



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA
E INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES E INFORMÁTICOS**

Tema:

“DESARROLLO DE UN SISTEMA DE RENTAS PARA AGILITAR LOS
TRÁMITES DE ALCABALAS, PLUSVALÍAS, RODAJES, TÍTULOS DE
MERCADOS, MULTAS Y DERECHOS DE INSTALACIÓN DE AGUA
POTABLE EN EL I. MUNICIPIO DEL CANTÓN QUERO”

Proyecto de Pasantía de Grado, previo la obtención del título de Ingeniero en
Sistemas Computacionales e Informáticos

AUTOR: Jorge Augusto Sánchez Villacrés

TUTOR: Ing. Clay Aldás

Ambato - Ecuador

Septiembre 2009

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de investigación sobre el tema: “DESARROLLO DE UN SISTEMA DE RENTAS PARA AGILITAR LOS TRÁMITES DE ALCABALAS, PLUSVALÍAS, RODAJES, TÍTULOS DE MERCADOS, MULTAS Y DERECHOS DE INSTALACIÓN DE AGUA POTABLE EN EL I. MUNICIPIO DEL CANTÓN QUERO”, de Jorge Augusto Sánchez Villacrés, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 57 del Capítulo IV Pasantías, del Reglamento de Graduación de Pregrado de la Universidad Técnica de Ambato.

Ambato septiembre, 2009

EL TUTOR

Ing. Clay Aldás

AUTORÍA

El presente trabajo de investigación titulado: “DESARROLLO DE UN SISTEMA DE RENTAS PARA AGILITAR LOS TRÁMITES DE ALCABALAS, PLUSVALÍAS, RODAJES, TÍTULOS DE MERCADOS, MULTAS Y DERECHOS DE INSTALACIÓN DE AGUA POTABLE EN EL I. MUNICIPIO DEL CANTÓN QUERO”. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato septiembre, 2009

Jorge Augusto Sánchez Villacrés
CC: 1803942026

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo de graduación conformada por los señores docentes Ing. M.Sc. Julio Balarezo e Ing. M.Sc. Javier Sánchez, aprueban el presente trabajo de graduación titulado “DESARROLLO DE UN SISTEMA DE RENTAS PARA AGILITAR LOS TRÁMITES DE ALCABALAS, PLUSVALÍAS, RODAJES, TÍTULOS DE MERCADOS, MULTAS Y DERECHOS DE INSTALACIÓN DE AGUA POTABLE EN EL I. MUNICIPIO DEL CANTÓN QUERO”, presentada por el señor Jorge Augusto Sánchez Villacrés; de acuerdo al Art. 57 del Reglamento de Graduación para obtener el título Terminal del tercer nivel de la Universidad Técnica de Ambato.

Ing. M.Sc. Alexis Sánchez Miño
PRESIDENTE DEL TRIBUNAL

Ing. M.Sc. Julio Balarezo
DOCENTE CALIFICADOR

Ing. M.Sc. Javier Sánchez
DOCENTE CALIFICADOR

DEDICATORIA:

Dedico el presente trabajo a mis padres y demás familiares que siempre creyeron en mí.

Jorge Augusto Sánchez Villacrés

AGRADECIMIENTO:

Agradezco a mis padres y familiares
por el apoyo constante que he
recibido durante todo este tiempo.

Jorge Augusto Sánchez Villacrés

ÍNDICE

Portada	i
Aprobación del tutor	ii
Autoría	iii
Aprobación de la comisión calificadora	iv
Dedicatoria	v
Agradecimiento	vi
Índice	vii
Resumen Ejecutivo	xiv
Introducción	xv

CAPÍTULO I EL PROBLEMA

1.1. Planteamiento del problema	1
1.1.1. Contextualización	1
1.1.2. Análisis crítico	1
1.1.3. Prognosis	2
1.2. Formulación del problema	3
1.2.1. Preguntas directrices	3
1.2.2. Delimitación del problema	3
1.3. Justificación	3
1.4. Objetivos de la investigación	4
1.4.1. Objetivo general	4
1.4.2. Objetivos específicos	4

CAPÍTULO II MARCO TEÓRICO

2.1. Antecedentes investigativos	5
2.2. Fundamentación	5
2.2.1. Fundamentación legal	5
2.2.2. Fundamentación metodológica	7

2.2.3. Fundamentación teórica	7
2.2.3.1. Desarrollo de un Sistema de Rentas	7
2.2.3.2. Base de Datos	8
2.2.3.3. Sistema Gestor de Base de Datos (SGBD)	12
2.2.3.4. PostgreSQL	16
2.2.3.5. Entorno de Desarrollo Integrado (IDE)	19
2.2.3.6. SharpDevelop	21
2.3. Variables	22
2.3.1. Variable independiente	22
2.3.2. Variable dependiente	22
2.4. Hipótesis	22

CAPÍTULO III METODOLOGÍA

3.1. Enfoque	23
3.1.1. Enfoque cualitativo	23
3.1.2. Enfoque cuantitativo	23
3.2. Modalidad básica	23
3.2.1. Investigación de campo	23
3.2.2. Investigación documental-bibliográfica	24
3.2.3. Proyecto factible	24
3.3. Nivel o tipo de investigación	24
3.4. Población y muestra	24
3.4.1. Población	24
3.4.2. Muestra	25
3.5. Recolección de información	25
3.5.1. Plan para recolección de la información	25
3.5.2. Plan para procesamiento de la información	25
3.6. Plan para análisis e interpretación de los resultados	25

CAPÍTULO IV ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. Análisis	26
---------------------	----

4.2. Interpretación	26
---------------------------	----

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones	28
5.2. Recomendaciones	28

CAPÍTULO VI PROPUESTA

6.1. Datos informativos	30
6.2. Antecedentes de la propuesta	30
6.3. Análisis	31
6.3.1. Análisis de factibilidades	31
6.3.1.1. Factibilidad técnica	31
6.3.1.2. Factibilidad operacional	31
6.3.1.3. Factibilidad económica	31
6.3.2. Análisis de requerimientos	32
6.3.3. Recolección de información	32
6.3.4. Alcance del sistema	32
6.3.5. Diagramas UML	32
6.3.5.1. Diagrama de casos de uso	33
6.4. Diseño	68
6.4.1. Diagrama de clases	69
6.4.2. Diagrama de secuencias	71
6.4.3. Diseño de la base de datos	73
6.4.3.1. Diseño lógico	74
6.4.3.2. Diseño físico	75
6.4.4. Diseño de interfaces	76
6.4.4.1. Diseño de interfaces de entrada	76
6.4.4.2. Diseño de interfaces de salida	78
6.5. Desarrollo	80
6.5.1. Desarrollo de la base de datos	80
6.5.1.1. Creación de la base de datos	80

6.5.1.2. Creación de tablas y secuencias	84
6.5.1.3. Descripción de los campos de las tablas	95
6.5.1.4. Creación de procedimientos almacenados	102
6.5.1.5. Usuarios y Privilegios	103
6.5.2. Desarrollo de la aplicación	104
6.5.2.1. Desarrollo de las clases	104
6.5.2.2. Desarrollo de la interfaz	104
6.6. Implantación	106
6.7. Conclusiones y recomendaciones	107
Bibliografía	108
ANEXOS	109
Anexo N° 1. Usuarios y privilegios	110
Anexo N° 2. Algunas clases desarrolladas	114
Anexo N° 3. Manual de instalación	155
Anexo N° 4. Manual de usuario	169

ÍNDICE DE TABLAS

Tabla N° 6.1. Casos de uso para el actor Jefe de Avalúos y Catastros	34
Tabla N° 6.2. Descripción del caso de uso modificar alcabala	34
Tabla N° 6.3. Descripción del caso de uso eliminar alcabala	35
Tabla N° 6.4. Descripción del caso de uso ingresar datos persona	35
Tabla N° 6.5. Descripción del caso de uso modificar datos persona	36
Tabla N° 6.6. Descripción del caso de uso eliminar datos persona	36
Tabla N° 6.7. Descripción del caso de uso generar reporte	37
Tabla N° 6.8. Descripción del caso de uso imprimir	37
Tabla N° 6.9. Descripción del caso de uso liquidar alcabala	38
Tabla N° 6.10. Descripción del caso de uso anular liquidación de alcabala	39
Tabla N° 6.11. Descripción del caso de uso anular pago de alcabala	40
Tabla N° 6.12. Descripción del caso de uso liquidar plusvalía	40
Tabla N° 6.13. Descripción del caso de uso anular pago de plusvalía	41
Tabla N° 6.14. Descripción del caso de uso eliminar plusvalía	41
Tabla N° 6.15. Descripción del caso de uso liquidar rodaje	42

Tabla N° 6.16. Descripción del caso de uso anular pago de rodaje	42
Tabla N° 6.17. Descripción del caso de uso eliminar rodaje	43
Tabla N° 6.18. Descripción del caso de uso emitir título de mercado	43
Tabla N° 6.19. Descripción del caso de uso anular pago de título de mercado	44
Tabla N° 6.20. Descripción del caso de uso eliminar título de mercado	44
Tabla N° 6.21. Descripción del caso de uso emitir multa	45
Tabla N° 6.22. Descripción del caso de uso anular pago de multa	45
Tabla N° 6.23. Descripción del caso de uso eliminar multa	46
Tabla N° 6.24. Descripción del caso de uso emitir título de medidor de agua	46
Tabla N° 6.25. Descripción del caso de uso anular pago de título de medidor de agua.....	47
Tabla N° 6.26. Descripción del caso de uso eliminar título de medidor de agua	48
Tabla N° 6.27. Descripción del caso de uso generar reporte alcabalas	49
Tabla N° 6.28. Descripción del caso de uso generar reporte plusvalías	49
Tabla N° 6.29. Descripción del caso de uso generar reporte rodajes	50
Tabla N° 6.30. Descripción del caso de uso generar reporte mercados	50
Tabla N° 6.31. Descripción del caso de uso generar reporte multas	51
Tabla N° 6.32. Descripción del caso de uso generar reporte medidor de agua	51
Tabla N° 6.33. Descripción del caso de uso imprimir	52
Tabla N° 6.34. Descripción del caso de uso ingresar datos persona	52
Tabla N° 6.35. Descripción del caso de uso modificar datos persona	53
Tabla N° 6.36. Descripción del caso de uso eliminar datos persona	53
Tabla N° 6.37. Descripción del caso de uso configurar base rodaje	54
Tabla N° 6.38. Descripción del caso de uso configurar mora tributaria	54
Tabla N° 6.39. Descripción del caso de uso pagar alcabala	55
Tabla N° 6.40. Descripción del caso de uso pagar plusvalía	56
Tabla N° 6.41. Descripción del caso de uso pagar rodaje	56
Tabla N° 6.42. Descripción del caso de uso pagar título mercado	57
Tabla N° 6.43. Descripción del caso de uso pagar multas	57
Tabla N° 6.44. Descripción del caso de uso pagar título de medidor de agua	58
Tabla N° 6.45. Descripción del caso de uso imprimir título/comprobante	59
Tabla N° 6.46. Descripción del caso de uso generar reporte recaudación alcabalas	59
Tabla N° 6.47. Descripción del caso de uso generar reporte recaudación plusvalías	60
Tabla N° 6.48. Descripción del caso de uso generar reporte recaudación rodajes	60
Tabla N° 6.49. Descripción del caso de uso generar reporte recaudación mercados	61

Tabla N° 6.50. Descripción del caso de uso generar reporte recaudación multas	61
Tabla N° 6.51. Descripción del caso de uso generar reporte recaudación medidores agua.....	62
Tabla N° 6.52. Descripción del caso de uso crear usuario	63
Tabla N° 6.53. Descripción del caso de uso modificar usuario	64
Tabla N° 6.54. Descripción del caso de uso eliminar usuario	64
Tabla N° 6.55. Descripción del caso de uso generar reporte alcabalas	65
Tabla N° 6.56. Descripción del caso de uso generar reporte plusvalías	65
Tabla N° 6.57. Descripción del caso de uso generar reporte rodajes	66
Tabla N° 6.58. Descripción del caso de uso generar reporte mercados	66
Tabla N° 6.59. Descripción del caso de uso generar reporte multas	67
Tabla N° 6.60. Descripción del caso de uso generar reporte medidor agua	67
Tabla N° 6.61. Descripción del caso de uso imprimir	68
Tabla N° 6.62. Descripción de los campos de la tabla “persona”	95
Tabla N° 6.63. Descripción de los campos de la tabla “vendedores”	96
Tabla N° 6.64. Descripción de los campos de la tabla “compradores”	96
Tabla N° 6.65. Descripción de los campos de la tabla “alcabala”	96
Tabla N° 6.66. Descripción de los campos de la tabla “plusvalia”	97
Tabla N° 6.67. Descripción de los campos de la tabla “rodaje”	98
Tabla N° 6.68. Descripción de los campos de la tabla “base_rodaje”	98
Tabla N° 6.69. Descripción de los campos de la tabla “mora tributaria”	99
Tabla N° 6.70. Descripción de los campos de la tabla “comerciante”	99
Tabla N° 6.71. Descripción de los campos de la tabla “mercados”	99
Tabla N° 6.72. Descripción de los campos de la tabla “multas”	100
Tabla N° 6.73. Descripción de los campos de la tabla “medidor”	101
Tabla N° 6.74. Descripción de los campos de la tabla “grupo”	101
Tabla N° 6.75. Descripción de los campos de la tabla “usuario”	102
Tabla N° 6.76. Grupos de usuarios	103

ÍNDICE DE FIGURAS

Figura N° 6.1. Casos de uso para el actor Jefe de Avalúos y Catastros	33
Figura N° 6.2. Casos de uso para el actor Jefe de Rentas	38
Figura N° 6.3. Casos de uso para el actor Jefe de Rentas (continuación)	48
Figura N° 6.4. Casos de uso para el actor Recaudador	55

Figura N° 6.5. . Casos de uso para el actor Administrador	63
Figura N° 6.6. Diagrama de clases	69
Figura N° 6.7. Diagrama de clases (continuación)	70
Figura N° 6.8. Diagrama de secuencia para el caso de uso “Ingresar alcabala”	71
Figura N° 6.9. Diagrama de secuencia para el caso de uso “Liquidar alcabala”	72
Figura N° 6.10. Diagrama de secuencia para el caso de uso “Pagar alcabala”	72
Figura N° 6.11. Diagrama de secuencia para el caso de uso “Generar reporte alcabalas”	73
Figura N° 6.12. Diseño lógico de la base de datos	74
Figura N° 6.13. Diseño físico de la base de datos	75
Figura N° 6.14. Diseño de interfaz de entrada	76
Figura N° 6.15. Interfaz de entrada. Ingreso de alcabala	77
Figura N° 6.16. Interfaz de entrada. Ingreso de persona	77
Figura N° 6.17. Interfaz de entrada. Liquidación de alcabala	78
Figura N° 6.18. Diseño de interfaz de salida	79
Figura N° 6.19. Interfaz de salida. Reporte	79
Figura N° 6.20. Consola de administración pgAdmin III	80
Figura N° 6.21. Ventana de autenticación para ingresar a pgAdmin III	80
Figura N° 6.22. Creación de una nueva base de datos	81
Figura N° 6.23. Creación de la base de datos “rentas”	81
Figura N° 6.24. Base de datos “rentas” creada	82
Figura N° 6.25. Creación de un nuevo esquema	82
Figura N° 6.26. Creación del esquema “esq”	83
Figura N° 6.27. Esquema “esq” creado	83
Figura N° 6.28. Cómo ejecutar código SQL	84
Figura N° 6.29. Ejecución de código SQL	85
Figura N° 6.30. Objetos creados en el esquema	85
Figura N° 6.31. Entorno de desarrollo de Windows Forms de SharpDevelop	105
Figura N° 6.32. Entorno de programación de SharpDevelop	105

RESUMEN EJECUTIVO

El I. Municipio del Cantón Quero, al no poseer ningún tipo de automatización en sus procesos de manejo de trámites y atención a sus usuarios, ha venido realizando un manejo de manera manual con todos los problemas que esto conlleva. Es por eso que el presente proyecto tiene como objetivo solucionar tales problemas mediante el desarrollo de un sistema que automatice el manejo de trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable.

Para llevar a cabo el desarrollo de dicho sistema, primeramente se procedió a estudiar cuál era la situación actual en los departamentos de Avalúos y Catastros, Rentas y Tesorería. Entonces se determinó los requerimientos del sistema. Se realizó el análisis del sistema, de sus factibilidades, requerimientos y alcance, para luego comenzar con el proceso de diseño. Posteriormente se procedió al desarrollo del sistema y realización de pruebas. Finalmente el sistema se puso en funcionamiento.

Cabe mencionar que el software que se utilizó en el desarrollo de la aplicación así como el motor de base de datos son software libre que no requiere el pago de ninguna licencia. Esto es muy positivo para la institución ya que puede hacer uso de un software de calidad sin costo alguno.

El sistema funciona de manera correcta y cumple con los objetivos propuestos: mantener la información íntegra y disponible, facilitar el trabajo al personal y mejorar la atención a los usuarios del cantón.

INTRODUCCIÓN

El presente proyecto tiene como objetivo el desarrollo de un sistema para agilizar los trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable en el I. Municipio del Cantón Quero. Está organizado de la siguiente manera:

Capítulo I titulado “El Problema”, donde se identifica el problema a resolver, se realiza un análisis crítico y se establecen los objetivos.

Capítulo II “Marco Teórico”, que contiene la fundamentación teórica de la investigación

Capítulo III denominado “Metodología”, donde se define cómo y con qué se va a llevar a cabo la investigación.

Capítulo IV titulado “Análisis e interpretación de resultados”, donde se analiza e interpreta los resultados obtenidos en la investigación.

Capítulo V denominado “Conclusiones y recomendaciones”, que contiene las conclusiones que se obtienen de la investigación y el análisis, y también se propone algunas recomendaciones.

Capítulo VI “Propuesta”, contiene la solución que se propone y la forma como se desarrolló.

Finalmente se encuentra la bibliografía que ha servido de apoyo en el desarrollo del proyecto, y los anexos que contienen documentación adicional.

CAPÍTULO I EL PROBLEMA

Tema:

“DESARROLLO DE UN SISTEMA DE RENTAS PARA AGILITAR LOS TRÁMITES DE ALCABALAS, PLUSVALÍAS, RODAJES, TÍTULOS DE MERCADOS, MULTAS Y DERECHOS DE INSTALACIÓN DE AGUA POTABLE EN EL I. MUNICIPIO DEL CANTÓN QUERO”

1.1. Planteamiento del problema

1.1.1. Contextualización

En muchas provincias del país existen municipios en los que la falta de automatización en los trámites y servicios causa que la atención a los usuarios sea ineficiente.

Los municipios de los cantones de la provincia del Tungurahua, especialmente de los más pequeños, no poseen ningún tipo de automatización en sus procesos de manejo de trámites y atención a sus usuarios. El I. Municipio del Cantón Quero no es la excepción. En esta institución se ha venido realizando el manejo de trámites de manera manual con todos los inconvenientes que esto implica.

1.1.2. Análisis crítico

Debido a que el manejo de las Rentas en el I. Municipio del Cantón Quero se ha venido realizando de manera manual, se generan varios inconvenientes como pérdida de tiempo, manejo tedioso de documentos y acumulación de trabajo para el personal.

La búsqueda de información, la realización de cálculos, generación de reportes son efectuadas manualmente por el personal. Esto provoca el consumo de buena parte del tiempo y recursos del personal que podrían ser utilizados en otras actividades más productivas además de causar insatisfacción e impaciencia en los usuarios.

Al mantener la información en medios tradicionales de almacenamiento de información como papel, se desperdicia muchos recursos y espacio. El acceso a dicha información es lento, molesto e ineficiente. Además se pueden producir confusión de documentos y graves pérdidas de información.

La institución piensa abandonar el sistema manual que se ha venido utilizando, y optar por la automatización con el fin de optimizar tiempo y recursos para de obtener un beneficio común.

1.1.3. Prognosis

Si la institución no realiza la automatización, la cantidad de usuarios insatisfechos aumentará y el desempeño del personal será cada vez peor y su trabajo más tedioso e ineficiente. Los usuarios tendrán que invertir más tiempo y paciencia para realizar sus trámites.

El manejo manual de la información podría causar una desorganización de documentos y confusión o pérdida de los mismos. Se gastarán muchos recursos y tiempo.

Si estos inconvenientes no se solucionan el trabajo del personal se acumulará causando tedio y baja productividad. Los usuarios perderán tiempo provocando impaciencia y dando paso a situaciones tensas.

1.2. Formulación del problema

¿Es posible agilizar los trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable con un sistema automatizado, en la I. Municipio del Cantón Quero?

1.2.1. Preguntas directrices

¿Cómo sustituir el manejo manual de los trámites?

¿Qué beneficios tendrá el personal?

¿Serán satisfechas las necesidades de los usuarios?

¿Cómo se mantendrá la información?

1.2.2. Delimitación del problema

La implementación del Sistema de Rentas se va a llevar a cabo en el I. Municipio del Cantón Quero, ubicado en la parroquia La Matriz del cantón Quero, en la provincia del Tungurahua.

El tiempo necesario para realizar dicho proyecto se ha estimado de seis meses.

1.3. Justificación

Los trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable en el I. Municipio del Cantón Quero, se realizan a diario y la cantidad de usuarios es grande, por lo que el manejo manual de dichos trámites es ineficiente.

Para solucionar el problema, es de vital importancia y utilidad el desarrollo de un Sistema que permita realizar los trámites de manera automatizada y rápida de modo que se libere al personal de tareas tediosas y satisfaga a la gran demanda de usuarios.

Con la automatización la búsqueda de información de usuarios, la realización de cálculos, la generación de reportes se realizarán de manera exacta e instantánea por lo que el tiempo para realizar los trámites se reducirá. Y esto en un Cantón donde cada día hay cada vez más usuarios que visitan la institución a diario, será de gran utilidad.

Además al almacenar la información en dispositivos computacionales evitará el manejo excesivo de medios tradicionales de almacenamiento de información como papel, ahorrando recursos en la institución. La información se mantendrá de manera organizada e íntegra.

En virtud de lo antes expuesto, la automatización de Rentas es de mucha importancia pues servirá de gran ayuda a la institución y a la colectividad

1.4. Objetivos de la investigación

1.4.1. Objetivo general

Desarrollar un Sistema de Rentas para agilizar los trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable en el I. Municipio del Cantón Quero.

1.4.2. Objetivos específicos

- Permitir un manejo automatizada de los trámites de Rentas en el I. Municipio del Cantón Quero.
- Facilitar las tareas del personal.
- Satisfacer a toda la demanda de usuarios.
- Mantener la información íntegra y disponible permanentemente.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes investigativos

Después de realizar una búsqueda en los archivos de tesis de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, puedo llegar a la conclusión de que no existen tesis con temas parecidos al que se trata en el presente trabajo.

2.2. Fundamentación

2.2.1. Fundamentación legal

I. Municipio del Cantón Quero

El I. Municipio del Cantón Quero aparece con la cantonización de la ciudad de Quero el día 27 de Julio de 1972, bajo el gobierno del General Guillermo Rodríguez Lara, según decreto ejecutivo No. 681.

El I. Municipio del Cantón Quero tiene la siguiente base legal:

- Art. 228 de la Constitución Política de la República que consagra la autonomía económica y administrativa de los municipios.
- Art. 17 de la Ley de Régimen Municipal, que determina que las municipalidades son autónomas y que salvo lo prescrito por la Constitución y esta Ley, ninguna función del Estado ni autoridad extraña a la municipalidad podrá interferir en su administración propia; estándole especialmente prohibido, entre otros aspectos a: “derogar, reformar o

suspender la ejecución de las Ordenanzas, Reglamentos, Resoluciones o Acuerdos de las autoridades municipales.

- Numeral 1 del Art.64 de la Ley de Régimen Municipal, por el que se faculta a los Concejos, normar a través de Ordenanzas, dictar Acuerdos o Resoluciones; determinar la política a seguirse y fijar las metas en cada uno de los ramos propios de su administración.
- Art.168 de la Ley de Régimen Municipal, que establece que la organización administrativa estará de acuerdo con las necesidades peculiares que deba atender y con los servicios públicos a prestarse, de acuerdo con los recursos financieros disponibles y consecuentemente a establecer una estructura de trabajo que permita el cumplimiento de objetivos.
- Art.169, que establece que el Reglamento Orgánico Funcional, determinará la estructura administrativa de cada municipalidad y considerando de que cada dependencia forme parte de una organización racionalmente integrada desde el punto de vista de la división del trabajo.
- Art. 172, de la Ley de Régimen Municipal que determina que la denominación de las dependencias de la administración municipal se ajustará a la nomenclatura de Dirección, Departamento y Sección, según sea la complejidad de la labor encomendada; y que habrá tantas Direcciones, cuantas convenga a la mejor y más racional agrupación por funciones afines que competan a la administración; y finalmente que la organización de cada Dirección, Departamento o Sección, constará en los respectivos Orgánicos Funcionales que serán aprobados por el Concejo Municipal.
- Art. 173 y 174 de la Ley de Régimen Municipal, que establece la estructura administrativa básica de Servicios Públicos: Obras Públicas, Financiera, Administrativa, Higiene y Salubridad y Educación Cultural, ajustada a las características propias de cada municipio, pudiendo refundirse dependencias o establecerse otras no previstas que aseguren una racional división del trabajo.

- Art. 179 de la Ley de Régimen Municipal, que establece que cada Dirección ejecutará el trabajo que le corresponde, por intermedio de los Departamentos o Secciones que para el efecto se crearen, con arreglo a la complejidad y volumen de trabajo y en consideración a los sistemas de trabajo por los que se vayan a ejecutar las obras o prestarse los servicios.

Software

El Sistema Gestor de Base de Datos que se utilizará es PostgreSQL 8.3 y la herramienta para desarrollar la aplicación es SharpDevelop 2.2. Las dos aplicaciones mencionadas no requieren pago de licencias pues son software libre.

2.2.2. Fundamentación metodológica

La metodología que se usará es el Modelo en Cascada que consiste en:

- Análisis de requerimientos.
- Diseño del Sistema.
- Diseño del programa.
- Codificación.
- Pruebas.
- Operación y mantenimiento.

2.2.3. Fundamentación teórica

2.2.3.1. Desarrollo de un Sistema de Rentas

El sistema de Rentas en el I. Municipio del Cantón Quero consiste en automatizar los trámites que se han venido realizando manualmente. Dicha automatización permitirá manejar, registrar y mantener de forma organizada la información concerniente a trámites de alcabalas, plusvalías, rodajes, títulos de mercados,

multas y derechos de instalación de agua potable que realizan los usuarios diariamente.

Los usuarios del Sistema dispondrán de una interfaz amigable y fácil de manejar. Dispondrán de la información que requieran instantáneamente para realizar los trámites de manera ágil y oportuna.

Para lograr la automatización de los trámites mencionados se requerirá de una Base de Datos y de una Herramienta de Desarrollo de Software (Entorno de Desarrollo Integrado - IDE) para la creación de la aplicación.

La aplicación será desarrollada utilizando programación por capas. El sistema tendrá tres capas:

1. Capa de presentación.
2. Capa de lógica de negocio.
3. Capa de datos.

2.2.3.2. Base de Datos

Una base de datos es un conjunto de datos relacionados (es decir que pertenecen a un mismo contexto) y almacenados sistemáticamente para su posterior uso. Se consideran como base de datos a una biblioteca pues está compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Hoy en día debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos. En este sentido una base de datos sería un conjunto de datos almacenados en un soporte informático de acceso directo.

Características de las Bases de Datos

- Versatilidad para representar la información. Ofrecer diferentes visiones de la información que se almacena en la BD.
- Desempeño. Respuesta en un tiempo adecuado, permitiendo el acceso simultáneo al mismo o diferente datos.
- Mínima redundancia.
- Capacidad de acceso. Responder adecuadamente a consultas previstas e imprevistas.
- Simplicidad. Los cambios en los requerimientos no deben suponer grandes cambios en el modelo de datos.
- Seguridad. Proteger los datos contra destrucción o pérdidas totales y/o parciales, causadas por el entorno (fuego, inundación, etc.), fallos del sistema, accesos no autorizados a la BD, etc.
- Privacidad. Impedir que personas no autorizadas tengan acceso a los datos.
- Integridad. Los datos deben ser correctos y deben corresponder a los requerimientos. La integridad debe hacer frente a fallos de hardware o software, o de acceso concurrente.

Ventajas de las Bases de Datos

Las bases de datos ofrecen muchas ventajas frente a los sistemas clásicos de ficheros, gracias a que se basan en una estructura de datos integrada y centralizada, eliminando así los problemas de redundancia y control de los datos.

Las ventajas de los sistemas de bases de datos son, entre otras, las siguientes:

- Independencia de los datos respecto a los tratamientos y viceversa. Esto hace que los cambios de los programas no afectan al diseño lógico y/o físico de la base de datos. Así mismo el aumento o disminución de la información, los cambios en la estructura física, etc., no implica alteración en los programas. Con esto se evita volver a programar las aplicaciones cuando se producen cambios en los datos.

- Mejor disponibilidad de los datos para los usuarios. Los usuarios no son propietarios de los datos, sino que estos son compartidos entre todos los usuarios y aplicaciones. Existe pues una mejor disponibilidad de los datos para todos los que los requieran, siempre que estén autorizados para acceder a los mismos.
- Eficiencia en la entrada y validación de los datos al sistema.
- Ahorro de recursos. Al almacenar los datos en almacenamiento secundario (discos magnéticos) ya no se requiere de medios de almacenamiento tradicional como el papel que ocupa mucho espacio físico y conlleva gastos.
- Reducción del espacio de almacenamiento. La desaparición (o disminución) de las redundancias y la aplicación de técnicas de compactación permite a las bases de datos ocupar menos espacio en almacenamiento secundario (disco magnético).

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.-** Bases de datos de sólo lectura para almacenar datos históricos.
- **Bases de datos dinámicas.-** En estas bases de datos la información almacenada va cambiando con el tiempo, permitiendo operaciones como ingreso, eliminación y actualización de datos, así como operaciones de consulta.

Según el contenido

- **Bases de datos bibliográficas.-** Estas bases de datos únicamente almacenan surrogantes (representantes) de la fuente primaria, que permite localizarla. Contienen resúmenes o extractos, cifras o números.
- **Bases de datos de texto completo.-** Almacenan las fuentes primarias, por ejemplo, todo el contenido de un conjunto de libros o revistas.
- **Directorios.-** Como las guías telefónicas en formato electrónico.
- **Bases de datos o "bibliotecas" de información Biológica.-** Almacenan información de ciencias de la vida o médicas.

Según el modelos de bases de datos

Se puede también clasificar a las bases de datos según el modelo de administración de datos. Un modelo de datos es una descripción de un contenedor de datos y de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos son abstracciones que permiten la implementación de un sistema eficiente de base de datos, por ejemplo algoritmos y conceptos matemáticos.

- **Bases de datos jerárquicas.-** Almacenan la información en una estructura jerárquica parecido a un árbol visto al revés, en donde un nodo padre de información (raíz) puede tener varios hijos (hojas).
- **Base de datos de red.-** Parecido al modelo jerárquico con la diferencia de que se permite que un mismo nodo tenga varios padres. Este modelo es usado generalmente por programadores más que por usuarios finales.
- **Base de datos relacional.-** Es el más usado hoy en día. Permite modelar problemas reales y administrar datos dinámicamente. Propuesto por Edgar Frank Codd pronto se convirtió en un paradigma en los modelos de base

de datos. Se fundamenta en el uso de relaciones que son conjuntos de datos llamados tuplas.

- En este modelo la forma de almacenamiento de los datos no tiene importancia. La información puede ser recuperada o almacenada mediante consultas que se pueden construir con el lenguaje estándar SQL (Structured Query Language o Lenguaje Estructurado de Consultas).
- **Bases de datos multidimensionales.-** Permiten la creación de Cubos OLAP. Son similares a las bases de datos relacionales, de las que se diferencian a un nivel conceptual.
- **Bases de datos orientadas a objetos.-** Se basa en modelos informáticos orientados a objetos. Almacena objetos completos (estado y comportamiento). Estas bases de datos incorporan conceptos del paradigma de objetos tales como: encapsulación, herencia y polimorfismo.
- **Bases de datos documentales.-** Permiten la indexación a texto completo y realizar búsquedas potentes.
- **Base de datos deductivas.-** Permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos

2.2.3.3. Sistema Gestor de Bases de Datos (SGBD)

Un Sistema Gestor de Base de Datos (SGBD) es un software que sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, Provee de un conjunto de herramientas que suministra al administrador, analistas, programadores y usuarios, los medios necesarios para describir, recuperar y manipular los datos almacenados en la BD, manteniendo la seguridad, integridad y confidencialidad de los mismos.

Objetivos de un Sistema Gestor de Bases de Datos

Los principales objetivos son:

- Evitar la redundancia de los datos.
- Proveer mecanismos de seguridad de los datos y la privacidad. Estos mecanismos de seguridad pueden ser contra: accesos indebidos o no autorizados a la BD, destrucción causada por el entorno, fallos del propio sistema (hardware, software).
- Asegurar la independencia de los programas y los datos. Esto permite la estructura de la base de datos (esquema) sin necesidad de modificar las aplicaciones que usan esos datos.
- Garantizar la integridad.
- Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas imprevistas.

Funciones de un Sistema Gestor de Bases de Datos

Las principales funciones que debe realizar un Sistema Gestor de Bases de Datos son:

- **Definición de los datos.-** Mediante el Lenguaje de Definición de Datos el SGBD permite describir y definir los esquemas de la base de datos. Este lenguaje debe permitir:
 - Crear y modificar objetos conceptuales.
 - Describir estructuras físicas.
 - Definir y modificar vistas.
 - Borrar objetos conceptuales.

El Lenguaje de Definición de Datos está compuesto por un conjunto de comandos que actúan sobre los objetos conceptuales y sobre los objetos físicos, que en su conjunto se les conoce con el nombre de diccionario de datos.

- **Manipulación de los datos.-** Permite el intercambio de datos entre los usuarios y la base de datos. Esto se realiza mediante el Lenguaje de Manipulación de Datos que es un conjunto de comandos que permite realizar operaciones como consultas, inserciones, modificaciones y borrado de los datos.

Lenguaje de Manipulación de Datos funciona de forma diferente, según el modelo de datos: en los modelos en red y jerárquicos se debe indicar los datos que se requieren y como se van a obtener esos datos; en el modelo relacional se indica que datos se quieren pero no la forma de obtenerlos.

- **Garantizar la seguridad e integridad de los datos.-** Los usuarios deben estar autorizados para realizar las operaciones correctas sobre la base de datos. Para ello se debe:
 - Controlar a los usuarios que acceden a la base de datos determinando que operaciones que están autorizados a realizar.
 - Validar las operaciones realizadas con los datos. Esto se consigue con las restricciones de integridad, por ejemplo las restricciones de integridad referencial.
 - Proteger los datos contra los accesos malintencionados y los fallos.
- **Gestión de las transacciones y el acceso concurrente.-** Una transacción es un conjunto de órdenes o comandos que son tratados como una unidad, lo que implica que una transacción solo se puede ejecutar completamente o ser anulada. Para controlar la ejecución simultánea de transacciones es necesario una buena gestión de acceso concurrente.

- **Capacidad de auditoria.-** El SGBD debe proveer al administrador de la base de datos información sobre quien accede a la base de datos y que operaciones realiza.
- Garantizar un tiempo de respuesta idóneo.

Ejemplos de Sistemas Gestores de Base de Datos

En el mercado se dispone de una multitud de Sistemas Gestores de Base de Datos, entre los cuales remencionan algunos a continuación:

SGBD libres

- PostgreSQL.
- MySQL.
- Firebird.
- SQLite.
- DB2.
- Apache Derby.

SGBD comerciales

- dBase.
- Fox Pro.
- IBM Informix.
- MAGIC.
- Microsoft Access.
- Microsoft SQL Server.
- Open Access.
- Oracle.
- Paradox.

- PervasiveSQL.
- Sybase.
- WindowBase.

2.2.3.4. PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS). Al igual que otros proyectos open source (código abierto), el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation. En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continuó el desarrollo del código de Ingres para crear un sistema de bases de datos objeto-relacionales llamado Postgres. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL, tras un breve período como Postgres95.

He aquí un resumen de la evolución de PostgreSQL:

- Ingres (1977-1985) Universidad de Berkeley.
- Postgres (**Posterior a Ingres**) (1986–1994), Michael Stonebraker.
- Postgres95 (1995) Andrew Yu & Jolly Chen publican el código en Internet.
- PostgreSQL 6 (1996–1999) PostgreSQL Global Development Group.
- PostgreSQL 7 (1999–2004) Mejoras en el rendimiento.
- PostgreSQL 8 (2005).

PostgreSQL es un proyecto open source, o de código abierto. Al ser de código abierto se puede obtener el código fuente, usar el programa, y modificarlo libremente sin las limitaciones del software propietario.

PostgreSQL es considerado como el sistema de bases de datos de código abierto más avanzado que existe. Las características que posee son equiparables a los productos comerciales de alta calidad tales como Oracle o DB2.

Características de PostgreSQL

Entre las características que posee, a partir de PostgreSQL 7.1.x. están las siguientes:

- **Multiplataforma.-** PostgreSQL puede funcionar sobre varios Sistemas Operativos como Linux o Windows.
- **DBMS Objeto-Relacional.-** PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas; por lo que es capaz de manejar consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays.
- **Altamente Extensible.-** PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- **Soporte SQL Comprensivo.-** PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- **Integridad Referencial.-** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

- **API Flexible.-** La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
- **Lenguajes Procedurales.-** PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle: PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- **MVCC.-** MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Con esto se evita que un usuario que requiera de una lectura de los datos tenga que esperar para acceder a la información debido a que otros usuarios están escribiendo en la base de datos.

Se puede decir que MVCC es mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. PostgreSQL logra esto manteniendo una ruta a todas las transacciones realizadas por los usuarios de la base de datos; siendo capaz de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

- **Cliente/Servidor.-** PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Existe un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectarse.
- **Write Ahead Logging (WAL).-** Esta característica incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que si la base de datos cae,

existirá un registro de las transacciones a partir del cual se podrá restaurar la base de datos.

Ventajas de usar PostgreSQL

Entre las ventajas de usar PostgreSQL están las siguientes:

- Instalación ilimitada. No existe costo asociado a la licencia del software.
- Estabilidad y confiabilidad. PostgreSQL no sufre caídas en varios años de operación de alta actividad.
- Diseñado para ambientes de con gran volumen de datos.
- Se puede disponer de varias herramientas gráficas de diseño y administración de base de datos. Entre las herramientas gráficas de alta calidad para administrar las bases de datos están pgAdmin, pgAccess, etc.; y para hacer diseño de bases de datos se tiene Tora, Data Architect, etc.
- El código fuente está disponible para todo mundo sin costo alguno. En todo el mundo todos los días comunidades de profesionales extienden y mejoran PostgreSQL; y estas nuevas características pueden ser disfrutadas por el público en general.
- Es multiplataforma. PostgreSQL está disponible para casi cualquier Unix, y en versión nativa para Windows

2.2.3.5. Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado (en inglés Integrated Development Environment - IDE) es un software que incorpora un conjunto de herramientas para programación en uno o varios lenguajes, que permiten el desarrollo de varios tipos de aplicaciones de software.

Un IDE integra en una sola aplicación un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs proveen un

ambiente de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc.

Componentes de un IDE

Un IDE incorpora varios componentes como:

- Editor de texto.
- Compilador.
- Intérprete.
- Herramientas de automatización.
- Depurador.
- Herramientas para la construcción de interfaces gráficas de usuarios.

Ejemplos de IDE

Algunos de los entornos de desarrollo integrado que existen son:

- Delphi y Turbo Delphi.
- Eclipse.
- JDeveloper de Oracle Corporation.
- KDevelop.
- MS Visual Studio .NET y Visual Studio Express de Microsoft.
- MonoDevelop.
- NetBeans.
- PowerBuilder de Sybase.
- #develop (SharpDevelop).
- Turbo C y Turbo C++ de Borland.
- Turbo Pascal de Borland.

2.2.3.6. SharpDevelop

SharpDevelop es un entorno de desarrollo integrado libre para los lenguajes de programación C#, Visual Basic .NET y Boo. Puede ser usado por programadores de dichos lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft, el Microsoft Visual Studio. La versión 1.1 de la aplicación puede importar proyectos de Visual Studio .NET. La versión 2.0 permite editarlos directamente. Funciona en plataforma Windows y tiene licencia LGPL.

Características de SharpDevelop

Entre las características de SharpDevelop están:

- Es Open Source (código abierto) con licencia LGPL.
- Posee un diseñador de Windows Forms.
- Completado de código. Soporta el uso de la combinación de teclas Ctrl + Espacio.
- Incorpora un depurador propio.
- Permite convertir aplicaciones en C# hacia Visual Basic .NET y viceversa. Además permite una conversión hacia Boo.
- Escrito enteramente en C#.
- Compilación de código directamente dentro del entorno de desarrollo integrado.
- Vista previa de documentación XML.
- Gran integración con plantillas a la hora de añadir o crear ficheros, proyectos o compiladores.
- Escritura de código C#, ASP.NET, ADO.NET, XML y HTML.
- Coloreado de sintaxis para los lenguajes C#, HTML, ASP, ASP.NET, VBScript, Visual Basic .NET, y XML.
- Llaves inteligentes en la escritura de código.
- Gestión de marcadores (favoritos).
- Soporte para plantillas de código.
- Es extensible mediante herramientas externas o complementos

2.3. Variables

2.3.1. Variable independiente

Desarrollo de un Sistema de Rentas.

2.3.2. Variable dependiente

Para agilizar trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable en el I. Municipio del Cantón Quero.

2.4. Hipótesis

El desarrollo del Sistema de Rentas en el I. Municipio del Cantón Quero permitirá agilizar los trámites que se realizan, haciendo de la atención a los usuarios una tarea rápida y eficiente. Además permitirá mantener la información íntegra y disponible. El beneficio será tanto para la institución como para la comunidad.

CAPÍTULO III METODOLOGÍA

3.1. Enfoque

La investigación se desarrollará basándose en un enfoque cuali-cuantitativo, pues así lo determina la naturaleza del problema y las características del objeto de estudio.

3.1.1. Enfoque cualitativo

El presente proyecto requiere de una investigación cualitativa porque se pondrá énfasis en los procesos y en la observación crítica, para la comprensión del fenómeno a estudiar.

3.1.2. Enfoque cuantitativo

La investigación tendrá también un enfoque cuantitativo pues se buscarán las causas que originaron el problema, y se explicará los hechos. Además se pondrá énfasis en los resultados.

3.2. Modalidad básica

3.2.1. Investigación de campo

Utilizaré la investigación de campo porque para la realización del proyecto necesitare de información de primera mano, en el lugar mismo de los hechos. Es decir, pretendo acercarme a la realidad de la institución, para observar y recavar información concreta.

3.2.2. Investigación documental-bibliográfica

Aplicaré la investigación bibliográfica ya que a través de ella obtendré fuentes de información y fundamentos teóricos para la utilización y manejo de las herramientas para el desarrollo del sistema.

3.2.3. Proyecto factible

Mi proyecto es factible de realizar porque su intención es la de sondear la realidad mediante la evaluación, el conocimiento de la naturaleza del problema y la realización del planteamiento, que permitan la ejecución y aplicación del proyecto.

3.3. Nivel o tipo de investigación

La investigación iniciará en un nivel exploratorio porque permite evaluar, definir y contextualizar la situación del problema; posteriormente, se realizará la investigación a un nivel descriptivo que facilitará un análisis crítico e identificación de las variables. Se pretende llegar finalmente al nivel explicativo con la comprobación de la hipótesis.

3.4. Población y muestra

3.4.1. Población

La población a investigar esta constituida por el personal que realiza los trámites de quien se obtendrá información para la realización del sistema. El personal pertenece a las áreas de Avalúos y Catastros, Rentas y Tesorería.

3.4.2. Muestra

Debido a que la población es pequeña todos los integrantes pasan a formar parte de la muestra.

3.5. Recolección de información

3.5.1. Plan para recolección de la información

Para una recolección óptima de la información de campo, se realizará lo siguiente: Observación y entrevista.

Con la observación se pretende conocer el proceso para realizar los trámites municipales. Con la entrevista se preguntará al personal sobre el funcionamiento de dichos trámites.

3.5.2. Plan para procesamiento de la información

Una vez realizada la observación y la entrevista se obtendrá información, la cual será sometida a:

- Revisión crítica.
- Organización.
- Procesamiento.

3.6. Plan para análisis e interpretación de los resultados

Luego de haber procesado la información y obtenido resultados, se procederá al análisis e interpretación de los mismos basándose en la fundamentación teórica; con lo cual se puede comprobar la hipótesis que se planteó.

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Para obtener los resultados se ha observado minuciosamente el manejo de los trámites y se ha realizado entrevistas con el personal de los departamentos de Avalúos y Catastros, Rentas y Tesorería. Con la información así adquirida se puede analizar e interpretar lo siguiente.

4.1. Análisis

El manejo de los trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable, es manual. Aunque se usa calculadora y la herramienta Excel los cálculos se realizan lentamente. La búsqueda de información es tediosa. Al mantener la información en papel, se desperdicia muchos recursos y espacio.

Todo esto genera varios inconvenientes como pérdida de tiempo, manejo ineficiente y graves pérdidas de la información, y se acumula el trabajo del personal.

4.2. Interpretación

La única forma en la que la institución pueda librarse de los problemas antes mencionados es automatizando el manejo de los trámites. Es por esto que la institución ha venido pensando en el desarrollo de un sistema con el fin optimizar tiempo y recursos, y mejorar la atención a los usuarios. Por lo tanto el desarrollo del presente sistema será la solución, anhelada por la institución, a sus inconvenientes.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El I. Municipio del Cantón Quero tiene muchos inconvenientes en los departamentos de Avalúos y Catastros, Rentas y Tesorería. Entre tales inconvenientes están pérdida de tiempo, manejo ineficiente y graves pérdidas de la información.
- Estos problemas son producto del manejo manual de los trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable.
- La solución a tales inconvenientes es la automatización del manejo de los trámites mencionados.
- El desarrollo de un sistema para el manejo de trámites ayudará al personal en sus labores cotidianas así como permitirá mejorar la atención a la comunidad.

5.2. Recomendaciones

- Se recomienda a la Institución seguir adelante en su deseo de abandonar el manejo manual de los trámites, y optar por la automatización.
- La Institución debería apoyar en todo lo necesario al desarrollo de un sistema para el manejo de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable.

- Se recomienda el uso de software libre, que no requiere de pago de licencias, como una ayuda a la economía de la Institución.

CAPÍTULO VI

PROPUESTA

6.1. Datos informativos

Institución:

Ilustre Municipio del Cantón Quero

Ubicación:

Av. 17 de Abril s/n y García Moreno

Tipo de institución:

Pública

6.2. Antecedentes de la propuesta

El manejo de trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua potable en el I. Municipio del Cantón Quero se ha venido realizando de manera manual. La información se mantiene en medios tradicionales de almacenamiento de información como papel, por lo que el acceso a la información es lento, además de que se pueden producir confusión de documentos y pérdidas de información.

Los cálculos se realizan manualmente o con la utilización ocasional del programa Excel, la emisión de títulos se los realiza con esta misma herramienta. La búsqueda de información y la generación de reportes son efectuadas de manera lenta por el personal.

Como consecuencia el trabajo del personal se acumula y los usuarios pierden tiempo. Por ello se concluye que es de vital importancia y utilidad la automatización de estos procesos de modo que se libere al personal de tareas tediosas y satisfaga a los usuarios.

6.3. Análisis

6.3.1. Análisis de factibilidades

6.3.1.1. Factibilidad técnica

El sistema es viable técnicamente, pues la institución posee hardware y software adecuados para su funcionamiento. Además la herramienta de desarrollo y el Motor de Base de Datos se adecuan al entorno.

6.3.1.2. Factibilidad operacional

La institución se compromete a apoyar en lo necesario para el desarrollo del sistema, el personal está abierto y dispuesto a colaborar; por lo que es viable operacionalmente.

6.3.1.3. Factibilidad económica

Debido a que la herramienta de desarrollo SharpDevelop 2.2 y el Motor de Base de Datos PostgreSQL 8.3 no requieren el pago de licencias, se obtiene un ahorro de dinero.

La institución posee, tanto en el servidor como en las máquinas clientes, hardware de buenas características y software legal; por lo que no gastará recursos económicos a este respecto.

6.3.2. Análisis de requerimientos

Los departamentos de Rentas, Avalúos y Catastros, y Tesorería del I. Municipio del Cantón Quero requieren mejorar el manejo de trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de agua

potable, para lo cual se ha optado por la automatización con el fin de optimizar tiempo y recursos.

