



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E S C I E N T I A H O M I N I S S A L U S "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA WEB PROTOTIPO PARA FACTURACIÓN ELECTRÓNICA DE UNA FERRETERÍA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

FRANCISCO JAVIER GUALLI GODOY
francisco.gualli@epn.edu.ec

DIRECTOR: MSc. GABRIEL ROBERTO LÓPEZ FONSECA
gabriel.lopez@epn.edu.ec

CODIRECTOR: MSc. FRANKLIN LEONEL SÁNCHEZ CATOTA
franklin.sanchez@epn.edu.ec

Quito, julio 2019

AVAL

Certificamos que el presente trabajo fue desarrollado por Francisco Javier Gualli Godoy bajo nuestra supervisión.

MSc. GABRIEL ROBERTO LÓPEZ FONSECA
DIRECTOR DEL TRABAJO DE TITULACIÓN

MSc. FRANKLIN LEONEL SÁNCHEZ CATOTA
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Francisco Javier Gualli Godoy, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejó constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

FRANCISCO JAVIER GUALLI GODOY

DEDICATORIA

A mis padres, por enseñarme el valor del trabajo duro y el esfuerzo que se realiza cada día por salir adelante.

A mis hermanos, por enseñarme que a pesar de todas las circunstancias de la vida siempre tienes alguien con quien contar.

A mi familia, por enseñarme muchas lecciones de vida y la madurez que conllevan todas las responsabilidades de un hogar.

A mis amigos, por enseñarme que no todo en la vida es tan serio, moderado o calmado.

AGRADECIMIENTOS

*Va por lo que venga, va por lo que fui
por toda la buena experiencia que gane
por demostrarme algo de interés
por esa confianza que me ayudo a crecer
por errores y fracasos que me hicieron entender
por el dulce cariño de una dulce mujer
por los pocos amigos que se pueden tener
por los bellos momentos en un rato familiar
por los bellos recuerdos que no se podrán borrar
por el toque genial que la música le da
a esta vida hundida en una dura realidad
es por eso que celebro con satisfacción
armonía y pura buena vibración*

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTOS	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	XII
ÍNDICE DE SEGMENTOS DE CÓDIGO	XIII
RESUMEN	XIV
ABSTRACT	XV
1. INTRODUCCIÓN	1
1.1. OBJETIVOS.....	1
1.2. ALCANCE	2
1.3. MARCO TEÓRICO	5
1.3.1. METODOLOGÍAS DE DESARROLLO	5
1.3.2. ARQUITECTURA N-CAPAS.....	8
1.3.3. BASE DE DATOS.....	9
1.3.4. .NET FRAMEWORK.....	12
1.3.5. TECNOLOGÍAS PARA DESARROLLO FRONT END	18
1.3.6. GITHUB	22
1.3.7. FACTURACIÓN ELECTRÓNICA.....	25
2. METODOLOGÍA	28
2.1. DISEÑO DEL PROTOTIPO	29
2.2. ANÁLISIS DE REQUERIMIENTOS	29
2.3. ESPECIFICACIÓN DE REQUERIMIENTOS	30
2.3.1. REQUERIMIENTOS FUNCIONALES.....	30
2.3.2. REQUERIMIENTOS NO FUNCIONALES	31
2.4. HISTORIAS DE USUARIO	32
2.4.1. FORMATO.....	32
2.4.2. DETALLE DE HISTORIAS DE USUARIO	33
2.5. DIAGRAMA DE CASO DE USOS.....	44
2.6. ROLES Y PERMISOS.....	44
2.6.1. ROLES Y PERMISOS DE ROL	44
2.7. ARQUITECTURA.....	47
2.8. DIAGRAMA ENTIDAD-RELACIÓN	48
2.8.1. DESCRIPCIÓN DE TABLAS.....	48
2.9. DIAGRAMA DE CLASES.....	52
2.9.1. CAPA DE ACCESO A DATOS	52
2.9.2. CAPA DE ENTIDADES	52
2.9.3. CAPA DE NEGOCIO.....	52
2.10. MÓDULOS DEL SISTEMA Y VISTAS DE USUARIO	62
2.10.1. MÓDULO DE AUTENTICACIÓN	62
2.10.2. MÓDULO DE ADMINISTRACIÓN	64
2.10.3. MÓDULO DE COMPROBANTES.....	104
2.10.4. MÓDULO DE REPORTES.....	121
2.10.5. MÓDULO DE CLIENTES	128

2.11. CODIFICACIÓN DEL SISTEMA.....	129
2.11.1. CODIFICACIÓN DE LA CAPA DE DATOS	129
2.11.2. CODIFICACIÓN DE LA CAPA DE ACCESO A DATOS	132
2.11.3. CODIFICACIÓN DE LA CAPA DE ENTIDADES.....	136
2.11.4. CODIFICACIÓN DE LA CAPA DE NEGOCIO	136
2.11.5. CODIFICACIÓN DE LA CAPA DE SERVICIO	137
2.11.6. CODIFICACIÓN DE LA CAPA DE PRESENTACIÓN.....	142
2.12. IMPLEMENTACIÓN DEL SISTEMA EN SERVIDOR WEB.....	151
2.12.1. PREPARACIÓN DEL SERVIDOR.....	151
2.12.2. MIGRACIÓN DE BASE DE DATOS	153
2.12.3. CONFIGURACIÓN IIS.....	154
2.12.4. DESPLIEGUE DEL SISTEMA EN EL SERVIDOR WEB	155
3. RESULTADOS Y DISCUSIÓN	158
3.1. CONTROL DE VERSIONES DE LA APLICACIÓN	158
3.2. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DATOS.....	161
3.3. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE ACCESO A DATOS	162
3.4. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE NEGOCIO	167
3.5. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE SERVICIO.....	168
3.6. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE PRESENTACIÓN.....	170
3.6.1. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE AUTENTICACIÓN	170
3.6.2. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE ADMINISTRACIÓN	174
3.6.3. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE COMPROBANTES	179
3.6.4. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE REPORTES	184
3.6.5. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE CLIENTES	188
3.7. CORRECCIÓN DE ERRORES EN LOS COMPONENTES DEL PROTOTIPO.	190
3.7.1. RESUMEN DE CORRECCIONES.....	190
4. CONCLUSIONES Y RECOMENDACIONES.....	192
4.1. CONCLUSIONES	192
4.2. RECOMENDACIONES.....	193
5. REFERENCIAS BIBLIOGRÁFICAS	194
ANEXOS.....	196
ORDEN DE EMPASTADO	197

ÍNDICE DE FIGURAS

Figura 1.1 Fases de desarrollo de la metodología RAD [2].	2
Figura 1.2 Arquitectura N-Capas [3].	3
Figura 1.3 Implementación del sistema en un servidor en la nube	3
Figura 1.4 Proceso cíclico de RAD	6
Figura 1.5 Ejemplo nueva capa en arquitectura N-Capas [3]	9
Figura 1.6 Esquema del patrón de diseño repositorio [6]	9
Figura 1.7 Relación con atributos [7]	10
Figura 1.8 Ejemplo de código HTML	18
Figura 1.9 Creación de un repositorio en GitHub.	23
Figura 1.10 Creación de una nueva rama en GitHub	24
Figura 1.11 Realizar cambios en un repositorio	24
Figura 1.12 Revisión de cambios en repositorio GitHub	25
Figura 1.13 Proceso de autorización de comprobantes electrónicos [17].	28
Figura 1.14 Campos requeridos para creación de clave de acceso [18].	28
Figura 2.1 Diagrama de caso de uso del prototipo	46
Figura 2.2 Arquitectura del prototipo	48
Figura 2.4 Diagrama de clases de la capa de acceso a datos	53
Figura 2.5 Diagrama de clases de la capa de entidades (PARTE I)	54
Figura 2.6 Diagrama de clases de la capa de entidades (PARTE II)	55
Figura 2.7 Diagrama de clases de la capa de negocio.	56
Figura 2.8 Diagrama de clases de la capa de servicio.	57
Figura 2.9 Diagrama de clases de la capa de presentación (PARTE I).	58
Figura 2.10 Diagrama de clases de la capa de presentación (PARTE II).	59
Figura 2.11 Diagrama de clases de la capa de presentación (PARTE III).	60
Figura 2.12 Diagrama de clases de la capa de presentación (PARTE IV).	61
Figura 2.13 Diagrama de actividades del módulo de autenticación	62
Figura 2.14 Vista de inicio de sesión usuario.	63
Figura 2.15 Vista de inicio de sesión cliente.	63
Figura 2.16 Vista de inicio de sesión usuario con rol de administrador.	64
Figura 2.17 Vista principal al ingreso del usuario.	64
Figura 2.18 Diagrama de clases del módulo de administración (Ingreso y actualización de entidad)	65
Figura 2.19 Diagrama de clases módulo de administración (Eliminar entidad)	65
Figura 2.20 Diagrama de clases módulo de administración (Visualizar información entidad)	66
Figura 2.21 Vista principal de administración de usuarios.	67
Figura 2.22 Vista para agregar un nuevo usuario al sistema.	67
Figura 2.23 Vista para actualizar un usuario existente del sistema.	68
Figura 2.24 Vista para ver información de un usuario existente del sistema.	68
Figura 2.25 Vista general para administrar certificados digitales.	69
Figura 2.26 Vista para agregar un nuevo certificados digital al sistema.	69
Figura 2.27 Vista para actualizar un certificado digital existente del sistema.	70
Figura 2.28 Vista para ver información de un certificado digital existente del sistema.	70
Figura 2.29 Vista general para administrar emisores.	71
Figura 2.30 Vista para agregar un nuevo emisor al sistema.	71
Figura 2.31 Vista para actualizar un emisor existente del sistema.	72
Figura 2.32 Vista para ver información de un emisor existente del sistema.	72
Figura 2.33 Vista para eliminar un emisor del sistema.	73
Figura 2.34 Vista general para administrar parámetros de configuración de red.	74
Figura 2.35 Vista para agregar un nuevo parámetro de configuración de red al sistema.	74
Figura 2.36 Vista para actualizar un parámetro de configuración de red existente del	

sistema.....	75
Figura 2.37 Vista para ver información de un parámetro de configuración de red existente del sistema.....	75
Figura 2.38 Vista general para administrar impuestos.....	76
Figura 2.39 Vista para agregar un nuevo impuesto al sistema.....	76
Figura 2.40 Vista para actualizar un impuesto existente del sistema.....	77
Figura 2.41 Vista para ver información de un impuesto existente del sistema.....	77
Figura 2.42 Vista general para administrar formas de pago.....	78
Figura 2.43 Vista para agregar una nueva forma de pago al sistema.....	79
Figura 2.44 Vista para actualizar una forma de pago existente del sistema.....	79
Figura 2.45 Vista para ver información de una forma de pago existente del sistema.....	80
Figura 2.46 Vista general para administrar tipos de comprobantes.....	80
Figura 2.47 Vista para actualizar un tipo de comprobante existente del sistema.....	81
Figura 2.48 Vista para ver información de un tipo de comprobante existente del sistema.....	81
Figura 2.49 Vista general para administrar sucursales.....	83
Figura 2.50 Vista para agregar una nueva sucursal al sistema.....	83
Figura 2.51 Vista para actualizar una sucursal existente del sistema.....	84
Figura 2.52 Vista para ver información de una sucursal existente del sistema.....	84
Figura 2.53 Vista para eliminar una sucursal del sistema.....	85
Figura 2.54 Vista general para administrar clientes.....	85
Figura 2.55 Vista para agregar un nuevo cliente al sistema.....	86
Figura 2.56 Vista para actualizar un cliente existente del sistema.....	86
Figura 2.57 Vista para ver información de un cliente existente del sistema.....	87
Figura 2.58 Vista para eliminar un cliente del sistema.....	87
Figura 2.60 Vista para agregar un nuevo transportista al sistema.....	89
Figura 2.61 Vista para actualizar un transportista existente del sistema.....	89
Figura 2.62 Vista para ver información de un transportista existente del sistema.....	90
Figura 2.63 Vista para eliminar un transportista del sistema.....	90
Figura 2.64 Vista general para administrar proveedores.....	92
Figura 2.65 Vista para agregar un nuevo proveedor al sistema.....	92
Figura 2.66 Vista para actualizar un proveedor existente del sistema.....	93
Figura 2.67 Vista para ver información de un proveedor existente del sistema.....	93
Figura 2.68 Vista para eliminar un proveedor del sistema.....	94
Figura 2.69 Vista general para administrar categorías de productos.....	94
Figura 2.70 Vista para agregar una categoría de productos al sistema.....	95
Figura 2.71 Vista para actualizar una categoría de productos existente del sistema.....	95
Figura 2.72 Vista para ver información de una categoría de productos existente del sistema.....	96
Figura 2.73 Vista para eliminar una categoría de productos del sistema.....	96
Figura 2.74 Vista general para administrar productos.....	98
Figura 2.75 Vista para agregar un producto al sistema.....	98
Figura 2.76 Vista para actualizar un producto existente del sistema.....	99
Figura 2.77 Vista para ver información de un producto existente del sistema.....	99
Figura 2.78 Vista para eliminar un producto del sistema.....	100
Figura 2.79 Vista general para administrar costos.....	100
Figura 2.80 Vista para agregar un costo al sistema.....	101
Figura 2.81 Vista para actualizar un costo existente del sistema.....	101
Figura 2.82 Vista para ver información de un costo existente del sistema.....	102
Figura 2.83 Vista general para administrar gastos.....	102
Figura 2.84 Vista para agregar un gasto al sistema.....	103
Figura 2.85 Vista para actualizar un gasto existente del sistema.....	103
Figura 2.86 Vista para ver información de un gasto existente del sistema.....	104
Figura 2.87 Diagrama de actividades para el ingreso de un nuevo comprobante	

electrónico	105
Figura 2.88 Vista para visualizar un comprobante electrónico.....	106
Figura 2.89 Vista para generar una cotización.....	107
Figura 2.90 Vista desplegada con lista de cotizaciones generadas.....	108
Figura 2.91 Vista para generar una factura.....	110
Figura 2.92 Vista desplegada con lista de facturas generadas.....	111
Figura 2.93 Vista para generar un comprobante de retención.....	112
Figura 2.94 Vista desplegada con lista de comprobantes de retención generados.....	113
Figura 2.95 Vista para generar una nota de crédito.....	114
Figura 2.96 Vista desplegada con lista de notas de crédito generadas.....	115
Figura 2.97 Vista para generar una nota de débito.....	117
Figura 2.98 Vista desplegada con lista de notas de débito generadas.....	118
Figura 2.99 Vista para generar una guía de remisión.....	119
Figura 2.100 Vista desplegada con lista de guías de remisión generadas.....	120
Figura 2.101 Diagrama de actividades del módulo de reportes.....	121
Figura 2.102 Vista de reporte gerencial de ventas.....	122
Figura 2.103 Vista de reporte gerencial de compras.....	123
Figura 2.104 Vista de reporte de compras versus ventas.....	123
Figura 2.105 Vista de reporte de productos más vendidos.....	124
Figura 2.106 Vista de reporte gerencial de tendencia de ventas.....	125
Figura 2.107 Vista de reporte de análisis de gastos.....	126
Figura 2.108 Vista de reporte de análisis de costos.....	127
Figura 2.109 Vista de reporte de movimientos de comprobantes.....	127
Figura 2.110 Diagrama de actividades del módulo de clientes.....	128
Figura 2.111 Vista de búsqueda de comprobantes de clientes.....	129
Figura 2.112 Estructura de una aplicación n-capas en Visual Studio 2017.....	133
Figura 2.113 Creación de un proyecto de biblioteca de clases (.NET Framework) en Visual Studio 2017.....	133
Figura 2.114 Creación de un elemento Entity Framework en Visual Studio 2017.....	134
Figura 2.115 Estructura proyecto DataAccessLayerTest.....	136
Figura 2.116 Estructura del contexto de datos generado por Entity Framework.....	136
Figura 2.117 Estructura del proyecto EntitiesLayer.....	137
Figura 2.118 Referencias del proyecto BusinessLayer.....	138
Figura 2.119 Estructura proyecto BusinessLayerTest.....	139
Figura 2.120 Estructura capa de servicio.....	140
Figura 2.121 Estructura referencias a servicios web.....	141
Figura 2.122 Estructura proyecto ServiceLayerTest.....	142
Figura 2.123 Referencias del proyecto PresentationLayer.....	143
Figura 2.124 Cookie del usuario almacenada en el navegador.....	145
Figura 2.125 Directorio de almacenamiento del módulo de administración.....	146
Figura 2.126 Administrador de paquetes NuGet.....	147
Figura 2.127 Módulo de comprobantes en capa de presentación.....	148
Figura 2.128 Plantillas de comprobantes electrónicos.....	148
Figura 2.129 Gráfico generado con ChartJS.....	150
Figura 2.130 Módulo de clientes en la capa de presentación.....	151
Figura 2.131 Administración de DNS de dominio en plataforma GoDaddy.....	151
Figura 2.132 Configuración IP pública.....	152
Figura 2.133 Configuración LAN.....	152
Figura 2.134 Configuración IPv4 del servidor de producción.....	152
Figura 2.135 Configuración para redirección de puertos.....	153
Figura 2.136 Carga de base de datos en SQL Server.....	153
Figura 2.137 Base de datos migrada a servidor de producción.....	154
Figura 2.138 Configuración de enlaces en IIS.....	154
Figura 2.139 Habilitar publicación Web Deploy en IIS.....	155

Figura 2.140 Perfil de publicación generado con herramienta Web Deploy.....	155
Figura 2.141 Menú de capa presentación.....	156
Figura 2.142 Perfil de publicación del proyecto para el servidor web.....	157
Figura 2.143 Salida del proceso de publicación del sistema en el servidor.....	157
Figura 2.144 Sistema publicado en servidor web.....	158
Figura 3.1 Panel de administración de control de versiones GitHub.....	159
Figura 3.2 Panel de archivos actualizados en GitHub.....	160
Figura 3.3 Panel de verificación de cambios realizados en archivos de GitHub.....	161
Figura 3.4 Ejemplo de tabla de auditoría.....	161
Figura 3.5 Método para testing de ingreso de una entidad usando capa de acceso a datos.....	162
Figura 3.6 Entidad ingresada en la capa de datos.....	162
Figura 3.7 Método para testing de actualización de una entidad usando capa de acceso a datos.....	163
Figura 3.8 Entidad actualizada en la capa de datos.....	163
Figura 3.9 Método para testing de eliminación de una entidad usando capa de acceso a datos.....	164
Figura 3.10 Entidad eliminada de la capa de datos.....	164
Figura 3.11 Método para testing de listado de entidades usando la capa de acceso a datos.....	164
Figura 3.12 Lista de entidades devueltas por la capa de acceso a datos.....	165
Figura 3.13 Método para testing de búsqueda de entidades usando la capa de acceso a datos.....	165
Figura 3.14 Lista de entidades encontradas devuelta por la capa de acceso a datos.....	166
Figura 3.15 Método para testing de ejecución de un procedimiento almacenado usando la capa de acceso a datos.....	166
Figura 3.16 Resultado de la ejecución del procedimiento almacenado devuelto por la capa de acceso a datos.....	167
Figura 3.17 Implementación clase de prueba.....	167
Figura 3.18 Explorador de pruebas de Visual Studio.....	168
Figura 3.19 Método para el <i>testing</i> del envío de comprobantes usando la capa de servicio.....	168
Figura 3.20 Respuesta de llamada al servicio web devuelta por la capa de servicio.....	168
Figura 3.21 Método para el testing de la verificación de un comprobante usando la capa de servicio.....	169
Figura 3.22 Resultado verificación del comprobante con el servicio web usando la capa de servicio.....	169
Figura 3.23 Intento de autenticación sin credenciales.....	170
Figura 3.24 Ingreso de caracteres especiales.....	171
Figura 3.25 Intento de ingreso sin credenciales como cliente.....	171
Figura 3.26 Intento de autenticación con credenciales incorrectas.....	172
Figura 3.27 Panel de selección de sucursal.....	172
Figura 3.28 Ingreso como administrador.....	173
Figura 3.29 Ingreso como administrador de sucursal.....	173
Figura 3.30 Ingreso como usuario.....	174
Figura 3.31 Ingreso como cliente.....	174
Figura 3.32 Intento de ingreso de nuevo usuario con campos vacíos.....	175
Figura 3.33 Ingreso de nuevo usuario.....	175
Figura 3.34 Mensaje de confirmación de registro de nuevo usuario.....	176
Figura 3.35 Intento de ingreso con credenciales de nuevo usuario.....	176
Figura 3.36 Ingreso al sistema con nuevo usuario.....	177
Figura 3.37 Actualización de información de usuario.....	177
Figura 3.38 Lista de usuarios registrados en el sistema.....	178
Figura 3.39 Intento de autenticación con credenciales de usuario inactivo.....	178

Figura 3.40 Mensaje de error en inicio de sesión.....	179
Figura 3.41 Generación nuevo comprobante.	180
Figura 3.42 Listado de comprobantes generados.	181
Figura 3.43 Mensaje de verificación de comprobante.....	181
Figura 3.44 Visualización de comprobante generado	181
Figura 3.45 Ventana de ingreso de destinatarios de correo electrónico.....	182
Figura 3.46 Mensaje de verificación de envío de comprobante.....	182
Figura 3.47 Comprobante recibido mediante correo electrónico.	183
Figura 3.48 Archivo PDF del comprobante electrónico recibido.	183
Figura 3.49 Archivo XML del comprobante electrónico recibido.....	183
Figura 3.50 Visualización de reporte gerencial de ventas.....	185
Figura 3.51 Reporte exportado como archivo PDF.....	186
Figura 3.52 Resultado de vista Vw_DatosVentas.	187
Figura 3.53 Resultado del procedimiento almacenado Sp_GetDatosVentas.....	187
Figura 3.54 Ingreso de identificación del cliente.....	188
Figura 3.56 Visualización de KARDEX del comprobante.....	189
Figura 3.57 Visualización de comprobante.....	189

ÍNDICE DE TABLAS

Tabla 1.1 Prefijos de las clases CSS para BootStrap [16].	22
Tabla 2.1 Formato de historia de usuario	32
Tabla 2.2 HU Autenticación en el sistema	33
Tabla 2.3 HU Modificar perfil personal.	33
Tabla 2.4 HU Administración de usuarios.	34
Tabla 2.5 HU Administración de sucursales	34
Tabla 2.6 HU Administración de emisores.	34
Tabla 2.7 HU Administración de certificados digitales	35
Tabla 2.8 HU Administración de parámetros de red	35
Tabla 2.9 HU Administración de impuestos manejados por el SRI	36
Tabla 2.10 HU Administración de formas de pago manejados por el SRI.	36
Tabla 2.11 HU Administración de tipos de comprobantes	37
Tabla 2.12 HU Administración de clientes	37
Tabla 2.13 HU Administración de transportistas	38
Tabla 2.14 HU Administración de proveedores	38
Tabla 2.15 HU Administración de categorías de productos	38
Tabla 2.16 HU Administración de productos	39
Tabla 2.17 HU Administración de costos.	39
Tabla 2.18 HU Administración de gastos.	40
Tabla 2.19 HU Generación de cotizaciones	40
Tabla 2.20 HU Generación de facturas	40
Tabla 2.21 HU Generación de notas de crédito	41
Tabla 2.22 HU Generación de notas de débito	41
Tabla 2.23 HU Generación de comprobantes de retención	42
Tabla 2.24 HU Generación de guías de remisión.	42
Tabla 2.25 HU Visualización de comprobantes electrónicos generados.	42
Tabla 2.26 HU Generación de reportes globales del negocio.	43
Tabla 2.27 HU Generación de reportes por sucursal	43
Tabla 2.28 Formularios para el módulo de administración	145
Tabla 2.29 Formularios para el módulo de comprobantes	146
Tabla 2.30 Formularios para el módulo de reportes.	149
Tabla 3.1 Tabla de resumen de correcciones.	190

ÍNDICE DE SEGMENTOS DE CÓDIGO

Segmento de código 1.1 Ejemplo de inserción de código CSS en un documento HTML	20
Segmento de código 2.1 Script para la creación de la tabla usuarios	130
Segmento de código 2.2 Script para la creación de la tabla usuarios	130
Segmento de código 2.3 Script para la creación del procedimiento almacenado Sp_GetDatosTendenciaVentas	131
Segmento de código 2.4 Script para la creación del trigger Guardar_Emisor_Logs	132
Segmento de código 2.5 Definición de la interfaz IRepository	134
Segmento de código 2.6 Ejemplo de implementación de la interfaz IRepository	135
Segmento de código 2.7 Ejemplo de implementación del patrón de diseño Factory	135
Segmento de código 2.8 Método para agregar una categoría	139
Segmento de código 2.9 Generación solicitud HTTP desde código C#	141
Segmento de código 2.10 Validación de campos del lado del cliente	144
Segmento de código 2.11 Definición clase Encoding	144
Segmento de código 2.12 Configuración de seguridad mediante formularios	144
Segmento de código 2.13 Llamada a método C# desde JavaScript (AJAX)	150

RESUMEN

El presente trabajo de titulación se enfoca en la creación de un sistema web prototipo que permita emitir comprobantes electrónicos como son facturas, retenciones, guías de remisión, notas de débito y notas de crédito autorizados a través de los servicios web proporcionados por el SRI, así mismo permitir un manejo centralizado de las sucursales con las que cuenta la ferretería MADERS HERRAJES y un manejo más eficiente del proceso de facturación.

En el primer capítulo del presente trabajo se describe principalmente los objetivos, alcance y el fundamento teórico con el cual se desarrolló el sistema, seguidamente en el siguiente capítulo se expone el proceso de diseño y codificación del prototipo, después en el capítulo tres se presentan las pruebas y resultados obtenidos de la implementación del prototipo y finalmente se muestran las conclusiones y recomendaciones obtenidas tras la finalización del trabajo de titulación.

Dicho sistema está realizado utilizando la arquitectura N-Capas bajo el lenguaje de programación C# con el framework ASP.NET además del uso de tecnologías Frontend como HTML5, CSS3, BootStrap y JavaScript e implementado en un servidor en Internet.

Una vez implementado el sistema y realizadas las pruebas correspondientes, se pudo verificar que las funciones cumplen con los requerimientos establecidos por la ferretería MADERS HERRAJES, además, se pudo comprobar la extensibilidad del sistema hacia nuevas características que pudieren desarrollarse en el futuro.

Palabras clave: Facturación Electrónica, N-Capas, ASP.NET, Cloud Computing, C#.

ABSTRACT

The present project is focused on the creation of a system that allows issuing electronic receipts such as invoices, retentions, referral guides, debit notes and credit notes authorized through the web services provided by the SRI, likewise allowing a centralized management of the branches with which the MADERS HERRAJES Hardware Store has and a more efficient management of the billing process.

The first chapter of the present project, describe the objectives, scope and the theoretical basis with which the system was developed, then in the next chapter the design and codification process of the prototype is exposed, then in chapter three the tests and results obtained from the prototype implementation and finally the conclusions and recommendations obtained after the completion of the project.

This system was made using the N-Layers architecture under the C # programming language with the ASP.NET framework, in addition the use of Frontend technologies like HTML5, CSS3, Bootstrap and JavaScript and implemented in a server on the Internet.

Once the system was implemented and the corresponding tests were carried out, it was possible to verify that the functionality satisfied the requirements established by MADERS HERRAJES Hardware Store, in addition, it was possible to verify the extensibility of the system towards new features that could be developed in the future.

Keywords: Electronic Invoicing, N-Layers, ASP.NET, Cloud Computing, C#.

1. INTRODUCCIÓN

Los avances tecnológicos actuales han permitido modernizar todo tipo de procesos permitiendo brindar mayor competitividad, rentabilidad y eficiencia, por lo que se ha vuelto necesario que la mayoría de estos se adapten y actualicen. Uno de estos procesos es el de facturación que genera altos costos relacionados con impresión, autorización, envío y conservación de documentos tributarios físicos, que hacen de la facturación electrónica una excelente alternativa para contar con un proceso que permita emitir, generar y enviar comprobantes en entornos digitales.

Actualmente en el Ecuador existen alrededor de 72000 empresas que realizan facturación electrónica que corresponden al 83% del total de contribuyentes. Si bien a la fecha no es una obligación para todos los contribuyentes emitir comprobantes electrónicos para el mes de enero del año 2020 lo será [1].

Por lo mencionado, se plantea una alternativa al modelo tradicional de facturación haciendo uso de medios digitales mediante el desarrollo de una herramienta de facturación electrónica que esté en capacidad de emitir y almacenar comprobantes electrónicos para la ferretería MADERS HERRAJES, la cual a su vez será capaz de enviar a sus clientes los comprobantes electrónicos de las transacciones realizadas.

• OBJETIVOS

El objetivo del presente trabajo de titulación es desarrollar un sistema web prototipo para facturación electrónica de una ferretería.

Los objetivos específicos son:

- Analizar las diferentes herramientas que servirán para el desarrollo e implementación del sistema.
- Diseñar cada uno de los componentes del sistema prototipo propuesto.
- Implementar cada uno de los componentes diseñados.
- Analizar los resultados de las pruebas funcionales del prototipo.

