

ESCUELA POLITÉCNICA DEL EJÉRCITO



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

Proyecto Previo a la Obtención del título de Ingeniero en Sistemas

TEMA:

**SISTEMA WEB DISTRIBUIDO PARA UNA APLICACIÓN DE CONTROL Y
ADMINISTRACIÓN DE BIENES DE LARGA DURACIÓN Y BIENES SUJETOS DE
CONTROL PARA EL DISTRITO METROPOLITANO DE QUITO**

AUTOR:

Rolando Xavier Orozco Alarcón

CERTIFICACION

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. ROLANDO XAVIER OROZCO ALARCON como requerimiento parcial a la obtención del título de INGENIEROS EN SISTEMAS E INFORMÁTICA.

Sangolquí, 11 de Septiembre del 2009.

Ing. Rolando Reyes.

DEDICATORIA

Quiero dedicar la elaboración de esta tesis con todo el cariño del mundo a mis padres Ing. Julio Cesar Orozco Lupera y Sra. Laura Natividad Alarcón Torres que me brindaron su apoyo incondicional para el logro de mi formación profesional.

A mi mejor amigo y hermano mayor Ing. Julio Fernando Orozco Alarcón y mis queridas hermanas Ing. Katia Alexandra Orozco Alarcón e Ing. Sandra Elizabeth Orozco Alarcón que gracias a su ejemplo y logros formaron en mí un hombre de bien tanto en lo personal como en lo profesional.

A mi tía Victoria quien desde el cielo ha sido mi sustento en momentos difíciles y guía para salir adelante.

A mi enamorada por estar siempre dispuesta a escucharme y aconsejarme en lo que necesite para ser una persona mejor.

Rolando Orozco A.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas aquellas personas que de una forma u otra intervinieron en la elaboración de esta Tesis, en especial al Municipio del Distrito Metropolitano de Quito por las facilidades que brindaron en el desarrollo del presente trabajo.

A mis familiares, amigos y profesores por su lealtad y ánimos.

Un agradecimiento especial a mi Director de Tesis Ing. Rolando Reyes así como a mí Co-Directora Ing. Sonia Cárdenas que en todo momento supieron guiarme e incentivarne a un feliz término en mis estudios.

Finalmente quiero agradecer a Dios por brindarme salud para poder seguir alcanzando mis metas e ideales en el transcurso de mi vida.

Rolando Orozco A.

RESUMEN

El presente proyecto consiste en llevar a cabo el desarrollo de una aplicación web denominada "SACBIM", cuyo objetivo primordial es la administración y control de los Bienes Municipales con el fin de obtener información global y verídica de todos sus activos, de tal forma que cubra con la necesidad que existe de automatizar los procesos que intervienen en el desarrollo del sistema tales como: Depreciación, Erogación, Transferencia de Bienes, Movimientos Temporales, Operaciones de un Bien, Históricos entre otros. SACBIM facilita el trabajo al usuario mediante una interfaz amigable ya que cuenta con controles personalizados del MDMQ los cuales brindan facilidad en el uso del sistema. En el entorno del sistema el usuario es capaz de crear sus propios códigos de barra según las últimas reformas en las Ordenanzas y Resoluciones respecto al control físico y contable de Bienes de Larga Duración en Entidades Públicas. SACBIM es un sistema web distribuido, desarrollado utilizando una metodología ágil denominada MSF Agile (Microsoft Solution Framework), la cual permite que el proyecto sea desarrollado en un corto tiempo y además incentiva a que el usuario interactúe constantemente en el desarrollo del sistema. Para el desarrollo del sistema se emplea la herramienta de programación Microsoft Visual Studio .NET 2005 y Oracle 10 g. como repositorio de datos, además se aplican conceptos como programación en capas, web services y DAL (Data Access Layer) para interactuar con la base de datos, y que en conjunto con el Sistema de Seguridades, Personas y Auditoria adquirido por el MDMQ contribuyen en la elaboración de un sistema robusto y de calidad.

INDICE DE CONTENIDO:

CAPITULO I INTRODUCCIÓN	10
1.1. ANTECEDENTES	10
1.2. PLANTEAMIENTO DEL PROBLEMA	11
1.3. JUSTIFICACIÓN E IMPORTANCIA	11
1.4. OBJETIVOS	12
1.4.1. OBJETIVO GENERAL.....	12
1.4.2. OBJETIVOS ESPECÍFICOS.....	13
1.5. ALCANCE.....	13
CAPITULO II MARCO TEÓRICO	15
2.1. TECNOLOGÍA	15
2.1.1. JAVASCRIPT Y VBSCRIPT	15
2.1.2. .NET FRAMEWORK.....	18
2.2. INGENIERÍA DE SOFTWARE	21
2.2.1. METODOLOGÍA	21
2.3. HERRAMIENTAS	26
2.3.1. HERRAMIENTAS DE MODELADO.....	26
2.3.2. HERRAMIENTAS DE PROGRAMACIÓN.....	28
2.4. MOTOR DE BASE DE DATOS.....	40
2.4.1. ORACLE ENTERPRISE 10 G.....	40
2.5. CONTROLES DEL PROGRAMA.....	41
2.5.1. UCFORMATOFECHA.....	41
2.5.2. UCMENSAJE	42
2.5.3. CALENDARIOWEB.....	42
2.5.4. UCDROPDOWNLISTENABLED.....	43
2.5.5. CATALOGOWEB	44
CAPITULO III. VISIONADO Y DISEÑO	46
3.1. VISION	46
3.1.1. FACTORES DE INCIDENCIA.....	46
3.1.2. VISIÓN DEL PROYECTO.....	46
3.2. DEFINICION DE PERSONAS	47
3.3. ARQUITECTURA	47
3.3.1. DESCRIPCIÓN DEL PROBLEMA.....	47
3.3.2. REQUERIMIENTOS Y ALCANCE DEL PROYECTO.....	48
3.3.3. ESTRATEGIA DE VERSIONAMIENTO O POR ETAPAS.....	54
3.3.4. CRITERIO DE ACEPTACIÓN.....	55
3.3.5. RIESGOS	56
3.3.6. ESTRUCTURA DEL PROYECTO.....	65
3.3.7. DISEÑO DE LA SOLUCIÓN.....	66
3.3.8. CONFIGURACIÓN DE HARDWARE Y SOFTWARE.....	84
3.3.9. MODELO DE BASE DE DATOS.....	89
3.3.10. REPORTES.....	101
3.3.11. EJECUCIÓN DE PRUEBAS.....	102

3.3.12.	CRONOGRAMA DE ACTIVIDADES.....	103
3.3.13.	DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS.....	104
CAPITULO IV. DESARROLLO Y PRUEBAS.....		105
4.1.	ES CENARIOS.....	105
4.1.1.	ADMINISTRACIÓN DE PARÁMETROS.....	105
4.1.2.	ADMINISTRACIÓN DE CUENTAS.....	123
4.1.3.	OPERACIONES.....	125
4.1.4.	HISTÓRICOS.....	127
4.1.5.	EROGACIONES.....	128
4.1.6.	DEPRECIACIONES.....	129
4.1.7.	TRANSFERENCIA DE CUSTODIO.....	130
4.1.8.	MOVIMIENTOS TEMPORALES.....	131
4.1.9.	REPORTES.....	133
4.2.	AMENAZAS Y VULNERABILIDADES.....	135
4.2.1.	OBJETIVOS.....	135
4.2.2.	FUERA DE ALCANCE EN LOS OBJETIVOS.....	135
4.2.3.	ESCENARIOS PRINCIPALES.....	136
4.2.4.	TECNOLOGÍAS.....	136
4.2.5.	MECANISMOS DE SEGURIDAD DE LAS APLICACIONES.....	136
4.2.6.	ELEMENTOS DE LA APLICACIÓN.....	137
4.2.7.	AMENAZAS.....	138
4.2.8.	VULNERABILIDADES.....	139
4.3.	PRUEBAS.....	139
4.3.1.	PRUEBAS UNITARIAS.....	139
CAPITULO V. CONCLUSIONES Y RECOMENDACIONES.....		162
5.1.	CONCLUSIONES.....	162
5.2.	RECOMENDACIONES.....	163
REFERENCIAS BIBLIOGRÁFICAS:.....		164
ANEXOS.....		165
MANUAL TÉCNICO.....		166
MANUAL DE USUARIO.....		205

LISTADO DE GRÁFICOS

FIGURA 1.	ARQUITECTURA FRAMEWORK.....	20
FIGURA 2.	MSF AGILE.....	21
FIGURA 3.	ITERACIONES DE MSF AGILE.....	22
FIGURA 4.	GOBIERNO DE MSF AGILE.....	22
FIGURA 5.	MENÚ TEAM EXPLORER EN VISUAL STUDIO.....	30
FIGURA 6.	TEAM EXPLORER EN VISUAL STUDIO.....	31
FIGURA 7.	ADMINISTRACIÓN DE TAREAS DEL PROYECTO.....	32
FIGURA 8.	OPCIONES DEL PROYECTO.....	33
FIGURA 9.	PORTAL DEL PROYECTO.....	34
FIGURA 10.	SOURCE SAFE DEL PROYECTO.....	35
FIGURA 11.	MENÚ DE HERRAMIENTA DE CODE SMITH.....	35
FIGURA 12.	OPCIONES DEL CODE SMITH.....	36
FIGURA 13.	EXPLORADOR DEL ESQUEMA DE CODE SMITH.....	37

FIGURA 14. EXPLORADOR DE PLANTILLAS.....	37
FIGURA 15. PROPIEDADES DE CODE SMITH	38
FIGURA 16. EDITOR DE MAPEO	38
FIGURA 17. PLANTILLAS DE CODE SMITH.....	39
FIGURA 18. CONTROL FORMATO DE FECHA	41
FIGURA 19. CONTROL MENSAJE.....	42
FIGURA 20. CONTROL CALENDARIO	43
FIGURA 21. CONTROL COMBO BOX	43
FIGURA 22. MENSAJE ADVERTENCIA.....	44
FIGURA 23. CONTROL CATÁLOGO	45
FIGURA 24. CASO DE USO CONTROL DE SEGURIDAD.....	66
FIGURA 25. CASO DE USO CONTROL DE BIENES.....	68
FIGURA 26. CASO DE USO REPORTES DE BIENES	74
FIGURA 27. CASO DE USO REPORTES DE BIENES	75
FIGURA 28. ARQUITECTURA DE LA SOLUCIÓN.....	77
FIGURA 29. CONFIGURACIÓN GENERAL DE LA SOLUCIÓN.....	85
FIGURA 30. COMPONENTES DEL SISTEMA	87
FIGURA 31. MODELADO DE BASE DE DATOS.....	89
FIGURA 32. CRONOGRAMA DE ITERACIONES	103
FIGURA 33. ESCENARIO DE COLOR.....	105
FIGURA 34. BÚSQUEDAS DE ESCENARIO	106
FIGURA 35. ESCENARIO DE OPERACIONES DE COLOR	106
FIGURA 36. ESCENARIO DE ESTADO	107
FIGURA 37. ESCENARIO DE BÚSQUEDA DEL ESTADO	108
FIGURA 38. ESCENARIO DE OPERACIONES DEL ESTADO	108
FIGURA 39. ESCENARIO DE USO	109
FIGURA 40. ESCENARIO DE BÚSQUEDA DE USO.....	110
FIGURA 41. ESCENARIO DE OPERACIONES DE USO	110
FIGURA 42. ESCENARIO DE TIPO DE MOVIMIENTO	111
FIGURA 43. ESCENARIO DE BÚSQUEDA DE TIPO DE MOVIMIENTO	112
FIGURA 44. ESCENARIO DE OPERACIONES DE TIPO DE MOVIMIENTO	112
FIGURA 45. ESCENARIO DE TIPO DE BIEN	113
FIGURA 46. ESCENARIO DE BÚSQUEDA DE TIPO DE BIEN.....	114
FIGURA 47. ESCENARIO DE OPERACIONES DE TIPO DE BIEN.....	114
FIGURA 48. ESCENARIO DE TIPO DE CUENTA	115
FIGURA 49. ESCENARIO DE BÚSQUEDA DE TIPO DE CUENTA.....	116
FIGURA 50. ESCENARIO DE OPERACIONES DE TIPO DE CUENTA	116
FIGURA 51. ESCENARIO DE ESTRUCTURA	117
FIGURA 52. ESCENARIO DE BÚSQUEDA DE ESTRUCTURA	118
FIGURA 53. ESCENARIO DE OPERACIONES DE ESTRUCTURA.....	118
FIGURA 54. ESCENARIO DE MOTIVO DE MOVIMIENTO.....	119
FIGURA 55. ESCENARIO DE BÚSQUEDA DE MOTIVO DE MOVIMIENTO	120
FIGURA 56. ESCENARIO DE OPERACIONES DE MOTIVO DE MOVIMIENTO	120
FIGURA 57. ESCENARIO DE ACCESORIOS	121
FIGURA 58. ESCENARIO DE BÚSQUEDA DE ACCESORIOS	121
FIGURA 59. ESCENARIO DE OPERACIONES DE ACCESORIOS	122
FIGURA 60. ESCENARIO DE CUENTAS	123
FIGURA 61. ESCENARIO DE OPERACIONES DE CUENTAS	124
FIGURA 62. ESCENARIO DE OPERACIONES.....	125
FIGURA 63. ESCENARIO DE CÓDIGO DE BARRAS.....	126
FIGURA 64. ESCENARIO DE HISTÓRICOS.....	127
FIGURA 65. ESCENARIO DE EROGACIONES	128
FIGURA 66. ESCENARIO DE OPERACIONES DE EROGACIONES	129
FIGURA 67. ESCENARIO DE DEPRECIACIONES	130
FIGURA 68. ESCENARIO DE TRANSFERENCIA DE CUSTODIO	131
FIGURA 69. ESCENARIO DE MOVIMIENTOS TEMPORALES	132

FIGURA 70. ESCENARIO REPORTE BIEN DETALLE	133
FIGURA 71. ESCENARIO REPORTE DEPRECIACIÓN	134
FIGURA 72. ESCENARIO REPORTE GRUPO DE BIENES	135
FIGURA 73. CREACIÓN DE PROYECTO DE PRUEBAS	140
FIGURA 74. MÉTODOS DE LA DAL DE BIENES	140

LISTADO DE TABLAS

TABLA 1. TABLA DE CATÁLOGO	16
TABLA 2. TABLA DE FUNCIONES	16
TABLA 3. TABLA DE VALIDACIONES	17
TABLA 4. TABLA DE LLAVES PRINCIPALES	18
TABLA 5. TABLA DE VARIAS FUNCIONES	18
TABLA 6. INSTALACIÓN FRAMEWORK.....	19
TABLA 7. MARCO DE TRABAJO	24
TABLA 8. RUP VS MSF AGILE.....	26
TABLA 9. REQUERIMIENTOS CANDIDATOS	53
TABLA 10. VALORACIÓN DE RIESGOS INHERENTES	57
TABLA 11. RIESGOS IDENTIFICADOS	58
TABLA 12. RIESGOS TOTALES.....	61
TABLA 13. VARIABLES PARA CÁLCULO NETO O RESIDUAL.....	62
TABLA 14. CALCULO VALOR NETO O RESIDUAL	64
TABLA 15. ESTRUCTURA DEL PROYECTO	65
TABLA 16. ACTORES CONTROL SEGURIDAD.....	66
TABLA 17. ACTORES CONTROL BIENES	68
TABLA 18. ACTORES REPORTES DE BIENES.....	74
TABLA 19. ACTORES ADMINISTRACIÓN DE BIENES.....	76
TABLA 20. ELEMENTOS DE LA SOLUCIÓN.....	78
TABLA 21. DICCIONARIO DE DATOS DE BIEN.....	91
TABLA 22. DICCIONARIO DE DATOS DE CUENTA	91
TABLA 23. DICCIONARIO DE DATOS DE UBICACIÓN DEL BIEN.....	92
TABLA 24. DICCIONARIO DE DATOS DE MOVIMIENTOS	93
TABLA 25. DICCIONARIO DE DATOS DE ESTADO	93
TABLA 26. DICCIONARIO DE DATOS DE HISTÓRICOS	94
TABLA 27. DICCIONARIO DE DATOS DE COLOR	95
TABLA 28. DICCIONARIO DE DATOS DE ESTRUCTURA.....	95
TABLA 29. DICCIONARIO DE DATOS DE TIPO.....	96
TABLA 30. DICCIONARIO DE DATOS DE TIPO DE BIEN.....	96
TABLA 31. DICCIONARIO DE DATOS DE GARANTÍA	97
TABLA 32. DICCIONARIO DE DATOS DE OPERACIÓN.....	97
TABLA 33. DICCIONARIO DE DATOS DE TIPO DE MOVIMIENTO	98
TABLA 34. DICCIONARIO DE DATOS DE MOTIVOS	98
TABLA 35. DICCIONARIO DE DATOS DE USO	99
TABLA 36. DICCIONARIO DE DATOS DE ACCESORIOS	99
TABLA 37. DICCIONARIO DE DATOS DE PERIODOS.....	100
TABLA 38. DICCIONARIO DE DATOS DE PERSONAS.....	100
TABLA 39. DICCIONARIO DE DATOS DE PROVEEDORES	101
TABLA 40. DICCIONARIO DE DATOS DE UBICACIÓN	101

CAPITULO I. INTRODUCCIÓN

1.1. Antecedentes

La tecnología avanza al ritmo en que algunas empresas de nuestro país, se obligan actualizar sus recursos y herramientas de acuerdo a sus necesidades, es por esta razón, que el Municipio del Distrito Metropolitano de Quito (MDMQ), desea realizar un sistema, mediante el cual administre todos sus bienes.

La Dirección Municipal de Gobierno Electrónico se crea como dependencia de la Alcaldía Metropolitana, por consiguiente tiene a su cargo el desarrollo y la ejecución de políticas adecuadas para la gobernabilidad electrónica institucional, interinstitucional, interurbana, e internacional correspondiente al gobierno electrónico.

Para cubrir las necesidades del sistema, el MDMQ establecerá las siguientes políticas para el área de desarrollo de aplicaciones:

- ✚ Unificar las relaciones colectivas a la conectividad, mediante el uso de las tecnologías de información y comunicación, que se encuentran dispersas en las dependencias municipales del MDMQ.
- ✚ Proyectar las políticas hacia el ámbito internacional, coordinarles y fiscalizar su ejecución.
- ✚ Brindar a las distintas instancias municipales, corporaciones y empresas metropolitanas, los procesos de incorporación de tecnología informática y las especificaciones que cubran en forma genérica sus necesidades.
- ✚ Definir una metodología común para la planificación estratégica de las tecnologías de Información y Comunicación en el ámbito de la municipalidad del MDMQ.

El desarrollo del presente proyecto proveerá un rol de supervisión sobre la operación de administración en actividades tales como: registros de bienes, generación de reportes, administración de cuentas, entre otras. Las mismas que al ser automatizadas permitirán alcanzar una excelencia en producción de la empresa y optimización de sus bienes.

1.2. Planteamiento del Problema

El Municipio del Distrito Metropolitano de Quito (MDMQ) creó nodos de gobierno electrónico en cada Administración zonal, empresa, corporación y fundación municipal, para integrar y centralizar el proceso de administración de la municipalidad del distrito Metropolitano de Quito. Los nodos zonales se encargan de aplicar y coordinar la gobernabilidad electrónica en las diferentes instancias a su cargo. Estos nodos son los encargados de procesar una aplicación idónea del uso de tecnología de información y Comunicación.

Actualmente el Sistema de Administración y Control de Bienes implementado por parte del MDMQ, se encuentra desarrollado en un lenguaje de bajo nivel, el cual no satisface las necesidades actuales como: Conexión de varias zonas municipales, Ingreso al sistema mediante la Web, Modificación de datos en cualquier terminal, entre otras; por lo tanto, la Dirección Municipal de Gobierno Electrónico ha visto la necesidad de implementar una aplicación Web para conocer la cantidad y valor real de sus Bienes de Larga Duración y Bienes Sujetos de Control, de todas sus Administraciones Zonales, tomando como referencia las últimas reformas en las Ordenanzas y Resoluciones respecto al control físico y contable de Bienes de Larga Duración en Entidades Públicas.

1.3. Justificación e Importancia

El desarrollo de este software cubrirá la necesidad que existe de automatizar los procesos que intervienen en la Administración de Bienes de Larga Duración y Bienes Sujetos de Control, garantizando un mayor control en la administración de los bienes

que posee la empresa y asignación de sus respectivos custodios; además traspasos internos, generación de reportes, cálculos de depreciación, etc.

Por tanto la automatización de los procesos que el MDMQ permite a este proyecto que logre ciertos beneficios los cuales se listan a continuación:

- ✚ Tener calidad en el software y obtener información oportuna en el sistema.
- ✚ Tener un control de todos los recursos materiales obtenidos en el MDMQ.
- ✚ Integración todos los nodos municipales para tener un total de Bienes que posee la empresa.
- ✚ Controlar la distribución adecuada de todas sus pertenencias.
- ✚ Conseguir un alto grado de eficiencia y eficacia en su administración.
- ✚ Minimizar la cantidad de documentación que actualmente se llevan en la adquisición de estos recursos necesarios para la producción de la empresa consiguiendo con esto generar consultas en menor tiempo y con mayor exactitud.
- ✚ Tener una seguridad al momento de adquirir un recurso para la empresa.

1.4. Objetivos

1.4.1. Objetivo General

- ✚ Desarrollar una aplicación Web distribuida para el Municipio del Distrito Metropolitano de Quito que permita administrar y controlar los Bienes de Larga Duración y Bienes Sujetos de Control con el fin de obtener información global y verídica de todos sus Bienes.

1.4.2. Objetivos Específicos

- ✚ Relevar y Optimizar los procesos que intervienen en la administración de Bienes de Larga Duración y Bienes Sujetos de Control Municipales.
- ✚ Realizar el análisis de requisitos funcionales y no funcionales de la Aplicación.
- ✚ Diseñar y Desarrollar la Aplicación automatizando los procesos que intervienen en la Administración y Control de Bienes Municipales.
- ✚ Realizar pruebas de la Aplicación para su futura implementación en el MDMQ.
- ✚ Realizar el Manual Técnico y de Usuario.

1.5. Alcance

La aplicación Web automatizará todos los procesos necesarios para su administración.

Uno de los factores determinantes en la decisión de implementar este Sistema es optimizar recursos y poseer una administración adecuada a todas las posesiones que tiene esta empresa.

En esta aplicación se realizarán las pruebas respectivas que satisfagan las necesidades del cliente.

Para respaldar la información realizada en el desarrollo del sistema se aplicará manuales técnicos y de usuarios que ayuden al cliente a utilizar la aplicación correctamente.

Aplicar la nueva metodología adquirida por el Municipio del Distrito Metropolitano de Quito.

Para el desarrollo de Aplicación Web se ha definido el siguiente alcance:

- ✚ Aplicación Multiempresa
- ✚ Proceso de Ingreso de nuevos Bienes de Larga Duración y Bienes Sujetos de Control Municipales
- ✚ Modificación de información de Bienes Municipales existentes
- ✚ Proceso de Traspasos
- ✚ Proceso de Baja
- ✚ Movimientos Temporales
- ✚ Proceso de Erogaciones
- ✚ Proceso de Depreciación
- ✚ Histórico de Movimientos
- ✚ Administración de Parámetros
- ✚ Reportes parametrizados Web
- ✚ Reportes Gerenciales Web

CAPITULO II. MARCO TEÓRICO

2.1. Tecnología

2.1.1. JavaScript y VBScript

JavaScript.-

Es un lenguaje de programación que no requiere compilación, se utiliza principalmente cuando se realizan páginas web.

JavaScript es un lenguaje orientado a objetos por las características que brinda en su programación, una de sus principales características es que dispone de herencia y puede extender su funcionalidad.

Todos los browsers pueden interpretar la codificación que brinda javascript en el desarrollo de una página web.

Se utiliza en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor.

JavaScript y Java son dos cosas distintas. Principalmente porque Java sí que es un lenguaje de programación completo. Lo único que comparten es la misma sintaxis.

El formato que debe tener javascript es:

```
 <script type="text/javascript" src="[URL]"></script>
```

[URL] la URL relativa o absoluta del recurso con código JavaScript, el cual tendrá extensión js.

También es posible incluir código directamente en el documento entre los elementos <script> y </script>, aunque no se recomienda:

```

<script type="text/javascript">
<!--
  // código JavaScript
-->
</script>

```

Las funciones que se serán utilizadas dentro del proyecto a desarrollarse son:

Catalogo.js

Este archivo es solo utilizado en la elaboración de catálogos.

fnResultado	Escribe el resultado del catálogo en la caja de texto del objeto que llamó al catálogo.
-------------	---

Tabla 1. Tabla de Catálogo

Funciones.js

marcarTodos(Ids)	Marcar todos los elementos tipo CHECKBOX que estén dentro de un grid.
------------------	---

Tabla 2. Tabla de Funciones

Validaciones.js

mostrarErrores()	Mostrar errores en un mensaje de MsgBox.
validaCampo()	Valida que un campo de texto no esté vacío
validaCampoDDL()	Valida que un campo DropDownList no esté vacío su valor.
validaCampoDDLNoSeleccionable()	Valida que un campo UCDropDownListEnabled no esté seleccionado un valor no permitido.
agregaError()	Agrega un nuevo error a la lista de errores.
validaInnerText()	Valida la propiedad InnerText.
validaComaPunto()	Valida que no estén ingresados los caracteres “.” y “,”

validaLongitud()	Valida la longitud de un campo. Existen 2 sobrecargas a este método. La primera solo valida un mínimo, y la segunda un mínimo y un máximo.
validaRango()	Valida que un valor este dentro de un rango.
validaIgualdad()	Valida que dos objetos tengan igual contenido.
validaNoIgualdad()	Valida que dos objetos no tengan igual contenido.
comparaFechas()	Compara si una fecha es mayor a otra.
ValidaNumeroOnPaste	Valida un número cuando se pega en un campo.

Tabla 3. Tabla de Validaciones

VBScript.-

Visual Basic Script es un lenguaje elaborado por Microsoft y es utilizado por navegadores para mejorar el desarrollo de la aplicación, con vbscript se puede crear botones interactivos y barras de desplazamiento.

Una de las principales características de vbscript es que es parecido al lenguaje de programación Visual Basic, su aprendizaje es mucho más sencillo que otros lenguajes de programación, es compatible con todo tipo de navegadores de internet. La decisión del lenguaje que debemos utilizar en nuestra aplicación depende del sitio que vamos a desarrollar.

Las funciones que se serán utilizadas dentro del proyecto a desarrollarse son:

Keys.vbs

KeyNumberPos()	Solo permite el ingreso de números positivos.
KeyNumber()	Solo permite el ingreso de números.
KeyMoney(sep)	Solo permite el ingreso de números con o sin decimales (sep es el separador).
KeyLock()	No permite hacer nada, el campo se encuentra bloqueado.
KeyDate()	Solo permite números y el signo "/"
KeyFone()	Solo permite números, paréntesis, - y ext.

KeyTime()	Solo permite números y el signo “.”
KeyChar()	Solo permite letras de la “a” – “z”.
KeyAlfaNum()	Solo permite números y letras.
KeyAlfaNumEsp()	Solo permite números, letras y espacios.
KeyString()	Solo permite letras y el espacio.
KeyUpper()	Transforma a mayúsculas.
KeyLower()	Transforma a minúsculas.

Tabla 4. Tabla de Llaves Principales

Varios.vbs

vbAlert()	Presenta un mensaje en una caja MsgBox.
vbConfirmar	Espera una respuesta a un mensaje presentado en una caja MsgBox.

Tabla 5. Tabla de Varias Funciones

2.1.2. .NET Framework

Es un marco de trabajo que se define como el componente principal de la plataforma .NET, el cual sirve para poder desarrollar aplicaciones y poder ejecutarlas luego en entornos de prueba o producción.

.NET Framework debe estar instalado en cualquier dispositivo de hardware para que la aplicación se pueda ejecutar.

En las aplicaciones de escritorio debe estar instalado en el lado del cliente ya que la aplicación interactúa con el mismo y se instala en el servidor cuando la aplicación es distribuida y que la parte funcional este en una sola computadora.

Para las Aplicaciones Web solo se necesita que el Framework este instalado en el lado del servidor y que el cliente tenga un navegador y una conexión de red al servidor, las

aplicaciones de consolas tienen la misma funcionalidad que las aplicaciones de escritorio y en Aplicaciones móviles se ejecutan en cualquier dispositivo móvil.

	CLIENTE	SERVIDOR
Aplicación de Escritorio	SI	SI (Distribuido)
Aplicación WEB	NO	SI
Aplicación de Consolas	SI	SI (Distribuido)
Aplicación Móvil	.NET COMPACT FRAMEWORK	

Tabla 6. Instalación Framework

.NET Framework se divide en tres variantes principales:

- ✚ **.NET Framework Redistributable Package.-** Es un componente pequeño de la plataforma .NET que se necesita para poder ejecutar aplicaciones. Este componente se instala en el proceso de producción, una vez que el desarrollo y las pruebas de la aplicación hayan concluido. El Framework de .NET esta compuesto por un entorno de Ejecución y una biblioteca de funcionalidad reutilizable.
- ✚ **.NET Framework SDK.-** Contiene herramientas de desarrollo de línea de comandos como compiladores, depuradores, entre otros. Esta variante se instala en los entornos de desarrollo de aplicaciones, y usan más los programadores que a los usuarios finales.

2.1.2.1. Arquitectura .NET Framework

La arquitectura de .NET Framework se divide en varias partes incluido el entorno de Ejecución de Aplicaciones, las bibliotecas de funcionalidad reutilizables y los compiladores con las herramientas de desarrollo para las aplicaciones.

Dentro del conjunto de la .NET Framework Class Library se distinguen 4 sub-componentes principales:

- ✚ **Base Class Library.-** Contiene la funcionalidad más comúnmente utilizada para el desarrollo de todo tipo de aplicaciones.
- ✚ **ADO.NET.-** Es un conjunto de clases que permiten interactuar con bases de datos relacionales y documentos XML como repositorios de información persistente.
- ✚ **ASP.NET.-** Es una tecnología dentro del .NET Framework para construir aplicaciones Web
- ✚ **Windows Forms.-** Es una tecnología dentro del .NET Framework para construir aplicaciones con interfaz de usuario y ventanas Windows q interactúan con el cliente.
- ✚ **Common Language Runtime.-** Se define como virtual o de maquina virtual y es el encargado de la ejecución de las aplicaciones y proveerles servicios en tiempo de ejecución. A los componentes de software que se ejecutan de esta manera se los conoce comúnmente como componentes manejados, ya que su ejecución es controlada por un entorno intermedio.

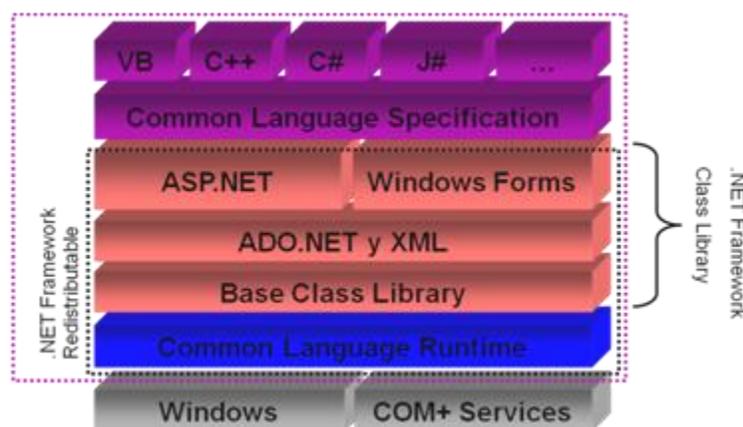


Figura 1. Arquitectura Framework¹

¹ Obtenido de documentación 5 estrellas Desarrollador Microsoft

2.2. Ingeniería de Software

2.2.1. Metodología

2.2.1.1. MSF for Agile Software Development (Microsoft Solution Framework)

Es un proceso ágil de desarrollo de software, está compuesto por escenarios y basado en contexto para la creación de aplicaciones .NET y otras aplicaciones orientadas a objetos.

MSF for Agile Software Development controla practicas de requisitos de calidad como seguridad y rendimiento, a su vez utiliza un enfoque que depende de su contenido para determinar la organización de un proyecto. Esta guía facilita la creación de un proceso que supera la mayoría de los procesos de desarrollo de software ágil y logra con los objetivos que formaron parte de la visión del proyecto.

La integración de MSF for Agile Software Development en Visual Studio Team System admite el diseño iterativo rápido con el continuo aprendizaje y perfeccionamiento.

La definición, el desarrollo y la comprobación del producto tienen lugar en iteraciones que se solapan y dan como resultado la finalización incremental del proyecto.

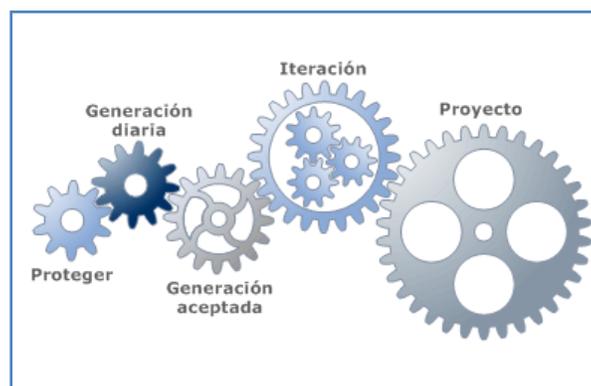


Figura 2. MSF Agile²

² Obtenido de documentación del MDMQ

Las distintas iteraciones tienen un foco diferente conforme se acerca el lanzamiento del proyecto. Las iteraciones pequeñas permiten reducir el margen de error de las estimaciones y proporcionan comentarios rápidos sobre la precisión de sus planes de proyecto. Cada iteración debe tener como resultado una parte estable del sistema global.

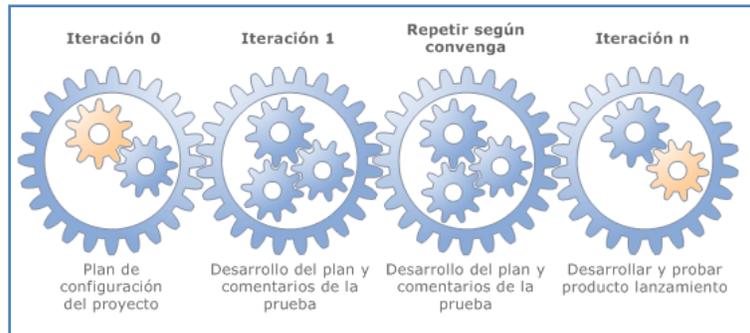


Figura 3. Iteraciones de MSF Agile

2.2.1.1.1. Gobierno

El gobierno se define como el control de tiempo y dinero en relación al flujo de valor. MSF for Agile Software Development define cinco puntos de control de gobierno, cada uno de los cuales se centra en contestar a una pregunta específica. Las actividades y secuencias de trabajo que conducen a los puntos de control se denominan secuencias de actividad. Las secuencias de trabajo para el gobierno no es lo mismo que la organización de trabajo que se agrupa en iteraciones y los ciclos más precisos que lo componen.

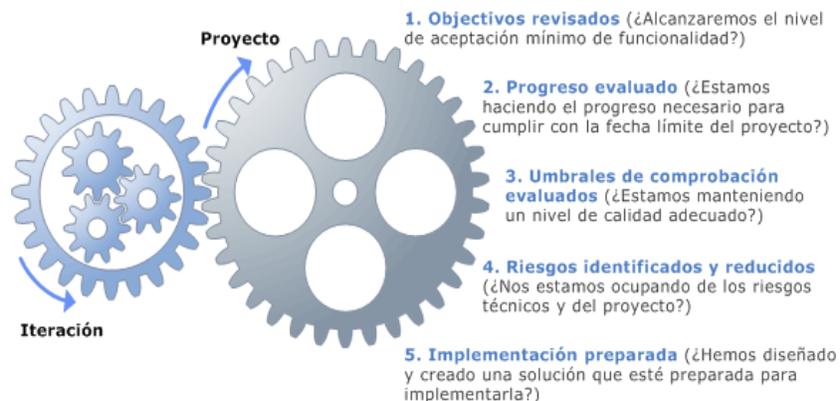


Figura 4. Gobierno de MSF Agile

2.2.1.1.2. Modelo de equipo

El modelo de equipo de Microsoft Solutions Framework agile es la estructuración de personas y sus actividades cuyo objetivo es el éxito del proyecto. Los principios fundamentales del modelo de equipo de MSF son:

- ✚ Un equipo de pares de plena confiabilidad, con responsabilidad compartida y comunicaciones abiertas. Cada persona que desempeña una función es responsable de una parte determinada de la calidad de la solución.
- ✚ Apoyo a todas las áreas principales que deben estar representadas en un proyecto de software con éxito. Están representadas todas las perspectivas con el fin de proporcionar un sistema de control que evite errores de omisión y decisiones parciales.

Cuando se crea el equipo de trabajo se debe seguir los siguientes pasos para la implementación de un proyecto mediante la metodología MSF for Agile Software Development.

Paso	Descripción
Primeras tareas	Cuando se crea un nuevo proyecto de equipo, también se crean elementos de trabajo para asignarlos y empezar a trabajar inmediatamente. Estos se encuentran en la hoja de cálculo Project Checklist.xls que se puede observar desde el portal del proyecto o en Team Explorer
Criterios de Salida	La lista de comprobación del proyecto contiene todos los elementos de trabajo que están marcados como criterios de salida para la iteración actual. Se utiliza para hacer un seguimiento de todo el trabajo importante que debe llevarse a cabo para completar la iteración correctamente
Actualizar el proyecto de equipo	Normalmente, el trabajo del proyecto empieza antes de crear el proyecto de equipo en un servidor Team Foundation Server. Se debe actualizar el servidor lo más pronto posible.
Productos de trabajo	Revise todas las plantillas de productos de trabajo de MSF for Agile Software Development que se encuentran en la carpeta Documentos en Team Explorer o en el portal del equipo. Se debe actualizar estas plantillas una vez que se cuente con la información requerida.
Elementos de trabajo	MSF for Agile Software Development utiliza elementos de trabajo relacionados.

<p>Punto de inicio en las secuencias de trabajo</p>	<ul style="list-style-type: none"> • Crear visión del proyecto • Planear una iteración • Una vez que estas dos secuencias de trabajo están en marcha, comienzan otras secuencias de trabajo y otras funciones empiezan a formar parte del proyecto de equipo.
--	--

Tabla 7. Marco de Trabajo

2.2.1.1.3. Elementos de Trabajo

Es un registro de la base de datos que Visual Studio Team Foundation utiliza para efectuar el seguimiento del trabajo y su estado. La base de datos de los elementos de trabajo y el almacén de métricas comunes hacen posible contestar a preguntas sobre el estado del proyecto en tiempo real. Se define cinco elementos de trabajo para asignar y dar seguimiento al trabajo las cuales definimos a continuación

2.2.1.1.3.1. Escenario

Un escenario es un tipo de elemento de trabajo, el cual registra un trayecto para la interacción del usuario a través del sistema. El escenario registra los pasos específicos que puede realizar para alcanzar un objetivo.

No siempre un escenario registra un trayecto satisfactorio es por esa razón que se debe ser concreto al momento de definirlos.

2.2.1.1.3.2. Requisito de Calidad de Servicio

Los requisitos de calidad de servicio documentan características del sistema, como rendimiento, carga, disponibilidad, esfuerzo, accesibilidad y capacidad de mantenimiento. Estos requisitos normalmente adquieren la forma de restricciones bajo las cuales debe funcionar el sistema.

2.2.1.1.3.3. Tarea

Cada función tiene sus propios requisitos para una tarea. Por ejemplo, el desarrollador utiliza tareas de desarrollo para asignar trabajo derivado de los escenarios o los

requisitos de calidad de servicio a los propietarios de los componentes. El ingeniero de pruebas utiliza las tareas de prueba para asignar el trabajo de escribir y ejecutar casos de pruebas. Una tarea también se puede utilizar para señalar regresiones o para sugerir que se realicen pruebas exploratorias.

2.2.1.1.3.4. Error

Un error es un elemento de trabajo que comunica que ha existido o existe un problema potencial en el sistema. El objetivo de abrir un error es crear un informe preciso de los errores de forma que permita al lector comprender el impacto total del problema. Los resultados de las pruebas deben mostrar el problema con claridad.

2.2.1.1.3.5. Riesgo

Un riesgo es un evento probable que puede tener un efecto negativo para el proyecto en el futuro. Un elemento de trabajo de riesgo documenta y realiza un seguimiento de los riesgos de un proyecto a nivel técnico y de organización. Los equipos que obtienen un entorno positivo de administración de riesgos tendrán más éxito a la hora de identificar y tratarlos eso no pasa con un entorno negativo.

2.2.1.1.3.6. Cuadro Comparativo con otra metodología

Existen varias diferencias con las metodologías tradicionales y ágiles las cuales detallamos con un ejemplo en el siguiente cuadro.

La siguiente tabla muestra las diferencias entre metodología RUP (Rational Unified process) con MSF Agile (Microsoft Solution Framework).

RUP	MSF AGILE
CARACTERISTICAS	
Documentación exhaustiva de todo el proyecto.	Las interacciones con el sistema son más importantes que las herramientas y los procesos que se utilizan.
Se enfoca en la planificación de procesos.	Es más importante crear un producto que funcione, que escribir documentación exhaustiva.
Alto costo al implementar algún cambio.	El cliente interactúa constantemente para el desarrollo

	del proyecto.
No ofrecer una buena solución para proyectos donde el entorno es volátil.	El desarrollo de un cambio es más importante que el seguimiento estricto de un plan.
VENTAJAS	
Se aplica en el desarrollo de proyectos de periodos extendidos como el caso de los proyectos de innovación.	Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto.
Seguimiento detallado en cada una de las fases.	Cuenta con plantillas para acelerar el proceso de documentación.
DESVENTAJAS	
La evaluación de riesgos es compleja	El análisis de riesgos es necesario, pero si se lo hace muy exhaustivo puede demorar o hasta frenar el avance del proyecto.

Tabla 8. RUP vs MSF Agile

2.3. Herramientas

2.3.1. Herramientas de Modelado

2.3.1.1. Visio Enterprise Architect

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación.

El navegador Internet Explorer incluye un visor de diagramas Visio, cuya extensión es vsd, llamado Visio Viewer.

Originalmente se estipulaba que sería una aplicación para dibujo técnico para el campo de Ingeniería y Arquitectura; con añadidos para desarrollar diagramas de negocios, su adquisición por Microsoft implicó drásticos cambios de directrices.

Con esta herramienta se puede diseñar diagramas que facilitan el entendimiento de los conceptos y transmisión de información de una forma más eficaz.

Ventajas de Visio Enterprise Architect

- Documentar y analizar los procesos de negocio.
- Diseñar, documentar y analizar los procesos de negocio gracias a las plantillas y formas para las que ofrecen soporte los proyectos BPM (Business Process Management), incluyendo las mejoras en calidad de las iniciativas Six Sigma y la documentación ISO 9000.
- Utiliza un modo de revisión para hacer un análisis de los procedimientos que se añaden en el modelado.
- Publicar diagramas desde Visio en el espacio de trabajo de Microsoft Sharepoint Portal Server o expórtelos utilizando el formato SVG (Scalable Vector Graphics) o actualizando con la función de Guardar como página Web.
- Utilizar las plantillas que ofrece Visio para recoger y estructurar ideas en una sesión de creativa. Exporte los diagramas de la reunión creativa a Microsoft Word, Microsoft Excel o XML para iniciar la creación de otros archivos de negocio.
- Elaborar y utilizar diagramas técnicos de una forma más sencilla.
- Elaborar y modificar diagramas con mayor rapidez.
- Crea soluciones conectadas y con un gran potencial, que se benefician de los servicios Web XML y del software conectado a Microsoft.NET- para establecer la conexión entre personas, información, sistemas y dispositivos. Conecte los diagramas a información empresarial relevante para que se comuniquen mejor y tomen medidas más decisivas.

2.3.2. Herramientas de Programación

2.3.2.1. ASP.NET 2005

Mediante ASP.NET se pueden realizar aplicaciones Web se lo puede definir como parte del Framework cuyo componente principal son los Web Form los cuales permiten separar la interfaz de usuario con la interfaz de aplicación. La interfaz de aplicación se encuentra al lado del servidor donde se almacenan las paginas, imágenes, etc.

Las páginas Web se encuentran realizadas por el protocolo HTTP (Hypertext Transfer Protocol), cuyo propósito es que los navegadores y los servidores Web se comuniquen unos con otros.

2.3.2.1.1. Ventajas de ASP.NET

- ✚ La parte ejecutable de una aplicación es compilada.
- ✚ Implementación y actualización de las aplicaciones sin reiniciar el servidor.
- ✚ Independiente del lenguaje de programación.
- ✚ Encapsulamiento de funcionalidad mediante controles de servidor y controles de usuario.

2.3.2.1.2. Elementos de las aplicaciones Web ASP.NET

2.3.2.1.2.1. Funciones de la gerencia de proyectos

Al crear una aplicación ASP.NET se pueden añadir páginas HTML y páginas de Formulario Web según sea el proyecto que se va a realizar. En una aplicación ASP.NET las páginas HTML son utilizadas para contenido estático y los Formularios Webs proporcionan una interfaz programable, excepto que la interfaz se muestra en un explorador Web u otro dispositivo Web.

- ✚ **Componentes:** Un componente es una función específica que puede realizar cualquier tipo de cálculo, validaciones, y realizar procesos complejos se lo puede interpretar como un servicio Web XML que se puede conectar mediante una red TCP/IP.
- ✚ **Datos:** La mayoría de las aplicaciones Web ASP.NET constan algún nivel de acceso a datos.
- ✚ **ADO.NET:** Proporciona un marco completo para acceder y administrar datos desde una variedad de fuentes, incluyendo bases de datos y archivos o flujos XML.

2.3.2.1.2.2. Otras funciones de infraestructura

ASP.NET proporciona un contexto en donde se pueden ejecutar esos elementos.

- ✚ **Seguridad:** ASP.NET incluye funciones de seguridad contra las que puede configurar y programar en su aplicación Web las cuales son: integridad, confidencialidad, confiabilidad, entre otras.
- ✚ **Rendimiento:** ASP.NET mantiene una memoria caché de salida que almacena páginas que han sido solicitadas anteriormente; al especificar las configuraciones de la memoria caché, puede controlar por cuanto tiempo se mantienen las páginas en la memoria caché y bajo qué circunstancias se actualizan.
- ✚ **Traza:** Debido a que las aplicaciones Web se ejecutan en el servidor, a menudo un servidor remoto, no cuentan con una salida que no sea la salida de la aplicación.

2.3.2.2. Visual Studio Team System

Visual Studio Team System se constituye un conjunto de herramientas de colaboración, desarrollo, métricas y reportes, tanto del lado del cliente como del servidor que permiten el uso de Microsoft Solution Frameworks Agile.

Este es un meta modelo que describe procesos de negocio y de ingeniería de software, de forma que se pueda implementar procesos de desarrollo de software más efectivamente.

2.3.2.1.3. Utilización de Herramientas

Team Explorer es un “add-in” de Microsoft Visual Studio que permite la conexión con el servidor de Team Foundation Server y la realización de muchas actividades dentro de la metodología.

Una vez instalado en Microsoft Visual Studio aparece un menú “Team” como consta en la gráfica:

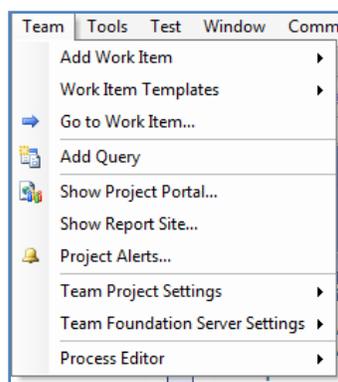


Figura 5. Menú Team Explorer en Visual Studio

Como se muestra se puede añadir nuevos ítems de trabajo, se puede acceder a las plantillas de ítems de trabajo, ir hacia un ítem de trabajo, añadir una consulta, mostrar el Portal del proyecto, mostrar el sitio del reporte, las alertas del proyecto, acceder a las configuración del proyecto de equipo y acceder a las configuraciones del servidor Team Foundation Server.

Adicionalmente, se cuenta con un explorador propiamente dicho que permite visualizar los proyectos y sus objetos relacionados, similar al explorador de proyectos de Visual Studio, el mismo se lo puede observar a continuación:

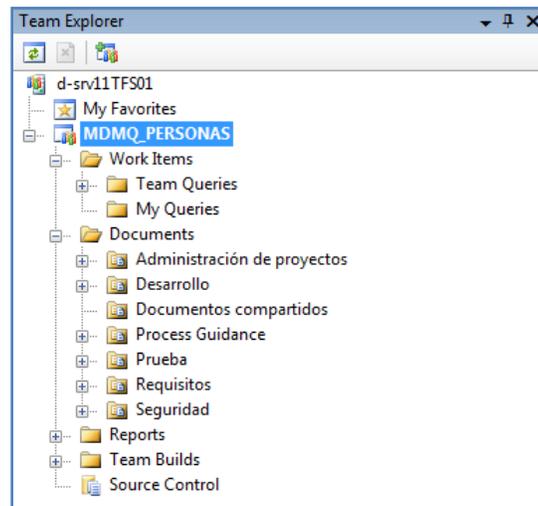


Figura 6. Team Explorer en Visual Studio

Tenemos los ítems de trabajo que se pueden consultar y administrar sus diferentes estados, otra sección para los documentos en la que se tiene:

- ✚ Administración de Proyectos
- ✚ Desarrollo
- ✚ Documentos compartidos
- ✚ Guía del Proceso
- ✚ Prueba
- ✚ Requisitos
- ✚ Seguridad

Después hay una sección para reportes de métricas, compilaciones y el acceso al Control de versiones que se revisará más adelante.

Para los ítems de trabajo se cuenta con interfaces dentro del mismo Visual Studio que permiten administrarlas y mantenerlas actualizadas, esto se lo puede ver a continuación:

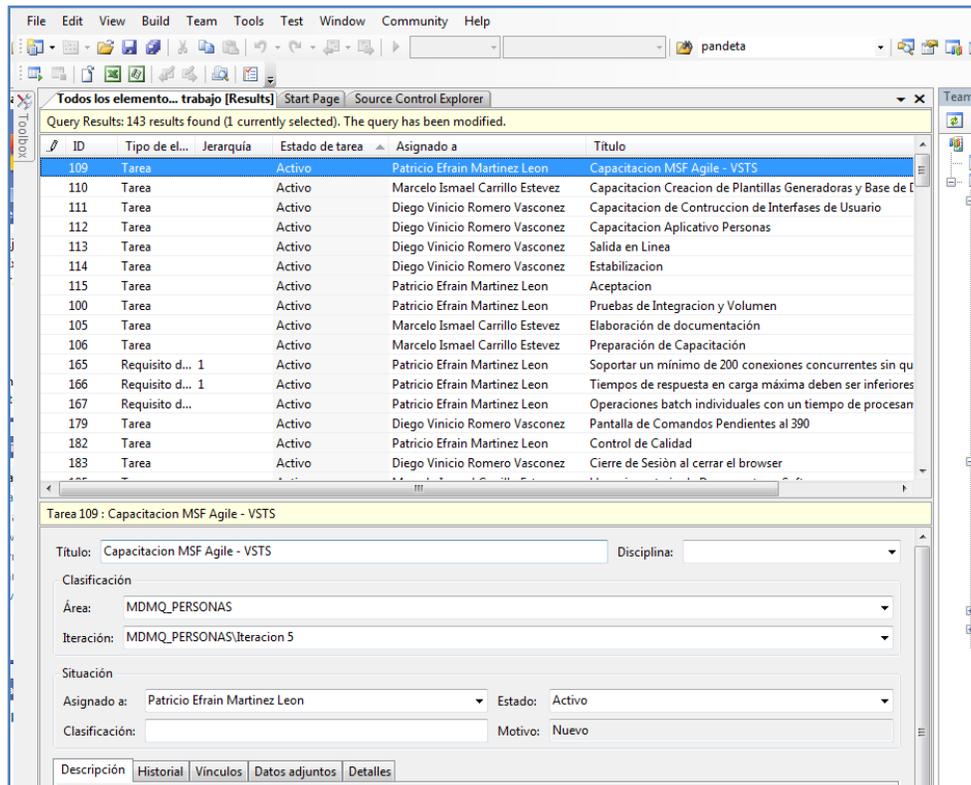


Figura 7. Administración de Tareas del Proyecto

Cabe anotar que Visual Studio Team System también incluye integración con algunas herramientas de Microsoft Office como es el caso de Microsoft Project y Microsoft Excel.

2.3.2.1.4. Portal del proyecto

En el Team Explorer si hacemos click derecho sobre el proyecto determinado nos aparece el siguiente menú emergente:

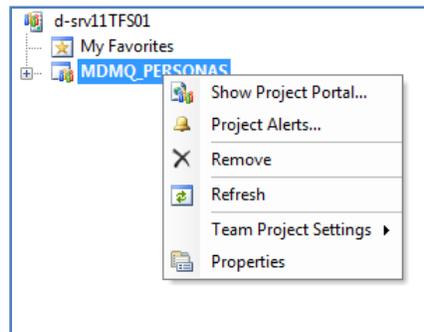


Figura 8. Opciones del Proyecto

Al seleccionar la primera opción “Show Project Portal” podemos a visualizar el Portal del Proyecto, este es un sitio web de Microsoft SharePoint Services que permite presentar el progreso del proyecto, mantener centralizados y publicados en línea documentos referentes al avance del proyecto, documentos de planificación, modelos, enlaces a prototipos, anuncios, etc.

Además se muestran un conjunto de reportes de métricas que permiten ver literalmente como va evolucionando el proyecto, también se cuenta con las guías del proceso que permiten que toda la gente involucrada en el proyecto conozca la metodología que se está utilizando y como puede proceder para interactuar de mejor manera y conseguir los objetivos buscados. A continuación se muestra la página principal del Portal del Proyecto:

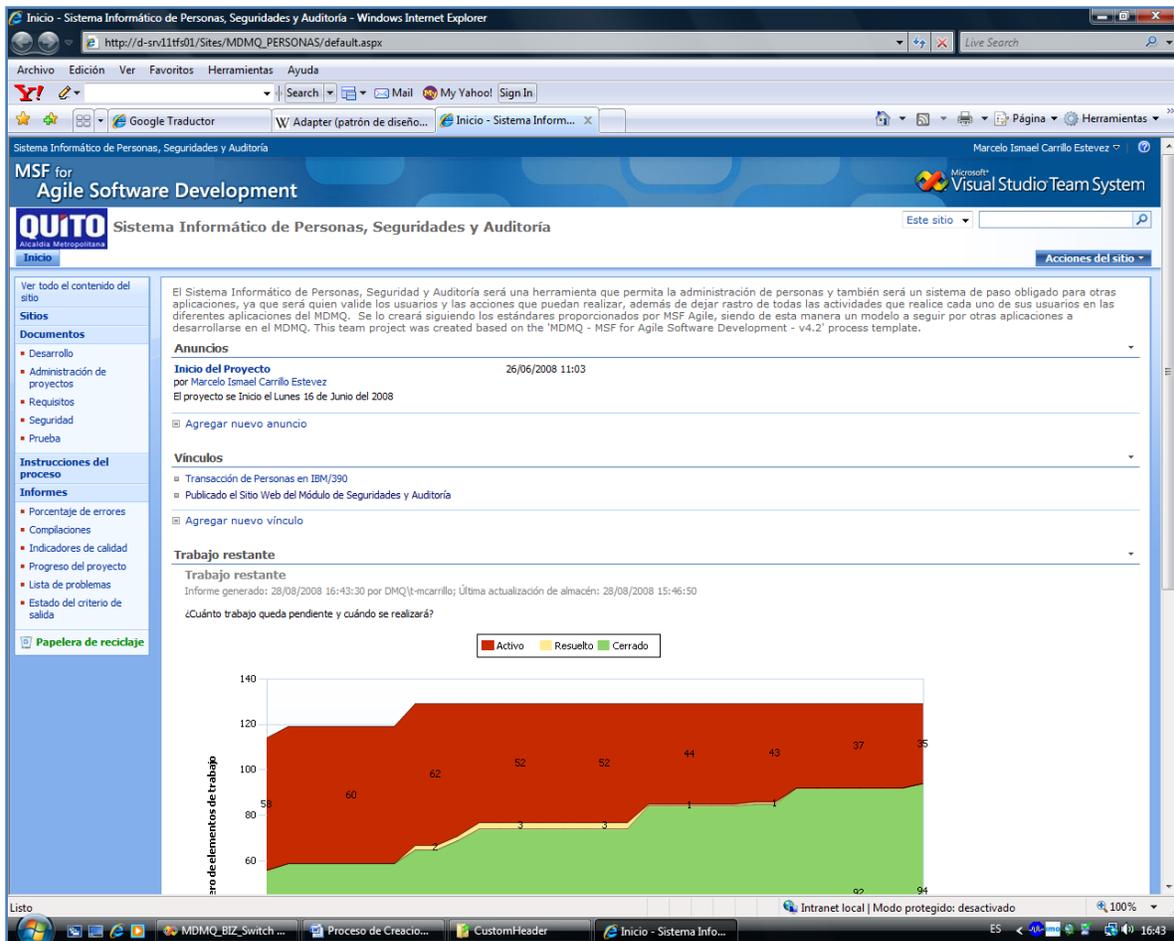


Figura 9. Portal del Proyecto

2.3.2.1.5. Control de Versiones

Esta es una parte muy importante de la solución Visual Studio Team System, ya que permite administrar los cambios en archivos en base a versiones, esto es de mucha utilidad sobre todo para el código fuente que es la sección primordial de un proyecto de desarrollo de software, ya que son los diseños que permiten la construcción de la solución final.

Otra ventaja que supone la utilización de un servidor de control de código fuente, es el hecho de que este se almacena en un sitio remoto a todas las máquinas de los desarrolladores, de tal suerte que si estas sufrieran un daño la última versión principal se encuentra disponible en el servidor.