6.3.3. Recolección de información

Para obtener la información se recurrió a la observación y análisis del manejo de los trámites, de los formularios, títulos de crédito y otros documentos, y también de los cálculos realizados manualmente o en el programa Excel.

6.3.4. Alcance del sistema

El sistema presta servicios a los siguientes departamentos:

- Avalúos y Catastros
- Rentas
- Tesorería (Recaudación)

Además consta de un módulo para Administración (Jefe del Centro de Cómputo).

6.3.5. Diagramas UML

UML (Unified Modeling Language) es un estándar para el modelado de aplicaciones construidas mediante orientación a objetos. Entre los diagramas UML están: diagrama de casos de uso, de clases, objetos, secuencia, colaboración, estados, actividades, componentes, etc. En el presente trabajo se utilizan, en las etapas de análisis y de diseño, los diagramas de casos de uso, de clases y de secuencias.

6.3.5.1. Diagrama de casos de uso

Permite modelar la funcionalidad del sistema con descripciones de las acciones ejecutadas para obtener un resultado.

Actores

Un actor es alguien o algo que hace uso del sistema. Los actores identificados son:

- Jefe de Avalúos y Catastros.
- Jefe de Rentas.
- Recaudador.
- Administrador.

Caso de uso

Un caso de uso es una funcionalidad completa tal y como la percibe un actor. A continuación están los diagramas con los casos de uso y su descripción.

- **Casos de uso para el actor Jefe de Avalúos y Catastros:**

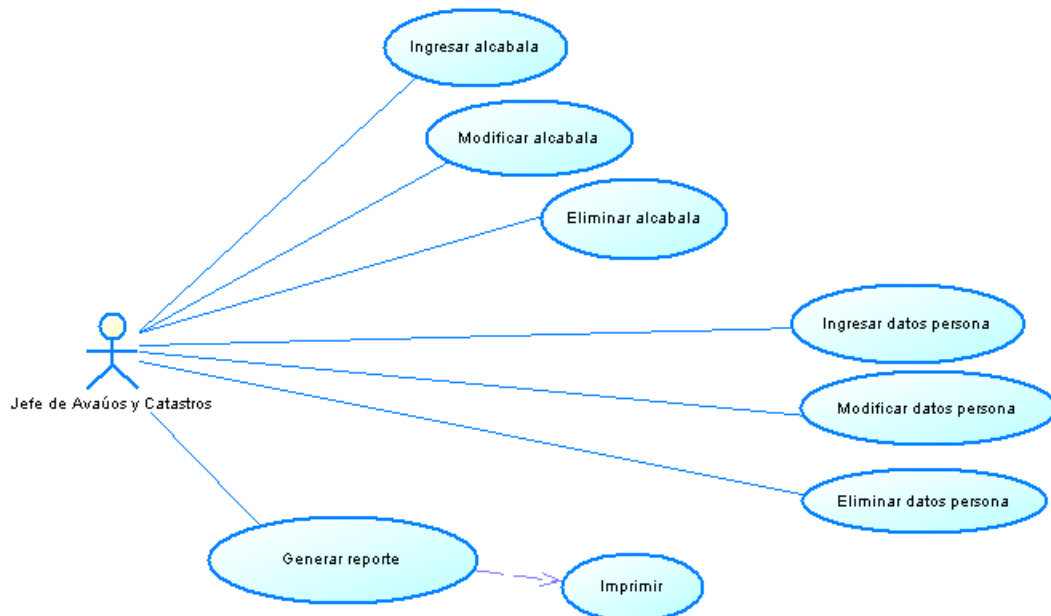


Figura N° 6.1. Casos de uso para el actor Jefe de Avalúos y Catastros

Descripción de los casos de uso:

Caso de uso	Ingresar alcabala
Actor	Jefe de Avalúos y Catastros
Descripción	El Jefe de Avalúos y Catastros ingresa los datos de una alcabala, para su posterior

	liquidación en el Dpto. de Rentas.	
Precondición	El actor conoce los datos de los compradores y vendedores, el objeto de venta, su ubicación y su valor (avalúo).	
Secuencia normal	Paso	Acción
	1	El actor elije la opción ingresar alcabala.
	2	El sistema visualiza la ventana de ingreso de alcabala.
	3	El actor teclea los datos de los compradores y vendedores, el objeto de venta, su ubicación y su valor.
	4	El actor acepta el ingreso de los datos.
Poscondición	El sistema ingresa los datos y la ventana se cierra.	
Excepciones	Paso	Acción
	4	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.1. Descripción del caso de uso ingresar alcabala

Caso de uso	Modificar alcabala	
Actor	Jefe de Avalúos y Catastros	
Descripción	El actor modifica los datos de una alcabala ingresada previamente.	
Precondición	El actor conoce el número de la alcabala a modificar. Dicha alcabala no debe estar aún liquidada.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción modificar alcabala.
	2	El sistema muestra la ventana para modificar alcabala.
	3	El actor ingresa el número de alcabala a buscar.
	4	El sistema muestra los datos de la alcabala especificada.
	5	El actor modifica los datos que requiere.
	6	El actor acepta la modificación de los datos.
Poscondición	El sistema modifica los datos y la ventana se cierra.	
Excepciones	Paso	Acción
	4	Si la alcabala especificada ya ha sido liquidada en el Dpto. de Rentas, el sistema no muestra datos y envía un mensaje advirtiendo que no se puede realizar la modificación.
	6	Si los datos modificados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.2. Descripción del caso de uso modificar alcabala

Caso de uso	Eliminar alcabala	
Actor	Jefe de Avalúos y Catastros	
Descripción	El actor elimina una alcabala.	
Precondición	El actor conoce el número de la alcabala a eliminar. Dicha alcabala no debe estar aún liquidada.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción eliminar alcabala.
	2	El sistema muestra la ventana para eliminar alcabala.
	3	El actor ingresa el número de alcabala a eliminar.
	4	El sistema muestra los datos de la alcabala especificada.
	5	El actor manda a eliminar la alcabala.
	6	El sistema muestra un mensaje de advertencia.
7	El actor confirma la eliminación de la alcabala.	
Poscondición	El sistema elimina la alcabala y se cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la alcabala especificada ya ha sido liquidada en el Dpto. de Rentas, el sistema no muestra datos y envía un mensaje advirtiendo que no se puede eliminar.

Tabla N° 6.3. Descripción del caso de uso eliminar alcabala

Caso de uso	Ingresar datos persona	
Actor	Jefe de Avalúos y Catastros	
Descripción	El actor ingresa datos de una nueva persona al sistema.	
Precondición	El actor conoce los datos de la persona, la cual no debe estar ya en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción personas.
	2	El sistema muestra la ventana con el listado de las personas.
	3	El actor elige la opción ingresar nueva persona.
	4	El sistema muestra la pantalla para ingreso de una nueva persona.
	5	El actor teclea los datos de la persona.
6	El actor acepta el ingreso de los datos.	
Poscondición	El sistema ingresa los datos de la persona.	
Excepciones	Paso	Acción
	6	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.
	6	Si la persona ya existe el sistema muestra un mensaje de alerta.

Tabla N° 6.4. Descripción del caso de uso ingresar datos persona

Caso de uso	Modificar datos persona	
Actor	Jefe de Avalúos y Catastros	
Descripción	El actor modifica datos de una persona especificada.	
Precondición	El actor conoce los datos actuales de la persona.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción personas.
	2	El sistema muestra la ventana con el listado de las personas.
	3	El actor busca y selecciona a la persona deseada.
	4	El actor elije la opción modificar datos de persona.
	5	El sistema muestra la pantalla para modificar los datos de persona.
	6	El actor modifica los datos que requiere.
	7	El actor acepta la actualización de los datos.
Poscondición	El sistema modifica los datos de la persona.	
Excepciones	Paso	Acción
	7	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.5. Descripción del caso de uso modificar datos persona

Caso de uso	Eliminar datos persona	
Actor	Jefe de Avalúos y Catastros	
Descripción	El actor elimina datos de una persona especificada.	
Precondición	El actor conoce los datos actuales de la persona. Dicha persona no debe haber realizado ningún tipo de trámite previamente para ser borrada.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción personas.
	2	El sistema muestra la ventana con el listado de las personas.
	3	El actor busca y elije a la persona deseada.
	4	El actor elije la opción eliminar datos de persona.
	5	El sistema muestra un mensaje de advertencia.
	6	El actor confirma la eliminación.
Poscondición	El sistema elimina los datos de la persona y se cierra la ventana.	
Excepciones	Paso	Acción
	6	Si la persona ha realizado algún tipo de trámite previamente, el sistema advierte que no se puede eliminar.

Tabla N° 6.6. Descripción del caso de uso eliminar datos persona

Caso de uso	Generar reporte
--------------------	-----------------

Actor	Jefe de Avalúos y Catastros	
Descripción	El actor genera un reporte de las alcabalas ingresadas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción reporte.
	2	El sistema muestra la pantalla de reporte de alcabalas ingresadas.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de las alcabalas ingresadas en el intervalo especificado
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han ingresado alcabalas el sistema no mostrará datos.

Tabla N° 6.7. Descripción del caso de uso generar reporte

Caso de uso	Imprimir	
Actor	Jefe de Avalúos y Catastros	
Descripción	El sistema imprime un reporte de alcabalas ingresadas.	
Precondición	El actor genera el reporte a imprimirse.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción para imprimir.
	2	El sistema muestra una ventana de opciones de impresión.
	3	El actor elige la impresora, tamaño y orientación del papel.
	4	El actor manda a imprimir el reporte.
	5	El sistema imprime el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	3	Si en la ventana de opciones no aparece ninguna impresora el sistema no podrá imprimir el reporte.

Tabla N° 6.8. Descripción del caso de uso imprimir

▪ **Casos de uso para el actor Jefe de Rentas:**



Figura N° 6.2. Casos de uso para el actor Jefe de Rentas

Descripción de los casos de uso:

Caso de uso	Liquidar alcabala	
Actor	Jefe de Rentas	
Descripción	El actor liquida una alcabala ingresada previamente por el Jefe de Avalúos y Catastros.	
Precondición	El actor conoce el número de alcabala. Esta alcabala debe haber sido ingresada previamente en Avalúos y Catastros, además no debe estar pagada aún.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción liquidar alcabala.
	2	El sistema muestra la ventana de liquidación de alcabala.
	3	El actor provee el número de alcabala y lo manda a buscar.
	4	El sistema visualiza los datos de la alcabala (ingresados en Avalúos y Catastros).
	5	El actor manda a realizar los cálculos de liquidación.
	6	El sistema realiza el cálculo y visualiza los resultados.
	7	El actor puede realizar descuento o exención si es el caso.
8	El actor ingresa una observación y acepta la liquidación.	
Poscondición	El sistema actualiza los datos y cierra la ventana.	

Excepciones	Paso	Acción
	4	Si la alcabala especificada no existe o si ya ha sido pagada, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la liquidación.

Tabla N° 6.9. Descripción del caso de uso liquidar alcabala

Caso de uso	Anular liquidación de alcabala	
Actor	Jefe de Rentas	
Descripción	El actor anula la liquidación de una alcabala.	
Precondición	El actor conoce el número de alcabala. Esta alcabala debe haber sido ingresada y liquidada previamente, además no debe estar pagada aún.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción anular liquidación de alcabala.
	2	El sistema muestra la ventana de anulación de liquidación de alcabala.
	3	El actor provee el número de alcabala y lo manda a buscar.
	4	El sistema visualiza los datos de la alcabala.
	5	El actor manda a anular la liquidación.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la anulación.
Poscondición	El sistema anula la liquidación de la alcabala especificada y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la alcabala especificada no existe, o si no ha sido liquidada aún, o si ya ha sido pagada, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la anulación.

Tabla N° 6.10. Descripción del caso de uso anular liquidación de alcabala

Caso de uso	Anular pago de alcabala	
Actor	Jefe de Rentas	
Descripción	El actor anula el pago de una alcabala.	
Precondición	El actor conoce el número de alcabala. Esta alcabala debe haber sido pagada previamente en Recaudación (Tesorería).	
Secuencia	Paso	Acción

normal	1	El actor escoge la opción anular pago de alcabala.
	2	El sistema muestra la ventana de anulación de pago de alcabala.
	3	El actor provee el número de alcabala y lo manda a buscar.
	4	El sistema visualiza los datos de la alcabala.
	5	El actor manda a anular el pago.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la anulación.
Poscondición	El sistema anula el pago de la alcabala especificada y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la alcabala especificada no existe, o si no ha sido pagada aún, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la anulación.

Tabla N° 6.11. Descripción del caso de uso anular pago de alcabala

Caso de uso	Liquidar plusvalía	
Actor	Jefe de Rentas	
Descripción	El actor ingresa los datos de una plusvalía y la liquida.	
Precondición	El actor conoce los datos: comprador, vendedor, superficie y ubicación del lote, precio, etc.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción liquidar plusvalía.
	2	El sistema muestra la ventana de liquidación de plusvalía.
	3	El actor teclea los datos: número de plusvalía, comprador, vendedor, superficie y ubicación del lote, precio, etc.
	4	El actor puede realizar exención si es el caso.
	5	El actor manda a realizar los cálculos de liquidación.
	6	El sistema realiza el cálculo y visualiza los resultados.
	7	El actor ingresa una observación y acepta la liquidación.
Poscondición	El sistema ingresa los datos y la liquidación de la plusvalía. Cierra la ventana.	
Excepciones	Pasos	Acción
	5 y 7	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.12. Descripción del caso de uso liquidar plusvalía

Caso de uso	Anular pago de plusvalía
Actor	Jefe de Rentas

Descripción	El actor anula el pago de una plusvalía.	
Precondición	El actor conoce el número de la plusvalía. Esta plusvalía debe haber sido pagada previamente en Recaudación (Tesorería).	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción anular pago de plusvalía.
	2	El sistema muestra la ventana de anulación de pago de plusvalía.
	3	El actor provee el número de plusvalía y lo manda a buscar.
	4	El sistema visualiza los datos de la plusvalía.
	5	El actor manda a anular el pago.
	6	El sistema muestra un mensaje de advertencia.
7	El actor confirma la anulación.	
Poscondición	El sistema anula el pago de la plusvalía especificada y cierra la ventana	
Excepciones	Paso	Acción
	4	Si la plusvalía especificada no existe, o si no ha sido pagada aún, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la anulación.

Tabla N° 6.13. Descripción del caso de uso anular pago de plusvalía

Caso de uso	Eliminar plusvalía	
Actor	Jefe de Rentas	
Descripción	El actor elimina una plusvalía.	
Precondición	El actor conoce el número de la plusvalía. Esta plusvalía no debe estar pagada.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción eliminar plusvalía.
	2	El sistema muestra la ventana de eliminación de plusvalía.
	3	El actor provee el número de plusvalía y lo manda a buscar.
	4	El sistema visualiza los datos de la plusvalía.
	5	El actor manda a eliminar la plusvalía.
	6	El sistema muestra un mensaje de advertencia.
7	El actor confirma la eliminación.	
Poscondición	El sistema elimina la plusvalía especificada y cierra la ventana	
Excepciones	Paso	Acción
	4	Si la plusvalía especificada no existe, o si ya está pagada, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la eliminación.

Tabla N° 6.14. Descripción del caso de uso eliminar plusvalía

Caso de uso	Liquidar rodaje	
Actor	Jefe de Rentas	
Descripción	El actor ingresa los datos de rodajes vehiculares y liquida.	
Precondición	El actor conoce los datos: persona, año, placa, avalúo, etc.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción liquidar rodaje.
	2	El sistema muestra la ventana de liquidación de rodajes.
	3	El actor busca o ingresa los datos de la persona responsable del vehículo.
	4	El actor ingresa uno o más rodajes. Para cada uno realiza los pasos del 5 al 9.
	5	El actor teclea los datos: año, placa, avalúo, clase de vehículo, matrícula, etc.
	6	El actor puede realizar exención si es el caso.
	7	El actor manda a realizar los cálculos de liquidación.
	8	El sistema realiza el cálculo y visualiza los resultados.
	9	El actor ingresa una observación y acepta.
	10	El actor acepta la liquidación.
Poscondición	El sistema ingresa los datos y la liquidación del rodaje.	
Excepciones	Pasos	Acción
	7 y 9	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.15. Descripción del caso de uso liquidar rodaje

Caso de uso	Anular pago de rodaje	
Actor	Jefe de Rentas	
Descripción	El actor anula el pago de un rodaje.	
Precondición	El actor conoce la placa del vehículo y el año del rodaje. Este rodaje debe haber sido pagado previamente en Recaudación (Tesorería).	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción anular pago de rodaje.
	2	El sistema muestra la ventana de anulación de pago de rodaje.
	3	El actor provee la placa y el año, y manda a buscar.
	4	El sistema visualiza los datos del rodaje.
	5	El actor manda a anular el pago.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la anulación.

Poscondición	El sistema anula el pago del rodaje especificado y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si el rodaje especificado no existe, o si no ha sido pagado aún, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la anulación.

Tabla N° 6.16. Descripción del caso de uso anular pago de rodaje

Caso de uso	Eliminar rodaje	
Actor	Jefe de Rentas	
Descripción	El actor elimina un rodaje vehicular.	
Precondición	El actor conoce la placa del vehículo y el año. Este rodaje no debe estar pagado.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción eliminar rodaje.
	2	El sistema muestra la ventana de eliminación de rodaje.
	3	El actor provee la placa del vehículo y el año, y lo manda a buscar.
	4	El sistema visualiza los datos del rodaje.
	5	El actor manda a eliminar el rodaje.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la eliminación.
Poscondición	El sistema elimina el rodaje especificado y cierra la ventana	
Excepciones	Paso	Acción
	4	Si el rodaje especificado no existe, o si ya está pagado, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la eliminación.

Tabla N° 6.17. Descripción del caso de uso eliminar rodaje

Caso de uso	Emitir título de mercado	
Actor	Jefe de Rentas	
Descripción	El actor ingresa datos y calcula el valor a pagar por puesto en el mercado.	
Precondición	El actor conoce los datos: comerciante, n° de puesto, dirección, etc.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción emitir título de mercado.
	2	El sistema muestra la ventana de título de mercado.
	3	El actor teclea los datos: comerciante, n° de puesto, dirección, etc.
	4	El actor manda a realizar los cálculos.
	5	El sistema realiza el cálculo y visualiza los resultados.
	6	El actor acepta la emisión.

Poscondición	El sistema ingresa los datos y cierra la ventana.	
Excepciones	Pasos	Acción
	4 y 6	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.18. Descripción del caso de uso emitir título de mercado

Caso de uso	Anular pago de título de mercado	
Actor	Jefe de Rentas	
Descripción	El actor anula el pago de título de mercado.	
Precondición	El actor conoce el número de título. Este título debe haber sido pagado previamente en Recaudación (Tesorería).	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción anular pago de título de mercado.
	2	El sistema muestra la ventana de anulación de título de mercado.
	3	El actor teclea el número de título y manda a buscar.
	4	El sistema visualiza los datos.
	5	El actor manda a anular el pago.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la anulación.
Poscondición	El sistema anula el pago del título especificado y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si el título especificado no existe, o si no ha sido pagado aún, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la anulación.

Tabla N° 6.19. Descripción del caso de uso anular pago de título de mercado

Caso de uso	Eliminar título de mercado	
Actor	Jefe de Rentas	
Descripción	El actor elimina un título de mercado.	
Precondición	El actor conoce el número de título. Este título no debe estar pagado.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción eliminar título de mercado.
	2	El sistema muestra la ventana de eliminación de título de mercado.
	3	El actor teclea el número de título y manda a buscar.
	4	El sistema visualiza los datos.
	5	El actor manda a eliminar el título.
	6	El sistema muestra un mensaje de advertencia.

	7	El actor confirma la eliminación.
Poscondición	El sistema elimina el título especificado y cierra la ventana	
Excepciones	Paso	Acción
	4	Si el título especificado no existe, o si ya está pagado, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la eliminación.

Tabla N° 6.20. Descripción del caso de uso eliminar título de mercado

Caso de uso	Emitir multa	
Actor	Jefe de Rentas	
Descripción	El actor ingresa datos y calcula el valor de la multa.	
Precondición	El actor conoce los datos: nombre de la persona, concepto, valor, interés, etc.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción emitir multa.
	2	El sistema muestra la ventana de emisión de multa.
	3	El actor teclea los datos: nombre de la persona, concepto, valor, interés, etc.
	4	El actor manda a realizar el cálculo.
	5	El sistema realiza el cálculo y visualiza los resultados.
	6	El actor acepta la emisión.
Poscondición	El sistema ingresa los datos y cierra la ventana.	
Excepciones	Paso	Acción
	4 y 6	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.21. Descripción del caso de uso emitir multa

Caso de uso	Anular pago de multa	
Actor	Jefe de Rentas	
Descripción	El actor anula el pago de multa	
Precondición	El actor conoce el número de título de multa. Esta multa debe haber sido pagada previamente en Recaudación (Tesorería).	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción anular pago multa.
	2	El sistema muestra la ventana de anulación de multa.
	3	El actor teclea el número de título de multa y manda a buscar.
	4	El sistema visualiza los datos.
	5	El actor manda a anular el pago.

	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la anulación.
Poscondición	El sistema anula el pago de la multa especificada y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si el título de multa especificado no existe, o si no ha sido pagado aún, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la anulación.

Tabla N° 6.22. Descripción del caso de uso anular pago de multa

Caso de uso	Eliminar multa	
Actor	Jefe de Rentas	
Descripción	El actor elimina una multa.	
Precondición	El actor conoce el número de título de multa. Esta multa no debe estar pagada.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción eliminar multa.
	2	El sistema muestra la ventana de eliminación de multa.
	3	El actor teclea el número de título de multa y manda a buscar.
	4	El sistema visualiza los datos.
	5	El actor manda a eliminar la multa.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la eliminación.
Poscondición	El sistema elimina la multa y cierra la ventana	
Excepciones	Paso	Acción
	4	Si el título de multa especificado no existe, o si ya está pagado, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la eliminación.

Tabla N° 6.23. Descripción del caso de uso eliminar multa

Caso de uso	Emitir título de medidor de agua	
Actor	Jefe de Rentas	
Descripción	El actor ingresa datos y calcula el valor a pagar.	
Precondición	El actor conoce los datos: nombre de la persona, valor, derecho de instalación, servicios, etc.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción emitir título de medidor de agua.
	2	El sistema muestra la ventana de título de medidor de agua.
	3	El actor teclea los datos: nombre de la persona, valor, derecho de

		instalación, servicios, etc.
	4	El actor manda a realizar el cálculo.
	5	El sistema realiza el cálculo y visualiza los resultados.
	6	El actor acepta la emisión.
Poscondición	El sistema ingresa los datos y cierra la ventana.	
Excepciones	Paso	Acción
	4 y 6	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.24. Descripción del caso de uso emitir título de medidor de agua

Caso de uso	Anular pago de título de medidor de agua	
Actor	Jefe de Rentas	
Descripción	El actor anula el pago de título de medidor de agua.	
Precondición	El actor conoce el número de título. Este título debe haber sido pagado previamente en Recaudación (Tesorería).	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción anular pago de título de medidor de agua.
	2	El sistema muestra la ventana de anulación de título de medidor de agua.
	3	El actor tecldea el número de título y manda a buscar.
	4	El sistema visualiza los datos.
	5	El actor manda a anular el pago.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la anulación.
Poscondición	El sistema anula el pago del título especificado y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si el título especificado no existe, o si no ha sido pagado aún, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la anulación.

Tabla N° 6.25. Descripción del caso de uso anular pago de título de medidor de agua

Caso de uso	Eliminar título de medidor de agua	
Actor	Jefe de Rentas	
Descripción	El actor elimina un título de medidor de agua.	
Precondición	El actor conoce el número de título. Este título no debe estar pagado.	
Secuencia	Paso	Acción

normal	1	El actor escoge la opción eliminar título de medidor de agua.
	2	El sistema muestra la ventana de eliminación de título de medidor de agua.
	3	El actor teclea el número de título y manda a buscar.
	4	El sistema visualiza los datos.
	5	El actor manda a eliminar el título.
	6	El sistema muestra un mensaje de advertencia.
	7	El actor confirma la eliminación.
Poscondición	El sistema elimina el título especificado y cierra la ventana	
Excepciones	Paso	Acción
	4	Si el título especificado no existe, o si ya está pagado, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar la eliminación.

Tabla N° 6.26. Descripción del caso de uso eliminar título de medidor de agua

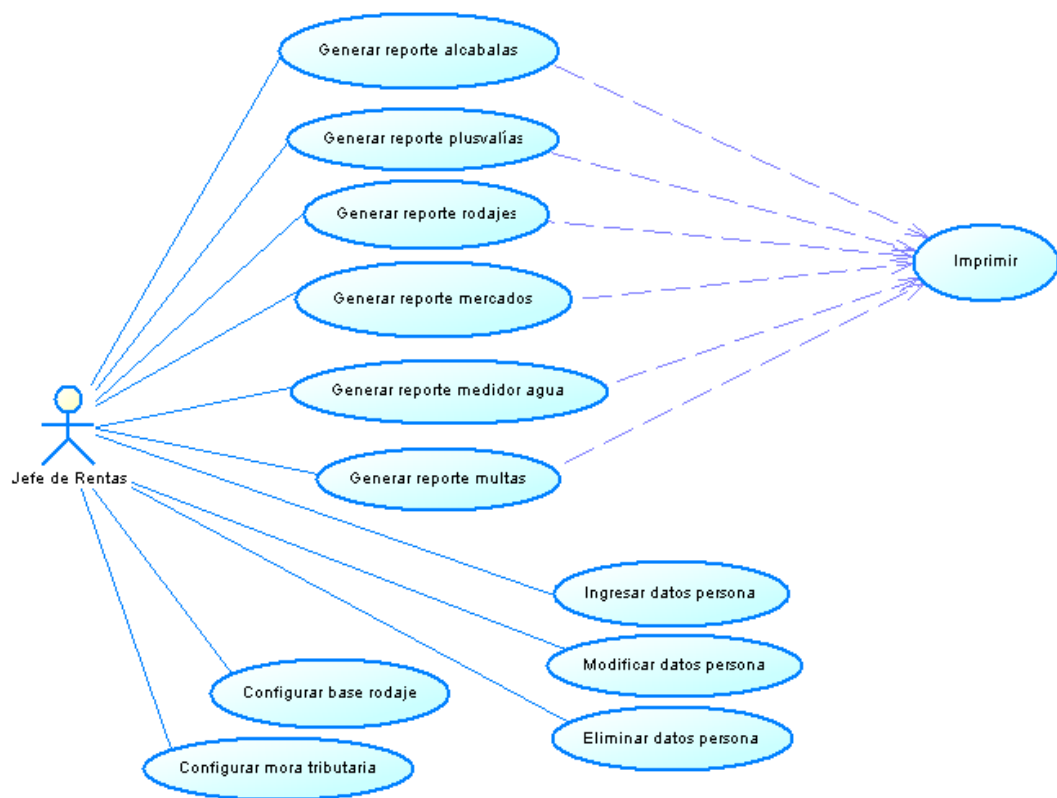


Figura N° 6.3. Casos de uso para el actor Jefe de Rentas (continuación)

Descripción de los casos de uso:

Caso de uso	Generar reporte alcabalas
Actor	Jefe de Rentas
Descripción	El actor genera un reporte de las alcabalas liquidadas en un lapso de tiempo

	especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte de alcabalas.
	2	El sistema muestra la pantalla de reporte de alcabalas liquidadas.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de las alcabalas liquidadas en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han liquidado alcabalas el sistema no mostrará ninguna información.

Tabla N° 6.27. Descripción del caso de uso generar reporte alcabalas

Caso de uso	Generar reporte plusvalías	
Actor	Jefe de Rentas	
Descripción	El actor genera un reporte de las plusvalías liquidadas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte de plusvalías.
	2	El sistema muestra la pantalla de reporte de plusvalías liquidadas.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de las plusvalías liquidadas en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han liquidado plusvalías el sistema no mostrará ninguna información.

Tabla N° 6.28. Descripción del caso de uso generar reporte plusvalías

Caso de uso	Generar reporte rodajes	
Actor	Jefe de Rentas	
Descripción	El actor genera un reporte de rodajes liquidados en un lapso de tiempo	

	especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción de reporte de rodajes.
	2	El sistema muestra la pantalla de reporte de rodajes liquidados.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de rodajes liquidados en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han liquidado rodajes el sistema no mostrará ninguna información.

Tabla N° 6.29. Descripción del caso de uso generar reporte rodajes

Caso de uso	Generar reporte mercados	
Actor	Jefe de Rentas	
Descripción	El actor genera un reporte títulos de mercado emitidos en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de títulos de mercado.
	2	El sistema muestra la pantalla de reporte de títulos de mercado emitidos.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de títulos de mercado emitidos en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han emitido títulos de mercado, el sistema no mostrará ninguna información.

Tabla N° 6.30. Descripción del caso de uso generar reporte mercados

Caso de uso	Generar reporte multas
Actor	Jefe de Rentas

Descripción	El actor genera un reporte de multas emitidas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de multas.
	2	El sistema muestra la pantalla de reporte de multas emitidas.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de multas emitidas en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han emitido multas, el sistema no mostrará ningún dato.

Tabla N° 6.31. Descripción del caso de uso generar reporte multas

Caso de uso	Generar reporte medidor de agua	
Actor	Jefe de Rentas	
Descripción	El actor genera un reporte títulos de medidor de agua emitidos en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción reporte de títulos de agua.
	2	El sistema muestra la pantalla de reporte de títulos medidor de agua emitidos.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de títulos de medidor de agua emitidos en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han emitido títulos de medidor de agua, el sistema no mostrará información.

Tabla N° 6.32. Descripción del caso de uso generar reporte medidor de agua

Caso de uso	Imprimir
--------------------	----------

Actor	Jefe de Rentas	
Descripción	El sistema imprime un reporte especificado.	
Precondición	El actor genera el reporte a imprimirse.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción para imprimir.
	2	El sistema muestra una ventana de opciones de impresión.
	3	El actor elige la impresora, tamaño y orientación del papel.
	4	El actor manda a imprimir el reporte.
	5	El sistema imprime el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	3	Si en la ventana de opciones no aparece ninguna impresora el sistema no podrá imprimir el reporte.

Tabla N° 6.33. Descripción del caso de uso imprimir

Caso de uso	Ingresar datos persona	
Actor	Jefe de Rentas	
Descripción	El actor ingresa datos de una nueva persona al sistema.	
Precondición	El actor conoce los datos de la persona, la cual no debe estar ya en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción personas.
	2	El sistema muestra la ventana con el listado de las personas.
	3	El actor elige la opción ingresar nueva persona.
	4	El sistema muestra la pantalla para ingreso de una nueva persona.
	5	El actor teclea los datos de la persona.
	6	El actor acepta el ingreso de los datos.
Poscondición	El sistema ingresa los datos de la persona.	
Excepciones	Paso	Acción
	6	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.
	6	Si la persona ya existe el sistema muestra un mensaje de alerta.

Tabla N° 6.34. Descripción del caso de uso ingresar datos persona

Caso de uso	Modificar datos persona	
Actor	Jefe de Rentas	
Descripción	El actor modifica datos de una persona especificada.	

Precondición	El actor conoce los datos actuales de la persona.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción personas.
	2	El sistema muestra la ventana con el listado de las personas.
	3	El actor busca y selecciona a la persona deseada.
	4	El actor elige la opción modificar datos de persona.
	5	El sistema muestra la pantalla para modificar los datos de persona.
	6	El actor modifica los datos que requiere.
	7	El actor acepta la modificación de los datos.
Poscondición	El sistema modifica los datos de la persona.	
Excepciones	Paso	Acción
	7	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.35. Descripción del caso de uso modificar datos persona

Caso de uso	Eliminar datos persona	
Actor	Jefe de Rentas	
Descripción	El actor elimina datos de una persona especificada.	
Precondición	El actor conoce los datos actuales de la persona. Dicha persona no debe haber realizado ningún tipo de trámite previamente para ser borrada.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción personas.
	2	El sistema muestra la ventana con el listado de las personas.
	3	El actor busca y elige a la persona deseada.
	4	El actor elige la opción eliminar datos de persona.
	5	El sistema muestra un mensaje de advertencia.
	6	El actor confirma la eliminación.
Poscondición	El sistema elimina los datos de la persona y se cierra la ventana.	
Excepciones	Paso	Acción
	6	Si la persona ha realizado algún tipo de trámite previamente, el sistema advierte que no se puede eliminar.

Tabla N° 6.36. Descripción del caso de uso eliminar datos persona

Caso de uso	Configurar base rodaje
--------------------	------------------------

Actor	Jefe de Rentas	
Descripción	El actor configura la base imponible para el cálculo del rodaje vehicular.	
Precondición	El actor conoce los rangos y valores de la base imponible, según manda la Ley Orgánica Municipal.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción configurar base imponible.
	2	El sistema muestra la ventana de configuración.
	3	El actor realiza los cambios necesarios, como valores y rangos.
	4	El actor acepta los cambios realizados.
Poscondición	El sistema actualiza los datos y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.37. Descripción del caso de uso configurar base rodaje

Caso de uso	Configurar mora tributaria	
Actor	Jefe de Rentas	
Descripción	El actor configura la tabla de Intereses por Mora Tributaria, para el cálculo del rodaje vehicular.	
Precondición	El actor conoce los valores y años de la tabla de Intereses por Mora Tributaria. Esta tabla le es entregada al Jefe de Rentas cada mes.	
Secuencia normal	Paso	Acción
	1	El actor escoge la opción configurar mora tributaria.
	2	El sistema muestra la ventana de configuración.
	3	El actor realiza los cambios necesarios, como valores y años.
	4	El actor acepta los cambios realizados.
Poscondición	El sistema actualiza los datos y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.38. Descripción del caso de uso configurar mora tributaria

▪ **Casos de uso para el actor Recaudador:**

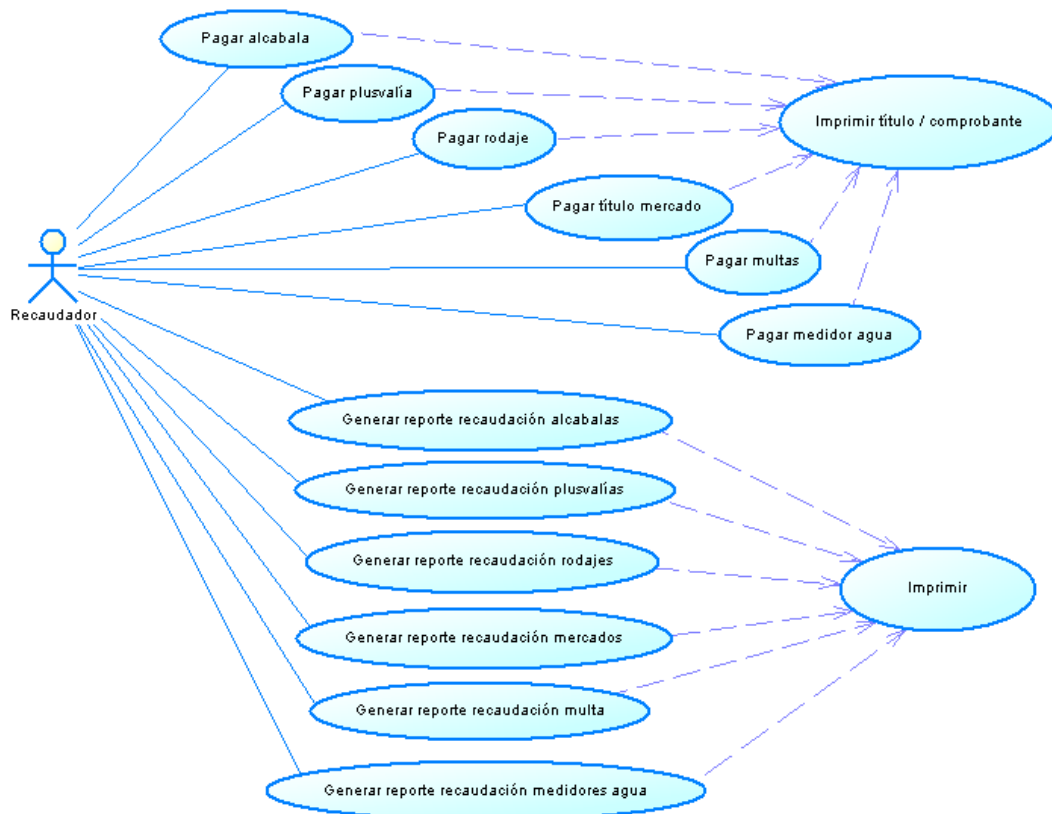


Figura N° 6.4. Casos de uso para el actor Recaudador

Descripción de los casos de uso:

Caso de uso	Pagar alcabala	
Actor	Recaudador	
Descripción	El actor realiza el pago de una alcabala. Esta alcabala debe haber sido liquidada previamente por el Jefe de Rentas.	
Precondición	El actor conoce el número de la alcabala a pagar.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción pagar alcabala.
	2	El sistema muestra la ventana de pago de alcabala.
	3	El actor teclea el número de alcabala y manda a buscar.
	4	El sistema visualiza los datos y el valor a pagar.
	5	El actor ejecuta la opción pagar luego de tomar el dinero.
	6	El sistema envía un mensaje advirtiéndole que se va imprimir el comprobante de pago.
Poscondición	El sistema actualiza el pago, imprime comprobante y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la alcabala especificada no existe, o si no ha sido liquidada aún,

		o si ya fue pagada, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar el pago.
--	--	--

Tabla N° 6.39. Descripción del caso de uso pagar alcabala

Caso de uso	Pagar plusvalía	
Actor	Recaudador	
Descripción	El actor realiza el pago de una plusvalía. Esta plusvalía debe haber sido ingresada y liquidada previamente por el Jefe de Rentas.	
Precondición	El actor conoce el número de la plusvalía a pagar.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción pagar plusvalía.
	2	El sistema muestra la ventana de pago de plusvalía.
	3	El actor teclea el número de plusvalía y manda a buscar.
	4	El sistema visualiza los datos y el valor a pagar.
	5	El actor ejecuta la opción pagar luego de tomar el dinero.
6	El sistema envía un mensaje advirtiendo que se va imprimir el comprobante de pago.	
Poscondición	El sistema actualiza el pago, imprime comprobante y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la plusvalía especificada no existe o si ya fue pagada, el sistema no mostrará datos y enviará un mensaje que advierte que no se puede realizar el pago.

Tabla N° 6.40. Descripción del caso de uso pagar plusvalía

Caso de uso	Pagar rodaje	
Actor	Recaudador	
Descripción	El actor realiza el pago de rodaje(s) vehicular(es). Este(os) debe(n) haber sido ingresado(s) y liquidado(s) previamente por el Jefe de Rentas.	
Precondición	El actor conoce nombres o cédula de la persona que debe pagar.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción pagar rodaje.
	2	El sistema muestra la ventana de pago de rodaje.
	3	El actor manda a buscar a la persona, para ver lo que ésta debe.
	4	El sistema visualiza el (los) rodaje(s) que debe pagar la persona especificada.
	5	El actor ejecuta la opción pagar luego de tomar el dinero.
6	El sistema envía un mensaje advirtiendo que se va imprimir el	

		(los) comprobante(s) de pago.
Poscondición	El sistema actualiza el pago, imprime comprobante(s) y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la persona especificada no debe ningún rodaje, el sistema no mostrará datos y enviará un mensaje que advierte que dicha persona no adeuda.

Tabla N° 6.41. Descripción del caso de uso pagar rodaje

Caso de uso	Pagar título mercado	
Actor	Recaudador	
Descripción	El actor realiza el pago de título(s) de puesto en el mercado. Este(os) debe(n) haber sido emitido(s) previamente por el Jefe de Rentas.	
Precondición	El actor conoce nombres o cédula de la persona que debe pagar.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción pagar título de mercado.
	2	El sistema muestra la ventana de pago de título de mercado.
	3	El actor manda a buscar a la persona, para ver lo que ésta debe.
	4	El sistema visualiza el (los) título(s) que debe pagar la persona especificada.
	5	El actor ejecuta la opción pagar luego de tomar el dinero.
	6	El sistema envía un mensaje advirtiendo que se va imprimir el (los) título(s) de crédito.
Poscondición	El sistema actualiza el pago, imprime título(s) y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la persona especificada no debe ningún título, el sistema no mostrará datos y enviará un mensaje que advierte que dicha persona no adeuda.

Tabla N° 6.42. Descripción del caso de uso pagar título mercado

Caso de uso	Pagar multas	
Actor	Recaudador	
Descripción	El actor realiza el pago de multa(s). Esta(as) debe(n) haber sido emitida(s) previamente por el Jefe de Rentas.	
Precondición	El actor conoce nombres o cédula de la persona que debe pagar.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción pagar multas.
	2	El sistema muestra la ventana de pago de multas.

	3	El actor manda a buscar a la persona, para ver lo que ésta debe.
	4	El sistema visualiza la (las) multa(s) que debe pagar la persona especificada.
	5	El actor ejecuta la opción pagar luego de tomar el dinero.
	6	El sistema envía un mensaje advirtiendo que se va imprimir el (los) título(s) de crédito.
Poscondición	El sistema actualiza el pago, imprime título(s) y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la persona especificada no debe ninguna multa, el sistema no mostrará datos y enviará un mensaje que advierte que dicha persona no adeuda.

Tabla N° 6.43. Descripción del caso de uso pagar multas

Caso de uso	Pagar título de medidor de agua	
Actor	Recaudador	
Descripción	El actor realiza el pago de título(s) de medidor de agua. Este(os) debe(n) haber sido emitido(s) previamente por el Jefe de Rentas.	
Precondición	El actor conoce nombres o cédula de la persona que debe pagar.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción pagar título de medidor de agua.
	2	El sistema muestra la ventana de pago de título de medidor de agua.
	3	El actor manda a buscar a la persona, para ver lo que ésta debe.
	4	El sistema visualiza el (los) título(s) que debe pagar la persona especificada.
	5	El actor ejecuta la opción pagar luego de tomar el dinero.
	6	El sistema envía un mensaje advirtiendo que se va imprimir el (los) título(s) de crédito.
Poscondición	El sistema actualiza el pago, imprime título(s) y cierra la ventana.	
Excepciones	Paso	Acción
	4	Si la persona especificada no debe ningún título, el sistema no mostrará datos y enviará un mensaje que advierte que dicha persona no adeuda.

Tabla N° 6.44. Descripción del caso de uso pagar título de medidor de agua

Caso de uso	Imprimir título / comprobante
Actor	Recaudador

Descripción	El sistema imprime el título de crédito o el comprobante de pago.	
Precondición	El actor genera el reporte a imprimirse.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción para imprimir.
	2	El sistema muestra una ventana de opciones de impresión.
	3	El actor elige la impresora, tamaño y orientación del papel.
	4	El actor manda a imprimir el título o comprobante.
5	El sistema imprime el título o comprobante.	
Poscondición	El sistema provee el título o comprobante impreso.	
Excepciones	Paso	Acción
	3	Si en la ventana de opciones no aparece ninguna impresora el sistema no podrá imprimir el título o comprobante.

Tabla N° 6.45. Descripción del caso de uso imprimir título/comprobante

Caso de uso	Generar reporte recaudación alcabalas	
Actor	Recaudador	
Descripción	El actor genera un reporte de recaudación de alcabalas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de recaudación de alcabalas.
	2	El sistema muestra la pantalla de reporte de recaudación de alcabalas.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de las alcabalas recaudadas en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han recaudado alcabalas el sistema no mostrará ninguna información.

Tabla N° 6.46. Descripción del caso de uso generar reporte recaudación alcabalas

Caso de uso	Generar reporte recaudación plusvalías
Actor	Recaudador

Descripción	El actor genera un reporte de recaudación de plusvalías en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de recaudación de plusvalías.
	2	El sistema muestra la pantalla de reporte de recaudación de plusvalías.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de las plusvalías recaudadas en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han recaudado plusvalías el sistema no mostrará ninguna información.

Tabla N° 6.47. Descripción del caso de uso generar reporte recaudación plusvalías

Caso de uso	Generar reporte recaudación rodajes	
Actor	Recaudador	
Descripción	El actor genera un reporte de recaudación de rodajes en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de recaudación de rodajes.
	2	El sistema muestra la pantalla de reporte de recaudación de rodajes.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de los rodajes recaudados en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han recaudado rodajes el sistema no mostrará ninguna información.

Tabla N° 6.48. Descripción del caso de uso generar reporte recaudación rodajes

Caso de uso	Generar reporte recaudación mercados
--------------------	--------------------------------------

Actor	Recaudador	
Descripción	El actor genera un reporte de recaudación de títulos de mercado en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte de recaudación de mercados.
	2	El sistema muestra la pantalla de reporte de recaudación de mercados.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de títulos de mercado recaudados en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han recaudado títulos de mercado el sistema no mostrará ninguna información.

Tabla N° 6.49. Descripción del caso de uso generar reporte recaudación mercados

Caso de uso	Generar reporte recaudación multas	
Actor	Recaudador	
Descripción	El actor genera un reporte de recaudación de multas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte de recaudación de multas.
	2	El sistema muestra la pantalla de reporte de recaudación de multas.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de multas recaudadas en el intervalo especificado.
5	El actor manda a imprimir el reporte.	
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han recaudado multas, el sistema no mostrará ninguna información.

Tabla N° 6.50. Descripción del caso de uso generar reporte recaudación multas

Caso de uso	Generar reporte recaudación medidores agua
--------------------	--

Actor	Recaudador	
Descripción	El actor genera un reporte de recaudación de títulos de medidor de agua en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de recaudación de títulos de medidor de agua.
	2	El sistema muestra la pantalla de reporte de recaudación de títulos de medidor de agua.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de títulos de medidor de agua recaudados en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han recaudado títulos de medidor de agua, el sistema no mostrará ninguna información.

Tabla N° 6.51. Descripción del caso de uso generar reporte recaudación medidores agua

- **Casos de uso para el actor Administrador:**

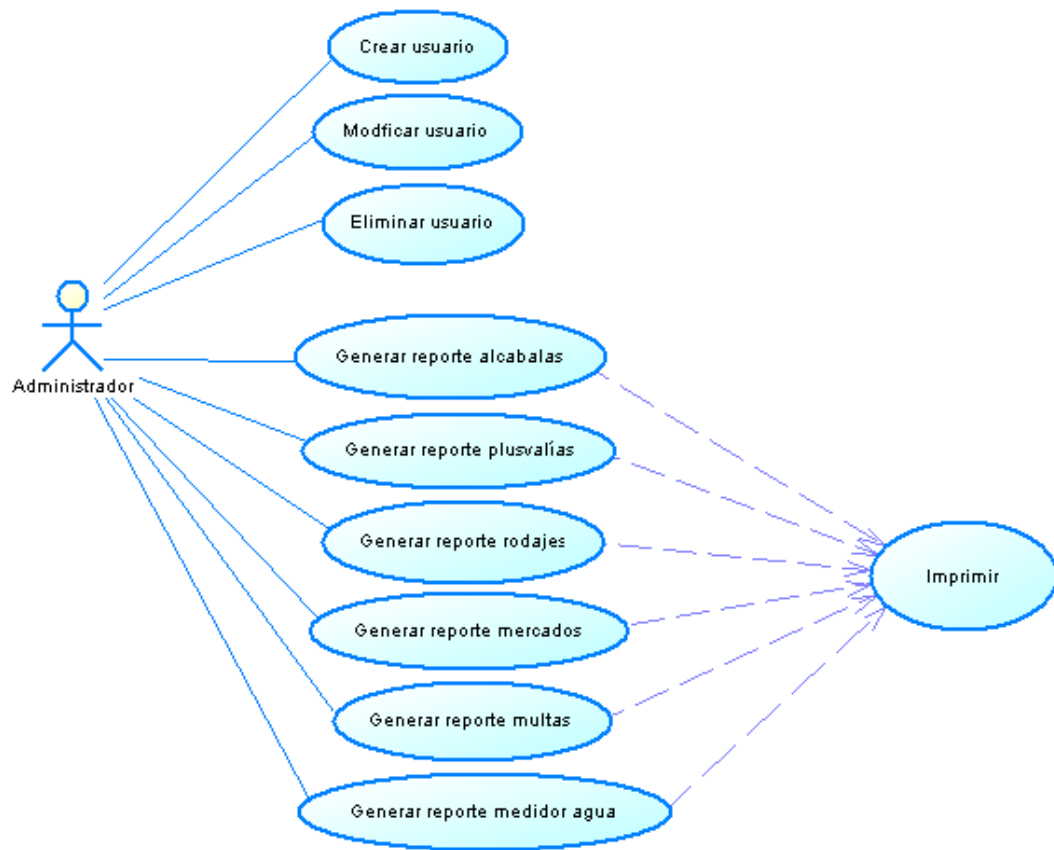


Figura N° 6.5. Casos de uso para el actor Administrador

Descripción de los casos de uso:

Caso de uso	Crear usuario	
Actor	Administrador	
Descripción	El actor crea un nuevo usuario del sistema.	
Precondición	El actor conoce los datos del usuario y la función que debe cumplir.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción usuarios.
	2	El sistema muestra la ventana con el listado de los usuarios del sistema.
	3	El actor elije la opción nuevo usuario.
	4	El sistema muestra la ventana para ingresar nuevo usuario.
	5	El actor teclea los datos del nuevo usuario y lo asocia a un grupo determinado (rentas, avalúos, tesorería o administración).
	6	El actor acepta la creación del nuevo usuario.
Poscondición	El sistema crea un nuevo usuario.	