- **ALCANCE**

El trabajo de titulación presentado propone el desarrollo de una aplicación web en base a la información de la ferretería MADERS HERRAJES utilizando un enfoque para 3 sucursales, con el principal objetivo de generar comprobantes electrónicos como son facturas, notas de crédito, notas de débito, comprobantes de retención y guías de remisión según el formato establecido por el Servicio de Rentas Internas del Ecuador. Además de brindar al usuario varias utilidades y funciones en base a los mismos como la sincronización de información online entre la matriz y las respectivas sucursales, resumen de movimientos (KARDEX), tendencia de ventas, análisis de costos, análisis de gastos, estado del negocio, y consulta de comprobantes online.

Para el desarrollo de software se utilizará la metodología de desarrollo RAD (*Rapid Application Development*) debido a la flexibilidad y los rápidos resultados obtenidos con esta metodología.

En la Figura 1.1 se presentan las fases de desarrollo de la metodología RAD.

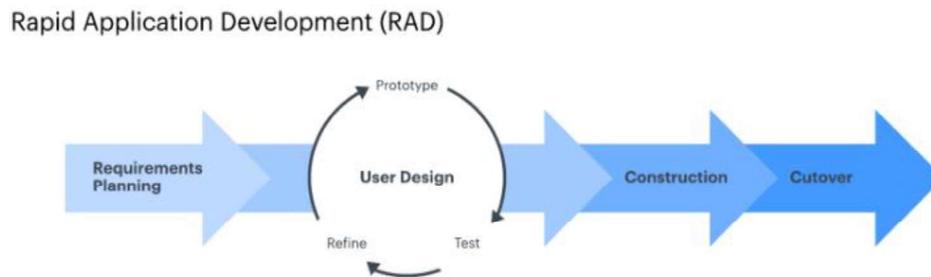


Figura 1.1 Fases de desarrollo de la metodología RAD [2].

Además de la metodología seleccionada se utilizará patrones de diseño y la arquitectura de n-capas para permitir la separación de responsabilidades y proporcionar mayor flexibilidad y un mejor mantenimiento del sistema, cabe recalcar que cada capa se encontrará en un proyecto diferente dentro de la misma solución, en la Figura 1.2 se muestran las capas presentes en la arquitectura.

Las capas a implementarse para el sistema serán: capa de datos que será generada en SQL Server, capa de acceso a datos que será generada utilizando *entity framework*, capa de entidades que al igual que la capa de acceso a datos será generada mediante *entity framework*, capa de lógica de negocio que será la capa con mayor importancia debido a que en esta se maneja toda la funcionalidad del sistema, capa de servicio o proxy que será la encargada de conectarse con los servicios web del SRI y la capa de presentación que maneja toda la interfaz gráfica del sistema.



Figura 1.2 Arquitectura N-Capas [3].

El sistema completo se lo realizará en el modelo de desarrollo web unificado ASP.NET con el lenguaje de programación orientada a objetos C#, además con la versión 5 del lenguaje de marcado y etiquetado HTML5, el uso de la versión 3 de hojas en estilo de cascada CSS3 y el *framework* Bootstrap. Para el almacenamiento de datos se utilizará SQL Server, el control de versiones será implementado en GitHub y todo el sistema será implementado en un servidor web.

El esquema de implementación del sistema en el servidor web, se muestra en la Figura 1.3.

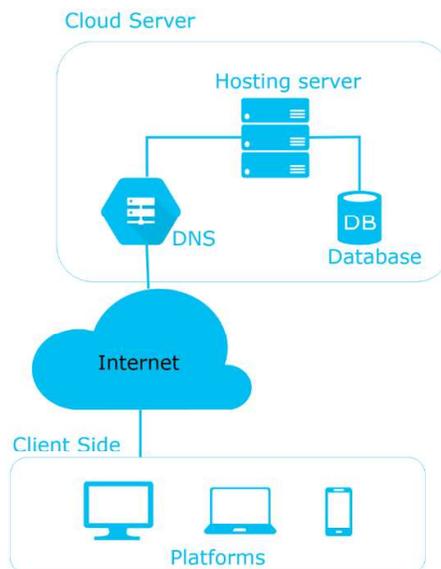


Figura 1.3 Implementación del sistema en un servidor en la nube

El sistema será implementado en un servidor web accesible desde Internet.

El sistema estará dividido en varios módulos descritos a continuación:

1. El primer módulo es el de autenticación que permitirá validar las credenciales de acceso para usuarios y clientes, además de validar los permisos para cada usuario.
2. El siguiente es el módulo de administración que permitirá la creación, edición, eliminación de los componentes necesarios para la facturación electrónica como son datos del emisor, datos del punto de emisión, datos del certificado digital utilizado para firmar los comprobantes, parámetros del SRI, datos del tipo de ambiente y secuencias de comprobantes, además de permitir la administración de clientes, proveedores, productos, transportistas.
3. El tercer módulo es el módulo de comprobantes que permite la generación, anulación, búsqueda y visualización de facturas, notas de débito, notas de crédito, guías de remisión y comprobantes de retención, así como la verificación de su validez en la plataforma del SRI a través del uso de sus servicios web.
4. El cuarto módulo corresponde al módulo de reportes que permite la creación, filtrado, búsqueda y listado de información referente a usuarios, clientes, productos, proveedores, ventas, comprobantes emitidos.
5. El último módulo es el módulo de clientes que permite filtrar y visualizar comprobantes electrónicos generados específicamente para cada cliente.

El sistema contará con 4 perfiles: Administrador del sistema, Administrador de sucursal, Usuario y Cliente que tendrán funciones compartidas y otras diferentes descritas a continuación.

- *Administrador del sistema:* Podrá ver, editar, agregar información de cualquier sucursal y generar estadísticas globales del negocio.
- *Administrador de sucursal:* Podrá ver, editar, agregar información de la sucursal asignada.
- *Usuario:* Podrá ver, editar, agregar información referente a comprobantes electrónicos.
- *Cliente:* Podrá únicamente visualizar comprobantes electrónicos generados por el sistema.

El escenario de pruebas, contará de un administrador del sistema, un administrador de sucursal, un usuario y dos clientes, en donde el usuario utilizando el navegador Google Chrome accederá al sistema alojado en Internet para generar comprobantes electrónicos.

El administrador del sistema efectuará cambios en los parámetros del comprobante

y podrá visualizar información de los comprobantes generados en todas las sucursales, así mismo el administrador de sucursal podrá realizar modificaciones en la sucursal en la que se encuentra asignado. Todo esto será realizado en el ambiente de pruebas del servicio web del SRI.

El presente proyecto de titulación tendrá un producto final demostrable.

- **MARCO TEÓRICO**

En esta sección se provee un resumen del fundamento teórico con el que se desarrolló el presente trabajo de titulación. En la sección 1.3.1 se presenta una introducción a metodologías ágiles para el desarrollo de software con especial enfoque en la metodología RAD utilizada para el proyecto y los casos en los que se debería aplicar esta metodología. Seguidamente, en la sección 1.3.2 se describe el enfoque principal de la arquitectura basada en capas. Más tarde, en la sección 1.3.3 se exponen las herramientas necesarias para el desarrollo de la base de datos. Después en la sección 1.3.4 se indican las características del .NET Framework y se presentan los conceptos más relevantes de esta tecnología en el desarrollo de aplicaciones web. Luego, en la sección 1.3.5 se exhiben las tecnologías existentes para el desarrollo Front End tales como HTML, CSS, JavaScript, JQuery, BootStrap. Posteriormente, en la sección 1.3.6 se provee una introducción al control de versiones con GitHub. Finalmente, en la sección 1.3.7 se define el que es la facturación electrónica y los requerimientos para su uso.

1.1.1. METODOLOGÍAS DE DESARROLLO

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de software a escoger las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.” [4].

1.1.1.1. Rapid Application Development (RAD)

Esta metodología de desarrollo ágil se enfoca en minimizar la planificación durante el desarrollo de software. RAD se basa en la realización de prototipos y componentes reutilizables. Al utilizar esta metodología de desarrollo se puede utilizar el conocimiento y lecciones aprendidas sobre el proyecto para dar forma al proceso de diseño o adaptarlo en la dirección requerida. [5]

Es recomendable utilizar RAD cuando los prototipos desarrollados son aceptables para ser implementados en el en el producto final reutilizando los componentes existentes.

A. Proceso cíclico de RAD

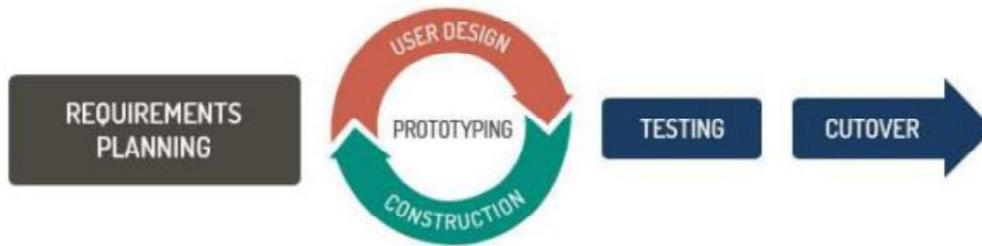


Figura 1.4 Proceso cíclico de RAD

1. *Requisitos de planificación:* En el periodo inicial, los implicados llegan a una resolución sobre el alcance del proyecto y los requerimientos del sistema que se desarrollaran en las siguientes etapas de la metodología.
2. *Diseño de usuario:* Se determina el flujo de datos y arquitectura del proyecto tomando en cuenta las opiniones de los usuarios. De esta manera se pueden crear modelos y prototipos desde el inicio del proyecto. Esta fase se repite para todos los prototipos a desarrollarse.
3. *Construcción rápida:* Una vez iniciado el diseño del usuario y diseño del sistema inicia también la fase de construcción rápida. En esta fase es en la que se realiza la codificación, pruebas e integración de componentes del sistema. Esta etapa se repite cuantas veces sea necesario, ya sea que se requiera un nuevo componente o se solicite un cambio en el sistema.
4. *Transición:* Esta etapa permite a los desarrolladores desplegar los entregables del proyecto de un ambiente de pruebas a un ambiente de producción en caso de ser requerido por el cliente.

B. Beneficios de los prototipos en RAD

Uno de los principales beneficios de utilizar RAD es poder recibir retroalimentación de los usuarios con facilidad y frecuencia ya que estos interactúan directamente con los prototipos desarrollados lo que les permite estar al tanto del proceso de desarrollo.

Algunos de los beneficios de los prototipos durante el ciclo de desarrollo se listan a continuación:

- *Inclusión del usuario:* En un modelo en cascada, los diseñadores y usuarios discuten sobre las características del sistema. A diferencia de este modelo, RAD

permite a los usuarios utilizar los prototipos desarrollados y brindar retroalimentación mientras se continúa con el desarrollo del proyecto.

- *Viabilidad*: RAD permite los desarrolladores evaluar la complejidad para la realización de nuevos componentes, permitiendo de esta manera trabajar en los más complejos al inicio del ciclo de vida del proyecto. Como consecuencia se tiene un producto final más robusto con posibilidad a implementarse mejoras.
- *Reducción y depuración de errores*: RAD permite la detección temprana de errores gracias al versionamiento de los prototipos desarrollados.

C. Ventajas de RAD

RAD presenta múltiples ventajas al ser utilizada como metodología de desarrollo, algunas de las ventajas se describen a continuación:

- *Progreso medible*: Con iteraciones frecuentes, es posible medir fácil y rápidamente los componentes y prototipos para mantenerlos dentro del plazo y presupuesto del proyecto.
- *Generación Rápida de Código Productivo*: RAD permite a los desarrolladores generar prototipos rápidamente, en cambio con el uso de metodologías de desarrollo tradicionales se necesita de más tiempo ya sea semanas o incluso meses debido a la planificación requerida.
- *Compartimento de componentes del sistema*: Los componentes generados deben ser funcionales e independientes para poder ser utilizados en una nueva versión o en caso de ser necesaria alguna actualización los otros componentes diseñados no se verían afectados.
- *Integración temprana del sistema*: A diferencia de las metodologías de desarrollo tradicionales, RAD se permite la integración del sistema casi de inmediato. De esta manera con RAD se identifica cualquier error en la integración del componente con los prototipos ya desarrollados.

D. Desventajas de RAD

- *Requiere Sistemas Modulares*: Dado que todos los componentes deben ser desarrollados dentro de ciclos iterativos, RAD define que cada componente desarrollado sea modular, permitiendo de esta manera intercambiar o alterar los elementos por los miembros del equipo de desarrollo.

- *Dificultad con proyectos a gran escala:* Por lo general las metodologías de desarrollo ágiles no manejan eficientemente proyecto a gran escala y RAD no es la excepción, el uso de RAD en proyectos a gran escala tiende a reducir la flexibilidad y el alcance del proyecto.
- *Demandas de interacción frecuente del usuario:* Obtener información y retroalimentación del usuario temprano y a menudo es sin duda un beneficio desde una perspectiva de diseño, pero puede ser un aspecto negativo, debido a que el equipo tiene que estar dispuesto a comunicarse con el usuario más seguido, además de depender de la disponibilidad del usuario.
- *Depende de desarrolladores expertos:* El uso de la metodología de desarrollo RAD requiere una gran habilidad del equipo de desarrollo debido que estos deben adaptarse rápidamente a la evolución de los componentes y prototipos del proyecto [5].

1.1.2. ARQUITECTURA N-CAPAS

La arquitectura basada en capas se enfoca principalmente en el agrupamiento de funcionalidad relacionada dentro de una aplicación en distintas capas que son colocadas verticalmente una encima de otra, la funcionalidad dentro de cada capa se relaciona con un rol o responsabilidad específica.

El dividir en capas una aplicación, permite la separación de responsabilidades lo que proporciona una mayor flexibilidad y un mejor mantenimiento.

Por ejemplo, en una aplicación de 3 capas presentación, lógica de negocio y acceso a datos, la responsabilidad de la capa de presentación es la de interactuar con el usuario, solicitando y proporcionando la información que el usuario requiere.

La responsabilidad de la capa de lógica de negocio es la de hacer cumplir las reglas del negocio o requerimientos de la aplicación mientras que la responsabilidad de la capa de acceso a datos es la de recuperar y modificar datos del origen de datos [3].

1.1.2.1. Objetivo de la Arquitectura basada en capas

El objetivo principal de la arquitectura basada en capas es el de estructurar una aplicación para soportar requerimientos operacionales complejos y disponer de una buena reusabilidad, escalabilidad, flexibilidad, seguridad, robustez y un fácil mantenimiento. La Figura 1.5 provee un ejemplo de implementación de una nueva capa de presentación la cual puede hacer uso completo de las capas de acceso a datos desarrolladas con anterioridad.

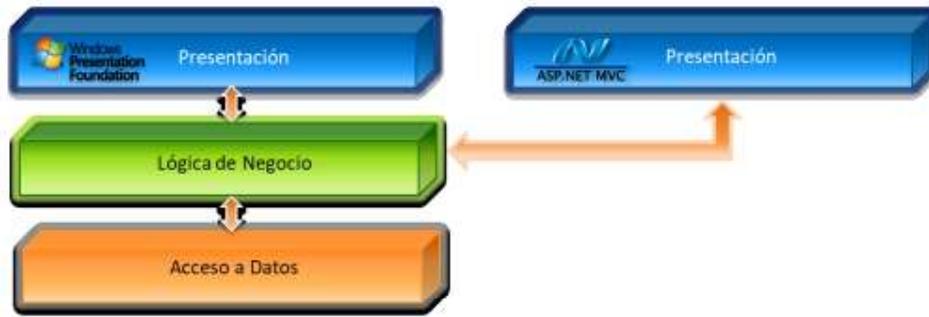


Figura 1.5 Ejemplo nueva capa en arquitectura N-Capas [3]

1.1.2.2. Patrones de diseño aplicados con la arquitectura N-Capas

En la arquitectura N-Capas se trabaja primordialmente con dos patrones de diseño:

- *Patrón de diseño Repository*. La capa de lógica de negocio utiliza la capa de acceso a datos para recuperar la información solicitada por la capa superior. El patrón *repository* es utilizado para separar la capa de lógica de negocio de la lógica que utiliza la capa de acceso a datos para recuperar la información y mapearlos a una clase de la capa de entidades. Esta separación se logra creando una interfaz en la capa de acceso a datos exponiendo los métodos disponibles para esta capa.

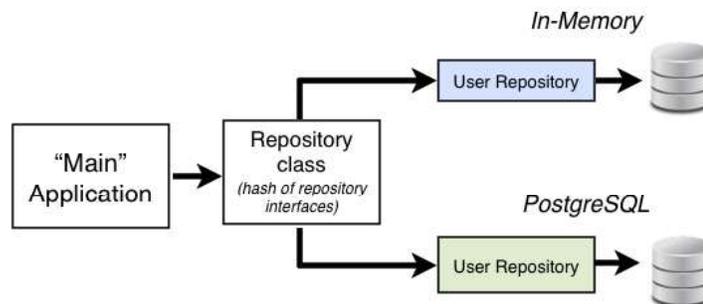


Figura 1.6 Esquema del patrón de diseño repository [6]

- *Patrón de diseño Factory*. La función del patrón de diseño Factory es la de crear un repositorio específico de la fuente de datos utilizada. De esta manera se puede manejar distintas fuentes de datos para la capa de acceso a datos.

1.1.3. BASE DE DATOS

Es un conjunto de datos ordenados bajo un formato estructurado definido por metadatos. Los metadatos también se pueden definir como información sobre los datos que se almacenan. Así mismo, con los metadatos se determina como se almacenará la información en la base de datos.

A lo largo de los últimos años, se han implementado varios modelos de bases de datos para el almacenamiento y la administración de información [7].

1.1.3.1. Modelo Relacional

Está basado primordialmente en los principios matemáticos de la teoría lógica de conjuntos, admite la recuperación fácil de información, impone la integridad de los datos y permite el acceso a los datos almacenados en la estructura establecida independientemente de las aplicaciones que accedan a dichos datos.

La base para este modelo son las relaciones entre las entidades. Una relación es un conjunto de filas y columnas agrupadas en una estructura parecida a una tabla que representa a una entidad compuesta por los datos relacionados. Cada una de las relaciones comprende de una o más columnas que representan a los atributos de la entidad donde cada atributo caracteriza a la entidad de alguna manera. Por ejemplo, en la Figura 1.7 el disco compacto (CD) viene a ser la entidad con atributos *CD_NAME* (el título), *ARTIST_NAME* (nombre del artista), y *COPYRIGHT_YEAR* (año de grabación).

ARTIST_NAME:FullName	CD_NAME:Title	COPYRIGHT_YEAR:Year
Jennifer Warnes	Famous Blue Raincoat	1991
Joni Mitchell	Blue	1971
William Ackerman	Past Light	1983
Kitaro	Kojiki	1990
Bing Crosby	That Christmas Feeling	1993
Patsy Cline	Patsy Cline: 12 Greatest Hits	1988

Figura 1.7 Relación con atributos [7]

En el modelo relacional los datos se almacenan en filas. Una fila es un conjunto de datos en donde los valores forman una instancia para cada atributo definido para la relación. A esta agrupación de valores también se lo designa como registros. Por ejemplo en la Figura 1.7 para el atributo *ARTIST_NAME* se tiene el valor “Jennifer Warnes”, para el atributo *CD_NAME* se tiene el valor “Famous Blue Raincoat” y para el atributo *COPYRIGHT_YEAR* se tiene el valor “1991. Juntos los tres valores forman un registro o fila.

1.1.3.2. SQL

Actualmente las bases de datos relacionales son los mecanismos preferidos para el almacenamiento de información. La mayoría de estas bases de datos son administradas por sistemas para el manejo de bases de datos relacionales o mejor conocidos como RDBMS (*Relational Database Management System*), tales como Microsoft SQL Server, MySQL, Oracle los cuales se basan en el lenguaje SQL (*Structured Query Language*) para la administración de las entidades almacenadas [7].

El lenguaje SQL carece de muchas de las capacidades de programación de la mayoría de los otros lenguajes. Generalmente se maneja SQL como un sublenguaje debido a que se usa en conjunto con otros lenguajes de programación que no están diseñados para administrar directamente datos proporcionando eficiencia en el acceso a la información.

A. Transacciones

Es una sucesión de una o más instrucciones SQL que agrupadas forman una unidad lógica de trabajo. Las instrucciones SQL que van formando la transacción generalmente están fuertemente relacionadas y ejecutan acciones dependientes entre sí. Así mismo, cada instrucción en la transacción realiza una parte de una tarea, pero se necesita que todos concluyan la tarea asignada. Mientras tanto, se agrupan las instrucciones como una sola transacción diciendo al RDBMS que las instrucciones tienen que ser ejecutadas una tras otra. Por lo tanto, para que la base de datos este en un estado consistente todas las instrucciones deben completarse correctamente [8].

1.1.3.3. Procedimientos almacenados

Un procedimiento almacenado permite realizar procesamiento de aplicaciones que están relacionadas con la base de datos en la misma base de datos. Por ejemplo, se podría implementar la lógica para una transferencia interbancaria entre clientes.

Para la creación de un procedimiento se utiliza la sentencia *CREATE PROCEDURE*, así mismo, para eliminar un procedimiento almacenado que ya no es necesario se utiliza la sentencia *DROP PROCEDURE* [8].

Para la creación de un procedimiento almacenado en la sentencia se deben especificar los siguientes parámetros:

- El nombre del procedimiento almacenado.
- El número y los tipos de datos de sus parámetros.
- Los nombres y tipos de datos de cualquier variable local utilizada por el procedimiento.

- La secuencia de instrucciones ejecutadas cuando se llama al procedimiento.

1.1.3.4. Triggers

Un *trigger* es un conjunto de instrucciones cuya activación es causada por modificaciones en los registros de la base de datos. A diferencia de los procedimientos almacenados, un *trigger* no se activa con las instrucciones *CALL* o *EXECUTE*. Así mismo, el *trigger* está asociado con una tabla de la base de datos. Cuando se modifican los datos de la tabla con instrucciones *INSERT*, *DELETE* o *UPDATE*, se desencadena el *trigger*, lo que significa que el RDBMS ejecuta las instrucciones SQL que componen al *trigger* [8].

1.1.3.5. Vistas

Una vista es una consulta SQL que es almacenada permanentemente en la base de datos y tiene un nombre asignado, puede verse como una tabla virtual. El resultado de la ejecución de la consulta almacenada es visible a través de la vista, y SQL permite el acceso a este resultado como si esta fuera una tabla real de la base de datos [8].

Las vistas son parte importante de SQL por múltiples razones como [8]:

- Permiten ajustar la apariencia de las tablas en base a las diferentes perspectivas de cada usuario.
- Permiten restringir el acceso a la información mostrando únicamente ciertas columnas de la tabla a los usuarios asignados.
- Simplifican el acceso a la base de datos presentando la estructura de datos en la forma más natural posible, ocultando consultas demasiado complejas.

1.1.4. .NET FRAMEWORK

.NET es un *framework* de desarrollo con una interfaz de programación para servicios de Windows y APIs. El *.NET framework* es el conjunto de múltiples tecnologías [9]:

- *Los lenguajes .NET*: incluyen Visual Basic, C#, F# y C++, aunque se han creado cientos más.
- *Common Language Runtime (CLR)*: Permite la ejecución de todos los programas desarrollados con .NET, además de proporcionar servicios para las

aplicaciones como comprobación de seguridad, gestión de memoria y optimización.

- *La biblioteca de clases de .NET Framework*: Recopila un conjunto de piezas funcionales pre-compiladas utilizadas por los desarrolladores las cuales se agrupan en conjuntos de diferentes tecnologías, como ADO.NET (tecnología para manejo de bases de datos) y Windows Forms (tecnología para el desarrollo de aplicaciones de escritorio).
- *ASP.NET*: Permite el desarrollo de aplicaciones web creadas con .NET, y tiene compatibilidad completa con la biblioteca de clases .NET. Así mismo, incluye múltiples servicios específicos para web como por ejemplo: almacenamiento de cookies, autenticación segura, soporte para manejo de protocolos de red como HTTP y FTP.
- *Visual Studio*: Cuenta con el soporte completo para el .NET *framework*. Además tiene un amplio conjunto de herramientas y funciones para el desarrollo y depuración de aplicaciones .NET.

.NET brinda una sólida plataforma para el desarrollo de aplicaciones, tales como:

- *Aplicaciones basadas en formularios Windows*: aplicaciones tradicionales de arquitectura cliente servidor.
- *Aplicaciones de consola*: aplicaciones tradicionales tipo DOS¹.
- *Aplicaciones web ASP.NET*: aplicaciones que se ejecutan en un servidor web y se acceden a través de un navegador web.
- *Bibliotecas de componentes*: aplicaciones que sirven como piezas complementarias a otros proyectos o como librerías para desarrolladores.
- *Controles personalizados de Windows*: además de los tipos de aplicaciones descritas también se pueden desarrollar controles propios para aplicaciones basadas en formularios Windows.
- *Controles personalizados web*: además de los tipos de aplicaciones descritas también se pueden desarrollar controles propios para aplicaciones web ASP.NET.

¹ **Disk Operating Systems (DOS)**: sistema operativo que se ejecuta desde una unidad de disco duro.

- *Servicios web*: son sistemas de software desarrollados para dar soporte a la interacción de máquina a máquina que opera a través de una red haciendo uso de protocolos y estándares como HTTP², XML³ y SOAP⁴.

1.1.4.1. Introducción a las tecnologías web de Microsoft

Microsoft ofrece una amplia gama de tecnologías que pueden ser utilizadas para crear aplicaciones web enriquecidas y publicarlas en Intranets y en Internet.

Además de la publicación de las aplicaciones Web, podemos utilizar estas tecnologías para desarrollar y alojar páginas web que ejecuten código tanto en el lado del servidor como en el navegador del usuario [10].

A. Tecnologías de hospedaje

Independientemente de la herramienta que se utilice para construir una aplicación web, es necesario utilizar un servidor web para alojar la aplicación.

Cuando los usuarios visitan un sitio web, el servidor responde generando código HTML y lo entrega al navegador del usuario para su visualización.

El servidor de hospedaje puede consultar una base de datos antes de generar el código HTML y puede realizar otras acciones, como el envío de mensajes de correo electrónico o guardar los archivos subidos por el usuario.

Es necesario probar la funcionalidad de las acciones de los usuarios en un servidor web.

Al crear aplicaciones web utilizando Visual Studio, es posible utilizar el servidor Web IIS Express incorporado a Visual Studio para ejecutar la aplicación, sin embargo, IIS Express no está diseñado para hospedar aplicaciones web en producción, por lo tanto, cuando se termina la construcción de la aplicación web y se encuentre lista para que los usuarios accedan a ella en una Intranet o en Internet, se debe utilizar un servidor web completamente funcional [10].

De las tecnologías de hospedaje del stack de Microsoft existen:

- *Microsoft Internet Information Server (IIS)*: Es una tecnología avanzada de alojamiento web. Es posible instalar servidores IIS en una red local, red perimetral o utilizar servidores IIS alojados por un proveedor de servicios de Internet (ISP⁵). IIS puede alojar cualquier sitio web ASP.NET, PHP o Node.js.

² **Hypertext Transfer Protocol (HTTP)**: protocolo usado para la transferencia de datos por la web.

³ **Extensible Markup Language (XML)**: se usa para definir documentos con un formato estándar que pueda ser leído por cualquier aplicación compatible.

⁴ **Simple Object Access Protocol (SOAP)**: es un método para transferir mensajes o pequeñas cantidades de información a través de Internet.

⁵ **Internet Service Provider (ISP)**: Es la empresa que brinda conexión a Internet a sus clientes.

Es posible escalar IIS para alojar sitios web grandes y concurridos mediante la configuración de granjas de servidores que conecten múltiples servidores IIS, todos dando servicio al mismo tiempo.

- *Microsoft Azure:* Es una plataforma en la nube que ofrece servicios sobre demanda para construir, implementar, hospedar y administrar aplicaciones web a través de centros de datos administrados por Microsoft. Cuando se utiliza los servicios de Microsoft Azure, se debe pagar solo por los recursos que el sitio web consume. Además, la construcción y manejo de infraestructura es manejada de forma automática a medida que el sitio web crece.

B. Tecnologías de ejecución de código

El código escrito para una aplicación web, debe ejecutarse en uno de los siguientes sitios [10].

- *En el servidor web:* El código que se ejecuta en el servidor web tiene acceso completo a los recursos del servidor web y a cualquier base de datos asociadas a él. Puede acceder a la base de datos con rapidez, enviar mensajes de correo electrónico y generar páginas web.
- *En el navegador web del usuario:* El código que se ejecuta en el navegador web del usuario responde rápidamente a las acciones del usuario, tales como clics del ratón, pero está más limitado en lo que puede hacer sin interactuar con el servidor web. Las aplicaciones ASP.NET también pueden utilizar la tecnología AJAX⁶ en la computadora del cliente. Podemos utilizar AJAX para actualizar una pequeña sección de una página HTML en lugar de volver a cargar la totalidad de la página desde el servidor.

1.1.4.2. ASP.NET

Podemos utilizar ASP.NET para desarrollar sitios web altamente funcionales, dinámicos y escalables que utilicen código en el lado del cliente y código en el lado del servidor. Es posible crear diferentes tipos de sitios web ASP.NET [10].

A. Modelos de programación

Cuando se hace uso de ASP.NET para construir una aplicación, se tienen tres estilos de programación. Cada modelo de programación tiene una estructura particular en el entorno de desarrollo y almacena el código en diferentes lugares de la jerarquía web [10].

⁶ **Asynchronous JavaScript And XML (AJAX):** Es una técnica de desarrollo web para crear aplicaciones interactivas.

