



**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN.**

**“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE  
CONTROL DEL CATASTRO TURÍSTICO PARA LA EMPRESA  
PÚBLICA METROPOLITANA DE GESTIÓN DEL DESTINO  
TURÍSTICO DE LA CIUDAD DE QUITO”**

**Previo a la obtención del Título:**

**INGENIERO EN SISTEMAS E INFORMÁTICA**

**Por: MARCO ROBERTO SALAZAR CEPEDA  
DANIEL ESTEBAN VALLEJO PRIETO**

**Sangolquí, Julio del 2012**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue realizado en su totalidad por los Sres. SALAZAR CEPEDA MARCO ROBERTO y VALLEJO PRIETO DANIEL ESTEBAN, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí, julio del 2012

---

Sr. Ing. Andrés de la Torre

## **DEDICATORIA**

Dedico esta Tesis a Dios, quien además de ser nuestro creador y redentor nunca me ha abandonado y siempre puedo sentir su presencia y bendición. Solo en Él y por Él se pueden hacer bien las cosas.

Quiero dedicar este trabajo además a mis padres, Marco Salazar y Sara Cepeda, cuya guía y enseñanza ha sido fundamental para llegar a ser la persona que ahora creo ser. Mil gracias Padre querido, por siempre estar incentivando en mí la búsqueda del conocimiento y Madre amada por tus cuidados y bendiciones.

Una dedicatoria muy especial para mi esposa Paola, quien siempre ha sabido demostrarme su amor y su apoyo. Gracias por escucharme, alentarme, aconsejarme y por estar a mi lado, sin ti no lo habría podido lograr.

**Marco**

## **DEDICATORIA**

Quiero dedicar esta Tesis a Dios, por ser el responsable de los grandes acontecimientos en mi vida.

Dedico principalmente este proyecto a mi Familia, a mis padres y hermanos que han sido y serán siempre mi gran apoyo en todas las etapas de mi vida.

También quiero dedicar esta tesis a la Chelita, a mi abuelita que siempre estaba pendiente de mi graduación, “Quiero que sepa que el Danisito Lindo ya se graduó”. Quiero incluir a esta dedicatoria a mi novia Carolina que fue un gran aporte para la culminación de esta etapa de mi vida.

**Daniel**

## **AGRADECIMIENTOS**

Queremos agradecer a DIOS en primer lugar por habernos permitido culminar este proyecto de vida.

Un agradecimiento muy especial a nuestro director, el Ing. Andrés de la Torre, por su apoyo incondicional para la realización de este trabajo. Su guía fue fundamental para lograrlo.

También deseamos agradecer a nuestro codirector de tesis, el Ing. Mauricio Campaña, quien estuvo con nosotros durante todo el proceso, exigiendo lo mejor de cada uno para lograr un producto de calidad.

No podemos dejar de mencionar y agradecer a nuestras familias, amigos y compañeros de trabajo, quienes aportaron cada uno con su grano de arena. Gracias por siempre confiar en nosotros y nunca dudar que lo lograríamos.

Finalmente, queremos reconocer nuestra gratitud a todo el personal de la Empresa Pública Metropolitana de Gestión del Destino Turístico de Quito, y en su representación a la Lcda. Luz Elena Coloma, directora de dicha entidad, por abrirnos las puertas y brindarnos todas las facilidades para la realización de este proyecto.

**Marco Roberto Salazar Cepeda**

**Daniel Esteban Vallejo Prieto**

# ÍNDICE DE CONTENIDOS

|  |           |
|--|-----------|
| <b>RESÚMEN .....</b>                                       | <b>1</b>  |
| <b>CAPÍTULO 1.....</b>                                     | <b>3</b>  |
| 1.1 TEMA .....   | 3         |
| 1.2 INTRODUCCIÓN .....                                     | 3         |
| 1.3 JUSTIFICACIÓN .....                                    | 4         |
| 1.4 OBJETIVOS .....  | 5         |
| 1.4.1 <i>Objetivo General</i> .....                        | 5         |
| 1.4.2 <i>Objetivos Específicos</i> .....                   | 6         |
| 1.5 ALCANCE .....  | 6         |
| 1.6 METODOLOGÍA .....                                      | 7         |
| 1.7 FACTIBILIDAD.....                                      | 8         |
| 1.7.1 <i>Factibilidad Técnica</i> .....                    | 8         |
| 1.7.2 <i>Factibilidad Económica</i> .....                  | 9         |
| 1.7.3 <i>Factibilidad Operativa</i> .....                  | 9         |
| <b>CAPÍTULO 2.....</b>                                     | <b>10</b> |
| 2.1 CONCEPTOS DE APLICACIONES DISTRIBUIDAS (N CAPAS) ..... | 10        |
| 2.1.1 <i>Introducción</i> .....                            | 10        |
| 2.1.2 <i>Concepto</i> .....                                | 10        |
| 2.1.3 <i>Aplicaciones Cliente – Servidor</i> .....         | 11        |
| 2.1.4 <i>Aplicaciones en n – Capas</i> .....               | 12        |
| 2.1.5 <i>Arquitectura de aplicaciones</i> .....            | 14        |
| 2.1.5.1 <i>Capa Servidor</i> .....                         | 14        |
| 2.1.5.2 <i>Capa Negocio</i> .....                          | 14        |
| 2.1.5.3 <i>Capa Presentación</i> .....                     | 14        |
| 2.2 PATRONES DE DISEÑO .....                               | 16        |
| 2.2.1 <i>Introducción</i> .....                            | 16        |

|           |   |    |
|-----------|---|----|
| 2.2.2     | <i>Historia</i> .....   | 16 |
| 2.2.3     | <i>Concepto</i> .....   | 18 |
| 2.2.3.1   | <i>SINGLETON</i> .....  | 19 |
| 2.2.3.2   | <i>FACTORY METHOD</i> .....   | 19 |
| 2.2.3.3   | <i>ADAPTER</i> .....  | 19 |
| 2.2.3.4   | <i>COMPOSITE</i> .....  | 20 |
| 2.2.3.5   | <i>FACADE</i> .....   | 20 |
| 2.2.3.6   | <i>OBSERVER</i> .....   | 21 |
| 2.2.3.7   | <i>STRATEGY</i> .....   | 21 |
| 2.2.3.8   | <i>ITERATOR</i> .....   | 22 |
| 2.3       | SELECCIÓN DE LA METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE .....          | 22 |
| 2.3.1     | <i>Introducción</i> .....   | 22 |
| 2.3.2     | <i>Comparación entre las Metodologías de Desarrollo de Software</i> ..... | 23 |
| 2.4       | PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE .....                    | 26 |
| 2.4.1     | <i>Introducción</i> .....   | 26 |
| 2.4.1.1   | <i>Dirigido por los Casos de Uso</i> .....                                | 27 |
| 2.4.1.2   | <i>Iterativo e Incremental</i> .....                                      | 27 |
| 2.4.1.3   | <i>Centrado en la Arquitectura</i> .....                                  | 28 |
| 2.4.2     | <i>Ciclo de vida del RUP</i> .....  | 28 |
| 2.4.2.1   | <i>Concepción, Inicio o Estudio de oportunidad</i> .....                  | 30 |
| 2.4.2.2   | <i>Elaboración</i> .....  | 30 |
| 2.4.2.3   | <i>Construcción</i> .....   | 30 |
| 2.4.2.4   | <i>Transición</i> .....   | 30 |
| 2.4.3     | <i>Flujos de Trabajo del RUP</i> .....                                    | 32 |
| 2.4.3.1   | <i>Flujos de Trabajo Fundamentales</i> .....                              | 32 |
| 2.4.3.1.1 | <i>Modelado del negocio</i> .....   | 32 |
| 2.4.3.1.2 | <i>Requerimientos</i> .....   | 32 |
| 2.4.3.1.3 | <i>Análisis y diseño</i> .....  | 32 |

|           |  |    |
|-----------|--|----|
| 2.4.3.1.4 | Implementación .....   | 33 |
| 2.4.3.1.5 | Pruebas .....  | 33 |
| 2.4.3.1.6 | Despliegue.....  | 33 |
| 2.4.3.2   | <i>Flujos de Trabajo de Soporte</i> .....                          | 34 |
| 2.4.3.2.1 | Gestión y configuración de cambios .....                           | 34 |
| 2.4.3.2.2 | Gestión del proyecto.....  | 34 |
| 2.4.3.2.3 | Entorno .....  | 34 |
| 2.5       | CONCEPTOS EN TORNO A LA METODOLOGÍA UML .....                      | 35 |
| 2.5.1     | <i>Introducción</i> .....  | 35 |
| 2.5.2     | <i>Objetivos</i> .....   | 36 |
| 2.5.3     | <i>Diagramas Utilizados</i> .....                                  | 37 |
| 2.5.3.1   | <i>Diagrama de Clases y Objetos</i> .....                          | 38 |
| 2.5.3.2   | <i>Diagrama de Casos de Uso</i> .....                              | 39 |
| 2.5.3.3   | <i>Diagrama de Interacción</i> .....                               | 39 |
| 2.5.3.3.1 | Diagrama de Secuencia .....  | 39 |
| 2.5.3.3.2 | Diagrama de Colaboración .....                                     | 40 |
| 2.5.3.4   | <i>Diagrama de Estados</i> .....                                   | 41 |
| 2.5.3.5   | <i>Diagrama de componentes</i> .....                               | 41 |
| 2.5.3.6   | <i>Diagrama de despliegue</i> .....                                | 42 |
| 2.6       | DESCRIPCIÓN DE LAS HERRAMIENTAS DE DESARROLLO .....                | 42 |
| 2.6.1     | <i>FRONT-END</i> .....   | 42 |
| 2.6.1.1   | Definición Herramientas de desarrollo .....                        | 42 |
| 2.6.1.2   | Comparación de lenguajes de Programación C# y Java (Eclipse) ..... | 43 |
| 2.6.1.3   | Selección de la Herramienta de Desarrollo .....                    | 44 |
| 2.6.1.4   | Comparación Herramientas de Desarrollo .....                       | 45 |
| 2.6.1.5   | Visual Studio .NET.....  | 46 |
| 2.6.1.5.1 | Características de C#.....   | 46 |
| 2.6.2     | <i>BACK-END</i> .....  | 47 |
| 2.6.2.1   | Selección del Motor de Base de Datos.....                          | 47 |



|                                    |   |           |
|------------------------------------|---|-----------|
| 2.6.2.2                            | Comparación Motores de Base de Datos .....          | 47        |
| 2.6.2.3                            | Microsoft SQL Server .....                          | 48        |
| <b>CAPÍTULO 3</b>                  | .....   | <b>50</b> |
| ANÁLISIS Y DISEÑO DE LA APLICACIÓN | .....   | 50        |
| 3.1                                | ESPECIFICACIÓN DE REQUERIMIENTOS.....               | 50        |
| 3.1.1                              | <i>Introducción</i> .....                           | 50        |
| 3.1.1.1                            | <i>Propósito</i> .....                              | 50        |
| 3.1.1.2                            | <i>Ámbito del Sistema</i> .....                     | 50        |
| 3.1.1.3                            | <i>Definiciones, Acrónimos y Abreviaturas</i> ..... | 52        |
| 3.1.1.3.1                          | Definiciones.....                                   | 52        |
| 3.1.1.3.2                          | Acrónimos y Abreviaturas .....                      | 53        |
| 3.1.1.4                            | <i>Referencias</i> .....                            | 54        |
| 3.1.1.5                            | <i>Visión general del documento</i> .....           | 55        |
| 3.1.2                              | <i>Descripción General</i> .....                    | 55        |
| 3.1.2.1                            | <i>Perspectiva del producto</i> .....               | 55        |
| 3.1.2.2                            | <i>Funciones del sistema</i> .....                  | 56        |
| 3.1.2.3                            | <i>Características de los Usuarios</i> .....        | 57        |
| 3.1.2.4                            | <i>Restricciones</i> .....                          | 58        |
| 3.1.2.4.1                          | Políticas de la empresa .....                       | 58        |
| 3.1.2.4.2                          | Funciones de auditoría.....                         | 59        |
| 3.1.2.4.3                          | Consideraciones acerca de la seguridad.....         | 59        |
| 3.1.2.5                            | <i>Suposiciones y dependencias</i> .....            | 59        |
| 3.1.3                              | <i>Requisitos Específicos</i> .....                 | 59        |
| 3.1.3.1                            | <i>Interfaces externas</i> .....                    | 59        |
| 3.1.3.1.1                          | Interfaz de Usuario .....                           | 60        |
| 3.1.3.1.2                          | Interfaces con otros sistemas.....                  | 60        |
| 3.1.3.1.3                          | Interfaces de comunicaciones .....                  | 60        |
| 3.1.3.2                            | <i>Funciones</i> .....                              | 61        |

|   |  |            |
|---|--|------------|
| 3.1.3.2.1                                       | Ingreso al Sistema.....                    | 61         |
| 3.1.3.2.2                                       | Mantenimiento del Catastro .....           | 63         |
| 3.1.3.2.3                                       | Mantenimiento de Recaudaciones .....       | 72         |
| 3.1.3.2.4                                       | Mantenimiento de Inspecciones .....        | 79         |
| 3.1.3.2.5                                       | Generación de Reportes.....                | 89         |
| 3.1.3.2.6                                       | Mantenimiento del Sistema .....            | 94         |
| 3.1.3.3   | Requisitos de Rendimiento .....            | 102        |
| 3.1.3.4   | Restricciones de Diseño .....              | 102        |
| 3.2   | DIAGRAMA ENTIDAD-RELACIÓN .....            | 103        |
| <b>CAPÍTULO 4.....</b>                          |  | <b>104</b> |
| DESARROLLO E IMPLANTACIÓN DE LA APLICACIÓN..... |  | 104        |
| 4.1   | DESARROLLO DEL SISTEMA.....                | 104        |
| 4.1.1   | <i>Componentes Desarrollados.....</i>      | <i>104</i> |
| 4.1.1.1   | Capa de Acceso a Datos .....               | 104        |
| 4.1.1.2   | Capa Transaccional .....                   | 105        |
| 4.1.1.3   | Capa de Negocio.....                       | 105        |
| 4.1.1.4   | Capa Útil.....                             | 105        |
| 4.1.1.5   | Capa de Entidades .....                    | 106        |
| 4.1.1.6   | Capa de Presentación de Usuario (Web)..... | 106        |
| 4.1.1.7   | Capa de Servicios Web.....                 | 106        |
| 4.2   | EJECUCIÓN DE PRUEBAS.....                  | 107        |
| 4.3   | IMPLANTACIÓN DEL SISTEMA Y TRANSICIÓN..... | 111        |
| 4.3.1   | <i>Implantación.....</i>                   | <i>112</i> |
| 4.3.2   | <i>Entrega del Sistema .....</i>           | <i>112</i> |
| 4.3.3   | <i>Manual de Usuario.....</i>              | <i>112</i> |
| <b>CAPÍTULO 5.....</b>                          |  | <b>113</b> |
| CONCLUSIONES Y RECOMENDACIONES.....             |  | 113        |
| 5.1   | CONCLUSIONES.....                          | 113        |
| 5.2   | RECOMENDACIONES.....                       | 114        |

|  |            |
|--|------------|
| <b>GLOSARIO DE TÉRMINOS</b> .....      | <b>116</b> |
| <b>BIBLIOGRAFÍA</b> .....              | <b>119</b> |
| REFERENCIAS MATERIAL IMPRESO .....     | 119        |
| REFERENCIAS MATERIAL ELECTRÓNICO ..... | 119        |
| <b>ANEXO A</b> .....                   | <b>123</b> |
| <b>ANEXO B</b> .....                   | <b>126</b> |
| <b>ANEXO C</b> .....                   | <b>127</b> |
| <b>ANEXO D</b> .....                   | <b>128</b> |
| <b>ANEXO E</b> .....                   | <b>129</b> |
| <b>ANEXO F</b> .....                   | <b>130</b> |
| <b>APÉNDICE</b> .....                  | <b>131</b> |

## LISTADO DE FIGURAS

|   |    |
|---|----|
| FIGURA N° 1: Aplicación Cliente - Servidor .....                            | 12 |
| FIGURA N° 2: Aplicación n - Capas .....                                     | 13 |
| FIGURA N° 3: Capa Presentación .....  | 15 |
| FIGURA N° 4: Ciclo de vida RUP.....   | 29 |
| FIGURA N° 5: Iteraciones en el RUP .....                                    | 31 |
| FIGURA N° 6: Diagrama de Clases y Objetos .....                             | 38 |
| FIGURA N° 7: Diagrama de Casos de Uso .....                                 | 39 |
| FIGURA N° 8: Diagrama de Secuencia .....                                    | 40 |
| FIGURA N° 9: Diagrama de Colaboración .....                                 | 40 |
| FIGURA N° 10: Diagrama de Estados .....                                     | 41 |
| FIGURA N° 11: Diagramas de Componentes .....                                | 41 |
| FIGURA N° 12: Diagrama de Despliegue.....                                   | 42 |
| FIGURA N° 13: Diagrama de Caso de Uso – Requisito 1 .....                   | 61 |
| FIGURA N° 14: Diagrama de Secuencia – Requisito 1 .....                     | 62 |
| FIGURA N° 15: Diagrama de Colaboración – Requisito 1.....                   | 62 |
| FIGURA N° 16: Diagrama de Caso de Uso – Requisito 2 .....                   | 64 |
| FIGURA N° 17: Diagrama de Secuencia – Requisito 2.....                      | 64 |
| FIGURA N° 18: Diagrama de Colaboración – Requisito 2.....                   | 65 |
| FIGURA N° 19: Diagrama de Caso de Uso – Requisito 3 .....                   | 66 |
| FIGURA N° 20: Diagrama de Secuencia – Requisito 3 .....                     | 67 |
| FIGURA N° 21: Diagrama de Colaboración – Requisito 3.....                   | 67 |
| FIGURA N° 22: Diagrama de Estados – Requisito 3.....                        | 68 |
| FIGURA N° 23: Diagrama de Caso de Uso – Requisito 4 .....                   | 69 |
| FIGURA N° 24: Diagrama de Secuencia (Flujo Normal) – Requisito 4.....       | 69 |
| FIGURA N° 25: Diagrama de Caso de Uso (Flujo Alternativo) – Requisito 4.... | 70 |
| FIGURA N° 26: Diagrama de Colaboración – Requisito 4.....                   | 70 |
| FIGURA N° 27: Diagrama de Caso de Uso – Requisito 5 .....                   | 71 |
| FIGURA N° 28: Diagrama de Secuencia – Requisito 5 .....                     | 71 |
| FIGURA N° 29: Diagrama de Colaboración – Requisito 5.....                   | 72 |
| FIGURA N° 30: Diagrama de Caso de Uso – Requisito 6 .....                   | 73 |
| FIGURA N° 31: Diagrama de Secuencia – Requisito 6.....                      | 74 |
| FIGURA N° 32: Diagrama de Colaboración – Requisito 6.....                   | 74 |
| FIGURA N° 33: Diagrama de Caso de Uso – Requisito 7 .....                   | 75 |
| FIGURA N° 34: Diagrama de Secuencia – Requisito 7 .....                     | 76 |
| FIGURA N° 35: Diagrama de Colaboración – Requisito 7.....                   | 76 |
| FIGURA N° 36: Diagrama de Caso de Uso – Requisito 8 .....                   | 77 |
| FIGURA N° 37: Diagrama de Secuencia – Requisito 8.....                      | 78 |
| FIGURA N° 38: Diagrama de Colaboración – Requisito 8.....                   | 79 |
| FIGURA N° 39: Diagrama de Caso de Uso – Requisito 9 .....                   | 80 |
| FIGURA N° 40: Diagrama de Secuencia – Requisito 9 .....                     | 81 |
| FIGURA N° 41: Diagrama de Colaboración – Requisito 9.....                   | 82 |
| FIGURA N° 42: Diagrama de Caso de Uso – Requisito 10 .....                  | 83 |
| FIGURA N° 43: Diagrama de Secuencia (Flujo Normal) – Requisito 10.....      | 83 |
| FIGURA N° 44: Diagrama de Secuencia (Flujo Alternativo) – Requisito 10..... | 84 |
| FIGURA N° 45: Diagrama de Colaboración – Requisito 10.....                  | 85 |
| FIGURA N° 46: Diagrama de Caso de Uso – Requisito 11 .....                  | 86 |

|   |     |
|---|-----|
| FIGURA N° 47: Diagrama de Secuencia – Requisito 11 .....                                    | 86  |
| FIGURA N° 48: Diagrama de Colaboración – Requisito 11 .....                                 | 87  |
| FIGURA N° 49: Diagrama de Caso de Uso – Requisito 12 .....                                  | 88  |
| FIGURA N° 50: Diagrama de Secuencia – Requisito 12 .....                                    | 88  |
| FIGURA N° 51: Diagrama de Colaboración – Requisito 12 .....                                 | 88  |
| FIGURA N° 52: Diagrama de Caso de Uso – Requisito 13 .....                                  | 89  |
| FIGURA N° 53: Diagrama de Secuencia – Requisito 13 .....                                    | 90  |
| FIGURA N° 54: Diagrama de Colaboración – Requisito 13 .....                                 | 90  |
| FIGURA N° 55: Diagrama de Caso de Uso – Requisito 14 .....                                  | 91  |
| FIGURA N° 56: Diagrama de Secuencia – Requisito 14 .....                                    | 91  |
| FIGURA N° 57: Diagrama de Colaboración – Requisito 14 .....                                 | 92  |
| FIGURA N° 58: Diagrama de Caso de Uso – Requisito 15 .....                                  | 93  |
| FIGURA N° 59: Diagrama de Secuencia – Requisito 15 .....                                    | 93  |
| FIGURA N° 60: Diagrama de Colaboración – Requisito 15 .....                                 | 93  |
| FIGURA N° 61: Diagrama de Caso de Uso – Requisito 16 .....                                  | 94  |
| FIGURA N° 62: Diagrama de Secuencia – Requisito 16 .....                                    | 95  |
| FIGURA N° 63: Diagrama de Colaboración – Requisito 16 .....                                 | 95  |
| FIGURA N° 64: Diagrama de Caso de Uso – Requisito 17 .....                                  | 96  |
| FIGURA N° 65: Diagrama de Secuencia – Requisito 17 .....                                    | 97  |
| FIGURA N° 66: Diagrama de Colaboración – Requisito 17 .....                                 | 97  |
| FIGURA N° 67: Diagrama de Caso de Uso – Requisito 18 .....                                  | 98  |
| FIGURA N° 68: Diagrama de Secuencia – Requisito 18 .....                                    | 99  |
| FIGURA N° 69: Diagrama de Caso de Uso – Requisito 18 .....                                  | 99  |
| FIGURA N° 70: Diagrama de Caso de Uso – Requisito 19 .....                                  | 100 |
| FIGURA N° 71: Diagrama de Secuencia – Requisito 19 .....                                    | 101 |
| FIGURA N° 72: Diagrama de Colaboración – Requisito 19 .....                                 | 101 |
| FIGURA N° 73: Diagrama de Entidad-Relación .....  | 103 |
| FIGURA N° 74: Diagrama de Componentes .....   | 107 |
| FIGURA N° 73: Tiempos esperados por transacción.....  | 109 |
| FIGURA N° 75: Porcentaje de transacciones Aceptables, A considerar y No<br>Aceptables ..... | 110 |
| FIGURA N° 75: Porcentaje de éxito de transacciones evaluadas .....                          | 111 |
| FIGURA N° 75: Diagrama de Despliegue .....  | 112 |

## LISTADO DE TABLAS

|  |     |
|--|-----|
| TABLA N° 1: Presupuesto del Proyecto .....   | 9   |
| TABLA N° 2: Patrones de Diseño.....  | 18  |
| TABLA N° 3: Criterios de ponderación para evaluación de metodologías de desarrollo de software ..... | 24  |
| TABLA N° 4: Evaluación de metodologías de desarrollo de software.....                                | 25  |
| TABLA N° 5: Presupuesto del Proyecto .....   | 45  |
| TABLA N° 6: Presupuesto del Proyecto .....   | 48  |
| TABLA N° 7: Definiciones de la Especificación de Requerimientos.....                                 | 53  |
| TABLA N° 8: Acrónimos de la especificación de requerimientos .....                                   | 54  |
| TABLA N° 9: Formularios de Inspección de acuerdo a cada Actividad.....                               | 79  |
| TABLA N° 10: Valores esperados para pruebas de stress.....   | 108 |
| TABLA N° 11: Valores obtenidos en ejecución de pruebas de stress.....                                | 108 |

## **RESÚMEN**

En la actualidad, el contar con un sistema informático, que permita ejercer un control efectivo sobre los establecimientos turísticos del Distrito Metropolitano de Quito y a la vez obtener reportes actualizados para la toma de decisiones, es una necesidad inherente a las funciones de la Empresa Pública Metropolitana de Gestión del Destino Turístico de Quito. Es por ello que el implementar un sistema que ejecute dichos procesos de manera eficaz ayuda en gran medida a conseguir estos objetivos.

Con el apoyo de los Departamentos de Sistemas y de Control de Calidad, se establecieron las pautas necesarias para iniciar el proyecto. Se analizó la metodología a utilizar, optándose por el RUP (Proceso Unificado de Rational) ya que es el estándar de desarrollo en dicho organismo, a la vez que proporcionaba las mejores condiciones para el producto a desarrollar.

El proyecto por su naturaleza web (dado que se requiere el acceso al mismo desde diversos puntos geográficos), se desarrolló completamente usando el entorno de desarrollo Visual Studio .NET (ASP.NET con lenguaje C#) y el motor de Base de Datos seleccionado fue SQL Server 2005, debido a que la empresa posee las licencias de uso de software para dichos aplicativos.

El proyecto se inició con la definición de los requerimientos funcionales y no funcionales de la empresa, siguiendo las pautas proporcionadas para este paso por la metodología IEEE 830.

Al especificar de manera clara todas las necesidades, se realizó el análisis y diseño del aplicativo, siguiendo los puntos establecidos en la metodología RUP. Una vez desarrollado el sistema, se realizaron pruebas unitarias, de integración y de estrés para verificar que el mismo cumpliera con los requisitos previamente definidos.



# CAPÍTULO 1

## 1.1 TEMA

Análisis, Diseño e Implementación de un Sistema de Control del Catastro Turístico para la Empresa Pública Metropolitana de Gestión del Destino Turístico de la ciudad de Quito.

## 1.2 INTRODUCCIÓN

La Empresa Pública Metropolitana de Gestión de Destino Turístico (EPMGDT), es una entidad de derecho público, cuya misión es desarrollar las actividades turísticas en el Distrito Metropolitano de Quito, tales como la generación y reconversión de nuevos productos turísticos, adecuación de la infraestructura turística, capacitación, formación y profesionalización en el sector turístico, entre otras.<sup>1</sup>

La EPMGDT es la entidad municipal encargada de realizar la promoción turística de la ciudad de Quito a nivel nacional e internacional. La empresa cuenta actualmente con 30 empleados de planta, los mismos que se dividen en las siguientes funciones: dirección general, gerentes de promoción internacional, promoción nacional, producto turístico, catastro turístico, secretarias, inspectores, diseñadores, personal de atención al público, soporte en sitio, bodegueros y choferes.

---

<sup>1</sup> Empresa Pública Metropolitana de Gestión de Destino Turístico - Acerca de Nosotros, [http://www.quito-turismo.gob.ec/index.php?option=com\\_content&task=view&id=25&Itemid=57](http://www.quito-turismo.gob.ec/index.php?option=com_content&task=view&id=25&Itemid=57)

La empresa cuenta con un sistema obsoleto de control del catastro, el mismo que no incluye un módulo de Inspecciones ni tampoco de Reportes Parametrizables, por lo que durante todos estos años, este proceso se lo ha venido realizando de forma manual, lo que conlleva una serie de problemas como son: Inexactitud, el mismo hecho de que se lo realice manualmente origina errores humanos; Inseguridad, como se sabe, un archivo de Excel no es tan seguro ni tan confiable como un DBMS, por lo que la información podría ser fácilmente cambiada; Tiempo y Recursos, ya que para elaborar los reportes solicitados por la Dirección, se requieren de alrededor de 2 días para una persona poder realizarlos.

Todo esto sin mencionar que el sistema actual de control de catastros posee algunas falencias en lo que a registro de información del establecimiento se refiere y sobretodo no contempla la funcionalidad del control de emisiones y pagos de las tasas turísticas a los establecimientos.

Por otro lado, es innegable que el vertiginoso avance de la tecnología ha motivado que las organizaciones en general, sientan la necesidad de implementar soluciones automatizadas, con el fin de incrementar su productividad, así como también de garantizar la seguridad de su información interna.

### **1.3 JUSTIFICACIÓN**

Para su funcionamiento, la Empresa Pública Metropolitana de Gestión de Destino Turístico requiere de un sistema informático que le permita administrar de manera efectiva la información referente de los locales turísticos que operan dentro del Distrito Metropolitano de Quito (catastro) y por ende el cobro de las tasas de

turismo a dichos establecimientos, ya que su presupuesto de funcionamiento depende en gran medida de este rubro.

Es fundamental además que como institución la EPMGDT pueda llevar de manera efectiva el control y sobretodo la información actualizada acerca de las inspecciones realizadas a los establecimientos turísticos, de tal forma que se puedan garantizar además de un trato justo y de alta calidad, la seguridad y el bienestar de los turistas de la ciudad de Quito.

El estudio de herramientas de desarrollo actuales y su potencialidad de aplicación a problemas como el propuesto, contribuye a la mejor formación profesional de los egresados de la Facultad y del país.

En lo referente a la seguridad, EPMGDT requiere que se utilicen métodos por medio de los cuales se pueda garantizar la veracidad de la información a ser recopilada, hecho que no podría ser asegurado mediante un procesamiento manual.

Por último, la ejecución del presente proyecto ha sido solicitada por el personal responsable del área de Catastros de la EPMGDT, los mismos que están dispuestos a destinar los fondos económicos necesarios para el desarrollo e implantación del sistema de Control del Catastro Turístico.

## **1.4 OBJETIVOS**

### **1.4.1 Objetivo General**

Analizar, diseñar e implementar un Sistema de Control del Catastro Turístico para la Empresa Pública Metropolitana de Gestión de Destino Turístico de la ciudad de Quito, mediante el uso de una aplicación web, a través de la cual se podrá realizar

el ingreso confiable de datos, para de esta manera facilitar el acceso a la información, el procesamiento de la misma y la obtención de reportes, garantizando la toma de decisiones gerenciales de forma oportuna.

#### **1.4.2 Objetivos Específicos**

- Conocer y entender el proceso actual sobre el control de Catastro Turístico en la Empresa Pública Metropolitana de Gestión de Destino Turístico.
- Establecer los fundamentos metodológicos a emplear para el análisis, diseño y desarrollo de la aplicación.
- Definir y Especificar los requerimientos para el sistema de gestión de Catastro Turístico.
- Realizar el Análisis y Diseño de la aplicación en base a los requerimientos obtenidos.
- Desarrollar los módulos de registro de Establecimientos por actividad, Control de Inspecciones a los establecimientos del catastro, Control de Emisiones y Pagos por cada uno de los Establecimientos, Permisos de funcionamiento para cada establecimiento y generación de Reportes Parametrizables, en base a los requerimientos establecidos.
- Implementar el sistema y realizar las pruebas respectivas.

#### **1.5 ALCANCE**

La implementación de este proyecto contempla:

- Establecer los requerimientos para el sistema de gestión de Control del Catastro Turístico.

- Realizar el Análisis y Diseño de la aplicación para gestionar el Control del Catastro Turístico de la Empresa Pública Metropolitana, utilizando el Proceso Unificado de Desarrollo RUP y el Lenguaje de Modelamiento (UML), apoyados en una herramienta CASE.
- Desarrollar e Implementar el módulo de registro de Establecimientos por actividad, Control de Inspecciones a los establecimientos del catastro, Control de los Pagos, Emisiones y Bajas para cada uno de los Establecimientos, Permisos de funcionamiento por cada establecimiento y generación de Reportes Parametrizables, además de realizar las pruebas correspondientes.

El proyecto debe cumplir con ciertas restricciones impuestas por la empresa, entre las cuales tenemos:

- La codificación utilizada para clasificar a los establecimientos turísticos, debe respetarse y no podrá ser modificada por el equipo de desarrollo.
- La implementación de un log de auditoría que implique modificación de datos por parte de los usuarios en el sistema.
- El sistema a desarrollarse, no abarcará los procesos contables ni de facturación o de retenciones, que por Ley deban aplicarse, por cuanto el cobro efectivo a los establecimientos turísticos lo realiza en la actualidad el Departamento de Recaudaciones del Ilustre Municipio de Quito

## **1.6 METODOLOGÍA**

La Metodología a utilizarse durante todo el proceso de desarrollo de la aplicación estará basada en RUP. Dicha metodología de desarrollo de software además de

ser iterativa e incremental, está centrada en los modelos que permitan describir el problema de forma natural.

Además se empleará el Lenguaje Unificado de Modelamiento, el cual contempla diagramas, para realizar de mejor manera el Análisis y Diseño de la aplicación, entre los cuales se pueden mencionar: Diagramas de Casos de Uso, Diagramas de Secuencia, Diagramas de Colaboración, Diagramas de Estado y Diagramas de Paquetes.

Para el Análisis y Diseño de la aplicación, se requerirá de un esfuerzo de diagnóstico y evaluación tanto de la situación actual de la empresa, como de los requerimientos de la misma, así como de la mejor manera para implementar la nueva tecnología en los procesos involucrados.

Finalmente, se requerirá realizar pruebas exhaustivas del correcto funcionamiento del sistema, así como de la conectividad de los equipos de las distintas sucursales antes de implementar finalmente el sistema.

## **1.7 FACTIBILIDAD**

### **1.7.1 Factibilidad Técnica**

El proyecto requiere de un equipo de hardware dedicado para el funcionamiento específico de la aplicación, el cual será proporcionado por la Empresa. En cuanto a la realización del presente trabajo se cuenta con suficientes fuentes de información, cuyas referencias se presentan en la bibliografía. De igual manera, el compromiso del Director de Tesis permite disponer del asesoramiento necesario para concluir exitosamente con el proyecto.