Excepciones	Paso	Acción
		6

Tabla N° 6.52. Descripción del caso de uso crear usuario

Caso de uso	Modificar usuario	
Actor	Administrador	
Descripción	El actor modifica datos de un usuario del sistema.	
Precondición	El actor conoce los datos actuales del usuario.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción usuarios.
	2	El sistema muestra la ventana con el listado de los usuarios del sistema.
	3	El actor busca y selecciona a la usuario deseado.
	4	El actor elige la opción modificar datos de usuario.
	5	El sistema muestra la pantalla para modificar los datos de usuario.
	6	El actor modifica los datos que requiere.
	7	El actor acepta la modificación de los datos.
Poscondición	El sistema modifica los datos de la persona.	
Excepciones	Paso	Acción
	7	Si los datos suministrados no están en el formato correcto el sistema envía un mensaje de alerta. Se deben volver a teclear.

Tabla N° 6.53. Descripción del caso de uso modificar usuario

Caso de uso	Eliminar usuario	
Actor	Administrador	
Descripción	El actor elimina datos de un usuario del sistema. Los usuarios del grupo “tesorería” no pueden ser eliminados.	
Precondición	El actor conoce los datos actuales del usuario.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción usuarios.
	2	El sistema muestra la ventana con el listado de los usuarios.
	3	El actor busca y elige al usuario deseado.
	4	El actor elige la opción eliminar usuario.
	5	El sistema muestra un mensaje de advertencia.
	6	El actor confirma la eliminación.

Poscondición	El sistema elimina los datos del usuario y se cierra la ventana.	
Excepciones	Paso	Acción
	6	Si el usuario a eliminar pertenece al grupo “tesorería”, el sistema advierte que no se puede eliminar.

Tabla N° 6.54. Descripción del caso de uso eliminar usuario

Caso de uso	Generar reporte alcabalas	
Actor	Administrador	
Descripción	El actor genera un reporte de las alcabalas ingresadas, liquidadas o recaudadas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de alcabalas ingresadas, liquidadas o recaudadas.
	2	El sistema muestra la pantalla de reporte.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de las alcabalas ingresadas, liquidadas o recaudadas en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han ingresado, liquidado o recaudado alcabalas el sistema no mostrará ninguna información.

Tabla N° 6.55. Descripción del caso de uso generar reporte alcabalas

Caso de uso	Generar reporte plusvalías	
Actor	Administrador	
Descripción	El actor genera un reporte de las plusvalías liquidadas o recaudadas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción de reporte de plusvalías liquidadas o recaudadas.
	2	El sistema muestra la pantalla de reporte.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.

	4	El sistema visualiza un listado de las plusvalías liquidadas o recaudadas en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han liquidado o recaudado plusvalías el sistema no mostrará ninguna información.

Tabla N° 6.56. Descripción del caso de uso generar reporte plusvalías

Caso de uso	Generar reporte rodajes	
Actor	Administrador	
Descripción	El actor genera un reporte de rodajes liquidados o recaudados en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte de rodajes liquidados o recaudados.
	2	El sistema muestra la pantalla de reporte.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de rodajes liquidados o recaudados en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han liquidado o recaudado rodajes el sistema no mostrará ninguna información.

Tabla N° 6.57. Descripción del caso de uso generar reporte rodajes

Caso de uso	Generar reporte mercados	
Actor	Administrador	
Descripción	El actor genera un reporte de los títulos de mercado emitidos o recaudados en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte de títulos de mercado emitidos o recaudados.
	2	El sistema muestra la pantalla de reporte.

	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de títulos de mercado emitidos o recaudados en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han emitido o recaudado títulos el sistema no mostrará ninguna información.

Tabla N° 6.58. Descripción del caso de uso generar reporte mercados

Caso de uso	Generar reporte multas	
Actor	Administrador	
Descripción	El actor genera un reporte de las multas emitidas o recaudadas en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte de multas emitidas o recaudadas.
	2	El sistema muestra la pantalla de reporte.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de multas emitidas o recaudadas en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han emitido o recaudado multas, el sistema no mostrará ninguna información.

Tabla N° 6.59. Descripción del caso de uso generar reporte multas

Caso de uso	Generar reporte medidor agua	
Actor	Administrador	
Descripción	El actor genera un reporte de los títulos de medidores de agua emitidos o recaudados en un lapso de tiempo especificado.	
Precondición	El actor determina el intervalo de tiempo.	
Secuencia normal	Paso	Acción
	1	El actor elije la opción de reporte títulos de medidores de agua

		emitidos o recaudados.
	2	El sistema muestra la pantalla de reporte.
	3	El actor provee la fecha inicial y la fecha final del intervalo de tiempo.
	4	El sistema visualiza un listado de títulos de medidores de agua emitidos o recaudados en el intervalo especificado.
	5	El actor manda a imprimir el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	4	Si en el intervalo especificado no se han emitido o recaudado títulos, el sistema no mostrará ninguna información.

Tabla N° 6.60. Descripción del caso de uso generar reporte medidor agua

Caso de uso	Imprimir	
Actor	Administrador	
Descripción	El sistema imprime un reporte especificado.	
Precondición	El actor genera el reporte a imprimirse.	
Secuencia normal	Paso	Acción
	1	El actor elige la opción para imprimir.
	2	El sistema muestra una ventana de opciones de impresión.
	3	El actor elige la impresora, tamaño y orientación del papel.
	4	El actor manda a imprimir el reporte.
	5	El sistema imprime el reporte.
Poscondición	El sistema provee un reporte completo.	
Excepciones	Paso	Acción
	3	Si en la ventana de opciones no aparece ninguna impresora el sistema no podrá imprimir el reporte.

Tabla N° 6.61. Descripción del caso de uso imprimir

6.4. Diseño

6.4.1. Diagrama de clases

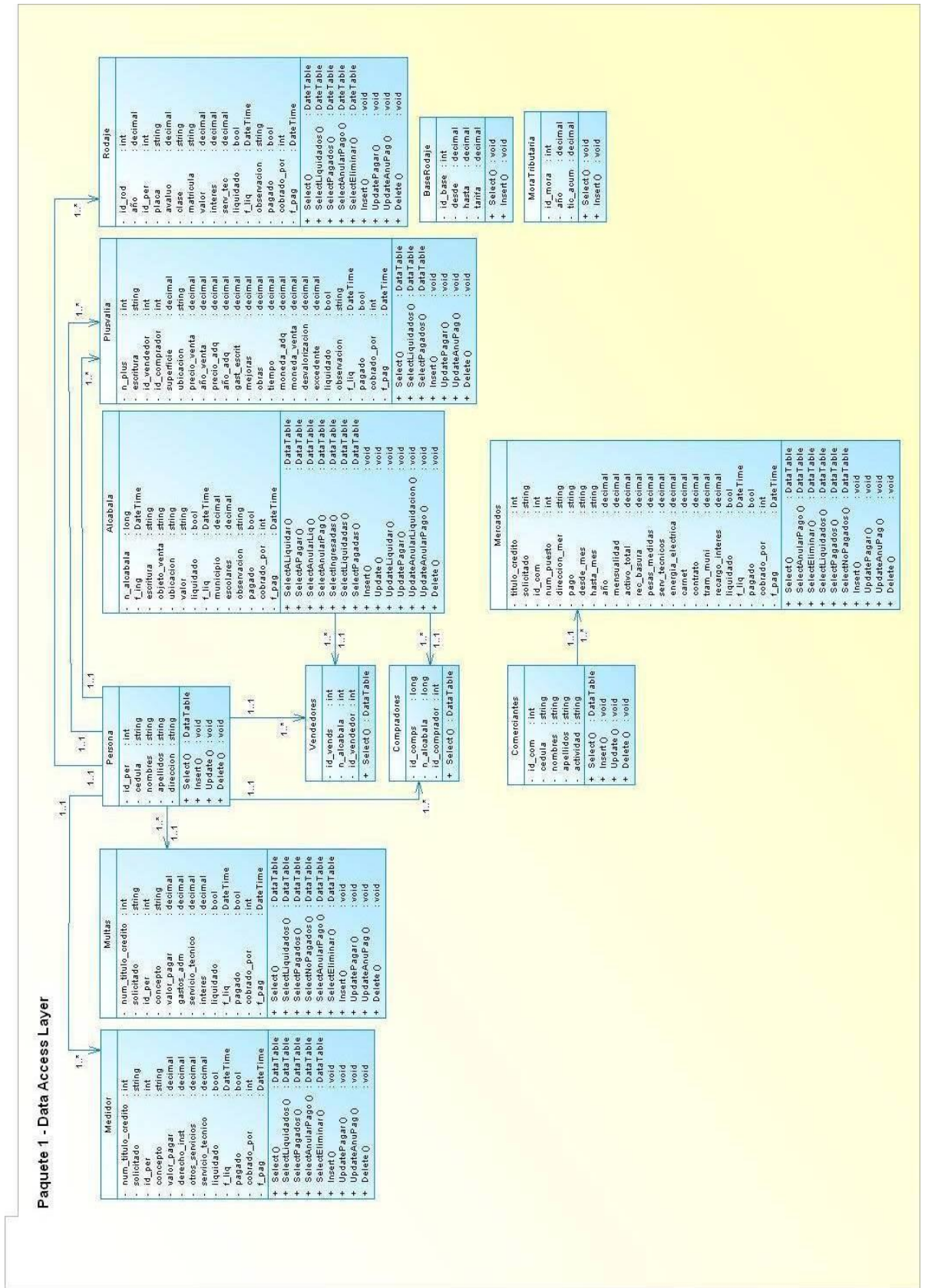


Figura N° 6.6. Diagrama de clases

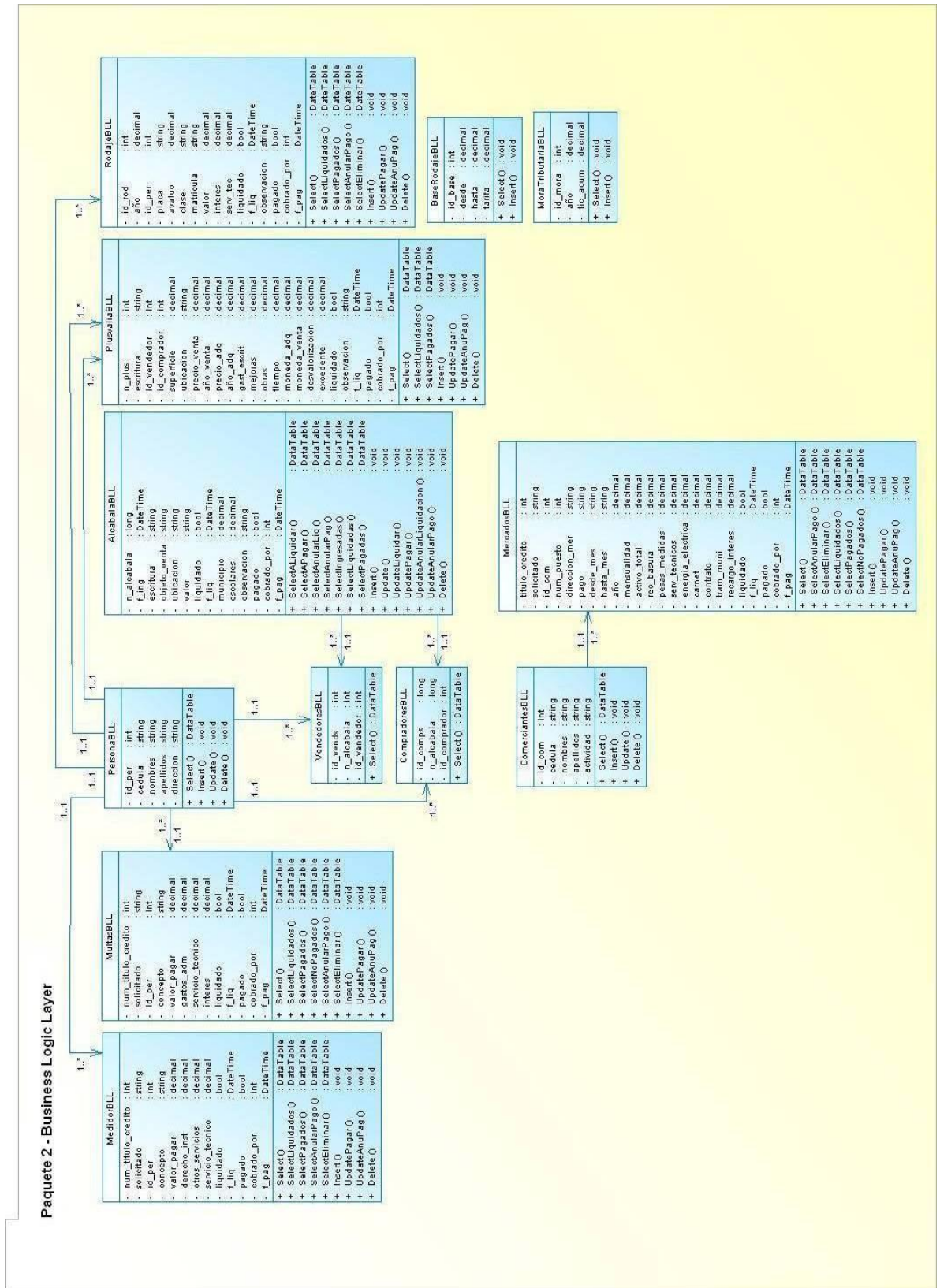


Figura N° 6.7. Diagrama de clases (continuación)

6.4.2. Diagrama de secuencias

Los diagramas de secuencias permiten ver la interacción que existe entre los objetos de las distintas clases a través de envío de mensajes, como una secuencia ordenada en el tiempo. A continuación se muestran algunos diagramas de secuencia.

Diagrama de secuencia para el caso de uso *Ingresar alcabala*, para el actor Jefe de Avalúos y Catastros:

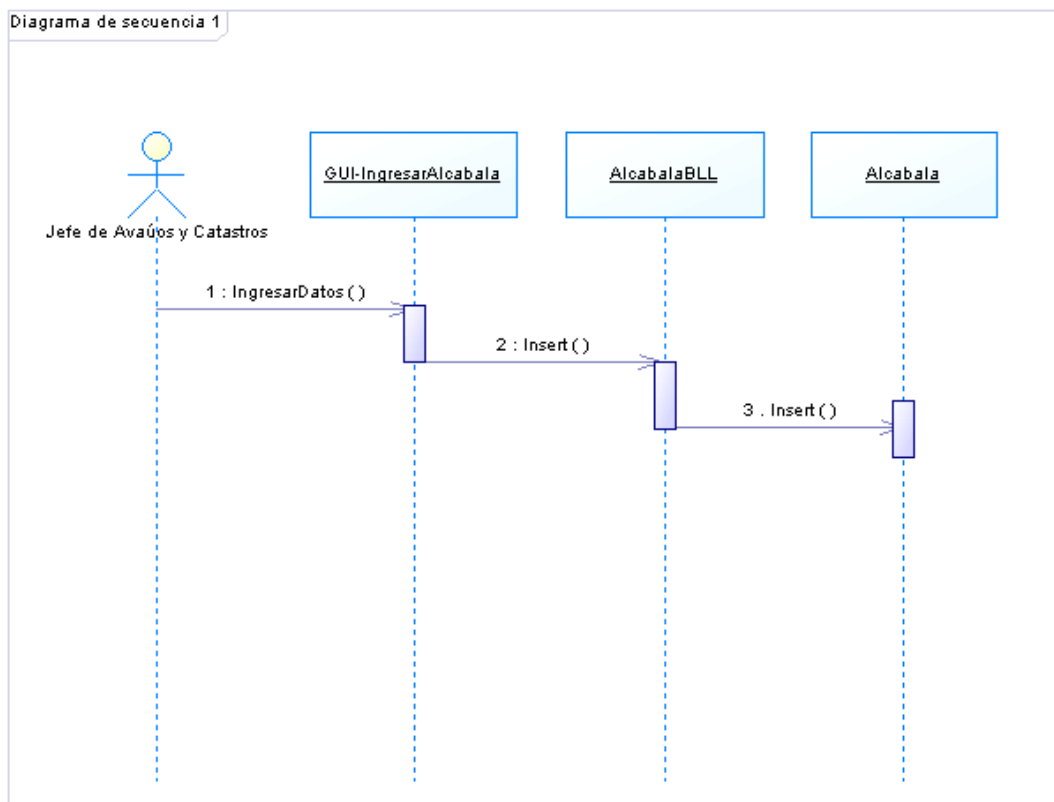


Figura N° 6.8. Diagrama de secuencia para el caso de uso "Ingresar alcabala"

Diagrama de secuencia para el caso de uso *Liquidar alcabala*, para el actor Jefe de Rentas:

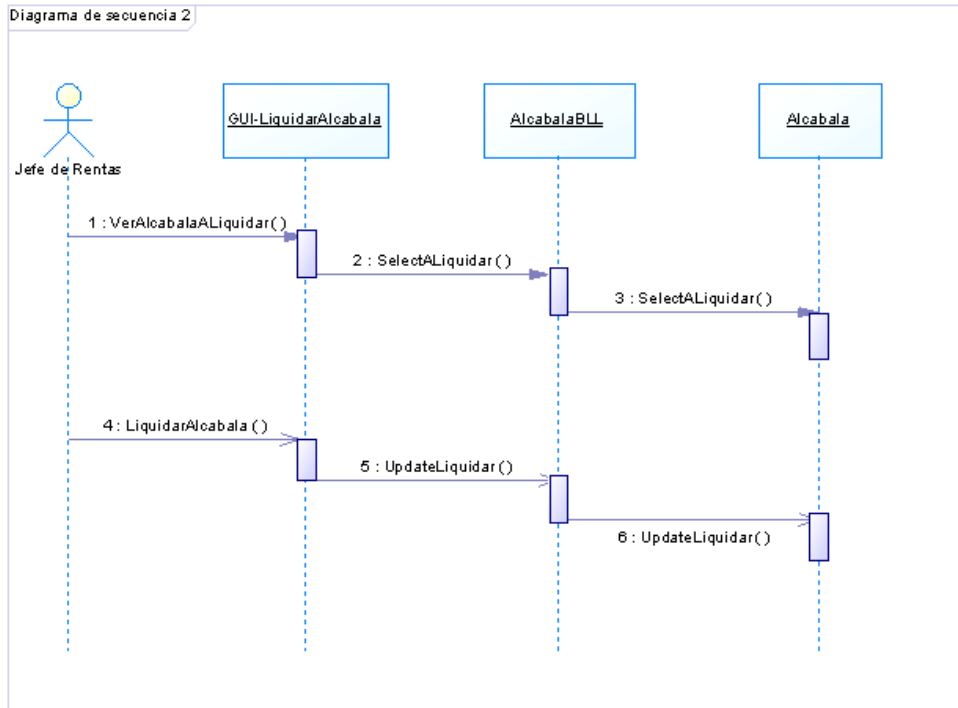


Figura N° 6.9. Diagrama de secuencia para el caso de uso "Liquidar alcabala"

Diagrama de secuencia para el caso de uso *Pagar alcabala*, para el actor Recaudador:

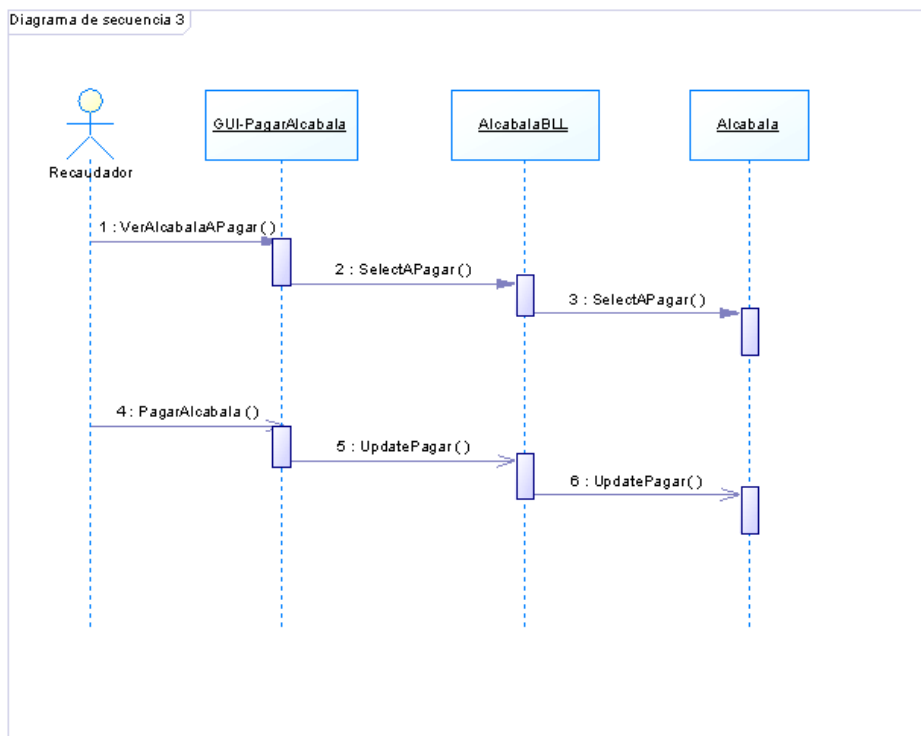


Figura N° 6.10. Diagrama de secuencia para el caso de uso "Pagar alcabala"

Diagrama de secuencia para el caso de uso *Generar reporte alcabalas*, para el actor Administrador:

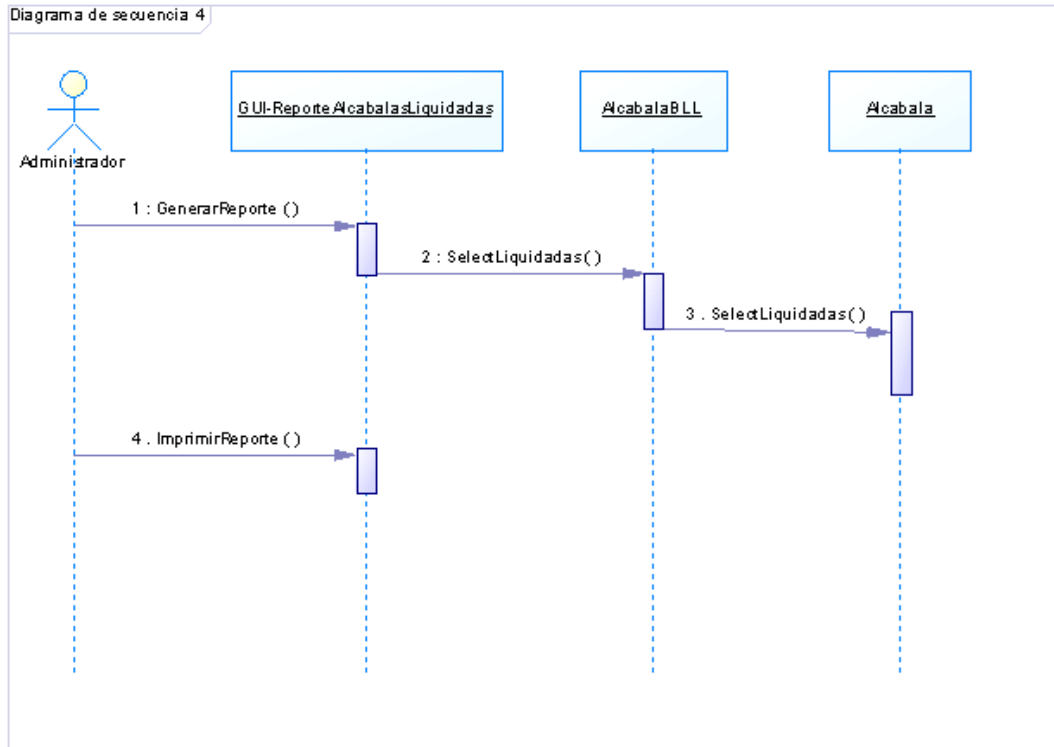


Figura N° 6.11. Diagrama de secuencia para el caso de uso "Generar reporte alcabalas"

6.4.3. Diseño de la base de datos

6.4.3.1. Diseño lógico

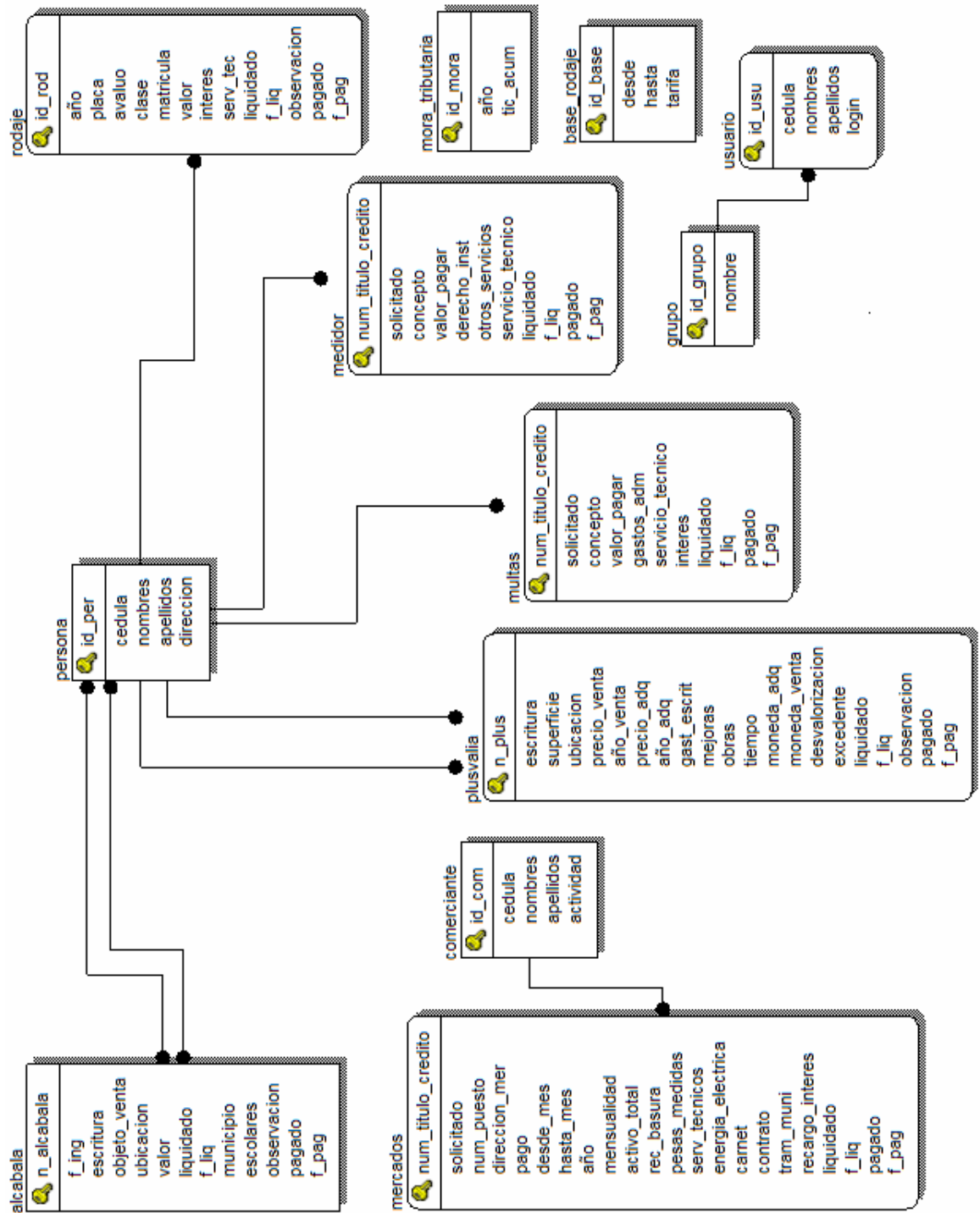


Figura N° 6.12. Diseño lógico de la base de datos

6.4.4. Diseño de interfaces

Las interfaces son esenciales para la comunicación entre los usuarios y el sistema, de modo que su diseño está orientado a que sean amigables y fáciles de usar.

6.4.4.1. Diseño de interfaces de entrada

Las interfaces de entrada permiten a los usuarios ingresar información al sistema. Tienen los siguientes elementos:

- Título.
- Botones para minimizar, restaurar y cerrar la ventana.
- Campos para ingreso de datos.
- Botones para aceptar y cancelar la operación de ingreso.

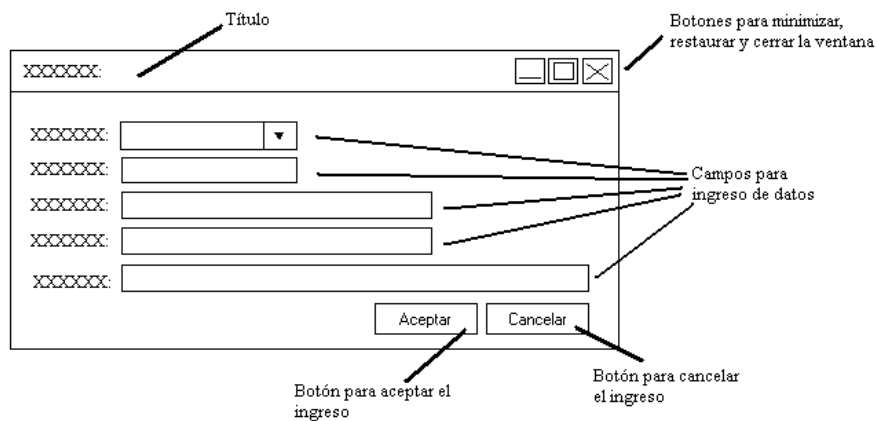


Figura N° 6.14. Diseño de interfaz de entrada

Algunos ejemplos de interfaz de entrada se muestran a continuación:

The screenshot shows a window titled "Ingreso de Alcabala". It features several input fields and buttons. At the top, there are three window control icons (minimize, maximize, close). The main content area includes:

- A text box for "Nº Alcabala".
- A text box for "Escritura de:".
- A list area for "Vendedor(es)" with "Agregar" and "Borrar" buttons to its right.
- A list area for "Comprador(es)" with "Agregar" and "Borrar" buttons to its right.
- A text box for "Objeto de venta:".
- A text box for "Ubicación:".
- A text box for "Valor:".
- At the bottom right, there are "Aceptar" and "Cancelar" buttons.

Figura N° 6.15. Interfaz de entrada. Ingreso de alcabala

The screenshot shows a window titled "Ingresar Persona". It features several input fields and buttons. At the top, there are three window control icons (minimize, maximize, close). The main content area includes:

- A dropdown menu for "Tipo:" with a downward arrow.
- A text box for "Cédula:".
- A text box for "Nombres:".
- A text box for "Apellidos:".
- A text box for "Dirección:".
- At the bottom right, there are "Aceptar" and "Cancelar" buttons.

Figura N° 6.16. Interfaz de entrada. Ingreso de persona

Figura N° 6.17. Interfaz de entrada. Liquidación de alcabala

6.4.4.2. Diseño de interfaces de salida

Las interfaces de salida permiten a los usuarios obtener y visualizar información del sistema, básicamente son reportes de liquidaciones y recaudación. Tienen los siguientes elementos:

- Título.
- Botones para minimizar, restaurar y cerrar ventana.
- Campos de selección de fechas para determinar el intervalo de tiempo del cual se desea obtener el reporte.
- Botón para visualizar los datos.
- Botón para visualizar resumen general.
- Botón para mandar a imprimir el reporte.
- Área de visualización de datos.

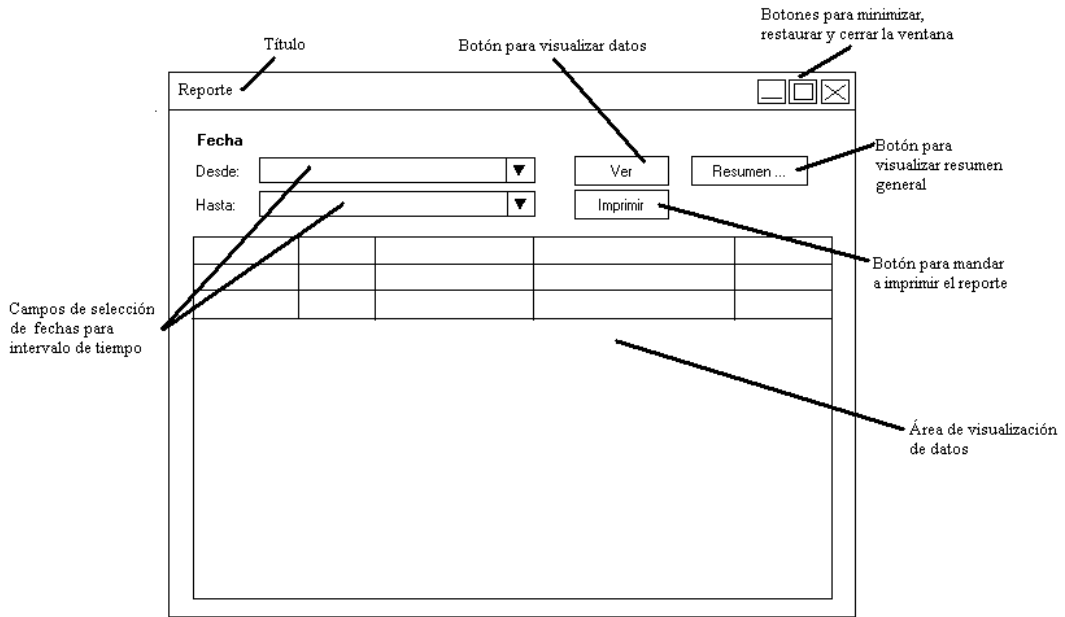


Figura N° 6.18. Diseño de interfaz de salida

Ejemplo de interfaz de salida (reporte):

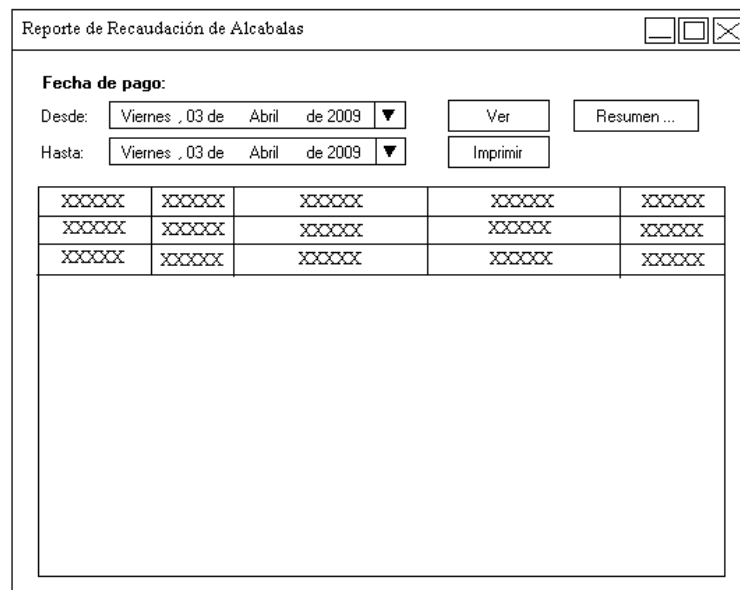


Figura N° 6.19. Interfaz de salida. Reporte

6.5. Desarrollo

6.5.1. Desarrollo de la base de datos

6.5.1.1. Creación de la base de datos

Para crear la base de datos en el motor PostgreSQL 8.3 primero se debe ingresar a la consola de administración pgAdmin III. Para ello ir a: Inicio > Programas > PostgreSQL 8.3 > pgAdmin III:

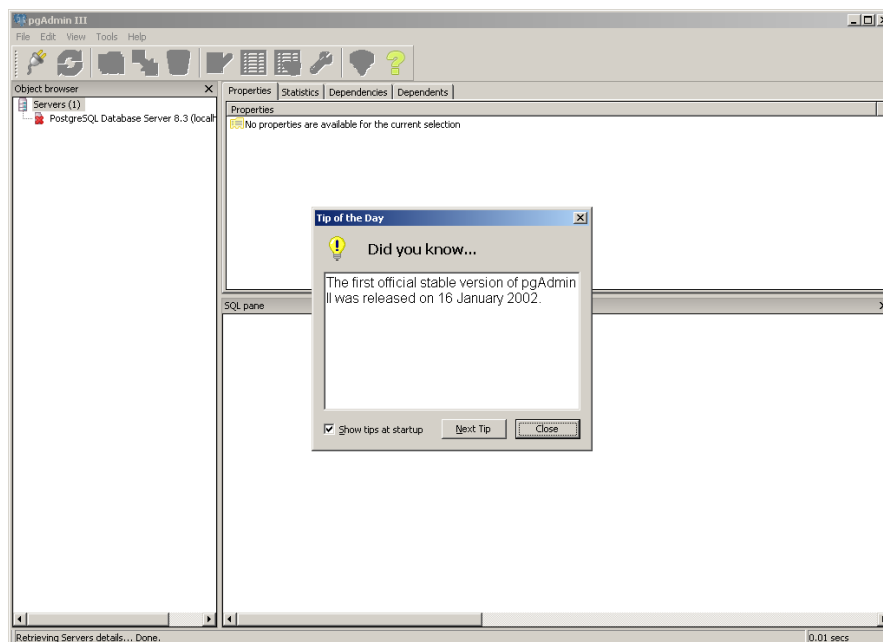


Figura N° 6.20. Consola de administración pgAdmin III

Para conectarse doble clic en “PostgreSQL Database Server 8.3 (localhost:5432)”, aparece una ventana en la que se debe ingresar la clave de súper usuario. OK:

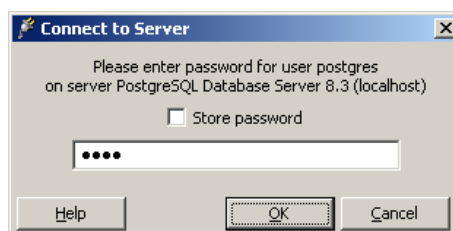


Figura N° 6.21. Ventana de autenticación para ingresar a pgAdmin III

Para crear la nueva base de datos: en el árbol Object browser, clic derecho en Databases > New Database...

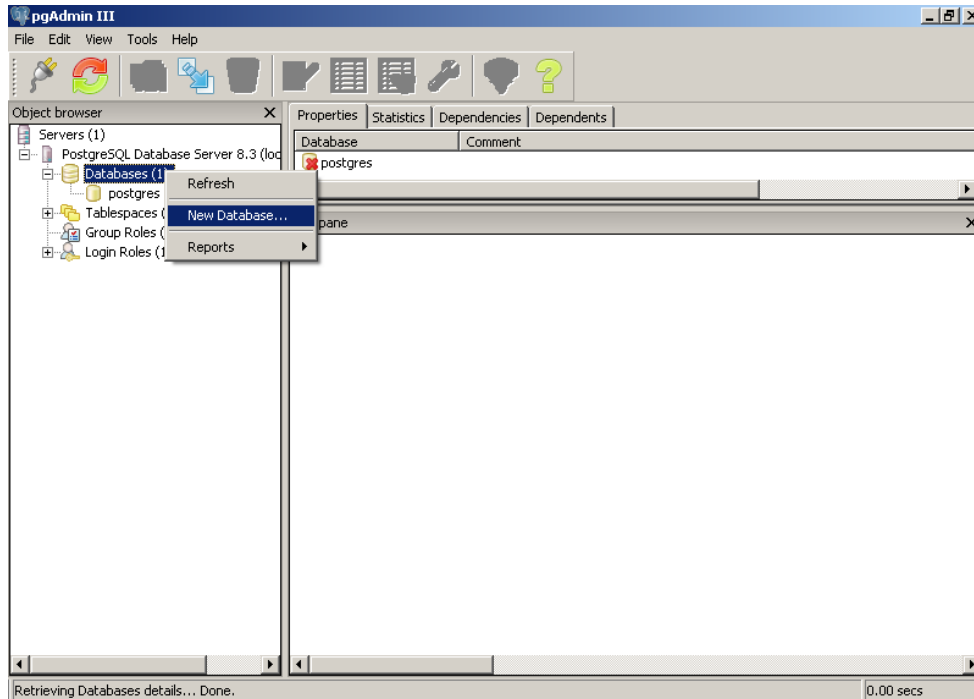


Figura N° 6.22. Creación de una nueva base de datos

En la ventana New Database... escribir: el nombre de la base de datos “rentas”, definimos como propietario de la base al súper usuario postgres y Tablespace a pg_default:

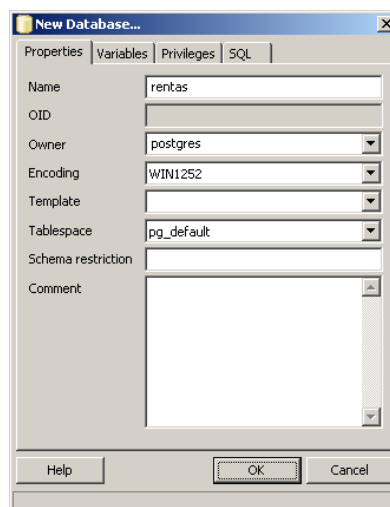


Figura N° 6.23. Creación de la base de datos “rentas”

Clic en OK y se crea la base de datos “rentas”:

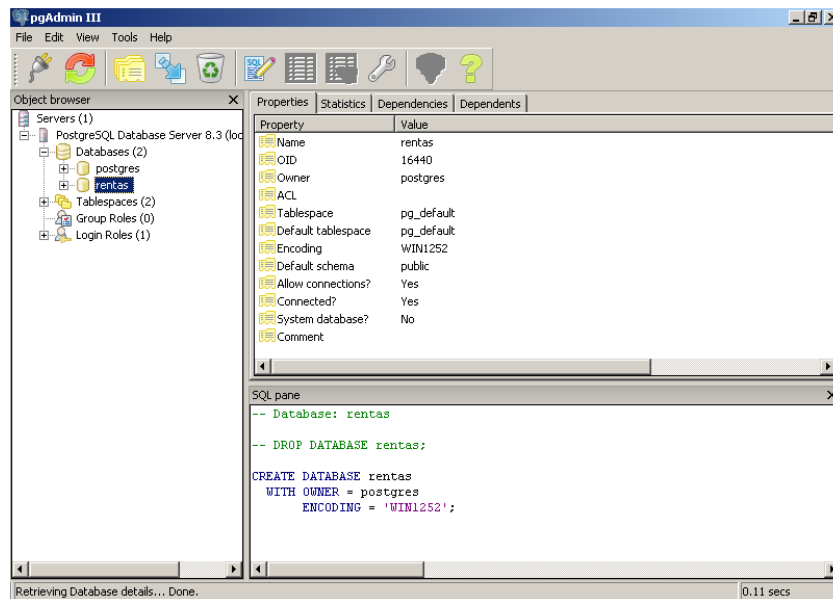


Figura N° 6.24. Base de datos “rentas” creada

Ahora se debe crear un esquema dentro de la base de datos “rentas”, para esto en el árbol Object browser dentro de la base de datos “rentas”, clic derecho en Schemas > New Schema...