- *Web Pages*: El modelo de programación ASP.NET *Web Pages* es un modelo de programación simple y fácil de aprender y es adecuado para sitios simples de acceso a datos.

Crear una aplicación web con este modelo es tan sencillo como crear una página HTML y agregar el código necesario para que el motor ASP.NET ejecute del lado del servidor para procesar la solicitud recibida.

- *Web Forms*: Cuando se emplea ASP.NET *Web Forms* para la construcción de un sitio web, se emplea un modelo de programación con controles ricos del lado del servidor y un ciclo de vida de página similar al desarrollo de aplicaciones de escritorio. Los controles integrados incluyen botones, cuadros de texto y grillas para mostrar datos en forma tabular. También es posible agregar controles de terceros o crear controles personalizados. Para responder a las acciones del usuario, se puede asociar a los controles del lado del servidor, controladores de eventos conteniendo código, todos los controles generados en el lado del cliente son mapeados a controles HTML.
- *MVC*: Cuando se emplea ASP.NET MVC para el desarrollo de un sitio web, se separa el código del lado del servidor en tres partes:
 - *Modelo*: Un modelo MVC define un conjunto de clases que representan los tipos de objetos que la aplicación web administra. Los modelos suelen incluir lógica de acceso a datos.
 - *Vistas*: Una vista MVC es un componente que construye las páginas web que conforman la interfaz de usuario de la aplicación web. Los controladores suelen pasar una instancia de una clase del modelo a una vista. La vista muestra las propiedades de la clase modelo.
 - *Controladores*: Un controlador MVC es una clase que se encarga de la interacción del usuario, crea y modifica clases del modelo y selecciona las vistas correspondientes.

Esta separación de código de modelo, vista y controlador garantiza que las aplicaciones MVC tengan una estructura lógica, incluso para los sitios más complejos. También mejora la capacidad de prueba de la aplicación. En última instancia, ASP.NET MVC proporciona un mayor control sobre el código HTML en comparación con *Web Pages* y *Web Forms*.

B. La API ASP.NET

Sea cual sea el modelo de programación elegido, se tiene acceso a las clases de la API ASP.NET. Estas clases están incluidas dentro del espacio de nombres System.Web del .NET Framework y pueden ser utilizadas para implementar funcionalidades web, tales como:

- *Configuración:* Utilizando archivos web.config, es posible configurar la aplicación web independientemente del modelo de programación. Los archivos web.config son archivos XML con etiquetas y atributos específicos que el motor en tiempo de ejecución de ASP.NET reconoce.
- *Autenticación y autorización:* Muchos sitios requieren que los usuarios inicien sesión mediante la introducción de un nombre de usuario y una contraseña, o proporcionando información adicional. Es posible utilizar proveedores de membresía de ASP.NET para autenticar y autorizar a los usuarios y restringir el acceso al contenido. Los proveedores de membresía pertenecen al espacio de nombres System.Web.Security
- *Almacenamiento en cache:* Para generar el código HTML de una página web compleja que puede requerir múltiples consultas de base de datos o llamadas a servicios web externos, ASP.NET puede tardar un prolongado periodo de tiempo. Es posible utilizar almacenamiento en caché, para atenuar este retraso. ASP.NET almacena en memoria caché una página ya generada, para que pueda devolver la misma página a solicitudes posteriores de los usuarios sin tener que generarla nuevamente

C. Compilación del código ASP.NET

Debido a que el código del lado del servidor ASP.NET utiliza el .NET Framework, debe ser escrito en un lenguaje de programación .NET administrado como C# o Visual Basic. Antes de ser ejecutado, este debe ser compilado a código nativo para que el CPU⁷ del servidor pueda procesarlo, este es un proceso de dos etapas:

1. *Compilación a código MSIL*⁸: Cuando se construye un sitio web en Visual Studio, el compilador de ASP.NET crea archivos .dll⁹ con todo el código

⁷ **Central Processing Unit (CPU):** Es la parte de una computadora en la que se encuentran los elementos que sirven para procesar datos.

⁸ **Microsoft Intermediate Language (MSIL):** Es el lenguaje de programación legible por humanos de más bajo nivel en el Common Language Infrastructure y en el .NET Framework

⁹ **Dynamic-Link Library (DLL):** Son archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.

compilado en MSIL. Este código es independiente tanto del lenguaje que se utilizó para escribir la aplicación como de la arquitectura del servidor.

2. *Compilación a código nativo*: Cuando se solicita una página por primera vez, el CLR¹⁰ compila el código MSIL a código nativo del CPU del servidor.

Este proceso de compilación de dos etapas permite a los componentes escritos en diferentes lenguajes trabajar juntos y permite la detección de muchas fallas en tiempo de compilación.

1.1.5. TECNOLOGÍAS PARA DESARROLLO FRONT END

1.1.5.1. Hyper Text Markup Language (HTML)

El lenguaje de marcado y etiquetado HTML es un predecesor del lenguaje SGML (*Standard Generalized Markup Language*). HTML es un lenguaje válido para la definición de documentos estructurados y etiquetas para representar estos documentos [12].

La estructura de los documentos HTML están organizados de manera estricta. Cada uno de estos documentos está definido por etiquetas específicas, donde algunas de estas son obligatorias y otras opcionales. Las etiquetas HTML funcionan mediante un mecanismo clave y valor que da un atributo al elemento HTML (por ejemplo, `<html lang="es">`). Casi todas estas etiquetas se usan en pares, una etiqueta para la apertura y otra para el cierre en donde los atributos y valores se declaran entre dichas etiquetas [11].

La Figura 1.8 presenta un ejemplo de la codificación básica requerida para una página HTML. Los elementos esenciales son las etiquetas DOCTYPE, html, head y body.

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="iso-8859-1">
5     <meta name="description" content="Ejemplo de HTML5">
6     <meta name="keywords" content="HTML5, CSS3, JavaScript">
7     <title>Este texto es el título del documento</title>
8     <link rel="stylesheet" href="misestilos.css">
9   </head>
10  <body>
11    <header>
12      <h1>Este es el título principal del sitio web</h1>
13    </header>
14    <p>Este es un parrafo dentro del sitio web</p>
15  </body>
16 </html>
```

Figura 1.8 Ejemplo de código HTML

¹⁰ **Common Language Runtime (CLR)**: Es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET

HTML5 es una redefinición completa del lenguaje HTML utilizado para la construcción de sitios y aplicaciones web. Se ha redefinido completamente dado el avance actual de este tipo de tecnologías donde se combina dispositivos móviles, trabajos en red y *cloud computing*.

HTML5 proporciona básicamente tres funcionalidades: estructura del contenido, estilo aplicado a ese contenido y funcionalidad. HTML5 es considerado como la combinación de HTML, CSS y JavaScript. Estas tecnologías dependen una de otra trabajando como una única unidad organizada bajo la estandarización de HTML5. HTML se encarga de la estructura del contenido, CSS de la presentación de esa estructura y contenido en pantalla y JavaScript proporciona la capacidad de realizar elementos dinámicos.

A pesar de toda esta integración, la estructura es la parte esencial del documento. Esta estructura proporciona los elementos necesarios para ubicar el contenido de sitios o aplicaciones web. Con la variedad de dispositivos que acceden a Internet y las múltiples interfaces que interactúan con la web la estructura se convierte en una parte fundamental del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio [11].

1.1.5.2. Cascading Style Sheets (CSS)

El lenguaje de hojas de estilo en cascada es un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. Inicialmente dentro de las etiquetas HTML se proporcionaba los estilos para cada elemento, pero a medida que este lenguaje evolucionó, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo satisfacer los requerimientos de diseñadores. En consecuencia, CSS pronto fue adoptado como la forma de separar la estructura de la presentación. Desde entonces, CSS ha crecido y ganado importancia individualmente, pero siempre desarrollándose en paralelo a HTML, enfocado en las necesidades de diseñadores.

CSS3 sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web, aunque oficialmente se trate de dos tecnologías completamente separadas.

Mediante reglas definidas en un documento CSS se puede modificar cualquier característica visual que posea un elemento, para poder indicar que elemento se requiere modificar se utilizan selectores y para indicar que característica es la que se va a modificar se utiliza la declaración.

La manera más directa de realizar la vinculación entre el documento HTML y el documento CSS es incrustándolo dentro de la cabecera del documento HTML como se puede observar en el Segmento de código 1.1. La razón para ubicarlo en la cabecera es porque así los CSS serán reconocidos antes de los demás elementos de la página [13].

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title> Ejemplo CSS </title>
  <!-- En style se ubica el código CSS -->
  <style>
    /*Se aplica el estilo a body -->*/
    body {
      /*Características de los elementos*/
      font-family: georgia, "times new roman", serif;
      color: blue;
      padding-left: 11em;
      background-color: #000000;
    }
  </style>
</head>
<body>
  <!-- Header 1 con el texto de ejemplo -->
  <h1> Letra de ejemplo </h1>
</body>
</html>
```

Segmento de código 1.1 Ejemplo de inserción de código CSS en un documento HTML.

1.1.5.3. JavaScript

“JavaScript es un lenguaje ligero e interpretado orientado a objetos con funciones de primera clase, más conocido como el lenguaje de *script* para páginas web [14]”. Este lenguaje de programación se utiliza en varios entornos de desarrollo web, como *node.js*, ya que es un lenguaje de programación con características de multiparadigma, que a su vez es dinámico.

JavaScript se basa en el estándar *ECMAScript*, dicho estándar en su versión 5.1 es soportado por todos los exploradores web modernos desde el año 2012, y los exploradores antiguos soportan por lo menos la versión 3, por lo que cualquier navegador puede interpretar JavaScript.

El modo de trabajo de JavaScript se basa en el acceso al documento y sus elementos mediante DOM¹¹. En el Segmento de Código 1.4 se observa la manera de acceso a

¹¹ **Document Object Model:** Es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML.

un documento nuevo, la creación de un nuevo elemento y luego el acceso ha dicho elemento mediante su identificador.

1.1.5.4. **BootStrap**

BootStrap es un *framework* constituido por archivos CSS, archivos con tipos de letra y archivos JavaScript que facilitan la creación de sitios web responsivos e interactivos que se adaptan a los distintos tamaños de dispositivos.

Desde que BootStrap fue lanzado en agosto de 2011, ha tomado una gran popularidad. Ha evolucionado de ser un proyecto hecho totalmente a base de CSS hasta llegar a hospedar complementos JavaScript e iconos que van de la mano con formularios y botones. Basicamente, permite un diseño web responsivo ya que cuenta con un sistema *Grid* robusto con 12 columnas con un ancho total de 750px, 970px o 1170px dependiendo del ancho de la pantalla del dispositivo.

1.1.5.5. **El sistema Grid de BootStrap**

BootStrap incluye un sistema *Grid* responsivo, diseñado de manera primaria para dispositivos móviles. El sistema *Grid* escala apropiadamente hasta 12 columnas a medida que crece el tamaño de la pantalla o *viewport* del dispositivo.

Es posible utilizar el sistema *Grid* de BootStrap para crear el diseño de una página web a través de una serie de filas y columnas donde se coloca el contenido.

El sistema *Grid* de BootStrap funciona de la siguiente manera [15]:

- Las filas deben ser colocadas dentro de un contenedor que utilice la clase *container* para un tamaño fijo (*fixed-width*) del *Grid* de 750px, 970px o 1170px dependiendo del ancho del dispositivo.
- Las filas también pueden estar contenidas en un contenedor que utilice la clase *container-fluid* para un ancho completo (*full-width*). En este caso siempre ocuparán el tamaño completo del dispositivo y no se limitarán a un tamaño máximo de *Grid* como en el caso de la clase *container*.
- Las filas se utilizan para agrupar a las columnas horizontalmente.
- El contenido debe ser colocado dentro de las columnas y únicamente las columnas deben ser hijas inmediatas de las filas.
- BootStrap contiene la clase CSS predefinida *.row* para definir una fila.
- BootStrap contiene clases predefinidas con prefijo *.col-* para definir columnas que junto a la clase *.row* se encuentran disponibles para crear diseños *Grid* de manera rápida y fácil.

- De manera predeterminada, el sistema *Grid* de Bootstrap divide una fila en 12 partes. El ancho de cada parte dependerá del tamaño de la pantalla.
- Las columnas del *Grid* definen su ancho especificando cuántas de las 12 partes de la fila abarcan. Por ejemplo, si se requiere que una fila sea dividida en tres columnas de anchos iguales cuando la página sea desplegada en una página de 1024px de ancho, debe utilizarse la clase `.col-md-4` para cada una de las 3 columnas. El valor 4 en la clase CSS `.col-md-4` indica que la columna ocupará 4 de las 12 partes en las que Bootstrap divide la fila.
- Bootstrap separa las columnas aplicando *padding*¹². Cada columna tiene un *padding* de 15px a la izquierda y de 15px a la derecha.

Tomando en cuenta las 12 partes en las que Bootstrap divide una fila, la **Tabla 1.1** muestra los prefijos de las clases CSS que se utilizan para definir columnas dependiendo del ancho del dispositivo. Se muestra también el ancho total de cada columna y el ancho disponible para el contenido de la columna quitando el espacio del *padding* de la izquierda y derecha de la columna.

Tabla 1.1 Prefijos de las clases CSS para Bootstrap [16].

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

1.1.6. GITHUB

GitHub es un servicio para alojamiento de repositorios de software gestionados por el sistema de control de versiones Git. Por tanto, Git es algo más general que sirve para

¹² **Padding:** Es una propiedad CSS que establece el espacio de relleno requerido por todos los lados de un elemento.

controlar el estado de un desarrollo a lo largo del tiempo, mientras que GitHub es algo más particular.

GitHub es una plataforma en línea construida para promover el alojamiento de código, el control de versiones y la colaboración entre personas que trabajan en un proyecto común. Los proyectos se pueden manejar desde cualquier lugar a través de la plataforma.

La plataforma GitHub ofrece manejo de proyectos tanto para proyectos de código abierto como para proyectos privados. Las características ofrecidas en cuanto a la dirección del proyecto del equipo incluyen páginas de GitHub *Flow* y GitHub. Estas funciones facilitan a los equipos con implementaciones regulares el manejo del flujo de trabajo.

Las principales características de la plataforma GitHub se listan a continuación [17]:

1. *Creación de un repositorio*: Un repositorio es un almacenamiento para todos los archivos de un proyecto. Incluye el historial de revisiones. Estos repositorios pueden ser públicos o privados. Los repositorios públicos son adecuados para proyectos de software de código abierto, mientras que los repositorios privados son principalmente adecuados para proyectos privados o de código cerrado.

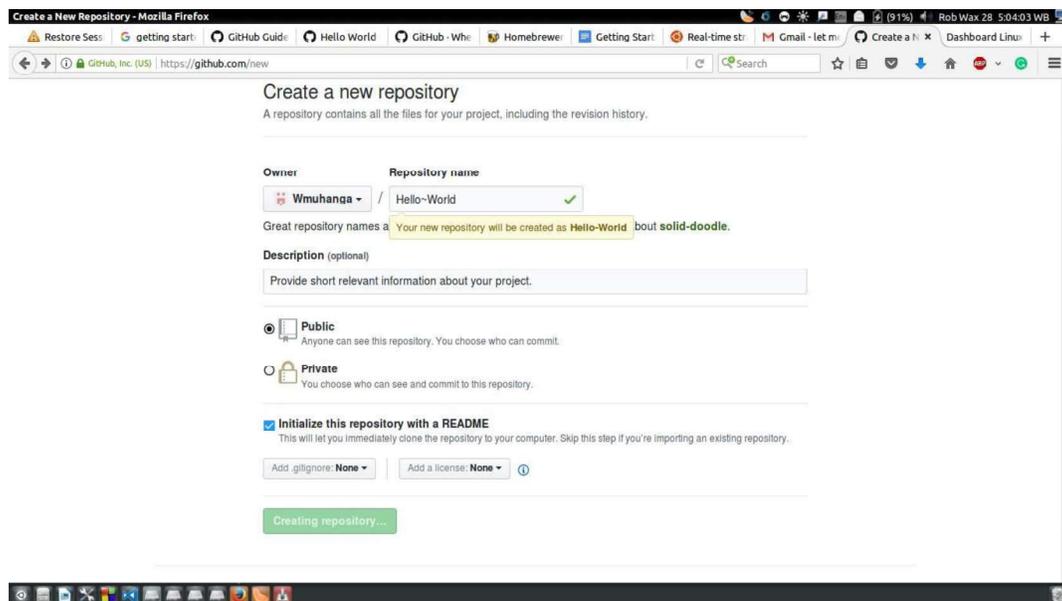


Figura 1.9 Creación de un repositorio en GitHub.

2. *Adición de una rama*: En GitHub, las ramas son una forma de trabajar simultáneamente en varias versiones de un único repositorio. De forma predeterminada, cualquier repositorio único creado se asigna a una rama llamada MASTER y se considera la rama final. En GitHub, las ramas son útiles

en la experimentación y edición de repositorios antes de comprometerlos con el maestro (rama final).

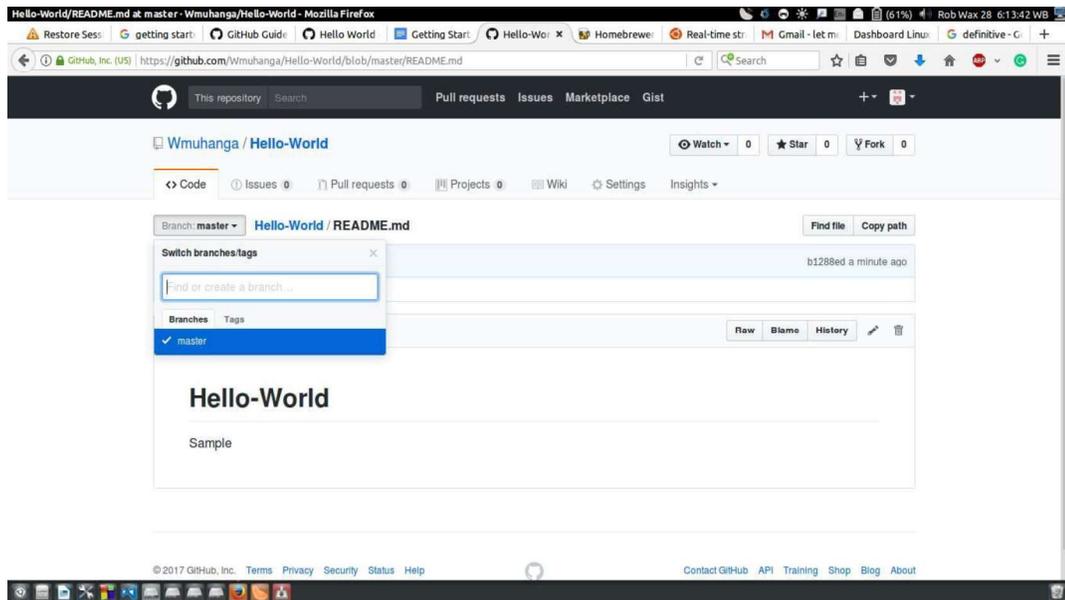


Figura 1.10 Creación de una nueva rama en GitHub.

3. *Hacer y comprometer cambios*: Este paso proporciona directrices sobre cómo se realizan los cambios en un repositorio y se guardan. En GitHub, *commits* es la palabra utilizada para referirse a los cambios guardados. Cada *commit* también está asociado con un mensaje de confirmación que contiene un historial de los cambios guardados y por qué se realizaron cambios particulares.

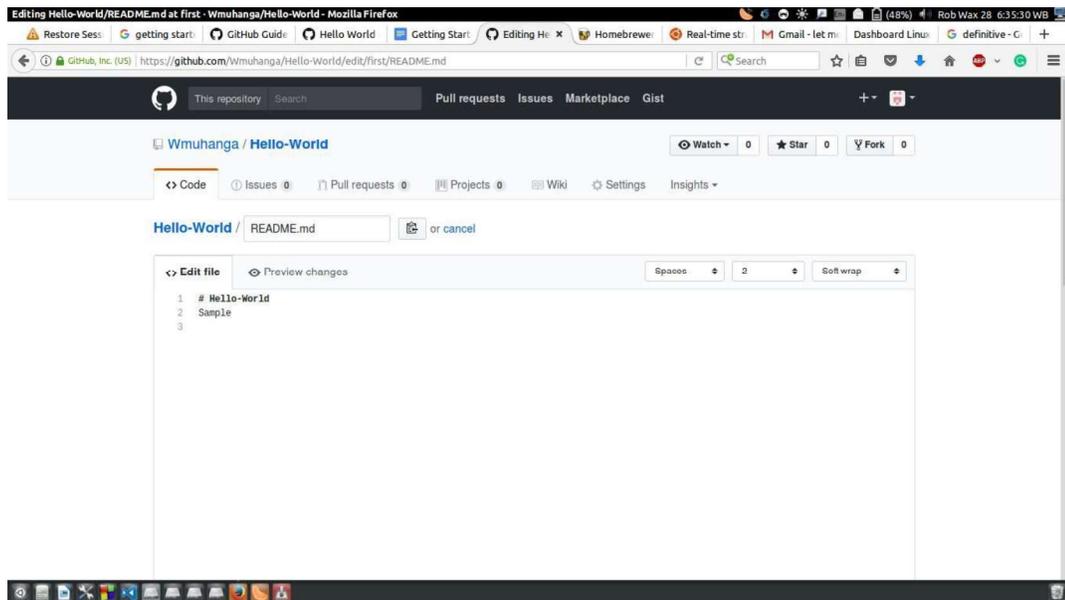


Figura 1.11 Realizar cambios en un repositorio.

4. *Apertura de una solicitud de extracción:* La solicitud de extracción es una función que permite a un contribuyente del proyecto proponer y solicitar a alguien que revise y fusione ciertos cambios en el proyecto. Las solicitudes de extracción también muestran la diferencia (diffs) de varias ramas. Los cambios, adiciones y sustracciones se muestran normalmente en colores rojo y verde.

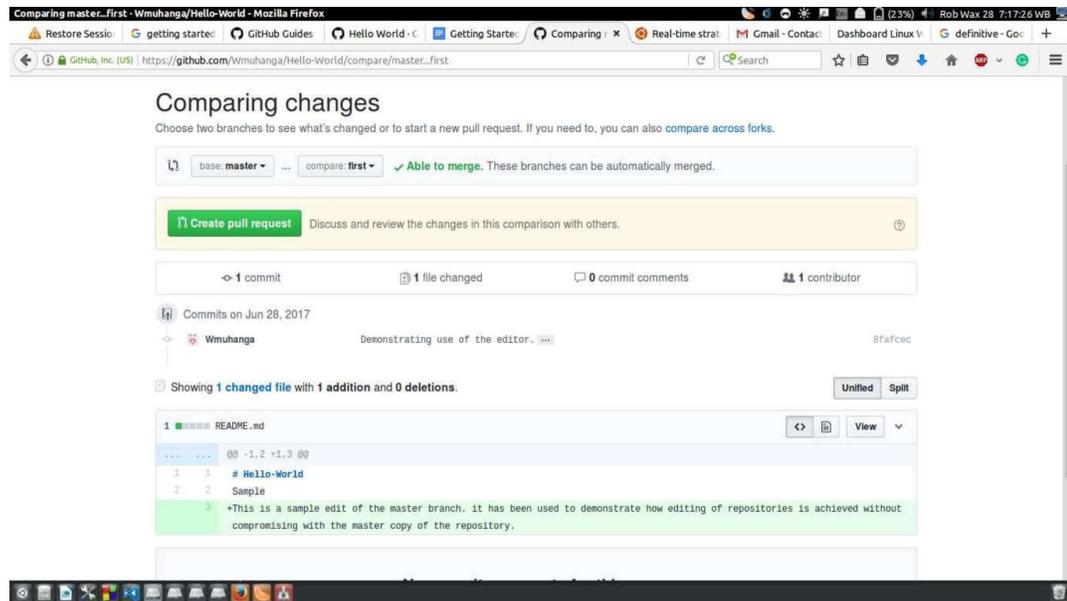


Figura 1.12 Revisión de cambios en repositorio GitHub.

1.1.7. FACTURACIÓN ELECTRÓNICA

La facturación electrónica es otra forma de emisión de comprobantes de venta que cumple con los requisitos legales y reglamentarios exigibles para su autorización por parte del SRI, garantizando la autenticidad de su origen e integridad de su contenido, ya que incluye en cada comprobante la firma electrónica del emisor.

Un comprobante electrónico tendrá validez legal siempre que contenga la firma electrónica del contribuyente emisor.

1.1.7.1. Beneficios de la facturación electrónica

- Tiene la misma validez que los documentos físicos.
- Reducción de tiempos de envío de comprobantes.
- Ahorro en el gasto de papelería física y su archivo.
- Contribuye al medio ambiente, debido al ahorro de papel y tintas de impresión.
- Mayor seguridad en el resguardo de los documentos.
- Menor probabilidad de falsificación.

- Procesos administrativos más rápidos y eficientes.

Documentos que pueden emitirse electrónicamente:

- Facturas
- Notas de crédito
- Notas de débito
- Comprobantes de retención
- Guías de remisión

1.1.7.2. Servicios Web expuestos por el SRI

Para la validación de comprobantes electrónicos a través de Internet el SRI expone servicios web para la autorización en línea de comprobantes electrónicos estandarizados a través de seguridad y certificados SSL¹³.

Para el intercambio de información entre el contribuyente y la Administración Tributaria, es requisito indispensable que el contribuyente cuente con acceso a la red de Internet con un ancho de banda de al menos 256Kbps.

Para garantizar que la conexión es segura se empleará certificados digitales SSL, es decir que el SRI emitirá un certificado válido cuando se realice la petición de los servicios web.

Existen dos tipos de ambientes disponibles para la invocación de los servicios web publicados por la Administración Tributaria:

- Uno es el ambiente de pruebas en donde cada contribuyente certificará que se aplicación funcione correctamente con cada tipo de comprobante electrónico.
- El segundo es para el ambiente de producción, al cual cada contribuyente deberá acceder una vez que ha realizado las pruebas y esté seguro que su aplicación funciona correctamente.

El sistema de autorización de documentos electrónicos soporta un proceso a través de un ambiente computacional seguro, que brinda alta disponibilidad y alto rendimiento, que opta por utilizar la infraestructura necesaria para brindar el servicio a la ciudadanía que realizan transferencias de bienes o prestación de servicio.

¹³ **Secure Sockets Layer:** Son los protocolos de seguridad de uso común que establecen un canal seguro entre dos ordenadores conectados a través de Internet o de una red interna.

La manera correcta de consumir las direcciones URL de los servicios web, es de manera asíncrona; es decir una vez que el contribuyente envíe el comprobante al servicio web y obtenga la respuesta de “RECIBIDA” en un tiempo de hasta 3 segundos, procederá a consumir la segunda dirección URL de autorización mediante la clave de acceso del comprobante, para obtener el resultado de autorización o rechazo del mismo.

Las URLs para el envío y autorización de comprobantes electrónicos en el ambiente de pruebas del SRI son [19]:

- <https://celcer.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantes?wsdl>
- <https://celcer.gob.ec/comprobantes-electronicos-ws/AutorizacionComprobantes?wsdl>

Las URLs para el envío y autorización de comprobantes electrónicos en el ambiente de producción del SRI son [19]:

- <https://cel.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantes?wsdl>
- <https://cel.gob.ec/comprobantes-electronicos-ws/AutorizacionComprobantes?wsdl>

1.1.7.3. Procedimiento para el envío de comprobantes electrónicos al SRI

Para realizar la facturación electrónica los contribuyentes deben requerir autorización para emisión de facturas de manera electrónica a través del portal del SRI, además deberán adquirir un certificado de firma digital en las entidades certificadoras.

El procedimiento a seguir para generar comprobantes electrónicos es el siguiente:

1. El emisor genera el comprobante electrónico en base al esquema XML del SRI.
2. Una vez generado el comprobante electrónico se firma con el certificado digital del emisor.
3. El emisor envía el comprobante firmado a la base de datos del SRI y cuando llega la información se valida y autoriza el comprobante.
4. El receptor puede verificar los comprobantes electrónicos autorizados mediante la página web del emisor, correo electrónico, otros medios o a través de la página web del SRI.

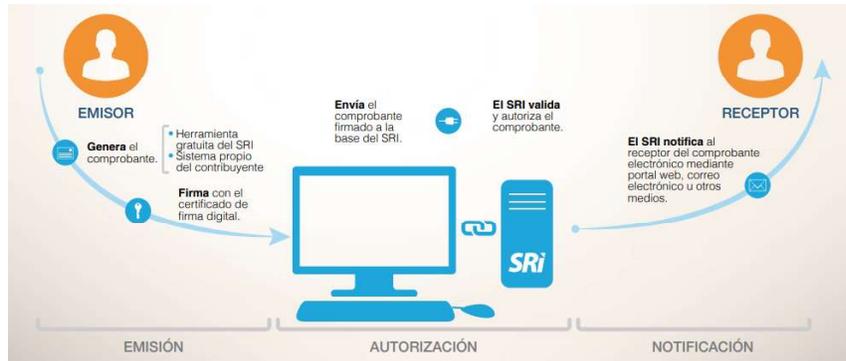


Figura 1.13 Proceso de autorización de comprobantes electrónicos [17].

Las claves en el esquema en línea estarán compuestas de 49 caracteres numéricos. La herramienta o sistema a utilizar por el contribuyente emisor deberá generar de manera automática la clave de acceso, la misma que otorgará el carácter único a cada uno de los comprobantes y servirá para que el SRI indique si el comprobante está autorizado o no.

