

# UNIVERSIDAD TÉCNICA DEL NORTE



**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TEMA:**

ESTUDIO DE LA TECNOLOGÍA WCF (WINDOWS COMMUNICATION FOUNDATION)  
EN EL DESARROLLO DE APLICACIONES EMPRESARIALES.

**AUTOR:**

LARA CASTILLO CHRISTIAN ORLANDO

**DIRECTOR:**

ING. DIEGO JAVIER TREJO ESPAÑA

**Ibarra – Ecuador**

**2018**

## UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA



### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

<b>DATOS DE CONTACTO</b>	
<b>CEDULA DE IDENTIDAD</b>	1002521415
<b>APELLIDOS Y NOMBRES</b>	CHRISTIAN ORLANDO LARA CASTILLO
<b>DIRECCIÓN</b>	IMBABURA - IBARRA – BARRIO LA FLORIDA
<b>EMAIL</b>	christianlara@hotmail.es
<b>TELÉFONO FIJO</b>	022604595
<b>TELÉFONO MÓVIL</b>	0998993726

<b>DATOS DE LA OBRA</b>	
<b>TITULO</b>	“ESTUDIO DE LA TECNOLOGÍA WCF (WINDOWS COMMUNICATION FOUNDATION) EN EL DESARROLLO DE APLICACIONES EMPRESARIALES.”
<b>AUTOR</b>	CHRISTIAN ORLANDO LARA CASTILLO
<b>FECHA</b>	23 NOVIEMBRE DEL 2018
<b>DIRECTOR</b>	ING. DIEGO TREJO

## 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD.

Yo, Christian **Orlando Lara Castillo**, con cédula de identidad No. 10025214145, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.



Nombre: Christian Orlando Lara Castillo

Cédula: 1002521415

Ibarra, 23 de noviembre del 2018



UNIVERSIDAD TÉCNICA DEL  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO**

Yo, Christian Orlando Lara Castillo, con cédula de identidad No. 1002522415, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: "ESTUDIO DE LA TECNOLOGÍA WCF (WINDOWS COMMUNICATION FOUNDATION) EN EL DESARROLLO DE APLICACIONES EMPRESARIALES." con el aplicativo: "IMPLEMENTACIÓN DE UN SISTEMA DE EMISIÓN DE PERMISOS DE FUNCIONAMIENTO DE LOCALES COMERCIALES EN LA CIUDAD DE OTAVALO.", que ha sido desarrollado para optar por el título de Ingeniero en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

Nombre: Christian Orlando Lara Castillo

Cedula: 1002521415

Ibarra, 23 de noviembre del 2018

## CERTIFICACIÓN DEL ASESOR

Certifico que la Tesis “ESTUDIO DE LA TECNOLOGÍA WCF (WINDOWS COMMUNICATION FOUNDATION) EN EL DESARROLLO DE APLICACIONES EMPRESARIALES.” con el aplicativo: “IMPLEMENTACIÓN DE UN SISTEMA DE EMISIÓN DE PERMISOS DE FUNCIONAMIENTO DE LOCALES COMERCIALES EN LA CIUDAD DE OTAVALO.”, ha sido realizada en su totalidad por el señor: Christian Orlando Lara Castillo, portador de la cédula de identidad número: 1002521415.



.....  
Ing. Diego Trejo

DIRECTOR DE TESIS

## CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

En la ciudad de Ibarra, 23 de noviembre del 2018.

El Autor:



Christian Orlando Lara Castillo

Cedula: 1002521415

## **DEDICATORIA**

Dedico el presente trabajo que es fruto del esfuerzo constante en primer lugar a mi Dios, el ser quien me guía en lo justo, ilumina mi vida y mis pensamientos.

Dedico mi triunfo con todo mi corazón a mis Padres, a mi querida Abuelita en especial, quien me enseñó que para ganar hay que luchar, pero sobre todo que, para tener éxitos en la vida, hay que ser perseverante y sentirse triunfador.

Igualmente, les dedico mi trabajo a aquellos amigos y amigas que como personas valiosas estuvieron ahí siempre aportando sus conocimientos, dándome su apoyo y animándome para hacer de este mi sueño posible.

A mí amada esposa, Doris Almeida que estuvo siempre brindándome su apoyo incondicional, su comprensión, su empeño, su fuerza y su inconmensurable amor.

## **AGRADECIMIENTO**

Un inmenso agradecimiento a la Universidad Técnica del Norte y a sus autoridades, por la oportunidad de poder culminar mi carrera, así como también a cada docente que transmitió sus valiosos conocimientos y experiencias en el camino de mi vida universitaria, de igual forma a mi asesor Ing. Diego Trejo por brindarme su paciencia y dedicación para seguir con mi proyecto y así lograr mi objetivo.

Agradezco también al Cuerpo de Bomberos de Otavalo, por haberme permitido hacer mi investigación para la implementación del sistema y darme la oportunidad de aportar a la comunidad con un granito de arena con el presente trabajo.

A mis compañeros de la Universidad y de mi vida les doy las gracias pues siempre me han demostrado que son amigos leales en los buenos y malos momentos, de los que siento una gran admiración por su amistad incondicional al brindarme su apoyo constante.



## RESUMEN

El Presente Proyecto de Tesis está orientado a la investigación de la tecnología WCF (Windows Communication Foundation) en el desarrollo de aplicaciones empresariales, donde el aplicativo está enfocado en un sistema que le permita a la Institución llamada: Cuerpo de Bomberos de Otavalo, controlar varios aspectos de los procesos de control que están involucrados en la generación de Permisos de funcionamiento hacia los locales comerciales de la ciudad de Otavalo y sus alrededores.

Las características de la arquitectura tecnológica WCF (Windows Communication Foundation); de la mano con la implementación del sistema, permiten cumplir con los estándares necesarios para poder solventar soluciones de alta fiabilidad, basados en un consolidado framework 4.5, servidor de base de datos SQL Server Express 2012 y la flexibilidad del lenguaje de programación C# de .NET.

En las diferentes fases del presente proyecto se plasma la correcta estructuración de artefactos brindados por la metodología RUP para facilitar su desarrollo que incluye el análisis y diseño, sus respectivas actividades de implementación y pruebas, toda esta información se detalla en el presente trabajo documentando paso a paso, referenciando en cada capítulo la aplicación de esta tecnología en el desarrollo del sistema.

Al finalizar se redacta las conclusiones y recomendaciones que ayudaran al lector a comprender los resultados de la investigación desarrollada.

## SUMMARY

The present thesis project is aimed at investigating the WCF (Windows Communication Foundation) technology in the development of business applications, where the application is focused on a system that will allow the institution called: "Departamento de Bomberos de Otavalo", to control multiple aspects of control processes that are involved in the generation of operating licenses to commercial premises of the city of Otavalo and its surroundings.

The characteristics of the technological architecture WCF (Windows Communication Foundation); hand in hand with the implementation of the system, capable of meeting the necessary standards to solve highly reliable solutions based on a framework 4.5, consolidated database server SQL Server Express 2012 and the flexibility of the programming language C # .NET.

In the different phases of this project the correct structuring of artifacts provided by the RUP methodology facilitating their development including analysis and design, their implementation and testing activities, all this information is detailed in this paper documenting step step, referencing each chapter the application of this technology in the development of the system.

At the end we have the conclusions and recommendations that will help the reader understand the results of research carried out it is drawn up.

## INDICE DE CONTENIDO

AUTORIZACIÓN DE USO Y PUBLICACIÓN .....	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO .....	¡Error! Marcador no definido.
CERTIFICACIÓN DEL ASESOR .....	iv
CONSTANCIAS .....	¡Error! Marcador no definido.
DEDICATORIA .....	vii
AGRADECIMIENTO .....	viii
RESUMEN .....	ix
SUMMARY .....	x
INDICE DE CONTENIDO .....	xi
ÍNDICE DE FIGURAS .....	xv
ÍNDICE DE TABLAS .....	xvii
CAPITULO I .....	1
CAPITULO I .....	2
1. INTRODUCCIÓN .....	2
1.1. Antecedentes .....	2
1.2. Problema .....	3
1.3. Objetivos del Proyecto .....	3
1.3.1. Objetivo General .....	3
1.3.2. Objetivos Específicos .....	3
1.4. Justificación .....	4
1.5. Alcance .....	6
1.5.1. Modulo Administrativo del Sistema .....	7
1.5.2. Módulo de Clientes .....	7
1.5.3. Módulo de Locales Comerciales .....	7
1.5.4. Módulo de Permisos de Funcionamiento .....	8
1.5.5. Módulo de Reportes Generales .....	8
1.5.6. Módulo de Inspecciones .....	8
1.5.7. Módulo de Conciliación Tesorería .....	8
1.5.8. Modulo para Gestión, Registro y Control de Personal. ....	9
CAPITULO II .....	11
2. MARCO TEÓRICO .....	11
2.1. Metodología RUP .....	11
2.2. Definición de WCF .....	12
2.3. Conceptos Básicos de WCF .....	12
2.3.1. Contrato de Servicio .....	12
2.3.2. Implementación del Contrato del Servicio .....	13
2.3.3. Alojamiento del servicio .....	13
2.3.4. Exposición de Puntos de acceso .....	14

2.3.5.	Arquitectura de WCF .....	15
2.3.6.	Contratos .....	16
2.3.7.	Mensajería .....	18
2.3.8.	Alojamiento y Activación .....	19
2.4.	Programación Básica de WCF .....	20
2.4.1.	Ciclo de vida de programación básica .....	20
2.4.1.1.	Definir el contrato de servicio (Microsoft, Ciclo de vida, 2013) .....	20
2.4.1.2.	Implementar el contrato .....	21
2.4.1.3.	Configurar el servicio.....	22
2.4.1.4.	Hospedar el servicio .....	22
2.4.2.	Diseño e implementación de servicios.....	22
2.4.3.	Configuración de servicios.....	23
2.4.4.	Servicios de Hospedaje .....	23
2.4.5.	Creación de Clientes (Microsoft, Creación de clientes, 2013).....	24
2.5.	Acerca de la Institución: Benemérito Cuerpo de Bomberos de Otavalo..	25
2.5.1.	Breve Historia.....	25
2.5.2.	Misión .....	26
2.5.3.	Visión.....	26
2.5.4.	Organigrama.....	27
2.5.5.	Servicios a la comunidad .....	27
2.5.5.1.	Proteger a la Comunidad.....	27
2.5.5.2.	Capacitaciones a estudiantes .....	27
2.5.5.3.	Capacitaciones Internas .....	28
2.5.5.4.	Control y Emisión de Permisos de Funcionamiento.....	28
<b>CAPITULO III .....</b>		<b>31</b>
<b>3.</b>	<b>ANÁLISIS Y DISEÑO .....</b>	<b>31</b>
3.1.	Requisitos .....	31
3.1.1.	Requerimientos Funcionales .....	31
3.1.2.	Requerimientos No Funcionales .....	33
3.2.	Casos de Uso .....	35
3.2.1.	Vista Global del Sistema con sus Actores y Casos de Uso.....	35
3.2.2.	Módulo Administrativo.....	36
3.2.2.1.	Caso de Uso: Mantener Módulos del Sistema .....	36
3.2.2.2.	Caso de Uso: Mantener Dispositivos en el Sistema.....	38
3.2.2.3.	Caso de Uso: Mantener Catálogo de Errores del Sistema.....	41
3.2.2.4.	Caso de Uso: Mantener Usuarios del Sistema.....	43

3.2.2.5. Caso de Uso: Iniciar Sesión.....	46
3.2.3. Caso de Uso: Mantener Clientes.....	48
3.2.4. Caso de Uso: Mantener Empleados .....	51
3.2.5. Caso de Uso: Mantener Locales comerciales .....	54
3.2.6. Caso de Uso: Registrar Informe de Inspección .....	56
3.2.7. Caso de Uso: Generar Permiso de Funcionamiento .....	58
3.2.8. Caso de Uso: Generar Conciliación Permisos y Rubros.....	60
3.2.9. Caso de Uso: Control de Asistencia .....	62
3.2.10. Caso de Uso: Mantener Tipos de Empleados.....	64
3.2.11. Caso de Uso: Mantener Tipos de Locales Comerciales .....	66
3.2.12. Caso de Uso: Mantener Tarifas Locales Comerciales.....	68
3.3. Modelo Conceptual del Sistema.....	70
3.4. Diseño Físico de la Base de Datos .....	75
3.5. Diagrama de Arquitectura del Sistema.....	77
3.6. Diseño de la Interfaz Gráfica .....	78
<b>CAPITULO IV.....</b>	<b>80</b>
<b>4. IMPLEMENTACIÓN .....</b>	<b>80</b>
4.1. Instalación y configuración de la Base de Datos SQL Server Express 2012. 80	
4.1.1. Requisitos del Sistema.....	80
4.1.2. Instalación.....	80
4.2. Implementación de la Base de Datos.....	85
4.2.1. Creación de Base de datos, Tablas y Relaciones. ....	85
4.2.1.1. Creación de la Base de Datos .....	85
4.2.1.2. Creación de las tablas del Sistema .....	87
4.2.2. Creación de los procedimientos almacenados Create, Read, Update, Delete (CRUD). ....	92
4.2.2.1. Create Registro de Cliente:.....	92
4.2.2.2. Read registro de Cliente.....	94
4.2.2.3. Update Registro de Cliente.....	95
4.2.2.4. Delete registro de Cliente.....	96
4.2.3. Creación y configuración del Proyecto en Capas en Visual Studio 2012 Xpress. 98	
4.2.3.1 DTO (Data Transfer Object). ....	99
4.2.3.2 DAL (Data Access Layer). ....	99
4.2.3.3 BLL (Business Logic Layer). ....	100
4.2.3.4 Servicio WCF (Windows Communication Foundation).....	102

4.3	Implementación de las Capas del Sistema.....	103
4.3.1.	DTO (Data Transfer Objctcs).....	103
4.3.2.	DAL (Data Access Layer).....	103
4.3.3.	BLL (Businnes Logic Layer).....	104
4.3.4.	WCF (Clase Cliente).....	104
4.3.4.1.	Data Contract.....	104
4.3.4.2.	Operation Contract.....	105
4.3.4.3.	Service Contract.....	105
4.4.	Publicación y Alojamiento .....	109
4.5.	Etapa de Pruebas .....	114
<b>CAPITULO V.....</b>		<b>118</b>
<b>5. CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>118</b>
5.1.	Conclusiones.....	118
5.2.	Recomendaciones.....	120
5.3.	Análisis de Impactos .....	121
5.3.1.	Proceso de Generación de permisos de Funcionamiento .....	121
5.3.2.	Registro y Actualización de Información.....	121
5.1.1.	Generación de Informes y Reportes .....	122
5.3.3.	Resumen de Análisis de Impactos .....	122
<b>BIBLIOGRAFÍA .....</b>		<b>124</b>
<b>Glosario de términos: .....</b>		<b>126</b>
<b>Anexo 1: Manual de Usuario. ....</b>		<b>132</b>

## ÍNDICE DE FIGURAS

Figura 2- 1 Arquitectura WCF (Microsoft, 2012).....	16
Figura 2- 2 Cuerpo de Bomberos de Otavalo (Cuerpo de bomberos de Otavalo, 2013) .	26
Figura 2- 3 Organigrama Institucional (Cuerpo de bomberos de Otavalo, 2013).....	27
Figura 3- 1 Vista Global del Sistema: Actores y Casos de Uso. (Fuente Propia, 2017)...	35
Figura 3- 2 Caso de Uso: Mantener Módulos del Sistema (Fuente Propia, 2017) .....	36
Figura 3- 3 Caso de Uso: Mantener Dispositivos en el Sistema (Fuente Propia, 2017)..	38
Figura 3- 4 Caso de Uso: Mantener Catálogo de Errores del Sistema (Fuente Propia, 2017) .....	41
Figura 3- 5 Caso de Uso: Mantener Usuarios del Sistema (Fuente Propia, 2017) .....	43
Figura 3- 6 Caso de Uso: Iniciar Sesión (Fuente Propia, 2017) .....	46
Figura 3- 7 Caso de Uso: Mantener Clientes (Fuente Propia, 2017) .....	48
Figura 3- 8 Caso de Uso: Mantener Empleados (Fuente Propia, 2017) .....	51
Figura 3- 9 Caso de Uso: Mantener Locales Comerciales (Fuente Propia, 2017) .....	54
Figura 3- 10 Caso de Uso: Registrar Informe de Inspección (Fuente Propia, 2017).....	56
Figura 3- 11 Caso de Uso: Generar Permiso de Funcionamiento (Fuente Propia, 2017)	58
Figura 3- 12 Caso de Uso: Generar Conciliación Permisos y Rubros (Fuente Propia, 2017) .....	61
Figura 3- 13 Caso de Uso Iniciar Control de Asistencia (Fuente Propia, 2017).....	62
Figura 3- 14 Caso de Uso: Mantener Tipos de Empleado. (Fuente Propia, 2017) .....	64
Figura 3- 15 Caso de Uso: Mantener Tipos de Locales Comerciales. (Fuente Propia, 2017) .....	66
Figura 3- 16 Caso de Uso: Mantener Tarifas Locales Comerciales.....	68
Figura 3- 17 Simbología Modelo Conceptual (Fuente Propia, 2017) .....	72
Figura 3- 18 Modelo Conceptual del Sistema. (Fuente Propia, 2017) .....	74
Figura 3- 19 Diseño Físico de la Base de Datos (Fuente Propia, 2017) .....	76
Figura 3- 20 Diagrama de Arquitectura del Sistema. (Fuente Propia, 2017) .....	77
Figura 3- 21 Interfaz Gráfica: Registrar. Modificar Usuarios del Sistema. (Fuente Propia, 2017) .....	78
Figura 4- 1 Selección Tipo de Instalación Sql Server Xpress 2012. ....	81
Figura 4- 2 Selección de Características Sql Server Xpress 2012.....	82
Figura 4- 3 Configuración Instancia Sql Server Xpress 2012. ....	82
Figura 4- 4 Configuración Servidor Sql Server Xpress 2012. ....	83
Figura 4- 5 Configuración Motor de BDD Sql Server Xpress 2012. ....	83
Figura 4- 6 Configuración de Reporting Services Sql Server Xpress 2012. ....	84

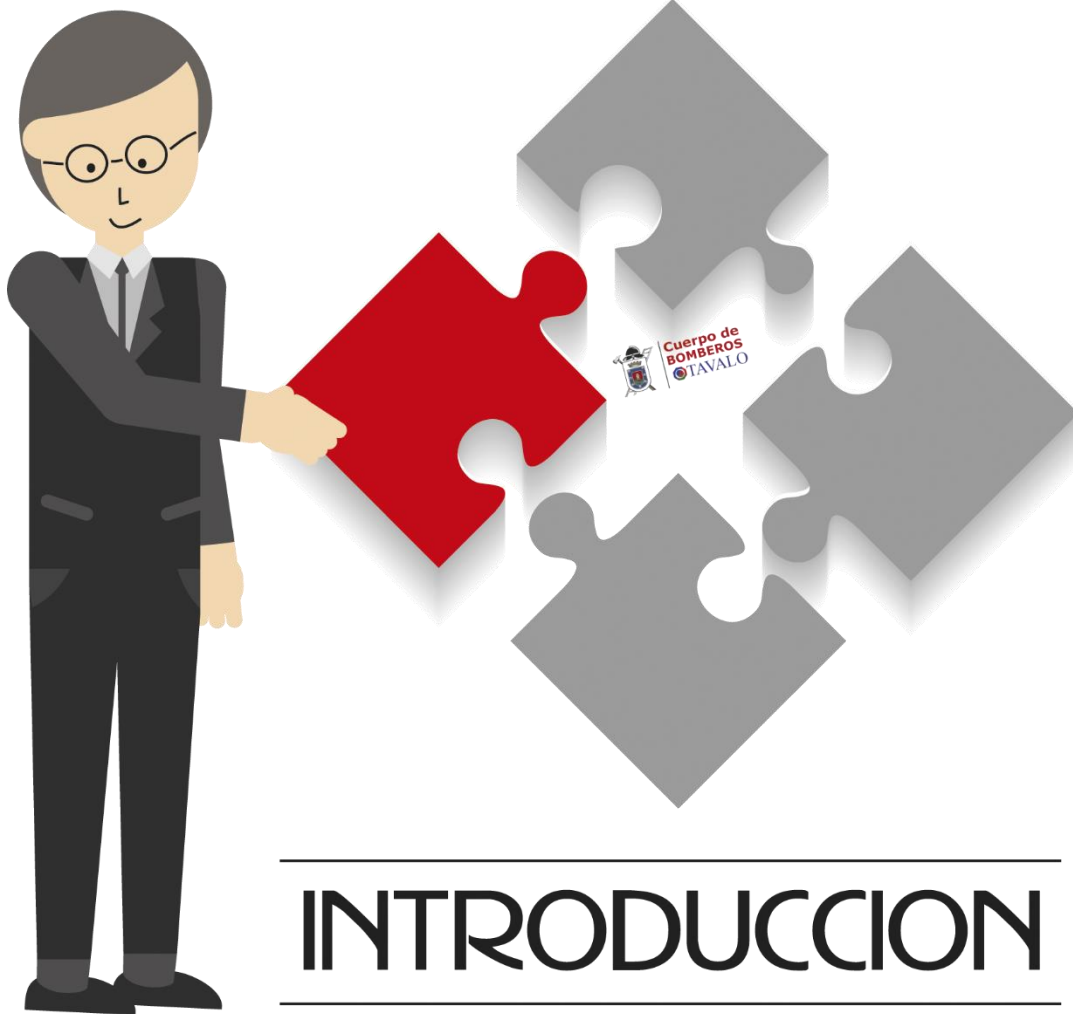
Figura 4- 7 Operación Completa Instalación Sql Server Xpress 2012. ....	85
<i>Figura 4- 8 Ruta Acceso Sql Server Management Studio 2012. (Fuente: Propia., 2015).</i>	85
Figura 4- 9 Iniciar Sesión Sql Server Management Studio 2012.....	86
Figura 4- 10 Crear Base de Datos Sql Server Management Studio 2012. ....	87
Figura 4- 11 Crear tablas y relaciones para: modulos_sistema, dispositivos_sistema, mensajes_error, log_error.....	88
Figura 4- 12 Crear tablas y relaciones para: tipos_empleado, empleados, usuarios_sistema, control_asistencia, time_stamp.....	90
Figura 4- 13 Crear tablas y relaciones para: tipos_local, tarifas_tipos_local, locales, permisos, conciliacion_recaudacion_permisos. ....	91
Figura 4- 14 Crear tablas y relaciones para: inspecciones, clientes.....	92
Figura 4- 15 Configuración de Capas de las clases del proyecto. ....	98
Figura 4- 16 Configuración de la capa de DTO para las clases del proyecto. ....	99
Figura 4- 17 Configuración de la capa DAL para las clases del proyecto.....	100
Figura 4- 18 Configuración de la capa de BLL para las clases del proyecto.....	101
Figura 4- 19 Configuración de la capa de Servicios WCF del proyecto. ....	102
<i>Figura 4- 20 Configuración del Servicio WCF: Aplicación.</i> .....	106
Figura 4- 21 Configuración del Servicio WCF: Web. ....	107
Figura 4- 22 Configuración del Servicio WCF: Preparar Servicio WCF para Alojamiento.....	109
Figura 4- 23 Publicación y Alojamiento del Servicio WCF: Perfil. ....	110
Figura 4- 24 Publicación y Alojamiento del Servicio WCF: Conexión.....	110
Figura 4- 25 Publicación y Alojamiento del Servicio WCF: Configuración.....	111
Figura 4- 26 Publicación y Alojamiento del Servicio WCF: Vista Previa. ....	111
Figura 4- 27 Publicación y Alojamiento del Servicio WCF: Verificación física. ....	112
Figura 4- 28 Publicación y Alojamiento del Servicio WCF:.....	112
Figura 4- 29 Publicación y Alojamiento del Servicio WCF:.....	113
Figura 4- 30 Publicación y Alojamiento del Servicio WCF: Descubrir entrada WSDL. ....	114
Figura 4- 31 Etapa de Pruebas: Configurar End Point con SOAP UI .....	114
Figura 4- 32 Etapa de Pruebas: End Point con las Transacciones.....	115
Figura 4- 33 Etapa de Pruebas: Ejecución de request servicio WCF .....	115
Figura 4- 34 Etapa de Pruebas: Verificación de afectación tabla clientes. ....	116



## ÍNDICE DE TABLAS

Tabla 2 - 1 ventajas y escenarios clave opciones de hospedaje para WCF .....	24
Tabla 3 - 1 Requerimientos Funcionales (Fuente: Propia., 2017).....	33
Tabla 3 - 2 Requerimientos no Funcionales. (Fuente: Propia., 2017) .....	34
Tabla 3 - 3 Tiempo de Entrega de Permisos (manual). (Fuente: Propia., 2017) .....	121
Tabla 3 - 4 Registro y actualización de Información (manual). (Fuente: Propia., 2017).	122
Tabla 3 - 5 Generación de Informes y reportes. (Fuente: Propia., 2017).....	122
Tabla 3 - 6 Tiempo de Entrega de Permisos (manual). (Fuente: Propia., 2017) .....	123

## CAPITULO I



---

# INTRODUCCION

---

## **CAPITULO I**

### **1. INTRODUCCIÓN**

#### **1.1. Antecedentes**

En la ciudad de Otavalo, acorde al crecimiento económico y social el cual a la par de las necesidades turísticas, el crecimiento comercial se ha incrementado el número de locales comerciales que brindan diferentes servicios y productos a las personas nacionales y extranjeras en los últimos años, por lo que el Benemérito Cuerpo de Bomberos se ha visto en la necesidad de llevar un control en las actividades referentes a los reglamentos y actividades relacionadas con la seguridad que deben tener en cada uno de los locales comerciales para salvaguardar la integridad del usuario del producto y servicio así como la del propietario, por lo que se lleva a cabo inspecciones regulares y controles a cada establecimiento.

Resultado de los controles es la generación de información acerca del estado actual de cada local comercial en la ciudad de Otavalo, así como la generación de nueva información en los casos donde se crean nuevos registros asociados con la actividad económica que se genera en el sector.

Los registros de inspecciones por parte de los empleados encargados se llevan de manera manual y se archivan sin un debido control, es difícil de encontrar el archivo físico cuando se necesita expedir un permiso de funcionamiento para comprobar si el local comercial supero los controles, lo que genera malestar en el cliente por la demora al buscarlo.

De igual manera no existe información exacta de la cantidad de locales comerciales que existen en la ciudad de Otavalo y sus alrededores lo que genera inconvenientes a la administración al momento de gestionar el personal para realizar las inspecciones.

En general todos los registros y gestión de empleados, clientes, locales comerciales, permisos de funcionamiento se los lleva de forma manual lo que conlleva a pérdida de tiempo en buscar información y una demora en los trámites para los clientes.

## **1.2. Problema**

Imposibilidad para manejo remoto de la información de la institución, ya que se requiere gestionar la emisión de permisos de funcionamiento en otras localidades de la ciudad.

Aglomeración de personas en el área administrativa de la institución ya que existe una sola persona y un sistema centralizado (una sola maquina) que se encarga de registrar manualmente la información de los clientes que requieren de un permiso de funcionamiento.

Lentitud en el proceso de emisión del permiso de funcionamiento ya que carecen de algunos procesos para registrar la información necesaria, además de existir información dispersa, que entorpece la generación del mismo.

## **1.3. Objetivos del Proyecto**

### **1.3.1. Objetivo General**

Estudiar la tecnología WCF en el Desarrollo de Aplicaciones Empresariales mediante la implementación de un aplicativo para emisión de permisos de funcionamiento de locales comerciales en la ciudad de Otavalo.

### **1.3.2. Objetivos Específicos**

- a) Investigar las herramientas del Framework.Net, WCF (Windows Communication Foundation) como marco de trabajo para la creación de aplicaciones orientadas a servicios.

- b) Implementar el aplicativo de Emisión de permisos de funcionamiento, basado en el marco de trabajo dado por WCF
- c) Aplicar metodología de desarrollo RUP.
- d) Aplicar patrones de diseño para la búsqueda de soluciones en el desarrollo del aplicativo.
- e) Validar la tecnología aplicada mediante la implantación y funcionamiento del aplicativo.
- f) Documentar la información recabada en esta investigación.

#### **1.4. Justificación**

- a) Este proyecto tendrá como finalidad ayudar a mejorar las funcionalidades del sistema para que las mismas puedan apuntar a una arquitectura orientada a Servicios, con lo que se logrará interoperabilidad, permitiendo que cualquier cliente creado en cualquier plataforma pueda conectarse con cualquier servicio.
- b) Emitir permisos de funcionamiento interconectando inicialmente un cliente ubicado de manera remota que podrá consumir los servicios para generación de permisos y consultas.
- c) Con la funcionalidad implementada el sistema podrá exponer sus servicios a diversas áreas administrativas remotas, mejorando significativamente la gestión de la información según las necesidades del manejo del negocio.
- d) Los clientes de la ciudad de Otavalo tendrán la posibilidad de hacer sus pagos en diferentes puntos de la ciudad en los que se configure un cliente que consuma los servicios expuestos por el sistema con esta tecnología, por lo que se agilizarán los procesos y se dará al cliente un mejor servicio

- e) Mejor arquitectura del sistema debido a que será adaptable, escalable y con la característica de poder ser mantenido a lo largo del tiempo para diferentes soluciones distribuidas.
- f) Mayor rapidez, seguridad y eficiencia en la emisión de los permisos de funcionamiento ya que con esta tecnología se pueden diversificar los puntos donde el cliente puede acceder a los servicios de pago que brinda la Institución.
- g) Eficiente uso de las comunicaciones ya que se puede usar diferentes tipos de bindings http que ofrece WCF, lo que permite inter operar entre distintas plataformas.
- h) WCF como tecnología es parte de .net framework en la cual otras tecnologías de comunicación de Microsoft se apegan debido al soporte que brinda para sus aplicaciones como lo son:
- **Cloud Workloads** - Escalamiento de aplicaciones para la Nube con el .NET Framework 4.5 en Windows Azure.
  - **Synchronous-Async support (Soporte Asíncrono).**- Los beneficios y el modelo síncrono y asíncrono en WCF son muy similares a ASP.NET, y permiten ampliar o reducir sus servicios mucho mejor en el servidor. También se ha habilitado la generación automática de métodos especiales para las operaciones asíncronas del servicio, que permite a las aplicaciones cliente recuperar fácilmente las operaciones de servicio de forma asíncrona.
  - **Visual Studio 2012** (última versión)
  - **Aplicaciones WPF** - Servicio de datos de WCF que expone los datos de Entity Data Model en una aplicación WPF.

- Los desarrolladores de servicios pueden crearlos con **Windows Communication Foundation** (WCF). Visual Studio 2012 facilita más que nunca el desarrollo de servicios de WCF con compatibilidad integrada para Web API.
- Soporte para Web Sockets (WCF and ASP.Net)
- Mapeo del protocolo HTTPS en WCF
- Soporte para streaming asíncrono en WCF.
- **Punto NET Framework 4.5.-** Es una actualización en contexto altamente compatible de .NET Framework 4. El uso conjunto de .NET Framework 4.5 y los lenguajes de programación C#, Visual Basic o F# le permite escribir aplicaciones Windows. .NET Framework 4.5 incluye mejoras importantes en el lenguaje y en la plataforma para C#, Visual Basic y F# que simplifican la escritura de código asíncrono y le permiten combinar flujo de control en el código sincrónico, además de proporcionar una interfaz de usuario con gran capacidad de respuesta y escalabilidad de las aplicaciones web. .NET Framework 4.5 agrega mejoras sustanciales a otras áreas funcionales como ASP.NET, Managed Extensibility Framework, Windows Communication Foundation, Windows Workflow Foundation y Windows Identity Foundation. .NET Framework 4.5 proporciona mejor rendimiento, confiabilidad y seguridad.

### 1.5. Alcance

Se implementará lógica de negocio para el manejo de la información para los módulos: ADMINISTRACION DEL SISTEMA, CLIENTES, LOCALES COMERCIALES, PERMISOS DE FUNCIONAMIENTO, REPORTES GENERALES, INSPECCIONES, CONCILIACIÓN TESORERÍA, GESTIÓN, REGISTRO Y CONTROL DE PERSONAL mediante los siguientes módulos:

### **1.5.1. Modulo Administrativo del Sistema**

Se encargará de validar los usuarios que acceden al sistema, así como de la administración y funcionalidad general de los módulos del sistema.

- a) Verificar Usuarios del sistema
- b) Validar Estado del Usuario del sistema
- c) Configuración de parámetros generales del sistema
- d) Manejo, parametrización y customización de errores controlados para cada módulo del sistema
- e) Registro de Log de errores para auditoría y control del sistema
- f) Configuración y manejos de acceso desde dispositivos al sistema

### **1.5.2. Módulo de Clientes**

Se encarga de manipular la información de los clientes propietarios de los locales comerciales

- g) Registrar cliente
- h) Modificar cliente
- i) Dar de baja cliente
- j) Mostrar cliente

### **1.5.3. Módulo de Locales Comerciales**

Se encarga de manipular la información de Locales Comerciales los que son dueños de los locales comerciales

- k) Registrar local comercial
- l) Modificar local comercial
- m) Dar de baja local comercial
- n) Mostrar local comercial



#### **1.5.4. Módulo de Permisos de Funcionamiento**

Se encarga de generar la información para la creación y legalización del Permiso de Funcionamiento para los Locales ya registrados.

- o) Registrar Permiso de Funcionamiento
- p) Generar Comprobante de Pago

#### **1.5.5. Módulo de Reportes Generales**

Se encarga de generar los reportes generales para:

- q) Clientes y Locales Registrados
- r) Locales Comerciales y estado de Permisos de Funcionamiento
- s) Reporte de Inspecciones a Locales comerciales por dirección
- t) Reportes de conciliación y cuadros con Tesorería.

#### **1.5.6. Módulo de Inspecciones**

Se encarga de registrar, controlar y conciliar los reportes físicos generados por los inspectores de los locales comerciales de cada localidad:

- u) Locales Comerciales Registrados
- v) Permisos de Funcionamiento
- w) Inspectores y reportes físicos.

#### **1.5.7. Módulo de Conciliación Tesorería**

Se encarga de enlazar, conciliar y cuadrar procesos referentes a la recaudación de valores generados en el proceso de emisión de Permisos de funcionamiento:

- x) Permisos de funcionamiento
- y) Recaudaciones

### **1.5.8. Modulo para Gestión, Registro y Control de Personal.**

Se encarga de registrar, controlar y gestionar los ámbitos:

- z) Control y gestión de Asistencia (mediante huella digital o acceso Biométrico), Permisos, cambios de guardia del personal de la institución.
- aa) Gestión de localidades para el personal que está encargado del proceso de inspección de locales comerciales dependiendo del sector y área de acción.