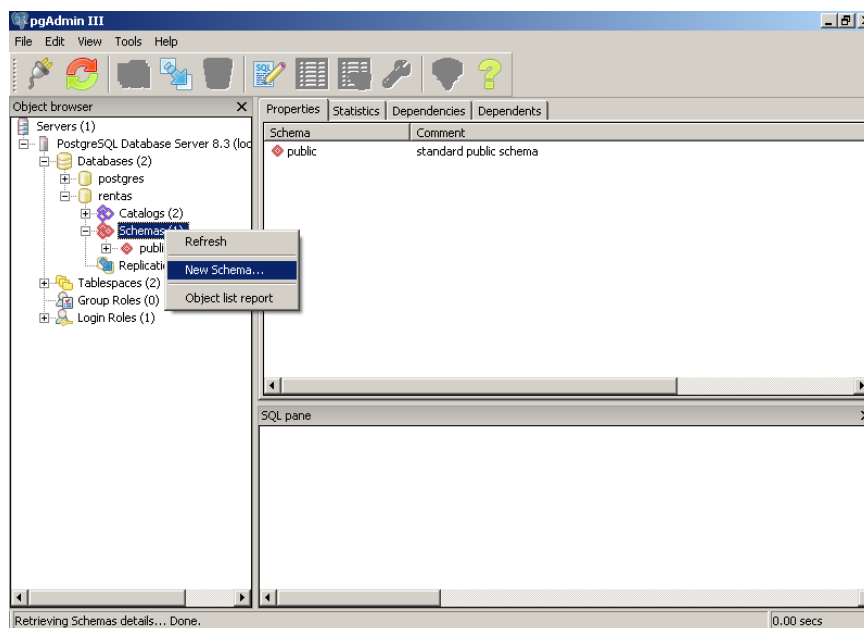


Figura N° 6.25. Creación de un nuevo esquema

En la ventana New Schema... escribir: el nombre del esquema como “esq” y definir como propietario al súper usuario postgres:

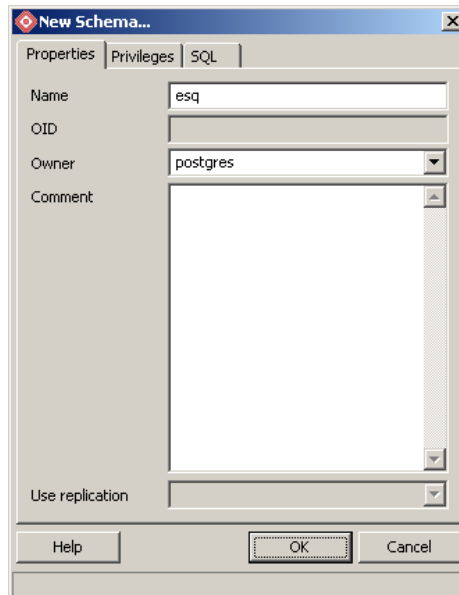


Figura N° 6.26. Creación del esquema “esq”

Clic en OK y se crea el esquema “esq”:

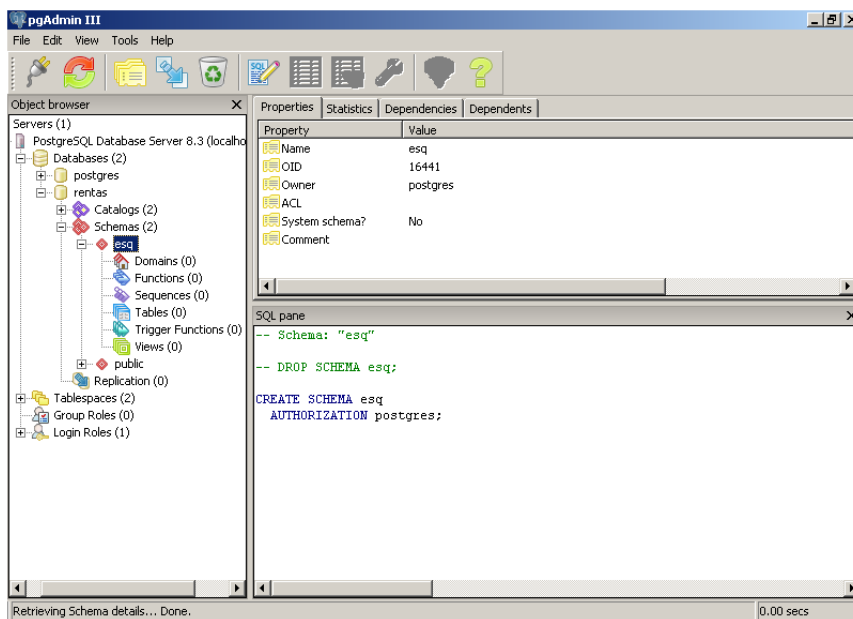


Figura N° 6.27. Esquema “esq” creado

Otra forma de hacer lo anterior es mediante la ejecución de código SQL:

Creación de la base de datos “rentas”:

```
CREATE DATABASE rentas WITH OWNER = postgres ENCODING = 'WIN1252';
```

Creación del esquema “esq”:

```
CREATE SCHEMA esq AUTHORIZATION postgres;
```

6.5.1.2. Creación de tablas y secuencias

Las tablas y secuencias son creadas mediante la ejecución de código SQL. Para esto seleccionamos en el Object browser a la base “rentas” y clic en el ícono

“Execute arbitrary SQL queries”  :

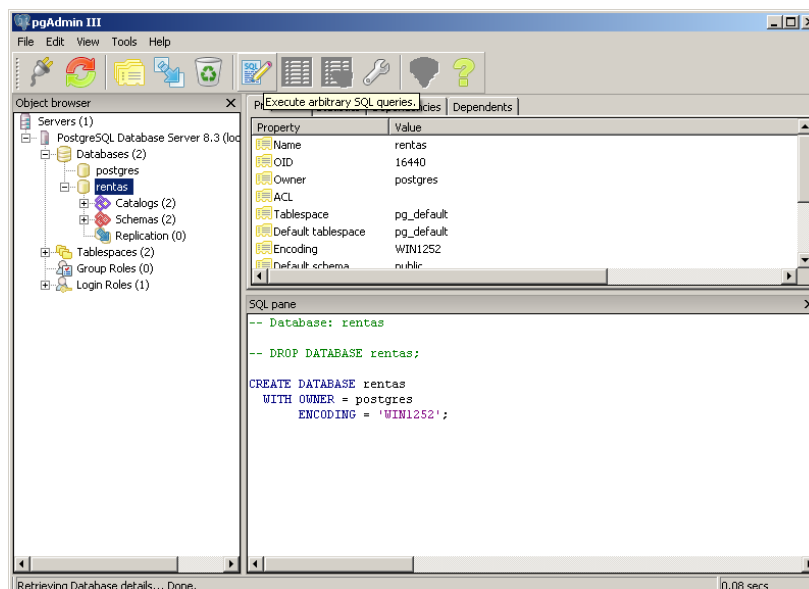


Figura N° 6.28. Cómo ejecutar código SQL

Aparece la ventana:

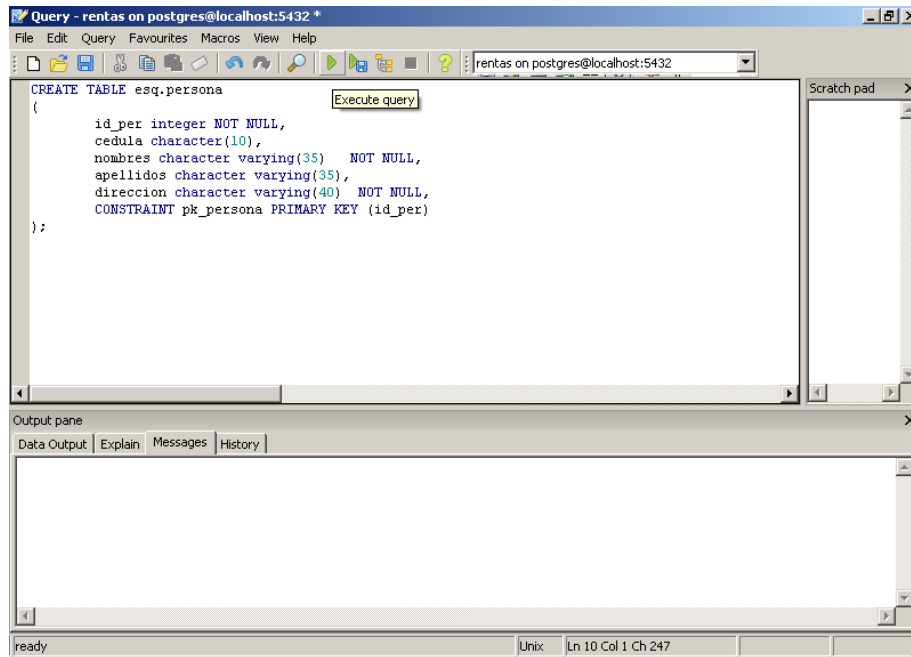



Figura N° 6.29. Ejecución de código SQL

Aquí escribir el código para ejecutarlo clic en el icono “Execute query”  y se crearán de esta manera los objetos:

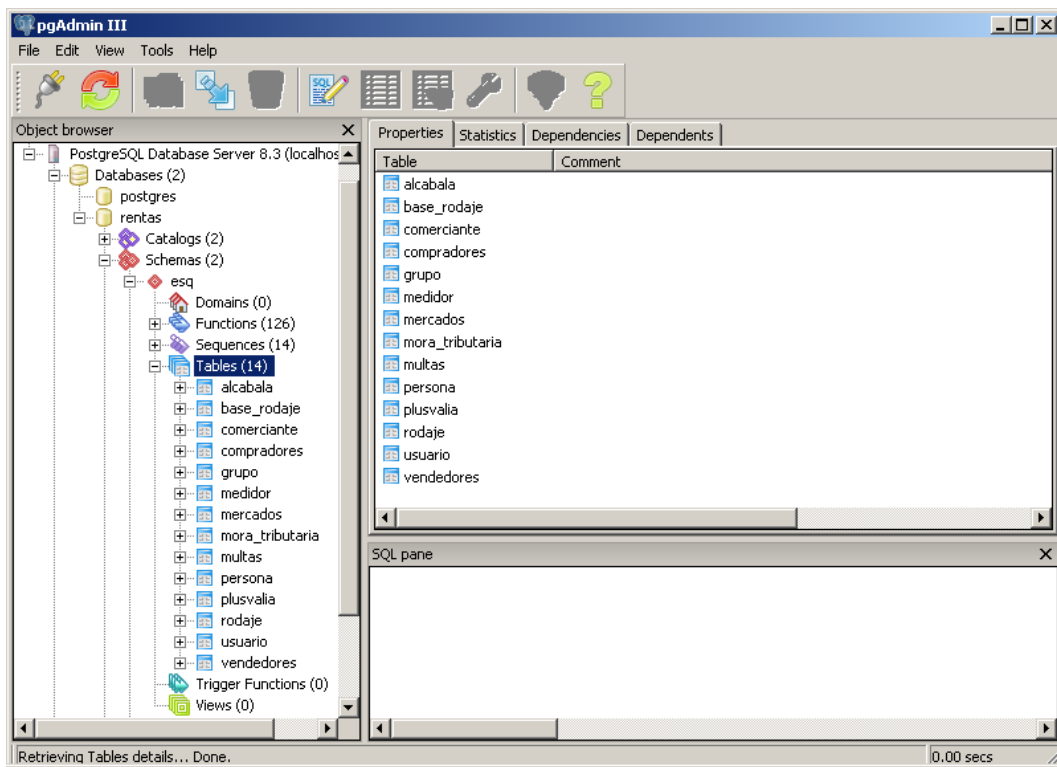


Figura N° 6.30. Objetos creados en el esquema

El código SQL completo se muestra a continuación:

Creación de la tabla “persona”:

```
CREATE TABLE esq.persona
(
    id_per integer NOT NULL,
    cedula character(10),
    nombres character varying(35) NOT NULL,
    apellidos character varying(35),
    direccion character varying(40) NOT NULL,
    CONSTRAINT pk_persona PRIMARY KEY (id_per)
);
```

Creación de la secuencia “seq_persona”:

```
CREATE SEQUENCE esq.seq_persona
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 1
CACHE 1;
```

Creación de la tabla “vendedores”:

```
CREATE TABLE esq.vendedores
(
    id_vends bigserial,
    n_alcabala bigint NOT NULL,
    id_vendedor integer NOT NULL,
    CONSTRAINT pk_vendedores PRIMARY KEY (id_vends),
```

```

        CONSTRAINT fk_vende FOREIGN KEY (id_vendedor) REFERENCES
esq.persona (id_per),  CONSTRAINT fk_alcabala_vend FOREIGN KEY (n_alcabala)
REFERENCES esq.alcabala (n_alcabala)
);

```

Creación de la tabla “compradores”:

```

CREATE TABLE esq.compradores
(
    id_comps bigserial,
    n_alcabala bigint NOT NULL,
    id_comprador integer NOT NULL,
    CONSTRAINT pk_compradores PRIMARY KEY (id_comps),
    CONSTRAINT fk_compra FOREIGN KEY (id_comprador) REFERENCES
esq.persona (id_per),
    CONSTRAINT fk_alcabala_comp FOREIGN KEY (n_alcabala) REFERENCES
esq.alcabala (n_alcabala)
);

```

Creación de la tabla “alcabala”:

```

CREATE TABLE esq.alcabala
(
    n_alcabala bigint,
    f_ing timestamp without time zone NOT NULL,
    escritura character varying(60) NOT NULL,
    objeto_venta text NOT NULL,
    ubicacion character varying(40) NOT NULL,
    valor numeric(10,2) NOT NULL,
    liquidado boolean NOT NULL,
    f_liq timestamp without time zone,
    municipio numeric(10,2),
    escolares numeric(10,2),
    observacion text,
    pagado boolean NOT NULL,

```

```

    cobrado_por integer,
    f_pag timestamp without time zone,
    CONSTRAINT pk_alcabala PRIMARY KEY (n_alcabala),
    CONSTRAINT fk_usuario FOREIGN KEY (cobrado_por) REFERENCES
esq.usuario(id_usu)
);

```

Creación de la secuencia “seq_alcabala”:

```

CREATE SEQUENCE esq.seq_alcabala
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 1
CACHE 1;

```

Creación de la tabla “plusvalia”:

```

CREATE TABLE esq.plusvalia
(
    n_plus integer,
    escritura character varying(30) NOT NULL,
    id_vendedor integer NOT NULL,
    id_comprador integer NOT NULL,
    superficie numeric(10,2) NOT NULL,
    ubicacion character varying(35) NOT NULL,
    precio_venta numeric(10,2) NOT NULL,
    año_venta numeric(4) NOT NULL,
    precio_adq numeric(10,2) NOT NULL,
    año_adq numeric(4) NOT NULL,
    gast_escrit numeric(10,2) NOT NULL,
    mejoras numeric(10,2) NOT NULL,
    obras numeric(10,2) NOT NULL,
    tiempo numeric(10,2) NOT NULL,
    moneda_adq numeric(10,2) NOT NULL,

```

```

moneda_venta numeric(10,2) NOT NULL,
desvalorizacion numeric(10,2) NOT NULL,
excedente numeric(10,2) NOT NULL,
liquidado boolean NOT NULL,
f_liq timestamp without time zone NOT NULL,
observacion character varying(35) NOT NULL,
pagado boolean NOT NULL,
cobrado_por integer,
f_pag timestamp without time zone,
CONSTRAINT pk_plusvalia PRIMARY KEY (n_plus),
CONSTRAINT fk_compra FOREIGN KEY (id_comprador) REFERENCES
esq.persona (id_per),
CONSTRAINT fk_vende FOREIGN KEY (id_vendedor) REFERENCES
esq.persona (id_per),
CONSTRAINT fk_usuario FOREIGN KEY (cobrado_por) REFERENCES
esq.usuario (id_usu)
);

```

Creación de la tabla “rodaje”:

```

CREATE TABLE esq.rodaje
(
    id_rod serial,
    año numeric(4) NOT NULL,
    id_per integer NOT NULL,
    placa character(7) NOT NULL,
    avaluo numeric(8,2) NOT NULL,
    clase character varying(35) NOT NULL,
    matricula character(6) NOT NULL,
    valor numeric(8,2) NOT NULL,
    interes numeric(8,2) NOT NULL,
    serv_tec numeric(8,2) NOT NULL,
    liquidado boolean NOT NULL,
    f_liq timestamp without time zone NOT NULL,
    observacion character varying(35) NOT NULL,

```



```

    pagado boolean NOT NULL,
    cobrado_por integer,
    f_pag timestamp without time zone,
    CONSTRAINT pk_rodaje PRIMARY KEY (id_rod),
    CONSTRAINT uni_anio UNIQUE (año, placa),
    CONSTRAINT fk_person FOREIGN KEY (id_per) REFERENCES esq.persona
(id_per),
    CONSTRAINT fk_usuario FOREIGN KEY (cobrado_por) REFERENCES
esq.usuario (id_usu)
);

```

Creación de la tabla “base_rodaje”:

```

CREATE TABLE esq.base_rodaje
(
    id_base integer,
    desde numeric(8,2) NOT NULL,
    hasta numeric(8,2) NOT NULL,
    tarifa numeric(8,2) NOT NULL,
    CONSTRAINT pk_base_rodaje PRIMARY KEY (id_base),
    CONSTRAINT uni_base_rodaje UNIQUE (desde, hasta)
);

```

Creación de la secuencia “seq_base_rodaje”:

```

CREATE SEQUENCE esq.seq_base_rodaje
INCREMENT 1
MINVALUE 1
MAXVALUE 100
START 1
CACHE 1;

```

Creación de la tabla “mora_tributaria”:

```

CREATE TABLE esq.mora_tributaria
(
id_mora integer,
año numeric(4) NOT NULL,
tic_acum numeric(8,2) NOT NULL,
CONSTRAINT pk_mora_tribu PRIMARY KEY (id_mora),
CONSTRAINT uni_año_mora_tri UNIQUE (año)
);

```

Creación de la secuencia “seq_mora_tributaria”:

```

CREATE SEQUENCE esq.seq_mora_tributaria
INCREMENT 1
MINVALUE 1
MAXVALUE 1000
START 1
CACHE 1;

```

Creación de la tabla “comerciante”:

```

CREATE TABLE esq.comerciante
(
    id_com integer,
    cedula character(10) NOT NULL,
    nombres character varying(35) NOT NULL,
    apellidos character varying(35) NOT NULL,
    actividad character varying(40) NOT NULL,
    CONSTRAINT pk_comerciant PRIMARY KEY (id_com)
);

```

Creación de la secuencia “seq_comerciante”;

```

CREATE SEQUENCE esq.seq_comerciante
INCREMENT 1

```

MINVALUE 1
MAXVALUE 2147483647
START 1
CACHE 1;

Creación de la tabla “mercados”:

```
CREATE TABLE esq.mercados
(
    num_titulo_credito serial,
    solicitado character varying(35) NOT NULL,
    id_com integer NOT NULL,
    num_puesto integer NOT NULL,
    direccion_mer character varying(40) NOT NULL,
    pago character varying(9) NOT NULL,
    desde_mes character varying(10) NOT NULL,
    hasta_mes character varying(10) NOT NULL,
    año numeric(4) NOT NULL,
    mensualidad numeric(6,2) NOT NULL,
    activo_total numeric(6,2) NOT NULL,
    rec_basura numeric(6,2) NOT NULL,
    pesas_medidas numeric(6,2) NOT NULL,
    serv_tecnicos numeric(6,2) NOT NULL,
    energia_electrica numeric(6,2) NOT NULL,
    carnet numeric(6,2) NOT NULL,
    contrato numeric(6,2) NOT NULL,
    tram_muni numeric(6,2) NOT NULL,
    recargo_interes numeric(6,2) NOT NULL,
    liquidado boolean NOT NULL,
    f_liq timestamp without time zone NOT NULL,
    pagado boolean NOT NULL,
    cobrado_por integer,
    f_pag timestamp without time zone,
    CONSTRAINT pk_mercados PRIMARY KEY (num_titulo_credito),
```

```

        CONSTRAINT fk_comerciant FOREIGN KEY (id_com) REFERENCES
esq.comerciante (id_com),
        CONSTRAINT fk_usuario FOREIGN KEY (cobrado_por) REFERENCES
esq.usuario (id_usu)
);

```

Creación de la tabla “multas”:

```

CREATE TABLE esq.multas
(
    num_titulo_credito serial,
    solicitado character varying(35) NOT NULL,
    id_per integer NOT NULL,
    concepto text NOT NULL,
    valor_pagar numeric(8,2) NOT NULL,
    gastos_adm numeric(8,2) NOT NULL,
    servicio_tecnico numeric(8,2) NOT NULL,
    interes numeric(8,2) NOT NULL,
    liquidado boolean NOT NULL,
    f_liq timestamp without time zone NOT NULL,
    pagado boolean NOT NULL,
    cobrado_por integer,
    f_pag timestamp without time zone,
    CONSTRAINT pk_multas PRIMARY KEY (num_titulo_credito),
    CONSTRAINT fk_person FOREIGN KEY (id_per) REFERENCES esq.persona
(id_per),
    CONSTRAINT fk_usuario FOREIGN KEY (cobrado_por) REFERENCES
esq.usuario (id_usu)
);

```

Creación de la tabla “medidor”:

```

CREATE TABLE esq.medidor
(
    num_titulo_credito serial,

```

```

solicitado character varying(35) NOT NULL,
id_per integer NOT NULL,
concepto text NOT NULL,
valor_pagar numeric(8,2) NOT NULL,
derecho_inst numeric(8,2) NOT NULL,
otros_servicios numeric(8,2) NOT NULL,
servicio_tecnico numeric(8,2) NOT NULL,
liquidado boolean NOT NULL,
f_liq timestamp without time zone NOT NULL,
pagado boolean NOT NULL,
cobrado_por integer,
f_pag timestamp without time zone,
CONSTRAINT pk_medidor PRIMARY KEY (num_titulo_credito),
CONSTRAINT fk_person FOREIGN KEY (id_per) REFERENCES esq.persona
(id_per),
CONSTRAINT fk_usuario FOREIGN KEY (cobrado_por) REFERENCES
esq.usuario (id_usu)
);

```

Creación de la tabla “grupo”:

```

CREATE TABLE esq.grupo
(
    id_grupo serial,
    nombre character varying(35) NOT NULL,
    CONSTRAINT pk_grupo PRIMARY KEY (id_grupo)
);

```

Creación de la tabla “usuario”:

```

CREATE TABLE esq.usuario
(
    id_usu integer,
    cedula character(10) NOT NULL,
    nombres character varying(35) NOT NULL,

```

```

apellidos character varying(35) NOT NULL,
login character varying(35) NOT NULL,
id_grupo integer NOT NULL,
CONSTRAINT pk_usuario PRIMARY KEY (id_usu),
CONSTRAINT fk_grupo FOREIGN KEY (id_grupo) REFERENCES esq_grupo
(id_grupo)
);

```

Creación de la secuencia “seq_usuario”:

```

CREATE SEQUENCE esq.seq_usuario
INCREMENT 1
MINVALUE 1
MAXVALUE 2147483647
START 1
CACHE 1;

```

6.5.1.3. Descripción de los campos de las tablas

A continuación se muestra la descripción de los campos de las tablas de la base de datos.

Tabla “persona”

La tabla “persona” contiene los datos de las personas que realizan los trámites.

Campo	Descripción
id_per	Clave primaria.
cedula	Cédula de identidad de la persona.
nombres	Nombres de la persona.
apellidos	Apellidos de la persona.
direccion	Dirección de la persona.

Tabla N° 6.62. Descripción de los campos de la tabla “persona”

Tabla “vendedores”

Campo	Descripción
id_vends	Clave primaria
n_alcabala	Clave foránea de la tabla “alcabala”
id_vendedor	Clave foránea de la tabla “persona”

Tabla N° 6.63. Descripción de los campos de la tabla “vendedores”

Tabla “compradores”

Campo	Descripción
id_comps	Clave primaria
n_alcabala	Clave foránea de la tabla “alcabala”
id_comprador	Clave foránea de la tabla “persona”

Tabla N° 6.64. Descripción de los campos de la tabla “compradores”

Tabla “alcabala”

La tabla “alcabala” contiene información de los trámites de alcabala.

Campo	Descripción
n_alcabala	Clave primaria.
f_ing	Fecha en que se ingresa un trámite de alcabala.
escritura	Tipo de escritura del trámite.
objeto_venta	Objeto de venta o lote que se esta vendiendo.
ubicacion	Ubicación del objeto de venta.
valor	Valor o avalúo del objeto de venta.
liquidado	Indica si el trámite de alcabala está o no liquidado.
f_liq	Fecha en la que se realizó la liquidación.
municipio	Valor del impuesto al municipio.
escolares	Valor del impuesto para construcciones escolares.
observacion	Observación.
pagado	Indica si el trámite de alcabala está o no pagado.
cobrado_por	Indica por cual recaudador fue cobrado.
f_pag	Fecha de pago de la alcabala.

Tabla N° 6.65. Descripción de los campos de la tabla “alcabala”

Tabla “plusvalía”

La tabla “plusvalia” contiene información sobre los trámites de plusvalías.

Campo	Descripción
n_plus	Clave primaria
escritura	Tipo de escritura del trámite.
id_vendedor	Clave foránea de la tabla “persona”
id_comprador	Clave foránea de la tabla “persona”
superficie	Superficie del lote.
ubicacion	Ubicación del lote.
precio_venta	Precio en que se vende el lote.
año_venta	Año en que se vende el lote.
precio_adq	Precio en que se adquirió el lote.
año_adq	Año en que se adquirió el lote.
gast_escrit	Valor de los gastos de escritura.
mejoras	Valor de la contribución de mejoras.
obras	Valor de obras realizadas.
tiempo	Valor del tiempo transcurrido.
moneda_adq	Valor de la moneda en el año de adquisición del lote.
moneda_venta	Valor de la moneda en el año de venta del lote.
desvalorizacion	Valor del porcentaje de desvalorización
excedente	Valor del impuesto sobre la fracción excedente.
liquidado	Indica si el trámite de plusvalía está o no liquidado.
f_liq	Fecha de liquidación de la plusvalía.
observacion	Observación.
pagado	Indica si el trámite de plusvalía está o no pagado.
cobrado_por	Indica por cual recaudador fue cobrado.
f_pag	Fecha de pago de la plusvalía.

Tabla N° 6.66. Descripción de los campos de la tabla “plusvalia”

Tabla “rodaje”

La tabla “rodaje” contiene información sobre los trámites de rodajes vehiculares.

Campo	Descripción
id_rod	Clave primaria.
año	Año del cual se está realizando el rodaje.
id_per	Clave foránea de la tabla “persona”.
placa	Placa del vehículo.
avaluo	Avalúo del vehículo.
clase	Clase de vehículo.
matricula	Nº matrícula
valor	Valor a pagar.
interes	Valor del interés.
serv_tec	Valor del servicio técnico administrativo.
liquidado	Indica si el trámite de rodaje está o no liquidado.
f_liq	Fecha de liquidación de rodaje.
observacion	Observación.
pagado	Indica si el trámite de rodaje está o no pagado.
cobrado_por	Indica por cual recaudador fue cobrado.
f_pag	Fecha de pago de rodaje.

Tabla Nº 6.67. Descripción de los campos de la tabla “rodaje”

Tabla “base_rodaje”

La tabla “base_rodaje” contiene información sobre la base imponible para el cálculo requerido en la liquidación de rodajes vehiculares.

Campo	Descripción
id_base	Clave primaria.
desde	Indica el límite inferior del intervalo.
hasta	Indica el límite superior del intervalo.
tarifa	Indica el valor o base imponible.

Tabla Nº 6.68. Descripción de los campos de la tabla “base_rodaje”

Tabla “mora_tributaria”

La tabla “mora_tributaria” contiene los valores de los intereses por mora tributaria, para el cálculo en la liquidación de rodaje vehicular.

Campo	Descripción
id_mora	Clave primaria.
año	Año del cual se quiere saber el valor.
tic_acum	Valor de interés

Tabla N° 6.69. Descripción de los campos de la tabla “mora tributaria”

Tabla “comerciante”

La tabla “comerciante” contiene la información de los comerciantes que ocupan puestos en las plazas y mercados.

Campo	Descripción
id_com	Clave primaria.
cedula	Cédula de identidad del comerciante.
nombres	Nombres del comerciante.
apellidos	Apellidos del comerciante.
actividad	Actividad específica del comerciante.

Tabla N° 6.70. Descripción de los campos de la tabla “comerciante”

Tabla “mercados”

La tabla “mercados” contiene información sobre los pagos de los puestos de los mercados y plazas.

Campo	Descripción
num_titulo_credito	Clave primaria.
solicitado	Indica por cual persona o departamento fue solicitado este trámite.
id_com	Clave foránea de la tabla “comerciante”
num_puesto	Número del puesto en la plaza o mercado.
direccion_mer	Dirección de la plaza o mercado.
pago	Indica si el pago es de todo el año o de algunos meses.
desde_mes	Indica desde que mes se está pagando.
hasta_mes	Indica hasta que mes se está pagando.
año	Indica el año.
mensualidad	Valor de la mensualidad.

activo_total	Valor del activo total.
rec_basura	Valor por recolección de basura.
pesas_medidas	Valor por aferición de pesas y medidas.
serv_tecnicos	Valor por servicios técnicos.
energia_electrica	Valor de la energía eléctrica.
carnet	Valor del carnet.
contrato	Valor del contrato.
tram_muni	Valor por trámite municipal.
recargo_interes	Valor de recargos y/o intereses.
liquidado	Indica si el trámite está o no liquidado.
f_liq	Fecha de liquidación.
pagado	Indica si el trámite está o no cobrado.
cobrado_por	Indica por cual recaudador fue cobrado.
f_pag	Fecha de pago.

Tabla N° 6.71. Descripción de los campos de la tabla “mercados”

Tabla “multas”

La tabla “multas” almacena información sobre las multas emitidas.

Campo	Descripción
num_titulo_credito	Clave primaria.
solicitado	Indica por cual persona o departamento fue solicitado la emisión de la multa.
id_per	Clave foránea de la tabla “persona”.
concepto	Concepto o razón de la multa.
valor_pagar	Valor de la multa.
gastos_adm	Valor por gastos administrativos.
servicio_tecnico	Valor por servicio técnico.
interes	Valor del interés.
liquidado	Indica si la multa está o no emitida.
f_liq	Fecha de emisión.
pagado	Indica si la multa está o no pagado.
cobrado_por	Indica por cual recaudador fue cobrado.
f_pag	Fecha de pago de la multa.

Tabla N° 6.72. Descripción de los campos de la tabla “multas”

Tabla “medidor”

La tabla “medidor” contiene información sobre las ventas de medidores de agua.

Campo	Descripción
num_titulo_credito	Clave primaria.
solicitado	Indica por cual persona o departamento fue solicitado el trámite.
id_per	Clave foránea de la tabla “persona”.
concepto	Concepto o razón del trámite.
valor_pagar	Valor a pagar.
derecho_inst	Valor por derecho de instalación.
otros_servicios	Valor por otros servicios.
servicio_tecnico	Valor del servicio técnico.
liquidado	Indica si el trámite está o no liquidado.
f_liq	Fecha de emisión.
pagado	Indica si el trámite está o no pagado.
cobrado_por	Indica por cual recaudador fue cobrado.
f_pag	Fecha de pago.

Tabla N° 6.73. Descripción de los campos de la tabla “medidor”

Tabla “grupo”

La tabla “grupo” contiene los grupos de usuario que tiene el sistema.

Campo	Descripción
id_grupo	Clave primaria.
nombre	Nombre del grupo de usuario.

Tabla N° 6.74. Descripción de los campos de la tabla “grupo”

Tabla “usuario”

La tabla “usuario” contiene los usuarios que tiene el sistema.

Campo	Descripción
id_usu	Clave primaria.
cedula	Cédula de identidad del usuario.
nombres	Nombres del usuario.
apellidos	Apellidos del usuario.
login	Login o nombre de usuario.
id_grupo	Clave foránea de la tabla "grupo".

Tabla N° 6.75. Descripción de los campos de la tabla "usuario"

6.5.1.4. Creación de procedimientos almacenados

Todas las consultas a la base de datos se realizan mediante procedimientos almacenados. Esto es porque el tiempo empleado para la consulta es menor, ya que estos procedimientos o funciones ya están compilados en el motor de bases de datos y se ejecutan directamente, no sucede así si se utiliza código SQL directo.

En PostgreSQL 8.3 se puede crear los procedimientos almacenados o funciones en varios lenguajes tales como: C, SQL, PL/pgSQL y otros. En el presente trabajo se utilizaron los lenguajes SQL y PL/pgSQL.

A continuación se muestran algunos ejemplos de las funciones o procedimientos almacenados que se crearon en el motor de base de datos:

Función (o procedimiento almacenado) que devuelve todas las filas de la tabla personas. Esta codificada en lenguaje SQL:

```
CREATE TYPE esq.tipo_per AS (idper integer, cedula character, nombres character varying);
CREATE OR REPLACE FUNCTION esq.sp_sel_per () RETURNS SETOF esq.tipo_per
AS
$BODY$
    SELECT id_per, cedula, apellidos || ' ' || nombres
    FROM esq.persona;
$BODY$
```

```
LANGUAGE 'sql';
```

Función (o procedimiento almacenado) que inserta una fila a la tabla personas. Está codificada en lenguaje PL/pgSQL:

```
CREATE OR REPLACE FUNCTION esq.sp_ins_persona (ced bpchar, nom character
varying, ape character varying, dir character varying) RETURNS integer AS
$BODY$
    BEGIN
        INSERT INTO esq.persona (id_per, cedula, nombres, apellidos, direccion)
        VALUES (nextval ('esq.seq_persona'), ced, nom, ape, dir);
        RETURN currval ('esq.seq_persona');
    END;
$BODY$
LANGUAGE 'plpgsql';
```

6.5.1.5. Usuarios y Privilegios

Grupos de usuarios

Los grupos de usuarios se corresponden con los departamentos de Avalúos y Catastros, Rentas, Tesorería y Centro de Cómputo:

Departamento	Grupo de usuarios
Avalúos y Catastros	avaluos
Rentas	rentas
Tesorería	tesoreria
Centro de Cómputo	administracion

Tabla N° 6.76. Grupos de usuarios

Cada grupo de usuarios tiene asignados los privilegios correspondientes sobre las tablas y secuencias, de acuerdo su función. Para ver el código SQL para asignar privilegios ver Anexo N° 1.

Usuarios

Los usuarios del sistema son creados desde la aplicación por el administrador del sistema, en este caso por el Jefe del Centro de Cómputo. Todo usuario pertenece a un grupo de usuarios determinado.

6.5.2. Desarrollo de la aplicación

6.5.2.1. Desarrollo de las clases

El lenguaje utilizado para la programación es C#. Todo ha sido desarrollado mediante clases. SharpDevelop permite la creación de Bibliotecas de Clases en las cuales se procede a la programación de las clases, luego a su depuración y compilación, para obtener como resultado archivos .dll que pueden ser usados en las distintas partes del proyecto, incluso en otros proyectos. Para revisar el código de algunas de las clases que se crearon ver el Anexo N° 2.

6.5.2.2. Desarrollo de la interfaz

El desarrollo de la interfaz de usuario se realizó de manera rápida puesto que la herramienta de desarrollo SharpDevelop posee un potente diseñador de Windows Forms (formularios) similar al Visual Studio .NET. Con esto se crean formularios y sus respectivos controles (botones, cajas de texto, etiquetas, grids, etc.) que permiten construir una interfaz amigable:

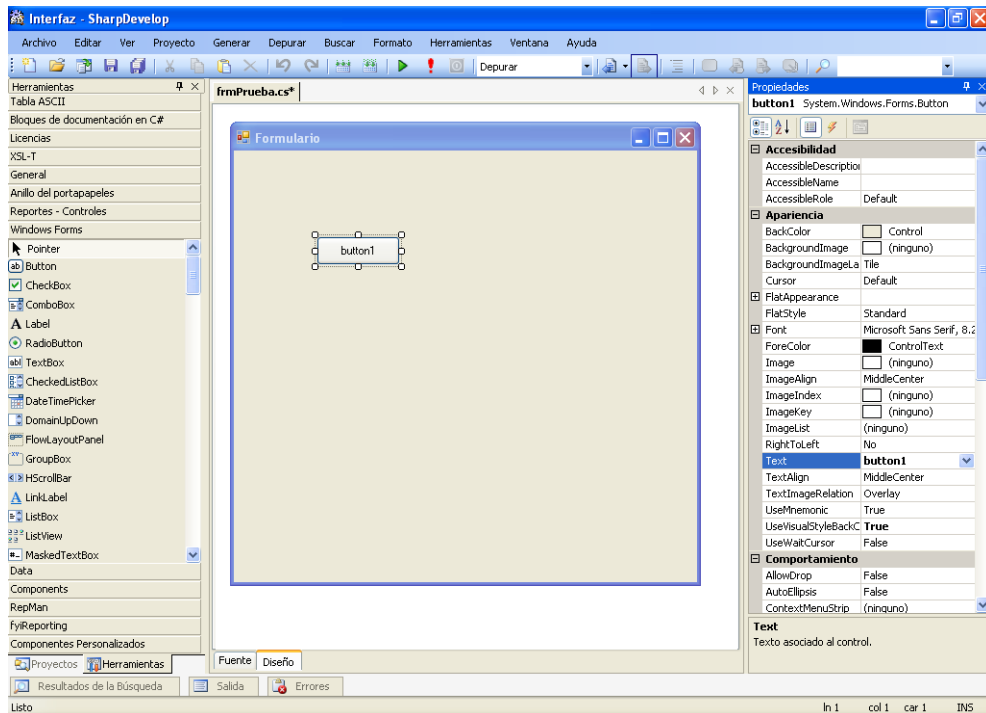


Figura N° 6.31. Entorno de desarrollo de Windows Forms de SharpDevelop

De igual forma la programación es fácil de desarrollar, pues SharpDevelop tiene un entorno amigable para crear eventos, clases, bibliotecas de clases, tiene coloreado de sintaxis para los lenguajes, etc.

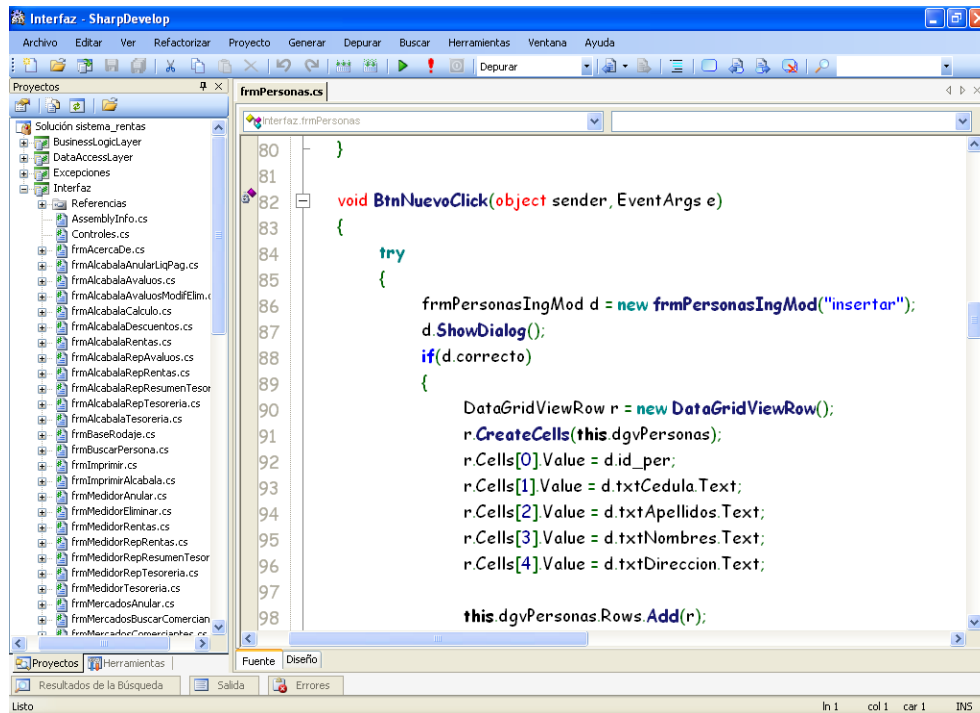


Figura N° 6.32. Entorno de programación de SharpDevelop

6.6. Implantación

La implantación del sistema se realizó de manera satisfactoria. El personal de los departamentos implicados fue debidamente capacitado. El funcionamiento del sistema es como se esperaba, y ha ayudado mucho a la institución y a la colectividad.

En el Anexo N° 3 se encuentra el manual de instalación en donde se describe las instalaciones del motor de base de datos en el equipo servidor, y de la aplicación y de otros programas adicionales en los equipos clientes. En el Anexo N° 4 se encuentra el manual de usuario que detalla el funcionamiento de la aplicación.

6.7. Conclusiones y recomendaciones

Las conclusiones que se obtienen son:

- La automatización tan anhelada por la institución se ha materializado gracias al desarrollo del presente proyecto. El problema causado por el manejo manual de los trámites es cosa del pasado.
- El desarrollo del sistema se realizó de manera satisfactoria de la manera antes descrita en este capítulo: después de analizar el problema de la institución, se realizó el análisis, diseño, para luego proceder al desarrollo mismo. Finalmente se realizaron pruebas y se puso el sistema en funcionamiento.
- Al personal de los departamentos de Avalúos y Catastros, Rentas y Tesorería se le ha facilitado el manejo de los trámites. La atención a los clientes ha mejorado notablemente, pues el tiempo para realizar los trámites se ha reducido.

- La información se mantiene de forma íntegra, consistente y está disponible permanentemente. Los usuarios hacen uso de dicha información sin riesgos de que ésta se confunda o pierda.

Algunas recomendaciones que se pueden dar son:

- Se recomienda al personal del Centro de Cómputo capacitar en el manejo del sistema al nuevo personal que pueda aparecer en lo futuro.
- Se recomienda a todos los usuarios del sistema mantener la privacidad de la información manteniendo en secreto su clave de acceso y más que todo cambiándola periódicamente.
- El I. Municipio de Quero debería seguir apoyando a la automatización de otros departamentos que así lo requieran.
- Si la institución en lo futuro necesitara el uso de alguna aplicación se aconseja que ésta sea de licencia libre, de modo que al no tener que pagar por su uso se obtenga una ayuda en el aspecto económico.

BIBLIOGRAFÍA

Páginas Web:

- <http://www.postgresql.org>
- <http://es.wikipedia.org>
- <http://www.dbrunas.com.ar/postgres/migrapg.pdf>
- <http://www.sobl.org/traduccion/practical-postgres/practical-postgres.html>
- <http://www.icsharpcode.net/OpenSource/SD/Features.aspx>

Libros:

- DATE, C. J. (1993). *Introducción a los Sistemas de bases de datos*. Quinta edición. Volumen 1. ADDISON-WESLEY IBEROAMERICANA, S.A. Wilmington, Delaware, E.U.A.
- WRIGHT, C. (2002). *Superutilidades para C#*. McGraw-Hill/Interamericana de España. Madrid. España.

Libro en pantalla:

Documentación (ayuda) de PostgreSQL 8.3

ANEXOS

ANEXO N° 1

USUARIOS Y PRIVILEGIOS

Creación de grupo de usuarios

```
CREATE GROUP rentas;  
CREATE GROUP tesoreria;  
CREATE GROUP avaluos;  
CREATE GROUP administracion;
```

Asignación de privilegios en el esquema “esq” a los grupos de usuarios

```
GRANT USAGE ON SCHEMA esq TO GROUP rentas;  
GRANT USAGE ON SCHEMA esq TO GROUP avaluos;  
GRANT USAGE ON SCHEMA esq TO GROUP tesoreria;  
GRANT USAGE ON SCHEMA esq TO GROUP administracion;
```

Asignación del privilegio para crear usuarios, al grupo de usuarios “administración”

```
ALTER ROLE administracion CREATEROLE;
```

Asignación de privilegios en los objetos del esquema “esq” a los grupos de usuario

Asignación de privilegios sobre tablas y secuencias, al grupo de usuarios “rentas”:

```
GRANT SELECT, UPDATE ON TABLE esq.alcabala TO GROUP rentas;  
GRANT SELECT ON TABLE esq.compradores TO GROUP rentas;  
GRANT SELECT ON TABLE esq.vendedores TO GROUP rentas;  
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.rodaje TO GROUP  
rentas;
```

```

GRANT USAGE ON SEQUENCE esq.rodaje_id_rod_seq TO GROUP rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.base_rodaje TO
GROUP rentas;
GRANT USAGE ON SEQUENCE esq.seq_base_rodaje TO GROUP rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.mora_tributaria TO
GROUP rentas;
GRANT USAGE ON SEQUENCE esq.seq_mora_tributaria TO GROUP rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.comerciante TO
GROUP rentas;
GRANT USAGE ON SEQUENCE esq.seq_comerciante TO GROUP rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.mercados TO
GROUP rentas;
GRANT USAGE ON SEQUENCE esq.mercados_num_titulo_credito_seq TO GROUP
rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.medidor TO GROUP
rentas;
GRANT USAGE ON SEQUENCE esq.medidor_num_titulo_credito_seq TO GROUP
rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.multas TO GROUP
rentas;
GRANT USAGE ON SEQUENCE esq.multas_num_titulo_credito_seq TO GROUP
rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.persona TO GROUP
rentas;
GRANT USAGE ON SEQUENCE esq.seq_persona TO GROUP rentas;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.plusvalia TO GROUP
rentas;
GRANT SELECT ON TABLE esq.grupo TO GROUP rentas;
GRANT SELECT ON TABLE esq.usuario TO GROUP rentas;

```

Asignación de privilegios sobre tablas y secuencias, al grupo de usuarios “avaluos”:

```

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.alcabala TO GROUP
avaluos;

```

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.compradores TO GROUP avaluos;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.vendedores TO GROUP avaluos;

GRANT USAGE ON SEQUENCE esq.seq_alcabala TO GROUP avaluos;

GRANT USAGE ON SEQUENCE esq.vendedores_id_vends_seq TO GROUP avaluos;

GRANT USAGE ON SEQUENCE esq.compradores_id_comps_seq TO GROUP avaluos;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.persona TO GROUP avaluos;

GRANT USAGE ON SEQUENCE esq.seq_persona TO GROUP avaluos;

GRANT SELECT ON TABLE esq.usuario TO GROUP avaluos;

GRANT SELECT ON TABLE esq.grupo TO GROUP avaluos;

Asignación de privilegios sobre tablas y secuencias, al grupo de usuarios “tesoreria”:

GRANT SELECT, UPDATE ON TABLE esq.alcabala TO GROUP tesoreria;

GRANT SELECT ON TABLE esq.compradores TO GROUP tesoreria;

GRANT SELECT ON TABLE esq.vendedores TO GROUP tesoreria;

GRANT SELECT, UPDATE ON TABLE esq.rodaje TO GROUP tesoreria;

GRANT SELECT ON TABLE esq.comerciante TO GROUP tesoreria;

GRANT SELECT, UPDATE ON TABLE esq.mercados TO GROUP tesoreria;

GRANT SELECT, UPDATE ON TABLE esq.medidor TO GROUP tesoreria;

GRANT SELECT, UPDATE ON TABLE esq.multas TO GROUP tesoreria;

GRANT SELECT ON TABLE esq.persona TO GROUP tesoreria;

GRANT SELECT, UPDATE ON TABLE esq.plusvalia TO GROUP tesoreria;

GRANT SELECT ON TABLE esq.usuario TO GROUP tesoreria;

GRANT SELECT ON TABLE esq.grupo TO GROUP tesoreria;

Asignación de privilegios sobre tablas, al grupo de usuarios “administracion”:

GRANT SELECT ON TABLE esq.alcabala TO GROUP administracion;

GRANT SELECT ON TABLE esq.compradores TO GROUP administracion;

GRANT SELECT ON TABLE esq.vendedores TO GROUP administracion;

```
GRANT SELECT ON TABLE esq.rodaje TO GROUP administracion;  
GRANT SELECT ON TABLE esq.comerciante TO GROUP administracion;  
GRANT SELECT ON TABLE esq.mercados TO GROUP administracion;  
GRANT SELECT ON TABLE esq.medidor TO GROUP administracion;  
GRANT SELECT ON TABLE esq.multas TO GROUP administracion;  
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.persona TO GROUP  
administracion;  
GRANT SELECT ON TABLE esq.plusvalia TO GROUP administracion;  
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE esq.usuario TO GROUP  
administracion;  
GRANT USAGE ON SEQUENCE esq.seq_usuario TO GROUP administracion;
```


ANEXO N° 2

ALGUNAS CLASES DESARROLLADAS

Algunas clases de la Capa de Acceso a los Datos (Data Access Layer)

Clase Conexion, permite la conexión con la base de datos:

```
using System;
namespace DataAccessLayer
{
    public static class Conexion
    {
        #region Datos

        private static string server = "";
        private static string port = "";
        private static string dataBase = "";
        private static string userID = "";
        private static string password = "";

        #endregion

        #region Propiedades

        public static string Server
        {
            set
            {
                server = value.Trim();
            }
            get
            {
                return server;
            }
        }

        public static string Port
        {
            set
            {
                port = value.Trim();
            }
            get
            {
                return port;
            }
        }

        public static string DataBase
        {
            set
            {
                dataBase = value.Trim();
            }
        }
    }
}
```

```

        get
        {
            return dataBase;
        }
    }

    public static string UserID
    {
        set
        {
            userID = value.Trim();
        }
        get
        {
            return userID;
        }
    }

    public static string Password
    {
        set
        {
            password = value.Trim();
        }
        get
        {
            return password;
        }
    }

    public static string CadenaConexion
    {
        get
        {
            return "Server=" + server + ";Port=" + port + ";User Id=" + userID +
";Password=" + password + ";Database=" + dataBase + ";";
        }
    }

    #endregion
}
}
}

```

Clase Persona, maneja todo lo concerniente a las personas:

```

using System;
using System.Data;
using Npgsql;
using NpgsqlTypes;

namespace DataAccessLayer
{
    public class Persona
    {
        public Persona()
        {
        }

        public Persona(string cedula, string nombres, string apellidos, string direccion)
        {
        }
    }
}

```

```

    {
        try
        {
            this.cedula = cedula;
            this.nombres = nombres;
            this.apellidos = apellidos;
            this.direccion = direccion;
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }
}

#region Datos

private string cedula;
private string nombres;
private string apellidos;
private string direccion;

#endregion

#region Propiedades

public string Cedula
{
    set { this.cedula = value;}
    get { return this.cedula;}
}

public string Nombres
{
    set { this.nombres = value;}
    get { return this.nombres;}
}

public string Apellidos
{
    set { this.apellidos = value;}
    get { return this.apellidos;}
}

public string Direccion
{
    set { this.direccion = value;}
    get { return this.direccion;}
}

#endregion

#region Métodos

public DataTable Select()
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    DataTable dtPersonas = new DataTable();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;
    }
}

```

```

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_per()";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlDataAdapter daPersona = new NpgsqlDataAdapter();
        daPersona.SelectCommand = cmdSelect;
        daPersona.Fill(dtPersonas);

        return dtPersonas;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Dispose();
    }
}

public DataTable SelectTodos()
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    DataTable dtPersonas = new DataTable();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;
        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_per2()";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlDataAdapter daPersona = new NpgsqlDataAdapter();
        daPersona.SelectCommand = cmdSelect;
        daPersona.Fill(dtPersonas);

        return dtPersonas;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Dispose();
    }
}

public string SelectDireccion(int idper)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    DataTable dtPersonas = new DataTable();
    try

```

```

        {
            cnRentas.ConnectionString = Conexion.CadenaConexion;
            NpgsqlCommand cmdSelect = new NpgsqlCommand();
            cmdSelect.CommandText = "esq.sp_sel_persona_dir(@idper)";
            cmdSelect.CommandType = CommandType.StoredProcedure;
            cmdSelect.Connection = cnRentas;

            NpgsqlParameter p1 = new NpgsqlParameter("@idper",

NpgsqlDbType.Integer);

            cmdSelect.Parameters.Add(p1);

            cmdSelect.Parameters[0].Value = idper;

            cnRentas.Open();
            return cmdSelect.ExecuteScalar().ToString();

        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
        finally
        {
            cnRentas.Close();
            cnRentas.Dispose();
        }
    }

    public int Insert()
    {
        NpgsqlConnection cnRentas = new NpgsqlConnection();

        try
        {
            cnRentas.ConnectionString = Conexion.CadenaConexion;

            NpgsqlCommand cmdInsert = new NpgsqlCommand();
            cmdInsert.CommandText = " esq.sp_ins_persona(@ced, @nom,

@ape, @dir)";

            cmdInsert.CommandType = CommandType.StoredProcedure;
            cmdInsert.Connection = cnRentas;

            NpgsqlParameter p1 = new NpgsqlParameter("@ced",

NpgsqlDbType.Char, 10);
            NpgsqlParameter p2 = new NpgsqlParameter("@nom",

NpgsqlDbType.Varchar, 35);
            NpgsqlParameter p3 = new NpgsqlParameter("@ape",

NpgsqlDbType.Varchar, 35);
            NpgsqlParameter p4 = new NpgsqlParameter("@dir",

NpgsqlDbType.Varchar, 40);

            cmdInsert.Parameters.Add(p1);
            cmdInsert.Parameters.Add(p2);
            cmdInsert.Parameters.Add(p3);
            cmdInsert.Parameters.Add(p4);
        }
    }
}