No.	Descripción de campo	Tipo de campo	Formato	Longitud	Requisito	Etiqueta o tag en archivo XML
1	Fecha de Emisión	Numérico	ddmmaaaa	8	Obligatorio	<claveAcceso>
2	Tipo de Comprobante		Tabla 4	2		
3	Número de RUC		1234567890001	13		
4	Tipo de Ambiente		Tabla 5	1		
5	Serie		001001	6		
6	Número del Comprobante (secuencial)		000000001	9		
7	Código Numérico		Numérico	8		
8	Tipo de Emisión		Tabla 2	1		
9	Dígito Verificador (módulo 11)		Numérico	1		

Figura 1.14 Campos requeridos para creación de clave de acceso [18].

2. METODOLOGÍA

En este capítulo se detallan 3 procesos fundamentales: el diseño del sistema, la codificación de los módulos y la implementación del prototipo. La fase de diseño comprende la arquitectura del prototipo, análisis de requerimientos y modelado del sistema. La fase de codificación, que se llevó a cabo siguiendo la metodología RAD enfocada en los requerimientos del sistema. La fase de implementación corresponde a la puesta en marcha del prototipo en un servidor en Internet, para finalmente ser sometido a pruebas.

2.1. DISEÑO DEL PROTOTIPO

En esta sección se define la arquitectura del prototipo en base a la arquitectura N-Capas, se definen los requerimientos funcionales y no funcionales de acuerdo con el alcance definido, se modelan la base de datos y la estructura del sistema mediante diagramas.

2.1.1. ANÁLISIS DE REQUERIMIENTOS

Para la obtención de los requerimientos del sistema prototipo, se realizó una entrevista disponible en el ANEXO 1 y seguimiento presencial del proceso de facturación de la ferretería MADERS HERRAJES.

Dicha entrevista y seguimiento al proceso de facturación en conjunto con el alcance definido para el trabajo de titulación arrojaron los requerimientos del sistema, mismos que se listan a continuación:

- Permitir generación de comprobantes electrónicos.
- Consumir los servicios web del SRI para la autorización de comprobantes.
- Permitir creación, modificación y lectura de usuarios.
- Permitir creación, modificación y lectura de sucursales.
- Permitir creación, modificación y lectura de emisores.
- Permitir carga de certificados digitales.
- Permitir operaciones CRUD en categorías de productos.
- Permitir operaciones CRUD en productos.
- Permitir creación, modificación y lectura de impuestos establecidos por el SRI.
- Permitir creación, modificación y lectura de formas de pago establecidas por el SRI.
- Permitir creación, modificación y lectura de transportistas.
- Permitir operaciones CRUD de proveedores.
- Permitir creación, modificación y lectura de costos del negocio.
- Permitir creación, modificación y lectura de gastos del negocio.
- Permitir generación de reportes en base a los comprobantes electrónicos generados.

2.1.2. ESPECIFICACIÓN DE REQUERIMIENTOS

En esta sección se especifican requerimientos funcionales y no funcionales para el prototipo.

2.1.2.1. REQUERIMIENTOS FUNCIONALES

Describen los servicios que prestará la aplicación. Se han definido los siguientes requerimientos funcionales:

- El sistema permite al administrador crear, leer, modificar e inactivar usuarios.
- El sistema permite al administrador y usuario modificar la información de su perfil y contraseña.
- El sistema permite al administrador de sucursal modificar la información referente a comprobantes de la sucursal a la cual está asignado.
- El sistema permite al administrador, administrador de sucursal y usuario generar comprobantes electrónicos con la sucursal en la que se encuentre asignado.
- El sistema permite al Administrador, Administrador de sucursal, usuario y cliente visualizar comprobantes electrónicos generados en el sistema y enviarlos mediante correo electrónico.
- El sistema permite al Administrador generar reportes consolidados de todas las sucursales en base a la información de los comprobantes generados.
- El sistema permite al administrador de sucursal generar reportes únicamente de la sucursal en la que se encuentra asignado.
- El sistema permite al administrador, administrador de sucursal y usuario agregar, leer, modificar clientes.
- El sistema permite al administrador, administrador de sucursal y usuario agregar, leer, modificar transportistas.
- El sistema permite al administrador, administrador de sucursal y usuario agregar, leer, modificar y eliminar proveedores.
- El sistema permite al administrador, administrador de sucursal, usuario verificar el estado de comprobantes a través del consumo de los servicios web del SRI.
- El sistema permite al administrador, administrador de sucursal agregar, leer, modificar costos en la sucursal asignada.

- El sistema permite al administrador, administrador de sucursal agregar, leer, modificar gastos en la sucursal asignada.
- El sistema permite al administrador, administrador de sucursal y usuario agregar, leer, modificar y eliminar categorías de productos.
- El sistema permite al administrador, administrador de sucursal y usuario agregar, leer, modificar y eliminar productos.

2.1.2.2. REQUERIMIENTOS NO FUNCIONALES

Describen propiedades del sistema como seguridad y disponibilidad. Se han definido los siguientes requerimientos no funcionales:

- Para el ingreso al sistema se utilizará un nombre de usuario y contraseña.
- Para el almacenamiento de credenciales en el navegador se utilizarán cookies cifradas con seguridad de Windows.
- Las contraseñas de los usuarios estarán almacenadas en la base de datos mediante un HASH utilizando el algoritmo SHA512.
- Las contraseñas no pueden estar en texto visible.
- No se pueden registrar usuarios duplicados.
- La validación de los clientes se realiza mediante el número de identificación registrado en el sistema.
- El cliente puede únicamente ver los comprobantes generados con su identificación y conociendo el número del comprobante, numero de autorización o clave de acceso del comprobante electrónico.
- Las contraseñas de los certificados digitales se almacenarán encriptadas en la base de datos.
- Para la carga de archivos al servidor se permitirán únicamente los archivos del formato especificado en el formulario.
- El tiempo de espera al servicio web del SRI tendrá una duración máxima especificada en los datos del emisor.
- Para el ingreso al sistema el cliente deberá proporcionar únicamente el número de identificación.

2.1.3. HISTORIAS DE USUARIO

Las historias de usuario son utilizadas para la descripción de requisitos de software, están clasificadas de acuerdo con los casos de uso establecidos para el sistema anteriormente.

2.1.3.1 FORMATO

El formato de las historias de usuario se muestra en la Tabla 2.1 y contiene los siguientes campos:

- *ID*: identificación única de la historia de usuario.
- *Rol*: indica el usuario al que está dirigida la funcionalidad
- *Nombre de historia de usuario*: título descriptivo de la historia de usuario.
- *Prioridad*: nivel de importancia de la historia de usuario, se han clasificado en: Alta, Media y Baja.
- *Descripción*: se describe los objetivos de la historia de usuario.
- *Criterios de aceptación*: indica la manera en que el usuario confirma que se ha cumplido el objetivo de la historia de usuario.

Tabla 2.1 Formato de historia de usuario

Historia de Usuario					
ID		Rol		Prioridad	
Nombre historia:					
Descripción:					
Criterios de aceptación:					

2.1.3.2. DETALLE DE HISTORIAS DE USUARIO

Se presentan las historias de usuario definidas según los requerimientos analizados.

Tabla 2.2 HU Autenticación en el sistema

Historia de Usuario					
ID	1	RoI	Administrador/ Administrador de sucursal/ Usuario/ Cliente	Prioridad	Alta
Nombre historia: Autenticarse en el sistema					
Descripción: El usuario podrá autenticarse en el sistema siempre y cuando el usuario este en estado activo mediante el ingreso de sus credenciales. En caso de que el usuario tenga el rol cliente se ingresará únicamente la identificación para el acceso al sistema.					
Criterios de aceptación: El sistema permite autenticarse y acceder a la aplicación.					

Tabla 2.3 HU Modificar perfil personal

Historia de Usuario					
ID	2	RoI	Administrador/ Administrador de sucursal/ Usuario	Prioridad	Baja
Nombre historia: Modificar perfil personal					
Descripción: El usuario podrá modificar la información personal del perfil asignado en el sistema.					
Criterios de aceptación: El sistema permite modificar la información del usuario.					

Tabla 2.4 HU Administración de usuarios

Historia de Usuario					
ID	3	Rol	Administrador	Prioridad	Alta
Nombre historia: Administración de usuarios del sistema					
<p>Descripción: El usuario podrá administrar los usuarios del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información del usuario exceptuando la contraseña 2. Crear nuevos usuarios del sistema. 3. Actualizar la información del usuario incluyendo la contraseña y si el usuario está en estado activo o inactivo. 					
Criterios de aceptación: El sistema permite administrar los usuarios del sistema					

Tabla 2.5 HU Administración de sucursales

Historia de Usuario					
ID	4	Rol	Administrador	Prioridad	Alta
Nombre historia: Administración de sucursales del sistema					
<p>Descripción: El usuario podrá administrar las sucursales del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de las sucursales del sistema. 2. Crear nuevas sucursales para el sistema. 3. Actualizar la información de las sucursales existentes. 4. Eliminar una sucursal del sistema si esta no tiene comprobantes generados. 					
Criterios de aceptación: El sistema permite administrar las sucursales del sistema					

Tabla 2.6 HU Administración de emisores

Historia de Usuario					
ID	5	Rol	Administrador	Prioridad	Alta
Nombre historia: Administración de emisores del sistema					
<p>Descripción: El usuario podrá administrar los emisores del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los emisores del sistema. 					

<ol style="list-style-type: none"> 2. Crear nuevos emisores para el sistema. 3. Actualizar la información de emisores existentes. 4. Eliminar un emisor del sistema si éste no tiene sucursales en las que se utilice el mismo para la generación de comprobantes.
<p>Criterios de aceptación: El sistema permite administrar los emisores del sistema</p>

Tabla 2.7 HU Administración de certificados digitales

Historia de Usuario					
ID	6	Rol	Administrador	Prioridad	Alta
Nombre historia: Administración de certificados digitales					
<p>Descripción: El usuario podrá administrar los certificados digitales del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los certificados digitales del sistema. 2. Ingresar nuevos certificados al sistema. 3. Actualizar la información de certificados existentes excepto la clave del certificado digital. <p>Los certificados digitales se almacenarán en el servidor y en la base de datos el path¹⁴ al archivo.</p>					
Criterios de aceptación: El sistema permite administrar los certificados digitales					

Tabla 2.8 HU Administración de parámetros de red

Historia de Usuario					
ID	7	Rol	Administrador	Prioridad	Alta
Nombre historia: Administración de parámetros de configuración					
<p>Descripción: El usuario podrá administrar los parámetros de configuración del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los parámetros de configuración del sistema. 2. Ingresar nuevos parámetros de configuración de red. 3. Actualizar la información de parámetros de configuración registrados en el sistema. 					

¹⁴ **Path:** Es una variable de entorno de los sistemas operativos POSIX y los sistemas de Microsoft, en ella se especifican las rutas en las cuales el intérprete de comandos debe buscar los programas a ejecutar.

Únicamente se podrá habilitar un parámetro de configuración a la vez ya que este se utilizará para el consumo de servicios web.

Criterios de aceptación: El sistema permite administrar los parámetros de configuración.

Tabla 2.9 HU Administración de impuestos manejados por el SRI

Historia de Usuario					
ID	8	RoI	Administrador	Prioridad	Alta
Nombre historia: Administración de impuestos manejados por el SRI					
Descripción: El usuario podrá administrar los impuestos manejados por el SRI con las operaciones descritas a continuación: <ol style="list-style-type: none">1. Leer información de los impuestos registrados en el sistema.2. Ingresar nuevos impuestos.3. Actualizar la información de impuestos registrados en el sistema.					
Criterios de aceptación: El sistema permite administrar los impuestos manejados por el SRI.					

Tabla 2.10 HU Administración de formas de pago manejados por el SRI

Historia de Usuario					
ID	9	RoI	Administrador	Prioridad	Alta
Nombre historia: Administración de formas de pago manejados por el SRI					
Descripción: El usuario podrá administrar las formas de pago manejadas por el SRI con las operaciones descritas a continuación: <ol style="list-style-type: none">1. Leer información de las formas de pago registradas en el sistema.2. Ingresar nuevas formas de pago.3. Actualizar la información de impuestos registradas en el sistema.					
Criterios de aceptación: El sistema permite administrar las formas de pago manejadas por el SRI.					

Tabla 2.11 HU Administración de tipos de comprobantes

Historia de Usuario					
ID	10	Rol	Administrador	Prioridad	Alta
Nombre historia: Administración de tipos de comprobantes					
<p>Descripción: El usuario podrá administrar los tipos de comprobantes del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los tipos de comprobantes registrados. 2. Ingresar nuevos tipos de comprobantes. 3. Actualizar la información de tipos de comprobantes y el número de secuencia en el que se encuentran por sucursal. 					
Criterios de aceptación: El sistema permite administrar los tipos de comprobantes del sistema.					

Tabla 2.12 HU Administración de clientes

Historia de Usuario					
ID	11	Rol	Administrador/ Administrador de sucursal/ Usuario	Prioridad	Media
Nombre historia: Administración de clientes					
<p>Descripción: El usuario podrá administrar los clientes del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los clientes registrados. 2. Ingresar nuevos clientes. 3. Actualizar la información de los clientes y el estado en el que se encuentran. 4. Eliminar clientes del sistema en caso de que no se hayan generado comprobantes con el mismo. <p>Existen diferentes tipos de clientes registrados para el SRI.</p> <ul style="list-style-type: none"> • Cliente • Sujeto Retenido • Destinatarios 					
Criterios de aceptación: El sistema permite administrar los clientes del sistema.					

Tabla 2.13 HU Administración de transportistas

Historia de Usuario					
ID	12	Rol	Administrador/ Administrador de sucursal/ Usuario	Prioridad	Media
Nombre historia: Administración de transportistas					
<p>Descripción: El usuario podrá administrar los transportistas del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 5. Leer información de los transportistas registrados. 6. Ingresar nuevos transportistas. 7. Actualizar la información de los transportistas. 8. Eliminar transportistas del sistema en caso de que no se hayan generado guías de remisión con el mismo. 					
Criterios de aceptación: El sistema permite administrar los transportistas del sistema.					

Tabla 2.14 HU Administración de proveedores

Historia de Usuario					
ID	13	Rol	Administrador/ Administrador de sucursal	Prioridad	Baja
Nombre historia: Administración de proveedores					
<p>Descripción: El usuario podrá administrar los proveedores del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los proveedores registrados. 2. Ingresar nuevos proveedores. 3. Actualizar la información de los proveedores. 4. Eliminar proveedores del sistema. 					
Criterios de aceptación: El sistema permite administrar los proveedores del sistema.					

Tabla 2.15 HU Administración de categorías de productos

Historia de Usuario					
ID	14	Rol	Administrador/ Administrador de sucursal	Prioridad	Media

Nombre historia: Administración de categorías de productos
<p>Descripción: El usuario podrá administrar las categorías de productos del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de las categorías registrados. 2. Ingresar nuevas categorías. 3. Actualizar la información de las categorías registradas. 4. Eliminar categorías del sistema si es que no se han registrado productos en la misma.
Criterios de aceptación: El sistema permite administrar las categorías de productos del sistema.

Tabla 2.16 HU Administración de productos

Historia de Usuario					
ID	15	Rol	Administrador/ Administrador de sucursal	Prioridad	Media
Nombre historia: Administración de productos					
<p>Descripción: El usuario podrá administrar los productos del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 2. Leer información de los productos registrados. 3. Ingresar nuevos productos. 4. Actualizar la información de los productos registrados. 5. Eliminar productos del sistema. 					
Criterios de aceptación: El sistema permite administrar los productos del sistema.					

Tabla 2.17 HU Administración de costos

Historia de Usuario					
ID	16	Rol	Administrador/ Administrador de sucursal	Prioridad	Alta
Nombre historia: Administración de costos					
<p>Descripción: El usuario podrá administrar los costos del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los costos registrados. 2. Ingresar nuevos costos. 3. Actualizar la información de los costos registrados en el presente mes. 					
Criterios de aceptación: El sistema permite administrar los productos del sistema.					

Tabla 2.18 HU Administración de gastos

Historia de Usuario					
ID	17	Rol	Administrador/ Administrador de sucursal	Prioridad	Alta
Nombre historia: Administración de gastos					
<p>Descripción: El usuario podrá administrar los gastos del sistema con las operaciones descritas a continuación:</p> <ol style="list-style-type: none"> 1. Leer información de los gastos registrados. 2. Ingresar nuevos gastos. 3. Actualizar la información de los gastos registrados en el presente mes. 					
Criterios de aceptación: El sistema permite administrar los productos del sistema.					

Tabla 2.19 HU Generación de cotizaciones

Historia de Usuario					
ID	18	Rol	Administrador/ Administrador de sucursal/ Usuario	Prioridad	Alta
Nombre historia: Generación de cotizaciones					
<p>Descripción: El usuario podrá generar cotizaciones de la sucursal asignada. En caso de que el usuario tenga un rol de administrador podrá generar cotizaciones de cualquiera de las sucursales del sistema.</p> <p>Los clientes válidos para este tipo de comprobantes son los de tipo cliente.</p>					
Criterios de aceptación: El sistema permite generar cotizaciones.					

Tabla 2.20 HU Generación de facturas

Historia de Usuario					
ID	19	Rol	Administrador/ Administrador de sucursal/ Usuario	Prioridad	Alta
Nombre historia: Generación de facturas					
<p>Descripción: El usuario podrá generar facturas de la sucursal asignada. En caso de que el usuario tenga un rol de administrador podrá generar facturas de cualquiera de las sucursales del sistema.</p>					

<p>El usuario podrá validar el estado del comprobante a través de los servicios web del SRI. Una vez finalizada la creación del comprobante se visualizará la factura. Los clientes válidos para este tipo de comprobantes son los de tipo cliente.</p>
<p>Criterios de aceptación: El sistema permite generar facturas.</p>

Tabla 2.21 HU Generación de notas de crédito

Historia de Usuario					
ID	20	Rol	Administrador/ Administrador de sucursal	Prioridad	Alta
Nombre historia: Generación de notas de crédito					
<p>Descripción: El usuario podrá generar notas de crédito de la sucursal asignada. En caso de que el usuario tenga un rol de administrador podrá generar notas de crédito de cualquiera de las sucursales del sistema. El usuario podrá validar el estado del comprobante a través de los servicios web del SRI. Una vez finalizada la creación del comprobante se visualizará la nota de crédito. Los clientes válidos para este tipo de comprobantes son los de tipo cliente.</p>					
Criterios de aceptación: El sistema permite generar notas de crédito.					

Tabla 2.22 HU Generación de notas de débito

Historia de Usuario					
ID	21	Rol	Administrador/ Administrador de sucursal	Prioridad	Alta
Nombre historia: Generación de notas de débito					
<p>Descripción: El usuario podrá generar notas de débito de la sucursal asignada. En caso de que el usuario tenga un rol de administrador podrá generar notas de débito de cualquiera de las sucursales del sistema. El usuario podrá validar el estado del comprobante a través de los servicios web del SRI. Una vez finalizada la creación del comprobante se visualizará la nota de débito. Los clientes válidos para este tipo de comprobantes son los de tipo cliente.</p>					
Criterios de aceptación: El sistema permite generar notas de débito.					

Tabla 2.23 HU Generación de comprobantes de retención

Historia de Usuario					
ID	22	Rol	Administrador/ Administrador de sucursal	Prioridad	Alta
Nombre historia: Generación de comprobantes de retención					
<p>Descripción: El usuario podrá generar retenciones de la sucursal asignada. En caso de que el usuario tenga un rol de administrador podrá retenciones de cualquiera de las sucursales del sistema. El usuario podrá validar el estado del comprobante a través de los servicios web del SRI. Una vez finalizada la creación del comprobante se visualizará la retención. Los clientes válidos para este tipo de comprobantes son los de tipo sujeto retenido.</p>					
Criterios de aceptación: El sistema permite generar comprobantes de retención.					

Tabla 2.24 HU Generación de guías de remisión

Historia de Usuario					
ID	23	Rol	Administrador/ Administrador de sucursal/ Usuario	Prioridad	Alta
Nombre historia: Generación de guías de remisión					
<p>Descripción: El usuario podrá generar guías de remisión de la sucursal asignada. En caso de que el usuario tenga un rol de administrador podrá retenciones de cualquiera de las sucursales del sistema. El usuario podrá validar el estado del comprobante a través de los servicios web del SRI. Una vez finalizada la creación del comprobante se visualizará la guía de remisión. Los clientes válidos para este tipo de comprobantes son los de tipo destinatario.</p>					
Criterios de aceptación: El sistema permite generar guías de remisión.					

Tabla 2.25 HU Visualización de comprobantes electrónicos generados

Historia de Usuario					
ID	24	Rol	Administrador/ Administrador de sucursal/ Usuario/ Cliente	Prioridad	Alta
Nombre historia: Visualización de comprobantes electrónicos generados.					

<p>Descripción: El usuario podrá visualizar los comprobantes electrónicos generados en la sucursal asignada. En caso de que el rol del usuario sea administrador podrá visualizar todos los comprobantes generados por el sistema. En caso de que el rol del usuario sea cliente podrá únicamente visualizar los comprobantes generados con su identificación El usuario podrá descargar el comprobante electrónico como archivo PDF¹⁵ o enviarlo por correo electrónico.</p>
<p>Criterios de aceptación: El sistema permite visualizar comprobantes electrónicos generados.</p>

Tabla 2.26 HU Generación de reportes globales del negocio

Historia de Usuario					
ID	25	Rol	Administrador	Prioridad	Alta
Nombre historia: Generación reportes globales del negocio.					
Descripción: El usuario podrá generar reportes consolidados de información de todo el negocio con capacidad de realizar filtros por fechas. El usuario será capaz de imprimir el reporte mostrado por el sistema con la fecha de generación y el rango de fechas del reporte.					
Criterios de aceptación: El sistema permite generar reportes globales del negocio.					

Tabla 2.27 HU Generación de reportes por sucursal

Historia de Usuario					
ID	26	Rol	Administrador/ Administrador de sucursal	Prioridad	Alta
Nombre historia: Generación reportes por sucursal.					
Descripción: El usuario podrá generar reportes de información de la sucursal en la que se encuentra asignado con capacidad de realizar filtros por fechas. El usuario será capaz de imprimir el reporte mostrado por el sistema con la fecha de generación y el rango de fechas del reporte. En caso de que el usuario tenga el rol de administrador podrá generar reportes de cualquier sucursal del sistema.					
Criterios de aceptación: El sistema permite generar reportes por sucursal.					

¹⁵**Portable Document Format (PDF):** Es un formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware. Este formato es de tipo compuesto.

2.1.4. DIAGRAMA DE CASO DE USOS

Después del análisis de los requerimientos funcionales se elaboró el diagrama de casos de uso que se muestra en la Figura 2.1.

Los actores que interactúan con el sistema son: administrador, administrador de sucursal, usuario y cliente. Los casos de uso definidos son las acciones que cada uno de los actores realizará en el sistema para conseguir algún resultado específico.

Para todos los casos definidos para el prototipo se debe tomar en cuenta la previa autenticación del usuario en el sistema exceptuando el mismo.

2.1.5. ROLES Y PERMISOS

Después del análisis del diagrama de casos de uso se han definido los siguientes roles y permisos para los actores del sistema:

2.1.5.1. ROLES Y PERMISOS DE ROL

Un rol es una colección de permisos, se definen cuatro roles que interactúan con el sistema:

- *Administrador*: actor encargado de gestionar el sistema, realiza acciones CRUD de usuarios, sucursales, emisores, tipos de comprobantes, certificados digitales, impuestos y formas de pago. Además, se encarga de la generación de reportes globales del negocio, tiene los siguientes permisos:
 - Todos los permisos en el sistema, es capaz de realizar operaciones CRUD tanto en configuración de parámetros esenciales para el proceso de facturación electrónica como para el correcto funcionamiento del sistema. Además, tiene acceso a la generación de reportes generales del negocio.
- *Administrador de sucursal*: actor encargado de gestionar la sucursal en la cual se encuentra asignado, realiza acciones CRUD de categorías de productos, productos, gastos, costos. Además, se encarga de la generación de notas de crédito, notas de débito, comprobantes de retención y guías de remisión, tiene los siguientes permisos:
 - Tiene todos los permisos en el sistema limitándose a la sucursal a la que se encuentra asignado. No puede modificar parámetros para el proceso de facturación electrónica y puede generar reportes únicamente de la sucursal correspondiente.

- *Usuario*: actor encargado de gestionar clientes, transportistas y productos. Además, se encarga de la generación de cotizaciones, facturas y guías de remisión, tiene los siguientes permisos:
 - Tiene permisos únicamente para generar cotizaciones, facturas, y guías de remisión. Además, tiene acceso a la gestión de clientes, transportistas y productos.

- *Cliente*: actor que accede al sistema mediante un número de identificación y que es capaz de visualizar los comprobantes electrónicos generados donde consta como receptor, tiene los siguientes permisos:
 - Tiene acceso únicamente a la visualización de comprobantes electrónicos generados con su número de identificación. Adicionalmente deberá proporcionar el número de comprobante, clave de acceso o número de autorización por razones de seguridad del sistema.

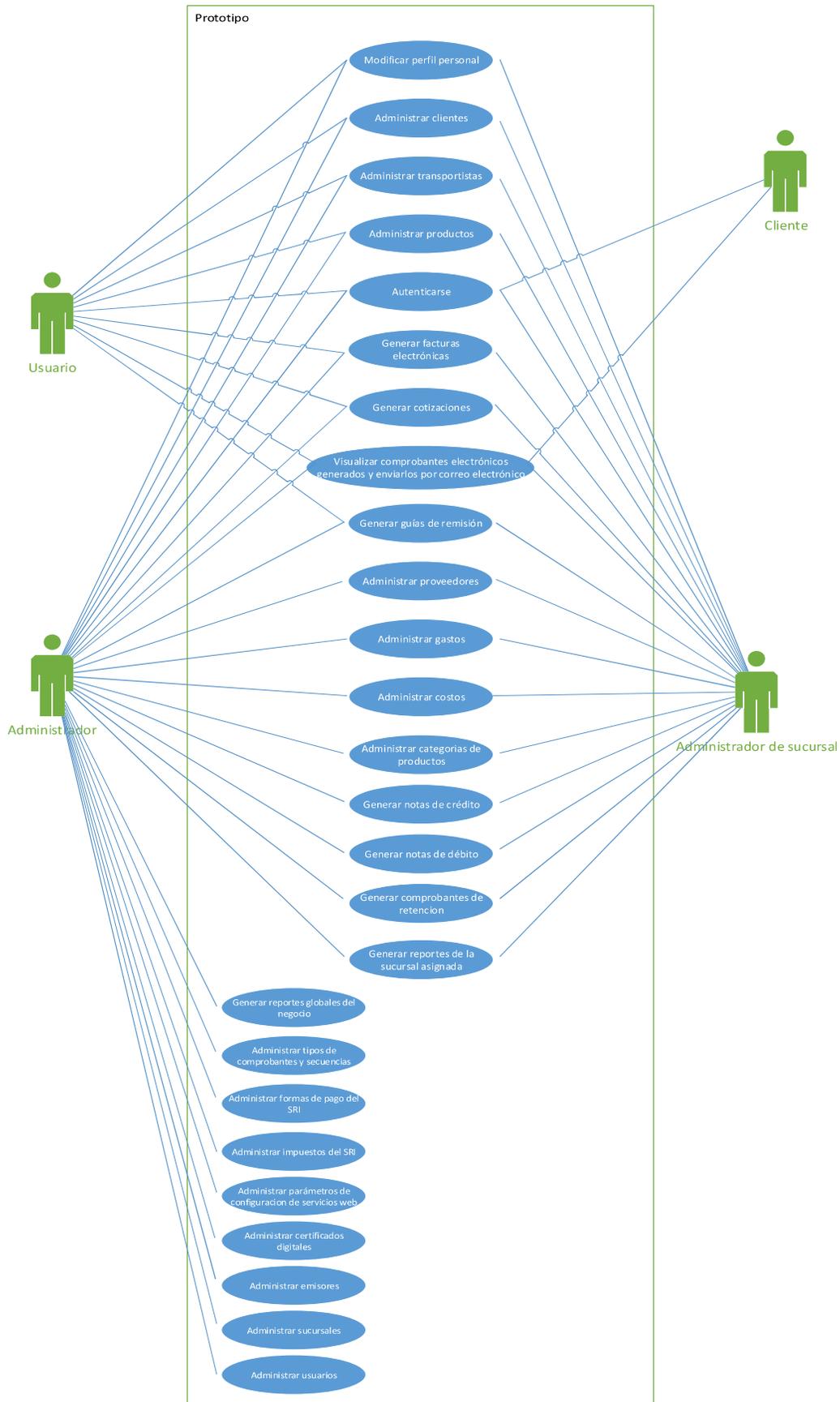


Figura 2.1 Diagrama de caso de uso del prototipo

2.1.6. ARQUITECTURA

El sistema será diseñado con una arquitectura N-Capas, las cuales serán:

- *Capa de datos*: es el servidor que almacena los datos de la aplicación. Puede ser accedida únicamente mediante la capa de acceso a datos.
- *Capa de acceso a datos*: proporciona la capacidad de persistir los datos de la aplicación. Pueden ser datos propios del sistema o datos expuestos por otros sistemas. Esta capa de persistencia se comunica con la capa de datos para proveer acceso a capas superiores. Esta capa permite hacer operaciones CRUD sobre la capa de datos.
- *Capa de lógica de negocio*: implementa la funcionalidad principal del sistema y encapsula la lógica que hace cumplir las reglas del negocio de la aplicación. Además, permite la independencia entre la aplicación cliente y la capa de acceso a datos.
- *Capa de servicio*: la función principal de esta capa es la de exponer o consumir servicios externos.
- *Capa de presentación*: la función principal de esta capa es la de implementar la funcionalidad requerida para que los usuarios puedan interactuar con el sistema a través de una interfaz de usuario. La capa de presentación está compuesta por elementos visuales utilizadas para mostrar información y aceptar entradas de usuario
- *Capa de entidades*: en esta capa se almacenan las entidades que permiten representar los datos de la capa de datos, y que son utilizadas por las todas las otras capas excepto la capa de datos.