---

# MARCO TEORICO

---

## CAPITULO II

### 2. MARCO TEÓRICO

#### 2.1. Metodología RUP

La metodología RUP es un proceso de desarrollo de software que conjuntamente con el lenguaje unificado UML que constituye una metodología estándar para el análisis, diseño implementación y documentación de sistemas orientados a objetos.

“ (Ivar jacobson, 1998)”

RUP comprende 2 aspectos importantes por los cuales se establecen las disciplinas:

##### **Proceso**

Las etapas de esta sección son:

- Modelado de Negocio
- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue

##### **Soporte**

En esta parte nos encontramos con las siguientes etapas:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP es la que permite que sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las cuatro fases descritas anteriormente:

- Inicio (también llamado Incepción o Concepción)
- Elaboración
- Desarrollo (también llamado Implementación, Construcción)

- Cierre (también llamado Transición)

(wikipedia, 2018)

## 2.2. Definición de WCF

WCF es un modelo de programación unificado orientado a servicios, el cual fue incluido en .NET 3.0 donde se intercambian las plataformas de interconexión tener la idea de un único framework para construir aplicativos conectados, lo cual permite implementar una arquitectura adaptable, escalable y con la característica de ser mantenido en el tiempo para las soluciones distribuidas. (Maximiliano Firtman, Leonardo Natale, 2010)

Mediante los estilos de comunicación provistos por el framework WCF permitirá que clientes de diversas plataformas consuman los servicios que se decida exponer a través del intercambio de mensajes.

## 2.3. Conceptos Básicos de WCF

### 2.3.1. Contrato de Servicio

Es una clase que va a definir qué métodos se va a exponer en WCF, para ese fin se usa 2 atributos decoradores: (Maximiliano Firtman, Leonardo Natale, 2010)

`[ServiceContract]`: Decora la clase servicio que se va a utilizar. (Maximiliano Firtman, Leonardo Natale, 2010)

`[OperationContract]`: Decora los métodos que serán publicados para que los clientes puedan consumirlo, sin este atributo los métodos serán inaccesibles. (Maximiliano Firtman, Leonardo Natale, 2010)

Para permitir la flexibilidad a la hora de su implementación esta clase debe ser generalmente una interface que pueda definir más de un contrato en la clase servicio.

Además de estos atributos existen otros que van a permitir serializar tipos de datos no primitivos; `[DataContract]/[DataMember]`, con estos atributos es posible definir los miembros que se utilizarán en las comunicaciones ya que determinarán un tipo de dato que se define para el intercambio de los mensajes entre el cliente y el servicio.

### 2.3.2. Implementación del Contrato del Servicio

En la implementación del contrato para el servicio se va a implementar una interface, que implemente la clase servicio que contendrá cada contrato

La idea es proporcionar un nivel alto de orientación conceptual para diseñar e implementar los servicios de WCF. Antes de diseñar e implementar su aplicación WCF, se recomienda lo siguiente:

- Entender lo que es un contrato de servicio, cómo funciona, y cómo crear uno. (Microsoft, Contratos, 2013)
- Comprender que los contratos especifican requisitos mínimos que la configuración en tiempo de ejecución o el entorno de alojamiento pueden no admitir.

### 2.3.3. Alojamiento del servicio

Con respecto al alojamiento del servicio se puede hacer uso de las alternativas o procesos de Windows que brinda WCF para poder albergar los servicios que se expondrán para que los clientes que requieran consumirlos puedan acceder de manera sencilla a sus funcionalidades. (Maximiliano Firtman, Leonardo Natale, 2010)

Las opciones de alojamiento de los servicios son:

**Internet Information Server (IIS).** - En este caso se recurre a la versión de 7.0 ya que soporta además de HTTP las opciones de TCP, MSMQ y pipes.

**Windows Process Activation Service (WAS).** - Proporciona soporte para otros protocolos sin tener que recurrir a IIS, como un componente Windows independiente de IIS convirtiéndose en un mecanismo para dar vida a los servicios de WCF con mínimas configuraciones. (Maximiliano Firtman, Leonardo Natale, 2010)

**Servicio Windows.** - Haciendo uso de un objeto *ServiceHost*, se gestiona el ciclo de vida del servicio donde es encargado (Maximiliano Firtman, Leonardo Natale, 2010)

**Aplicación.** - Usando *ServiceHost* se puede hacer que el servicio sea mantenido en un formulario o en una aplicación de consola, esta opción puede ser usada para realizar pruebas o depuración en caso de presentarse problemas en las funcionalidades internas en el servicio.

#### 2.3.4. Exposición de Puntos de acceso

Para la comunicación entre un cliente y el servicio se hace uso de puntos de acceso denominados *Endpoints*, los cuales permiten acceder a las operaciones que son propias de las clases que se están exponiendo en el servicio.

Para configurar los servicios es necesario de 3 piezas fundamentales en los cuales se especifica donde se encuentran, como conectarse a ellos y que es lo que exponen: (Microsoft, Endpoints: Addresses, Bindings, and Contracts, 2013)

**Dirección [Address].** - Define la ruta del servicio para ser consumido por parte del cliente, donde se indica una URL donde se encuentra el recurso, protocolo usado, nombre de la computadora donde se encuentra el servicio, número de puerto (opcional) del protocolo que se está empleando. Ejemplo: (Maximiliano Firtman, Leonardo Natale, 2010)

`http://localhost:9095/BrowserWeb/BrowserServlet`

**Vinculación [Binding].** - En la vinculación se debe especificar como el mundo exterior se comunica con un servicio, en este momento es donde se podrá configurar varios aspectos como seguridad, protocolos de transporte, comunicación sincrónica o asincrónica etc (Maximiliano Firtman, Leonardo Natale, 2010)

Algunos de los bindings ya definidos por WCF se podrán usar según sus características las cuales ya están listas para ser empleadas: (Maximiliano Firtman, Leonardo Natale, 2010).

- BasicHttpBinding: Cumple con la especificación WS-I Basic Profile 1.1
- WSHttpBinding: Cumple con la especificación WS-\* para soportar transacciones distribuidas, http, https, seguridad, codificación de los mensajes usando MTOM para datos binarios, etc.
- NetTcpBinding: TCP es el protocolo de transporte para los mensajes mediante codificación binaria que requiera un buen performance en una LAN entre computadoras que usen sistema operativo de Windows.
- NetNamedBinding: En este caso emplea named pipes para las comunicaciones entre procesos en una misma computadora, soporta seguridad y transacciones
- Contract: Define que operaciones va a exponer el servicio hacia el exterior.

### **2.3.5. Arquitectura de WCF**

Como cualquier arquitectura definida en el software, WCF se compone de diferentes capas que conforman su funcionalidad la cual se va a describir a continuación de acuerdo al gráfico siguiente:



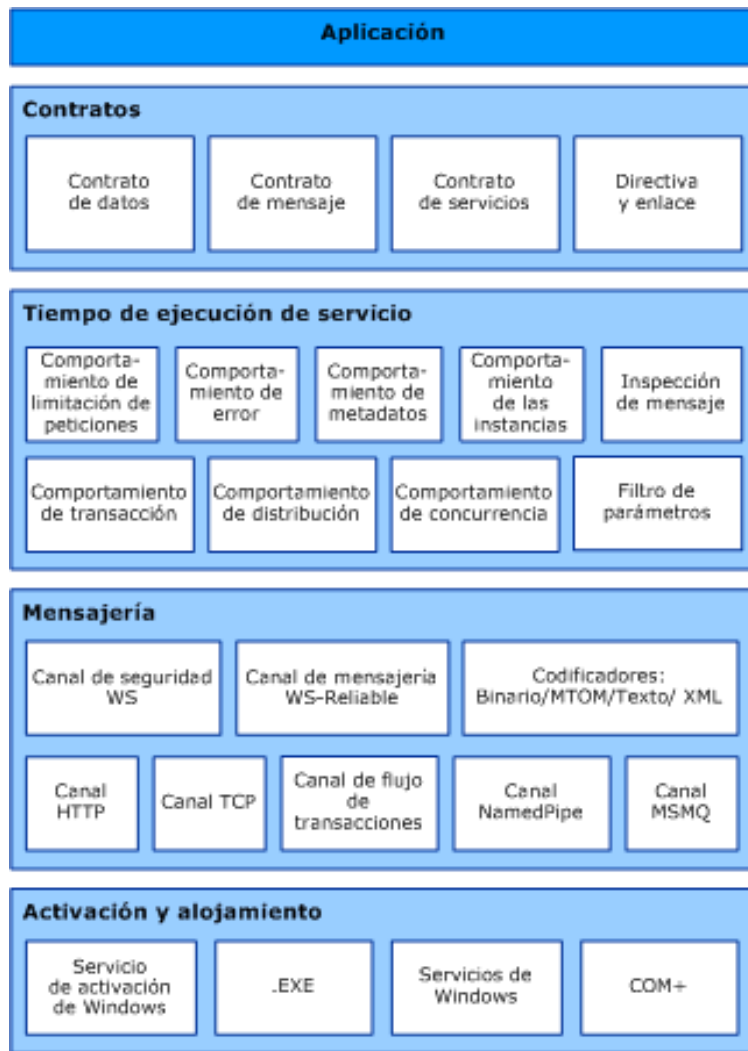


Figura 2- 1 Arquitectura WCF (Microsoft, *Arquitectura de WCF*, 2013)

<https://i-msdn.sec.s-msft.com/dynimg/IC5864.jpeg>

### 2.3.6. Contratos

Los contratos especifican varios aspectos del sistema de mensajes que va a usar el sistema:

**Contrato de Datos.** - Describe cada parámetro que constituye cada mensaje que un servicio puede crear o utilizar, mediante Lenguaje de definición de esquemas XML (XSD) se definen los parámetros de mensaje, permitiendo a cualquier sistema que entienda XML procesar los documentos. (Microsoft, *Data Contracts*, 2013)

**Contrato del Mensaje.** - Define partes específicas del mensaje utilizando los protocolos SOAP y permite el control más fino sobre las partes del mensaje, cuando la interoperabilidad exige tal precisión.

**Contrato de Servicios.** - especifica las firmas de método actuales del servicio y se distribuye como una interfaz en uno de los lenguajes de programación compatibles, como Visual Basic o Visual C#. (Maximiliano Firtman, Leonardo Natale, 2010)

**Directiva y Enlace.** - Estipulan las condiciones exigidas para comunicarse con un servicio. Por ejemplo, el enlace debe especificar (como mínimo) el transporte utilizado (por ejemplo, HTTP o TCP) y una codificación. Las directivas incluyen los requisitos de seguridad y otras condiciones que se deben cumplir para comunicarse con un servicio.

La capa del tiempo de ejecución del servicio contiene los comportamientos que solo se producen durante la operación actual del servicio, es decir, los comportamientos en tiempo de ejecución del servicio. La limitación de peticiones controla cuántos mensajes se procesan que puede variar si la demanda para el servicio crece a un límite preestablecido. Un comportamiento de error especifica lo que sucede cuando se produce un error interno en el servicio, por ejemplo, controlando qué información se comunica al cliente. (Demasiada información puede dar ventaja a un usuario malintencionado para organizar un ataque.) El comportamiento de los metadatos rige cómo y si los metadatos se ponen a disposición del mundo externo. El comportamiento de la instancia especifica cuántas instancias del servicio se pueden ejecutar (por ejemplo, un SINGLETON especifica solo una instancia para procesar todos los mensajes). El comportamiento de la transacción habilita la recuperación de operaciones de transacción si se produce un error. El comportamiento de la expedición es el control de cómo la infraestructura WCF procesa un mensaje. (Microsoft, Introducción a la extensibilidad, 2013)

La extensibilidad habilita la personalización de procesos en tiempo de ejecución. Por ejemplo, la inspección del mensaje es la facilidad para inspeccionar partes de un mensaje y la filtración de parámetros permite que se realicen acciones preestablecidas basándose en filtros que actúan en encabezados del mensaje. (Microsoft, Introducción a la extensibilidad, 2013)

### **2.3.7. Mensajería**

La capa de la mensajería se crea de canales. Un canal es un componente que procesa un mensaje de alguna manera, por ejemplo, autenticando un mensaje. Un conjunto de canales también se conoce como una pila de canales. Los canales funcionan en los mensajes y encabezados del mensaje. Esto es diferente de la capa en tiempo de ejecución del servicio, que se ocupa principalmente de procesar el contenido de los cuerpos de los mensajes.

Hay dos tipos de canales: (Microsoft, Arquitectura de WCF, 2013)

- a) Los canales de transporte leen y escriben mensajes de la red (o algún otro punto de la comunicación con el mundo externo). Algunos transportes utilizan un codificador para convertir los mensajes (que se representan como conjuntos de información XMLs) hacia y desde la representación de la secuencia de bytes utilizada por la red. Son ejemplos de transportes HTTP, canalizaciones con nombre, TCP y MSMQ. Son ejemplos de codificaciones XML y binario optimizado.
  
- b) Los canales de protocolo implementan protocolos de procesamiento de mensajes, a menudo leyendo o escribiendo encabezados adicionales en el mensaje. Los ejemplos de tales protocolos incluyen WS-Security y WS-Reliability.

La capa de la mensajería muestra los posibles formatos y modelos de intercambio de los datos. WS-Security es una implementación de la especificación WS-Security que habilita la seguridad en la capa del mensaje. El canal de mensajería WS-Reliable habilita la garantía de entrega del mensaje. Los codificadores presentan una variedad de codificaciones que se pueden utilizar para satisfacer las necesidades del mensaje. El canal HTTP especifica que el Protocolo de transporte de hipertexto se utiliza para la entrega del mensaje. El canal TCP especifica de manera similar el protocolo TCP. El canal de flujo de transacciones rige los modelos de mensajes de transacción. El canal de la canalización con nombre habilita la comunicación entre procesos. El canal de MSMQ habilita la interoperación con aplicaciones MSMQ.

#### **2.3.8. Alojamiento y Activación**

En su forma final, un servicio es un programa. Como otros programas, un servicio se debe ejecutar en un ejecutable. Esto se conoce como un servicio con host propio.

Los servicios también se pueden hospedar ejecutar en un ejecutable administrado por un agente externo, como IIS o Servicio de activación de Windows (WAS). WAS permite activar automáticamente aplicaciones WCF cuando se implementan en un equipo que ejecuta WAS. Los servicios también se pueden ejecutar manualmente como ejecutables (archivos .exe). Un servicio también se puede ejecutar automáticamente como un servicio de Windows. Los componentes COM+ también se pueden hospedar como servicios WCF. (Microsoft, Arquitectura de WCF, 2013)

## 2.4. Programación Básica de WCF

### 2.4.1. Ciclo de vida de programación básica

Describe el ciclo de vida del diseño, creación e implementación de aplicaciones de servicio y cliente de WCF. (Microsoft, Programación básica de WCF, 2013)

#### 2.4.1.1. Definir el contrato de servicio (Microsoft, Ciclo de vida, 2013)

Un contrato de servicio especifica la firma de un servicio, los datos que intercambia y el resto de los datos necesarios contractualmente. Para obtener la interoperabilidad más amplia posible de aplicaciones cliente, se recomienda que marque sus tipos con los atributos **DataContractAttribute** y **DataMemberAttribute**.

### Clases o interfaces

Tanto las clases como las interfaces representan una agrupación de funcionalidad y, por consiguiente, ambas se pueden utilizar para definir un contrato de servicios WCF. Sin embargo, se recomienda que utilice las interfaces porque modelan directamente los contratos de servicios. Sin una implementación, las interfaces no hacen más que definir una agrupación de métodos con ciertas firmas. Implemente una interfaz de contrato de servicio y habrá implementado un servicio WCF.

Todas las ventajas de las interfaces administradas se aplican a las interfaces de contrato de servicio:

- c) Las interfaces del contrato de servicio pueden extender cualquier número de otras interfaces del contrato de servicio.
- d) Una única clase puede implementar cualquier número de contratos de servicios implementando esas interfaces del contrato de servicio.

- e) Puede modificar la implementación de un contrato de servicios cambiando la implementación de la interfaz, mientras el contrato de servicios sigue siendo el mismo.
- f) Puede controlar la versión de su servicio implementando la interfaz antigua y la nueva. Los clientes antiguos se conectan a la versión original, mientras los clientes más nuevos pueden conectarse a la versión más nueva.

La manera preferida de crear un contrato de Windows Communication Foundation (WCF) es utilizar una interfaz. Este contrato especifica la colección y estructura de mensajes requeridas para obtener acceso a las operaciones que el servicio proporciona. Esta interfaz define los tipos de entrada y salida aplicando la clase **ServiceContractAttribute** a la interfaz y la clase **OperationContractAttribute** a los métodos que desee exponer.

```
1 using System.ServiceModel;
2
3 [ServiceContract]
4 public interface ICalculator
5 {
6     [OperationContract]
7     double Add(double n1, double n2);
8     [OperationContract]
9     double Subtract(double n1, double n2);
10    [OperationContract]
11    double Multiply(double n1, double n2);
12    [OperationContract]
13    double Divide(double n1, double n2);
14 }
```

#### 2.4.1.2. Implementar el contrato

Para implementar un contrato de servicio, cree una clase que implemente el contrato y especifique comportamientos personalizados que deba tener el tiempo de ejecución. (Microsoft, 2013).

#### **2.4.1.3. Configurar el servicio**

Especificando los extremos y el resto de la información de comportamiento. (Microsoft, 2013).

#### **2.4.1.4. Hospedar el servicio**

WCF proporciona un modelo de programación unificado para generar aplicaciones orientadas a servicios. Este modelo de programación permanece coherente y es independiente del entorno de tiempo de ejecución en el que se implementa el servicio. En la práctica, esto significa que el código de sus servicios tiene prácticamente el mismo aspecto independientemente de la opción de hospedaje.

Estas opciones de hospedaje pueden desde ejecutarse dentro de una aplicación de consola hasta entornos de servidor como un servicio de Windows que se ejecuta dentro de un proceso de trabajo administrado por Internet Information Server (IIS) o por Windows Process Activation Service (WAS). Los programadores eligen el entorno de hospedaje que satisface los requisitos de implementación del servicio. Estos requisitos pueden derivarse de la plataforma en la que se implementa la aplicación, del transporte en el que debe enviar y recibir los mensajes, o del tipo de proceso que recicla y otra administración de procesos requerida para garantizar una disponibilidad adecuada, o de alguna otra administración o requisitos de confiabilidad.

#### **2.4.2. Diseño e implementación de servicios**

Describe cómo diseñar e implementar un contrato de servicios, elegir un modelo de intercambio de mensajes, especificar un contrato de errores y otros aspectos básicos de los servicios. (Microsoft, Diseño e implementación de servicios, 2013)

### 2.4.3. Configuración de servicios

Describe cómo configurar un servicio de WCF para admitir los requisitos de contrato, personalizar el comportamiento en tiempo de ejecución local e indicar la dirección para publicar el servicio. (Microsoft, Configuración de servicios, 2013)

### 2.4.4. Servicios de Hospedaje

Describe los fundamentos de los servicios de hospedaje en una aplicación. (Microsoft, Servicios de hospedaje, 2013)

Entorno de hospedaje	Escenarios comunes	Ventajas y limitaciones clave
Aplicación administrada ("autohospedada")	<ul style="list-style-type: none"> <li>• Aplicaciones de consola utilizadas durante el desarrollo.</li> <li>• WinForm enriquecidos y aplicaciones de cliente de WPF que obtienen acceso a servicios.</li> </ul>	<ul style="list-style-type: none"> <li>• Flexible.</li> <li>• Fácil de implementar.</li> <li>• No es una solución empresarial de servicios.</li> </ul>
Windows Services (conocidos anteriormente como servicios NT)	<ul style="list-style-type: none"> <li>• Un servicio de WCF de ejecución prolongada hospedado fuera de IIS.</li> </ul>	<ul style="list-style-type: none"> <li>• Duración de proceso del servicio controlada por el sistema operativo, no activada por mensaje.</li> <li>• Admitida por todas las versiones de Windows.</li> <li>• Entorno seguro.</li> </ul>
IIS 5.1, IIS 6,0	<ul style="list-style-type: none"> <li>• Ejecución de un servicio de WCF en paralelo con contenido ASP.NET en Internet utilizando el protocolo HTTP.</li> </ul>	<ul style="list-style-type: none"> <li>• Reciclaje de procesos.</li> <li>• Cierre por inactividad.</li> <li>• Supervisión de estado de procesos.</li> <li>• Activación por mensajes.</li> <li>• Sólo HTTP.</li> </ul>
Servicio de activación de procesos de Windows (WAS)	<ul style="list-style-type: none"> <li>• Ejecución de un servicio de WCF sin instalar IIS en Internet utilizando varios protocolos de transporte.</li> </ul>	<ul style="list-style-type: none"> <li>• No se requiere IIS.</li> <li>• Reciclaje de procesos.</li> <li>• Cierre por inactividad.</li> <li>• Supervisión de estado de procesos.</li> <li>• Activación por mensajes.</li> <li>• Funciona con HTTP, TCP, canalizaciones con nombre y MSMQ.</li> </ul>
IIS 7.0	<ul style="list-style-type: none"> <li>• Ejecución de un servicio de WCF con contenido ASP.NET.</li> </ul>	<ul style="list-style-type: none"> <li>• Ventajas de WAS.</li> <li>• Se integra con contenido de IIS y ASP.NET.</li> </ul>



	<ul style="list-style-type: none"> <li>• Ejecución de un servicio de WCF en Internet utilizando varios protocolos de transporte.</li> </ul>	
--	---	--

Tabla 2 - 1 ventajas y escenarios clave opciones de hospedaje para WCF

(Microsoft, Servicios de hospedaje, 2013)

#### 2.4.5. Creación de Clientes (Microsoft, Creación de clientes, 2013)

En WCF, los servicios y clientes modelan los contratos utilizando los atributos, interfaces y métodos administrados. Para conectar con un servicio en una aplicación cliente, es necesario obtener la información del tipo de contrato de servicios.

Normalmente, esto se consigue mediante Herramienta de utilidad de metadatos de ServiceModel (Svcutil.exe), que descarga los metadatos del servicio, los convierte en un archivo de código fuente administrado con el lenguaje elegido, y crea un archivo de configuración de aplicación cliente que puede utilizarse para configurar el objeto de cliente WCF.

Por ejemplo, si se crea un objeto de cliente WCF para invocar un MyCalculatorService, y se sabe que los metadatos de ese servicio están publicados en `http://computerName/MyCalculatorService/Service.svc?wsdl`, el siguiente ejemplo de código muestra cómo utilizar Svcutil.exe para obtener un archivo `ClientCode.vb` que contenga el contrato de servicios en código administrado:

```
svcutil /language:vb /out:ClientCode.vb /config:app.config
http://computerName/MyCalculatorService/Service.svc?wsd
```

Puede compilar este código de contrato en la aplicación cliente, o en otro ensamblado que la aplicación cliente puede utilizar para crear un objeto de cliente WCF. Puede utilizar el archivo de configuración y configurar el objeto de cliente para conectarse correctamente con el servicio.

## 2.5. Acerca de la Institución: Benemérito Cuerpo de Bomberos de Otavalo

### 2.5.1. Breve Historia



Otavalo como una ciudad eminentemente turística y artesanal, afectada por el desarrollo constante ofertado por sus ciudadanos y las necesidades de la población que a la par de su crecimiento, inquietó a un grupo de visionarios jóvenes otavaleños, quienes siendo padres de familia, comerciantes, transportistas, etc., veían cada vez más de cerca, los riesgos ante los que estaban expuestos los habitantes del cantón, por lo que a mediados de la década de los 60 mediante concesión entre los pobladores y las diferentes coyunturas políticas de la época; solicitan mediante oficio la creación del primer cuerpo de bomberos en la ciudad de Otavalo.

Posteriormente, con las personas encargadas de dirigir la Institución y con la ayuda de las autoridades cantonales se logra la donación de un terreno que pertenecía a la Sub Estación Eléctrica Municipal, ubicada en la calle Quito entre Sucre y Bolívar, donde se construyó el primer edificio. Inmediatamente se legalizan los nombramientos de los funcionarios y en el registro oficial No 2593 del 1 de octubre de 1969, se crea el Cuerpo de Bomberos de Otavalo, a través del entonces Ministerio de Previsión Social y Trabajo, ratificándose como primer jefe al señor Marco Tulio Teanga Saa; y como secretario pagador al señor Nelson Aurelio Orbe Mena.

Desde entonces, el trabajo cotidiano del Cuerpo de Bomberos de Otavalo ha estado encaminado al servicio oportuno y emergente a la ciudadanía, liderados a través de los años por los señores comandantes.



Figura 2- 2 Cuerpo de Bomberos de Otavalo (*Cuerpo de bomberos de Otavalo, 2013*)

<http://www.bomberosotavalo.gob.ec/images/stories/historia1.jpg>

### **2.5.2. Misión**

“La misión de los Cuerpos de Bomberos es eminentemente técnica y humana, para salvar vidas a través de la atención efectiva de Emergencias y Gestión de Riesgos, sembrando una cultura de prevención y alerta temprana para seguridad y bienestar de la sociedad, con personal altamente calificado, eficiente y eficaz; con una infraestructura y equipos adecuados para brindar a la ciudadanía un servicio oportuno y de calidad.” (Cuerpo de bomberos de Otavalo, 2013)

### **2.5.3. Visión**

“Ser reconocidos como un organismo líder en Atención de Emergencias y Gestión de Riesgos a nivel local, regional y nacional, por su servicio efectivo con un alto índice de calidad, vinculando a la comunidad, en un ambiente profesional y ético con abnegación y disciplina, acorde al avance tecnológico, contribuyendo al desarrollo sustentable y pertinente de la sociedad en términos generacionales, de género y con identidad cultural.” (Cuerpo de bomberos de Otavalo, 2013)

## 2.5.4. Organigrama



Figura 2- 3 Organigrama Institucional (*Cuerpo de bomberos de Otavalo, 2013*)  
<http://www.bomberosotavalo.gob.ec/images/stories/organigrama.png>

## 2.5.5. Servicios a la comunidad

### 2.5.5.1. Proteger a la Comunidad

El objetivo primordial de la Institución es proteger la vida y bienes de los ciudadanos, con el compromiso de servicio eficiente, eficaz, profesional, humanitario que brinda el cuerpo de bomberos de Otavalo con los servicios enfocados hacia la comunidad. (Cuerpo de bomberos de Otavalo, 2013)

### 2.5.5.2. Capacitaciones a estudiantes

Se dan capacitaciones mediante brigadas estudiantiles a los colegios de la zona en lo que es:

- a) Primeros Auxilios
- b) Instrucción formal.
- c) Manejo de gas licuado de petróleo.
- d) Prevención en hogares.

- e) Evacuación en caso de terremotos.

### **2.5.5.3. Capacitaciones Internas**

Se capacita al personal con especialidades en algunos cursos:

- a) BAGER (Bases Administrativas para la Gestión de Riesgos)
- b) COPCIF (Curso de Operaciones de Prevención y Control de Incendios Forestales)
- c) SCI (Sistema de Comando de Incidentes) Intermedio.
- d) Cierres contables y apertura contable nuevo año.

### **2.5.5.4. Control y Emisión de Permisos de Funcionamiento**

Con respecto al proceso de gestión, generación y registro de permisos de funcionamiento no existe a la mano un documento formal que detalle dicho proceso, solamente la descripción verbal de las personas que trabajan en la institución y están relacionadas con las actividades que tienen que ver con el proceso de emisión de este documento hacia los propietarios de los locales comerciales. Este proceso se detalla de manera general de la siguiente manera:

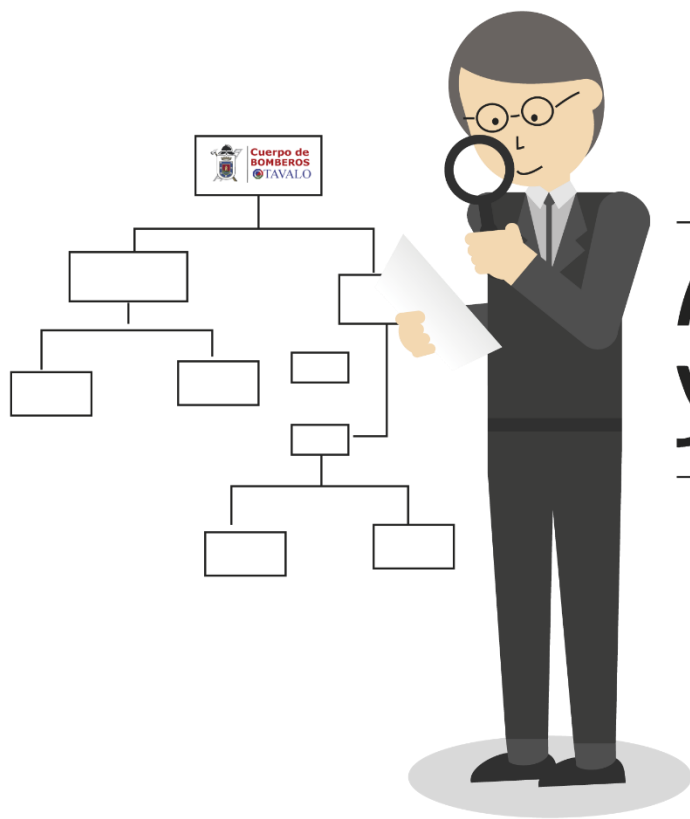
El proceso de permisos de funcionamiento tiene lugar debido a la necesidad de llevar a cabo un control a los negocios que generan productos y servicios para los pobladores o gente extranjera con el fin de evitar desgracias debido a los accidentes que pueden ser producidos por descuidos o por falta de información, en este sentido el cuerpo de Bomberos brinda capacitaciones a los propietarios de los locales además de las visitas regulares de los miembros de la institución en calidad de inspectores para realizar inspecciones donde se determinan parámetros de riesgo acorde a las actividades que lleve a cabo el local comercial o negocio.

Las Inspecciones se llevan a cabo previa solicitud del propietario del Local Comercial del año en curso y son realizadas por miembros capacitados del cuerpo de bomberos que se ubican en el lugar donde opera el negocio, allí se revisan aspectos como la medición de riesgos que pueden presentarse por el uso ya sea de materiales inflamables, seguridad humana, riesgos de incendios, protección preventiva, año de la construcción, estos son parámetros que le ayudan al inspector a valorar el estado del local comercial, toda esta información recolectada por el inspector se registra en formularios físicos que posteriormente es usada para considerar si se genera o no el permiso de funcionamiento.

Para la generación del permiso de funcionamiento la secretaria solamente debe emitir el documento si es solicitado por el cliente dueño del local comercial y que conste en el informe físico del inspector que el local comercial cumple con todas las normas de seguridad que lo permitan laborar de forma segura.

La secretaria luego de verificar que no existen observaciones negativas en el informe, procede a emitir el permiso de funcionamiento para el local comercial, anexando el informe físico de inspección con una copia del permiso de funcionamiento emitido.

Como se ha descrito de manera general el proceso de generación de permisos de funcionamiento se lo ha llevado de manera manual y los detalles de las necesidades del sistema que gestionara este proceso se plasman en los requerimientos funcionales descritos más adelante.



---

# ANALISIS Y DISEÑO

---

## CAPITULO III

### 3. ANÁLISIS Y DISEÑO

Para el proceso de desarrollo del proyecto se va a hacer uso de la metodología RUP en las diferentes etapas de este desarrollo, esta metodología incluye prácticas claves, planeación estratégica y administración de riesgos en cada fase.

#### 3.1. Requisitos

##### 3.1.1. Requerimientos Funcionales

Código	Descripción del Requerimiento	Caso de Uso	Actor
RF1	El sistema debe permitir registrar un catálogo de los tipos de locales comerciales.	Registrar Tipos de Local comercial (Crear, Modificar)	Secretaria Supervisor
RF2	El sistema debe permitir registrar un catálogo de las tarifas por cada tipo de local comercial, el mismo que tendrá vigencia de un año.	Registrar Tarifas por Tipo Local Comercial (Crear, Modificar)	Secretaria Supervisor
RF3	El sistema debe permitir al usuario registrar información relacionada al Cliente.	Registrar Clientes (Crear, Modificar)	Secretaria Supervisor
RF4	El sistema debe permitir ingresar a las personas que son empleados de la institución al sistema y asignarles un rol de acuerdo a la función que desempeñan: empleado, secretaria, tesorero, inspector.	Registrar Empleados (Crear, Modificar)	Secretaria Supervisor



RF5	El sistema debe permitir al usuario registrar información relacionada al local comercial al que pertenece el cliente.	Registrar Local Comercial (Crear, Modificar)	Secretaria Supervisor
RF6	El sistema debe permitir generar permisos de funcionamiento anual al propietario del local comercial.	Generar Permiso de Funcionamiento (Crear)	Secretaria Supervisor
RF7	El sistema debe permitirle al cliente consultar el estado del permiso de funcionamiento de su local comercial.	Consultar Permiso de Funcionamiento	Secretaria Supervisor
RF8	El sistema debe permitirle al usuario con el rol de inspector registrar información de la inspección física del local comercial en el sistema.	Registrar Informe de Inspección (Crear, Modificar)	Inspector Supervisor
RF9	El sistema debe permitirles consultar reportes acerca del estado de la recaudación de los permisos de funcionamiento anual / mensual	Consultar Reporte de Permisos de Funcionamiento Anual / Mensual	Secretaria Tesorero
RF10	El sistema debe permitir el acceso desde internet a las funciones principales del sistema al empleado encargado de cada módulo.	Iniciar Sesión	Usuario
RF11	El sistema debe permitir llevar un registro de asistencia para los empleados y miembros de la institución	Control de Asistencia (Crear)	Secretaria
RF12	El sistema debe permitir generar reportes mensuales, anuales, por persona acerca del registro	Consultar Control de Asistencia	Usuario

	de asistencia de los empleados y miembros de la institución		
RF13	El sistema debe permitir generar información consolidada acerca de la recaudación de permisos de funcionamiento de manera mensual, anual, para que sea consumida por un sistema externo contable para su respectivo procesamiento.	Generar Conciliación Permisos y Rubros	Tesorero

Tabla 3 - 1 Requerimientos Funcionales (Fuente: Propia., 2017)

### 3.1.2. Requerimientos No Funcionales

Código	Descripción del Requerimiento
RNF1	Usabilidad: El sistema debe proveer interfaces que faciliten la interacción entre los usuarios y el sistema, además si se presentan mensajes de error se debe permitir facilitar al usuario identificarlos y reportarlos al administrador del sistema
RNF2	<i>Disponibilidad:</i> Las funcionalidades de los servicios del sistema para el usuario estarán expuestas diariamente hasta las 5:30 de la tarde, excepto los servicios de consulta y solicitudes expuestas hacia el cliente los cuales estarán disponibles las 24h, cuando un servicio se interrumpe se levanta otro en su lugar.
RNF3	<i>Confiabilidad:</i> Los servicios deben ser tolerantes a fallos y las operaciones a realizar,
RNF4	<i>Seguridad:</i> El acceso al sistema debe ser restringido al uso de claves de claves asignadas a cada uno de los usuarios y

	dependiendo de su rol tendrá acceso a un módulo o módulos del sistema.
RNF5	<i>Mantenimiento y Actualización:</i> El sistema deberá estar documentado, cada uno de los módulos que forman parte de la solución propuesta deberán estar debidamente documentados
RNF6	<i>Capacidad de administración:</i> Ya que el sistema está basado en servicios se necesita un fácil acceso a la información acerca de lo que sucede en el host en el que viven los servicios WCF.