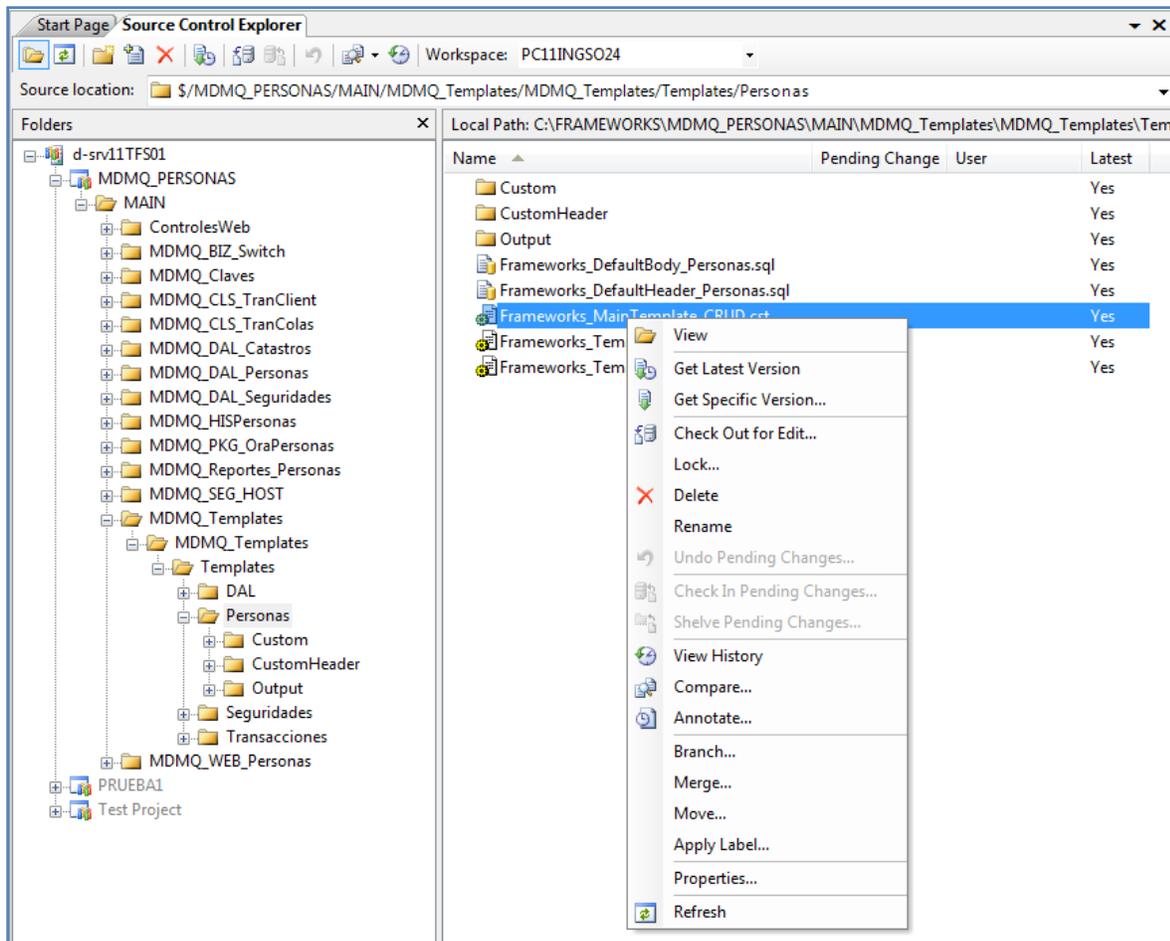


Figura 10. Source Safe del Proyecto

2.3.2.3. CodeSmith Studio

CodeSmith Studio es la IDE de desarrollo de las plantillas, es decir es el programa que se utiliza para modificar plantillas ya existentes y escribir nuevas plantillas. En la parte superior contiene un menú y una barra de herramientas como se lo puede observar en la imagen:



Figura 11. Menú de Herramienta de Code Smith

Este menú y barra es muy similar a otras IDE de desarrollo como Visual Studio, se tiene opciones para crear nuevos archivos, abrir, guardar, guardar todos, cortar, copiar, pegar, deshacer, rehacer, indentar, compilar, ejecutar, en fin todas aquellas acciones a las que están acostumbrados los profesionales del desarrollo de software.

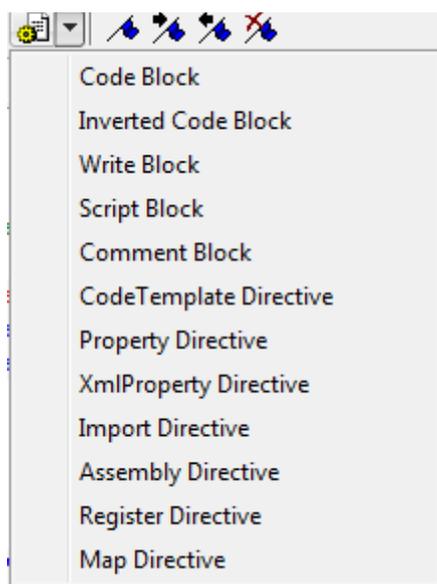


Figura 12. Opciones del Code Smith

Cabe destacar en la imagen el botón “Insert Content” que tiene un menú contextual y que nos permite añadir automáticamente diferentes combinaciones de tags y “esqueletos” de directivas para incluir código dinámico dentro de las plantillas y permitiéndonos además economizar tiempo en el desarrollo mismo de la plantilla y reduciendo la curva de aprendizaje de la sintaxis de las plantillas.

Por defecto a la izquierda presenta un control visual del Schema Explorer en el cual se navega a través de todas las conexiones que se tengan configuradas y en cada una de ellas a través de los metadatos de la base. En la siguiente gráfica se presenta el árbol que tiene para la conexión actual con la base Oracle.

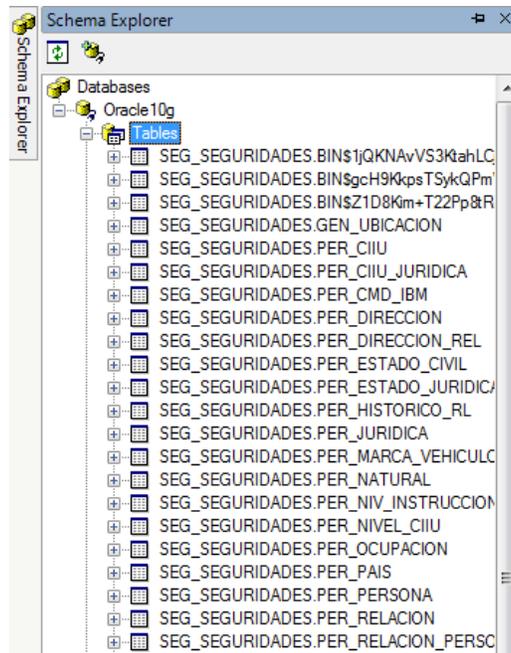


Figura 13. Explorador del Esquema de Code Smith

Existen otros paneles que ofrece Code Smith como el panel “Template Explorer” o Explorador de Plantillas este nos muestra el conjunto de plantillas que incluye CodeSmith.

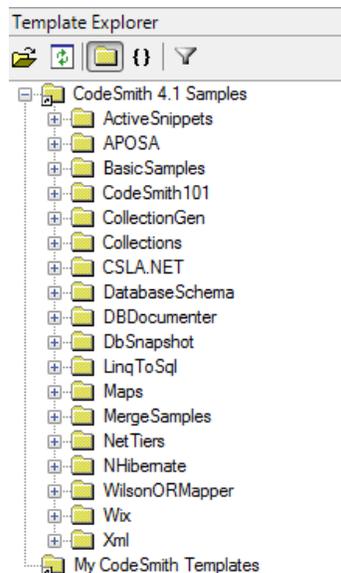


Figura 14. Explorador de Plantillas

Estos conjuntos de plantillas nos permiten generar código para muchas cosas, como capas de acceso a datos, diccionarios de datos y demás. Debajo de este se presenta el panel de configuración de propiedades que se lo muestra en la grafica a continuación:

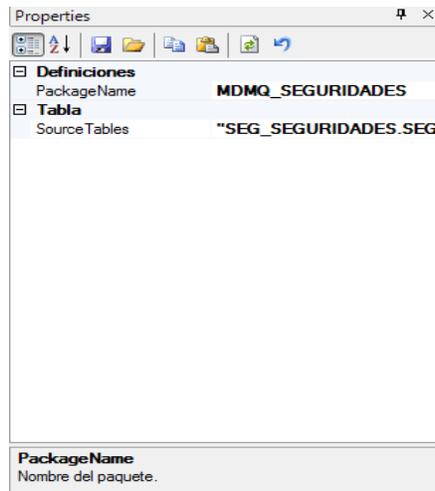


Figura 15. Propiedades de Code Smith

En la misma posición que el panel de propiedades pero como una pestaña adicional se encuentra el panel de Editor de Mapeo o “Map Editor” que permite editar o simplemente ver la correspondencia asignada de tipos de datos de diferentes fuentes como Oracle, SQL, etc. hacia C# o VB .NET. La imagen a continuación muestra la correspondencia entre tipos Oracle y .NET:

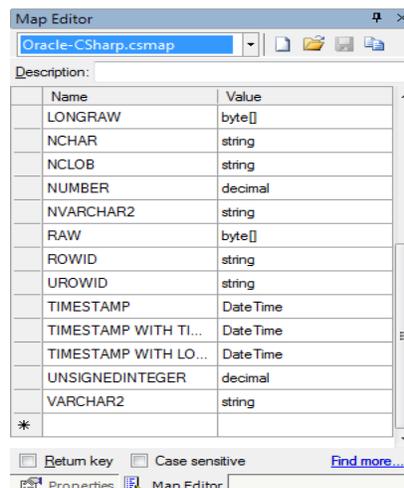


Figura 16. Editor de Mapeo

Finalmente la parte más importante de la interfaz es aquella en la que se escribe el código y se lo ejecuta, la Interface de Plantilla nos permite editar el código de la plantilla, nos muestra con colores las palabras y tags (etiquetas) reservadas y si estamos generando un código conocido se presentan en colores las palabras reservadas del lenguaje destino. Tenemos dos pestañas Template y Output en la primera se colocará el código de la plantilla y en la segunda podremos ver, después de la ejecución, el resultado final, es decir, el código generado por la plantilla. En las imágenes se presenta el detalle de ambas pestañas, a continuación:

The image shows two side-by-side screenshots of the Code Smith interface. The left screenshot shows the 'Template' tab with a SQL query template. The right screenshot shows the 'Output' tab with the rendered SQL query.

```

Frameworks_Template_CRUD_Bodyzst | Frameworks_MainTemplate_CRUDzst |
513      (i_<%=GetProperty( foreign ) %> IS NUL
514      <%=counter3++; %> <%=AddCustomCode( tbTab
515      <%=GetOrderBy( tbTable ) %> ) A)
516      WHERE indice < i_indice+i_filas
517      AND indice >= i_indice;
518  END IF;
519  IF (i_modos = 4) THEN
520
521  SELECT indice INTO w_indice FROM
522  (SELECT rownum indice, B.* FROM
523  (SELECT <%=GetAlias( tbTable.Name ) %>.*<%=
524  FROM <%= tbTable.Name %> <%=GetAlias( tbT
525  <%= int counter4 = 0; %>
526  <%= if ( tbTable.ForeignKeyColumns.Count>0
527  WHERE --Num Foraneas: <%=tbTable.Foreign
528  <%=foreach( SchemaExplorer.ColumnSchema fc
529  <%= if ( foreign.IsPrimaryKeyMember==true) {
530  <%= if ( IncludeCondition( tbTable, "i_" + G
531  (i_<%=GetProperty( foreign ) %> IS NUL
532  <%=else{ %> 1=1 <%= if ( counter4<tbTable.
533  WHERE --Num primarias: <%=tbTable.Pri
534  <%=foreach( SchemaExplorer.ColumnSchema fc
535  <%= if ( IncludeCondition( tbTable, "i_" + G
536  (i_<%=GetProperty( foreign ) %> IS NUL
537  <%=else{ %> 1=1 <%=if ( counter4<tbTable.F
538  <%=GetOrderBy( tbTable ) %>
539  )B
540  )A
541  WHERE <%= for ( int i = 0; i < tbTable.Pri
542  A.<%=tbTable.PrimaryKey.MemberColumns[i]
543
544  w_indice := (ceil(w_indice/i_filas)-1)*i
545
546  OPEN datos FOR
547  SELECT * FROM (
548  SELECT A.*, ROWNUM indice
549  FROM (
550  SELECT <%= for ( int i = 0; i < tbTable.Co
551  <%=tbTable.Columns[i].Name %> <%=i
552  FROM <%= tbTable.Name %> <%=AddCustomCode
    
```

```

Frameworks_Template_CRUD_Bodyzst | Frameworks_MainTemplate_CRUDzst |
2126      ,US_NOMBRE, US_APELLIDO, US_ESTADO, US_FECHA_ESI
2127      FROM SEG_GRUPO_OFICINA GRU , SEG_GRUPO,
2128      WHERE --Num Foraneas: 4
2129      (i_empresa_id IS NULL OR GO_EMPRESA_ID
2130      AND (i_grupo_id IS NULL
2131      AND (i_login_id IS NULL
2132      AND TRUE AND GO_EMPRESA_ID = EM_CODIG
2133      ORDER BY GO_EMPRESA_ID, GO_GRUPO_ID, GO
2134      )B
2135  )A
2136  WHERE
2137  A.GO_EMPRESA_ID = i_empresa_id AND
2138  A.GO_GRUPO_ID = i_grupo_id AND
2139  A.GO_LOGIN_ID = i_login_id AND
2140  A.GO_OFICINA_ID = i_oficina_id ;
2141
2142  w_indice := (ceil(w_indice/i_filas)-1)*i
2143
2144  OPEN datos FOR
2145  SELECT * FROM (
2146  SELECT A.*, ROWNUM indice
2147  FROM (
2148  SELECT
2149  GO_EMPRESA_ID ,
2150  GO_GRUPO_ID ,
2151  GO_LOGIN_ID ,
2152  GO_OFICINA_ID ,
2153  GO_ESTADO ,GR_NOMBRE,GR_DESCRIPC
2154  ,US_NOMBRE, US_APELLIDO, US_ESTADO, US_FECHA_ESI
2155  FROM SEG_GRUPO_OFICINA, SEG_GRUPO, SEG_C
2156  WHERE --Num Foraneas: 4
2157  (i_empresa_id IS NULL OR GO_EMPRESA_ID
2158  AND (i_grupo_id IS NULL
2159  AND (i_login_id IS NULL
2160  AND TRUE AND GO_EMPRESA_ID = EM_CODIG
2161  ORDER BY GO_EMPRESA_ID, GO_GRUPO_ID, GO
2162  WHERE indice <= w_indice+i_filas
2163  AND indice > w_indice;
2164  END IF;
2165
    
```

Figura 17. Plantillas de Code Smith

2.4. Motor de Base de Datos

2.4.1. Oracle Enterprise 10 g

Oracle es el manejador de Base de datos líder en el mundo gracias a sus características que le han permitido posicionarse como el más robusto, confiable, seguro mundialmente.

Se considera a Oracle como uno de los sistemas de bases de datos más eficientes debido a:

- ✚ Soporte de transacciones
- ✚ Estabilidad
- ✚ Escalabilidad
- ✚ Es multiplataforma

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS³ con licencia libre como PostgreSQL, MySql o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

Todas las bases de datos Oracle son compatibles y se encuentran disponibles para muchas plataformas diferentes, esto permite que las bases de datos puedan ser portables y escalables sin necesidad de cambiar las aplicaciones existentes.

2.4.1.1. Beneficios de Oracle

- ✚ Continua disponibilidad a pesar de fallas de sistema, situaciones anómalas, siniestros, errores humanos y planes de mantenimiento.

³ <http://www.alegsa.com.ar/Dic/RDBMS.php>

- ✚ Estándar de la industria con soporte para decenas de miles de aplicaciones disponibles tanto de Oracle como de sus partners.
- ✚ Soporta más tipos de datos que cualquiera otra Base de Datos, incluyendo XML, audio, documentos, texto y más.

2.4.1.2. Características de Oracle

- ✚ El más integral y abierto acceso a Servicios Web a través de SQL, Java, XML e interfaces Web estándares.
- ✚ Administración eficiente de datos del orden de los terabytes, a través de operaciones de paralelismo avanzado.
- ✚ Sistema de seguridad basados en usuarios, grupos y roles, alertas, backups y restauración de datos, auditorias, etc.

2.5. Controles del Programa

2.5.1. UCFormatoFecha

UCFormatoFecha es un UserControl creado por Frameworks Cía. Ltda., con la finalidad de mostrar al usuario el formato de una fecha que debe ser ingresada en un campo. Este es un UserControl que se puede utilizar junto con el CalendarioWEB.



Figura 18. Control Formato de Fecha

Facilidades que presta UCFormatoFecha

- ✚ Muestra al usuario el formato de la fecha que debe ingresar en un campo.
- ✚ El formato depende de la cultura, en este caso, el formato es fijo DD/MM/YYYY.

2.5.2. UCMensaje

UCMensaje es un UserControl creado por Frameworks Cía. Ltda., con la finalidad de mostrar al usuario mensajes de la aplicación que no necesariamente son mensajes de error, por ejemplo, el mensaje de que un determinado proceso ha terminado correctamente y sin errores.



Figura 19. Control Mensaje

Facilidades que presta UCMensaje

- ✚ Muestra al usuario un mensaje.
- ✚ Tiene mensajes por defecto, que se encuentran en el Web.Config de la aplicación. Estos mensajes son:

```
<add key="MsgError" value="HA OCURRIDO UN ERROR EN LA APLICACION,  
POR FAVOR INTENTE MAS TARDE."/>
```

```
<add key="MsgExito" value="LA OPERACION HA SIDO REALIZADA CON  
EXITO."/>
```

- ✚ Este control por defecto siempre se cargará con la propiedad Visible = false;

2.5.3. CalendarioWEB

CalendarioWEB es un UserControl creado por Frameworks Cía. Ltda., con la finalidad de estandarizar en las aplicaciones que controle el MDMQ, el manejo de un solo calendario que permita la selección y asignación de las fechas.

Este control tiene las siguientes presentaciones:

1. Una caja de texto, acompañado de una imagen a la derecha, donde se puede ingresar directamente una fecha. En el evento "LooseFocus" de la caja de texto se valida que la fecha ingresada tenga el formato correcto. En caso de que la caja de texto quede vacía, no se realiza la validación.
2. Una página con un calendario, que aparece al hacer clic en la imagen. Cuando se escoge una *fecha*, *esta página se cierra y la fecha escogida aparece en la caja de texto.*



Figura 20. Control Calendario

Facilidades que presta CalendarioWEB

- ✚ Es un calendario que se maneja con JavaScript, por lo que no es necesario en la aplicación realizar viajes al servidor para obtener los datos.
- ✚ Realiza validaciones de la fecha escrita de acuerdo a la cultura escogida.
- ✚ Se puede modificar los colores del calendario, solo con modificar la hoja de estilos.

2.5.4. UCDropDownListEnabled

UCDropDownListEnabled es un UserControl creado por Frameworks Cía. Ltda., con la finalidad de heredar las propiedades que tiene un combo y aumentar la funcionalidad de que aparezcan elementos con un mensaje (NO SELECCIONABLE) de manera que no pueden ser escogidos por parte del usuario.

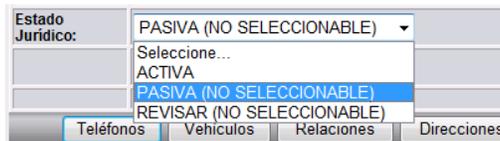


Figura 21. Control Combo Box

En caso de que el usuario escoja una de las opciones que tienen el mensaje NO SELECCIONABLE, al momento de hacer clic en el botón “Aceptar”, le aparecerá un mensaje como el siguiente:

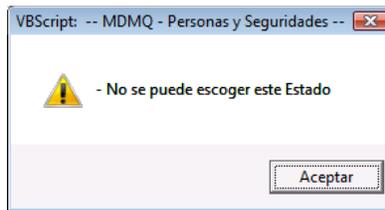


Figura 22. Mensaje Advertencia

Facilidades que presta este UCDropDownListEnabled

- ✚ Todas las propiedades que tiene un DropDownList.
- ✚ Permite incluir en la lista de elementos que no estén activos (no seleccionables) dentro de la aplicación y que si deben ser visualizados pero no deben ser escogidos por el usuario.

2.5.5. CatalogoWEB

CatalogoWEB es un UserControl creado por Frameworks Cía. Ltda., con la finalidad de estandarizar en las aplicaciones que controle el MDMQ, el manejo de búsquedas de datos en las aplicaciones.

Para facilidad de explicación de este UserControl, tomaremos como ejemplo la búsqueda de Países.

El control tiene las siguientes presentaciones:

1. Una caja de texto, acompañado de una imagen a la derecha y otra caja de texto (que está deshabilitada), donde irá la descripción del código ingresado en la primera caja de texto. En el evento "OnBlur" (cuando se presiona la tecla TAB o cuando se selecciona otro campo mediante el mouse) de la primera caja de texto se valida el código ingresado. En caso de que la caja de texto quede vacía, no se valida nada. Si en la primera caja de texto se digita la tecla "?", tiene el mismo efecto que hacer clic en la imagen, aparecerá la segunda presentación de este UserControl.

2. Una pantalla donde están las diferentes formas de búsqueda, necesarias para el manejo del UserControl.

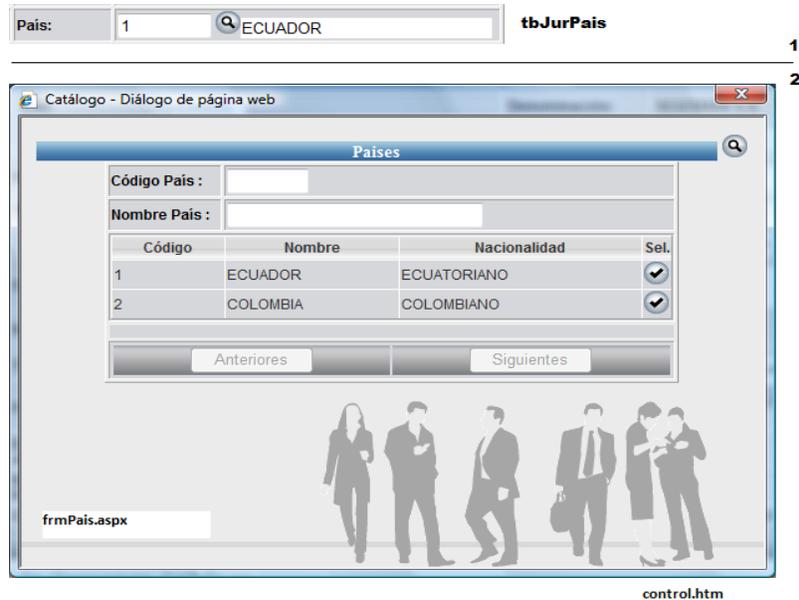


Figura 23. Control Catálogo

Facilidades que presta CatalogoWEB

- ✚ Facilita las búsquedas.
- ✚ Cuando el control de la página se encuentra sobre este UserControl, al presionar la tecla “?” y cuando se hace clic en la imagen del UserControl , aparece la pantalla de búsqueda.
- ✚ Si se digita un código en la primera caja de texto, el evento “OnBlur” lanza la búsqueda de el código digitado, utilizando un webService que debe incluirse en el mismo proyecto. Y trae la descripción (en caso de existir) mostrándola en la segunda caja de texto. En caso de no encontrar la descripción, se presenta las palabras “NO ENCONTRADO...”.

CAPITULO III. VISIONADO Y DISEÑO

La metodología MSF Agile consta con varios documentos para su desarrollo tales como: documento de visión, documento de arquitectura, documento de escenarios, documento de vulnerabilidades y documento de pruebas.

3.1. VISION

3.1.1. Factores de Incidencia

- ✚ Administración de Cuentas, Parámetros.
- ✚ Control de Bienes.
- ✚ Reportes.

3.1.2. Visión del Proyecto

El Sistema Web de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control (SACBIM), cumplirá con las necesidades que el Municipio del Distrito Metropolitano de Quito (MDMQ) requiere para su desarrollo y su implementación será realizada con tecnología de punta para que su funcionamiento sea óptimo y ayude a solucionar los problemas en la Administración de Bienes Municipales.

Con la estandarización del MDMQ para todos sus proyectos, se podrá añadir más procesos que interactúen con la Administración de Bienes Municipales y así formar un sistema robusto y eficiente.

3.2. DEFINICION DE PERSONAS

El conocimiento, capacidad y habilidad que debe tener el usuario administrador y el usuario MDMQ se define a continuación:

- ✚ **Usuario Administrador:** Es la persona quien cuenta con los permisos para poder administrar los datos del Sistema.
- ✚ **Usuario MDMQ:** Es la persona que cuenta con los permisos necesarios para el manejo de Control de Bienes.

El objetivo, motivo e interés que debe tener el usuario administrador y el usuario MDMQ se define a continuación:

- ✚ **Usuario Administrador:** Realizar la Administración de los Bienes del MDMQ.
- ✚ **Usuario MDMQ:** Realizar el control de los Bienes Municipales.

3.3. ARQUITECTURA

3.3.1. Descripción del Problema

En la actualidad, el MDMQ, se encuentran en la tarea de realizar un Sistema de Administración de Bienes Municipales ya que el sistema actual no cumple con algunos requisitos fundamentales para su administración, por lo cual se han encontrado algunos problemas tales como:

Algunas características de los Bienes no se encuentran parametrizadas.

- ✚ Existe información inconsistente de los Bienes al momento de realizar inventarios.
- ✚ El código de barras no está integrado con la codificación gubernamental.
- ✚ El sistema actual no se encuentra realizado para que su utilización sea multiempresa.

El Distrito Metropolitano de Quito (DMQ) dispone de un Sistema de Seguridades, Personas y Auditoria que sirve como herramienta para la administración de personas y base para la implementación de nuevos proyectos, a su vez cambia la metodología existente y establece un estándar para el desarrollo de aplicaciones web (internet o intranet) que usen la plataforma tecnológica instalada. Por lo tanto, el Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control debe seguir con el estándar propuesto por el DMQ e interactuar con el Sistema de Seguridades, Personas y Auditoria para su implementación.

3.3.2. Requerimientos y Alcance del Proyecto

3.3.2.1. Requerimientos

3.3.2.1.1. Requerimientos de Negocio

La administración de la información de Bienes Municipales en forma coherente y consolidada, se debe considerar el tipo de Bien Municipal que se debe ingresar.

Elaboración de procesos que se utilice para la Administración y control de los Bienes Municipales.

3.3.2.1.2. Requerimientos de Usuario

El desarrollo de la Aplicación debe ser en un ambiente amigable y fácil de usar para el usuario.

Deberá ser realizado mediante la Web para facilitar el acceso desde navegador o browser al usuario.

Debe ser parametrizable para que al usuario le facilite la interacción con el sistema.

3.3.2.1.3. Requerimientos de Sistemas

✚ Aplicación multiempresa parametrizable.

El Sistema debe ser distribuido y en la Web para que los usuarios de diferentes zonas municipales puedan ingresar e interactuar con la aplicación.

✚ Generación de códigos de barra por tipo de bien.

El sistema debe generar un código de barra secuencial ascendente por clase, en el que se integrará la cuenta contable, la clase del bien, la variedad y el ordinal. En el caso de bienes compuestos se debe generar un código de barra general para identificar al bien como tal y además generar códigos relacionados con el general para identificar específicamente a cada una de las partes y accesorios de dichos bienes.

✚ Registro de información de nuevos Bienes Muebles Municipales que contemple los siguientes campos:

- Código interno del sistema
- Identificación de Cuentas Contables
- Tipo de Activo: Larga Duración y Sujeto a Control
- Clase (Item)
- Serie
- Modelo
- Marca
- Color
- Estructura
- Material
- Ubicación (Información recursiva hasta cinco niveles),
- Usuario
- Fotografía
- Proveedor
- Número de Factura
- Vida Útil

- Garantías
- Motivo de movimiento
- Estado
- Cantidad
- Características (Campo abierto para características adicionales)
- Fecha de Factura (Fecha de Adquisición)
- Número Acta
- Fecha Acta (Fecha de ingreso al sistema)
- Valor de adquisición (Valor de la Factura)
- Valor Ajustado (Valor luego de una revalorización o erogación)
- Valor Depreciación Acumulada

Solo en caso de vehículos

- Placa
- Disco
- Nro. Chasis
- Nro. Motor

✚ Modificación de información de bienes existentes

Las modificaciones deben contemplar las correspondientes validaciones, controles e historial de todos los cambios realizados.

✚ Administración de Traspasos

El Sistema debe ofrecer la opción de traspasos de bienes en forma individual y grupal dependiendo el tipo del bien es decir que se puede cambiar de custodio un bien o a su vez cambiar todos los bienes de un custodio, Unidad Administrativa o Dirección a otros.

✚ Administración de Bajas

Las Bajas de los bienes deben ser en forma grupal o individual dependiendo del tipo de bien e incluirán validaciones, controles e historial de las mismas.

Proceso de Movimientos Temporales.

Los movimientos temporales deben ser de forma individual o grupal dependiendo el tipo del bien.

Proceso de Erogaciones.

Las erogaciones se aplican a los bienes compuestos, las cuales pueden afectar a los valores contables que un bien puede tener.

Administración de Parámetros.

Se parametriza los valores que interactúan con las características del bien ya que facilitan la administración y control para el usuario.

Proceso de Depreciación.

Las depreciaciones se deben efectuar de manera individual ya que se determinan mediante periodos de tiempo para cambiar los valores contables de los bienes municipales.

Histórico de movimientos.

El histórico de movimientos se debe a si el usuario realizo alguna operación contable al bien municipal y se lo puede utilizar en forma individual o grupal.

3.3.2.1.4. Requerimientos candidatos

La siguiente tabla muestra los requerimientos candidatos de SACBIM que se debe realizar para el desarrollo de un proyecto.

SISTEMA	SUBSISTEMA	PROCESO	PROCEDIMIENTO				
SACBIM	1. Control de Seguridad	1.1 Conexión con Sistema de Seguridades del MDMQ.	1.1.1 Gestión de Usuarios				
			1.1.2 Gestión de Perfiles				
			1.1.3 Gestión de Menú				
			1.1.4 Auditorias				
	2. Control de Bienes	2.1 Ingreso de Bienes	2.1 Ingreso de Bienes	2.1.1 Tipo de Bien a registrar			
				2.1.2 Registrar Bien Municipal			
				2.1.3 Generación código de barras			
				2.1.4 Impresión código de barras			
				2.2 Modificación de Bienes	2.2 Modificación de Bienes	2.2.1 Tipo de Bien a Modificar	
						2.2.2 Búsqueda de Bien Municipal	
						2.2.3 Actualización de Bien Municipal	
						2.2.4 Reimpresión código de barras	
		2.3 Baja de Bienes	2.3 Baja de Bienes	2.3.1 Tipo de Bien escogido			
				2.3.2 Elegir datos de baja			
				2.3.3 Realización del proceso de Baja.			
				2.4 Erogación de Bienes	2.4 Erogación de Bienes	2.4.1 Búsqueda de Bien Municipal	
		2.4.2 Baja de piezas en mal estado					
		2.4.3 ingreso de pieza en mal estado					
		2.4.4 Calculo de presupuesto a comprar					
		2.4.5 Ejecución de Proceso de Erogación.					
		2.5 Depreciación de Bienes	2.5 Depreciación de Bienes	2.5.1 Búsqueda de Bien Municipal			
				2.5.2 Periodo de tiempo de Bien a depreciar			
				2.5.3 Ejecución de proceso de depreciación			
		2.6 Históricos de Movimiento	2.6 Históricos de Movimiento	2.6.1 Búsqueda de Bien			
				2.6.2 Consulta de Información de Bienes			
		2.7 Traspasos de Bienes	2.7 Traspasos de Bienes	2.7 Traspasos de Bienes	2.7.1 Búsqueda de Bien		
					2.7.2 Ingreso de custodio		
					2.7.3 Asignación de Bien Municipal		
					2.7.4 Asignación del lugar del Bien Municipal		
					2.8 Movimientos Temporales	2.8 Movimientos Temporales	2.8 Movimientos Temporales
2.8.2 Ingreso de custodio temporal							
2.8.3 Asignación de Bien Temporal							

			2.8.4 Asignación del Lugar Temporal del Bien.
	3. Reportes de Bienes	3.1 Reportes Parametrizados	3.1.1 Consulta de acuerdo a roles
			3.1.2 Impresión de reportes parametrizados
		3.2 Reportes Gerenciales	3.2.1 Generación de reportes de acuerdo a los proceso (Actas)
			3.2.2 Impresión de reportes gerenciales (Actas)
	4. Administración	4.1 Administración de Parámetros del Sistema	4.1.1 Crear nuevos parámetros para el uso del sistema
			4.1.2 Modificación de parámetros para el uso del sistema
		4.2 Administración de Cuentas del Sistema	4.2.1 Crear nuevas Cuentas para el uso del sistema
			4.2.2 Modificación de Cuentas para el uso del sistema

Tabla 9. Requerimientos Candidatos

3.3.2.2. Alcance del Proyecto

3.3.2.2.1. Objetivos de Negocio

- ✚ Implementar el Sistema de Bienes con la estandarización del Distrito Metropolitano de Quito.
- ✚ Parametrizar las características de los Bienes para una mejor administración.
- ✚ Proporcionar un adecuado control a los Bienes Municipales.
- ✚ Proporcionar una debida codificación de los Bienes propuesta por el departamento Gubernamental.

3.3.2.2.2. Fuera de Alcance en los Objetivos de Negocio

- ✚ Implementación a producción del proyecto SACBIM.
- ✚ Migración total de los Datos para el Sistema de Administración de Bienes de Larga Duración y Bienes Sujetos a Control.
- ✚ Cualquier desarrollo en IBM390.
- ✚ Realizar cambios en los siguientes otros sistemas informáticos del MDMQ como: RRHH y Financiero.
- ✚ Procesos de respaldos o backup de los datos del Sistema.

3.3.2.2.3. Objetivos Técnicos

- ✚ Implementar una metodología de desarrollo (MSF Agile), ya que es utilizada como un estándar en el desarrollo de sistemas informáticos en la MDMQ.
- ✚ Desarrollar el Modulo de Bienes Muebles, basándose en la funcionalidad del Sistema de Seguridades, Personas y Auditoria para adaptarlo a las necesidades del Municipio de Quito.
- ✚ Interactuar con el Sistema de RRHH a nivel de consulta (Vistas).

3.3.2.2.4. Fuera de Alcance en los Objetivos Técnicos

- ✚ Realizar cambios de cualquier información a los sistemas que interactúan con el Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control.
- ✚ Cualquier desarrollo de protocolos de comunicaciones.
- ✚ Operación o Producción del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control, el mismo que estará a cargo del personal del MDMQ.
- ✚ Procesos de respaldos o backup de los datos del Sistema.

3.3.3. Estrategia de Versionamiento o por Etapas

Se utiliza la estrategia de versionamiento que permite una evolución continua para el proyecto a desarrollarse.

La solución del proyecto se ha dividido por 6 etapas las cuales garantizara la funcionalidad del sistema y una evaluación continua, las cuales se describen a continuación:

Primera Etapa.- Implementación de la metodología MSF Agile en la elaboración del proyecto e implementación del diseño estándar en la interfaz de usuario.

Segunda Etapa.- Conexión con el Sistema de Seguridades, Personas y Auditoria. Diseño inicial del estándar de interfaz de usuario. Implementación de plantillas de generación automática de la base de datos y DAL de las transacciones.

Tercera Etapa.- Desarrollo de los diferentes Procesos que realiza la Administración de Bienes.

Cuarta Etapa.- Implementación de Reportes Gerenciales y Parametrizados para el proyecto.

Quinta Etapa.- Implementación de administración de parámetros del sistema.

Sexta Etapa.- Se realizará una recopilación de la documentación técnica de esta fase en el Documento Técnico del Sistema de Administración de Bienes Municipales; además, se prepararán los Manuales de Usuario y Técnicos.

Es importante considerar que en cada etapa se realizará la documentación y las pruebas correspondientes que garanticen la calidad del producto.

3.3.4. Criterio de Aceptación

Para la aceptación del sistema se considerará la entrega de los siguientes elementos:

La entrega de la documentación técnica considerada en la metodología MSF Agile, que se resume en los siguientes elementos.

- ✚ Documento de Arquitectura
- ✚ Documento de Visión del Proyecto
- ✚ Documentos de Personas
- ✚ Documentos de Escenarios
- ✚ Documento de Amenazas y Vulnerabilidades
- ✚ Documento de Pruebas
- ✚ Cronograma de Actividades
- ✚ El cumplimiento de todos los casos de uso que cubran la funcionalidad necesaria para la satisfacción de los procesos.
- ✚ CD que incluye todo el software desarrollado incluyendo Manuales de Usuario y Técnico.

3.3.5. Riesgos

3.3.5.1. Preliminar del Planteamiento

- ✚ Aplicación Multiempresa
 - Falta de seguridad en aplicación de roles
 - Concurrencia de datos

- ✚ Proceso de Ingreso de nuevos Bienes de Larga Duración y Bienes Sujetos de Control Municipales.

- ✚ Mal ingreso de Datos
 - Doble ingreso de Bien Municipal
 - Mala generación de código de barras

- ✚ Modificación de información de Bienes Municipales existentes
 - Reimpresión de código de barras

- ✚ Proceso de Traspasos
 - Se asigne dos Bienes Municipales a un solo custodio
 - No se registre en el sistema el traspaso realizado
 - No se pueda registrar un Custodio ya que no existe en RRHH

- ✚ Proceso de Baja
 - Asignar a un custodio un Bien dado de Baja

- ✚ Movimientos Temporales
 - Asignar un Bien a custodios que ya no trabajan en la empresa

- ✚ Proceso de Depreciación
 - No se establecen los periodos de depreciación de los Bienes

✚ Reportes Parametrizados

- Parámetros para la generación de los reportes Parametrizados

✚ Reportes Gerenciales

- Parámetros para la generación de los reportes Gerenciales.

3.3.5.2. Lista Primaria de riesgos identificados

A continuación se estimara el valor de riesgo inherente el cual consiste en asignar a los riesgos calificaciones dentro de un rango, los cuales son: 1 a 5 (insignificante (1), baja (2), media (3), moderada (4) o alta (5)) el cual se muestra en el siguiente cuadro:

Valoración de riesgo inherente

IMPACTO	ALTO	4	5	5
	MEDIO	3	3	5
	BAJO	1	2	4
		BAJO	MEDIO	ALTO

FRECUENCIA O PROBABILIDAD DE OCURRENCIA

Tabla 10. Valoración de Riesgos Inherentes

La siguiente tabla muestra los riesgos identificados con su respectiva valoración que se debe realizar para el desarrollo de un proyecto.

Nº	Riesgos Identificados	Impacto	Probabilidad	Valor
1	No exista parametrización de las características de los Bienes	Alto	Alto	5
2	Concurrencia de datos de los Bienes	Alto	Medio	5
3	Mal ingreso de Datos	Medio	Medio	3
4	Doble ingreso de Bien Municipal	Alto	Medio	5

5	Mala generación de código de barras	Alto	Bajo	4
6	Reimpresión de código de barras	Medio	Medio	3
7	Se asigne dos Bienes Municipales a un solo custodio	Medio	Bajo	3
8	No se registre en el sistema el traspaso realizado	Alto	Bajo	4
9	Consultas a vistas de sistemas que interactúan fallen	Alto	Alto	5
10	Asignar a un custodio un Bien dado de Baja	Medio	Bajo	3
11	Asignar un Bien a custodios que ya no trabajan en la empresa	Bajo	Bajo	1
12	No se establecen los periodos de depreciación de los Bienes	Medio	Bajo	3
13	Parámetros para la generación de los reportes Parametrizados	Alto	Medio	5
14	Parámetros para la generación de los reportes Gerenciales	Alto	Alto	5
15	Cambio de Fecha para generar el proceso de depreciación	Alto	Medio	5
16	Aplastar varios botones a la vez	Medio	Medio	3
17	Datos erróneos para los asientos contables	Alto	Alto	5

Tabla 11. Riesgos Identificados

La siguiente tabla muestra los riesgos obtenidos en el desarrollo de un proyecto.

No	Riesgo	Descripción	Impacto	Codificación del impacto	Causas	Probabilidad ocurrencia
1	No exista parametrización de las características de los Bienes	Se puedan añadir más características de los Bienes y al momento del ingreso no se las puede identificar.	<ul style="list-style-type: none"> Datos erróneos de un Bien. Inconsistencia de Datos. 	Alto	No se establecen correctamente los requerimientos para el sistema.	Alto
2	Concurrencia de Datos	Existen demasiados datos en la red al mismo tiempo	<ul style="list-style-type: none"> Los datos colisionan y no se pueden actualizar datos Genera error en la base de datos. Se cuelga el Sistema. 	Alto	Dos o mas usuarios pueden utilizar el sistema al mismo tiempo.	Medio
3	Mal ingreso de datos	Digitación errónea de los Bienes Municipales	<ul style="list-style-type: none"> Mala impresión de código de barras. Mala asignación de Categoría de los Bienes. 	Medio	Error por parte del supervisor al realizar el ingreso de los Bienes al Sistema.	Medio
4	Doble ingreso del Bien Municipal	Ingreso de un Bien Municipal ya existente en el sistema	<ul style="list-style-type: none"> Existe duplicación del mismo Bien en el sistema. Doble asignación de nomenclatura en el sistema para el mismo Bien. 	Alto	Descuido por parte del supervisor al momento de ingresar un Bien.	Medio
5	Mala generación de código de barras	Se puede generar un código erróneo de barras.	<ul style="list-style-type: none"> No existe lectura de barras. El Bien no existe en Sistema 	Alto	Mal ingreso de datos al Sistema. Error al categorizar dicho Bien Municipal.	Bajo
6	Reimpresión de código de Barras	Es posible que el código de Barras que fue imprimido anteriormente se deteriore.	<ul style="list-style-type: none"> No es posible la lectura de su código de barras. No se podrá determinar qué Bien y a que custodio pertenece. 	Medio	Los usuarios que realicen el inventario tendrán problemas al momento de verificar cada Bien Municipal.	Medio
7	Se asigne dos Bienes Municipales a	Traspaso de un Bien a un custodio dos o más veces.	<ul style="list-style-type: none"> Sobrescribir la información del Bien que posee el custodio. 	Medio	Descuido al momento del Traspaso de un Bien, ya que se puede	Bajo

	un solo custodio		<ul style="list-style-type: none"> Desconocer que Bien Municipal le pertenece a dicho custodio. 		asignar un Bien a un custodio que ya fue asignado anteriormente.	
8	No se registre en el sistema el traspaso realizado	Al momento de cambiar un Bien Municipal de custodio se puede omitir su cambio al sistema y puede generar error al momento de buscar dicho Bien.	<ul style="list-style-type: none"> El Bien que fue asignado a un nuevo custodio no esté ingresado en sistema. Existe problema al momento de realizar un inventario. 	Alto	Descuido por parte del supervisor al momento de ingresar o modificar un nuevo custodio.	Bajo
9	Consultas a Sistemas que interactúan fallen	No se puedan consultar datos necesarios para el Sistema.	<ul style="list-style-type: none"> Consulta de personas que trabajan en el municipio. Consulta de valores de Asientos Contables 	Alto	Fallo en la conexión con los sistemas que interactúan con el Sistema de Bienes Municipales.	Alto
10	Asignar a un custodio un Bien dado de Baja	Asignar por negligencia un Bien inservible a un custodio	<ul style="list-style-type: none"> En el sistema dicho Bien pertenece a un custodio y dicho Bien no existe. 	Medio	Al momento que genere reportes tendrán problemas al momento de buscar dicho Bien Municipal.	Bajo
11	Asignar un Bien a custodios que ya no trabajan en la empresa	Asignar por negligencia un Bien a un custodio que no existe.	<ul style="list-style-type: none"> Se asigna un Bien Municipal a un custodio que no trabaja en la empresa 	Bajo	Falta de actualización de datos en el sistema de RRHH.	Bajo
12	No se establecen los periodos de depreciación de los Bienes	No se establece periodos de la depreciación de un Bien Municipal.	<ul style="list-style-type: none"> No se puede calcular la depreciación de un Bien. Un Bien no se deprecia nunca. 	Medio	No se coloca una formula consistente que garantice la depreciación de un Bien Municipal.	Bajo
13	Parámetros para la generación de los reportes Parametrizados	Se debe establecer los parámetros necesarios para generarlos.	<ul style="list-style-type: none"> Los reportes que se imprimen no son los deseados. 	Alto	No se establecen los parámetros de búsqueda para los reportes.	Medio
14	Parámetros	Se debe establecer	<ul style="list-style-type: none"> Los reportes que se 	Alto	No se establecen los	Alto

	para la generación de los reportes Gerenciales	los parámetros necesarios para generarlos.	imprimen no son los deseados.		datos correctos para los reportes.	
15	Cambio de Fecha para generar el proceso de depreciación	Al cambiar la fecha del sistema se podría haber confusión en el sistema para el proceso de depreciación.	<ul style="list-style-type: none"> No está bien configurado la hora del sistema. Confusión de periodos de tiempo 	Alto	Mala configuración regional del sistema	Medio
16	Aplatar varios botones a la vez	Colgar el sistema y no efectuar el proceso correctamente.	<ul style="list-style-type: none"> No se efectúen algunos procesos establecidos. 	Medio	Fallo de la base de datos o de algún proceso en el sistema.	Medio
17	Datos erróneos para los asientos contables.	Fallo interno del sistema o falta de datos en la descripción del Bien Municipal.	<ul style="list-style-type: none"> Inconsistencia de Datos. No existe un balance con las cuentas contables. 	Alto	<ul style="list-style-type: none"> Falta de validaciones en el Sistema. Cálculos mal realizados en el Sistema 	Alto

Tabla 12. Riesgos Totales

3.3.5.3. Cálculo de Riesgo Neto o Residual

Para el cálculo del riesgo neto ocuparemos la efectividad de cada control descrito para la solución de los riesgos propuestos.

Control	Efectividad
Ninguno	1
Bajo	2
Medio	3
Alto	4
Destacado	5

Tabla 13. Variables para Cálculo Neto o Residual

El proyecto cuenta con el siguiente Riesgo Neto para el desarrollo del Sistema de Bienes Municipales:

La siguiente tabla muestra el cálculo de riesgo neto o residual para el desarrollo de un proyecto SACBIM.

		Calidad de Gestión				
Nº	Actividades	Nivel Riesgo	Controles	Efectividad	Promedio (*)	Riesgo Residual(**)
1	No exista parametrización de las características de los Bienes	5	Crear parámetros para las características de los Bienes.	4	3,5	1,42
			Identificar correctamente las características de los Bienes Municipales.	3		
2	Concurrencia de datos	5	Realizar restricciones para usuarios que estén ocupando el sistema	3	3	1,66
			Exista fiabilidad en la ejecución del Programa	4		
			Sincronización de procesos	2		
3	Mal ingreso de Datos	3	Validación de datos	3		

			Mascaras en los datos a ingresar	3	2,6	1,15
			Verificaciones de longitudes y caracteres	2		
4	Doble ingreso de Bien Municipal	5	Comparación del Bien que se desea ingresar con los Bienes existentes en la empresa	2	2	2,5
5	Mala generación de código de barras	4	Categorizar según sea el Bien y que nunca se repita la codificación del Bien Municipal	3	3	1,33
6	Reimpresión de código de barras	3	Búsqueda de Bien para la reimpresión de código de barra	4	4	0,75
7	Se asigne dos Bienes Municipales a un solo custodio	3	Validación de Bienes asignados a custodios	5	5	0,6
8	No se registre en el sistema el traspaso realizado	4	Mensajes de avisos al momento que se realice una operación.	4	4	1
9	No se pueda registrar un Custodio ya que no existe en RRHH	5	Consultar la existencia de una persona en la Empresa	3	2,5	2
			Ingresar nuevo custodio temporal	2		
10	Asignar a un custodio un Bien dado de Baja	3	Validación de Bienes de Baja	3		
			Actualización de Bienes Municipales	2	2,5	1,2
11	Asignar un Bien a custodios que ya no trabajan en la empresa	1	Búsqueda en RRHH de personal de la empresa	4		
			Validación de personas que trabajan en la empresa	3	3,5	0,28
12	No se establecen los periodos de depreciación de los Bienes	3	Parametrizar los periodos de depreciación de los Bienes	3	3	1
13	Parámetro para la generación de los reportes parametrizados	5	Criterio de Búsqueda en reportes parametrizados	5		
			Autenticación para reportes parametrizados	5	5	1
			Impresión para reportes parametrizados	5		
14	Parámetro para la generación de los reportes Gerenciales	5	Asignación correcta de datos en reportes gerenciales	5		
			Autenticación para reportes gerenciales	5	5	1
			Impresión para reportes gerenciales	5		
15	Cambio de Fecha para generar el proceso de depreciación	5	Verificación de fecha en el Sistema en la pantalla de Depreciación	4	4	1,25

16	Aplastar varios botones a la vez	3	Validaciones en la solución	4		
			Dividirse por paneles para llenar datos	3	3,5	1,42
17	Datos erróneos para los asientos contables.	5	Verificación ingreso de datos	4	3,5	1,42
			Verificación de Cálculos de valores de los Bienes.	3		
Perfil de riesgo (Riesgo residual total) (***)						1,23

(*) Promedio de los datos de efectividad

(**) Resultado de la división entre nivel de riesgo / Promedio de efectividad

(***) Promedio: Se considera un mismo peso de ponderación a los RI.

Tabla 14. Calculo Valor Neto o Residual

3.3.6. Estructura del Proyecto

Posiciones y Responsabilidades

El desarrollo de este proyecto será realizado por la metodología MSF agile la misma que será detallada mediante este documento.

Este proyecto ha sido realizado utilizando la metodología MSF agile la misma que identifica los siguientes roles y responsabilidades requeridas durante las iteraciones que se requieren para la solución del proyecto.

La estructura organizacional es la siguiente:

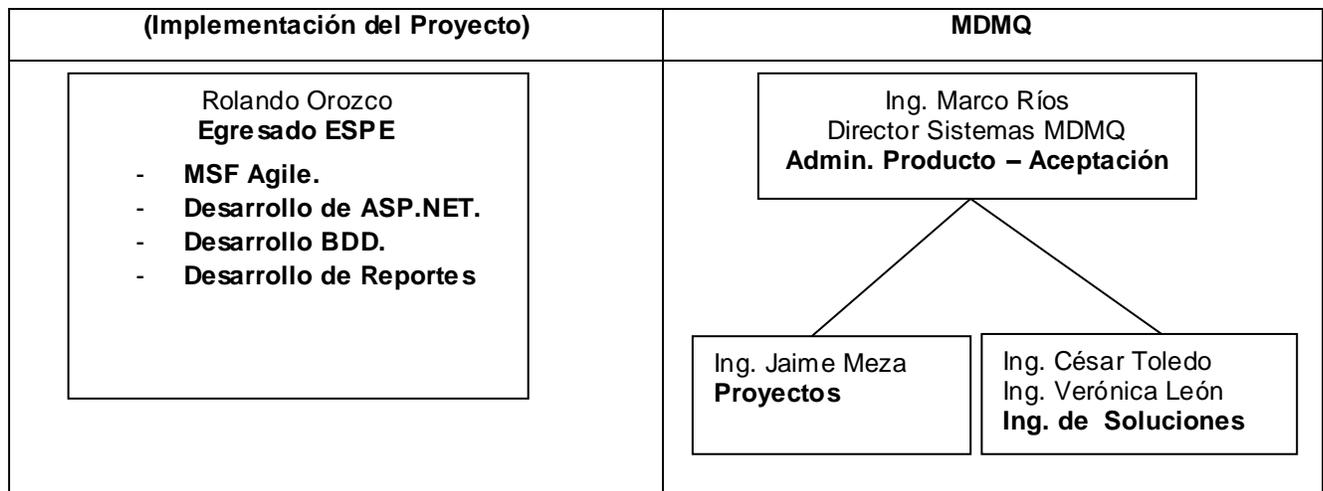


Tabla 15. Estructura del Proyecto

3.3.7. Diseño de la Solución

3.3.7.1. Inventario de Casos de Uso

3.3.7.1.1. Caso 1: Control de Seguridad

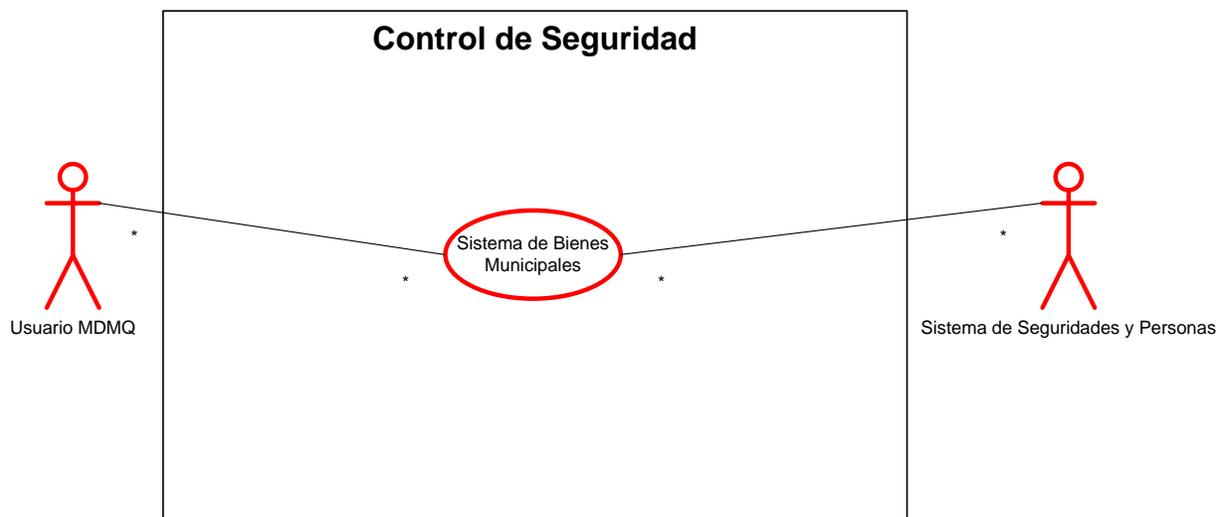


Figura 24. Caso de Uso Control de Seguridad

Actores:

Usuario-MDMQ	Persona que trabaja en el MDMQ y que necesita realizar un proceso.
Sistema de Seguridades y Personas	Sistema con el cual el proyecto debe interactuar para realizar las seguridades requeridas.

Tabla 16. Actores Control Seguridad

Descripción del Proceso

El usuario realiza toda la gestión de seguridades que debe tener un sistema para que sea confiable, íntegro, óptimo y eficiente.

Actores de Entrada de Datos.

1. Usuario-MDMQ

Actores de Salida de Datos

1. Usuario-MDMQ
2. Sistema de Seguridades y Personas.

Precondiciones

1. El usuario está autenticado y autorizado en el Sistema de Personas.
2. El menú debe estar generado mediante el Sistema de Personas.

Flujo Básico de Eventos

1. Se valida si el Usuario-MDMQ tiene las autorizaciones necesarias para poder ejecutar la acción que desea realizar en el Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control.
2. Dependiendo del Rol que tenga el Usuario-MDMQ se podrá realizar diferentes acciones.
3. Se realiza gestión de Auditoria para el sistema de bienes.

Flujos Alternativos

1. La conexión del Sistema de Personas será implícitamente para consulta.

Flujo de Excepción

1. Si el usuario no es autenticado.
 - a. Emitir un mensaje y no permitir el ingreso a la aplicación
2. Si el usuario no está autorizado a ejecutar una transacción.
 - a. Emitir un mensaje y no permitir la ejecución de la transacción

Resultado

El presente caso de uso genera como resultado el proceso de seguridades de toda la aplicación de Bienes Municipales.

3.3.7.1.2. Caso 2: Control de Bienes

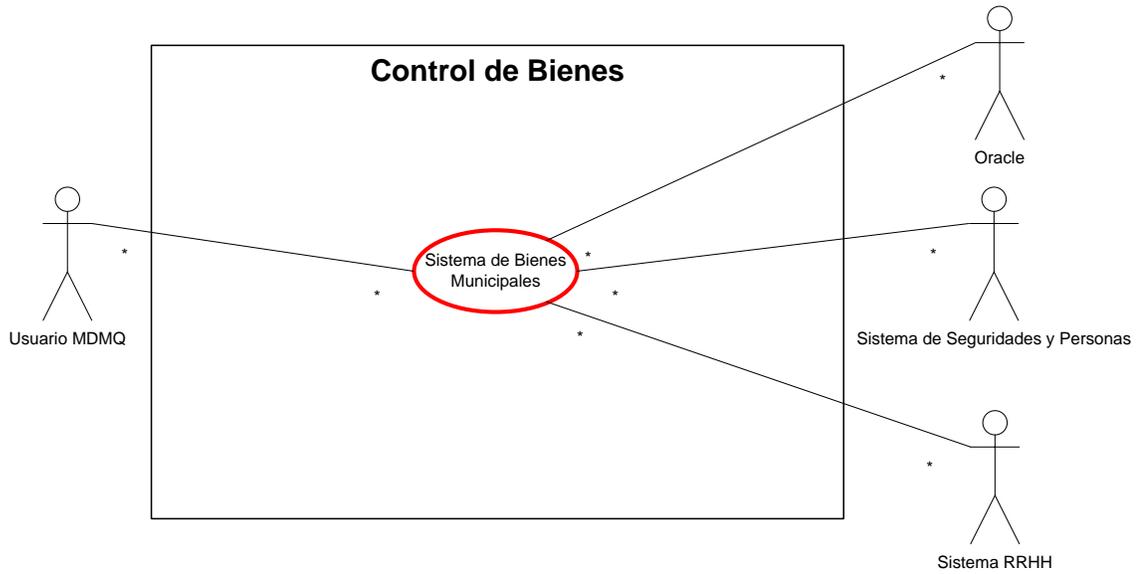


Figura 25. Caso de Uso Control de Bienes

Actores:

Usuario-MDMQ	Persona que trabaja en el MDMQ y que necesita realizar un proceso.
Sistema de Seguridades y Personas	Sistema con el cual el proyecto debe interactuar para realizar las seguridades requeridas.
ORACLE10g	Base de datos.
Sistema RRHH	Se utiliza como consulta de información de la ubicación de los Bienes Municipales y custodios.