```

```

        cmdInsert.Parameters[0].Value = this.cedula;
        cmdInsert.Parameters[1].Value = this.nombres;
        cmdInsert.Parameters[2].Value = this.apellidos;
        cmdInsert.Parameters[3].Value = this.direccion;

        cnRentas.Open();

        return Convert.ToInt32(cmdInsert.ExecuteScalar());
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Close();
        cnRentas.Dispose();
    }
}

public int Insert(string cedula, string nombres, string apellidos, string direccion)
{
    try
    {
        this.cedula = cedula;
        this.nombres = nombres;
        this.apellidos = apellidos;
        this.direccion = direccion;

        return this.Insert();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public void Update(int idper, string cedula, string nombres, string apellidos, string
direccion)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();

    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdInsert = new NpgsqlCommand();
        cmdInsert.CommandText = "esq.sp_upd_persona(@idper, @ced,
@nom, @ape, @dir)";

        cmdInsert.CommandType = CommandType.StoredProcedure;
        cmdInsert.Connection = cnRentas;

```

```

NpgsqlDbType.Integer);
NpgsqlDbType.Varchar, 10);
NpgsqlDbType.Varchar, 35);
NpgsqlDbType.Varchar, 35);
NpgsqlDbType.Varchar, 40);

NpgsqlParameter p1 = new NpgsqlParameter("@idper",
NpgsqlParameter p2 = new NpgsqlParameter("@ced",
NpgsqlParameter p3 = new NpgsqlParameter("@nom",
NpgsqlParameter p4 = new NpgsqlParameter("@ape",
NpgsqlParameter p5 = new NpgsqlParameter("@dir",

cmdInsert.Parameters.Add(p1);
cmdInsert.Parameters.Add(p2);
cmdInsert.Parameters.Add(p3);
cmdInsert.Parameters.Add(p4);
cmdInsert.Parameters.Add(p5);

cmdInsert.Parameters[0].Value = idper;
cmdInsert.Parameters[1].Value = cedula;
cmdInsert.Parameters[2].Value = nombres;
cmdInsert.Parameters[3].Value = apellidos;
cmdInsert.Parameters[4].Value = direccion;

cnRentas.Open();

cmdInsert.ExecuteNonQuery();

}
catch(NpgsqlException errorBD)
{
    throw errorBD;
}
catch(Exception error)
{
    throw error;
}
finally
{
    cnRentas.Close();
    cnRentas.Dispose();
}
}

public void Delete(int idper)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    DataTable dtPersonas = new DataTable();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;
        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_del_persona(@idper)";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@idper",

cmdSelect.Parameters.Add(p1);

cmdSelect.Parameters[0].Value = idper;

```

```

        cnRentas.Open();
        cmdSelect.ExecuteNonQuery();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Close();
        cnRentas.Dispose();
    }
}
#endregion
}
}

```

Clase Alcabala, maneja las alcabalas:

```

using System;
using Npgsql;
using NpgsqlTypes;
using System.Data;
using System.Collections.Generic;

namespace DataAccessLayer
{
    public class Alcabala
    {
        public Alcabala()
        {
        }

        public Alcabala(long n_alcabala, string escritura, string objeto, string ubicacion, decimal
valor)
        {
            this.n_alcabala = n_alcabala;
            this.escritura = escritura;
            this.objeto_venta = objeto;
            this.ubicacion = ubicacion;
            this.valor = valor;
        }

        #region Datos

        private long n_alcabala;
        private DateTime f_ing;

```



```

private string escritura;
private string objeto_venta;
private string ubicacion;
private decimal valor;
private bool liquidado;
private DateTime f_liq;
private decimal municipio;
private decimal escolares ;
private string observacion;
private bool pagado;
private DateTime f_pag ;

#endregion

#region Propiedades

public long N_alcabala
{
    set
    {
        this.n_alcabala = value;
    }
    get
    {
        return this.n_alcabala;
    }
}

public DateTime F_ing
{
    set
    {
        this.f_ing = value;
    }
    get
    {
        return this.f_ing;
    }
}

public string Escritura
{
    set
    {
        this.escritura = value;
    }
    get
    {
        return this.escritura;
    }
}

public string Objeto_venta
{
    set
    {
        this.objeto_venta = value;
    }
    get
    {
        return this.objeto_venta;
    }
}

```

```

public string Ubicacion
{
    set
    {
        this.ubicacion = value;
    }
    get
    {
        return this.ubicacion;
    }
}

```

```

public decimal Valor
{
    set
    {
        this.valor = value;
    }
    get
    {
        return this.valor;
    }
}

```

```

public bool Liquidado
{
    set
    {
        this.liquidado = value;
    }
    get
    {
        return this.liquidado;
    }
}

```

```

public DateTime F_liq
{
    set
    {
        this.f_liq = value;
    }
    get
    {
        return this.f_liq;
    }
}

```

```

public decimal Municipio
{
    set
    {
        this.municipio = value;
    }
    get
    {
        return this.municipio;
    }
}

```

```

public decimal Escolares
{
    set

```

```

        {
            this.escolares = value;
        }
        get
        {
            return this.escolares;
        }
    }

    public string Observacion
    {
        set
        {
            this.observacion = value;
        }
        get
        {
            return this.observacion;
        }
    }

    public bool Pagado
    {
        set
        {
            this.pagado = value;
        }
        get
        {
            return this.pagado;
        }
    }

    public DateTime F_pag
    {
        set
        {
            this.f_pag = value;
        }
        get
        {
            return this.f_pag;
        }
    }
}

#endregion

#region Métodos

public DataTable SelectALiquidar(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_alcabala_a_liq(@nalca)";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;
    }
    catch
    {
    }
}

```

```

NpgsqlDbType.Bigint);

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",

cmdSelect.Parameters.Add(p1);

cmdSelect.Parameters[0].Value = n_alca;

        DataTable dt = new DataTable();
        NpgsqlDataAdapter da = new NpgsqlDataAdapter();
        da.SelectCommand = cmdSelect;
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Dispose();
    }
}

public DataTable SelectAPagar(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_alcabala_a_pag(@nalca)";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",

NpgsqlDbType.Bigint);

cmdSelect.Parameters.Add(p1);

cmdSelect.Parameters[0].Value = n_alca;

        DataTable dt = new DataTable();
        NpgsqlDataAdapter da = new NpgsqlDataAdapter();
        da.SelectCommand = cmdSelect;
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {

```

```

        cnRentas.Dispose();
    }
}

public DataTable SelectAnularLiq(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_alcabala_anu_liq(@nalca)";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

        cmdSelect.Parameters.Add(p1);

        cmdSelect.Parameters[0].Value = n_alca;

        DataTable dt = new DataTable();
        NpgsqlDataAdapter da = new NpgsqlDataAdapter();
        da.SelectCommand = cmdSelect;
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Dispose();
    }
}

public DataTable SelectAnularPag(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_alcabala_anu_pag(@nalca)";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

        cmdSelect.Parameters.Add(p1);

        cmdSelect.Parameters[0].Value = n_alca;

```

```

        DataTable dt = new DataTable();
        NpgsqlDataAdapter da = new NpgsqlDataAdapter();
        da.SelectCommand = cmdSelect;
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Dispose();
    }
}

public long SelectSecuencia()
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_alcabala_seq()";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        cnRentas.Open();

        return Convert.ToInt64(cmdSelect.ExecuteScalar());
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Close();
        cnRentas.Dispose();
    }
}

public DataTable SelectIngresadas(DateTime fini, DateTime ffin)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();

    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();

```

```

cmdSelect.CommandText = "esq.sp_sel_alcabala_ings (@fini,
@ffin)";

cmdSelect.CommandType = CommandType.StoredProcedure;
cmdSelect.Connection = cnRentas;

NpgsqlDbType.Date);
NpgsqlDbType.Date);

NpgsqlParameter p1 = new NpgsqlParameter("@fini",
NpgsqlParameter p2 = new NpgsqlParameter("@ffin",

cmdSelect.Parameters.Add(p1);
cmdSelect.Parameters.Add(p2);

cmdSelect.Parameters[0].Value = fini;
cmdSelect.Parameters[1].Value = ffin;

NpgsqlDataAdapter da = new NpgsqlDataAdapter();
da.SelectCommand = cmdSelect;

DataTable dt = new DataTable();
da.Fill(dt);

return dt;
}
catch(NpgsqlException errorBD)
{
throw errorBD;
}
catch(Exception error)
{
throw error;
}
}

```

```

public DataTable SelectLiquidadas(DateTime fini, DateTime ffin)
{
NpgsqlConnection cnRentas = new NpgsqlConnection();

try
{
cnRentas.ConnectionString = Conexion.CadenaConexion;

NpgsqlCommand cmdSelect = new NpgsqlCommand();
cmdSelect.CommandText = "esq.sp_sel_alcabala_liqs (@fini,
@ffin)";

cmdSelect.CommandType = CommandType.StoredProcedure;
cmdSelect.Connection = cnRentas;

NpgsqlParameter p1 = new NpgsqlParameter("@fini",
NpgsqlParameter p2 = new NpgsqlParameter("@ffin",

cmdSelect.Parameters.Add(p1);
cmdSelect.Parameters.Add(p2);

cmdSelect.Parameters[0].Value = fini;
cmdSelect.Parameters[1].Value = ffin;

NpgsqlDataAdapter da = new NpgsqlDataAdapter();
da.SelectCommand = cmdSelect;

```

```

        DataTable dt = new DataTable();
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectPagadas(DateTime fini, DateTime ffin)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();

    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_alcabala_pags (@fini,
@ffin)";

        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@fini",
NpgsqlDbType.Date);

        NpgsqlParameter p2 = new NpgsqlParameter("@ffin",
NpgsqlDbType.Date);

        cmdSelect.Parameters.Add(p1);
        cmdSelect.Parameters.Add(p2);

        cmdSelect.Parameters[0].Value = fini;
        cmdSelect.Parameters[1].Value = ffin;

        NpgsqlDataAdapter da = new NpgsqlDataAdapter();
        da.SelectCommand = cmdSelect;

        DataTable dt = new DataTable();
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectPagadasPorVentanilla(DateTime fini, DateTime ffin, int cobrado)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();

    try

```



```

        {
            cnRentas.ConnectionString = Conexion.CadenaConexion;

            NpgsqlCommand cmdSelect = new NpgsqlCommand();
            cmdSelect.CommandText = "esq.sp_sel_alcabala_pags_ventanilla
(@fini, @ffin, @cobrado)";

            cmdSelect.CommandType = CommandType.StoredProcedure;
            cmdSelect.Connection = cnRentas;

            NpgsqlParameter p1 = new NpgsqlParameter("@fini",
NpgsqlDbType.Date);
            NpgsqlParameter p2 = new NpgsqlParameter("@ffin",
NpgsqlDbType.Date);
            NpgsqlParameter p3 = new NpgsqlParameter("@cobrado",
NpgsqlDbType.Integer);

            cmdSelect.Parameters.Add(p1);
            cmdSelect.Parameters.Add(p2);
            cmdSelect.Parameters.Add(p3);

            cmdSelect.Parameters[0].Value = fin;
            cmdSelect.Parameters[1].Value = ffin;
            cmdSelect.Parameters[2].Value = cobrado;

            NpgsqlDataAdapter da = new NpgsqlDataAdapter();
            da.SelectCommand = cmdSelect;

            DataTable dt = new DataTable();
            da.Fill(dt);

            return dt;
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }

    public void Insert(long n_alcabala, string escritura, string objeto, string ubicacion,
decimal valor, List<Vendedores> vends, List<Compradores> comps)
    {
        NpgsqlConnection cnRentas = new NpgsqlConnection();
        cnRentas.ConnectionString = Conexion.CadenaConexion;
        cnRentas.Open();
        NpgsqlTransaction trans = cnRentas.BeginTransaction();

        try
        {
            NpgsqlCommand cmdInsert = new NpgsqlCommand();
            cmdInsert.CommandText = "esq.sp_ins_alcabala(@nalca, @escr ,
@objeto, @ubic , @val) ";

            cmdInsert.CommandType = CommandType.StoredProcedure;
            cmdInsert.Connection = cnRentas;
            cmdInsert.Transaction = trans;

            NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);
            NpgsqlParameter p2 = new NpgsqlParameter("@escr",
NpgsqlDbType.Varchar, 60);

```

```

NpgsqlDbType.Text);
NpgsqlDbType.VarChar, 40);
NpgsqlDbType.Numeric);

NpgsqlParameter p3 = new NpgsqlParameter("@objeto",
NpgsqlParameter p4 = new NpgsqlParameter("@ubic",
NpgsqlParameter p5 = new NpgsqlParameter("@val",

cmdInsert.Parameters.Add(p1);
cmdInsert.Parameters.Add(p2);
cmdInsert.Parameters.Add(p3);
cmdInsert.Parameters.Add(p4);
cmdInsert.Parameters.Add(p5);

cmdInsert.Parameters[0].Value = n_alcabala;
cmdInsert.Parameters[1].Value = escritura;
cmdInsert.Parameters[2].Value = objeto;
cmdInsert.Parameters[3].Value = ubicacion;
cmdInsert.Parameters[4].Value = valor;

cmdInsert.ExecuteNonQuery();
//*****CONTRATANTES

List<NpgsqlCommand>();

List<NpgsqlCommand> comandVende = new
for(int i = 0; i < vends.Count; i++)
{

NpgsqlCommand cmdInsertVend = new
cmdInsertVend.CommandText =
cmdInsertVend.CommandType =
cmdInsertVend.Connection = cnRentas;

NpgsqlParameter pr1 = new NpgsqlParameter("@nalca",
NpgsqlParameter pr2 = new NpgsqlParameter("@id_ven",

cmdInsertVend.Parameters.Add(pr1);
cmdInsertVend.Parameters.Add(pr2);

cmdInsertVend.Parameters[0].Value = vends[i].N_alcabala;
cmdInsertVend.Parameters[1].Value =

vends[i].Id_vendedor;

cmdInsertVend.Transaction = trans;
comandVende.Add(cmdInsertVend);
cmdInsertVend.Dispose();
}

List<NpgsqlCommand> comandCompra = new
for(int j = 0; j < comps.Count; j++)
{
NpgsqlCommand cmdInsertComp = new
cmdInsertComp.CommandText =
cmdInsertComp.CommandType =
CommandType.StoredProcedure;

```

```

        cmdInsertComp.Connection = cnRentas;

        NpgsqlParameter pr1 = new NpgsqlParameter("@nalca",
        NpgsqlDbType.Bigint);
        NpgsqlParameter pr2 = new NpgsqlParameter("@id_com",
        NpgsqlDbType.Integer);

        cmdInsertComp.Parameters.Add(pr1);
        cmdInsertComp.Parameters.Add(pr2);

        cmdInsertComp.Parameters[0].Value =
        comps[j].N_alcabala;
        cmdInsertComp.Parameters[1].Value =
        comps[j].Id_comprador;

        cmdInsertComp.Transaction = trans;
        comandCompra.Add(cmdInsertComp);
        cmdInsertComp.Dispose();
    }

    for(int k = 0; k < comandVende.Count; k++)
        comandVende[k].ExecuteNonQuery();

    for(int l = 0; l < comandCompra.Count; l++)
        comandCompra[l].ExecuteNonQuery();

    trans.Commit();
}
catch(NpgsqlException errorBD)
{
    trans.Rollback();
    throw errorBD;
}
catch(Exception error)
{
    trans.Rollback();
    throw error;
}
finally
{
    cnRentas.Close();
    cnRentas.Dispose();
}
}

public void UpdateLiquidar(long n_alca, decimal muni, decimal escol, string observ)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdUpdate = new NpgsqlCommand();
        cmdUpdate.CommandText = "esq.sp_upd_alcabala_liq(@nalca,
        @muni, @esc, @obs)";

        cmdUpdate.CommandType = CommandType.StoredProcedure;
        cmdUpdate.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
        NpgsqlDbType.Bigint);
        NpgsqlParameter p2 = new NpgsqlParameter("@muni",
        NpgsqlDbType.Numeric);
        NpgsqlParameter p3 = new NpgsqlParameter("@esc",
        NpgsqlDbType.Numeric);

```

```

NpgsqlDbType.Text);

        NpgsqlParameter p4 = new NpgsqlParameter("@obs",

        cmdUpdate.Parameters.Add(p1);
        cmdUpdate.Parameters.Add(p2);
        cmdUpdate.Parameters.Add(p3);
        cmdUpdate.Parameters.Add(p4);

        cmdUpdate.Parameters[0].Value = n_alca;
        cmdUpdate.Parameters[1].Value = muni;
        cmdUpdate.Parameters[2].Value = escol;
        cmdUpdate.Parameters[3].Value = observ;

        cnRentas.Open();

        cmdUpdate.ExecuteNonQuery();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Close();
        cnRentas.Dispose();
    }
}

public void UpdatePagar(long n_alca, int cobrado)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdUpdate = new NpgsqlCommand();
        cmdUpdate.CommandText = "esq.sp_upd_alcabala_pag(@nalca,
@cobrado)";

        cmdUpdate.CommandType = CommandType.StoredProcedure;
        cmdUpdate.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);
        NpgsqlParameter p2 = new NpgsqlParameter("@cobrado",
NpgsqlDbType.Integer);

        cmdUpdate.Parameters.Add(p1);
        cmdUpdate.Parameters.Add(p2);

        cmdUpdate.Parameters[0].Value = n_alca;
        cmdUpdate.Parameters[1].Value = cobrado;

        cnRentas.Open();

        cmdUpdate.ExecuteNonQuery();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
}

```

```

        catch(Exception error)
        {
            throw error;
        }
        finally
        {
            cnRentas.Close();
            cnRentas.Dispose();
        }
    }

    public void Update(long n_alcabala, string escritura, string objeto, string ubicacion,
decimal valor, List<Vendedores> vends, List<Compradores> comps)
    {
        NpgsqlConnection cnRentas = new NpgsqlConnection();
        cnRentas.ConnectionString = Conexion.CadenaConexion;
        cnRentas.Open();
        NpgsqlTransaction trans = cnRentas.BeginTransaction();

        try
        {
            NpgsqlCommand cmdUpdate = new NpgsqlCommand();
            cmdUpdate.CommandText = "esq.sp_upd_alcabala(@escr , @objeto,
@ubic , @val, @nalca)";

            cmdUpdate.CommandType = CommandType.StoredProcedure;
            cmdUpdate.Connection = cnRentas;
            cmdUpdate.Transaction = trans;

            NpgsqlParameter p1 = new NpgsqlParameter("@escr",
NpgsqlDbType.VarChar, 60);
            NpgsqlParameter p2 = new NpgsqlParameter("@objeto",
NpgsqlDbType.Text);
            NpgsqlParameter p3 = new NpgsqlParameter("@ubic",
NpgsqlDbType.VarChar, 40);
            NpgsqlParameter p4 = new NpgsqlParameter("@val",
NpgsqlDbType.Numeric);
            NpgsqlParameter p5 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

            cmdUpdate.Parameters.Add(p1);
            cmdUpdate.Parameters.Add(p2);
            cmdUpdate.Parameters.Add(p3);
            cmdUpdate.Parameters.Add(p4);
            cmdUpdate.Parameters.Add(p5);

            cmdUpdate.Parameters[0].Value = escritura;
            cmdUpdate.Parameters[1].Value = objeto;
            cmdUpdate.Parameters[2].Value = ubicacion;
            cmdUpdate.Parameters[3].Value = valor;
            cmdUpdate.Parameters[4].Value = n_alcabala;

            cmdUpdate.ExecuteNonQuery();
            //*****CONTRATANTES

            NpgsqlCommand cmdDeleteVend = new NpgsqlCommand();
            cmdDeleteVend.CommandText = "esq.sp_del_vendedores(@nalca)";
            cmdDeleteVend.CommandType = CommandType.StoredProcedure;
            cmdDeleteVend.Connection = cnRentas;
            cmdDeleteVend.Transaction = trans;

            NpgsqlParameter par1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

```

```

cmdDeleteVend.Parameters.Add(par1);

cmdDeleteVend.Parameters[0].Value = n_alcabala;
cmdDeleteVend.ExecuteNonQuery();

//
NpgsqlCommand cmdDeleteComp = new NpgsqlCommand();
cmdDeleteComp.CommandText =
"esq.sp_del_compradores(@nalca)";
cmdDeleteComp.CommandType = CommandType.StoredProcedure;
cmdDeleteComp.Connection = cnRentas;
cmdDeleteComp.Transaction = trans;

NpgsqlParameter para1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

cmdDeleteComp.Parameters.Add(para1);

cmdDeleteComp.Parameters[0].Value = n_alcabala;
cmdDeleteComp.ExecuteNonQuery();
/***/

List<NpgsqlCommand> comandVende = new
for(int i = 0; i < vends.Count; i++)
{
NpgsqlCommand cmdInsertVend = new
cmdInsertVend.CommandText =
"esq.sp_ins_vendedores(@nalca, @id_ven) ";
cmdInsertVend.CommandType =
CommandType.StoredProcedure;
cmdInsertVend.Connection = cnRentas;

NpgsqlParameter pr1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);
NpgsqlParameter pr2 = new NpgsqlParameter("@id_ven",
NpgsqlDbType.Integer);

cmdInsertVend.Parameters.Add(pr1);
cmdInsertVend.Parameters.Add(pr2);

cmdInsertVend.Parameters[0].Value = vends[i].N_alcabala;
cmdInsertVend.Parameters[1].Value =
vends[i].Id_vendedor;

cmdInsertVend.Transaction = trans;
comandVende.Add(cmdInsertVend);
cmdInsertVend.Dispose();
}

List<NpgsqlCommand> comandCompra = new
for(int j = 0; j < comps.Count; j++)
{
NpgsqlCommand cmdInsertComp = new
cmdInsertComp.CommandText =
"esq.sp_ins_compradores(@nalca, @id_com) ";
cmdInsertComp.CommandType =
CommandType.StoredProcedure;

```

```

        cmdInsertComp.Connection = cnRentas;

        NpgsqlParameter pr1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);
        NpgsqlParameter pr2 = new NpgsqlParameter("@id_com",
NpgsqlDbType.Integer);

        cmdInsertComp.Parameters.Add(pr1);
        cmdInsertComp.Parameters.Add(pr2);

        cmdInsertComp.Parameters[0].Value =
comps[j].N_alcabala;
        cmdInsertComp.Parameters[1].Value =
comps[j].Id_comprador;

        cmdInsertComp.Transaction = trans;
        comandCompra.Add(cmdInsertComp);
        cmdInsertComp.Dispose();
    }

    for(int k = 0; k < comandVende.Count; k++)
        comandVende[k].ExecuteNonQuery();

    for(int l = 0; l < comandCompra.Count; l++)
        comandCompra[l].ExecuteNonQuery();

    trans.Commit();
}
catch(NpgsqlException errorBD)
{
    trans.Rollback();
    throw errorBD;
}
catch(Exception error)
{
    trans.Rollback();
    throw error;
}
finally
{
    cnRentas.Close();
    cnRentas.Dispose();
}
}

public void UpdateAnularLiquidacion(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdUpdate = new NpgsqlCommand();
        cmdUpdate.CommandText =
"esq.sp_upd_alcabala_anu_liq(@nalca)";
        cmdUpdate.CommandType = CommandType.StoredProcedure;
        cmdUpdate.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

        cmdUpdate.Parameters.Add(p1);

```

```

        cmdUpdate.Parameters[0].Value = n_alca;

        cnRentas.Open();

        cmdUpdate.ExecuteNonQuery();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Close();
        cnRentas.Dispose();
    }
}

public void UpdateAnularPago(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdUpdate = new NpgsqlCommand();
        cmdUpdate.CommandText =
"esq.sp_upd_alcabala_anu_pag(@nalca)";
        cmdUpdate.CommandType = CommandType.StoredProcedure;
        cmdUpdate.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

        cmdUpdate.Parameters.Add(p1);

        cmdUpdate.Parameters[0].Value = n_alca;

        cnRentas.Open();

        cmdUpdate.ExecuteNonQuery();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Close();
        cnRentas.Dispose();
    }
}

public void Delete(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();

```



```

try
{
    cnRentas.ConnectionString = Conexion.CadenaConexion;

    NpgsqlCommand cmdDelete = new NpgsqlCommand();
    cmdDelete.CommandText = "esq.sp_del_alcabala(@nalca)";
    cmdDelete.CommandType = CommandType.StoredProcedure;
    cmdDelete.Connection = cnRentas;

    NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

    cmdDelete.Parameters.Add(p1);

    cmdDelete.Parameters[0].Value = n_alca;

    cnRentas.Open();

    cmdDelete.ExecuteNonQuery();
}
catch(NpgsqlException errorBD)
{
    throw errorBD;
}
catch(Exception error)
{
    throw error;
}
finally
{
    cnRentas.Close();
    cnRentas.Dispose();
}
}

```

```

public void Secuencia(int i)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmd = new NpgsqlCommand();
        cmd.CommandText = "esq.sp_alcabala_seq(@i)";
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@i",
NpgsqlDbType.Integer);

        cmd.Parameters.Add(p1);

        cmd.Parameters[0].Value = i;

        cnRentas.Open();

        cmd.ExecuteNonQuery();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
}

```

```

        }
        catch(Exception error)
        {
            throw error;
        }
        finally
        {
            cnRentas.Close();
            cnRentas.Dispose();
        }
    }

#endregion

}
}
}

```

Clase Vendedores, maneja vendedores en trámites de alcabalas:

```

using System;
using Npgsql;
using System.Data;
using NpgsqlTypes;

namespace DataAccessLayer
{
    public class Vendedores
    {
        public Vendedores()
        {
        }

        #region Datos

        private long id_vends;
        private long n_alcabala;
        private int id_vendedor;

        #endregion

        #region Propiedades

        public long Id_vends
        {
            set
            {
                this.id_vends = value;
            }
            get
            {
                return this.id_vends;
            }
        }

        public long N_alcabala
        {
            set

```

```

        {
            this.n_alcabala = value;
        }
        get
        {
            return this.n_alcabala;
        }
    }

    public int Id_vendedor
    {
        set
        {
            this.id_vendedor = value;
        }
        get
        {
            return this.id_vendedor;
        }
    }
}

#endregion

#region Métodos

public DataTable Select(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_vendedores(@nalca)";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

        cmdSelect.Parameters.Add(p1);

        cmdSelect.Parameters[0].Value = n_alca;

        DataTable dt = new DataTable();
        NpgsqlDataAdapter da = new NpgsqlDataAdapter();
        da.SelectCommand = cmdSelect;
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Dispose();
    }
}

```

```

        #endregion
    }
}

```

Clase Compradores, maneja compradores en los trámites de alcabalas:

```

using System;
using Npgsql;
using NpgsqlTypes;
using System.Data;

namespace DataAccessLayer
{
    public class Compradores
    {
        public Compradores()
        {
        }

        #region Datos

        private long id_comps;
        private long n_alcabala;
        private int id_comprador;

        #endregion

        #region Propiedades

        public long Id_comps
        {
            set
            {
                this.id_comps = value;
            }
            get
            {
                return this.id_comps;
            }
        }

        public long N_alcabala
        {
            set
            {
                this.n_alcabala = value;
            }
            get
            {
                return this.n_alcabala;
            }
        }

        public int Id_comprador
        {
            set
            {
                this.id_comprador = value;
            }
        }
    }
}

```

```

        }
        get
        {
            return this.id_comprador;
        }
    }
#endregion

#region Métodos

public DataTable Select(long n_alca)
{
    NpgsqlConnection cnRentas = new NpgsqlConnection();
    try
    {
        cnRentas.ConnectionString = Conexion.CadenaConexion;

        NpgsqlCommand cmdSelect = new NpgsqlCommand();
        cmdSelect.CommandText = "esq.sp_sel_compradores(@nalca)";
        cmdSelect.CommandType = CommandType.StoredProcedure;
        cmdSelect.Connection = cnRentas;

        NpgsqlParameter p1 = new NpgsqlParameter("@nalca",
NpgsqlDbType.Bigint);

        cmdSelect.Parameters.Add(p1);

        cmdSelect.Parameters[0].Value = n_alca;

        DataTable dt = new DataTable();
        NpgsqlDataAdapter da = new NpgsqlDataAdapter();
        da.SelectCommand = cmdSelect;
        da.Fill(dt);

        return dt;
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
    finally
    {
        cnRentas.Dispose();
    }
}
#endregion
}

```

Algunas clases de la Capa de Regla de Negocio (Business Logic Layer)

Clase PersonaBLL con reglas de negocio para manejo de personas:

```

using System;
using Npgsql;
using NpgsqlTypes;
using DataAccessLayer;
using System.Data;

namespace BusinessLogicLayer
{
    public class PersonaBLL
    {
        public PersonaBLL()
        {
            this.per = new Persona();
        }

        public PersonaBLL(string cedula, string nombres, string apellidos, string direccion)
        {
            try
            {
                this.per = new Persona();

                //cedula
                if( String.IsNullOrEmpty(cedula) )
                    this.per.Cedula = null;
                else if( !BLL.SoloDigitos(cedula) )
                    throw new Exception("La cédula es incorrecta");
                else if(cedula.Length < 10 || cedula.Length > 10
                    dígitos");
                throw new Exception("La cédula debe tener exactamente 10

                else
                    this.per.Cedula = cedula;

                //nombres
                if( String.IsNullOrEmpty(nombres) )
                    throw new Exception("El nombre es requerido");
                else
                    this.per.Nombres = BLL.Cadena(nombres, 35);

                //apellidos
                if( String.IsNullOrEmpty(apellidos) )
                    this.per.Apellidos = null;
                else if( !BLL.SoloLetrasyEspacios(apellidos) )
                    throw new Exception("El apellido es incorrecto");
                else
                    this.per.Apellidos = BLL.Cadena(apellidos, 35);

                //dirección
                if( String.IsNullOrEmpty(direccion) )
                    throw new Exception("La dirección es requerida");
                else
                    this.per.Direccion = BLL.Cadena(direccion, 40);
            }
            catch(Exception error)
            {
                throw error;
            }
        }

        #region Datos

        private Persona per;
    }
}

```

```

#endregion

#region Propiedades

public string Cedula
{
    set
    {
        try
        {
            string cedula = value;
            if( String.IsNullOrEmpty(cedula) )
                this.per.Cedula = null;
            else if( !BLL.SoloDigitos(cedula) )
                throw new Exception("La cédula es incorrecta");
            else if(cedula.Length < 10 || cedula.Length > 10)
                throw new Exception("La cédula debe tener
exactamente 10 dígitos");
            else
                this.per.Cedula = cedula;
        }
        catch(Exception error)
        {
            throw error;
        }
    }
    get
    {
        return this.per.Cedula;
    }
}

public string Nombres
{
    set
    {
        try
        {
            string nombres = value;
            if( String.IsNullOrEmpty(nombres) )
                throw new Exception("El nombre es requerido");
            else
                this.per.Nombres = BLL.Cadena(nombres, 35);
        }
        catch(Exception error)
        {
            throw error;
        }
    }
    get
    {
        return this.per.Nombres;
    }
}

public string Apellidos
{
    set
    {
        try
        {
            string apellidos = value;

```

```

        if( String.IsNullOrEmpty(apellidos) )
            this.per.Apellidos = null;
        else if( !BLL.SoloLetrasyEspacios(apellidos) )
            throw new Exception("El apellido es incorrecto");
        else
            this.per.Apellidos = BLL.Cadena(apellidos, 35);
    }
    catch(Exception error)
    {
        throw error;
    }
}
get
{
    return this.per.Apellidos;
}
}

public string Direccion
{
    set
    {
        try
        {
            string direccion = value;
            if( String.IsNullOrEmpty(direccion) )
                throw new Exception("La dirección es requerida");
            else
                this.per.Direccion = BLL.Cadena(direccion, 40);
        }
        catch(Exception error)
        {
            throw error;
        }
    }
    get
    {
        return this.per.Direccion;
    }
}

}

#endregion

#region Métodos

public DataTable Select()
{
    try
    {
        return this.per.Select();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectTodos()

```



```

    {
        try
        {
            return this.per.SelectTodos();
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }

    public string SelectDireccion(int idper)
    {
        try
        {
            return this.per.SelectDireccion(idper);
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }

    public int Insert()
    {
        try
        {
            return this.per.Insert();
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }

    public int Insert(string cedula, string nombres, string apellidos, string direccion)
    {
        try
        {
            //cedula
            if( String.IsNullOrEmpty(cedula) )
                this.per.Cedula = null;
            else if( !BLL.SoloDigitos(cedula) )
                throw new Exception("La cédula es incorrecta");
            else if(cedula.Length < 10 || cedula.Length > 10)
                throw new Exception("La cédula debe tener exactamente 10
dígitos");
            else
                this.per.Cedula = cedula;
        }
    }

```

```

//nombres
if( String.IsNullOrEmpty(nombres) )
    throw new Exception("El nombre es requerido");
else
    this.per.Nombres = BLL.Cadena(nombres, 35);

//apellidos
if( String.IsNullOrEmpty(apellidos) )
    this.per.Apellidos = null;
else if( !BLL.SoloLetrasYEspacios(apellidos) )
    throw new Exception("El apellido es incorrecto");
else
    this.per.Apellidos = BLL.Cadena(apellidos, 35);

//dirección
if( String.IsNullOrEmpty(direccion) )
    throw new Exception("La dirección es requerida");
else
    this.per.Direccion = BLL.Cadena(direccion, 40);

return this.per.Insert();
}
catch(NpgsqlException errorBD)
{
    throw errorBD;
}
catch(Exception error)
{
    throw error;
}
}

public void Update(int idper, string cedula, string nombres, string apellidos, string
direccion)
{
    try
    {
        //cédula
        if( String.IsNullOrEmpty(cedula) )
            this.per.Cedula = null;
        else if( !BLL.SoloDigitos(cedula) )
            throw new Exception("La cédula es incorrecta");
        else if(cedula.Length < 10 || cedula.Length > 10)
            throw new Exception("La cédula debe tener exactamente 10
dígitos");
        else
            this.per.Cedula = cedula;

        //nombres
        if( String.IsNullOrEmpty(nombres) )
            throw new Exception("El nombre es requerido");
        else
            this.per.Nombres = BLL.Cadena(nombres, 35);

        //apellidos
        if( String.IsNullOrEmpty(apellidos) )
            this.per.Apellidos = null;
        else if( !BLL.SoloLetrasYEspacios(apellidos) )
            throw new Exception("El apellido es incorrecto");
        else
            this.per.Apellidos = BLL.Cadena(apellidos, 35);
    }
}

```

```

        //dirección
        if( String.IsNullOrEmpty(direccion) )
            throw new Exception("La dirección es requerida");
        else
            this.per.Direccion = BLL.Cadena(direccion, 40);

        this.per.Update(idper, cedula, nombres, apellidos, direccion);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public void Delete(int idper)
{
    try
    {
        this.per.Delete(idper);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

#endregion
}
}
}

```

Clase AlcabalaBLL con reglas de negocio para manejo de alcabalas:

```

using System;
using Npgsql;
using DataAccessLayer;
using System.Collections.Generic;
using System.Data;

namespace BusinessLogicLayer
{
    public class AlcabalaBLL
    {
        public AlcabalaBLL()
        {
            this.alca = new Alcabala();
        }

        #region Datos
    }
}

```

```

private Alcabala alca;

#endregion

#region Métodos

public DataTable SelectALiquidar(long n_alca)
{
    try
    {
        return this.alca.SelectALiquidar(n_alca);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectAPagar(long n_alca)
{
    try
    {
        return this.alca.SelectAPagar(n_alca);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectAnularLiq(long n_alca)
{
    try
    {
        return this.alca.SelectAnularLiq(n_alca);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectAnularPag(long n_alca)
{
    try
    {
        return this.alca.SelectAnularPag(n_alca);
    }
    catch(NpgsqlException errorBD)
    {

```

```

        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public long SelectSecuencia()
{
    try
    {
        return alca.SelectSecuencia();
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectIngresadas(DateTime fini, DateTime ffin)
{
    try
    {
        return this.alca.SelectIngresadas(fini, ffin);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectLiquidadas(DateTime fini, DateTime ffin)
{
    try
    {
        return this.alca.SelectLiquidadas(fini, ffin);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectPagadas(DateTime fini, DateTime ffin)
{
    try
    {
        return this.alca.SelectPagadas(fini, ffin);
    }
    catch(NpgsqlException errorBD)
    {

```

```

        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public DataTable SelectPagadasPorVentanilla(DateTime fini, DateTime ffin, int cobrado)
{
    try
    {
        return this.alca.SelectPagadasPorVentanilla(fini, ffin, cobrado);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public void Insert(long n_alcabala, string escritura, string objeto, string ubicacion,
decimal valor, List<VendedoresBLL> vends, List<CompradoresBLL> comps)
{
    try
    {
        List<Vendedores> vendedores = new List<Vendedores>();
        for(int i = 0; i < vends.Count; i++)
        {
            Vendedores ven = new Vendedores();
            ven.N_alcabala = vends[i].N_alcabala;
            ven.Id_vendedor = vends[i].Id_vendedor;
            vendedores.Add(ven);
        }
        List<Compradores> compradores = new List<Compradores>();
        for(int i = 0; i < comps.Count; i++)
        {
            Compradores com = new Compradores();
            com.N_alcabala = comps[i].N_alcabala;
            com.Id_comprador = comps[i].Id_comprador;
            compradores.Add(com);
        }

        //escritura
        if( String.IsNullOrEmpty(escritura) )
            throw new Exception("El campo \"Escritura\" es
requerido");

        else
            escritura = BLL.Cadena(escritura, 60);

        //objeto
        if( String.IsNullOrEmpty(objeto) )
            throw new Exception("Objeto de venta es requerido");

        //ubicación
        if( String.IsNullOrEmpty(ubicacion) )
            throw new Exception("La ubicación es requerida");

        else
            ubicacion = BLL.Cadena(ubicacion, 40);
    }
}

```

```

        //valor
        if(valor < 0 || valor > Convert.ToDecimal(99999999.99))
            throw new Exception("Valor no puede ser menor que $0" +
BLL.Separador() + "00, o mayor que $ 99999999" + BLL.Separador() + "99");

        this.alca.Insert(n_alcabala, escritura, objeto, ubicacion, valor,
vendedores, compradores);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public void Update(long n_alcabala, string escritura, string objeto, string ubicacion,
decimal valor, List<VendedoresBLL> vends, List<CompradoresBLL> comps)
{
    try
    {
        List<Vendedores> vendedores = new List<Vendedores>();
        for(int i = 0; i < vends.Count; i++)
        {
            Vendedores ven = new Vendedores();
            ven.N_alcabala = vends[i].N_alcabala;
            ven.Id_vendedor = vends[i].Id_vendedor;
            vendedores.Add(ven);
        }
        List<Compradores> compradores = new List<Compradores>();
        for(int i = 0; i < comps.Count; i++)
        {
            Compradores com = new Compradores();
            com.N_alcabala = comps[i].N_alcabala;
            com.Id_comprador = comps[i].Id_comprador;
            compradores.Add(com);
        }

        //escritura
        if( String.IsNullOrEmpty(escritura) )
            throw new Exception("El campo \"Escritura\" es
requerido");
        else
            escritura = BLL.Cadena(escritura, 60);

        //objeto
        if( String.IsNullOrEmpty(objeto) )
            throw new Exception("Objeto de venta es requerido");

        //ubicación
        if( String.IsNullOrEmpty(ubicacion) )
            throw new Exception("La ubicación es requerida");
        else
            ubicacion = BLL.Cadena(ubicacion, 40);

        //valor
        if(valor < 0 || valor > Convert.ToDecimal(99999999.99))
            throw new Exception("Valor no puede ser menor que $0" +
BLL.Separador() + "00, o mayor que $ 99999999" + BLL.Separador() + "99");
    }
}

```

```

        this.alca.Update(n_alcabala, escritura, objeto, ubicacion, valor,
vendedores, compradores);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public void UpdateLiquidar(long n_alca, decimal muni, decimal escol, string observ)
{
    try
    {
        //municipio
        if(muni < 0 || muni > Convert.ToDecimal(99999999.99))
            throw new Exception("Municipio no puede ser menor que
$0" + BLL.Separador() + "00, o mayor que $ 99999999" + BLL.Separador() + "99");

        //escolares
        if(escol < 0 || escol > Convert.ToDecimal(99999999.99))
            throw new Exception("Const. Escolares no puede ser menor
que $0" + BLL.Separador() + "00, o mayor que $ 99999999" + BLL.Separador() + "99");

        //observación
        if( String.IsNullOrEmpty(observ) )
            throw new Exception("La observación es requerida");

        alca.UpdateLiquidar(n_alca, muni, escol, observ);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public void UpdatePagar(long n_alca, int cobrado)
{
    try
    {
        alca.UpdatePagar(n_alca, cobrado);
    }
    catch(NpgsqlException errorBD)
    {
        throw errorBD;
    }
    catch(Exception error)
    {
        throw error;
    }
}

public void UpdateAnularLiquidacion(long n_alca)
{
    try

```



```

        {
            alca.UpdateAnularLiquidacion(n_alca);
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }

    public void UpdateAnularPago(long n_alca)
    {
        try
        {
            alca.UpdateAnularPago(n_alca);
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }

    public void Delete(long n_alca)
    {
        try
        {
            alca.Delete(n_alca);
        }
        catch(NpgsqlException errorBD)
        {
            throw errorBD;
        }
        catch(Exception error)
        {
            throw error;
        }
    }

    #endregion
}
}

```

ANEXO N° 3

MANUAL DE INSTALACIÓN

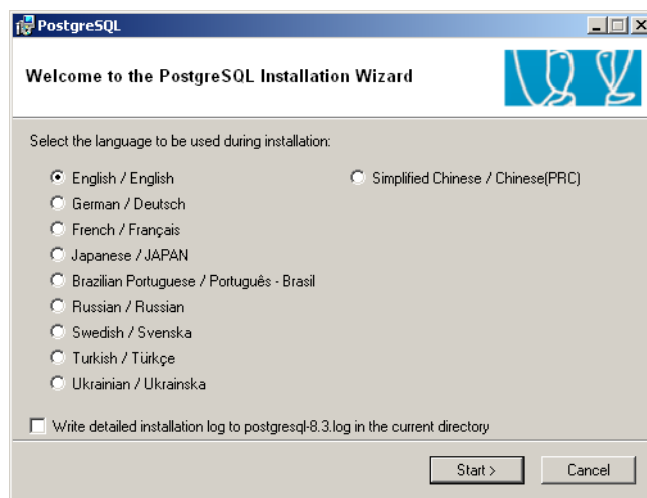
Software a instalar en el equipo Servidor

Instalación del motor de base de datos PostgreSQL 8.3

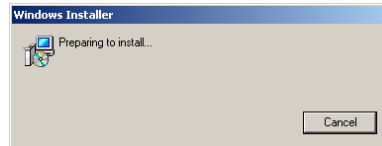
Para instalar el motor de base de datos PostgreSQL 8.3 se debe tener primero el instalador que es el archivo postgresql-8.3.0-1.zip. Este archivo hay que descomprimirlo y se obtiene los archivos siguientes:

- postgresql-8.3.msi
- postgresql-8.3-int.msi
- README.txt
- SETUP.bat
- UPGRADE.bat

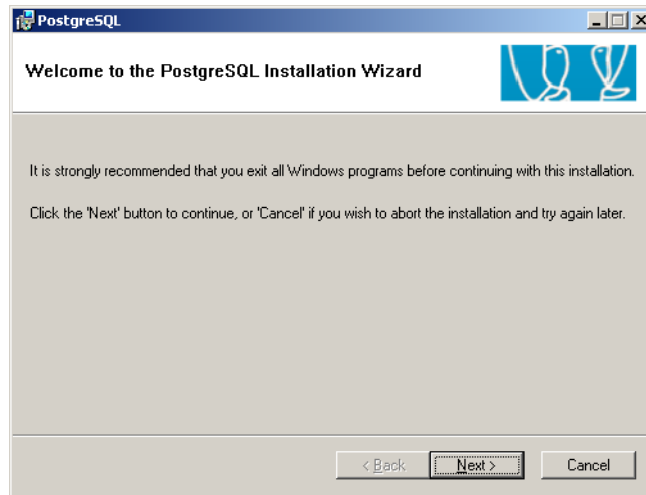
Ejecutar el archivo postgresql-8.3.msi y aparece el asistente de instalación. Escoger el idioma que se será utilizado durante la instalación, en este caso inglés (escoger la opción English). Clic en Start.



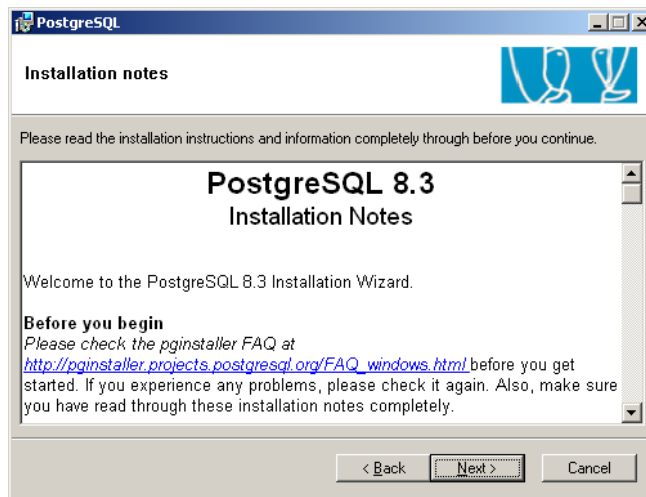
Se prepara la instalación:



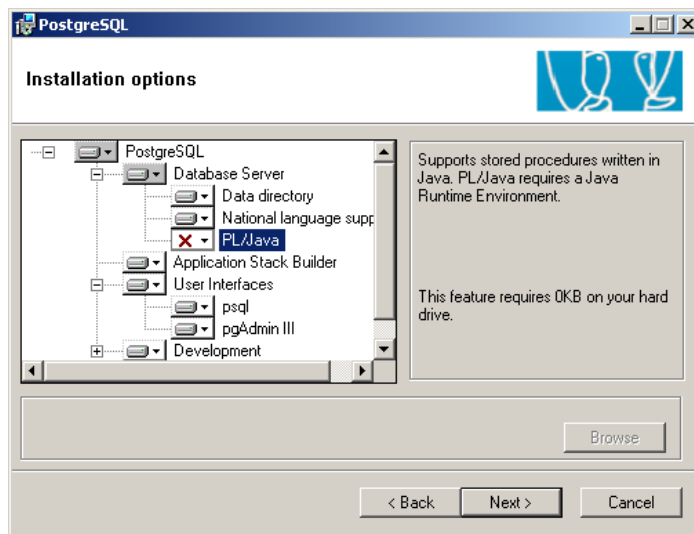
Aparece la pantalla de bienvenida. Next.



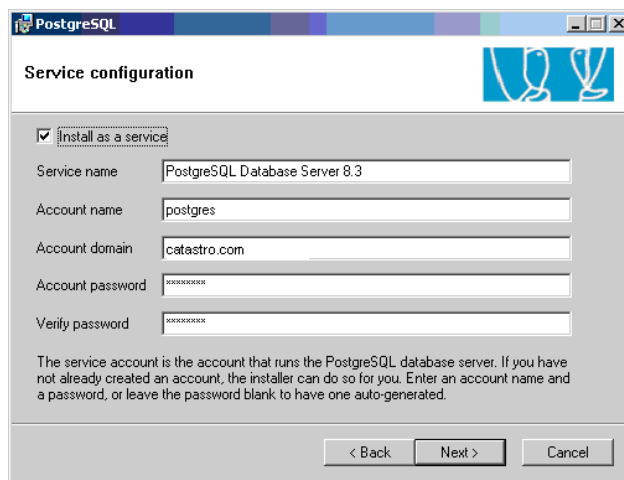
Aparecen las notas de instalación. Next.



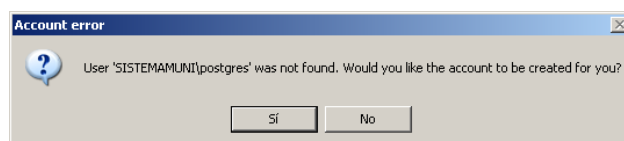
Luego se determina las opciones de instalación, escogiendo los componentes que se desean instalar. Next



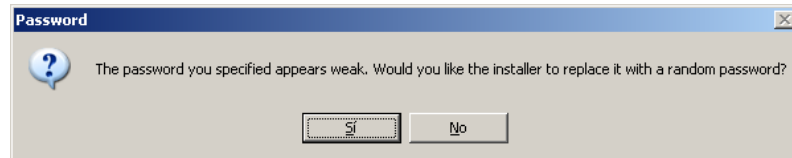
Se procede a configurar el servicio. Se debe verificar que la casilla “Install as a service” esté chequeada, escribir el nombre de la cuenta, el nombre del dominio que es “catastro.com” y una clave. Next.