*Tabla 3 - 2 Requerimientos no Funcionales. (Fuente: Propia., 2017)*

### 3.2. Casos de Uso

#### 3.2.1. Vista Global del Sistema con sus Actores y Casos de Uso

A continuación, se cran los casos de uso para definir, identificar y dividir la funcionalidad del sistema, mediante esquemas o modelos que representen la vista del sistema del mundo real de los requerimientos funcionales descritos anteriormente.

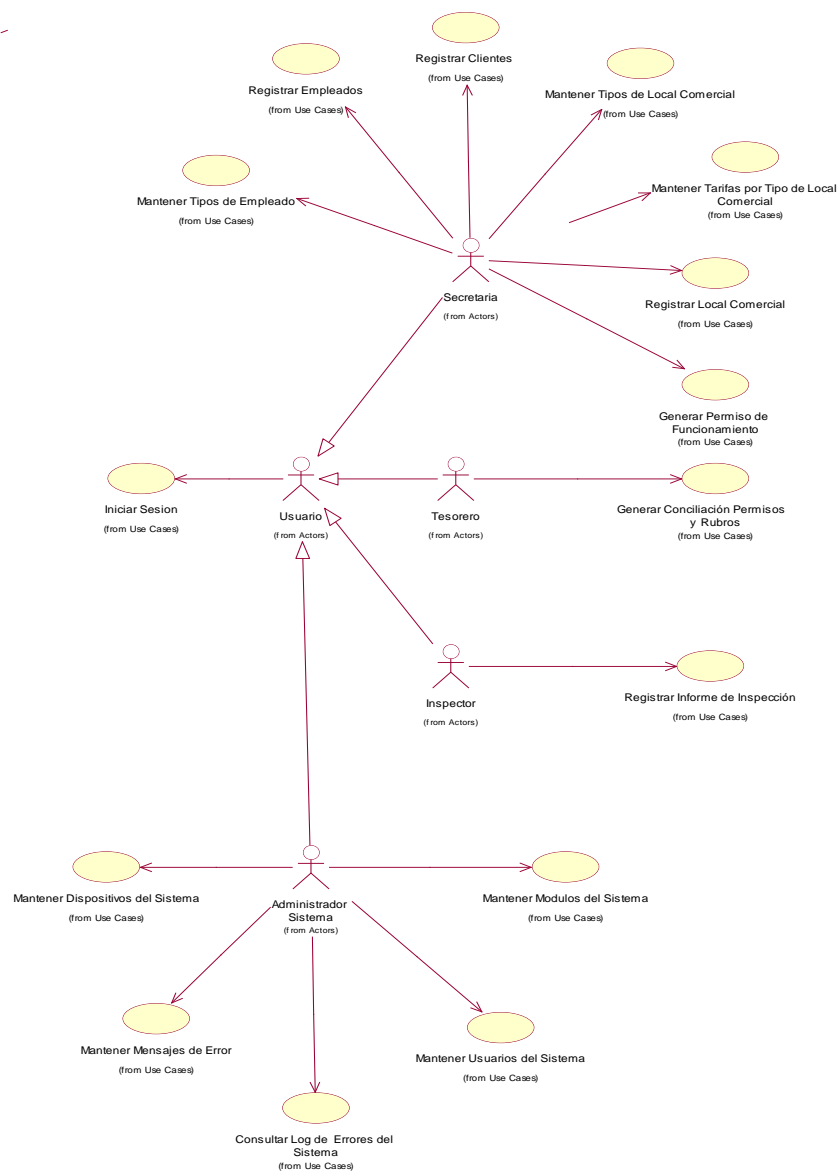


Figura 3- 1 Vista Global del Sistema: Actores y Casos de Uso. (Fuente: Propia., 2017)

### 3.2.2. Módulo Administrativo

#### 3.2.2.1. Caso de Uso: Mantener Módulos del Sistema.

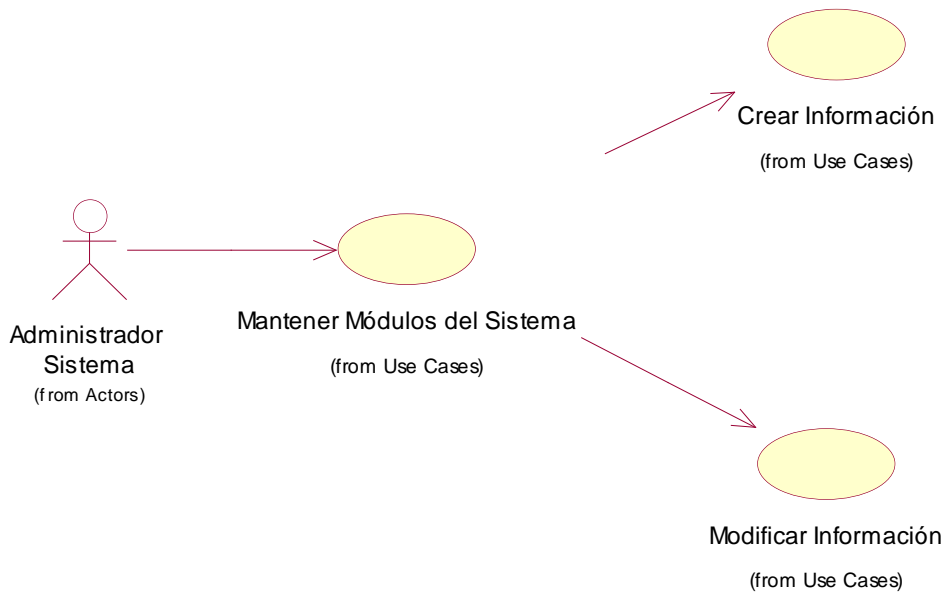


Figura 3- 2 Caso de Uso: Mantener Módulos del Sistema (Fuente: Propia., 2017)

#### Caso de Uso: Mantener Módulos del Sistema

**Actor:** Administrador del Sistema.

**Precondiciones:**

El usuario ha sido admitido con el rol de Administrador del Sistema (Inicio de sesión aceptada por el sistema).

**Post condiciones:**

El módulo del sistema será dado de alta en el sistema con la información completa en sus dos flujos básicos:

- **Flujo Básico: Crear Modulo del Sistema**

g) El caso de uso inicia cuando el usuario indica “Crear Modulo del Sistema”.

- h) El sistema muestra un formulario de ingreso donde se va a ingresar información como la descripción, dependencia, status.
- i) El usuario indica “Guardar”.
- j) El sistema realiza las validaciones generales de los datos ingresados.
- k) El caso de uso finaliza.

- **Flujo Alternativo:**

- Validaciones generales:**

- En el paso d), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

- **Flujo Básico: Modificar Modulo del Sistema**

- a) El caso de uso inicia cuando el usuario indica “Modificar Modulo del Sistema”.
- b) El sistema muestra un formulario con una lista de módulos disponibles en el sistema.
- c) El Usuario selecciona el módulo que va a modificar.
- d) El sistema muestra un formulario con la información del módulo seleccionado donde se puede modificar información como la descripción, dependencia y status
- e) El usuario indica “Guardar”
- f) El sistema realiza las validaciones generales de los datos ingresados.
- g) El caso de uso finaliza

- **Flujo Alternativo:**

**Validaciones generales:**

En el paso f), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

### 3.2.2.2. Caso de Uso: Mantener Dispositivos en el Sistema

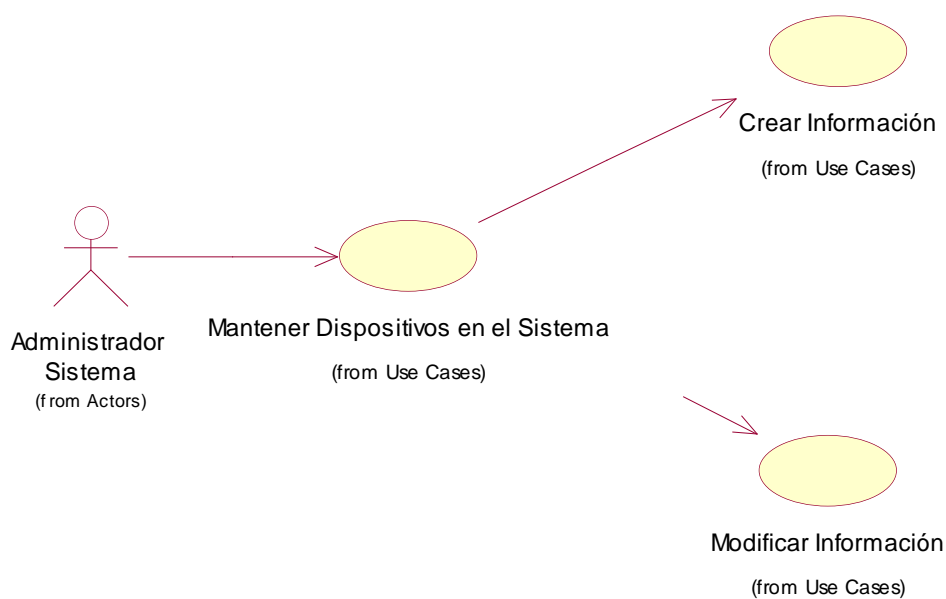


Figura 3- 3 Caso de Uso: Mantener Dispositivos en el Sistema (Fuente: Propia., 2017)

### Caso de Uso: Mantener Dispositivos en el Sistema

**Actor:** Administrador del Sistema

**Precondiciones:**

El usuario ha sido admitido con el rol de Administrador del Sistema (Inicio de sesión aceptada por el sistema)

## **Post condiciones:**

El dispositivo del sistema será dado de alta en el sistema con la información completa en sus dos flujos básicos:

- **Flujo Básico: Crear Dispositivo en el Sistema**

- a) El caso de uso inicia cuando el usuario indica “Crear Dispositivo en el Sistema”.
- b) El sistema muestra un formulario con una lista de módulos disponibles en el sistema.
- c) El Usuario selecciona el módulo que va a usar para registrar el dispositivo.
- d) El sistema muestra un formulario de ingreso donde se va a ingresar información como el código, descripción, tipo, status, dirección ip.
- e) El usuario indica “Guardar”
- f) El sistema realiza las validaciones generales de los datos ingresados.
- g) El caso de uso finaliza

- **Flujo Alternativo:**

### **Validaciones generales:**

En el paso f), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

- **Flujo Básico: Modificar Modulo del Sistema**

- a) El caso de uso inicia cuando el usuario indica “Modificar Dispositivo del Sistema”.



- b) El sistema muestra un formulario con una lista de los dispositivos disponibles en el sistema.
- c) El Usuario selecciona el dispositivo que va a modificar.
- d) El sistema muestra un formulario con la información del dispositivo seleccionado donde se puede modificar la información: descripción, tipo, status, dirección ip.
- e) El usuario indica "Guardar"
- f) El sistema realiza las validaciones generales de los datos ingresados.
- g) El caso de uso finaliza

- **Flujo Alternativo:**

**Validaciones generales:**

En el paso f), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

### 3.2.2.3. Caso de Uso: Mantener Catálogo de Errores del Sistema

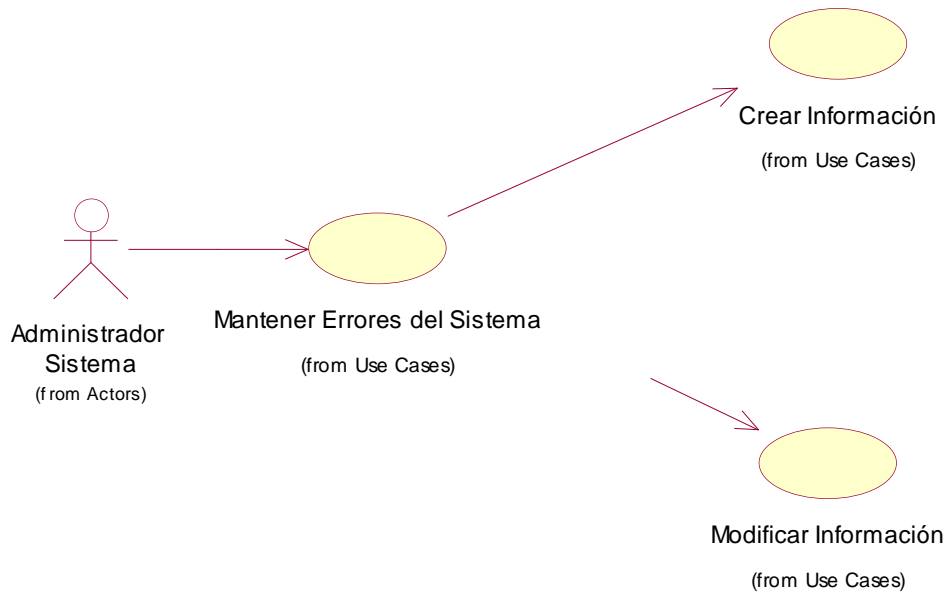


Figura 3- 4 Caso de Uso: Mantener Catálogo de Errores del Sistema (*Fuente: Propia., 2017*)

### Caso de Uso: Mantener Catálogo de Errores del Sistema

**Actor:** Administrador del Sistema

#### Precondiciones:

El Usuario del sistema ha sido admitido con el rol de Administrador del Sistema (Inicio de sesión aceptada por el sistema)

#### Post condiciones:

El error del sistema será dado de alta en el sistema con la información completa en sus dos flujos básicos:

- **Flujo Básico: Crear Modulo del Sistema**

a) El caso de uso inicia cuando el usuario indica "Crear Error del Sistema".

- b) El sistema muestra un formulario de ingreso donde se va a ingresar información como el tipo, descripción y el mensaje de error.
  - c) El usuario indica “Guardar”
  - d) El sistema realiza las validaciones generales de los datos ingresados.
  - e) El caso de uso finaliza
- **Flujo Alternativo:**

### **Validaciones generales:**

En el paso d), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

- **Flujo Básico: Modificar Modulo del Sistema**
- a) El caso de uso inicia cuando el usuario indica “Modificar Error del Sistema”.
  - b) El sistema muestra un formulario con una lista de errores disponibles en el sistema, donde el usuario ingresa el id del error para filtrar el registro que se necesita modificar.
  - c) El Usuario selecciona el error que va a modificar.
  - d) El sistema muestra un formulario con la información del error seleccionado donde se puede modificar información como el tipo, descripción y el mensaje de error
  - e) El usuario indica “Guardar”
  - f) El sistema realiza las validaciones generales de los datos ingresados.
  - g) El caso de uso finaliza

### Flujo Alternativo:

### Validaciones generales:

En el paso f), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

#### 3.2.2.4. Caso de Uso: Mantener Usuarios del Sistema

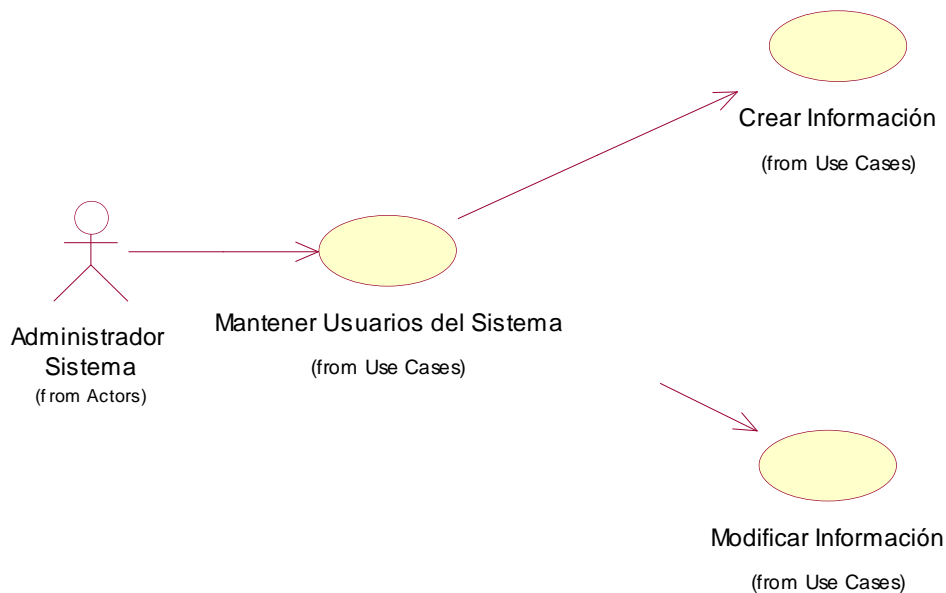


Figura 3- 5 Caso de Uso: Mantener Usuarios del Sistema (Fuente: Propia., 2017)

### Caso de Uso: Mantener Usuarios del Sistema

**Actor:** Administrador del Sistema

**Precondiciones:**

El Usuario del sistema ha sido admitido con el rol de Administrador del Sistema  
(Inicio de sesión aceptada por el sistema)

**Post condiciones:**

El Usuario del sistema será dado de alta en el sistema con la información completa en sus dos flujos básicos:

- **Flujo Básico: Crear Usuario del Sistema**
  - a) El caso de uso inicia cuando el usuario indica “Crear Error del Sistema”.
  - b) El sistema muestra un formulario con una lista de empleados disponibles en el sistema, donde el usuario ingresa el id del empleado para filtrar el registro que se necesita.
  - c) El Usuario selecciona al empleado que va a usar para registrarlo como usuario del sistema.
  - d) El sistema muestra un formulario de ingreso donde se va a ingresar información como el password, caducidad password y el módulo del sistema al que tendrá acceso.
  - e) El usuario indica “Guardar”
  - f) El sistema realiza las validaciones generales de los datos ingresados.
  - g) El caso de uso finaliza
- **Flujo Alternativo:** No aplica.

### **Validaciones generales:**

En el paso f), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

- **Flujo Básico: Modificar Usuario del Sistema**

- a) El caso de uso inicia cuando el usuario indica “Modificar Usuario del Sistema”.
- b) El sistema muestra un formulario con una lista de los usuarios disponibles en el sistema, donde el usuario ingresa el id del usuario para filtrar el registro que se necesita modificar.
- c) El Usuario selecciona el registro que va a modificar.
- h) El sistema muestra un formulario con la información del usuario seleccionado donde se puede modificar información como: password, caducidad password y el módulo del sistema al que tendrá acceso.
- d) El usuario indica “Guardar”
- e) El sistema realiza las validaciones generales de los datos ingresados.
- f) El caso de uso finaliza

### **Flujo Alternativo:**

#### **Validaciones generales:**

En el paso e), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

### 3.2.2.5. Caso de Uso: Iniciar Sesión

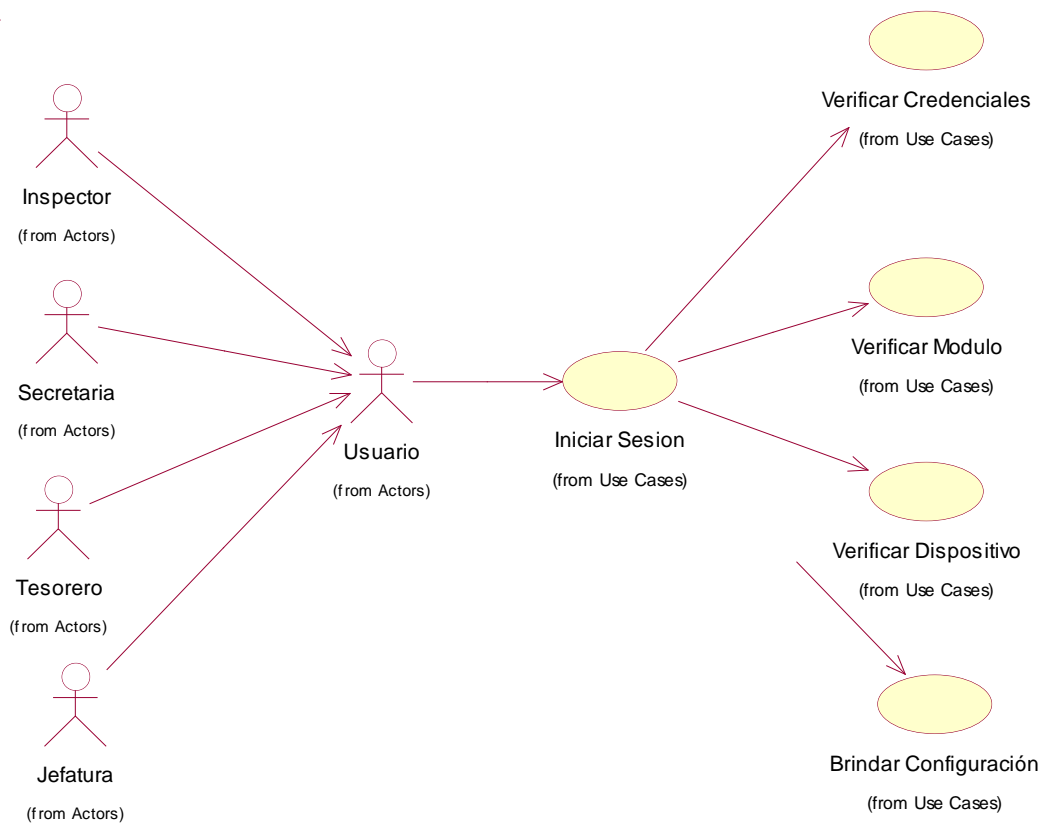


Figura 3- 6 Caso de Uso: Iniciar Sesión (Fuente: Propia., 2017)

#### Caso de Uso: Iniciar Sesión

**Actor:** Usuario

**Precondiciones:**

El usuario debe estar previamente registrado en el sistema

**Post condiciones:**

El Usuario obtendrá inicio de sesión conjuntamente con la configuración del módulo del sistema al que corresponde.

**Flujo Básico:**

- a) El caso de uso inicia cuando el usuario indica "Iniciar Sesión".

- b) El sistema muestra un formulario de ingreso donde se va a ingresar información como el nombre del usuario y contraseña.
- c) El usuario indica “Iniciar Sesión”
- d) El sistema verifica que el usuario y su contraseña coincida con la información que esté registrada en el sistema
- e) El sistema verifica información acerca del módulo al que está registrado el usuario.
- f) El sistema verifica adicionalmente el/los dispositivos(s) permitido(s) al usuario de acuerdo al módulo asignado.
- g) El sistema otorga la sesión correspondiente al usuario, incluyendo el acceso a la configuración del módulo asignado.
- h) El caso de uso finaliza

#### **Flujo Alternativo:**

##### **Usuario/Contraseña incorrectos:**

En el paso d), si se genera un error en la verificación del usuario/contraseña se debe presentar un mensaje de error: “La información de Usuario/Contraseña son incorrectos”, y se termina el caso de uso.

##### **Modulo no asignado:**

En el paso e), si se genera un error en la verificación de la asignación del módulo al usuario se debe presentar un mensaje de error: “La información del módulo para este usuario no existe.”, y se termina el caso de uso.

##### **Dispositivo inexistente o deshabilitado:**

En el paso f), si se genera un error en la verificación del dispositivo desde el que se está accediendo al sistema, al usuario se debe presentar un mensaje de error: “El



dispositivo no está permitido para acceder a este módulo: ID MODULO”, y se termina el caso de uso.

Configuración Inexistente:

En el paso g), si se genera un error en la obtención de la configuración para el módulo registrado para el usuario o no existiese esta configuración, al usuario se debe presentar un mensaje de error: “No existe la parametrización para el módulo: ID MODULO.”, y se termina el caso de uso.

Configuración faltante:

En el paso g), si se genera un error en la obtención de la configuración por falta de parametrización de algún campo para el módulo registrado, al usuario se debe presentar un mensaje de error: “Existe inconsistencias en la parametrización de la configuración el módulo: ID MODULO, en el campo: ID CAMPO”, y se termina el caso de uso.

### 3.2.3. Caso de Uso: Mantener Clientes

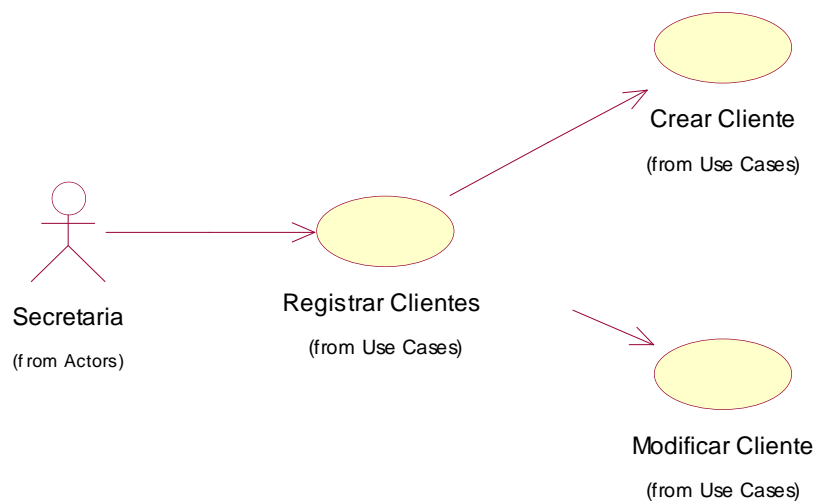


Figura 3- 7 Caso de Uso: Mantener Clientes (Fuente: Propia., 2017)

## **Caso de Uso: Mantener Clientes**

**Actor:** Secretaría

### **Precondiciones:**

El usuario ha sido admitido con el rol de secretaria (Inicio de sesión aceptada por el sistema)

### **Post condiciones:**

El usuario será dado de alta en el sistema con la información completa en sus dos flujos básicos

### **Flujo Básico: Crear Cliente**

- a) El caso de uso inicia cuando la secretaria indica "Registrar Clientes".
- b) El sistema muestra un formulario de ingreso donde se va a ingresar información básica del cliente como es no cédula o ruc, nombres, apellidos, dirección, teléfonos.
- c) La secretaria indica "Guardar"
- d) El sistema realiza validaciones generales a los datos ingresados y registra la información.
- e) El caso de uso finaliza

### **Flujo Alternativo:**

#### **Validaciones generales:**

En el paso d), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

### **Cliente Existente:**

En el paso d), si se genera un error por cliente existente se muestra el mensaje: "Cliente: No. Cédula, ya se encuentra creado en el sistema".

### **Flujo Básico: Modificar Cliente**

- a) El caso de uso inicia cuando la secretaria indica "Modificar Clientes".
- b) El sistema muestra un formulario de ingreso donde se va a ingresar la cédula del cliente para realizar su búsqueda y listarlo en el mismo formulario
- c) La secretaria selecciona al Cliente.
- d) El Sistema muestra un formulario con los datos del cliente modificables: nombres, apellidos, dirección, teléfonos.
- e) La secretaria luego de realizar las modificaciones indica "Guardar"
- f) El sistema realiza validaciones generales a los datos ingresados y registra la información.
- g) El caso de uso finaliza

### **Flujo Alternativo:**

#### **Validaciones generales:**

En el paso f), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

### **Cliente Inexistente:**

En el paso b), si se genera un error por cliente inexistente se muestra el mensaje: "Cliente: No. Cédula, no se encuentra registrado en el sistema".

### 3.2.4. Caso de Uso: Mantener Empleados

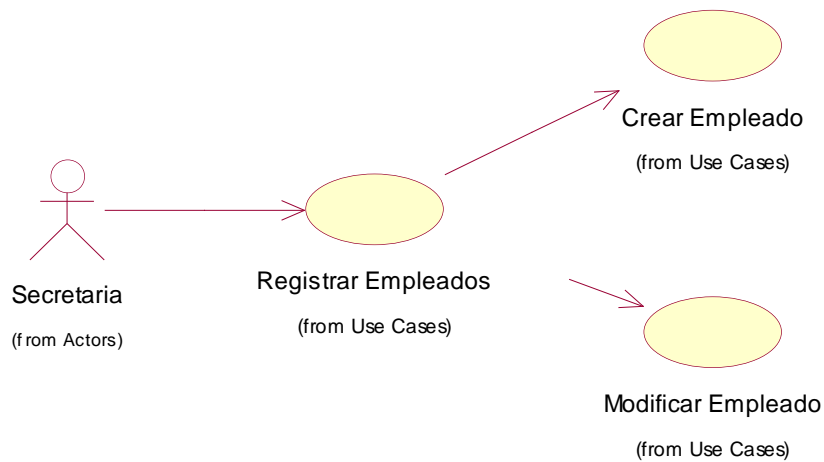


Figura 3- 8 Caso de Uso: Mantener Empleados (Fuente: Propia., 2017)

#### Caso de Uso: Registrar Empleado

**Actor:** Secretaria

#### Precondiciones:

- Se requiere que el dispositivo lector de huella digital esté conectado al sistema.
- El usuario ha sido admitido con el rol de Secretaria (Inicio de sesión aceptada por el sistema)

#### Flujo Básico: Crear

- a) El caso de uso inicia cuando la secretaria indica “Registrar Empleado”.
- b) El sistema muestra un formulario donde se ingresa la información del empleado como: CEDULA, NOMBRES, APELLIDOS, SEXO, DIRECCIÓN, TELÉFONO, FECHA DE INICIO CONTRATO, FECHA FINALIZACIÓN CONTRATO, HUELLA DIGITAL, TIPO EMPLEADO

- c) La secretaria en el mismo formulario indica “Cargar Archivo Huella Digital Empleado”, donde se muestra un formulario con la opción Huella Digital
- d) El sistema muestra en el mismo formulario las tarifas disponibles de acuerdo a la capacidad del tipo del local comercial seleccionado: (P) pequeño, (M) mediano, (Grande) grande.
- e) La secretaria selecciona la tarifa para el local comercial
- f) El sistema marca el tipo de local comercial y la tarifa seleccionadas
- g) La secretaria adicionalmente ingresa información básica del local comercial como la ubicación, teléfonos, capacidad y año de inicio de actividades.
- h) La secretaria indica “Guardar”
- i) El sistema realiza validaciones generales a los datos ingresados y registra la información.
- j) El caso de uso finaliza

### **Flujo Básico: Modificar**

- a) El caso de uso inicia cuando la secretaria indica “Registrar Empleado”.
- b) El sistema muestra un formulario de consulta en el cual se debe ingresar el id del cliente al cual se debe asignar el nuevo local comercial.
- c) La secretaria selecciona el cliente resultante del proceso de búsqueda en el formulario.
- d) El sistema muestra un formulario donde se selecciona el tipo de local comercial el cual va a ser asignado al registro.
- e) La secretaria selecciona el tipo de local comercial.
- f) El sistema muestra en el mismo formulario las tarifas disponibles de acuerdo a la capacidad del tipo del local comercial seleccionado: (P) pequeño, (M) mediano, (Grande) grande.

- g) La secretaria selecciona la tarifa para el local comercial
- h) El sistema marca el tipo de local comercial y la tarifa seleccionadas
- i) La secretaria adicionalmente ingresa información básica del local comercial como la ubicación, teléfonos, capacidad y año de inicio de actividades.
- j) La secretaria indica “Guardar”
- k) El sistema realiza validaciones generales a los datos ingresados y registra la información.
- l) El caso de uso finaliza

**Flujo Alternativo:**

**Cliente inexistente:**

En el paso b), si en el caso de que el cliente no exista se presentara un mensaje alertando “Usuario No registrado en el Sistema” y se termina el caso de uso.

**Validaciones generales:**

En el paso k), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo.

### 3.2.5. Caso de Uso: Mantener Locales comerciales

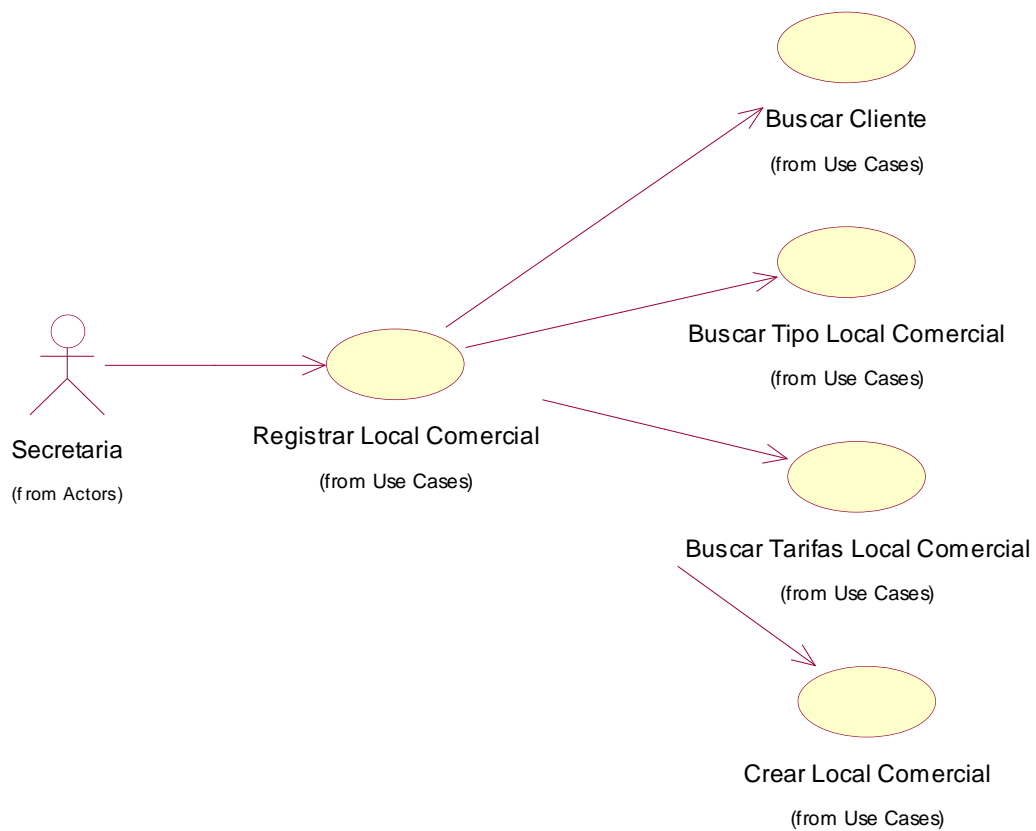


Figura 3- 9 Caso de Uso: Mantener Locales Comerciales (*Fuente: Propia., 2017*)

#### **Caso de Uso: Registrar Local Comercial**

**Actor:** Secretaria

**Precondiciones:**

- Se requiere que el cliente haya sido ingresado previamente al sistema
- El usuario ha sido admitido con el rol de Secretaria (Inicio de sesión aceptada por el sistema)

### **Flujo Básico:**

- a) El caso de uso inicia cuando la secretaria indica “Registrar Local Comercial”.
- b) El sistema muestra un formulario de consulta en el cual se debe ingresar el id del cliente al cual se debe asignar el nuevo local comercial.
- c) La secretaria selecciona el cliente resultante del proceso de búsqueda en el formulario.
- d) El sistema muestra un formulario donde se selecciona el tipo de local comercial el cual va a ser asignado al registro.
- e) La secretaria selecciona el tipo de local comercial.
- f) El sistema muestra en el mismo formulario las tarifas disponibles de acuerdo a la capacidad del tipo del local comercial seleccionado: (P) pequeño, (M) mediano, (Grande) grande.
- g) La secretaria selecciona la tarifa para el local comercial
- h) El sistema marca el tipo de local comercial y la tarifa seleccionadas
- i) La secretaria adicionalmente ingresa información básica del local comercial como la ubicación, teléfonos, capacidad y año de inicio de actividades.
- j) La secretaria indica “Guardar”
- k) El sistema realiza validaciones generales a los datos ingresados y registra la información.
- l) El caso de uso finaliza



### Flujo Alternativo:

### Cliente inexistente:

En el paso b), si en el caso de que el cliente no exista se presentara un mensaje alertando “Usuario No registrado en el Sistema” y se termina el caso de uso.

