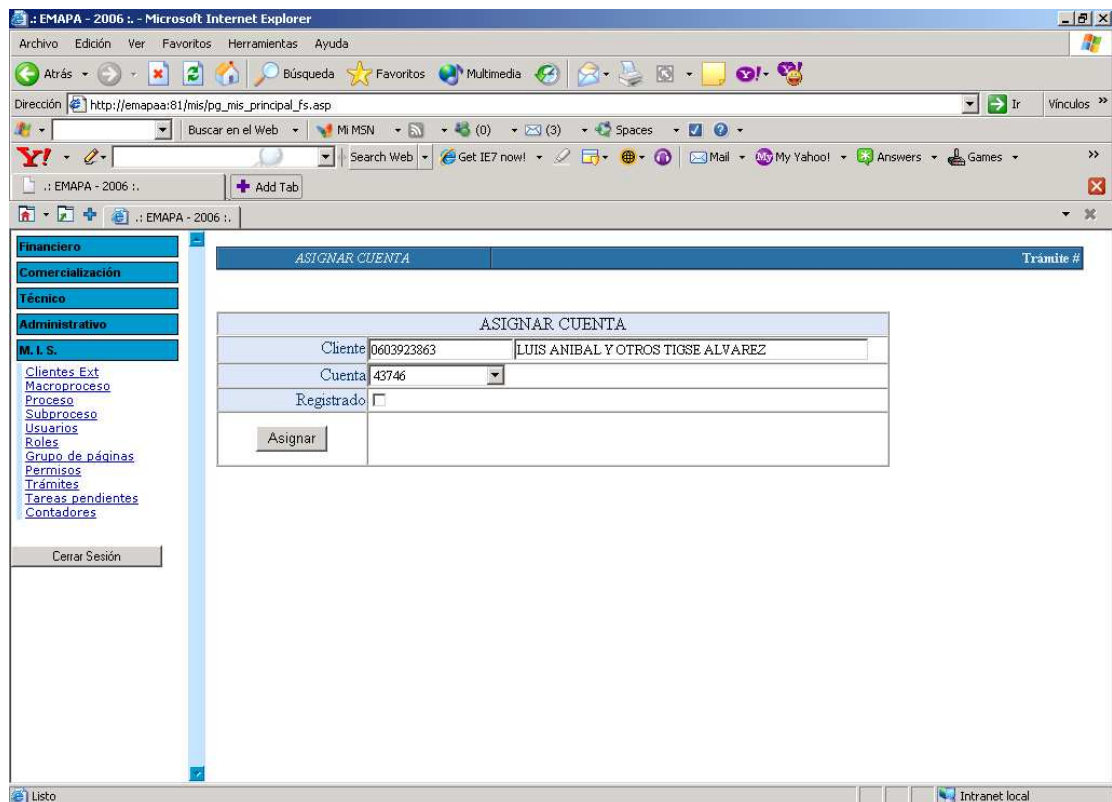
Si decidió aceptar el pago pasaremos al siguiente paso

4. Asignación de Cuenta

En esta página Ud. puede asignar a una cuenta de las que posee el cliente el cargo a pagarse por la factura, para lo que debe seleccionar en el combo el número de cuenta al que desee asignar el valor.

Si el cliente no posee una cuenta registrada, la aplicación en lugar de la cuenta mostrara un mensaje que diga “Cliente sin Cuenta”.

En cualquiera de los dos casos se procederá a registrar, y luego a dar clic sobre el botón Asignar, y pasamos al siguiente punto.



5. Facturación

En esta página Ud. posee información sobre los valores a pagar por el cliente de acuerdo al servicio solicitado.

Aquí Ud. puede seleccionar la forma de pago que el cliente requiere que puede ser en efectivo o a su vez se puede ser crédito para lo que se debe crear un convenio de pago a plazos que se irán cargando a la cuenta que haya seleccionado en el paso anterior para que sea pagada cada mes.

Cabe indicar que si el cliente no posee una cuenta registrada en la empresa, no se puede generar un convenio por lo que la única forma de pago será en efectivo.

VENTA DE AGUA EN TANQUERO Tramite #4202

FACTURA

Numero de Factura	1388435	Fecha Facturacion	07/11/2006 9:21:00
Cliente Externo CI/RUC	0603923863	LUIS ANIBAL Y OTROS TIGSE ALVAREZ	
Pagada	<input type="checkbox"/>		
Forma de Pago	Efectivo		

Guardar

DETALLE DE FACTURA Tramite # 4202

Id	Servicio	Cantidad	Valor Unitario	IVA	Valor Total
4721008	Venta de agua en tanquero	24	0.45		10.8
4721009	Transporte en tanquero	2	10		20