### 1.7.2 Factibilidad Económica

A continuación se presenta el Presupuesto General para el desarrollo de la Tesis, tomando en cuenta diferentes aspectos como bibliografía, materiales de oficina, depreciación de equipos, diseñador gráfico, programadores, etc. El financiamiento estará a cargo de los tesisistas, por lo que el proyecto es factible económicamente.

| Concepto              | Costo/Semana (\$) | Semanas | Costo Total (\$) |
|-----------------------|-------------------|---------|------------------|
| <b>Ingresos</b>       |                   |         |                  |
| Aporte Tesisistas     |                   |         | 6530             |
| <b>Total Ingresos</b> |                   |         | <b>6530</b>      |
| <b>Egresos</b>        |                   |         |                  |
| Personal              |                   |         |                  |
| Programadores         | 200               | 28      | 5600             |
| Recursos              |                   |         |                  |
| Mat. Bibliográfico    |                   |         | 200              |
| Internet              | 10                | 28      | 280              |
| Mat. Oficina/Copias   |                   |         | 150              |
| Uso de equipos        |                   |         | 300              |
| <b>Total Egresos</b>  |                   |         | <b>6530</b>      |

TABLA N° 1: Presupuesto del Proyecto

### 1.7.3 Factibilidad Operativa

Los egresados tienen la capacidad suficiente para realizar el presente trabajo gracias a los conocimientos adquiridos en las materias del área de Desarrollo de Software cursados durante su carrera en la Facultad de Sistemas e Informática.

El proyecto cuenta con el apoyo y auspicio de la EPMGDT por lo cual se dispone de la colaboración del personal que posee el conocimiento técnico y la experiencia respecto de los procesos, lo cual será de utilidad para la elaboración del proyecto. Adicionalmente no existirá resistencia al cambio, por cuanto el sistema simplificará la carga de trabajo actual.

## CAPÍTULO 2

### GENERALIDADES DEL PROYECTO

#### 2.1 CONCEPTOS DE APLICACIONES DISTRIBUIDAS (N CAPAS)

##### 2.1.1 Introducción

En la actualidad, se evidencia una gran revolución tecnológica, existe un gran adelanto con respecto a las comunicaciones, cualquier tipo de información es entregada casi inmediatamente al usuario final, entre los cambios más relevantes, esta lo referente al soporte y canal de transmisión de la información.

Con el Internet y las tecnologías web, se ha conseguido que los usuarios se familiaricen con información hipermedia, incluyendo texto, imágenes, audio y vídeo. Todo ordenador conectado a la gran red constituye para todos una fuente fácil de entrada a la información y servicios.

Este hecho hace que las aplicaciones web hoy en día sean las más utilizadas al momento de implementar un sistema empresarial.

##### 2.1.2 Concepto

Una aplicación es distribuida, cuando contiene distintos componentes que son ejecutados en diferentes entornos, normalmente en distintas plataformas que se conectan a través de una red. Las aplicaciones distribuidas pueden clasificarse en **cliente-servidor** y en aplicaciones **n-capas**.



### **2.1.3 Aplicaciones Cliente – Servidor**

Las aplicaciones cliente-servidor, tienen dos procesos que se encargan de proporcionar a través del servidor, datos que son solicitados por el cliente, y a su vez el servidor de procesar los datos que son entregados por el cliente.

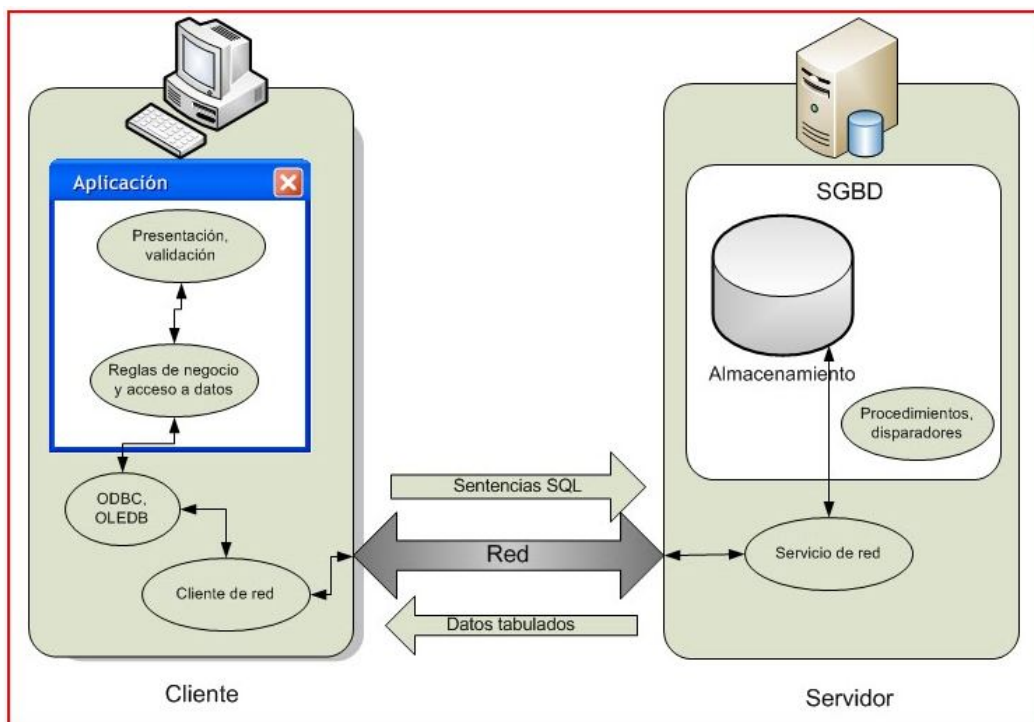
De estos dos procesos, el primero se encarga de entregar los datos que son solicitados, y de procesar los datos que le envían, este proceso es realizado en el servidor. El otro proceso que hace referencia al cliente, es ejecutado en la máquina del usuario final que maneja la aplicación, este a través de dicha aplicación cumple su función principal, esta es la solicitud de datos al servidor, los datos son presentados al usuario para que este haga sus cálculos o transacciones y después termina el proceso enviando al servidor los cambios.

Con este esquema de aplicación, generalmente en el servidor se realiza la gestión de base de datos (SGBD Sistema Gestor de Base de Datos). Este SGBD, en el caso de SQL Server u Oracle, del lado del servidor ejecuta un servicio de red que recibe todas las peticiones que van llegando en forma de sentencias SQL, las mismas que son transmitidas al verdadero proceso de la base de datos, es decir este se encarga de seleccionar los registros que le fueron indicados en cada petición, o en el caso de enviar a realizar una operación como puede ser una actualización de registros. Si esta petición generó registros, este resultado se transforma en datos tabulados, que son enviados a través de la red hacia el equipo que realizó la solicitud.

Un SGBD no consiste solo en ser un almacén de datos, este es una parte importante en los procesos de la aplicación distribuida.

Por otro lado, también están las estaciones de trabajo, en las que se encuentran las aplicaciones del cliente. Para que un cliente pueda realizar peticiones al servidor, debe contar con el cliente de red que corresponda al SGBD.

La función de un cliente no es más que permitir al usuario, solicitar datos a través de una interfaz y visualizarlos en pantalla, realizar operaciones con ellos y finalmente enviar las actualizaciones al servidor.



**FIGURA N° 1:** Aplicación Cliente - Servidor

Se puede concluir, que las aplicaciones cliente-servidor deben ser utilizadas únicamente en casos puntuales, en los que temas de rendimiento o simplicidad de aplicaciones son los factores determinantes.

#### 2.1.4 Aplicaciones en n – Capas

En una aplicación n-capas se encuentran un sin número de procesos, los mismos que estarán distribuidos en capas tanto lógicas como físicas. Estos procesos

pueden ser ejecutados en varios equipos, que no necesariamente utilizan una misma plataforma o sistema operativo.

Los equipos utilizados en una aplicación n-capas, cumplen un rol individual o especial, ya que cada uno tiene su propia configuración y a su vez cumplen con el papel que se les ha asignado dentro de la estructura de la aplicación, con esto se garantiza la optimización de recursos.

En una aplicación web, es posible encontrar un servidor contenedor de datos, en el cual su SGBD puede poseer varios procedimientos almacenados que estén acorde a la lógica de la base de datos, después es se puede hallar un segundo equipo que contenga los componentes necesarios en los que este inmersa la lógica o las reglas del negocio de la aplicación, entre estas reglas es factible encontrar la validación de seguridad o la recuperación de los datos. En un tercer equipo, en el que suele implementarse la aplicación web ASP.NET, servidor IIS. Finalmente está el navegador en el que se pueden incluir validaciones de usuario en las propias páginas.

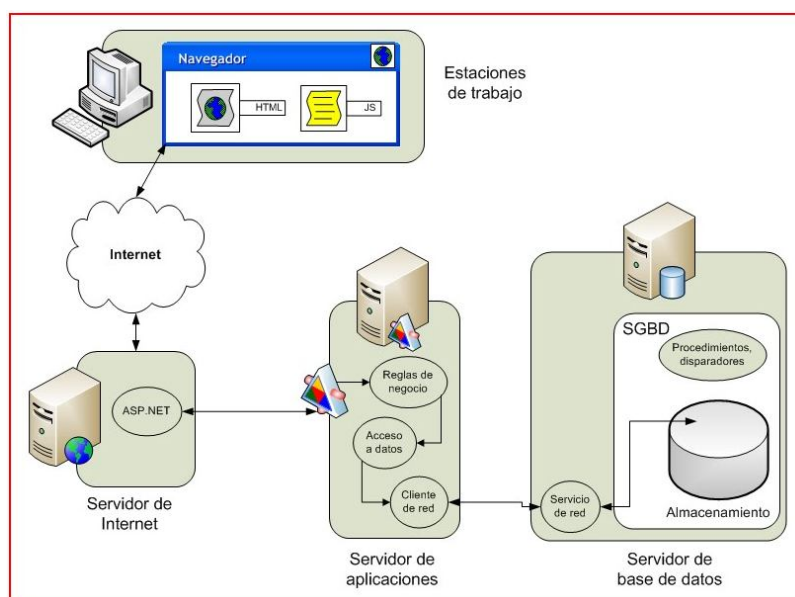


FIGURA N° 2: Aplicación n - Capas

Una aplicación distribuida en n-capas demanda de un mayor esfuerzo al momento de diseño, lo que da como resultado una pequeña pérdida de rendimiento frente a las aplicaciones cliente-servidor, la diferencia está en que al momento de realizar la implantación de una aplicación n-capas se solucionan muchos inconvenientes como la escalabilidad y la portabilidad, lo que hace que su uso sea necesario en sistemas muy grandes o complejos.

### **2.1.5 Arquitectura de aplicaciones**

Las aplicaciones distribuidas n-capas, según los distintos roles que deben manejar las mismas, serán clasificadas de la siguiente manera según su funcionalidad:

#### **2.1.5.1 Capa Servidor**

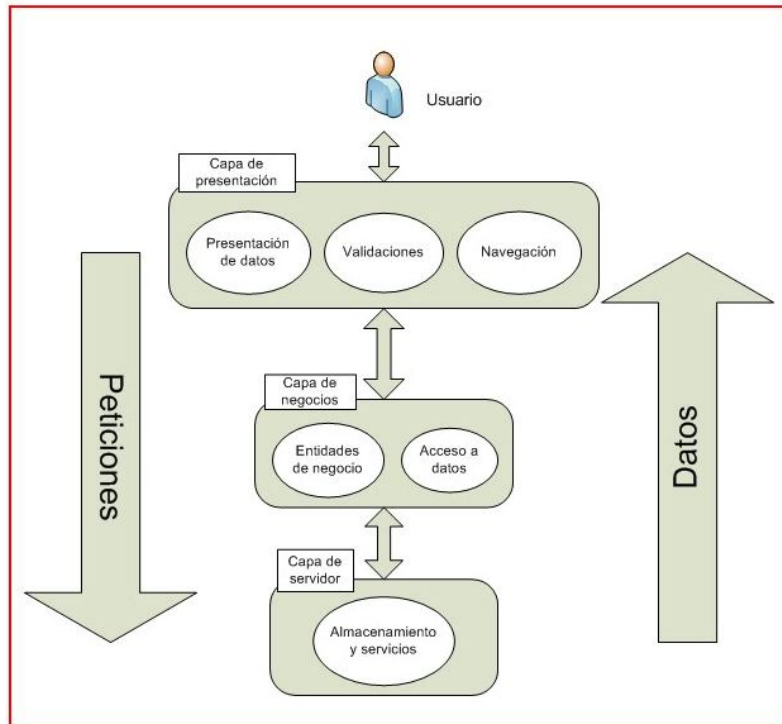
Esta capa se encarga de reunir a cada uno de los elementos que reciben las solicitudes de acceso a servicios del sistema o los que se asocian a solicitudes de datos, y de entregar la información solicitada.

#### **2.1.5.2 Capa Negocio**

En la capa del negocio residen todas las reglas para poder acceder a la información, en esta también se gestionan los procesos internos de la aplicación.

#### **2.1.5.3 Capa Presentación**

La capa de presentación está a cargo de interactuar con el usuario que accede a la aplicación.



**FIGURA N° 3:** Capa Presentación

Ahora se puede identificar como interactúa la capa de presentación con el usuario, realizando solicitudes de datos o simplemente realizando acciones en la aplicación. Las peticiones serán atendidas por la capa de negocio, la misma que está a cargo de gestionar la comunicación con la capa de servidor para realizar la tarea solicitada. De igual manera si la solicitud realizada, espera que se le entreguen datos, la capa del servidor se encargará de comunicarse con la capa del negocio, para que finalmente entregue la información a la capa de presentación.

## **2.2 PATRONES DE DISEÑO**

### **2.2.1 Introducción**

El diseño es un modelo de un sistema a ser desarrollado, el mismo se realiza con varios principios y técnicas, que permiten llegar a describir un sistema a un gran detalle como para que este sea implementado. Para crear un sistema los principios y reglas no son suficientes, cuando se habla de diseño se puede evidenciar que los buenos profesionales utilizan esquemas y estructuras de solución que se usan en reiteradas ocasiones en función de cómo se presente el problema. Con esto se quiere decir que es importante basarse en una estructura bien organizada, y no necesariamente tener una notable inteligencia al momento de desarrollar un sistema.

Estos esquemas y estructuras son definiciones reusables, y a su vez permiten no inventar cosas que ya existen, o cuyo uso se ha probado no es el más conveniente.

Un buen profesional pone en práctica el uso de estas herramientas, reutiliza un esquema de solución ante la presencia de problemas similares.

### **2.2.2 Historia**

El concepto de "patrón de diseño" que se tiene en Ingeniería del Software se ha tomado prestado de la arquitectura. En 1977 se publica el libro "A Pattern Language: Towns/Building/Construction", de Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King y Shlomo Angel, Oxford University Press. Contiene numerosos patrones con una notación específica de Alexander.

Alexander dice que “Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”<sup>2</sup>. Un patrón de diseño es una estructura o esquema de solución que se aplica cuando surge algún problema, cuando se aplica un patrón, no se lo debe hacer de forma mecánica, este requiere de adaptación al inconveniente. Por esta razón, Alexander indica que los numerosos usos de un patrón no se repiten dos veces de la misma forma.

No se había profundizado en cuanto a una definición clara para los patrones de diseño, por esta razón varios diseñadores se lanzaron a plasmar las ideas de Alexander y las asociaron a su propio concepto. El catálogo más famoso de patrones se encuentra en “Design Patterns: Elements of Reusable Object-Oriented Software”, de Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, 1995, Addison-Wesley, también conocido como el libro GOF (Gang-Of-Four).

Siguiendo el libro de GOF los patrones se clasifican según el propósito para el que han sido definidos:

- **Creacionales:**

Solucionan problemas de creación de instancias. Ayudan a encapsular y abstraer dicha creación.

---

<sup>2</sup> Patrones de diseño software, <http://www.proactiva-calidad.com/java/patrones/index.html>

- **Estructurales:**

Solucionan problemas de composición (agregación) de clases y objetos.

- **De Comportamiento:**

Soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

Según el ámbito se clasifican en patrones de clase y de objeto:

| Propósito<br>Ámbito | Creación   | Estructural  | Comportamiento   |
|---------------------|--|--|--|
| Clase               | ✓ <b>Factory Method</b>  | ✓ <b>Adapter</b>   | Interpreter<br>Template Method   |
| Objeto              | Abstract Factory<br>Builder<br>Prototype<br>✓ <b>Singleton</b> | ✓ <b>Adapter</b><br>Bridge<br>✓ <b>Composite</b><br>Decorator<br>✓ <b>Facade</b><br>Flyweight<br>Proxy | Chain of Responsibility<br>Command<br>✓ <b>Iterator</b><br>Mediator<br>Memento<br>✓ <b>Observer</b><br>State<br>✓ <b>Strategy</b><br>Visitor |

**TABLA N° 2:** Patrones de Diseño

### 2.2.3 Concepto

Se conoce como patrones de diseño a las soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

A continuación, una descripción de los patrones de diseño más conocidos y utilizados.



### **2.2.3.1 SINGLETON**

#### **Uso**

- Asegurar que una clase tiene una sola instancia y proporcionar un punto de acceso global a ella.

#### **Ventajas**

- Es necesario cuando hay clases que tienen que gestionar de manera centralizada un recurso.
- Una variable global no garantiza que sólo se instancie una vez.

### **2.2.3.2 FACTORY METHOD**

#### **Uso**

- Separar la clase que crea los objetos, de la jerarquía de objetos a instanciar.

#### **Ventajas**

- Centralización de la creación de objetos.
- Facilita la escalabilidad del sistema.
- El usuario se abstrae de la instancia a crear.

### **2.2.3.3 ADAPTER**

#### **Uso**

- Convertir la interfaz de una clase en otra interfaz esperada por los clientes.
- Permite que clases con interfaces incompatibles se comuniquen.

### **Ventajas**

- Se quiere utilizar una clase ya existente y su interfaz no se corresponde con la interfaz que se necesita.
- Se quiere envolver código no orientado a objeto con forma de clase.

### **2.2.3.4 COMPOSITE**

#### **Uso**

- Componer objetos en jerarquías todo-parte y permitir a los clientes tratar objetos simples y compuestos de manera uniforme.

#### **Ventajas**

- Permite tratamiento uniforme de objetos simples y complejos así como composiciones recursivas.
- Simplifica el código de los clientes, que sólo usan una interfaz.
- Facilita añadir nuevos componentes sin afectar a los clientes.

### **2.2.3.5 FACADE**

#### **Uso**

- El patrón FACADE simplifica el acceso a un conjunto de clases proporcionando una única clase que todos utilizan para comunicarse con dicho conjunto de clases.

#### **Ventajas**

- Los clientes no necesitan conocer las clases que hay tras la clase FACADE.

- Se pueden cambiar las clases “ocultadas” sin necesidad de cambiar los clientes. Sólo hay que realizar los cambios necesarios en FACADE.

### **2.2.3.6 OBSERVER**

#### **Uso**

- Definir una dependencia 1:n de forma que cuando el objeto 1 cambie su estado, los n objetos sean notificados y se actualicen automáticamente.

#### **Ventajas**

- En un toolkit de GUI, se puede separar los objetos de presentación (vistas) de los objetos de datos, de forma que se puedan tener varias vistas sincronizadas de los mismos datos (editor-subscriptor).

### **2.2.3.7 STRATEGY**

#### **Uso**

- Encapsular algoritmos relacionados en clases y hacerlos intercambiables.
- Se permite que la selección del algoritmo se haga según el objeto que se trate.

#### **Ventajas**

- Se permite cambiar el algoritmo dinámicamente.

- Se eliminan sentencias condicionales para seleccionar el algoritmo deseado.

### **2.2.3.8 ITERATOR**

#### **Uso**

- Proporcionar una forma de acceder a los elementos de una colección de objetos de manera secuencial sin revelar su representación interna.
- Define una interfaz que declara métodos para acceder secuencialmente a la colección.

#### **Ventajas**

- La clase que accede a la colección solamente a través de dicha interfaz permanece independiente de la clase que implementa la interfaz.

## **2.3 SELECCIÓN DE LA METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE**

### **2.3.1 Introducción**

Las Metodologías para el Desarrollo de Software definen las pautas, modelos, procesos y técnicas a seguir para la realización exitosa de un proyecto de desarrollo de software. Dichas metodologías son utilizadas para estructurar, planear y controlar el proceso de desarrollo de aplicaciones y surgen ante la necesidad de garantizar la calidad de un producto de software. Es por ello que se podría argumentar que la ausencia del uso de una metodología en el desarrollo de

un proyecto de software puede llegar a garantizar también la ausencia de calidad o el fracaso de este.

Las metodologías para el desarrollo de software, pretenden crear un marco de trabajo homogéneo y guiar a los programadores al crear un nuevo aplicativo. En la actualidad, existe una gran variedad de metodologías para la creación del software, las cuales se podrían clasificar en dos grupos importantes:

- Metodologías Tradicionales, se conoce así a las metodologías que poseen una orientación al control de los procesos, el establecimiento riguroso de las actividades a desarrollar, así como de las herramientas a utilizar y notaciones que se usarán. Entre las principales metodologías tradicionales existen RUP, MSF y CMMI.
- Metodologías Ligeras o Agiles, son aquellas cuya orientación principal es la interacción con el cliente y el desarrollo incremental del software, elaborando versiones parcialmente funcionales del sistema en intervalos cortos de tiempo, para que al cliente pueda evaluarlos y sugerir cambios en el producto según se va desarrollando. Entre las más destacadas se pueden mencionar a XP, Scrum, MSF Agile, AUP y Crystal Clear.

### **2.3.2 Comparación entre las Metodologías de Desarrollo de Software**

Para la selección de la metodología a utilizar, se han escogido como candidatas a RUP y XP, como representantes de las metodologías tradicionales y ágiles, respectivamente. Para evaluarlas, se tomarán en cuenta los siguientes criterios:

- Adaptación a la naturaleza del software a desarrollar.
- Adaptación a la tecnología para la cual se desarrollará el software.

- Basado en normas y estándares preestablecidos.
- Comunicación entre los integrantes del grupo de trabajo y con el cliente.
- Control de la calidad del producto y los entregables.
- Permite el uso de Herramientas CASE.
- Nivel de conocimiento de los desarrolladores de la metodología.

Para cada una de las características anteriormente señaladas, se definirá el nivel de importancia que posee cada una para el desarrollo del aplicativo, en un valor de ponderación Alto (3), Medio (2) y Bajo (1), como se muestra en la siguiente tabla:

| <b>Literal</b> | <b>Criterio</b>   | <b>Ponderación</b> |
|----------------|---|--------------------|
| A              | Adaptación a la naturaleza del software a desarrollar               | 3                  |
| B              | Adaptación a la tecnología para la cual se desarrollará el software | 3                  |
| C              | Basado en normas y estándares preestablecidos                       | 2                  |
| D              | Comunicación entre los integrantes del grupo de trabajo             | 2                  |
| E              | Control de la calidad del producto y los entregables                | 3                  |
| F              | Permite el uso de Herramientas CASE                                 | 2                  |
| G              | Flexibilidad para cambios   | 1                  |
| H              | Nivel de conocimiento de los desarrolladores                        | 2                  |

**TABLA N° 3:** Criterios de ponderación para evaluación de metodologías de desarrollo de software

A continuación se presenta la tabla de evaluación de las metodologías para el desarrollo de software, en la cual según el nivel en el que aplica una característica para cada metodología se la calificará basándose en el siguiente criterio: Aplica totalmente (2), Aplica parcialmente (1), No aplica (0).

| Característica | Ponderación | Grado Aplica |    | Puntaje Final |    |
|----------------|-------------|--------------|----|---------------|----|
|                |             | RUP          | XP | RUP           | XP |
| A              | 3           | 2            | 2  | 6             | 6  |
| B              | 3           | 2            | 2  | 6             | 6  |
| C              | 2           | 2            | 1  | 4             | 2  |
| D              | 2           | 2            | 2  | 4             | 4  |
| E              | 3           | 2            | 1  | 6             | 3  |
| F              | 2           | 2            | 0  | 4             | 0  |
| G              | 1           | 1            | 3  | 2             | 6  |
| H              | 2           | 2            | 1  | 4             | 2  |
| TOTAL PUNTAJE  |             |              |    | 36            | 29 |

**TABLA N° 4:** Evaluación de metodologías de desarrollo de software

En base a los resultados obtenidos, se puede resaltar que las características que más se destacan en la evaluación de la metodología RUP por sobre XP son las siguientes:

- C (Basado en normas y estándares preestablecidos).
- E (Control de la calidad del producto y los entregables).
- F (Permite el uso de Herramientas CASE).

Esto junto al resto de atributos de la metodología RUP, en los cuales obtiene un puntaje similar a XP, garantiza el éxito del proyecto y la entrega de un producto de calidad al cliente, por lo cual se la selecciona como metodología de desarrollo.

## **2.4 PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE**

### **2.4.1 Introducción**

El Proceso Unificado de Rational (RUP, por sus siglas en inglés), es un proceso de desarrollo de software, el cual contempla una serie de actividades y pasos necesarios para convertir los requisitos del usuario en un sistema de software.<sup>3</sup>

El RUP es un proceso genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones y diferentes tamaños de proyectos. El RUP permite además, asegurar la producción de software de alta calidad que cumple con los requerimientos del usuario, dentro de un presupuesto y tiempo establecidos.

---

<sup>3</sup> JACOBSON, I., BOOCH, G. y RUMBAUGH, J, “El Proceso Unificado de Desarrollo de Software”. Pag. 4, Addison-Wesley. 2000



Las tres características esenciales del RUP son:

- Dirigido por los Casos de Uso
- Iterativo e Incremental
- Centrado en la Arquitectura

#### **2.4.1.1 Dirigido por los Casos de Uso**

Con esto se refiere a la utilización de los Casos de Uso para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Los Casos de Uso son la base para la implementación de las fases y disciplinas del RUP.

Un Caso de Uso es una secuencia de pasos que se deben ejecutar para la consecución de un fin u objetivo. Se lo relaciona directamente con los requerimientos, ya que un Caso de Uso es la sucesión de actividades que conlleva la realización e implementación de un requerimiento planteado por el cliente.

#### **2.4.1.2 Iterativo e Incremental**

Es el modelo utilizado por RUP para el desarrollo de un proyecto de software. Este modelo plantea la implementación del proyecto a realizar en Iteraciones, con lo cual se pueden evaluar los objetivos a cumplir en cada ciclo y así poder ir avanzando en el desarrollo del proyecto iteración por iteración, con lo cual se tienen varias ventajas, entre las que resalta el hecho de poder generar pequeños avances del proyectos que son entregables al cliente, quien puede probar dichas entregas parciales mientras se continúa con el desarrollo de la siguiente iteración

del proyecto. De esta manera, el proyecto va creciendo hasta completarlo en su totalidad.

### **2.4.1.3 Centrado en la Arquitectura**

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.<sup>4</sup> El RUP define la arquitectura de un sistema como una arquitectura ejecutable; es decir, una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades, construida como un prototipo evolutivo en el tiempo.

### **2.4.2 Ciclo de vida del RUP**

El RUP tiene dos dimensiones:

- El eje horizontal representa tiempo y muestra los diferentes ciclos de vida del proceso.
- El eje vertical representa las disciplinas, que agrupan actividades de una manera lógica de acuerdo a su naturaleza.

La primera dimensión representa el aspecto dinámico del proceso y se expresa en términos de fases y de iteraciones. La segunda dimensión representa el aspecto estático del proceso, se describe en términos de los componentes del proceso: las disciplinas, actividades, flujos de trabajo, artefactos y roles.

---

<sup>4</sup> **El Proceso Unificado de Desarrollo de Software,** [www.utim.edu.mx/~mgarcia/DOCUMENTO/ADSI2/ApuntesRUP.pdf](http://www.utim.edu.mx/~mgarcia/DOCUMENTO/ADSI2/ApuntesRUP.pdf)

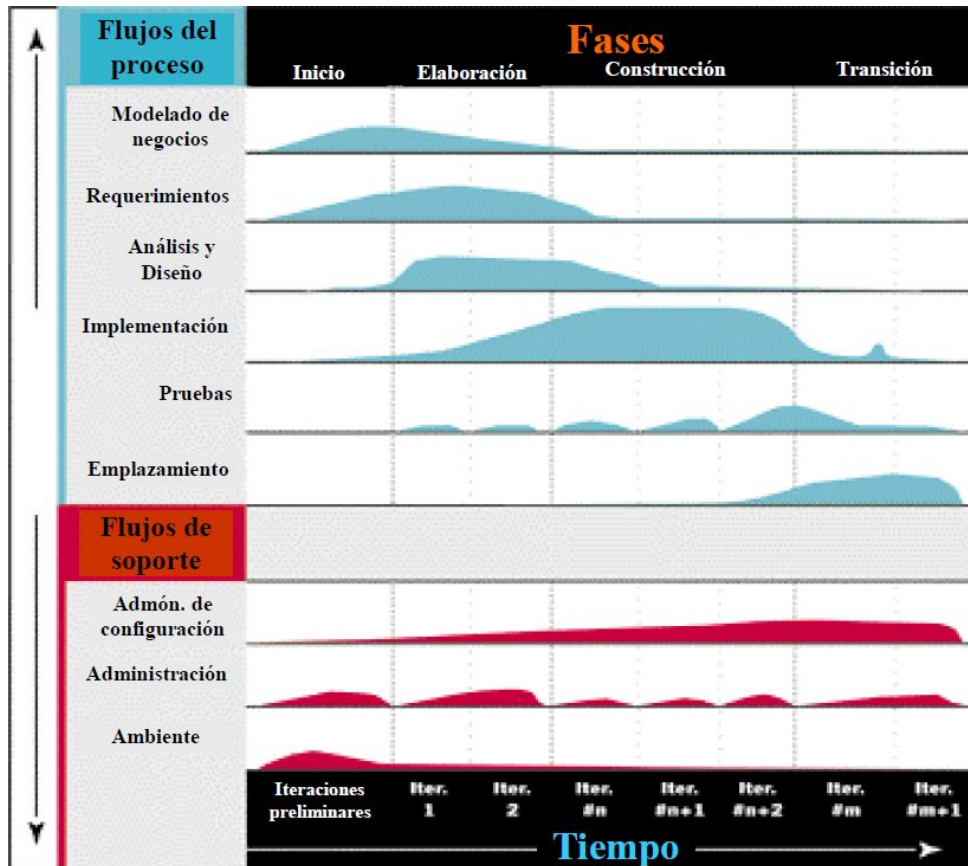


FIGURA N° 4: Ciclo de vida RUP

En la figura, se puede observar como el énfasis de cada disciplina varía en un cierto punto en el tiempo, y durante cada una de las etapas. Por ejemplo, en las iteraciones iniciales, se requiere de un mayor esfuerzo en la obtención de requerimientos y en el análisis; mientras que, en las iteraciones finales se realiza un mayor énfasis en poner en práctica la ejecución del proyecto en sí.

El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales. En cada extremo de una fase se realiza una evaluación para determinar si los objetivos de la fase se han cumplido. Una evaluación satisfactoria permite que el proyecto se mueva a la próxima fase.

#### **2.4.2.1 Concepción, Inicio o Estudio de oportunidad**

- Define el ámbito y objetivos del proyecto.
- Se define la funcionalidad y capacidades del producto.

#### **2.4.2.2 Elaboración**

Tanto la funcionalidad como el dominio del problema se estudian en profundidad

- Se define una arquitectura básica.
- Se planifica el proyecto considerando recursos disponibles.

#### **2.4.2.3 Construcción**

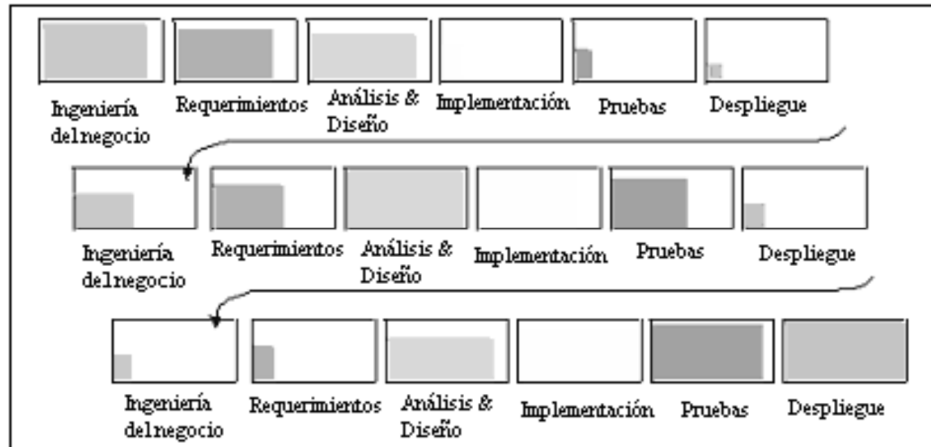
El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación

- Las fases de estudio y análisis generaron una arquitectura básica que en este punto es refinada de manera incremental, conforme se construye.
- Gran parte del trabajo es aplicado a la programación y ejecución de pruebas.
- Se documenta tanto el sistema construido como el manejo del mismo.
- Esta fase proporciona un producto construido junto con la documentación.

#### **2.4.2.4 Transición**

- Se libera el producto y se entrega al usuario para un uso real.
- Se incluyen tareas de instalación, configuración, entrenamiento, soporte, mantenimiento, etc.

- Los manuales de usuario se completan y complementan con la información anterior.
- Estas tareas se realizan también en iteraciones, de requerirse.



**FIGURA N° 5:** Iteraciones en el RUP

En la figura se visualiza el ciclo de vida de un software siguiendo el enfoque Iterativo Incremental, en el cual se puede observar que en cada iteración se realiza una pequeña parte de cada disciplina en paralelo, aumentando así poco a poco hasta concluir con la realización de todas las disciplinas con un número de iteraciones prudente.