### Validaciones generales:

En el paso k), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo.

### 3.2.6. Caso de Uso: Registrar Informe de Inspección

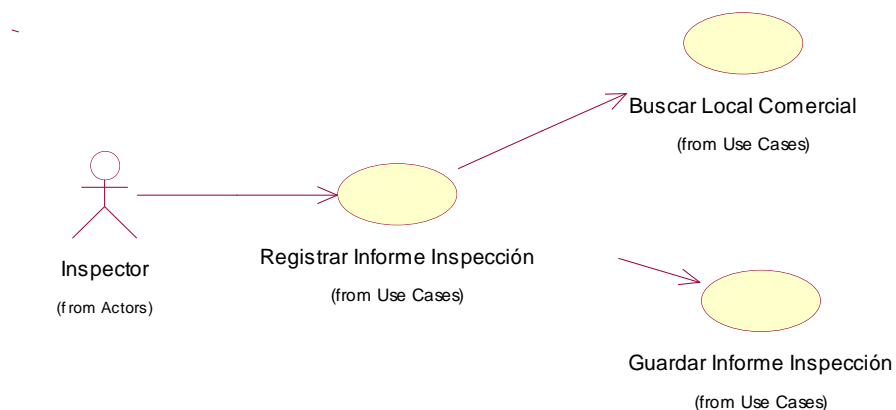


Figura 3- 10 Caso de Uso: Registrar Informe de Inspección (Fuente: Propia., 2017)

### Caso de Uso: Registrar Informe de Inspección

**Actor:** Inspector

### Precondiciones:

- Se requiere que el cliente haya sido ingresado previamente al sistema
- Se requiere que el local comercial haya sido ingresado previamente al sistema.

- Se requiere que el empleado este registrado en el sistema como usuario del sistema y con el rol de Inspector.
- El usuario ha sido admitido con el rol de Inspector (Inicio de sesión aceptada por el sistema)

**Flujo Básico:**

- a) El caso de uso inicia cuando la secretaria indica “Registrar Informe de Inspección”
- b) El sistema muestra un formulario de consulta donde se ingresa el Id del local comercial al cual se desea ingresar la información.
- c) El inspector selecciona el local comercial resultante del proceso de búsqueda en el formulario.
- d) El sistema muestra un formulario de ingreso en los cuales se ingresan los campos: año correspondiente a la inspección, fecha de la inspección, seguridad humana, material inflamable, riesgo de incendios, protección contra incendios, observaciones inspección y resultado de la inspección.
- e) El Inspector indica “Guardar”
- f) El sistema realiza validaciones generales a los datos ingresados y registra la información.
- g) El caso de uso finaliza.

### Flujo Alternativo:

### Informe no favorable:

En el paso b), si el sistema detecta que no existe el local comercial al cual se le va a registrar en informe muestra un mensaje indicando: “El local comercial al que hace referencia no se encuentra registrado.” y se termina el caso de uso.

### Validaciones generales:

En el paso f), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo.

### 3.2.7. Caso de Uso: Generar Permiso de Funcionamiento

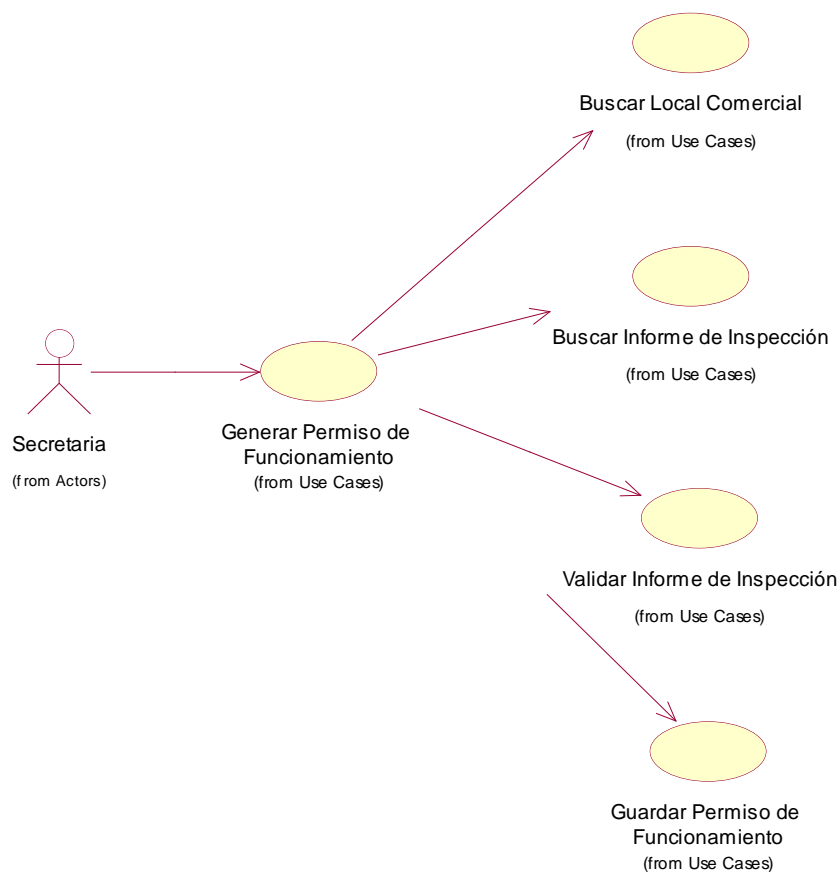


Figura 3- 11 Caso de Uso: Generar Permiso de Funcionamiento (Fuente: Propia., 2017)

## **Caso de Uso:** Generar Permiso de Funcionamiento

**Actor:** Secretaría

### **Precondiciones:**

- Se requiere que el cliente haya sido ingresado previamente al sistema.
- Se requiere que el local comercial haya sido ingresado previamente al sistema.
- Se requiere de la existencia del informe favorable de la inspección física del local comercial
- El usuario ha sido admitido con el rol de Secretaria (Inicio de sesión aceptada por el sistema)

### **Flujo Básico:**

- a) El caso de uso inicia cuando la secretaria indica “Generar Permiso de Funcionamiento”
- b) El sistema muestra un formulario donde se ingresa el Id del cliente para listar el local comercial al cual se va a generar el permiso de funcionamiento.
- c) La secretaria selecciona el local comercial resultante del proceso de búsqueda en el formulario.
- d) El sistema muestra un formulario donde se presenta una vista previa con la información del informe de la inspección física de dicho local comercial.
- e) La secretaria valida que el informe de la inspección del local comercial sea favorable e indica “Aceptar Informe”.
- f) El sistema muestra un formulario de ingreso donde se ingresa la información: tipo de permiso, Nro Permiso, Nro. Recibo, fecha permiso, valor permiso.
- g) La secretaria indica “Generar Permiso”.

- h) El Sistema muestra una vista previa del Permiso con un mensaje de confirmación.
- i) La secretaria indica “Aceptar”.
- j) El sistema realiza validaciones generales a los datos ingresados y registra la información.
- k) El caso de uso finaliza.

**Flujo Alternativo:**

**Local Comercial inexistente:**

En el paso b), si en el caso de que con el id del cliente no se listará local comercial alguno, se presentará un mensaje alertando “El cliente no registra local comercial.” y se termina el caso de uso.

**Informe no favorable:**

En el paso e), si la secretaria valida que no existe un informe favorable de la inspección física del local comercial indica “Cancelar” y se termina el caso de uso.

**3.2.8. Caso de Uso: Generar Conciliación Permisos y Rubros**

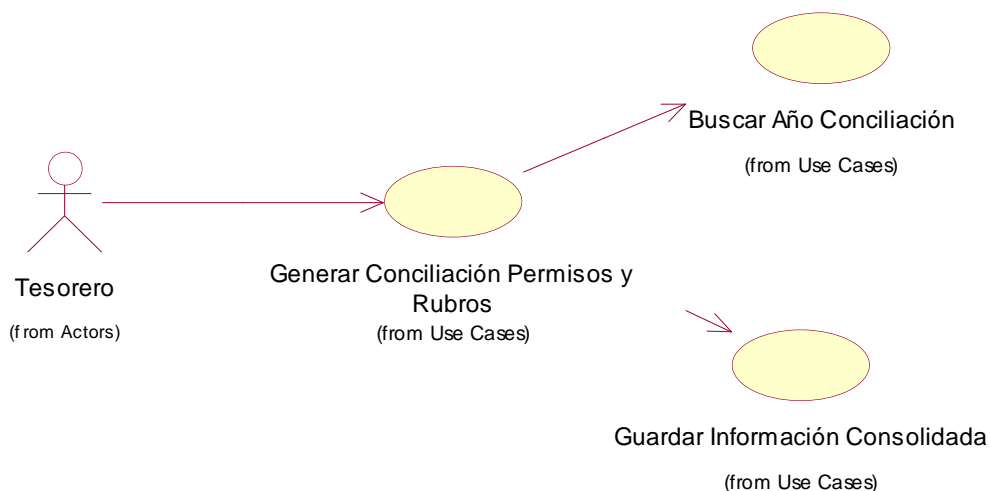


Figura 3- 12 Caso de Uso: Generar Conciliación Permisos y Rubros (*Fuente: Propia., 2017*)

**Caso de Uso:** Generar Conciliación Permisos y Rubros

**Actor:** Tesorero

**Precondiciones:**

- Se requiere que el empleado este registrado en el sistema como usuario del sistema y con el rol de Tesorero.
- El usuario ha sido admitido con el rol de Tesorero (Inicio de sesión aceptada por el sistema)

**Flujo Básico:**

- h) El caso de uso inicia cuando la secretaria indica “Generar Conciliación Permisos y Rubros”
- i) El sistema muestra un formulario de consulta donde se ingresa el año que se desea generar la información.
- j) El Tesorero indica “Generar Información”.
- k) El Sistema guarda la información consolidada en el repositorio: CONSOLIDACION RECAUDACION PERMISOS, agrupando los valores generados en los permisos de funcionamiento de acuerdo al año seleccionado por cada tipo de local comercial con información generada en forma MENSUAL y ANUAL.
- l) El caso de uso finaliza.

**Flujo Alternativo:**

No existen flujos alternos para este caso de uso.

### 3.2.9. Caso de Uso: Control de Asistencia

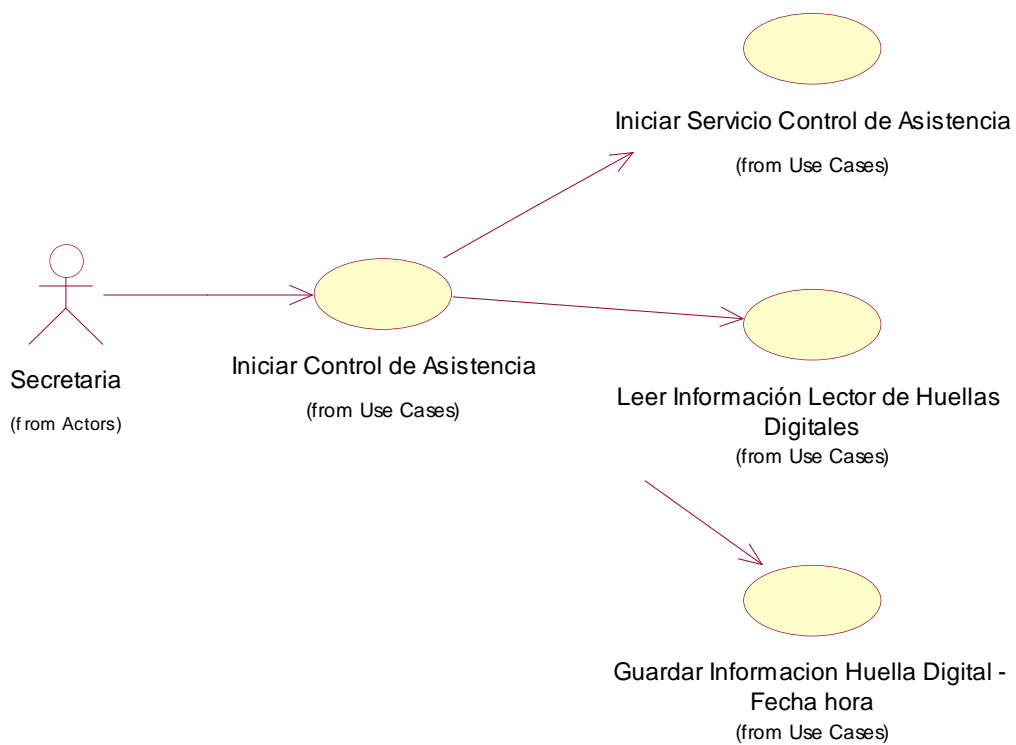


Figura 3- 13 Caso de Uso Iniciar Control de Asistencia (*Fuente: Propia., 2017*)

#### **Caso de Uso:** Iniciar Control de Asistencia

**Actor:** Secretaria

#### **Precondiciones:**

- Se requiere que el dispositivo denominado “Lector de huella digital”, esté conectado al computador el cual servirá de receptor de la huella digital de cada empleado.
- Se requiere que el empleado que usa el lector de huella digital para registrarse diariamente este registrado como empleado en el sistema, no es necesario que inicie sesión.

- Se requiere que el empleado que inicia el servicio de Control de Asistencia esté registrado como usuario del sistema y con el rol de Secretaria.
- El usuario ha sido admitido con el rol de Secretaria (Inicio de sesión aceptada por el sistema)

### **Flujo Básico:**

- a) El caso de uso inicia cuando la secretaria indica “Iniciar Control de Asistencia”
- b) El sistema muestra en pantalla un formulario donde se debe confirmar el inicio del Control de Asistencia
- c) La Secretaria indica “Iniciar Servicio”.
- d) El Sistema inicia el servicio y lee la información del lector de huellas digitales cada vez que un empleado hace uso del dispositivo para registrarse
- e) El sistema guarda esta información en el repositorio TIME\_STAMP de acuerdo al ID del empleado en el sistema
- f) El sistema presenta una pantalla que muestra las personas que registran su ingreso y salida en el transcurso de ese día.
- g) El servicio permanecerá activo hasta que la secretaria detiene el servicio al finalizar el día
- h) El caso de uso finaliza.

### **Flujo Alternativo:**

#### **Empleado no registrado:**

En el paso d) si el sistema detecta que el empleado no está registrado en el sistema muestra el mensaje: “Empleado No registrado en el sistema.”.



### 3.2.10. Caso de Uso: Mantener Tipos de Empleados

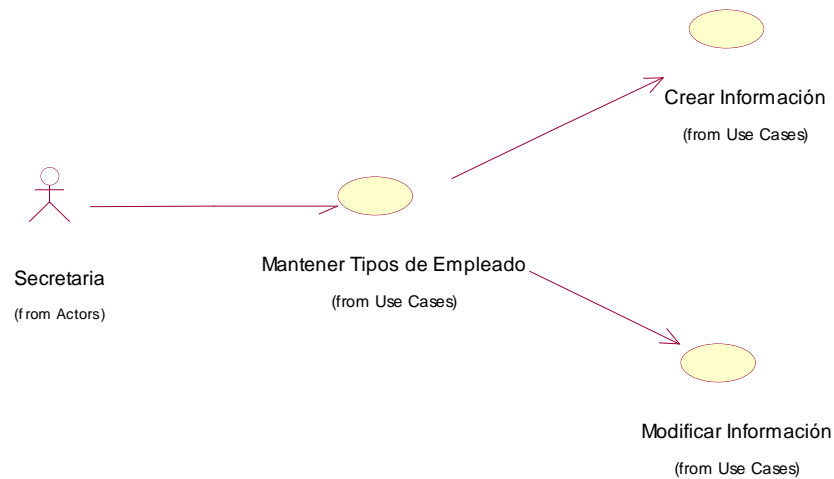


Figura 3- 14 Caso de Uso: Mantener Tipos de Empleado. (Fuente: Propia., 2017)

#### Caso de Uso: Mantener Tipos de empleados

**Actor:** Secretaria

#### Precondiciones:

- Se requiere que el empleado esté registrado como usuario del sistema y con el rol de Secretaria.
- El usuario ha sido admitido con el rol de Secretaria (Inicio de sesión aceptada por el sistema)

El Tipo de Empleado será dado de alta en el sistema con la información completa en sus dos flujos básicos:

- **Flujo Básico: Crear Tipo de Empleado**
  - a) El caso de uso inicia cuando el usuario indica “Crear, Modificar Tipos de empleado”.
  - b) El sistema muestra un formulario donde el usuario ingresa el **Cargo** y la **Dependencia** la que pertenecerá el **Tipo de Empleado** que se va a crear.

- c) El usuario indica "Guardar"
- d) El sistema realiza las validaciones generales de los datos ingresados.
- e) El caso de uso finaliza

- **Flujo Alternativo:**

**Validaciones generales:**

En el paso d), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

- **Flujo Básico: Modificar Tipo de Empleado**

- a) El caso de uso inicia cuando el usuario indica "Crear, Modificar Tipos de empleado".
- b) El sistema muestra un formulario, donde el usuario ingresa el id del tipo de empleado para filtrar el registro que se necesita modificar.
- f) El sistema muestra un formulario con la información del tipo de empleado seleccionado donde se puede modificar información del Cargo y la Dependencia.
- c) El usuario indica "Guardar"
- d) El sistema realiza las validaciones generales de los datos ingresados.
- e) El caso de uso finaliza

- **Flujo Alternativo:**

**Validaciones generales:**

En el paso e), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

### Registro Inexistente:

En el paso b), si no existe el registro el Sistema debe presentar el mensaje: “Registro Inexistente”.

### 3.2.11. Caso de Uso: Mantener Tipos de Locales Comerciales

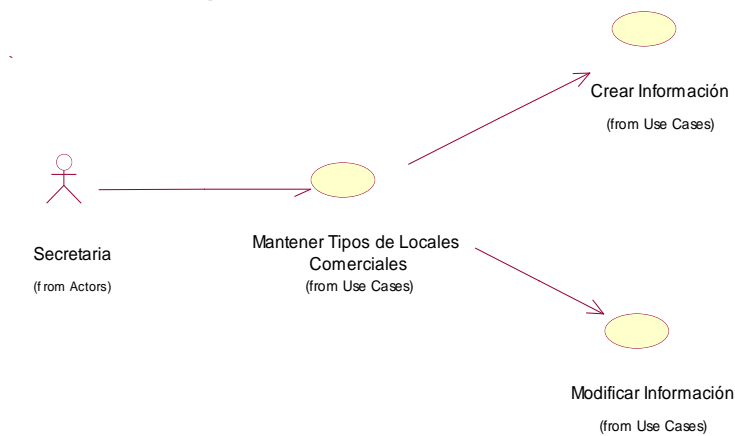


Figura 3- 15 Caso de Uso: Mantener Tipos de Locales Comerciales. (Fuente: Propia., 2017)

### Caso de Uso: Mantener Tipos de Locales Comerciales

**Actor:** Secretaria

### Precondiciones:

- Se requiere que el empleado esté registrado como usuario del sistema y con el rol de Secretaria.
- El usuario ha sido admitido con el rol de Secretaria (Inicio de sesión aceptada por el sistema)

El Tipo de Local Comercial será dado de alta en el sistema con la información completa en sus dos flujos básicos:

- **Flujo Básico: Crear Tipo de Local Comercial**

- a) El caso de uso inicia cuando el usuario indica “Crear, Modificar Tipos de Locales Comerciales”.
- b) El sistema muestra un formulario donde el usuario ingresa la **Descripción** del Tipo de Local Comercial.
- c) El usuario indica “Guardar”
- d) El sistema realiza las validaciones generales de los datos ingresados.
- e) El caso de uso finaliza

**Flujo Alternativo:**

**Validaciones generales:**

En el paso d), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

- **Flujo Básico: Modificar Tipo de Empleado**

- a) El caso de uso inicia cuando el usuario indica “Crear, Modificar Tipos de Locales Comerciales”.
- b) El sistema muestra un formulario, donde el usuario ingresa el id del tipo de Local Comercial para filtrar el registro que se necesita modificar.
- f) El sistema muestra un formulario con la información del Tipo de Local Comercial seleccionado donde se puede modificar información de su **Descripción**.
- c) El usuario indica “Guardar”
- d) El sistema realiza las validaciones generales de los datos ingresados.
- e) El caso de uso finaliza

### Flujo Alternativo:

### Validaciones generales:

En el paso d), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

### Registro Inexistente:

En el paso b), si no existe el registro el Sistema debe presentar el mensaje: “Registro Inexistente”.

### 3.2.12. Caso de Uso: Mantener Tarifas Locales Comerciales

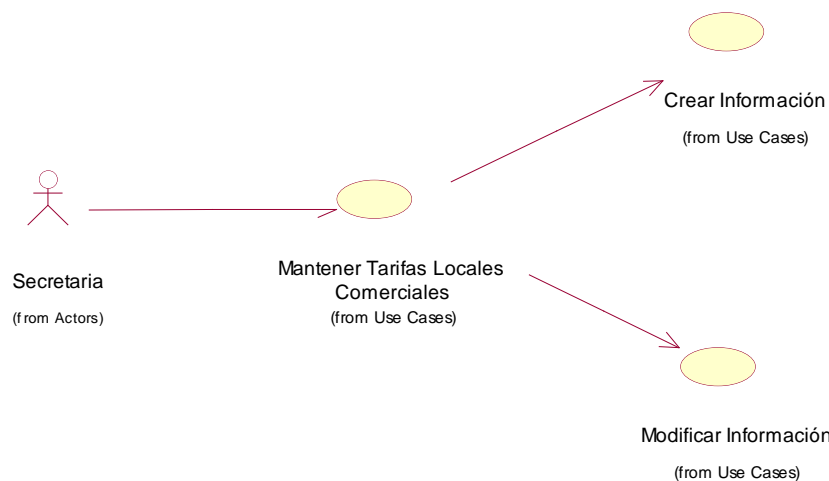


Figura 3- 16 Caso de Uso: Mantener Tarifas Locales Comerciales.

(Fuente: Propia., 2017)

### Caso de Uso: Mantener Tarifas Locales Comerciales

**Actor:** Secretaria

## **Precondiciones:**

- Se requiere que el empleado esté registrado como usuario del sistema y con el rol de Secretaria.
- El usuario ha sido admitido con el rol de Secretaria (Inicio de sesión aceptada por el sistema)

Las Tarifas de Local Comercial será dado de alta en el sistema con la información completa en sus dos flujos básicos:

- **Flujo Básico: Crear Tipo de Local Comercial**
  - a) El caso de uso inicia cuando el usuario indica “Crear, Modificar Tarifas Locales Comerciales”.
  - b) El sistema muestra un formulario donde el usuario ingresa el Id del Tipo de Local Comercial donde se mostrará en modo solo lectura la descripción del Tipo de Local Comercial que se seleccionó.
  - c) Se habilitarán los campos: **Valor Tipo Pequeño, Valor Tipo Mediano, Valor Tipo Grande, Año de Vigencia** donde se debe ingresar de manera obligatoria esta información.
  - d) El usuario indica “Guardar”
  - e) El sistema realiza las validaciones generales de los datos ingresados.
  - f) El caso de uso finaliza

## **Flujo Alternativo:**

### **Validaciones generales:**

En el paso e), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

- **Flujo Básico: Modificar Tipo de Empleado**

- a) El caso de uso inicia cuando el usuario indica “Crear, Modificar Tarifas Locales Comerciales”.
- b) El sistema muestra un formulario donde el usuario ingresa el Id del Tipo de Local Comercial donde se mostrará en modo solo lectura la descripción del Tipo de Local Comercial que se seleccionó.
- c) Se habilitarán los campos: **Valor Tipo Pequeño, Valor Tipo Mediano, Valor Tipo Grande, Año de Vigencia** donde se puede modificar esta información previamente ingresada.
- d) El usuario indica “Guardar”
- e) El sistema realiza las validaciones generales de los datos ingresados.
- f) El caso de uso finaliza

- **Flujo Alternativo:**

#### **Validaciones generales:**

En el paso d), si se genera un error en las validaciones generales se debe presentar un mensaje de error explicando el motivo, el código del error y el campo en el cual se produjo y se termina el caso de uso.

#### **Registro Inexistente:**

En el paso b), si no existe el registro el Sistema debe presentar el mensaje: “Registro Inexistente”.

### **3.3. Modelo Conceptual del Sistema**

Aquí se define las entidades y sus relaciones mediante el modelo conceptual: **ENTIDAD-RELACIÓN** las cuales que van a participar en el diseño del sistema, así

como sus respectivos atributos los cuales estarán representados de la siguiente manera:

- La entidad estará representada por un cuadrado con el respectivo nombre:
- Los atributos estarán representados por un ovalo con la descripción del atributo, el atributo clave estará representado por el nombre subrayado y en negritas.
- La relación estará representada por un rombo en el cual la descripción estará representada por un verbo, la cardinalidad se colocará en la línea de inicio y fin de cada relación.



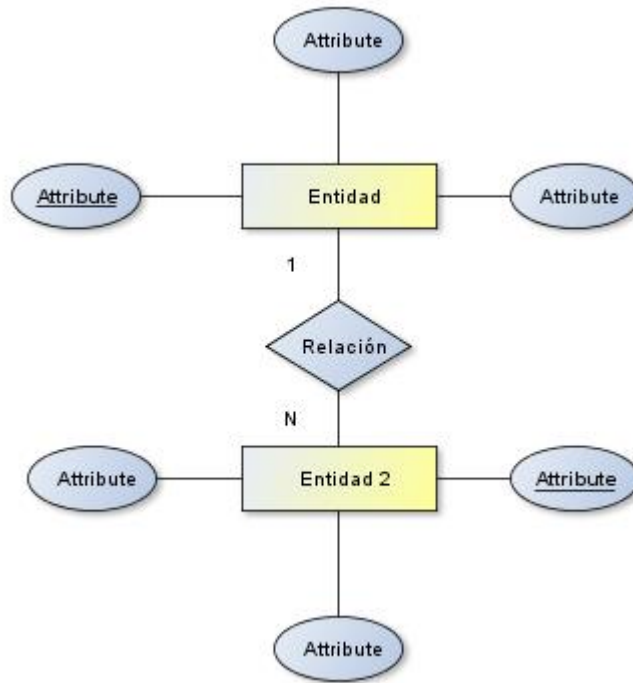
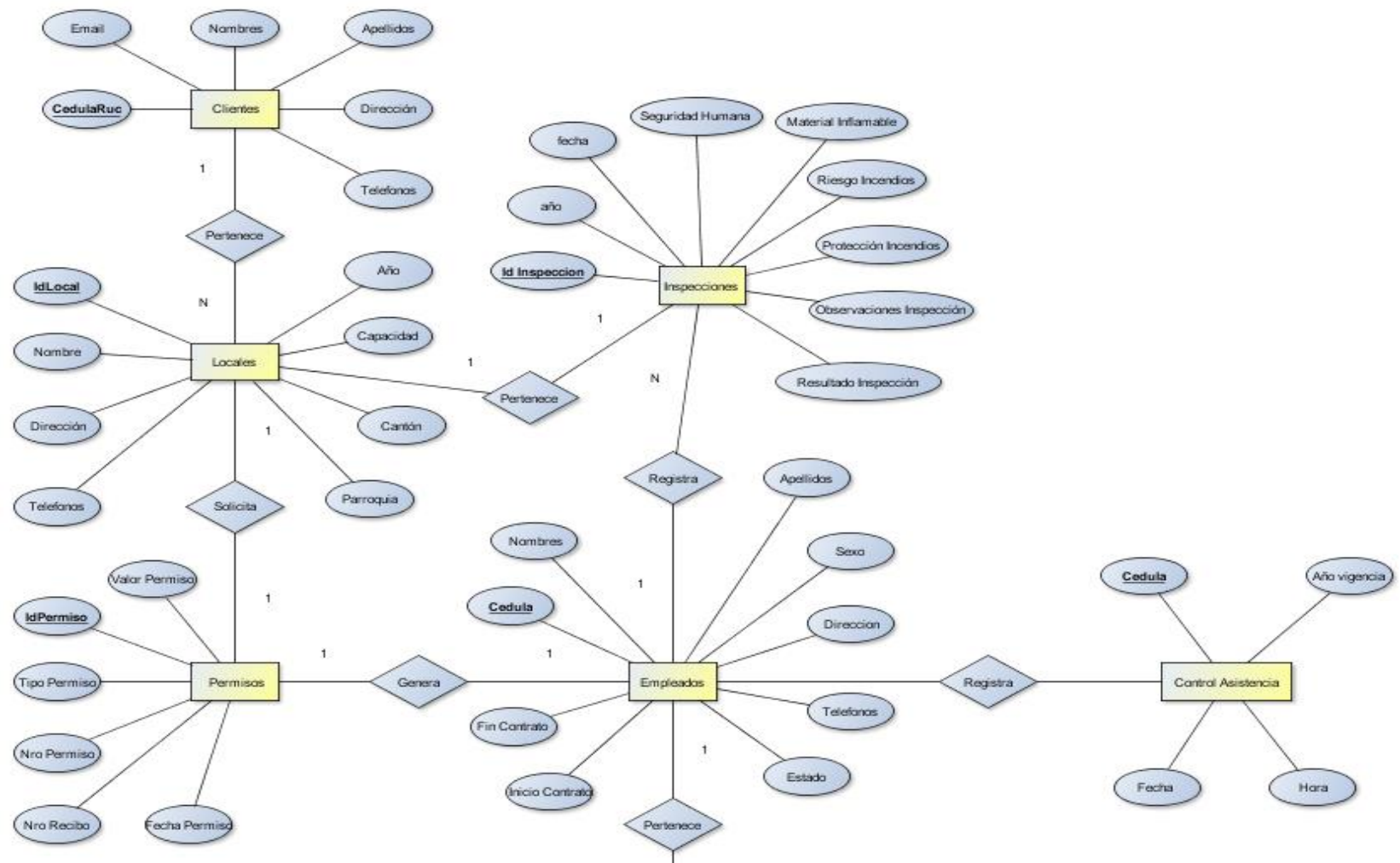


Figura 3- 17 Simbología Modelo Conceptual (*Fuente: Propia., 2017*)