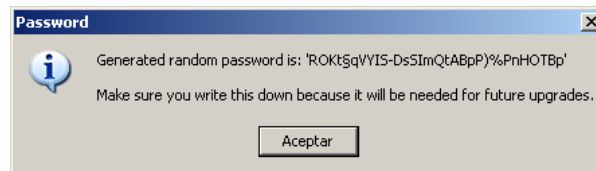
Aparece un mensaje advirtiendo que la cuenta de usuario no existe y si se desea crearlo, clic en Sí:



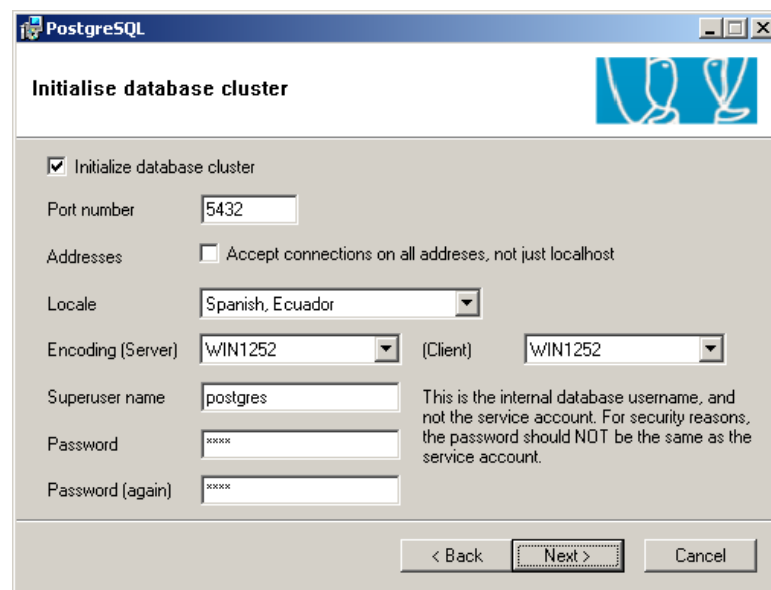
Otro mensaje advierte que la clave que se ha dado es poco segura y que si se desea que se genere una clave aleatoria. Se puede decir que no se genere, o en caso contrario Clic en Sí:



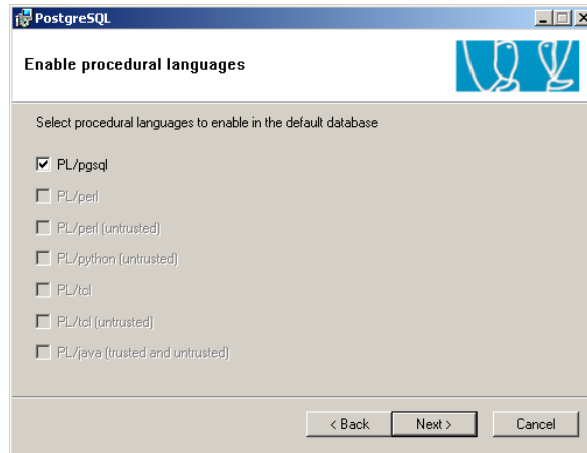
Un mensaje muestra la clave generada. Aceptar:



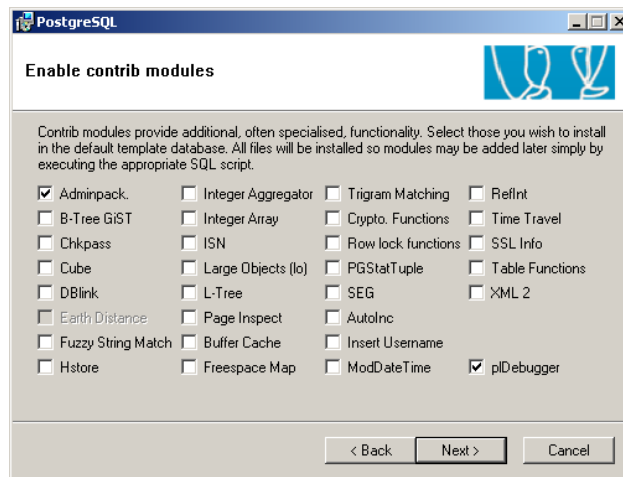
En la ventana que aparece verificar que la casilla “Initialize database cluster” esté chequeada, el puerto que usará la aplicación es el 5432. El nombre del super usuario es postgres, escribir una clave. Next.



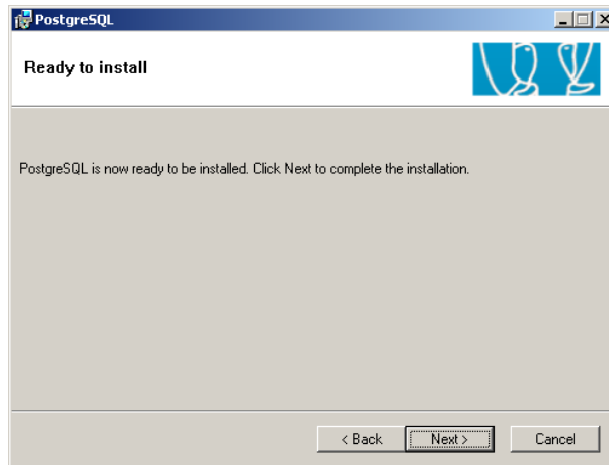
Se muestran los lenguajes procedimentales disponibles, chequear PL/pgsql. Next.



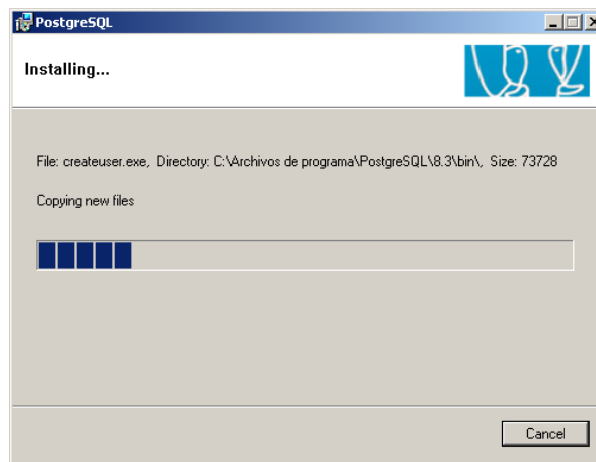
Aparecen los módulos adicionales que se pueden instalar. Next.



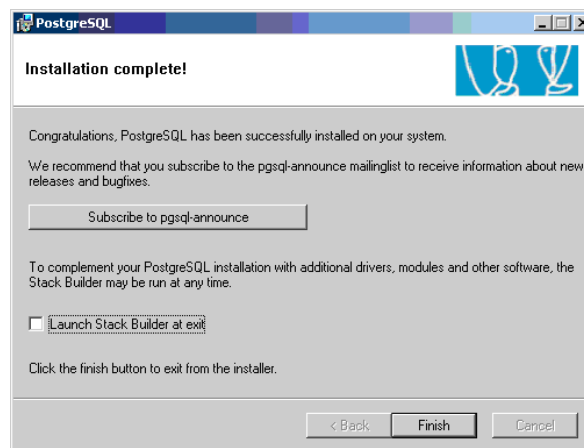
Para comenzar la instalación clic en Next.



Y comienza el proceso de instalación:



Finalmente aparece la siguiente ventana:



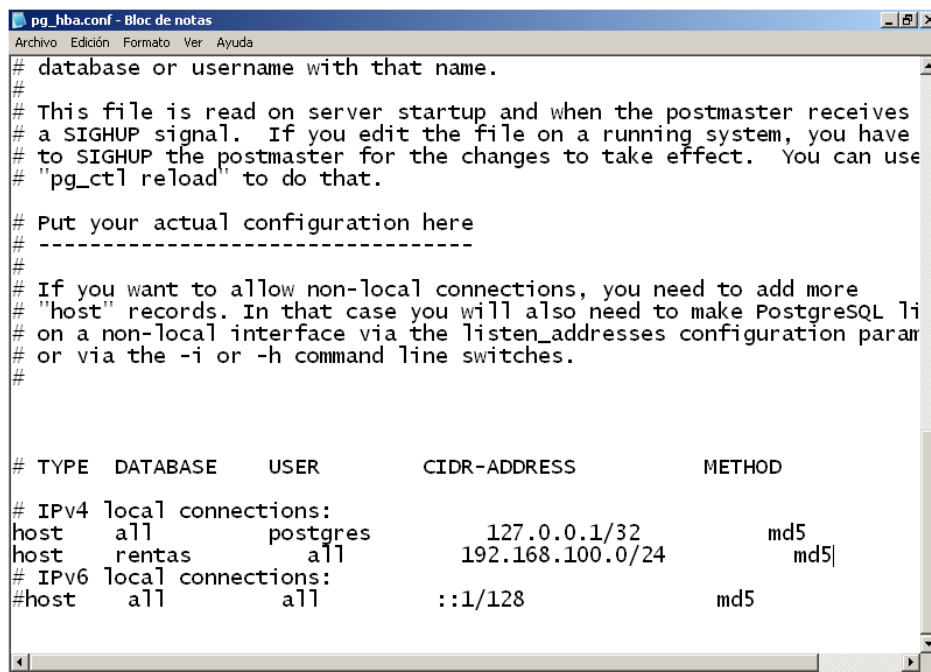
La instalación se ha realizado correctamente, clic en Finish. El software ha sido instalado.

Configuración de PostgreSQL 8.3

Para que los distintos usuarios de la base de datos y los equipos de la red puedan conectarse para hacer uso del motor de base de datos, se debe configurar los siguientes archivos:

- pg_hba.conf
- postgresql.conf

Para configurar el archivo pg_hba.conf ir a Inicio > Programas > PostgreSQL 8.3 > Configuration files > Edit pg_hba.conf. Aparece el editor de texto mostrando el contenido del archivo:



```

pg_hba.conf - Bloc de notas
Archivo Edición Formato Ver Ayuda
# database or username with that name.
#
# This file is read on server startup and when the postmaster receives
# a SIGHUP signal. If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect. You can use
# "pg_ctl reload" to do that.
#
# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL li
# on a non-local interface via the listen_addresses configuration param
# or via the -i or -h command line switches.
#
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# IPv4 local connections:
host all postgres 127.0.0.1/32 md5
host rentas all 192.168.100.0/24 md5
# IPv6 local connections:
#host all all ::1/128 md5

```

Al final del archivo, en la sección “Put your actual configuration here”, deben estar las siguientes líneas:

```

host all postgres 127.0.0.1/32 md5
host rentas all 192.168.100.0/24 md5

```

La primera línea indica que el súper usuario postgres se conecta a todas las bases de datos y puede también conectarse desde el equipo actual. La segunda línea indica que todos los demás usuarios del sistema pueden usar solamente a la base de datos “rentas” y conectarse desde los equipos que pertenezcan a la red 192.168.100.0/24. Una vez modificado el archivo se guardan los cambios.

Para configurar el archivo postgresql.conf ir a Inicio > Programas > PostgreSQL 8.3 > Configuration files > Edit postgresql.conf. Aparece el editor de texto mostrando el contenido del archivo:

```
# -----  
# CONNECTIONS AND AUTHENTICATION  
# -----  
# - Connection Settings -  
listen_addresses = '*' | # what IP address(es) to listen on;  
# comma-separated list of address  
# defaults to 'localhost', '*'  
# (change requires restart)  
port = 5432 # (change requires restart)  
max_connections = 100 # (change requires restart)  
# Note: Increasing max_connections costs ~400 bytes of shared memory per  
# connection slot, plus lock space (see max_locks_per_transaction). You  
# also need to raise shared_buffers to support more connections.  
#superuser_reserved_connections = 3 # (change requires restart)  
#unix_socket_directory = '' # (change requires restart)  
#unix_socket_group = '' # (change requires restart)  
#unix_socket_permissions = 0777 # begin with 0 to use octal notation  
# (change requires restart)  
#bonjour_name = '' # defaults to the computer name  
# (change requires restart)  
# - Security and Authentication -
```

En la sección “CONNECTIONS AND AUTHENTICATION” modificar de modo que esté la siguiente línea así:

```
listen_addresses = '*'
```

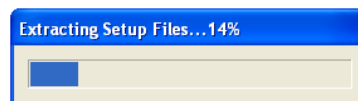
Verificar que la línea no esté como comentario (es decir, que esté sin el símbolo # al inicio). Esta línea indica que se escucharán a todas las direcciones IP de la red. Guardar los cambios.

Finalmente para actualizar los cambios ir a Inicio > Programas > PostgreSQL 8.3 > Reload configuration. Y está listo.

Software a instalar en los equipos Clientes

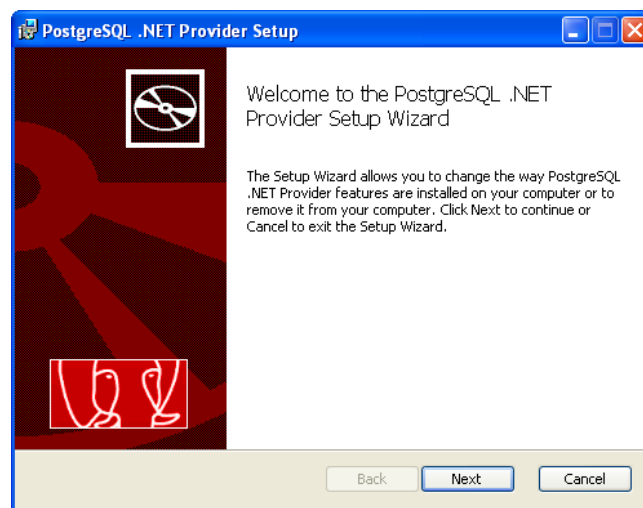
Instalación del .NET Framework 2.0

Para que la aplicación pueda funcionar se requiere que en el equipo esté instalado el .NET Framework 2.0. Para instalarlo se debe tener el archivo dotnet2.exe. Doble clic sobre el archivo y se comienza a instalar:

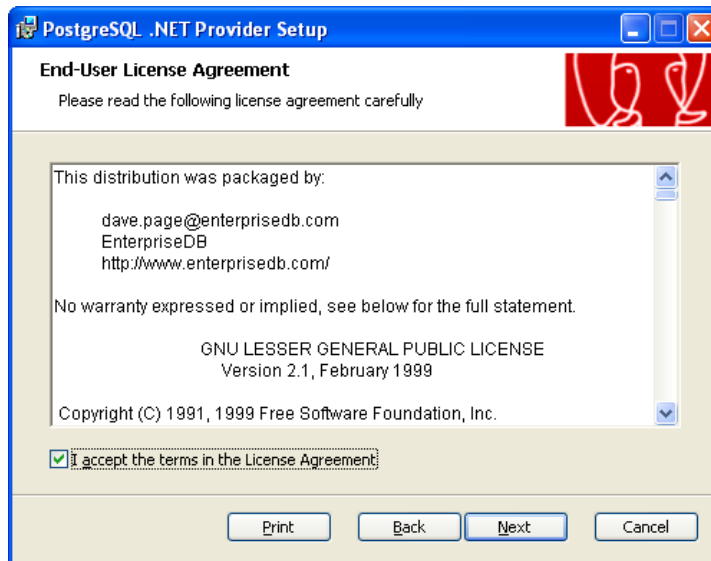


Instalación del proveedor .NET para PostgreSQL

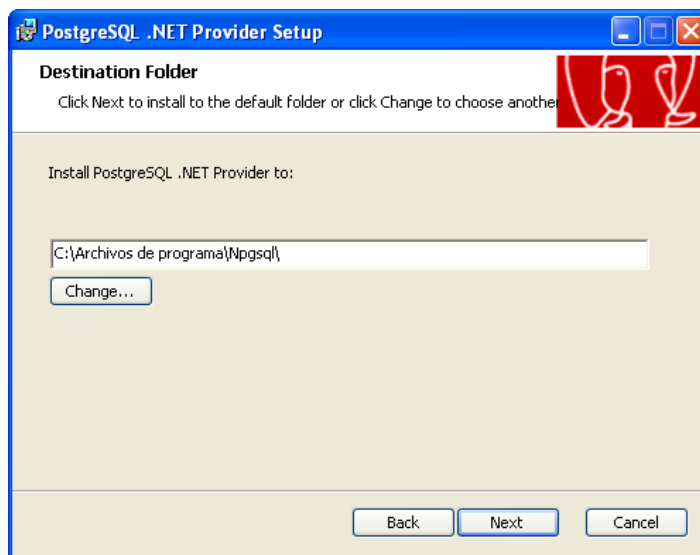
Se requiere además, para el funcionamiento del programa, que esté instalado en el equipo el proveedor .NET para PostgreSQL. Para instalarlo se debe tener el archivo npgsql.msi. Doble clic sobre el archivo. Aparece el asistente de instalación. Clic en Next:



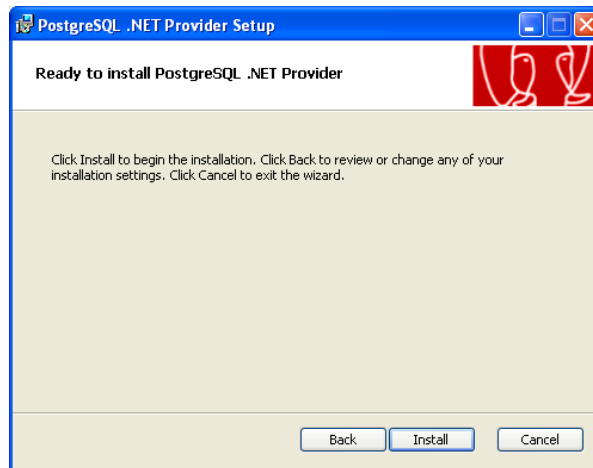
Chequear la casilla "I accept the terms in the License Agreement". Next:



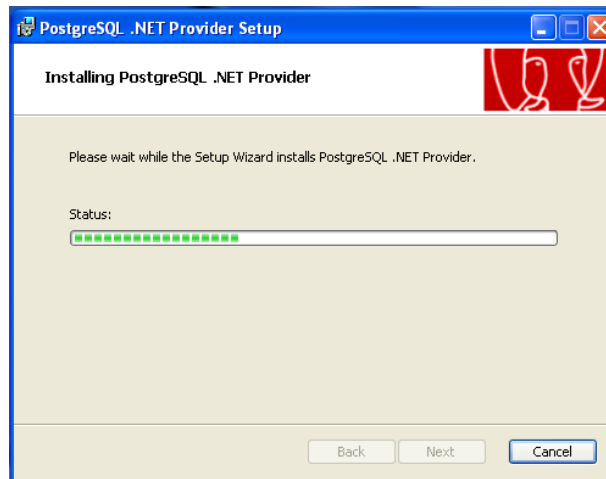
Escoger la carpeta de destino de la instalación. Next:



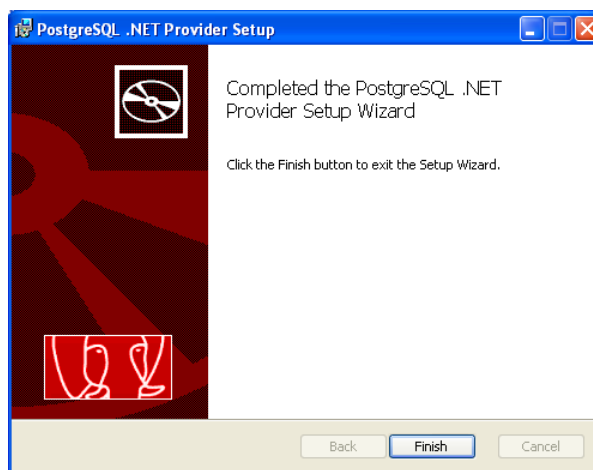
Clic en Install, para dar comienzo a la instalación:



Comienza el proceso de instalación:



Por último clic en Finish.

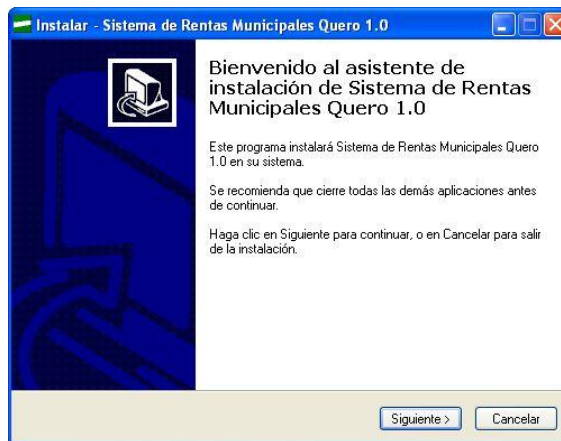


Instalación de “Sistema de Rentas Municipales Quero 1.0”

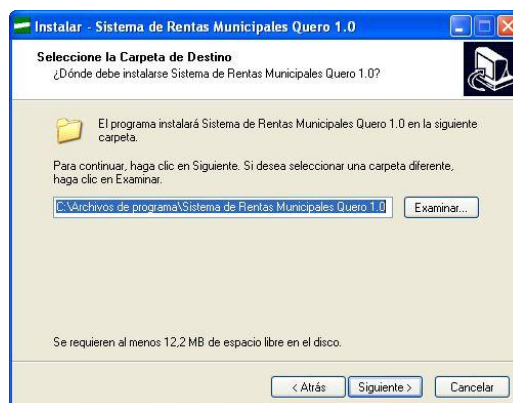
Para instalar el Sistema de Rentas Municipales Quero 1.0 primero se debe tener el instalador que es el archivo ejecutable:

Instalador_SistemaRentasMunicipalesQuero1.0.exe

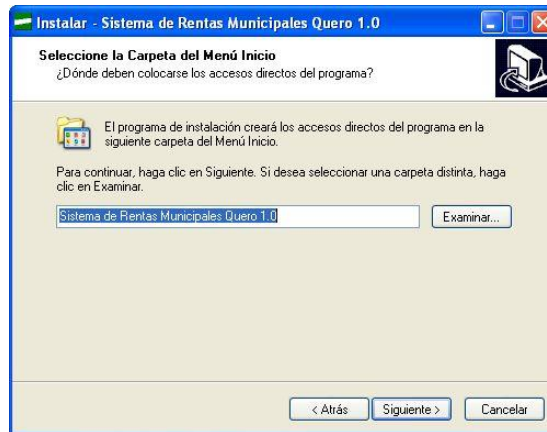
Entonces se procede a ejecutarlo haciendo doble clic sobre el archivo y aparece el asistente de instalación:



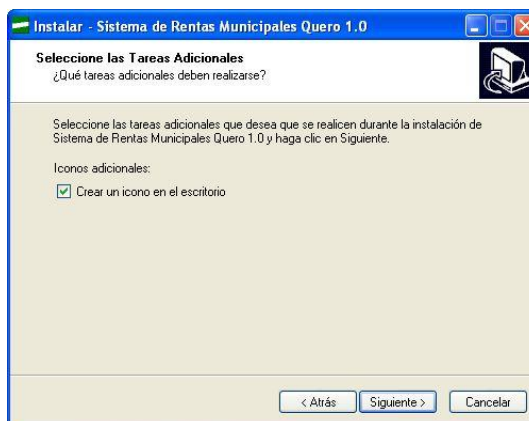
Clic en el botón Siguiente >.



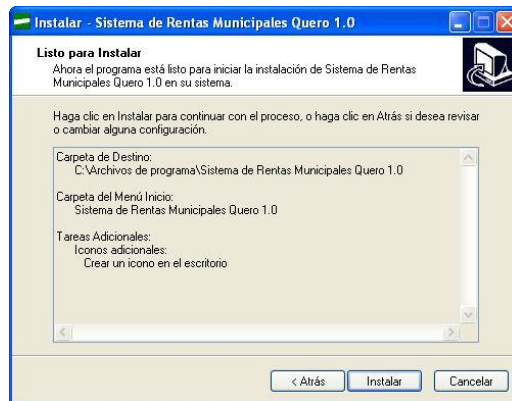
En la ventana que aparece se selecciona el directorio o carpeta de destino de instalación. Aparece la carpeta C:\Archivos de Programa por defecto, pero se puede cambiar dando clic en Examinar... Clic en Siguiente >.



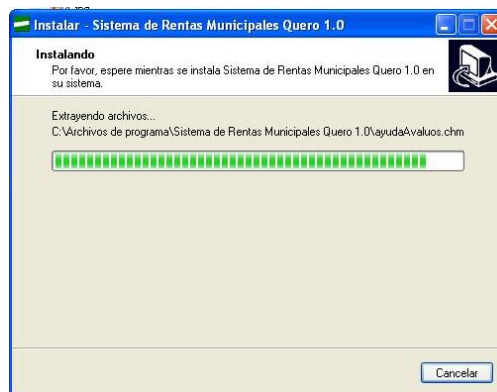
Aparece la ventana de creación de accesos directos. Siguiente >.



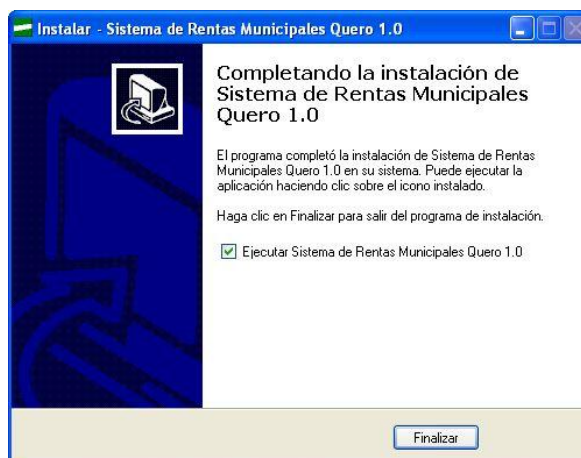
En la ventana de tareas adicionales chequear la casilla Crear un ícono en el escritorio. Siguiente >.



Aparece la ventana previa a la instalación con un resumen de la configuración de instalación. Clic en Instalar.



Se lleva a cabo la instalación. Y finalmente aparece la ventana informando que la instalación se ha completado. Clic en Finalizar:



El programa está listo para utilizarse.

ANEXO N° 4

MANUAL DE USUARIO

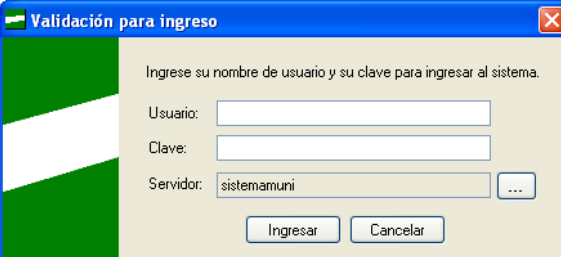
Introducción


Este sistema permite el manejo automatizado de trámites de alcabalas, plusvalías, rodajes, títulos de mercados, multas y derechos de instalación de medidores de agua potable. Es muy fácil de utilizar, por lo que su aprendizaje no tiene dificultad. El sistema brinda sus servicios a los departamentos de Avalúos y Catastros, Rentas, Tesorería y Centro de Cómputo, los cuales obtendrán un manejo eficiente de la información y una atención de calidad.

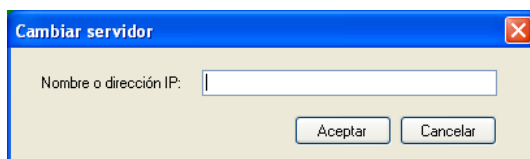
Ingresar al sistema

Para ingresar al sistema ir a Inicio > Todos los programa > Sistema de Rentas Municipales Quero 1.0 > Sistema de Rentas Municipales Quero 1.0.

Aparecerá la ventana de validación para ingresar al sistema:

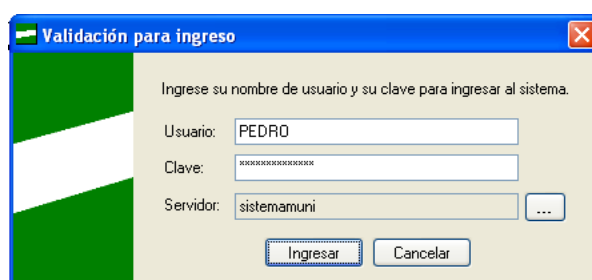


En dicha ventana se encuentran los campos Usuario, Clave y Servidor. Este último campo indica a que servidor se va conectar, de modo que si se desea cambiar de servidor simplemente hacer clic en el botón  , y aparecerá la ventana:



En donde se ingresará el nombre o la dirección IP del servidor.

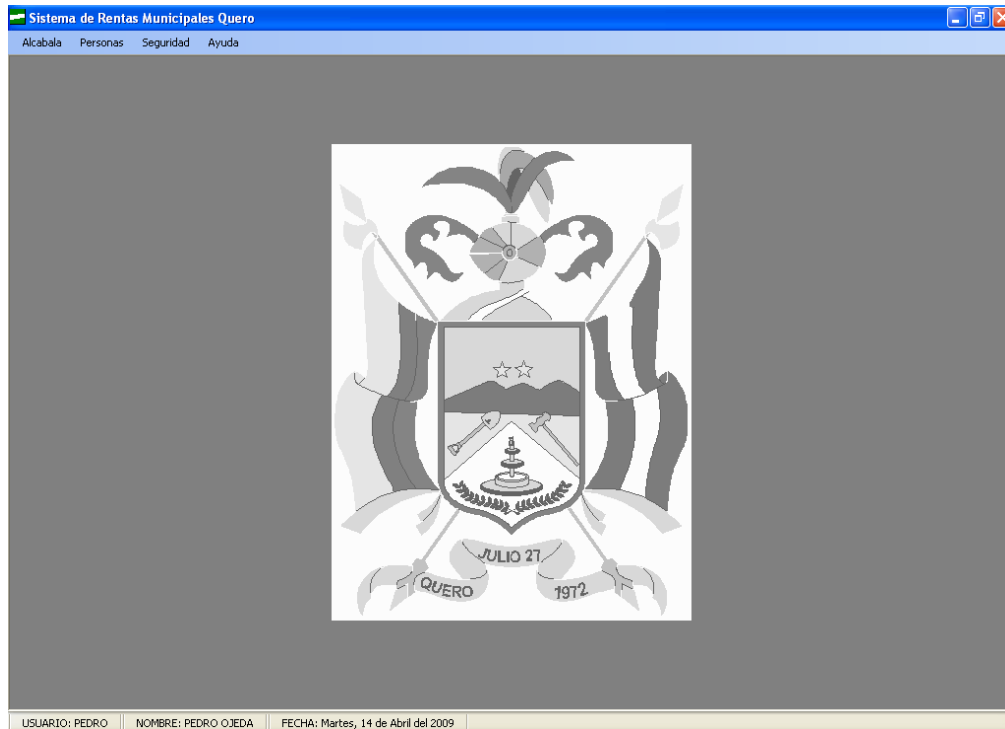
En cuanto a los campos Usuario y Clave se debe ingresar el nombre de usuario y la clave respectivamente, para ingresar al sistema:



El sistema cuenta con cuatro módulos correspondientes a los departamentos de Avalúos y Catastros, Rentas, Tesorería y Centro de Cómputo; por lo que las opciones no serán las mismas para todos los usuarios. A continuación se describe la funcionalidad de cada uno de los módulos.

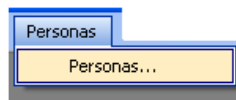
Módulo Departamento de Avalúos y Catastros

La ventana principal de este módulo es la siguiente:

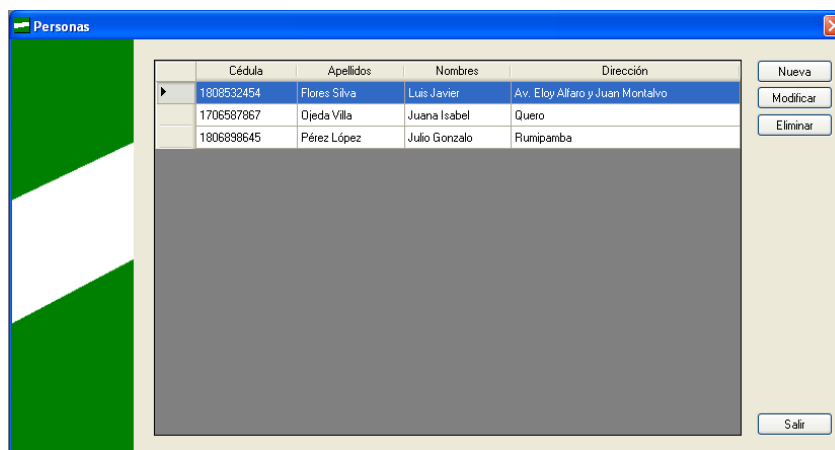


Entre las opciones de la barra de menú están: Alcabala, Personas, Seguridad y Ayuda.

Menú Personas



En este menú está la opción Personas... que nos permitirá manejar los datos de las personas almacenados en el sistema:



Para ingresar los datos de una nueva persona clic en el botón Nueva, y en la ventana que aparece ingresar los datos:

The dialog box 'Ingresar nueva persona' contains the following fields and controls:

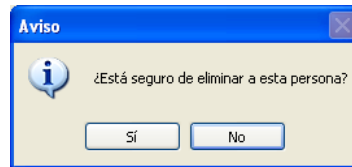
- Tipo: (dropdown menu)
- Cédula:
- Nombres:
- Apellidos:
- Dirección:
- Buttons: Aceptar, Cancelar

Se puede modificar los datos de las personas seleccionando el registro de la persona y haciendo clic en Modificar. Aparece una ventana con los datos de la persona, entonces se debe realizar los cambios:

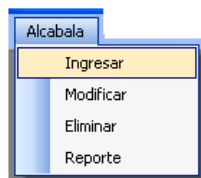
The dialog box 'Modificar datos de persona' contains the following fields and controls:

- Tipo: (dropdown menu)
- Cédula:
- Nombres:
- Apellidos:
- Dirección:
- Buttons: Aceptar, Cancelar

Por último, se puede eliminar los datos de una persona que no haya realizado algún trámite (caso contrario no se podrá eliminar), seleccionando el registro y haciendo clic en Eliminar, aparecerá un mensaje pidiendo una confirmación:



Menú Alcabala



En este menú están todas las tareas que se pueden realizar, en el departamento de Avalúos y Catastros, con los trámites de alcabalas. En el menú Alcabala se tiene las opciones: Ingresar, Modificar, Eliminar y Reporte.

Ingresar una alcabala.- Para ingresar una nueva alcabala ir al menú Alcabala > Ingresar, aparece la siguiente ventana:

A screenshot of a software window titled 'Ingreso de Alcabala'. The window has a blue title bar and a green and white decorative sidebar on the left. The main area contains several input fields and buttons. The fields are: 'N° Alcabala:' with the value '66'; 'Escritura de:'; 'Vendedor(es):' with a large text area and 'Agregar' and 'Borrar' buttons; 'Comprador(es):' with a large text area and 'Agregar' and 'Borrar' buttons; 'Objeto de venta:'; 'Ubicación:'; and 'Valor:'. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

El número de alcabala se genera automáticamente. Los demás campos deben ser ingresados:

- En escritura ingresar el tipo de escritura (compra/venta, donación, partición, etc.).
- Para ingresar los compradores y vendedores clic en el botón Agregar y aparece la ventana para buscar personas:

Buscar Persona

Búsqueda

Por cédula. Por nombre.

Ingrese la cédula:

Ingrese los apellidos y luego los nombres:

Cédula	Apellidos y Nombres
--------	---------------------

Personas... Aceptar Cancelar

La búsqueda de personas se puede hacer por el número de cédula o por los nombres. A medida que se va ingresando la cédula o los nombres sigue apareciendo el listado de personas:

Buscar Persona

Búsqueda

Por cédula. Por nombre.

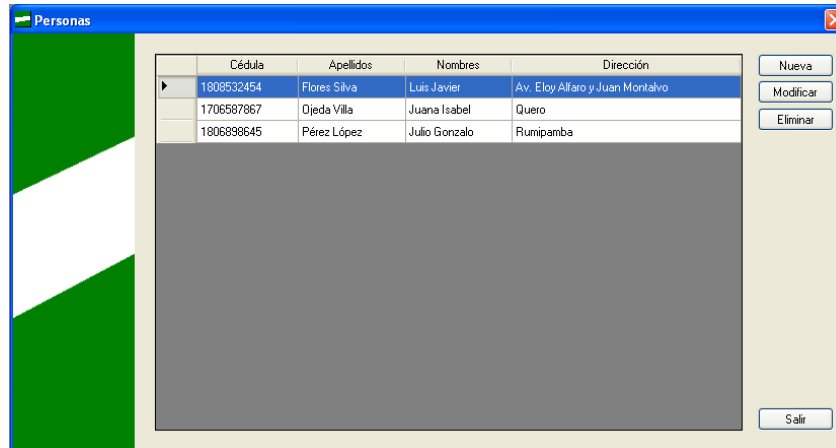
Ingrese la cédula:

Ingrese los apellidos y luego los nombres:

Cédula	Apellidos y Nombres
1808532454	Flores Silva Luis Javier
1806898645	Pérez López Julio Gonzalo

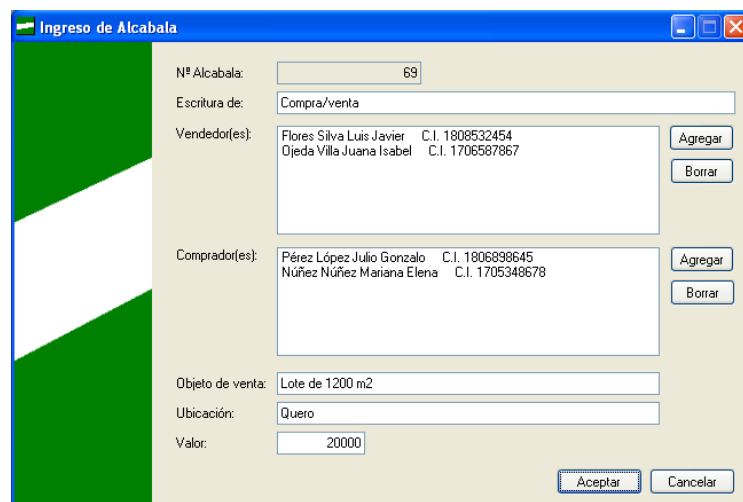
Personas... Aceptar Cancelar

En caso de que la persona no conste en la lista, se trata de una persona nueva así que se debe ingresar los datos, para ello clic en el botón Personas...



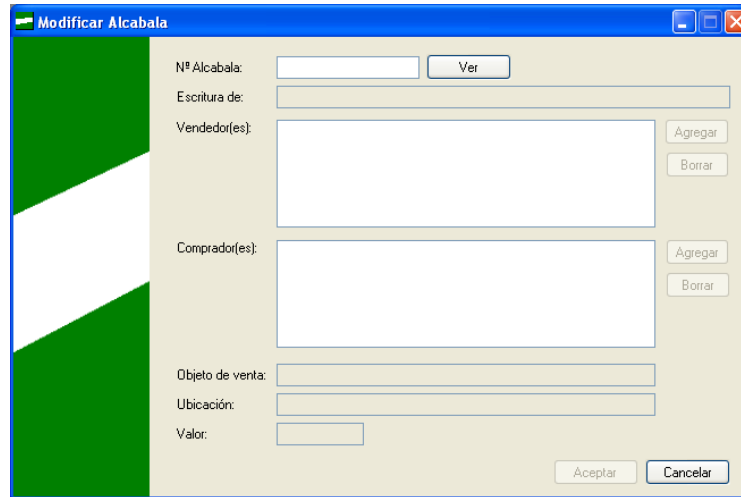
Se puede ver un listado de las personas, y se puede ingresar, modificar o eliminar.

- Finalmente se ingresa el objeto que se está vendiendo, su ubicación y valor o avalúo. Clic en Aceptar:



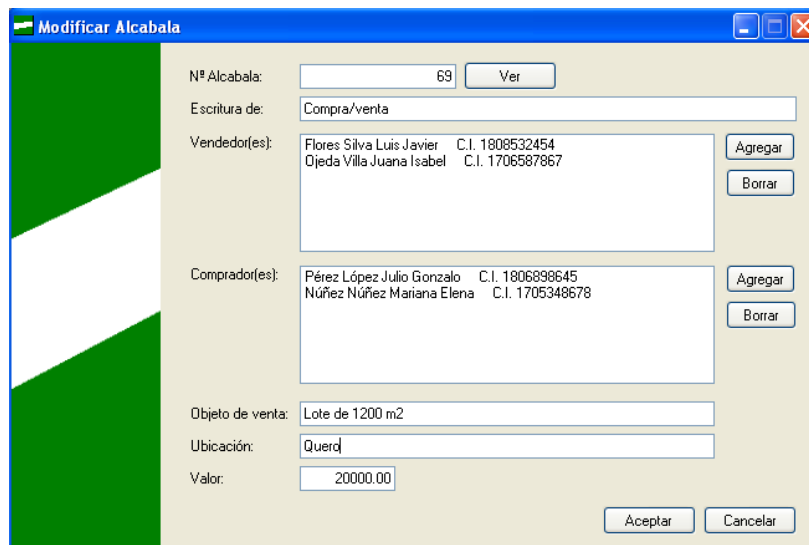
El trámite de alcabala ha sido ingresado y está listo para ser liquidado en el departamento de Rentas.

Modificar una alcabala.- Para modificar una alcabala ir al menú Alcabala > Modificar, aparece la siguiente ventana:



The screenshot shows a window titled "Modificar Alcabala" with a blue title bar. The window contains several input fields and buttons. On the left side, there is a green and white diagonal striped graphic. The fields are: "Nº Alcabala:" with an empty text box and a "Ver" button; "Escritura de:" with an empty text box; "Vendedor(es):" with an empty list box and "Agregar" and "Borrar" buttons; "Comprador(es):" with an empty list box and "Agregar" and "Borrar" buttons; "Objeto de venta:" with an empty text box; "Ubicación:" with an empty text box; and "Valor:" with an empty text box. At the bottom right, there are "Aceptar" and "Cancelar" buttons.

En el campo Nº Alcabala se ingresa el número de la alcabala a modificar y clic en Ver para visualizarla. Entonces se procede a realizar las modificaciones y Aceptar:



The screenshot shows the same "Modificar Alcabala" window, but now with data entered. The "Nº Alcabala:" field contains the number "69" and the "Ver" button is visible. The "Escritura de:" field contains the text "Compra/venta". The "Vendedor(es):" list box contains two entries: "Flores Silva Luis Javier C.I. 1806532454" and "Ojeda Villa Juana Isabel C.I. 1706587867". The "Comprador(es):" list box contains two entries: "Pérez López Julio Gonzalo C.I. 1806896645" and "Núñez Núñez Mariana Elena C.I. 1705348678". The "Objeto de venta:" field contains "Lote de 1200 m2". The "Ubicación:" field contains "Querú". The "Valor:" field contains "20000.00". The "Aceptar" and "Cancelar" buttons are at the bottom right.

Eliminar una alcabala.- Se puede eliminar una alcabala que no haya sido liquidada, para ello menú Alcabala > Eliminar. En la ventana que aparece se ingresa el número de la alcabala a liquidar y clic en Ver para visualizar el trámite:

Eliminar Alcabala

Nº Alcabala: 69 Ver

Escritura de: Compra/venta

Vendedor(es): Flores Silva Luis Javier C.I. 1808532454
Djeda Villa Juana Isabel C.I. 1706587867

Comprador(es): Pérez López Julio Gonzalo C.I. 1806898645
Núñez Núñez Mariana Elena C.I. 1705348678

Objeto de venta: Lote de 1200 m2

Ubicación: Quero

Valor: 20000.00

Eliminar Cancelar

Clic en eliminar, y aparecerá un mensaje de confirmación:

Advertencia

¿Está seguro de eliminar esta alcabala?

Sí No

Reporte de alcabalas ingresadas.- Para generar e imprimir reportes de las alcabalas ingresadas: menú Alcabala > Reporte. Aparece la ventana de reportes:

Reporte de Alcabalas Ingresadas

Fecha de ingreso:

Desde: Martes , 14 de Abril de 2009 Ver

Hasta: Martes , 14 de Abril de 2009 Imprimir

En la parte superior izquierda de la ventana se encuentra los campos Desde y Hasta en los cuales se escoge las fechas del intervalo de tiempo del cual se quiere obtener el reporte. Clic en Ver:

N° Alcabala	F. Ingreso	Escritura	Vendedor	Comprador	Objeto de Venta	Ubicación	Valor
69	14/04/2009	Compra/venta	Flores Silva Luis ...	Pérez López Julio...	Lote de 1200 m2	Quero	20000.00

Para imprimir el reporte clic en Imprimir.

Impresora: Epson FX-1180
Tamaño del papel: reporte
Orientación:
 Vertical
 Horizontal
Imprimir Cancelar

Se escoge la impresora, tamaño y orientación del papel, y se imprime el reporte:

I. Municipio de Quero
Alcabalas Ingresadas
Martes, 14 de Abril de 2009

Nº Alcabala	F. Ingreso	Escritura	Vendedor	Comprador	Objeto de venta	Ubicación	Valor
69	14/04/2009	Compra/Venta	Flores Silva Luis Javier 1808532454 y otros	C.I. Pérez López Julio Gonzalo 1808898843 y otros	Lote de 1200 m2	Quero	20000.00

Martes, 14 de Abril del 2009 1 de 1

Menú Seguridad

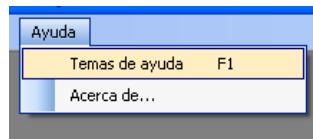


Contiene la opción Cambiar clave, que permite al usuario actual cambiar su clave:

 A screenshot of a dialog box titled 'Cambiar clave de usuario'. It contains three input fields: 'Clave actual:', 'Nueva clave:', and 'Repita la nueva clave:'. Each field is filled with asterisks. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

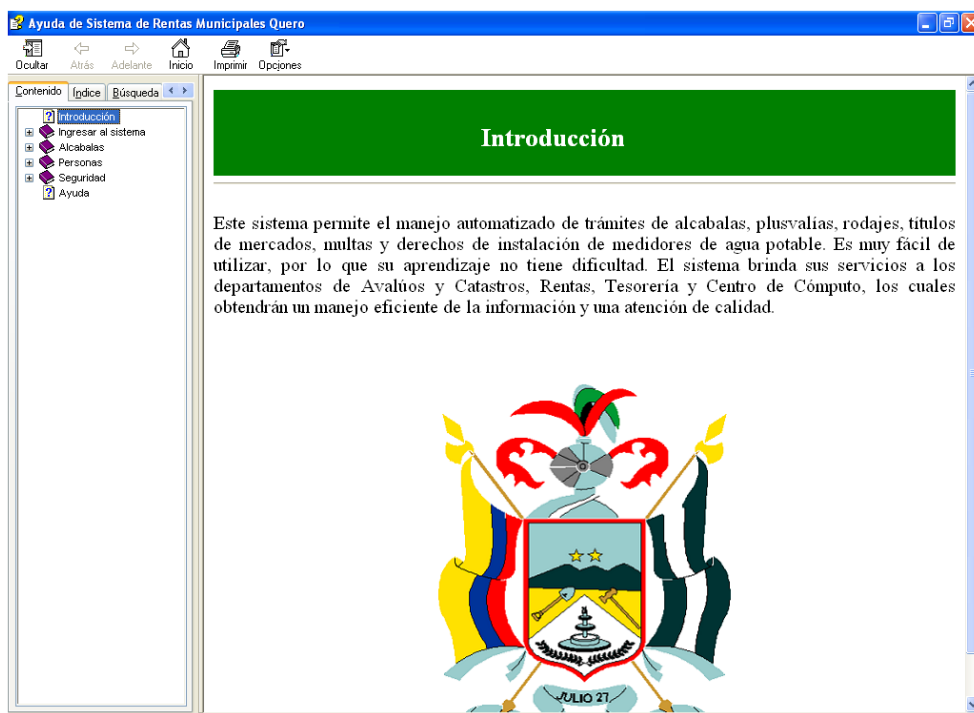
Se ingresa primero la clave actual, luego se ingresa la nueva clave, por último se repite la nueva clave, y Aceptar.