Tabla 17. Actores Control Bienes

Descripción del Proceso

El usuario necesita realizar varios subprocesos para la Administración y Control de Bienes Municipales tales como: Ingreso, Modificación, Dar de Baja, Erogación, Depreciación, Movimientos Temporales, Histórico de un Bien y Traspasos de Bienes.

Actores de Entrada de Datos.

1. Usuario-MDMQ

Actores de Salida de Datos

1. Sistema de Seguridades y Personas
2. ORACLE10g
3. Usuario-MDMQ
4. Sistema RRHH

Precondiciones

Subproceso Ingreso

- ✚ Proveedor entrega el Bien Municipal solicitado con su respectiva documentación (factura y adjudicación o contrato).
- ✚ Analista Administrativo verifica que las características del Bien Municipal que fue entregado sea el mismo con lo que solicitó anteriormente y que la documentación sea la correcta.

Subproceso Modificación

- ✚ El Bien Municipal debe estar ingresado en el repositorio de datos.

Subproceso Dar de Baja

- ✚ Custodio comunica al Departamento de Bienes que el bien está inservible u obsoleto.
- ✚ Analista Administrativo verifica el estado del bien.
- ✚ Si fuere necesario el Analista Administrativo solicita un informe técnico a quién corresponda.
- ✚ Analista Administrativo realiza el traspaso del bien, a la bodega de bienes.
- ✚ Cuando existen varios bienes de similares condiciones se solicita la baja de dichos Bienes Municipales.

- ✚ Concejo de Bajas aprueba la solicitud de baja, especificando el destino de los bienes, es decir si van hacer: donados, destruidos o se van al remate.

Subproceso Erogación

- ✚ Analista Administrativo con la adjudicación y factura registra la erogación.
- ✚ Analista Administrativo imprime y legaliza el acta de erogación.

Subproceso Histórico de Movimiento

- ✚ Realizar cualquier proceso anterior para determinar el histórico del bien.

Subproceso Traspaso de Bienes

- ✚ Custodio actual del bien llena el formulario: Control de Movimientos de Activos Fijos.
- ✚ El formulario es legalizado por el custodio actual y el nuevo custodio.

Subproceso Movimientos Temporales

- ✚ Se asigna el acta de Control de Movimientos de Activos Fijos.
- ✚ El formulario es legalizado para el movimiento temporal del Bien Municipal.

Flujo Básico de Eventos

Subproceso Ingreso

- ✚ Analista Administrativo registra las características propias de cada Bien Municipal adquiridos por el MDMQ.
- ✚ Sistema genera el código de barra correspondiente de acuerdo a la ley gubernamental.
- ✚ Analista Administrativo imprime el acta entrega /recepción.
- ✚ Analista Administrativo, el Jefe de Bienes y Responsable legalizan el acta que fue imprimida.
- ✚ Analista Administrativo imprime y coloca físicamente el código de barra al Bien Municipal.
- ✚ Analista Financiero consulta Bienes de larga duración y Bienes Sujetos de Control ingresados físicamente.
- ✚ Analista Financiero realiza la conciliación física.

- ✚ Asignación de un Custodio al Bien Ingresado.

Subproceso Modificación

- ✚ Ingreso Físico del código de barras al Bien Municipal que se desea modificar.
- ✚ Modificar datos no relevantes con el código de barras que fue generado anteriormente.
- ✚ Reimpresión de código de barras.
- ✚ Actualización de datos cambiados en el repositorio de datos del MDQM.

Subproceso Dar de Baja

- ✚ Analista Administrativo da de baja el bien en el sistema, seleccionando el destino del Bien Municipal
- ✚ Analista Administrativo imprime Acta de Baja.
- ✚ Jefe de Bienes, Analista Administrativo y Responsable legalizan el acta.
- ✚ Analista Financiero consulta bienes de larga duración y bienes sujetos dados de baja físicamente.

Subproceso Erogación

- ✚ Analista Financiero consulta bienes de larga duración que fueron erogados.
- ✚ Analista Financiero realiza la erogación contablemente.

Subproceso Depreciación

- ✚ Jefe de bienes realiza la depreciación.
- ✚ Analista Financiero consulta Bienes Municipales depreciados.

Subproceso Histórico de Movimiento

- ✚ Consulta de Bienes Municipales
- ✚ Funciones realizadas anteriormente por diferentes procesos
- ✚ Impresión de Información de Bienes Municipales.

Subproceso Traspaso de Bienes

- ✚ Analista Administrativo registra el cambio de custodio.
- ✚ Imprime el acta de traspaso.
- ✚ Legaliza el acta.

Subproceso Movimientos Temporales

- ✚ Analista Administrativo registra el cambio temporal del Bien Municipal.
- ✚ Imprime el acta de traspaso Temporal.
- ✚ Legaliza el acta.

Flujos Alternativos

Subproceso Ingreso

El ingreso de un bien se puede realizar por los siguientes motivos:

Por Adquisición.- Cuando el DMQ adquiere un Bien Municipal.

Por Transferencia Gratuita.- Caso similar al de adquisición, con la variante de que el proveedor puede ser un ente contable o una empresa y que el valor de un Bien Municipal es una estimación con base en el valor de mercado.

Reposición.- El bien ingresa con las características y valor correspondientes a un Bien Municipal perdido o extraviado.

Erogación Capitalizable.- Consiste cuando un Bien Municipal adquirido aumenta la capacidad productiva, esto puede ser con la vida útil estimada o con su valor.

Existe la condición de preexistencia.

Por constatación física (sobrantes).- Luego de una constatación física existen Bienes Municipales que no fueron registrados, por lo cual se realiza el ingreso con un valor estimado.

Subproceso Modificación

La Modificación de un Bien Municipal se puede realizar por los siguientes motivos:

Por Cambio de Datos.- Se puede modificar un Bien Municipal cuando los datos no modifiquen el código de barra generado.

Por Pérdida o Gasto de Código de Barras.- En el caso que un código de barras este deteriorado por el tiempo, se puede generar una reimpresión de su respectivo código del Bien Municipal.

Subproceso Dar de Baja

La baja de un bien también se puede dar por los siguientes motivos:

- ✚ Transferencia Gratuita.
- ✚ Siniestro.

- ✚ Constatación Física.

En los cuales se procede a realizar la baja sin necesidad de realizar primero el traspaso. El custodio puede solicitar un informe técnico y adjuntar a su informe para dar de baja a un Bien Municipal.

Subproceso Movimientos Temporales

- ✚ Un custodio requiere un Bien Municipal por determinado tiempo.

Flujo de Excepción

Subproceso Ingreso

- ✚ Si las características del Bien Municipal que recibe no coinciden con las características del Bien Municipal solicitado, se procede a devolver el Bien para el cambio correspondiente.
- ✚ Si la documentación presenta errores, se procede a devolver al proveedor para que realice las correcciones necesarias.

Subproceso Modificación

- ✚ Pueden existir errores al momento del ingreso de un Bien Municipal, por consiguiente para la modificación es necesario verificar sus características con la documentación adquirida.

Subproceso Dar de Baja

- ✚ Si el concejo de bajas no aprueba la baja del Bien Municipal seguirá en funcionamiento hasta que ocurra lo contrario.

Subproceso Erogación

- ✚ El cálculo de la erogación es incorrecta.

Subproceso Traspaso de Bien

- ✚ No se define el destino del Bien Municipal.

Subproceso Movimientos Temporales

- ✚ No se define el destino del Bien Municipal.

Resultado

El presente caso de uso genera como resultado toda la Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control.

3.3.7.1.3. Caso 3: Reportes de Bienes

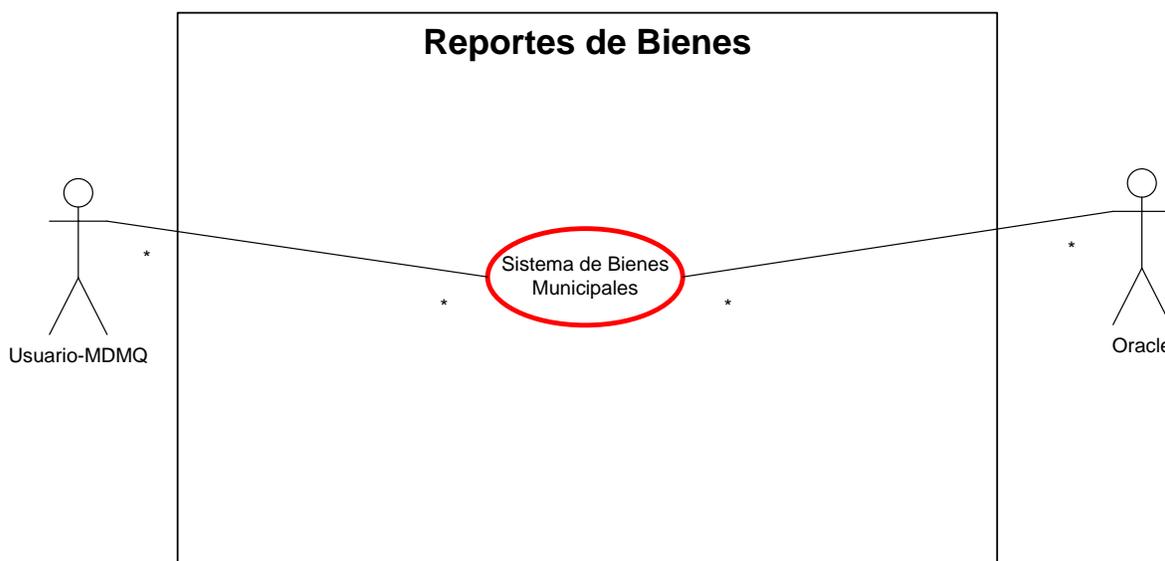


Figura 26. Caso de Uso Reportes de Bienes

Actores:

Usuario-MDMQ	Persona que trabaja en el MDMQ y que necesita realizar un proceso.
Oracle	Base de Datos del Sistema

Tabla 18. Actores Reportes de Bienes

Descripción del Proceso

El usuario realiza el proceso de reportes parametrizables y gerenciales de acuerdo a parámetros determinados por el MDMQ.

Actores de Entrada de Datos.

1. Usuario-MDMQ

Actores de Salida de Datos

1. Usuario-MDMQ
2. Oracle

Precondiciones

1. Realizar cualquier subproceso del caso de uso control de Bienes.

Flujo Básico de Eventos

1. Usuario –MDMQ escoge que tipo de reporte desea.
2. Usuario-MDMQ selecciona con que parámetro se va a generar los reportes.
3. Usuario – MDMQ imprime los reportes generados por el sistema.

Flujos Alternativos

1. Se generara reportes pequeños dependiendo de su funcionalidad.

Flujo de Excepción

1. No se imprime con los datos deseados.

Resultado

El presente caso de uso genera como resultado el proceso de generación de reportes parametrizados y gerenciales.

3.3.7.1.4. Caso 4: Administración de Bienes

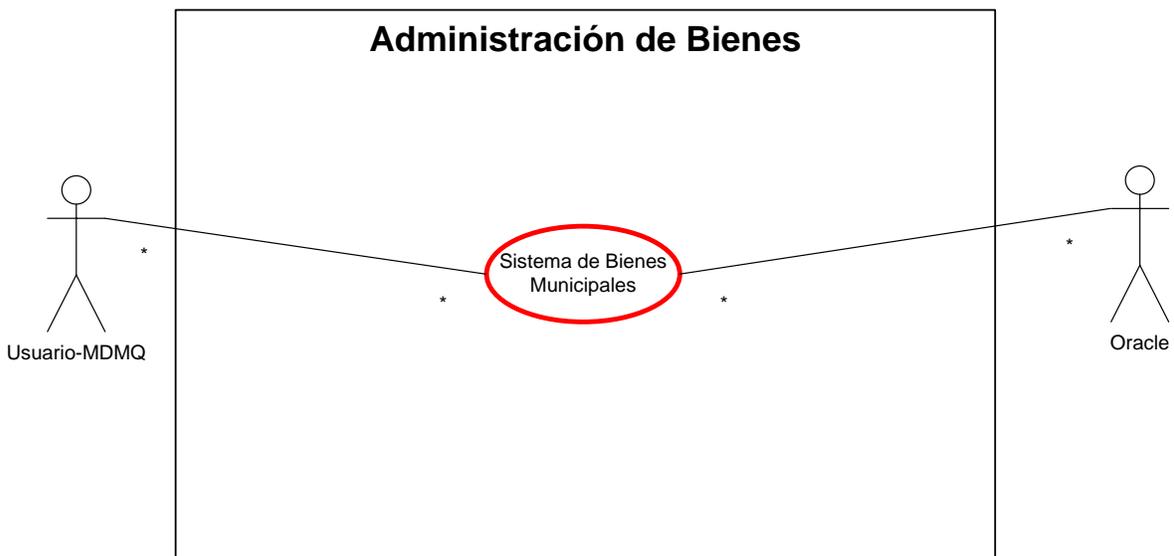


Figura 27. Caso de Uso Reportes de Bienes

Actores:

Usuario-MDMQ	Persona que trabaja en el MDMQ y que necesita realizar un proceso.
Oracle	Base de Datos del Sistema

Tabla 19. Actores Administración de Bienes

Descripción del Proceso

El usuario realiza el proceso de Administración de Cuentas y Parámetros que serán utilizados en el sistema por el personal del MDMQ.

Actores de Entrada de Datos.

1. Usuario-MDMQ

Actores de Salida de Datos

1. Usuario-MDMQ
2. Oracle

Precondiciones

Subproceso de Administración de Cuentas

Analista Administrativo ingresa las cuentas de los bienes simples y compuestos según su codificación y nivel de cuenta.

Subproceso de Administración de Parámetros

Analista Administrativo realiza las operaciones necesarias para la administración de los parámetros del sistema los cuales son: Accesorios, Color, Estado, Estructura, Tipo del Bien, Motivo de Movimiento, Tipo de Movimiento, Tipo de Cuenta y Uso.

Flujo Básico de Eventos

Subproceso de Administración de Cuentas

1. Usuario-MDMQ escoge que el tipo de cuenta a ingresar.
2. Usuario-MDMQ ingresa datos de cuenta.

Subproceso de Administración de Parámetros

1. Usuario-MDMQ escoge el parámetro para realizar las operaciones del Sistema.

2. Usuario-MDMQ ingresa datos dependiendo del parámetro escogido.

Flujos Alternativos

1. Se ingresaran los datos dependiendo del tipo de cuenta y parámetro escogido.

Flujo de Excepción

1. No se ingresan los datos correctamente.

Resultado

El presente caso de uso genera como resultado el proceso de administración de cuentas y parámetros para uso del sistema.

3.3.7.2. Diseño General de la solución

El diseño general de la solución requiere la implementación de los componentes descritos en el siguiente gráfico:

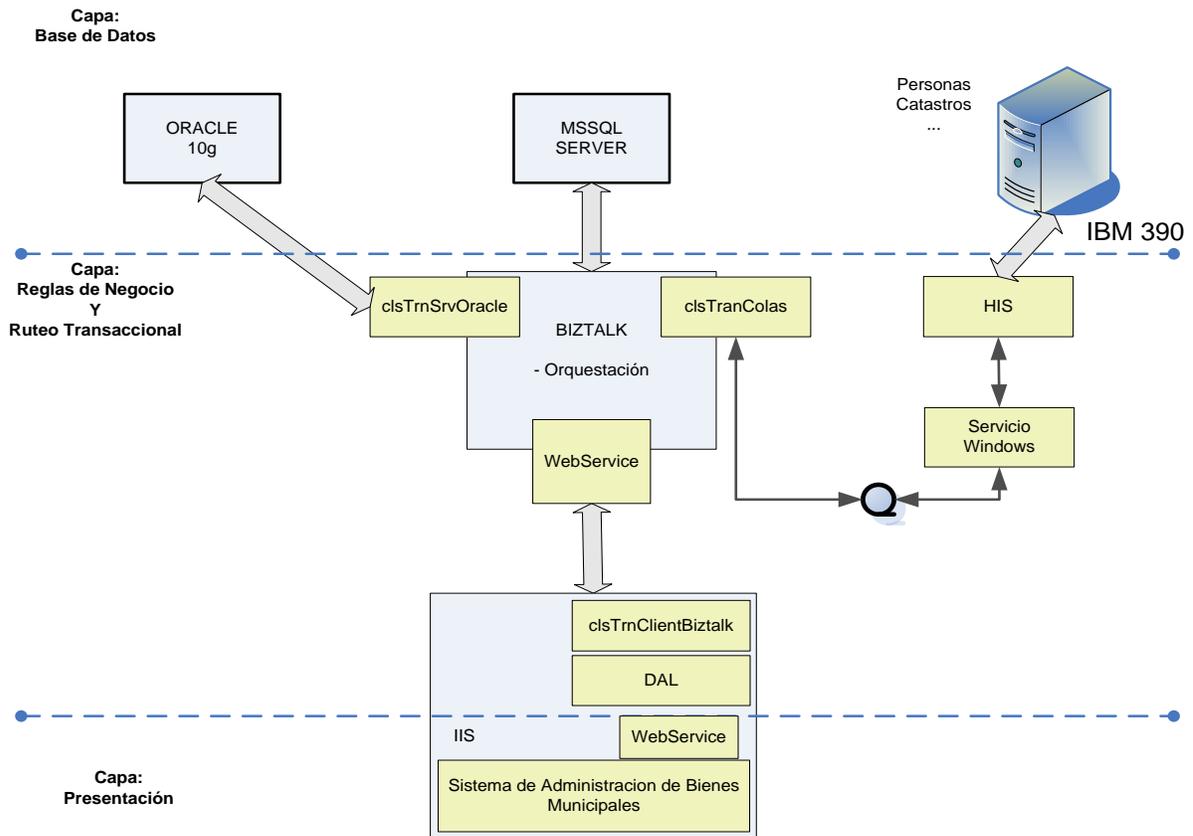


Figura 28. Arquitectura de la Solución

Los elementos que constituyen esta solución son los siguientes:

#	Nombre	Descripción
1	Oracle 10g	Base de datos donde se van a guardar los datos del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos
2	MSSQL Server	Base de datos para que funcione BizTalk.
3	IBM 390	Servidor de aplicaciones de MDMQ.
4	BizTalk	Actúa como centro para conectar varios sistemas diferentes a través de un sistema de mensajes y orquestación basado en XML.
5	clsTmSrvOracle	Componente que llama a BizTalk para ejecutar las transacciones en la base de datos Oracle.
6	COMTI	Componentes de comunicación para el intercambio de información del el servidor IBM 390.
7	WebService	Servicio web donde se publican las orquestaciones BizTalk.
8	IIS	Internet Information Services. Servidor Web de Microsoft.
9	clsTmClientBiztalk	Componente que permite a las aplicaciones cliente ejecutar sus transacciones a través del Web Service publicado por BizTalk
10	DAL	Data Access Layer – Capa de acceso a datos.
11	Sistema de Administración y Control de Bienes Municipales	Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control del Municipio de Quito.

Tabla 20. Elementos de la Solución

3.3.7.3. Descripción detallada de Procesos

Proceso de Seguridades

Entrada: Ingresar datos de Usuario – MDMQ

Salida: Seguridades del Sistema

Explicación del Proceso: Este proceso consta en la conexión con el Sistema de Personas, Seguridades y Auditorias para poder realizar la autorización a diferentes Usuarios-MDMQ.

Se basa principalmente en la autenticación de Usuarios para la utilización del Sistema.

Este proceso realizara todas las operaciones que un sistema de seguridades realice para utilización del sistema.

Proceso de Ingreso del Bien Municipal

Entrada: Se ingresa datos de un Bien adquirido por el MDMQ tales como:

- + Código interno del sistema
- + Identificación de Cuentas Contables
- + Tipo de Activo: Activo Fijo y Sujeto a Control
- + Clase (Item)
- + Serie
- + Modelo
- + Marca
- + Color
- + Estructura
- + Material
- + Ubicación (Información recursiva hasta cinco niveles) ,
- + Usuario
- + Fotografía
- + Proveedor
- + Número de Factura
- + Vida Útil
- + Garantías
- + Motivo de movimiento
- + Estado
- + Cantidad
- + Características (Campo abierto para características adicionales)
- + Fecha de Factura (Fecha de Adquisición)
- + Número Acta
- + Fecha Acta (Fecha de ingreso al sistema)
- + Valor de adquisición (Valor de la Factura)
- + Valor Ajustado (Valor luego de una revalorización o erogación)
- + Valor Deprecación Acumulada

Solo en caso de vehículos

- + Placa
- + Disco
- + Nro. Chasis
- + Nro. Motor

Salida: Registros de un Bien Municipal ingresado en la Base de Datos.

Explicación del proceso: El proceso de ingreso de Bienes Municipales se lo realiza con la interfaz de ingreso de Datos, el cual ha sido diseñado por herramientas de plantillas.

La interfaz de ingreso consta con varios datos parametrizados para agilizar el ingreso de un Bien Municipal al usuario – MDMQ.

Dependiendo de las características como es en el caso de un computador se podrá ingresar solo los ítems correspondientes a esa categoría.

Este archivo será procesado y se lo subirá a la base de datos de Bienes (que se encuentra en Oracle 10g).

Al momento en que se está subiendo la información a la base de datos, se debe controlar las validaciones respectivas para el ingreso de Bienes, obteniendo un mensaje satisfactorio cuando el proceso haya llegado a su fin.

Se generará automáticamente el código de barras con la codificación gubernamental, y se podrá imprimir para su respectivo Bien Municipal.

Proceso de Modificación de Bienes Municipales

Entrada: Código de Barras, Código del Bien.

Salida: Bien Municipal modificado.

Explicación del proceso: El proceso de Modificación se lo realiza con la interfaz de modificación de Bienes Municipales. Tomara como búsqueda el código de Barras del Bien Municipal.

Detallará todas las características de dicho Bien y se lo podrá modificar características del Bien el cual modificara el código de Barras y se lo podrá imprimir de nuevo.

En el caso que el código de Barras este distorsionado o no se pueda entender su codificación, se podrá realizar la búsqueda por medio de otros parámetros, y al momento de localizar dicho Bien, el sistema tendrá la opción de reimprimir su código de barras.

Proceso de Dar de Baja a un Bien Municipal

Entrada: Código de Barras, Código del Bien.

Salida: Bien Municipal Dado de Baja

Explicación del proceso: Este proceso se lo realiza en el caso que un Bien ya no esté apto para su utilización y que se deba dar de Baja por alguno de los siguientes motivos.

- ✚ Transferencia Gratuita
- ✚ Siniestro
- ✚ Constatación Física

El Sistema debe dar un informe sobre la Baja del Bien Municipal. Una vez que el Bien Municipal haya sido dado de Baja su estado debería cambiar.

Proceso de Erogación

Entrada: Código de Barras, Código del Bien.

Salida: Erogación de un Bien Municipal

Explicación del proceso: El proceso de erogación consta que una de las partes de un Bien Municipal necesita ser cambiada.

El precio total de un Bien Municipal debe ser erogado para poder obtener el precio actual del Bien.

Proceso de Depreciación

Entrada: Rango de fecha

Salida: Bienes Depreciados y Bienes no depreciados

Explicación del proceso: El proceso de depreciación obtendrá un rango de tiempo de la base de datos. El rango de tiempo será asignado automáticamente según se haya depreciado los bienes en un periodo de tiempo. El sistema obtendrá el número de bienes depreciados y bienes no depreciados.

Proceso de Histórico de Movimiento

Entrada: Código de Barras, Código del Bien.

Salida: información de Bienes Municipales

Explicación del proceso: El proceso consta de la información de un Bien Municipal, al momento de realizar el proceso de Ingreso de Bienes un trigger insertara la información a la tabla de Histórico y la cual será guardada en la Base de Datos Oracle 10g.

La información que se encuentra en esta tabla podrá ser consultada dependiendo de diferentes parámetros descritos por el MDMQ.

La información podrá ser imprimida para documentación del MDMQ.

Proceso de Traspaso de un Bien

Entrada: Código de Barras, Código de Bien.

Salida: Traspaso de un Bien a Custodio

Explicación del proceso: Este proceso se basa en el traspaso o asignación de un Bien Municipal a un Custodio dentro de MDMQ, por lo cual se deberá designar la ubicación de un Bien.

Un custodio deberá ser asignado a un Bien Municipal con su respectiva ubicación, estos datos deberá ser guardados dentro de la base de Bienes.

La búsqueda de un Bien podrá ser realizada mediante diferentes parámetros descritos por el MDMQ.

Proceso de Movimientos Temporales

Entrada: Código de Barras, Código de Bien.

Salida: Traspaso de un Bien Temporalmente

Explicación del proceso: Este proceso se basa en el movimiento temporal de un bien a una diferente ubicación, para esto deberá realizar los mismo pasos que cuando ocurre cuando se traspasa un Bien Municipal.

Proceso de Reportes Gerenciales

Entrada: Datos del Bien para Reporte

Salida: Reporte Gerencial.

Explicación del proceso: Este proceso consta en la realización de reportes gerenciales (Actas).

Proceso de Administración de Parámetros

Entrada: Parámetros.

Salida: Parámetros establecidos.

Explicación del proceso: Este proceso consta en la parametrización del sistema lo cual mediante una administración de Parámetros podremos aumentar o disminuir información según se requiera para la utilización del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control.

Proceso de Administración de Cuentas

Entrada: Datos Cuenta.

Salida: Nueva cuenta del sistema.

Explicación del proceso: Este proceso consta en el ingreso de las cuentas que conforman los bienes de larga duración y bienes sujetos a control del sistema lo cual mediante una administración podremos aumentar o disminuir información según se requiera para la utilización del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control.

3.3.7.4. Especificaciones Funcionales del Sistema

1. Para la Autenticación de los Usuarios se utiliza el Sistema adquirido por el MDMQ el cual se encarga de realizar el proceso de Seguridades del Sistema de Bienes Municipales.
2. Se realizara búsqueda de un Bien Municipal por Código de Barra o su Código Principal.
3. La aplicación será desarrollada en ambiente web, utilizando Microsoft Visual Studio .Net 2005, el lenguaje de programación Visual C#. El desarrollo utilizará ASP.NET y SCRIPTS (Javascript y VBScript) para garantizar una adecuada presentación de la aplicación Web.
4. Permitirá la administración de parámetros que necesite el Sistema de Bienes Municipales.

5. Permitirá consultas mediante una vista al Sistema de RRHH.
6. Permitirá la ejecución de las operaciones del Bien:
7. El registro o ingreso de nuevos Bienes, considerando los tipos de Bien que se ingresan.
8. Modificación de la información existente generando los históricos.
9. Consultas de los históricos.
10. Consulta y permitir dar de Baja a los Bienes
11. Depreciaciones de Bienes Municipales
12. Consulta y Erogación de Bienes Municipales
13. Reportes de los procesos de Bienes Municipales.
14. Reportes gerenciales.
15. Los reportes serán desarrollados en Reporting Services.
16. Utilizará dentro de su arquitectura un Servicio Web (Web Service) genérico que permitirá la ejecución de cualquier procedimiento almacenado y componentes DAL que pueden ser utilizados en las diferentes aplicaciones.

3.3.8. Configuración de Hardware y Software

3.3.8.1. Configuración General de Hardware y Sistemas Operativos

La configuración general de Hardware y Software es la siguiente:

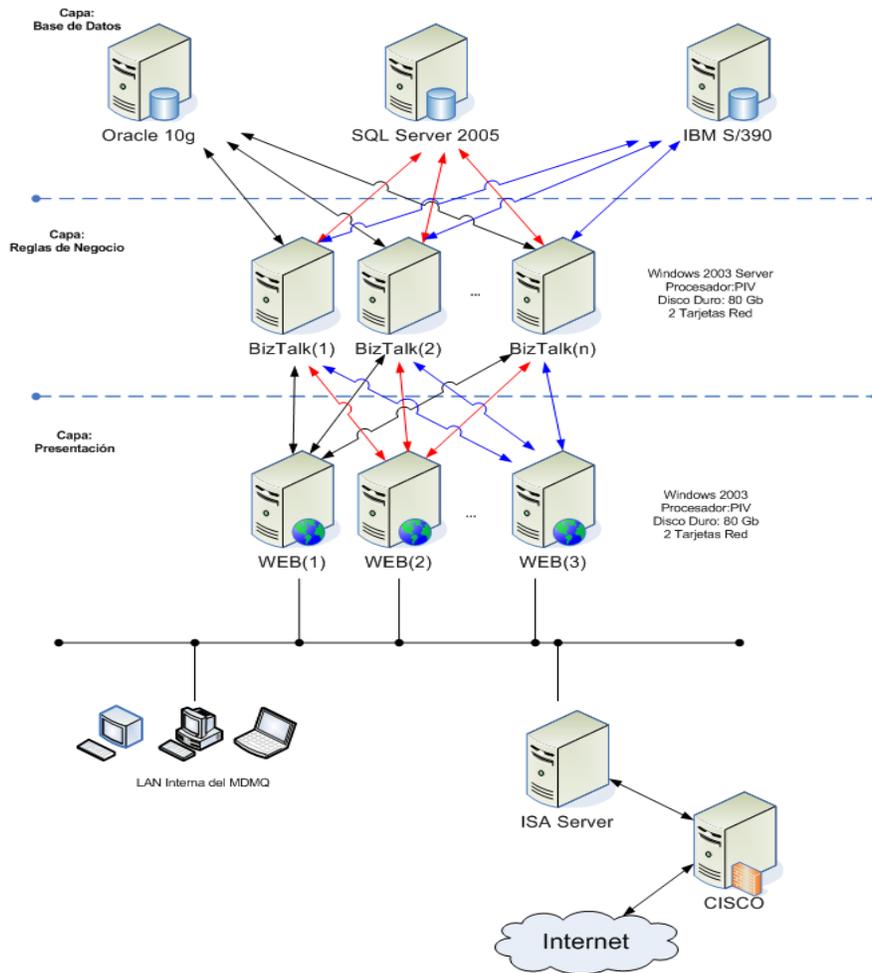


Figura 29. Configuración general de la Solución

En este gráfico se puede observar la presencia de tres repositorios de bases de datos (capa de base de datos):

- ✚ Oracle 10g
- ✚ SQL Server 2005
- ✚ IBM S/390

Cada una de las bases de datos se encuentra en servidores separados. Esta configuración también podría realizarse en un servidor HP-UX con particiones virtuales independientes, donde se ubicaría Oracle 10g y CICS en particiones UNIX y SQL Server en una partición Windows 2003 Server.

A continuación viene una granja de servidores BizTalk (Windows 2003 Server), cada uno de estos computadores que forme parte de esta granja constituyen la capa de reglas de negocio:

- ✚ Debe contar con 2 tarjetas de red, una para conectarse a las bases de datos y otro para conectarse a la granja WEB.
- ✚ Debe tener instalado el componente clsTrnSrvOracle.
- ✚ Debe tener instalado el componente de conexión al IBM S/390, denominados HIS o COMTIs.
- ✚ Debe tener instalado el Web Service de interface de la orquestación de BizTalk, y la orquestación Biztalk.

A continuación viene la granja WEB (Windows 2003), cada uno de los computadores que forme parte de esta granja corresponden a la capa de presentación:

- ✚ Debe tener instalado el componente clsTrnClientBiztalk.
- ✚ Debe tener instalado el componente DAL de seguridades y personas.
- ✚ Debe tener instalado los componentes DAL del Sistema de Bienes.
- ✚ Debe tener las páginas aspx del sitio web del Sistema de Bienes.

Sobre la capa de presentación, se encuentra la red LAN interna del MDMQ, la misma que permite a computadores conectados a ella y utilizando el navegador Internet Explorer 6.0 o superior, el acceso a la aplicación de Administración de Bienes Municipales.

3.3.8.2. Componentes del Sistema

A continuación se presenta un gráfico que ilustra todos los componentes de software que formarán parte del Sistema de Bienes de Larga Duración y Bienes Sujetos a Control del MDMQ:

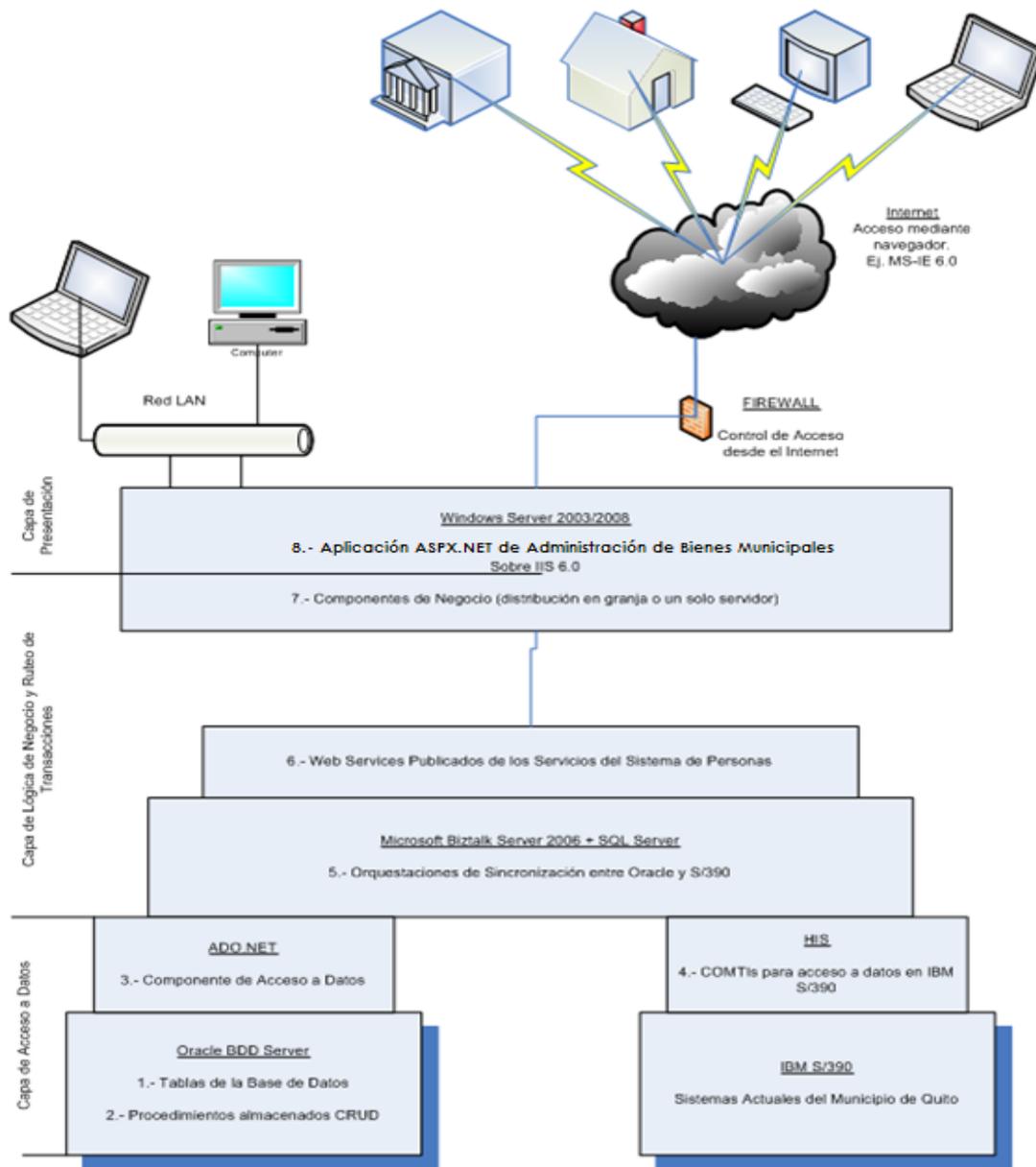


Figura 30. Componentes del Sistema

Como se puede observar en el presente gráfico, el software sobre el cual se va a implementar el Sistema de Bienes Municipales, tiene las siguientes características:

Para la capa de Datos:

- ✚ La base de datos Oracle 10g, donde estarán las tablas de datos y los procedimientos almacenados. Los procedimientos almacenados incluyen los CRUD de las tablas, que serán generados inicialmente mediante plantillas de CodeSmith. Para conectarse a la base de datos, hay que desarrollar en ADO.NET el componente de acceso a datos DAL, que también serán generados inicialmente por medio de plantillas de CodeSmith. Es importante considerar que los resultados de CodeSmith podrán ser modificados luego por el programador para afinar funcionalidades de la aplicación dependiendo de cada caso.
- ✚ IBM S/390, donde están las aplicaciones del MDMQ. Hay que desarrollar COMTIS para acceso a los datos del IBM S/390.

Para la capa de Negocios:

- ✚ BizTalk. Hay que desarrollar la orquestación necesaria para lograr la sincronización entre Oracle y el IBM S/390.
- ✚ Web Services. Hay que publicar los Web Services que genere la orquestación para que otros sistemas se puedan comunicar con el Sistema de Personas.
- ✚ DAL. Hay que desarrollar las DAL o componentes de negocio.

Para la capa de Presentación:

- ✚ Sitio web. Hay que desarrollar la aplicación web (con paginas aspx), del Sistema de Administración de Bienes Municipales.

3.3.9. Modelo de Base de Datos

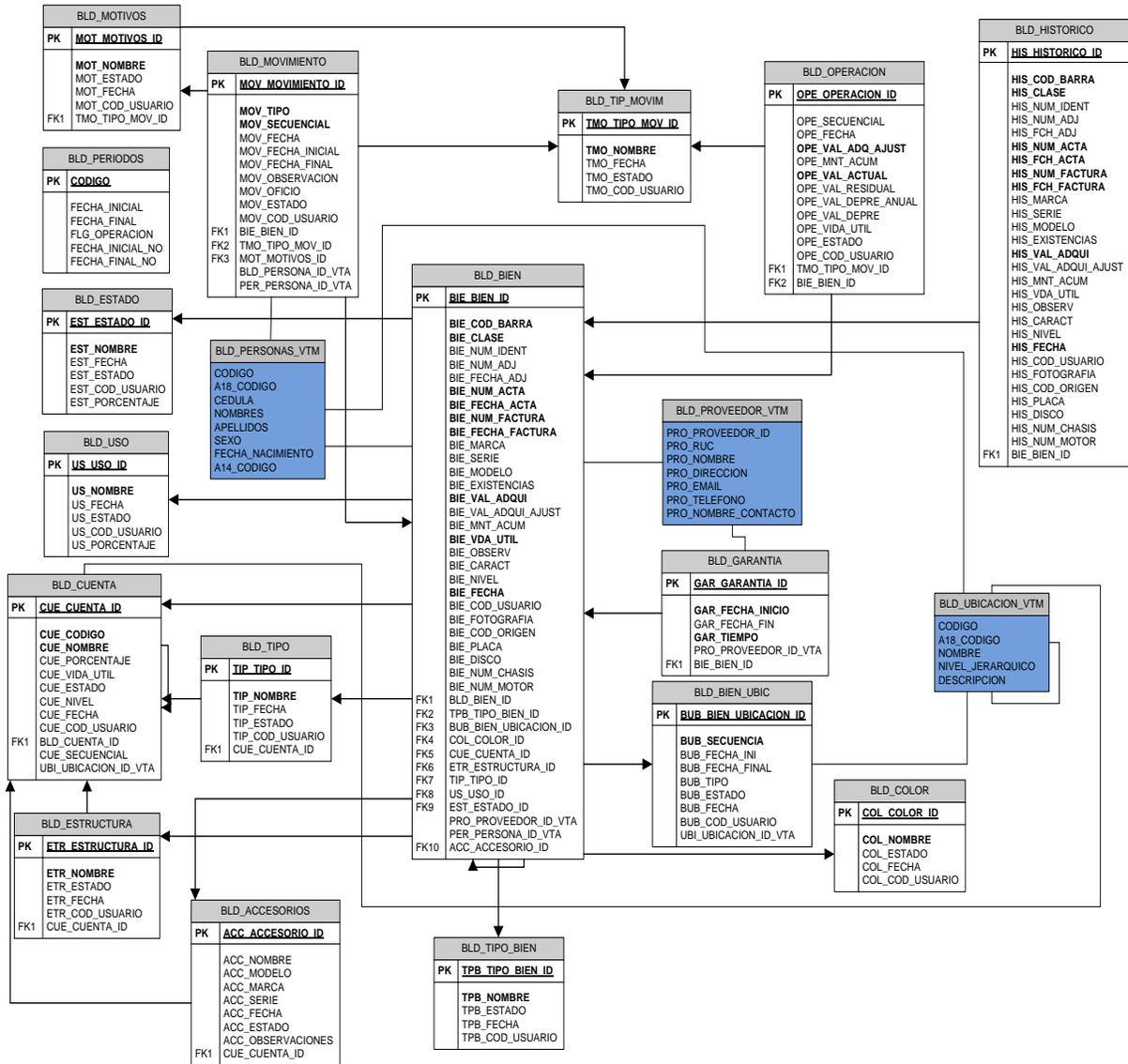


Figura 31. Modelado de Base de Datos

3.3.9.1. Diccionario de Datos

Datos de la tabla de Bien

TABLA: BLD_BIEN			
Almacena los datos propios del Bien			
CAMPO	TIPO	LONGITUD	DESCRIPCION
BIE_BIEN_ID	NUMBER not null,	(10, 0)	Código propio de la tabla (CLAVE PRIMARIA)
BIE_COD_BARRA	VARCHAR2 not null,	70	Código identificador físico y contable del Bien
BIE_CLASE	VARCHAR2 not null,	80	Nombre de la clase que representa el Bien
BIE_NUM_IDENT	VARCHAR2 null,	5	Número de identificador municipal
BIE_NUM_ADJ	NUMBER null,	(10, 2)	Número de adjudicación al proveedor
BIE_FECHA_ADJ	DATE null,		Fecha de adjudicación al proveedor
BIE_NUM_ACTA	NUMBER not null,	(10, 0)	Número de acta de ingreso
BIE_FECHA_ACTA	DATE not null,		Fecha de ingreso
BIE_NUM_FACTURA	VARCHAR2 not null,	15	Número de la factura con la que se recibe el Bien
BIE_FECHA_FACTURA	DATE not null,		Fecha de la factura con la que se recibe el Bien
BIE_MARCA	VARCHAR2 null,	50	Marca registrada en la factura del Bien
BIE_SERIE	VARCHAR2 null,	50	Serie registrada en la factura del Bien
BIE_MODELO	VARCHAR2 null,	50	Modelo registrado en la factura del Bien
BIE_EXISTENCIAS	NUMBER null,	(4, 0)	Número de Bienes de iguales características
BIE_VAL_ADQUI	NUMBER not null,	(15, 2)	Valor registrado en la factura con la que se recibe el Bien
BIE_VAL_ADQUI_AJUST	NUMBER null,	(15, 2)	Valor registrado en la factura con la que se recibe el Bien
BIE_MNT_ACUM	NUMBER null,	(15, 2)	Monto de depreciación acumulada
BIE_VDA_UTIL	NUMBER not null,	(3, 0)	Número de años de vida útil del Bien
BIE_OBSERV	VARCHAR2 null,	1024	Acotaciones del ingreso del Bien
BIE_CHARACTER	VARCHAR2 null,	1024	Características adicionales del Bien
BIE_NIVEL	NUMBER null,	(10, 0)	Nivel de ingreso del Bien
BIE_FECHA	DATE default 'sysdate' not null,		Fecha de registro del Bien
BIE_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien
BIE_FOTOGRAFIA	BLOB null,		Fotografía del Bien
BIE_COD_ORIGEN	VARCHAR2 null,	100	Código de Barras original de cada Bien
BIE_PLACA	VARCHAR2 null,	10	Número de placa del vehiculo

BIE_DISCO	VARCHAR2 null,	50	Número de disco del vehiculo
BIE_NUM_CHASIS	VARCHAR2 null,	50	Número de chasis del vehiculo
BIE_NUM_MOTOR	VARCHAR2 null,	50	Número del motor del vehiculo
BLD_BIE_ID	NUMBER null,	(10, 0)	Código del padre del Bien
TPB_TIPO_BIEN_ID	NUMBER null,	(10, 0)	Código del Tipo de Bien a ingresar (Largo plazo, sujeto a control)
BUB_BIEN_UBICACION_ID	NUMBER null,	(10, 0)	Código de la ubicación del Bien
COL_COLOR_ID	NUMBER null,	(10, 0)	Código que representa el color de un Bien
GAR_GARANTIA_ID	NUMBER null,	(10, 0)	Código de garantía que ofrece el proveedor
CUE_CUENTA_ID	NUMBER null,	(10, 0)	Código que representa a que cuenta pertenece el Bien
ETR_ESTRUCTURA_ID	NUMBER null,	(10, 0)	Código de la estructura del Bien
TIP_TIPO_ID	NUMBER null,	(10, 0)	Código del material del Bien
PER_PERSONA_ID	NUMBER null,	(10, 0)	Código de información de persona
US_USO_ID	NUMBER null,	(10, 0)	Código de Uso del Bien
EST_ESTADO_ID	NUMBER null,	(10, 0)	Código de Estado de Bien
PRO_PROVEEDOR_ID	NUMBER null	(10, 0)	Código del proveedor del Bien

Tabla 21. Diccionario de Datos de Bien

Datos de la tabla de Cuenta

TABLA: BLD_CUENTA			
Se almacena las cuentas de los bienes de forma recursiva			
CAMPO	TIPO	LONGUITUD	DESCRIPCION
CUE_CUENTA_ID	NUMBER not null,	(5, 0)	Código propio de la tabla
CUE_CODIGO	VARCHAR2 null,	50	Código que tenía en DB2
CUE_NOMBRE	VARCHAR2 null,	50	Nombre de la cuenta
CUE_PORCENTAJE	NUMBER null,	(3, 2)	Porcentaje que tiene el ítem del valor total.
CUE_VIDA_UTIL	NUMBER null,	(2, 0)	Vida útil de la cuenta contable.
CUE_ESTADO	VARCHAR2 null,	1	Activo o Histórico.
CUE_NIVEL	NUMBER not null,	(1, 0)	Nivel en el que se encuentra la cuenta.
CUE_FECHA	DATE null,		Fecha de creación de la cuenta
CUE_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra la cuenta
BLD_CUENTA_ID	NUMBER null,	(5, 0)	Código de recursividad.
CUE_SECUENCIAL	NUMBER not null,	(4, 0)	Número secuencial de la cuenta a la que pertenece.
UBI_UBICACION_ID	NUMBER null,	(7, 0)	Código de Ubicación (multiempresa)

Tabla 22. Diccionario de Datos de Cuenta

Datos de la tabla Ubicación del Bien

TABLA: BLD_BIEN_UBICACION			
Permite controlar las posibles ubicaciones de un bien y los prestamos de bienes a otras dependencias.			
CAMPO	TIPO	LON.	DESCRIPCION
BUB_BIEN_UBICACION_ID	NUMBER not null,	(7, 0)	Código propio de la tabla
BUB_SECUENCIA	NUMBER not null,	(4, 0)	Secuencial ascendente de la ubicación del Bien
BUB_FECHA_INI	DATE null,		Fecha inicio donde se encuentra el Bien
BUB_FECHA_FINAL	DATE null,		Fecha Final donde se encuentra el Bien
BUB_TIPO	VARCHAR2 null,	1	Tipo de ubicación. Indica si esta ubicación es a la que pertenece el bien o es la ubicación en la que está por un préstamo temporal o también puede ser su ubicación física P: Si es a la que pertenece R: Si es préstamo U: Si es su ubicación física
BUB_ESTADO	VARCHAR2 null,	1	Indica cual es la ubicación actual del bien A: ACTIVO H: HISTORICO
			Un bien puede tener dos ubicaciones con estado "A" pero el tipo debe ser diferente.
BUB_FECHA	DATE null,		Fecha de ingreso o cambio de ubicación
BUB_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra la cuenta
UBI_UBICACION_ID	NUMBER null,	(7, 0)	Código de la Ubicación la cual es recursiva

Tabla 23. Diccionario de Datos de Ubicación del Bien

Datos de la tabla de Movimientos del Bien

TABLA: BLD_MOVIMIENTO			
Se almacena los ingresos, bajas y traspasos para el control físico de los bienes			
CAMPO	TIPO	LONGITUD	DESCRIPCION
MOV_MOVIMIENTO_ID	NUMBER not null,	(4, 0)	Código propio de la tabla
MOV_TIPO	NUMBER null,	(4, 0)	Código del tipo de movimiento (ingreso, baja, etc)
MOV_SECUENCIAL	NUMBER not null,	(4, 0)	Secuencial que se relaciona con un

			bien
MOV_FECHA	DATE null,		Fecha del movimiento
MOV_FECHA_INICIAL	DATE null,		Fecha inicial del movimiento que se realiza
MOV_FECHA_FINAL	DATE null,		Fecha final del movimiento que se realiza
MOV_OBSERVACION	VARCHAR2 null,	1024	Detalle del movimiento
MOV_OFICIO	VARCHAR2 null,	20	Oficio que se relaciona con el movimiento
MOV_ESTADO	VARCHAR2 null,	1	Identifica el último movimiento
MOV_COD_USUARIO	VARCHAR2 null,	30	Usuario responsable del registro
BLD_PERSONA_ID	NUMBER null,	(10, 0)	Código del custodio de origen
BIE_BIEN_ID	NUMBER null,	(7, 0)	Código del bien
TMO_TIPO_MOV_ID	NUMBER null,	(3, 0)	Código del tipo de movimiento (ingreso, baja, etc)
MOT_MOTIVOS_ID	NUMBER null,	(4, 0)	Código del motivo que se realiza el movimiento
PER_PERSONA_ID	NUMBER null,	(10, 0)	Código de custodio destino. (Actual custodio del bien)

Tabla 24. Diccionario de Datos de Movimientos

Datos de la tabla de Estado

TABLA: BLD_ESTADO			
Almacena los posibles estados de un Bien			
CAMPO	TIPO	LONGUITUD	DESCRIPCION
EST_ESTADO_ID	NUMBER not null,	(3, 0)	Código propio de la tabla
EST_NOMBRE	VARCHAR2 null,	50	Nombre del estado
EST_FECHA	DATE null,		Fecha de registro del estado
EST_ESTADO	VARCHAR2 null,	1	estado del Bien
EST_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra el estado
EST_PORCENTAJE	NUMBER null,	(3, 2)	Porcentaje de estado

Tabla 25. Diccionario de Datos de Estado

Datos de la tabla Histórico

TABLA: BLD_HISTORICO			
Almacena los históricos de los Bienes Municipales			
CAMPO	TIPO	LONGUITUD	DESCRIPCION
BIE_BIEN_ID	NUMBER not null,	(7, 0)	Código propio de la tabla (CLAVE PRIMARIA)
BIE_COD_BARRA	VARCHAR2 not null,	70	Código identificador físico y contable del Bien
BIE_CLASE	VARCHAR2 not null,	50	Nombre de la clase que representa el Bien
BIE_NUM_IDENT	VARCHAR2 null,	5	Número de identificador municipal
BIE_NUM_ADJ	NUMBER null,	(7, 2)	Número de adjudicación al proveedor
BIE_FECHA_ADJ	DATE null,		Fecha de adjudicación al proveedor
BIE_NUM_ACTA	NUMBER not null,	(7, 0)	Número de acta de ingreso
BIE_FECHA_ACTA	DATE not null,		Fecha de ingreso
BIE_NUM_FACTURA	VARCHAR2 not null,	15	Número de la factura con la que se recibe el Bien
BIE_FECHA_FACTURA	DATE not null,		Fecha de la factura con la que se recibe el Bien
BIE_MARCA	VARCHAR2 null,	50	Marca registrada en la factura del Bien
BIE_EXISTENCIAS	NUMBER null,	(4, 0)	Número de Bienes de iguales características
BIE_VAL_ADQUI	NUMBER not null,	(15, 2)	Valor registrado en la factura con la que se recibe el Bien
BIE_VAL_ADQUI_AJUST	NUMBER null,	(15, 2)	Valor registrado en la factura con la que se recibe el Bien
BIE_MNT_ACUM	NUMBER null,	(15, 2)	Monto de depreciación acumulada
BIE_VDA_UTIL	NUMBER not null,	(3, 0)	Número de años de vida útil del Bien
BIE_OBSERV	VARCHAR2 null,	1024	Acotaciones del ingreso del Bien
BIE_CHARACTER	VARCHAR2 null,	1024	Características adicionales del Bien
BIE_NIVEL	NUMBER null,	(1, 0)	Nivel de ingreso del Bien
BIE_FECHA	DATE default 'sysdate' not null,		Fecha de registro del Bien
BIE_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien
BIE_FOTOGRAFIA	BLOB null,		Fotografía del Bien
BIE_COD_ORIGEN	VARCHAR2 null,	50	Código de Barras original de cada Bien
BIE_PLACA	VARCHAR2 null,	10	Número de placa del vehículo
BIE_DISCO	VARCHAR2 null,	50	Número de disco del vehículo
BIE_NUM_CHASIS	VARCHAR2 null,	50	Número de chasis del vehículo
BIE_NUM_MOTOR	VARCHAR2 null,	50	Número del motor del vehículo
BLD_BIE_ID	NUMBER null,	(7, 0)	Código del padre del Bien

Tabla 26. Diccionario de Datos de Históricos

Datos de la tabla de Color

TABLA: BLD_COLOR			
Parámetros de Colores			
CAMPO	TIPO	LONGITUD	DESCRIPCION
COL_COLOR_ID	NUMBER not null,	(3, 0)	Código propio de la tabla
COL_NOMBRE	VARCHAR2(20) null,	20	Nombre del Color
COL_ESTADO	VARCHAR2 null,	1	Estado en el que se encuentra el color
COL_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra el Color
COL_FECHA	DATE null,		Fecha de creación de registro

Tabla 27. Diccionario de Datos de Color

Datos de la tabla de Estructura

TABLA: BLD_ESTRUCTURA			
Almacena la estructura de una clase			
CAMPO	TIPO	LONGITUD	DESCRIPCION
ETR_ESTRUCTURA_ID	NUMBER not null,	(5, 0)	Código propio de la tabla
ETR_NOMBRE	VARCHAR2 null,	50	Nombre de la estructura
ETR_ESTADO	VARCHAR2 null,	1	Indica el estado del campo A: ACTIVO H: HISTORICO
ETR_FECHA	DATE null,		Fecha de creación de la estructura
ETR_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien
CUE_CUENTA_ID	NUMBER null,	(5, 0)	Código de la tabla cuenta

Tabla 28. Diccionario de Datos de Estructura

Datos de la tabla Tipo

TABLA: BLD_TIPO			
Almacena el tipo de una clase			
CAMPO	TIPO	LONGITUD	DESCRIPCION
TIP_TIPO_ID	NUMBER not null,	(3, 0)	Código propio de la tabla
TIP_NOMBRE	VARCHAR null,	50	Nombre del Topo de la clase
TIP_ESTADO	VARCHAR2 null,	1	Indica el estado del campo A: ACTIVO H: HISTORICO
TIP_FECHA	DATE null,		Fecha de creación del tipo
TIP_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien
CUE_CUENTA_ID	NUMBER null,	(5, 0)	Código de la tabla cuenta

Tabla 29. Diccionario de Datos de Tipo

Datos de tabla de Tipo del Bien

TABLA: BLD_TIPO_BIEN			
Representa el tipo del Bien			
CAMPO	TIPO	LONGITUD	DESCRIPCION
TPB_TIPO_BIEN_ID	NUMBER not null,	(2, 0)	Código propio de la tabla
TPB_NOMBRE	VARCHAR2 null,	50	Nombre del tipo de Bien
TPB_ESTADO	VARCHAR2 null,	1	Indica el estado del campo A: ACTIVO H: HISTORICO
TPB_FECHA	DATE null,		Fecha de creación del tipo de bien
TPB_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien

Tabla 30. Diccionario de Datos de Tipo de Bien

Datos de la tabla Garantía

TABLA: BLD_GARANTIA			
Representa la garantía de los Bienes			
CAMPO	TIPO	LONGITUD	DESCRIPCION
GAR_GARANTIA_ID	NUMBER null,	(3, 0)	Código propio de la tabla
GAR_FECHA_INICIO	DATE default 'sysdate' not null,		Fecha de inicio de garantía
GAR_FECHA_FIN	DATE null,		Fecha calculada de finalización de la garantía
GAR_TIEMPO	NUMBER null,	(2, 0)	Tiempo de garantía
PRO_PROVEEDOR_ID	NUMBER null,	(10, 0)	código de proveedor

Tabla 31. Diccionario de Datos de Garantía

Datos de la tabla de Operación

TABLA: BLD_OPERACION			
Se almacena los erogaciones, depreciaciones, bajas e ingresos para el control de operaciones de de los Bienes			
CAMPO	TIPO	LONGITUD	DESCRIPCION
OPE_OPERACION_ID	NUMBER not null,	(4, 0)	Código propio de la tabla
OPE_SECUENCIAL	NUMBER null,	(4, 0)	Número secuencial ascendente de las operaciones de los Bienes
OPE_FECHA	DATE null,		Fecha de realización de la operación
OPE_VAL_ADQ_AJUST	NUMBER null,	(15, 2)	Valor de adquisición ajustado se actualiza cuando
			se produce una erogación a dicho bien
OPE_MNT_ACUM	NUMBER null,	(15, 2)	La suma de las depreciaciones anuales
			que se realiza sobre el bien
OPE_VAL_ACTUAL	NUMBER null,	(15, 2)	Resta entre el valor ajustado y la depreciación acumulada representa el valor en libros
OPE_VAL_RESIDUAL	NUMBER null,	(15, 2)	Mínimo valor que alcanza el Bien
OPE_VAL_DEPRE_ANUAL	NUMBER null,	(15, 2)	Valor que se deprecia anualmente al Bien
OPE_VAL_DEPRE	NUMBER null,	(15, 2)	Valor de la depreciación de un Bien
OPE_VIDA_UTIL	NUMBER null,	(2, 0)	Vida útil del Bien
OPE_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien
TMO_TIPO_MOV_ID	NUMBER null,	(3, 0)	Tipo de movimiento a realizarse: Erogación Dar de Baja Ingreso Depreciación
BIE_BIEN_ID	NUMBER null,	(7, 0)	Código del Bien que realiza la operación

Tabla 32. Diccionario de Datos de Operación

Datos de la tabla de Tipo de Movimiento

TABLA: BLD_TIP_MOVIM			
Se almacena que tipo de proceso que va a realizar el Bien Municipal			
CAMPO	TIPO	LONGITUD	DESCRIPCION
TMO_TIPO_MOV_ID	NUMBER not null,	(3, 0)	Código propio de la tabla
TMO_NOMBRE	VARCHAR2 null,	50	Nombre del motivo que se va a realizar
TMO_ESTADO	VARCHAR2 null,	1	Indica el estado del campo A: ACTIVO H: HISTORICO
TMO_FECHA	DATE null,		Fecha de realización del Motivo
TMO_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien

Tabla 33. Diccionario de Datos de Tipo de Movimiento

Datos de la tabla de Motivos

TABLA: BLD_MOTIVOS			
Se almacena los motivos por el cual se hace un movimiento a un Bien Municipal			
CAMPO	TIPO	LONGITUD	DESCRIPCION
MOT_MOTIVOS_ID	NUMBER not null,	(10, 0)	Código propio de la tabla
MOT_NOMBRE	VARCHAR2 null,	50	Motivo por el cual se realiza el tipo de movimiento
MOT_ESTADO	VARCHAR2 null,	1	Indica el estado del campo A: ACTIVO H: HISTORICO
MOT_FECHA	DATE null,		Fecha de realización del Registro
MOT_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien
TMO_TIPO_MOV_ID	NUMBER null,	(3, 0)	Código de la tabla tipo de movimiento a realizarse Ingreso Modificación Dar de Baja Erogación Depreciación

Tabla 34. Diccionario de Datos de Motivos

Datos de la tabla de Uso

TABLA: BLD_USO			
Se almacena el uso del Bien respecto al tiempo			
CAMPO	TIPO	LONGITUD	DESCRIPCION
US_USO_ID	NUMBER not null,	(3, 0)	Código propio de la tabla
US_NOMBRE	VARCHAR2 null,	40	Nombre del uso
US_FECHA	DATE null,		Fecha de creación
US_ESTADO	VARCHAR2 null,	1	Estado del Bien según su uso
US_COD_USUARIO	VARCHAR2 null,	30	Usuario que registra en Bien
US_PORCENTAJE	NUMBER null,	(3, 2)	Porcentajes de acuerdo al tiempo de uso

Tabla 35. Diccionario de Datos de Uso

Datos de la tabla de Accesorios

TABLA: BLD_ACCESORIOS			
Se almacena el accesorio del Bien			
CAMPO	TIPO	LONGITUD	DESCRIPCION
ACC_ACCESORIO_ID	NUMBER not null,	(3, 0)	Código propio de la tabla
ACC_NOMBRE	VARCHAR2 null,	40	Nombre del uso
ACC_MODELO	DATE null,		Fecha de creacion
ACC_MARCA	VARCHAR2 null,	1	Estado del Bien según su uso
ACC_SERIE	NUMBER null,	(10, 0)	Usuario que registra en Bien
ACC_FECHA	NUMBER null,	(3, 2)	Porcentajes de acuerdo al tiempo de uso
ACC_OBSERVACIONES	VARCHAR2 null,	50	Observaciones del Accesorio
ACC_ESTADO	VARCHAR2 null,	1	Estado del Accesorio
ACC_COD_USUARIO	VARCHAR2 null,	30	Código del Usuario

Tabla 36. Diccionario de Datos de Accesorios

Datos de la tabla de Periodos

TABLA: BLD_PERIODOS			
Se almacena el periodo de las depreciaciones			
CAMPO	TIPO	LONGITUD	DESCRIPCION
CODIGO	NUMBER not null,	(3, 0)	Código propio de la tabla
FECHA_INICIAL	DATE null,		Nombre del uso
FECHA_FINAL	DATE null,		Fecha de creación
FLG_OPERACION	VARCHAR2 null,	1	Estado del Bien según su uso
FECHA_INICIAL_NO	DATE null,		Usuario que registra en Bien
FECHA_FINAL_NO	DATE null,		Porcentajes de acuerdo al tiempo de uso

Tabla 37. Diccionario de Datos de Periodos

Datos de la Vista de Personas

TABLA: PER_PERSONAS_VISTA			
Trabajadores del MDMQ			
CAMPO	TIPO	LONGITUD	DESCRIPCION
CODIGO	NUMBER(10,0) not null,	(10, 0)	Código propio de la tabla
A18_CODIGO	NUMBER(10,0) null	(10, 0)	Código de ubicación de usuario
CEDULA	VARCHAR2 not null	10	Cédula de Persona
NOMBRES	VARCHAR2 null	50	Nombre de Persona
APELLIDOS	VARCHAR2 null	50	Apellido de las personas
SEXO	VARCHAR2 null	1	Sexo de la Persona
FECHA_NACIMIENTO	DATE null		Fecha de Nacimiento
A14_ID	NUMBER null	(5, 0)	Régimen a la cual pertenece la Persona

Tabla 38. Diccionario de Datos de Personas

Datos de la Vista de Proveedores

TABLA: BLD_PROVEEDOR_VISTA			
Proveedores del MDMQ			
CAMPO	TIPO	LONGITUD	DESCRIPCION
PRO_PROVEEDOR_ID	NUMBER(10,0) not null,	(10, 0)	Código propio de la tabla
PRO_RUC	VARCHAR2 not null	13	Ruc del proveedor

PRO_NOMBRE	VARCHAR2 null	50	Nombre del Proveedor
PRO_MAIL	VARCHAR2 null	50	Mail del Proveedor
PRO_DIRECCION	VARCHAR2 null	50	Dirección del Proveedor
PRO_TELEFONO	VARCHAR2 null	20	Teléfono del Proveedor
PRO_NOMBRE_CONTACTO	VARCHAR2 null	50	Nombre de cualquier persona a quien se pueda contactar

Tabla 39. Diccionario de Datos de Proveedores

Datos de la Vista de Ubicación

TABLA: BLD_UBICACION_VISTA			
Ubicación del MDMQ, conexión con RRHH			
CAMPO	TIPO	LONGITUD	DESCRIPCION
CODIGO	NUMBER not null,	(10, 0)	Código propio de la tabla
A18_CODIGO	NUMBER null	(10, 0)	Recursividad de la ubicación
NOMBRE	VARCHAR2 null	50	Nombre de la Dependencia
NIVEL_JERARQUICO	VARCHAR2 null	50	Nivel en el que se encuentra la ubicación
DESCRIPCION	VARCHAR2 null	50	Descripción de la Ubicación

Tabla 40. Diccionario de Datos de Ubicación

3.3.10. Reportes

Listado de Reportes generales para SACBIM

- ✚ Detalle de Bien Municipal.
- ✚ Listado alfabético de Bienes con opciones parametrizadas
- ✚ Listado de Depreciaciones de Bien.
- ✚ Listado de Históricos.
- ✚ Acta de Bien
- ✚ Acta de Baja
- ✚ Acta de Transferencia de Bien
- ✚ Acta de Movimientos Temporales
- ✚ Acta de Erogación

3.3.11. Ejecución de Pruebas

Visual Studio Team Edition incluye herramientas de prueba integradas con Microsoft Visual Studio 2005, en este proyecto se utilizarán tales herramientas.