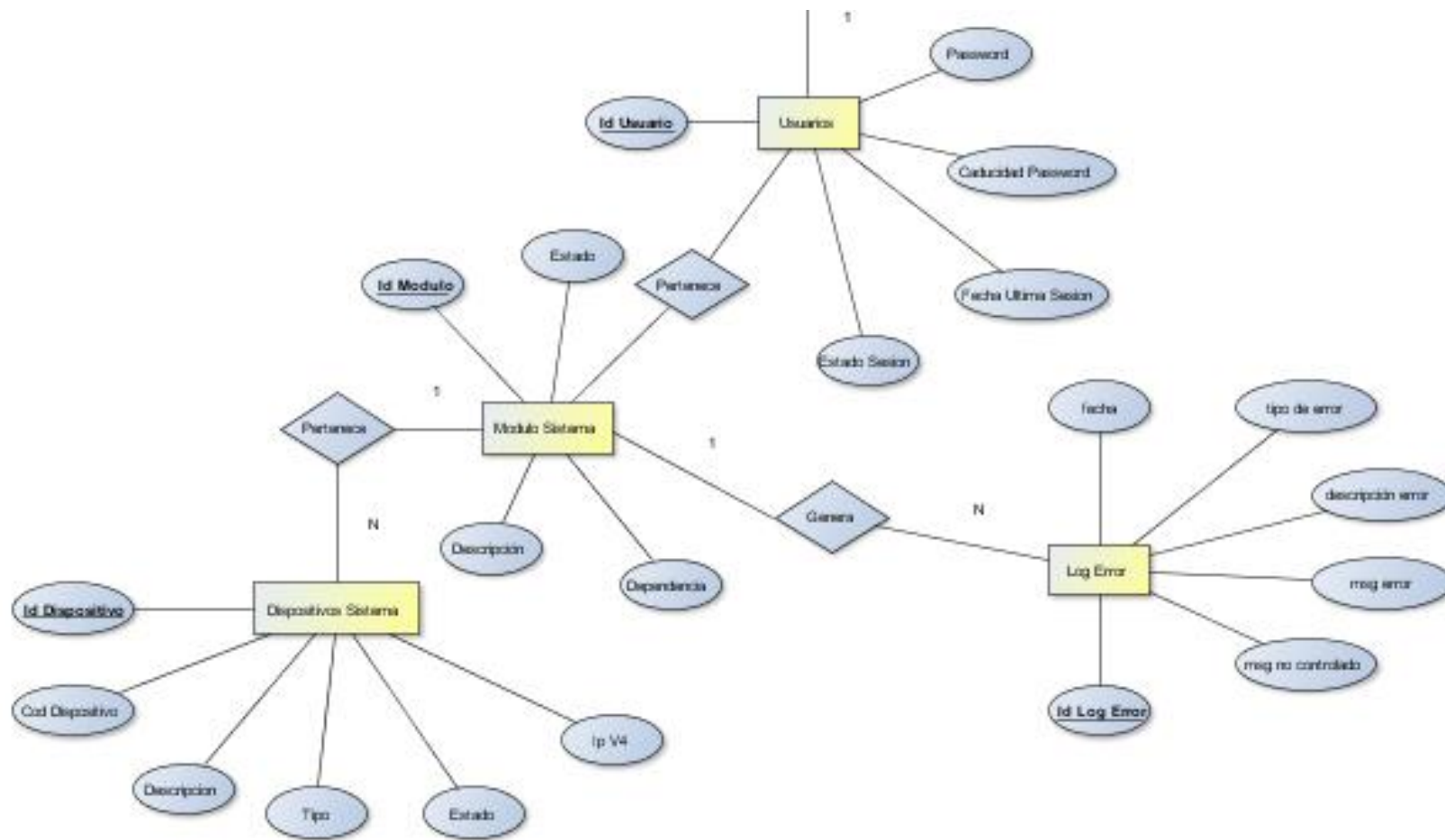
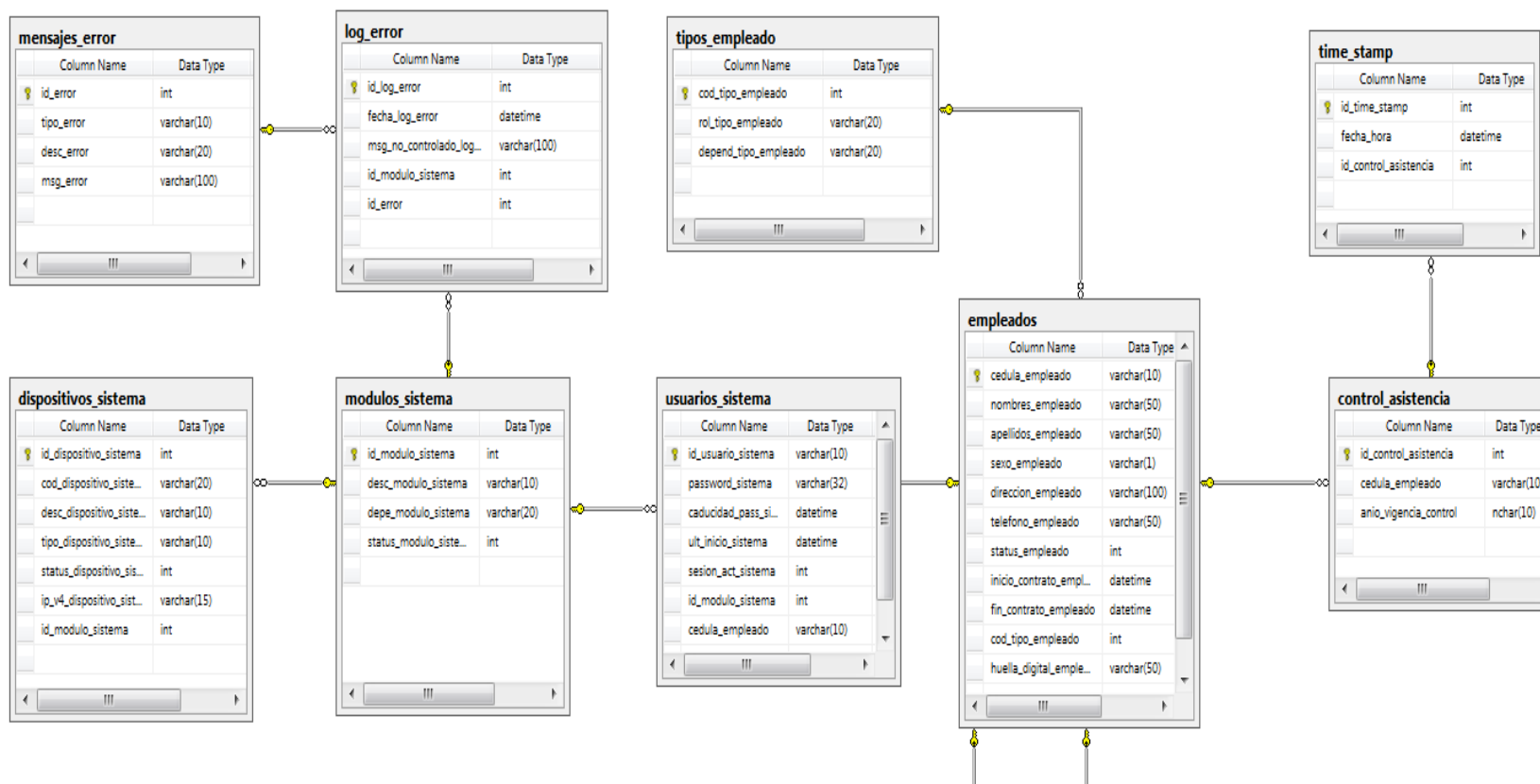


Figura 3- 18 Modelo Conceptual del Sistema. (Fuente: Propia., 2017)

### 3.4. Diseño Físico de la Base de Datos

En este modelo físico están representados las entidades, atributos y relaciones que se analizaron en el modelo conceptual:



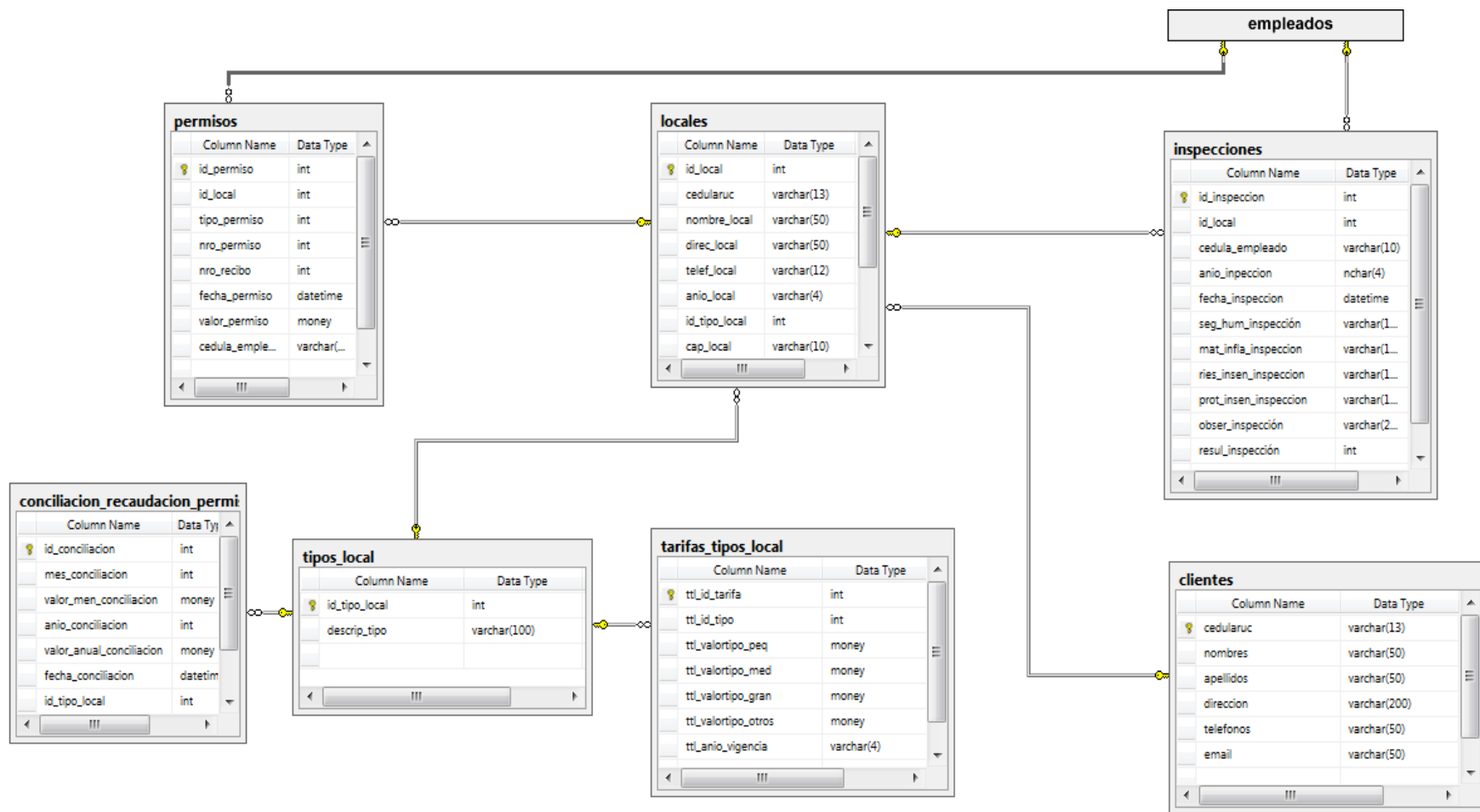


Figura 3- 19 Diseño Físico de la Base de Datos (Fuente: Propia., 2017)

### 3.5. Diagrama de Arquitectura del Sistema

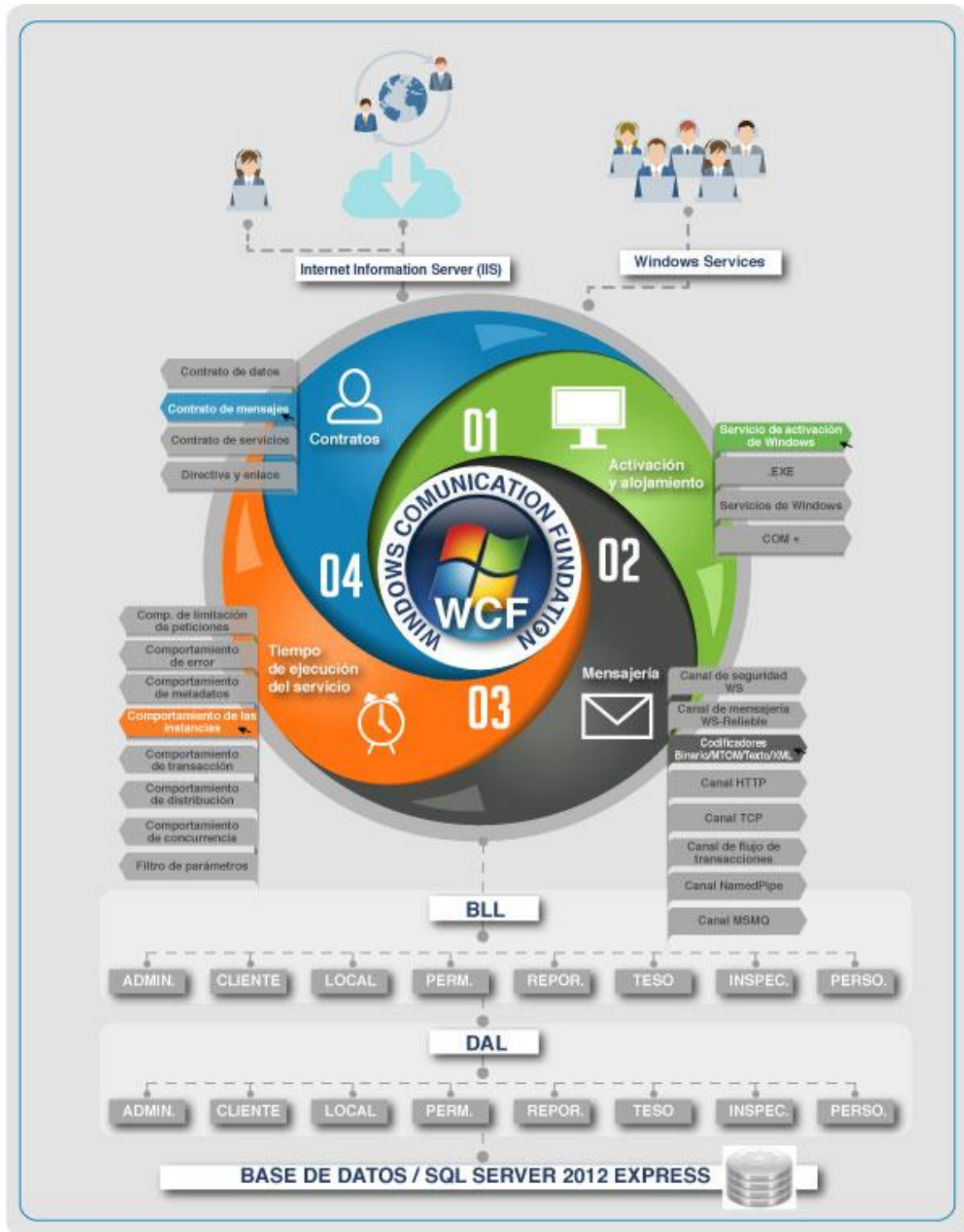


Figura 3- 20 Diagrama de Arquitectura del Sistema. (Fuente: Propia., 2017)

### 3.6. Diseño de la Interfaz Gráfica

Aquí se muestra la interfaz gráfica que se le va a presentar al usuario final en cada funcionalidad que integra el sistema:



Figura 3- 21 Interfaz Gráfica: Registrar. Modificar Usuarios del Sistema. (Fuente: Propia., 2017)

Todos los demás diseños de pantallas se encuentran en el manual de usuario como ANEXOS.





## CAPITULO IV

### 4. IMPLEMENTACIÓN

#### 4.1. Instalación y configuración de la Base de Datos SQL Server Express 2012.

##### 4.1.1. Requisitos del Sistema.

**Sistema Operativo:** Windows 7; Windows Server 2008 R2; Windows Server 2008 Service Pack 2; Windows Vista Service Pack 2, NET Framework 3.5 Service Pack 1 y poder acceder al centro de descargas para poder actualizar con la versión 4.0.

**Hardware (32 bits):** Equipo con procesador Intel o compatible a 1 GHz o superior (se recomienda 2 GHz o superior). 512 MB de RAM como mínimo (se recomienda 2 GB o más), 2.2 GB de espacio disponible en disco.

**Hardware (64 bits):** Procesador a 1,4 GHz o superior. 512 MB de RAM como mínimo (se recomienda 2 GB o más), 2.2 GB de espacio disponible en disco.

##### 4.1.2. Instalación.

a) Descargar la aplicación de forma gratuita de la dirección:

<https://www.microsoft.com/es-es/download/details.aspx?id=29062>

b) Luego de descargar la aplicación se lanza el ejecutable:

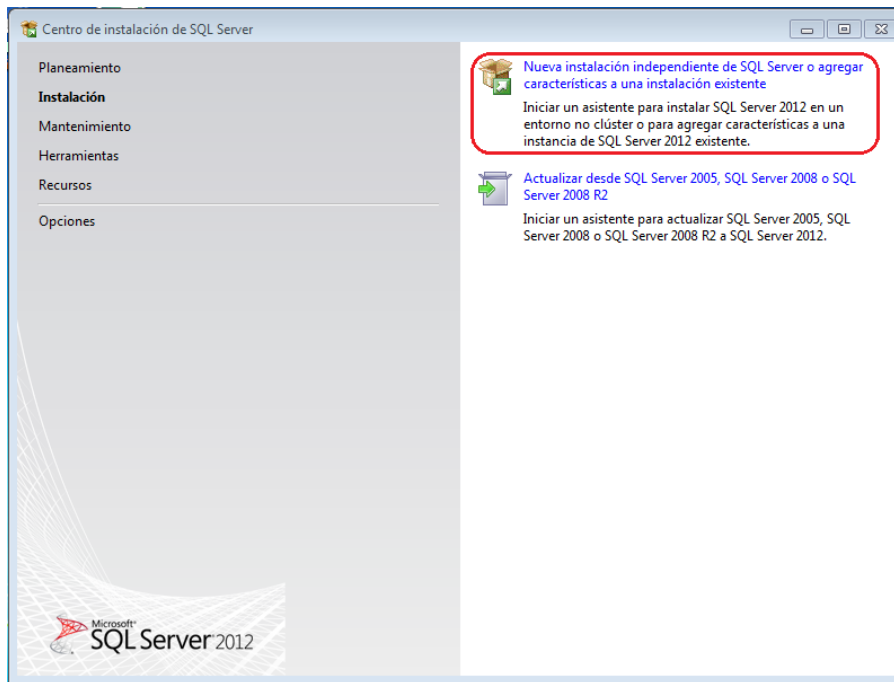


Figura 4- 1 Selección Tipo de Instalación Sql Server Xpress 2012.

(Fuente: Propia., 2017)

- c) Luego de un par de siguientes, presionar siguiente para buscar actualizaciones, se copian los archivos del programa de instalación, se comprueban las reglas auxiliares y se elige las características a instalar, pulsar siguiente y se empiezan a copiar los archivos.

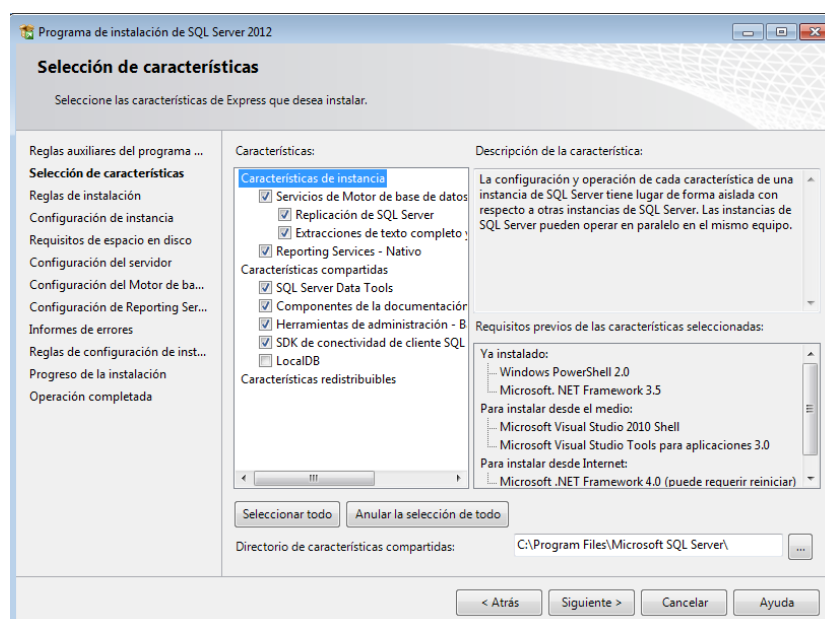


Figura 4- 2 Selección de Características Sql Server Xpress 2012

(Fuente: Propia., 2017)

- d) Se muestra la pantalla de elección de las características es la configuración de la instancia. Aquí especificar el nombre y directorio raíz:

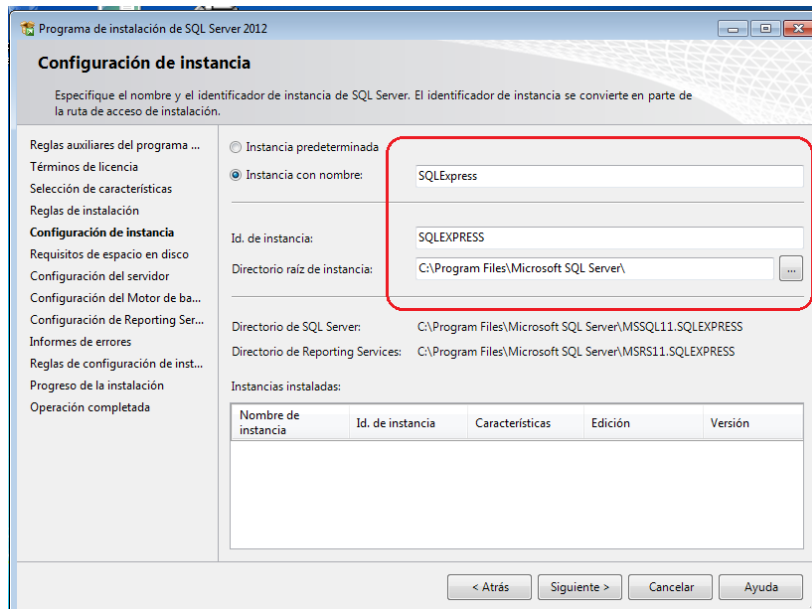


Figura 4- 3 Configuración Instancia Sql Server Xpress 2012.

(Fuente: Propia., 2017)

- e) Continuar con la configuración de las cuentas que ejecutaran los distintos servicios incluidos (el motor del sgbd, reporting services y sql server browser). Recordar que Sql Server Browser es como un "agente" que publica en la red la existencia de la instancia de SQL Server. Por defecto viene deshabilitado y si no es por causa mayor mejor dejarlo así:

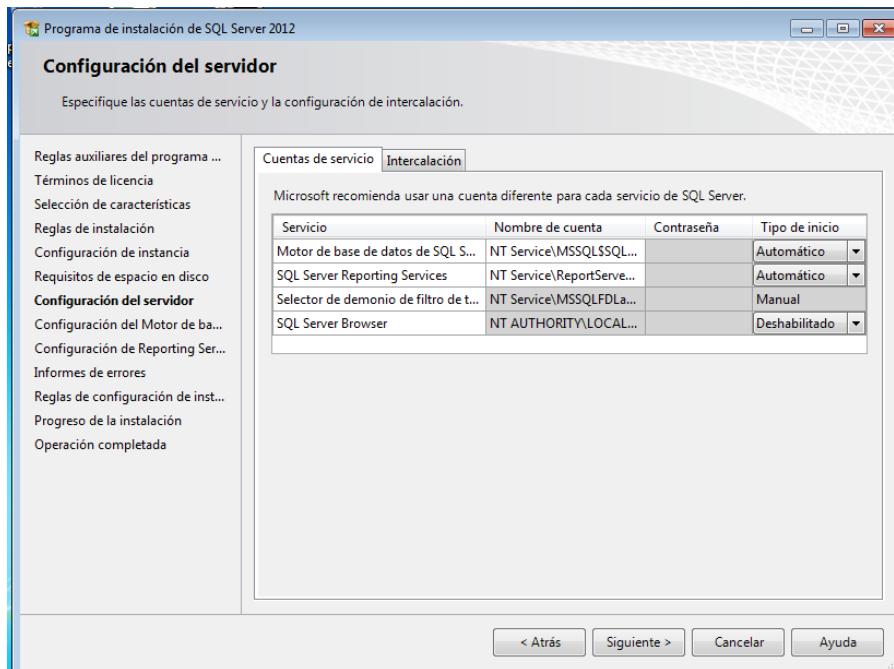


Figura 4- 4 Configuración Servidor Sql Server Xpress 2012.

(Fuente: Propia., 2017)

- f) El siguiente paso es elegir el modo de autenticación y los administradores de la base de datos. Generalmente conviene elegir mixto si esta base de datos se necesita usar en la red local o autenticación de Windows si directamente es de uso local y no para no complicar la instalación.

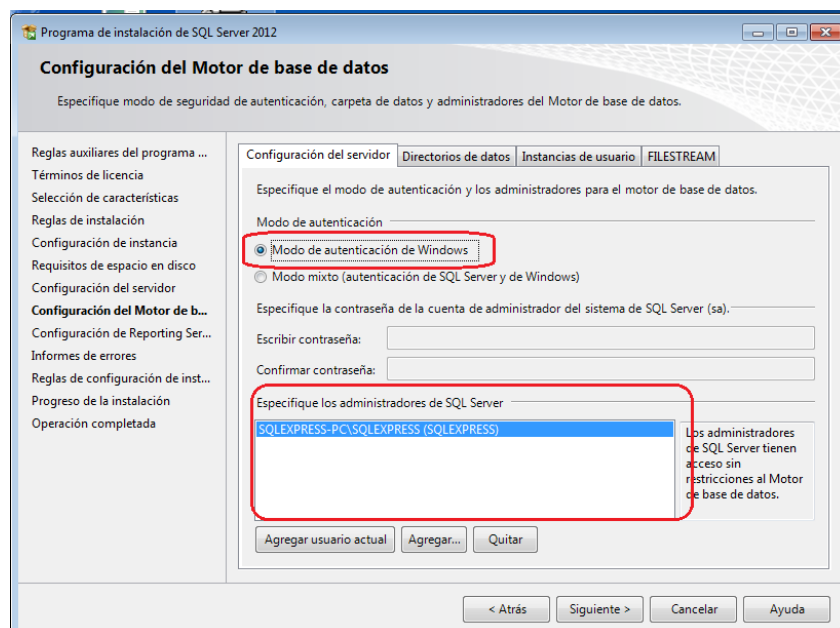


Figura 4- 5 Configuración Motor de BDD Sql Server Xpress 2012.

(Fuente: Propia., 2017)

- g) Para la configuración por defecto de Reporting Services, simplemente basta con dejar la primera opción seleccionada, la que permite usarlo en modo nativo.

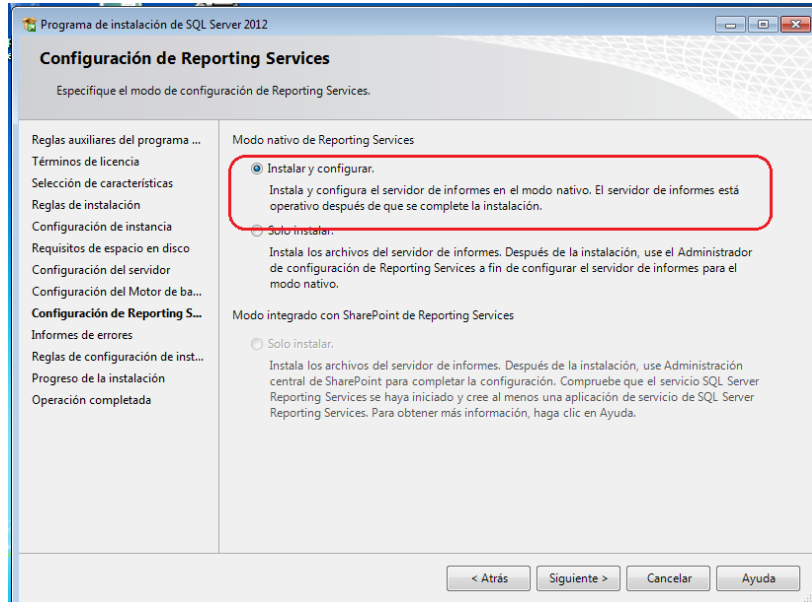


Figura 4- 6 Configuración de Reporting Services Sql Server Xpress 2012.

(Fuente: Propia., 2017)

- h) El último siguiente se refiere a la privacidad y la notificación de errores a Microsoft y ya estaría listo e instalado:

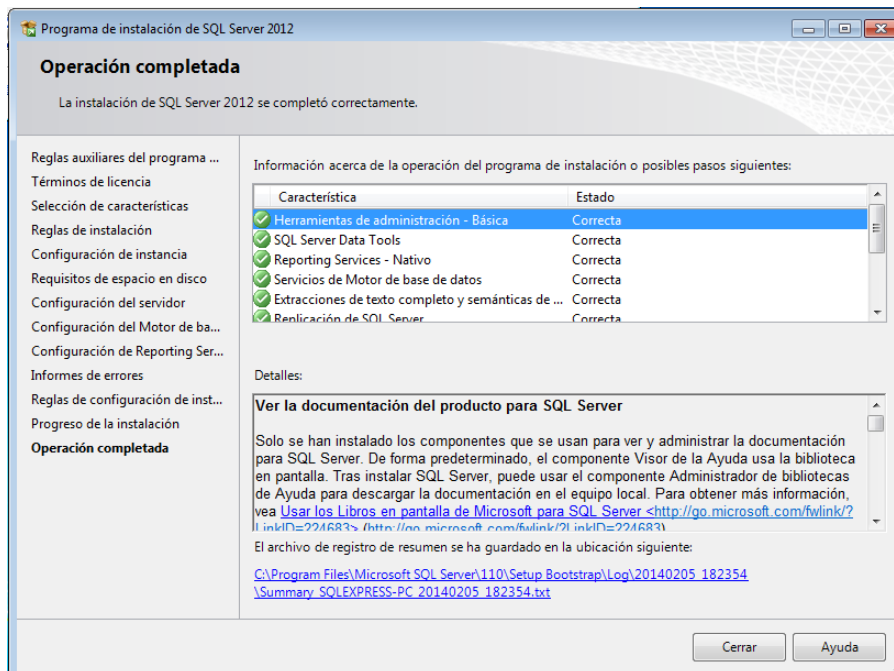


Figura 4- 7 Operación Completa Instalación Sql Server Xpress 2012.

(Fuente: Propia., 2017)

- i) Reiniciar el sistema para finalizar la instalación de la versión express de sql server 2012 con servicios avanzados.

## 4.2. Implementación de la Base de Datos.

### 4.2.1. Creación de Base de datos, Tablas y Relaciones.

#### 4.2.1.1. Creación de la Base de Datos

- a) Buscar la aplicación Inicio->Microsoft SQL Server 2012 -> Sql Server Management Studio:

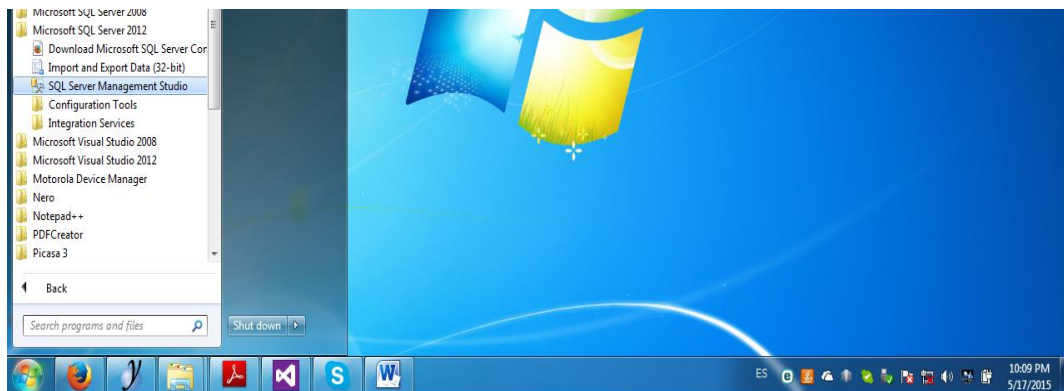


Figura 4- 8 Ruta Acceso Sql Server Management Studio 2012. (Fuente: Propia., 2017)

- b) En este caso como se escogió anteriormente la autenticación de Windows para iniciar sesión en el Sql Server 2012 presionar “Conectar”.



Figura 4- 9 Iniciar Sesión Sql Server Management Studio 2012.

(Fuente: Propia., 2017)

- c) Proceder a crear la base de datos bdd\_BOMBEROS en la cual se va a definir los objetos más adelante:

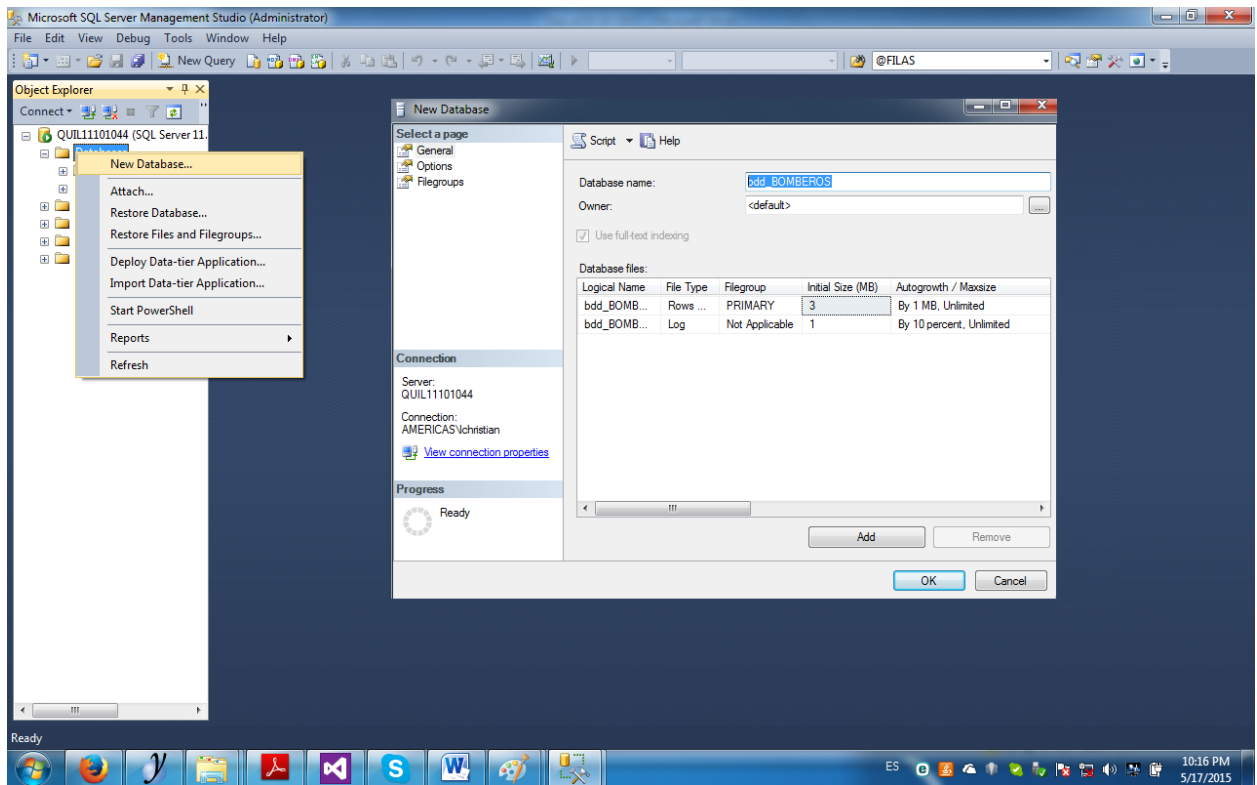


Figura 4- 10 Crear Base de Datos Sql Server Management Studio 2012.