Todas estas etapas se ejecutan en cada interacción en mayor o menor medida de acuerdo a como avanza el desarrollo del proyecto.

## **2.4.3 Flujos de Trabajo del RUP**

### **2.4.3.1 Flujos de Trabajo Fundamentales**

#### **2.4.3.1.1 Modelado del negocio**

Esta etapa tiene como objetivos comprender la estructura y la dinámica de la organización, entender los problemas de los procesos actuales e identificar las posibles mejoras y descubrir los procesos de negocio. Utiliza el modelo de Casos de Uso del Negocio para vislumbrar los procesos del negocio y los clientes, el Modelo de Objetos del Negocio para describir cada caso de uso del Negocio con los Trabajadores, además se utilizan los Diagramas de Actividad y de Clases.

#### **2.4.3.1.2 Requerimientos**

Tiene por objetivos establecer lo que el sistema debe hacer (especificar los requisitos), definir los límites del sistema, realizar una estimación del costo y tiempo de desarrollo. Utiliza el Modelo de Casos de Uso para definir el sistema que comprende los actores y relaciones, además utiliza los Diagramas de Estado de cada caso de uso.

#### **2.4.3.1.3 Análisis y diseño**

Permite definir la arquitectura del sistema y tiene como metas el convertir los requisitos en especificaciones de implementación. El análisis se refiere a transformar casos de uso en clases, y el diseño se refiere a refinar el análisis para poder implementar los diagramas de clases, los diagramas de colaboración de

cada caso de uso, los diagramas de secuencia, el diagrama de estados de las clases y el modelo de despliegue de la arquitectura.

#### **2.4.3.1.4 Implementación**

Es utilizada para implementar las clases de diseño como componentes de programación, probar los componentes individualmente e integrar los componentes en un sistema ejecutable. Utiliza el Modelo de Implementación, conjuntamente con los Diagramas de Componentes para comprender cómo se organizan los componentes y dependen unos de otros.

#### **2.4.3.1.5 Pruebas**

Su fin principal es el de verificar la integración de los componentes (prueba de integración), verificar que todos los requisitos han sido implementados (pruebas del sistema), asegurar que los defectos detectados han sido resueltos antes de la entrega final.

#### **2.4.3.1.6 Despliegue**

Esta disciplina posee como objetivos el asegurar que el producto está preparado para el cliente, proceder a su entrega y recepción por parte del cliente. En esta disciplina se realizan las actividades de probar el software en su entorno final, empaquetarlo, distribuirlo e instalarlo, así como la tarea de enseñar al usuario acerca del manejo del sistema.

### **2.4.3.2 Flujos de Trabajo de Soporte**

#### **2.4.3.2.1 Gestión y configuración de cambios**

Es esencial para controlar el número de artefactos producidos por la cantidad de personal que trabajan en un proyecto conjuntamente. Los controles sobre los cambios son de extrema ayuda para evitar costosas confusiones como la compostura de algo que ya se había arreglado, conflictos debido a la actualización simultánea de código fuente o múltiples versiones, etc.

#### **2.4.3.2.2 Gestión del proyecto**

La gestión del proyecto tiene como meta el administrar el riesgo y superar restricciones con óptimas soluciones para entregar un producto que satisface las necesidades de los clientes con éxito. Su propósito consiste en proveer pautas para administrar el proyecto, dirigir al personal y administrar el riesgo.

#### **2.4.3.2.3 Entorno**

Esta disciplina hace énfasis sobre las actividades necesarias para configurar el proceso que engloba el desarrollo de un proyecto y su propósito es proveer a la organización que desarrollará el software un ambiente en el cual basarse, el cual provee los procesos y las herramientas para poder producir el software.



## 2.5 CONCEPTOS EN TORNO A LA METODOLOGÍA UML

### 2.5.1 Introducción

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía *Rational* fundada por Booch.<sup>5</sup> El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (*Object Modelling Tool*). El primer borrador de UML apareció en octubre de 1995. En esa misma época, otro reconocido investigador, Jacobson, se unió a *Rational* y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Todas estas colaboraciones condujeron a la definición de la primera versión de UML.

El Lenguaje Unificado de Modelado, UML es una notación estándar para el modelado de sistemas de software, resultado de una propuesta de estandarización promovida por el consorcio OMG (Object Management Group) en 1996, del cual forman parte las empresas más importantes que se dedican al desarrollo de software.

El UML tiene una notación gráfica muy comprensiva, la cual permite representar en mayor o menor medida todas las fases de un proyecto de software. Es decir permite diagramar o graficar desde el análisis con los casos de uso, el diseño con los diagramas de clases hasta la implementación y configuración con los diagramas de despliegue.

---

<sup>5</sup> JACOBSON, I., BOOCH, G. y RUMBAUGH, J, “El lenguaje Unificado de Modelado, Manual de Referencia”. Pag. 5, Addison-Wesley. Tercera Edición

UML es un lenguaje de modelado, lo cual viene a ser una simplificación de la realidad. El objetivo del modelado de un sistema es capturar las partes esenciales del sistema y transponerlas en un diagrama que sea fácilmente entendible. Para facilitar este modelado, se realiza una abstracción y se plasma en una notación gráfica. Esto se conoce como modelado visual.

UML sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. Además, es independiente del lenguaje de programación, de tal forma que los diseños realizados usando UML se pueden implementar en cualquier lenguaje que soporte las posibilidades de UML (lenguajes orientados a objetos).

UML al ser un método formal de modelado aporta varias ventajas, entre las cuales es posible mencionar<sup>6</sup>:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén sincronizados y actualizados.

### **2.5.2 Objetivos**

Son varios los objetivos que persigue el UML; sin embargo, se pueden resumir sus funciones en las siguientes:

---

<sup>6</sup> El Lenguaje Unificado de Modelado (UML), <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

- Visualizar: UML permite expresar de manera gráfica un sistema de forma que otra persona lo pueda entender.
- Especificar: UML permite detallar las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos del UML sirven como documentación del sistema desarrollado y a la vez, sirven para una futura revisión.

Un modelo UML está compuesto por tres clases de bloques de construcción<sup>7</sup>:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

### 2.5.3 Diagramas Utilizados

Un diagrama es la representación gráfica de un conjunto de elementos junto con sus relaciones; es decir, un diagrama ofrece una vista del sistema a modelar. UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas:

- Diagrama de Clases y Objetos.

---

<sup>7</sup> El Lenguaje Unificado de Modelado (UML), <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

- Diagrama de Casos de Uso.

Para modelar el comportamiento dinámico del sistema están los diagramas de:

- Diagrama de Interacción.
- Diagrama de Colaboración.
- Diagrama de Estados.

Los diagramas enfocados a la implementación del sistema son:

- Diagrama de Componentes.
- Diagrama de Despliegue.

### 2.5.3.1 Diagrama de Clases y Objetos

El diagrama de clases muestra un conjunto de clases, interfaces y sus relaciones. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos.

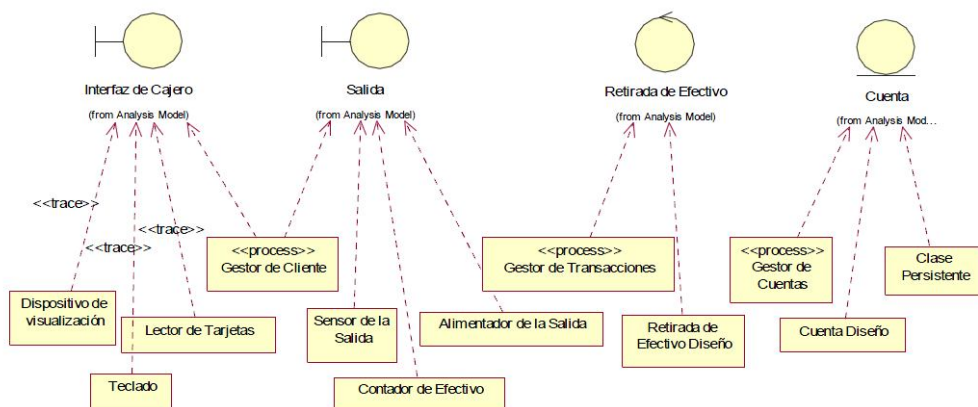


FIGURA N° 6: Diagrama de Clases y Objetos

### 2.5.3.2 Diagrama de Casos de Uso

El diagrama de casos de usos representa gráficamente los casos de uso que posee un sistema. Se define un caso de uso como cada interacción con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que debe hacer el sistema propuesto.

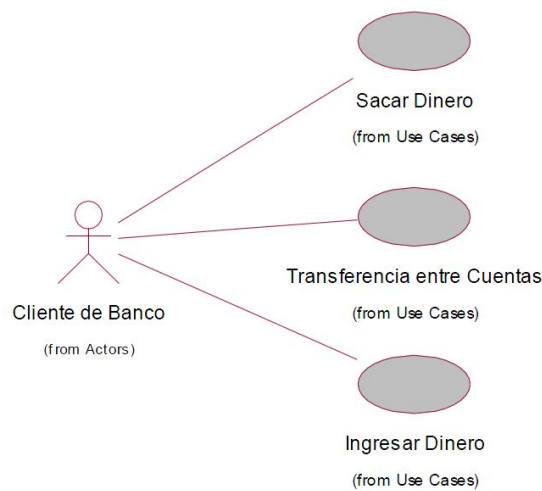


FIGURA N° 7: Diagrama de Casos de Uso

### 2.5.3.3 Diagrama de Interacción

Estos diagramas, según la interacción que representan, a su vez se dividen en 2 tipos de diagramas: Diagramas de Secuencia y Diagramas de Colaboración.

#### 2.5.3.3.1 Diagrama de Secuencia

Hace énfasis en la interacción entre los objetos y los mensajes que intercambian entre sí, junto con el orden cronológico de los mismos. En el diagrama de secuencia se muestra la interacción de los objetos que componen un sistema de forma temporal.

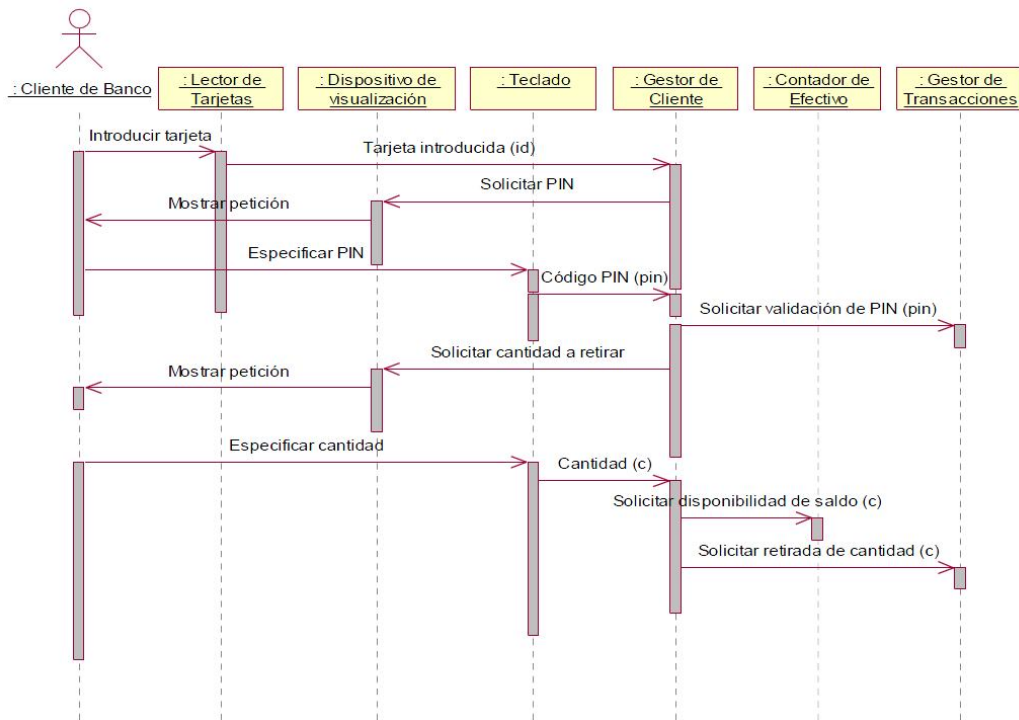


FIGURA N° 8: Diagrama de Secuencia

### 2.5.3.3.2 Diagrama de Colaboración

De igual manera, muestra la interacción entre los objetos, resaltando la organización estructural de los mismos, en lugar del orden de los mensajes intercambiados.

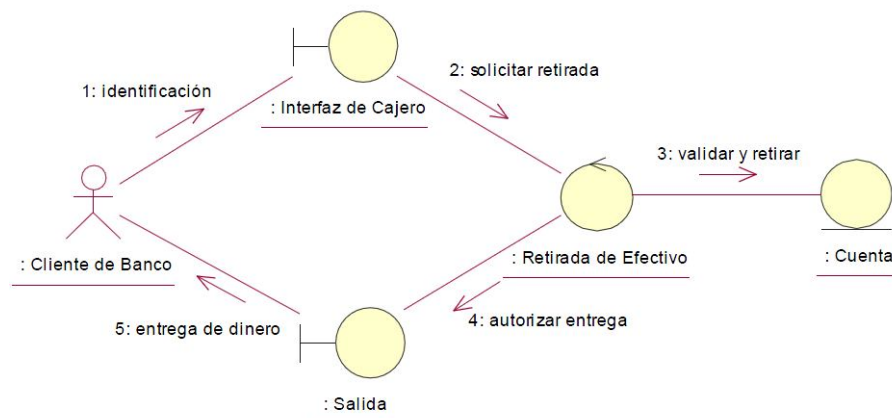


FIGURA N° 9: Diagrama de Colaboración

### 2.5.3.4 Diagrama de Estados

Permite modelar el comportamiento del sistema de acuerdo con eventos. Sirven para establecer el estado de las entidades del sistema en un momento determinado.

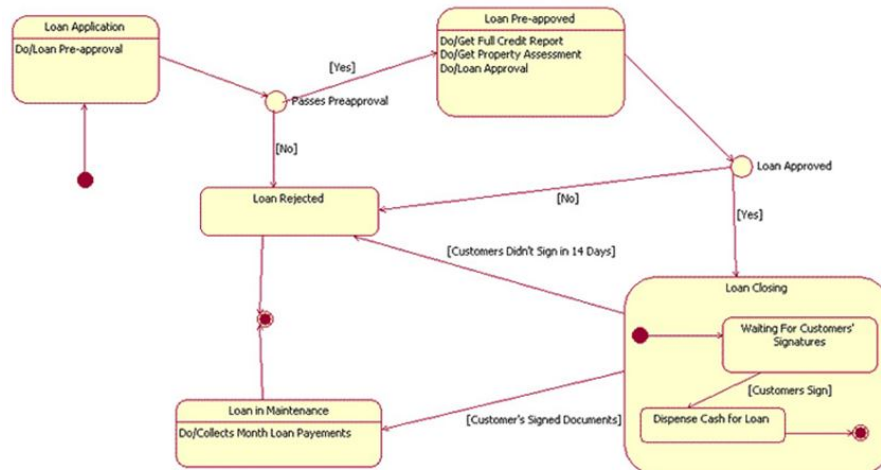


FIGURA N° 10: Diagrama de Estados

### 2.5.3.5 Diagrama de componentes

Muestra la organización y las dependencias entre un conjunto de componentes.

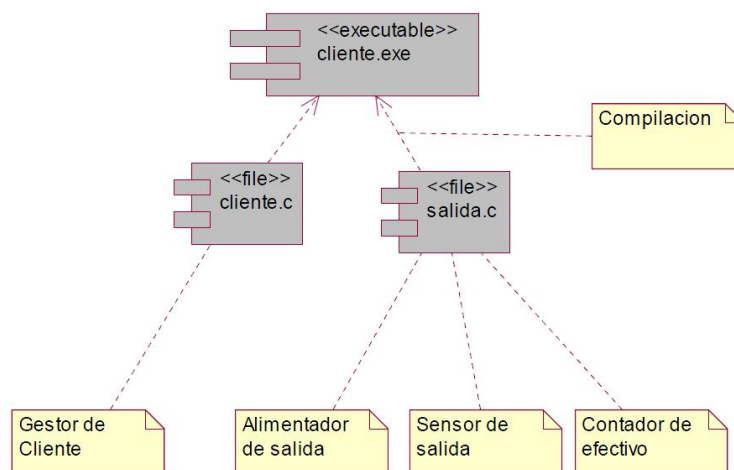


FIGURA N° 11: Diagramas de Componentes

### 2.5.3.6 Diagrama de despliegue

Permite representar los dispositivos que se encuentran en un sistema y su distribución en el mismo.

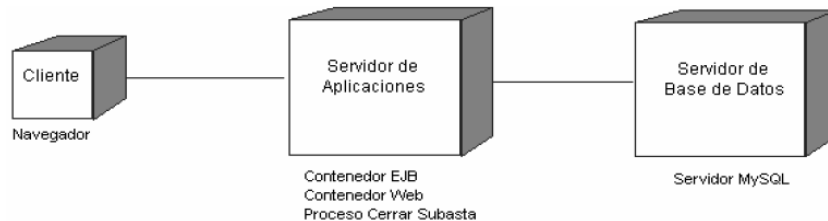


FIGURA N° 12: Diagrama de Despliegue

## 2.6 DESCRIPCIÓN DE LAS HERRAMIENTAS DE DESARROLLO

A continuación se detallan las herramientas que serán utilizadas para el desarrollo de este proyecto de tesis, se hace referencia a dichas herramientas para tener un panorama claro sobre su alcance, beneficios y contribución de las mismas.

Es necesario aclarar que la EPMGDT con anterioridad a la realización del presente proyecto de tesis, adquirió las herramientas **Visual Studio .NET 2005** y **SQL Server 2005** para el desarrollo de aplicaciones y como motor de base de datos, respectivamente. En la investigación realizada, se ha comprobado que son herramientas muy robustas y varios de sus beneficios serán mencionados en párrafos posteriores.

### 2.6.1 FRONT-END

#### 2.6.1.1 Definición Herramientas de desarrollo

Se conoce como herramienta de desarrollo a los programas que están inmersos en el desarrollo de una aplicación, es decir son programas específicos que permiten



al usuario la creación de nuevas aplicaciones. Estos pueden ser de gran importancia, como un compilador, ensamblador o un editor, o de importancia secundaria como un entorno de desarrollo integrado (IDE).

### **2.6.1.2 Comparación de lenguajes de Programación C# y Java (Eclipse)**

Muchos sostienen que C# es una copia de Java. Si se toman ciertos fragmentos de código (convenientemente) escritos en alguno de estos lenguajes, podría ser imposible determinar si es C# o es Java.

Lo cierto es que parte de la filosofía empleada por ambos lenguajes es la misma. Ambos se ejecutan sobre una máquina virtual, y esta característica los convierte en lenguajes potencialmente multiplataforma; ambos poseen especificaciones de subconjuntos de recursos de lenguaje para implementaciones en diversos dispositivos (por ejemplo, móviles), y ambos tienen una librería de clases con muchas características en común.

Sin embargo, C# presenta la ventaja de integrarse mejor con aplicaciones nativas de la plataforma sobre la cual se encuentra trabajando. Claro que, en caso de acceder a recursos nativos, perderá la característica de ser multiplataforma, pero esta característica no siempre es deseada. En algún momento podría presentarse la necesidad de invocar métodos de librerías nativas creadas en C++ para Windows de una manera sencilla y eficiente. Luego se podría discutir en muchos puntos cuál es mejor que cuál; en la Red, los foros de discusión sobre programación se encuentran plagados de peleas de este tipo. La realidad es que hoy día C# es más fuerte en plataformas basadas en Windows, mientras que Java es más fuerte en una gran diversidad de plataformas menos populares (celulares, tarjetas

inteligentes, etc.) y es el candidato ideal si hoy desea construir una aplicación que deba ejecutarse sin cambios en Linux y Windows, aunque esta realidad se encuentre próxima a cambiar.<sup>8</sup>

### 2.6.1.3 Selección de la Herramienta de Desarrollo

Para realizar la selección de las herramientas de desarrollo se otorgará una valoración a ciertas características o requisitos que debe cumplir la herramienta según una ponderación establecida.

Las características a evaluar se mencionan a continuación:

- **Facilidad de Interacción con Servicios Web.** Es importante que exista esta interacción ya que EPMGDT por una parte se alimentará de otras fuentes de información a través de servicios web.
- **Facilidad de Interacción con Aplicaciones Distribuidas.** Se prevé que EPMGDT interactúe con otros sistemas, que se encuentren en la misma o diferente plataforma.
- **Facilidad de Interacción con Dispositivos Portables.** En el futuro existe la posibilidad de manejar ciertas funcionalidades de EPMGDT a través de dispositivos portables.
- **Conocimiento Adquirido.** Es importante que las personas involucradas en la implementación de EPMGDT tengan el conocimiento y la experiencia necesaria en la herramienta seleccionada para cumplir con los tiempos de desarrollo y con la calidad esperada.

---

<sup>8</sup> Lenguaje C,  
[http://www.cepeu.edu.py/LIBROS\\_ELECTRONICOS\\_2/Lenguaje%20c.pdf](http://www.cepeu.edu.py/LIBROS_ELECTRONICOS_2/Lenguaje%20c.pdf)

- **Soporte.** Para la implementación de nuevas tecnologías o inconvenientes presentados en el transcurso del desarrollo de EPMGDT es importante que se tenga el soporte adecuado con respecto a la herramienta de desarrollo.

#### 2.6.1.4 Comparación Herramientas de Desarrollo

A través de la siguiente comparación, se determinará la mejor herramienta de desarrollo para la implementación del sistema EPMGDT. La ponderación utilizada es 3 para alta, 2 para media y 1 para baja.

| Criterio   | Ponderación | Evaluación |           | Cálculo   |           |
|--|-------------|------------|-----------|-----------|-----------|
|  |             | Eclipse    | C#        | Eclipse   | C#        |
| Facilidad de Interacción con Servicios Web             | 3           | 2          | 3         | 6         | 9         |
| Facilidad de Interacción con Aplicaciones Distribuidas | 3           | 3          | 3         | 9         | 9         |
| Facilidad de Interacción con Dispositivos Portables    | 2           | 1          | 2         | 2         | 4         |
| Conocimiento Adquirido                                 | 3           | 1          | 3         | 3         | 9         |
| Soporte  | 1           | 1          | 1         | 1         | 1         |
| <b>Valor Total</b>                                     | <b>11</b>   | <b>8</b>   | <b>12</b> | <b>21</b> | <b>32</b> |

TABLA N° 5: Presupuesto del Proyecto

En base a estos resultados se ha seleccionado el lenguaje de programación Visual C#, que forma parte de la herramienta Visual Studio .NET.

### **2.6.1.5 Visual Studio .NET**

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación como Visual C++, **Visual C#**, Visual J#, ASP.NET y Visual Basic.NET.

La primera versión lanzada al mercado fue **Visual Studio 6.0** el 30 de julio de 1998, la última versión estable es **Visual Studio 2010**, salió al mercado el 12 de abril del 2010.

#### **2.6.1.5.1 Características de C#**

Entre los beneficios del lenguaje de programación Visual C# es posible destacar que es un lenguaje de programación orientado a objetos, la sintaxis del mismo se asemeja a la de C++ y Java, en él se combina la alta productividad de Visual Basic con la flexibilidad y poder de C++. Este lenguaje gestiona automáticamente la memoria, solucionando así los problemas más comunes de programación que existen en C o C++. A través de la plataforma .NET es posible interactuar con otros componentes realizados en otros lenguajes .NET de manera muy fácil, también se puede interactuar con componentes no gestionados fuera de la plataforma .NET, por esta razón se puede integrar con facilidad en sistemas ya creados, al poseer una librería de clases muy completa y muy bien diseñada, esta permitirá disminuir en un gran porcentaje los tiempos de desarrollo.

## 2.6.2 BACK-END

### 2.6.2.1 Selección del Motor de Base de Datos

Para realizar la selección del motor de base de datos se otorgará una valoración a ciertas características o requisitos que debe cumplir el motor según una ponderación establecida.

Las características a evaluar se mencionan a continuación:

- **Almacenamiento de Imágenes.** Se prevé que la base de datos para EPMGDT en algún momento almacene imágenes escaneadas.
- **Backup / Recovery.** Es importante que se pueda hacer un buen uso de los respaldos y en ciertas ocasiones de la recuperación de la base de datos de EPMGDT.
- **Conocimiento Adquirido.** Las personas involucradas en la implementación de EPMGDT deben tener el conocimiento y la experiencia necesaria en el lenguaje y el motor de base de datos para cumplir con los tiempos de desarrollo y con la calidad esperada.
- **Soporte.** Para la implementación de nuevas tecnologías o inconvenientes presentados en el transcurso del desarrollo de EPMGDT es importante que se tenga el soporte adecuado con respecto al motor de base de datos.

### 2.6.2.2 Comparación Motores de Base de Datos

A través de la siguiente comparación, se determinará el mejor motor de base de datos para la implementación del sistema EPMGDT. La ponderación utilizada es 3 para alta, 2 para media y 1 para baja.

| Criterio                   | Ponderación | Evaluación |                 |            | Cálculo   |                 |            |
|----------------------------|-------------|------------|-----------------|------------|-----------|-----------------|------------|
|                            |             | MySQL      | SQL Server 2005 | Oracle 10g | MySQL     | SQL Server 2005 | Oracle 10g |
| Almacenamiento de Imágenes | 3           | 1          | 3               | 3          | 3         | 9               | 9          |
| Backup / Recovery          | 3           | 1          | 3               | 3          | 3         | 9               | 9          |
| Conocimiento Adquirido     | 3           | 1          | 3               | 1          | 3         | 9               | 3          |
| Soporte                    | 2           | 2          | 2               | 2          | 6         | 6               | 6          |
| <b>Valor Total</b>         | <b>11</b>   | <b>5</b>   | <b>11</b>       | <b>9</b>   | <b>15</b> | <b>33</b>       | <b>27</b>  |

**TABLA N° 6:** Presupuesto del Proyecto

En base a estos resultados se ha seleccionado el motor de base de datos SQL Server 2005.

### 2.6.2.3 Microsoft SQL Server

Microsoft SQL Server es un sistema para la gestión de base de datos creado por Microsoft basado en el modelo relacional, sus lenguajes de consultas son T-SQL y ANSI SQL.

La primera versión lanzada al mercado fue **SQL Server 1-0** en 1989, la última versión estable es **SQL Server 2008**, salió al mercado en el 2008.

Entre los beneficios que brinda el motor de base de datos mencionado, se encuentran las siguientes: compatibilidad integrada con XML, cuenta con características de escalabilidad, disponibilidad y seguridad, las necesarias para manejar los componentes de almacenamiento de sitios Web de gran tamaño.

## **CAPÍTULO 3**

### **ANÁLISIS Y DISEÑO DE LA APLICACIÓN**

#### **3.1 ESPECIFICACIÓN DE REQUERIMIENTOS**

##### **3.1.1 Introducción.**

###### **3.1.1.1 Propósito.**

A continuación, se proporcionará una visión clara y objetiva de los requerimientos que deberán ser contemplados para la realización del Sistema de Control del Catastro Turístico para la Empresa Pública Metropolitana de Gestión de Destino Turístico de la ciudad de Quito.

El propósito es definir y documentar los requerimientos del sistema siguiendo las pautas establecidas en la norma IEEE 830. En este procedimiento se identificarán todos los requisitos que contendrá el nuevo sistema. Los requerimientos identificados en esta etapa han sido validados y podrían sufrir ciertas variaciones a medida que se avance en el desarrollo del sistema.

La siguiente especificación estará dirigida tanto al usuario del sistema como al equipo de desarrollo.

###### **3.1.1.2 Ámbito del Sistema.**

El software se denominará CatastrosQT, el mismo que estará encaminado a realizar las siguientes actividades:

- Ingreso, consulta y mantenimiento de los establecimientos turísticos del distrito metropolitano de Quito, en base a la plantilla definida para tal efecto por el Ministerio de Turismo.



- Registro de las Emisiones, Pagos y Bajas de los valores que cada establecimiento debe pagar para obtener el permiso anual de funcionamiento.
- Impresión de la LUAF (Licencia Única Anual de Funcionamiento).
- Emisión de Reportes con información actualizada.
- Administración de usuarios del sistema, mediante la implementación de perfiles de usuarios específicos.

El sistema CatastrosQT no abarcará los procesos contables ni de facturación o de retenciones que por Ley se deban aplicar por cuanto el cobro efectivo a los establecimientos turísticos lo realiza en la actualidad el Departamento de Recaudaciones del Ilustre Municipio de Quito.

Uno de los beneficios más importantes que se obtendrán con el uso del sistema CatastrosQT será el de optimizar el manejo de la información referente al cumplimiento o no de la normativa por parte de los establecimientos turísticos, a la vez que se agiliza el proceso de registro de dicha información y se facilita la generación de reportes actualizados y oportunos a los directivos de la EPMGDT.

Los principales objetivos que se esperan alcanzar con el sistema CatastrosQT son:

- Agilizar la tarea de quien registra la información de los establecimientos turísticos del distrito metropolitano.
- Facilitar el control de los establecimientos que incumplen con la normativa vigente.
- Reducir al mínimo el tiempo requerido para la emisión de reportes gerenciales.

- Minimizar la posibilidad de errores y llevar un log de auditoria de cambios realizados en el sistema.

### 3.1.1.3 Definiciones, Acrónimos y Abreviaturas

#### 3.1.1.3.1 Definiciones

| Término            | Definición   |
|--------------------|--|
| Baja de Emisiones  | Eliminación o Anulación de la emisión de valores que se tiene para un establecimiento, el cual se realiza debido a un error en la misma.   |
| BC                 | Cuarto con Baño Compartido   |
| CBP                | Cuarto con Baño Privado  |
| Detalles           | Número de ítems que posee un establecimiento y que aplican para evaluarlo en una mayor o menor categoría y en base a los cuales también puede variar el valor de la emisión.                                 |
| Emisión            | Emisión de valores que un establecimiento debe cancelar correspondientes al permiso anual de funcionamiento. Este se calcula en base a la actividad, tipo y categoría a la que pertenece el establecimiento. |
| Número de Registro | Número de registro turístico otorgado por el Ministerio de Turismo a los establecimientos catastrados.   |

|                   |  |
|-------------------|--|
| Pago de Emisiones | Registro del pago total o parcial que realiza un establecimiento al valor emitido para un año en particular.   |
| Perfil de usuario | Nivel de capacidad de acción que posee un usuario del sistema, basado en una colección de recursos que le son asignados, los cuales habilitan o restringen el acceso del mismo a las opciones del sistema. |
| Recurso           | Opción del sistema o sección del mismo que se asigna o restringe a un usuario en base al perfil asignado al mismo.   |
| Servidor          | Computador en el cual se instala el sistema y al cual deben acceder los usuarios para poder hacer uso del mismo.   |

**TABLA N° 7:** Definiciones de la Especificación de Requerimientos

### 3.1.1.3.2 Acrónimos y Abreviaturas

| <b>Término</b> | <b>Definición</b>   |
|----------------|---|
| BD             | Base de Datos   |
| BPM            | Business Process Model (Modelo de Proceso de Negocios)  |
| CatastrosQT    | Sistema de Control del Catastro Turístico para la Empresa Pública Metropolitana de Gestión de Destino Turístico de la ciudad de Quito |
| GUI            | Graphic User Interface (Interfaz Gráfica de Usuario)  |

|                  |   |
|------------------|---|
| DBMS             | Data Base Management System (Sistema Administrador de Base de Datos)  |
| IEEE             | Institute of Electrical & Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)   |
| IIS              | Internet Information Service (Servicio de Información de Internet)  |
| SQL              | Structured Query Language (Lenguaje de Consulta Estructurado)   |
| Sistema LUAE-BPM | Sistema informático implementado por el Municipio de Quito, el mismo que podrá dar inicio al trámite de registro de un establecimiento turístico. |
| TCP/IP           | Transfer Control Protocol / Internet Protocol (Protocolo de Control de Transporte / Protocolo de Internet)  |
| Web Service      | Servicio Web que permite el intercambio de datos entre aplicaciones.  |

**TABLA N° 8:** Acrónimos de la especificación de requerimientos

#### **3.1.1.4 Referencias**

- Especificación de Requisitos según el estándar de IEEE 830.
- CATASTROS MANUAL DE PROCESOS - emitido por la Gerencia Nacional de Recursos Turísticos del Ministerio de Turismo en el año 2006 (Anexo D).

- Ordenanza Municipal 0243 - establece los valores a cobrar a los establecimientos turísticos en base a su actividad, tipo y categoría (Anexo E).

### **3.1.1.5 Visión general del documento**

A continuación se detallan los requisitos de software que serán contemplados en el sistema CatastrosQT, cuya especificación de requerimientos está dividida en 2 partes:

- La primera que contempla la Descripción General de todos aquellos factores que tienen incidencia en los requerimientos del sistema.
- La segunda parte que se refiere a la descripción de los Requisitos Específicos en lo que tiene que ver con interfaces, funciones, requerimientos de rendimiento, de diseño, atributos del sistema y otros.

## **3.1.2 Descripción General**

### **3.1.2.1 Perspectiva del producto**

El sistema CatastrosQT es una aplicación en esencia independiente, por lo que básicamente podrá operar sin necesidad de interactuar con otros sistemas. Al momento el Municipio está implementando un nuevo sistema LUAE-BPM, el mismo que podrá registrar de manera inicial el trámite de solicitud del registro turístico de un establecimiento nuevo o la re-inspección de un establecimiento que haya sido registrado anteriormente en el sistema CatastrosQT y que requiera la renovación de su licencia.

Dado que el sistema CatastrosQT no va a manejar los procesos contables o de tributación involucrados en los cobros realizados a los establecimientos, el mismo no requiere enlazarse con el sistema contable de la EPMGDT.