Team Edition permite crear, administrar, editar y ejecutar pruebas, así como obtener y almacenar los resultados de las pruebas. Los tipos de prueba integrados en Visual Studio que se utilizarán en este proyecto son las Pruebas Unitarias.

Para ejecutar las pruebas se utiliza el IDE de Visual Studio. Se realizarán grupos de pruebas o pruebas individuales que no sea el tipo de prueba manual desde una línea de comando.

3.3.11.1. Pruebas unitarias

Las pruebas unitarias sirven para utilizar otro código fuente llamando directamente a los métodos de una clase, pasando los parámetros apropiados, y posteriormente, si se incluyen instrucciones Assert, pueden comparar los valores que se generan con los valores esperados. Los métodos de pruebas unitarias residen en clases Test, que se almacenan en archivos de código fuente.

Con pruebas unitarias, no sólo se pueden probar métodos públicos, sino también privados. En ambos casos, las pruebas unitarias se crean automáticamente, desde el código que se desea probar. Visual Studio genera los resultados directamente.

3.3.12.

Cronograma de actividades

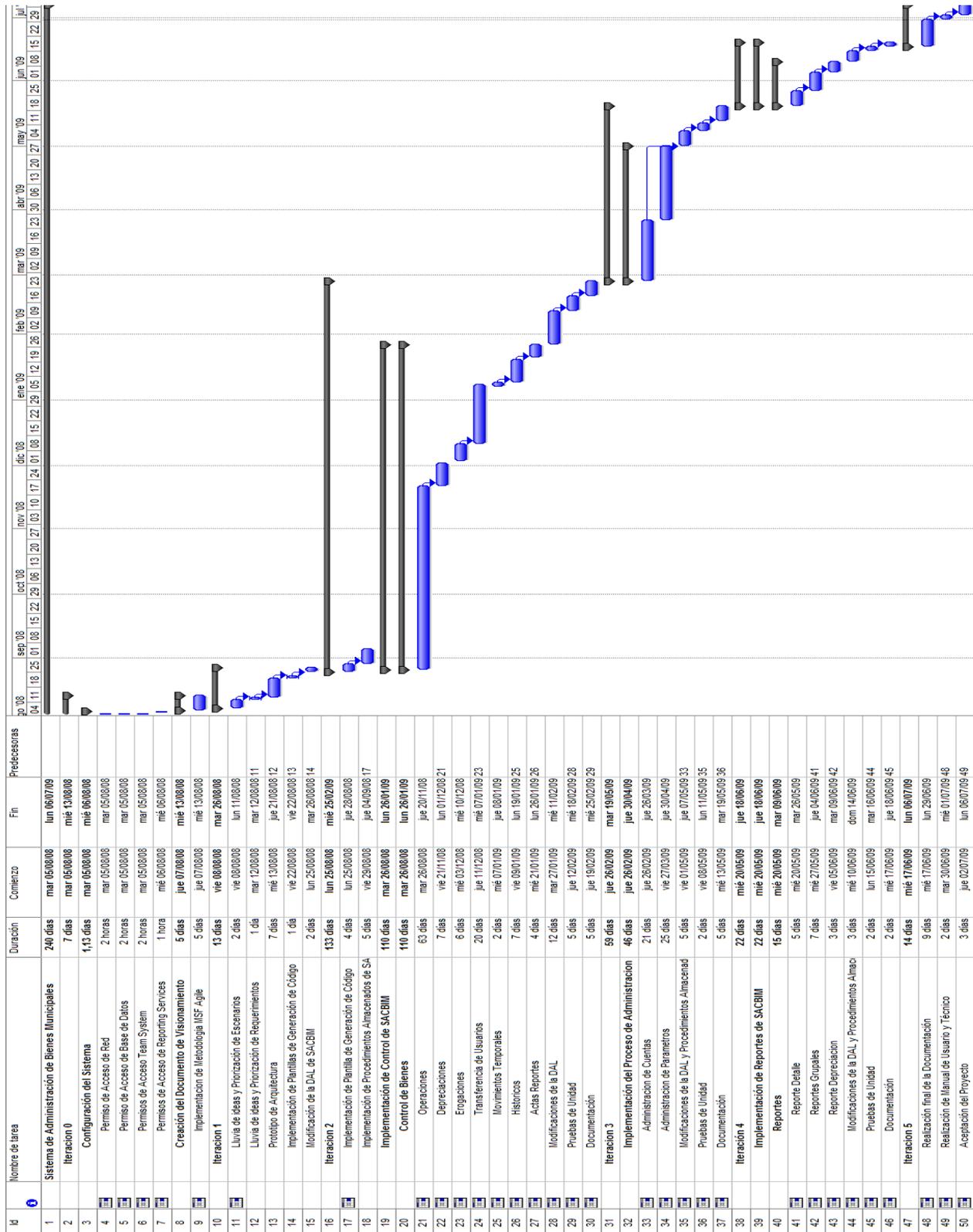


Figura 32. Cronograma de Iteraciones

3.3.13. Definiciones, acrónimos y abreviaturas

SACBIM.- Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos de Control.

CRUD.- Create, Read, Update, Delete (Creación, Lectura, Actualización y Eliminación de datos). Procesos de mantenimiento de datos utilizando paquetes y procedimientos almacenados en la Base de Datos Oracle.

DAL.- Data Application Layer. Son librerías que permiten el llamado a las funciones CRUD desde las aplicaciones de la capa de presentación.

MDMQ.- Municipio del Distrito Metropolitano de la ciudad de Quito.

MSF Agile. - Microsoft Solutions Framework Agile. Metodología de desarrollo de Sistemas informáticos de Microsoft para proyectos de corta duración que utiliza herramientas de Visual Studio Team Foundation Server para reemplazar la documentación tradicional requerida en el desarrollo de otros Sistemas Informáticos.

Bien Municipal.- Jurídicamente se entiende por bien todo aquello de carácter material o inmaterial susceptible de tener un valor.

Bien de Larga Duración.- Bien tangible o derecho de propiedad que dispone el Estado o entidades del sector público, destinado a utilizarse en las actividades de la entidad, susceptible de contabilizarse y reflejarse en los estados financieros, cuyo costo individual de adquisición, fabricación, donación o avalúo sea igual o superior a \$100 y la vida útil mayor a un año.

Bien Sujeto de Control.- Bien tangible de propiedad del Estado o de las entidades del sector público que, sin embargo de tener una vida útil superior a un año y utilizarse en las actividades de la entidad, tienen un costo individual de adquisición, fabricación, donación o avalúo, inferior a \$100.

Empleado.- Persona que trabaja en el MDMQ.

Web.- Interfaz de usuario GUI basada en exploradores (browser) de Internet.

CAPITULO IV. DESARROLLO Y PRUEBAS

4.1. ESCENARIOS

4.1.1. Administración de Parámetros

4.1.1.1. Administración de Colores

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación de los registros de Color relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración de Colores deberá ser escogida por un combo box el cual mostrara la lista de Colores registradas en el sistema.

CODIGO COLOR	NOMBRE COLOR	FECHA COLOR	ESTADO COLOR	Editar	Borrar
72	BLANCO	05/03/2009	A		
73	NEGRO	04/03/2009	A		
74	VERDE	05/03/2009	A		
85	AMARILLO	15/04/2009	A		
86	PLOMO	15/04/2009	A		
87	ROJO	15/04/2009	A		
88	ROSADO	15/04/2009	A		

Figura 33. Escenario de Color

Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.

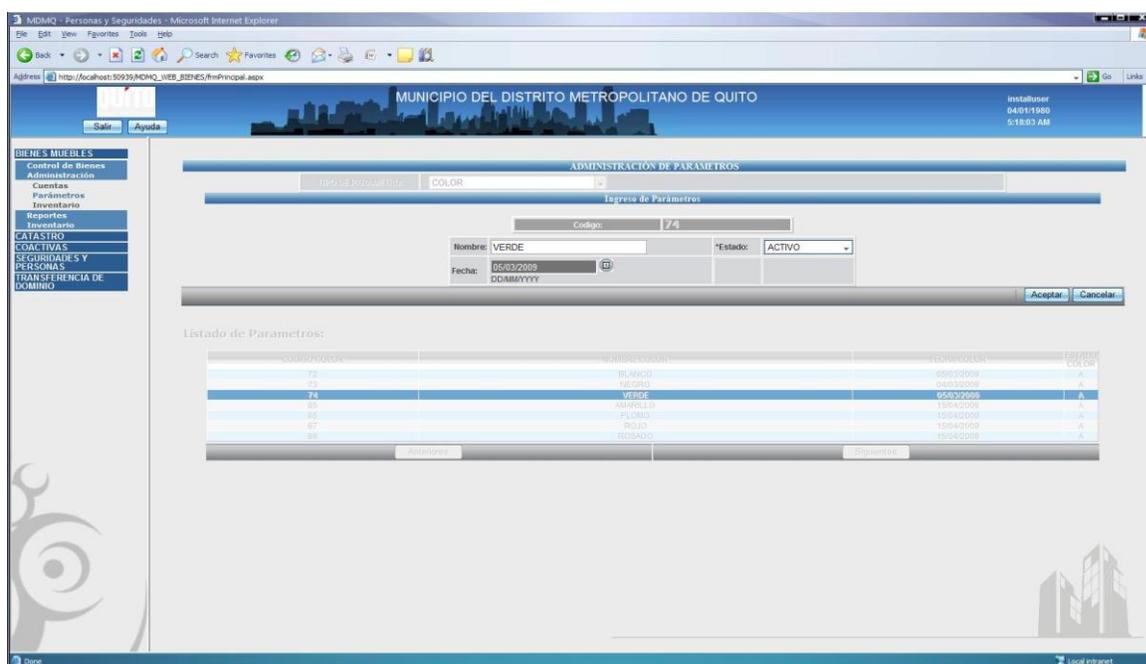


Buscar: Tipo Parametro: TIPO DE BIEN Código Parametro: Buscar Cancelar

Figura 34. Búsquedas de Escenario

En el programa se permite la selección de una de ellas para realizar cambios o modificaciones o eliminaciones de las mismas.

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos del color a su vez se permite además el ingreso de nuevos colores, registrando su nombre del color, fecha de creación, estado del color.



MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO

ADMINISTRACION DE PARAMETROS

Ingreso de Parametros

Nombre: VERDE Estado: ACTIVO Fecha: 05/03/2009

CODIGO COLOR	NOMBRE COLOR	FECHA CREACION	ESTADO
72	BLANCO	05/03/2009	A
73	ROJO	05/03/2009	A
74	VERDE	05/03/2009	A
75	AMARILLO	05/03/2009	A
76	FLORIDO	05/03/2009	A
77	ROJO	05/03/2009	A
78	ROSDADO	05/03/2009	A

Figura 35. Escenario de Operaciones de Color

En caso de modificar o añadir un parámetro nuevo del color se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

La creación de un nuevo color incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10 días

4.1.1.2. Administración de Estado

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación del registros del Estado relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración del Estado deberá ser escogida por un combo box el cual mostrara la lista del Estado registradas en el sistema.

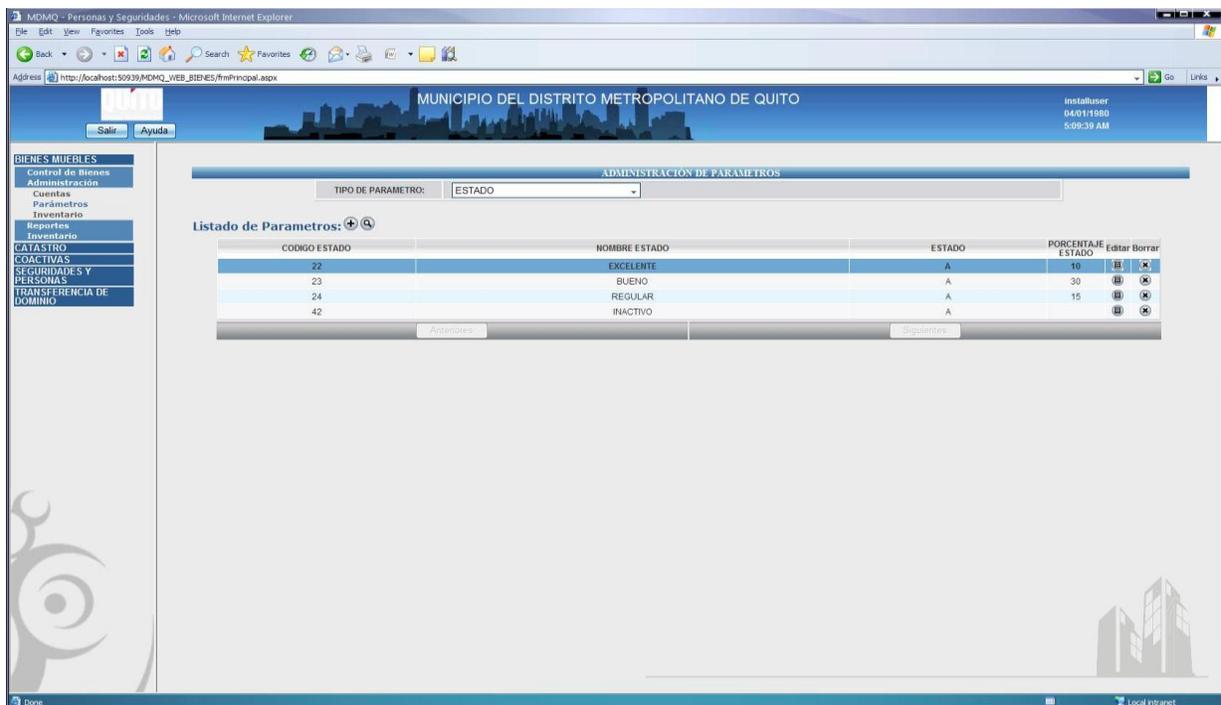


Figura 36. Escenario de Estado

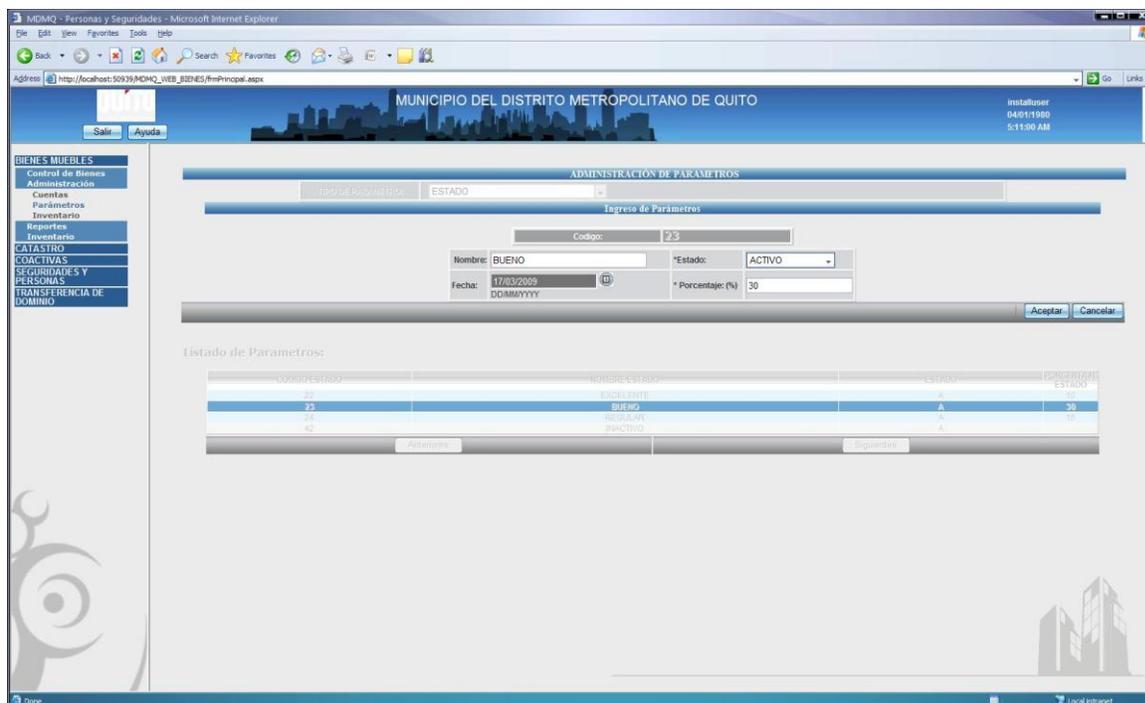
Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.



Figura 37. Escenario de Búsqueda del Estado

En el programa se permite la selección de una de ellas para realizar cambios o modificaciones o eliminaciones de las mismas.

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos del estado, a su vez se permite además el ingreso de nuevos estados, registrando su nombre del estado, fecha de creación, estado y su porcentaje.



CODIGO ESTADO	NOMBRE ESTADO	ESTADO	PORCENTAJE
22	EXCELENTE	A	10
23	BUENO	A	30
24	REGULAR	A	15
42	INACTIVO	A	

Figura 38. Escenario de Operaciones del Estado

En caso de modificar o añadir un parámetro nuevo del estado se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

No se podrá ingresar más del 100% en el total de los datos ingresados en el sistema.
La creación de un nuevo estado incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10 días

4.1.1.3. Administración de Uso

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación del los registros del Uso relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración del Uso deberá ser escogida por un combo box el cual mostrara la lista del Estado registradas en el sistema.

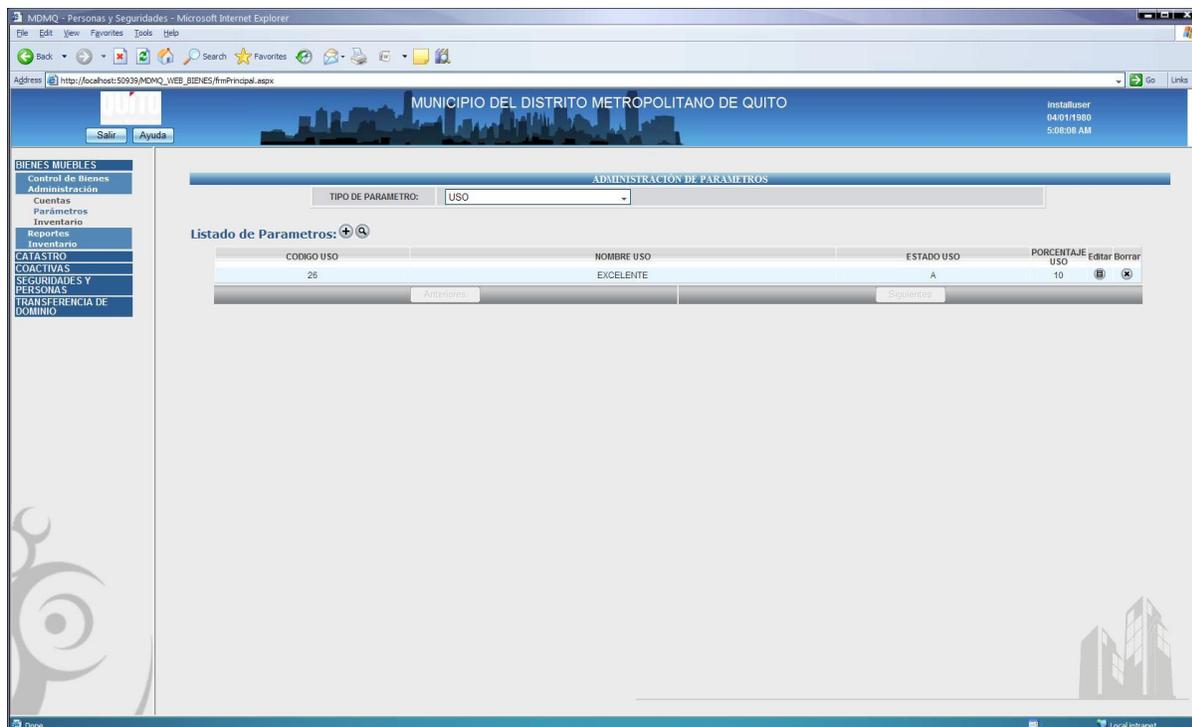


Figura 39. Escenario de Uso

Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.

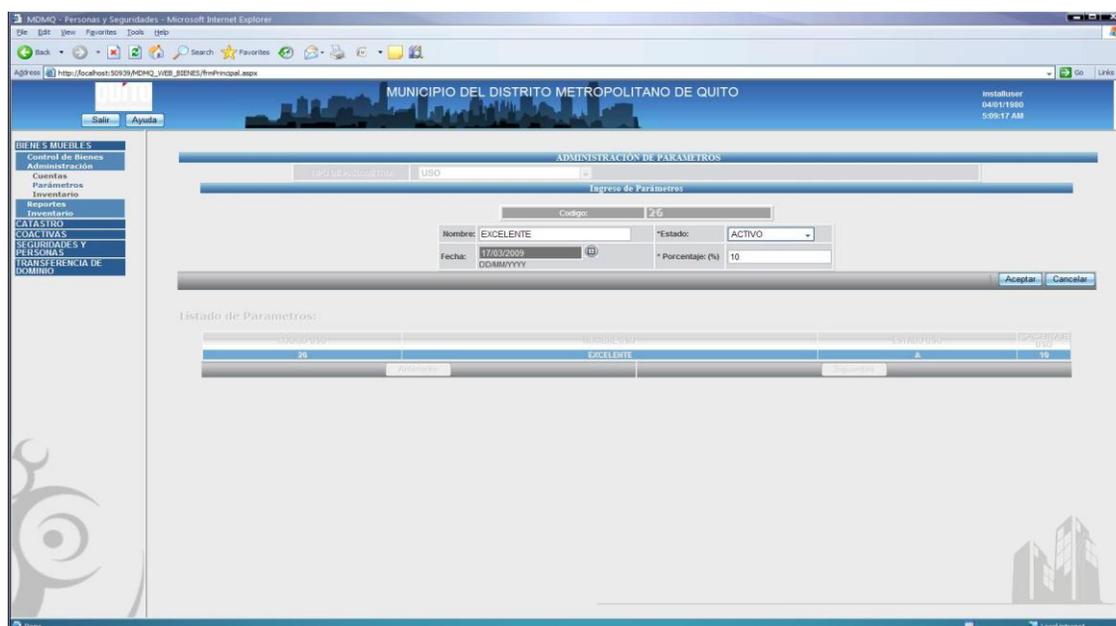


The image shows a search bar with the following elements: a 'Buscar:' label, a dropdown menu for 'Tipo Parametro' with 'TIPO DE BIEN' selected, a text input field for 'Código Parametro', and two buttons labeled 'Buscar' and 'Cancelar'.

Figura 40. Escenario de Búsqueda de Uso

En el programa se permite la selección de una de ellas para realizar cambios o modificaciones o eliminaciones de las mismas.

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos del uso, a su vez se permite además el ingreso de nuevos estados, registrando su nombre del uso, fecha de creación, estado y su porcentaje.



The screenshot shows a web application interface for 'MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO'. The main section is titled 'ADMINISTRACIÓN DE PARÁMETROS'. It features a search bar with 'USO' entered. Below it is a form for 'Ingreso de Parámetros' with fields for 'Codigo' (26), 'Nombre' (EXCELENTE), 'Estado' (ACTIVO), 'Fecha' (17/03/2009), and 'Porcentaje (%)' (10). There are 'Aceptar' and 'Cancelar' buttons. Below the form is a table titled 'Estado de Parámetros' with columns for 'CODIGO USO', 'NOMBRE USO', 'ESTADO USO', and 'PORCENTAJE USO'. The table contains one row with the values 26, EXCELENTE, A, and 10.

CODIGO USO	NOMBRE USO	ESTADO USO	PORCENTAJE USO
26	EXCELENTE	A	10

Figura 41. Escenario de Operaciones de Uso

En caso de modificar o añadir un parámetro nuevo del uso se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

No se podrá ingresar más del 100% en el total de los datos ingresados en el sistema.

La creación de un nuevo uso incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10 días

4.1.1.4. Administración de Tipo de Movimiento

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación del los registros del Tipo de Movimiento relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración del Tipo de Movimiento deberá ser escogida por un combo box el cual mostrara la lista del Tipo de Movimiento registradas en el sistema.

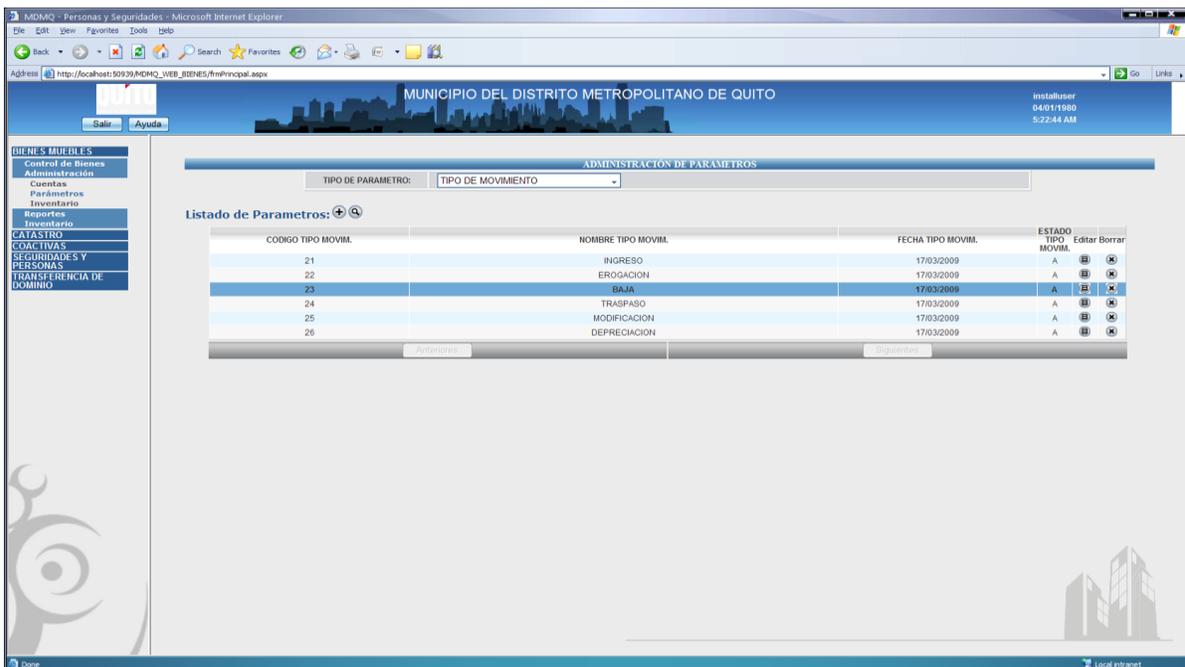


Figura 42. Escenario de Tipo de Movimiento

Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.

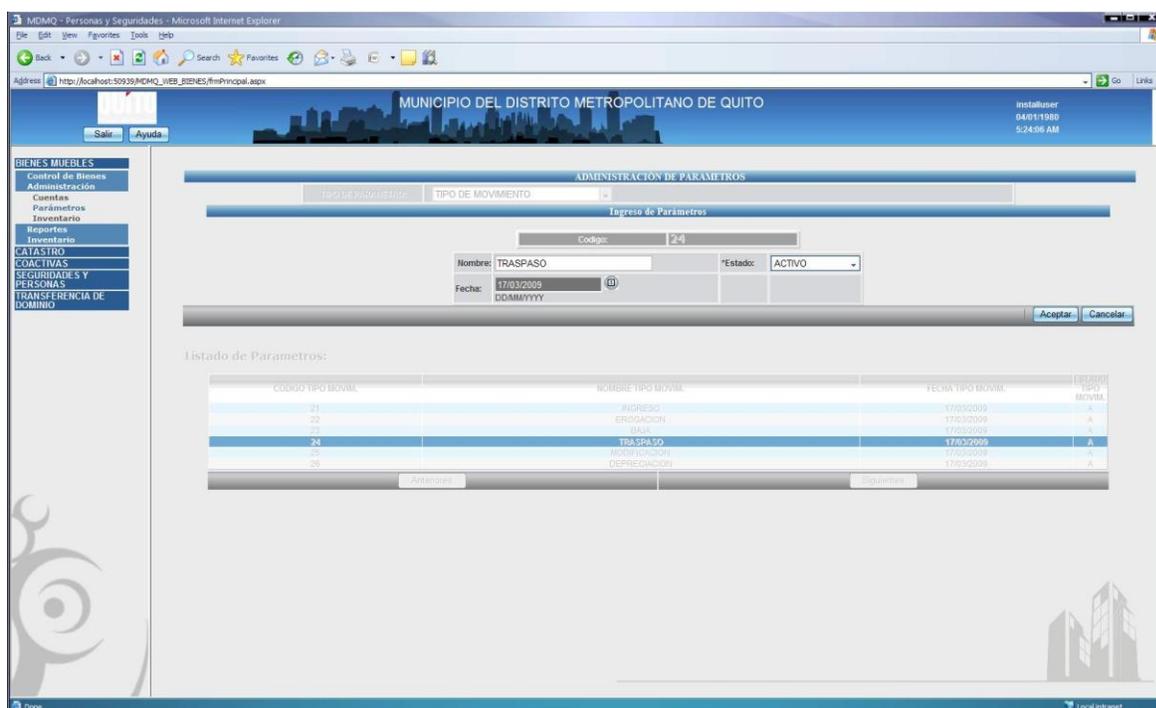


A search bar with the following elements: a 'Buscar:' label, a dropdown menu for 'Tipo Parametro' with 'TIPO DE BIEN' selected, a text input field for 'Código Parametro', and two buttons labeled 'Buscar' and 'Cancelar'.

Figura 43. Escenario de Búsqueda de Tipo de Movimiento

En el programa se permite la selección de una de ellas para realizar cambios o modificaciones o eliminaciones de las mismas.

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos del Tipo de Movimiento, a su vez se permite además el ingreso de nuevos estados, registrando su nombre del uso, fecha de creación, y su estado.



The screenshot shows a web application interface for 'MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO'. The main area is titled 'ADMINISTRACIÓN DE PARÁMETROS' and 'TIPO DE MOVIMIENTO'. It features a form for 'Ingreso de Parámetros' with fields for 'Codigo' (24), 'Nombre' (TRASPASO), 'Estado' (ACTIVO), and 'Fecha' (17/03/2009). Below the form is a table titled 'Listado de Parámetros' with columns for 'CODIGO TIPO MOVIM', 'NOMBRE TIPO MOVIM', 'FECHA TIPO MOVIM', and 'ESTADO TIPO MOVIM'. The table contains five rows of data.

CODIGO TIPO MOVIM	NOMBRE TIPO MOVIM	FECHA TIPO MOVIM	ESTADO TIPO MOVIM
21	INGRESO	17/03/2009	A
22	EROGACION	17/03/2009	A
23	BAJE	17/03/2009	A
24	TRASPASO	17/03/2009	A
25	MODIFICACION	17/03/2009	A
26	DEPRECIACION	17/03/2009	A

Figura 44. Escenario de Operaciones de Tipo de Movimiento

En caso de modificar o añadir un parámetro nuevo del Tipo de Movimiento se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

La creación de un nuevo Tipo de movimiento incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10 días

4.1.1.5. Administración de Tipo del Bien

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación de los registros del Tipo de Bien relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración del Tipo del Bien deberá ser escogida por un combo box el cual mostrara la lista del Tipo del Bien registradas en el sistema.

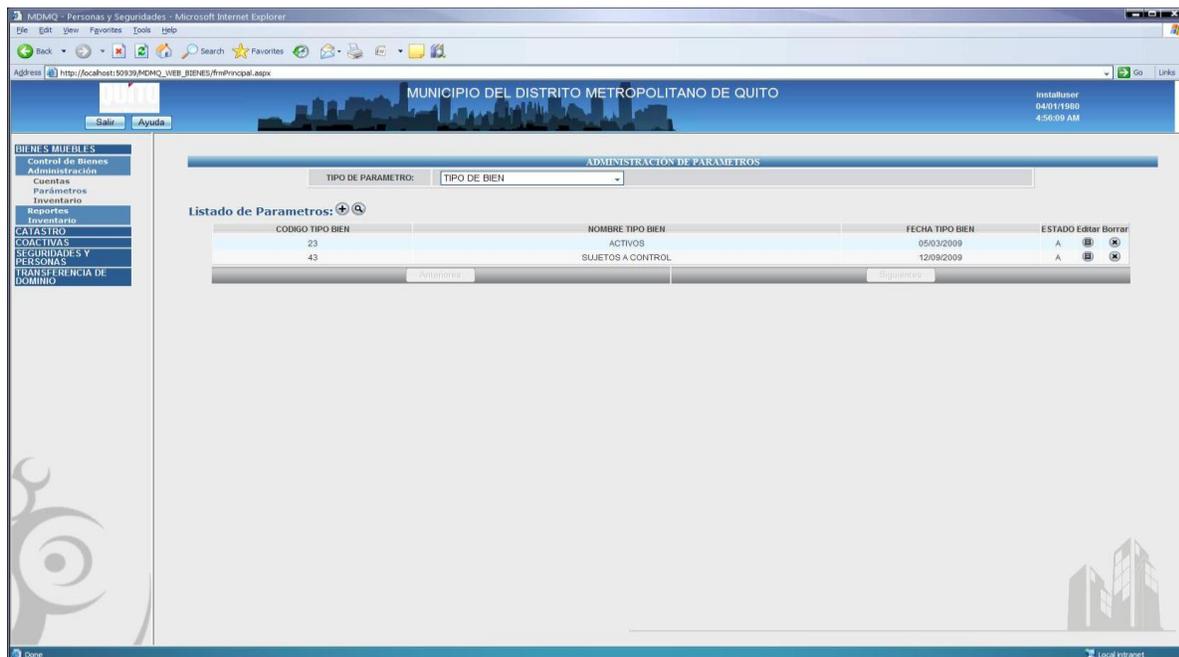


Figura 45. Escenario de Tipo de Bien

Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.



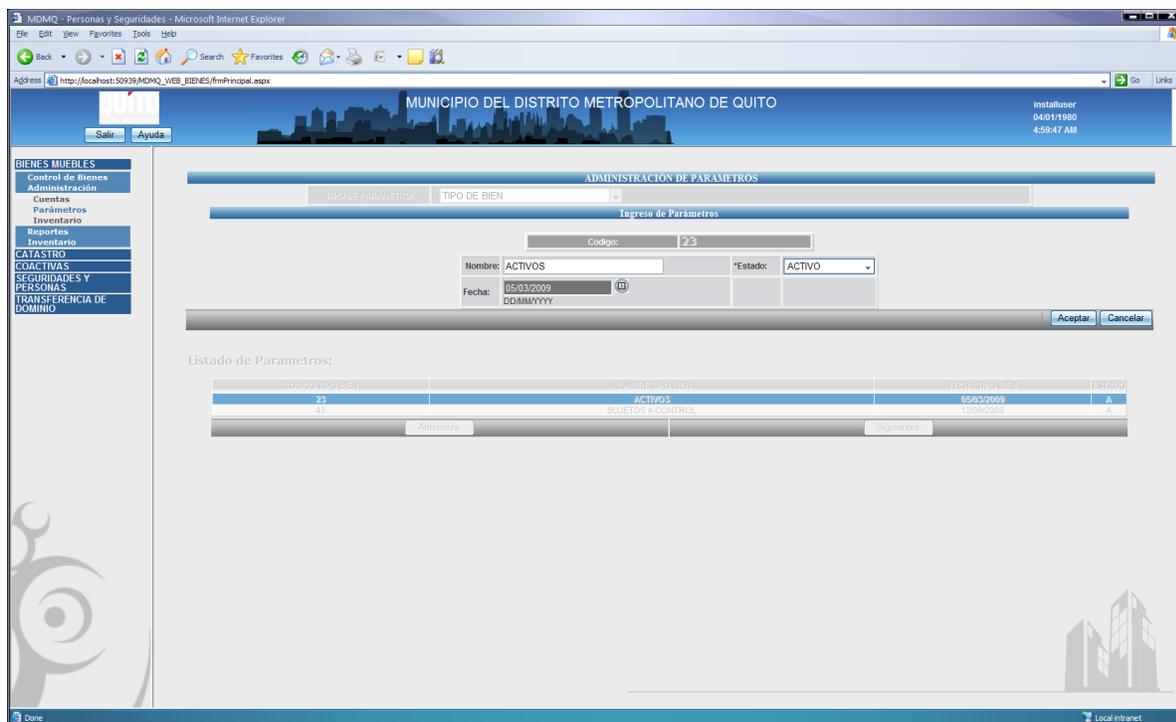
The screenshot shows a search bar with the following elements:

- Label: **Buscar:**
- Field: **Tipo Parametro:** with a dropdown menu showing **TIPO DE BIEN**.
- Field: **Código Parametro:** with an empty input box.
- Buttons: **Buscar** and **Cancelar**.

Figura 46. Escenario de Búsqueda de Tipo de Bien

En el programa se permite la selección de una de ellas para realizar cambios o modificaciones o eliminaciones de las mismas.

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos del Tipo del Bien, a su vez se permite además el ingreso de nuevos tipos de bienes, registrando su nombre del uso, fecha de creación, y su estado.



The screenshot shows a web application interface for 'MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO'. The main content area is titled 'ADMINISTRACIÓN DE PARAMETROS' and contains the following elements:

- Search bar: **Tipo de parámetro:** with a dropdown menu showing **TIPO DE BIEN**.
- Section: **Ingreso de Parámetros**
- Form fields:
 - Código:** 23
 - Nombre:** ACTIVOS
 - *Estado:** ACTIVO
 - Fecha:** 05/03/2009
- Buttons: **Aceptar** and **Cancelar**.
- Section: **Listado de Parámetros:**
- Table with columns: **Código de parámetro**, **Nombre de parámetro**, **Fecha de creación**, **Estado**.
- Table data:

Código de parámetro	Nombre de parámetro	Fecha de creación	Estado
23	ACTIVOS	05/03/2009	A
43	SUJETOS A CONTROL	12/08/2008	A
- Buttons: **Anteriores** and **Siguientes**.

Figura 47. Escenario de Operaciones de Tipo de Bien

En caso de modificar o añadir un parámetro nuevo del Tipo del Bien se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

La creación de un nuevo tipo de movimiento incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10 días

4.1.1.6. Administración de Tipo de Cuenta

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación del los registros del Tipo de Cuenta relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración del Tipo de Cuenta deberá ser escogida por un combo box el cual mostrara la lista del Tipo de Cuenta registradas en el sistema.

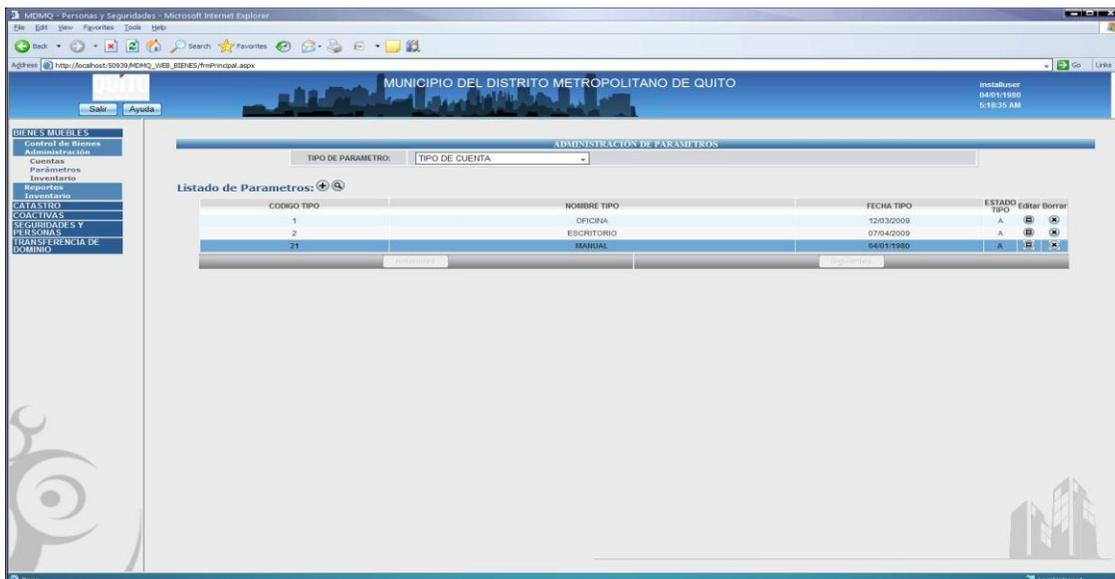


Figura 48. Escenario de Tipo de Cuenta

Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.

Buscar: Tipo Parametro: TIPO DE BIEN Código Parametro: [Buscar] [Cancelar]

Figura 49. Escenario de Búsqueda de Tipo de Cuenta

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos del Tipo de Cuenta, a su vez se permite además el ingreso de nuevos tipos de cuenta, registrando su nombre del uso, fecha de creación, su estado, y el padre el cual será determinado por un catalogo de búsqueda.

Cuenta	Nombre	Fecha	Estado
1	OFICINA	12/04/2009	A
2	ESCRITORIO	07/04/2009	A
21	BAÑAL	04/01/2009	A

Figura 50. Escenario de Operaciones de Tipo de Cuenta

En caso de modificar o añadir un parámetro nuevo del Tipo del Bien se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

La búsqueda de tipo de cuenta se realizara mediante un catalogo el cual muestra las clases de una cuenta.

La creación de un nuevo tipo de movimiento incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.1.7. Administración de Estructura

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación del los registros de Estructura relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración de la Estructura deberá ser escogida por un combo box el cual mostrara la lista del Tipo de Cuenta registradas en el sistema.

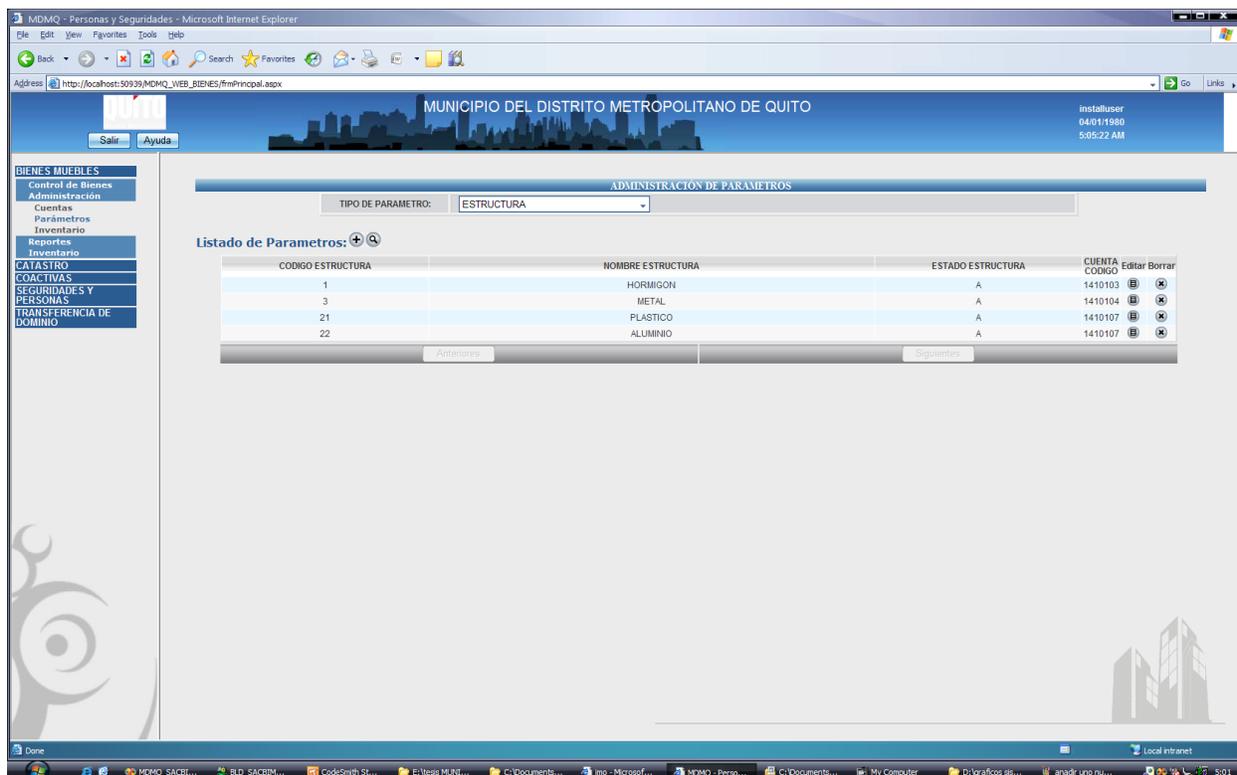


Figura 51. Escenario de Estructura

Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.

Buscar: Tipo Parametro: TIPO DE BIEN Código Parametro: Buscar Cancelar

Figura 52. Escenario de Búsqueda de Estructura

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos de la Estructura, a su vez se permite además el ingreso de nuevos estados, registrando su nombre de la estructura, fecha de creación, su estado, y el padre el cual será determinado por un catalogo de búsqueda .

MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO
 Ingresar 04011488 5:07:41 AM

ADMINISTRACIÓN DE PARAMETROS
 TIPO DE PARAMETRO: ESTRUCTURA

Ingreso de Parámetros

Código: 22

Cuenta: 1410107 EQUIPOS, SISTEMAS Y PAQUET

Nombre: ALUMNIO *Estado: ACTIVO

Fecha: 15/04/2009 DDMMYYYY

Aceptar Cancelar

Listado de Parámetros:

CODIGO ESTRUCTURA	NOMBRE ESTRUCTURA	ESTADO ESTRUCTURA	BIENES CODIGO
1	HORMIGON	A	1410103
3	METAL	A	1410104
21	PLASTICO	A	1410107
22	ALUMNIO	A	1410107

Anteriores Siguientes

Figura 53. Escenario de Operaciones de Estructura

En caso de modificar o añadir un parámetro nuevo de la Estructura se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

La búsqueda de tipo de cuenta se realizara mediante un catalogo el cual muestra el grupo general de las clases.

La creación de un nuevo tipo de movimiento incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.1.8. Administración de Motivo de Movimiento

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación del los registros de Motivo de Movimientos relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración de la Motivo de Movimientos deberá ser escogida por un combo box el cual mostrara la lista del Motivo de Movimientos registradas en el sistema.

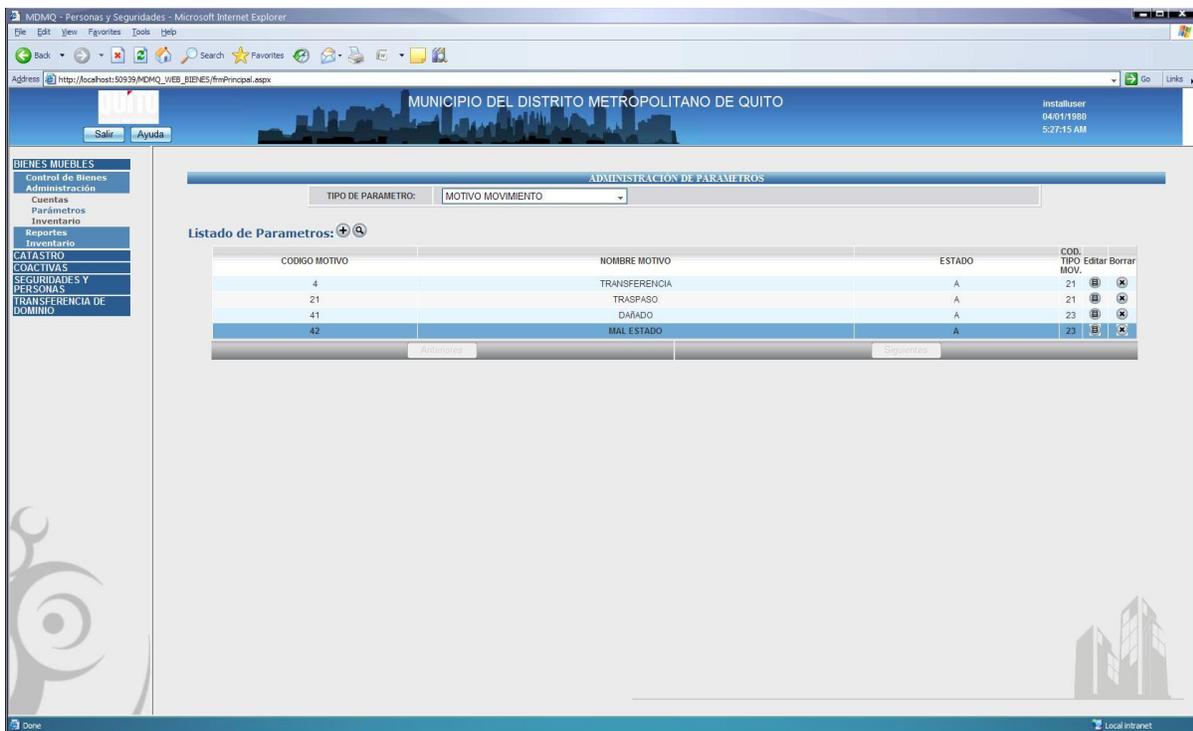


Figura 54. Escenario de Motivo de Movimiento

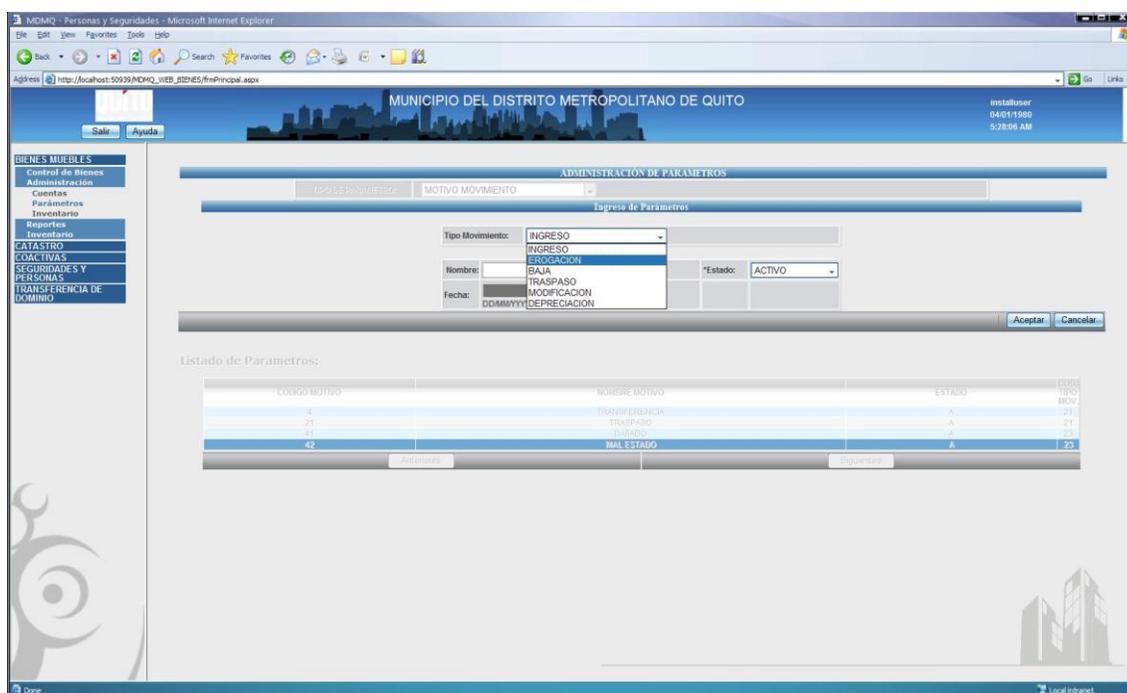
Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.