En la arquitectura basada en capas, cada capa se comunica con la inferior adyacente, esto es, la capa superior consume la funcionalidad de la capa inferior.

En la Figura 2.2 se presenta la arquitectura del sistema y la interacción entre sus capas.

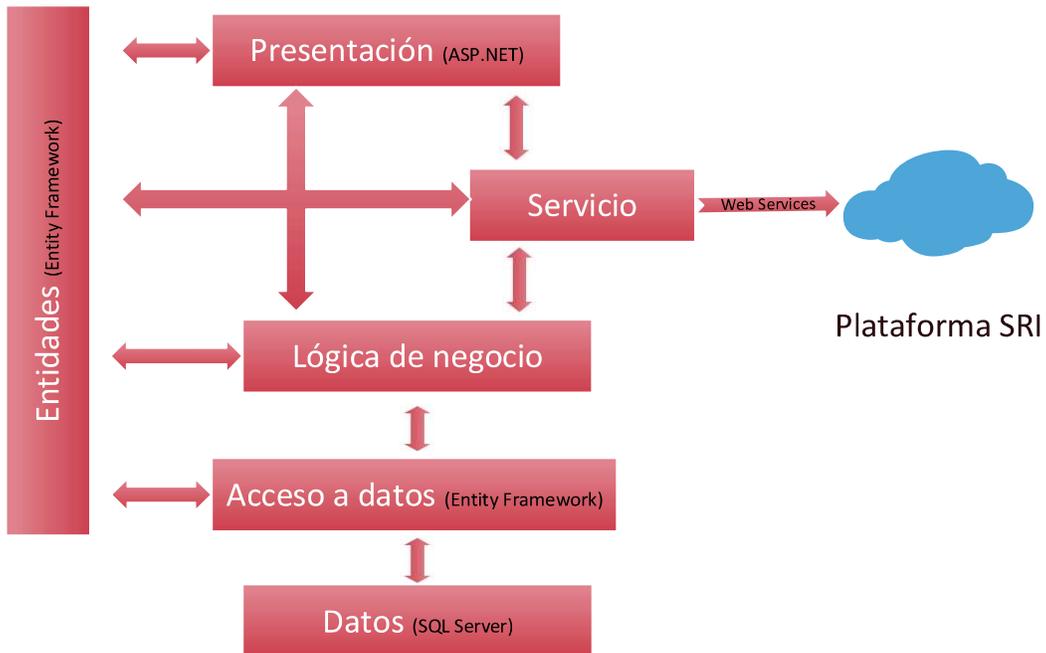


Figura 2.2 Arquitectura del prototipo.

2.1.7. DIAGRAMA ENTIDAD-RELACIÓN

En base a los requerimientos de usuario establecidos se elaboró el diagrama Entidad-Relación que se muestra en la Figura 2.33. Este diagrama sirvió para el modelado de la base de datos que se utilizó para el almacenamiento de información de usuarios, sucursales, emisores, comprobantes y clientes. El diagrama fue generado utilizando la herramienta Entity Framework 6 [20].

2.1.7.1. DESCRIPCIÓN DE TABLAS

La tabla *Change_Log_Database* se encarga de almacenar todas las transacciones realizadas en la base de datos, transacciones como alteración de tablas, procedimientos almacenados, *triggers*, vistas.

La tabla *Change_Log_Table* se encarga de almacenar en las transacciones *insert*, *update* y *delete* de las tablas más importantes del sistema.

La tabla *Sucursales* se encarga de almacenar las sucursales del sistema.

La tabla *Usuarios* se encarga de almacenar la información de usuarios en el sistema. Tiene la restricción de no admitir ingresos con el mismo nombre de usuario en cuanto el usuario este en estado activo.

La tabla *Logs_Usuarios* almacena los datos de todos los accesos de los usuarios al sistema.

La tabla *Certificados* almacena la información de los certificados digitales almacenados en el sistema.

La tabla *Emisores* almacena la información de cada uno de los emisores del sistema, está relacionada a la sucursal y al certificado digital.

La tabla `Transportistas` almacena la información de los transportistas del sistema. Tiene la restricción de que el número de identificación debe ser único.

La tabla `Telefonos_Transportistas` almacena la información de los teléfonos de transportistas.

La tabla `Correos_Transportistas` almacena la información de los correos de transportistas.

La tabla `Tipos_Clientes` almacena los diferentes clientes disponibles para el sistema como clientes, sujetos retenidos o destinatarios. Tiene la restricción de que el tipo de cliente ingresado debe ser único.

La tabla `Clientes` almacena la información de los clientes del sistema, al igual que la tabla `transportistas` tiene la restricción que cada cliente debe tener un número de identificación único.

La tabla `Cliente_Tipo_Cliente` almacena la información entre la relación de la tabla `clientes` y la tabla `tipos de clientes` debido a que un cliente puede ser de diferentes tipos.

La tabla `Telefonos_Clientes` almacena la información de los teléfonos de clientes.

La tabla `Correos_Clientes` almacena la información de los correos electrónicos de los clientes ingresados.

La tabla `Proveedores` almacena la información de proveedores del sistema. Tiene la restricción de ingresar un número de identificación único.

La tabla `Telefonos_Proveedores` almacena la información de los teléfonos de proveedores.

La tabla `Correos_Proveedores` almacena la información de los correos electrónicos de proveedores.

La tabla `Gastos` almacena los gastos ingresados para cada sucursal por el administrador general o administrador de sucursal.

La tabla `Costos` almacena los gastos ingresados para cada sucursal por el administrador general o administrador de sucursal.

La tabla `Red` almacena los parámetros de configuración de los servicios web del SRI. Tiene la restricción de tener en estado activo únicamente un registro.

La tabla `Tipos_Impuestos` almacena los tipos de impuestos establecidos por el SRI. Tiene la restricción de tener un solo impuesto con el mismo nombre en estado activo.

La tabla `Impuestos` almacena los impuestos establecidos por el SRI.

La tabla `Categorias_Productos` almacena la información de categorías de productos del sistema. Tiene la restricción de que la combinación entre el código de la categoría y el nombre sea única.

La tabla `Productos` almacena información de todos los productos registrados en el sistema. Al igual que la tabla de categorías de productos tiene la restricción de que la combinación entre el código del producto y el nombre sea única. Las columnas `ICE` e `IRBPNR` son válidas para determinar si el producto tiene alguno de estos impuestos.

La tabla `Producto_Impuesto` almacena información de los productos y su relación con los impuestos que afectan directamente al precio.

La tabla Formas_Pago almacena información de las formas de pago del SRI.

La tabla Tipos_Comprobantes almacena los diferentes tipos de comprobantes del sistema. Tiene la restricción de ingresar un único tipo de comprobante en estado activo.

La tabla Secuencia_Comprobantes almacena el número de secuencia de los comprobantes por cada sucursal.

La tabla Comprobantes almacena la información genérica para todos los tipos de comprobantes.

La tabla Datos_Emision almacena los datos de emisión de cada comprobante generado.

La tabla Info_Adicional almacena información extra de los comprobantes generados.

La tabla Facturas almacena la información de facturas generadas.

La tabla Detalle_Factura almacena el detalle de los productos agregados a la factura.

La tabla Detalle_Forma_Pago_Factura almacena la información de la forma de pago seleccionada con los valores para la factura.

La tabla Cotizaciones almacena la información de cotizaciones generadas.

La tabla Detalle_Cotizacion almacena el detalle de los productos agregados a la cotización.

La tabla Retenciones almacena información de los comprobantes de retención generados por el sistema.

La tabla Detalle_Retencion almacena la información de los impuestos retenidos por el contribuyente.

La tabla Notas_Credito almacena la información de la nota de crédito generada por el sistema.

La tabla Detalle_Nota_Credito almacena la información de los productos agregados a la nota de crédito.

La tabla Notas_Debito almacena la información de las notas de débito generadas por el sistema.

La tabla Detalle_Nota_Debito almacena la información de los valores de la nota de débito.

La tabla Detalle_Forma_Pago_Nota_Debito almacena la información de la forma de pago registrada para la nota de débito.

La tabla Guias_Remision almacena la información de las guías de remisión generadas por el sistema.

La tabla Detalle_Guia_Remision almacena la información de los destinatarios asociados a la guía de remisión.

La tabla Detalle_Productos_Guia_Remision almacena la información de los productos asociados a cada destinatario en la guía de remisión.

La tabla Respuestas_Web_Service almacena la información de las solicitudes realizadas a los servicios web del SRI.

2.1.8. DIAGRAMA DE CLASES

En esta sección se presenta el diagrama de clases de cada una de las capas utilizadas para el desarrollo del sistema.

2.1.8.1. CAPA DE ACCESO A DATOS

La Figura 2.4 presenta el diagrama de clases correspondiente a la capa de acceso a datos donde la interfaz `IRepository` expone los métodos disponibles para el consumo de las clases que lo implementen. La clase `Repository` implementa la interfaz con objetos genéricos para la utilización de la misma clase para todo el sistema. Con la ayuda de *entity framework* se genera el contexto desde la base de datos, donde se mapean las entidades como propiedades y los resultados de procedimientos almacenados se mapean como funciones.

2.1.8.2. CAPA DE ENTIDADES

La capa de Entidades es generada mediante *entity framework* el cual mapea las tablas y procedimientos almacenados directamente desde la base de datos a clases. La Figura 2.55 y Figura 2.66 presentan el diagrama de clases de entidades y resultado de procedimientos almacenados.

2.1.8.3. CAPA DE NEGOCIO

En la Figura 2.7 se presenta el diagrama de clases de la capa de negocio.

2.1.8.4. CAPA DE SERVICIO

Para la capa de servicio se emplean las clases `EnvioComprobantes` y `VerificaComprobantes` útiles para verificar el estado, y enviar comprobantes a través de los servicios web del SRI. La Figura 2.88 presenta el diagrama de clases de esta capa.

2.1.8.5. CAPA DE PRESENTACIÓN

La Figura 2.9, Figura 2.10, Figura 2.11 y la Figura 2.12 presentan el diagrama de clases de la capa de presentación.



Figura 2.4 Diagrama de clases de la capa de acceso a datos.



Figura 2.5 Diagrama de clases de la capa de entidades (PARTE I).

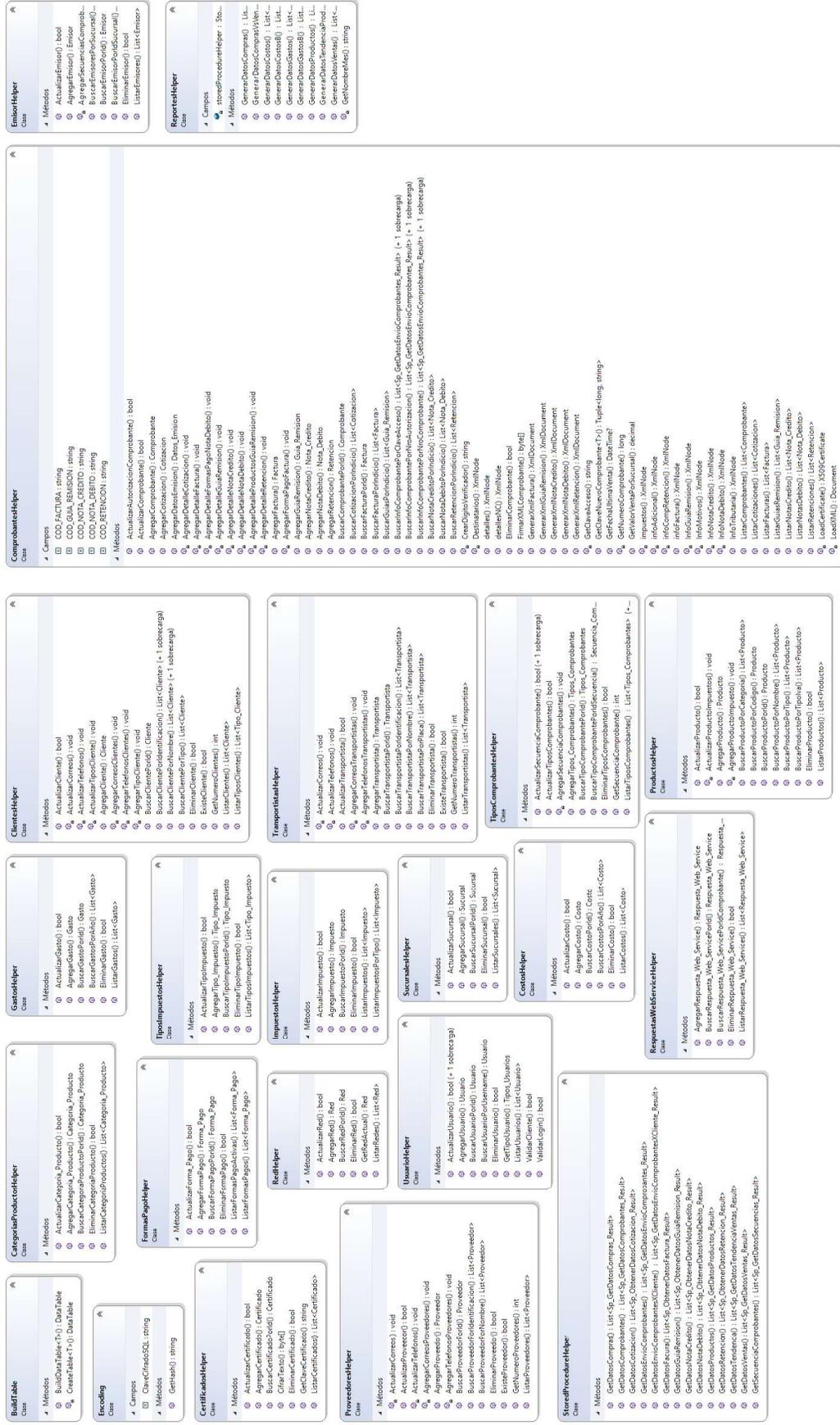
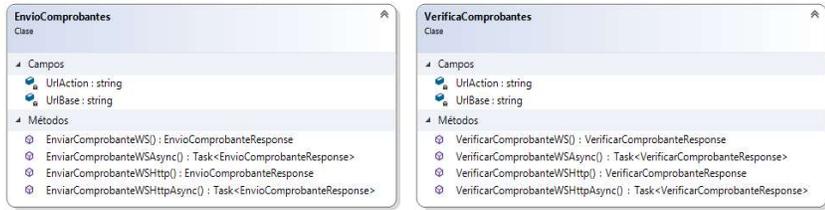


Figura 2.7 Diagrama de clases de la capa de negocio.



Clases generadas automáticamente por el servicio web cuando se agrega la referencia al servicio.

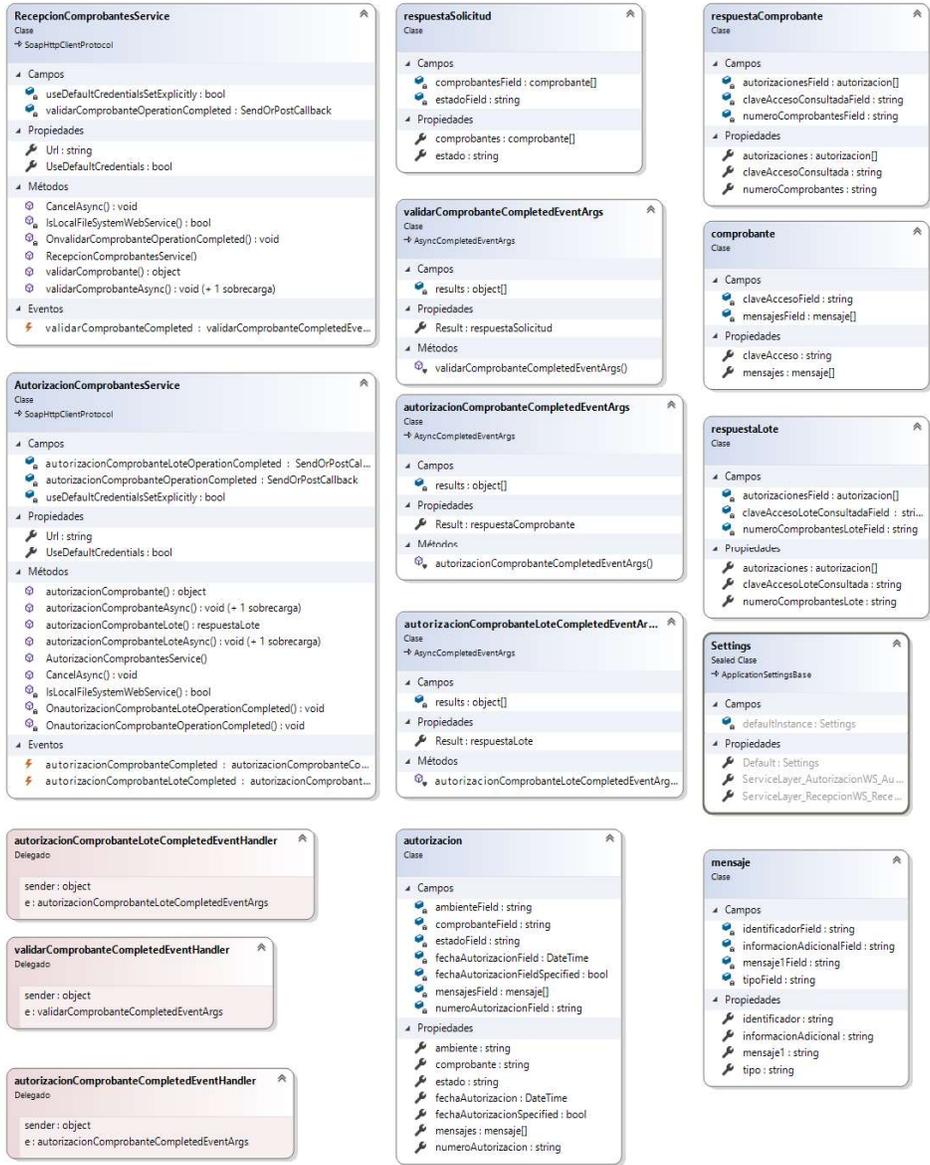


Figura 2.8 Diagrama de clases de la capa de servicio.

```

Global
Class
↳ HttpApplication

Métodos
Application_Error() : void
Application_Start() : void

```

```

Utilities
Class

Métodos
LoadLibrary() : IntPtr
LoadNativeAssembly() : void
LoadNativeAssembly() : void

```

```

Main
Class
↳ MasterPage

Campos
actualEmisor : Emisor
actualUser : Usuario
binSalir : LinkButton
cphContent : ContentPlaceHolder
cphFooter : ContentPlaceHolder
cphHeader : ContentPlaceHolder
frmPrincipal : HtmlForm
idSucursal : int
imgUser : Image
lblNombreBanco : Label
lblAdministracion : HtmlGenericControl
lblCategoria : HtmlGenericControl
lblClientes : HtmlGenericControl
lblComprobantes : HtmlGenericControl
lblConfiguracion : HtmlGenericControl
lblCostos : HtmlGenericControl
lblDashboard : HtmlGenericControl
lblGastos : HtmlGenericControl
lblInasCredito : HtmlGenericControl
lblInasDebito : HtmlGenericControl
lblProveedores : HtmlGenericControl
lblReportes : HtmlGenericControl
lblRetenciones : HtmlGenericControl
lblSistema : HtmlGenericControl
titulo : HtmlGenericControl

tiposAcceso : Tipo_Acceso
tiposLogin : Tipos_Login
update : UpdateProgress

usuarioIdHelper : UsuarioIdHelper

Métodos
binSalir_Click() : void
Page_Load() : void

```

```

Wirm_Login
Class
↳ Page

Campos
btnCliente : LinkButton
btnIngresar : LinkButton
btnIngresoGlobal : Button
btnIngresoMátrix : Button
btnIngresoSucursal : Button
chkContraseña : HtmlGenericControl
chkUsuario : HtmlGenericControl
form1 : HtmlForm
hdFldSucursal : HiddenField
hdContraseña : HtmlGenericControl
lblUsuario : HtmlGenericControl
listaSucursales : List<Sucursal>
prlnIngresoGlobal : Panel
rptPass : RequiredFieldValidator
rptSucursales : Repeater
rptPassword : TextBox
txtUsername : TextBox

usuarioIdHelper : UsuarioIdHelper

Métodos
btnCliente_Click() : void
btnIngresar_Click() : void
btnIngresoGlobal_Click() : void
btnIngresoSucursal_Click() : void
Page_Load() : void
rptSucursales_tembDataBound() : void

```

```

Wirm_Dashboard
Class
↳ Page

Campos
ASCENDING : string
datos : List<SP_GetDatosComprobantes_Result>
DESCENDING : string
dBalance : HtmlGenericControl
dChart : HtmlGenericControl
dInasCredito : HtmlGenericControl
dInasDebito : HtmlGenericControl
dRetenciones : HtmlGenericControl
gridDashboard : GridView
idSucursal : int
lblNombreBanco : Label
lblNombreSucursal : Label
lblNumeroGlobalRecepcion : Label
lblNumeroGlobalDebito : Label
lblNumeroGlobalCredito : Label
lblNumeroRetenciones : Label
rptSucursales : Repeater

storedProcedureHelper : StoredProcedureHelper
sucursales : List<Sucursal>
usuarioIdHelper : UsuarioIdHelper
tiposAcceso : Tipo_Acceso
usuarioIdHelper : UsuarioIdHelper

Métodos
CargarDatosSucursales() : void
gridDashboard_RowDataBound() : void
gridDashboard_Sorting() : void
Page_Load() : void
rptSucursales_tembDataBound() : void
SortGridView() : void
Tipos_anidados

```

```

Wirm_Mostrat_Comprobante
Class
↳ Page

Campos
btnEnviarCorreo : ImageButton
comprobantesActual : Comprobante
comprobantesIdHelper : ComprobantesIdHelper
correos : List<string>
form1 : HtmlForm
ReportViewer1 : ReportViewer
ScriptManager1 : ScriptManager
storedProcedureHelper : StoredProcedureHelper

Métodos
btnEnviarCorreo_Click() : void
CargarComprobante() : void
EnviarCorreo() : void
EnviarCorreo_Click() : void
Page_Load() : void

```

Figura 2.9 Diagrama de clases de la capa de presentación (PARTE I).



Figura 2.11 Diagrama de clases de la capa de presentación (PARTE III).

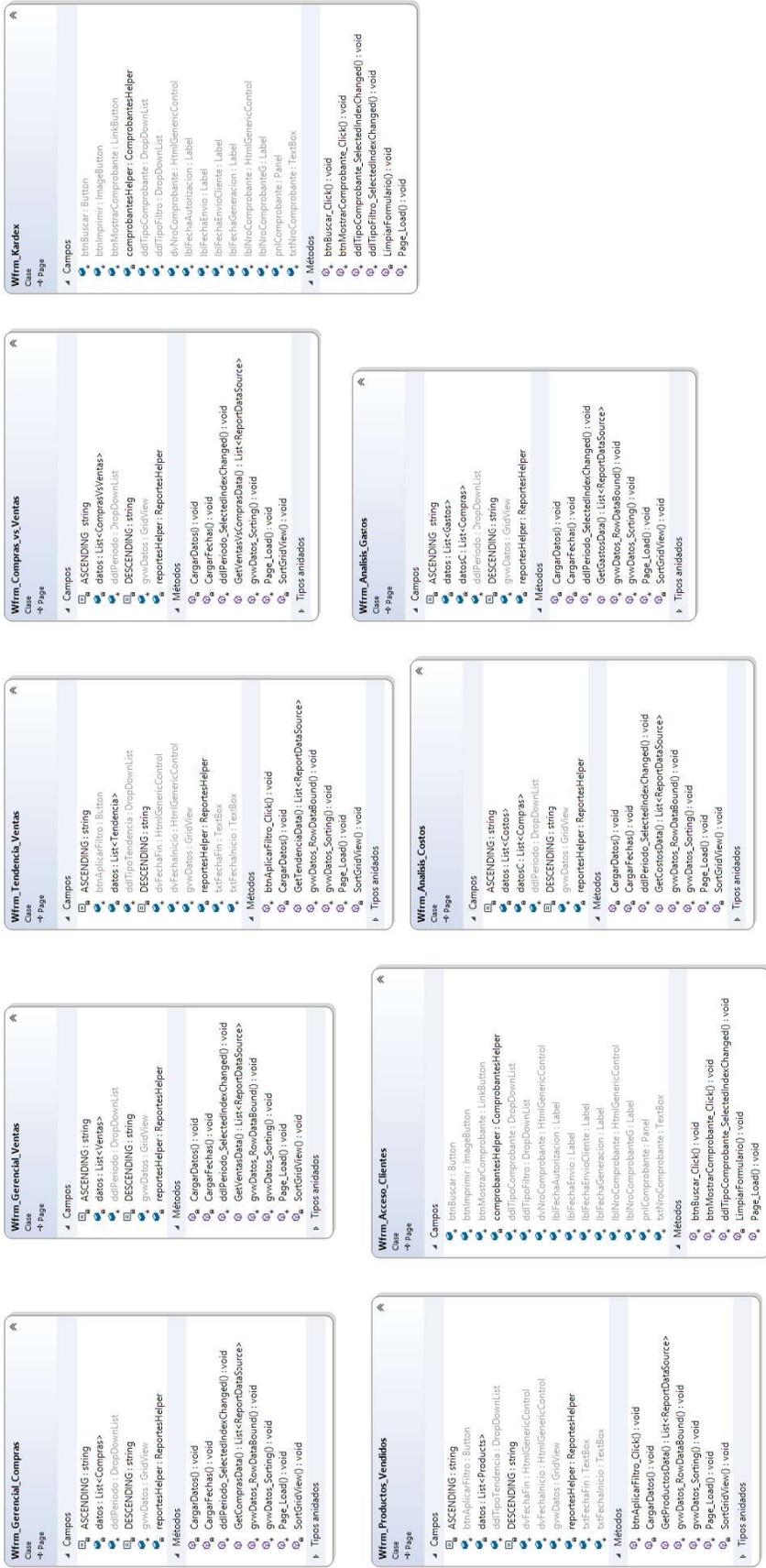


Figura 2.12 Diagrama de clases de la capa de presentación (PARTE IV).

2.1.9. MÓDULOS DEL SISTEMA Y VISTAS DE USUARIO

A continuación, se presentan los módulos del sistema y el diseño de las interfaces de usuario para uno de los componentes de los módulos.

Los procesos se modelaron mediante diagramas de actividades y muestran los procesos como un flujo de trabajo entre los usuarios y el sistema.

Se ha definido un diagrama de flujo por cada módulo ya que el proceso es similar para todos los componentes del módulo.

Las vistas que se detallan han sido diseñadas en base a los requerimientos definidos anteriormente, para el diseño de los diagramas se ha utilizado la herramienta Visio¹⁶. Es importante aclarar que las interfaces finales variaron con respecto a las presentadas en esta fase por motivos de presentación de la aplicación.

2.1.9.1. MÓDULO DE AUTENTICACIÓN

Permite validar credenciales de los usuarios y clientes para el ingreso al sistema. La Figura 2.13 presenta el diagrama de actividades del módulo de autenticación tanto para usuarios como para clientes.

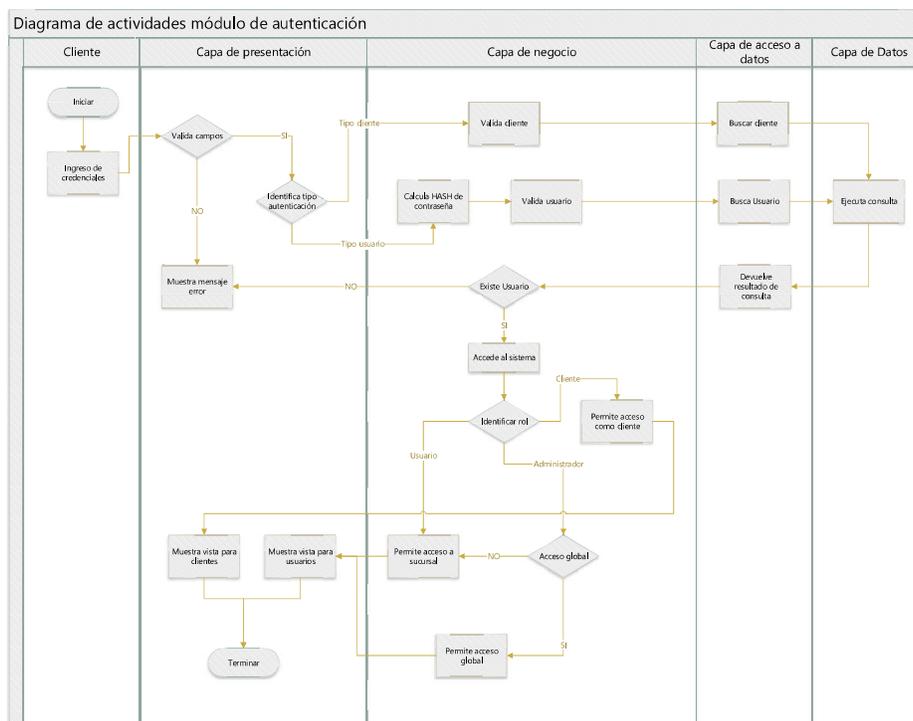


Figura 2.13 Diagrama de actividades del módulo de autenticación

¹⁶ Microsoft Visio: Es un software para dibujar una variedad de diagramas.