(Fuente: Propia., 2017)

#### 4.2.1.2. Creación de las tablas del Sistema

De acuerdo al diseño del diagrama conceptual el cual contiene las entidades, atributos y relaciones, se procede a implementar las tablas del Sistema:

- a) Se crean las tablas: `modulos_sistema`, `dispositivos_sistema`, `mensajes_error`, `log_error` y sus respectivas relaciones:

Column Name	Data Type	Allow Nulls
<code>id_modulo_sistema</code>	int	<input type="checkbox"/>
<code>desc_modulo_sistema</code>	varchar(10)	<input type="checkbox"/>
<code>depe_modulo_sistema</code>	varchar(20)	<input type="checkbox"/>
<code>status_modulo_sistema</code>	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
<code>id_error</code>	int	<input type="checkbox"/>
<code>tipo_error</code>	varchar(10)	<input type="checkbox"/>
<code>desc_error</code>	varchar(20)	<input type="checkbox"/>
<code>msg_error</code>	varchar(100)	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
<code>id_dispositivo_sistema</code>	int	<input type="checkbox"/>
<code>cod_dispositivo_sistema</code>	varchar(20)	<input type="checkbox"/>
<code>desc_dispositivo_sistema</code>	varchar(10)	<input type="checkbox"/>
<code>tipo_dispositivo_sistema</code>	varchar(10)	<input type="checkbox"/>
<code>status_dispositivo_sistema</code>	int	<input type="checkbox"/>
<code>ip_v4_dispositivo_sistema</code>	varchar(15)	<input checked="" type="checkbox"/>
<code>id_modulo_sistema</code>	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
<code>id_log_error</code>	int	<input type="checkbox"/>
<code>fecha_log_error</code>	datetime	<input type="checkbox"/>
<code>msg_no_controlado_log_error</code>	varchar(100)	<input checked="" type="checkbox"/>
<code>id_modulo_sistema</code>	int	<input checked="" type="checkbox"/>
<code>id_error</code>	int	<input checked="" type="checkbox"/>



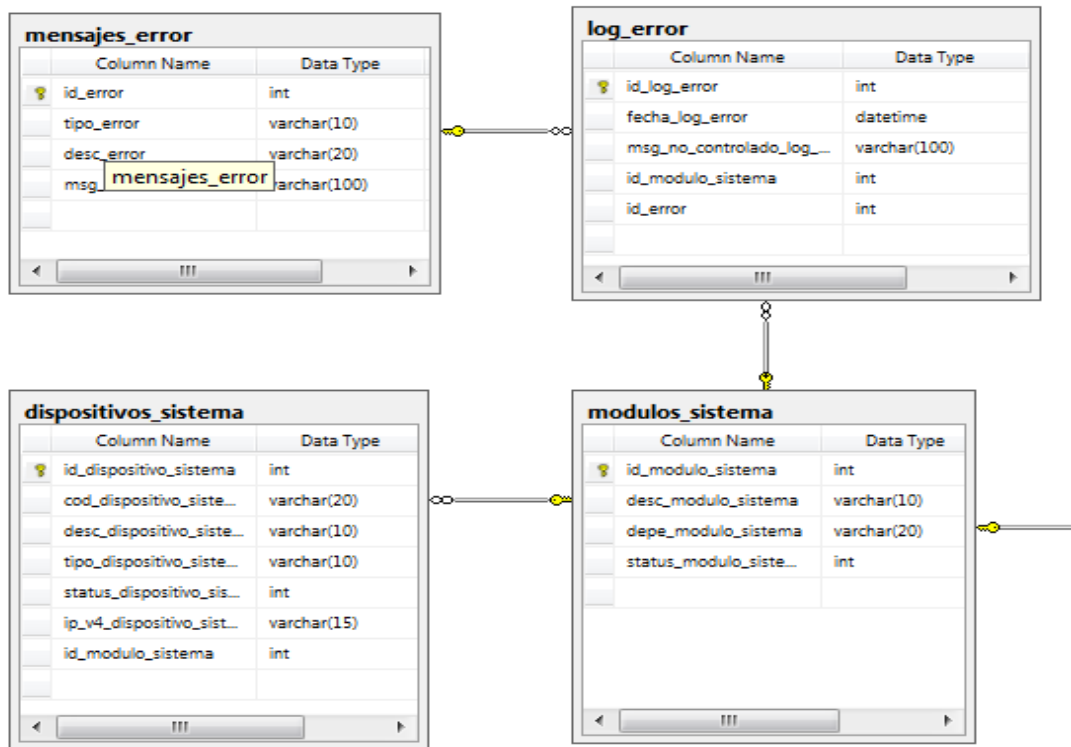


Figura 4- 11 Crear tablas y relaciones para: modulos\_sistema, dispositivos\_sistema, mensajes\_error, log\_error

(Fuente: Propia., 2017)

- b) Crear las tablas: tipos\_empleado, empleados, usuarios\_sistema, control\_asistencia, time\_stamp, esta última se creará y asociará a control\_asistencia para alivianar la carga de información la cual será significativa con el tiempo:

QUIL11101044.bdd\_...bo.tipos\_empleado X QUIL11101044.bdd

Column Name	Data Type	Allow Nulls
<b>cod_tipo_empleado</b>	int	<input type="checkbox"/>
rol_tipo_empleado	varchar(20)	<input type="checkbox"/>
depend_tipo_empleado	varchar(20)	<input type="checkbox"/>

QUIL11101044.bdd\_...o.usuarios\_sistema X QUIL11101044.bdd

Column Name	Data Type	Allow Nulls
<b>id_usuario_sistema</b>	varchar(10)	<input type="checkbox"/>
password_sistema	varchar(32)	<input type="checkbox"/>
caducidad_pass_sistema	datetime	<input type="checkbox"/>
ult_inicio_sistema	datetime	<input checked="" type="checkbox"/>
sesion_act_sistema	int	<input type="checkbox"/>
id_modulo_sistema	int	<input type="checkbox"/>
cedula_empleado	varchar(10)	<input type="checkbox"/>

QUIL11101044.bdd\_...S - dbo.empleados X QUIL11101044.bdd

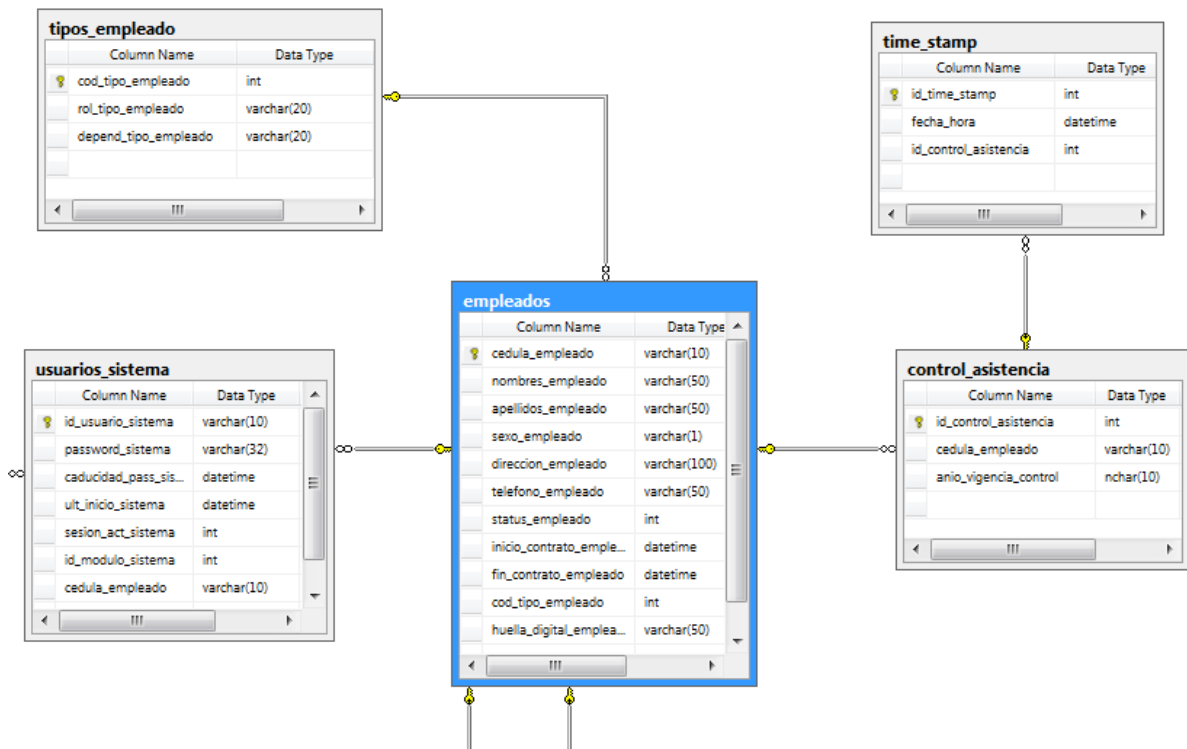
Column Name	Data Type	Allow Nulls
<b>cedula_empleado</b>	varchar(10)	<input type="checkbox"/>
nombres_empleado	varchar(50)	<input type="checkbox"/>
apellidos_empleado	varchar(50)	<input type="checkbox"/>
sexo_empleado	varchar(1)	<input type="checkbox"/>
direccion_empleado	varchar(100)	<input type="checkbox"/>
telefono_empleado	varchar(50)	<input type="checkbox"/>
status_empleado	int	<input type="checkbox"/>
inicio_contrato_empleado	datetime	<input type="checkbox"/>
fin_contrato_empleado	datetime	<input checked="" type="checkbox"/>
cod_tipo_empleado	int	<input type="checkbox"/>
huella_digital_empleado	varchar(50)	<input checked="" type="checkbox"/>

QUIL11101044.bdd\_...S - dbo.time\_stamp X QUIL11101044.bdd

Column Name	Data Type	Allow Nulls
<b>id_time_stamp</b>	int	<input type="checkbox"/>
fecha_hora	datetime	<input type="checkbox"/>
id_control_asistencia	int	<input type="checkbox"/>

QUIL11101044.bdd\_B...control\_asistencia X QUIL11101044.bdd

Column Name	Data Type	Allow Nulls
<b>id_control_asistencia</b>	int	<input type="checkbox"/>
cedula_empleado	varchar(10)	<input type="checkbox"/>
anio_vigencia_control	nchar(10)	<input type="checkbox"/>



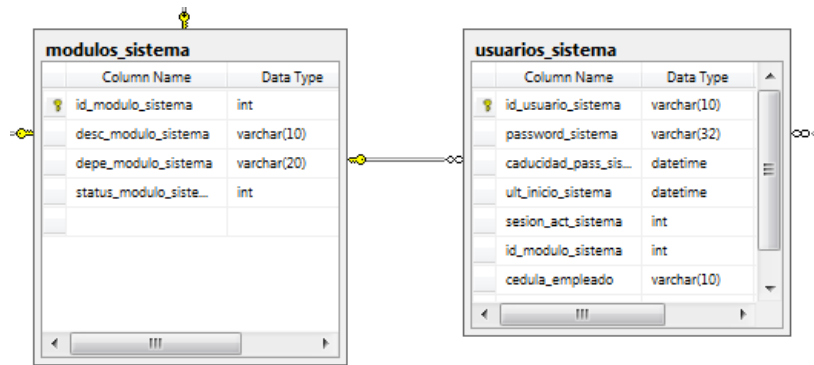


Figura 4- 12 Crear tablas y relaciones para: tipos\_empleado, empleados, usuarios\_sistema, control\_asistencia, time\_stamp.

(Fuente: Propia., 2017)

- c) Crear las tablas: tipos\_local, tarifas\_tipos\_local, locales, permisos, conciliacion\_recaudacion\_permisos.

Column Name	Data Type	Allow Nulls
id_tipo_local	int	<input type="checkbox"/>
descrip_tipo	varchar(100)	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
tll_id_tarifa	int	<input type="checkbox"/>
tll_id_tipo	int	<input type="checkbox"/>
tll_valortipo_peq	money	<input type="checkbox"/>
tll_valortipo_med	money	<input type="checkbox"/>
tll_valortipo_gran	money	<input type="checkbox"/>
tll_valortipo_otros	money	<input type="checkbox"/>
tll_anio_vigencia	varchar(4)	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
id_local	int	<input type="checkbox"/>
cedularuc	varchar(13)	<input type="checkbox"/>
nombre_local	varchar(50)	<input type="checkbox"/>
direc_local	varchar(50)	<input type="checkbox"/>
telef_local	varchar(12)	<input checked="" type="checkbox"/>
anio_local	varchar(4)	<input type="checkbox"/>
id_tipo_local	int	<input type="checkbox"/>
cap_local	varchar(10)	<input type="checkbox"/>
canton_local	varchar(50)	<input checked="" type="checkbox"/>
parroquia_lo...	varchar(50)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
id_conciliacion	int	<input type="checkbox"/>
mes_conciliacion	int	<input type="checkbox"/>
valor_men_conciliacion	money	<input type="checkbox"/>
anio_conciliacion	int	<input type="checkbox"/>
valor_anual_conciliacion	money	<input type="checkbox"/>
fecha_conciliacion	datetime	<input type="checkbox"/>
id_tipo_local	int	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
id_permiso	int	<input type="checkbox"/>
id_local	int	<input type="checkbox"/>
tipo_permiso	int	<input type="checkbox"/>
nro_permiso	int	<input type="checkbox"/>
nro_recibo	int	<input type="checkbox"/>
fecha_permiso	datetime	<input type="checkbox"/>
valor_permiso	money	<input type="checkbox"/>
cedula_empleado	varchar(10)	<input type="checkbox"/>



Figura 4- 13 Crear tablas y relaciones para: tipos\_local, tarifas\_tipos\_local, locales, permisos, conciliacion\_recaudacion\_permisos.

(Fuente: Propia., 2017)

d) Crear las tablas inspecciones, clientes:

Column Name	Data Type	Allow Nulls	Column N...	Data Type	Allow Nulls
<b>id_inspeccion</b>	int	<input type="checkbox"/>	<b>cedularuc</b>	varchar(13)	<input type="checkbox"/>
id_local	int	<input type="checkbox"/>	nombres	varchar(50)	<input type="checkbox"/>
cedula_employado	varchar(10)	<input type="checkbox"/>	<b>apellidos</b>	varchar(50)	<input type="checkbox"/>
anio_inspeccion	nchar(4)	<input type="checkbox"/>	direccion	varchar(200)	<input type="checkbox"/>
fecha_inspeccion	datetime	<input type="checkbox"/>	telefonos	varchar(50)	<input checked="" type="checkbox"/>
seg_hum_inspeccion	varchar(100)	<input type="checkbox"/>	email	varchar(50)	<input checked="" type="checkbox"/>
mat_infla_inspeccion	varchar(100)	<input type="checkbox"/>			
ries_insen_inspeccion	varchar(100)	<input type="checkbox"/>			
prot_insen_inspeccion	varchar(100)	<input type="checkbox"/>			
obser_inspeccion	varchar(200)	<input type="checkbox"/>			
resul_inspeccion	int	<input type="checkbox"/>			

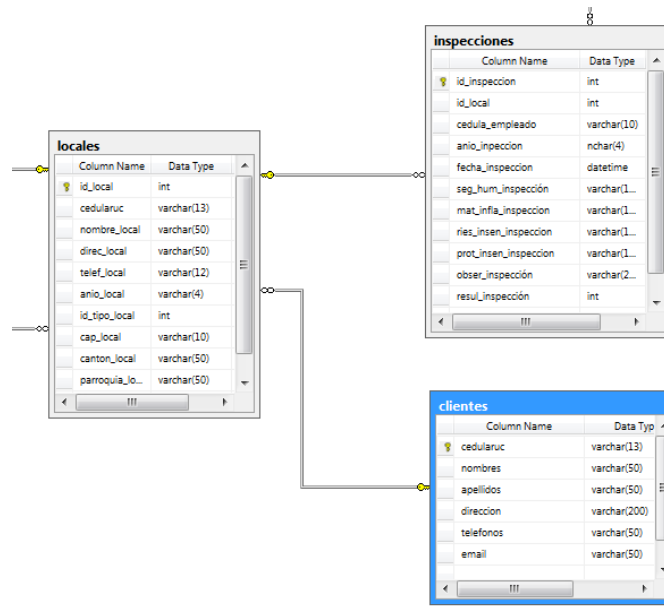


Figura 4- 14 Crear tablas y relaciones para: inspecciones, clientes.

(Fuente: Propia., 2017)

## 4.2.2. Creación de los procedimientos almacenados Create, Read, Update, Delete (CRUD).

### 4.2.2.1. Create Registro de Cliente:

**Nombre del Procedimiento:** InsertarCliente

**Parámetros de Entrada:**

Nombre del parámetro	Tipo de Dato
@cedularuc	INT
@nombres	VARCHAR(50)
@apellidos	VARCHAR(50)
@direccion	VARCHAR(200)
@telefonos	VARCHAR(10)
@email	VARCHAR(50)

**Parámetros de Salida:**

Nombre del parámetro	Tipo de Dato
@TRANSACCION	INT
@CODIGO_ERROR	INT
@MENSAJE_ERROR	VARCHAR(255)

**Código Fuente:**

```
1. CREATE PROCEDURE [dbo].[InsertarCliente]
2. @cedularuc INT = NULL,
```

```

3. @nombres VARCHAR(50) = NULL,
4. @apellidos VARCHAR(50) = NULL,
5. @direccion VARCHAR(200) = NULL,
6. @telefonos VARCHAR(50) = NULL,
7. @email VARCHAR(50) = NULL,
8. @TRANSACCION INT OUTPUT,
9. @CODIGO_ERROR INT OUTPUT,
10. @MENSAJE_ERROR VARCHAR(255) OUTPUT
11. AS
12. BEGIN
13. DECLARE @CONT INT;
14. DECLARE @FILAS INT;
15. DECLARE @TRANS VARCHAR(15);
16. SET NOCOUNT OFF;
17. SET @FILAS = (SELECT COUNT(1) FROM clientes WITH (NOLOCK)
WHERE cedularuc = @cedularuc);
18. SET @TRANS = 'INSERT'
19. BEGIN TRY
20. IF @FILAS = 0
21. BEGIN
22. BEGIN TRANSACTION @TRANS
23. INSERT INTO
clientes(cedularuc,nombres,apellidos,direccion,telefonos,email)
24. VALUES
(@cedularuc,@nombres,@apellidos,@direccion,@telefonos,@email)
;

25. SET @CONT = @@ROWCOUNT;
26. PRINT 'FILAS' + CAST(@FILAS AS VARCHAR(10));
27. PRINT 'CONTADOR' + CAST(@CONT AS VARCHAR(10));

28. END
29. IF @CONT = 1 AND @FILAS = 0
30. BEGIN
31. SET @TRANSACCION = 1;
32. SET @CODIGO_ERROR = 0;
33. SET @MENSAJE_ERROR = 'OK';
34. COMMIT TRANSACTION @TRANS;
35. END
36. ELSE IF @CONT = 0 OR @FILAS > 1
37. BEGIN
38. SET @TRANSACCION = 0;
39. SET @CODIGO_ERROR = 0;
40. SET @MENSAJE_ERROR = 'OK';
41. ROLLBACK TRANSACTION @TRANS;
42. END
43. ELSE
44. BEGIN
45. SET @TRANSACCION = -1;
46. SET @CODIGO_ERROR = -1;
47. SET @MENSAJE_ERROR = 'ERROR: Ha ocurrido un error inesperado
a nivel de Base de Datos. Vuelva a intentarlo más tarde.';
48. ROLLBACK TRANSACTION @TRANS;
49. END
50. END TRY

51. BEGIN CATCH
52. SET @TRANSACCION = -1;
53. SET @CODIGO_ERROR = ERROR_NUMBER();
54. SET @MENSAJE_ERROR = SUBSTRING(ERROR_MESSAGE(), 1, 255);
55. ROLLBACK TRANSACTION @TRANS;
56. END CATCH

```

#### 4.2.2.2. Read registro de Cliente.

**Nombre del Procedimiento:** BuscarCliente

**Parámetros de Entrada:**

Nombre del parámetro	Tipo de Dato
@cedularuc	INT

**Parámetros de Salida:**

Nombre del parámetro	Tipo de Dato
@TRANSACCION	INT
@CODIGO_ERROR	INT
@MENSAJE_ERROR	VARCHAR(255)

**Código Fuente:**

```
1. CREATE PROCEDURE [dbo].[BuscarCliente]
2. @cedularuc INT = NULL,
3. @TRANSACCION INT OUTPUT,
4. @CODIGO_ERROR INT OUTPUT,
5. @MENSAJE_ERROR VARCHAR(255) OUTPUT
6. AS
7. BEGIN
8. DECLARE @CONT INT;
9. SET NOCOUNT ON;
10. BEGIN TRY
11. SELECT * FROM clientes WITH (NOLOCK) WHERE cedularuc =
    @cedularuc;
12. SET @CONT = @@ROWCOUNT;
13. PRINT 'CONTADOR ' + CAST(@CONT AS VARCHAR(10));
14. IF @CONT >= 1
15. BEGIN
16. SET @TRANSACCION = 1;
17. SET @CODIGO_ERROR = 0;
18. SET @MENSAJE_ERROR = 'OK';
19. END
20. ELSE IF @CONT = 0
21. BEGIN
22. SET @TRANSACCION = 0;
23. SET @CODIGO_ERROR = 0;
24. SET @MENSAJE_ERROR = 'OK';
25. END
26. ELSE
27. BEGIN
28. SET @TRANSACCION = -1;
29. SET @CODIGO_ERROR = -1;
30. SET @MENSAJE_ERROR = 'ERROR: Ha ocurrido un error
    inesperado a nivel de Base de Datos. Vuelva a intentarlo
    más tarde.';
31. END
32. RETURN 0;
33. END TRY
34. BEGIN CATCH
35. SET @TRANSACCION = -1;
36. SET @CODIGO_ERROR = ERROR_NUMBER();
37. SET @MENSAJE_ERROR = SUBSTRING(ERROR_MESSAGE(), 1, 255);
38. END CATCH
39. END
```

### 4.2.2.3. Update Registro de Cliente.

Nombre del Procedimiento: ActualizarCliente

#### Parámetros de Entrada:

Nombre del parámetro	Tipo de Dato
@cedularuc	INT
@nombres	VARCHAR(50)
@apellidos	VARCHAR(50)
@direccion	VARCHAR(200)
@telefonos	VARCHAR(10)
@email	VARCHAR(50)

#### Parámetros de Salida:

Nombre del parámetro	Tipo de Dato
@TRANSACCION	INT
@CODIGO_ERROR	INT
@MENSAJE_ERROR	VARCHAR(255)

#### Código Fuente:

```
1. CREATE PROCEDURE [dbo].[ActualizarCliente]
2. @cedularuc INT = NULL,
3. @nombres VARCHAR(50) = NULL,
4. @apellidos VARCHAR(50) = NULL,
5. @direccion VARCHAR(200) = NULL,
6. @telefonos VARCHAR(50) = NULL,
7. @email VARCHAR(50) = NULL,
8. @TRANSACCION INT OUTPUT,
9. @CODIGO_ERROR INT OUTPUT,
10. @MENSAJE_ERROR VARCHAR(255) OUTPUT
11. AS
12. BEGIN

13. DECLARE @CONT INT;
14. DECLARE @FILAS INT;
15. DECLARE @TRANS VARCHAR(15);
16.
17. SET NOCOUNT OFF;
18.
19. SET @FILAS = (SELECT COUNT(1) FROM clientes WITH (NOLOCK)
                WHERE cedularuc = @cedularuc);
20. SET @TRANS = 'UPDATE'
21. BEGIN TRY
22. BEGIN TRANSACTION @TRANS
23. UPDATE clientes
24. SET nombres = ISNULL(@nombres,nombres),
25. apellidos = ISNULL(@apellidos,apellidos),
26. direccion = ISNULL(@direccion,direccion),
27. telefonos = ISNULL(@telefonos,telefonos),
28. email = ISNULL(@email,email)
29. WHERE cedularuc = @cedularuc;
30. SET @CONT = @@ROWCOUNT;
31. PRINT 'FILAS ' + CAST(@FILAS AS VARCHAR(10));
32. PRINT 'CONTADOR ' + CAST(@CONT AS VARCHAR(10));

33. IF @CONT = @FILAS AND @CONT >= 1
34. BEGIN
35. SET @TRANSACCION = 1;
36. SET @CODIGO_ERROR = 0;
37. SET @MENSAJE_ERROR = 'OK';
```



```

38. COMMIT TRANSACTION @TRANS;
39. END
40. ELSE IF @CONT = 0 OR @FILAS = 0
41. BEGIN
42. SET @TRANSACCION = 0;
43. SET @CODIGO_ERROR = 0;
44. SET @MENSAJE_ERROR = 'OK';
45. ROLLBACK TRANSACTION @TRANS;
46. END
47. ELSE
48. BEGIN
49. SET @TRANSACCION = -1;
50. SET @CODIGO_ERROR = -1;
51. SET @MENSAJE_ERROR = 'ERROR: Ha ocurrido un error inesperado
a nivel de Base de Datos. Vuelva a intentarlo más tarde.';
52. ROLLBACK TRANSACTION @TRANS;
53. END
54. END TRY
55. BEGIN CATCH
56. SET @TRANSACCION = -1;
57. SET @CODIGO_ERROR = ERROR_NUMBER();
58. SET @MENSAJE_ERROR = SUBSTRING(ERROR_MESSAGE(), 1, 255);
59. ROLLBACK TRANSACTION @TRANS;
60. END CATCH
61. END

```

#### 4.2.2.4. Delete registro de Cliente.

**Nombre del Procedimiento:** EliminarCliente

##### Parámetros de Entrada:

Nombre del parámetro	Tipo de Dato
@cedularuc	INT

##### Parámetros de Salida:

Nombre del parámetro	Tipo de Dato
@TRANSACCION	INT
@CODIGO_ERROR	INT
@MENSAJE_ERROR	VARCHAR(255)

##### Código Fuente:

```

1. CREATE PROCEDURE [dbo].[EliminarCliente]
2. @cedularuc INT = NULL,
3. @TRANSACCION INT OUTPUT,
4. @CODIGO_ERROR INT OUTPUT,
5. @MENSAJE_ERROR VARCHAR(255) OUTPUT
6. AS
7. BEGIN
8. DECLARE @CONT INT;
9. DECLARE @FILAS INT;
10. DECLARE @TRANS VARCHAR(15);
11. DECLARE @DEPEN VARCHAR(15);
12. SET NOCOUNT OFF;
13. SET @FILAS = (SELECT COUNT(1) FROM clientes WITH (NOLOCK)
WHERE cedularuc = @cedularuc);
14. SET @DEPEN = (SELECT COUNT(1) FROM locales WITH (NOLOCK)
WHERE cedularuc = @cedularuc);
15. IF @DEPEN >= 1 BEGIN
16. SET @TRANSACCION = -1;

```

```

17. SET @CODIGO_ERROR = -1;
18. SET @MENSAJE_ERROR = 'ERROR: No es posible eliminar el
registro ya que existe dependencia en LOCALES.';
19. END
20. ELSE BEGIN
21. SET @TRANS = 'DELETE'
22. BEGIN TRY
23. BEGIN TRANSACTION @TRANS
24. DELETE FROM CLIENTES
25. WHERE cedularuc= @cedularuc;
26. SET @CONT = @@ROWCOUNT;
27. PRINT 'FILAS ' + CAST(@FILAS AS VARCHAR(10));
28. PRINT 'CONTADOR ' + CAST(@CONT AS VARCHAR(10));

29. IF @CONT = @FILAS AND @CONT >= 1
30. BEGIN
31. SET @TRANSACCION = 1;
32. SET @CODIGO_ERROR = 0;
33. SET @MENSAJE_ERROR = 'OK';
34. COMMIT TRANSACTION @TRANS;
35. END
36. ELSE IF @CONT = 0 OR @FILAS = 0
37. BEGIN
38. SET @TRANSACCION = 0;
39. SET @CODIGO_ERROR = 0;
40. SET @MENSAJE_ERROR = 'OK';
41. ROLLBACK TRANSACTION @TRANS;
42. END
43. ELSE
44. BEGIN
45. SET @TRANSACCION = -1;
46. SET @CODIGO_ERROR = -1;
47. SET @MENSAJE_ERROR = 'ERROR: Ha ocurrido un error
inesperado a nivel de Base de Datos. Vuelva a intentarlo
más tarde.';
48. ROLLBACK TRANSACTION @TRANS;
49. END
50. END TRY
51. BEGIN CATCH
52. SET @TRANSACCION = -1;
53. SET @CODIGO_ERROR = ERROR_NUMBER();
54. SET @MENSAJE_ERROR = SUBSTRING(ERROR_MESSAGE(), 1, 255);
55. ROLLBACK TRANSACTION @TRANS;
56. END CATCH
57. END
58. END

```

### 4.2.3. Creación y configuración del Proyecto en Capas en Visual Studio 2012 Xpress.

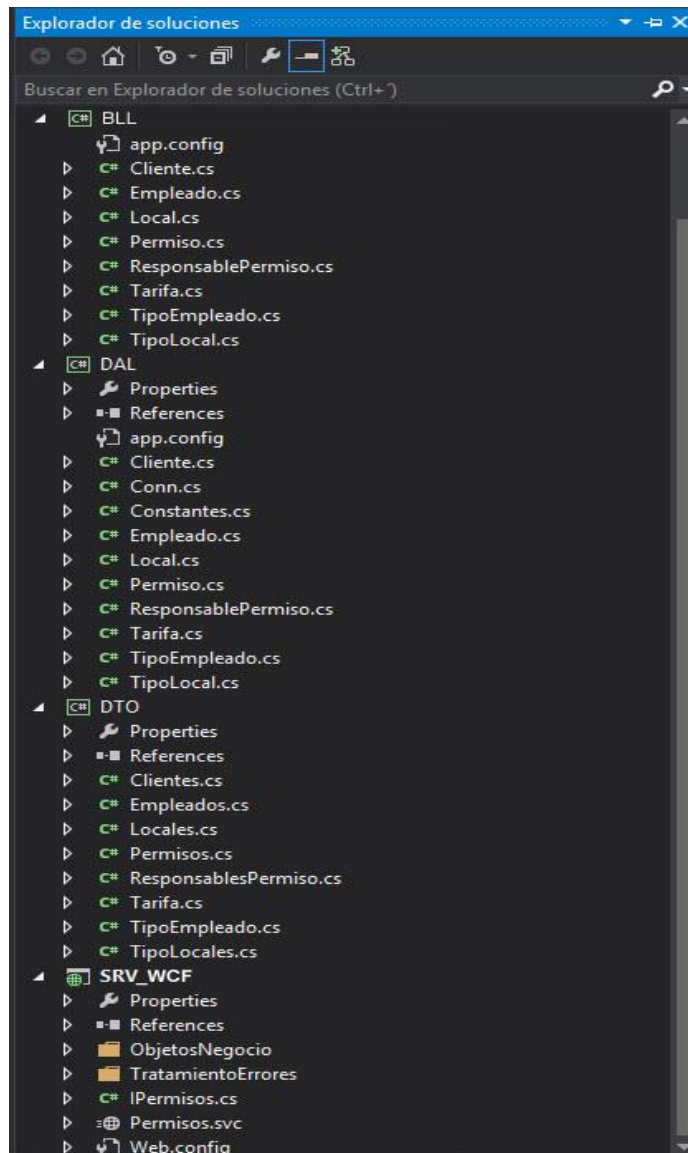


Figura 4- 15 Configuración de Capas de las clases del proyecto.

(Fuente: Propia., 2017)

La configuración para el proyecto se va usar el patrón de diseño N CAPAS en el cual usaremos 4 capas las cuales corresponden a: DTO, DAL, BLL, SERVICIOS WCF.

### 4.2.3.1 DTO (Data Transfer Object).

Esta capa va a contener las clases de los datos que son transferidos por cada llamada, pero que son entregados en una sola invocación, en la pantalla de abajo se detalla la configuración necesaria para esta capa:

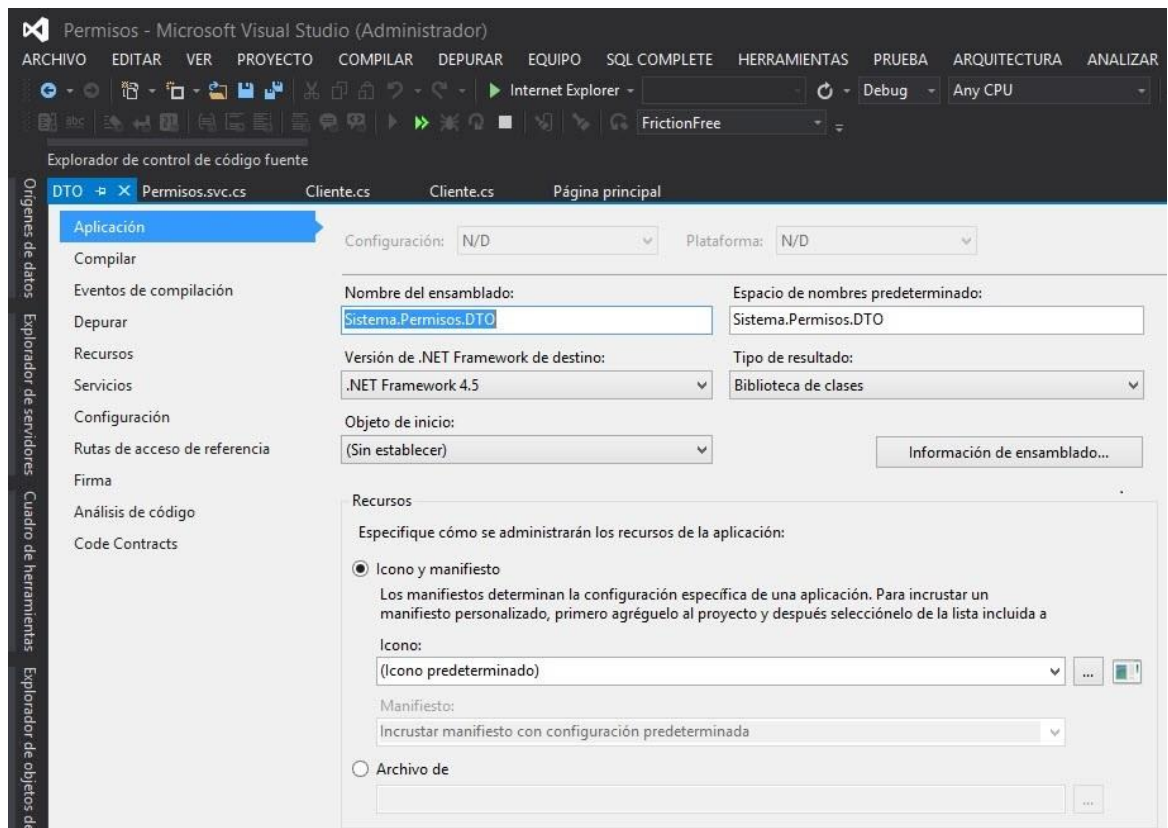


Figura 4- 16 Configuración de la capa de DTO para las clases del proyecto.

(Fuente: Propia., 2017)

### 4.2.3.2 DAL (Data Access Layer).

Esta capa va a contener las clases encargadas de interconectar la capa de DTO con la capa de Acceso a Datos.