The screenshot shows a search bar with the following elements: a 'Buscar:' label, a 'Tipo Parametro:' dropdown menu currently set to 'TIPO DE BIEN', a 'Codigo Parametro:' text input field, and two buttons labeled 'Buscar' and 'Cancelar'.

Figura 55. Escenario de Búsqueda de Motivo de Movimiento

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos de Motivo de Movimientos, a su vez se permite además el ingreso de nuevos motivo de movimientos, registrando su nombre del motivo, fecha de creación, su estado, y el tipo de movimiento a ejecutarse.



The screenshot displays the 'ADMINISTRACIÓN DE PARAMETROS' interface for 'MOTIVO MOVIMIENTO'. It features a navigation menu on the left with categories like 'BIENES MUEBLES', 'CATASTRO', and 'SEGURIDADES Y PERSONAS'. The main area contains a form for 'Ingreso de Parametros' with fields for 'Tipo Movimiento' (dropdown), 'Nombre' (text), 'Fecha' (DDMMYYYY), and 'Estado' (dropdown). Below the form is a table titled 'Listado de Parametros' with columns for 'Codigo Motivo', 'Nombre Motivo', 'Estado', and 'Codigo Tipo Mov'. The table lists three entries: 'TRANSFERENCIA', 'TRASPASO', and 'REAL ESTADO'.

codigo motivo	nombre motivo	estado	codigo tipo mov
21	TRANSFERENCIA	A	21
21	TRASPASO	A	21
42	REAL ESTADO	A	21

Figura 56. Escenario de Operaciones de Motivo de Movimiento

En caso de modificar o añadir un parámetro nuevo de la Estructura se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

La creación de un nuevo de tipo de movimiento incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.1.9. Administración de Accesorios

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación del los registros de Accesorios relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

La Administración de Accesorios deberá ser escogida por un combo box el cual mostrara la lista de Accesorios registradas en el sistema.

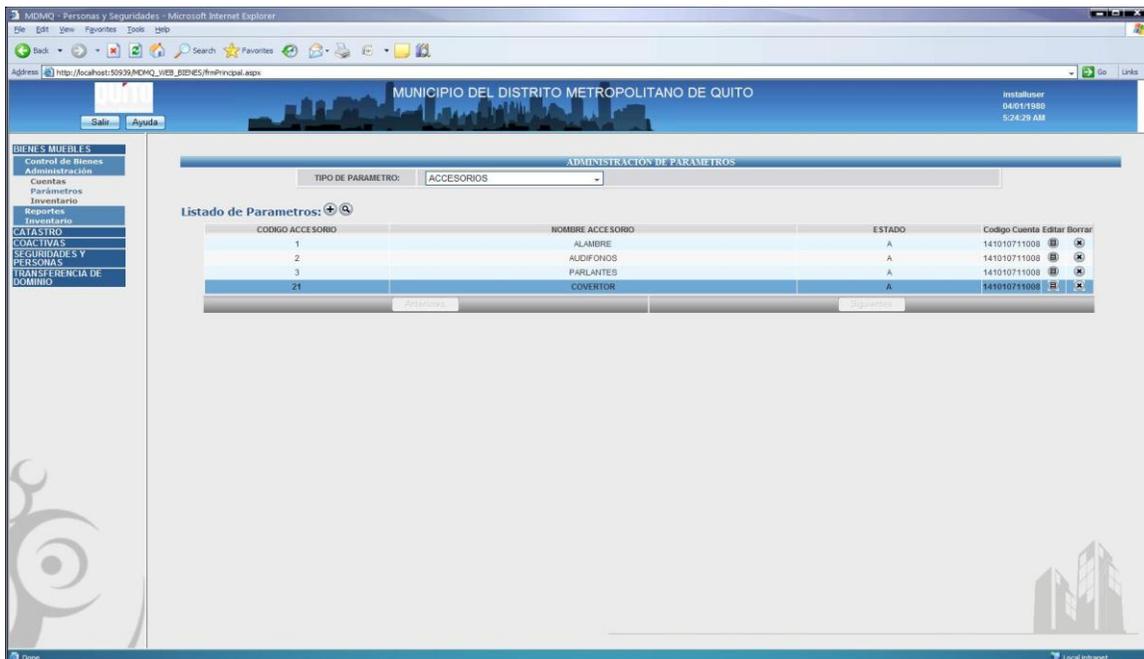


Figura 57. Escenario de Accesorios

Se incluyen búsquedas por código del parámetro o por el nombre del parámetro.



Figura 58. Escenario de Búsqueda de Accesorios

Al momento de seleccionar un elemento para editar se abrirá una pantalla alterna la cual podemos modificar los elementos de Accesorios, a su vez se permite además el ingreso de nuevos accesorios, registrando su nombre del accesorio, fecha de creación, su estado, un catalogo de cuentas, modelo, marca, serie, y las observaciones.

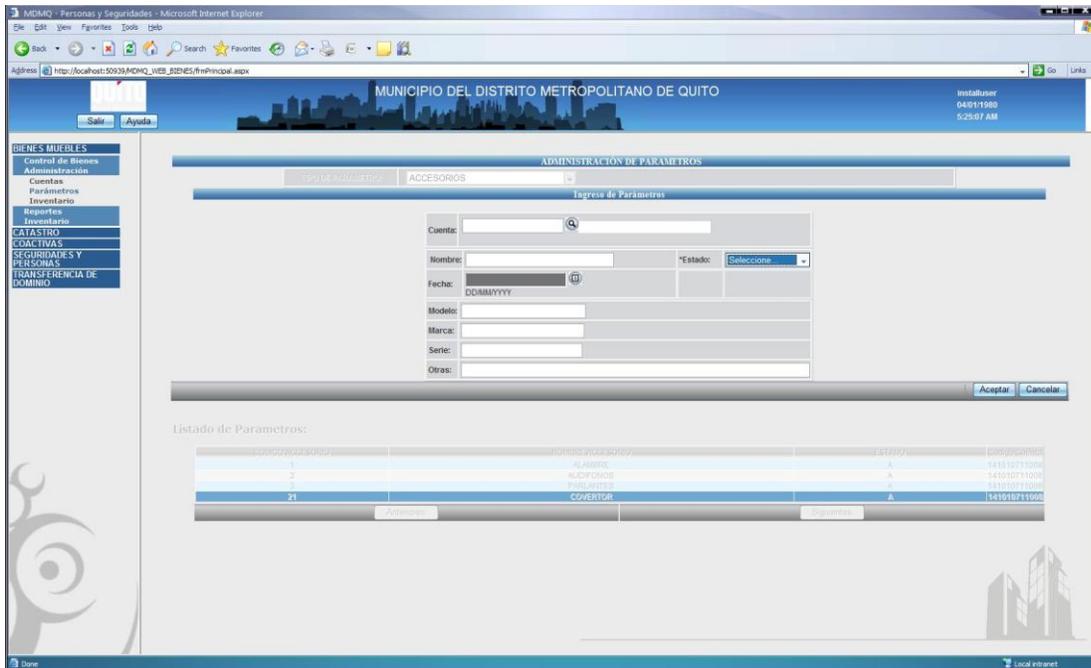


Figura 59. Escenario de Operaciones de Accesorios

En caso de modificar o añadir un parámetro nuevo de la Estructura se mostrara un mensaje de aviso.

En caso de borrar un elemento deseado se mostrara un mensaje de aviso.

La creación de un nuevo Accesorio incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.2. Administración de Cuentas

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Eliminación de los registros de Cuentas relacionadas con Bien Municipal, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

El escenario de Cuentas muestra todos los elementos de la Cuenta para que el Usuario MDMQ pueda elegir la operación que desee, según muestra en la siguiente figura.

Código	Cuentas	Fecha	Vida Útil	Editar	Borrar
1	ACTIVO	08/03/2009			
14	INVERSIONES EN BIENES DE LARGA DURACION	08/03/2009			
141	BIENES DE ADMINISTRACION	08/03/2009			
14101	BIENES MUEBLES	08/03/2009			
1410103	MOBILIARIO	08/03/2009			
14101031	USO GENERAL	08/03/2009			
141010310002	ABACO	08/03/2009			
141010310072	AFICHE	08/03/2009			
141010310144	AJEDREZ	08/03/2009			
141010310162	ALACENA	10/03/2009			

Figura 60. Escenario de Cuentas

Se puede modificar, añadir o eliminar las cuentas de acuerdo como el Usuario-MDMQ vea que es pertinente hacerlo.

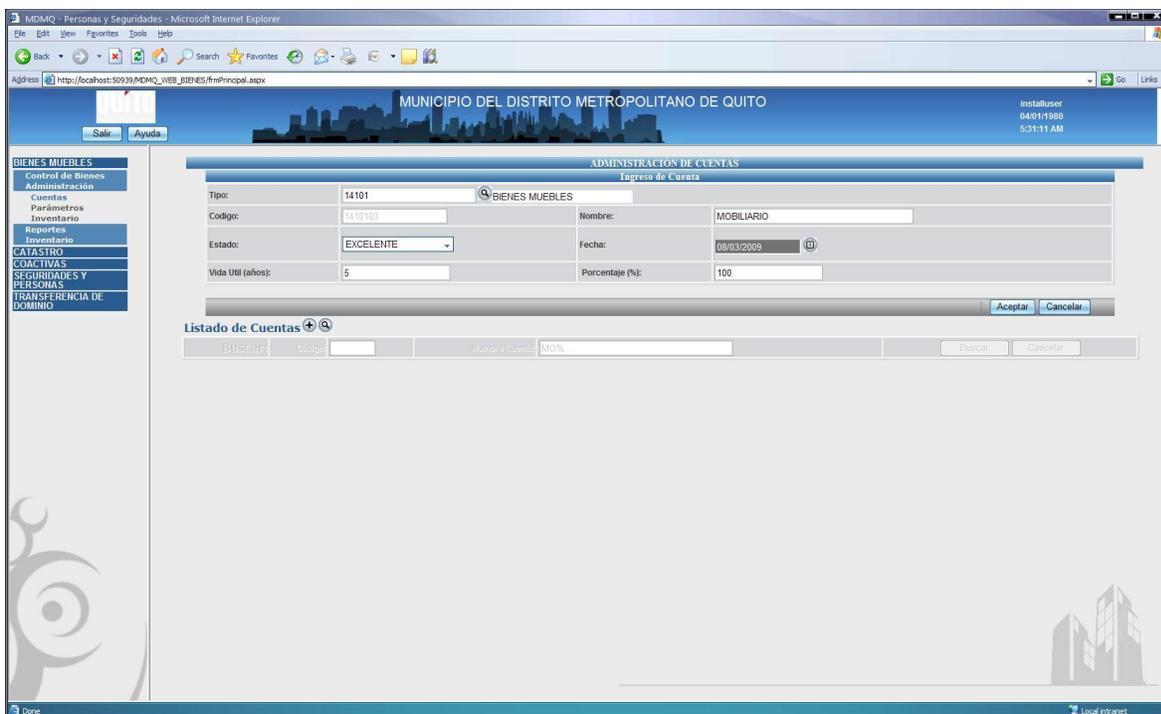


Figura 61. Escenario de Operaciones de Cuentas

En la cuenta se podrá crear Bienes Compuestos o Simples dependiendo las circunstancias, y su vida útil dependerá del Grupo General de la Cuenta.

Para distinguir el nivel de cada clase se creará un catálogo con todas las clases de los Bienes.

La creación de una nueva Cuenta incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.3. Operaciones

Personas

El Usuario MDMQ debe realizar la Creación, Consulta, Actualización o Dar de Baja a los registros de los Bienes Municipales, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

Se mostrara la pantalla principal en el cual el Usuario-MDMQ deberá escoger la opción que desee como puede ser añadir, modificar o dar de baja un bien municipal.

En la operación de ingreso se realizara por etapas ya que es un requerimiento de usuario, en la primera etapa necesitaríamos ingresar datos como la foto del Bien, la clase del bien y el numero secuencial ascendente para la generación del código de barra.

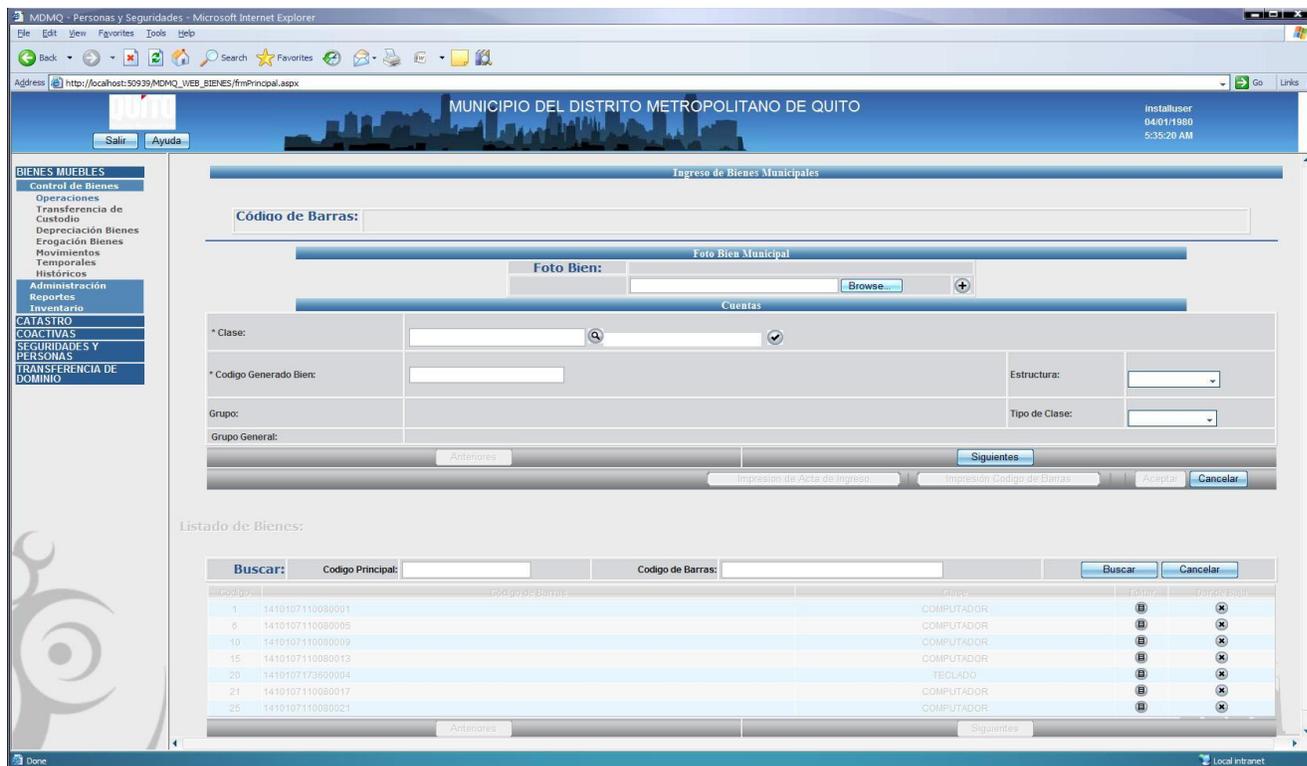


Figura 62. Escenario de Operaciones

El escenario constara con la generación de Código de Barra.

	Codigo de Barras	Codigo	Clase
1	Distrito Metropolitano de Quito  914101071100800259 COMPUTADOR	141010711008	COMPUTADOR
2	Distrito Metropolitano de Quito  914101071100800269 COMPUTADOR MONITOR	141010714836	MONITOR
3	Distrito Metropolitano de Quito  914101071100800279 COMPUTADOR MOUSE	141010714841	MOUSE
4	Distrito Metropolitano de Quito  914101071100800289 COMPUTADOR TECLADO	141010717360	TECLADO

Figura 63. Escenario de Código de Barras

En el escenario de operaciones se tendrá la opción de modificar un Bien siempre y cuando no modifique su código de barra generado.

Si el Usuario-MDMQ escoge la opción dar de Baja, todo el Bien Municipal deberá dar de baja lógicamente, lo cual significa el cambio de estado de un Bien ya que por cuestión de Reportes no se podrá borrar de la base de datos.

La creación de un nuevas Operaciones incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.4. Históricos

Personas

El Usuario MDMQ debe realizar la Consulta de históricos de los registros de los Bienes Municipales, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

Muestra los históricos de los ingresos y modificación de los Bienes Municipales y siguiendo con el estándar de las demás pantallas se podrá buscar el Bien por su código de Barra o su codificación normal.

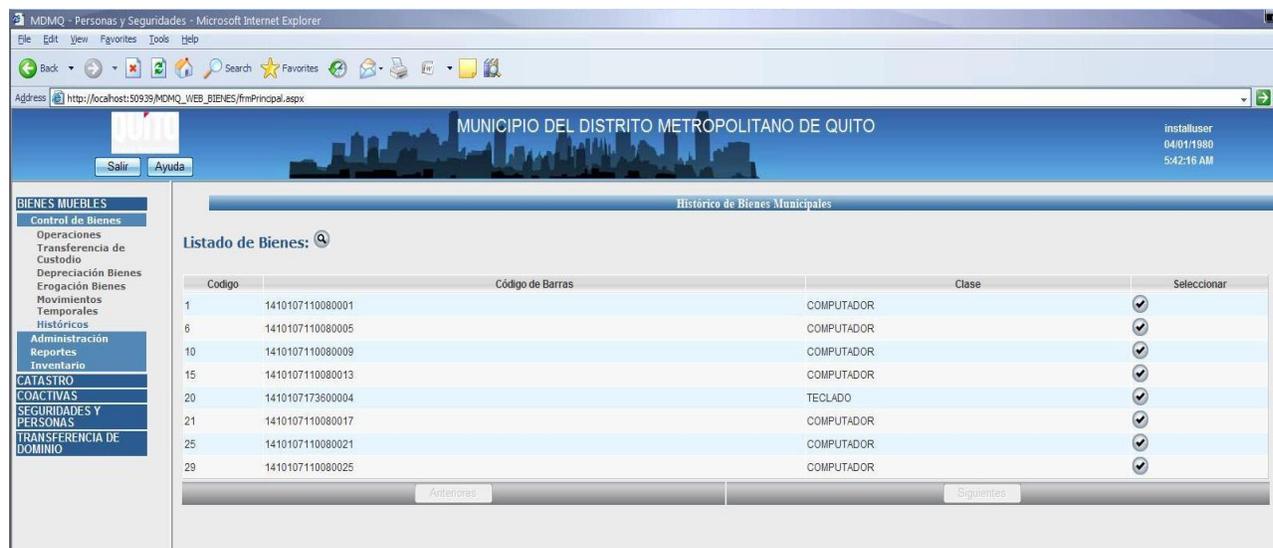


Figura 64. Escenario de Históricos

Una vez seleccionado el Bien Municipal el sistema identificara si es un Bien Simple o Compuesto y su información serán mostrados en pantalla.

La creación de un nuevo Histórico incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.5. Erogaciones

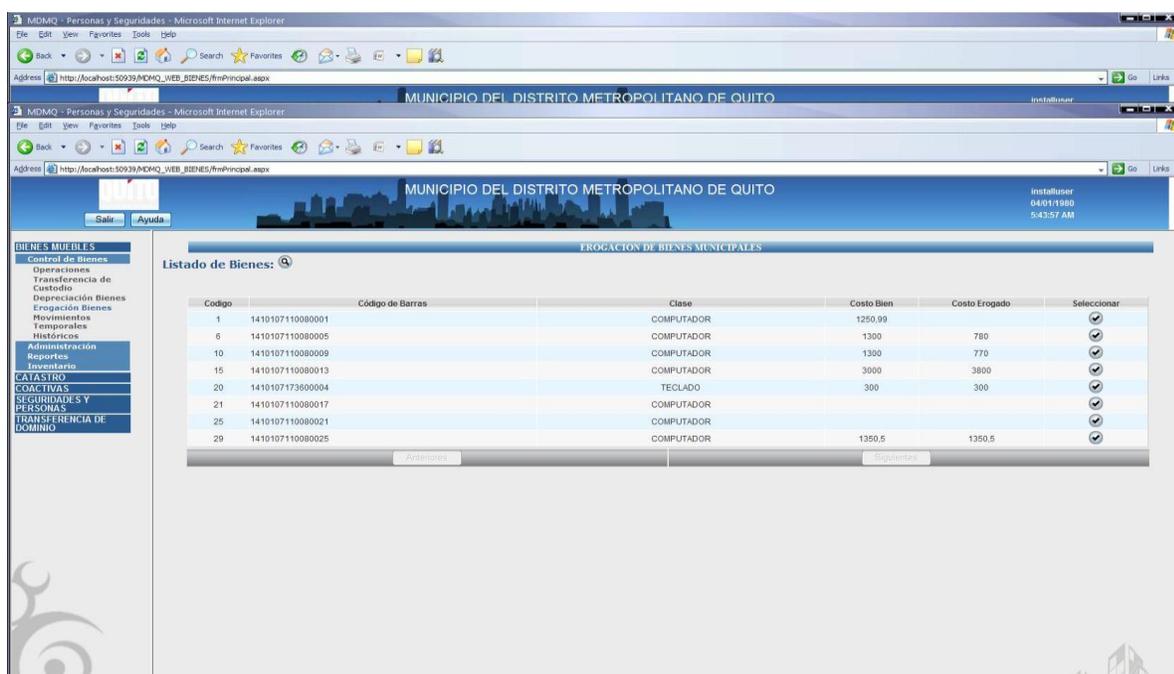
Personas

El Usuario MDMQ debe realizar las erogaciones de los registros de los Bienes Municipales, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

Las adiciones, mejoras y reposiciones vitales destinadas a aumentar la vida útil estimada, la capacidad productiva, o el valor de los bienes de larga duración, de conformidad con los criterios técnicos, deberán registrarse como erogaciones capitalizables que aumentan el valor contable del bien. Los costos asignados a las partes o piezas que se reemplacen o sustituyan, deberán disminuir el valor contable respectivo; así como también la correspondiente Depreciación Acumulada.

El valor de la erogación se expresa como: Valor contable + Valor de erogación



The screenshot displays a web application interface for the 'MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO'. The main content area is titled 'EROGACION DE BIENES MUNICIPALES' and shows a 'Listado de Bienes' table. The table has the following columns: 'Codigo', 'Codigo de Barras', 'Clase', 'Costo Bien', 'Costo Erogado', and 'Seleccionar'. The data rows are as follows:

Codigo	Codigo de Barras	Clase	Costo Bien	Costo Erogado	Seleccionar
1	1410107110080001	COMPUTADOR	1250,99		<input checked="" type="checkbox"/>
6	1410107110080005	COMPUTADOR	1300	780	<input checked="" type="checkbox"/>
10	1410107110080009	COMPUTADOR	1300	770	<input checked="" type="checkbox"/>
15	1410107110080013	COMPUTADOR	3000	3800	<input checked="" type="checkbox"/>
20	1410107173800004	TECLADO	300	300	<input checked="" type="checkbox"/>
21	1410107110080017	COMPUTADOR			<input checked="" type="checkbox"/>
25	1410107110080021	COMPUTADOR			<input checked="" type="checkbox"/>
29	1410107110080025	COMPUTADOR	1350,5	1350,5	<input checked="" type="checkbox"/>

Figura 65. Escenario de Erogaciones

Un Bien Compuesto se divide según sus elementos, para la erogación primero debemos dar de baja un elemento del Bien Municipal para poder remplazarlo por otro.

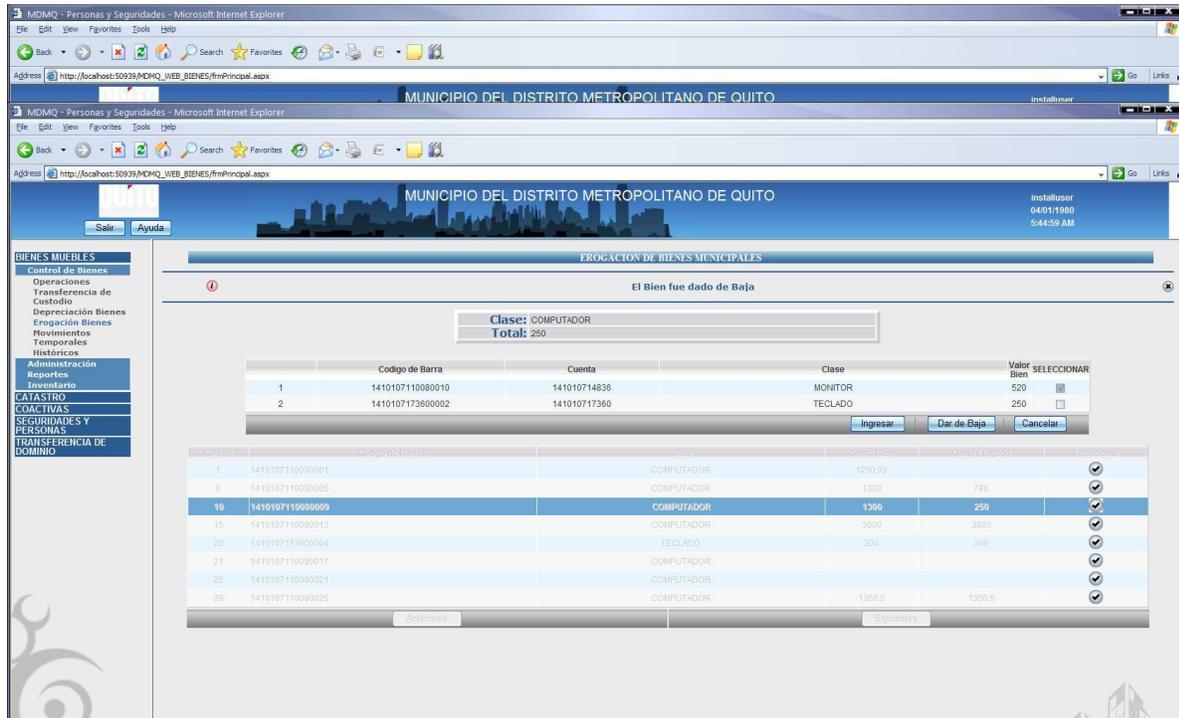


Figura 66. Escenario de Operaciones de Erogaciones

La creación de una nueva Erogación incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.6. Depreciaciones

Personas

El Usuario MDMQ debe realizar las depreciaciones de los registros de los Bienes Municipales, para lo cual debe esperar los periodos que se han determinado por el MDMQ para depreciar sus bienes.

Descripción del Escenario

Los valores contables del bien deberán disminuir su valor con la depreciación ya que por cada periodo de tiempo el usuario realizara el proceso de depreciación de acuerdo a la vida útil del bien.

Se asignaran intervalos de tiempo en cada depreciación y durante ese periodo de tiempo el sistema denegara al usuario el realizar la operación.

FECHA ACTUAL:	viernes, 12 de junio de 2009
FECHA INICIAL:	01/07/2008
FECHA FINAL:	30/06/2009

Figura 67. Escenario de Depreciaciones

La ejecución de la Depreciación incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.7. Transferencia de Custodio

Personas

El Usuario MDMQ debe realizar las transferencias de los registros de los Bienes Municipales, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

Un bien puede tomar otro custodio a lo largo de su vida útil, es por eso que se crea este escenario el cual consta en el traslado de un bien a una nueva ubicación y a un nuevo custodio.

A su vez el Bien Municipal será trasladado tanto sus componentes como sus accesorios para que una nueva persona se haga cargo del nuevo bien.

TRANSFERENCIA CUSTODIO			
Listado de Bienes: 🔍			
Codigo	Código de Barras	Clase	Seleccionar
9	1410107110080005	COMPUTADOR	<input checked="" type="checkbox"/>
13	1410107110080009	COMPUTADOR	<input checked="" type="checkbox"/>

Anteriores
Siguientes



Figura 68. Escenario de Transferencia de Custodio

La ejecución de la Transferencia de Custodio incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.8. Movimientos Temporales

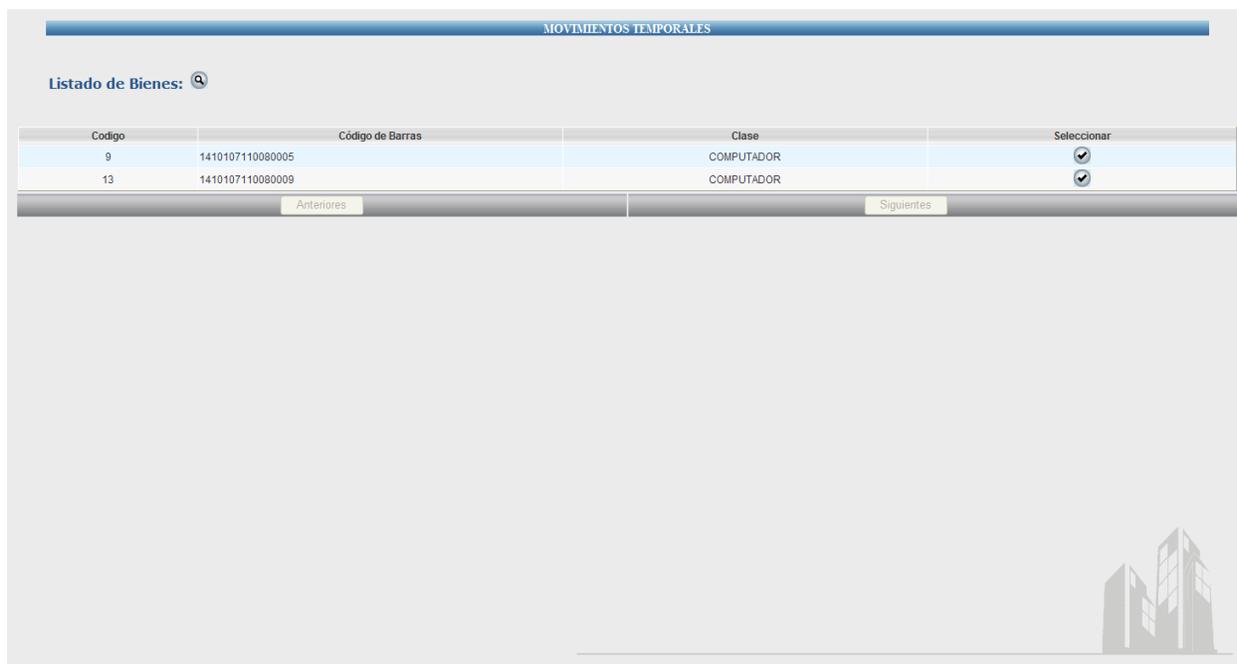
Personas

El Usuario MDMQ puede realizar transferencias temporales de los registros de los Bienes Municipales, para lo cual debe ingresar a la pantalla correspondiente y realizar las operaciones requeridas.

Descripción del Escenario

Un bien puede tomar otro custodio temporalmente a lo largo de su vida útil, es por eso que se crea este escenario el cual consta en el traslado de un bien temporalmente a una nueva ubicación y a un nuevo custodio.

A su vez el Bien Municipal será trasladado tanto sus componentes como sus accesorios para que una nueva persona se haga cargo del nuevo bien.



Codigo	Código de Barras	Clase	Seleccionar
9	1410107110080005	COMPUTADOR	<input checked="" type="checkbox"/>
13	1410107110080009	COMPUTADOR	<input checked="" type="checkbox"/>

Anteriores Sigüientes

Figura 69. Escenario de Movimientos Temporales

La ejecución de Movimientos Temporales incluye además la creación de los parámetros básicos de seguridad de la misma. Estos son:

- ✚ Tiempo máximo de duración de la sesión en horas. Default = 8
- ✚ Número de días para validez de la clave: Default = 30 días
- ✚ Número de días para aviso de caducidad de la clave. Default= 10

4.1.9. Reportes

4.1.9.1. Reporte Detalle Bien Personas

El Usuario MDMQ puede realizar una consulta de un bien por medio de parámetros.

Descripción del Escenario

Un Bien puede ser consultado de acuerdo a varios atributos que el usuario deberá escoger al momento de realizar un reporte.

REPORTES DE DETALLE DE BIEN	
Codigo Bien:	<input type="text"/>
Codigo de Barras Actual:	<input type="text"/>
Codigo Barras Anterior:	<input type="text"/>
Codigo NIM:	<input type="text"/>

Figura 70. Escenario Reporte Bien Detalle

4.1.9.2. Reporte Depreciación Personas

El Usuario MDMQ puede realizar una consulta de un bien por medio de parámetros.

Descripción del Escenario

Un Bien puede ser consultado de acuerdo a varios atributos que el usuario deberá escoger al momento de realizar un reporte.

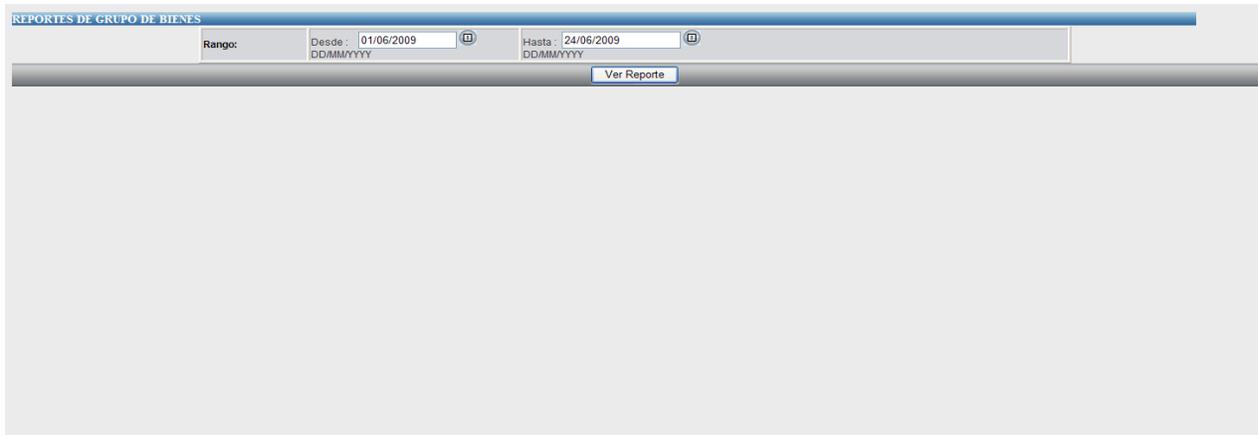


Figura 71. Escenario Reporte Depreciación

4.1.9.3. Reporte Grupo Bien Personas

El Usuario MDMQ puede realizar una consulta de varios bienes por medio de parámetros.

Descripción del Escenario

Varios bienes pueden ser consultados de acuerdo a varios atributos que el usuario deberá escoger al momento de realizar un reporte.

Figura 72. Escenario Reporte Grupo de Bienes

4.2. AMENAZAS Y VULNERABILIDADES

4.2.1. Objetivos

- ✚ Obtener información efectiva de los registros del sistema de Bienes Municipales.
- ✚ Organizar los registros de los Bienes Municipales.
- ✚ Implementación de un Sistema Informático a partir de una modelo como lo es el Sistema de Personas.
- ✚ Implementar una metodología de desarrollo (MSF Agile).
- ✚ Parametrizar las características de los Bienes para una mejor administración.
- ✚ Proporcionar un adecuado control a los Bienes Municipales.
- ✚ Proporcionar una debida codificación de los Bienes propuesta por el departamento Gubernamental.

4.2.2. Fuera de Alcance en los Objetivos

- ✚ Implementación a producción del proyecto SACBIM.
- ✚ Migración total de los Datos para el Sistema de Administración de Bienes de Larga Duración y Bienes Sujetos de Control.
- ✚ Cualquier desarrollo en IBM390.

- ✚ Realizar cambios en los siguientes otros sistemas informáticos del MDMQ como: RRHH y Financiero.
- ✚ Operación o Producción del Sistema de Personas, el mismo que estará a cargo del personal del MDMQ.
- ✚ Procesos de respaldos o backup de los datos del Sistema.

4.2.3. Escenarios Principales

- ✚ Administración de los datos de Bienes: El usuario podrá realizar la administración de parámetros y cuentas contables las cuales necesitan realizar una acción de inserción, modificación, borrado, actualización y consulta de registros de Bienes.
- ✚ Control de Bienes: El usuario debe realizar las operaciones necesarias para realizar la administración de Bienes Municipales, Cada proceso debe ser independiente a otro.
- ✚ Reportes: El usuario deberá escoger los reportes que sean necesarios para la administración de los Bienes Municipales.

4.2.4. Tecnologías

- ✚ Servidor Web: Internet Information Server (IIS).
- ✚ Lógica de presentación: ASP.NET (C#).
- ✚ Lógica de Negocios: Librerías de clases en C#.
- ✚ Lógica de Acceso a Datos: MDMQ_CLS_TransSrvOracle (componente desarrollado para facilitar el acceso a los datos) y Procedimientos almacenados.
- ✚ Servidor de Base de Datos: Oracle 10g.

4.2.5. Mecanismos de Seguridad de las Aplicaciones

- ✚ El ingreso al sistema deberá ser controlado por el Sistema.
- ✚ Los procesos administrativos se realizan solamente en aplicaciones internas, en las cuales el usuario se ha autenticado previamente en el sistema. No se permite la administración por medio de accesos remotos.
- ✚ Se validara todas las páginas para que salte mensajes de error en el momento que ocurran.

4.2.6. Elementos de la Aplicación

4.2.6.1. Flujos de Datos

A través de la página de Ingreso al “Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control” el usuario ingresa su Login y Clave, los cuales son validados de acuerdo a la estructura de seguridades registrada en la base de datos, y solo si son correctos, se permite el ingreso del usuario al Sistema.

Cuando el Usuario desea ejecutar una transacción, se debe validar si el usuario se encuentra autorizado para poder ejecutar dicha transacción.

Cualquier dato que el usuario ingrese al sistema será enviado a su respectiva reposición de datos.

4.2.6.2. Puntos de Entrada

- ✚ Página de Login (Ingreso). Es accesible a todos los usuarios de la intranet; los datos de esta página se validan a nivel del cliente y del servidor utilizando controles disponibles en la librería.
- ✚ Consulta de parámetros en la Base de Datos.

4.2.6.3. Puntos de Salida

- ✚ La comunicación con ORACLE.
- ✚ Las consultas con la información requerida obtenida de la Base de Datos.
- ✚ Las transacciones correctamente a la Base de Datos sin que ocurra excepciones del sistema.
- ✚ Los reportes.

4.2.7. Amenazas

- ✦ La falta de apoyo por parte del cliente puede causar que no se cumplan los requerimientos a cabalidad.
- ✦ Se debe evaluar la totalidad de los escenarios de llamadas que van a ser considerados en el procesamiento, pues la omisión de algún caso o la falta de comprensión del mismo puede ser la causa para que su procesamiento sea realizado en forma incorrecta.
- ✦ La falta de un equipo de control de calidad puede ocasionar la obtención de un producto que no cumpla con las especificaciones establecidas o que se cumplan parcialmente.
- ✦ En caso de no disponer de ambientes separados de desarrollo, control de calidad y producción se corre peligro de que el producto terminado no cumpla con las especificaciones requeridas y la salida en línea puede causar problemas. Además, en caso de contingencias no se dispondría de instalables para nuevos equipos.
- ✦ Un atacante (hacker) obtiene el nombre de usuario y la clave monitoreando la red entre el navegador y el servidor de Web y puede cambiar valores importantes que ocasionarían un daño para la institución.
- ✦ Manipulación de Datos mediante inyecciones de SQL.
- ✦ Un atacante realiza una inyección de SQL utilizando la pantalla de búsquedas.
- ✦ Un atacante lanza un ataque de diccionario hacia la pantalla de login.
- ✦ Manipulación de parámetros. Un usuario malicioso altera los datos de un parámetro con el objetivo de influenciar las decisiones realizadas por la aplicación.
- ✦ Acceso no autorizado. Un atacante intenta adivinar el nombre de cuenta de un usuario y su clave con el objetivo de ganar acceso legítimo a la aplicación.

4.2.8. Vulnerabilidades

- ✚ Si el sistema Web llega a ser invadido por un atacante, este podría asignarse un nivel de usuario con todos los privilegios, lo que le permitiría realizar un gran daño al sistema.
- ✚ No actualizar los datos del sistema cuando corresponda.
- ✚ Gran número de usuario utilizando el mismo sistema podría causar lentitud en consulta de datos.
- ✚ Demasiada confianza en la validación de la entrada de datos desde el cliente.
- ✚ Construcción dinámica de instrucciones SQL sin utilizar parámetros con tipos.
- ✚ Cadenas de conexión a la base de datos en texto claro.
- ✚ Secretos almacenados en código.
- ✚ Tener conocimiento de claves de acceso al sistema podría causar daños e inconsistencia de datos.

4.3. PRUEBAS

4.3.1. Pruebas Unitarias

Las pruebas unitarias sirven para utilizar otro código fuente llamando directamente a los métodos de una clase, pasando los parámetros apropiados, y posteriormente, si se incluyen instrucciones Assert, pueden comparar los valores que se generan con los valores esperados. Los métodos de pruebas unitarias residen en clases Test, que se almacenan en archivos de código fuente.

Con pruebas unitarias, no sólo se pueden probar métodos públicos, sino también privados. En ambos casos, las pruebas unitarias se crean automáticamente, desde el código que se desea probar. Visual Studio genera los resultados directamente.

Para la realización de pruebas el Visual Studio creara un proyecto aparte denominado MDMQ_PRUEBAS_SACBIM el cual contendrá todos los métodos de pruebas generados por el sistema.

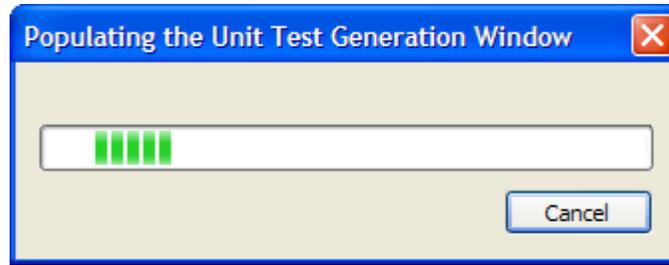


Figura 73. Creación de Proyecto de Pruebas

Para generar las pruebas se deberá escoger en la capa de MDMQ_DAL_BIENES los métodos que el usuario desea probar.

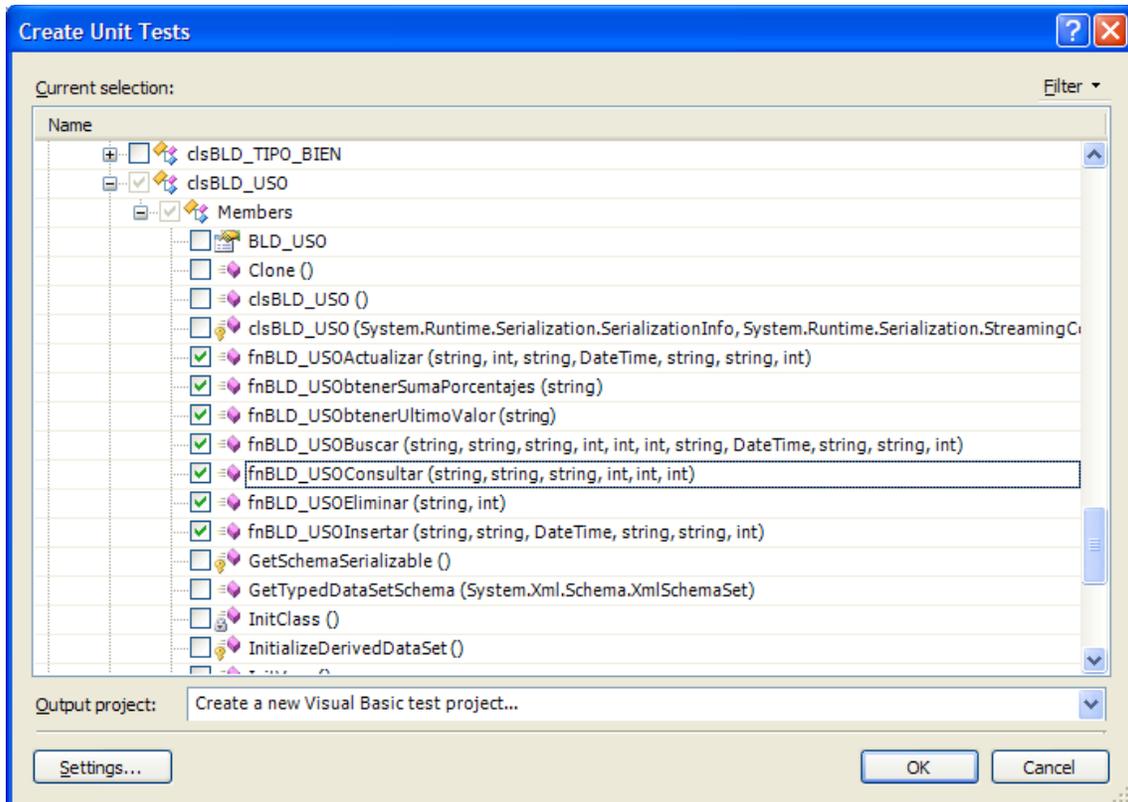


Figura 74. Métodos de la DAL de Bienes

Se generaran automáticamente los métodos de prueba de cada método deseado.
La estructura que tendrá para todas las tablas será la misma que la siguiente.

PRUEBAS USO

```
/// <summary>
///A test for fnBLD_USOActualizar (string, int, string, DateTime, string, string, int)
///</summary>
[TestMethod()]
public void fnBLD_USOActualizarTest()
{
    clsBLD_USO target = new clsBLD_USO();

    string strIdentifier = null; // TODO: Initialize to an appropriate value

    int intUS_USO_ID = 0; // TODO: Initialize to an appropriate value

    string strUS_NOMBRE = null; // TODO: Initialize to an appropriate value

    DateTime dtmUS_FECHA = new DateTime(); // TODO: Initialize to an appropriate value

    string strUS_ESTADO = null; // TODO: Initialize to an appropriate value
    string strUS_COD_USUARIO = null; // TODO: Initialize to an appropriate value
    int intUS_PORCENTAJE = 0; // TODO: Initialize to an appropriate value
    int expected = 0;
    int actual;

    actual = target.fnBLD_USOActualizar(strIdentifier, intUS_USO_ID, strUS_NOMBRE,
    dtmUS_FECHA, strUS_ESTADO, strUS_COD_USUARIO, intUS_PORCENTAJE);

    Assert.AreEqual(expected, actual,
    "MDMQ_DAL_BIENES.clsBLD_USO.fnBLD_USOActualizar did not return the expected
    value." +
    "");
    Assert.Inconclusive("Verify the correctness of this test method.");
}

/// <summary>
///A test for fnBLD_USOObtenerSumaPorcentajes (string)
///</summary>
[TestMethod()]
public void fnBLD_USOObtenerSumaPorcentajesTest()
{
    clsBLD_USO target = new clsBLD_USO();
    string strIdentifier = null; // TODO: Initialize to an appropriate value
```

```

int expected = 0;
int actual;

actual = target.fnBLD_USObtenerSumaPorcentajes(strIdentifier);

Assert.AreEqual(expected, actual,
"MDMQ_DAL_BIENES.clsBLD_USO.fnBLD_USObtenerSumaPorcentajes did not return
the expected value.");
Assert.Inconclusive("Verify the correctness of this test method.");
}

/// <summary>
///A test for fnBLD_USObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_USObtenerUltimoValorTest()
{
clsBLD_USO target = new clsBLD_USO();

string strIdentifier = null; // TODO: Initialize to an appropriate value

int expected = 0;
int actual;

actual = target.fnBLD_USObtenerUltimoValor(strIdentifier);

Assert.AreEqual(expected, actual,
"MDMQ_DAL_BIENES.clsBLD_USO.fnBLD_USObtenerUltimoValor did not return the
expected value.");
Assert.Inconclusive("Verify the correctness of this test method.");
}

/// <summary>
///A test for fnBLD_USOBuscar (string, string, string, int, int, int, string, DateTime, string,
string, int)
///</summary>
[TestMethod()]
public void fnBLD_USOBuscarTest()
{
clsBLD_USO target = new clsBLD_USO();

string strIdentifier = null; // TODO: Initialize to an appropriate value

```

```

string strIndice = null; // TODO: Initialize to an appropriate value
string strQuery = null; // TODO: Initialize to an appropriate value
int intModo = 0; // TODO: Initialize to an appropriate value
int intFilas = 0; // TODO: Initialize to an appropriate value
int intUS_USO_ID = 0; // TODO: Initialize to an appropriate value
string strUS_NOMBRE = null; // TODO: Initialize to an appropriate value
DateTime dtmUS_FECHA = new DateTime(); // TODO: Initialize to an appropriate value
string strUS_ESTADO = null; // TODO: Initialize to an appropriate value
string strUS_COD_USUARIO = null; // TODO: Initialize to an appropriate value
int intUS_PORCENTAJE = 0; // TODO: Initialize to an appropriate value

int expected = 0;
int actual;

actual = target.fnBLD_USOBuscar(strIdentifier, strIndice, strQuery, intModo, intFilas,
intUS_USO_ID, strUS_NOMBRE, dtmUS_FECHA, strUS_ESTADO,
strUS_COD_USUARIO, intUS_PORCENTAJE);

Assert.AreEqual(expected, actual,
"MDMQ_DAL_BIENES.clsBLD_USO.fnBLD_USOBuscar did not return the expected
value.");
Assert.Inconclusive("Verify the correctness of this test method.");
}

/// <summary>
///A test for fnBLD_USOConsultar (string, string, string, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_USOConsultarTest()
{
clsBLD_USO target = new clsBLD_USO();

string strIdentifier = null; // TODO: Initialize to an appropriate value

string strIndice = null; // TODO: Initialize to an appropriate value

string strQuery = null; // TODO: Initialize to an appropriate value

```

```

int intModo = 0; // TODO: Initialize to an appropriate value

int intFilas = 0; // TODO: Initialize to an appropriate value

int intUS_USO_ID = 0; // TODO: Initialize to an appropriate value

int expected = 0;
int actual;

actual = target.fnBLD_USOConsultar(strIdentifier, strIndice, strQuery, intModo, intFilas,
intUS_USO_ID);

Assert.AreEqual(expected, actual,
"MDMQ_DAL_BIENES.clsBLD_USO.fnBLD_USOConsultar did not return the expected
value.");
Assert.Inconclusive("Verify the correctness of this test method.");
}

/// <summary>
///A test for fnBLD_USOEliminar (string, int)
///</summary>
[TestMethod()]
public void fnBLD_USOEliminarTest()
{
clsBLD_USO target = new clsBLD_USO();

string strIdentifier = null; // TODO: Initialize to an appropriate value

int intUS_USO_ID = 0; // TODO: Initialize to an appropriate value

int expected = 0;
int actual;

actual = target.fnBLD_USOEliminar(strIdentifier, intUS_USO_ID);

Assert.AreEqual(expected, actual,
"MDMQ_DAL_BIENES.clsBLD_USO.fnBLD_USOEliminar did not return the expected
value.");
Assert.Inconclusive("Verify the correctness of this test method.");
}

```

```

/// <summary>
///A test for fnBLD_USOInsertar (string, string, DateTime, string, string, int)
///</summary>
[TestMethod()]
public void fnBLD_USOInsertarTest()
{
clsBLD_USO target = new clsBLD_USO();

string strIdentifier = null; // TODO: Initialize to an appropriate value

string strUS_NOMBRE = null; // TODO: Initialize to an appropriate value

DateTime dtmUS_FECHA = new DateTime(); // TODO: Initialize to an appropriate value

string strUS_ESTADO = null; // TODO: Initialize to an appropriate value

string strUS_COD_USUARIO = null; // TODO: Initialize to an appropriate value

int intUS_PORCENTAJE = 0; // TODO: Initialize to an appropriate value

int expected = 0;
int actual;

actual = target.fnBLD_USOInsertar(strIdentifier, strUS_NOMBRE, dtmUS_FECHA,
strUS_ESTADO, strUS_COD_USUARIO, intUS_PORCENTAJE);

Assert.AreEqual(expected, actual,
"MDMQ_DAL_BIENES.clsBLD_USO.fnBLD_USOInsertar did not return the expected
value.");
Assert.Inconclusive("Verify the correctness of this test method.");
}

```

PRUEBAS TIPO BIEN

```

/// <summary>
///A test for fnBLD_TIPO_BIENActualizar (string, int, string, string, DateTime, string,
string)
///</summary>
[TestMethod()]
public void fnBLD_TIPO_BIENActualizarTest()
{
}

```

```
/// <summary>  
///A test for fnBLD_TIPO_BIENBuscar (string, string, string, int, int, int, string, string,  
DateTime, string, string)  
///</summary>
```

```
[TestMethod()]  
public void fnBLD_TIPO_BIENBuscarTest()  
{  
}
```

```
/// <summary>  
///A test for fnBLD_TIPO_BIENConsultar (string, string, string, int, int, int)  
///</summary>
```

```
[TestMethod()]  
public void fnBLD_TIPO_BIENConsultarTest()  
{  
}
```

```
/// <summary>  
///A test for fnBLD_TIPO_BIENEliminar (string, int)  
///</summary>
```

```
[TestMethod()]  
public void fnBLD_TIPO_BIENEliminarTest()  
{  
}
```

```
/// <summary>  
///A test for fnBLD_TIPO_BIENInsertar (string, string, string, DateTime, string, string)  
///</summary>
```

```
[TestMethod()]  
public void fnBLD_TIPO_BIENInsertarTest()  
{  
}
```

```
/// <summary>  
///A test for fnBLD_TIPOBIENbtenerUltimoValor (string)  
///</summary>
```

```
[TestMethod()]  
public void fnBLD_TIPOBIENbtenerUltimoValorTest()  
{  
}
```

PRUEBA TIPO

```
/// <summary>
///A test for fnBLD_TIPOActualizar (string, int, string, DateTime, string, string, string)
///</summary>
[TestMethod()]
public void fnBLD_TIPOActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIPOObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_TIPOObtenerUltimoValorTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIPOBuscar (string, string, string, int, int, int, string, DateTime, string,
string, string)
///</summary>
[TestMethod()]
public void fnBLD_TIPOBuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIPOConsultar (string, string, string, int, int, int, string)
///</summary>
[TestMethod()]
public void fnBLD_TIPOConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIPOEliminar (string, int)
///</summary>
[TestMethod()]
public void fnBLD_TIPOEliminarTest()
{
}
```

✚ PRUEBA TIPO MOVIMIENTO

```
/// <summary>
///A test for fnBLD_TIP_MOVIMActualizar (string, int, string, DateTime, string, string)
///</summary>
[TestMethod()]
public void fnBLD_TIP_MOVIMActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIP_MOVIMBuscar (string, string, string, int, int, int, string, DateTime,
string, string)
///</summary>
[TestMethod()]
public void fnBLD_TIP_MOVIMBuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIP_MOVIMConsultar (string, string, string, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_TIP_MOVIMConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIP_MOVIMEliminar (string, int)
///</summary>
[TestMethod()]
public void fnBLD_TIP_MOVIMEliminarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_TIP_MOVIMInsertar (string, string, DateTime, string, string)
///</summary>
[TestMethod()]
public void fnBLD_TIP_MOVIMInsertarTest()
{
}
```

✚ PRUEBA PERSONAS

```
/// <summary>
///A test for fnBLD_PERSONASBuscar (string, string, string, int, int, int, string, string,
string, string, DateTime, int)
///</summary>
[TestMethod()]
public void fnBLD_PERSONASBuscarTest()
{
}

/// <summary>
///A test for fnBLD_PERSONASConsultar (string, string, string, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_PERSONASConsultarTest()
{
}
```

✚ PRUEBA PERIODOS

```
/// <summary>
///A test for fnBLD_PERIODOSActualizar (string, int, DateTime, DateTime, string,
DateTime, DateTime)
///</summary>
[TestMethod()]
public void fnBLD_PERIODOSActualizarTest()
{
}

/// <summary>
///A test for fnBLD_PERIODOSBuscar (string, string, string, int, int, int, DateTime,
DateTime, string, DateTime, DateTime)
///</summary>
[TestMethod()]
public void fnBLD_PERIODOSBuscarTest()
{
}

/// <summary>
///A test for fnBLD_PERIODOSInsertar (string, DateTime, DateTime, string, DateTime,
DateTime)
///</summary>
[TestMethod()]
public void fnBLD_PERIODOSInsertarTest()
{
}
```

PRUEBAS OPERACIÓN

```
/// <summary>
///A test for fnBLD_OPERACIONActualizar (string, int, int, DateTime, decimal, decimal,
decimal, decimal, decimal, decimal, int, string, string, int, int)
///</summary>
[TestMethod()]
public void fnBLD_OPERACIONActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_OPERACIONConsultar (string, string, string, int, int, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_OPERACIONConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_OPERACIONInsertar (string, int, DateTime, decimal, decimal,
decimal, decimal, decimal, decimal, int, string, string, int, int)
///</summary>
[TestMethod()]
public void fnBLD_OPERACIONInsertarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_OPERACIONObtenerSecuencial (string, int)
///</summary>
[TestMethod()]
public void fnBLD_OPERACIONObtenerSecuencialTest()
{
}
```

PRUEBAS MOVIMIENTOS

```
/// <summary>
///A test for fnBLD_MOVIMIENTOActualizar (string, int, int, DateTime, DateTime,
DateTime, string, string, string, string, int, int, long, long, int)
///</summary>
[TestMethod()]
public void fnBLD_MOVIMIENTOActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_MOVIMIENTOConsultar (string, string, string, int, int, int, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_MOVIMIENTOConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_MOVIMIENTOInsertar (string, int, DateTime, DateTime, DateTime,
string, string, string, string, int, int, long, long, int)
///</summary>
[TestMethod()]
public void fnBLD_MOVIMIENTOInsertarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_MOVIMIENTOSObtenerSecuencial (string, int)
///</summary>
[TestMethod()]
public void fnBLD_MOVIMIENTOSObtenerSecuencialTest()
{
}
```

PRUEBAS MOTIVOS

```
/// <summary>
///A test for fnBLD_MOTIVOSConsultar (string, string, string, int, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_MOTIVOSConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_MOTIVOSBuscar (string, string, string, int, int, int, string, string,
DateTime, string, int)
///</summary>
[TestMethod()]
public void fnBLD_MOTIVOSBuscarTest()
{
}
```

PRUEBAS HISTORICOS

```
/// <summary>
///A test for fnBLD_HISTORICOConsultar (string, string, string, int, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_HISTORICOConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_HISTORICOEliminar (string, int)
///</summary>
[TestMethod()]
public void fnBLD_HISTORICOEliminarTest()
{
}
```