### **3.1.2.2 Funciones del sistema**

Para que el sistema CatastrosQT cumpla con las necesidades de automatización de los usuarios, debe cumplir con las siguientes funciones:

- Ingreso y Mantenimiento de la información de los establecimientos turísticos que se encuentren dentro del Distrito Metropolitano de Quito.
- Registro de Emisiones, Pagos y Bajas de los valores que deben pagar dichos establecimientos en base a la normativa municipal vigente.
- Llevar control del cumplimiento de la normativa vigente por parte de los establecimientos turísticos, mediante el registro de las inspecciones realizadas.
- Emisión de reportes que permitan llevar un control de la información registrada en el sistema a nivel de Información de Establecimientos, Recaudaciones, Inspecciones y Administrativos (internos del sistema).
- Administración de los usuarios del sistema, mediante el uso de perfiles y roles para limitar el acceso a las opciones del sistema, según corresponda el perfil de cada usuario.

### 3.1.2.3 Características de los Usuarios

El sistema contemplará la existencia de seis perfiles o tipos de usuarios:

- **Usuario Administrador** que tendrá la capacidad de modificar los permisos de otros usuarios en el sistema, mediante la asignación/remoción de recursos a los perfiles de los mismos o la creación de nuevos perfiles. Dicho usuario deberá poseer un nivel de instrucción superior en el área de sistemas, ya que poseerá un control total de la información del mismo y de la Base de Datos en general.
- **Usuario de Consulta** el cual podrá solamente visualizar la información correspondiente a las Emisiones, Inspecciones y Recaudaciones realizadas a los establecimientos turísticos. El perfil requerido para este usuario es una persona que posea conocimientos básicos de computación.
- **Usuario Inspector** que tendrá la capacidad de registrar el resultado de las inspecciones realizadas a los establecimientos turísticos en el sistema. El usuario con este perfil, podrá a su vez acceder a visualizar la información de los establecimientos turísticos, así como a los reportes relacionados con las inspecciones de los establecimientos, sin embargo no podrá visualizar/modificar la información correspondiente a las Emisiones.
- **Usuario Operador** el cual será la persona encargada de realizar el ingreso/modificación de la información de los establecimientos turísticos y que deberá contar con conocimientos básicos de computación. Este tipo de usuario podrá a su vez acceder a los reportes relacionados con la información de los establecimientos, sin embargo no podrá

visualizar/modificar la información correspondiente a las Emisiones ni Inspecciones.

- ***Usuario de Recaudaciones*** quien será la persona encargada de registrar en el sistema la información correspondiente a las Emisiones, Pagos y Bajas de los establecimientos turísticos y que deberá contar con conocimientos básicos de computación. Este tipo de usuario podrá a su vez acceder a los reportes relacionados con la información de los recaudaciones y establecimientos, sin embargo no podrá visualizar/modificar la información correspondiente a las Inspecciones.
- ***Usuario Supervisor de Inspecciones*** el cual será la persona encargada de verificar el ingreso/modificación de la información de los establecimientos turísticos y a su vez de asignar las tareas de inspección a los usuarios inspectores. También podrá acceder a los reportes relacionados con la información de los establecimientos y de inspecciones. Este tipo de usuario deberá contar con conocimientos básicos de computación.

#### **3.1.2.4 Restricciones**

##### **3.1.2.4.1 Políticas de la empresa**

La codificación utilizada para clasificar a los establecimientos turísticos, es una especificación de la EPMGDT y no podrá ser modificada por el equipo de desarrollo.



#### **3.1.2.4.2 Funciones de auditoría**

Dado que la información sobre los establecimientos turísticos y su cumplimiento o no de la normativa es de carácter crítico, se deberá llevar un Log de Auditoría de los usuarios y las acciones realizadas en el sistema que modifiquen dicha información.

#### **3.1.2.4.3 Consideraciones acerca de la seguridad**

Los archivos de configuración que contengan información acerca de las cadenas de conexión a la BD así como información que se considere sensible, deberán ser encriptadas mediante algún mecanismo de encriptación de la información con un algoritmo que imposibilite su visualización a personas no autorizadas.

#### **3.1.2.5 Suposiciones y dependencias**

Se supondrá que el sistema se ejecutará en el lado del Servidor bajo la plataforma Microsoft Windows Server 2003 o superior, con una versión del IIS 6.0 o superior. Del lado del cliente se puede utilizar cualquier navegador web, requiriéndose en el caso del Internet Explorer la versión 6.0 o superior. Adicionalmente se asumirá la utilización del programa Microsoft SQL Server 2005 Express como DBMS del servidor.

### **3.1.3 Requisitos Específicos**

#### **3.1.3.1 Interfaces externas**

A continuación se especificarán los requisitos referentes a la interfaz de usuario, interfaces con otros sistemas e interfaz de comunicaciones.

#### **3.1.3.1.1 Interfaz de Usuario**

Uno de los requisitos de usuario, la GUI deberá ser sencilla de utilizar, con menús y botones que permitan su fácil identificación y manipulación. En lo referente al formulario de registro de información de los establecimientos turísticos, se requiere que dichos formularios se muestren con los campos que apliquen a cada actividad turística, en base a la normativa definida en el documento “CATASTROS MANUAL DE PROCESOS” (Anexo D).

#### **3.1.3.1.2 Interfaces con otros sistemas**

El sistema CatastrosQT deberá exponer un servicio web el mismo que le permita al sistema LUAE-BPM iniciar el trámite de registro de un establecimiento turístico nuevo o en su defecto, la necesidad de realizar una re-inspección de un establecimiento ya existente para renovar su licencia de funcionamiento.

Adicionalmente el sistema LUAE-BPM proveerá de un servicio web el mismo que deberá ser invocado una vez que el trámite de inspección o re-inspección haya concluido y se tenga el resultado al trámite registrado inicialmente.

#### **3.1.3.1.3 Interfaces de comunicaciones**

Debido a que el sistema se lo podrá ejecutar en cualquier navegador web, el protocolo a utilizar es el TCP/IP. De igual manera, la intercomunicación entre el servidor de aplicaciones y el DBMS se la realizará mediante dicho protocolo de comunicaciones.

### 3.1.3.2 Funciones

#### 3.1.3.2.1 Ingreso al Sistema

**Requisito 1:** El sistema debe validar las credenciales de usuario para garantizar el acceso seguro al sistema y a la vez mostrar las opciones que le corresponden al perfil de usuario.

**Precondiciones:** Al usuario se le debe haber registrado previamente en el sistema. El usuario debe poseer las credenciales de acceso al mismo.

**Entradas:** Nombre de usuario, contraseña.

**Flujo Normal:** El sistema debe validar que las credenciales ingresadas pertenezcan a un usuario activo registrado en la BD. En caso de ser fallida la validación de las credenciales del usuario se debe mostrar un mensaje de error para que el usuario intente nuevamente. Una vez validado esto, se debe evaluar los recursos a los cuales se debe dar acceso al usuario en base al perfil del mismo.

**Salida:** Validación de credenciales de usuario y listado recursos que correspondan al perfil.

#### Diagrama Caso de Uso

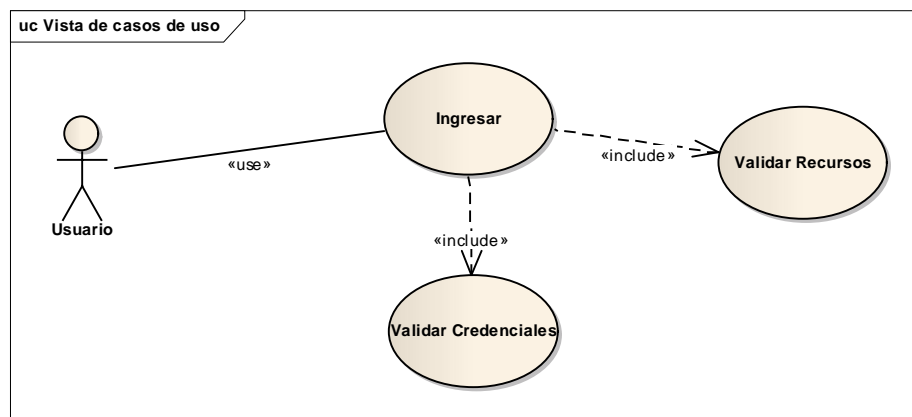


FIGURA N° 13: Diagrama de Caso de Uso – Requisito 1

## Diagrama de Secuencia

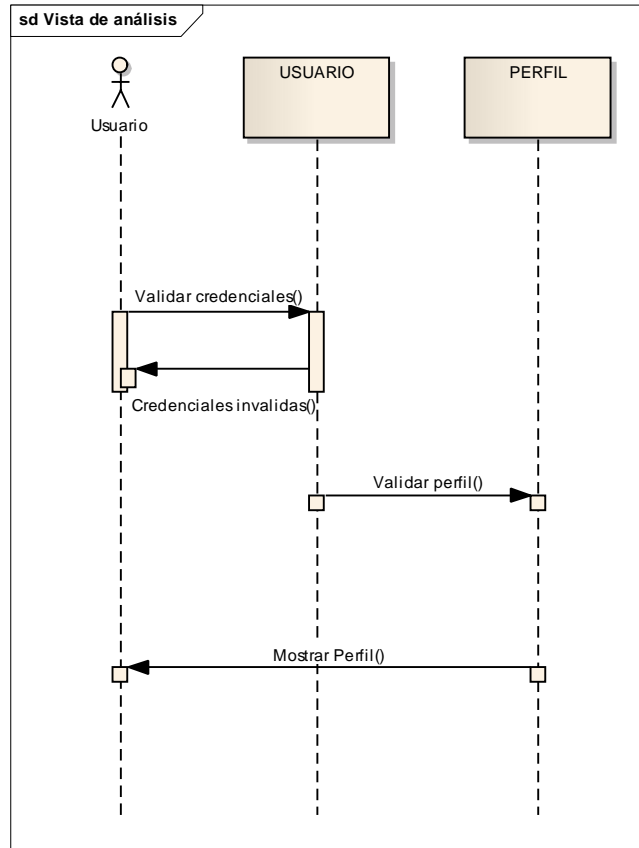


FIGURA N° 14: Diagrama de Secuencia – Requisito 1

## Diagrama de Colaboración

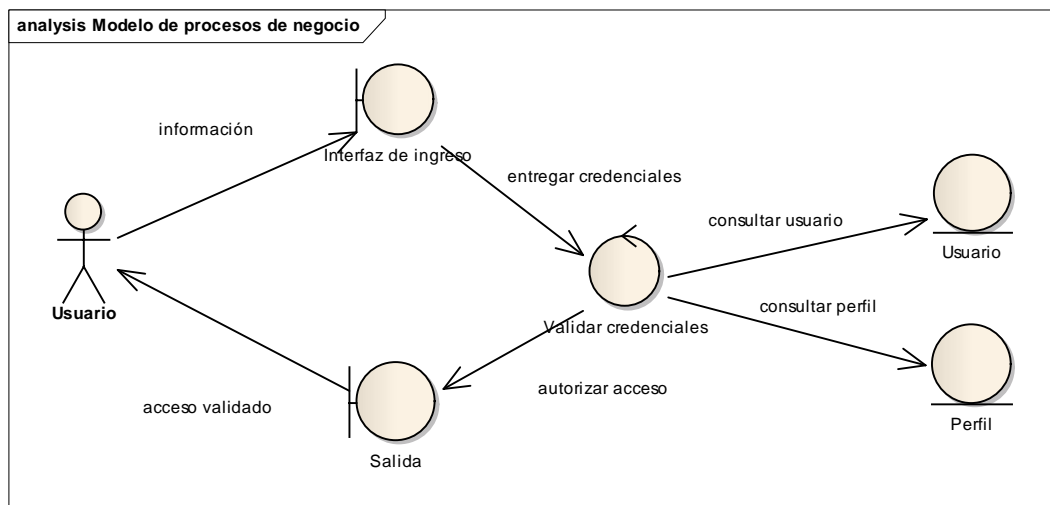


FIGURA N° 15: Diagrama de Colaboración – Requisito 1

### **3.1.3.2.2 Mantenimiento del Catastro**

**Requisito 2:** El sistema debe proveer de un servicio web que le permita al sistema LUAE-BPM registrar en CatastrosQT una solicitud de registro o de re-inspección de un establecimiento turístico (tramite LUAE-BPM). En el caso que dentro de los datos enviados por el sistema LUAE-BPM conste el número de registro turístico del establecimiento (aplica solo para establecimientos ya existentes), el sistema deberá validar adicionalmente que dicho número se encuentre registrado en el sistema, caso contrario, emitirá un mensaje de error y no se notificará al usuario inspector.

**Precondiciones:** El servicio web debe estar activo.

**Entradas:** Información básica del establecimiento.

**Flujo Normal:** El sistema debe validar si el sistema LUAE-BPM envió un número de registro turístico, de ser así se validará que el mismo esté registrado en la base de datos. Si no existe, se mostrará un mensaje de error al usuario, caso contrario registrará en la BD la información enviada por el sistema LUAE-BPM, una vez realizado esto debe notificar dicha solicitud de trámite al usuario supervisor de inspecciones.

**Salida:** Información básica del establecimiento registrada en la BD.

## Diagrama Caso de Uso

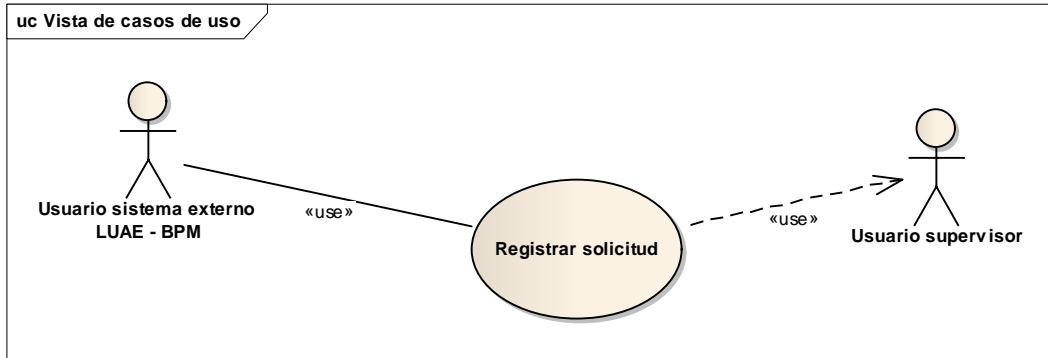


FIGURA N° 16: Diagrama de Caso de Uso – Requisito 2

## Diagrama Secuencia

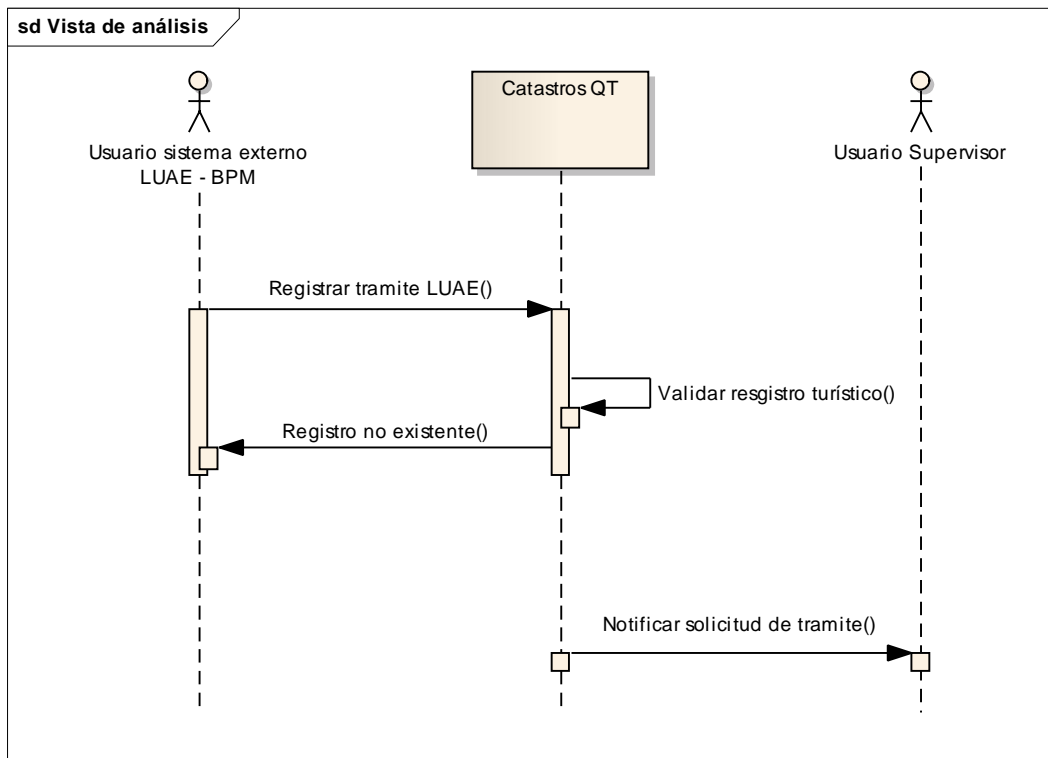
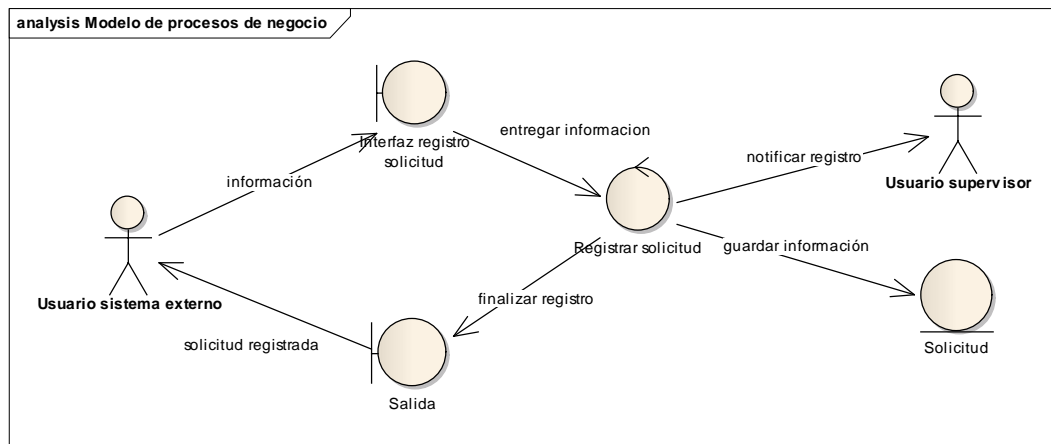


FIGURA N° 17: Diagrama de Secuencia – Requisito 2

## Diagrama de Colaboración



**FIGURA N° 18:** Diagrama de Colaboración – Requisito 2

**Requisito 3:** Independientemente de si se inició o no un trámite de registro por parte del Sistema LUAE-BPM, el sistema CatastrosQT debe permitir al Usuario Operador insertar y/o modificar la información de los establecimientos turísticos del catastro, basándose en las especificaciones que aplican a cada actividad, definidas en el documento “CATASTROS MANUAL DE PROCESOS”.

En base a la actividad, tipo y categoría del establecimiento, el sistema deberá calcular automáticamente el número de detalles del establecimiento.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema. El establecimiento a ser consultado debe haber sido registrado previamente en el sistema.

**Entradas:** Identificación, Tipo de organización, Tipo de establecimiento, Tipo de local donde funciona, Equipos disponibles, Cuántas habitaciones, camas y plazas posee (si el establecimiento es de alojamiento), Maquinaria disponible, Espacio disponible, Servicios prestados, Información de transporte turístico, Servicios complementarios, Especialidad (si el establecimiento presta servicio de alimentos

y bebidas), Cuáles son los productos más vendidos (si el establecimiento es una agencia de viajes), Cuáles son los destinos más vendidos y sus porcentajes, Número de personas que laboran en el establecimiento y Observaciones en caso de existir alguna.

**Flujo Normal:** El sistema debe validar que aquellos campos definidos como obligatorios (Número de Registro, RUC, Nombre Comercial, Propietario/Razón Social, Actividad y Calle Principal) se hayan ingresado correctamente y sean válidos. En caso de no serlo, el proceso no continuará.

El sistema debe determinar el número de detalles que correspondan a un establecimiento en base a su actividad, tipo y categoría. Para el caso de los establecimientos de Alojamiento, se debe sumar las habitaciones BC y CBP. Para el caso de los establecimientos de Alimentos y Bebidas, será igual al número de mesas. Para el resto de actividades, el número de detalles será de uno.

Una vez realizado esto, el sistema debe registrar el establecimiento nuevo o actualizarlo en caso de ya estar registrado en la BD.

**Salida:** Información del establecimiento registrada/actualizada exitosamente.

### Diagrama Caso de Uso

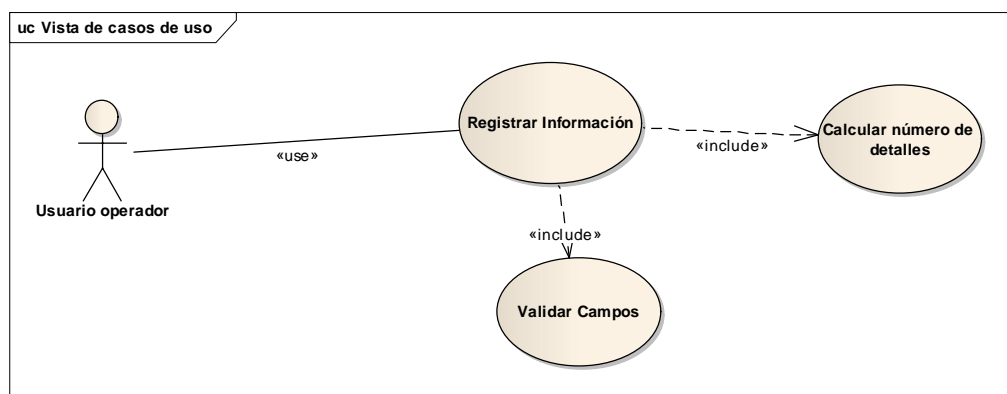


FIGURA N° 19: Diagrama de Caso de Uso – Requisito 3



## Diagrama de Secuencia

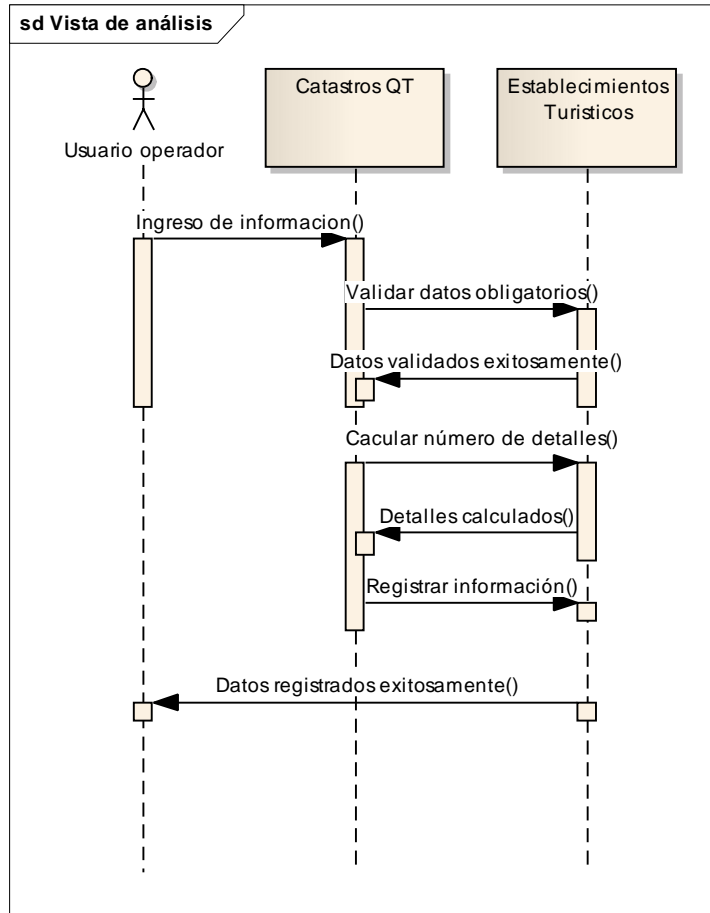


FIGURA N° 20: Diagrama de Secuencia – Requisito 3

## Diagrama de Colaboración

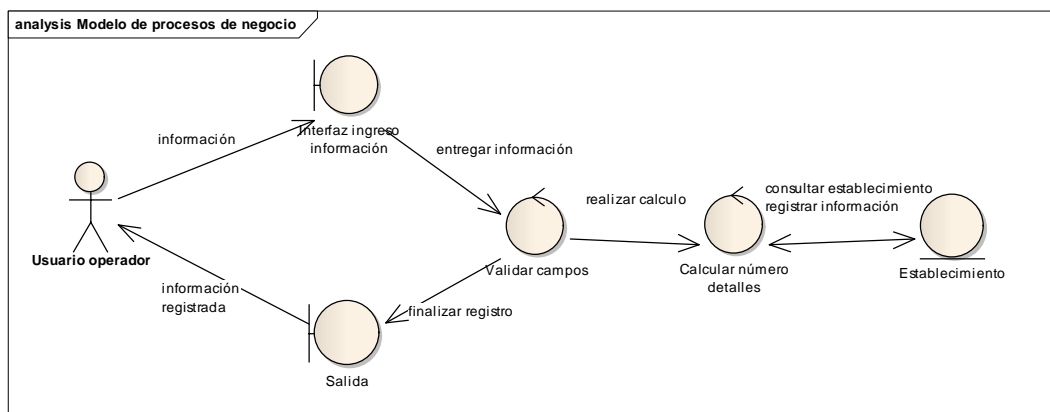


FIGURA N° 21: Diagrama de Colaboración – Requisito 3

## Diagrama de Estados

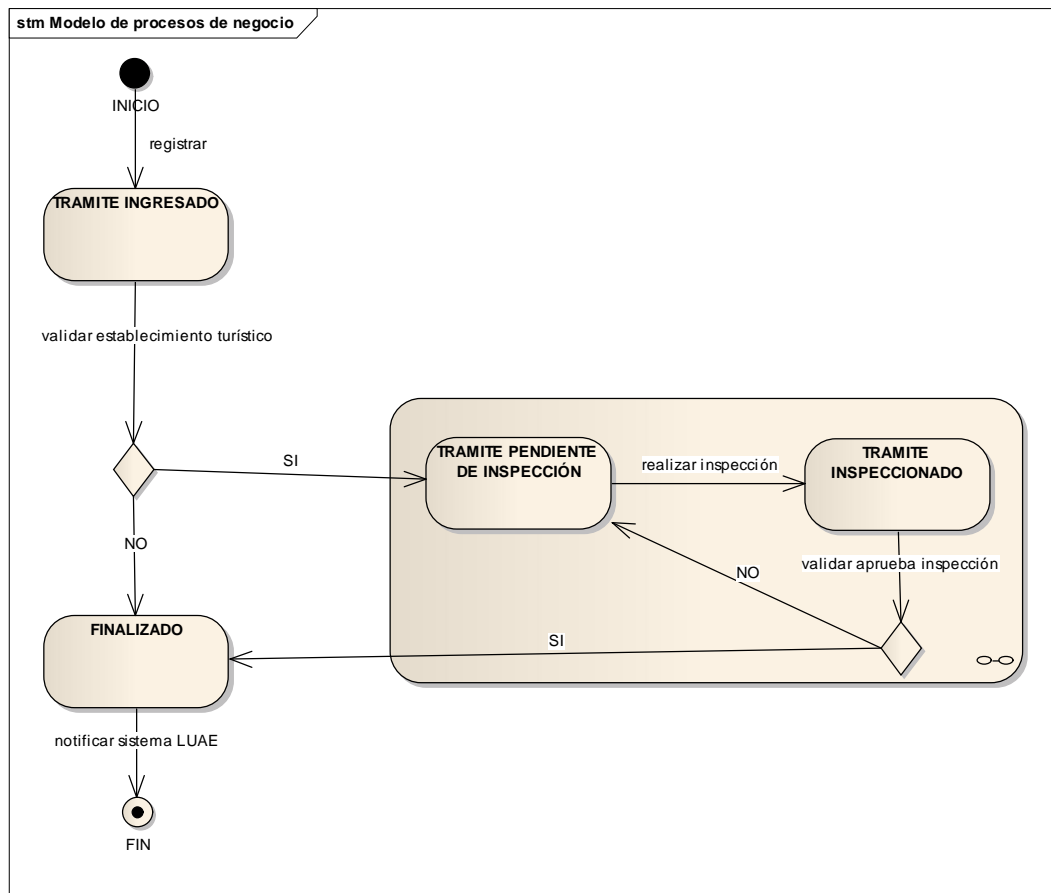


FIGURA N° 22: Diagrama de Estados – Requisito 3

**Requisito 4:** El sistema deberá permitir consultar los establecimientos registrados en base a cualquiera de los siguientes criterios: Nombre, Número de Registro, Representante Legal, RUC o por Número de Trámite. En caso de desconocer cualquiera de estos criterios, el sistema debe permitir consultar los establecimientos que pertenezcan a una determinada Actividad y/o Tipo y/o Categoría.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Nombre, Número de Registro, Representante Legal, RUC, Número de Trámite.

**Flujo Normal:** El sistema debe consultar en la BD y obtener el listado de todos los establecimientos registrados que coincidan con el criterio de búsqueda seleccionado.

**Flujo Alternativo:** El sistema permitirá consultar los establecimientos que pertenezcan a una determinada Actividad, Tipo y/o Categoría, en base a la selección de dichas propiedades de un control de Vista de Árbol.

**Salida:** Listado de establecimientos consultados.

### Diagrama Caso de Uso

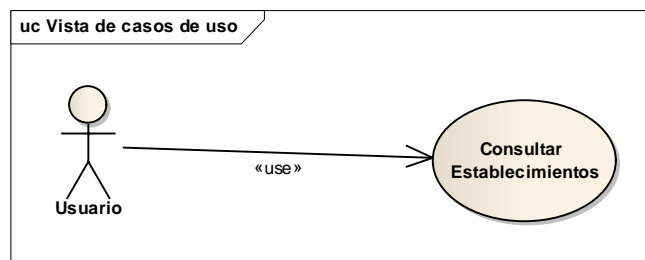


FIGURA N° 23: Diagrama de Caso de Uso – Requisito 4

### Diagrama de Secuencia

**Flujo Normal:**

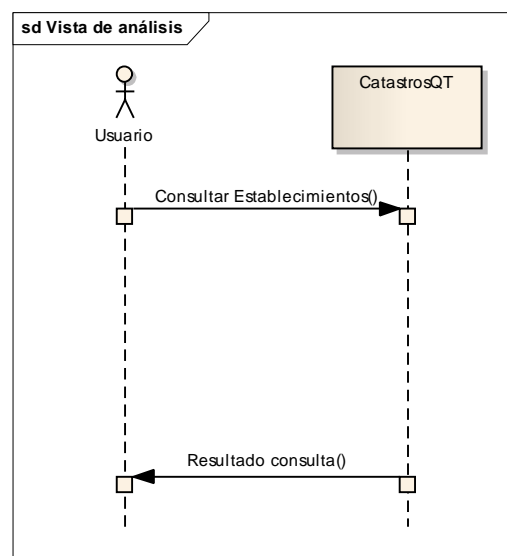


FIGURA N° 24: Diagrama de Secuencia (Flujo Normal) – Requisito 4

## Flujo Alternativo:

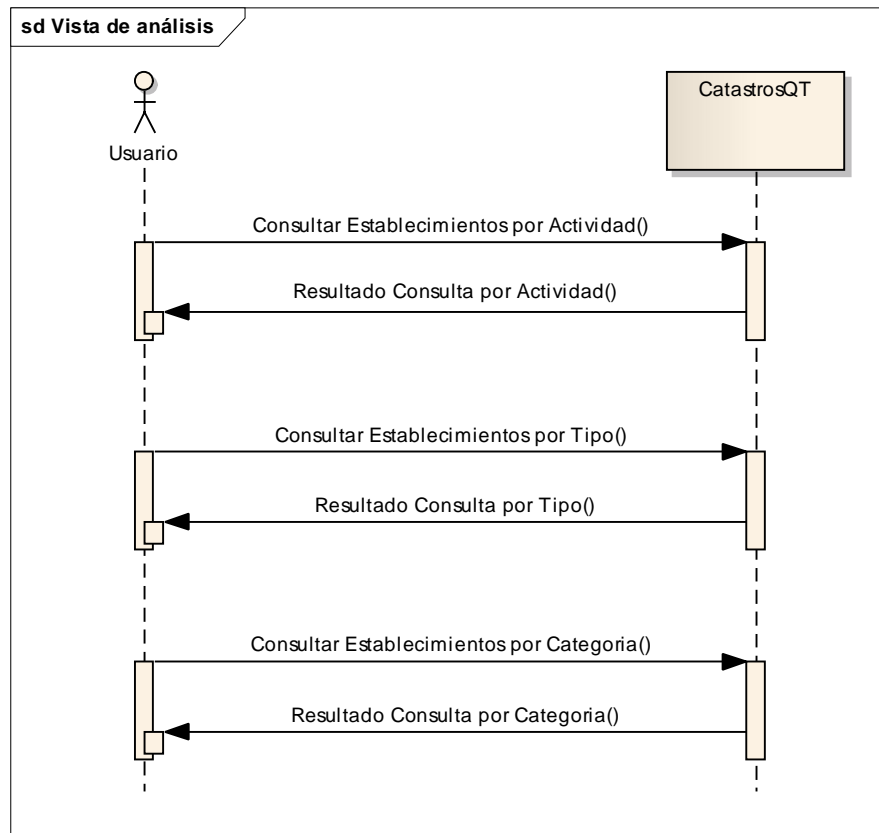


FIGURA N° 25: Diagrama de Caso de Uso (Flujo Alternativo) – Requisito 4

## Diagrama de Colaboración

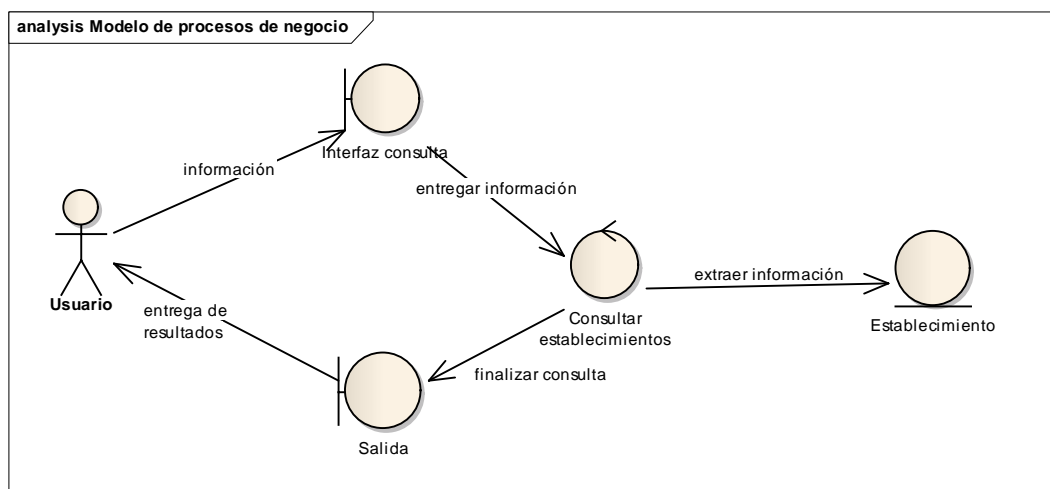


FIGURA N° 26: Diagrama de Colaboración – Requisito 4

**Requisito 5:** El sistema debe proveer de un servicio web que le permita al sistema LUAE-BPM consultar el estado del trámite registrado anteriormente en la BD de CatastrosQT.