Menú Ayuda

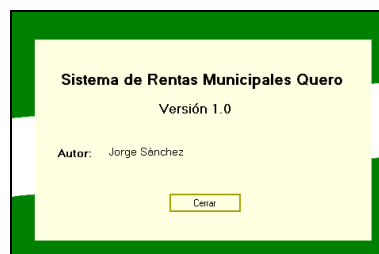


Tiene las siguientes opciones:

Temas de ayuda.- Con esta opción se muestra la ayuda del sistema:

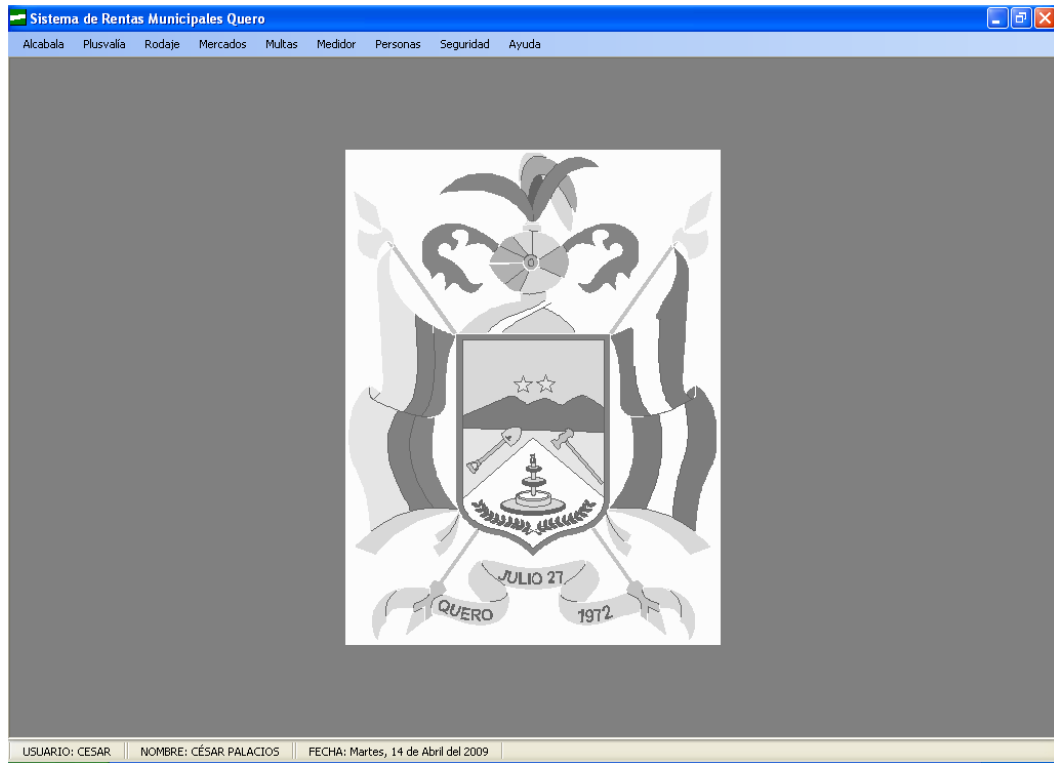


Acerca de.- Muestra la ventana Acerca de...



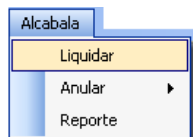
Módulo Departamento de Rentas

La ventana principal tiene el siguiente aspecto:



Entre las opciones de la barra de menú están: Alcabala, Plusvalía, Rodaje, Mercados, Multas, Medidor, Personas, Seguridad y Ayuda.

Menú Alcabala



En este menú están todas las operaciones que se pueden realizar, en el departamento de Rentas, con los trámites de alcabalas. En el menú Alcabala se tiene las opciones: Liquidar, Anular y Reporte.

Liquidar una alcabala.- Ir a menú Alcabala > Liquidar, aparece la ventana:

Liquidación de Alcabalas

Nº Alcabala:

Avalúo

Escritura de:

Vendedor(es):

Comprador(es):

Objeto de venta:

Ubicación:

Valor:

Liquidación

Municipio: Exento

Const. Escolares:

Total:

Observación:

En el campo Nº Alcabala se ingresa el número de la alcabala a liquidar y clic en Ver:

Liquidación de Alcabalas

Nº Alcabala:

Avalúo

Escritura de:

Vendedor(es): Flores Silva Luis Javier C.I. 1808532454
Ojeda Villa Juana Isabel C.I. 1706587867

Comprador(es): Pérez López Julio Gonzalo C.I. 1806898645
Núñez Núñez Mariana Elena C.I. 1705348678

Objeto de venta:

Ubicación:

Valor:

Liquidación

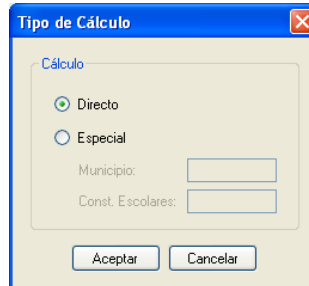
Municipio: Exento

Const. Escolares:

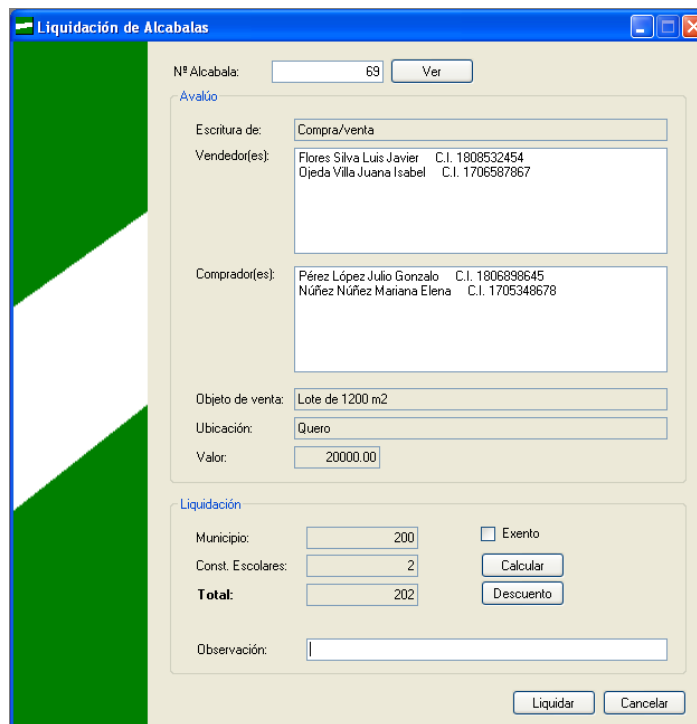
Total:

Observación:

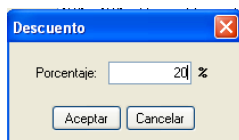
Una vez que ya se ha visualizado el trámite se procede a su liquidación. Para ello clic en Calcular, aparece la ventana que pide el tipo de cálculo: directo o especial



Con la opción Directo se hace el cálculo normal, con Especial se debe ingresar el valor del impuesto al Municipio y Construcciones Escolares. Clic en Aceptar, y aparecerá los resultados del cálculo:



Si por alguna razón se debe realizar un descuento clic en Descuento. Aparece una ventana en la cual se ingresa el porcentaje del descuento.

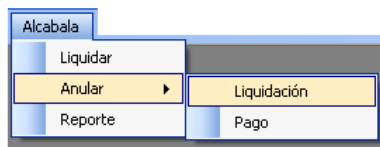


De igual forma, si por alguna razón este trámite debe ser exento de pago se debe chequear la casilla Exento Exento para que los valores de Municipio y Const. Escolares sean 0:

Municipio:	<input type="text" value="0.00"/>
Const. Escolares:	<input type="text" value="0.00"/>
Total:	<input type="text" value="0.00"/>

Por último se debe ingresar una observación, y clic en Liquidar para finalizar la liquidación:

Anular alcabalas.- Hay dos tipos de anulación de alcabalas: anulación de la liquidación y anulación del pago:



Para anular una liquidación de alcabala ir a menú Alcabala > Anular > Liquidación. En la ventana que aparece, en el campo N° Alcabala se ingresa el número de alcabala a anular y clic en Ver:

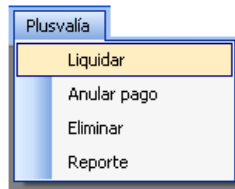
Se muestra el trámite. Clic en Anular liquidación, aparece un mensaje para confirmar la anulación:

Para anular el pago de una alcabala: menú Alcabala > Anular > Pago. En la ventana que aparece en el campo Nº Alcabala se ingresa el número de la alcabala cuyo pago se desea anular y clic en Ver:

Clic en Anular pago, aparece un mensaje para confirmar la anulación del pago:

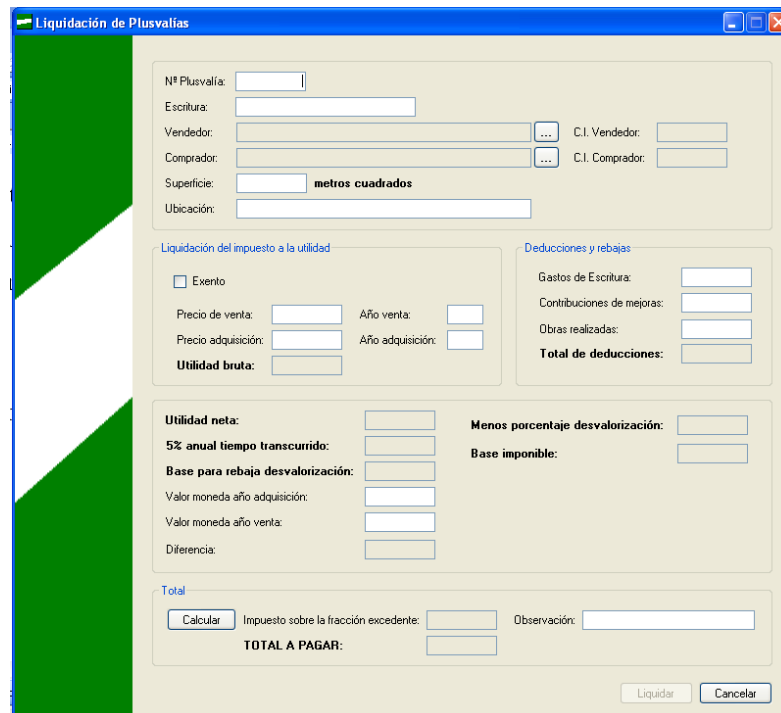
Reporte de alcabalas liquidadas.- Para generar e imprimir reportes de las alcabalas liquidadas: menú Alcabala > Reporte. Aparece la ventana de reportes:

Menú Plusvalía




En este menú están todas las operaciones que se pueden realizar con los trámites de plusvalía en el departamento de Rentas. En el menú Plusvalía se tiene las opciones: Liquidar, Anular pago, Eliminar y Reporte.

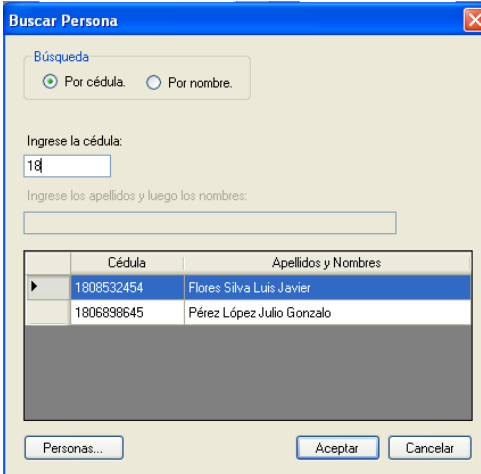
Liquidar una plusvalía.- Para liquidar una plusvalía ir a menú Plusvalía > Liquidar:

A screenshot of a software window titled 'Liquidación de Plusvalías'. The window contains several input fields and sections for calculating plusvalía tax. The fields include: 'Nº Plusvalía', 'Escritura', 'Vendedor', 'Comprador', 'Superficie' (with a unit of 'metros cuadrados'), and 'Ubicación'. There are also fields for 'C.I. Vendedor' and 'C.I. Comprador'. The 'Liquidación del impuesto a la utilidad' section has a checkbox for 'Exento', and fields for 'Precio de venta', 'Año venta', 'Precio adquisición', and 'Año adquisición'. The 'Deducciones y rebajas' section has fields for 'Gastos de Escritura', 'Contribuciones de mejoras', and 'Obras realizadas'. The 'Utilidad neta' section has fields for 'Utilidad bruta', '5% anual tiempo transcurrido', 'Base para rebaja desvalorización', 'Valor moneda año adquisición', 'Valor moneda año venta', and 'Diferencia'. The 'Menos porcentaje desvalorización' and 'Base imponible' sections have their own fields. At the bottom, there is a 'Total' section with a 'Calcular' button, 'Impuesto sobre la fracción excedente', 'Observación', and 'TOTAL A PAGAR' field. There are also 'Liquidar' and 'Cancelar' buttons at the bottom right.

Se debe ingresar los siguientes datos:

- El número de plusvalía.
- El tipo de escritura (compra/venta, partición, etc.).

- Vendedor y comprador. Para llenar estos campos hacer clic en  para buscar o ingresar a la persona:



Cédula	Apellidos y Nombres
1808532454	Flores Silva Luis Javier
1806898645	Pérez López Julio Gonzalo

- Superficie y ubicación del lote o terreno.
- Precio y año de venta del lote o terreno.
- Precio y año de adquisición del lote o terreno.
- Valor de los gastos de escritura.
- Valor de contribución de mejoras.
- Valor de obras realizadas.
- Valores de la moneda en los años de adquisición y venta.

Luego se procede a calcular haciendo clic en Calcular y se ingresa una observación:

Nº Plusvalía:	45	
Escritura:	Compra/venta	
Vendedor:	Ojeda Villa Juana Isabel	C.I. Vendedor: 1706587867
Comprador:	Núñez Núñez Mariana Elena	C.I. Comprador: 1705348678
Superficie:	3500 metros cuadrados	
Ubicación:	San Vicente	

Liquidación del impuesto a la utilidad		Deducciones y rebajas	
<input type="checkbox"/> Exento		Gastos de Escritura:	10
Precio de venta:	10000	Contribuciones de mejoras:	5
Precio adquisición:	1000	Obras realizadas:	12
Año venta:	2008	Total de deducciones:	27
Año adquisición:	1999		
Utilidad bruta:	9000		

Utilidad neta:	8973	Menos porcentaje desvalorización:	0.00
5% anual tiempo transcurrido:	3140.55	Base imponible:	5832.45
Base para rebaja desvalorización:	5832.45		
Valor moneda año adquisición:	0		
Valor moneda año venta:	0		
Diferencia:	0		

Total	
<input type="button" value="Calcular"/>	Impuesto sobre la fracción excedente: 29.16
	Observación: Ninguna
TOTAL A PAGAR:	29.16

Finalmente clic en Liquidar.

Anular el pago de una plusvalía.- Se puede anular el pago de una plusvalía en menú Plusvalía > Anular pago. En la ventana que aparece se debe teclear el número de plusvalía a anular y clic en Ver:

Anulación de Pago de Plusvalía

Nº Plusvalía:

Escritura:

Vendedor: C.I. Vendedor:

Comprador: C.I. Comprador:

Superficie: **metros cuadrados**

Ubicación:

Liquidación del impuesto a la utilidad

Precio de venta: Año venta:

Precio adquisición: Año adquisición:

Utilidad bruta:

Deducciones y rebajas

Gastos de Escritura:

Contribuciones de mejoras:

Obras realizadas:

Total de deducciones:

Utilidad neta: **Menos porcentaje desvalorización:**

5% anual tiempo transcurrido: **Base imponible:**

Base para rebaja desvalorización:

Valor moneda año adquisición:

Valor moneda año venta:

Diferencia:

Total

Impuesto sobre la fracción excedente: Observación:

TOTAL A PAGAR:

Clic en Anular pago. Sale un mensaje de advertencia pidiendo confirmar la anulación:

Advertencia

¿Está seguro de anular el pago de esta plusvalía?

Eliminar una plusvalía.- Las plusvalías que aún no han sido pagadas se pueden eliminar. Para ello ir a menú Plusvalía > Eliminar. En la ventana que aparece se debe teclear el número de plusvalía a eliminar y clic en Ver:

Eliminar plusvalía

Nº Plusvalía:

Escritura:

Vendedor: C.I. Vendedor:

Comprador: C.I. Comprador:

Superficie: **metros cuadrados**

Ubicación:

Liquidación del impuesto a la utilidad

Precio de venta: Año venta:

Precio adquisición: Año adquisición:

Utilidad bruta:

Deducciones y rebajas

Gastos de Escritura:

Contribuciones de mejoras:

Obras realizadas:

Total de deducciones:

Utilidad neta: **Menos porcentaje desvalorización:**

5% anual tiempo transcurrido: **Base imponible:**

Base para rebaja desvalorización:

Valor moneda año adquisición:

Valor moneda año venta:

Diferencia:

Total

Impuesto sobre la fracción excedente: Observación:

TOTAL A PAGAR:

Clic en eliminar y confirmar la eliminación:

Advertencia

¿Está seguro de que desea eliminar esta plusvalía?

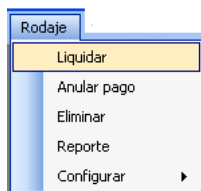
Reporte de plusvalías liquidadas.- Para generar e imprimir reportes de las plusvalías liquidadas: menú Plusvalía > Reporte. Aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

I. Municipio de Quero
Plusvalías Liquidadas
Martes, 14 de Abril de 2009

Nº Plusvalía	F. Liquidación	Escritura	Vendedor	Comprador	Precio Venta	Año Venta	Precio Adq.	Año Adq.	Total
45	14/04/2009	Compra/venta	Ojeda Villa Juana Isabél	Mariana Elena Núñez Núñez	10000.00	2008	1000.00	1999	29.16
TOTALES:									29.16

Martes, 14 de Abril del 2009 Directo Financiero Jefe de Rentas Tesorero Municipal 1 de 1

Menú Rodaje



En este menú están todas las tareas que se pueden realizar con los trámites de rodaje vehicular en el departamento de Rentas. En el menú Rodaje se tiene las opciones: Liquidar, Anular pago, Eliminar, Reporte y Configurar.

Liquidar un rodaje vehicular.- Ir a menú Rodaje > Liquidar. Aparece la ventana:

Clic en Buscar persona... para encontrar o ingresar los datos de la persona que desea hacer el trámite. Una persona puede realizar varios rodajes simultáneamente por lo que por cada vehículo del cual se desea hacer el trámite se debe ingresar un registro. Para ingresar un registro clic en Agregar:

A screenshot of a software dialog box titled "Agregar". The dialog box contains the following fields and controls:

- Año: 2009
- Placa: TCP-256
- Avalúo vehículo: 12000
- Clase: Camioneta
- Matrícula año anterior: 233423
- Exento
- Serv. Téc. Adm.: 0.50
- Valor: 15.00
- Interés: 0.00
- Calcular (button)
- Total: 15.5
- Observación: Ninguna
- Aceptar (button)
- Cancelar (button)

En la ventana que aparece se ingresa lo siguiente:

- Año del cual se hace el rodaje.
- Placa, avalúo, clase y número de matrícula del vehículo.
- El valor por servicios técnicos/administrativos.

Luego clic en Calcular para obtener los valores, se ingresa una observación y Aceptar.

Se puede modificar un rodaje seleccionándolo y haciendo clic en Modificar. Además se puede eliminar un rodaje no deseado haciendo clic en Eliminar.

Una vez que se han ingresado el (los) rodaje(s) clic en Liquidar:

Persona

Nombre: Pérez López Julio Gonzalo Buscar persona...

C.I.: 1806898645

Rodajes vehiculares

	Año	Placa	Avalúo	Clase	Matrícula	Valor	Interés	Serv. Téc. Adm.
▶	2009	TCP-256	12000	Camioneta	233423	15.00	0.00	0.50

Agregar
Modificar
Eliminar

Liquidar Cancelar

Anular el pago de un rodaje.- Ir a menú Rodaje > Anular pago:

Anulación de Pago de Rodajes

Nº Rodaje: Buscar

Nombre: C.I.:

Año:

Placa:

Avalúo vehículo:

Clase:

Matrícula año anterior:

Valor: Interés: Servicio Téc. Adm.: Total:

Observación:

Anular pago Cancelar

Para identificar al rodaje deseado clic en Buscar. Ingresar la placa del vehículo y el año del rodaje.

Buscar Rodaje

Placa: TCP0256

Año: 2009

Aceptar Cancelar

Se visualiza el trámite. Clic en Anular pago:

Confirmar la anulación:

Eliminar un rodaje.- Se puede eliminar los rodajes solo si no han sido pagados aún, caso contrario no se podrán eliminar. Menú Rodaje > Eliminar. En la ventana que aparece clic en Buscar. Una vez identificado el trámite clic en Eliminar:

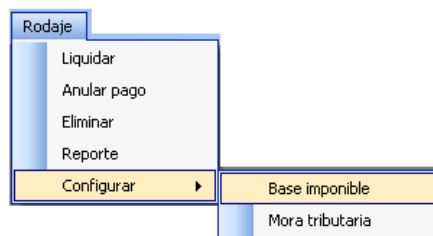
Confirmar la eliminación:

Reporte de rodajes liquidados.- Para generar e imprimir reportes de los rodajes liquidados: menú Rodaje > Reporte. Aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

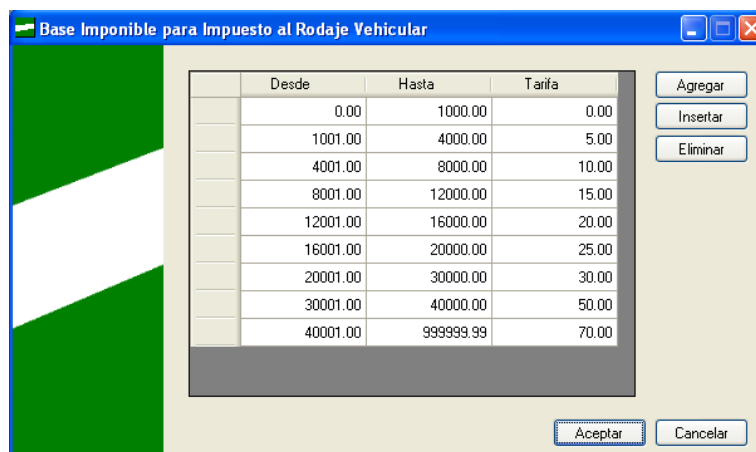
I. Municipio de Quero													
Rodajes Vehiculares Liquidados													
Martes, 14 de Abril de 2009													
Título N°	F. Liq.	Nombre	Cédula	Año	Placa	Avaluo	Clase	Matricula	Valor	Interés	Serv. Téc.	Total	Observación
4	14/04/2009	Pérez López, Julio Gonzalo	1806898645	2009	TCPO256	12000.00	Camioneta	233423	15.00	0.00	0.50	15.50	Ninguna
TOTALES:									15.00	0.00	0.50	15.50	

Martes, 14 de Abril del 2009 Directo Financiero Jefe de Rentas Tesorero Municipal 1 de 1

Configuración.- Se puede configurar la tabla de la Base Imponible y la tabla de intereses por Mora Tributaria, que son utilizados para calcular la liquidación de rodajes:

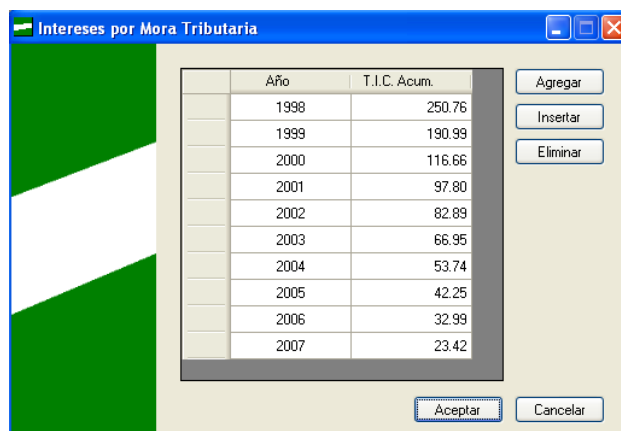


Para configurar la tabla de la Base Imponible ir a menú Rodaje > Configurar > Base imponible:



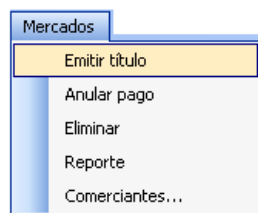
Se realiza las modificaciones necesarias agregando, insertando o eliminando rangos y tarifas. Aceptar.

Para configurar la tabla de intereses por Mora Tributaria ir a menú Rodaje > Configurar > Mora tributaria:



Se realiza las modificaciones necesarias agregando, insertando o eliminando años y sus valores correspondientes. Aceptar.

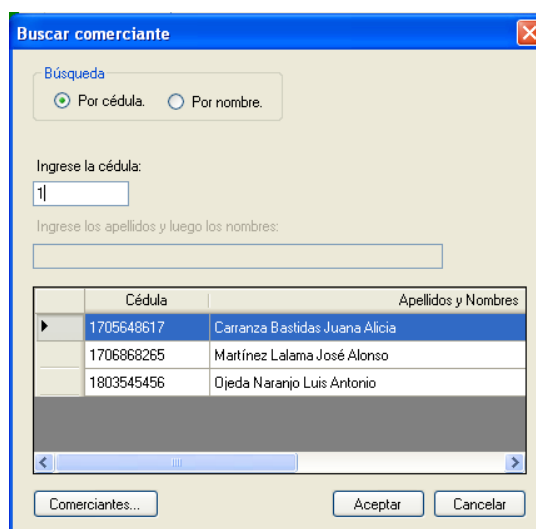
Menú Mercados



En este menú están todas las tareas que se pueden realizar con los trámites de mercados en el departamento de Rentas. En el menú Mercados se tiene las opciones: Emitir título, Anular pago, Eliminar, Reporte y Comerciantes.

Emitir un título de mercado.- Ir a menú Mercados > Emitir título. En la ventana que aparece se ingresa lo siguiente:

- Departamento que solicita la emisión del título.
- Nombre del comerciante. Para ello clic en para buscar al comerciante:



- Número de puesto en el mercado o plaza.
- Se debe definir si el pago es de todo el año o solo de algunos meses.
- Valor de la mensualidad.
- Valor del activo total.
- Valor por recolección de basura.
- Valor por aferición de pesas y medidas.
- Valor por servicios administrativos.
- Valor por consumo de energía eléctrica.

- Valor del carnet.
- Valor del contrato.
- Valor por trámite municipal.
- Valor de recargo y/o intereses.

Luego se obtiene los resultados haciendo clic en Calcular. Aceptar.

Emisión de título de mercado

Solicitado por: Administración de Mercados

Comerciante: Carranza Bastidas Juana Alicia

C.I.: 1705648617 Actividad: Minorista

Nº Puesto: 120

Dirección: Av. 17 de Abril y Eloy Alfaro

Pago

Anual Por meses

Desde mes de: Enero Hasta mes de: Diciembre

Año: 2009

Especies Valoradas	
Carnet:	10.00
Contrato:	4.00
Trámite municipal:	0.80
Subtotal 2:	14.80

Mensualidad:	60.00
Activo total:	25.00
Recolección de basura:	6.00
Aferición pesas y medidas:	5.00
Servicios administrativos:	5.00
Energía eléctrica:	0.00
Subtotal 1:	101.00

Otros

Recargo y/o intereses: 0.00

Total

Total: 115.80

Anular el pago de un título de mercado.- Ir a menú Mercados > Anular pago. En la ventana que aparece se debe teclear el número de título de mercado cuyo pago se desea anular y clic en Ver:

Clic en Anular pago y confirmar la anulación:

Eliminar un título de mercado.- Se puede eliminar sólo los títulos de mercado que no han sido pagados aún. Para ello ir a menú Mercados > Eliminar. En la ventana que aparece se debe teclear el número de título de mercado a eliminar y clic en Ver:

Eliminar título de mercado

Título N°:

Solicitado por:

Comerciante:

C.I.: Actividad:

N° Puesto:

Dirección:

Pago:

Año: Desde mes de: Hasta mes de:

Especies Valoradas	
Carnet:	<input type="text" value="10.00"/>
Contrato:	<input type="text" value="4.00"/>
Trámite municipal:	<input type="text" value="0.80"/>
Subtotal 2:	<input type="text" value="14.80"/>

Mensualidad:	<input type="text" value="60.00"/>
Activo total:	<input type="text" value="25.00"/>
Recolección de basura:	<input type="text" value="6.00"/>
Aferición pesas y medidas:	<input type="text" value="5.00"/>
Servicios administrativos:	<input type="text" value="5.00"/>
Energía eléctrica:	<input type="text" value="0.00"/>
Subtotal 1:	<input type="text" value="101.00"/>

Otros

Recargo y/o intereses:

Total

TOTAL:

Clic en Eliminar y confirmar la eliminación:

Advertencia

¿Está seguro de eliminar este título de mercado?

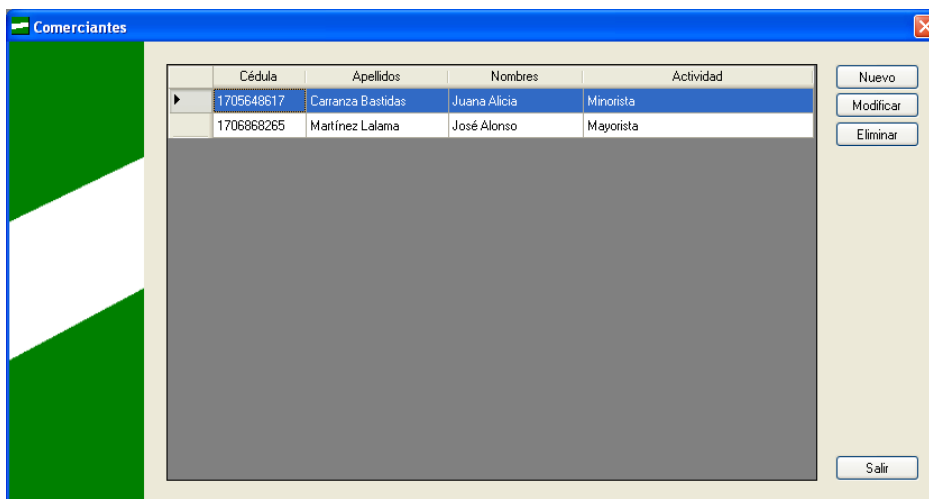
Reporte de títulos de mercado emitidos.- Para generar e imprimir reportes de los títulos de mercado emitidos ir a menú Mercados > Reporte. Aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

I. Municipio de Ouseo
Títulos de Mercado Emitidos
Martes, 14 de Abril de 2009

Título N°	F. Emisión	Comerciante	Dirección	Desde	Hasta	Año	Mensualidad	Act. Total	Rec. Basura	Pesas y Med.	Serv. Adm.	Energ. Elec.	Carnet	Contrato	Tram. Munic.	Recarg./inter.	Total
3	14/04/2009	Carranza Bastidas Juana Alicia	Av. 17 de Abril y Eloy Alfaro	Enero	Diciembre	2009	60.00	25.00	6.00	5.00	5.00	0.00	10.00	4.00	0.80	0.00	115.80
TOTALES:							60.00	25.00	6.00	5.00	5.00	0.00	10.00	4.00	0.80	0.00	115.80

Martes, 14 de Abril del 2009 Directo Financiero Jefe de Rentas Tesorero Municipal 1 de 1

Comerciantes.- La opción Comerciantes... permite ver una lista de todos los comerciantes ingresados en el sistema:



Para ingresar un nuevo comerciante clic en Nuevo. Ingresar los datos y Aceptar:

Ingresar nuevo comerciante

Cédula: 1803545456

Nombres: Luis Antonio

Apellidos: Ojeda Naranjo

Actividad: Minorista

Aceptar Cancelar

Para modificar los datos de un comerciante se debe seleccionar el registro y luego dar clic en Modificar. Realizar los cambios y Aceptar:

Modificar datos de comerciante

Cédula: 1803545456

Nombres: Luis Antonio

Apellidos: Djeda Naranjo

Actividad:

Aceptar Cancelar

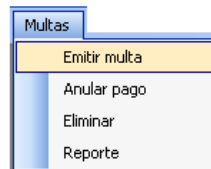
Para eliminar seleccionar el registro y clic en Eliminar. Se debe confirmar la eliminación:

Aviso

¿Está seguro de eliminar a este comerciante?

Sí No

Menú Multas



En este menú están todas las tareas que se pueden realizar con las multas en el departamento de Rentas. En el menú Multas se tiene las opciones: Emitir multa, Anular pago, Eliminar y Reporte.

Emitir una multa.- Ir a menú Multas > Emitir multa. En la ventana que aparece se ingresa lo siguiente:

- Departamento que solicita la emisión de la multa.
- Persona a quien va dirigida la multa, clic en para buscarla:

Cédula	Apellidos y Nombres
1808532454	Flores Silva Luis Javier
1806898645	Pérez López Julio Gonzalo

- Concepto o razón de la multa.
- Valor a pagar.
- Valor por gastos administrativos.
- Valor por servicios técnicos.
- Valor de interés.

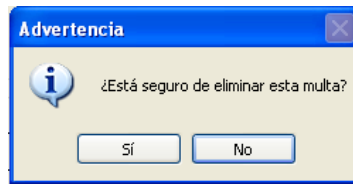
Luego se obtiene los resultados haciendo clic en Calcular. Finalmente clic en Aceptar:

Anular el pago de una multa.- El pago de las multas se puede anular con la opción Multas > Anular pago. En la ventana que aparece se debe teclear el número de título de la multa pagada y clic en Ver:

Clic en anular pago, y confirmar la anulación de pago:

Eliminar una multa.- Las multas que aún no han sido pagadas se pueden eliminar. Para esto ir a menú Multas > Eliminar. En la ventana que aparece se debe teclear el número de título de la multa a eliminar y clic en Ver:

Clic en Eliminar y confirmar la eliminación:



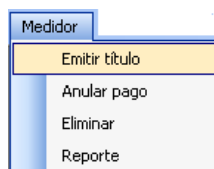
Reporte de multas emitidas.- Para generar e imprimir reportes de las multas emitidas ir a menú Multas > Reporte. Aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

I. Municipio de Quero
Multas Emitidas
Miércoles, 15 de Abril de 2009

Título N°	F. emisión	Nombre o razón	Dirección	Solicitado por	Concepto	V. pagar	Gastos Adm.	Serv. téc.	Interés	Total
3	15/04/2009	Palacios Garces Juan Alfredo	Rumipamba, Parque central	Comisaria Municipal	Infracción de ordenanza de construcción	50.00	0.50	0.50	2.00	53.00
TOTALES:						50.00	0.50	0.50	2.00	53.00

Miércoles, 15 de Abril del 2009 Directo Financiero Jefe de Rentas Tesorero Municipal 1 de 1

Menú Medidor



En este menú están todas las tareas que se pueden realizar con los trámites de venta de medidores de agua, en el departamento de Rentas. En el menú Medidor se tiene las opciones: Emitir título, Anular pago, Eliminar y Reporte.

Emitir un título de venta de medidor de agua.- Para hacer esto ir a menú Medidor > Emitir título. En la ventana que aparece se ingresa lo siguiente:

- Departamento que solicita la emisión del título.
- Datos de la persona, clic en :

- Concepto.
- Valor a pagar.
- Valor por derecho de instalación.
- Valor por otros servicios.
- Valor por servicios técnicos.

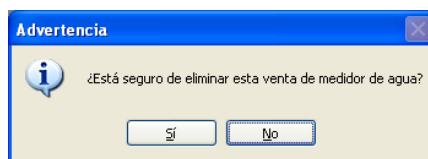
Para obtener los resultados clic en Calcular. Por último Aceptar.

Anular el pago de un título de venta de medidor de agua.- Ir a menú Medidor > Anular pago. En la ventana que aparece se debe teclear el número de título y clic en Ver:

Clic en Anular pago y confirmar:

Eliminar un título de venta de medidor de agua.- Los títulos de venta de medidor de agua que aún no han sido pagados se pueden eliminar. Para esto ir a menú Medidor > Eliminar. En la ventana que aparece se debe teclear el número de título y clic en Ver:

Clic en Eliminar. Confirmar la eliminación:

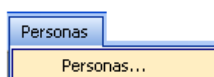


Reporte de título de venta de medidor de agua emitidos.- Para generar e imprimir reportes de los títulos de venta de medidor de agua emitidos ir a menú Medidor > Reporte. Aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

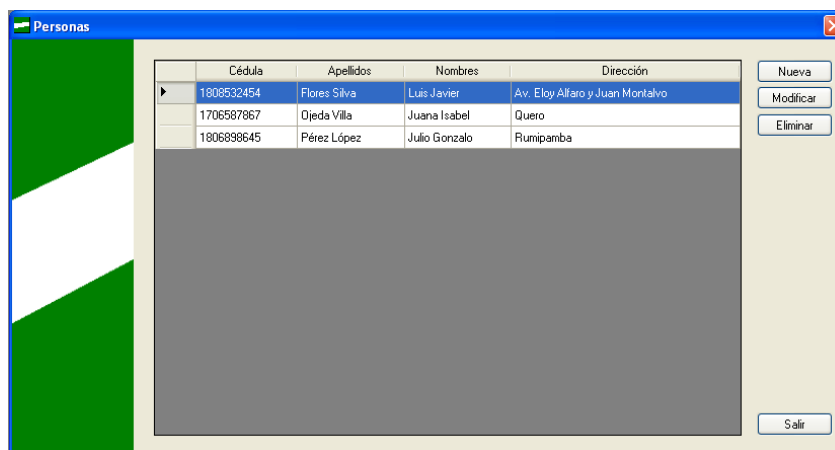
I Municipio de Quero										
Títulos de Venta de Medidor de Agua Emitidos										
Miércoles, 15 de Abril de 2009										
Título N°	F. Emisión	Nombre o razón	Dirección	Solicitado por	Concepto	V. pagar	Der. inst	Otros serv.	Serv. téc.	Total
3	15/04/2009	Flores Silva Luis Javier	Quero, Av. Eloy Alfaro y Juan Montalvo	Departamento de Agua Potable	Pago por venta de medidor de agua	30.00	15.00	5.00	0.60	50.60
TOTALES:						30.00	15.00	5.00	0.60	50.60

Miércoles, 15 de Abril del 2009 Directo Financiero Jefe de Rentas Tesorero Municipal 1 de 1

Menú Personas



En este menú está la opción Personas... que nos permitirá manejar los datos de las personas almacenados en el sistema:



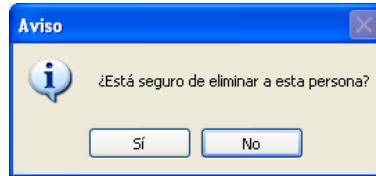
Para ingresar los datos de una nueva persona clic en el botón Nueva, y en la ventana que aparece ingresar los datos:

The dialog box "Ingresar nueva persona" contains the following fields: "Tipo" (dropdown menu set to "Particular"), "Cédula" (text box with "1705348678"), "Nombres" (text box with "Mariana Elena"), "Apellidos" (text box with "Núñez Núñez"), and "Dirección" (text box with "San Antonio"). At the bottom right are "Aceptar" and "Cancelar" buttons.

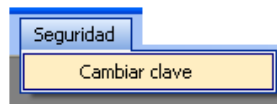
Se puede modificar los datos de las personas seleccionando el registro de la persona y haciendo clic en Modificar. Aparece una ventana con los datos de la persona, entonces se debe realizar los cambios:

The dialog box "Modificar datos de persona" contains the following fields: "Tipo" (dropdown menu set to "Particular"), "Cédula" (text box with "1808532454"), "Nombres" (text box with "Luis Javier"), "Apellidos" (text box with "Flores Silva"), and "Dirección" (empty text box). At the bottom right are "Aceptar" and "Cancelar" buttons.

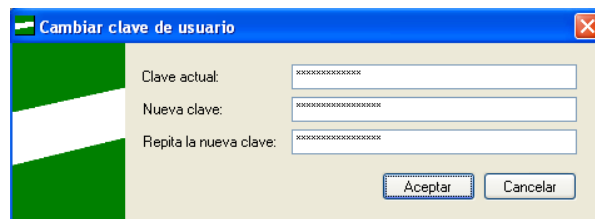
Por último, se puede eliminar los datos de una persona que no haya realizado algún trámite (caso contrario no se podrá eliminar), seleccionando el registro y haciendo clic en Eliminar, aparecerá un mensaje pidiendo una confirmación:



Menú Seguridad

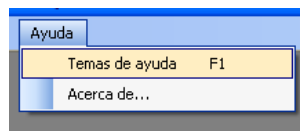


Contiene la opción Cambiar clave, que permite al usuario actual cambiar su clave:



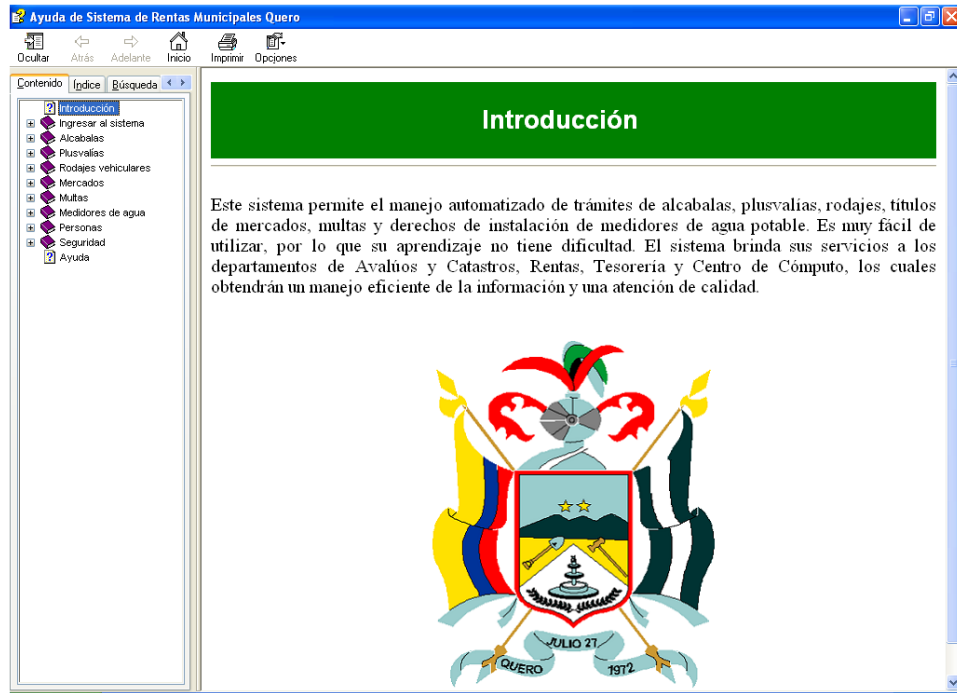
Se ingresa primero la clave actual, luego se ingresa la nueva clave, por último se repite la nueva clave, y Aceptar.

Menú Ayuda

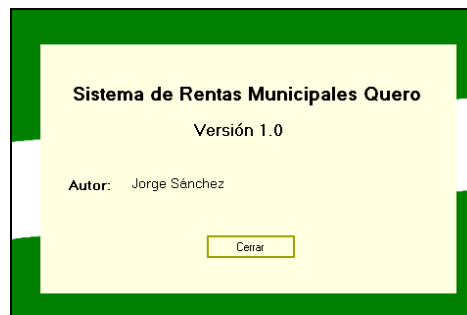


Tiene las siguientes opciones:

Temas de ayuda.- Con esta opción se muestra la ayuda del sistema:

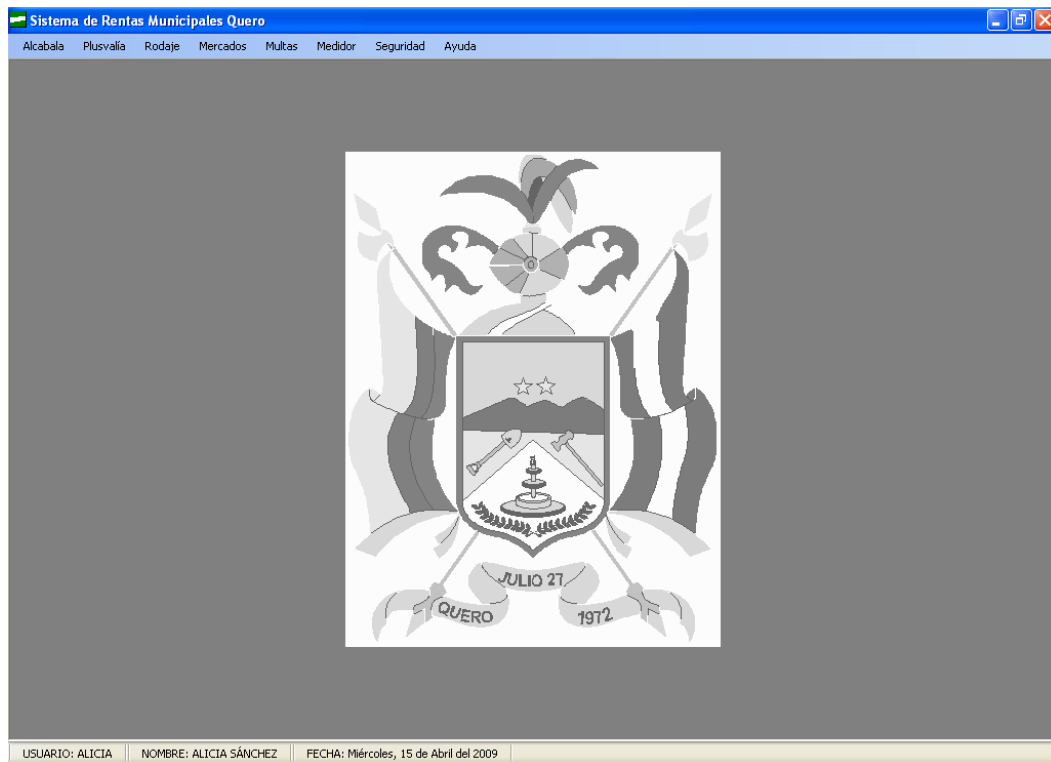


Acerca de.- Muestra la ventana Acerca de...



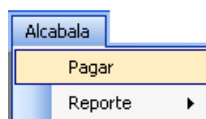
Módulo Departamento de Recaudación (Tesorería)

La ventana principal es la siguiente:



Entre las opciones de la barra de menú están: Alcabala, Plusvalía, Rodaje, Mercados, Multas, Medidor, Seguridad y Ayuda.

Menú Alcabala



En este menú están todas las tareas que se pueden realizar, en el departamento de Recaudación, con los trámites de alcabalas. En el menú Alcabala se tiene las opciones: Pagar y Reporte.

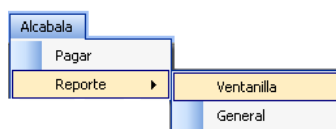
Pagar una alcabala.- Para pagar una alcabala ir a menú Alcabala > Pagar. En la ventana que aparece se teclea el número de la alcabala a pagar, y clic en Ver:

Se puede visualizar todos los datos y valores así como el valor total a pagar. Una vez que el recaudador ha cobrado el dinero debe hacer clic en Pagar. Inmediatamente se debe imprimir el comprobante de pago, por lo que aparece la ventana de impresión:

Y se obtiene el comprobante de pago impreso:

PAGO DE IMPUESTO A ALCABALA				Nº. 89	
Fecha de ingreso:	14/04/2009	Fecha de liquidación:	14/04/2009	Fecha de pago:	15/04/2009
Escritura de:	Compra/venta				
Vendedor:	Flores Silva Luis Javier	C.I. 1808532454;	Cjeda Villa Juana Isabel	C.I. 1706587867	
Comprador:	Pérez López Julio Gonzalo	C.I. 1806898645;	Núñez Núñez Mariana Elena	C.I. 1705348678	
Objeto de venta:	Lote de 1200 m2				
Ubicación:	Quero				
Valor:	20000.00				
Municipio	200.00	Observación:	Según LORM Art. 344 literal a		
Const. Escolares	2.00				
Total a pagar	202.00				

Reporte de alcabalas recaudadas.- Existen dos tipos de reporte: de ventanilla y general:



El reporte de ventanilla muestra las alcabalas recaudadas en la ventanilla actual, es decir las alcabalas recaudadas por el usuario actual. El reporte general, por el contrario, muestra la recaudación de todas las ventanillas, es decir de todos los usuarios del sistema en el módulo de recaudación.

En ambos casos aparece la ventana de reportes:



En la parte superior izquierda de la ventana se encuentra los campos Desde y Hasta en los cuales se escoge las fechas del intervalo de tiempo del cual se quiere obtener el reporte. Clic en Ver:

Nº Alcabala	F. Liquidación	F. Pago	Escritura	Vendedor	Comprador	Objeto de
69	14/04/2009	15/04/2009	Compra/venta	Flores Silva Luis Javier...	Pérez López Julio Gonzal...	Lote de 12
TOTALES						

Para imprimir el reporte clic en Imprimir.