PRUEBAS GARANTIA

```
/// <summary>
///A test for fnBLD_GARANTIAActualizar (string, int, DateTime, DateTime, int, long, int)
///</summary>
[TestMethod()]
public void fnBLD_GARANTIAActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_GARANTIAConsultar (string, string, string, int, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_GARANTIAConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_GARANTIAInsertar (string, DateTime, DateTime, int, long, int)
///</summary>
[TestMethod()]
public void fnBLD_GARANTIAInsertarTest()
{
}
```

✚ PRUEBAS ESTRUCTURA

```
/// <summary>  
///A test for fnBLD_ESTRUCTURAActualizar (string, int, string, string, DateTime, string,  
string)
```

```
///</summary>
```

```
[TestMethod()]
```

```
public void fnBLD_ESTRUCTURAActualizarTest()
```

```
{  
}
```

```
/// <summary>
```

```
///A test for fnBLD_ESTRUCTURABuscar (string, string, string, int, int, int, string, string,  
DateTime, string, string)
```

```
///</summary>
```

```
[TestMethod()]
```

```
public void fnBLD_ESTRUCTURABuscarTest()
```

```
{  
}
```

```
/// <summary>
```

```
///A test for fnBLD_ESTRUCTURAConsultar (string, string, string, int, int, int, string)
```

```
///</summary>
```

```
[TestMethod()]
```

```
public void fnBLD_ESTRUCTURAConsultarTest()
```

```
{  
}
```

```
/// <summary>
```

```
///A test for fnBLD_ESTRUCTURAEliminar (string, int)
```

```
///</summary>
```

```
[TestMethod()]
```

```
public void fnBLD_ESTRUCTURAEliminarTest()
```

```
{  
}
```

```
/// <summary>
```

```
///A test for fnBLD_ESTRUCTURAIinsertar (string, string, string, DateTime, string, string)
```

```
///</summary>
```

```
[TestMethod()]
```

```
public void fnBLD_ESTRUCTURAIinsertarTest()
```

```
{  
}
```

```
/// <summary>
///A test for fnBLD_ESTRUCTURAObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_ESTRUCTURAObtenerUltimoValorTest()
{
}
```

PRUEBAS ESTADO

```
/// <summary>
///A test for fnBLD_ESTADOActualizar (string, int, string, DateTime, string, string, int)
///</summary>
[TestMethod()]
public void fnBLD_ESTADOActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ESTADOBuscar (string, string, string, int, int, int, string, DateTime,
string, string, int)
///</summary>
[TestMethod()]
public void fnBLD_ESTADOBuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ESTADOConsultar (string, string, string, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_ESTADOConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ESTADOInsertar (string, string, DateTime, string, string, int)
///</summary>
[TestMethod()]
public void fnBLD_ESTADOInsertarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ESTADOObtenerSumaPorcentajes (string)
///</summary>
[TestMethod()]
public void fnBLD_ESTADOObtenerSumaPorcentajesTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ESTADOObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_ESTADOObtenerUltimoValorTest()
{
}
```

PRUEBAS CUENTA

```
/// <summary>
///A test for fnBLD_CUENTABuscar (string, string, string, int, int, string, string, string, int,
int, int, int, DateTime, string, string, int, int)
///</summary>
[TestMethod()]
public void fnBLD_CUENTABuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_CUENTAConsultar (string, string, string, int, int, string, string)
///</summary>
[TestMethod()]
public void fnBLD_CUENTAConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_CUENTAEliminar (string, string)
///</summary>
[TestMethod()]
public void fnBLD_CUENTAEliminarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_CUENTAInsertar (string, string, string, string, int, int, int, int,
DateTime, string, string, int, int)
///</summary>
[TestMethod()]
public void fnBLD_CUENTAInsertarTest()
{
}
```

PRUEBAS COLOR

```
/// <summary>
///A test for fnBLD_COLORActualizar (string, int, string, string, DateTime, string)
///</summary>
[TestMethod()]
public void fnBLD_COLORActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_COLORBuscar (string, string, string, int, int, int, string, string,
DateTime, string)
///</summary>
[TestMethod()]
public void fnBLD_COLORBuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_COLORConsultar (string, string, string, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_COLORConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_COLORInsertar (string, string, string, DateTime, string)
///</summary>
[TestMethod()]
public void fnBLD_COLORInsertarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_COLORObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_COLORObtenerUltimoValorTest()
{
    clsBLD_COLOR target = new clsBLD_COLOR();
}
```

PRUEBAS BIEN UBICACIÓN

```
/// <summary>
///A test for fnBLD_BIEN_UBICActualizar (string, int, int, DateTime, DateTime, string,
string, DateTime, string, string)
///</summary>
[TestMethod()]
public void fnBLD_BIEN_UBICActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIEN_UBICConsultar (string, string, string, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_BIEN_UBICConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIEN_UBICInsertar (string, int, DateTime, DateTime, string, string,
DateTime, string, string)
///</summary>
[TestMethod()]
public void fnBLD_BIEN_UBICInsertarTest()
{
}
```

```

/// <summary>
///A test for fnBLD_UBICACIONBIENObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_UBICACIONBIENObtenerUltimoValorTest()
{
}

```

🚦 PRUEBAS BIEN

```

/// <summary>
///A test for fnBLD_BIENActualizar (string, int, string, string, string, string, DateTime, int,
DateTime, string, DateTime, string, string, string, int, decimal, decimal, decimal, int,
string, string, int, DateTime, string, byte[], string, string, string, string, string, int, long, int,
int, int, int, int, int, long, long, int, string, string, string, int)
///</summary>
[TestMethod()]
public void fnBLD_BIENActualizarTest()
/// <summary>
///A test for fnBLD_BIENActualizarDepreciacion (string, decimal, int, decimal)
///</summary>
[TestMethod()]
public void fnBLD_BIENActualizarDepreciacionTest()
{
}

```

```

/// <summary>
///A test for fnBLD_BIENActualizarErogacion (string, decimal, int)
///</summary>
[TestMethod()]
public void fnBLD_BIENActualizarErogacionTest()
{
}

```

```

/// <summary>
///A test for fnBLD_BIENActualizarTransferencia (string, int, int, int)
///</summary>
[TestMethod()]
public void fnBLD_BIENActualizarTransferenciaTest()
{
}

```

```
/// <summary>
///A test for fnBLD_BIENBuscar (string, string, string, int, int, int, string, string, string,
string, DateTime, int, DateTime, string, DateTime, string, string, string, int, decimal,
decimal, decimal, int, string, string, int, DateTime, string, string, string, string, string,
string, int, long, int, int, int, int, int, long, long, int, string, string, string, int)
///</summary>
```

```
[TestMethod()]
public void fnBLD_BIENBuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIENBuscarSecuencial (string, string, string, int, int, string)
///</summary>
```

```
[TestMethod()]
public void fnBLD_BIENBuscarSecuencialTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIENConsultar (string, string, string, int, int, int, int, long, int, int, int,
int, int, int, int, string)
///</summary>
```

```
[TestMethod()]
public void fnBLD_BIENConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIENConsultarDepreciacion (string, string, string, int, int, DateTime,
DateTime)
///</summary>
```

```
[TestMethod()]
public void fnBLD_BIENConsultarDepreciacionTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIENEliminar (string, int)
///</summary>
```

```
[TestMethod()]
public void fnBLD_BIENEliminarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIENeliminarAccesorios (string, int)
///</summary>
[TestMethod()]
public void fnBLD_BIENeliminarAccesoriosTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIENInsertar (string, int, string, string, string, string, DateTime, int,
DateTime, string, DateTime, string, string, string, int, decimal, decimal, decimal, int,
string, string, int, DateTime, string, byte[], string, string, string, string, string, int, long, int,
int, int, int, int, int, long, long, int, string, string, string, int)
///</summary>
[TestMethod()]
public void fnBLD_BIENInsertarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_BIENObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_BIENObtenerUltimoValorTest()
{
}
```

PRUEBAS ACCESORIOS

```
/// <summary>
///A test for fnBLD_ACCESORIOSActualizar (string, int, string, string, string, string,
DateTime, string, string, string, string)
///</summary>
[TestMethod()]
public void fnBLD_ACCESORIOSActualizarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ACCESORIOSBuscar (string, string, string, int, int, int, string, string,
string, string, DateTime, string, string, string, string)
///</summary>
[TestMethod()]
public void fnBLD_ACCESORIOSBuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ACCESORIOSConsultar (string, string, string, int, int, int, string)
///</summary>
[TestMethod()]
public void fnBLD_ACCESORIOSConsultarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ACCESORIOSEliminar (string, int)
///</summary>
[TestMethod()]
public void fnBLD_ACCESORIOSEliminarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_ACCESORIOSObtenerUltimoValor (string)
///</summary>
[TestMethod()]
public void fnBLD_ACCESORIOSObtenerUltimoValorTest()
{
}
```

PRUEBAS UBICACIÓN

```
/// <summary>
///A test for fnBLD_UBICACIONBuscar (string, string, string, int, int, string, string, int,
string)
///</summary>
[TestMethod()]
public void fnBLD_UBICACIONBuscarTest()
{
}
```

```
/// <summary>
///A test for fnBLD_UBICACIONConsultar (string, string, string, int, int, string)
///</summary>
[TestMethod()]
public void fnBLD_UBICACIONConsultarTest()
{
}
```

CAPITULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- ✚ Se obtuvo una optimización de los procesos de control y administración de Bienes, lo cual permitió cumplir con las expectativas generadas por la empresa.
- ✚ Se diseñó una interfaz amigable y flexible lo que permitirá al usuario trabajar de una manera ágil, eficiente y efectiva.
- ✚ El servidor de base de datos Oracle brinda más confiabilidad, seguridad y eficiencia al momento de interactuar con la información almacenada, que el servidor IBM S/390 que trabaja con Cobol.
- ✚ El aplicar estándares en el desarrollo de la automatización de procesos dentro del Municipio del Distrito Metropolitano de Quito son importantes para contar con un sistema de calidad.
- ✚ Por requerimiento del Municipio del Distrito Metropolitano de Quito se realizó el cambio de metodología de Microsoft Solution Framework a Microsoft Solution Framework Agile lo que permitió obtener resultados visibles en cortos periodos de tiempo mediante el uso de los escenarios definidos por iteraciones con lo cual permitió realizar de mejor manera la gestión de calidad para eliminar errores cumpliendo con el objetivo de toda metodología.
- ✚ Se verificó el correcto funcionamiento del sistema mediante la realización de pruebas unitarias, las cuales ayudaron a comprobar el desempeño de las funciones que interactúan con los procesos del sistema.

5.2. RECOMENDACIONES

- ✚ En cuanto al uso de la metodología se recomienda tener más cuidado en las primeras etapas de desarrollo del sistema debido a que los requerimientos son de vital importancia para el producto final y así se evitara cambios que posteriormente produzcan un impacto significativo para la aplicación.
- ✚ Se recomienda la realización de prácticas pre-profesionales orientadas a la aplicación de metodologías, así como herramientas de desarrollo con el fin de incrementar el conocimiento y la destreza del estudiante.
- ✚ Se aconseja al Municipio del Distrito Metropolitano de Quito hacer uso de un dispositivo móvil que interactúe con el Sistema de Bienes Municipales para verificar que la información ingresada sea la correcta.
- ✚ Se recomienda a futuro al Municipio del Distrito Metropolitano de Quito que se automaticen los procesos de inventario y bodega debido a que son de gran importancia para la administración y control de los Bienes.
- ✚ Se necesita constar con un buen servidor de base de datos el cual permita tener un buen rendimiento en la aplicación.

Referencias Bibliográficas:

- ✚ Tomado de: <http://www.librosweb.es/>
- ✚ Tomado de: <http://msdn.microsoft.com/es-es/library/bb398874.aspx>
- ✚ Tomado de: <http://technet2.microsoft.com/windowsserver/es/library/935f1331-8872-48b9-a3dc-d59c88dde2c53082.mspx?mfr=true>
- ✚ Tomado de: http://www.programacion.com/asp/articulo/aspnet_quees/
- ✚ Tomado de: <http://www.softonic.com/pocketpc/programacion>
- ✚ Tomado de: <http://www.dotnetheaven.com/UploadFile/ursbgiri/.aspx>
- ✚ Tomado de: [http://msdn.microsoft.com/es-es/library/fda2bad5\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/fda2bad5(VS.80).aspx)
- ✚ Tomado de: <http://www.progratium.com/manuales/sql/sql000.htm>
- ✚ Tomado de: Microsoft Solutions Framework site at <http://www.microsoft.com/MSF>
- ✚ Tomado de: <http://www.microsoft.com/spanish/msdn/vs2005/editions/team.mspix>
- ✚ Tomado de: Desarrollador 5 estrellas Microsoft site at <http://www.mslatam.com/latam/msdn/comunidad/msdn2005/>
- ✚ Tomado de: <http://www.codesmithtools.com/>
- ✚ Tomado de: <http://www.orape.net/article68.html>
- ✚ Tomado de: http://education.oracle.com/pls/web_prod-plq-dad/web_prod.view_pdf
- ✚ Tomado de: http://www.miliuco.net/docs/oracle_select.htm
- ✚ Tomado de: <http://www.monografias.com/trabajos48/desarrollo-software-agil/desarrollo-software-agil.shtml>
- ✚ Tomado de: <http://mexinetica.com/~lanjoe9/costos05-2/equipo02-valoractual-neto.ppt.ppt>
- ✚ Tomado de: <http://www.asfi.gov.bo/archivos/Editorial0805.pdf>
- ✚ Tomado de: <http://scruz334.blogspot.es/i2007-11/>

ANEXOS

Manual Técnico

Sistema de Bienes SACBIM

Preparado para:

Municipio del Distrito Metropolitano de Quito

10 de junio de 2009

Versión 1.0 Release

Preparado por:

Sr. Rolando Orozco

Resumen

El Gobierno Electrónico del Municipio del Distrito Metropolitano de Quito implementó el Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control (SACBIM), con la finalidad de satisfacer las necesidades tecnológicas del Municipio de Quito, de tal manera que utilizando tecnología de punta, se faciliten sus procesos y actividades con el propósito de cumplir eficaz y eficientemente las demandas de los diversos usuarios, en lo que respecta a la información del registro municipal.

El presente documento es una descripción técnica de todos los componentes desarrollados que forman parte del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control del Gobierno Electrónico del Municipio del Distrito Metropolitano de Quito.

Se inicia con la presentación detallada de la arquitectura implementada para el sistema, que incluye diagramas y descripciones de los componentes y su disposición tanto lógica como física.

Finalmente, se presenta cada componente de la solución con sus atributos y métodos.

INDICE

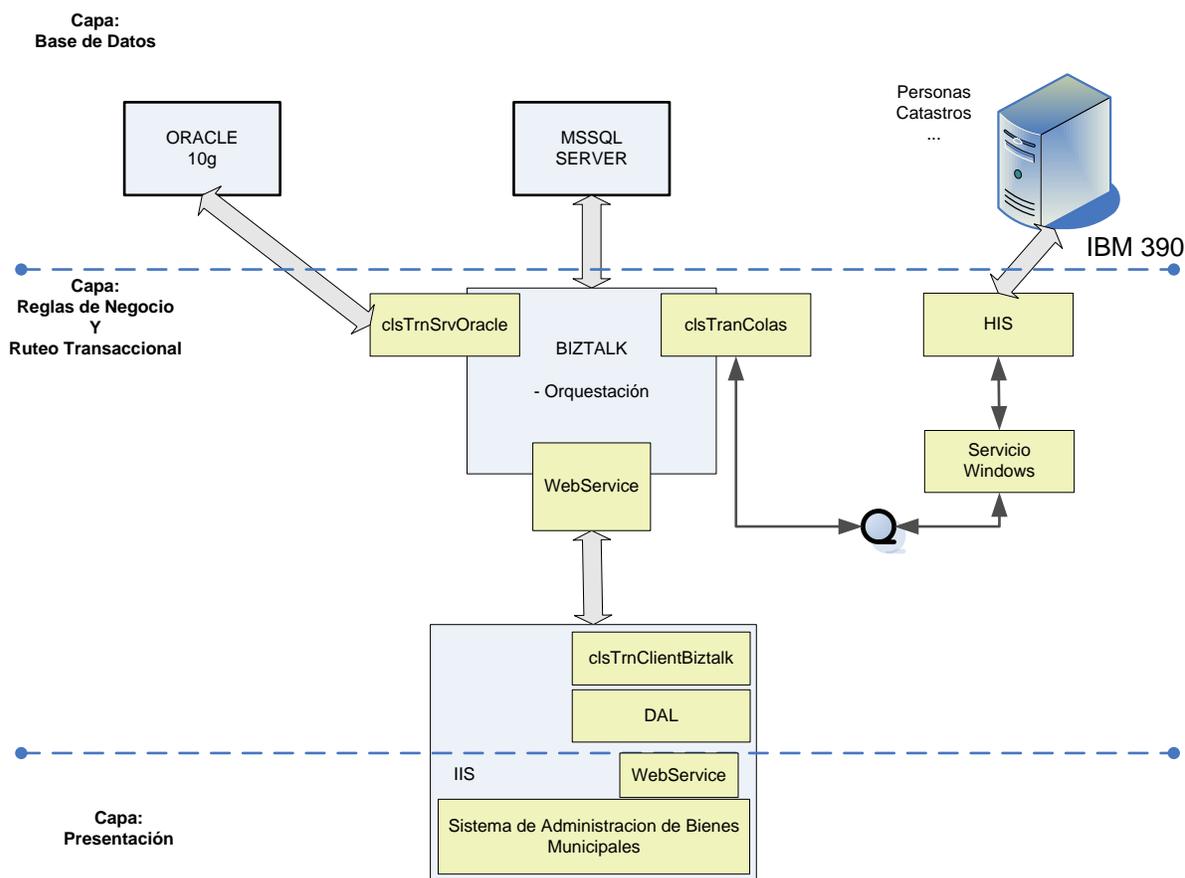
Resumen.....	2
Índice.....	3
1. Arquitectura del sistema.....	4
1.1. Diseño General del Sistema.....	4
1.2. Configuración de Hardware y Software.....	5
1.2.1. Configuración General de Hardware y Sistemas Operativos.....	5
1.2.2. Componentes del Sistema.....	7
2. MDMQ_DAL_Bienes.....	9
2.1. Diagrama de clases.....	10
2.2. Clases de este componente.....	10
2.2.1. Clase clsBLD_BIEN.....	11
2.2.2. Clase clsBLD_PERIODOS.....	13
2.2.3. Clase clsBLD_ESTADO.....	15
2.2.4. Clase clsBLD_USO.....	16
2.2.5. Clase clsBLD_OPERACION.....	18
2.2.6. Clase clsBLD_MOVIMIENTO.....	20
2.2.7. Clase clsBLD_TIPO.....	21
2.2.8. Clase clsBLD_TIPO_BIEN.....	23
2.2.9. Clase clsBLD_ESTRUCTURA.....	24
2.2.10. Clase clsBLD_TIP_MOVIM.....	26
2.2.11. Clase clsBLD_MOTIVOS.....	27
2.2.12. Clase clsBLD_ACCESORIOS.....	29
2.2.13. Clase clsBLD_GARANTIA.....	31
2.2.14. Clase clsBLD_HISTORICO.....	33
2.2.15. Clase clsBLD_COLOR.....	34
2.2.16. Clase clsBLD_CUENTA.....	36
2.2.17. Clase clsBLD_PERSONA.....	37
2.2.18. Clase clsBLD_UBICACION.....	39
2.2.19. Clase clsBLD_BIEN_UBIC.....	40

1. Arquitectura del sistema

En el presente capítulo se explicará cómo está formada la Arquitectura del Software del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control del Gobierno Electrónico del MDMQ.

1.1. Diseño General del Sistema

El diseño general de la solución requiere la implementación de los componentes descritos en el siguiente gráfico:



Los elementos que constituyen esta solución son los siguientes:

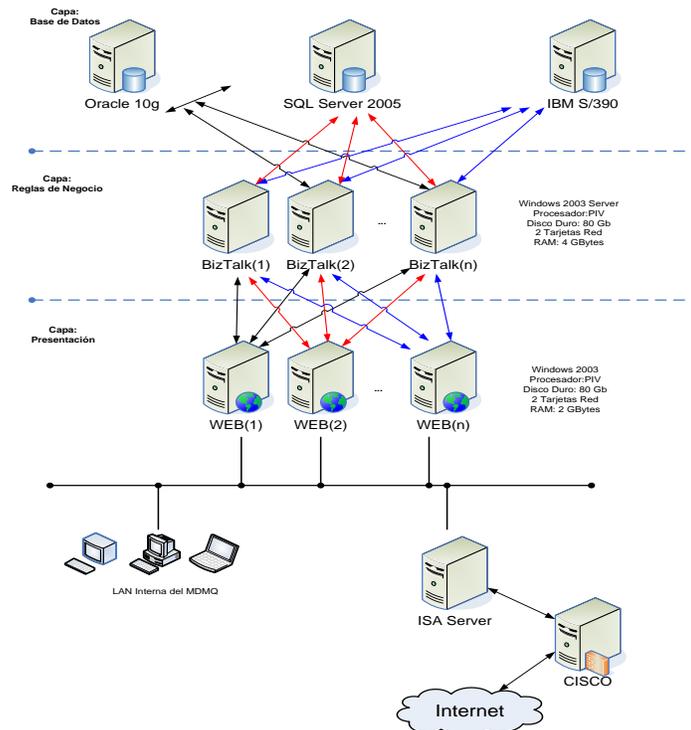
#	Nombre	Descripción
1	Oracle 10g	Base de datos donde se van a guardar los datos del sistema de Bienes Municipales
2	MSSQL Server	Base de datos para que funcione BizTalk Server y SQL Reporting Services.
3	IBM 390	Host IBM S/390 del Municipio de Quito.
4	BizTalk	Actúa como centro para conectar varios sistemas diferentes a través de un sistema de mensajes y orquestación basado en XML.
5	clsTrnSrvOracle	Componente que llama a BizTalk para ejecutar las transacciones en la

		base de datos Oracle.
6	clsTranColas	Componente de uso de colas. Utilizado desde la orquestación BizTalk.
7	Servicio Windows MDMQ_Seg_Host	Servicio Windows que se encarga de leer las colas para procesar las transacciones llamando al HIS.
8	HIS – COMTI	Componentes de comunicación para el intercambio de información del el servidor IBM 390. En este caso se utiliza el componente LibHisBK04.DLL
9	WebService	Servicio web donde se publican las orquestaciones BizTalk. Debe ser identificado de la siguiente forma: <i>http://aaa.bbb.ccc.ddd</i> <i>/MDMQ_BIZ_Switch_Proxy/MDMQ_BIZ_Switch_MDMQBIZSeguridades_p</i> <i>rEjecutarTran.asmx</i>
10	IIS	Internet Information Services. Servidor Web de Microsoft.
11	clsTrnClientBiztalk	Componente que permite a las aplicaciones cliente ejecutar sus transacciones a través del Web Service publicado por BizTalk
12	DAL	Data Access Layer – Capa de acceso a datos. Componentes que hacen llamadas a clsTrnClientBiztalk para ejecutar las transacciones en los servidores por medio de BizTalk Server
13	Sistema de Administración de Bienes Municipales	Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control.

1.2. Configuración de Hardware y Software

1.2.1. Configuración General de Hardware y Sistemas Operativos

La configuración general de Hardware y Software es la siguiente:



En este gráfico se puede observar la presencia de tres repositorios de bases de datos (en la capa de base de datos):

- ✚ Oracle 10g
- ✚ SQL Server 2005
- ✚ IBM S/390

Al momento de la realización de este documento, cada una de las bases de datos se encuentra en servidores separados. Esta configuración también podría realizarse en un servidor HP-UX con particiones virtuales independientes, donde se ubicaría Oracle 10g y CICS en particiones UNIX y SQL Server en una partición Windows 2003 Server con Service Pack II.

A continuación viene una granja de servidores BizTalk (Windows 2003 Server SP II), cada uno de estos computadores que forme parte de esta granja, constituyen la capa de reglas de negocio:

- ✚ Debe contar con 2 tarjetas de red, una para conectarse a las bases de datos y otro para conectarse a la granja WEB.
- ✚ Debe tener instalado el componente MDMQ_CLS_TransrvOracle para conectarse a la base de datos.
- ✚ Debe tener instalado el Web Service de interface de la orquestación de BizTalk (MDMQ_BIZ_Switch_MDMQBizSeguridades_prtEjecutarTran.aspx?op=fnEjecutar), y la orquestación BizTalk.

A continuación viene la granja WEB (Windows 2003), cada uno de los computadores que forme parte de esta granja corresponden a la capa de presentación:

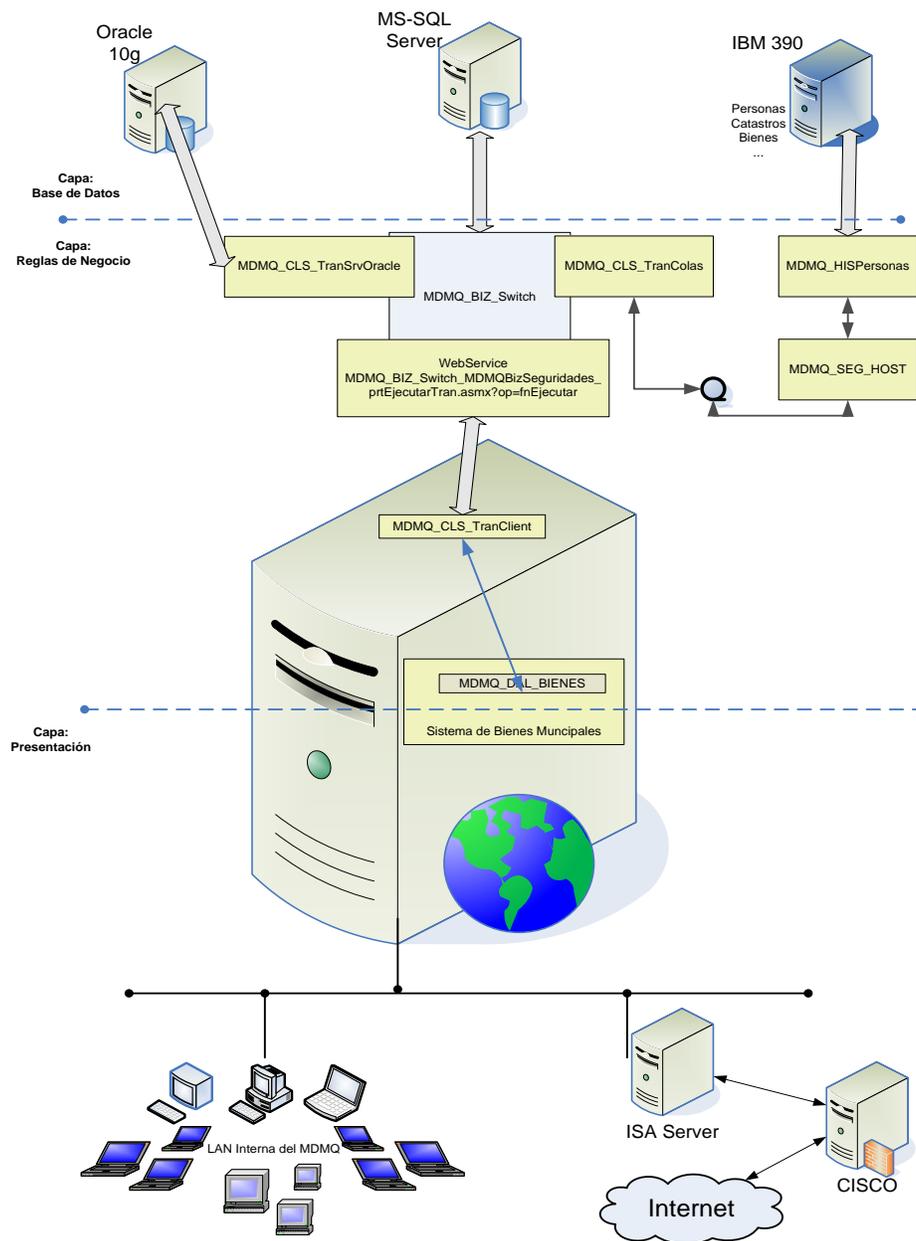
- ✚ Debe tener instalado el componente MDMQ_CLS_TransClient.
- ✚ Debe tener instalado el componente de acceso a datos del sistema de bienes: MDMQ_DAL_BIENES.
- ✚ Debe tener las páginas aspx del sitio web del Sistema de Bienes Municipales, dentro del Instalador de estas páginas, se incluyen también las DAL antes mencionadas.

Es importante considerar, que debido a la simplicidad de esta aplicación, las DAL se encuentran dentro de la aplicación web, sin embargo, esto no es una obligación, y en

aplicaciones más complejas se las puede incluir en la capa de negocios y en otra granja de servidores que contenga las DAL y la clsTranClient.

1.2.2. Componentes del Sistema

A continuación se presenta un gráfico que ilustra todos los componentes de software que forman parte del Sistema de Bienes Municipales del Gobierno Electrónico del MDMQ:



Como se puede observar en el anterior gráfico, el software sobre el cual funciona el Sistema SACBIM, tiene las siguientes características:

En la Capa de Datos:

- ✦ La base de datos Oracle 10g, es donde están las tablas de datos y los procedimientos almacenados del sistema. Hay que tener en cuenta que los CRUD de las tablas, fueron generados mediante plantillas en CodeSmith.
- ✦ La base de datos MS-SQL Server, que es la base de datos con utiliza BizTalk para su funcionamiento. Así como también es el repositorio de los Reportes del Sistema.
- ✦ IBM S/390, que es donde están las aplicaciones del MDMQ.

En la capa de Reglas de Negocio:

- ✦ Existe el componente MDMQ_BIZ_Switch que tiene la orquestación, necesaria para lograr la sincronización entre Oracle y el IBM S/390.
- ✦ Para la conexión con la base de datos Oracle 10g, se desarrolló el componente MDMQ_CLS_TransrvOracle.
- ✦ Para controlar la sincronización con el IBM S/390 se desarrolló el componente MDMQ_CLS_TransColas, el cual por medio de colas de Windows MSMQ, se comunica con el componente MDMQ_SEG_HOST que es un servicio Windows para controlar las colas, el cual llama al MDMQ_HISPersonas que es quien finalmente tiene los elementos COMTI necesarios para la comunicación con el IBM S/390, siendo este componente quien se encargue de mantener la sincronización con el IBM S/390.
- ✦ Existe también el componente MDMQ_BIZ_Switch_MDMQBizSeguridades_prtEjecutarTran.asmx?op=fnEjecutar que es un servicio web generado por la orquestación, para que otros sistemas se puedan comunicar con la orquestación que se encuentra en BizTalk que para este caso está relacionado con el Sistema SACBIM.
- ✦ Las componentes de negocio de este sistema tienen los siguientes nombres: MDMQ_DAL_BIENES, que tienen todas las llamadas a los procedimientos almacenados de la base de datos. Estos componentes fueron generados mediante plantillas CodeSmith. Estas DAL están formadas por archivos xsd (XML Schema Definition - es un lenguaje de esquema utilizado para describir la estructura y las

restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así una percepción del tipo de documento con un nivel alto de abstracción).

En la capa de Presentación:

- ✚ Se encuentra el Sitio web del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control.

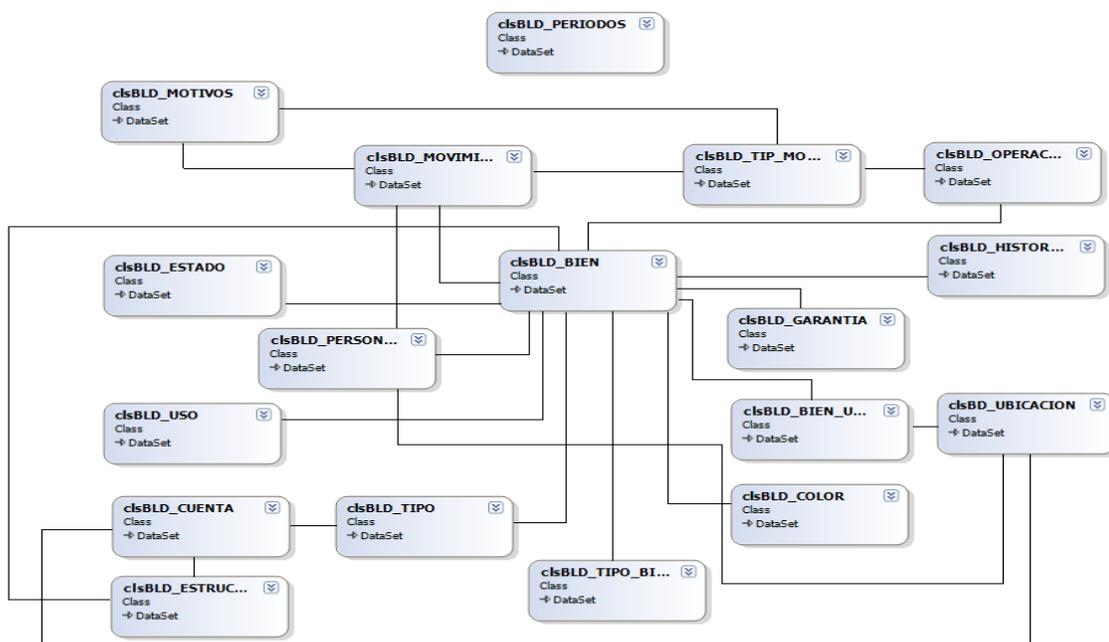
2. MDMQ_DAL_Bienes

MDMQ_DAL_Bienes es un componente, para el Gobierno Electrónico del Municipio del Distrito Metropolitano de Quito, como una parte del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control.

La finalidad de este componente es crear una capa entre los componentes de negocio y los componentes de la base de datos, facilitando de esta manera la llamada de los métodos desde cualquiera de las dos capas.

Este componente es generado por CodeSmith.

2.1. Diagrama de clases



2.2. Clases de este componente

Las clases que forman este componente son:

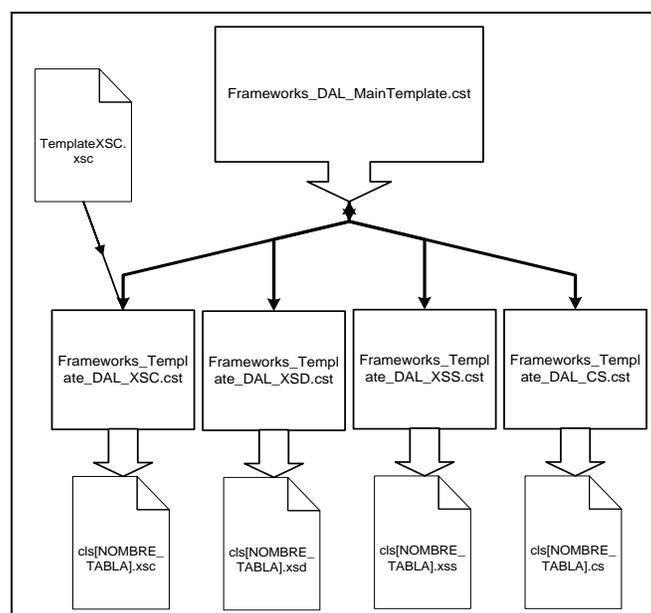
clsBLD_PERIODOS.xsd	clsBLD_BIEN.xsd
clsBLD_MOTIVOS.xsd	clsBLD_GARANTIA.xsd
clsBLD_MOVIMIENTOS.xsd	clsBLD_BIEN_UBICACION.xsd
clsBLD_ESTADO.xsd	clsBLD_COLOR.xsd
clsBLD_PERSONAS.xsd	clsBLD_TIPO_BIEN.xsd
clsBLD_USO.xsd	clsBLD_OPERACION.xsd
clsBLD_CUENTA.xsd	clsBLD_HISTORICO.xsd
clsBLD_TIPO.xsd	clsBLD_UBICACION.xsd
clsBLD_ESTRUCTURA.xsd	clsBLD_TIPO_MOVIMIENTO.xsd

Cada uno de estas clases está formado por 5 archivos. Como ejemplo tomaremos la primera clase, es decir a clsSEG_APLICACION.xsd y revisaremos su contenido.

Los archivos que forman esta clase son:

clsBLD_BIEN.cs	Tiene la definición de los métodos que van a ser utilizados por la clase.
clsBLD_BIEN.Designer.cs	Tiene la definición de la clase, del DataSet que maneja, de las tablas, de las relaciones entre las tablas y de los campos que maneja.
clsBLD_BIEN.xsc	
clsBLD_BIEN.xsd	Se presenta en forma gráfica la estructura XML de los datos que va a manejar la clase.
clsBLD_BIEN.xss	

El siguiente es un gráfico de la plantilla utilizada para la generación de los diferentes archivos:



2.2.1. Clase clsBLD_BIEN

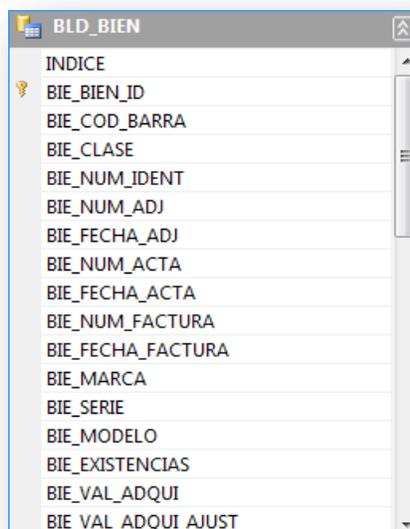
Una de las clases más importantes para el sistema es el clsBLD_BIEN.cs, en este archivo se encuentra la definición de los métodos que son utilizados por la clase. En esta clase encontramos un contenido como el siguiente:



Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_BIENConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_bien_qry_fun
<code>public int fnBLD_BIENBuscar</code>	Buscar en la tabla. Realiza búsquedas por todos los campos de la tabla correspondiente. Llama al procedimiento almacenado: bld_bien_fnd_fun
<code>public int fnBLD_BIENInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_bien_add_fun

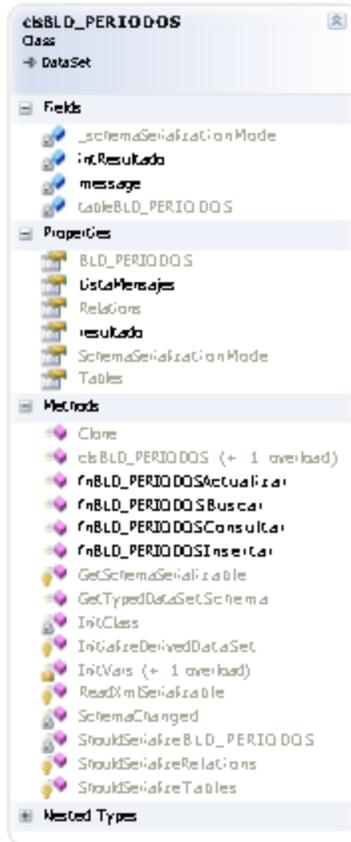
<code>public int fnBLD_BIENActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_bien_upd_fun</code>
<code>public int fnBLD_BIENEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_bien_del_fun</code>
<code>public int fnBLD_BIENActualizarDepreciacion</code>	Actualiza registro de erogación de la tabla. Llama al procedimiento almacenado: <code>bld_bien_depreciacion_upd_fun</code>
<code>public int fnBLD_BIENActualizarErogacion</code>	Actualiza registro de erogación de la tabla. Llama al procedimiento almacenado: <code>bld_bien_erogacion_upd_fun</code>
<code>public int fnBLD_BIENEliminarAccesorios</code>	Elimina los accesorios que están registrados. Llama al procedimiento almacenado: <code>bld_bien_accesorio_del_fun</code>
<code>public int fnBLD_BIENBaja</code>	Dar de Baja a un registro de la tabla. Llama al procedimiento almacenado: <code>bld_bien_baja_fun</code>
<code>public int fnBLD_BIENBuscarSecuencial</code>	Buscar Secuencial para el código de Barra. Llama al procedimiento almacenado: <code>bld_bien_barra_fun</code>
<code>public int fnBLD_BIENObtenerUltimoValor</code>	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: <code>bld_ultimo_valor_bienes_fun</code>

En el archivo `clsBLD_BIEN.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



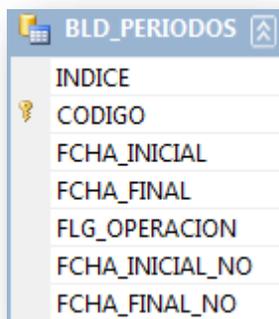
2.2.2. Clase `clsBLD_PERIODOS`

En esta clase se define los métodos para que el sistema pueda interactuar con las depreciaciones. En esta clase encontramos un contenido como el siguiente:



Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_PERIODOSConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_periodos_qry_fun
<code>public int fnBLD_PERIODOSBuscar</code>	Buscar en la tabla. Realiza búsquedas por todos los campos de la tabla correspondiente. Llama al procedimiento almacenado: bld_periodos_fnd_fun
<code>public int fnBLD_PERIODOSInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_periodos_add_fun
<code>public int fnBLD_PERIODOSActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_periodos_upd_fun
<code>public int fnBLD_PERIODOSEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: bld_periodos_del_fun

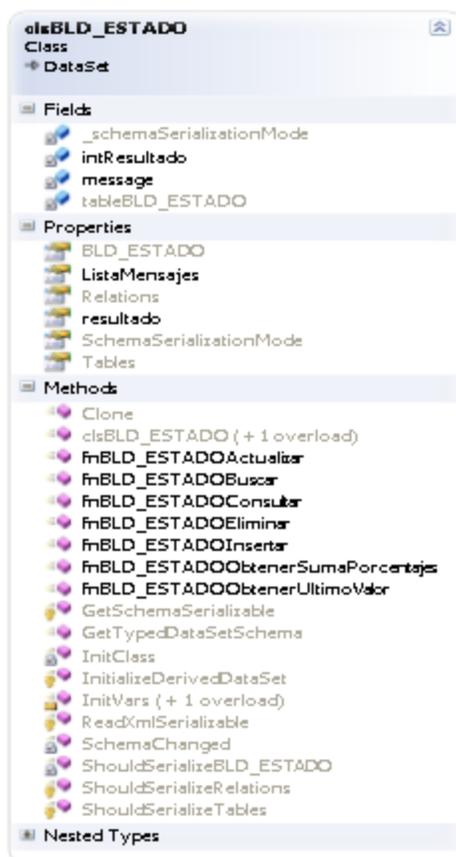
En el archivo clsBLD_PERIODOS.xsd se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



INDICE
KEY CODIGO
FCHA_INICIAL
FCHA_FINAL
FLG_OPERACION
FCHA_INICIAL_NO
FCHA_FINAL_NO

2.2.3. Clase clsBLD_ESTADO

En esta clase se define los métodos del parámetro de Estados, los cuales se utilizan para determinar en qué estado se encuentra un Bien Municipal.



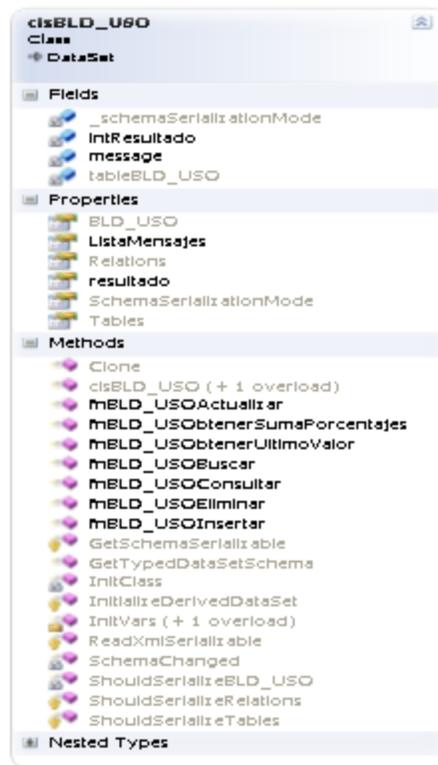
Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_ESTADOConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_estado_qry_fun</code>
<code>public int fnBLD_ESTADOBuscar</code>	Buscar en la tabla. Realiza búsquedas por todos los campos de la tabla correspondiente. Llama al procedimiento almacenado: <code>bld_estado_fnd_fun</code>
<code>public int fnBLD_ESTADOInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_estado_add_fun</code>
<code>public int fnBLD_ESTADOActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_estado_upd_fun</code>
<code>public int fnBLD_ESTADOEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_estado_del_fun</code>
<code>public int fnBLD_ESTADOObtenerUltimoValor</code>	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: <code>bld_ultimo_valor_estados_fun</code>
<code>public int fnBLD_ESTADOObtenerSumaPorcentajes</code>	Obtiene la suma de porcentajes de los Estados. Llama al procedimiento almacenado: <code>bld_suma_porcentaje_estado_fun</code>

En el archivo `clsBLD_ESTADO.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



2.2.4. Clase clsBLD_USO

En esta clase se define los métodos del parámetro de Uso, los cuales se utilizan para determinar en qué uso se encuentra un Bien Municipal.



Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_USOConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_uso_qry_fun
<code>public int fnBLD_USOBuscar</code>	Buscar en la tabla. Realiza búsquedas por todos los campos de la tabla correspondiente. Llama al procedimiento almacenado: bld_uso_fnd_fun
<code>public int fnBLD_USOInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_uso_add_fun
<code>public int fnBLD_USOActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_uso_upd_fun
<code>public int fnBLD_USOEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: bld_uso_del_fun
<code>public int fnBLD_USOObtenerUltimoValor</code>	Obtiene el último valor del código de Bien.

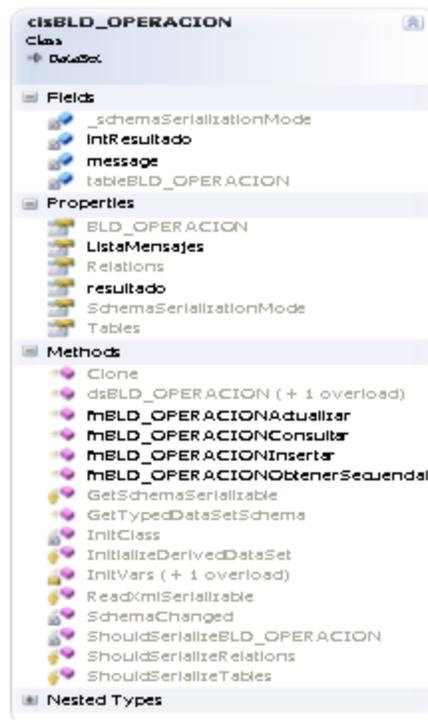
	Llama al procedimiento almacenado: bld_ultimo_valor_uso_fun
<code>public int fnBLD_USOObtenerSumaPorcentajes</code>	Obtiene la suma de porcentajes de los Estados. Llama al procedimiento almacenado: bld_suma_porcentaje_uso_fun

En el archivo clsBLD_USO.xsd se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



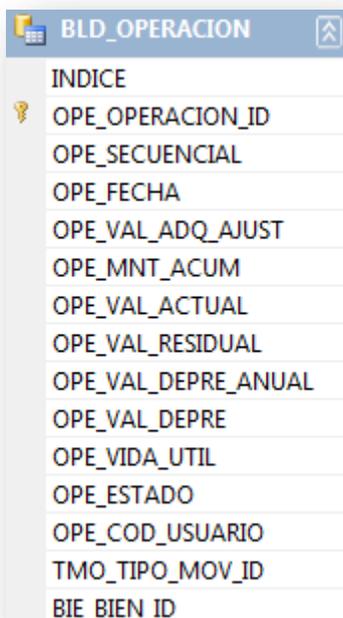
2.2.5. Clase clsBLD_OPERACION

En esta clase se define los métodos de operación del sistema, los cuales se utilizan para determinar los valores que fueron utilizados en varios procesos.



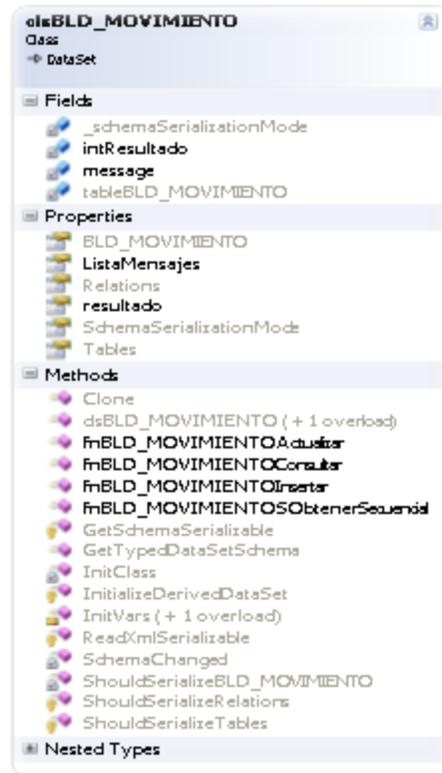
Datos	
<code>private int</code> intResultado	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[]</code> message	Mensaje de la ejecución.
<code>public int</code> resultado	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[]</code> ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int</code> fnBLD_OPERACIONConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_operacion_qry_fun</code>
<code>public int</code> fnBLD_OPERACIONInsertar	Insertar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_operacion_add_fun</code>
<code>public int</code> fnBLD_OPERACIONActualizar	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_operacion_upd_fun</code>
<code>public int</code> fnBLD_OPERACIONEliminar	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_operacion_del_fun</code>
<code>public int</code> fnBLD_OPERACIONObtenerSecuencial	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: <code>bld_ultimo_valor_operacion_fun</code>

En el archivo `clsBLD_OPERACION.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



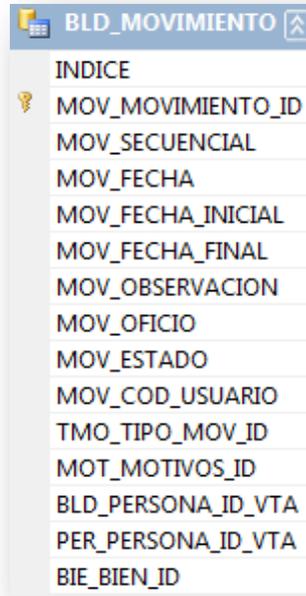
2.2.6. Clase clsBLD_MOVIMIENTO

En esta clase se define los métodos de movimiento del sistema, los cuales se utilizan para determinar los movimientos físicos que se realizan en varios procesos del sistema.



Datos	
private int intResultado	Resultado de la ejecución.
clsTrnClientBiztalk.clsMessage[] message	Mensaje de la ejecución.
public int resultado	Propiedad. Resultado de la ejecución.
public clsTrnClientBiztalk.clsMessage[] ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
public int fnBLD_MOVIMIENTOConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_movimiento_qry_fun
public int fnBLD_MOVIMIENTOInsertar	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_movimiento_add_fun
public int fnBLD_MOVIMIENTOActualizar	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_movimiento_upd_fun
public int fnBLD_MOVIMIENTOEliminar	Eliminar un registro de la tabla. Llama al procedimiento almacenado: bld_movimiento_del_fun
public int fnBLD_MOVIMIENTOSObtenerSecuencial	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: bld_ultimo_valor_movimiento_fun

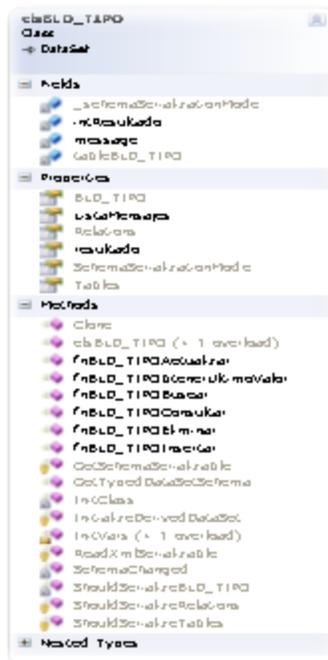
En el archivo clsBLD_MOVIMIENTO.xsd se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



BLD_MOVIMIENTO	
	INDICE
KEY	MOV_MOVIMIENTO_ID
	MOV_SECUENCIAL
	MOV_FECHA
	MOV_FECHA_INICIAL
	MOV_FECHA_FINAL
	MOV_OBSERVACION
	MOV_OFICIO
	MOV_ESTADO
	MOV_COD_USUARIO
	TMO_TIPO_MOV_ID
	MOT_MOTIVOS_ID
	BLD_PERSONA_ID_VTA
	PER_PERSONA_ID_VTA
	BIE_BIEN_ID

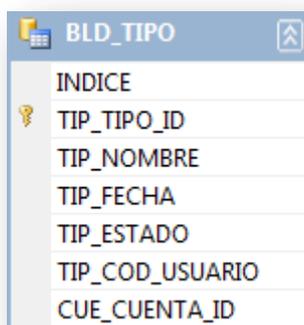
2.2.7. Clase clsBLD_TIPO

En esta clase se define los métodos de tipo de grupo del sistema, los cuales se utilizan para determinar a qué tipo se refiere el usuario al escoger una cuenta.



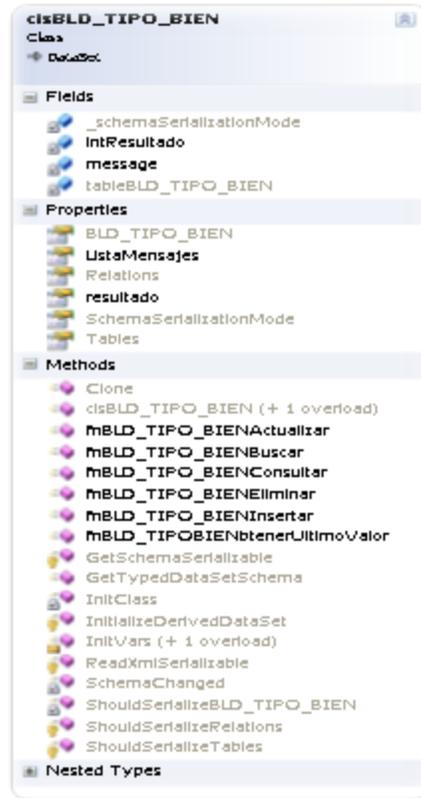
Datos	
<code>private int</code> intResultado	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[]</code> message	Mensaje de la ejecución.
<code>public int</code> resultado	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[]</code> ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int</code> fnBLD_TIPOConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_tipo_qry_fun</code>
<code>public int</code> fnBLD_TIPOInsertar	Insertar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_tipo_add_fun</code>
<code>public int</code> fnBLD_TIPOActualizar	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_tipo_upd_fun</code>
<code>public int</code> fnBLD_TIPOEliminar	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_tipo_del_fun</code>
<code>public int</code> fnBLD_TIPOObtenerUltimoValor	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: <code>bld_ultimo_valor_tipo_fun</code>
<code>public int</code> fnBLD_TIPOBuscar	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_tipo_fnd_fun</code>

En el archivo `clsBLD_TIPO.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



2.2.8. Clase `clsBLD_TIPO_BIEN`

En esta clase se define los métodos de tipo de bien del sistema, los cuales se utilizan para determinar a qué tipo pertenece dicho Bien.



Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_TIPO_BIENConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_tipo_bien_qry_fun
<code>public int fnBLD_TIPO_BIENInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_tipo_bien_add_fun
<code>public int fnBLD_TIPOA_BIENActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_tipo_bien_upd_fun
<code>public int fnBLD_TIPO_BIENEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: bld_tipo_del_bien_fun
<code>public int fnBLD_TIPOBIENObtenerUltimoValor</code>	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: bld_ultimo_valor_tipo_bien_fun
<code>public int fnBLD_TIPO_BIENBuscar</code>	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_tipo_bien_fnd_fun

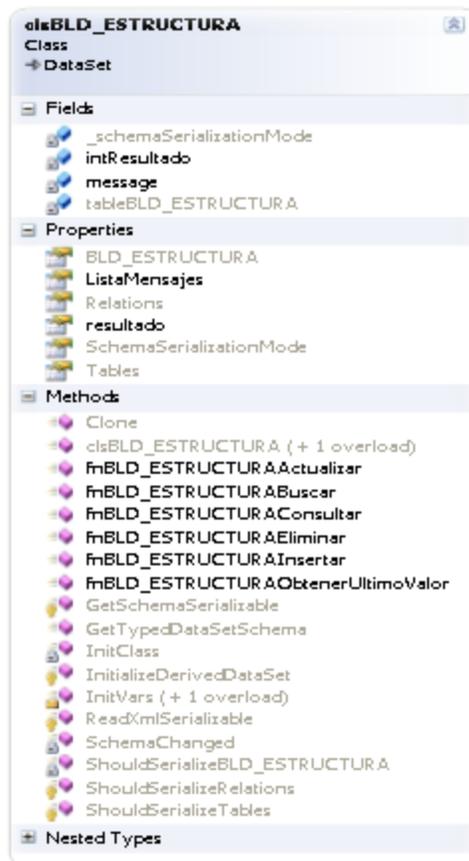
En el archivo clsBLD_TIPO_BIEN.xsd se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



BLD_TIPO_BIEN	
INDICE	
TPB_TIPO_BIEN_ID	Primary Key
TPB_NOMBRE	
TPB_ESTADO	
TPB_FECHA	
TPB_COD_USUARIO	
FLG_MAYOR	

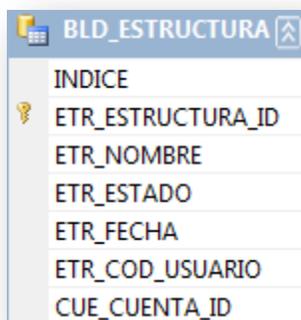
2.2.9. Clase clsBLD_ESTRUCTURA

En esta clase se define los métodos de estructura de clase del sistema, los cuales se utilizan para determinar a qué tipo se refiere el usuario al escoger una cuenta.