**Precondiciones:** El servicio web debe encontrarse activo.

**Entradas:** Número de trámite

**Flujo Normal:** El sistema debe obtener de la BD la información del estatus actual del trámite generado desde LUAE-BPM, así como el usuario asignado al mismo.

**Salida:** Información del trámite consultado.

### Diagrama Caso de Uso

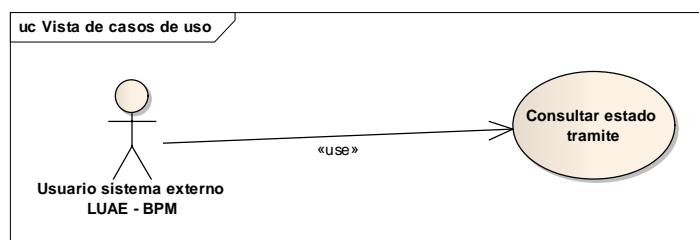


FIGURA N° 27: Diagrama de Caso de Uso – Requisito 5

### Diagrama de Secuencia

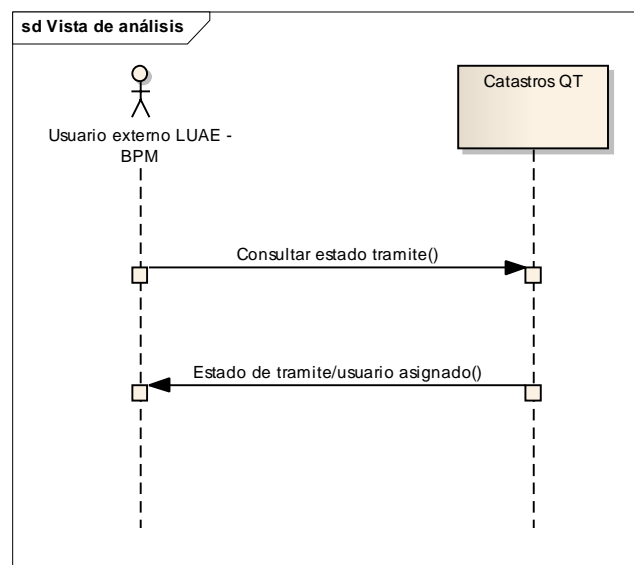


FIGURA N° 28: Diagrama de Secuencia – Requisito 5

## Diagrama de Colaboración

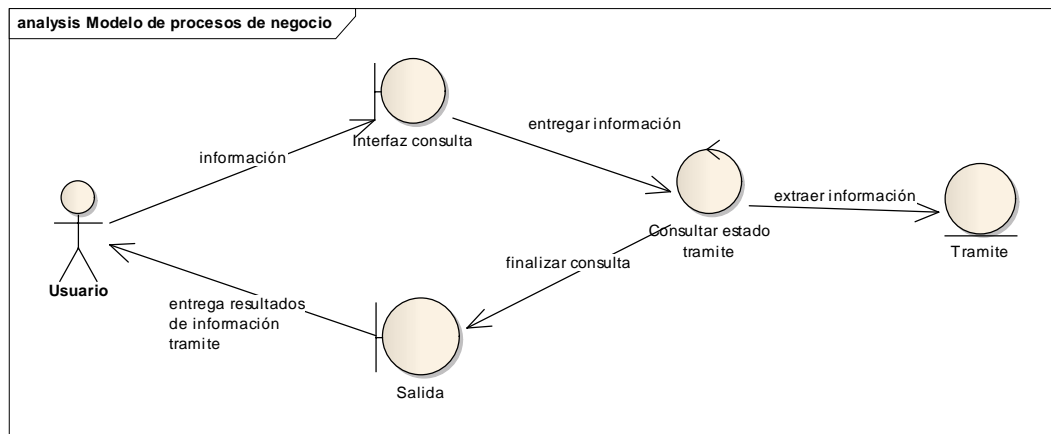


FIGURA N° 29: Diagrama de Colaboración – Requisito 5

### 3.1.3.2.3 Mantenimiento de Recaudaciones

**Requisito 6:** El sistema debe permitir a un Usuario de Recaudaciones registrar y/o modificar la Emisión de valores que un establecimiento debe pagar con periodicidad anual, siempre y cuando el mismo se encuentre en estado ACTIVO. En caso de que el establecimiento posea valores emitidos anteriormente para el año seleccionado, el sistema le permitirá ingresar una nueva emisión, siempre y cuando el valor no exceda el máximo para la actividad, tipo y categoría del establecimiento.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema. El usuario debe haber seleccionado el establecimiento al cual desea emitir el valor.

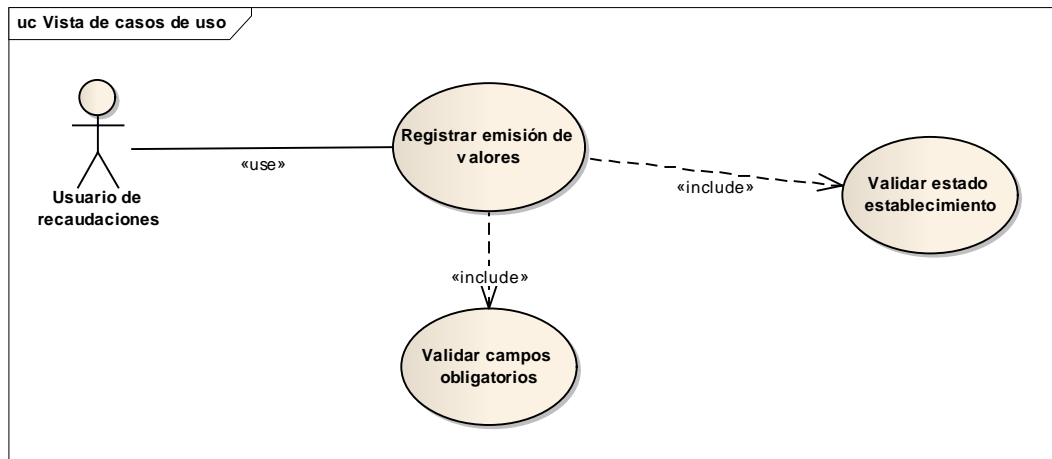
**Entradas:** Año, Fecha de Emisión, Patente Municipal, Valor correspondiente a: Turismo, CAPTUR, Bomberos y Medio Ambiente.

**Flujo Normal:** El sistema debe validar que el estado del establecimiento sea ACTIVO antes de continuar. Adicionalmente se debe verificar que los valores ingresados sean válidos y que para el caso de la tasa de Turismo sea mayor a cero.

El sistema debe validar si existen emisiones anteriores para el establecimiento en el año seleccionado, de ser el caso, debe validar que con la nueva emisión no se exceda el valor límite impuesto para la Actividad, Tipo y Categoría del establecimiento. En caso de que con la nueva emisión se sobrepase dicho valor, deberá emitir un mensaje de error alertando sobre este tema, caso contrario, registrará la información correspondiente.

**Salida:** Información de la emisión registrada/actualizada en la BD.

### Diagrama Caso de Uso



**FIGURA N° 30:** Diagrama de Caso de Uso – Requisito 6

## Diagrama de Secuencia

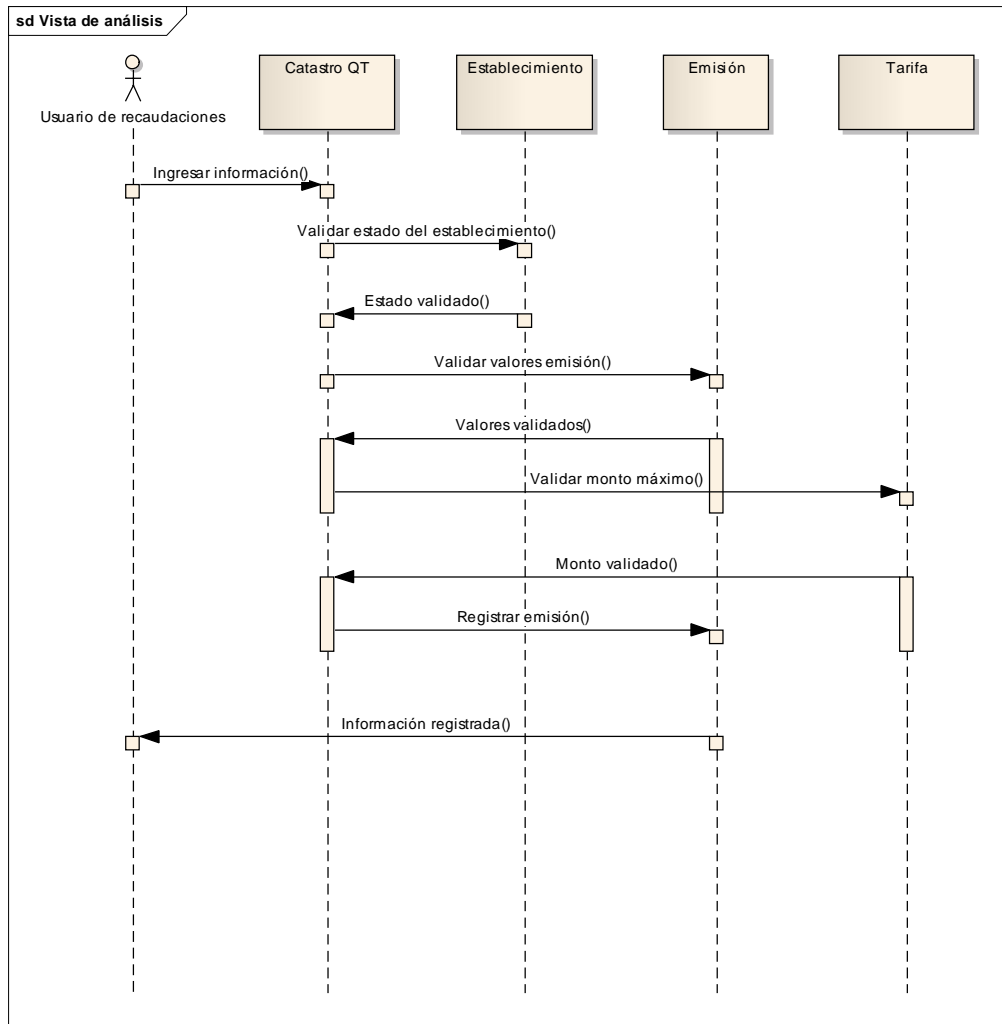


FIGURA N° 31: Diagrama de Secuencia – Requisito 6

## Diagrama de Colaboración

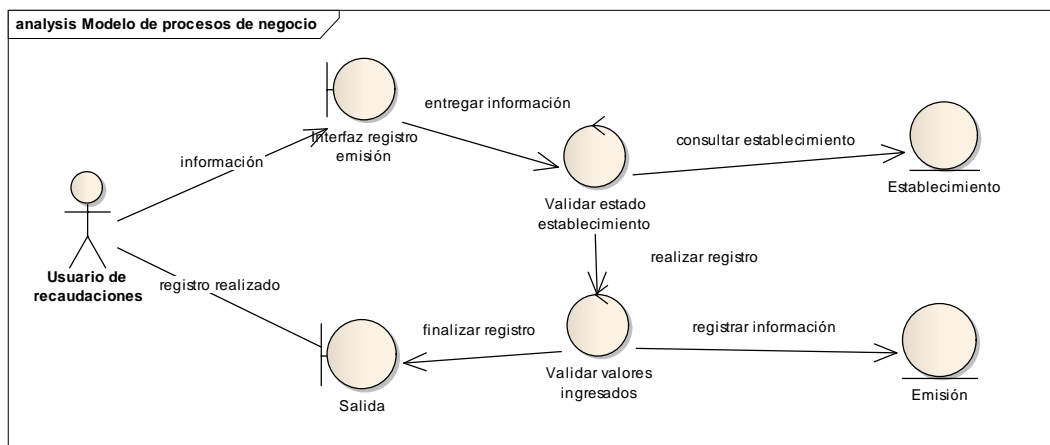


FIGURA N° 32: Diagrama de Colaboración – Requisito 6



**Requisito 7:** El sistema debe permitir a un Usuario de Recaudaciones, registrar y/o modificar el Pago correspondiente a una Emisión de valores registrada anteriormente. El valor del pago a registrar puede ser parcial o total con respecto al valor de la emisión.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema. Se debe haber registrado con anterioridad una emisión de valores para el establecimiento requerido.

**Entradas:** Año, Emisión a pagar, Fecha de Pago, Número de Título de Crédito, Valor correspondiente a: Turismo, CAPTUR, Bomberos y Medio Ambiente.

**Flujo Normal:** El sistema debe validar que el estado del establecimiento sea ACTIVO antes de continuar. Adicionalmente se debe verificar que los valores ingresados sean válidos y que para el caso de la tasa de Turismo sea mayor a cero. En caso de que la emisión registre pagos parciales anteriores, se debe cerciorar que junto al valor actual de pago no sobrepasa el valor total de la emisión. Finalmente se registra en la BD dichos valores.

**Salida:** Información del pago registrada/actualizada en la BD.

### Diagrama Caso de Uso

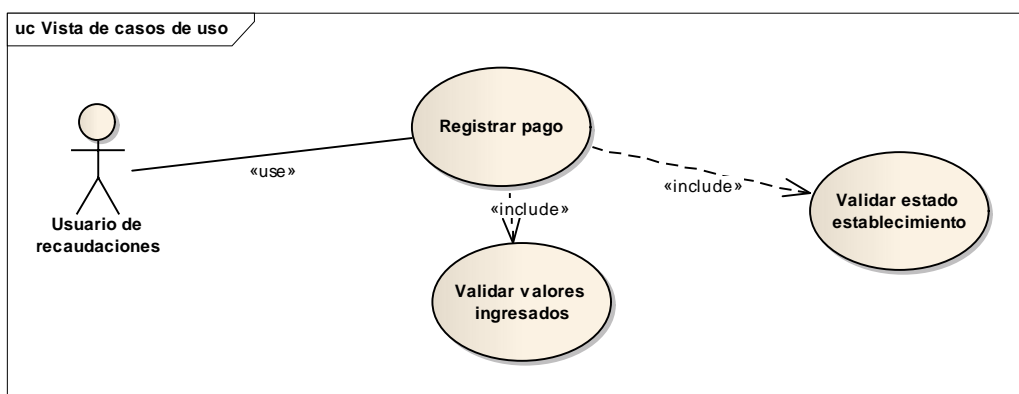


FIGURA N° 33: Diagrama de Caso de Uso – Requisito 7

## Diagrama de Secuencia

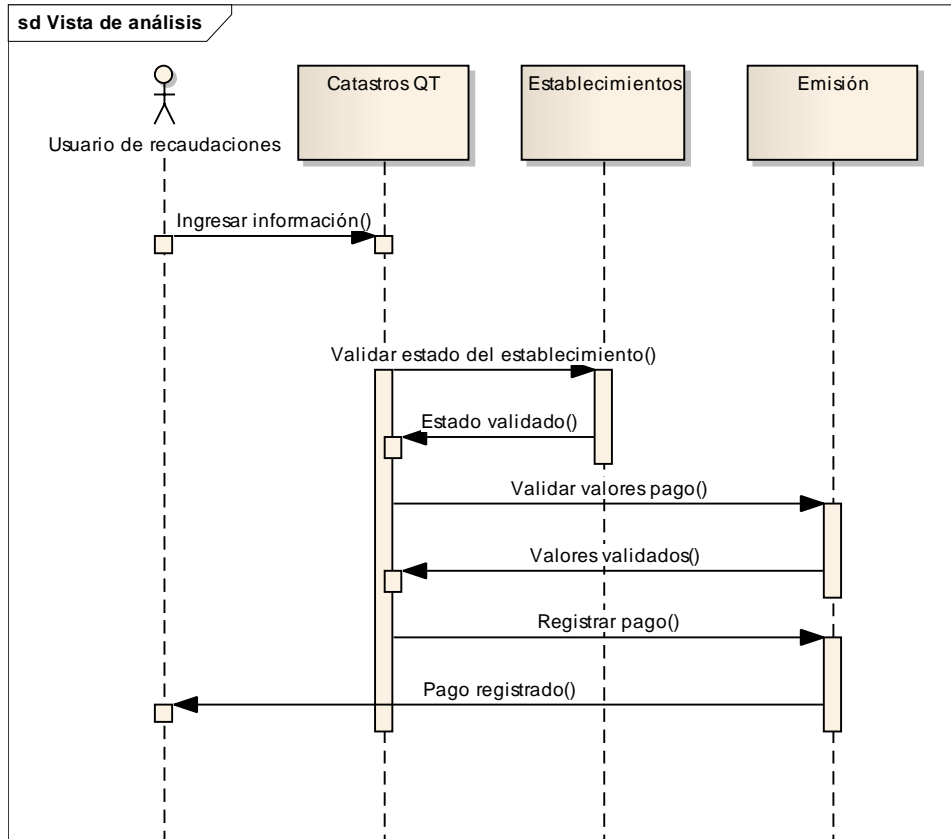


FIGURA N° 34: Diagrama de Secuencia – Requisito 7

## Diagrama de Colaboración

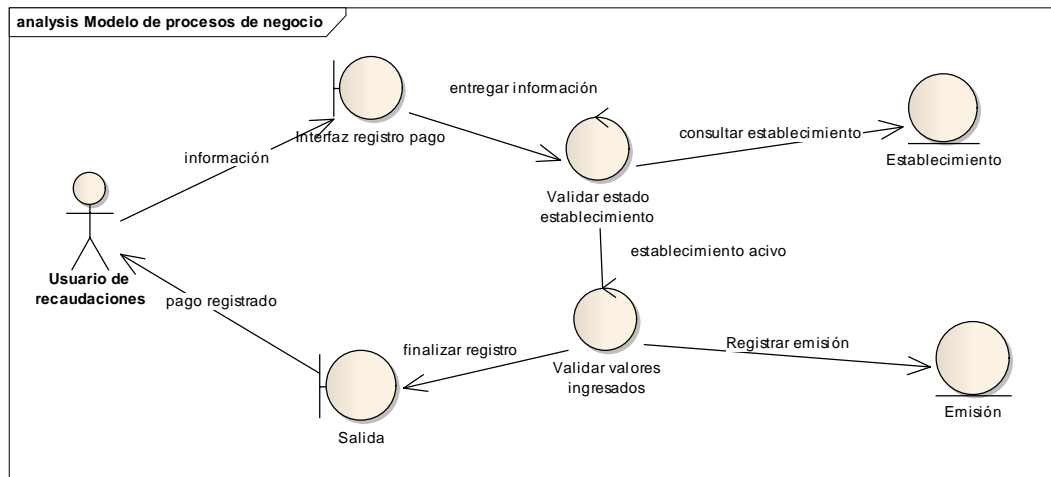


FIGURA N° 35: Diagrama de Colaboración – Requisito 7

**Requisito 8:** El sistema debe permitir a un Usuario de Recaudaciones, registrar y/o modificar la Baja correspondiente a una Emisión de valores registrada anteriormente.

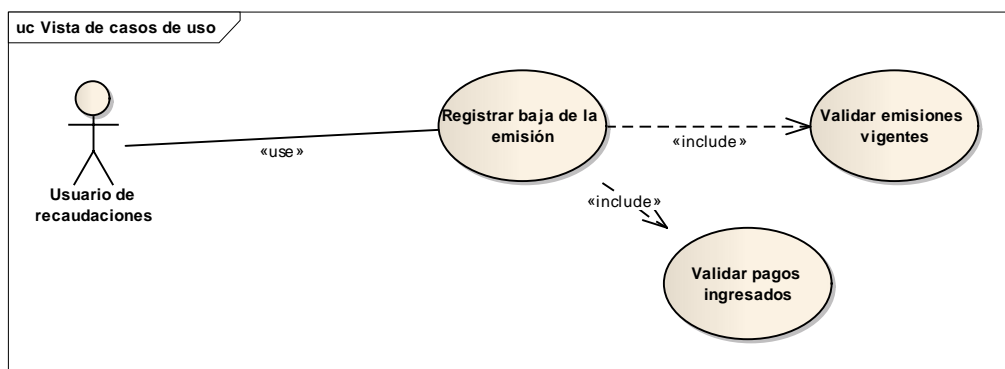
**Precondiciones:** El usuario debe haber ingresado previamente al sistema. Se debe haber registrado con anterioridad una emisión de valores para el establecimiento requerido.

**Entradas:** Año, Emisión a dar de Baja, Fecha de Baja y Numero de Resolución.

**Flujo Normal:** El sistema debe validar que para el año seleccionado existan emisiones vigentes. En caso de que la emisión registre pagos parciales o totales anteriores, no se debe permitir dar de baja la emisión. Finalmente se registra en la BD la baja de la emisión.

**Salida:** Información de la baja de una emisión registrada/actualizada en la BD.

### Diagrama Caso de Uso



**FIGURA N° 36:** Diagrama de Caso de Uso – Requisito 8

## Diagrama de Secuencia

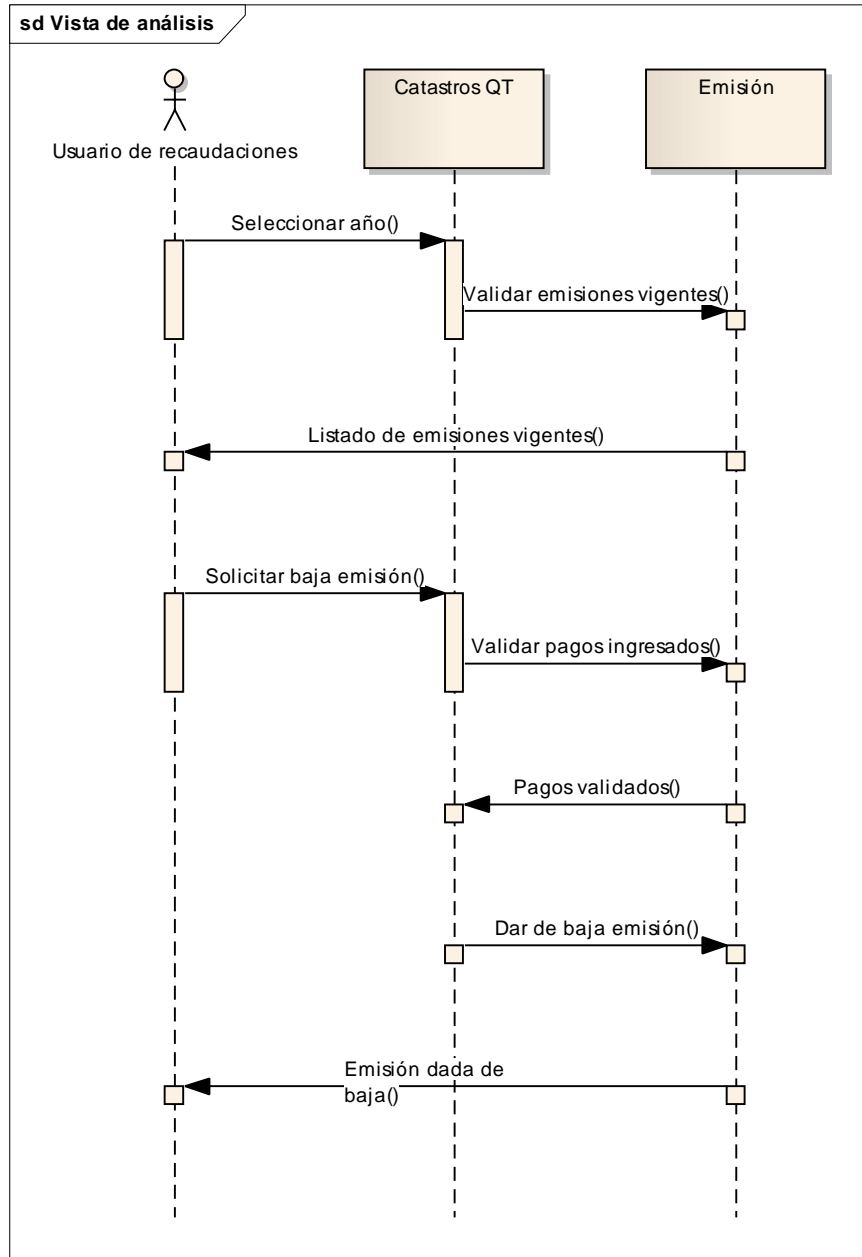


FIGURA N° 37: Diagrama de Secuencia – Requisito 8

## Diagrama de Colaboración

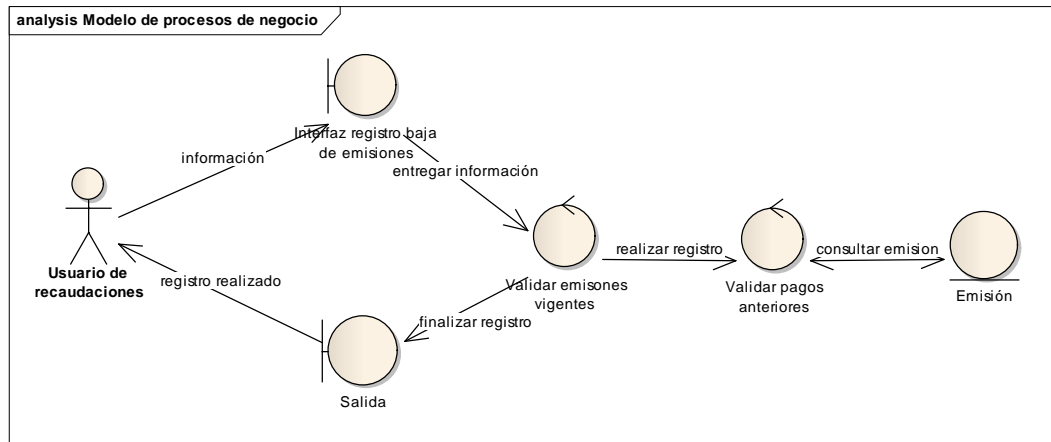


FIGURA N° 38: Diagrama de Colaboración – Requisito 8

### 3.1.3.2.4 Mantenimiento de Inspecciones

**Requisito 9:** El sistema debe permitir al Usuario Inspector, registrar el formulario de inspección que corresponda a un establecimiento del catastro en base a su actividad turística. Los formularios que aplican a cada actividad se listan en la siguiente tabla:

| FORMULARIOS                               | ACTIVIDADES |                     |                           |                   |                      |
|---|-------------|---------------------|---------------------------|-------------------|----------------------|
|   | ALOJAMIENTO | ALIMENTOS Y BEBIDAS | DIVERSION Y ESPARCIMIENTO | AGENCIAS DE VIAJE | TRANSPORTE TURISTICO |
| FORMULARIO DE CONTROL DE INCENDIOS        | X           | X                   | X                         | X                 | X                    |
| FORMULARIO DE CONTROL SANITARIO           |             | X                   |                           |                   |                      |
| FORMULARIO DE GUÍAS PRÁCTICAS AMBIENTALES | X           | X                   | X                         |                   |                      |

TABLA N° 9: Formularios de Inspección de acuerdo a cada Actividad

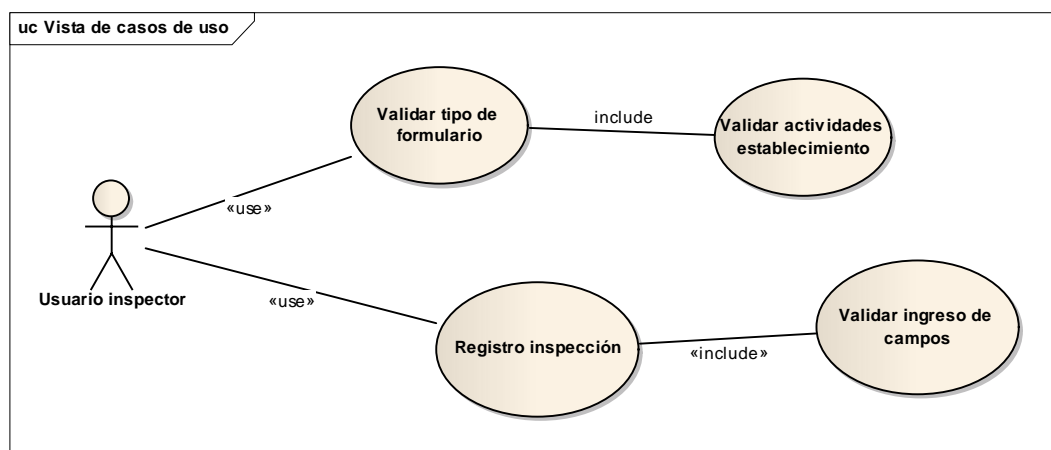
**Precondiciones:** El usuario debe haber ingresado previamente al sistema. Se debe haber registrado con anterioridad la información del establecimiento.

**Entradas:** Número de Registro del establecimiento, Año al que aplica la Inspección, Ítems del Formulario que aplique, Aprueba Inspección, Fecha de Inspección, Hora de Inspección, Fecha de Próxima Inspección, Observaciones, Nombre Inspector y Responsable del Establecimiento.

**Flujo Normal:** En base a la actividad del establecimiento, el sistema debe determinar el(los) formulario(s) de inspección que se debe(n) aplicar. Además, el sistema debe validar que todos los ítems de la inspección hayan sido registrados con su correspondiente calificación. Finalmente se registra en la BD la inspección junto con todos los formularios correspondientes.