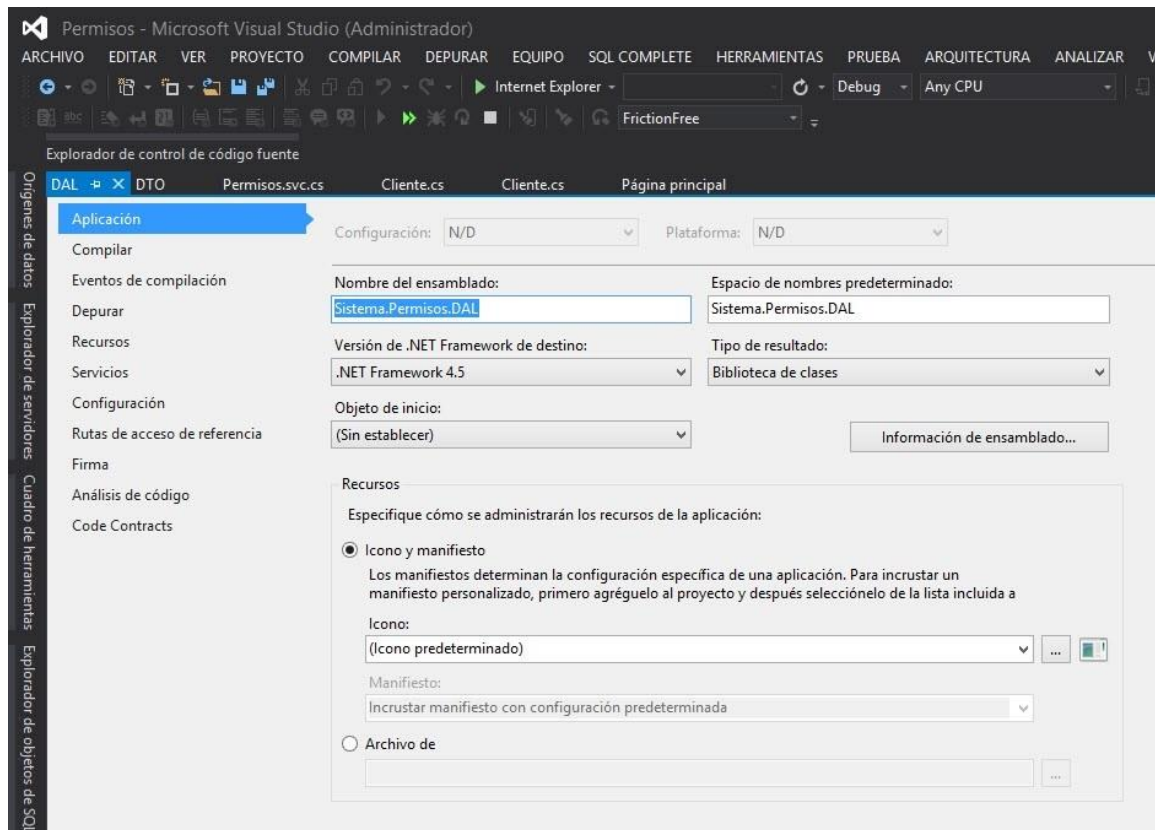


Figura 4- 17 Configuración de la capa DAL para las clases del proyecto.

(Fuente: Propia., 2017)

#### 4.2.3.3 BLL (Business Logic Layer).

Esta capa contiene las clases encargadas de gestionar las funcionalidades relacionadas con la lógica de negocio necesaria para orquestar los diferentes procesos del sistema:

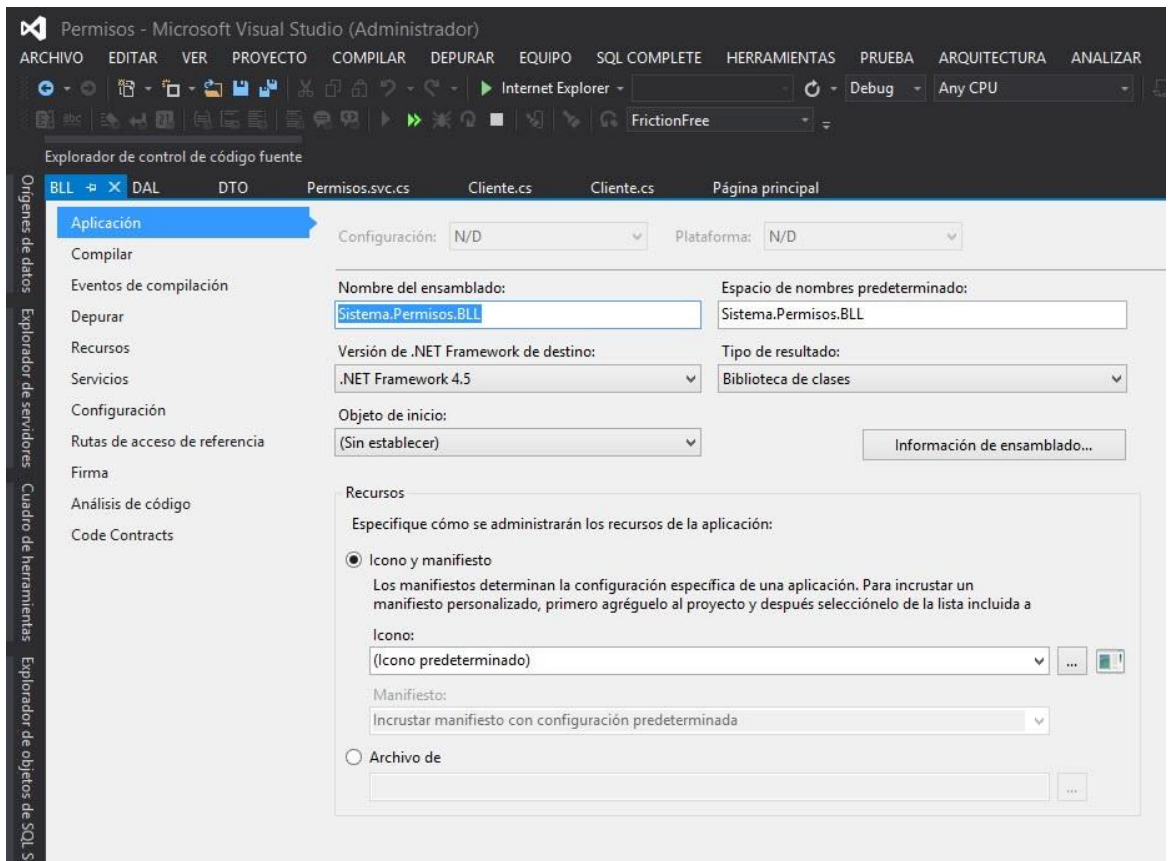


Figura 4- 18 Configuración de la capa de BLL para las clases del proyecto.

(Fuente: Propia., 2017)

#### 4.2.3.4 Servicio WCF (Windows Communication Foundation).

En esta capa se definen los diferentes conceptos y características necesarias para la definición de las interfaces WCF:

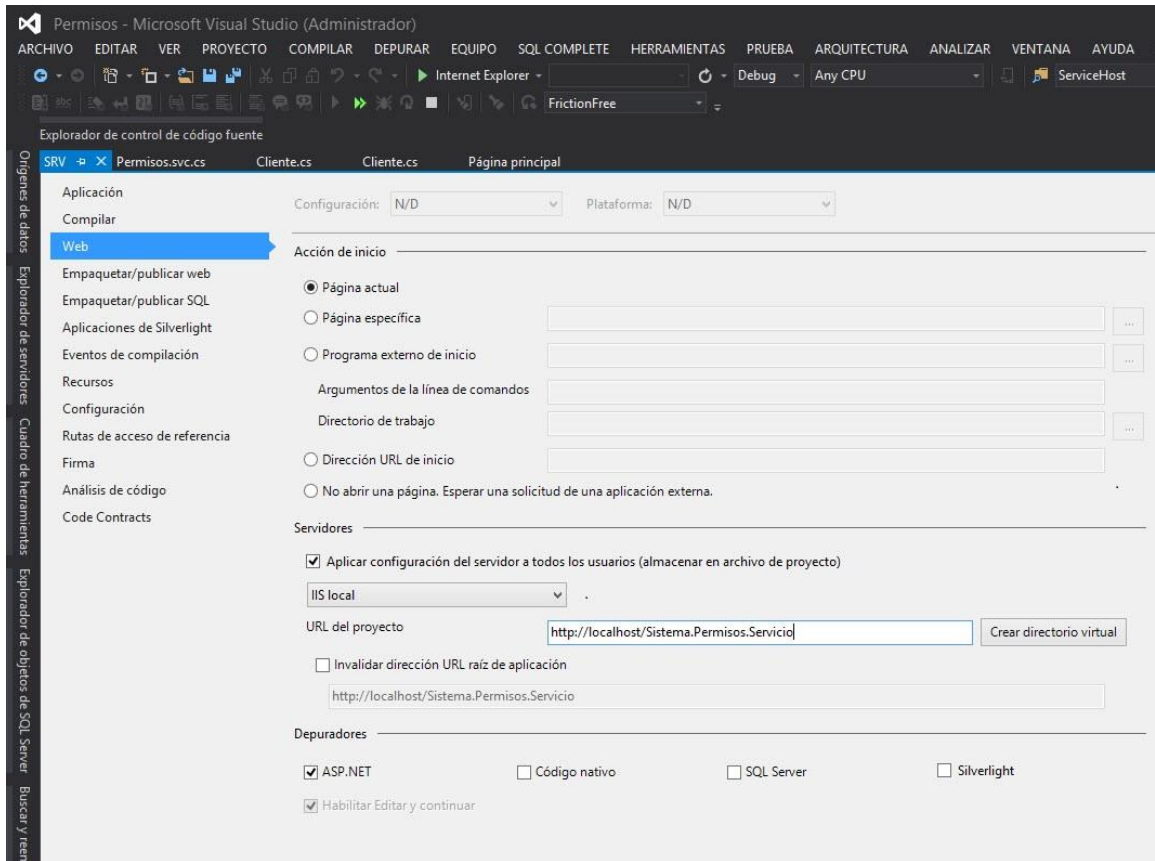


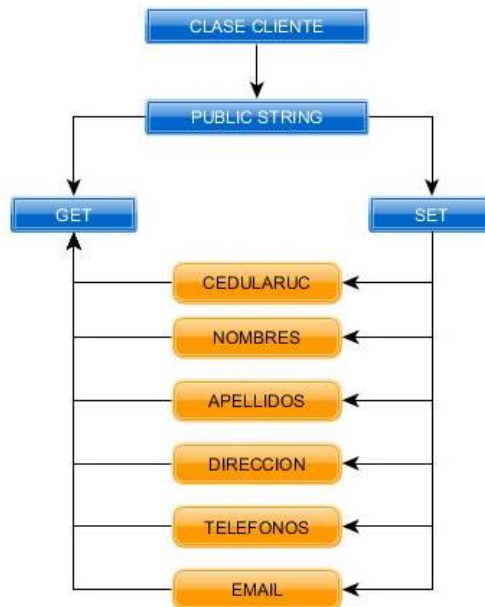
Figura 4- 19 Configuración de la capa de Servicios WCF del proyecto.

(Fuente: Propia., 2017)

### 4.3 Implementación de las Capas del Sistema

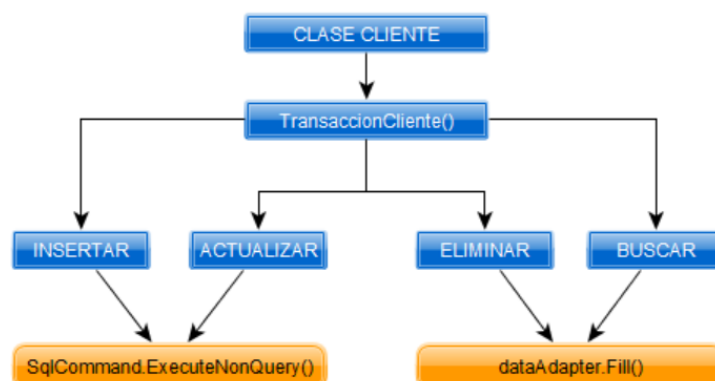
#### 4.3.1. DTO (Data Transfer Objects).

Definir la CLASE CLIENTE la cual encapsula la información mediante los métodos accesoros GET, SET de cada propiedad del campo de la entidad CLIENTE.



#### 4.3.2. DAL (Data Access Layer).

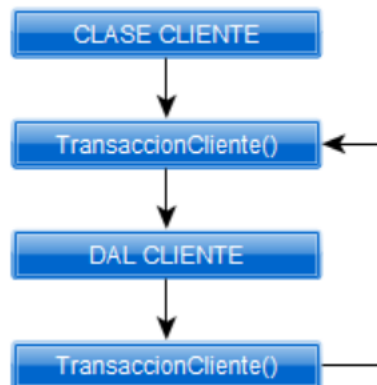
Crear la CLASE CLIENTE en la cual se define el método publico TransaccionCliente() donde vamos a controlar mediante parámetros si la operación requerida por el cliente es INSERTAR, MODIFICAR, ELIMINAR O BUSCAR los registros:





### 4.3.3. BLL (Business Logic Layer).

En esta capa definir la clase: CLIENTE donde se hace la llamada hacia el método publico definido en la capa DAL CLIENTE llamada de la misma forma TransaccionCliente() la cual retornara resultado exitoso o erróneo dependiendo del estado de la ejecución que hayan generado los parámetros de entrada.



### 4.3.4. WCF (Clase Cliente).

#### 4.3.4.1. Data Contract

```
1.     [DataContract]
2.     public class Cliente
3.     {
4.         #region Declaración de Variables
5.         private DTO.Clientes[] listaClientes;
6.         private MensajeSistema mensajeSistema;
7.         #endregion
8.
9.         #region Propiedades
10.        [DataMember(Order = 0)]
11.        public DTO.Clientes[] ListaClientes
12.        {
13.            get
14.            {
15.                return listaClientes;
16.            }
17.            set
18.            {
19.                listaClientes = value;
20.            }
21.        }
22.        [DataMember(Order = 1)]
23.        public MensajeSistema MensajeSistema
24.        {
25.            get
26.            {
27.                return mensajeSistema;
28.            }
29.            set
30.            {
31.                mensajeSistema = value;
32.            }
33.        }
34.        #endregion
35.    }
```

#### 4.3.4.2. Operation Contract

```
1. #region TransaccionCliente
2. [OperationContract(IsTerminating = false,
3. IsInitiating = true,
4. IsOneWay = false,
5. AsyncPattern = false,
6. Action =
7. "urn:Sistema.Permisos.Servicio.IPermisos.TransaccionCliente",
8. ReplyAction =
9. "urn:Sistema.Permisos.Servicio.IPermisos.TransaccionCliente",
10. ProtectionLevel = ProtectionLevel.None)]
11. Sistema.Permisos.Servicio.Cliente TransaccionCliente(
12. string @cedularuc
13. ,string @nombres
14. ,string @apellidos
15. ,string @direccion
16. ,string @telefonos
17. ,string @email
18. ,string @operacion
19. );
```

#### 4.3.4.3. Service Contract

```
1. [ServiceContract(Namespace = "urn:Sistema.Permisos.Servicio",
2. Name = "Permisos",
3. SessionMode = SessionMode.Allowed,
4. ProtectionLevel = ProtectionLevel.None)]
5.
6. public interface IPermisos
7. {
8.     #region Definición de Contrato
9.
10.     #region TransaccionCliente
11.     [OperationContract(IsTerminating = false, IsInitiating
12. = true, IsOneWay = false, AsyncPattern = false, Action =
13. "urn:Sistema.Permisos.Servicio.IPermisos.TransaccionCliente",
14. ReplyAction =
15. "urn:Sistema.Permisos.Servicio.IPermisos.TransaccionCliente",
16. ProtectionLevel = ProtectionLevel.None)]
17. Sistema.Permisos.Servicio.Cliente TransaccionCliente(
18. string
19. @cedularuc
20. ,string @nombres
21. ,string
22. @apellidos
23. ,string
24. @direccion
25. ,string
26. @telefonos
27. ,string @email
28. ,string
29. @operacion
30. );
```

#### Configuración del Servicio:

- a) Crear el Servicio WCF en la pestaña aplicación se hace referencia al nombre con el que se va a generar el ensamblado, la versión del framework con el que se va a trabajar en este caso será : .Net Framework 4.5 , el espacio de nombres y la salida la cual será Biblioteca de Clases:

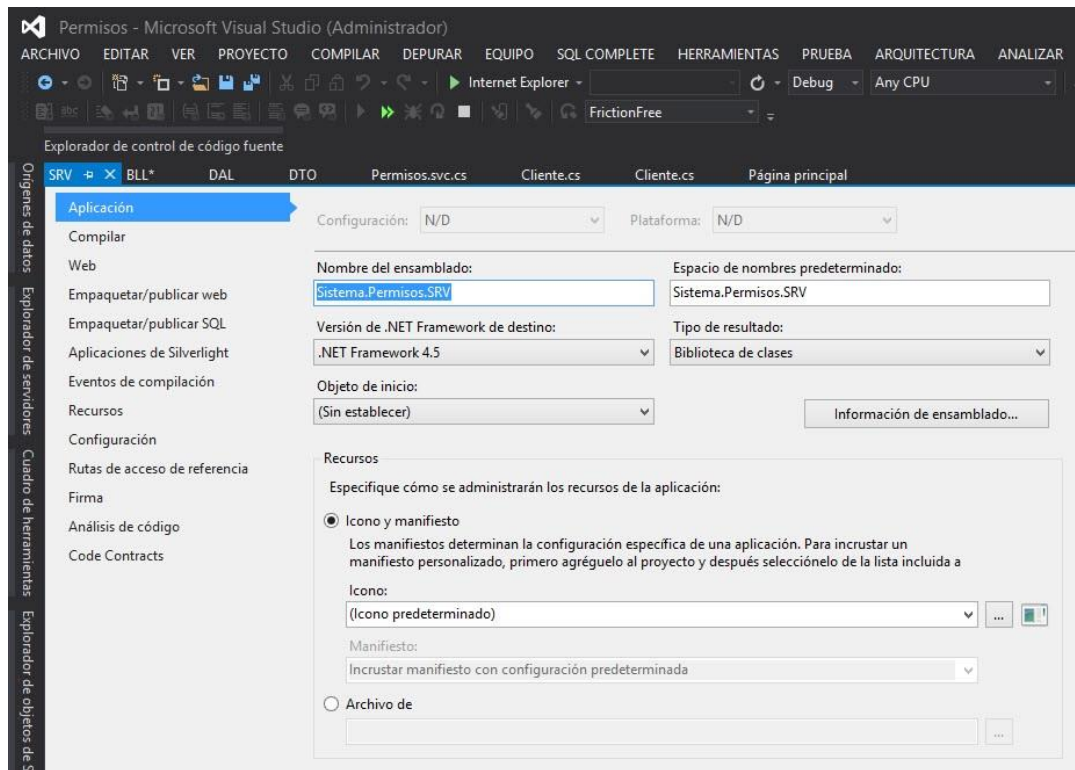


Figura 4- 20 Configuración del Servicio WCF: Aplicación.

(Fuente: Propia., 2017)

- b) En la pestaña Web se especifica la acción de inicio que será: Pagina Actual, en servidores aplicar la configuración del servidor haciendo referencia a Internet Information Server con la respectiva URL donde se van a alojar los servicios, en depuradores elegir “ASP.NET”.

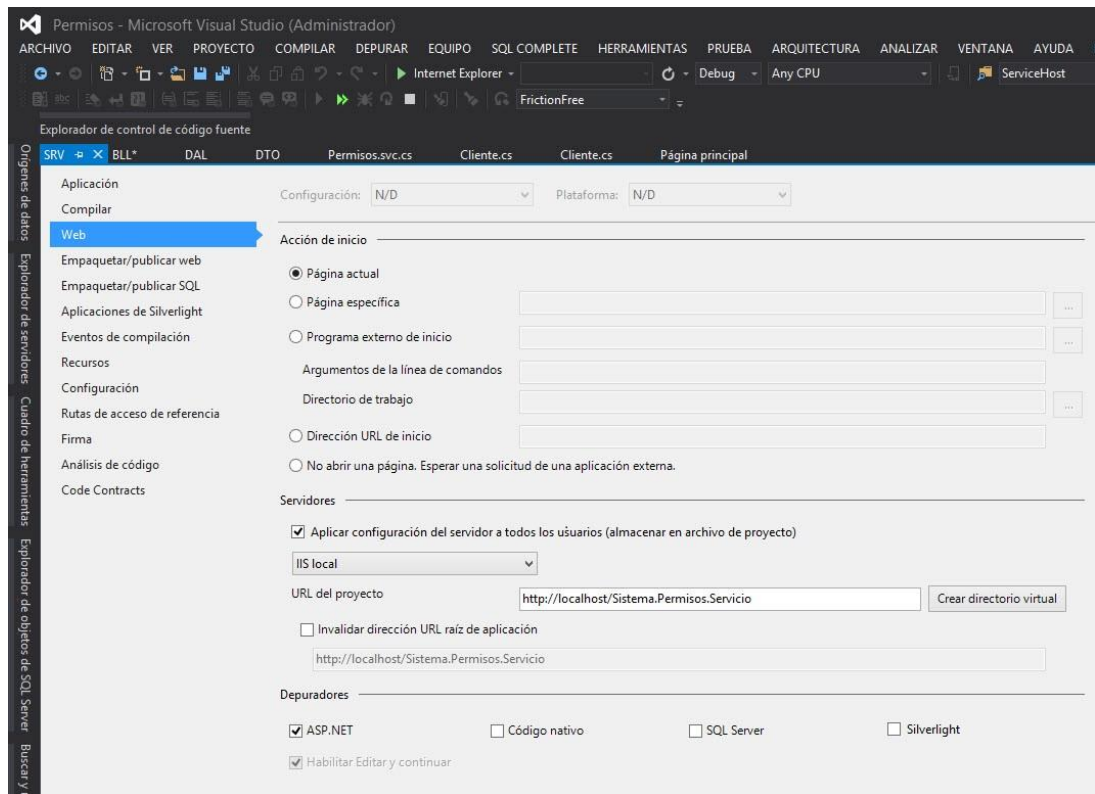
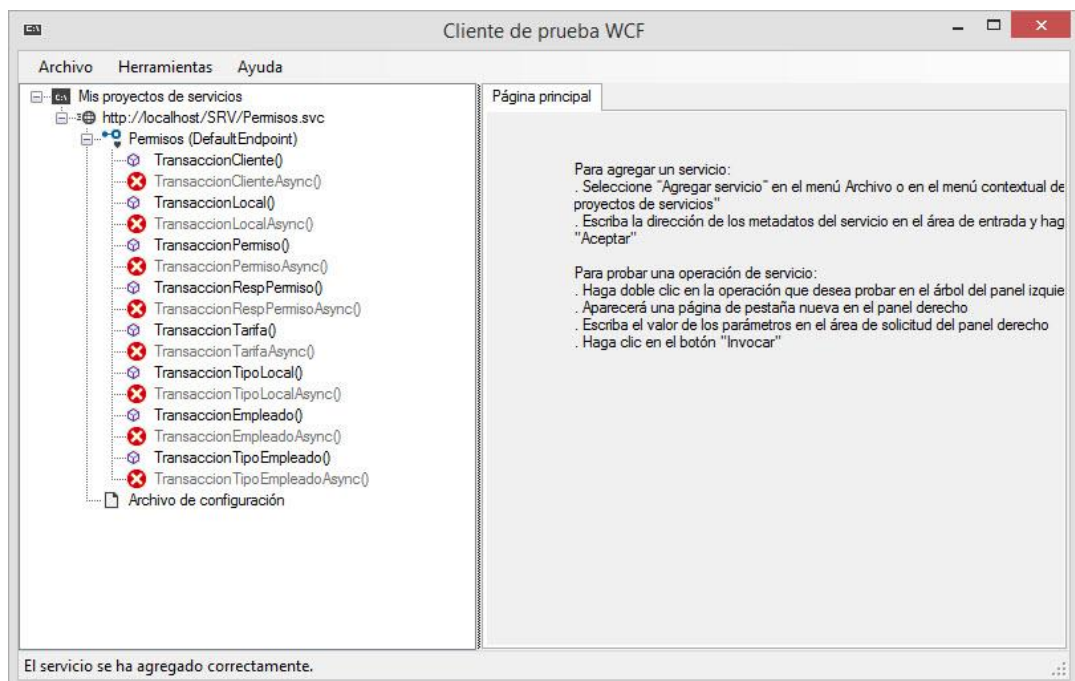
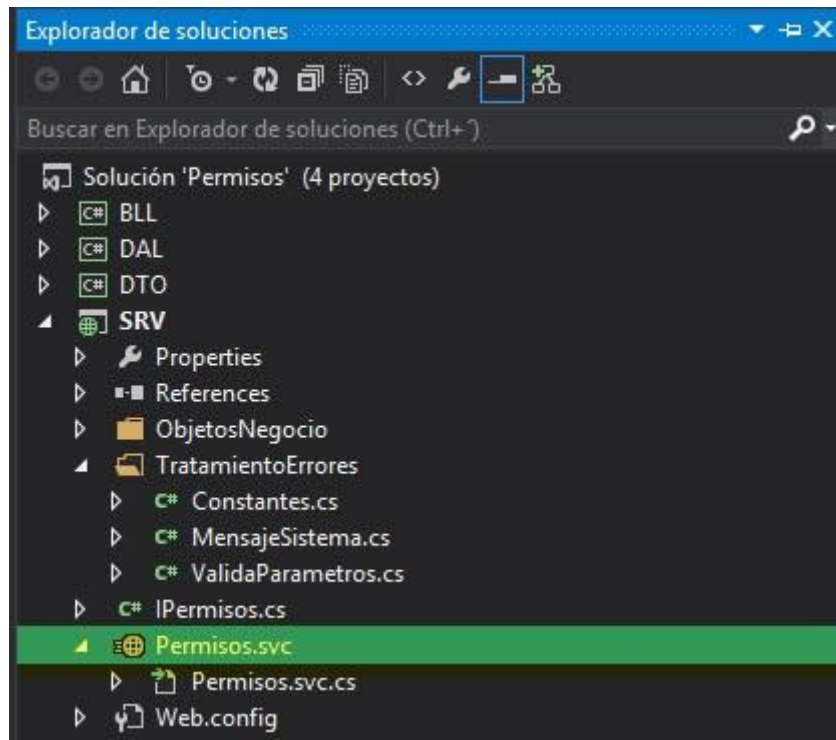


Figura 4- 21 Configuración del Servicio WCF: Web.

(Fuente: Propia., 2017)

- c) Para el servicio WCF: **Permisos.svc** configurar un cliente de prueba con el que se va a publicar y alojar el servicio para poder generar la dirección WSDL para que el servicio pueda ser consumido desde cualquier aplicación.



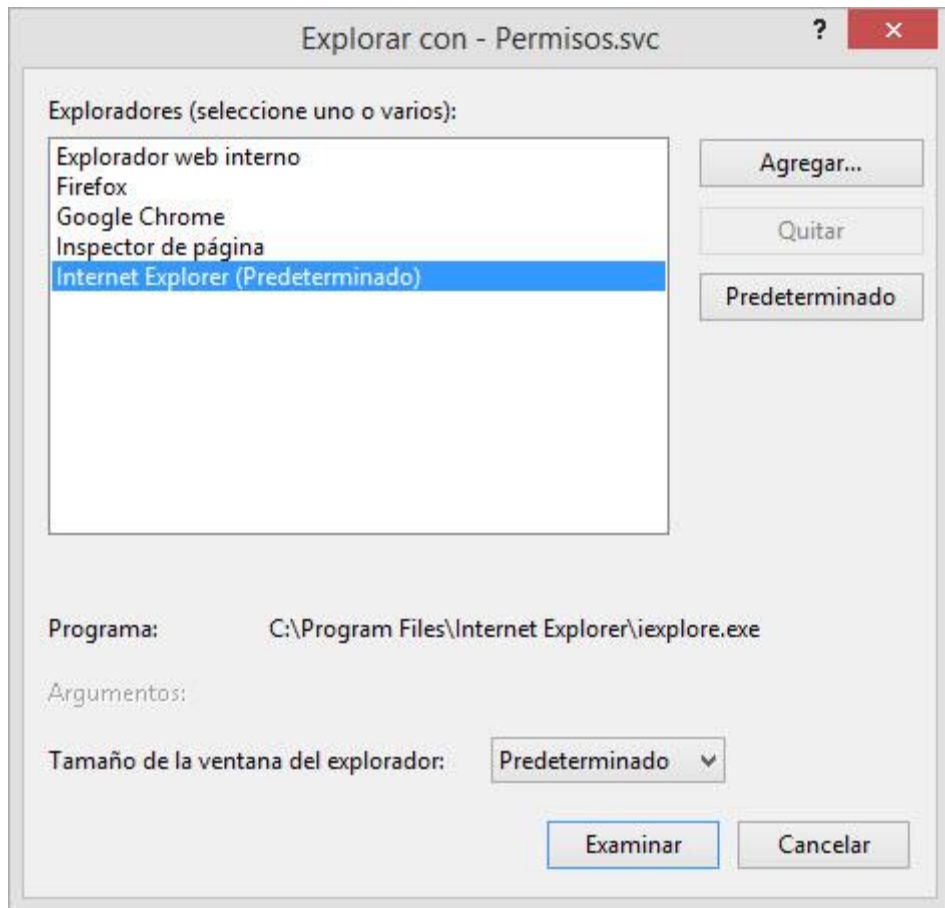


Figura 4- 22 Configuración del Servicio WCF: Preparar Servicio WCF para Alojamiento.

(Fuente: Propia., 2017)

#### 4.4. Publicación y Alojamiento

- a) Para la publicación del Servicio WCF se elige las opciones de publicación web en las cuales se definen: Perfil, Conexión, Configuración, Vista Previa, tal como se detalla en las figuras siguientes:

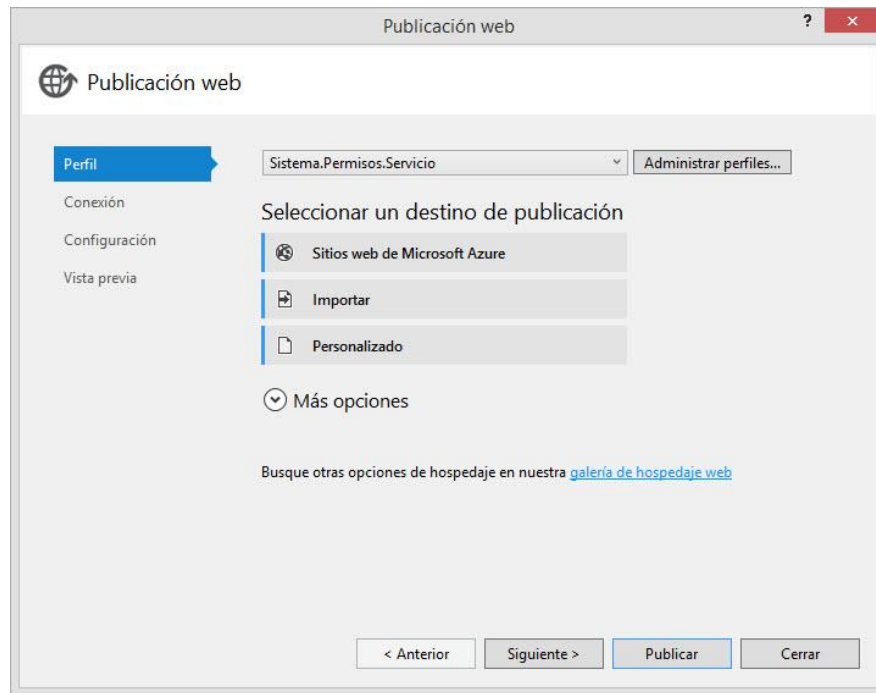


Figura 4- 23 Publicación y Alojamiento del Servicio WCF: Perfil.

(Fuente: Propia., 2017)

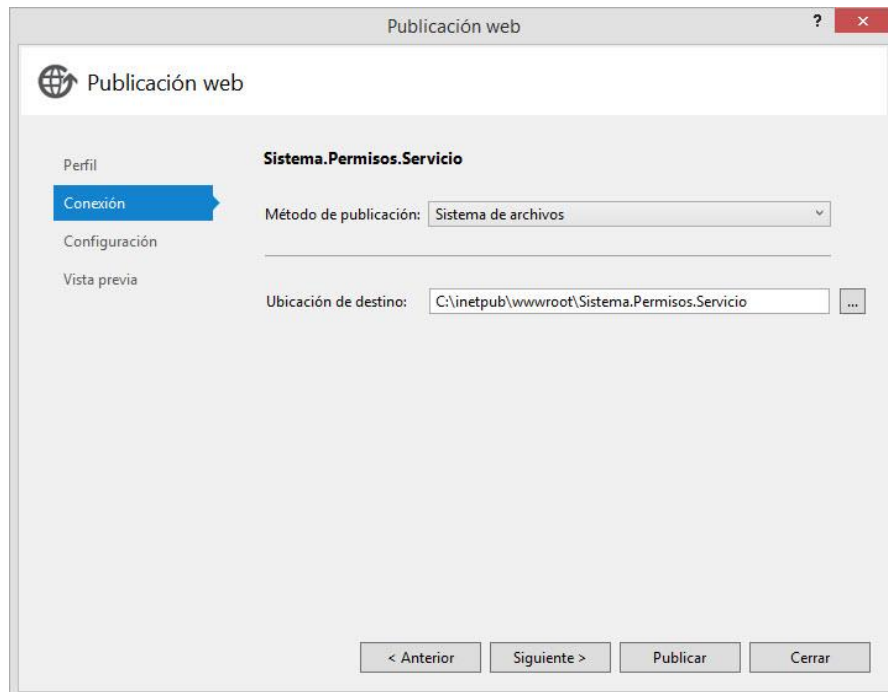


Figura 4- 24 Publicación y Alojamiento del Servicio WCF: Conexión.

(Fuente: Propia., 2017)

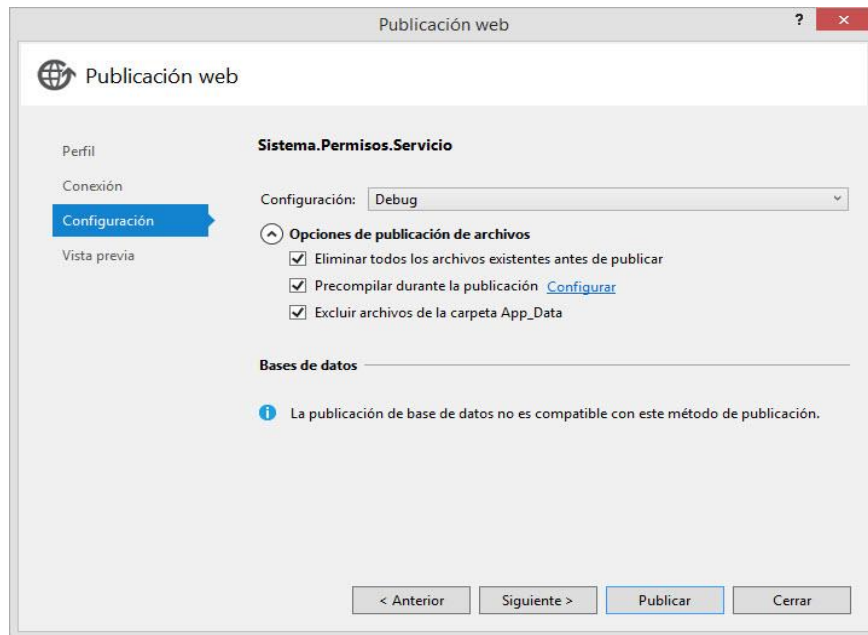


Figura 4- 25 Publicación y Alojamiento del Servicio WCF: Configuración.