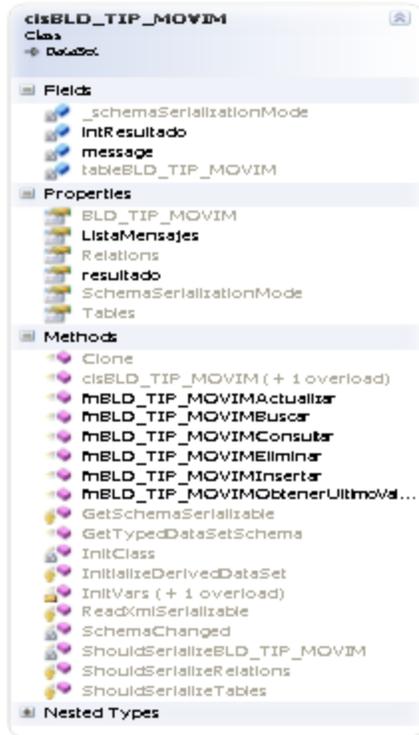
Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_ESTRUCTURAConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_estructura_qry_fun</code>
<code>public int fnBLD_ESTRUCTURAInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_estructura_add_fun</code>
<code>public int fnBLD_ESTRUCTURAActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_estructura_upd_fun</code>
<code>public int fnBLD_ESTRUCTURAEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_estructura_del_fun</code>
<code>public int fnBLD_ESTRUCTURAObtenerUltimoValor</code>	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: <code>bld_ultimo_valor_estructura_fun</code>
<code>public int fnBLD_ESTRUCTURABuscar</code>	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_estructura_fnd_fun</code>

En el archivo `clsBLD_ESTRUCTURA.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



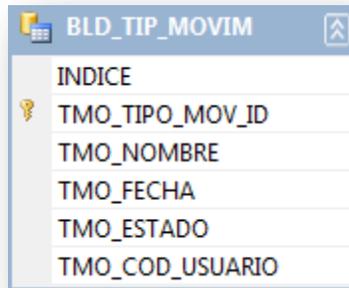
2.2.10. Clase `clsBLD_TIP_MOVIM`

En esta clase se define los métodos de tipo de movimiento del sistema, los cuales se utilizan para determinar a qué cuantas operaciones puede realizar el usuario en el sistema.



Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_TIP_MOVIMConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_tip_movim_qry_fun</code>
<code>public int fnBLD_TIP_MOVIMInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_tip_movim_add_fun</code>
<code>public int fnBLD_TIP_MOVIMActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_tip_movim_upd_fun</code>
<code>public int fnBLD_TIP_MOVIMEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_tip_movim_del_fun</code>
<code>public int fnBLD_TIP_MOVIMObtenerUltimoValor</code>	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: <code>bld_ultimo_valor_tip_movim_fun</code>
<code>public int fnBLD_TIP_MOVIMBuscar</code>	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_tip_movim_fnd_fun</code>

En el archivo clsBLD_TIP_MOVIM.xsd se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



BLD_TIP_MOVIM	
	INDICE
🔑	TMO_TIPO_MOV_ID
	TMO_NOMBRE
	TMO_FECHA
	TMO_ESTADO
	TMO_COD_USUARIO

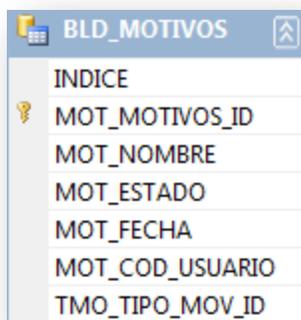
2.2.11. Clase clsBLD_MOTIVOS

En esta clase se define los métodos de motivos del sistema, los cuales se utilizan para determinar los posibles motivos que se determinaran por los tipos de movimientos.

31

Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_MOTIVOSConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_motivos_qry_fun</code>
<code>public int fnBLD_MOTIVOSInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_motivos_add_fun</code>
<code>public int fnBLD_MOTIVOSActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_motivos_upd_fun</code>
<code>public int fnBLD_MOTIVOSEliminar</code>	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_motivos_del_fun</code>
<code>public int fnBLD_MOTIVOSObtenerUltimoValor</code>	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: <code>bld_ultimo_valor_motivos_fun</code>
<code>public int fnBLD_MOTIVOSBuscar</code>	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_motivos_fnd_fun</code>

En el archivo `clsBLD_MOTIVOS.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



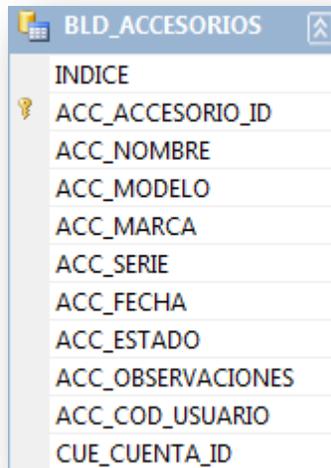
2.2.12. Clase clsBLD_ACCESORIOS

En esta clase se define los métodos de accesorios del sistema, los cuales se utilizan para determinar los accesorios de los bienes.

Datos	
<code>private int</code> intResultado	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[]</code> message	Mensaje de la ejecución.
<code>public int</code> resultado	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[]</code> ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int</code> fnBLD_ACCESORIOSConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_accesorios_qry_fun
<code>public int</code> fnBLD_ACCESORIOSInsertar	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_accesorios_add_fun
<code>public int</code> fnBLD_ACCESORIOSActualizar	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_accesorios_upd_fun
<code>public int</code> fnBLD_ACCESORIOSEliminar	Eliminar un registro de la tabla. Llama al procedimiento almacenado: bld_accesorios_del_fun
<code>public int</code> fnBLD_ACCESORIOSObtenerUltimoValor	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado:

	bld_ultimo_valor_accesorios_fun
<code>public int fnBLD_ACCESORIOSBuscar</code>	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_accesorios_fnd_fun

En el archivo `clsBLD_ACCESORIOS.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



2.2.13. Clase `clsBLD_GARANTIA`

En esta clase se define los métodos de garantía del sistema, los cuales se utilizan para determinar la garantía del bien adquirido.

Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_GARANTIAConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_garantia_qry_fun</code>
<code>public int fnBLD_GARANTIAInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_garantia_add_fun</code>
<code>public int fnBLD_GARANTIAActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: <code>bld_garantia_upd_fun</code>

En el archivo `clsBLD_GARANTIA.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



2.2.14. Clase `clsBLD_HISTORICO`

En esta clase se define los métodos de históricos del sistema, los cuales se utilizan para determinar las consultas históricas de los bienes.

Datos	
<code>private int</code> intResultado	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[]</code> message	Mensaje de la ejecución.
<code>public int</code> resultado	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[]</code> ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int</code> fnBLD_HISTORICOConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_historico_qry_fun</code>
<code>public int</code> fnBLD_HISTORICOEliminar	Eliminar un registro de la tabla. Llama al procedimiento almacenado: <code>bld_historico_del_fun</code>

En el archivo `clsBLD_HISTORICO.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:

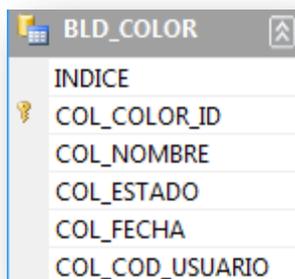
INDICE
HIS_HISTORICO_ID
HIS_COD_BARRA
HIS_CLASE
HIS_NUM_IDENT
HIS_NUM_ADJ
HIS_FCH_ADJ
HIS_NUM_ACTA
HIS_FCH_ACTA
HIS_NUM_FACTURA
HIS_FCH_FACTURA
HIS_MARCA
HIS_SERIE
HIS_MODELO

2.2.15. Clase clsBLD_COLOR

En esta clase se define los métodos de color del sistema, los cuales se utilizan para determinar el color de cada bien.

Datos	
<code>private int</code> intResultado	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[]</code> message	Mensaje de la ejecución.
<code>public int</code> resultado	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[]</code> ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int</code> fnBLD_COLORConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_color_qry_fun
<code>public int</code> fnBLD_COLORInsertar	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_color_add_fun
<code>public int</code> fnBLD_COLORActualizar	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_color_upd_fun
<code>public int</code> fnBLD_COLOREliminar	Eliminar un registro de la tabla. Llama al procedimiento almacenado: bld_color_del_fun
<code>public int</code> fnBLD_COLORObtenerUltimoValor	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: bld_ultimo_valor_color_fun
<code>public int</code> fnBLD_COLORBuscar	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_color_fnd_fun

En el archivo `clsBLD_COLOR.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



2.2.16. Clase `clsBLD_CUENTA`

En esta clase se define los métodos de color del sistema, los cuales se utilizan para determinar el color de cada bien.

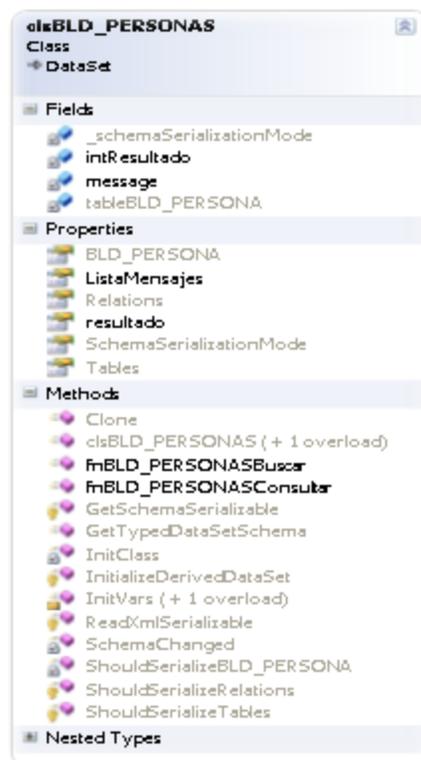
Datos	
<code>private int</code> intResultado	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[]</code> message	Mensaje de la ejecución.
<code>public int</code> resultado	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[]</code> ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int</code> fnBLD_CUENTAConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_cuenta_gry_fun
<code>public int</code> fnBLD_CUENTAInsertar	Insertar un registro en la tabla. Llama al procedimiento almacenado: bld_cuenta_add_fun
<code>public int</code> fnBLD_CUENTAActualizar	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_cuenta_upd_fun
<code>public int</code> fnBLD_CUENTAEliminar	Eliminar un registro de la tabla. Llama al procedimiento almacenado: bld_cuenta_del_fun
<code>public int</code> fnBLD_CUENTABuscar	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_cuenta_fnd_fun

En el archivo clsBLD_CUENTA.xsd se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



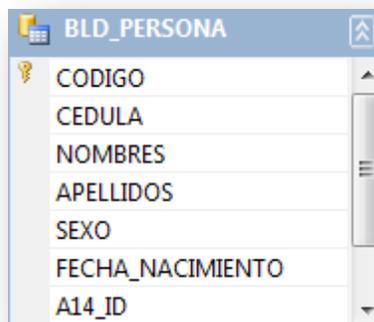
2.2.17. Clase clsBLD_PERSONA

En esta clase se define los métodos de persona del sistema, esta clase se conecta con una vista generada para el sistema SACBIM la cual se utilizan para determinar los custodios de los bienes.



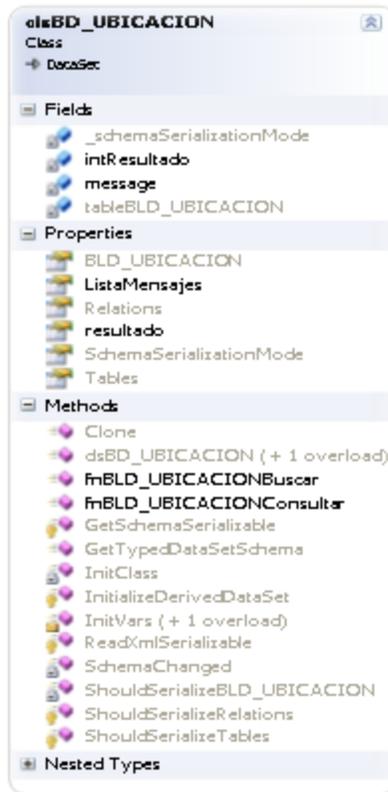
Datos	
<code>private int</code> intResultado	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[]</code> message	Mensaje de la ejecución.
<code>public int</code> resultado	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[]</code> ListaMensajes	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int</code> fnBLD_PERSONASConsultar	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_personas_qry_fun</code>
<code>public int</code> fnBLD_PERSONASBuscar	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: <code>bld_personas_fnd_fun</code>

En el archivo `clsBLD_PERSONAS.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:



2.2.18. Clase `clsBLD_UBICACION`

En esta clase se define los métodos de ubicación del sistema, esta clase se conecta con una vista generada para el sistema SACBIM la cual se utilizan para determinar las ubicaciones de los bienes.



Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_UBICACIONConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_ubicacion_qry_fun
<code>public int fnBLD_UBICACIONBuscar</code>	Buscar de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_ubicacion_fnd_fun

En el archivo clsBLD_UBICACION.xsd se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:

BLD_UBICACION	
🔑	CODIGO
	A18_CODIGO
	NOMBRE
	NIVEL_JERARQUICO
	DESCRIPCION
	INDICE

2.2.19. Clase clsBLD_BIEN_UBIC

En esta clase se define los métodos de color del sistema, los cuales se utilizan para determinar el color de cada bien.

Datos	
<code>private int intResultado</code>	Resultado de la ejecución.
<code>clsTrnClientBiztalk.clsMessage[] message</code>	Mensaje de la ejecución.
<code>public int resultado</code>	Propiedad. Resultado de la ejecución.
<code>public clsTrnClientBiztalk.clsMessage[] ListaMensajes</code>	Propiedad. Listado de mensajes de la ejecución
Métodos	
<code>public int fnBLD_BIEN_UBICConsultar</code>	Consulta de la tabla. Principalmente por el código o por campos principales de búsqueda. Llama al procedimiento almacenado: bld_bien_ubic_qry_fun
<code>public int fnBLD_BIEN_UBICInsertar</code>	Insertar un registro en la tabla. Llama al procedimiento almacenado:

	bld_bien_ubic_add_fun
<code>public int fnBLD_BIEN_UBICActualizar</code>	Actualizar un registro en la tabla. Llama al procedimiento almacenado: bld_bien_ubic_upd_fun
<code>public int fnBLD_BIEN_UBICObtenerUltimoValor</code>	Obtiene el último valor del código de Bien. Llama al procedimiento almacenado: bld_ultimo_valor_bien_ubic_fun

En el archivo `clsBLD_BIEN_UBIC.xsd` se puede ver la tabla (o tablas y sus relaciones) que van a manejarse en la clase, en forma gráfica:

BLD_BIEN_UBIC	
INDICE	
 BUB_BIEN_UBICACION_ID	
BUB_SECUENCIA	
BUB_FECHA_INI	
BUB_FECHA_FINAL	
BUB_TIPO	
BUB_ESTADO	
BUB_FECHA	
BUB_COD_USUARIO	
UBI_UBICACION_ID_VTA	

Manual de Usuario

SISTEMA DE BIENES MDMQ

Preparado para:

Municipio del Distrito Metropolitano de Quito

06 de junio de 2009

Versión 1.0 Release

Preparado por:

Sr. Rolando Orozco

Resumen

El presente documento presenta la documentación específica relativa del funcionamiento del Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control, es decir, aquí se detalla cada una de las funcionalidades con las que cuenta el sistema y el proceso que se sigue para su correcta utilización.

Contiene el detalle del funcionamiento de la Administración y Control de los Bienes Municipales tales como: Administración de Parámetros, Operaciones de Bienes, Depreciación, Erogación entre otros.

Finalmente se presentan los reportes implementados, los datos que reciben para filtrar la información y el resultado de los mismos.

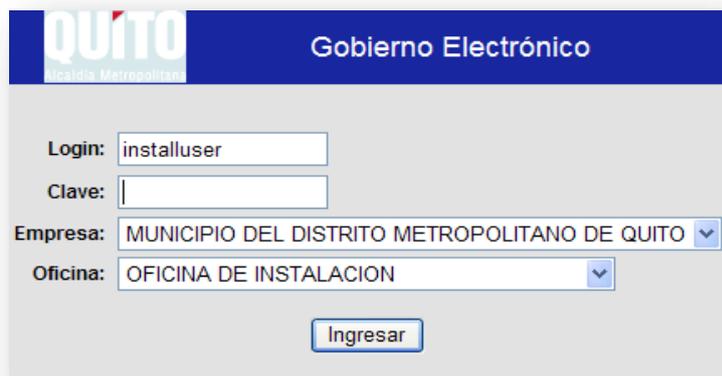
Índice de Contenido

1. Sistema de Bienes Municipales.....	1
1.1 Ingreso al Sistema.....	4
1.2 Menú Principal.....	5
1.3 Control de Bienes.....	6
1.3.1 Operaciones.....	6
1.3.2 Transferencia de Custodios.....	19
1.3.3 Movimientos Temporales.....	22
1.3.4 Depreciación.....	24
1.3.5 Erogaciones.....	25
1.3.6 Históricos.....	27
1.4 Administración.....	29
1.4.1 Cuentas.....	29
1.4.2 Parámetros.....	32
1.5 Reportes.....	48
1.5.1 Detalle del Bien.....	48
1.5.2 Grupo de Bienes.....	49
1.5.3 Depreciación de Bienes.....	50

6. Sistema de Bienes Municipales

1.1. Ingreso al Sistema

Para ingresar al Sistema de Administración y Control de Bienes de Larga Duración y Bienes Sujetos a Control se debe autenticar el usuario (nombre, clave), a su vez se debe escoger a que dependencia se quiere conectar (Empresa, Oficina).

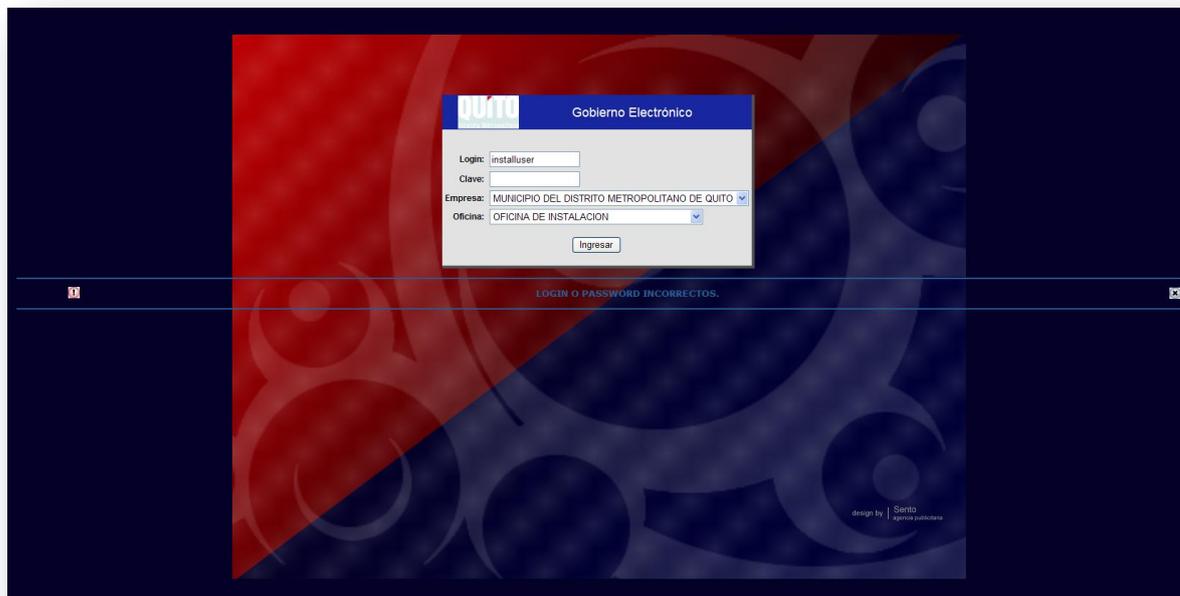


The screenshot shows the login interface for the 'Gobierno Electrónico' of Quito. The header includes the 'QUITO' logo and 'Alcaldía Metropolitana'. The form contains the following fields:

- Login:** A text input field containing the text 'installuser'.
- Clave:** A text input field for the password.
- Empresa:** A dropdown menu with the selected option 'MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO'.
- Oficina:** A dropdown menu with the selected option 'OFICINA DE INSTALACION'.

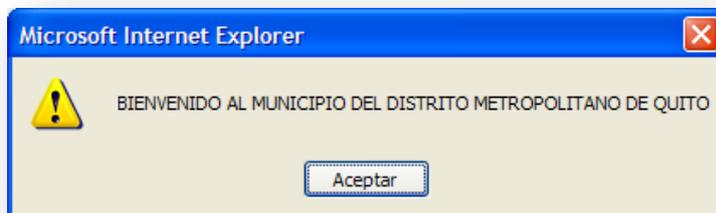
Below the form is a button labeled 'Ingresar'.

En el caso que el usuario introduzca erróneamente o pertenezca a esa independencia el nombre de usuario o la clave, se emitirá un mensaje de error y no tendrá acceso al Sistema de Bienes.



1.2. Menú Principal

Al ingresar al Sistema se desplegará un mensaje de Bienvenida al usuario identificado.



La página principal consta de un Menú de los proyectos actuales en el MDMQ en la parte izquierda y una cabecera la cual muestra el usuario ingresado, la fecha que ingreso y la hora actual del sistema.



El menú consta de cuatro secciones las cuales son: Control de Bienes, Administración y Reportes. Cada sección consta de sus subíndices.



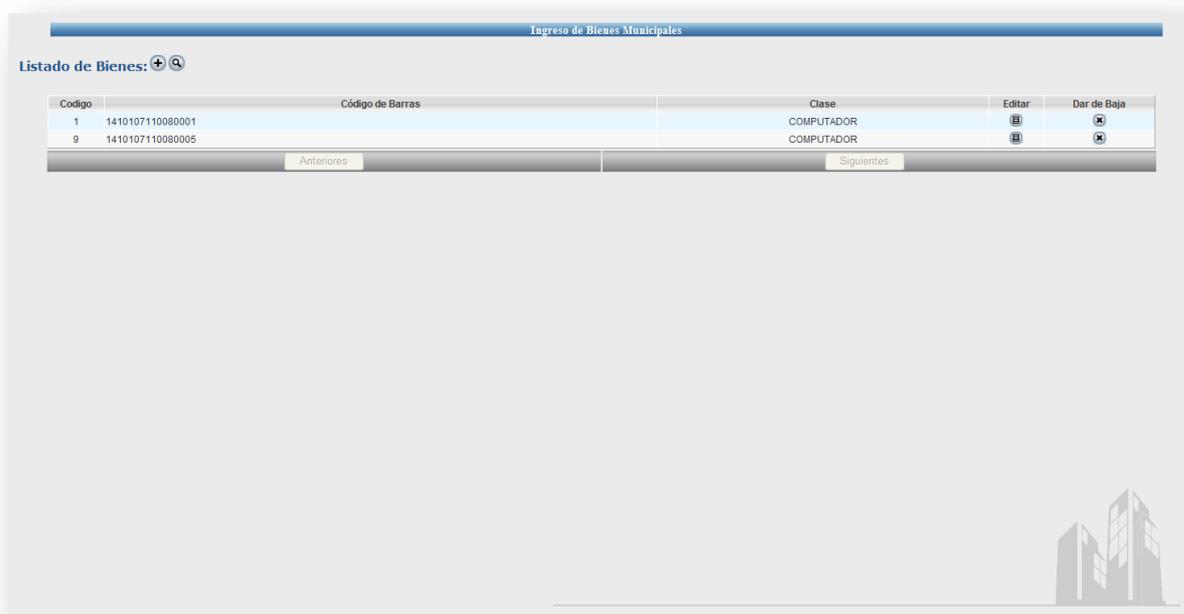
1.3. Control de Bienes

Dentro de la sección de Control de Bienes tenemos Operaciones, Transferencia de Custodio, Depreciación Bienes, Erogación Bienes, Movimientos Temporales e Históricos.



1.3.1. Operaciones

Dentro de la interfaz de Operaciones el usuario tiene la opción de realizar cualquier operación que vea conveniente tales como: Ingreso, Modificación, Búsqueda de Bien y Eliminación.



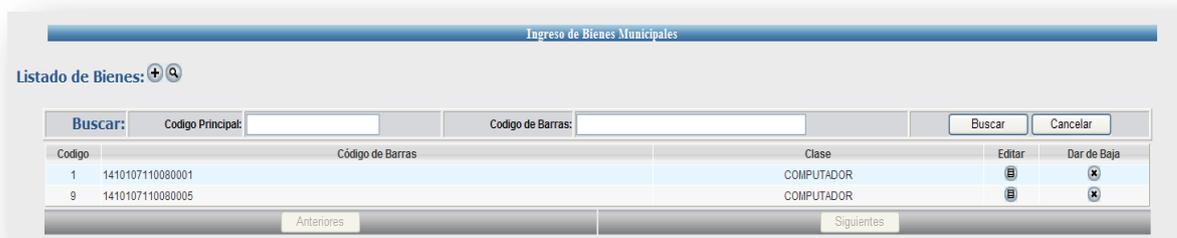
A la derecha del título "Listado de Bienes" se observan las herramientas   , la primera permite ingresar un nuevo Bien y la segunda permite realizar búsquedas.

En cada registro de la tabla se tiene las herramientas  para editar los contenidos y  para dar de Baja al Bien.

El catalogo de Bienes consta de los siguientes atributos:

Nombre	Descripción
Código	Numérico de 4 posiciones
Código de Barra	Alfanumérico de 70 posiciones
Clase	Puede ser Activo o Inactivo

Cuando se realiza una consulta de tipo de Bienes en la parte inferior de los botones se despliegan unos criterios por los cuales selecciona la búsqueda que aparece abajo, el módulo permite una navegación a través de esta lista mediante los botones que se encuentran en la parte inferior los mismos que permiten avanzar o retroceder en la lista.



Se puede realizar búsquedas de empresas por Código y por el Código de Barras. Con el botón Buscar se realiza la operación. Con el botón Cancelar se cancela la búsqueda.

Si el usuario realiza una búsqueda y no existe dicho bien el sistema desplegara un mensaje de error.

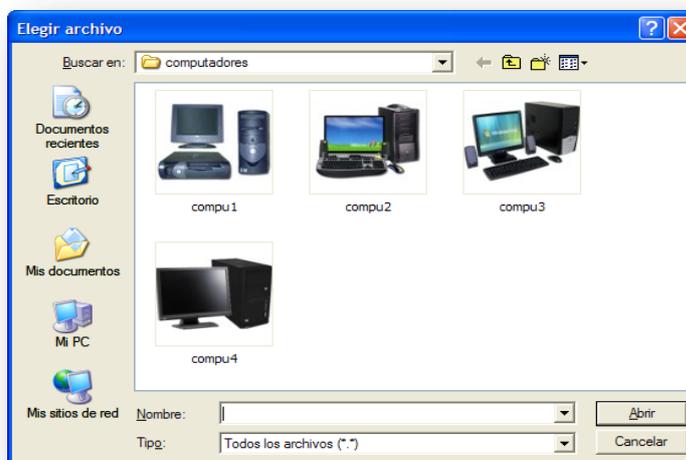


Si el usuario opta por la opción de insertar un nuevo Bien , se mostrara la pantalla de ingreso de datos, la cual deberá ser llenada con los datos que corresponden.

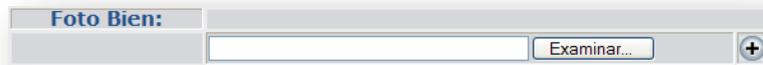
El panel de Cuenta consta de los siguientes datos a ser llenados

Nombre	Descripción
Clase	Control de catalogo de Cuentas.
Código Generado Bien	Código automáticamente generado para el ingreso del Bien.
Grupo	El grupo al cual pertenece la cuenta.
Grupo General	El grupo general al cual pertenece la cuenta.
Estructura	La estructura a la cual pertenece el Grupo.
Tipo de Clase	El tipo de el cual pertenece la clase

Para insertar la foto del Bien se deberá escoger la opción de Examinar para obtener la ruta que se encuentra la fotografía, las cuales estarán almacenadas en una carpeta destinada para ello.



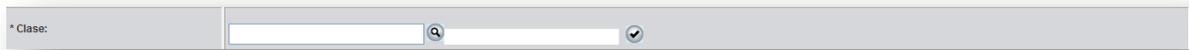
Para mostrar la foto en pantalla el usuario debe aplastar el botón que se encuentra en la parte izquierda del control Examinar.



Lo cual mostrara la Foto del Bien



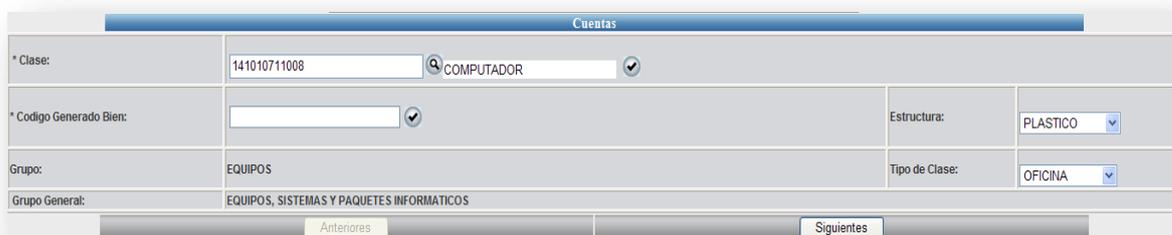
Para el ingreso de una clase se ha utilizado un control el cual llama a un catalogo para escoger cualquier opción que el usuario desee.



El usuario podrá realizar búsquedas mediante Código de Cuenta y por su Cuenta, lo cual permitirá escoger una cuenta específica. Si el usuario no sabe el nombre completo de la cuenta se podrá escribir alguna letra que lo identifique con el símbolo “%” lo cual permitirá encontrar todas las cuentas que contengan esas letras.



Una vez escogido la cuenta que el usuario requiera, se deberá aplastar el botón  que se encuentra en la parte derecha del control de catalogo de clase. El sistema buscara automáticamente llenara las estructuras y los tipos que contengas esta Cuenta, además, se obtendrá el Grupo y el Grupo General de la misma cuenta.

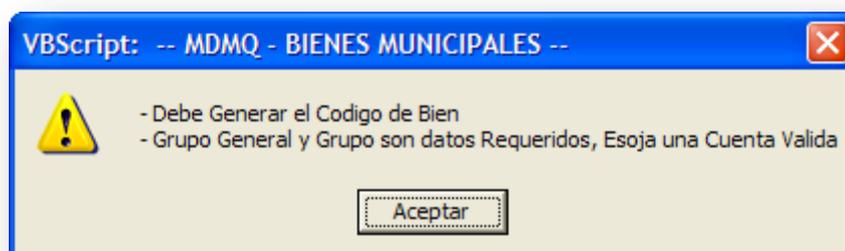


Se generara automáticamente el código del Bien aplastando el botón que se encuentra alado de la caja de texto de Código Generado Bien.



Para poder seguir llenando los datos requeridos se deberá aplastar el botón siguiente o anterior según el usuario se requiera.

En el caso que no se llenen los datos requeridos de las Cuentas al momento de querer cambiar de panel con el botón Siguiete, el sistema emitirá un mensaje mostrando los datos que faltan por generar o escribir.



El segundo panel a ser llenado es Datos de Adquisición, el cual consta de los siguientes datos.

Nombre	Descripción
Proveedor	Catalogo de Proveedores.
Garantía	Garantía de un Bien .
Valor Adquisición	Valor del Bien.
Numero Elementos	Número de elementos de un Bien.
Valor Adquisición Ajustada	Valor del Bien.
Numero Adjudicación	Numero requerido de adjudicación del Bien.
Numero Factura	Numero de Factura del Bien adquirido.
Fecha Adjudicación	Fecha de adjudicación del Bien
Fecha Factura	Fecha de factura del Bien adquirido.

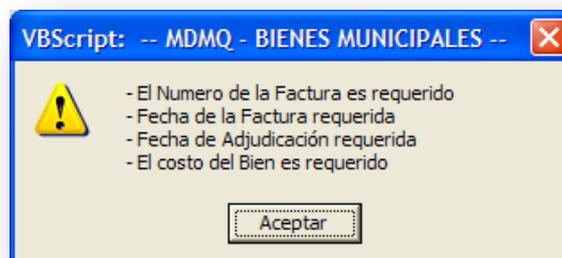
Al igual que en el catalogo de cuentas el catalogo del proveedor tiene las mismas funciones, el cual se eligiera el proveedor el Bien Municipal.



El usuario deberá llenar los datos requeridos del panel de Adquisición, en el caso de las fechas se mostrara un control para que el usuario pueda escoger la fecha que se desee.



Al momento de aplastar el botón siguiente existen varios datos requeridos por el sistema.



En el siguiente panel se encuentra las características del Bien con los siguientes atributos.

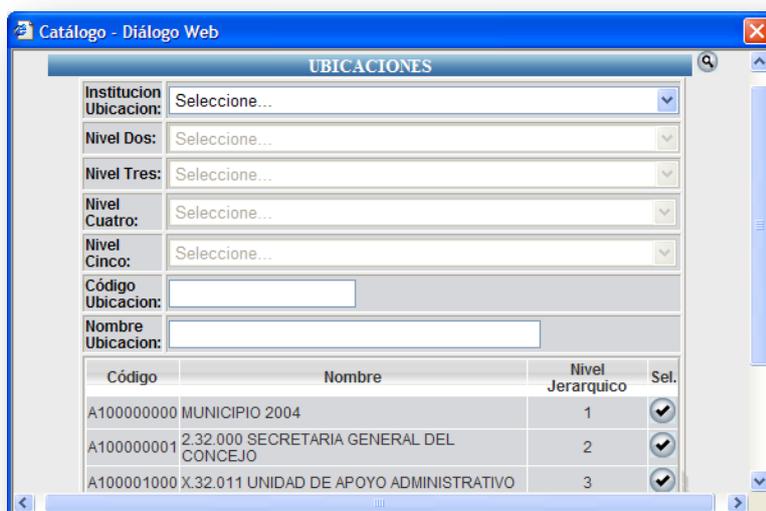
Nombre	Descripción
Modelo	El modelo del Bien
Marca	La marca del Bien
Serie	La serie del Bien
Color	El color del Bien
Otras	Alguna otra característica del Bien

Para el caso del panel de Características por no ser datos de suma importancia para el ingreso del Bien no existen validaciones para este panel.

En el panel siguiente el cual es Datos de Ingreso se encuentran los siguientes atributos a ser llenados.

Nombre	Descripción
Tipo de Ingreso	El motivo del ingreso del Bien
Numero de Acta	Numero de Acta para el Bien
Vida Útil	Vida Útil del Bien
Tipo de Bien	Activo Fijo o Sujeto a Control
Fecha Acta	Fecha del Acta para el Bien
Ubicación Física	Ubicación Física donde se encuentra el Bien.
Estado	Estado que se adquiere el Bien
Accesorios	Si se desea o no accesorios para dicho Bien.
Observaciones	Alguna observación necesaria para el ingreso del Bien.

Se debe llenar los datos requeridos en el Panel de Ingreso pero en el caso del catálogo de Ubicación se ha dividido en 5 niveles para saber la ubicación del Bien mediante un catálogo.



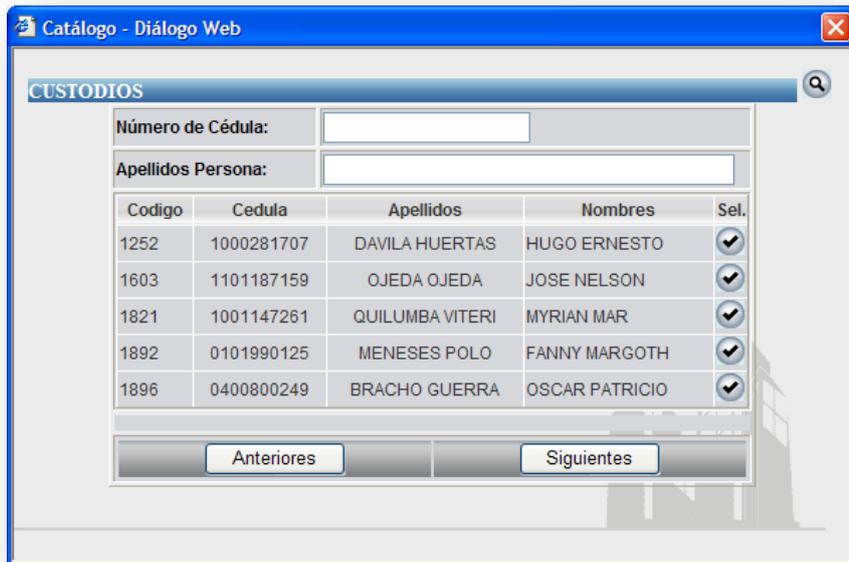
El caso de las validaciones de este panel se debe a que un Bien Activo Fijo debe tener un valor mayor a \$100 caso contrario será un Bien Sujeto de Control.

Para el panel del Custodio se mostrara los siguientes atributos.

Nombre	Descripción
Custodio	Catalogo de Custodio.
Ubicación	Ubicación donde trabaja el custodio.
Cedula	Cedula del Custodio.



Para este panel se deberá escoger un custodio para la asignación del Bien mediante un catálogo mediante las personas que trabajan para el MDMQ.



Finalmente si el usuario escogió el ítem para acceder al panel de Accesorios se mostrara dicho panel con sus siguientes atributos.

Nombre	Descripción
Accesorio	Escoger el Accesorio.
Cantidad	Número de Elementos
Costo	Costo total del accesorio.



Al terminar de llenar los paneles de Ingreso del Bien Municipal se deberá aplastar el botón aceptar para que el sistema tenga constancia de los datos ingresados.

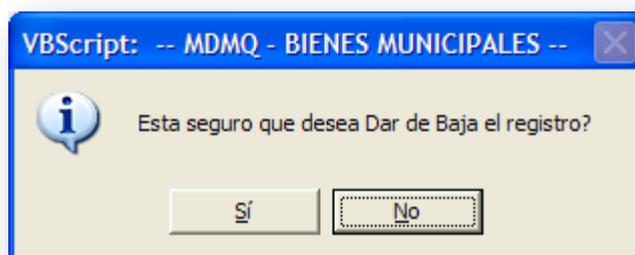
El sistema mostrara un mensaje de Éxito afirmando que el ingreso del Bien fue satisfactorio.



Se generara automáticamente los códigos de barras del Bien Ingresado para el MDMQ.

	Código de Barras	Código	Clase
Código de Barras:	1 Distrito Metropolitano de Quito  COMPUTADOR	141010711008	COMPUTADOR
	2 Distrito Metropolitano de Quito  COMPUTADOR MONITOR	141010714836	MONITOR
	3 Distrito Metropolitano de Quito  COMPUTADOR MOUSE	141010714841	MOUSE
	4 Distrito Metropolitano de Quito  COMPUTADOR TECLADO	141010717360	TECLADO

Si el usuario desea eliminar o realizar una modificación en alguno de los registros debe realizar un click sobre los botones mencionados los cuales se encuentran en las últimas columnas del set de datos desplegado. En este caso, una vez seleccionado el botón Dar de Bajas  aparece el siguiente mensaje que permite validar si el usuario se encuentra seguro de dar de baja el registro seleccionado.



Si el usuario selecciona la opción No, el mensaje de alerta desaparecerá.

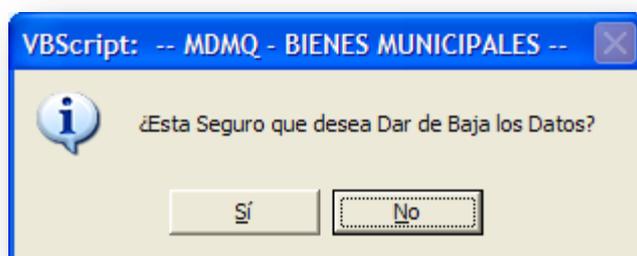
Si el usuario selecciona la opción Si aparece una pantalla que muestra el nombre del bien que se va a dar de baja con su respectivo código. En el caso de que sea un bien compuesto aparecerá un listado con los complementos del bien que se va a dar de baja.

Dentro de esta pantalla se encuentran los siguientes datos:

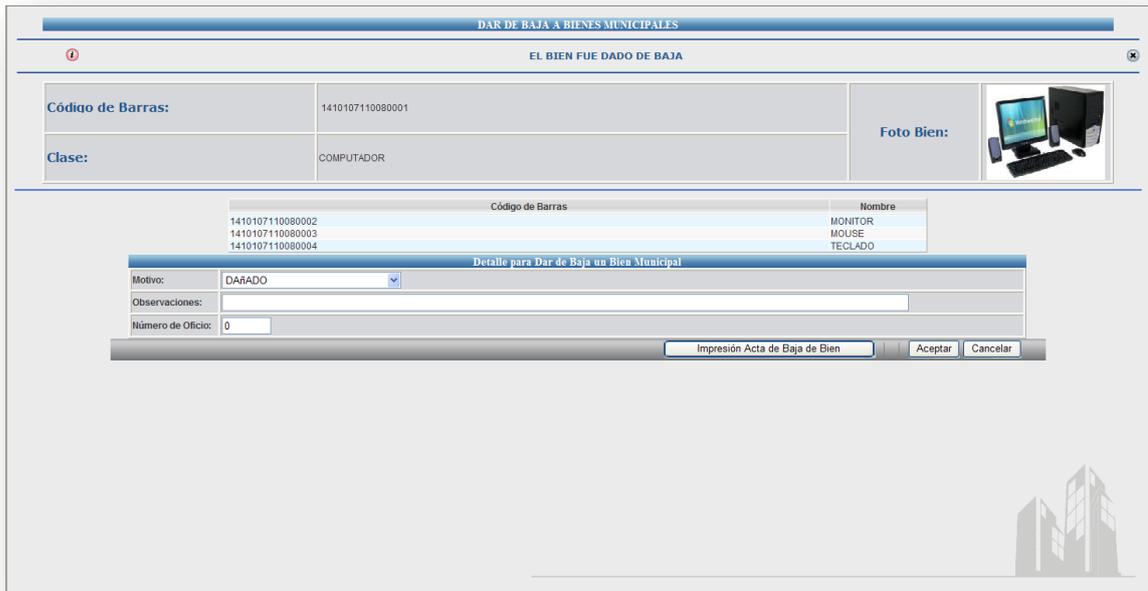
Nombre	Descripción
Motivo	Indica la causa por la que se va a dar de baja el bien.
Observaciones	Permite agregar un detalle del bien.
Número de Oficio	Indica el orden secuencial para el Acta de Baja de Bien.

La pantalla para Dar de Baja un bien es la siguiente:

Si el usuario da clic en el botón Aceptar aparecerá un mensaje de confirmación para Dar de Baja el bien seleccionado. Si el usuario da clic en No, el mensaje de información se cerrará quedando la pantalla para Dar de Baja un bien.



Si el usuario da clic en Si aparecerá otra pantalla que indica que el bien ha sido dado de baja.



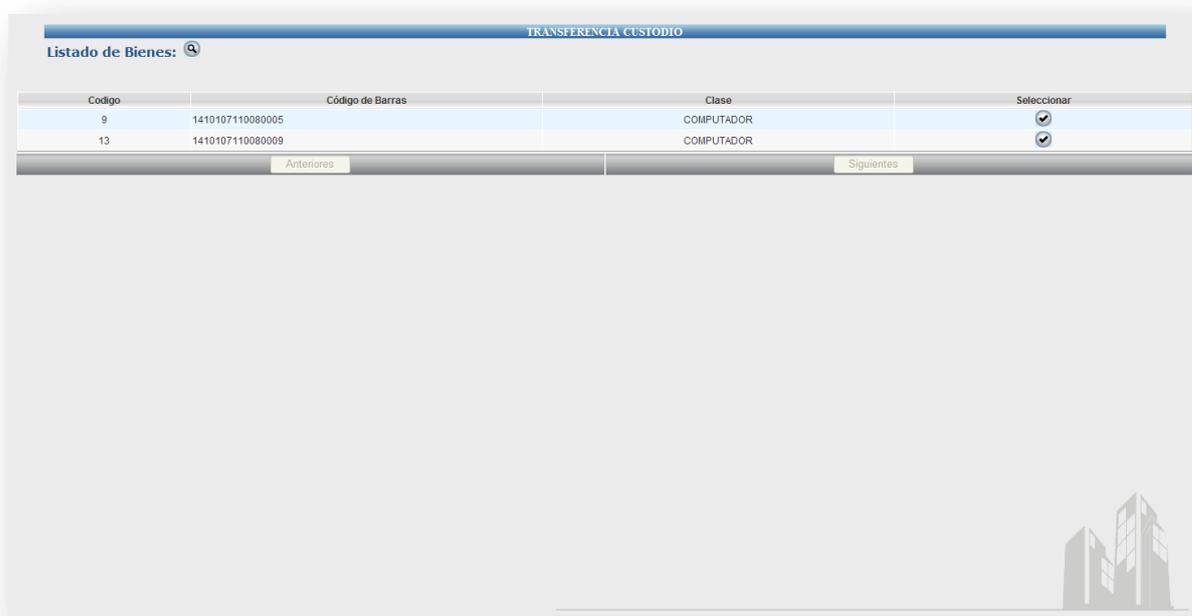
Si el usuario da clic en el botón Imprimir aparecerá una nueva pantalla en la que se muestre el acta correspondiente al bien que se da de baja, la cual contiene información de bien como son: código del bien, clase del bien, el grupo al que pertenece, el tipo de bien que es, el valor de adquisición, etc.



En caso de producirse alguna inconsistencia o algún mensaje explicativo para el usuario se despliega la siguiente pantalla:

1.3.2. Transferencia de Custodios

Dentro de la interfaz de Transferencia de Custodio el usuario tiene la opción de escoger un Bien Municipal para poder realizar el proceso de transferencia de custodio.



En la pantalla de Listado de Bienes podemos utilizar la opción de Buscar  para que se active la opción de buscar por el código del Bien o por su código de Barra.



El usuario deberá escoger un Bien para asignar un nuevo Custodio. Se abrirá una pantalla la cual mostrara el detalle del Bien tales como: Código de Barra, la Clase del Bien, el Custodio Actual, la Ubicación del Custodio y la Ubicación del Bien.

Código de Barras:	1410107110080005	Foto Bien :
Clase:	COMPUTADOR	
Custodio Actual:	DAVILA HUERTAS HUGO ERNESTO	
Ubicacion Custodio:	MUNICIPIO 2004	
Ubicacion Bien:	MUNICIPIO 2004	

Además, la pantalla de Cambio también será mostrada en pantalla los datos que se requieren para completar este proceso.

Nombre	Descripción
Nombre Custodio	Nombre del nuevo Custodio.
Ubicación	Ubicación del Bien.
Motivo	La causa porque se realiza este proceso.
Oficio	Numero de Oficio
Observaciones	Detalle del Traspaso del Bien

Detalle de Custodio

Nuevo Custodio:	<input type="text"/>
Ubicación:	<input type="text"/>
Motivo:	CAMBIO CUSTODIO
Oficio:	0
Observaciones:	<input type="text"/>

Una vez que el usuario lleno los datos requeridos para el traspaso de bienes, el usuario podrá escoger entre aceptar sus cambios o cancelar el proceso.

TRANSFERENCIA CUSTODIO

Código de Barras:	1410107110080005	Foto Bien :
Clase:	COMPUTADOR	
Custodio Actual:	DAVILA HUERTAS HUGO ERNESTO	
Ubicacion Custodio:	MUNICIPIO 2004	
Ubicacion Bien:	MUNICIPIO 2004	

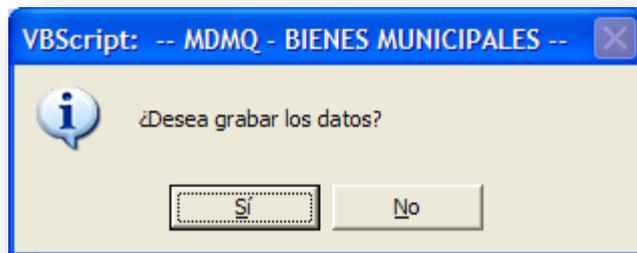
Detalle de Custodio

Nuevo Custodio:	1821
Ubicación:	A100000001 2.32.000 SECRETARIA GENERAL DEL CONCEJO
Motivo:	CAMBIO CUSTODIO
Oficio:	3
Observaciones:	CAMBIO DE CUSTODIO

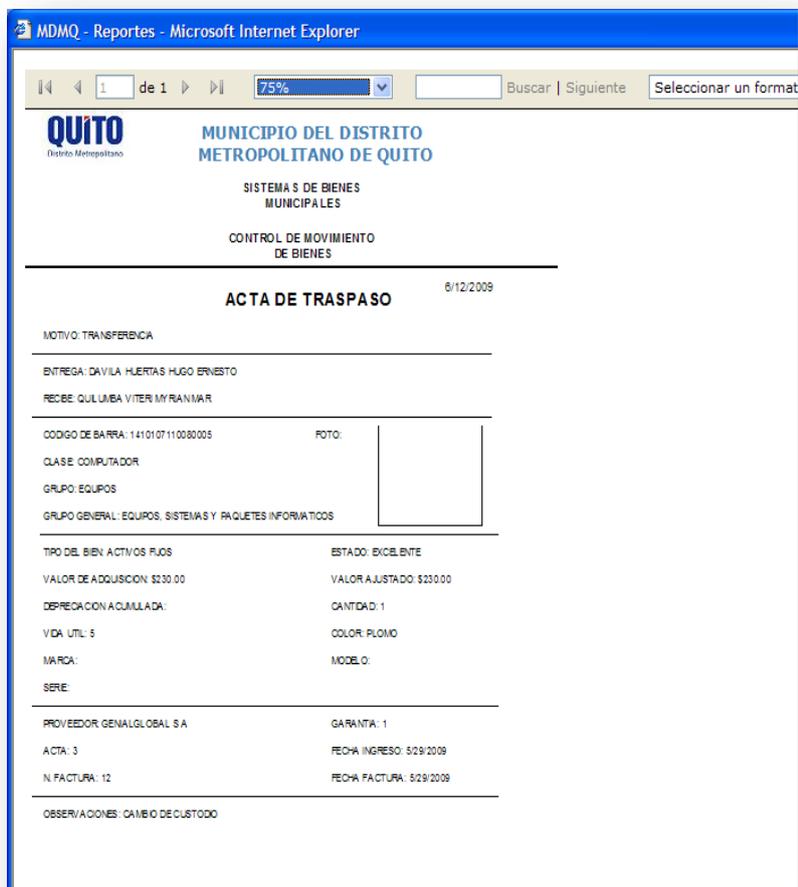
Listado de Bienes:

Código	Código de Barras	Clase	Seleccionar
9	1410107110080005	COMPUTADOR	<input checked="" type="checkbox"/>
13	1410107110080009	COMPUTADOR	<input checked="" type="checkbox"/>

El sistema mostrara en pantalla un mensaje de confirmación para ingresar los datos al sistema.

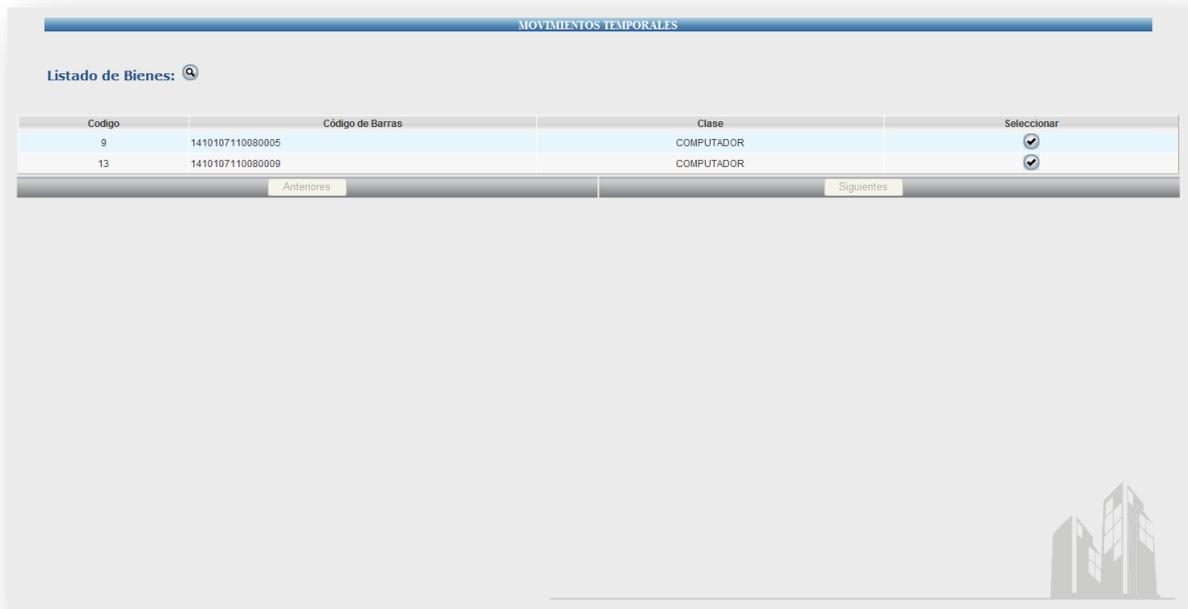


Una vez realizado el proceso de Transferencia de Bienes, el sistema habilitara la opción de imprimir el reporte respectivo.



1.3.3. Movimientos Temporales

Dentro de la interfaz de Movimientos Temporales el usuario tiene la opción de escoger un Bien Municipal para realizar el proceso de traspasar un bien temporalmente a un custodio.



Una vez escogido un Bien Municipal el proceso de Movimientos Temporales es similar al proceso de Transferencia de Custodio.

Se abrirá una pantalla la cual mostrara el detalle del Bien tales como: Código de Barra, la Clase del Bien, el Custodio Actual, la Ubicación del Custodio y la Ubicación del Bien.

Código de Barras:	1410107110080009	Foto Bien :	
Clase:	COMPUTADOR		
Custodio Actual:	BRACHO GUERRA OSCAR PATRICIO		
Ubicación Custodio:	MUNICIPIO 2004		
Ubicación Bien:	MUNICIPIO 2004		

Además, la pantalla de Cambio también será mostrada en pantalla los datos que se requieren para completar este proceso.

Nombre	Descripción
Nombre Custodio	Nombre del nuevo Custodio.
Ubicación	Ubicación del Bien.
Motivo	La causa porque se realiza este proceso.
Oficio	Numero de Oficio
Observaciones	Detalle del Traspaso del Bien

Detalle de Custodio

Nuevo Custodio:	<input type="text"/>
Ubicación:	<input type="text"/>
Motivo:	CAMBIO CUSTODIO
Oficio:	0
Observaciones:	<input type="text"/>

Una vez que el usuario lleno los datos requeridos para el traspaso de bienes, el usuario podrá escoger entre aceptar sus cambios o cancelar el proceso.

MOVIMIENTOS TEMPORALES

Código de Barras:	1410107110080009	Foto Bien :	
Clase:	COMPUTADOR		
Custodio Actual:	BRACHO GUERRA OSCAR PATRICIO		
Ubicación Custodio:	MUNICIPIO 2004		
Ubicación Bien:	MUNICIPIO 2004		

Detalle de Custodio

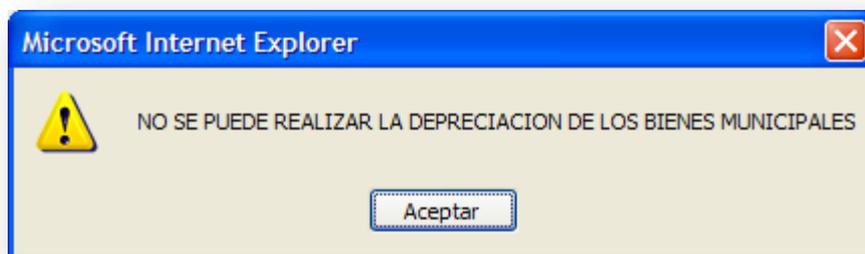
Nuevo Custodio:	<input type="text"/>
Ubicación:	<input type="text"/>
Motivo:	CAMBIO CUSTODIO
Oficio:	<input type="text"/>
Observaciones:	<input type="text"/>

Listado de Bienes:

Código	Código de Barras	Clase	Seleccionar
9	1410107110080005	COMPUTADOR	<input checked="" type="checkbox"/>
13	1410107110080009	COMPUTADOR	<input checked="" type="checkbox"/>

1.3.4. Depreciación

La pantalla de Depreciación dependerá de los periodos previstos por el MDMQ, en el caso que no se encuentre en un periodo permitido aparecerá en pantalla un mensaje de error.



La pantalla de Depreciación constara con un encabezado de los periodos permitidos para depreciar los Bienes del MDMQ.

FECHA ACTUAL:	viernes, 12 de junio de 2009
FECHA INICIAL:	01/07/2008
FECHA FINAL:	30/06/2009

Una vez realizada la depreciación de los Bienes Municipales aparecerá en pantalla la cantidad de Bienes depreciados y cuantos no fueron depreciados.

BIENES DEPRECIADOS:	0
BIENES SIN DEPRECIAR:	0

Erogaciones

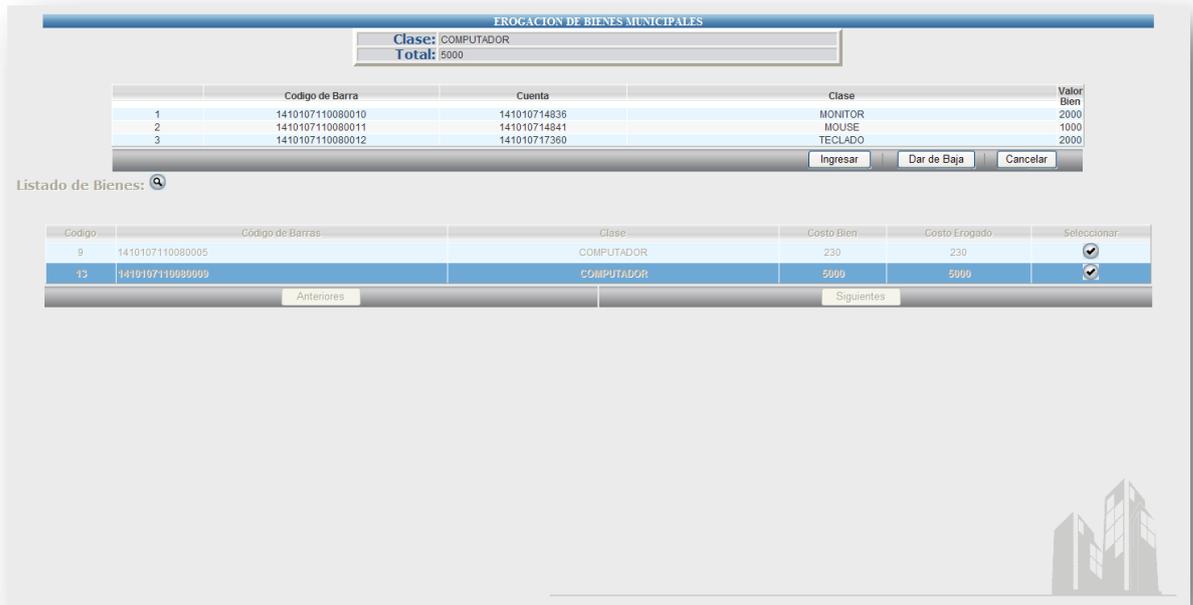
El usuario deberá escoger entre un Bien Municipal ingresado para realizar el proceso de Erogación de Bienes.