**Salida:** Inspección registrada en la BD.

### Diagrama Caso de Uso



**FIGURA N° 39:** Diagrama de Caso de Uso – Requisito 9

## Diagrama de Secuencia

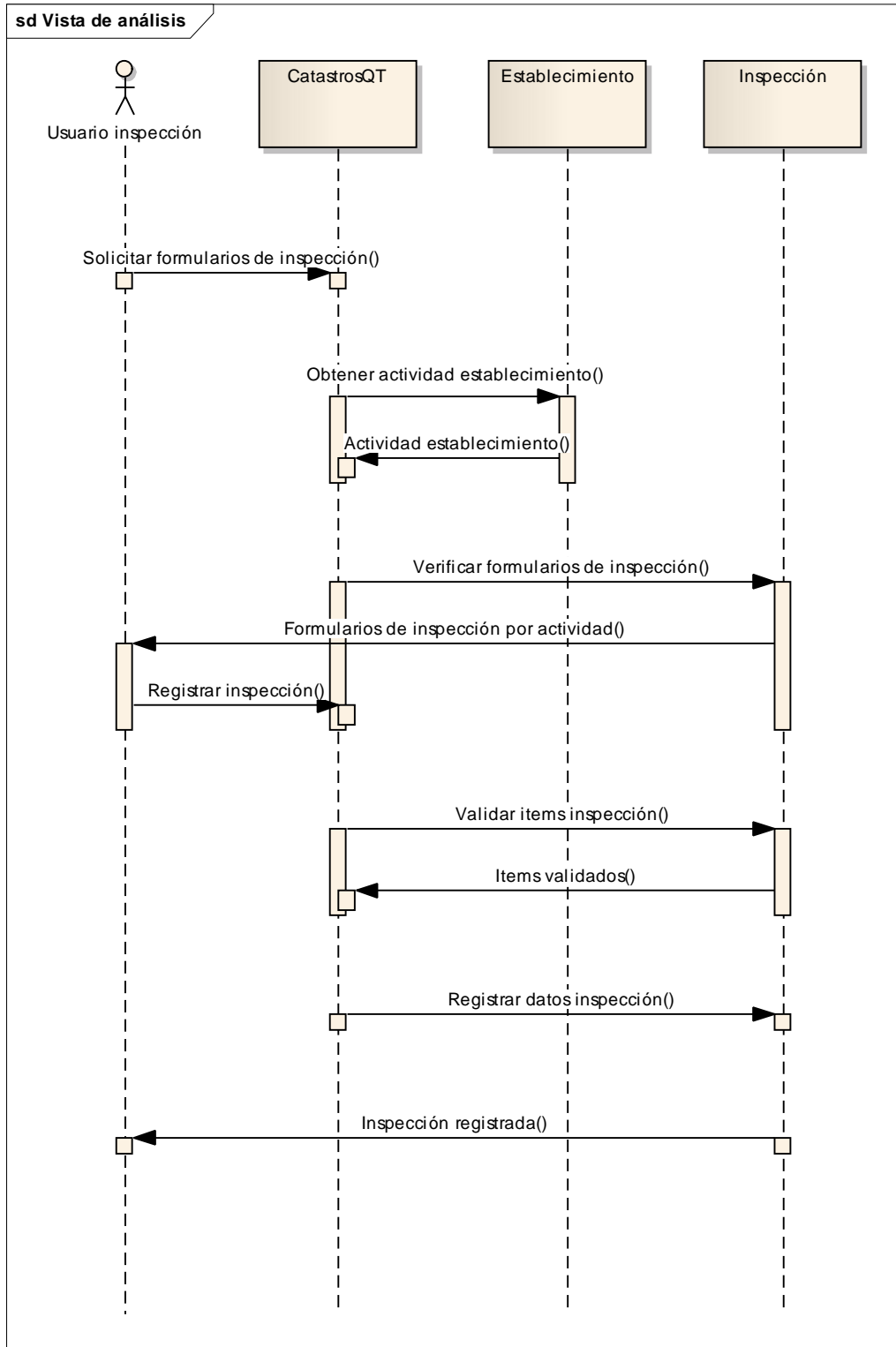


FIGURA N° 40: Diagrama de Secuencia – Requisito 9

## Diagrama de Colaboración

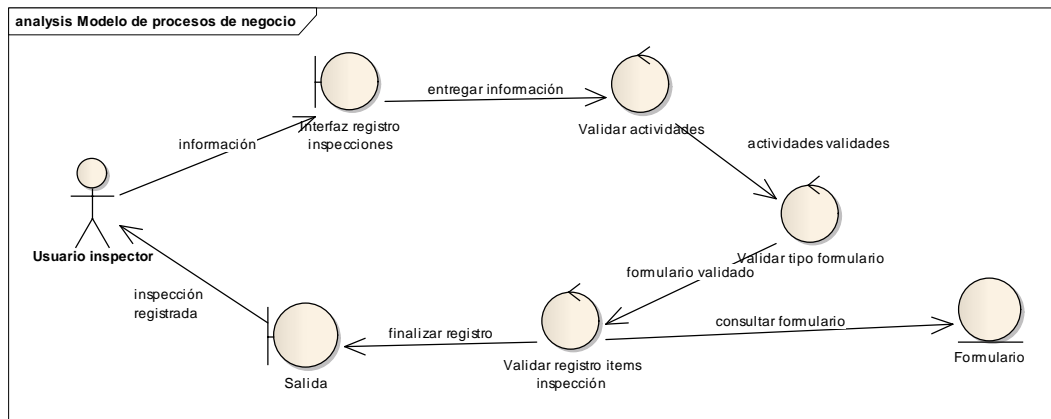


FIGURA N° 41: Diagrama de Colaboración – Requisito 9

**Requisito 10:** El sistema debe permitir al Usuario Supervisor de Inspecciones, eliminar el último formulario de inspección registrado para un establecimiento anteriormente en el sistema. En caso de que la inspección que se requiera eliminar haya estado relacionada con un Trámite LUAE-BPM, el sistema no deberá permitir eliminar dicha inspección siempre y cuando esta haya sido EXITOSA.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema. Se debe haber registrado con anterioridad al menos una inspección para el establecimiento requerido.

**Entradas:** Número de Registro del establecimiento, Número de Inspección.

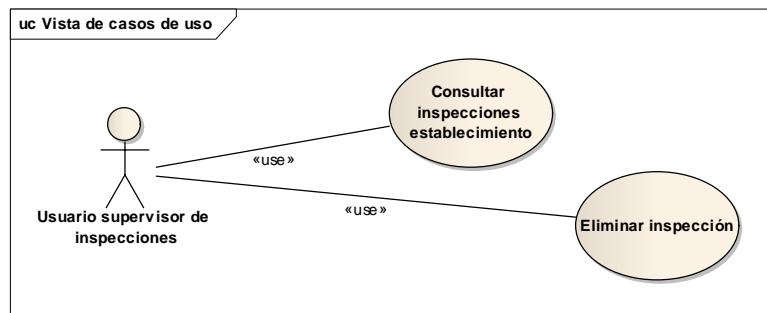
**Flujo Normal:** El usuario debe consultar en el sistema el establecimiento para el cual desea eliminar la inspección. Luego de esto, Debe seleccionar el año para el cual esta registrada la inspección. Al usuario se le mostrará el formulario de inspección más reciente registrado para el establecimiento y año seleccionados. Al seleccionar la opción de “Eliminar Inspección”, la misma se debe registrar con estado ELIMINADO.



**Flujo Alternativo:** El sistema validará si la inspección a eliminar se encuentra relacionada con un Trámite LUAE-BPM, de ser así, no permitirá dar de baja dicha inspección si la misma se encuentra en estado EXITOSA.

**Salida:** Inspección eliminada en la BD.

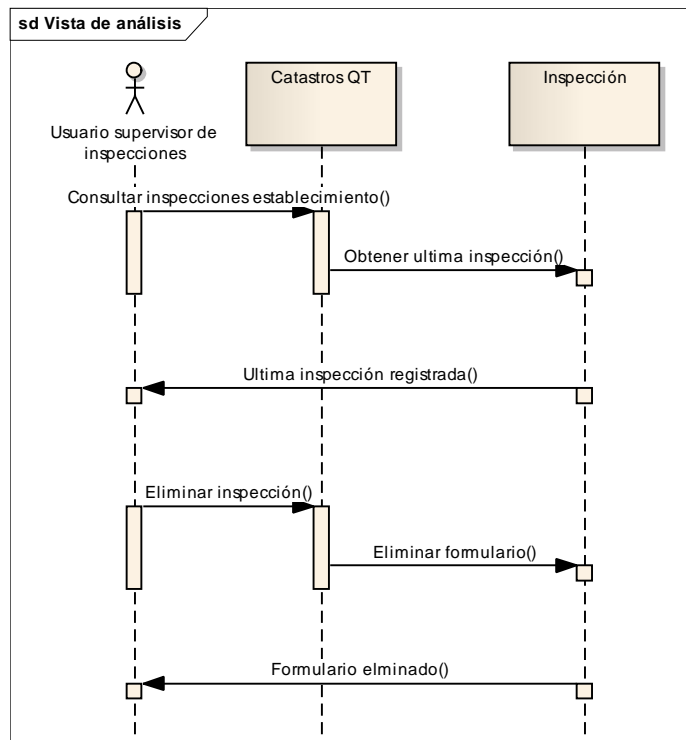
**Diagrama Caso de Uso**



**FIGURA N° 42:** Diagrama de Caso de Uso – Requisito 10

**Diagrama de Secuencia**

**Flujo Normal:**



**FIGURA N° 43:** Diagrama de Secuencia (Flujo Normal) – Requisito 10

### Flujo Alternativo:

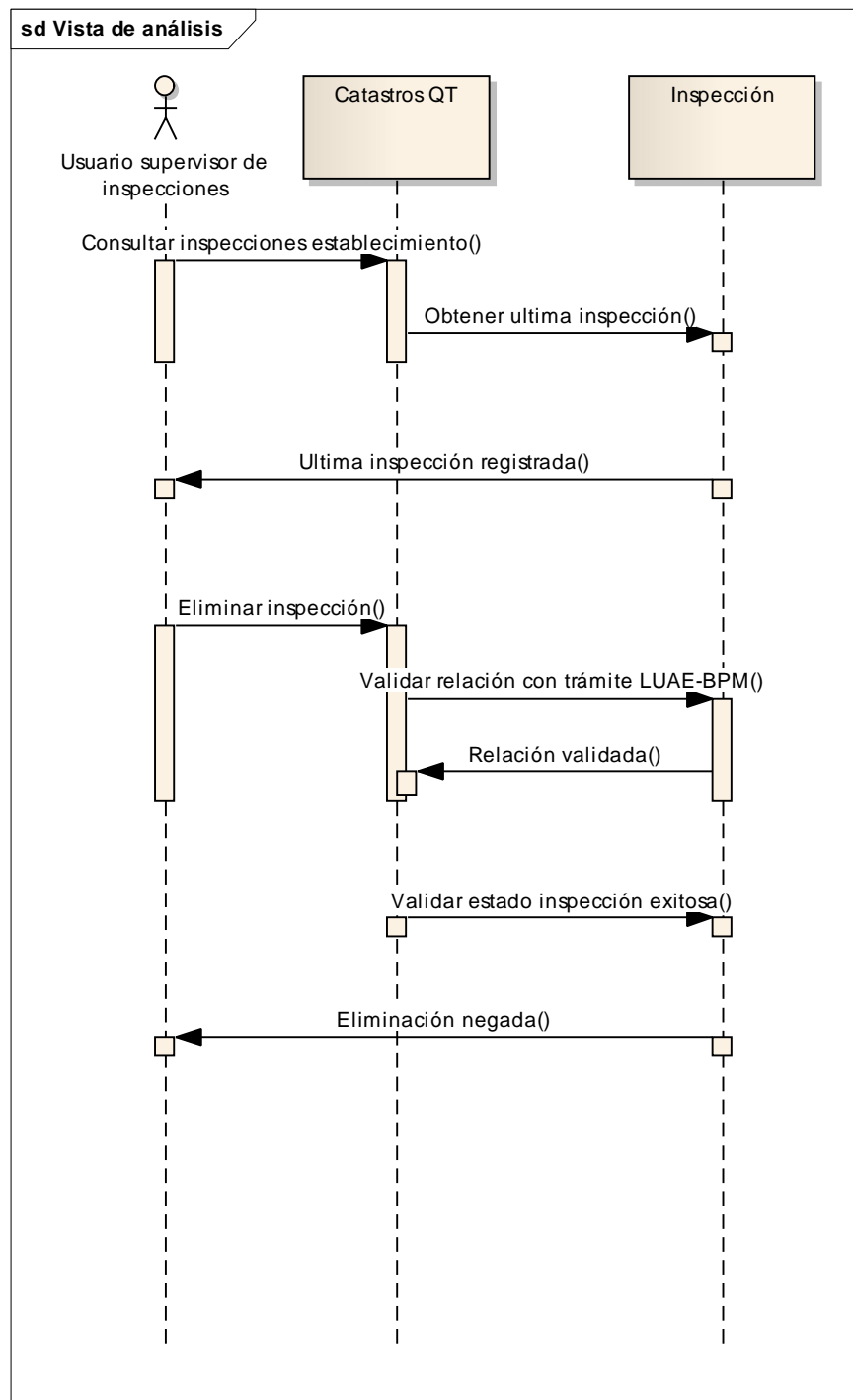


FIGURA N° 44: Diagrama de Secuencia (Flujo Alternativo) – Requisito 10

## Diagrama de Colaboración

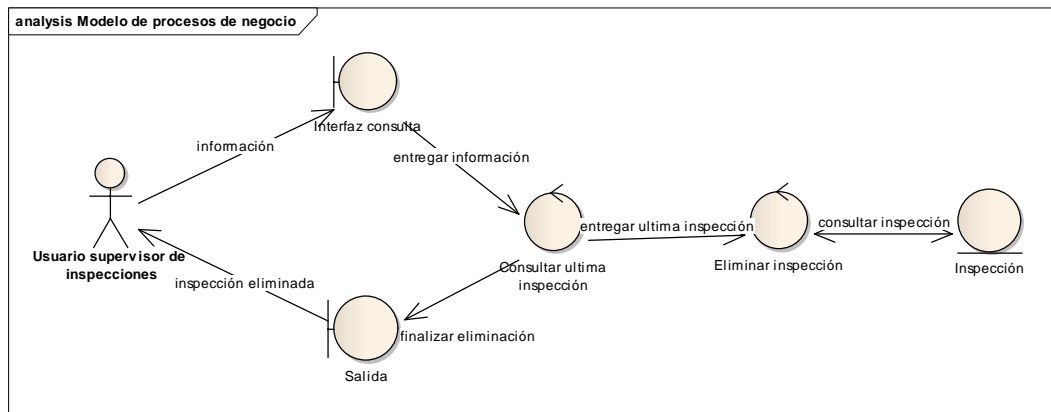


FIGURA N° 45: Diagrama de Colaboración – Requisito 10

**Requisito 11:** El sistema debe permitir al Usuario Operador, imprimir la LUAF, en el formato establecido por la EPMGDT para aquellos establecimientos que han aprobado la Inspección para un determinado año, siempre y cuando el establecimiento se encuentre al día en el pago de los valores emitidos.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema. Se debe haber registrado con anterioridad al menos una inspección para el establecimiento requerido.

**Entradas:** Número de Registro del establecimiento, Año, Persona que solicita, Identificación del solicitante, Lugar de impresión y Observaciones.

**Flujo Normal:** El sistema debe validar que el establecimiento haya aprobado la inspección para el año solicitado y que se encuentre al día en el pago de sus emisiones. A continuación, debe registrar en la BD la información de solicitud de impresión de la LUAF, generando dicho documento en el formato especificado e imprimiéndolo automáticamente.

## Diagrama Caso de Uso

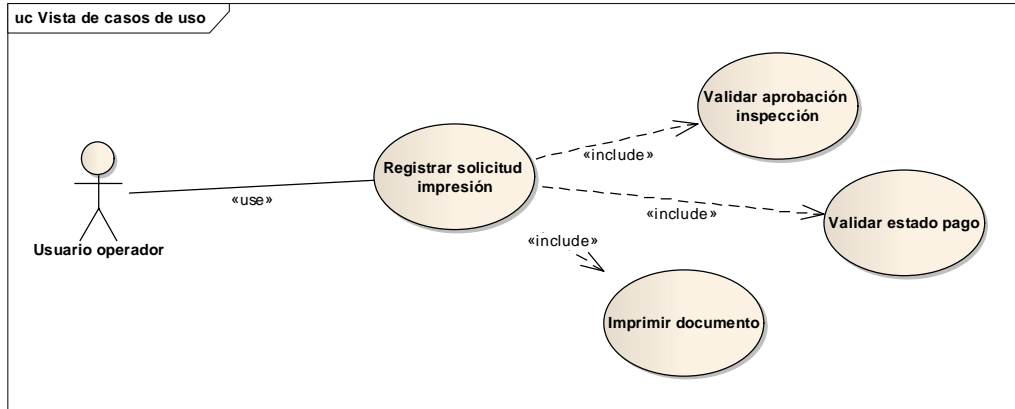


FIGURA N° 46: Diagrama de Caso de Uso – Requisito 11

## Diagrama de Secuencia

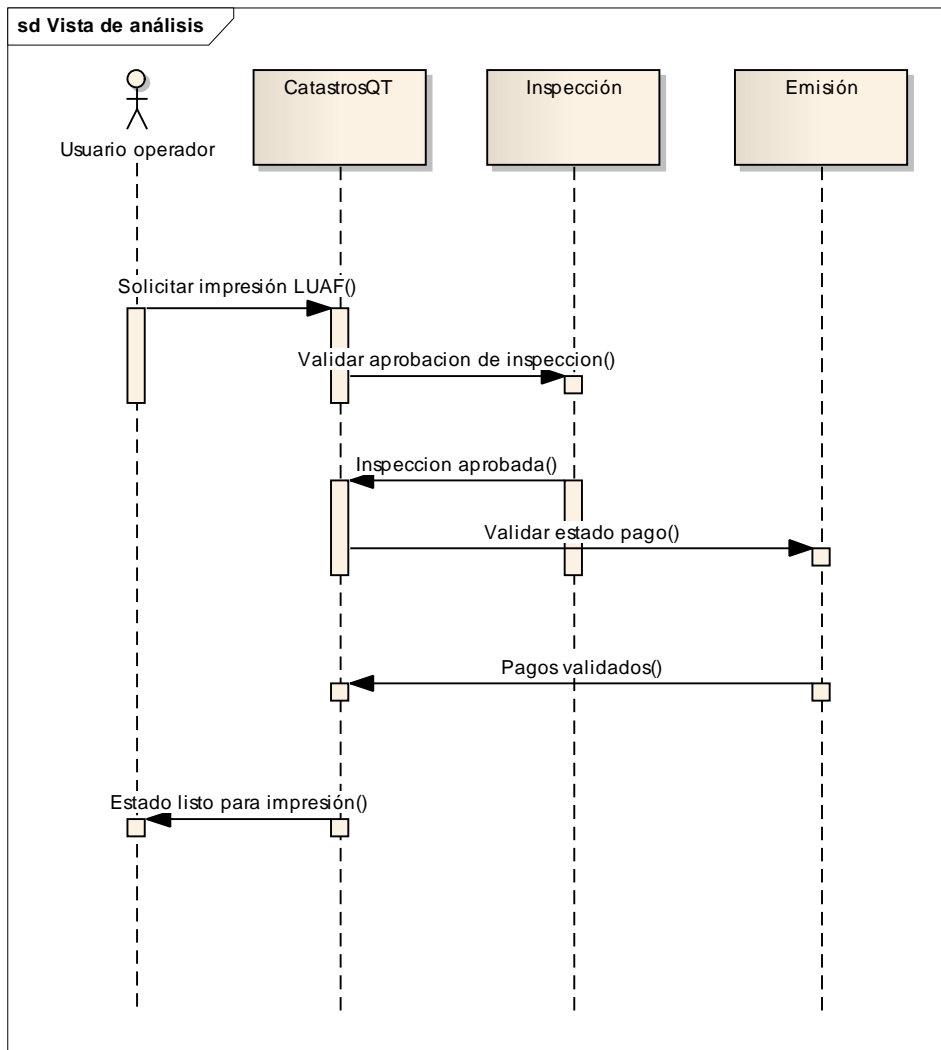
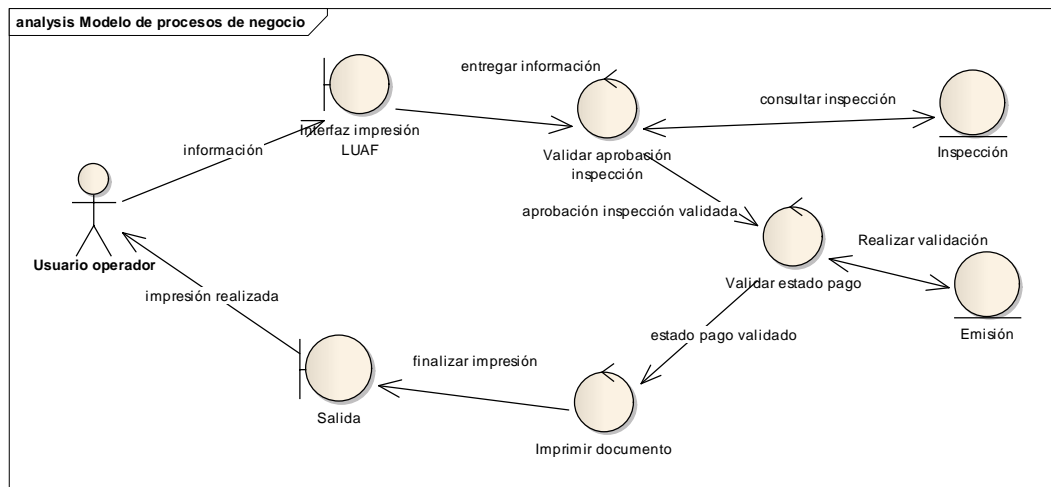


FIGURA N° 47: Diagrama de Secuencia – Requisito 11

## Diagrama de Colaboración



**FIGURA N° 48:** Diagrama de Colaboración – Requisito 11

**Requisito 12:** Una vez cumplido el proceso de inspección para un trámite generado desde el sistema LUAE-BPM, el sistema debe invocar al servicio web provisto por LUAE-BPM para registrar en dicho sistema que el trámite ha concluido y el resultado del mismo sea que haya sido exitosa o no la inspección.

**Precondiciones:** Se debe haber culminado el registro de inspecciones para un establecimiento determinado.

**Entradas:** Número de Trámite, Resultado de la Inspección.

**Flujo Normal:** El sistema debe ejecutar el servicio web de LUAE-BPM de manera automática, una vez que el establecimiento ha finalizado el proceso de inspección. En caso de recibir error, el sistema reintentará de manera automática ejecutar nuevamente el servicio web luego de un período de tiempo determinado.

**Salida:** Servicio web de LUAE-BPM ejecutado.

## Diagrama Caso de Uso

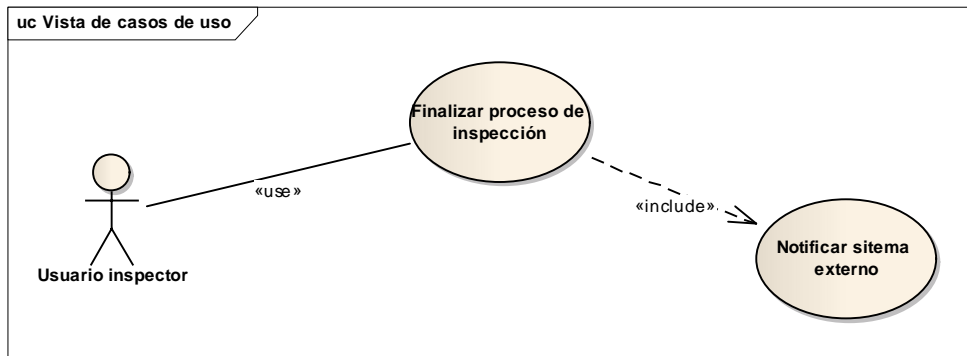


FIGURA N° 49: Diagrama de Caso de Uso – Requisito 12

## Diagrama de Secuencia

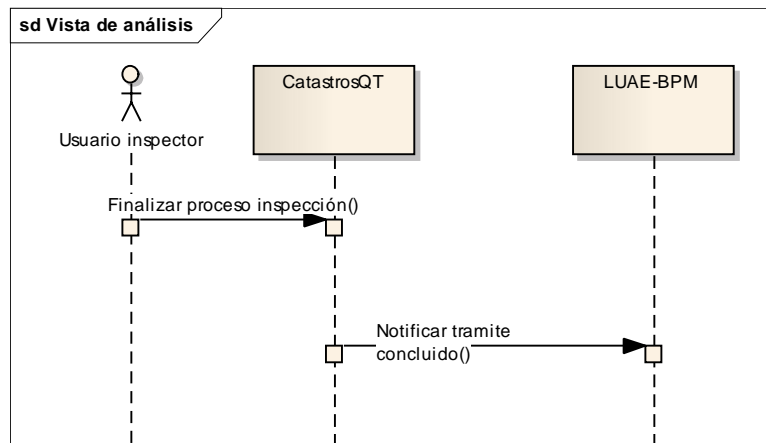


FIGURA N° 50: Diagrama de Secuencia – Requisito 12

## Diagrama de Colaboración

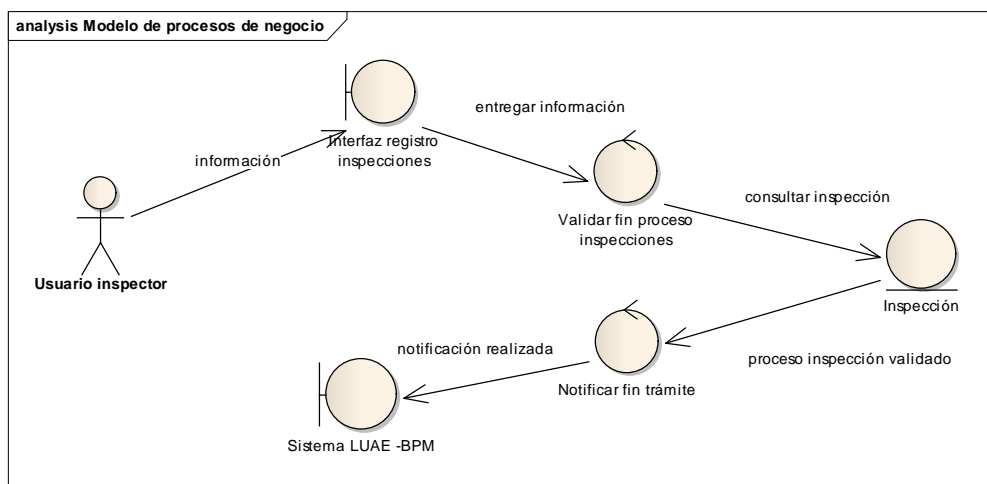


FIGURA N° 51: Diagrama de Colaboración – Requisito 12

### 3.1.3.2.5 Generación de Reportes

**Requisito 13:** El sistema debe proveer de un esquema de reportes que permita al Usuario Consulta obtener la información registrada en el aplicativo de los Establecimientos del catastro registrados, permitiendo además filtrar dichos reportes por Fecha Inicio, Fecha Hasta, Actividad, Tipo, Categoría, Administración Zonal, Sector Turístico y Parroquia. Adicionalmente, se debe permitir exportar la información del reporte a Excel.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Tipo de Reporte, Fecha Inicio, Fecha Hasta, Actividad, Tipo, Categoría, Administración Zonal, Sector Turístico y Parroquia.

**Flujo Normal:** El sistema debe extraer la información de la BD en base al tipo de reporte seleccionado y los filtros seleccionados.

**Salida:** Información correspondiente al reporte solicitado.

#### Diagrama Caso de Uso

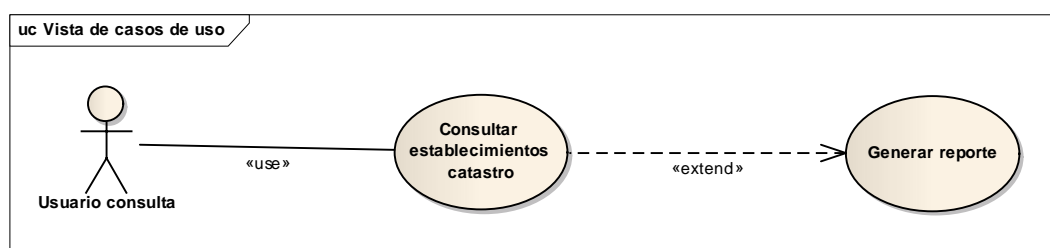


FIGURA N° 52: Diagrama de Caso de Uso – Requisito 13

## Diagrama de Secuencia

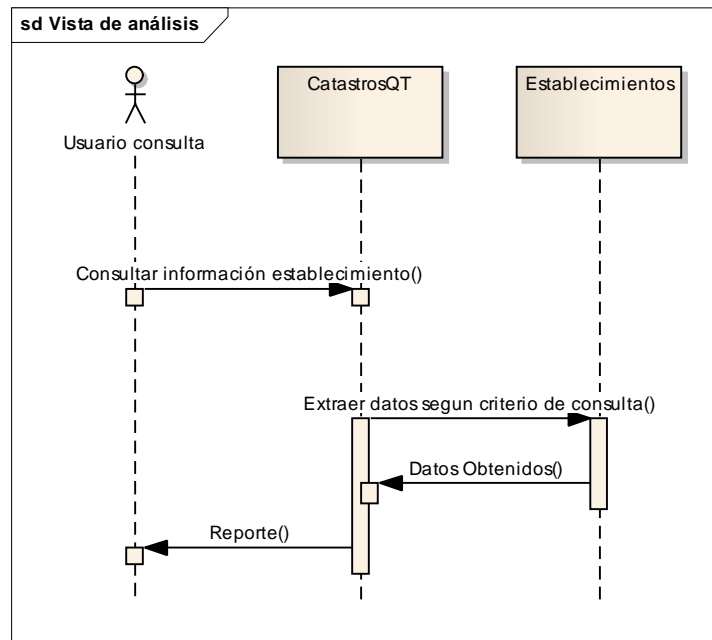


FIGURA N° 53: Diagrama de Secuencia – Requisito 13

## Diagrama de Colaboración

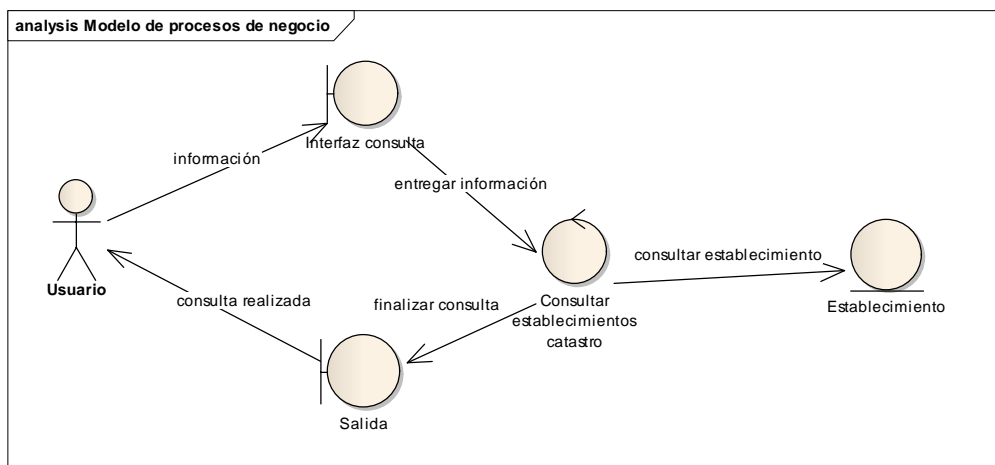


FIGURA N° 54: Diagrama de Colaboración – Requisito 13

**Requisito 14:** El sistema debe proveer de un esquema de reportes que permita consultar la información de las Emisiones, Pagos y Bajas registradas, permitiendo además filtrar dichos reportes por Año, Fecha Inicio, Fecha Hasta, Actividad, Tipo, Categoría, Administración Zonal, Sector Turístico y Parroquia. Adicionalmente, se debe permitir exportar la información del reporte a Excel.



**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Año, Tipo de Reporte, Fecha Inicio, Fecha Hasta, Actividad, Tipo, Categoría, Administración Zonal, Sector Turístico y Parroquia.

**Flujo Normal:** El sistema debe extraer la información de la BD en base al tipo de reporte seleccionado y los filtros seleccionados.

**Salida:** Información correspondiente al reporte solicitado.

### Diagrama Caso de Uso

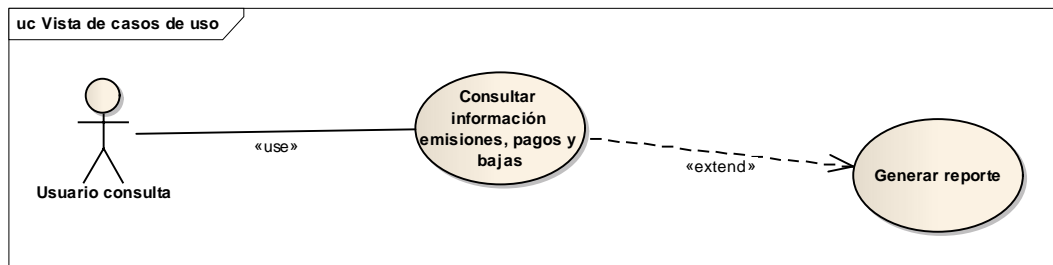


FIGURA N° 55: Diagrama de Caso de Uso – Requisito 14

### Diagrama de Secuencia

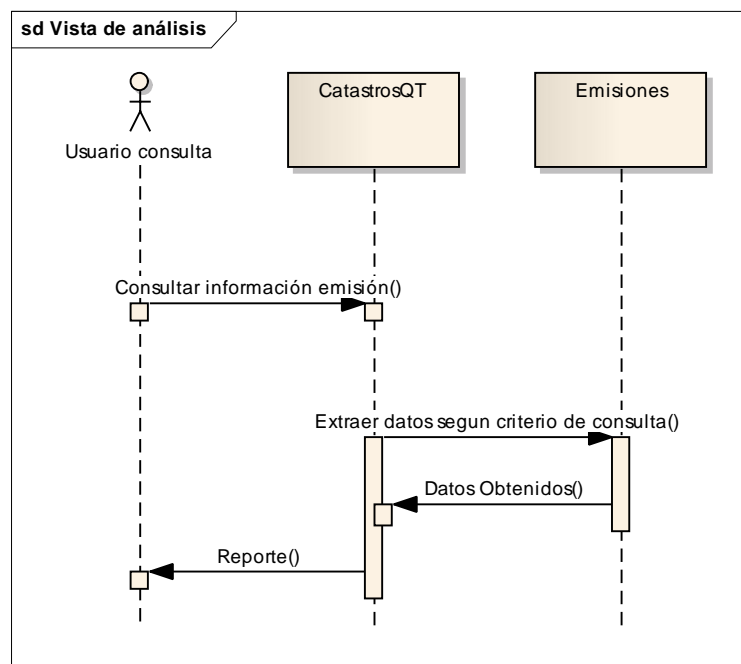


FIGURA N° 56: Diagrama de Secuencia – Requisito 14

## Diagrama de Colaboración

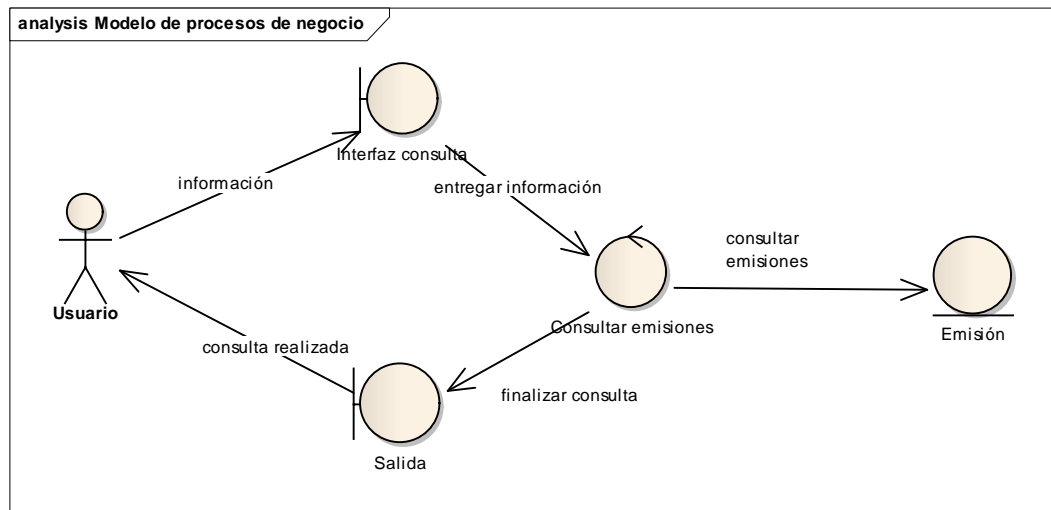


FIGURA N° 57: Diagrama de Colaboración – Requisito 14

**Requisito 15:** El sistema debe proveer de un esquema de reportes que permita consultar la información de las Inspecciones registradas, permitiendo además filtrar dichos reportes por Año, Fecha Inicio, Fecha Hasta, Actividad, Tipo, Categoría, Administración Zonal, Sector Turístico y Parroquia. Adicionalmente, se debe permitir exportar la información del reporte a Excel.