Impresora: Epson FX-1180

Tamaño del papel: reporte

Orientación

Vertical

Horizontal

Imprimir Cancelar

Se escoge la impresora, tamaño y orientación del papel, y se imprime el reporte:

I. Municipio de Quero
Listado de Recaudación de Alcabalas
Miércoles, 15 de Abril de 2009

N° Alcabala	F. liquid.	F. pago	Escritura	Vendedor	Comprador	Objeto de venta	Ubicación	Valor	Municipio	Const. Escolares	Total
89	14/04/2009	15/04/2009	Compra/Venta	Flores Silva Luis Javier C.I. 1808532454 y otros	Pérez López Julio Gonzalo C.I. 1806988645 y otros	Lote de 1200 m2	Quero	20000.00	200.00	2.00	202.00
TOTAL RECAUDACIÓN:									200.00	2.00	202.00

Miércoles, 15 de Abril del 2009

1 de 1

Por último hay la posibilidad de obtener un resumen general de la recaudación, para ello clic en el botón Resumen... Aparece una ventana con el resumen:

Resumen

RESUMEN DE RECAUDACIÓN GENERAL DIARIO
ALCABALAS
Miércoles, 15 de Abril de 2009

Municipio:

Const. Escolares:

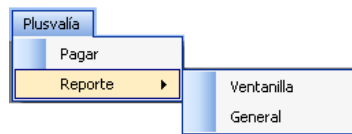
Total Recaudado:

N° de alcabalas:

Se puede imprimir el resumen:

RESUMEN DE RECAUDACIÓN GENERAL DIARIO	
ALCABALAS	
Miércoles, 15 de Abril de 2009	
Municipio	200.00
Const. Escolares	2.00
Total Recaudado	202.00
N° de alcabalas	1

Menú Plusvalía



En este menú están todas las tareas que se pueden realizar, en el departamento de Recaudación, con los trámites de plusvalía. En el menú Plusvalía se tiene las opciones: Pagar y Reporte.

Pagar una plusvalía.- Ir a menú Plusvalía > Pagar. En la ventana que aparece se teclea el número de la plusvalía a pagar, y clic en Ver:

A screenshot of a software window titled 'Pago de Plusvalía'. The window contains several input fields and calculated values. At the top, there is a field for 'Nº Plusvalía' with the value '45' and a 'Ver' button. Below this, there are fields for 'Escritura' (Compra/venta), 'Vendedor' (Ojeda Villa Juana Isabel), 'Comprador' (Núñez Núñez Mariana Elena), 'Superficie' (3500.00 metros cuadrados), and 'Ubicación' (San Vicente). There are also fields for 'C.I. Vendedor' (1706587867) and 'C.I. Comprador' (1705348678). The window is divided into sections: 'Liquidación del impuesto a la utilidad' with fields for 'Precio de venta' (10000.00), 'Año venta' (2008), 'Precio adquisición' (1000.00), 'Año adquisición' (1999), and 'Utilidad bruta' (9000.00); 'Deducciones y rebajas' with fields for 'Gastos de Escritura' (10.00), 'Contribuciones de mejoras' (5.00), 'Obras realizadas' (12.00), and 'Total de deducciones' (27.00); 'Utilidad neta' (8973.00), '5% anual tiempo transcurrido' (3140.55), 'Base para rebaja desvalorización' (5832.45), 'Valor moneda año adquisición' (0.00), 'Valor moneda año venta' (0.00), 'Diferencia' (0), 'Menos porcentaje desvalorización' (0.00), and 'Base imponible' (5832.45). At the bottom, there is a 'Total' section with 'Impuesto sobre la fracción excedente' (29.16), 'Observación' (Ninguna), and 'TOTAL A PAGAR' (29.16). There are 'Pagar' and 'Salir' buttons at the bottom right.

Se muestra todos los datos y valores, así como el valor total a pagar. Una vez que el recaudador ha cobrado el dinero debe hacer clic en Pagar. Inmediatamente se debe imprimir el comprobante de pago, por lo que aparece la ventana de impresión, y se manda a imprimir el comprobante de pago:

PAGO DE IMPUESTO A LA PLUSVALÍA					
Fecha de pago:	15/04/2009	Fecha de Liquidación:	14/04/2009	Nº:	45
Ubicación:	San Vicente				
Vendedor:	Ojeda Villa Juana Isabel	C.I. Vendedor:	1706587867		
Comprador:	Núñez Núñez Mariana Elena	C.I. Comprador:	1705348678		
Precio de Venta:	10000.00	Año Venta:	2008		
Precio de Adquisición:	1000.00	Año Adquisición:	1999		
Total a pagar	29.16				

Reporte de plusvalías recaudadas.- Existen reportes por ventanilla y general. El reporte de ventanilla muestra las plusvalías recaudadas en la ventanilla actual. El reporte general muestra la recaudación de todas las ventanillas. En ambos casos aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

I. Municipio de Ocuaro							
Listado de Recaudación de Plusvalías							
Miércoles, 15 de Abril de 2009							
Nº Plusvalía	F. Liquidación	F. Pago	Escritura	Vendedor	Comprador	Ubicación	Total
45	14/04/2009	15/04/2009	Compra/venta	Ojeda Villa Juana Isabel	Núñez Núñez Mariana Elena	San Vicente	29.16
TOTAL RECAUDACIÓN:							29.16

1 de 1

Se puede ver e imprimir un resumen general de la recaudación haciendo clic en el botón Resumen...

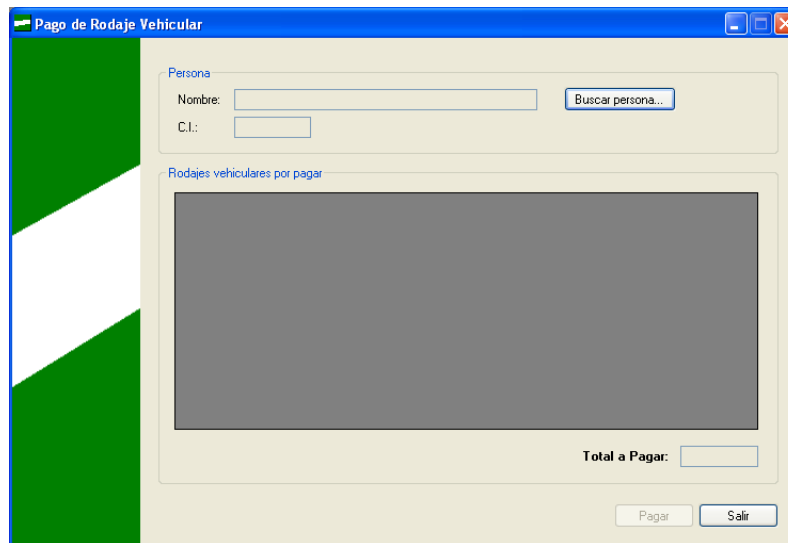
RESUMEN DE RECAUDACIÓN GENERAL DIARIO PLUSVALÍAS	
Miércoles, 15 de Abril de 2009	
Impuesto fracción excedente	29.16
Total Recaudado	29.16
Nº de plusvalías	1

Menú Rodaje

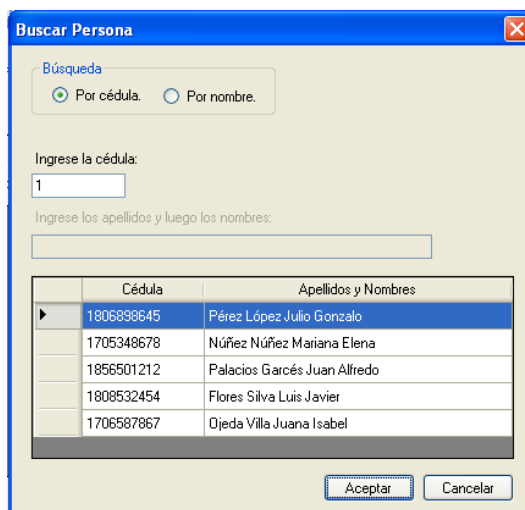


En este menú están todas las tareas que se pueden realizar, en el departamento de Recaudación, con los trámites de rodajes vehiculares. En el menú Rodaje se tiene las opciones: Pagar y Reporte.

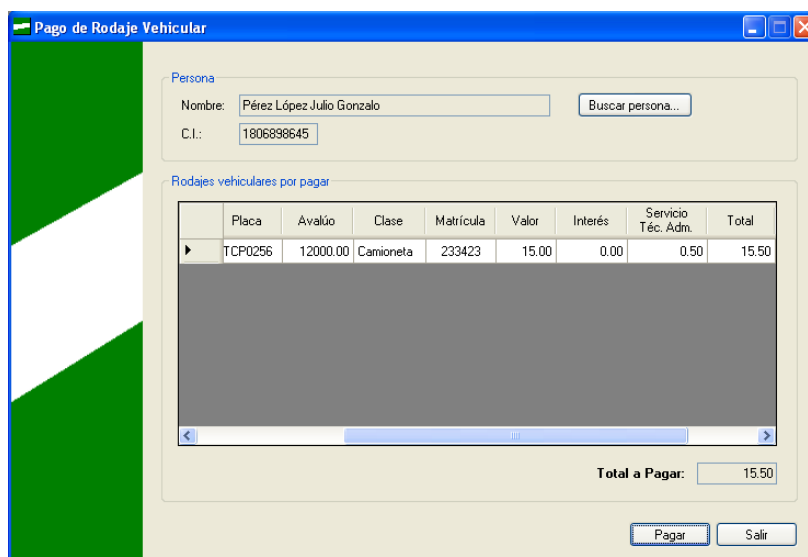
Pagar rodajes vehiculares.- Para esto ir a menú Rodaje > Pagar. Aparece la ventana:



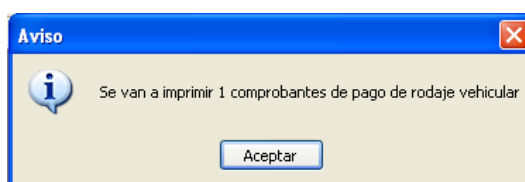
Clic en Buscar persona..., aparece la ventana de búsqueda de personas:



Elegir a la persona. Y aparecen los datos:



Se muestra el (los) rodaje(s) con sus respectivos datos y valores, así como el valor total a pagar. Una vez que el recaudador ha cobrado el dinero debe hacer clic en Pagar. Inmediatamente se debe imprimir el (los) comprobante(s) de pago, por lo que aparece un mensaje advirtiendo cuántos comprobantes se van a imprimir:



Y luego aparece la ventana de impresión, y se manda a imprimir el (los) comprobante(s) de pago:

PAGO DE IMPUESTO AL RODAJE VEHICULAR		
Fecha de pago:	15/04/2009	Fecha de emisión: 14/04/2009
		Nº: 4
Año:	2009	
Nombre:	Pérez López Julio Gonzalo	C.I.: 1806898645
Placa:	TCP0256	
Avalúo vehículo:	12000.00	
Clase:	Camioneta	
Mat. Ant. Nº:	233423	
Valor	15.00	
Interés	0.00	
Servicio Téc. Adm.	0.50	
Total a pagar	15.50	

Reporte de rodajes vehiculares recaudados.- Hay reportes por ventanilla y general. El reporte de ventanilla muestra los rodajes recaudados en la ventanilla actual. El reporte general muestra la recaudación de todas las ventanillas. En ambos casos aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

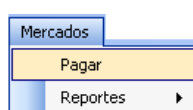
I. Municipio de Quero													
Listado de Recaudación de Rodajes Vehiculares													
Miércoles, 15 de Abril de 2009													
Título Nº	F. Liq.	F. Pag.	Nombre	Cédula	Año	Placa	Avalúo	Clase	Matricula	Valor	Interés	Serv. Téc.	Total
4	14/04/2009	15/04/2009	Pérez López Julio Gonzalo	1806898645	2009	TCP0256	12000.00	Camioneta	233423	15.00	0.00	0.50	15.50
TOTAL RECAUDACIÓN:										15.00	0.00	0.50	15.50

Miércoles, 15 de Abril del 2009 1 de 1

Se puede ver e imprimir un resumen general de la recaudación haciendo clic en el botón Resumen...

RESUMEN DE RECAUDACIÓN GENERAL DIARIO RODAJES Miércoles, 15 de Abril de 2009	
Valor	15.00
Interés	0.00
Servicio Téc. Adm.	0.50
Total Recaudación	15.50
Nº de rodajes	1

Menú Mercados



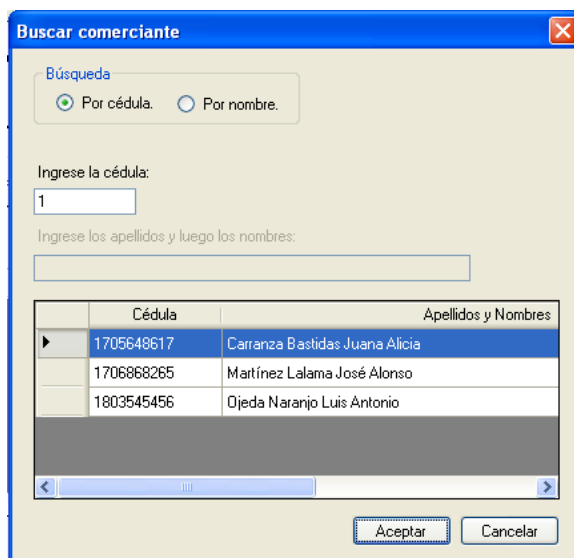
En este menú están todas las tareas que se pueden realizar, en el departamento de Recaudación, con los trámites de mercados. En el menú Mercados se tiene las opciones: Pagar y Reportes.

Pagar un título de mercado.- Para esto ir a menú Mercados > Pagar. Aparece la ventana:

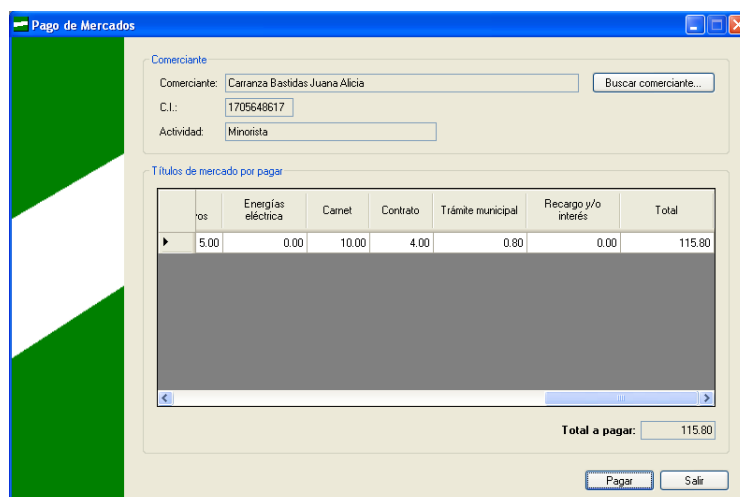
 A screenshot of a software window titled 'Pago de Mercados'. The window has a blue title bar and a green and white striped sidebar on the left. The main area contains:

- A section labeled 'Comerciante' with three input fields: 'Comerciante:', 'C.I.:', and 'Actividad:'. A 'Buscar comerciante...' button is to the right of the first field.
- A section labeled 'Títulos de mercado por pagar' with a large, empty grey rectangular area.
- A 'Total a pagar:' label followed by an input field.
- At the bottom right, there are two buttons: 'Pagar' and 'Salir'.

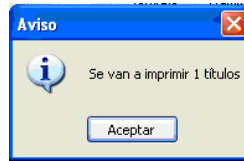
Clic en Buscar comerciante..., aparece la ventana de búsqueda de comerciantes:



Elegir al comerciante. Y aparecen los datos:



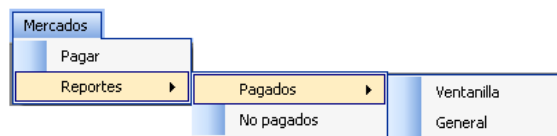
Se muestra todo lo que debe pagar el comerciante con sus respectivos datos y valores, así como el valor total a pagar. Una vez que el recaudador ha cobrado el dinero debe hacer clic en Pagar. Inmediatamente se debe imprimir el (los) título(s), por lo que aparece un mensaje advirtiendo cuántos títulos se van a imprimir:



Y luego aparece la ventana de impresión, y se manda a imprimir el (los) título(s):

TÍTULO DE CRÉDITO MERCADOS					
Fecha de pago:	15/04/2009	Fecha de emisión:	14/04/2009	Titulo Nº: 3	
Solicitado por:	Administración de Mercados			C.I.: 1705648617	
Comerciante:	Carranza Bastidas Juana Alicia				
Actividad:	Minorista				
Nº Puesto:	120				
Dirección:	Av. 17 de Abril y Eloy Alfaro				
Desde mes:	Enero	Hasta mes:	Diciembre		
Año:	2009				
Mensualidad	60.00	Carnet	10.00	Subtotales 1 + 2	115.80
Activo total	25.00	Contrato	4.00	Recargo y/o intereses	0.00
Rec. de basura y aseo p.	6.00	Trámite municipal	0.80	Total a pagar	115.80
Aferición de pesas y medidas	5.00	Subtotal 2	14.80		
Servicios administrativos	5.00				
Energía eléctrica	0.00				
Subtotal 1	101.00				

Reporte de títulos de mercado.- Se puede obtener reportes de los títulos que han sido pagados y de los no han sido pagados aún:



Para los pagados se dispone de reportes por ventanilla y general. El reporte de ventanilla muestra los títulos de mercado recaudados en la ventanilla actual. El reporte general muestra la recaudación de todas las ventanillas. En ambos casos aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

I. Municipio de Quero
Listado de Recaudación de Mercados
Miércoles, 15 de Abril de 2009

Título Nº	F. Emisión	F. Pago	Comerciante	Desde	Hasta	Año	Mensualidad	Act. Total	Rec. Basura	Pesas y med.	Serv. Adm.	Energ. Elec.	Carnet	Contrato	Tram. Munic.	Recarg/inter	Total
3	14042009	15042009	Carranza Bastidas Juana Alicia	Enero	Diciembre	2009	60.00	25.00	6.00	5.00	5.00	0.00	10.00	4.00	0.80	0.00	115.80
TOTAL RECAUDACIÓN:							60.00	25.00	6.00	5.00	5.00	0.00	10.00	4.00	0.80	0.00	115.80

Miércoles, 15 de Abril del 2009

1 de 1

Se puede ver e imprimir un resumen general de la recaudación haciendo clic en el botón Resumen...

RESUMEN DE RECAUDACIÓN GENERAL DIARIO
MERCADOS
Miércoles, 15 de Abril de 2009

Mensualidad	60.00
Activo total	25.00
Rec. de basura y aseo p	6.00
Aferición de pesas y medidas	5.00
Servicios administrativos	5.00
Energía eléctrica	0.00
Carnet	10.00
Contrato	4.00
Trámite municipal	0.80
Recargo y/o intereses	0.00
Total Recaudado	115.80
Nº de títulos	1

Para obtener un reporte de los títulos no pagados, también se determina el lapso de tiempo y se visualiza e imprime el reporte:

I. Municipio de Quero
Lista de Títulos de Mercado No Pagados
Del 14 de Abril del 2009 al 15 de Abril del 2009

Título Nº	F. Emisión	Comerciante	Dirección	Desde	Hasta	Año	Mensualidad	Act. Total	Rec. Basura	Pesas y med.	Serv. Adm.	Energ. Elec.	Carnet	Contrato	Tram. Munic.	Recarg/inter	Total
3	14042009	Carranza Bastidas Juana Alicia	Av. 11 de Abril y Eloy Alfaro	Enero	Diciembre	2009	60.00	25.00	6.00	5.00	5.00	0.00	10.00	4.00	0.80	0.00	115.80
TOTAL ES:							60.00	25.00	6.00	5.00	5.00	0.00	10.00	4.00	0.80	0.00	115.80

Miércoles, 15 de Abril del 2009

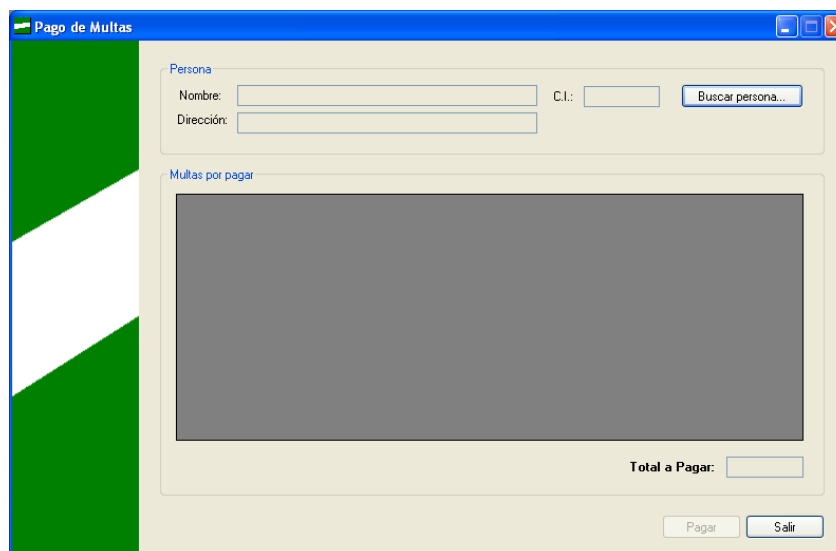
1 de 1

Menú Multas

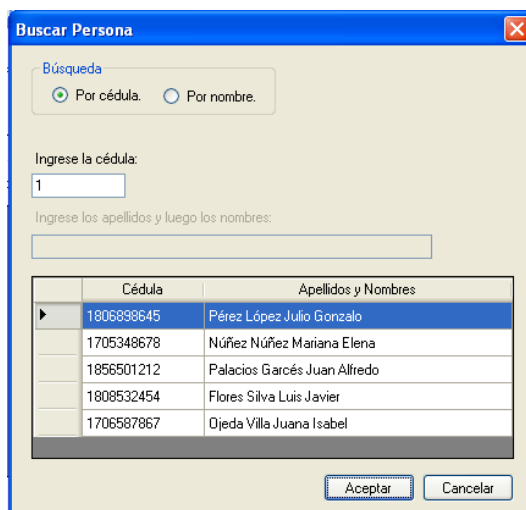


En este menú están todas las tareas que se pueden realizar, en el departamento de Recaudación, con las multas. En el menú Multas se tiene las opciones: Pagar y Reportes.

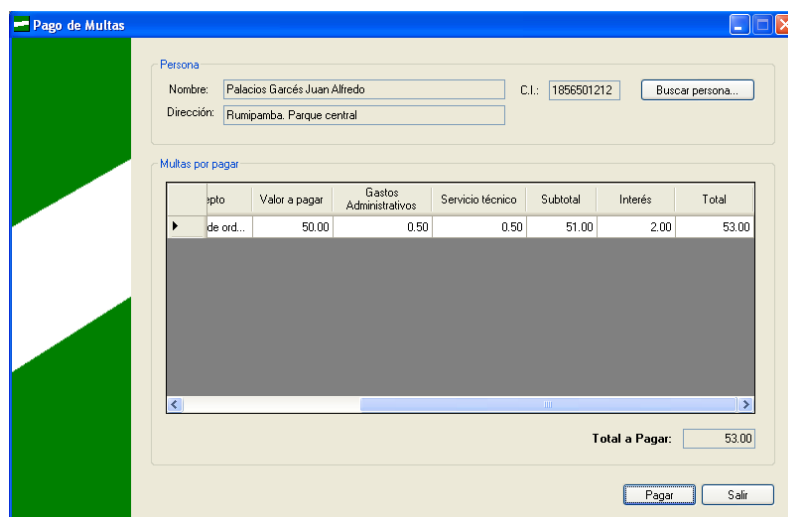
Pagar una multa.- Para esto ir a menú Multas > Pagar. Aparece la ventana:

A screenshot of a software window titled 'Pago de Multas'. The window has a blue title bar and a green and white striped sidebar on the left. The main area is divided into two sections. The top section is labeled 'Persona' and contains three input fields: 'Nombre:', 'Dirección:', and 'C.I.:'. To the right of the 'C.I.' field is a button labeled 'Buscar persona...'. The bottom section is labeled 'Multas por pagar' and contains a large, empty gray rectangular area. At the bottom right of this section is a label 'Total a Pagar:' followed by an empty input field. At the very bottom of the window are two buttons: 'Pagar' and 'Salir'.

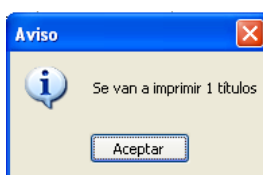
Clic en Buscar persona..., aparece la ventana de búsqueda de personas:



Elegir a la persona. Y aparecen los datos:



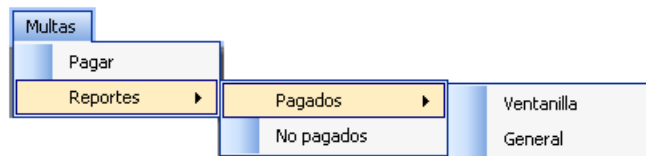
Se muestra la (las) multa(s) con sus respectivos datos y valores, así como el valor total a pagar. Una vez que el recaudador ha cobrado el dinero debe hacer clic en Pagar. Inmediatamente se debe imprimir el (los) título(s), por lo que aparece un mensaje advirtiendo cuántos títulos se van a imprimir:



Y luego aparece la ventana de impresión, y se manda a imprimir el (los) título(s):

TÍTULO DE CRÉDITO DE MULTA			
Fecha de pago:	15/04/2009	Fecha de emisión:	15/04/2009
		Título N°:	3
Solicitado por:	Comisaría Municipal		
Nombre o razón:	Palacios Garcés Juan Alfredo	C.I.:	1858501212
Dirección:	Rumipamba, Parque central		
Concepto:	Infracción de ordenanza de construcción		
Valor a pagar	50.00	Subtotal	51.00
Gastos Adminis.	0.50	Interés	2.00
Servicio Técnico	0.50	Total a pagar	53.00
Subtotal	51.00		

Reporte de multas.- Se puede obtener reportes de las multas que han sido pagadas y de las que no han sido pagadas aún:



Para las multas pagadas se dispone de reportes por ventanilla y general. El reporte de ventanilla muestra las multas recaudadas en la ventanilla actual. El reporte general muestra la recaudación de todas las ventanillas. En ambos casos aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

I. Municipio de Quero										
Listado de Recaudación de Multas										
Miércoles, 15 de Abril de 2009										
Título N°	F. emisión	F. pago	Nombre o razón	Solicitado por	Concepto	Valor a pagar	Gastos Adm.	Serv. Téc.	Interés	Total
3	15/04/2009	15/04/2009	Palacios Garcés Juan Alfredo	Comisaría Municipal	Infracción de ordenanza de construcción	50.00	0.50	0.50	2.00	53.00
TOTAL RECAUDACIÓN:						50.00	0.50	0.50	2.00	53.00

Miércoles, 15 de Abril del 2009 1 de 1

Para obtener un resumen general de la recaudación, clic en Resumen...

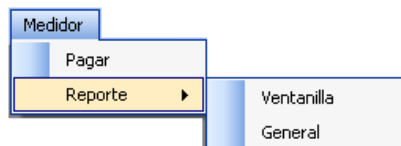
RESUMEN DE RECAUDACIÓN GENERAL DIARIO	
MULTAS	
Miércoles, 15 de Abril de 2009	
Valor a pagar	50.00
Gastos Adminis.	0.50
Servicio Técnico	0.50
Interés	2.00
Total Recaudado	53.00
Nº de títulos	1

Para las multas no pagadas también se determina el lapso de tiempo y se visualiza e imprime el reporte:

I. Municipio de Quero										
Lista de Multas No Pagadas										
Miércoles, 15 de Abril de 2009										
Título Nº	F. emisión	Nombre o razón	Dirección	Solicitado por	Concepto	V. pagar	Gastos Adm.	Serv. téc.	Interés	Total
3	15/04/2009	Palacios Carces Juan Alfredo	Rumipamba, Parque central	Comisaria Municipal	Infracción de ordenanza de construcción	30.00	0.50	0.50	2.00	53.00
TOTALES:						50.00	0.50	0.50	2.00	53.00

Miércoles, 15 de Abril del 2009 1 de 1

Menú Medidor



En este menú están todas las tareas que se pueden realizar, en el departamento de Recaudación, con los trámites de venta de medidores de agua. En el menú Medidor se tiene las opciones: Pagar y Reporte.

Pagar un título de venta de medidor de agua.- Ir a menú Medidor > Pagar.
Aparece la ventana:

Persona

Nombre: C.I.:

Dirección:

Ventas de medidor por pagar

--	--

Total a Pagar:

Clic en **Buscar persona...**, aparece la ventana de búsqueda de personas:

Búsqueda

Por cédula. Por nombre.

Ingrese la cédula:

Ingrese los apellidos y luego los nombres:

Cédula	Apellidos y Nombres
1806898645	Pérez López Julio Gonzalo
1705348678	Núñez Núñez Mariana Elena
1856501212	Palacios Garcés Juan Alfredo
1808532454	Flores Silva Luis Javier
1706587867	Ojeda Villa Juana Isabel

Elegir a la persona. Y aparecen los datos:

Pago de Venta de Medidor de Agua

Persona

Nombre: Flores Silva Luis Javier C.I.: 1808532454

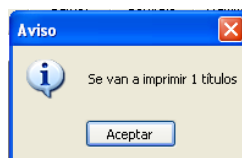
Dirección: Quero, Av. Eloy Alfaro y Juan Montalvo

Ventas de medidor por pagar

Concepto	Valor a pagar	Derecho de Instalación	Otros Servicios	Servicio Técnico	Total
venta d...	30.00	15.00	5.00	0.60	50.60

Total a Pagar: 50.60

Se muestra el (los) título(s) que debe pagar la persona con sus respectivos datos y valores, así como el valor total a pagar. Una vez que el recaudador ha cobrado el dinero debe hacer clic en Pagar. Inmediatamente se debe imprimir el (los) título(s), por lo que aparece un mensaje advirtiendo cuántos títulos se van a imprimir:



Y luego aparece la ventana de impresión, y se manda a imprimir el (los) título(s):

TÍTULO DE CRÉDITO PAGO VENTA MEDIDOR			
Fecha de pago:	15/04/2009	Fecha de emisión:	15/04/2009
		Título N°:	3
Solicitado por:	Departamento de Agua Potable		
Nombre o razón:	Flores Silva Luis Javier	C.I.:	1808532454
Dirección:	Quero. Av. Eloy Alfaro y Juan Montalvo		
Concepto:	Pago por venta de medidor de agua		
		Subtotal	45.00
Valor a pagar	30.00	Servicio Técnico	0.60
Derecho Instalación	15.00	Otros Servicios	5.00
Subtotal	45.00	Total a pagar	50.60

Reporte de títulos de venta de medidor de agua.- Existen reportes por ventanilla y general. El reporte de ventanilla muestra los títulos recaudados en la ventanilla actual. El reporte general muestra la recaudación de todas las ventanillas. En ambos casos aparece la ventana de reportes, se determina las fechas, se visualiza e imprime el reporte:

I. Municipio de Quero											
Listado de Recaudación de Medidores de Agua											
Miércoles, 15 de Abril de 2009											
Título N°	F. emisión	F. pago	Nombre o razón	Dirección	Solicitado por	Concepto	V. pagar	Der. inst.	Otros serv.	Serv. téc.	Total
3	15/04/2009	15/04/2009	Flores Silva Luis Javier	Quero. Av. Eloy Alfaro y Juan Montalvo	Departamento de Agua Potable	Pago por venta de medidor de agua	30.00	15.00	5.00	0.60	50.60
TOTAL RECAUDACIÓN:							30.00	15.00	5.00	0.60	50.60

Miércoles, 15 de Abril del 2009 1 de 1

Se puede ver e imprimir un resumen general de la recaudación:

RESUMEN DE RECAUDACIÓN GENERAL DIARIO MEDIDORES DE AGUA Miércoles, 15 de Abril de 2009	
Valor a pagar	30.00
Derecho Instalación	15.00
Servicio Técnico	0.60
Otros Servicios	5.00
Total Recaudado	50.60
Nº de títulos	1

Menú Seguridad

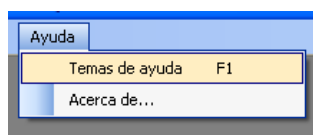


Contiene la opción Cambiar clave, que permite al usuario actual cambiar su clave:

 A screenshot of a dialog box titled 'Cambiar clave de usuario'. It contains three input fields: 'Clave actual:', 'Nueva clave:', and 'Repita la nueva clave:'. Each field is filled with a series of asterisks. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

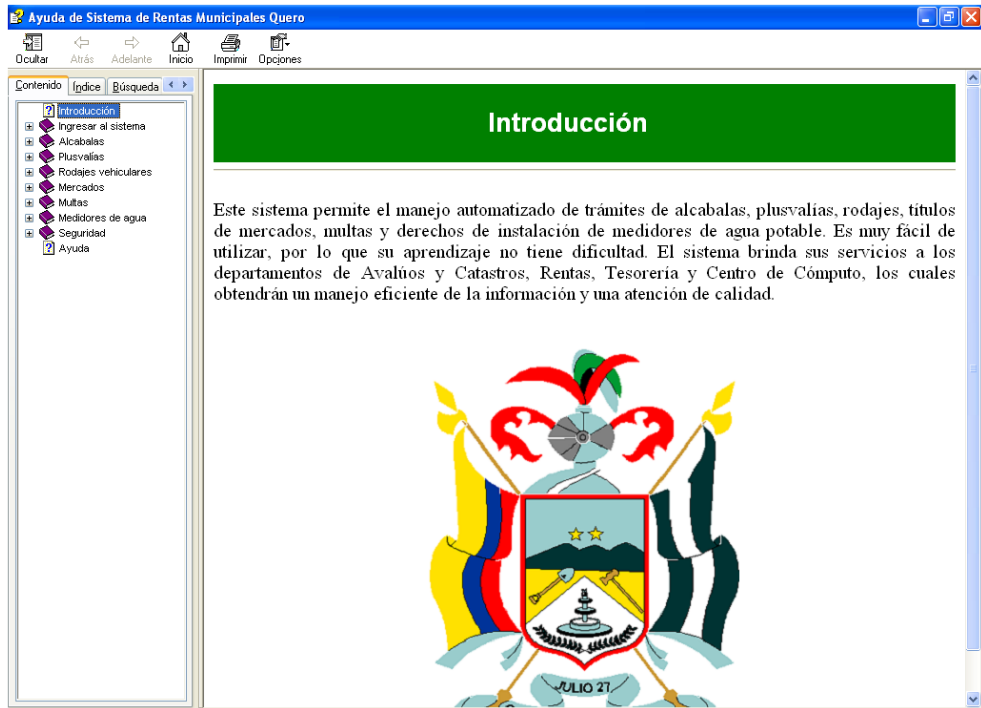
Se ingresa primero la clave actual, luego se ingresa la nueva clave, por último se repite la nueva clave, y Aceptar.

Menú Ayuda

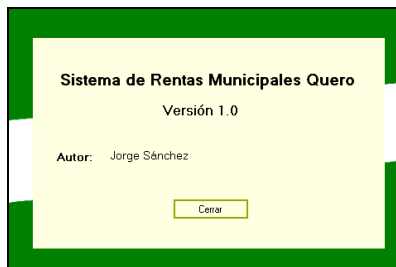


Tiene las siguientes opciones:

Temas de ayuda.- Con esta opción se muestra la ayuda del sistema:

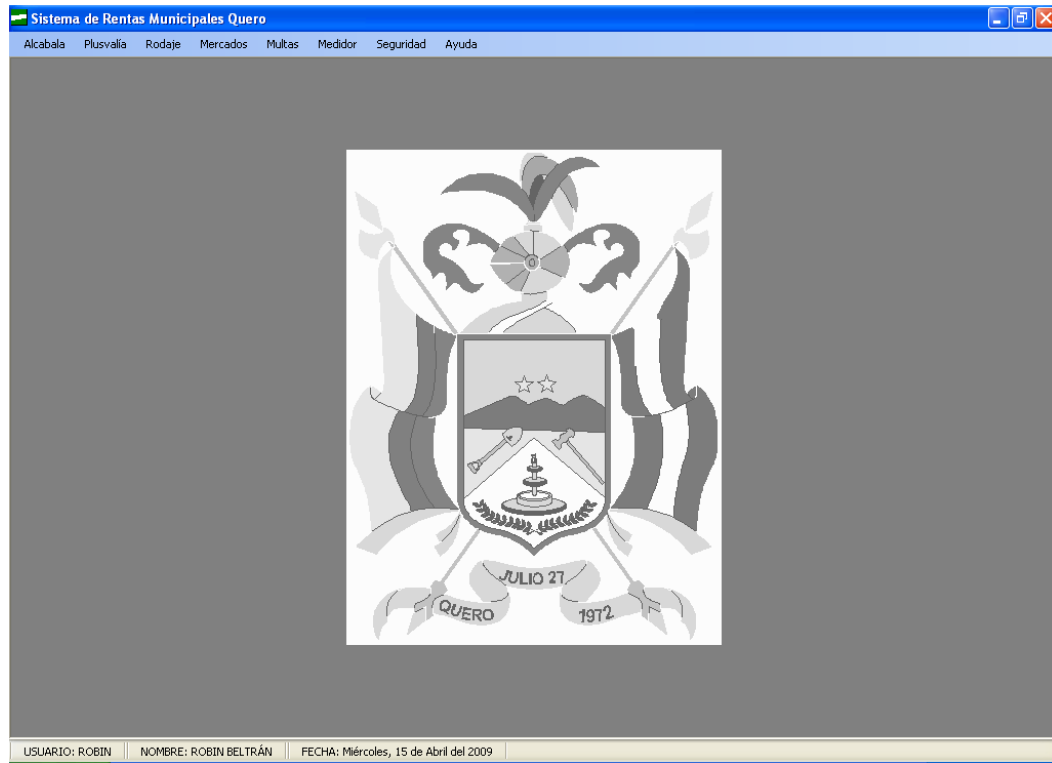


Acerca de.- Muestra la ventana Acerca de...



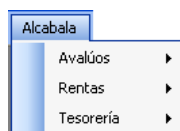
Módulo de Administración (Centro de Cómputo)

La ventana principal es la siguiente:



Entre las opciones de la barra de menú están: Alcabala, Plusvalía, Rodaje, Mercados, Multas y Medidor; estas opciones permiten obtener reportes de los distintos trámites que se han realizado en los departamentos de Avalúos y Catastros, Rentas y Tesorería. Están también las opciones Seguridad y Ayuda.

Menú Alcabala



En este menú están todas las tareas que puede realizar el administrador del sistema con los trámites de alcabalas. En el menú Alcabala se tiene las opciones: Avalúos, Rentas y Tesorería.

Avalúos.- La opción Alcabala > Avalúos > Reporte, permite obtener reportes de las alcabalas que se han ingresado en el Dpto. de Avalúos y Catastros.

Rentas.- La opción Alcabala > Rentas > Reporte, permite obtener reportes de las alcabalas que se han liquidado en el Dpto. de Rentas.

Tesorería.- La opción Alcabala > Tesorería > Reporte, permite obtener reportes de las alcabalas que se han recaudado en Tesorería o Recaudación.

Menú Plusvalía

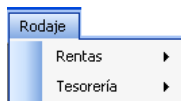


En este menú están todas las tareas que puede realizar el administrador del sistema con los trámites de plusvalía. En el menú Plusvalía se tiene las opciones: Rentas y Tesorería.

Rentas.- La opción Plusvalía > Rentas > Reporte, permite obtener reportes de las plusvalías que se han liquidado en el Dpto. de Rentas.

Tesorería.- La opción Plusvalía > Tesorería > Reporte, permite obtener reportes de las plusvalías que se han recaudado en Tesorería.

Menú Rodaje



En este menú están todas las tareas que pueden realizar el administrador del sistema con los trámites de rodajes vehiculares. En el menú Rodaje se tiene las opciones: Rentas y Tesorería.

Rentas.- La opción Rodaje > Rentas > Reporte, permite obtener reportes de los rodajes que se han liquidado en el Dpto. de Rentas.

Tesorería.- La opción Rodaje > Tesorería > Reporte, permite obtener reportes de los rodajes que se han recaudado en Tesorería.

Menú Mercados

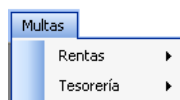


En este menú están todas las tareas que puede realizar el administrador del sistema con los títulos de mercado. En el menú Mercados se tiene las opciones: Rentas y Tesorería.

Rentas.- La opción Mercados > Rentas > Reporte, permite obtener reportes de los títulos de mercado que se han emitido en el Dpto. de Rentas.

Tesorería.- La opción Mercados > Tesorería > Reporte, permite obtener reportes de los títulos de mercados que se han recaudado en Tesorería.

Menú Multas

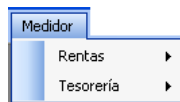


En este menú están todas las tareas que puede realizar el administrador del sistema con los trámites de multas. En el menú Multas se tiene las opciones: Rentas y Tesorería.

Rentas.- La opción Multas > Rentas > Reporte, permite obtener reportes de las multas que se han emitido en el Dpto. de Rentas.

Tesorería.- La opción Multas > Tesorería > Reporte, permite obtener reportes de las multas que se han recaudado en Tesorería.

Menú Medidor

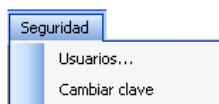


En este menú están todas las tareas que pueden realizar el administrador del sistema con los títulos de venta de medidor de agua. En el menú Medidor se tiene las opciones: Rentas y Tesorería.

Rentas.- La opción Medidor > Rentas > Reporte, permite obtener reportes de los títulos de venta de medidor de agua que se han emitido en el Dpto. de Rentas.

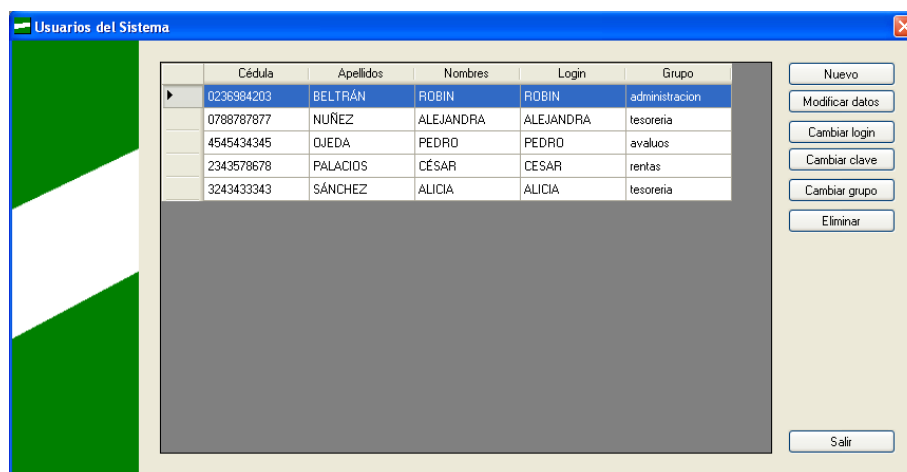
Tesorería.- La opción Medidor > Tesorería > Reporte, permite obtener reportes de los títulos de venta de medidor de agua que se han recaudado en Tesorería.

Menú Seguridad



Contiene las opciones Usuarios y Cambiar clave.

Usuarios.- Esta es la opción más importante de todo el módulo puesto que permite administrar a todos los usuarios que acceden al sistema. Para ello ir a menú Seguridad > Usuarios; aparece la ventana con el listado de todos los usuarios del sistema:



Se puede crear nuevos usuarios, hacer modificaciones y eliminar usuarios:

Para crear un nuevo usuario del sistema clic en Nuevo. Aparece la ventana para ingreso de usuarios. Allí se debe ingresar los datos siguientes:

- Cédula de identidad del nuevo usuario.
- Nombres y apellidos del usuario.
- El "login" o nombre de usuario y la clave. Estos dos datos se pedirán al usuario cada vez que deseen ingresar al sistema.
- Grupo de usuarios al que pertenecerá el nuevo usuario, este puede ser "avaluos", "rentas", "tesoreria" o "administracion", según la función que deba cumplir.

The dialog box titled "Ingresar nuevo usuario" contains the following fields and controls:

- Cédula: 1700425867
- Nombres: MARIANA
- Apellidos: FLORES
- Login: MARIANA
- Clave: *****
- Grupo: tesoreria (dropdown menu)
- Buttons: Aceptar, Cancelar

Para modificar los datos personales del usuario se debe seleccionar al usuario y clic en Modificar datos. En la ventana que aparece se realizan las modificaciones.

Para modificar el “login” o nombre de usuario, seleccionar el usuario y clic en Cambiar login:

The dialog box titled "Cambiar login" contains the following fields and controls:

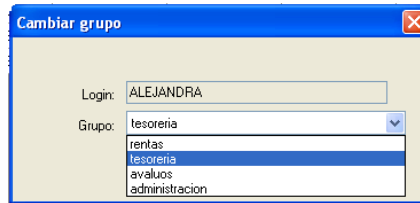
- Login actual: ALEJANDRA
- Login nuevo: (empty text box)
- Clave: (empty text box)
- Buttons: Aceptar, Cancelar

Para cambiar la clave, seleccionar el usuario y clic en Cambiar clave:

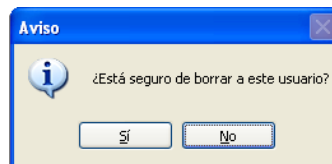
The dialog box titled "Cambiar clave" contains the following fields and controls:

- Login: ALEJANDRA
- Nueva clave: *****
- Repita la clave: *****
- Buttons: Aceptar, Cancelar

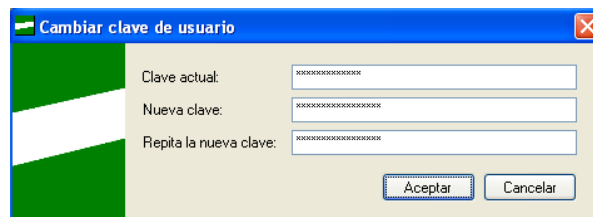
Para cambiar el grupo de usuarios al que pertenece, seleccionar el usuario y clic en Cambiar grupo:



Para eliminar a un usuario, seleccionar el usuario y clic en Eliminar. Se debe confirmar la eliminación:

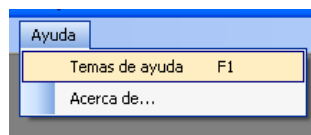


Cambiar clave.- La opción Seguridad > Cambiar clave, permite al usuario administrador cambiar su clave:



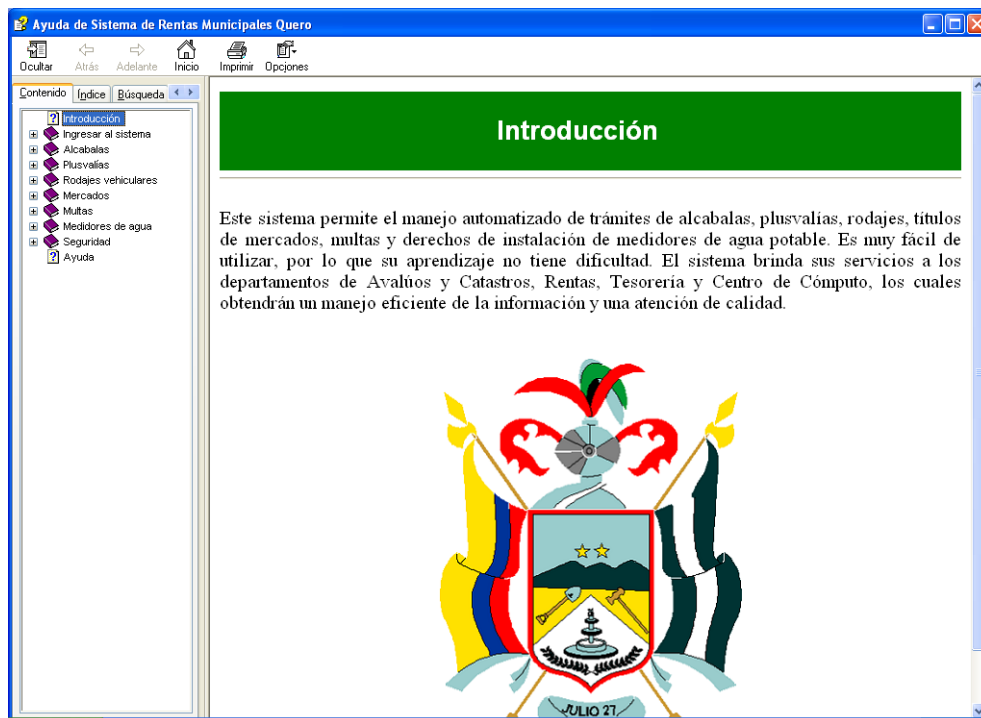
Se ingresa primero la clave actual, luego se ingresa la nueva clave, por último se repite la nueva clave, y Aceptar.

Menú Ayuda



Tiene las siguientes opciones:

Temas de ayuda.- Con esta opción se muestra la ayuda del sistema:



Acerca de.- Muestra la ventana Acerca de...