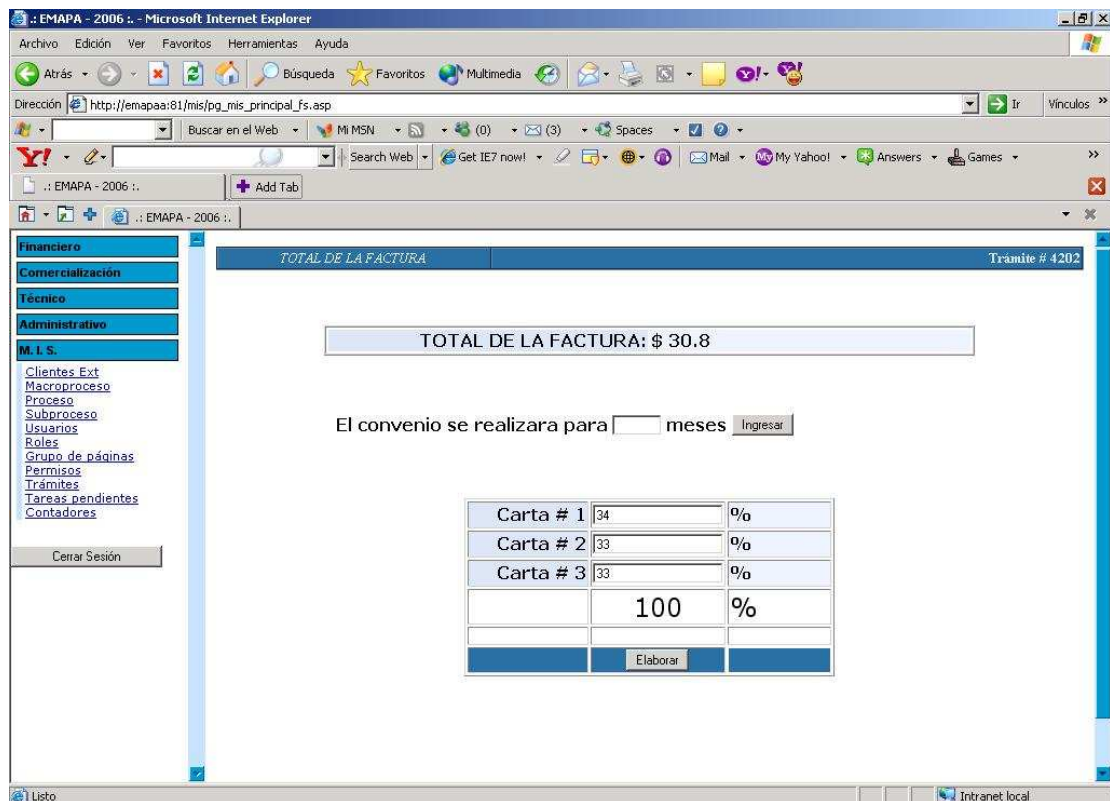
Luego de seleccionar la forma de pago se procederá a Guardar la opción escogida y en caso de ser efectivo, la factura queda lista para su recaudación; en caso de escoger un convenio seguiremos al siguiente paso.

6. Generación de Convenio

Si Ud, ha llegado a este punto quiere decir que el cliente posee un número de cuenta registrado en la empresa y que la forma de pago escogida haya sido crédito.

En este punto Ud. debe ingresar el número de meses convenido por el cliente para realizar el pago por el presente servicio y luego dar click sobre el botón ingresar

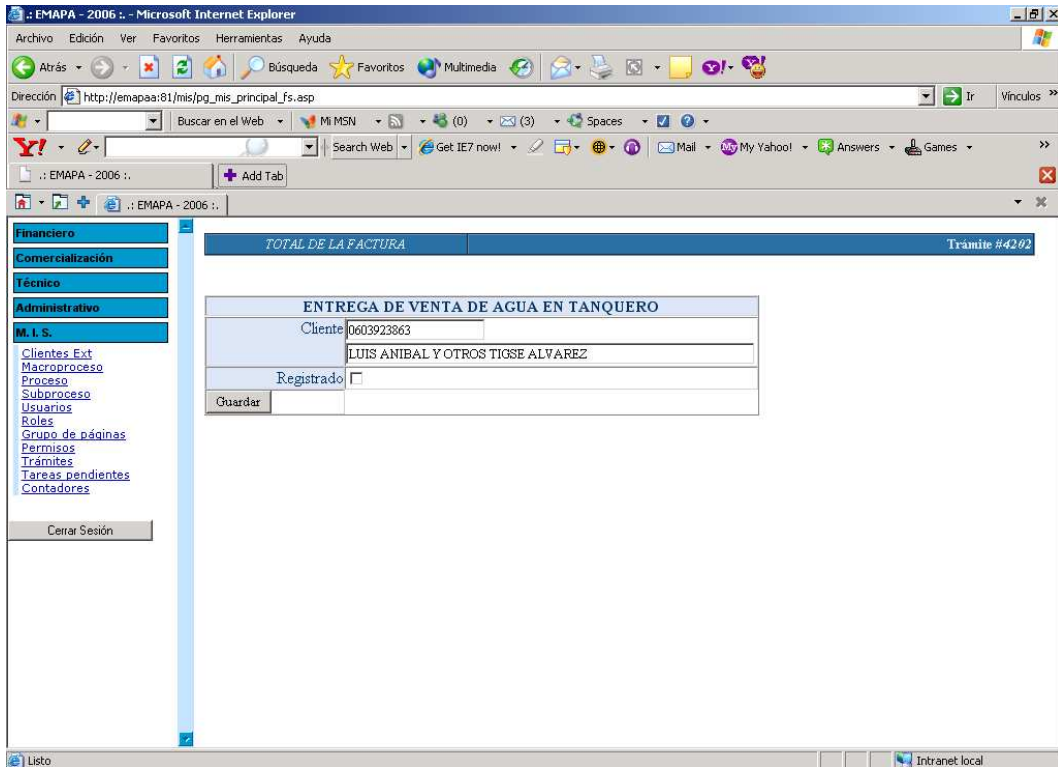
Se desplegara información del valor a cancelar en cada mes incluyendo el interés, si esta de acuerdo con los valores a cancelar deberá dar click sobre el botón elaborar y se creara el convenio.



7. Entrega de Agua

Una vez que se ha pagado por el servicio el cliente deberá presentar la factura para que la entrega de agua se haga efectiva.

Para lo que solamente cuando presente la factura se dará click sobre la opción registrado y luego sobre la opción guardar.



Una vez realizados estos pasos, la aplicación finalizará.