**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Año, Tipo de Reporte, Fecha Inicio, Fecha Hasta, Actividad, Tipo, Categoría, Administración Zonal, Sector Turístico y Parroquia.

**Flujo Normal:** El sistema debe extraer la información de la BD en base al tipo de reporte seleccionado y los filtros seleccionados.

**Salida:** Información correspondiente al reporte solicitado.

## Diagrama Caso de Uso

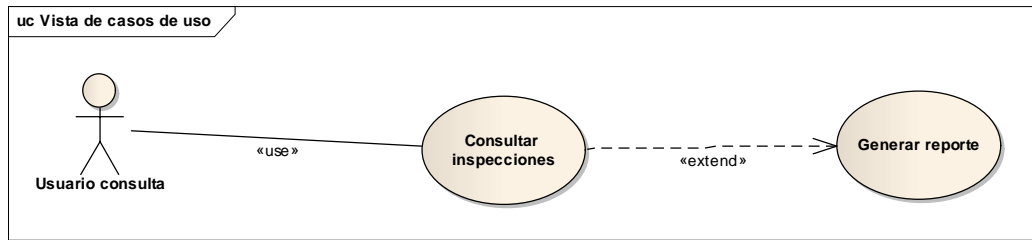


FIGURA N° 58: Diagrama de Caso de Uso – Requisito 15

## Diagrama de Secuencia

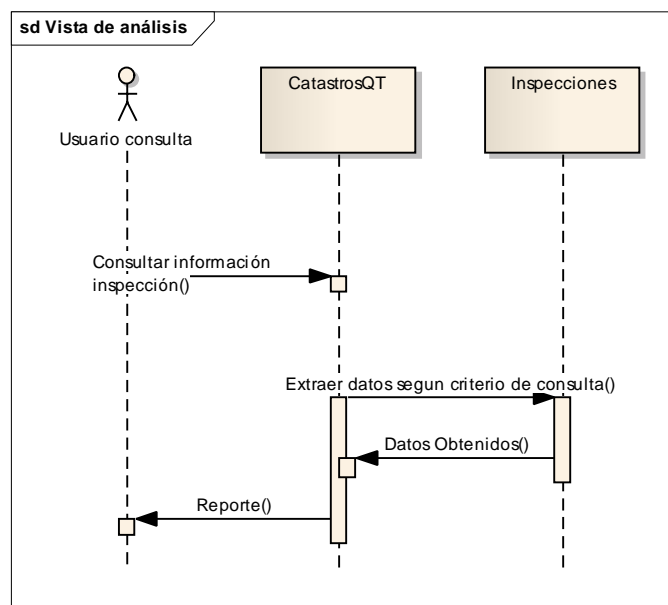


FIGURA N° 59: Diagrama de Secuencia – Requisito 15

## Diagrama de Colaboración

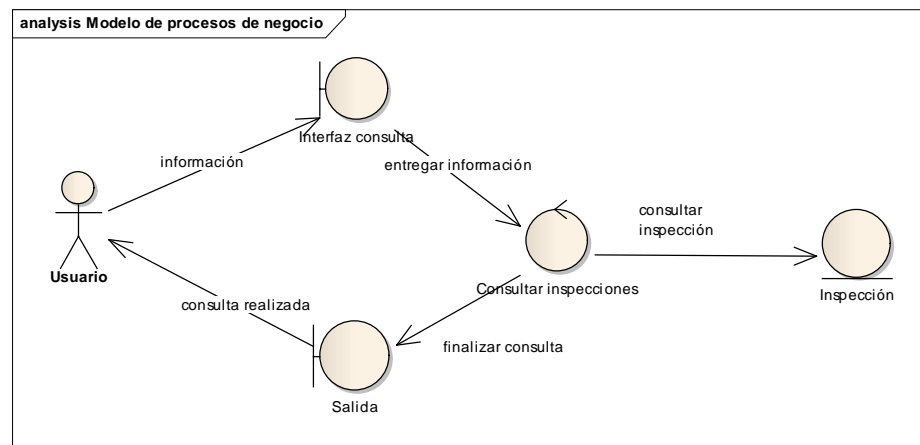


FIGURA N° 60: Diagrama de Colaboración – Requisito 15

### 3.1.3.2.6 Mantenimiento del Sistema

**Requisito 16:** El sistema debe permitir a cada usuario modificar su contraseña de tal manera que la misma permanezca en sigilo solo del usuario a quien se la asignó.

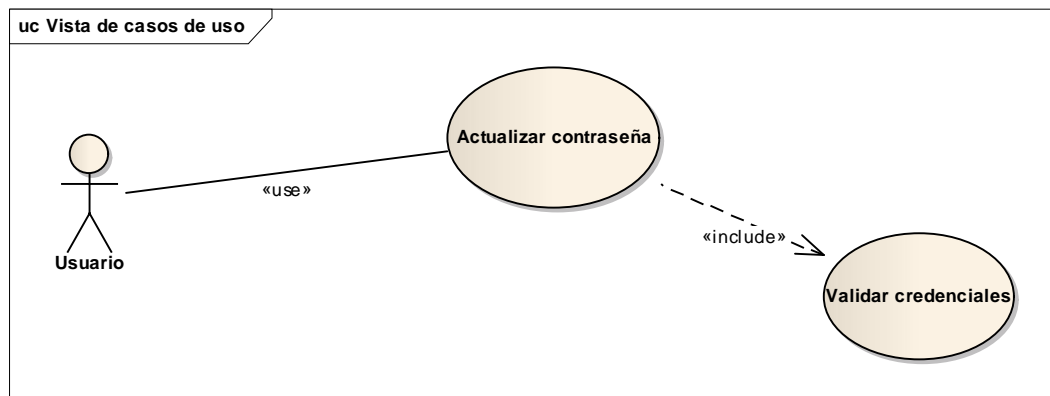
**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Usuario, Contraseña Actual, Nueva Contraseña.

**Flujo Normal:** El sistema debe validar que la Contraseña Actual ingresada coincida con aquella registrada en la BD. Luego de dicha validación debe actualizar la contraseña del usuario. En caso de que la contraseña ingresada sea incorrecta, debe cancelar el proceso y mostrar un mensaje de error al usuario.

**Salida:** Contraseña de usuario actualizada.

#### Diagrama Caso de Uso



**FIGURA N° 61:** Diagrama de Caso de Uso – Requisito 16

## Diagrama de Secuencia

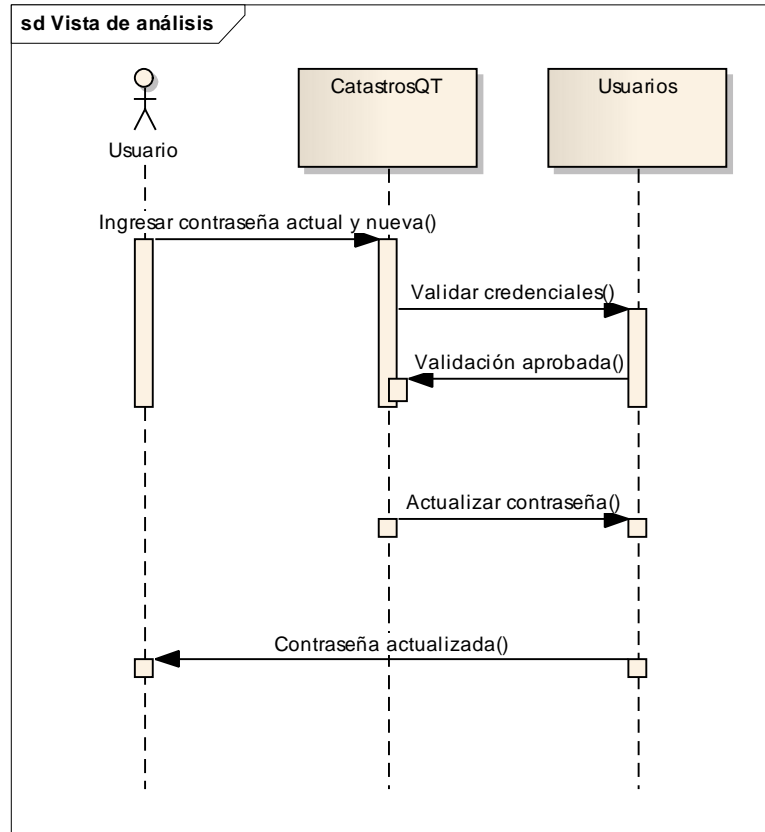


FIGURA N° 62: Diagrama de Secuencia – Requisito 16

## Diagrama de Colaboración

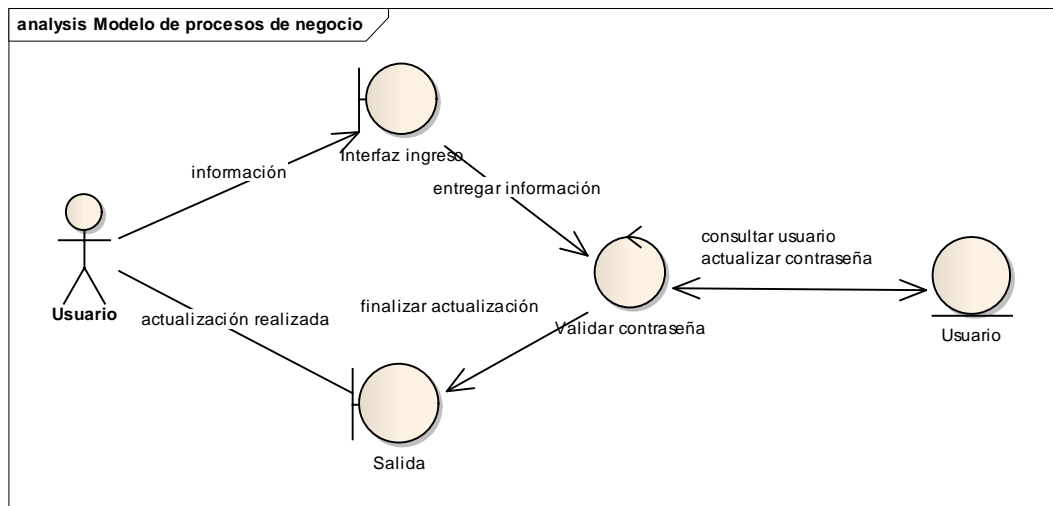


FIGURA N° 63: Diagrama de Colaboración – Requisito 16

**Requisito 17:** El sistema debe permitir al usuario Administrador actualizar el Valor de Detalle y Valor Máximo que debe aplicarse para el cobro de los valores anuales por funcionamiento a los establecimientos turísticos, en base a su Actividad, Tipo y Categoría.

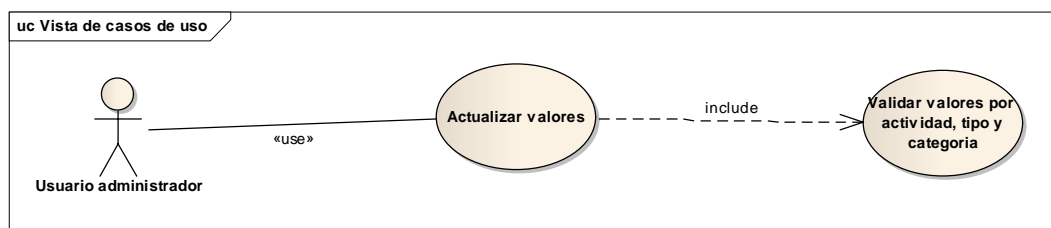
**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Actividad, Tipo, Categoría, Valor de Detalle y Valor Máximo.

**Proceso:** El sistema debe consultar en la DB el Valor de Detalle y Valor Máximo para la Actividad, Tipo y Categoría seleccionadas y desplegar las mismas en la pantalla. Luego de esto, el usuario ingresará los valores correspondientes a cada rubro y al seleccionar el botón “Actualizar”, el sistema deberá actualizar dichos valores en la BD.

**Salida:** Valores de cobro actualizados.

### Diagrama Caso de Uso



**FIGURA N° 64:** Diagrama de Caso de Uso – Requisito 17

## Diagrama De Secuencia

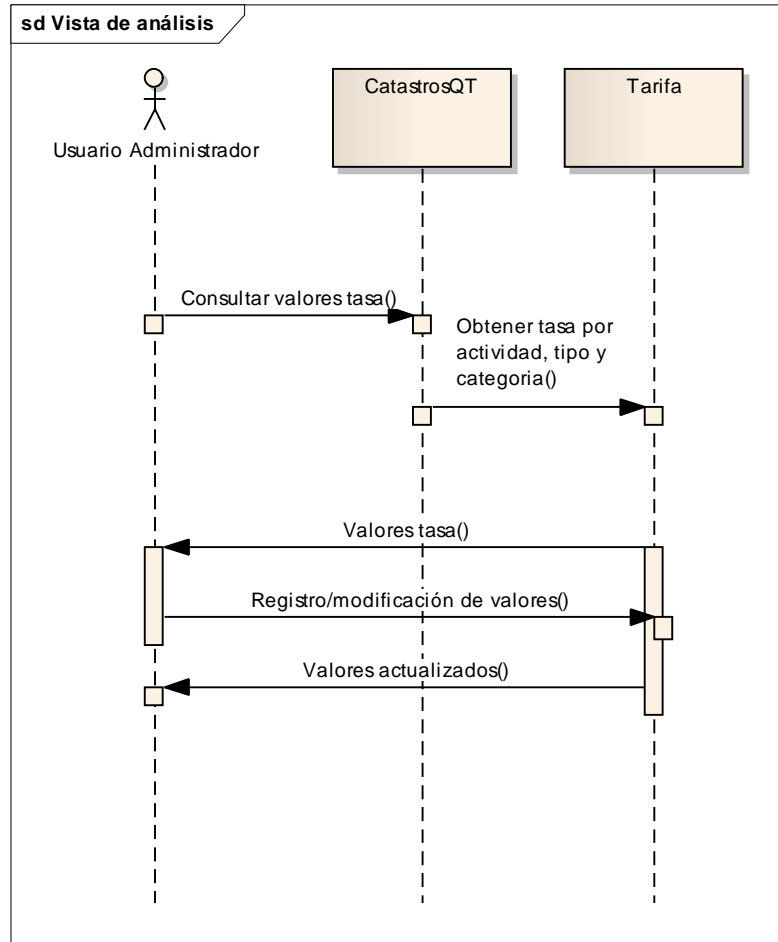


FIGURA N° 65: Diagrama de Secuencia – Requisito 17

## Diagrama de Colaboración

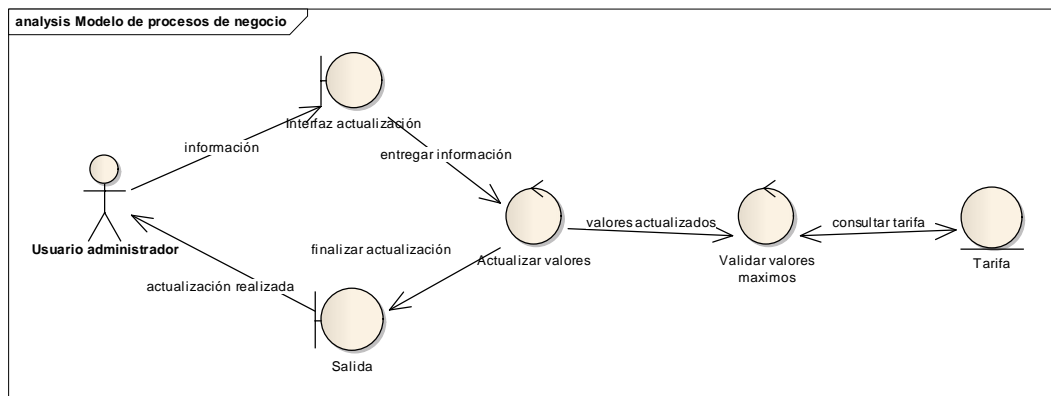


FIGURA N° 66: Diagrama de Colaboración – Requisito 17

**Requisito 18:** El sistema debe permitir al usuario Administrador realizar el ingreso/consulta/modificación de los usuarios registrados en el sistema.

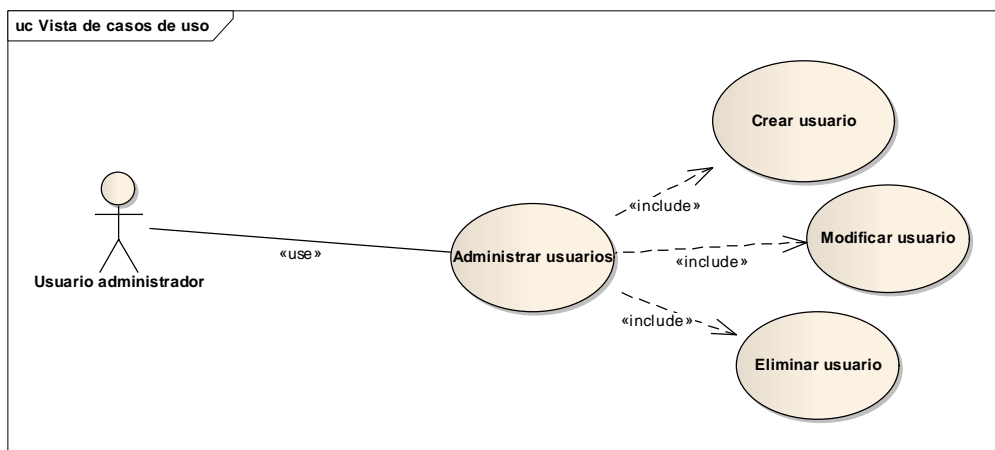
**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Usuario, Nombre Completo, Email, Perfil, Ubicación y Estado.

**Flujo Normal:** Al seleccionar la opción de “Mantenimiento de Usuarios”, el sistema debe desplegar el listado de todos los usuarios registrados en el sistema. El usuario puede entonces seleccionar agregar un nuevo usuario o modificar la información de uno ya existente. Finalmente, una vez registrada la información, el sistema debe ingresar/actualizar en la BD la información del usuario.

**Salida:** Usuario ingresado/actualizado.

### Diagrama Caso de Uso



**FIGURA N° 67:** Diagrama de Caso de Uso – Requisito 18



## Diagrama de Secuencia

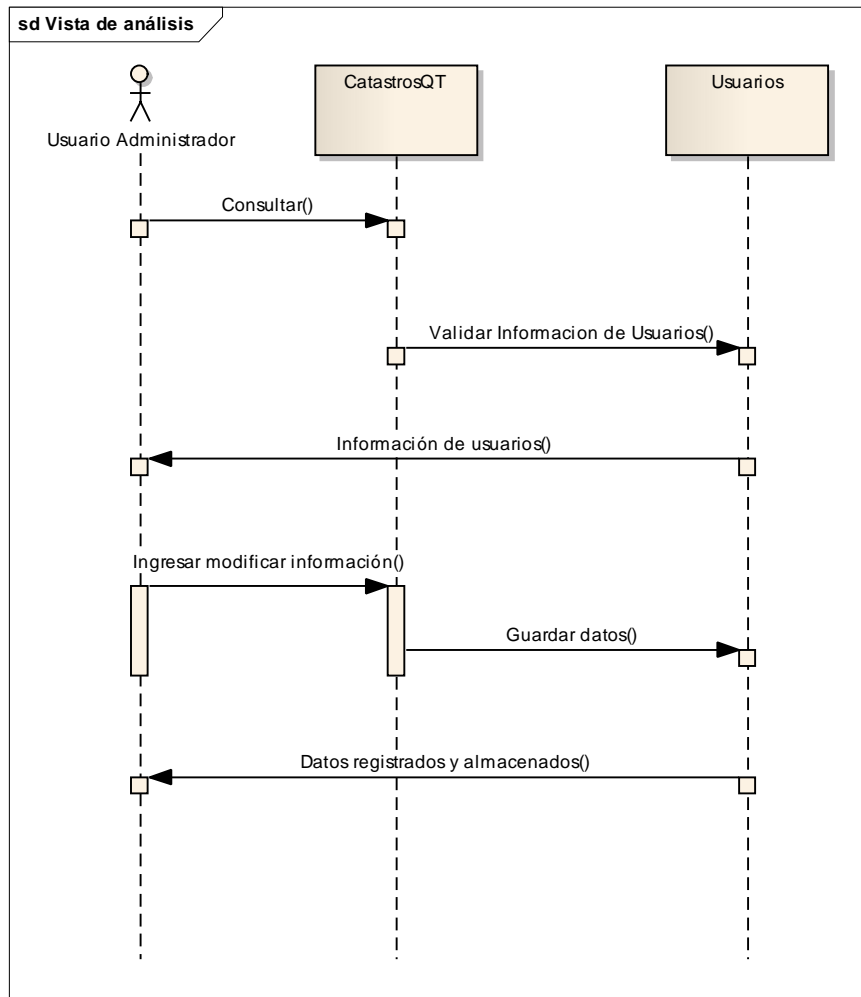


FIGURA N° 68: Diagrama de Secuencia – Requisito 18

## Diagrama de Colaboración

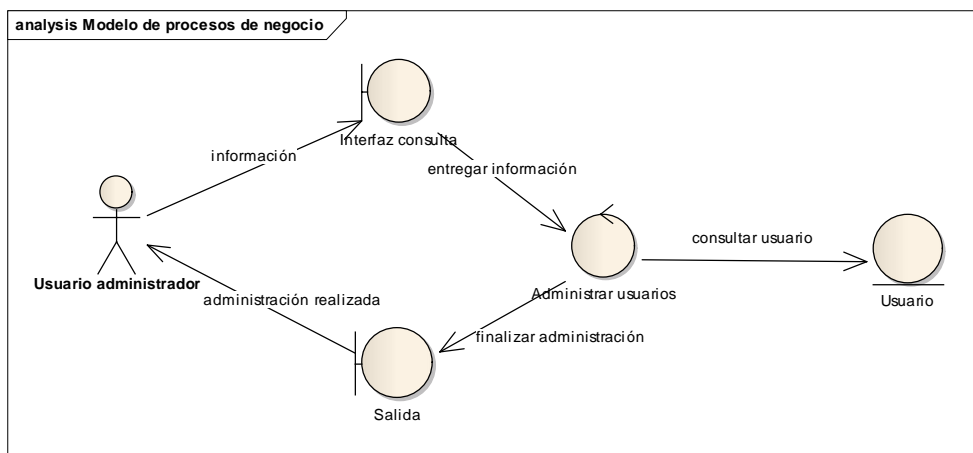


FIGURA N° 69: Diagrama de Caso de Uso – Requisito 18

**Requisito 19:** El sistema debe permitir al usuario Administrador realizar el ingreso/consulta/modificación de los perfiles registrados en el sistema y los recursos asignados a cada perfil.

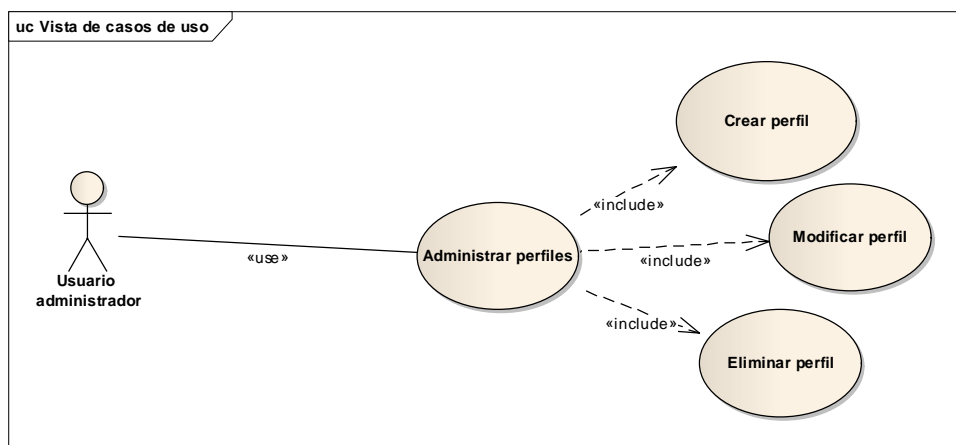
**Precondiciones:** El usuario debe haber ingresado previamente al sistema.

**Entradas:** Perfil, Recurso y Estado.

**Flujo Normal:** Al seleccionar la opción de “Mantenimiento de Perfiles”, el sistema debe consultar todos los perfiles registrados anteriormente en el sistema y desplegar el listado de todos los recursos registrados en el sistema, señalando aquellos que aplican al perfil escogido. El usuario puede entonces marcar/desmarcar los recursos según apliquen al perfil o agregar un nuevo perfil. Finalmente, una vez registrada la información, el sistema debe ingresar/actualizar en la BD la información del perfil.

**Salida:** Perfil ingresado/actualizado.

### Diagrama Caso de Uso



**FIGURA N° 70:** Diagrama de Caso de Uso – Requisito 19

## Diagrama de Secuencia

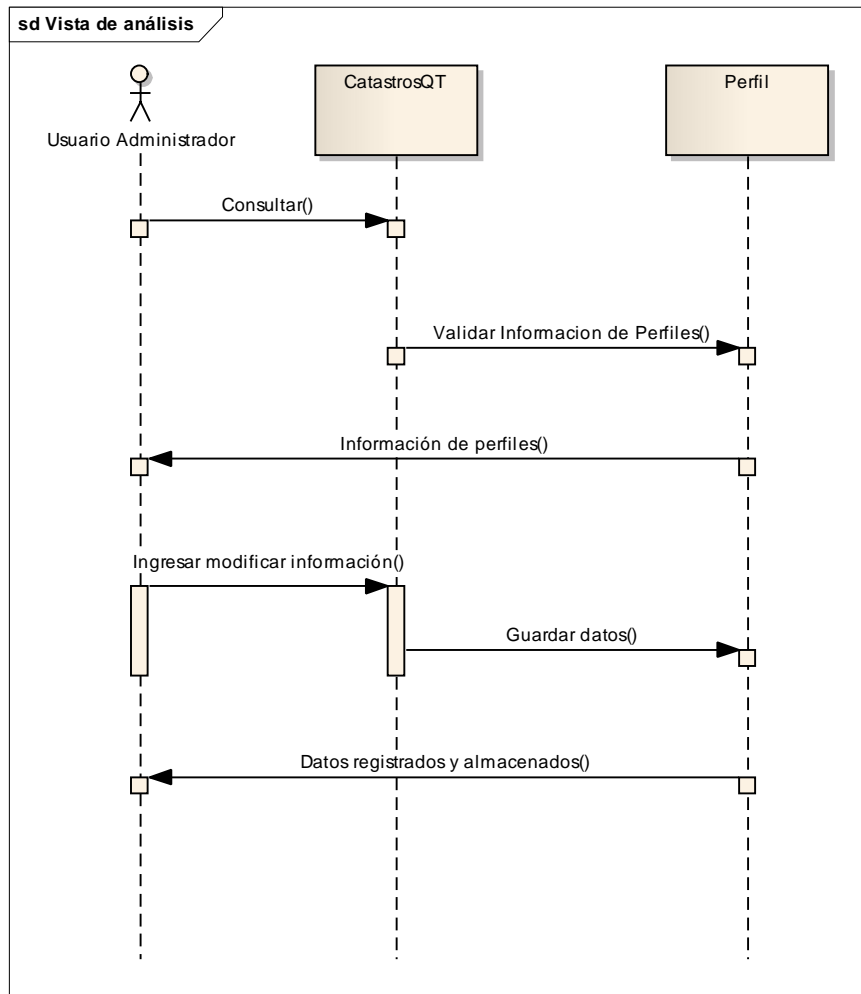


FIGURA N° 71: Diagrama de Secuencia – Requisito 19

## Diagrama de Colaboración

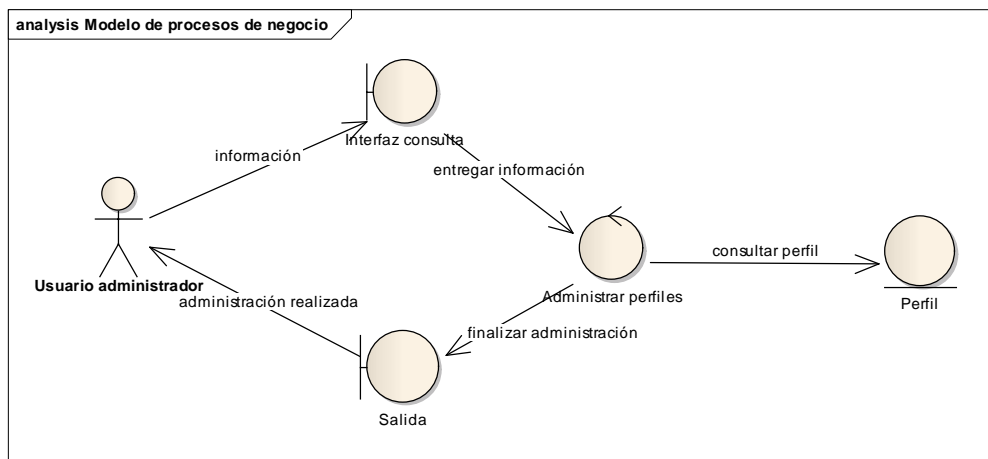


FIGURA N° 72: Diagrama de Colaboración – Requisito 19

### **3.1.3.3 Requisitos de Rendimiento**

La carga que se espera deberá soportar el sistema es reducida, por cuanto no se espera tener más de 3 (tres) usuarios simultáneos conectados.

Se ha definido que el número total de estaciones que deberán poder acceder al sistema CatastrosQT es de 6 (seis), lo cual será soportado fácilmente por el servidor de aplicaciones y el sistema en sí.

### **3.1.3.4 Restricciones de Diseño**

Dado que se ha optado por utilizar la Metodología Orientada a Objetos, el diseño se lo realizara mediante herramientas CASE con orientación a objetos. Así mismo, ya que la herramienta de desarrollo seleccionada es Visual Studio .NET 2005, misma que es ideal para desarrollar basándose en un diseño orientado a objetos, las restricciones de diseño entonces serán referentes a las herramientas a utilizar.



## **CAPÍTULO 4**

### **DESARROLLO E IMPLANTACIÓN DE LA APLICACIÓN**

#### **4.1 DESARROLLO DEL SISTEMA**

El desarrollo del sistema CatastrosQT se lo realiza en base a la información recolectada en el proceso de Análisis y Diseño, así como también utilizando la herramienta de desarrollo Visual Studio .NET 2005, con el lenguaje de programación C#.

##### **4.1.1 Componentes Desarrollados**

A continuación, se describe brevemente los módulos construidos como parte del desarrollo del sistema.

###### **4.1.1.1 Capa de Acceso a Datos**

Para administrar la conexión al motor de Base de Datos SQL Server 2005, se generó una clase individual llamada EjecutaSQL, la misma que se encarga de gestionar la invocación de los Procedimientos Almacenados (SPs) de una manera centralizada. Dentro de esta capa se encuentran el resto de clases para acceso a datos de acuerdo a su función, entre ellas tenemos a DatosEntorno, DatosEstablecimiento, DatosInspeccion, DatosLUAF, DatosLUAE, DatosReporte y DatosUsuario.

#### **4.1.1.2 Capa Transaccional**

Sirve para la ejecución de varios procesos en un solo paso, que requieren mantener una integridad relacional, es decir que todas las transacciones se manejen de forma atómica, por lo que si una falla, las anteriores deben deshacerse y las posteriores no deben ejecutarse. Dentro de esta capa se encuentran procesos complejos de registro de información como son el de registro de Establecimientos y el registro de Inspecciones. Las clases dentro de esta capa se encuentran divididas de acuerdo a su función, entre ellas tenemos a TransaccionEstablecimiento y TransaccionInspeccion.

#### **4.1.1.3 Capa de Negocio**

Es utilizada para comunicar a las capas de Presentación con la capa de Acceso a Datos. Permite centralizar las reglas del negocio que deben aplicarse a ciertos procesos. Dentro de esta capa se encuentran varias clases separadas de acuerdo a su función, entre ellas tenemos a NegocioEntorno, NegocioEstablecimiento, NegocioInspeccion, NegocioLUAF, NegocioLUAE, NegocioReporte y NegocioUsuario. Esta capa también es utilizada como intermedio para la ejecución de operaciones solicitadas desde la capa de Presentación que requieren mantener atomicidad, cuya invocación es trasladada a la capa Transaccional.

#### **4.1.1.4 Capa Útil**

Contiene funcionalidad de transformación de datos o cálculos generales que son utilizados en varias de las capas del aplicativo. Además contempla las características de envío de emails, log de errores y constantes del sistema. Al interior de esta capa se encuentran varias clases de acuerdo a su función, entre

ellas tenemos a DatosUtil, FechaUtil, NumeroUtil, MailPersonalizado, Constante, Loggeador y Validador.

#### **4.1.1.5 Capa de Entidades**

Es utilizada para definir las entidades de datos que serán generadas para un paso de información mas simple, limpio y controlado entre las distintas capas del sistema. Dentro de esta capa se encuentran varias clases de acuerdo a su función, entre ellas tenemos a Coordinada, Establecimiento, Inspeccion, Reporte, Rubro, Tarifa y Usuario.

#### **4.1.1.6 Capa de Presentación de Usuario (Web)**

Representa a la capa del aplicativo que es visible al usuario del mismo y que le permite interactuar con el sistema. Desde esta capa se originan todas las peticiones de consulta, inserción o modificación de la información del aplicativo CatastrosQT.