2.1.9.1.1. Inicio de sesión usuario

Esta vista permite a los usuarios autenticarse en el sistema ingresando las credenciales de acceso correspondientes. Se valida el rol que este usuario posee y se le muestra la vista correspondiente. La Figura 2.14 muestra el diseño de la vista de inicio de sesión para un usuario del sistema.

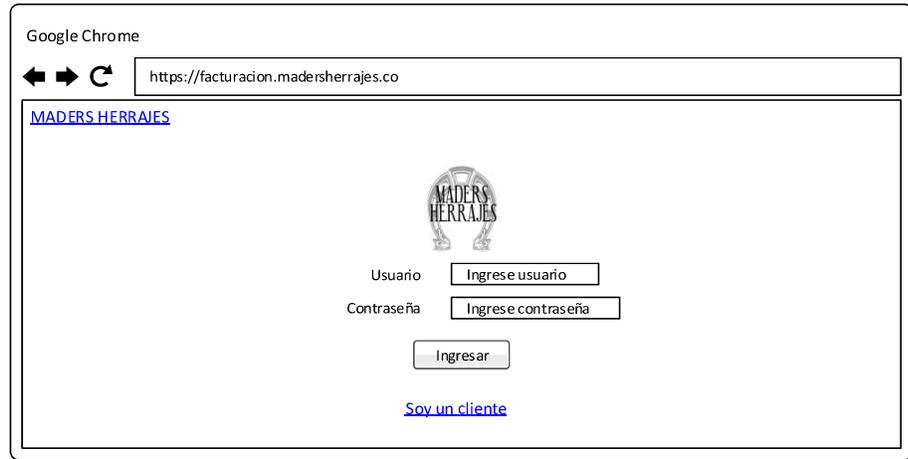


Figura 2.14 Vista de inicio de sesión usuario.

2.1.9.1.2. Inicio de sesión cliente

Esta vista permite a los clientes autenticarse en el sistema ingresando el número de identificación. Este se valida a través de la base de datos y le muestra la vista correspondiente al acceso. La Figura 2.15 muestra el diseño de la vista para ingreso de credenciales de usuarios con el rol cliente.

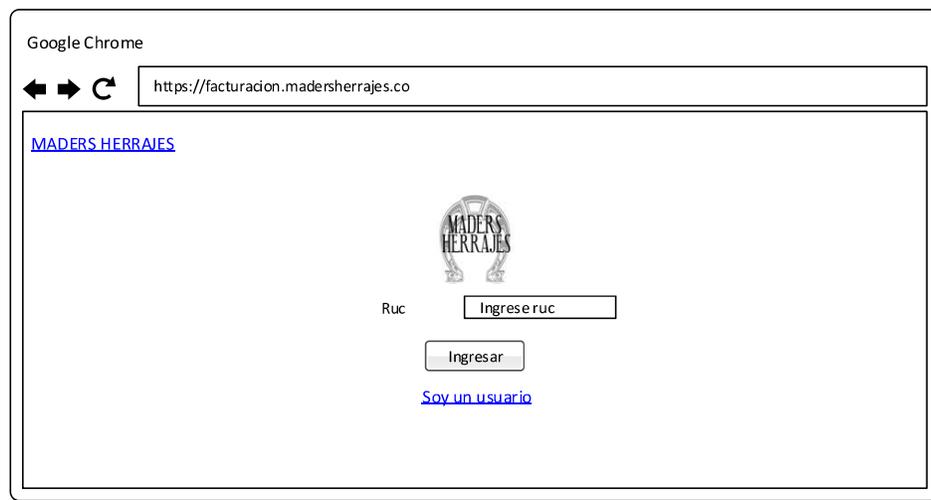


Figura 2.15 Vista de inicio de sesión cliente.

Una vez autenticado en el sistema se muestra la vista general para el usuario el cual contiene información relevante a la sucursal a la cual pertenece, tal como se muestra en la

Figura 2.17, en caso de tener un rol administrativo y haber seleccionado la vista global del negocio se mostrará la vista con la información completa de todas las sucursales del negocio. La Figura 2.16 muestra la vista empleada para la selección de sucursal.

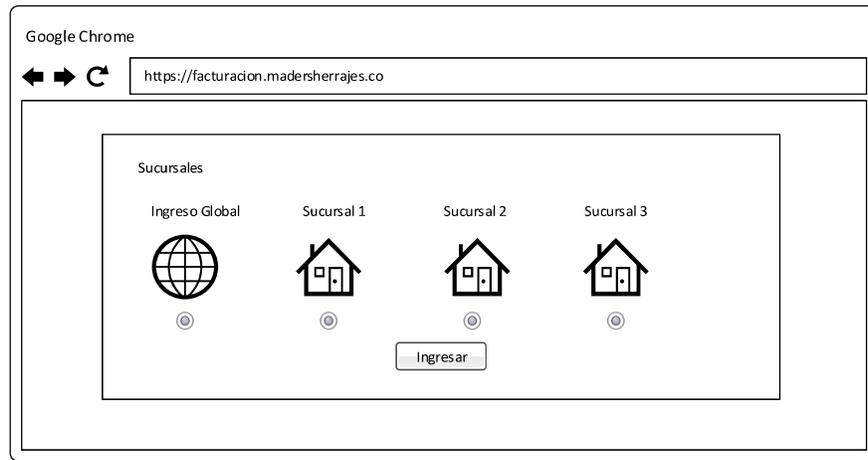


Figura 2.16 Vista de inicio de sesión usuario con rol de administrador.

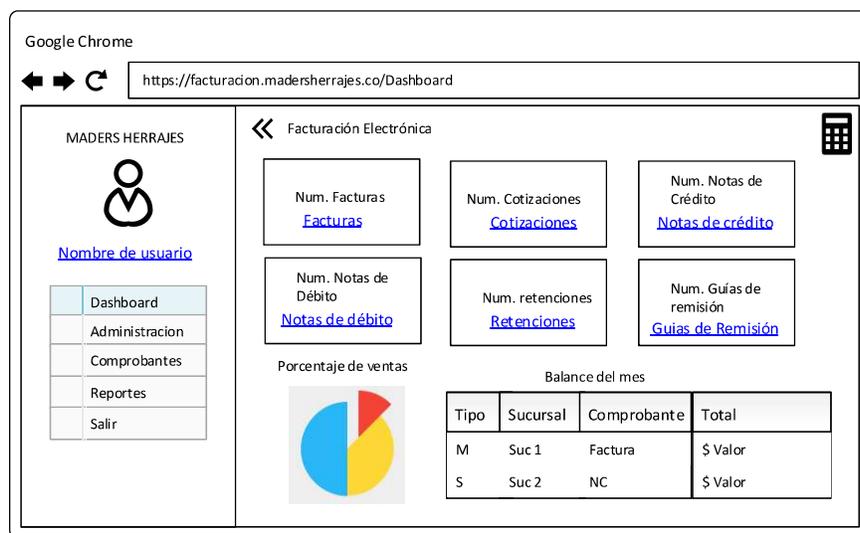


Figura 2.17 Vista principal al ingreso del usuario.

2.1.9.2. MÓDULO DE ADMINISTRACIÓN

Permite administrar los componentes del sistema. La Figura 2.18 presenta el diagrama de actividades para los procesos de ingreso y actualización de los componentes. Así mismo, la Figura 2.19 muestra el diagrama de actividades para el proceso de eliminación de una entidad del sistema.

La Figura 2.20 muestra el diagrama de actividades correspondiente al proceso de visualización de información de una entidad.

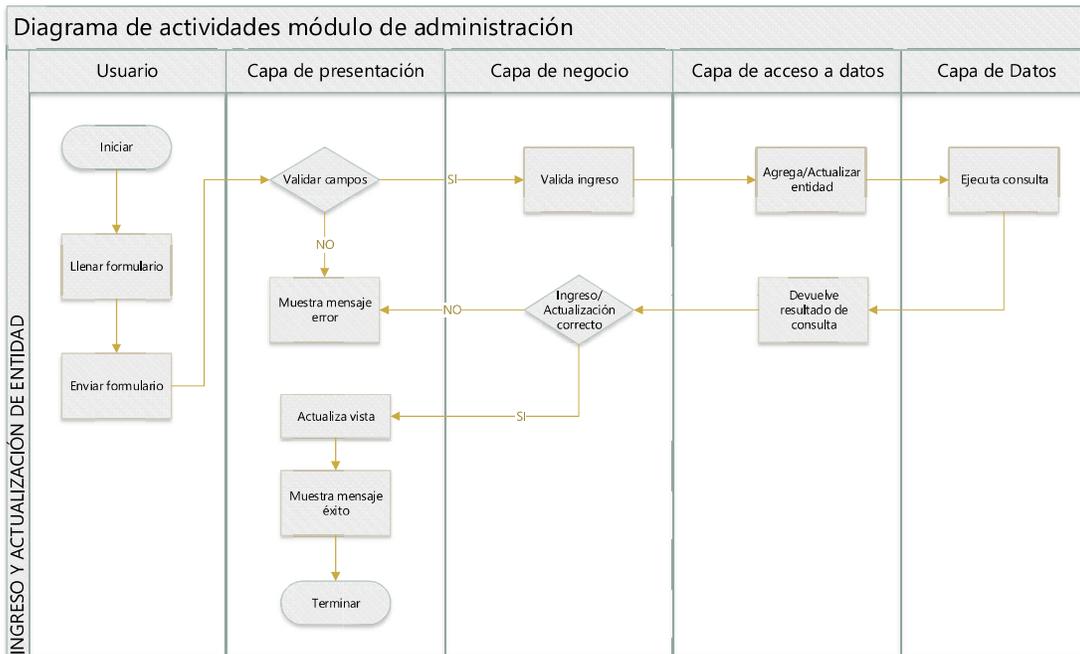


Figura 2.18 Diagrama de clases del módulo de administración (Ingreso y actualización de entidad).

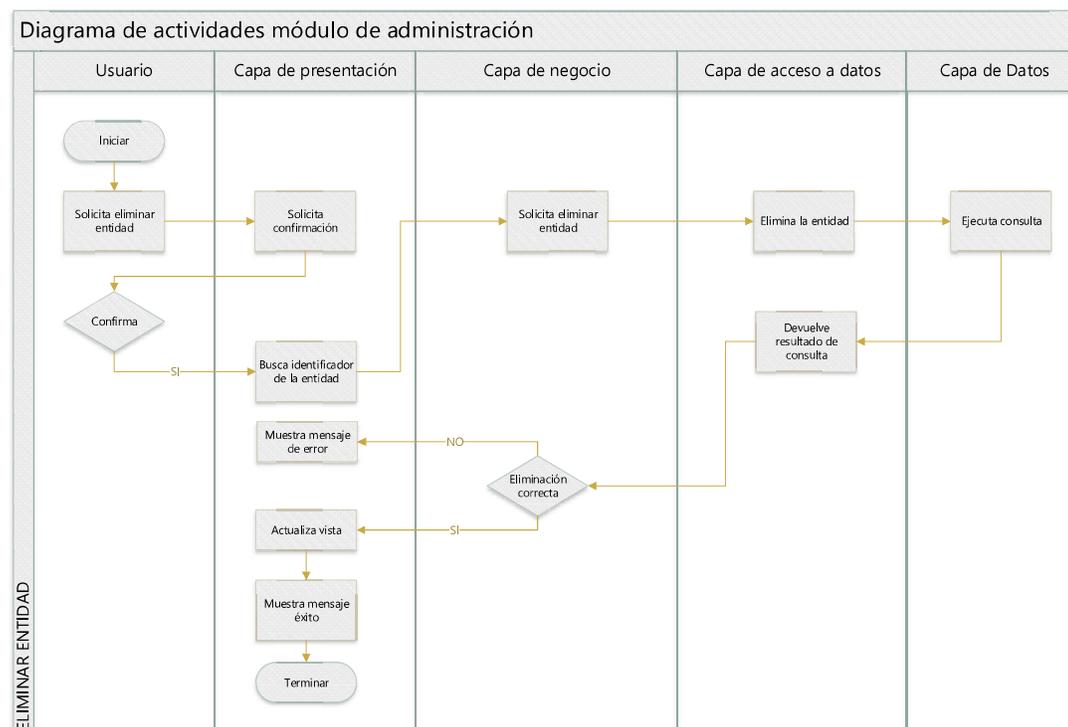


Figura 2.19 Diagrama de clases módulo de administración (Eliminar entidad).

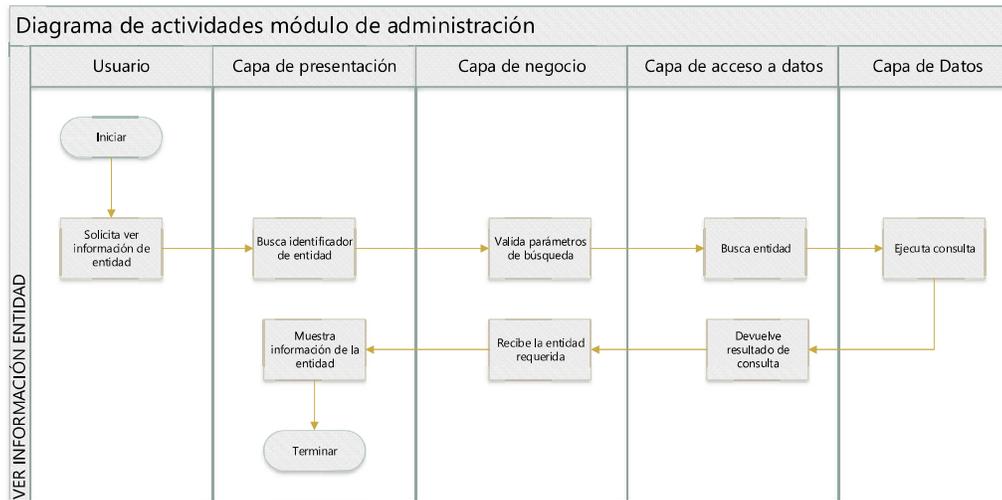


Figura 2.20 Diagrama de clases módulo de administración (Visualizar información entidad).

2.1.9.2.1. Administración de usuarios

Esta vista permite administrar usuarios y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.21 muestra la vista general de administración de usuarios que visualiza el usuario. Se visualiza la información más relevante del usuario y las acciones disponibles para cada usuario.

Para los usuarios ingresados se han establecido las acciones ver y editar.

La Figura 2.22 presenta la vista para el ingreso de un nuevo usuario al sistema.

Las vistas para la visualización de los datos extendidos del usuario y modificación se presentan en la Figura 2.24 y Figura 2.23 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.2.2. Administración de certificados digitales

Esta vista permite administrar certificados digitales y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.25 muestra la vista general de certificados digitales que visualiza el usuario. Se visualiza la información más relevante del certificado y las acciones disponibles para cada certificado.

Para los certificados ingresados se han establecido las acciones ver y editar.

La Figura 2.26 muestra la vista para agregar un nuevo certificado digital al sistema.

Las vistas para la visualización de los datos extendidos del certificado y modificación se presentan en la Figura 2.28 y Figura 2.27 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

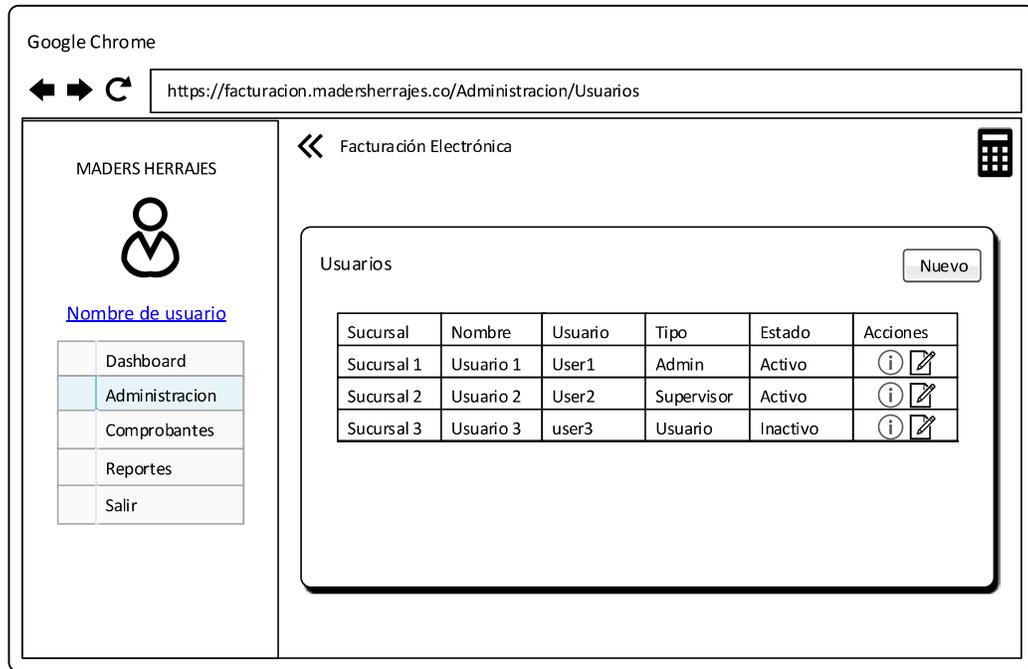


Figura 2.21 Vista principal de administración de usuarios.

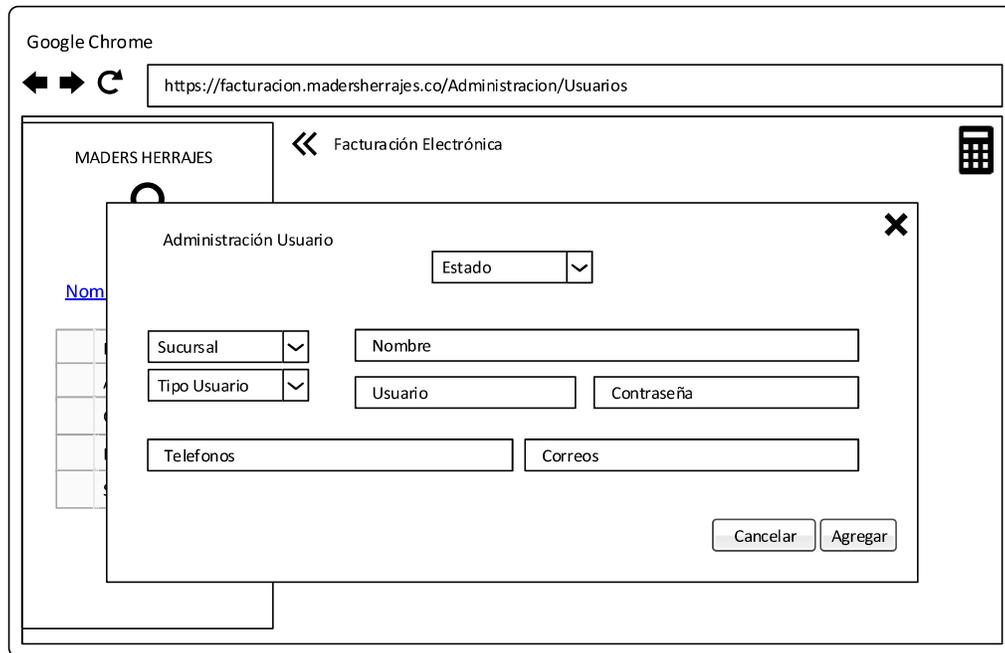


Figura 2.22 Vista para agregar un nuevo usuario al sistema.

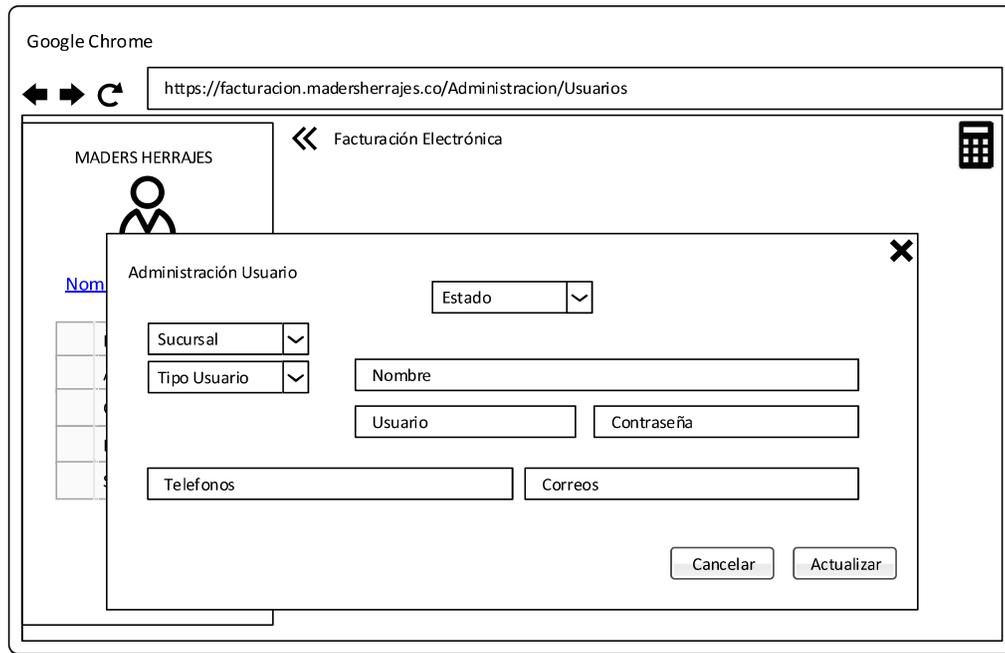


Figura 2.23 Vista para actualizar un usuario existente del sistema.

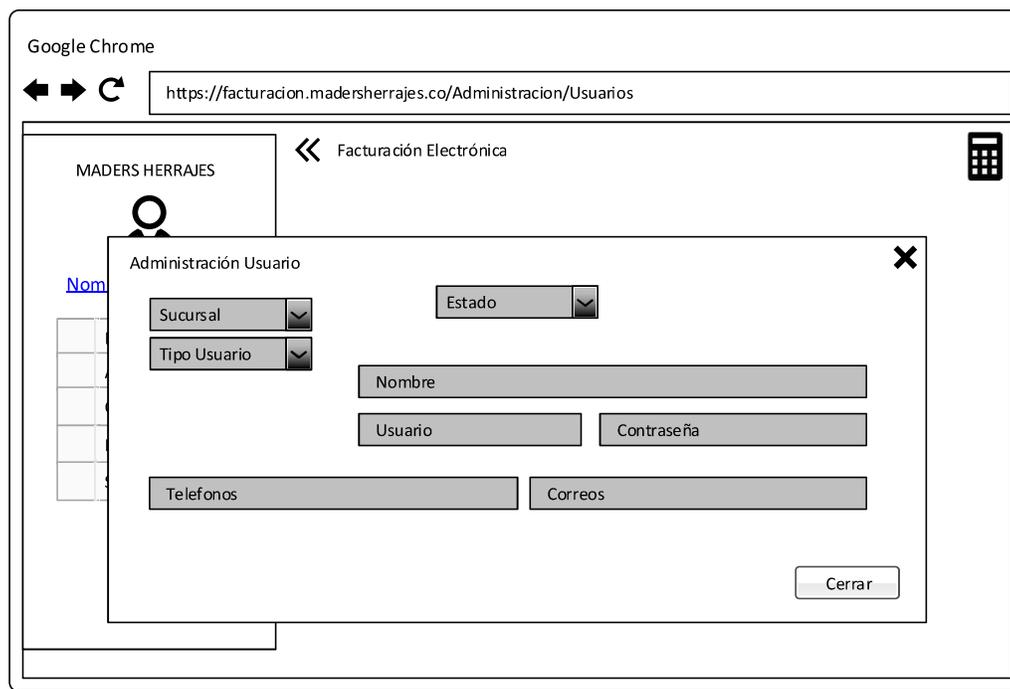


Figura 2.24 Vista para ver información de un usuario existente del sistema.

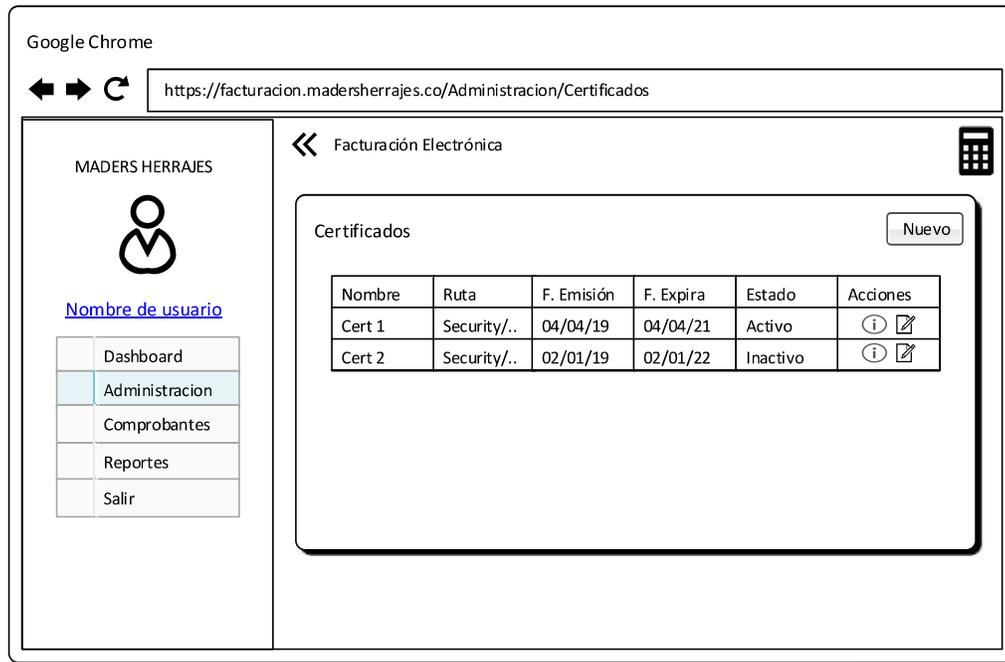


Figura 2.25 Vista general para administrar certificados digitales.

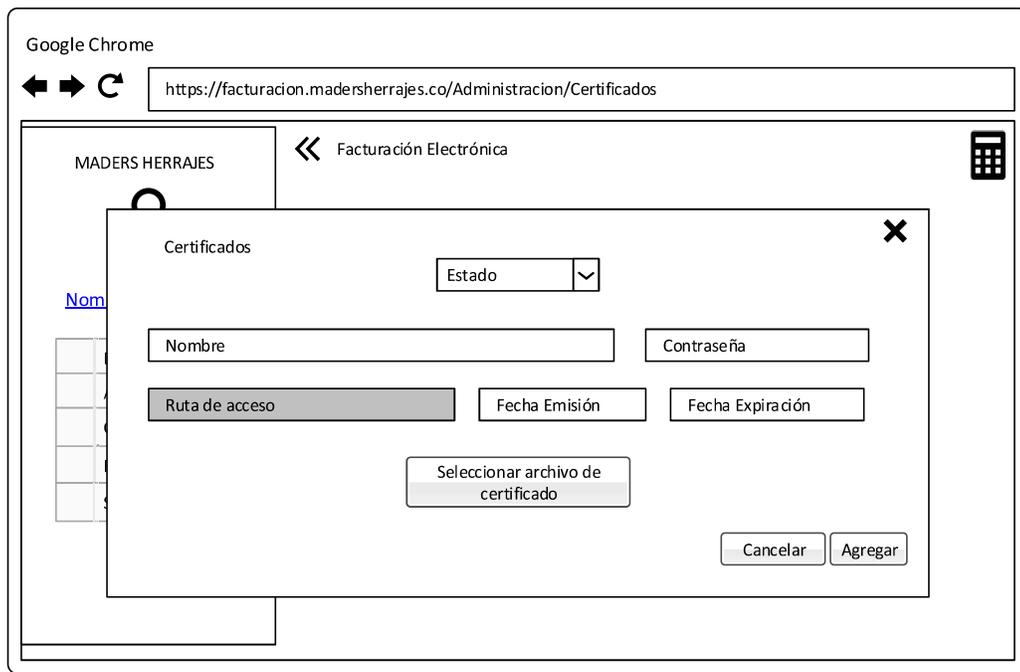


Figura 2.26 Vista para agregar un nuevo certificados digital al sistema.