EROGACION DE BIENES MUNICIPALES					
Listado de Bienes: 					
Codigo	Código de Barras	Clase	Costo Bien	Costo Erogado	Seleccionar
9	1410107110080005	COMPUTADOR	230	230	<input checked="" type="checkbox"/>
13	1410107110080009	COMPUTADOR	5000	5000	<input checked="" type="checkbox"/>

Anteriores
Siguintes

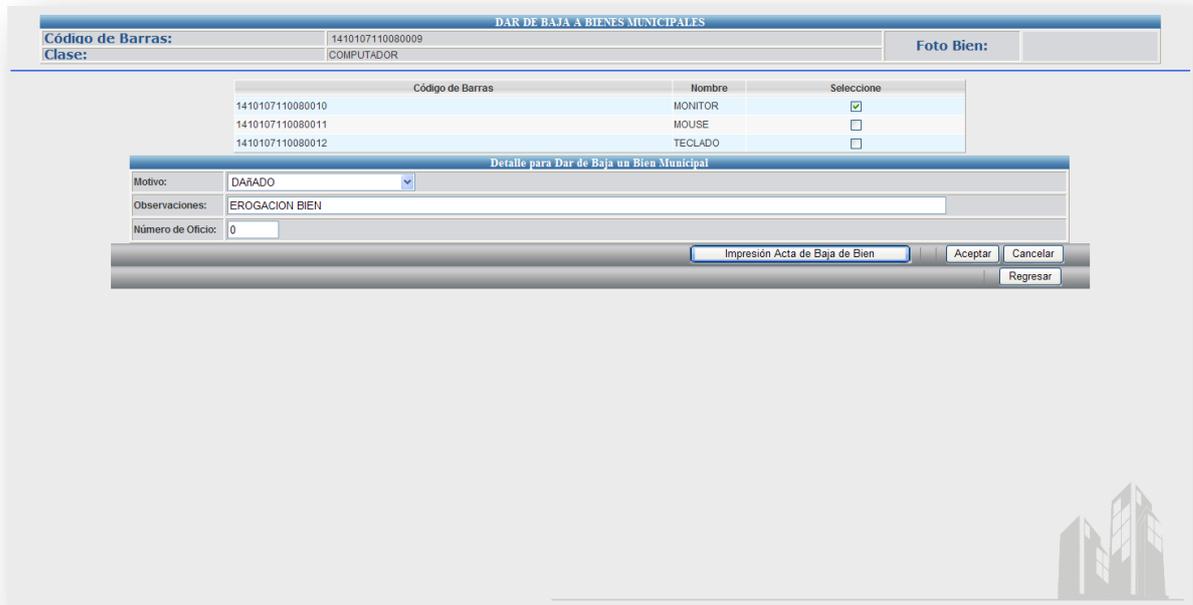


Al momento de escoger un Bien, el sistema mostrara en pantalla la información del valor actual con sus respectivos componentes del bien compuesto.



El usuario deberá dar de Baja las partes del Bien compuesto que desea erogar aplastando el botón Dar de Baja.

El usuario podrá escoger que componente del Bien Compuesto debe ser dado de Baja en la pantalla de Dar de Baja.



Una vez que se dio de baja alguno de los componentes del Bien Compuesto se deberá ingresar un nuevo componente en remplazo al que fue erogado, para

eso el usuario deberá apastar el botón de ingresar lo cual realizara el proceso de operación de un Bien.

The screenshot shows a web form titled "Ingreso de Bienes Municipales". At the top, there is a text input field for "Código de Barras:". Below this is a section titled "Foto Bien Municipal" which contains a large grey placeholder box and a small "Examinar..." button with a plus icon. Underneath is a section titled "Cuentas" containing several form fields: "* Clase:" with a dropdown menu, "* Código Generado Bien:" with a dropdown menu, "Estructura:" with a dropdown menu, "Grupo:" with a dropdown menu, and "Tipo de Clase:" with a dropdown menu. At the bottom of the form are two buttons: "Anteriores" and "Siguietes".

Cuando el usuario haya realizado este proceso, el valor del Bien cambiara según sea la cantidad que ingreso de sus componentes.

1.3.5. Históricos

La pantalla de Históricos consta de los movimientos que el usuario realizo en el Bien Municipal. En la primera pantalla muestra todos los bienes que tengan al menos un movimiento realizado.

The screenshot shows a web page titled "Histórico de Bienes Municipales". It features a table with the following data:

Codigo	Código de Barras	Clase	Seleccionar
9	1410107110080005	COMPUTADOR	<input checked="" type="checkbox"/>
13	1410107110080009	COMPUTADOR	<input checked="" type="checkbox"/>

Below the table are two buttons: "Anteriores" and "Siguietes".

Una vez que el usuario escoja un Bien Municipal aparecerá en pantalla la información detallada del Bien, con una tabla de todos los movimientos que realizo.

Historico de Bienes Municipales

Código de Barras:	Codigo de Barra	Codigo	Clase	Foto Bien:
	1410107110080005	141010711008	COMPUTADOR	
	1410107110080006	141010714836	MONITOR	
	1410107110080007	141010714841	MOUSE	
	1410107110080008	141010717360	TECLADO	

Detalle de Bienes						
Código de Barras	Motivo	Clase	Fecha Creacion	Valor Adquisicion	Valor Actual	Usuario
1410107110080005	INGRESO	COMPUTADOR	29/05/2009	230	230	installuser
1410107110080005	ACTUALIZACION	COMPUTADOR	29/05/2009	230	230	installuser

[Anteriores](#)
[Siguietes](#)
[Impresión Historico](#) [Cancelar](#)

Listado de Bienes:

Código	Código de Barras	Clase	Seleccionar
9	1410107110080005	COMPUTADOR	<input checked="" type="checkbox"/>
13	1410107110080009	COMPUTADOR	<input checked="" type="checkbox"/>

[Anteriores](#)
[Siguietes](#)

Si el usuario desea podrá imprimir un reporte histórico el cual muestra toda la información detallada del bien.

MDMQ - Reportes - Microsoft Internet Explorer

QUITO
MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO
SISTEMA S DE BIENES MUNICIPALES
CONTROL DE MOVIMIENTO DE BIENES

HISTORIAL DEL BIEN

CODIGO DE BARRA: 1410107110080005 FOTO:

CLASE: COMPUTADOR

GRUPO: EQUIPOS

GRUPO GENERAL: EQUIPOS, SISTEMAS Y PAQUETES INFORMATICOS

MOVIMIENTOS

ACTUALIZACION FECHA: 5/29/2009

ORIGEN:

DESTINO:

RESPONSABLE: installuser

INGRESO FECHA: 5/29/2009

ORIGEN:

DESTINO:

RESPONSABLE: installuser

Ejecutado por: installuser installuser Pág: 1/1

1.4. Administración

Dentro de la sección de Administración tenemos Cuentas y Parámetros.



1.4.1. Cuentas

En Administración de Cuenta el usuario tendrá la posibilidad de realizar las operaciones respectivas para las cuentas de los Bienes.

En la primera pantalla muestra Todas las cuentas creadas por el usuario con su respectiva información, las cuales el usuario podrá eliminar o modificar según sea el caso.

Código	Cuentas	Fecha	Vida Útil	Editar	Borrar
1	ACTIVO	08/03/2009			
14	INVERSIONES EN BIENES DE LARGA DURACION	08/03/2009			
141	BIENES DE ADMINISTRACION	08/03/2009			
14101	BIENES MUEBLES	08/03/2009			
1410103	MOBILIARIO	08/03/2009	10		
14101031	USO GENERAL	08/03/2009			
141010310002	ABACO	08/03/2009	10		
141010310072	AFICHE	08/03/2009	10		
141010310144	AJEDREZ	08/03/2009	10		
141010310182	ALACENA	10/03/2009	10		

El usuario si desea modificar cualquier clase deberá aplastar el botón , se abrirá la pantalla de edición con los siguientes campos que pueden ser editados.

Nombre	Descripción
Tipo de Cuenta	La cuenta que contendrá a la Clase ingresada.
Código	Código propio de la Clase.

Estado	El estado de la cuenta.
Vida Útil	Vida Útil del grupo de la Clase.
Nombre	Nombre de la Cuenta.
Fecha	Fecha de creación de la Clase.
Porcentaje	Porcentaje que tiene la Clase.

ADMINISTRACIÓN DE CUENTAS
Ingreso de Cuenta

Tipo de Cuentas: 14101031 USO GENERAL

Código: 141010310002 Nombre: ABACO

Estado: EXCELENTE Fecha: 08/03/2009

Vida Útil (años): 10 Porcentaje (%):

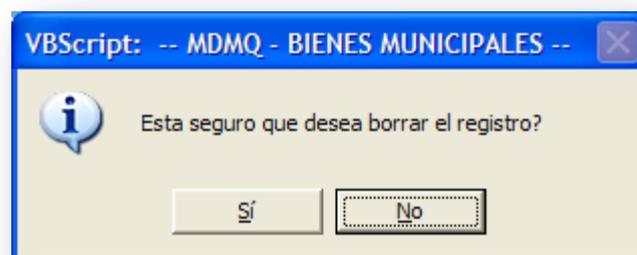
Aceptar Cancelar

Listado de Cuentas

Código	Cuentas	Fecha	Vida Útil	Editar	Borrar
1	ACTIVO	08/03/2009	10	E	X
14	INVERSIONES EN BIENES DE LARGA DURACION	08/03/2009	10	E	X
141	BIENES DE ADMINISTRACION	08/03/2009	10	E	X
14101	BIENES MUEBLES	08/03/2009	10	E	X
1410103	MOBILIARIO	08/03/2009	10	E	X
14101031	USO GENERAL	08/03/2009	10	E	X
141010310002	ABACO	08/03/2009	10	E	X
141010310072	AFICHE	08/03/2009	10	E	X
141010310144	ALJEDREZ	08/03/2009	10	E	X
141010310162	ALACENA	10/03/2009	10	E	X

Anteriores Siguietes

Si el usuario desea borrar una clase de la cuenta, deberá aplastar el icono  que se encuentra en la tabla. El usuario debe estar seguro que la cuenta que desea borrar no contenga otras cuentas. Aparecerá un mensaje de confirmación para borrar el dato que el usuario crea conveniente.



Si el usuario desea ingresar una nueva cuenta deberá aplastar el botón  que se encuentra en la parte superior izquierda de la pantalla. Se deberá colocar una cuenta la cual contenga a la que va a ser ingresada para eso utilizamos el control de Catalogo y llenamos el dato requerido por el programa.

ADMINISTRACIÓN DE CUENTAS
Ingreso de Cuenta

Tipo de Cuentas:

Se abrirá el catálogo de Cuentas la cual podemos buscar y escoger la que el usuario desee.

Catálogo - Diálogo Web

CUENTAS

Código Cuenta :

Nombre Cuenta :

Código	Nombre	Sel.
1	ACTIVO	<input checked="" type="checkbox"/>
14	INVERSIONES EN BIENES DE LARGA DURACION	<input checked="" type="checkbox"/>
141	BIENES DE ADMINISTRACION	<input checked="" type="checkbox"/>
14101	BIENES MUEBLES	<input checked="" type="checkbox"/>
1410103	MOBILIARIO	<input checked="" type="checkbox"/>

Una vez que el usuario escogió la cuenta aparecerá el formulario a ser llenado dependiendo de cual haya seleccionado.

ADMINISTRACIÓN DE CUENTAS
Ingreso de Cuenta

Tipo de Cuentas: 1410103 MOBILIARIO

Código: Nombre:

Estado: EXCELENTE Fecha: 08/03/2009

ADMINISTRACIÓN DE CUENTAS
Ingreso de Cuenta

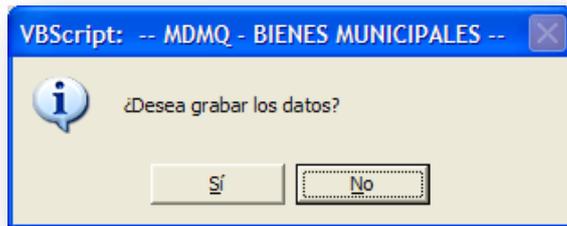
Tipo de Cuentas: 14101031 USO GENERAL

Código: Nombre:

Estado: EXCELENTE Fecha: 08/03/2009

Vida Útil (años): Porcentaje (%):

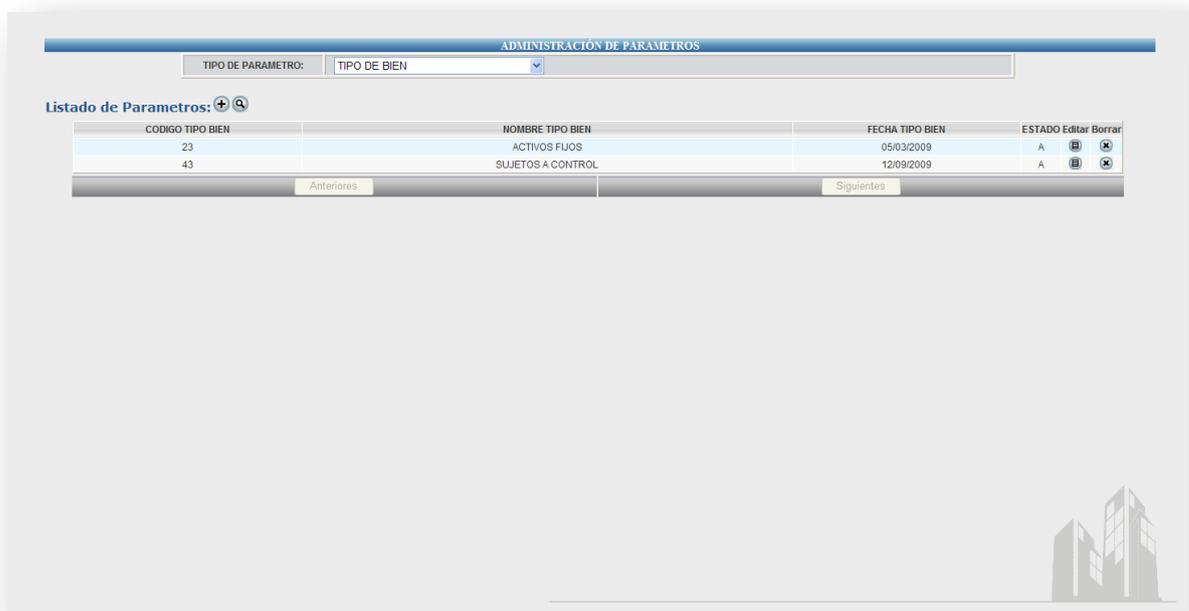
Para finalizar el usuario deberá aceptar la operación que está realizando, la cual aparecerá un mensaje de confirmación.



1.4.2. Parámetros

1.4.2.1. Tipo de Bien

Se mostrara en pantalla todos los Tipos de Bien que el usuario creo en la aplicación en una tabla la cual tenemos las opciones de modificar y eliminar.



Si el usuario aplasta la opción de modificar  , el sistema mostrara todos los atributos que el usuario podrá modificar.

Nombre	Descripción
Nombre de Tipo de Bien	Nombre del nuevo Tipo de Bien
Fecha	Fecha de Creación.
Estado	Estado del Tipo de Bien.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: TIPO DE BIEN

Ingreso de Parámetros

Codigo: 23

Nombre: ACTIVOS FIJOS *Estado: ACTIVO

Fecha: 05/03/2009
DDMMYYYY

Aceptar Cancelar

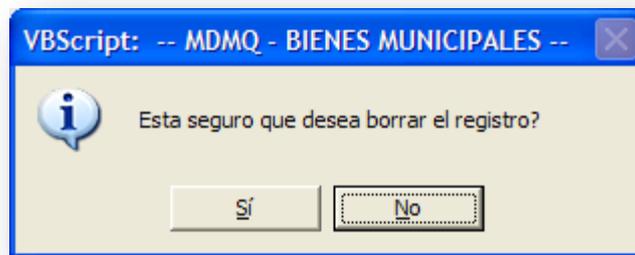
Listado de Parametros:

CODIGO TIPO BIEN	NOMBRE TIPO BIEN	FECHA TIPO BIEN	ESTADO
23	ACTIVOS FIJOS	05/03/2009	A
43	SUJETOS A CONTROL	12/09/2009	A

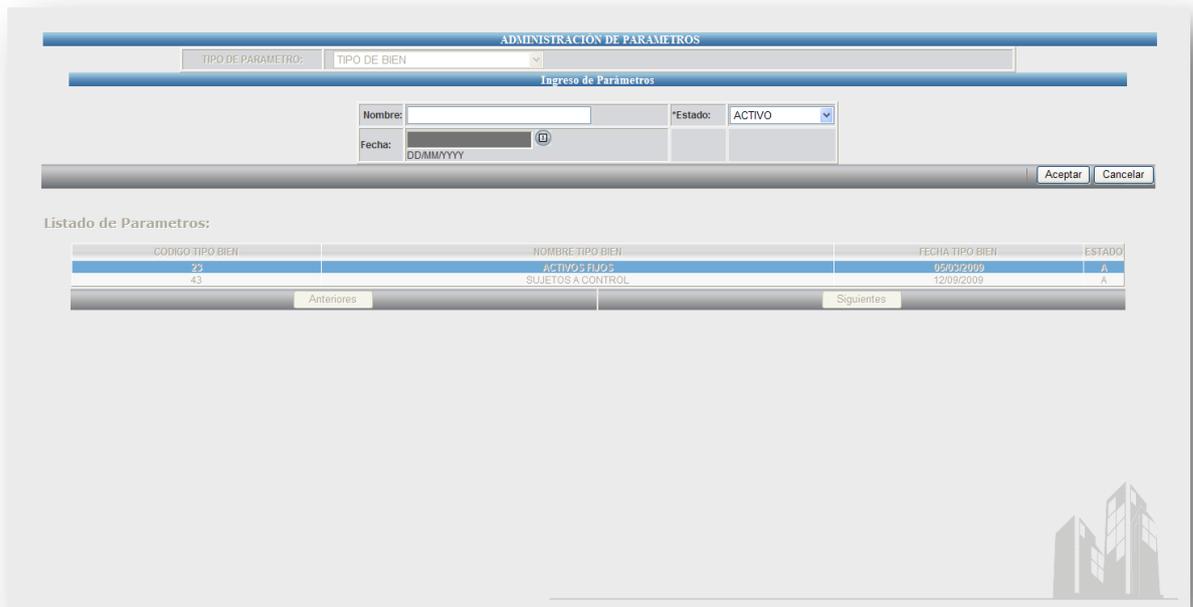
Anteriores Siguintes



Para borrar un registro el usuario deberá utilizar el botón de Borrar que se encuentra en la tabla,  lo cual el sistema mostrara un mensaje de confirmación.

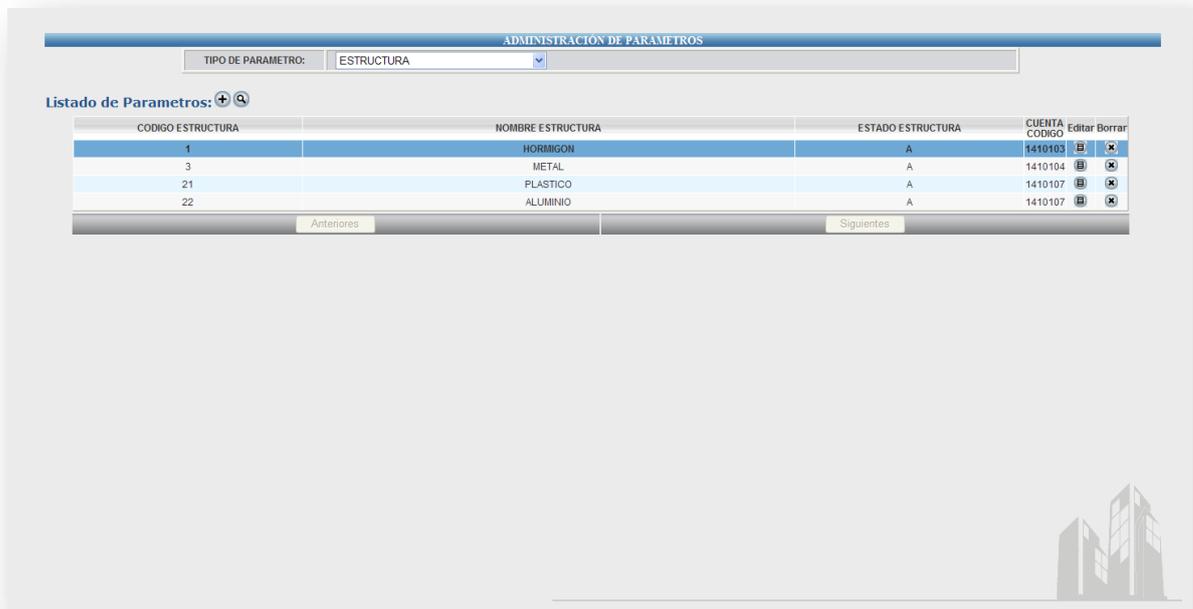


El usuario puede agregar otros tipos de bien los cuales se mostraran en el sistema.



1.4.2.2. Estructura

Al escoger la opción de estructura automáticamente cambiara los datos que se mostraron en pantalla por los datos que pertenecen a la Estructura del Bien.



Al igual que el parámetro de Tipo de Bien el parámetro de Estructura se puede borrar y modificar, al momento de modificar se mostrara en pantalla los siguientes atributos.

Nombre	Descripción
Cuenta	Catalogo de Cuentas
Nombre de Estructura	Nombre de la nueva Estructura
Fecha	Fecha de Creación.
Estado	Estado de la Estructura.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: ESTRUCTURA

Ingreso de Parametros

Código: 1

Cuenta: 1410103 MOBILIARIO

Nombre: HORMIGON *Estado: ACTIVO

Fecha: DD/PP/AAAA
DDMMYYYY

Aceptar Cancelar

Listado de Parametros:

CODIGO ESTRUCTURA	NOMBRE ESTRUCTURA	ESTADO ESTRUCTURA	CUENTA CODIGO
1	HORMIGON	A	1410103
3	METAL	A	1410104
21	PLASTICO	A	1410107
22	ALUMINIO	A	1410107

Anteriores Siguietes



Si el usuario desea ingresar una nueva Estructura, se mostrara en pantalla los atributos vacios los cuales deberán ser llenados por el usuario.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: ESTRUCTURA

Ingreso de Parametros

Cuenta:

Nombre: *Estado: ACTIVO

Fecha:
DDMMYYYY

Aceptar Cancelar

Listado de Parametros:

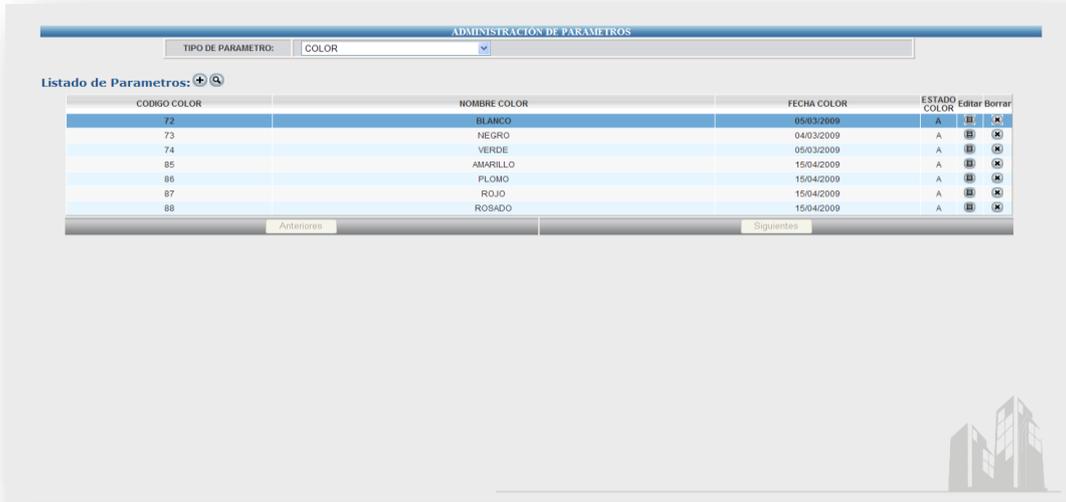
CODIGO ESTRUCTURA	NOMBRE ESTRUCTURA	ESTADO ESTRUCTURA	CUENTA CODIGO
1	HORMIGON	A	1410103
3	METAL	A	1410104
21	PLASTICO	A	1410107
22	ALUMINIO	A	1410107

Anteriores Siguietes



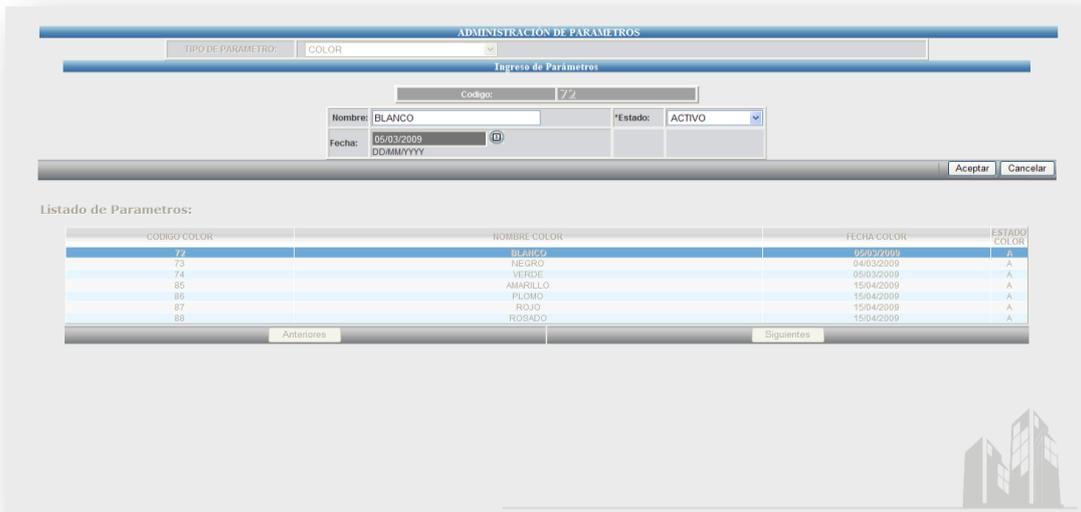
1.4.2.3. Color

Al momento que el usuario escoja la opción de Color, aparecerá en pantalla todos los colores que el sistema posee.

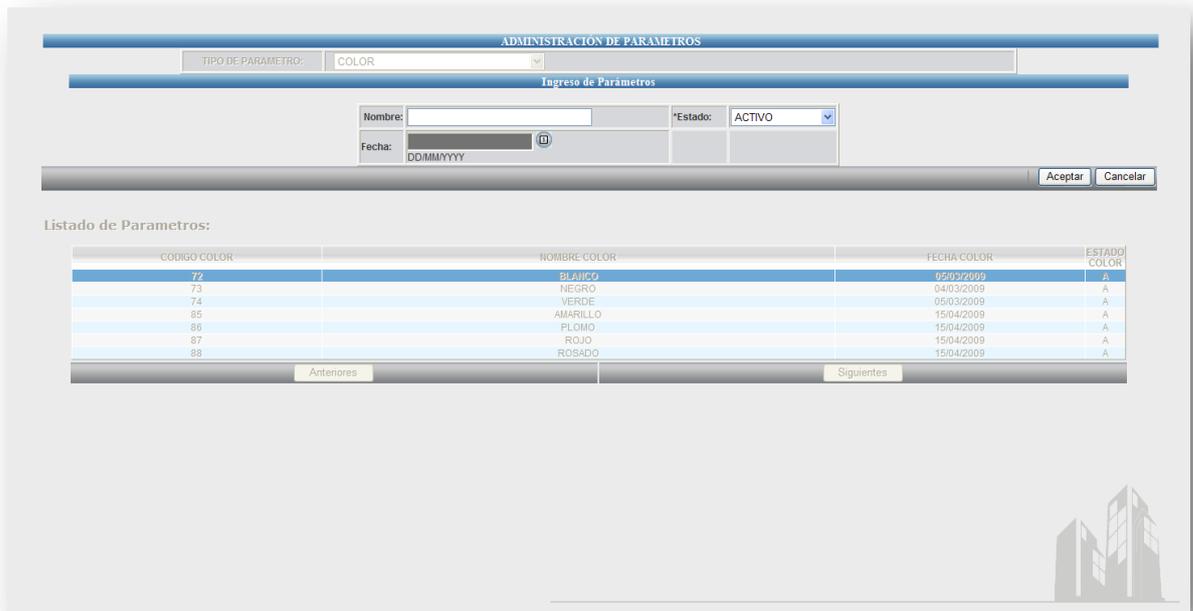


Al momento de modificar un color el usuario deberá aplastar el botón de modificar para que el sistema muestre en pantalla la información del color.

Nombre	Descripción
Nombre del Color	Nombre del Color.
Fecha	Fecha de Creación.
Estado	Estado del Color.

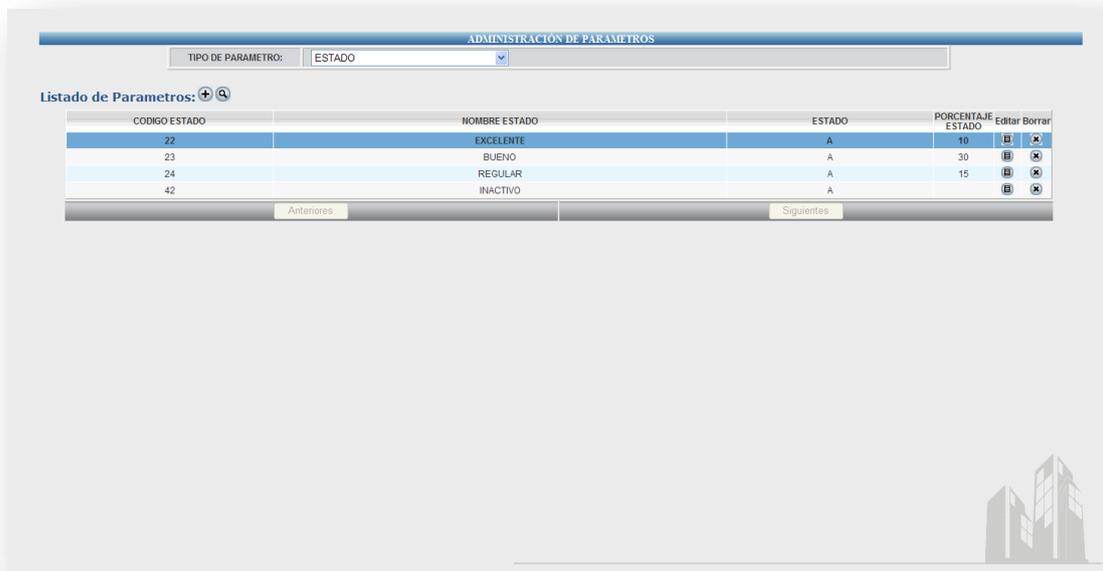


Si el usuario desea agregar un dato nuevo deberá aplastar el botón de añadir y se mostrara en pantalla los atributos vacios.



1.4.2.4. Estado

Al escoger la opción de Estado se mostrara en pantalla los estados que el usuario creo para la aplicación.



En el caso de los parámetros de Estados al momento de modificar aparecerá en pantalla los atributos que podrán ser modificados por el usuario.

Nombre	Descripción
Nombre del Estado	Nombre del Estado.
Fecha	Fecha de Creación.
Estado	Estado del Estado.
Porcentaje	Porcentaje del Estado

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: ESTADO

Ingreso de Parámetros

Código: 22

Nombre: EXCELENTE *Estado: ACTIVO

Fecha: 17/03/2009 *Porcentaje: (%) 10

DDMMYYYY

Aceptar Cancelar

Listado de Parámetros:

CODIGO ESTADO	NOMBRE ESTADO	ESTADO	PORCENTAJE ESTADO
22	EXCELENTE	A	10
23	BUENO	A	30
24	REGULAR	A	15
42	INACTIVO	A	

Anteriores Siguietes

Si el usuario desea añadir un nuevo atributo del Estado aparecerá en pantalla los atributos a ser llenados.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: ESTADO

Ingreso de Parámetros

Nombre: *Estado: ACTIVO

Fecha: *Porcentaje: (%) 0

DDMMYYYY

Aceptar Cancelar

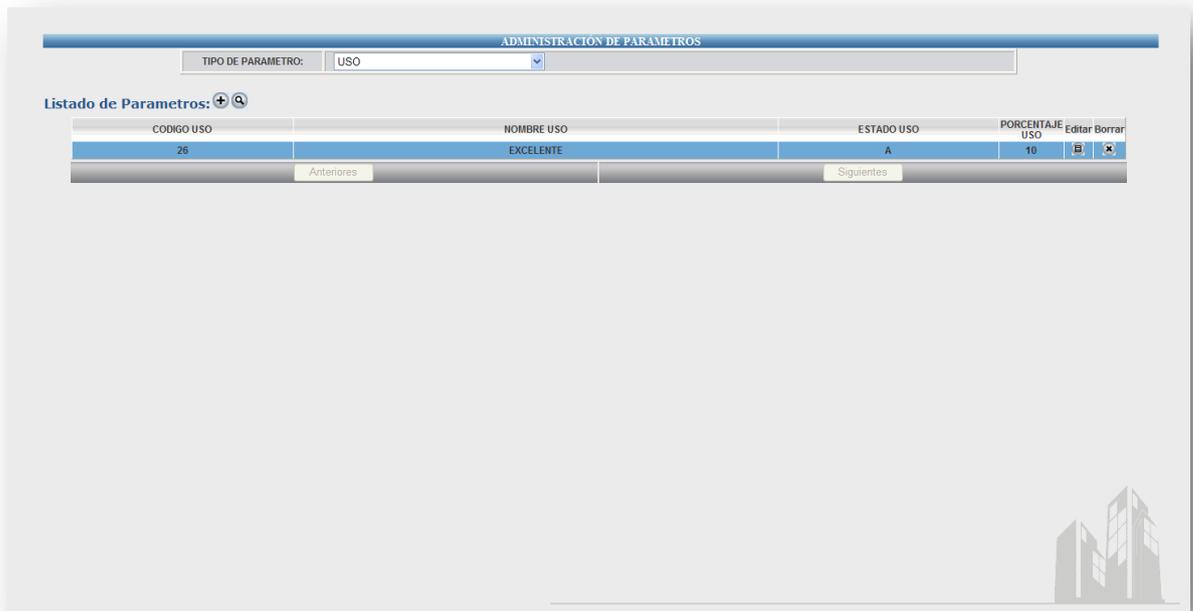
Listado de Parámetros:

CODIGO ESTADO	NOMBRE ESTADO	ESTADO	PORCENTAJE ESTADO
22	EXCELENTE	A	10
23	BUENO	A	30
24	REGULAR	A	15
42	INACTIVO	A	

Anteriores Siguietes

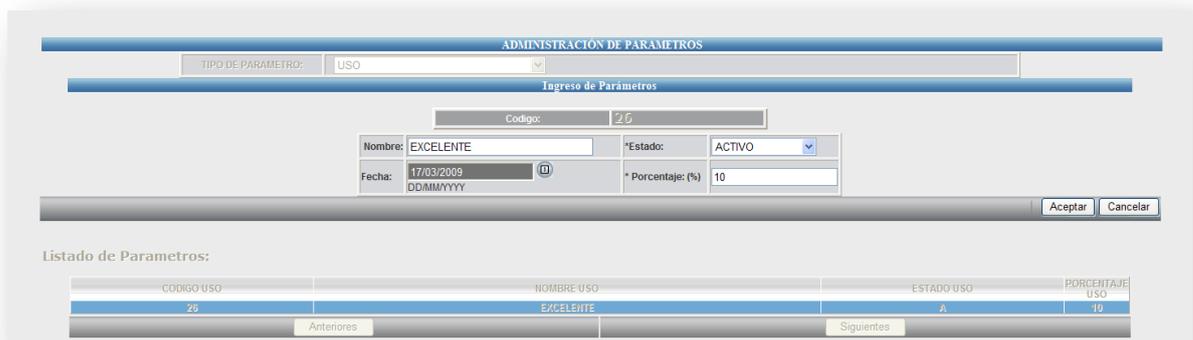
1.4.2.5. Uso

Al escoger la opción de Uso se mostrara en pantalla los estados que el usuario creo para la aplicación.



En el caso del parámetro del Uso al momento de modificar aparecerá en pantalla los atributos que podrán ser modificados por el usuario.

Nombre	Descripción
Nombre del Uso	Nombre del Uso.
Fecha	Fecha de Creación.
Estado	Estado del Uso.
Porcentaje	Porcentaje del Uso



Si el usuario desea añadir un nuevo atributo del Uso aparecerá en pantalla los atributos a ser llenados.

ADMINISTRACION DE PARAMETROS

TIPO DE PARAMETRO: USO

Ingreso de Parametros

Nombre: *Estado: ACTIVO

Fecha: DDMMYYYY *Porcentaje: (%) 0

Aceptar Cancelar

Listado de Parametros:

CODIGO USO	NOMBRE USO	ESTADO USO	PORCENTAJE USO
26	EXCELENTE	A	10

Anteriores Siguietes

1.4.2.6. Tipo de Cuenta

Al escoger la opción de Tipo de Cuenta automáticamente cambiara los datos que se mostraron en pantalla por los datos que pertenecen al tipo del bien.

ADMINISTRACION DE PARAMETROS

TIPO DE PARAMETRO: TIPO DE CUENTA

Listado de Parametros:

CODIGO TIPO	NOMBRE TIPO	FECHA TIPO	ESTADO TIPO	Editar	Borrar
1	OFICINA	12/03/2009	A		
2	ESCRITORIO	07/04/2009	A		
21	MANUAL	04/01/1980	A		
22	OPTICO	30/04/2009	A		
23	PLASMA	07/03/2009	A		

Anteriores Siguietes

Al igual que el parámetro de Estructura el parámetro de Tipo de Cuenta se puede borrar y modificar, al momento de modificar se mostrara en pantalla los siguientes atributos.

Nombre	Descripción
Cuenta	Catalogo de Cuentas
Nombre de Tipo de Cuenta	Nombre del nuevo Tipo de Cuenta
Fecha	Fecha de Creación.
Estado	Estado del Tipo de Cuenta.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: TIPO DE CUENTA

Ingreso de Parametros

Codigo: 1

Cuenta: 141010711008 COMPUTADOR

Nombre: OFICINA *Estado: ACTIVO

Fecha: 12/03/2009 DD/MM/YYYY

Aceptar Cancelar

Listado de Parametros:

CODIGO TIPO	NOMBRE TIPO	FECHA TIPO	ESTADO TIPO
1	OFICINA	12/03/2009	A
2	ESCRITORIO	07/04/2009	A
21	MANUAL	04/01/1980	A
22	OPTICO	30/04/2009	A
23	PLASMA	07/03/2009	A

Anteriores Siguietes

Si se desea ingresar un nuevo Tipo de Cuenta, se mostrara en pantalla los atributos vacios los cuales deberán ser llenados por el usuario.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: TIPO DE CUENTA

Ingreso de Parametros

Cuenta:

Nombre: *Estado: ACTIVO

Fecha: DD/MM/YYYY

Aceptar Cancelar

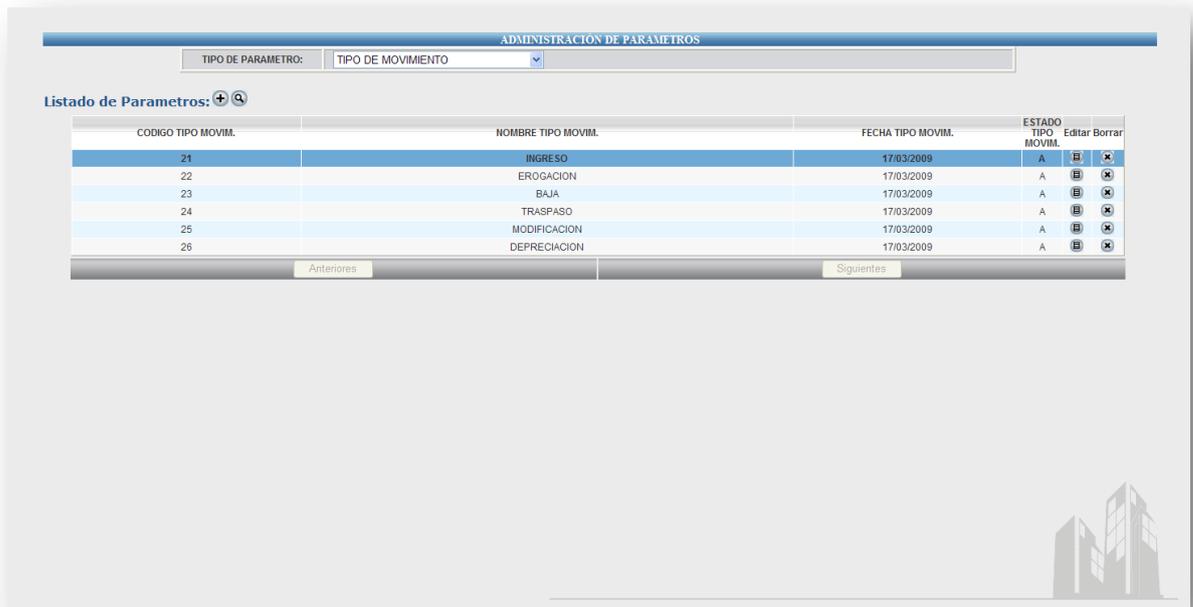
Listado de Parametros:

CODIGO TIPO	NOMBRE TIPO	FECHA TIPO	ESTADO TIPO
1	OFICINA	12/03/2009	A
2	ESCRITORIO	07/04/2009	A
21	MANUAL	04/01/1980	A
22	OPTICO	30/04/2009	A
23	PLASMA	07/03/2009	A

Anteriores Siguietes

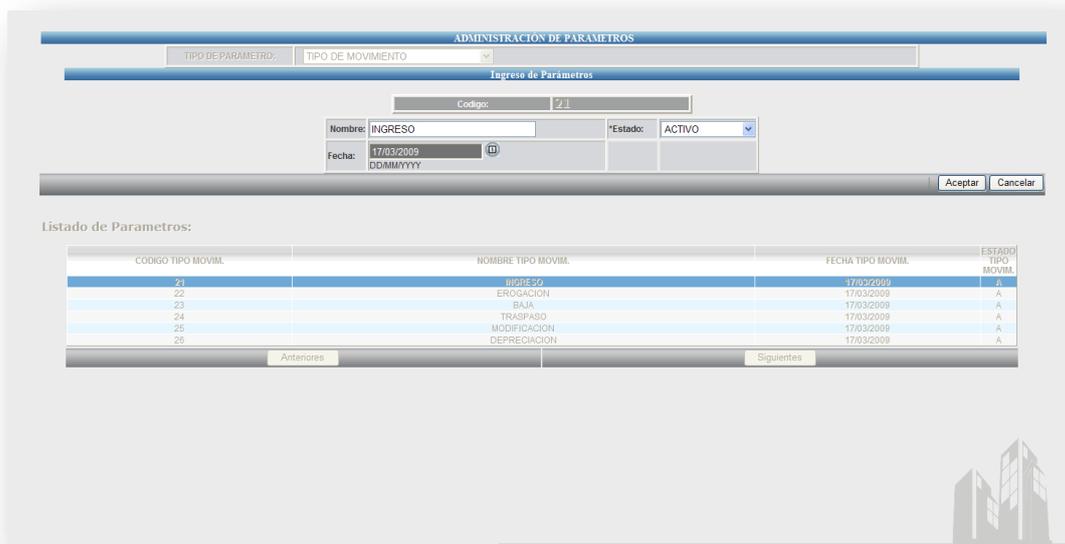
1.4.2.7. Tipo de Movimiento

Al escoger la opción de Tipo de Movimiento automáticamente cambiara los datos que se mostraron en pantalla por los datos que pertenecen al tipo de movimiento del bien.



Al igual que los anteriores parámetros el parámetro de Tipo de Movimiento se puede borrar y modificar, al momento de modificar se mostrara en pantalla los siguientes atributos.

Nombre	Descripción
Nombre de Tipo de Movimiento	Nombre del nuevo Tipo de Movimiento
Fecha	Fecha de Creación.
Estado	Estado del Tipo de Cuenta.



Si se desea ingresar un nuevo Tipo de Movimiento, se mostrara en pantalla los atributos vacios los cuales deberán ser llenados por el usuario.

The screenshot shows the 'ADMINISTRACIÓN DE PARAMETROS' interface. At the top, there is a dropdown menu for 'TIPO DE PARAMETRO:' set to 'TIPO DE MOVIMIENTO'. Below this is the 'Ingreso de Parametros' form with fields for 'Nombre:', 'Fecha:' (format DDMMYYYY), and '*Estado:' (set to 'ACTIVO'). There are 'Aceptar' and 'Cancelar' buttons. Below the form is a table titled 'Listado de Parametros:' with columns: CODIGO TIPO MOVIML., NOMBRE TIPO MOVIML., FECHA TIPO MOVIML., and ESTADO TIPO MOVIML.

CODIGO TIPO MOVIML.	NOMBRE TIPO MOVIML.	FECHA TIPO MOVIML.	ESTADO TIPO MOVIML.
21	IMPRESO	17/03/2009	A
22	EROSION	17/03/2009	A
23	BAJA	17/03/2009	A
24	TRASPASO	17/03/2009	A
25	MODIFICACION	17/03/2009	A
26	DEPRECIACION	17/03/2009	A

Navigation buttons 'Anteriores' and 'Siguientes' are located below the table.

1.4.2.8. Motivo

Al escoger la opción de Motivo automáticamente cambiara los datos que se mostraron en pantalla por los datos que pertenecen al motivo del bien.

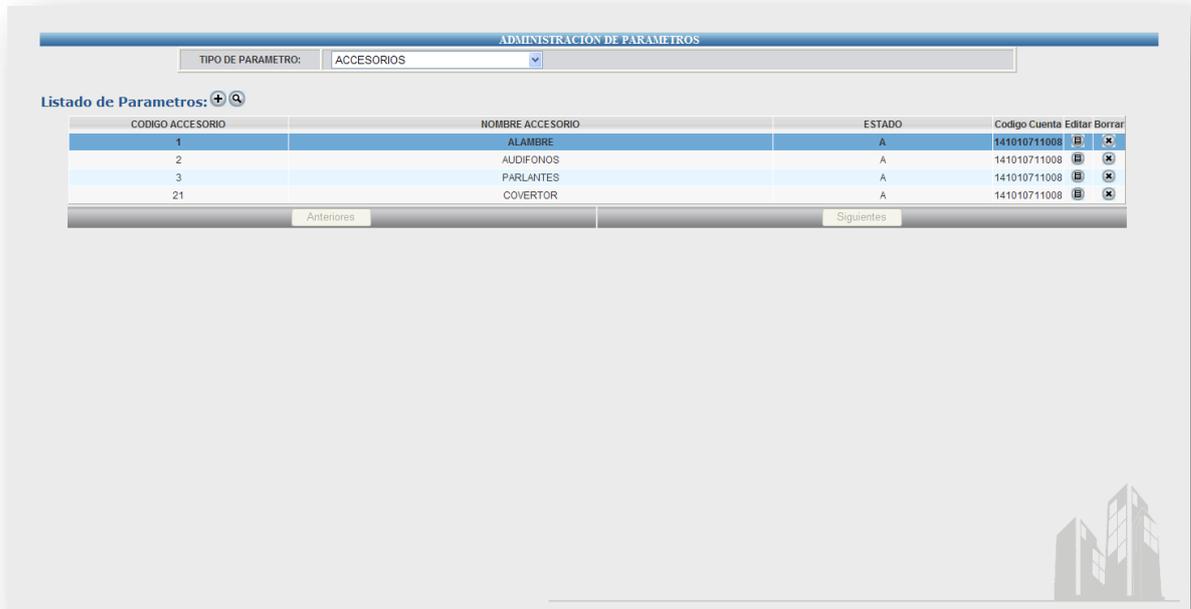
The screenshot shows the 'ADMINISTRACIÓN DE PARAMETROS' interface with 'TIPO DE PARAMETRO:' set to 'MOTIVO MOVIMIENTO'. Below is the 'Listado de Parametros:' table with columns: CODIGO MOTIVO, NOMBRE MOTIVO, ESTADO, and COD. TIPO MOV. Each row has 'Editar' and 'Borrar' icons. Navigation buttons 'Anteriores' and 'Siguientes' are at the bottom.

CODIGO MOTIVO	NOMBRE MOTIVO	ESTADO	COD. TIPO MOV.
4	TRANSFERENCIA	A	21
21	TRASPASO	A	21
41	DAÑADO	A	23
42	MAL ESTADO	A	23
61	CAMBIO CUSTODIO	A	24
62	DESPIDO	A	24

Al igual que los anteriores parámetros el parámetro de Motivo se puede borrar y modificar, al momento de modificar se mostrara en pantalla los siguientes atributos.

1.4.2.9. Accesorios

Al escoger la opción de Accesorios automáticamente cambiara los datos que se mostraron en pantalla por los datos que pertenecen al bien.



Al igual que los anteriores parámetros el parámetro de Accesorios se puede borrar y modificar, al momento de modificar se mostrara en pantalla los siguientes atributos.

Nombre	Descripción
Cuenta	Catalogo de Tipo de Movimiento.
Nombre del Accesorio	Nombre del Motivo.
Fecha	Fecha de Creación.
Estado	Estado del Motivo.
Modelo	Modelo del Accesorio
Marca	Marca del Accesorio
Serie	Serie del Accesorio
Otras	Alguna característica adicional para el Accesorio.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: ACCESORIOS

Ingreso de Parámetros

Código: 1

Cuenta: 141010711008 COMPUTADOR

Nombre: ALAMBRE *Estado: ACTIVO

Fecha: 05/03/2009 DD/MM/YYYY

Modelo: SONNY

Marca: DF

Serie: we

Otras: RWER

Aceptar Cancelar

Listado de Parámetros:

CODIGO ACCESORIO	NOMBRE ACCESORIO	ESTADO	Código Cuenta
1	ALAMBRE	A	141010711008
2	AUDIFONOS	A	141010711008
3	PARLANTES	A	141010711008
21	COVERTOR	A	141010711008

Anteriores Sigüentes

Si se desea ingresar un nuevo Accesorio, se mostrara en pantalla los atributos vacios los cuales deberán ser llenados por el usuario.

ADMINISTRACIÓN DE PARAMETROS

TIPO DE PARAMETRO: ACCESORIOS

Ingreso de Parámetros

Cuenta:

Nombre: *Estado: ACTIVO

Fecha: DD/MM/YYYY

Modelo:

Marca:

Serie:

Otras:

Aceptar Cancelar

Listado de Parámetros:

CODIGO ACCESORIO	NOMBRE ACCESORIO	ESTADO	Código Cuenta
1	ALAMBRE	A	141010711008
2	AUDIFONOS	A	141010711008
3	PARLANTES	A	141010711008
21	COVERTOR	A	141010711008

Anteriores Sigüentes

1.5. Reportes

1.5.1. Detalle del Bien

Para el primer reporte que se presentara en pantalla es el de detalle del Bien, el cual mostrara todas las características de cualquier bien que el usuario quiera consultar.

El usuario deberá llenar cualquier atributo que el desee para mostrar el reporte.

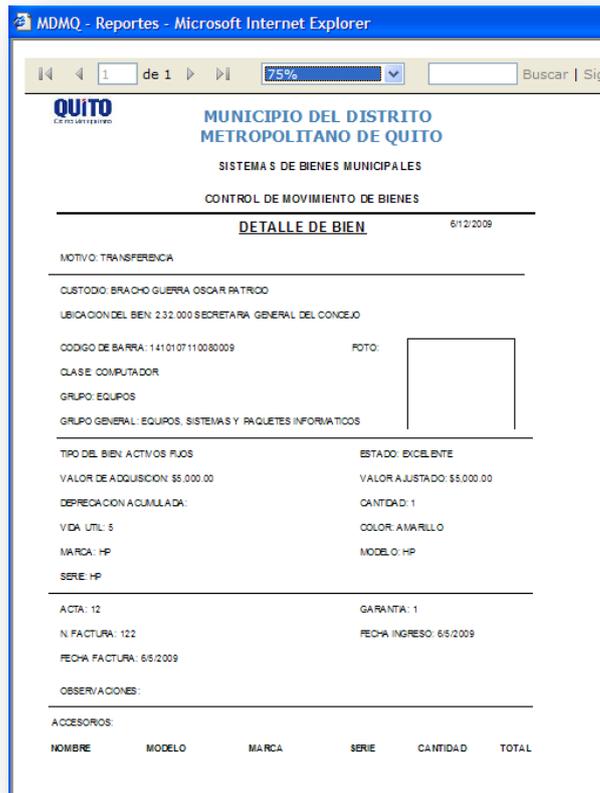
Nombre	Descripción
Código Bien	Código propio de cada Bien.
Código de Barra Actual	Código de Barra generado.
Código de Barra Anterior	Código de Barra del sistema anterior.
Código NIM	Código que otorga el MDMQ al Bien.

REPORTES DE DETALLE DE BIEN

Codigo Bien:	<input type="text"/>
Codigo de Barras Actual:	<input type="text"/>
Codigo Barras Anterior:	<input type="text"/>
Codigo NIM:	<input type="text"/>



Si el usuario desea ver el Bien deberá aplastar el botón de ver reporte el cual mostrara el reporte generado en pantalla.



1.5.2. Grupo de Bienes

El reporte que se presentara en pantalla es el de grupo del Bien, el cual mostrara todas las características de los bienes que el usuario quiera consultar. El usuario deberá llenar cualquier atributo que el desee para mostrar el reporte.

Nombre	Descripción
Código Bien	Código propio de cada Bien.
Código de Barra Actual	Código de Barra generado.
Código de Barra Anterior	Código de Barra del sistema anterior.
Código NIM	Código que otorga el MDMQ al Bien.

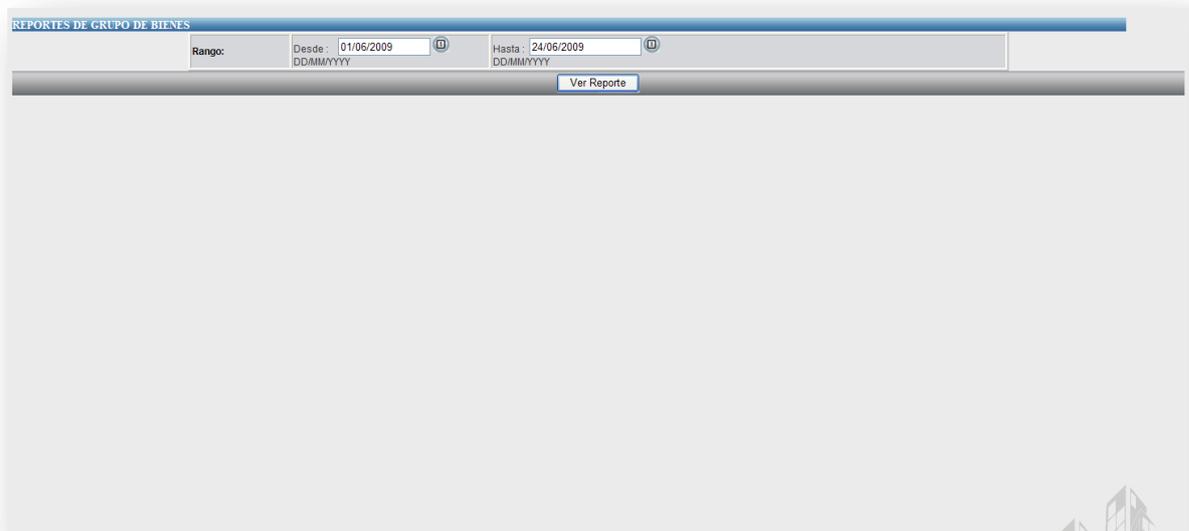
Si el usuario desea ver el Bien deberá aplastar el botón de ver reporte el cual mostrara el reporte generado en pantalla.

1.5.3. Depreciación de Bienes

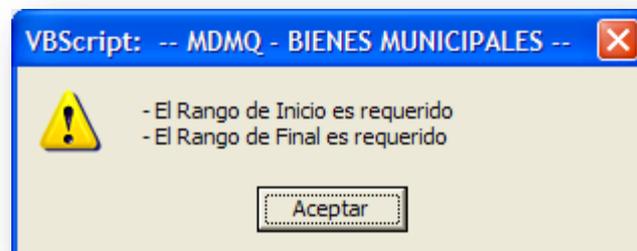
El reporte que se presentara en pantalla es el de depreciación de Bienes, el cual mostrara el detalle de depreciación del Bien, a partir de un rango dado.

El usuario deberá llenar cualquier atributo que el desee para mostrar el reporte.

Nombre	Descripción
Desde	Empiezo de Rango de Fecha a Consultar.
Hasta	Final de Rango de Fecha a Consultar.



Si el usuario no llena los datos requeridos el sistema mostrara mensajes de advertencia al usuario.



Si el usuario desea ver el Bien deberá aplastar el botón de ver reporte el cual mostrara el reporte generado en pantalla.

MDMQ - Reportes - Microsoft Internet Explorer

1 de 1 100% Buscar | Siguiete Seleccionar un formato Exporta

QUITO
Distrito Metropolitano

MUNICIPIO DEL DISTRITO METROPOLITANO DE QUITO

SISTEMAS DE BIENES MUNICIPALES

REPORTES DE BIENES

REPORTE VALOR EN LIBROS

NOMBRE BIEN	VALOR	VALOR RES.	DEP ANUAL	DEP ACUMULADA	DEPRECIADOS
COMPUTADOR	\$7,500.00	\$750.00		\$1,350.00	2
TOTALES:	\$7,500.00	\$750.00		\$1,350.00	2

Realizado por: installuser installuser Pág : 1/1