Dentro de esta capa, además del lenguaje de programación C#, se utiliza JavaScript, el mismo que es un lenguaje que no requiere ser compilado, ya que se ejecuta en el lado del cliente y es interpretado por el navegador web usado por el usuario. JavaScript sirve principalmente para realizar las validaciones de datos en el lado del cliente y evitar procesamiento innecesario en el lado del servidor.

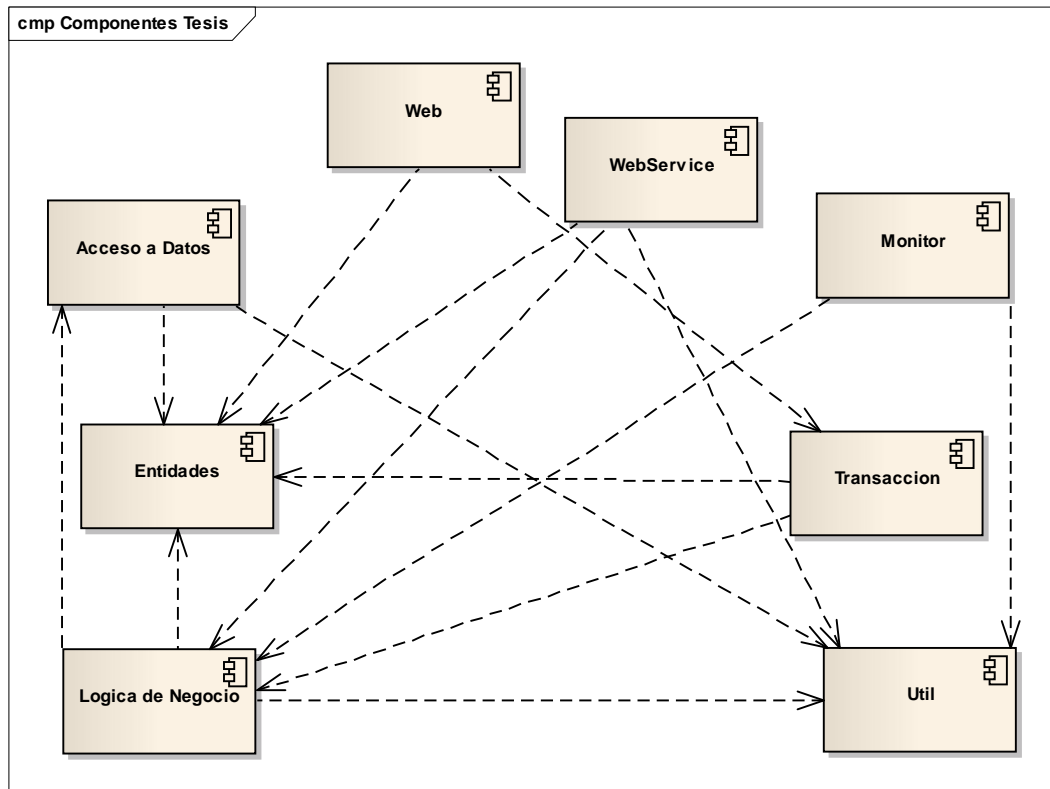
#### **4.1.1.7 Capa de Servicios Web**

Sirve para la interconexión con aplicativos externos como el sistema LUAE-BPM, el mismo que utiliza XML como forma nativa de intercambio de información.



Mediante esta capa, el sistema LUAE-BPM puede registrar solicitudes de trámite, para que las mismas sean procesadas dentro de CatastrosQT.

A continuación se presenta el diagrama de componentes, donde se puede visualizar la interacción de las capas anteriormente mencionadas:



**FIGURA N° 74:** Diagrama de Componentes

## 4.2 EJECUCIÓN DE PRUEBAS

Una vez culminado el desarrollo del sistema y para verificar que la versión final del aplicativo sea entregada libre de errores, es necesario ejecutar pruebas funcionales y de stress, en conjunto con los usuarios finales, las mismas que se realizan de manera exitosa.

Para la realización de pruebas de stress, se tomaron en cuenta los siguientes criterios:

- Tiempo de Ejecución: 60 minutos.
- Concurrencia: 30 usuarios.
- Aplicación: CatastrosQT.
- Ambiente: TEST.

La funcionalidad a ser evaluada fue la más transaccional, crítica e importante a criterio de los responsables del área de catastros de la EPMGDT:

- Ingresar un Trámite LUAE
- Registrar un formulario de Inspección
- Obtener el Reporte General de Establecimientos

A continuación se detallan los valores de ejecución esperados para las pruebas de stress a ejecutar sobre las transacciones anteriormente mencionadas:

| Transacción                      | Usuarios | Transacciones | TPS   | Tiempo aceptable | Tiempo Inaceptable |
|----------------------------------|----------|---------------|-------|------------------|--------------------|
| Ingresar Trámite                 | 10       | 30            | 0,005 | 3                | 5                  |
| Realizar Inspección              | 10       | 30            | 0,005 | 3                | 5                  |
| Reporte General Establecimientos | 10       | 30            | 0,005 | 4                | 6                  |

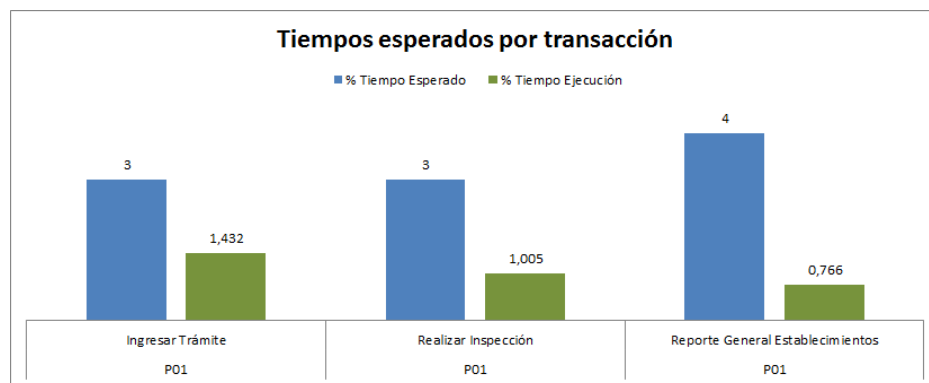
**TABLA N° 10:** Valores esperados para pruebas de stress

A continuación se detallan valores de ejecución obtenidos, luego de la realización de las pruebas de stress con el aplicativo LoadRunner:

| Transacción                      | Usuarios | Transacciones | TPS   | Tiempo Mínimo | Tiempo Promedio | Tiempo Máximo |
|----------------------------------|----------|---------------|-------|---------------|-----------------|---------------|
| Ingresar Trámite                 | 10       | 31            | 0,008 | 0,119         | 1,432           | 2,946         |
| Realizar Inspección              | 10       | 36            | 0,008 | 0,195         | 1,005           | 2,176         |
| Reporte General Establecimientos | 10       | 30            | 0,008 | 0,137         | 0,766           | 4,402         |

**TABLA N° 11:** Valores obtenidos en ejecución de pruebas de stress

Como se puede observar en la tabla anterior, los tiempos de respuesta son menores a los esperados, con lo cual se valida que el sistema cumple con los requisitos mínimos establecidos para tal efecto, tal y como se puede ver en la siguiente gráfica:



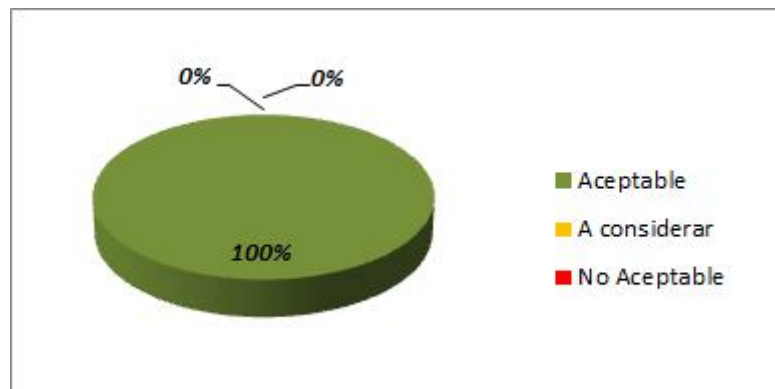
**FIGURA N° 75:** Tiempos esperados por transacción

Dentro del conjunto de pruebas a realizar se han establecido los siguientes criterios de aceptación respecto del rendimiento de las pruebas:

- **Aceptable:** Implica que para la transacción evaluada, el tiempo promedio obtenido es menor o igual al tiempo aceptable esperado, con lo cual se valida que el proceso se realiza en un tiempo mejor que el esperado.
- **A considerar:** Aplica cuando el tiempo promedio obtenido de ejecución de la transacción se encuentra entre el tiempo aceptable esperado y el tiempo inaceptable esperado. En cuyo caso, debe validarse la transacción para determinar si la misma por su nivel de complejidad es aceptada por los responsables de la evaluación de las pruebas.
- **No Aceptable:** Aplica en los casos en los que para la transacción ejecutada, el tiempo promedio obtenido es mayor al tiempo inaceptable esperado,

conlleando con esto que el proceso tome más tiempo de lo que debería tomar, lo cual implica que el proceso debe mejorarse notablemente.

A continuación se incluye un gráfico con la evaluación de los tiempos obtenidos en la ejecución de las pruebas de stress, en la cual se muestran los porcentajes de transacciones Aceptables, A considerar y No Aceptables:



**FIGURA N° 76:** Porcentaje de transacciones Aceptables, A considerar y No Aceptables

Finalmente, se evaluarán el porcentaje de éxito o error de las transacciones ejecutadas, obtenidos en las pruebas. Una transacción exitosa implica que la misma se realizó sin error, registrándose correctamente la información en el repositorio de Base de Datos. Una transacción con error se refiere a aquella que tuvo un resultado fallido, el cual pudo ocasionarse por cualquiera de las siguientes condiciones: Fallo en la infraestructura y/o conexión a la BD, Respuesta de error por datos incorrectos o Errores en el Aplicativo.

Como se puede observar en la siguiente gráfica, todas las transacciones ejecutadas fueron exitosas.



**FIGURA N° 77:** Porcentaje de éxito de transacciones evaluadas

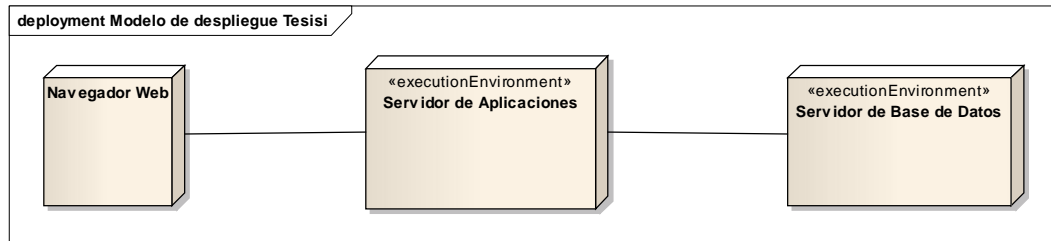
Los resultados que se obtuvieron en la ejecución de las pruebas de stress generaron las siguientes conclusiones:

- El 100% de las transacciones evaluadas tuvieron un tiempo de respuesta Aceptable, un 0% de transacciones A considerar y un 0% de transacciones evaluadas como Inaceptables.
- De todas las transacciones ejecutadas, existió un 0% transacciones fallidas.
- El 100% de las transacciones alcanzaron los TPS (Transacciones Por Segundo) esperados.

Los resultados obtenidos garantizan que el sistema sea fiable y que su rendimiento sea mejor al esperado por los usuarios, lo cual constituye un valor agregado para los mismos.

#### **4.3 IMPLANTACIÓN DEL SISTEMA Y TRANSICIÓN**

La Implantación del sistema, junto con el proceso de Transición y entrega del mismo se explican más adelante. A continuación se muestra un diagrama de como quedará implementada la solución a nivel de despliegue:



**FIGURA N° 78:** Diagrama de Despliegue

### 4.3.1 Implantación

El Manual de Instalación del Sistema especifica las características mínimas de hardware y software que son requeridas para la correcta ejecución del mismo. De igual manera se explican los pasos para la instalación del aplicativo y las configuraciones a realizar. El Manual de Instalación del Sistema se encuentra en el Anexo A.

### 4.3.2 Entrega del Sistema

Adjunto al presente documento, se entregarán el correspondiente Manual de Usuario y de Instalación, así como un CD en el cual se encuentra el código fuente de la aplicación y todo lo necesario para que el sistema pueda funcionar.

### 4.3.3 Manual de Usuario

El Manual de Usuario constituye una guía de uso del sistema, la cual resume las características funcionales del mismo así como presenta una explicación paso a paso de como aprovechar los beneficios del sistema. El Manual de Usuario se encuentra en el Anexo B.

## CAPÍTULO 5

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 CONCLUSIONES

- El Sistema de Control del Catastro Turístico, CatastrosQT, reemplaza procesos manuales que actualmente son obsoletos, con una administración centralizada y organizada, que dará mayor seguridad y consistencia a la información; garantizando además, la obtención de reportes actualizados e inmediatos y reduciendo considerablemente el esfuerzo requerido para lograrlo.
- La especificación de requerimientos fue un proceso fundamental en el inicio del proyecto, en este se involucraron tanto a los usuarios expertos como a los desarrolladores, por lo que fue el pilar sobre la cual se elaboró el sistema. De esta manera se garantizó que el sistema cumpla con las necesidades de los usuarios finales.
- El uso de la metodología RUP, a pesar de requerir un mayor esfuerzo por parte de los equipos de trabajo, permitió que se definan de manera clara y precisa los requisitos funcionales y optimizó el desarrollo del software ya que los casos de uso y diagramas relacionados permitieron a los programadores entender de mejor manera las necesidades de los usuarios.
- En el diseño del aplicativo CatastrosQT, se estableció que este debía ser un aplicativo Web. De esta forma se agregó una gran cantidad de beneficios a la institución, entre los que podemos mencionar: Simplicidad de Acceso a la información desde cualquier parte del mundo, Mantener la Información centralizada y actualizada, Facilidad de actualización del

aplicativo pues no requiere instalarse en cada equipo, Gran escalabilidad del mismo, etc.

- Desarrollar la solución de software con una activa participación de todos los involucrados; y, en especial de los usuarios, hace que estos se sientan parte del éxito del proyecto, lo cual reduce en gran medida el miedo a lo nuevo y evita el rechazo del sistema.

## **5.2 RECOMENDACIONES**

- Al momento de implementar o realizar el desarrollo de un sistema, se recomienda hacer uso de las metodologías de desarrollo más destacadas y que se ajusten a las necesidades del aplicativo, se debe incluir también estándares de calidad en el diseño y en el desarrollo de software para que su arquitectura sea entendible y su código fuente legible, esto servirá a los desarrolladores al momento de realizar modificaciones o nuevas implementaciones en el sistema.
- Se sugiere implementar modelos de validación de seguridad en conjunto con el equipo de desarrollo del sistema externo LUAE-BPM para garantizar la seguridad de la información que ambos sistemas comparten, ya que al usarse servicios web, estos son vulnerables a ataques de personas no autorizadas.
- Se propone a la EPMGDT la realización de un proyecto de mejoramiento de continuación al presente, el cual permitirá una mayor integración con el sistema del Municipio de Quito, de tal manera que el registro de los pagos de los establecimientos sea automático y no requiera de un procesamiento manual.



- Se recomienda que el usuario Administrador del sistema sea una persona con alto grado de conocimiento de informática, por cuanto será la persona encargada de velar por la integridad de la información ingresada a la base de datos, así como de realizar la parametrización de valores que se requieran posteriormente.
- Es conveniente que el responsable de departamento de sistemas de la EPMGDT realice respaldos periódicos (de preferencia a diario) de la base de datos, de tal manera que en caso de ocurrir algún problema físico con el servidor se cuente con la información lo más actualizada posible y que la misma pueda ser restaurada, reduciendo al mínimo los problemas causados por dicho fallo.

## GLOSARIO DE TÉRMINOS

### A

**AUP:** Proceso Unificado Agil

### B

**BD:** Base de Datos

**BPM:** Modelo de Proceso de Negocios

### C

**Catastro:** Local turístico que opera dentro del Distrito Metropolitano

**CAPTUR:** Cámara Provincial de Turismo de Pichincha

**CASE (Herramienta):** Ingeniería de Software Asistida por Computadora

**CD:** Disco Compacto

**CMMI:** Integración de Modelos de Madurez de Capacidades

### D

**DBMS:** Sistema Administrador de Base de Datos.

### E

**EPMGDT:** Empresa Pública Metropolitana de Gestión de Destino Turístico.

### G

**GUI:** Interfaz Gráfica de Usuario.

### H

**HIPERMEDIA:** Hipermedia es el término con que se designa al conjunto de métodos o procedimientos para escribir, diseñar o componer contenidos que tengan texto, video, audio, mapas u otros medios, y que además tenga la posibilidad de interactuar con los usuarios.

## **I**

**IDE:** Entorno Integrado de Desarrollo.

**IIS:** Servicio de Información de Internet.

## **L**

**LoadRunner:** Sistema de HP para ejecución de pruebas de carga y stress.

**LUAF:** Licencia Única Anual de Funcionamiento.

## **M**

**MSF:** Microsoft Solution Framework. Conjunto de principios, modelos, disciplinas, guías y modelos para el desarrollo de aplicaciones.

## **O**

**OMG:** Grupo de Administración de Objetos.

**OMT:** Herramienta de Modelado de Objetos.

## **R**

**RUC:** Registro Único de Contribuyente.

**RUP:** Proceso Unificado de Desarrollo.

## **S**

**Scrum:** Marco de trabajo para la gestión y desarrollo ágil de software. Se basa en procesos iterativos e incrementales.

**SGBD:** Sistema Gestor de Base de Datos.

**SP:** Procedimiento Almacenado de una Base de Datos.

**SQL:** Lenguaje de Consulta Estructurado.

## **T**

**TCP/IP:** Protocolo de Control de Transmisión/Protocolo de Internet.

**TOOLKIT:** Conjunto o grupo de herramientas.

**TPS:** Transacciones Por Segundo.

## **U**

**UML:** Lenguaje Unificado de Modelado.

## **W**

**Web Service:** Servicio Web que permite el intercambio de datos entre aplicaciones.

## **X**

**XP:** Programación Extrema (eXtreme Programming).

## **BIBLIOGRAFÍA**

### **REFERENCIAS MATERIAL IMPRESO**

#### **RUP**

El Proceso Unificado de Desarrollo de Software. I. Jacobson, G. Booch, J. Rumbaugh. Addison-Wesley, 2000, ISBN: 84-7829-036-2

#### **UML**

El lenguaje Unificado de Modelado, Manual de Referencia. J. Rumbaugh, I. Jacobson, G. Booch. Addison-Wesley, Tercera Edición, ISBN: 84-7829-037-0

### **REFERENCIAS MATERIAL ELECTRÓNICO**

#### **INTRODUCCION**

Empresa Pública Metropolitana de Gestión de Destino Turístico - Acerca de Nosotros. Internet.

[http://www.quito-turismo.gob.ec/index.php?option=com\\_content&task=view&id=25&Itemid=57](http://www.quito-turismo.gob.ec/index.php?option=com_content&task=view&id=25&Itemid=57)

Acceso último: 11/10/2011.

#### **APLICACIONES DISTRIBUIDAS**

Lauro Soto; Aplicaciones Distribuidas. Internet.

<http://www.mitecnologico.com/Main/AplicacionesDistribuidas>

Acceso último: 15/11/2011

Pedro Cuesta Morales; Desarrollo de Aplicaciones Distribuidas basadas en Tecnologías Web. PDF

<http://trevinca.ei.uvigo.es/~pcuesta/publicaciones/TecWeb.pdf>

Acceso último: 15/11/2011

Jesús Villalobos; Aplicaciones Distribuidas. Internet.

<http://knol.google.com/k/aplicaciones-distribuidas>

Acceso último: 17/11/2011

## **HERRAMIENTAS DE DESARROLLO**

Programación C#. PDF.

[http://www.cepeu.edu.py/LIBROS\\_ELECTRONICOS\\_2/Lenguaje%20c.pdf](http://www.cepeu.edu.py/LIBROS_ELECTRONICOS_2/Lenguaje%20c.pdf)

Acceso último: 10/01/2012.

Carrillo Ayala Alma Nelly / Cervantes Estrada Freddy / Gutiérrez Cervantes Miguel; SQL SERVER. PDF.

<http://www.uaem.mx/posgrado/mcruz/cursos/miic/sql.pdf>

Acceso último: 09/12/2012.

## **IEEE**

Especificación de Requisitos según el estándar de IEEE 830. Internet.

<http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

Acceso último: 20/02/2012

## **PATRONES DE DISEÑO**

Ramiro Lago; Patrones de diseño software. Internet.

<http://www.proactiva-calidad.com/java/patrones/index.html>

Acceso último: 20/11/2011

Joaquín García; IngenieroSoftware. Internet.

<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>

Acceso último: 21/11/2011

Reutilización del Software. PDF.

<http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf>

Acceso último: 21/11/2011

Nicolás Tedeschi, ¿Qué es un Patrón de Diseño?. Internet.

<http://msdn.microsoft.com/es-es/library/bb972240.aspx>

Acceso último: 21/11/2011

## **RUP**

Gustavo Torossi; El Proceso Unificado de Desarrollo de Software. PDF.

[www.utim.edu.mx/~mgarcia/DOCUMENTO/ADSI2/ApuntesRUP.pdf](http://www.utim.edu.mx/~mgarcia/DOCUMENTO/ADSI2/ApuntesRUP.pdf)

Acceso último: 15/11/2011.

Robin Castro; Estructura básica del RUP. PDF.

[http://www.icesi.edu.co/biblioteca\\_digital/bitstream/item/399/1/rcastro\\_estructura-bas-puds.pdf](http://www.icesi.edu.co/biblioteca_digital/bitstream/item/399/1/rcastro_estructura-bas-puds.pdf)

Acceso último: 22/11/2011.

Vilma Quispe, Dante Huamantuco, José Vargas; MONOGRAFIA  
METODOLOGIA RUP (RATIONAL UNIFIED PROCESS)

<http://www.monografias.com/trabajos-pdf4/metodologia-rup-una-puno/metodologia-rup-una-puno.pdf>

Acceso último: 03/11/2011.

## **UML**

Donald Bell; UML basics: An introduction to the Unified Modeling Language.  
Internet.

<http://www.ibm.com/developerworks/rational/library/769.html>

Acceso último: 04/12/2011.

Enrique Hernández; El Lenguaje Unificado de Modelado (UML). PDF.

<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

Acceso último: 09/12/2011.



## ANEXO A

### MANUAL DE INSTALACIÓN DEL SISTEMA

#### **Requerimientos de hardware y software**

##### ***Requerimientos Mínimos de Hardware – Servidor de Aplicación***

Procesador Pentium III de 1.5 GHz

512 MB de memoria RAM

Espacio libre de al menos 250 MB en disco duro

Tarjeta de vídeo SVGA ó superior

Doble Tarjeta de Red (una para la LAN y la otra para la red del Municipio)

##### ***Requerimientos Mínimos de Hardware – Servidor de Base de Datos***

Procesador Pentium III de 1.5 GHz

512 MB de memoria RAM

Espacio libre de al menos 150 MB en disco duro

Tarjeta de vídeo SVGA ó superior

Tarjeta de Red

##### ***Requerimientos Mínimos de Hardware – Cliente***

Procesador Pentium III de 1.5 GHz

256 MB de memoria RAM

Espacio libre de al menos 50 MB en disco duro

Tarjeta de vídeo SVGA ó superior

Tarjeta de Red

Resolución de Pantalla de 1024 x 800

##### ***Requerimientos Mínimos de Software – Servidor de Aplicación***

Windows Server 2003 o superior

Microsoft .NET Framework 2.0 con SP1

IIS v6.0 o superior

##### ***Requerimientos Mínimos de Software – Servidor de Base de Datos***

Windows Server 2003 o superior

SQL Server 2005 Express o superior

##### ***Requerimientos Mínimos de Software – Cliente***

Windows XP o superior

Internet Explorer 7.0 o superior

## Pasos para la instalación del sistema

A continuación se enumeran las diferentes actividades que se deben realizar para la instalación del aplicativo CatastrosQT:

1. Ejecutar el paquete de instalación **CatastrosQT.Instalacion.msi**, el cual copia los archivos necesarios por el aplicativo dentro del path **[drive:]\Inetpub\wwwroot** en el directorio **CatastrosQT**.
2. Dentro del **Servidor de Información de Internet (IIS)**, navegar hasta el directorio virtual **Sitio Web predeterminado**, dentro de este se encuentra el directorio **CatastrosQT**. Dar clic derecho sobre ese directorio y seleccionar la opción de **Propiedades**.

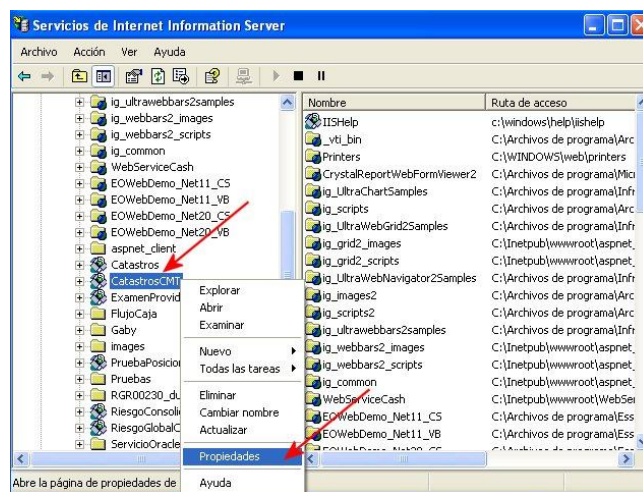


FIGURA 01: Configuración del Directorio Virtual

3. En el cuadro de diálogo que se muestra, dar clic en el botón **CREAR**

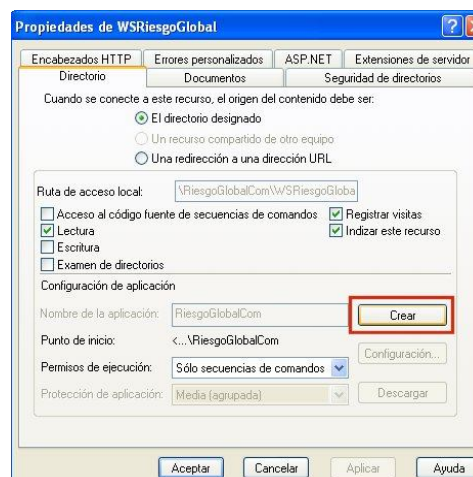


FIGURA 02: Creación del Directorio Virtual

4. Finalmente, dentro de la pestaña de **Seguridad de Directorios**, verificar que las opciones del cuadro de diálogo **Métodos de Autenticación**, que se muestra al dar clic en el botón **Modificar**, seleccionar la opción de **Acceso Anónimo**

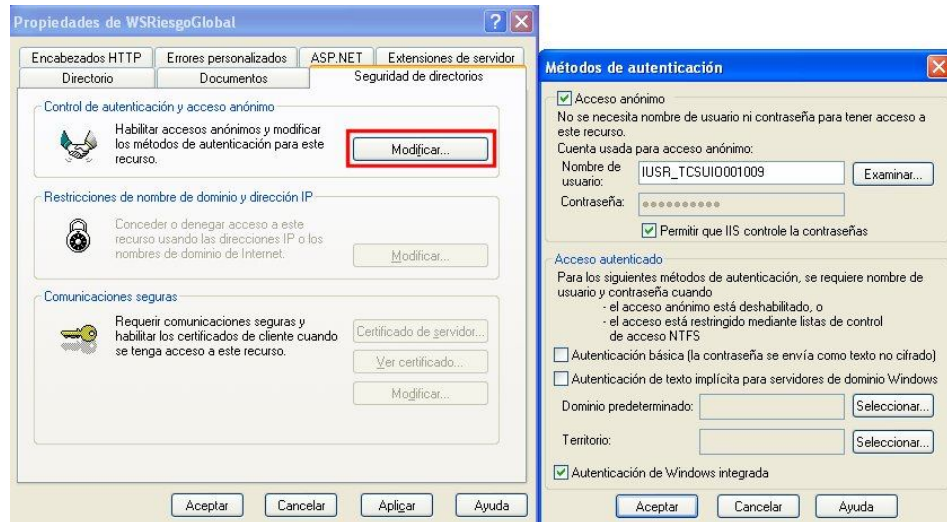


FIGURA 03: Configuración del Directorio Virtual

5. Dentro del Servidor de SQL Server, ejecutar los siguientes scripts en el orden indicado:

- 1.- Creacion\_CatastrosQT.sql
- 2.- Insercion\_Inicial\_Datos.sql
- 3.- Migracion\_Datos.sql

## **ANEXO B**

### **MANUAL DE USUARIO**

El documento se encuentra dentro del contenido del CD.

## **ANEXO C**

### **Ordenanza Municipal 0243**

El documento se encuentra dentro del contenido del CD.

## **ANEXO D**

### **CATASTROS - MANUAL DE PROCESOS**

El documento se encuentra dentro del contenido del CD.

## ANEXO E

### CARTA DE AUSPICIO DEL SISTEMA



Quito  
Turismo

Av. 6 de Diciembre N25-96 entre Av. Colón y Pasaje Batallas  
Teléfonos: (593 2) 299-3300 - Fax: (593 2) 2993341  
www.quito.com.ec  
www.quito-turismo.gob.ec

Quito, 21 de Julio de 2011

Señor  
Ingeniero  
Mauricio Campaña  
**Coordinador de Carrera**  
**Departamento de Ciencias de la Computación**  
**Escuela Politécnica del Ejército**  
Presente.-

De mi consideración:

De acuerdo a la solicitud presentada por los Señores Marco Roberto Salazar Cepeda, portador de la cédula de ciudadanía No. 180234455-4 y Daniel Esteban Vallejo Prieto, portador de la cédula de ciudadanía No. 171744784-9, egresados de la Facultad de Ingeniería de Sistemas e Informática para desarrollar su tesis de grado con el apoyo de la Empresa Publica Metropolitana de Gestión de Destino Turístico - QUITO TURISMO, me permito poner en conocimiento que la empresa EMQT, otorga el auspicio para que se desarrolle el tema "**Análisis, Diseño e Implantación del Sistema de Catastros para la Empresa Pública Metropolitana QUITO TURISMO**".

La **Empresa Pública Metropolitana QUITO TURISMO**, brindará todo el apoyo requerido para el desarrollo del tema planteado, el mismo que es de gran importancia para nuestra organización y contribuirá al progreso tecnológico de la misma con la implantación del mencionado sistema.

Atentamente,

Alicia Díaz Campaña  
Jefe de Servicios Turísticos  
Empresa Publica Metropolitana QUITO TURISMO



**Quito**  
DISTRITO  
METROPOLITANO

## ANEXO F

### CARTA DE ACEPTACION DEL SISTEMA



#### CERTIFICADO DE ACEPTACIÓN DEL SISTEMA CATASTROSQT

Por este medio acepto de conformidad el Sistema de Control del Catastro Turístico para la Empresa Pública Metropolitana de Gestión del Destino Turístico (CatastrosQT), desarrollado por los señores Marco Roberto Salazar Cepeda y Daniel Esteban Vallejo Prieto como parte de su proyecto de tesis de grado, el mismo que cumple satisfactoriamente con los requerimientos de la EPMGDT expuestos en reuniones previas, y que ha sido validado en las pruebas funcionales realizadas por los colaboradores responsables del catastro turístico de la EPMGDT en las fechas 15 al 29 de Mayo del 2012.

Hemos validado que los procesos de registro y procesamiento de información del aplicativo CatastrosQT, así como las observaciones realizadas hasta el momento han sido implementados satisfactoriamente.

Atentamente,

Patricio Velásquez  
Director de Calidad y Asistencia Técnica  
Empresa Pública Metropolitana de Gestión del Destino Turístico



## **APÉNDICE**

### **BIOGRAFÍA**

#### **Marco Roberto Salazar Cepeda**

El Sr. Marco Roberto Salazar Cepeda, nació en Ambato, el 13 de febrero de 1980.

Cursó sus estudios escolares en la escuela Juan León Mera “La Salle” de Ambato en el año de 1991.

Sus estudios secundarios los realizó en el Colegio Diocesano “San Pio X” y se graduó de bachiller nacional con especialidad en físico-matemático, en el año 1997. Ese mismo año viajó a los Estados Unidos de Norte América, como estudiante de intercambio, en donde se graduó de bachiller general del estado de Nueva York en el año 2008.

En ese año ingresó a la Facultad de Ingeniería en Sistemas de la ESCUELA POLITECNICA DEL EJERCITO, donde realiza sus estudios en la carrera de Ingeniería de Sistemas e Informática. Finalmente, termina la carrera y egresa de la misma en el año 2003.

## **BIOGRAFÍA**

### **Daniel Esteban Vallejo Prieto**

El Sr. Daniel Vallejo, nació en Quito el 31 de octubre de 1983, cursó sus estudios escolares en la Unidad Educativa “BORJA #3” en el año de 1995.

Sus estudios secundarios los realizó en la Unidad Educativa “LA SALLE” y se graduó de bachiller nacional con especialidad en ciencias experimentales, en el año 2011.

Ese mismo año ingresó a la Facultad de Ingeniería de Sistemas e Informática de la ESCUELA POLITÉCNICA DEL EJERCITO de donde egresa en el año 2007.

**HOJA DE LEGALIZACIÓN DE FIRMAS**

**ELABORADA(O) POR**

---

Marco Roberto Salazar Cepeda

---

Daniel Esteban Vallejo Prieto

**DIRECTOR DE LA CARRERA**

---

Sr. Ing. Mauricio Campaña

Lugar y fecha: Sangolquí, julio del 2012