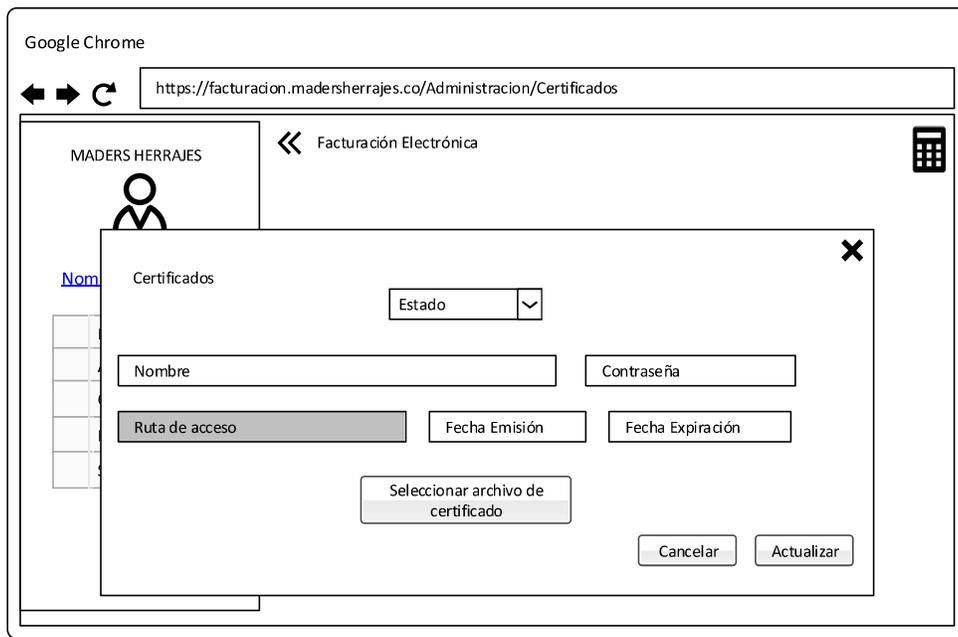


Figura 2.27 Vista para actualizar un certificado digital existente del sistema.

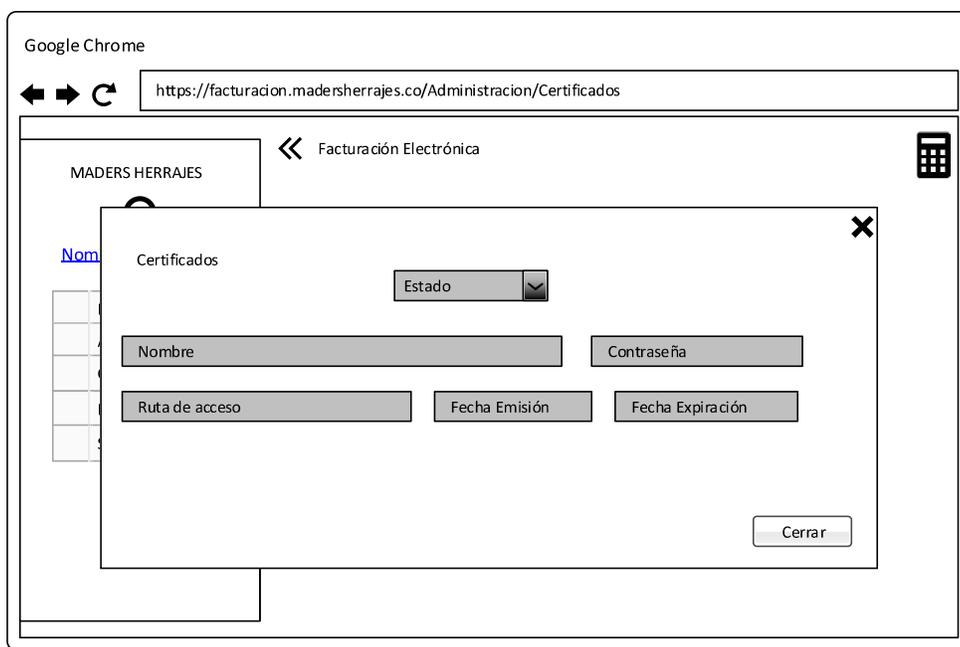


Figura 2.28 Vista para ver información de un certificado digital existente del sistema.

2.1.9.2.3. Administración de emisores

Esta vista permite administrar emisores y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.29 muestra la vista general de emisores al ingresar a la vista. Se visualiza la información más relevante de los emisores y las acciones disponibles para cada emisor.

Para los emisores ingresados se han establecido las acciones copiar, ver, editar y eliminar.

La Figura 2.30 presenta la vista para agregar un nuevo emisor al sistema.

Las vistas para la visualización de los datos extendidos del emisor y modificación se presentan en la Figura 2.32 y Figura 2.31 respectivamente.

La vista para eliminar el emisor se presenta en la Figura 2.33. Únicamente se podrán eliminar emisores con los cuales no se hayan generado comprobantes electrónicos.

Para el ingreso se valida que los campos estén llenos correctamente.

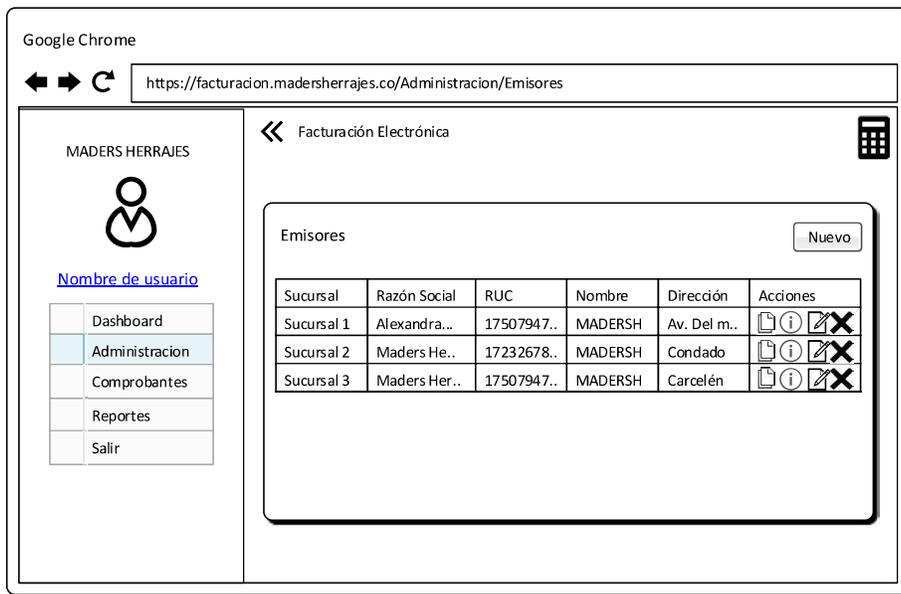


Figura 2.29 Vista general para administrar emisores.

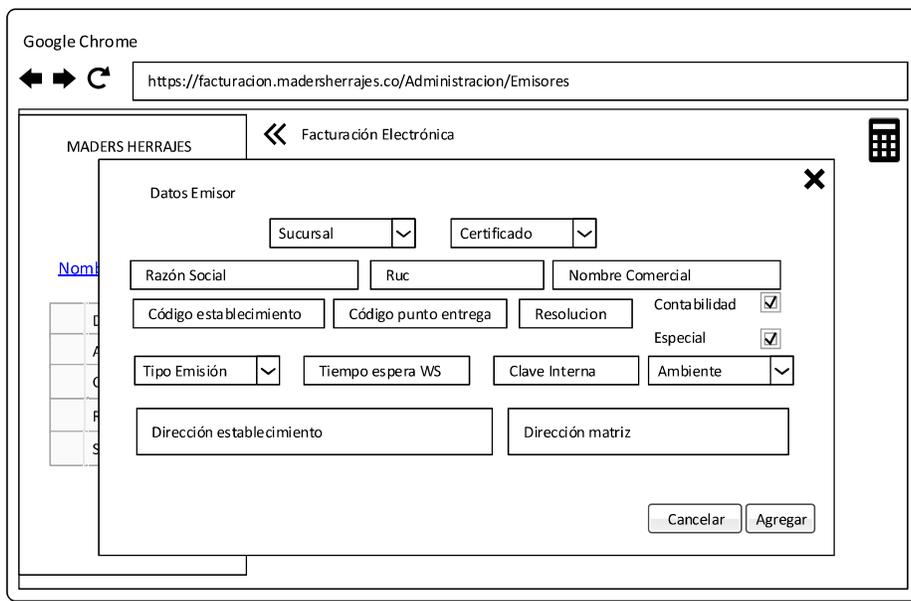


Figura 2.30 Vista para agregar un nuevo emisor al sistema.

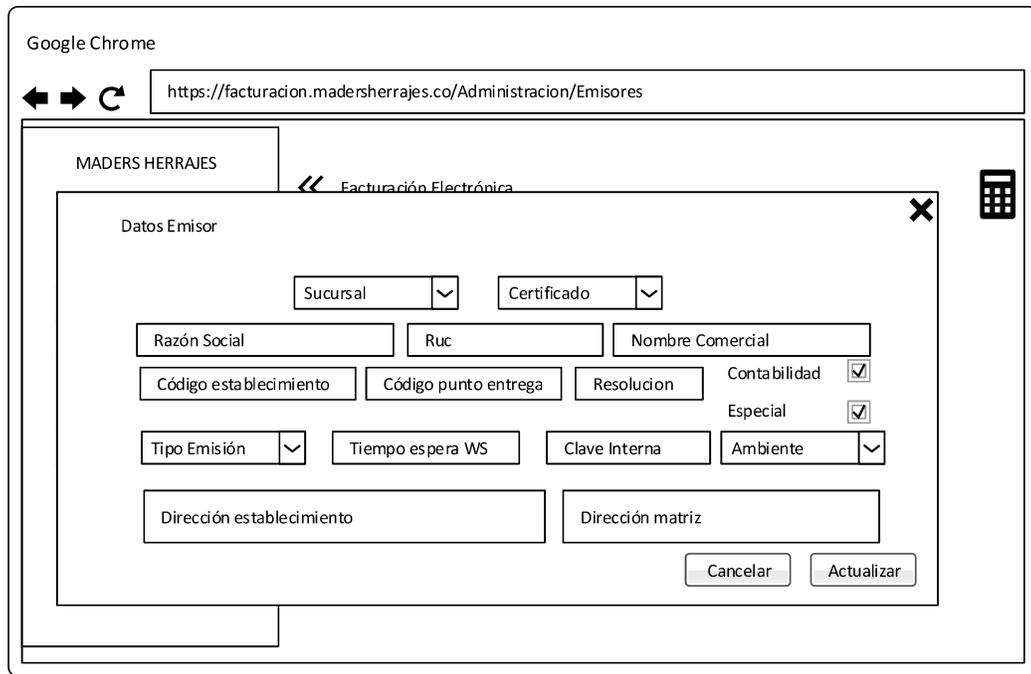


Figura 2.31 Vista para actualizar un emisor existente del sistema.

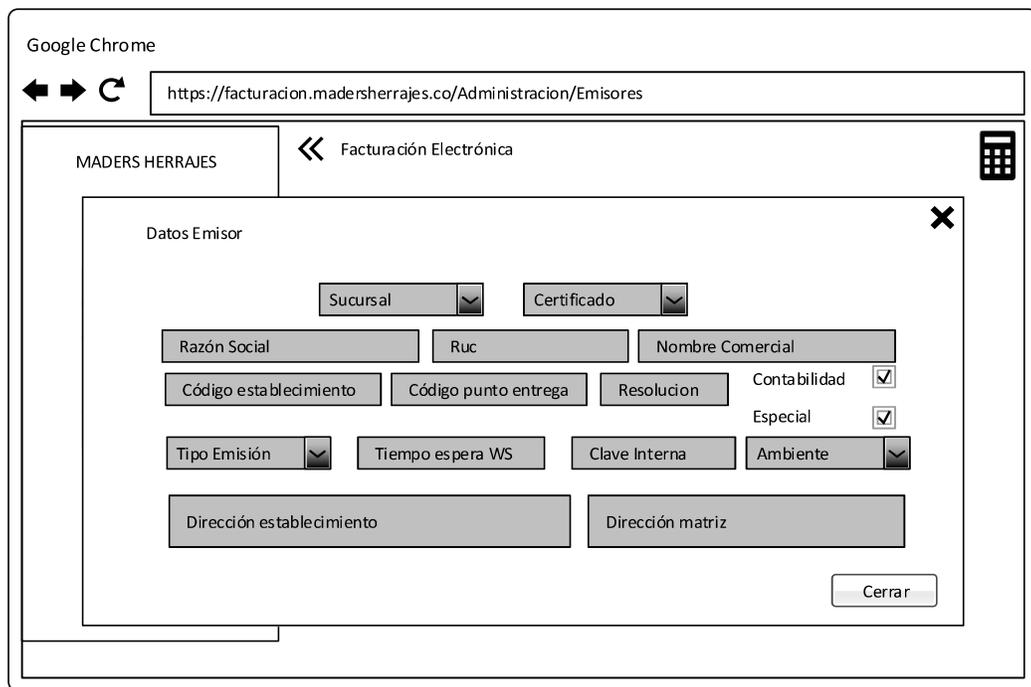


Figura 2.32 Vista para ver información de un emisor existente del sistema.

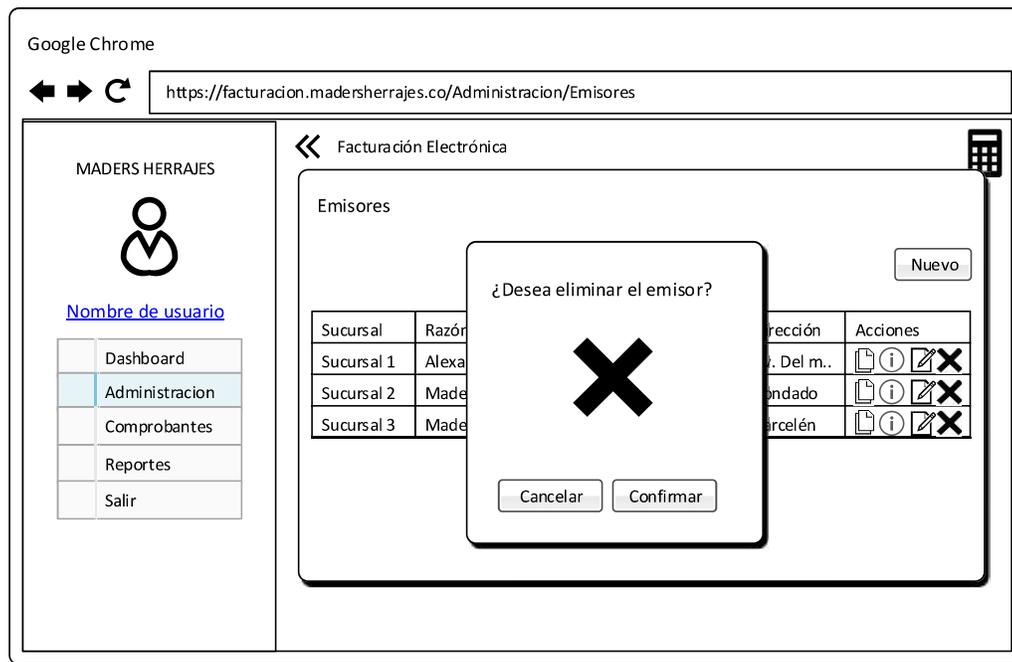


Figura 2.33 Vista para eliminar un emisor del sistema.

2.1.9.2.4. Administración de configuración de red

Esta vista permite administrar parámetros de configuración de servicios web y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.34 muestra la vista general de parámetros de configuración que visualiza el usuario.

Para los parámetros de configuración ingresados se han establecido las acciones ver y editar.

La Figura 2.35 presenta la vista para agregar una nueva configuración de red al sistema.

Las vistas para la visualización de los datos extendidos de la configuración de parámetros y modificación se presentan en la Figura 2.37 y Figura 2.36 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.2.5. Administración de impuestos del SRI

Esta vista permite administrar los impuestos que maneja el SRI y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.38 muestra la vista general de impuestos que visualiza el usuario. Se visualiza la información más relevante de los impuestos y las acciones disponibles para cada impuesto.

Para los impuestos ingresados se han establecido las acciones ver y editar.

La Figura 2.39 presenta la vista para agregar un nuevo impuesto al sistema.

Las vistas para la visualización de los datos extendidos del impuesto y modificación se presentan en la Figura 2.41 y Figura 2.40 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

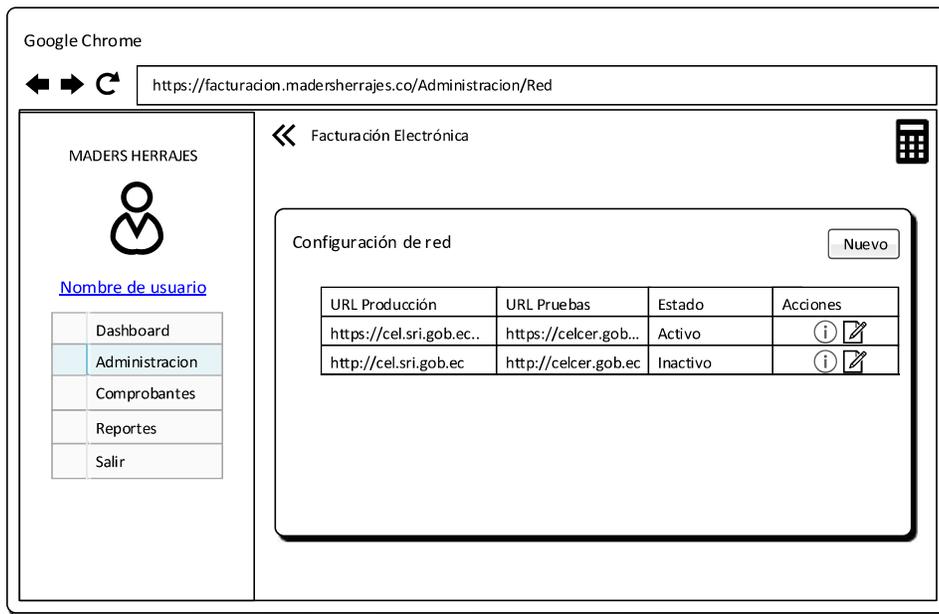


Figura 2.34 Vista general para administrar parámetros de configuración de red.

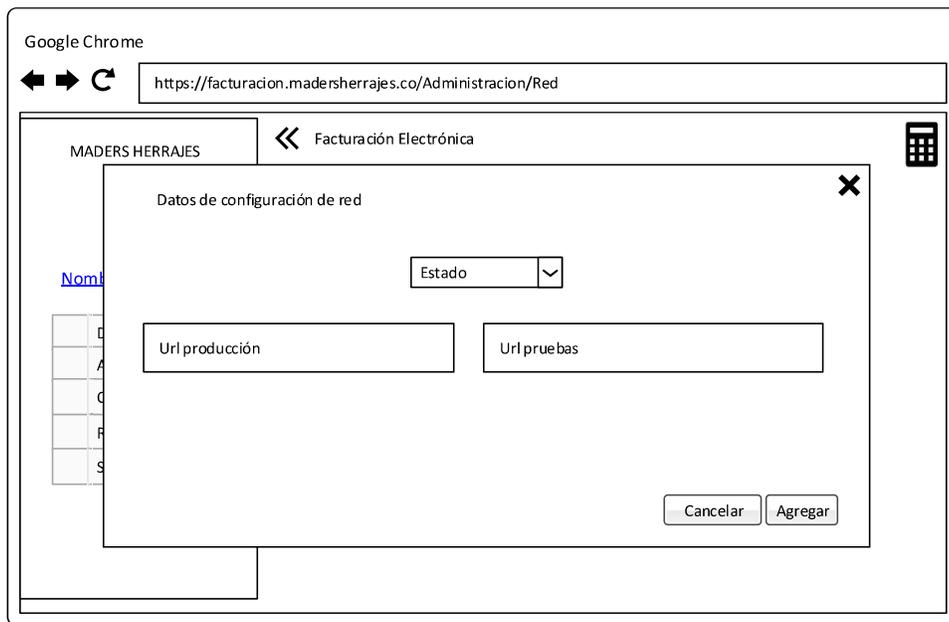


Figura 2.35 Vista para agregar un nuevo parámetro de configuración de red al sistema.

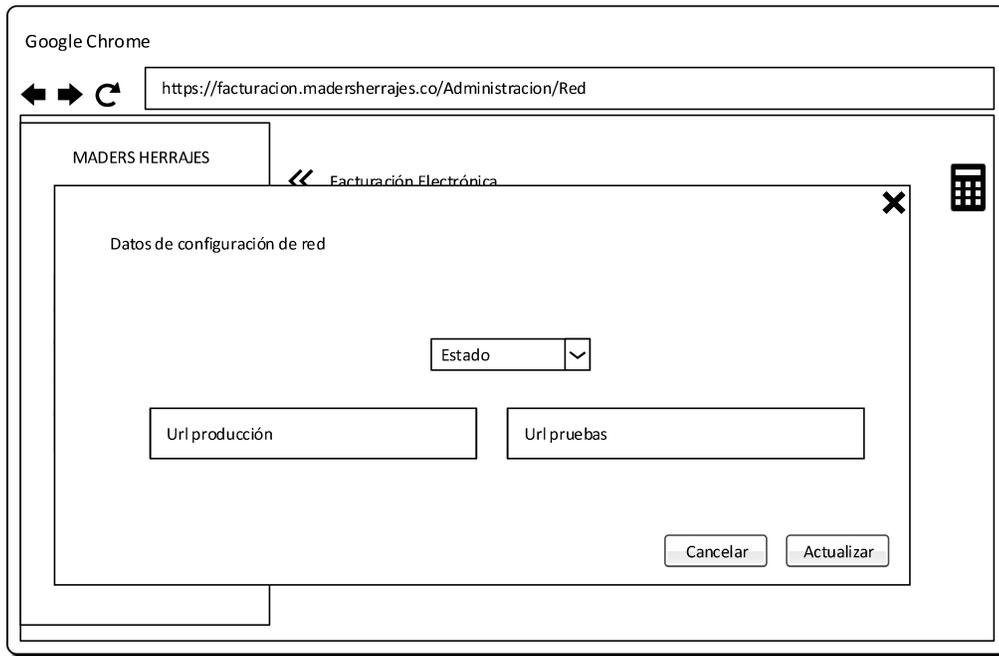


Figura 2.36 Vista para actualizar un parámetro de configuración de red existente del sistema.

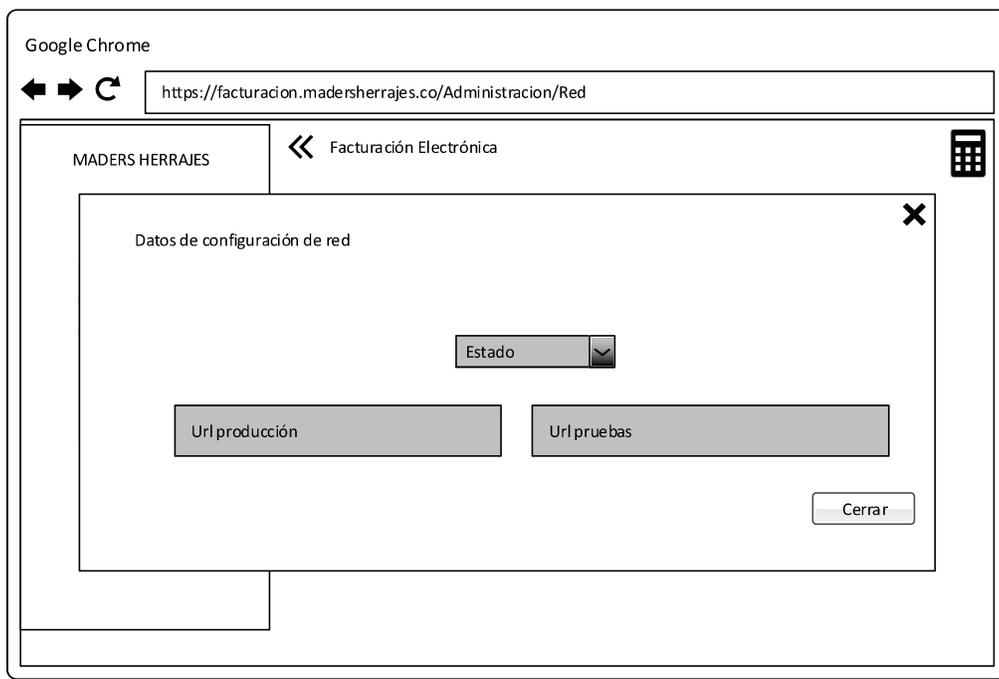


Figura 2.37 Vista para ver información de un parámetro de configuración de red existente del sistema.

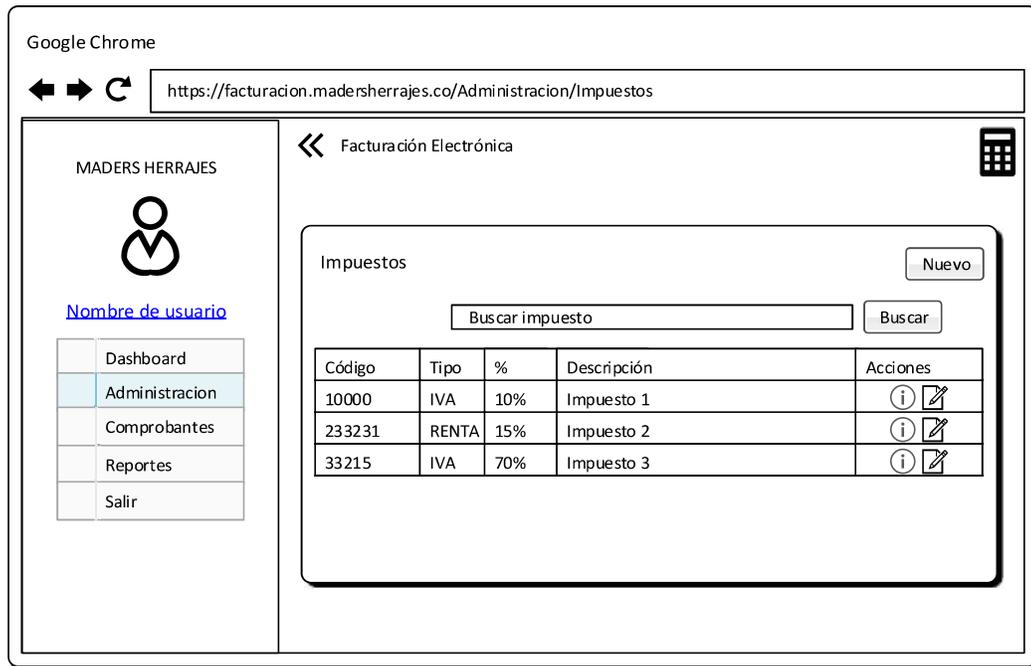


Figura 2.38 Vista general para administrar impuestos.

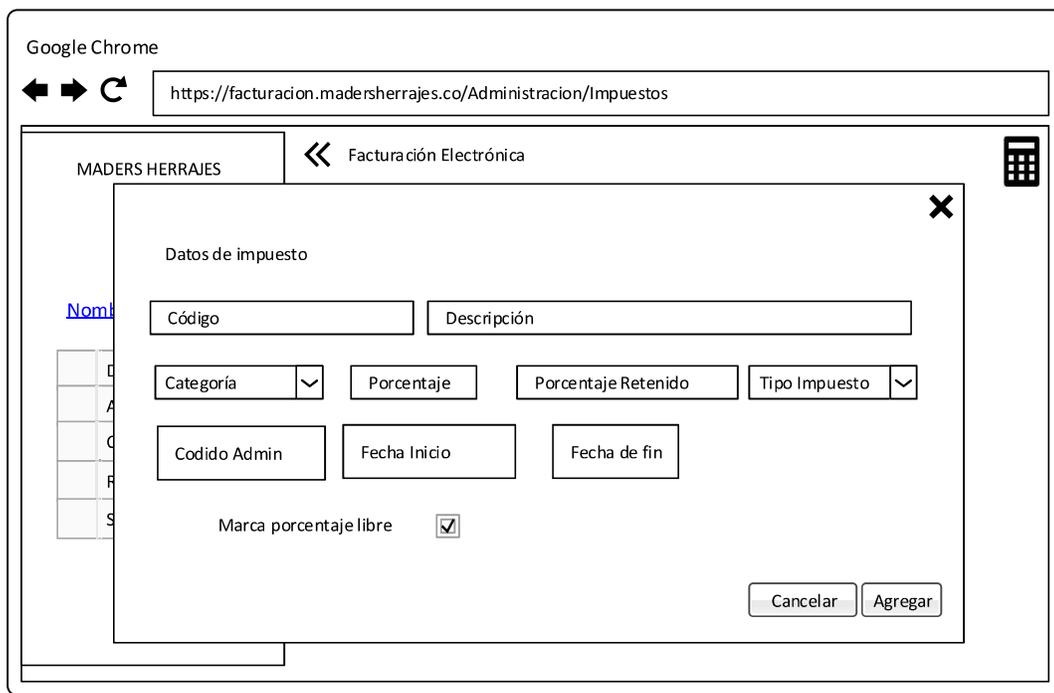


Figura 2.39 Vista para agregar un nuevo impuesto al sistema.