(Fuente: Propia., 2017)

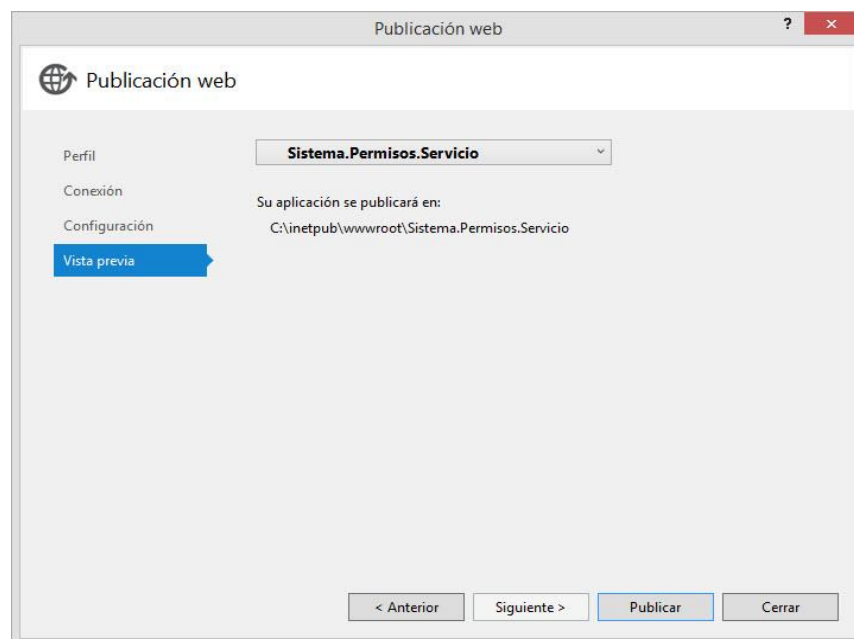


Figura 4- 26 Publicación y Alojamiento del Servicio WCF: Vista Previa.

(Fuente: Propia., 2017)

- b) Al terminar la configuración elegir Publicar y se generara el directorio en la ruta especificada con el ensamblado del Servicio WCF en el servidor web del IIS.



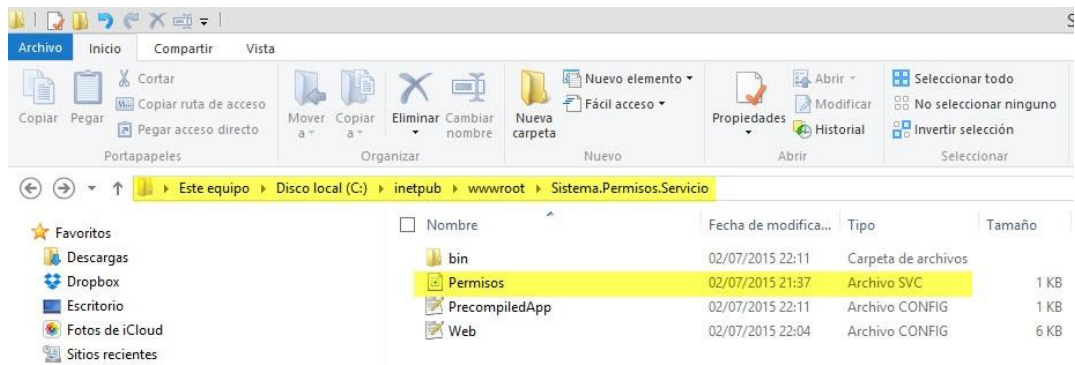


Figura 4- 27 Publicación y Alojamiento del Servicio WCF: Verificación física.

(Fuente: Propia., 2017)

- c) En el Administrador de IIS elegir el servicio publicado para verificar la generación de la entrada WSDL del servicio WCF, pulsar la opción que se encuentra en la esquina superior derecha: “Examinar: 80 (http)”.

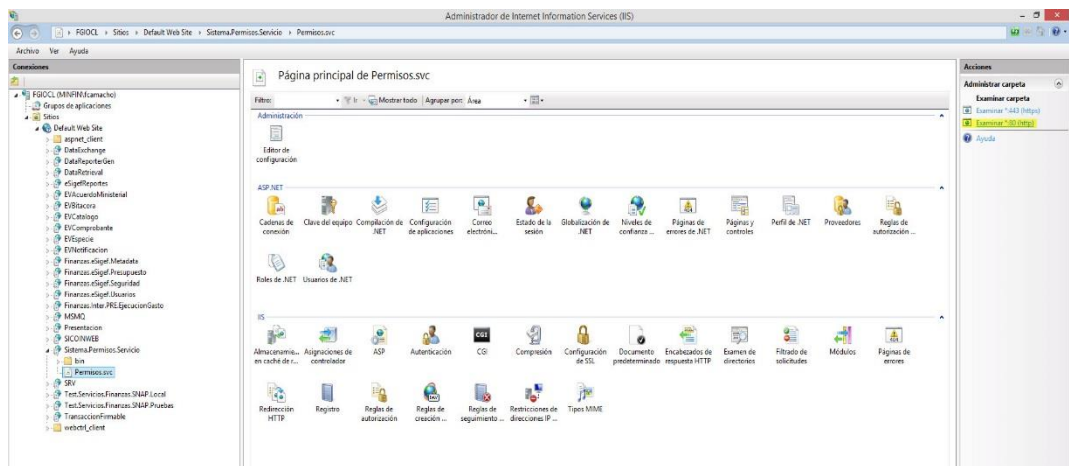


Figura 4- 28 Publicación y Alojamiento del Servicio WCF: Servicio WCF – Entrada WSDL. (Fuente: Propia., 2017)

d) Verificar la creación del servicio y la entrada WSDL respectivamente.

Creó un servicio.

Para probarlo, deberá crear un cliente y usarlo para llamar al servicio. Para ello, puede usar la herramienta svcutil.exe en la línea de comandos con la siguiente sintaxis:

```
svcutil.exe http://localhost/SRV/Permisos.svc?wsdl
```

También puede tener acceso a la descripción del servicio como un solo archivo:

```
http://localhost/SRV/Permisos.svc?singleWsdl
```

Esto generará un archivo de configuración y un archivo de código que contiene la clase de cliente. Agregue los dos archivos a la aplicación cliente y use la clase de cliente generada para llamar al servicio. Por ejemplo:

```
C#  
class Test  
{  
    static void Main()  
    {  
        PermisosClient client = new PermisosClient();  
  
        // Use la variable 'client' para llamar a operaciones en el servicio.  
  
        // Cierre siempre el cliente.  
        client.Close();  
    }  
}
```

Visual Basic

```
Class Test  
Shared Sub Main()  
    Dim client As PermisosClient = New PermisosClient()  
    ' Use la variable 'client' para llamar a operaciones en el servicio.  
  
    ' Cierre siempre el cliente.  
    client.Close()  
End Sub  
End Class
```

Figura 4- 29 Publicación y Alojamiento del Servicio WCF:  
WCF – Entrada WSDL. (Fuente: Propia., 2017)

```
<?xml version="1.0" encoding="UTF-8"?>  
<wsi:definitions xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:mexc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsa10="http://www.w3.org/2005/08/addressing" xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wsdl-1.0-base" xmlns:wsse="http://schemas.xmlsoap.org/ws/2004/09/secure" xmlns:wsm="http://schemas.xmlsoap.org/ws/2004/09/mex" xmlns:wsi="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:Sistema.Permisos.Servicio" name="Permisos">  
  <wsi:schema targetNamespace="urn:Sistema.Permisos.Servicio/Imports">  
    <xsd:import namespace="urn:Sistema.Permisos.Servicio" schemaLocation="http://localhost/SRV/Permisos.svc?xsd=xsd1"/>  
    <xsd:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" schemaLocation="http://localhost/SRV/Permisos.svc?xsd=xsd1"/>  
    <xsd:import namespace="http://schemas.datacontract.org/2004/07/Sistema.Permisos.DTO" schemaLocation="http://localhost/SRV/Permisos.svc?xsd=xsd3"/>  
  </wsi:schema>  
</wsi:definitions>  
<wsi:types>  
  <wsi:message name="Permisos_TransaccionCliente_InputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionCliente"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionCliente_OutputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionClienteResponse"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionLocal_InputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionLocal"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionLocal_OutputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionLocalResponse"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionPermiso_InputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionPermiso"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionPermiso_OutputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionPermisoResponse"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionRespPermiso_InputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionRespPermiso"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionRespPermiso_OutputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionRespPermisoResponse"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionTarifa_InputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionTarifa"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionTarifa_OutputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionTarifaResponse"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionTipoLocal_InputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionTipoLocal"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionTipoLocal_OutputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionTipoLocalResponse"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionEmpleado_InputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionEmpleado"/>  
  </wsi:message>  
  <wsi:message name="Permisos_TransaccionEmpleado_OutputMessage">  
    <wsi:part name="parameters" element="tns:TransaccionEmpleadoResponse"/>  
  </wsi:message>  
</wsi:types>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:m:ac="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsop="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsa10="http://www.w3.org/2005/08/addressing" xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wsseuc" xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:Sistema.Permisos.Servicio" name="Permisos">
  <wsdl:types>
    <xsd:schema targetNamespace="urn:Sistema.Permisos.Servicio" xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
      <xsd:import namespace="http://schemas.datacontract.org/2004/07/Sistema.Permisos.Servicio"/>
      <xsd:element name="TransaccionCliente">
        <xsd:complexType base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
          <xsd:sequence base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
            <xsd:element name="cedularuc" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="nombres" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="apellidos" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="direccion" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="telefonos" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="email" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="operacion" type="xsd:string" nillable="true" minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="TransaccionClienteResponse">
        <xsd:complexType base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
          <xsd:sequence base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
            <xsd:element name="TransaccionClienteResult" type="q1:Cliente" nillable="true" minOccurs="0" maxOccurs="1" xmlns:q1="http://schemas.datacontract.org/2004/07/Sistema.Permisos.Servicio"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="TransaccionLocal">
        <xsd:complexType base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
          <xsd:sequence base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
            <xsd:element name="id_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="cedularuc_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="nombre_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="direc_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="telef_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="anio_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="tipo_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="cap_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="canton_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="parroquia_local" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="operacion" type="xsd:string" nillable="true" minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="TransaccionLocalResponse">
        <xsd:complexType base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
          <xsd:sequence base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
            <xsd:element name="TransaccionLocalResult" type="q2:Local" nillable="true" minOccurs="0" maxOccurs="1" xmlns:q2="http://schemas.datacontract.org/2004/07/Sistema.Permisos.Servicio"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="TransaccionPermiso">
        <xsd:complexType base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
          <xsd:sequence base="xsd:string" minOccurs="1" maxOccurs="1" nillable="true">
            <xsd:element name="id_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="cedularuc_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="nombre_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="direc_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="telef_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="anio_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="tipo_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="cap_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="canton_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="parroquia_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
            <xsd:element name="operacion_permiso" type="xsd:string" nillable="true" minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:binding name="PermisosSoap" type="soap:Fault" soap:binding="soap:Fault" style="fault" use="literal"/>
  <wsdl:binding name="PermisosHttp" type="PermisosSoap" soap:binding="PermisosSoap" style="fault" use="literal"/>
  <wsdl:service name="Permisos" base="PermisosHttp" soap:binding="PermisosSoap" style="fault" use="literal"/>
  <wsdl:operation name="TransaccionCliente" type="PermisosSoap" soap:binding="PermisosSoap" style="fault" use="literal"/>
  <wsdl:operation name="TransaccionLocal" type="PermisosSoap" soap:binding="PermisosSoap" style="fault" use="literal"/>
  <wsdl:operation name="TransaccionPermiso" type="PermisosSoap" soap:binding="PermisosSoap" style="fault" use="literal"/>

```

Figura 4- 30 Publicación y Alojamiento del Servicio WCF: Descubrir entrada WSDL.  
(Fuente: Propia., 2017)

#### 4.5. Etapa de Pruebas

- a) Para esta etapa se usara SOAP UI en el cual se puede consumir los servicios WCF expuestos en la entrada WSDL, por lo que se copia la entrada WSDL generada anteriormente.

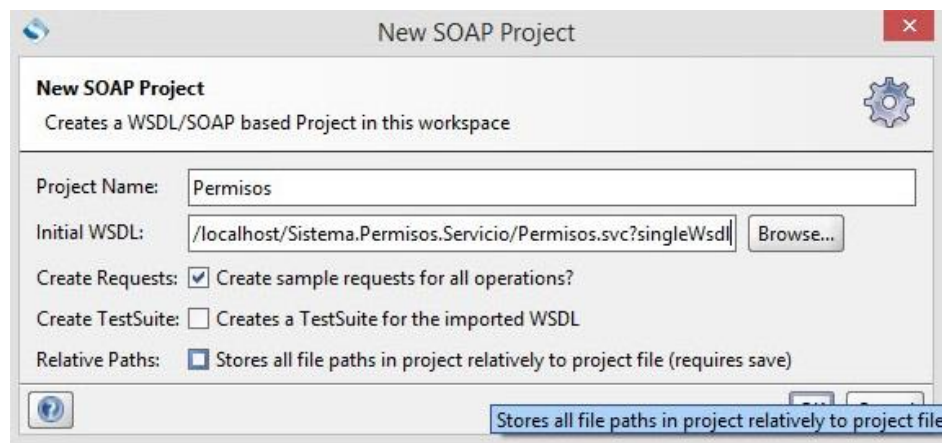


Figura 4- 31 Etapa de Pruebas: Configurar End Point con SOAP UI para consumir servicio WCF. (Fuente: Propia., 2017)

- b) Se crea el EndPoint con los request que apuntaran a las transacciones definidas en la interfaz del servicio WCF:

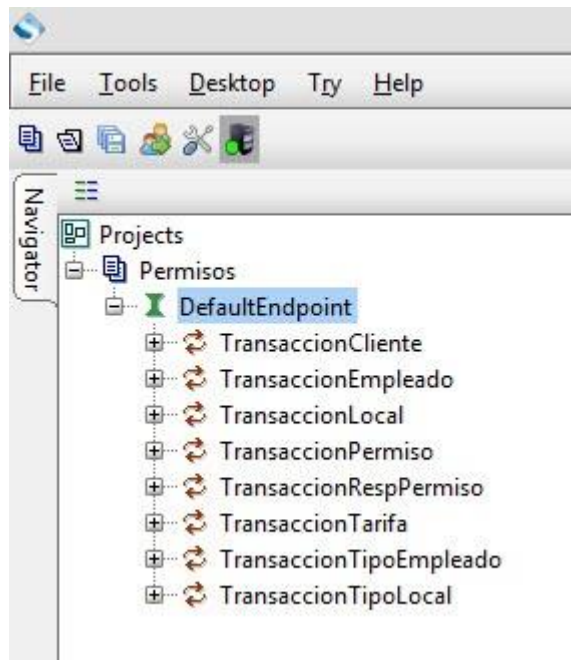


Figura 4- 32 Etapa de Pruebas: End Point con las Transacciones de los servicios WCF. (Fuente: Propia., 2017)

- c) Para probar elegir el request referente a la transacción cliente y llenar la información del cliente.

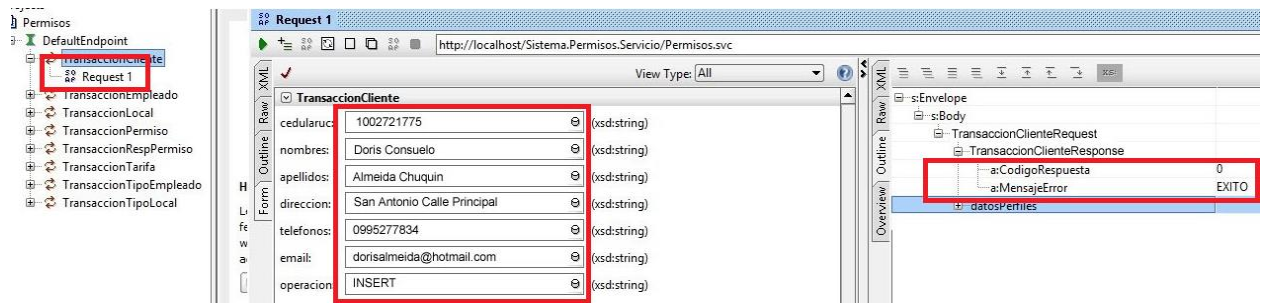


Figura 4- 33 Etapa de Pruebas: Ejecución de request servicio WCF Ingresar Cliente. (Fuente: Propia., 2017)

- d) Comprobar afectación en la base de datos CLIENTE después de la ejecución del servicio

select \*from clientes

100 %

Results Messages

	cedularuc	nombres	apellidos	direccion	telefonos	email
1	1002521415	Christian Orlando	Lara Castillo	Ibarra La Florida	0998993726	christianlara@hotmail.es
2	1002521416	Jorge Anibal	Realpe Rosero	Los Ceibos y Eucaliptos	0994265017	jorgerealpe@hotmail.com
3	1002721775	Doris Consuelo	Almeida Chuquin	San Antonio Calle Principal	0995277834	dorisalmeida@hotmail.com

Figura 4- 34 Etapa de Pruebas: Verificación de afectación tabla clientes.

(Fuente: Propia., 2017)



---

# CONCLUSIONES RECOMENDACIONES

---

## CAPITULO V

### 5. CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Conclusiones

- a) Con la implementación de esta tecnología (Windows Communication Foundation), la flexibilidad del sistema permite que sea usado desde cualquier Cliente implementado en cualquier lenguaje de programación que puede incluir herramientas libres, debido a que la lógica de negocio, sus métodos que la exponen se encuentran de manera independiente y expuesta en un servicio WCF.
- b) Gracias al sistema se lleva un control adecuado de la información y los entes que están inmersos en el proceso de generación de permisos de funcionamiento como lo son empleados y administradores.
- c) El sistema facilita a los administradores y empleados de la Institución las actividades de control relacionados a los permisos de funcionamiento mediante reportes que les ayudan a tomar decisiones a corto plazo, basados en la información recaudatoria y estado de los locales comerciales.
- d) Gracias a las características del sistema, se organiza de manera eficaz la información que antes se llevaba de forma física permitiendo tener un repositorio digital confiable y ágil a la hora de administrar el sistema, inclusive desde cualquier lugar donde exista internet.
- e) El sistema permite una fácil administración de la información sin afectar su integridad y credibilidad.

- f) Los usuarios del sistema aplican tiempos mínimos en la generación de información hacia el cliente, mejorando su productividad frente a los procesos realizados en la institución lo que se traduce en una mejor atención y satisfacción del cliente.
  
- g) La tecnología y la implementación del sistema va de la mano de la infraestructura del hardware por lo que si se requiere mayor disponibilidad del sistema es necesaria mayor inversión económica acorde a como escale el sistema.
  
- h) Windows Communication Foundation (WCF), permite el uso de características con un coste de aprendizaje mínimo es altamente aplicable para proyectos con tendencia a funcionalidades amigables al usuario y que requieren de un framework, sin que este necesite recargar las peticiones que se ejecuten proporcionando mensajería interoperable, segura, confiable y por transacciones. en el desarrollo, que ofrezca componentes.



## 5.2. Recomendaciones

- a) Para la buena operatividad del sistema se recomienda infraestructura de hardware que soporte su arquitectura, para este caso y para abaratar costos es recomendable usar un solo servidor para la instalación de los servicios para el manejo de las capas de lógica de negocio y servidor de base de datos, respecto al sistema el mismo puede ser alojado en cualquier hosting que use .net.
- b) Se recomienda a la Institución documentar la información de los procesos manuales que se lleven a cabo a manera de guías o instructivos que faciliten futuras implementaciones o soluciones informáticas que las pueden cubrir.
- c) Es necesario un seguimiento del sistema informático, documentando los errores que se generen a lo largo de su funcionamiento ya que esto facilitara al equipo encargado del mismo la detección de fallos.
- d) Para el mantenimiento del sistema Se recomienda guiarse en el uso correcto de los lineamientos que establece este documento que relaciona la metodología de desarrollo de software: RUP, enfatizando en el levantamiento de los casos de uso, debido a que este artefacto es la base para elaborar el corazón del sistema.

### 5.3. Análisis de Impactos

Para el análisis de impactos se toma en cuenta varios factores en los cuales se cuantifica de acuerdo a la relación de los diferentes aspectos funcionales del sistema con el usuario, como son:

- Proceso de generación de permisos de funcionamiento.
- Registro y actualización de Información.
- Generación de informes y reportes.

#### 5.3.1. Proceso de Generación de permisos de Funcionamiento

Para el análisis del proceso de entrega del permiso de funcionamiento se toma en cuenta los factores: Búsqueda de Información (Cliente, Local Comercial, Informe de Inspección, Permisos Anteriores), Generación de Permiso (manual) de donde se obtiene la siguiente información:

Actividades	Tiempo por Actividad (Minutos)	Tiempo Total por Día (horas)
Búsqueda de Información (Cliente, Local Comercial, Informe de Inspección, Permisos Anteriores)	10 min.	5,3 h
Generación de Permiso (manual)	5 min.	2,7 h

Tabla 3 - 3 Tiempo de Entrega de Permisos (manual). (Fuente: Propia., 2017)

#### 5.3.2. Registro y Actualización de Información

Para el análisis del proceso de Registro y actualización de información se toma en cuenta los factores como son registro actualización de Empleados, Clientes, Locales Comerciales, Informe de Inspección

Actividades	Tiempo por Actividad (Minutos)	Tiempo Total por Día (horas)
Registro y enrolamiento de Empleados		

	60 min por empleado.	4 h
Registro de Clientes y Locales Comerciales	25 min. Por cliente	8 h
Registro y Actualización de Informes de Inspección	15 min	2,5 h

Tabla 3 - 4 Registro y actualización de Información (manual). (Fuente: Propia., 2017)

### 5.1.1. Generación de Informes y Reportes

Para el análisis del proceso de generación de Informes y reportes Registro y actualización de información se toma en cuenta los reportes relacionados con:

Actividades	Tiempo por Actividad (Minutos)	Tiempo Total por Día (horas)
Reporte de Empleados	240 min	4 h
Informe de Control de Asistencia	480 min	8 h
Reporte de Clientes	60 min	1 h
Reporte de Locales Comerciales	240 min	4 h
Reporte de Tarifas a Locales Comerciales	60 min	1 h
Generación de Informe de conciliación de Permisos de Funcionamiento Anual y por Tipo de Local Comercial	2280 min	48 h

Tabla 3 - 5 Generación de Informes y reportes. (Fuente: Propia., 2017)

### 5.3.3. Resumen de Análisis de Impactos

Actividades	Antes	Después
Búsqueda de Información (Cliente, Local Comercial, Informe de Inspección, Permisos Anteriores)	10 min.	1 min
Generación de Permiso (manual)	5 min.	10 seg
Registro y enrolamiento de Empleados	60 min por empleado.	5 min
Registro de Clientes y Locales Comerciales	25 min. Por cliente	5 min
Registro y Actualización de Informes de Inspección	15 min	5 min
Reporte de Empleados	240 min	1 min
Informe de Control de Asistencia	480 min	1 min

Reporte de Clientes	60 min	1 min
Reporte de Locales Comerciales	240 min	1 min
Reporte de Tarifas a Locales Comerciales	60 min	1 min
Generación de Informe de conciliación de Permisos de Funcionamiento Anual y por Tipo de Local Comercial	2280 min	1 min

*Tabla 3 - 6 Tiempo de Entrega de Permisos (manual). (Fuente: Propia., 2017)*

#### **Beneficios Visibles**

Los beneficios visibles aportados por el sistema propuesto están dados por los siguientes aspectos:

- Reducción de costos en documentación, papelería, mantenimiento y espacio físico.
- Control y seguimiento de los ingresos de la Institución.
- Información actualizada de los clientes de la Institución.
- Mejor empleo de los recursos, tanto materiales como financieros.
- Procesamiento de la información más rápido.
- Facilidad de acceso a la información, en cualquier momento.
- Información confiable.
- Se podrá generar reportes inmediatamente.

#### **Beneficios Imperceptibles**

Entre los beneficios del sistema propuesto se pueden incluir:

- Capacitación del personal.
- Optimización de tiempo en las actividades diarias del personal.
- Mayor seguridad de la información.
- Mejoramiento de la calidad del servicio.

## BIBLIOGRAFÍA

Cuerpo de bomberos de Otavalo. (2013). *Historia*. From Cuerpo de Bomberos de Otavalo: <http://www.bomberosotavalo.gob.ec/>

Fuente: Propia. (2017). *Fuente Propia*. Ibarra: Fuente Propia.

Ivar jacobson, G. B. (1998). *The Unified Software Development Process (ISBN 0-201-57169-2)*.

Maximiliano Firtman, Leonardo Natale. (2010). *Visual Studio .NET Framework 3.5 para profesionales*. Mexico: Alfaomega.

Microsoft. (2013). *Arquitectura de WCF*. Recuperado el 01 de Mayo de 2015, de Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/vstudio/ms733128%28v=vs.100%29.aspx>

Microsoft. (2013). *Ciclo de vida*. Retrieved Marzo 01, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms732098%28v=vs.110%29.aspx>

Microsoft. (2013). *Configuración de servicios*. Retrieved Marzo 20, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms733830%28v=vs.110%29.aspx>

Microsoft. (2013). *Contratos*. Retrieved July 10, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/vstudio/aa702732%28v=vs.100%29.aspx>

Microsoft. (2013). *Creación de clientes*. Retrieved Marzo 25, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms730825%28v=vs.110%29.aspx>

Microsoft. (2013). *Data Contracts*. Retrieved Diciembre 01, 2014 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms733127%28v=vs.110%29.aspx>

Microsoft. (2013). *Diseño e implementación de servicios*. Retrieved Marzo 10, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms729746%28v=vs.110%29.aspx>

Microsoft. (2013). *Endpoints: Addresses, Bindings, and Contracts*. Retrieved Abril 15, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/en-us/library/ms733107.aspx>

Microsoft. (2013). *Introducción a la extensibilidad*. Retrieved Febrero 10, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms789051%28v=vs.110%29.aspx>

Microsoft. (2013). *msdn.microsoft.com*. From [msdn.microsoft.com](https://msdn.microsoft.com/es-es/library/vstudio/aa702732%28v=vs.100%29.aspx): <https://msdn.microsoft.com/es-es/library/vstudio/aa702732%28v=vs.100%29.aspx>

Microsoft. (2013). *Programación básica de WCF*. Retrieved Marzo 01, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms731067%28v=vs.110%29.aspx>

Microsoft. (2013). *Servicios de hospedaje*. Retrieved Marzo 21, 2015 from Msdn Microsoft: <https://msdn.microsoft.com/es-es/library/ms730158%28v=vs.110%29.aspx>

Microsoft. (2013). *WCF*. From [www.microsoft.com](http://www.microsoft.com): <https://msdn.microsoft.com/es-es/library/vstudio/ms732098%28v=vs.100%29.aspx>

*wikipedia*. (2018, 4 4). From [wikipedia:](https://es.wikipedia.org/wiki/Proceso_Unificado_Racional) [https://es.wikipedia.org/wiki/Proceso\\_Unificado\\_Racional](https://es.wikipedia.org/wiki/Proceso_Unificado_Racional)

## **Glosario de términos:**

**Alojar:** Hospedar, aposentar.

**Binding:** Propiedad como se conecta el cliente a un servicio.

**Binario:** Sistema de numeración en el que los números se representan utilizando solamente las cifras cero y uno (0 y 1).

**Cardinalidad:** En Ingeniería del software, la cardinalidad es referente a la cantidad de relaciones que puede tener una entidad con otra.

**C#:** Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma.

**Cloud:** Término usado en informática para representar a las aplicaciones que se encuentran en internet.

IIS: Internet Information Server

**DTO:** (Objeto de transferencia de datos), es un patrón de diseño se utiliza para transferir datos entre los subsistemas de aplicaciones de software. DTO se utilizan a menudo en conjunción con los objetos de acceso a datos para recuperar datos de una base de datos”.

**Entity Framework:** Conjunto de tecnologías en ADO.NET que dan soporte al desarrollo de software orientado a datos, que permite a los desarrolladores .NET tener mecanismos para acceder y guardar datos en una base de datos.

**Encapsula:** La operación de refactorización Encapsular campo permite crear rápidamente una propiedad a partir de un campo existente y, a continuación, actualizar el código de forma transparente con referencias a la nueva propiedad.

**Endpoint:** Identifica una dirección de red. Esta es una clase abstracta de .NET.

**Framework:** (plataforma, entorno, marco de trabajo). Desde el punto de vista del desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.

**Host:** Se refiere a las computadoras conectadas a una red, que proveen y utilizan servicios de ella.

**Hosting:** Quiere decir dar, hospedar o alojar. Aplicado al Internet, significa poner una página web en un servidor de Internet para que ella pueda ser vista en cualquier lugar del mundo entero con acceso al Internet.

**Http:** HTTP de **Hyper Text Transfer Protocol** (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

**IIS:** Los servicios de Internet Information Server (o IIS), son los servicios de software que admiten la creación, configuración y administración de sitios Web, además de otras funciones de Internet. Los servicios de Microsoft Internet Information Server incluyen el Protocolo de transferencia de noticias a través de la red (NNTP), el Protocolo de transferencia de archivos (FTP) y el Protocolo simple de transferencia de correo (SMTP). **Iteraciones:** Acción y efecto de iterar. Iteración acto de repetir un proceso con el objetivo de alcanzar una meta deseada, objetivo o resultado. Cada repetición del proceso también se le denomina una "iteración", y los resultados de una iteración se utilizan como punto de partida para la siguiente iteración.

**Https:** Hypertext Transfer Protocol Secure, más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.



**Link:** Enlaces de hipertexto, son URLs que forman parte de una página web. Son accesos directos a otros sitios web a otras partes de la misma página. Normalmente están subrayados y con un color distinto al resto del texto.

**Microsoft:** Microsoft es actualmente la mayor compañía de desarrollo de software para ordenadores y dispositivos informáticos. Si se conoce Windows, trabaja con Office o se ha usado alguna vez una consola de DOS, todos estos productos llevan su marca; **Microsoft Corporation.**

**Microsoft Azure:** Es la plataforma de informática en la nube de Microsoft, una colección cada vez mayor de servicios integrados (análisis, proceso, bases de datos, móviles, redes, almacenamiento y web).

**Metodología:** Conjunto de métodos que se siguen en una disciplina científica, en un estudio o en una exposición doctrinal.

**MSMQ (Microsoft Message Queuing):** Es una herramienta de Microsoft para realizar el manejo de colas de mensajes, viene con Windows NT/2000 y cuenta con una interfaz que permite escribir programas en lenguaje C o utilizando componentes COM para realizar el envío y recepción de mensajes.

**NET:** El nombre de dominio .net es un dominio de nivel superior genérico (gTLD por sus siglas en inglés) utilizado en el Sistema de Nombres de Dominio de Internet. Su nombre deriva de la palabra inglesa «network», que significa red, lo que indica que en un principio estaba previsto para organizaciones relacionadas con tecnologías de redes, como es el caso de los proveedores de servicios de Internet y otras empresas de infraestructura.

**Namespace:** (tubería nombrada (named pipe en inglés), también llamada FIFO por su comportamiento, es una extensión del concepto tradicional de tuberías utilizado en los **Named Pipe:** En ingeniería de software por su comportamiento, es uno de los

métodos de Comunicación entre procesos donde el primer elemento en entrar es el primero en salir.

**Repositorio:** Es un término utilizado en el dominio de las herramientas CASE. El repositorio podría definirse como la base de datos fundamental para el diseño; no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos software necesarios para el trabajo de programación.

**Request:** Solicitud de información o datos que una computadora cliente efectúa a un servidor.

**RUP (Rational Unified Process):** Proceso de desarrollo de software y junto con el lenguaje unificando de modelo UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistema orientados a abiertos.

**Service:** Servicio.

**Servicehost:** Servicio de acogida.

**System.ServiceModel:** Modelo del sistema de Servicio.

**Servicecontractattribute:** Indica que una interfaz o una clase define un contrato de servicio en una aplicación de Windows Communication Foundation (WCF).

**Streaming:** Es También denominado transmisión, lectura en continuo, difusión en flujo.

**OperationContractAttribute:** Indica que un método define una operación que forma parte de un contrato de servicio en una aplicación Windows Communication Foundation (WCF).

**TCP:** Transmission Control Protocol (en español Protocolo de Control de Transmisión) o TCP, es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 y 1974 por Vint Cerf y Robert Kahn.

**Timeout:** Tiempo de espera.

**WAS:** Windows Process Activation Service, generaliza el modelo de proceso de Internet Information Services (IIS), lo que elimina la dependencia de HTTP. Todas las características de IIS que anteriormente solo estaban disponibles para aplicaciones HTTP ahora lo están para las aplicaciones que hospedan servicios Windows Communication Foundation (WCF) con protocolos distintos de HTTP. IIS 7.0 usa también el servicio WAS (Windows Process Activation Service) para la activación basada en mensajes a través de HTTP.

**WPF:** Windows Presentation Foundation (WPF) es una tecnología de Microsoft, presentada como parte de Windows Vista. Permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web.

**WCF:** Windows Communication Foundation (WCF) es un marco de trabajo para la creación de aplicaciones orientadas a servicios. Con WCF, es posible enviar datos como mensajes asíncronos de un extremo de servicio a otro. Un extremo de servicio puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación. Un extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio. Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios.

**Windows:** Windows es una palabra del idioma inglés que significa “ventana”. Su utilización en español, en cambio, está casi exclusivamente vinculada a un sistema informático desarrollado por la empresa Microsoft y comercializado desde 1985.

Windows Azure:

**Workload:** En informática, es la carga de trabajo o la cantidad de procesamiento que el ordenador ha sido designado para hacer en un momento dado.

**WS-I Basic Profile 1.1:** Consiste en un conjunto de especificaciones de servicios Web no propietarios, junto con aclaraciones, refinamientos, interpretaciones y ampliaciones de dichas especificaciones que promueven la interoperabilidad.

**XML:** Siglas en inglés Xtensible Markup Language, es un language de marcas desarrollado por el World Wide Web Consortium (W3C), utilizado para almacenar datos en forma legible, permitiendo definir la gramática de lenguajes específicos de la misma manera que HTML.

## **Anexo 1: Manual de Usuario.**

El anexo correspondiente al manual de usuario se anexa de manera digital en el medio magnético en cd adjunto a este documento.