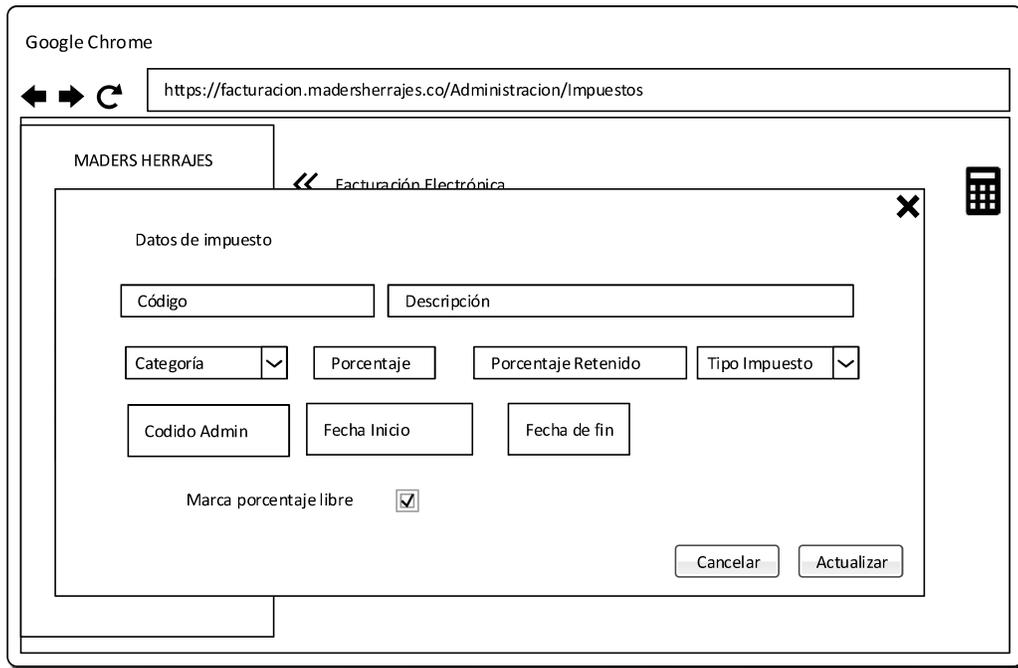


Figura 2.40 Vista para actualizar un impuesto existente del sistema.

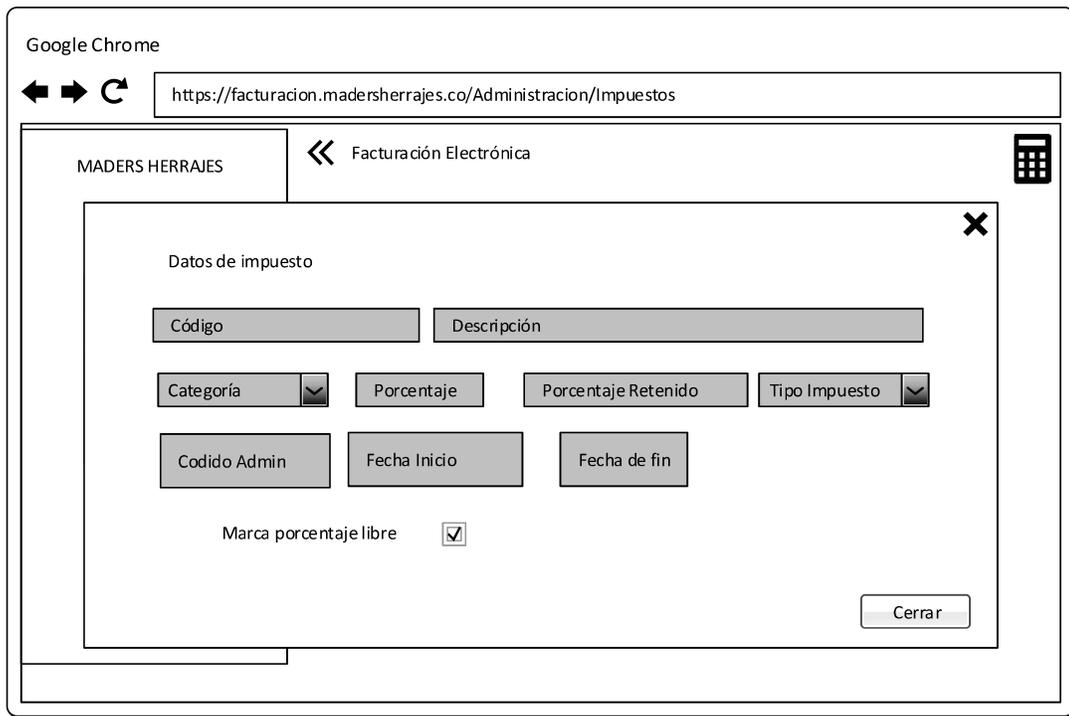


Figura 2.41 Vista para ver información de un impuesto existente del sistema.

2.1.9.2.6. Administración de formas de pago del SRI

Esta vista permite administrar las formas de pago que maneja el SRI y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.42 muestra la vista general de formas de pago que visualiza el usuario. Se visualiza la información más relevante de los impuestos y las acciones disponibles para cada forma de pago.

Para las formas de pago ingresadas se han establecido las acciones ver y editar.

La Figura 2.43 presenta la vista para agregar una nueva forma de pago al sistema.

Las vistas para la visualización de los datos extendidos de la forma de pago y modificación se presentan en la Figura 2.45 y Figura 2.44 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.2.7. Administración de tipos de comprobantes

Esta vista permite administrar los tipos de comprobantes manejados por el sistema y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.46 muestra la vista general de tipos de comprobantes que visualiza el usuario. Se visualiza la información más relevante de los impuestos y las acciones disponibles para cada tipo de comprobante.

Para tipos de comprobantes ingresados se han establecido las acciones ver, editar.

Las vistas para la visualización de los datos extendidos del tipo de comprobante y modificación se presentan en la Figura 2.48 y Figura 2.47 respectivamente.

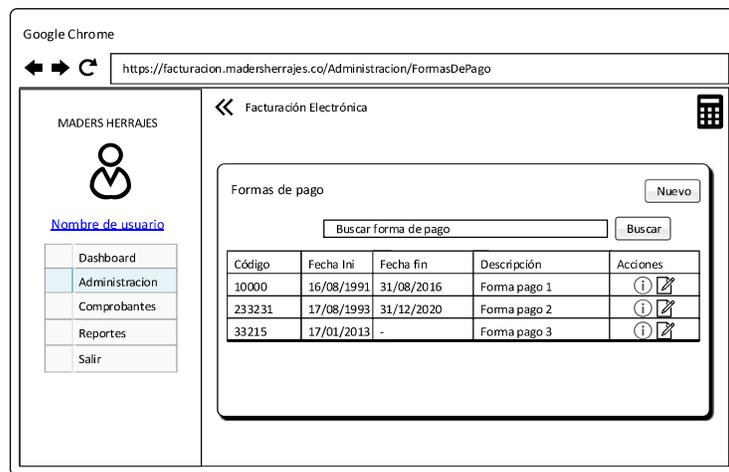


Figura 2.42 Vista general para administrar formas de pago.

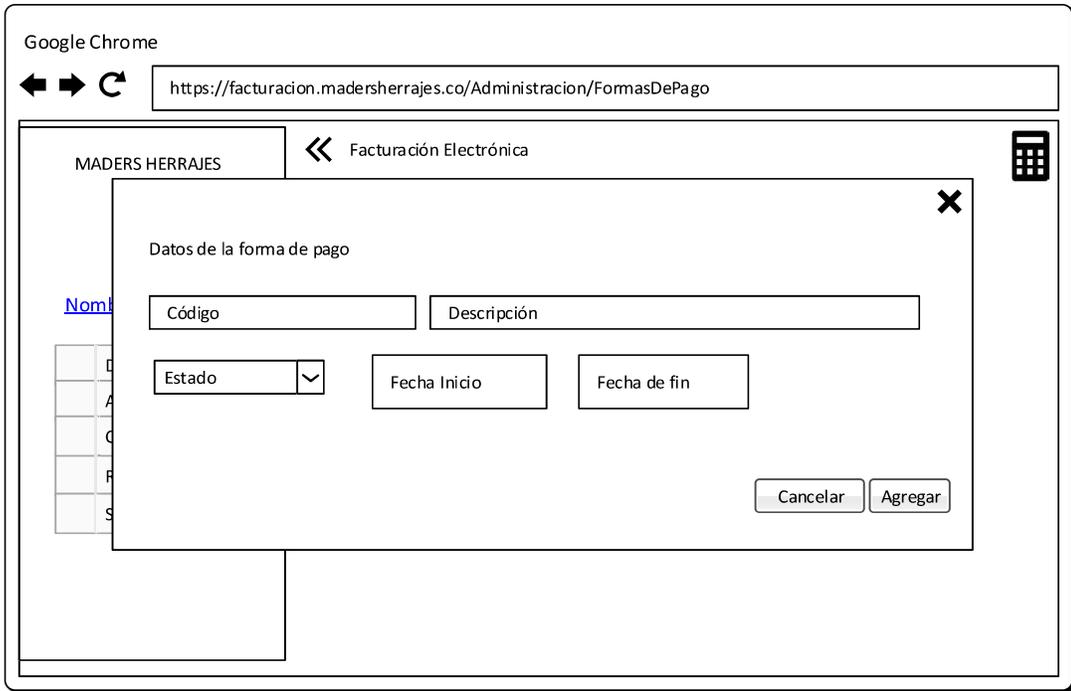


Figura 2.43 Vista para agregar una nueva forma de pago al sistema.

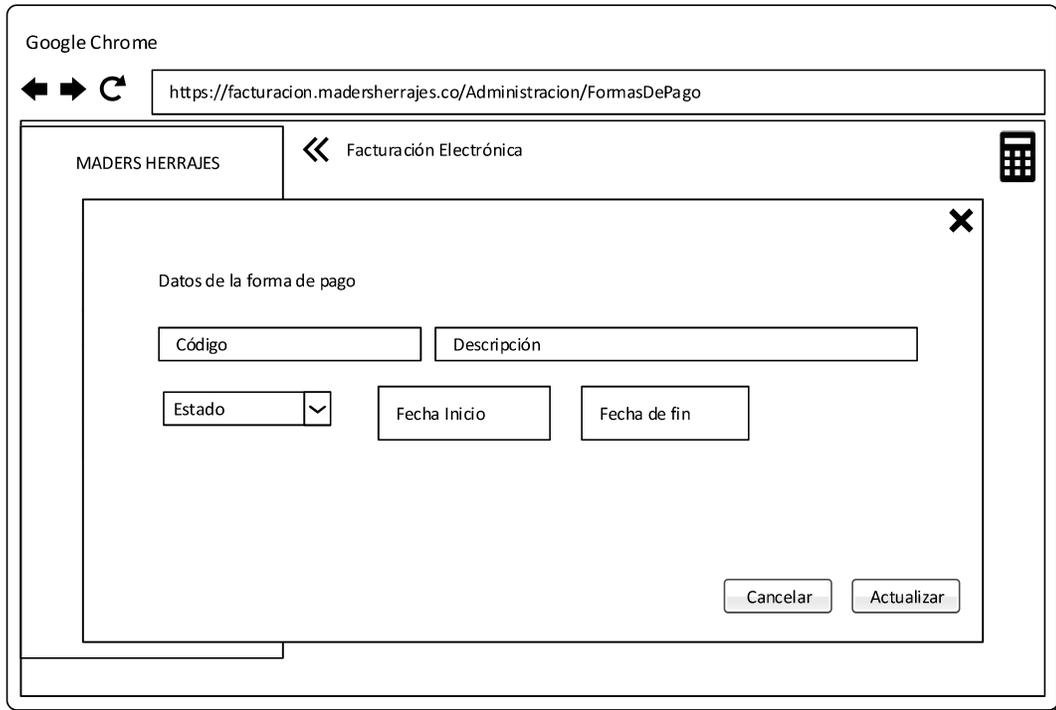


Figura 2.44 Vista para actualizar una forma de pago existente del sistema.

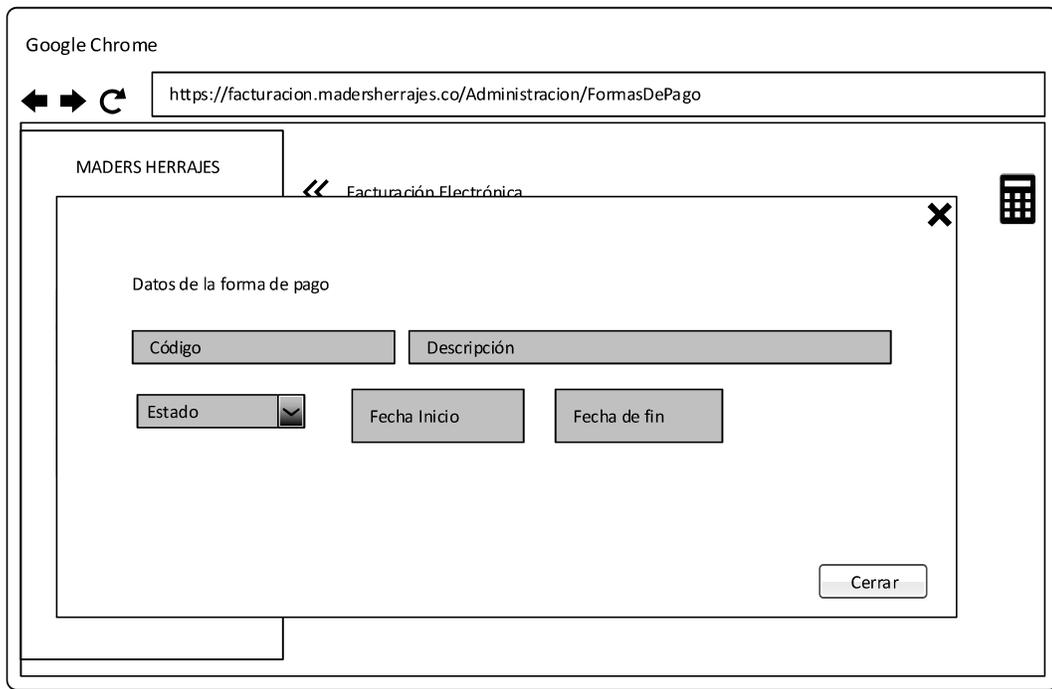


Figura 2.45 Vista para ver información de una forma de pago existente del sistema.

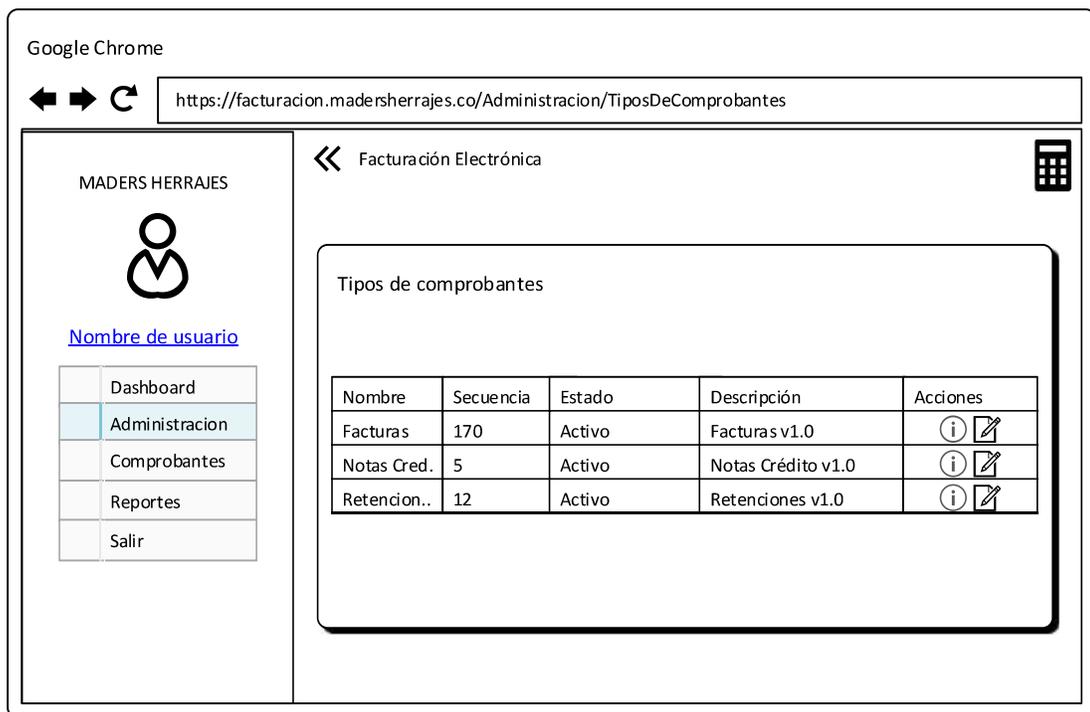


Figura 2.46 Vista general para administrar tipos de comprobantes.

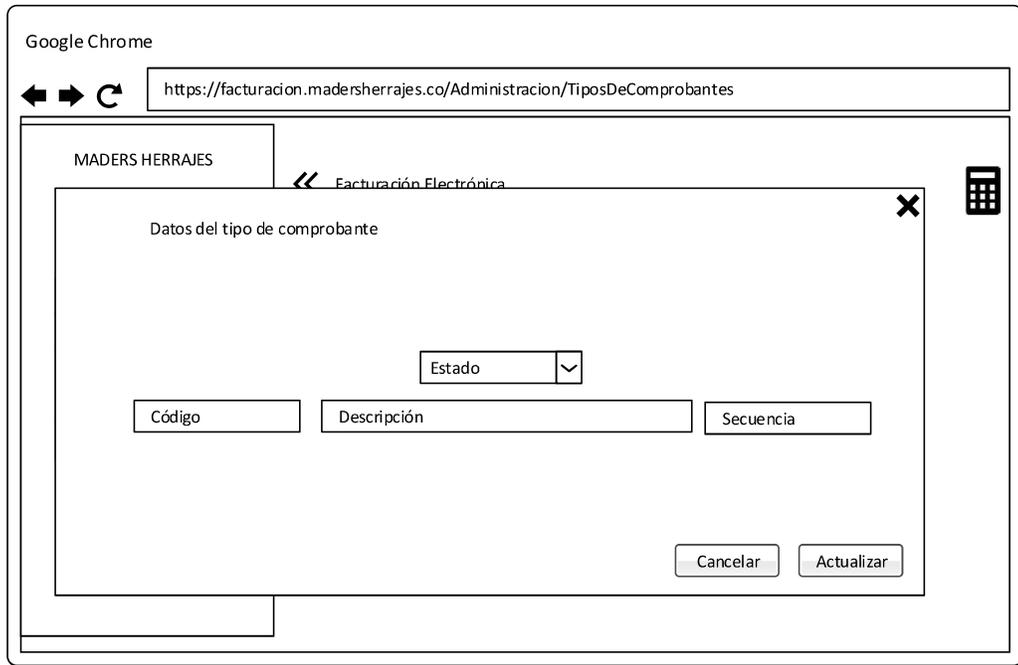


Figura 2.47 Vista para actualizar un tipo de comprobante existente del sistema.

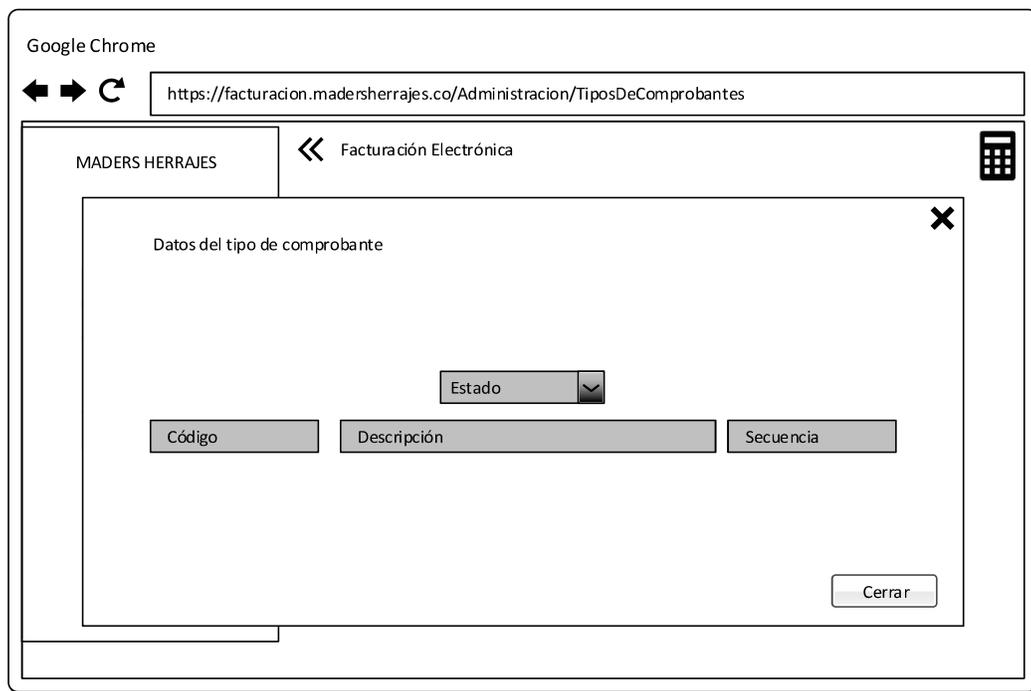


Figura 2.48 Vista para ver información de un tipo de comprobante existente del sistema.

2.1.9.2.8. Administración de sucursales

Esta vista permite administrar las sucursales del sistema y es accesible únicamente por el rol de administrador, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.49 muestra la vista general de tipos de comprobantes que visualiza el usuario. Se visualiza la información más relevante de los impuestos y las acciones disponibles para cada sucursal.

Para sucursales ingresadas se han establecido las acciones ver, editar y eliminar.

La Figura 2.50 presenta la vista para agregar una nueva sucursal al sistema.

Las vistas para la visualización de los datos extendidos de sucursales y modificación se presentan en la Figura 2.52 y Figura 2.51 respectivamente.

La vista para eliminar la sucursal se presenta en la Figura 2.53. Únicamente se podrán eliminar las sucursales en las que no se hayan generado comprobantes.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.2.9. Administración de clientes

Esta vista permite administrar la lista de clientes del sistema y es accesible por los roles de administrador, administrador de sucursal y usuario.

La Figura 2.54 muestra la vista general de los clientes que visualiza el usuario. Se visualiza la información más relevante de los clientes y las acciones disponibles para cada cliente.

Para clientes ingresados se han establecido las acciones ver, editar y eliminar.

La Figura 2.55 presenta la vista para el ingreso de nuevos clientes al sistema.

Las vistas para la visualización de los datos extendidos de clientes y modificación se presentan en la Figura 2.57 y Figura 2.56 respectivamente.

La vista para eliminar el cliente se presenta en la Figura 2.58. Únicamente se podrán eliminar los clientes con los que no se hayan generado comprobantes electrónicos.

Para el ingreso se valida que los campos estén llenos correctamente.

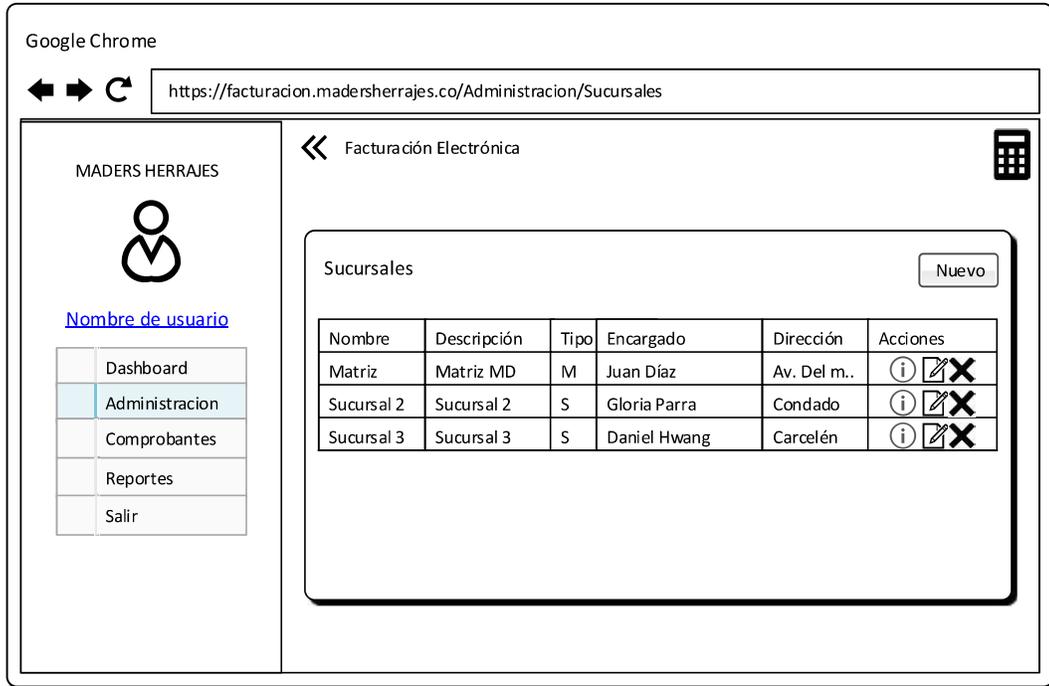


Figura 2.49 Vista general para administrar sucursales.

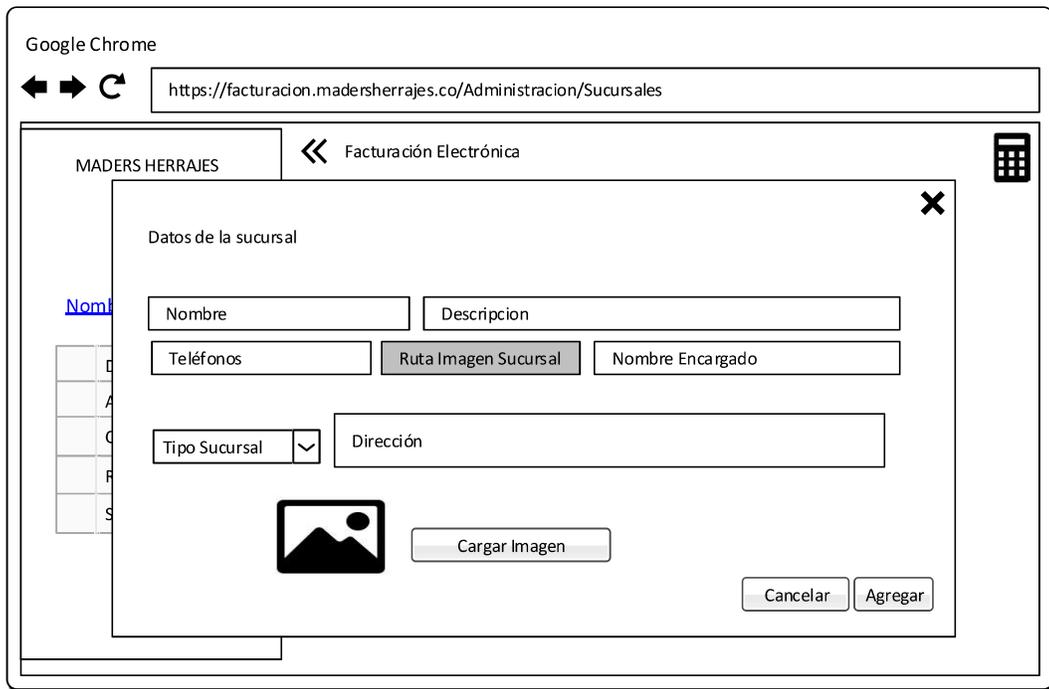


Figura 2.50 Vista para agregar una nueva sucursal al sistema.

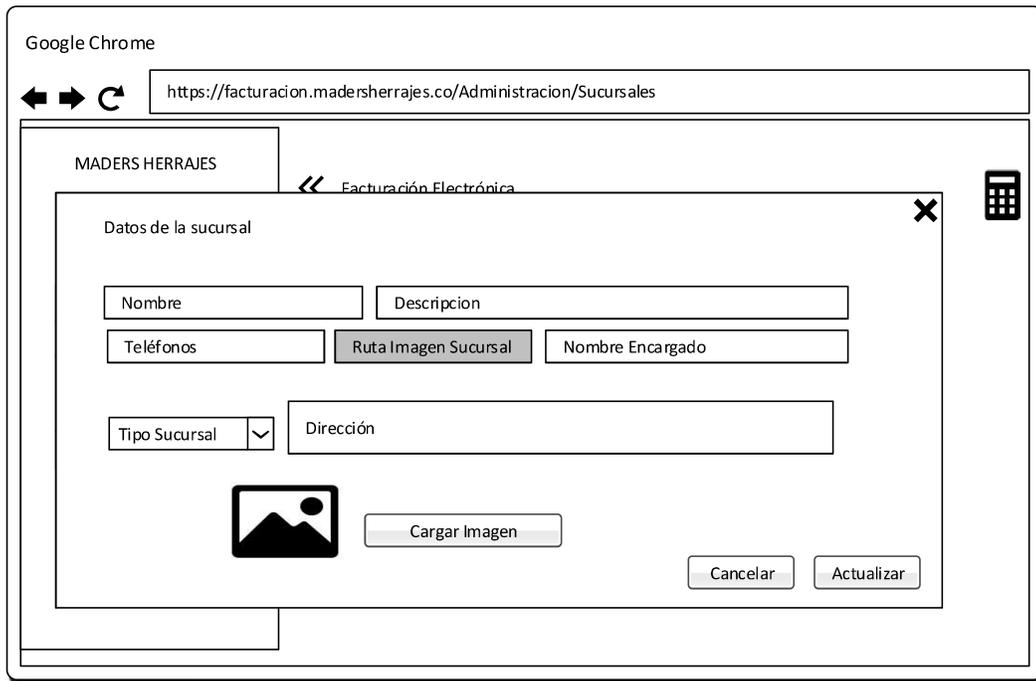


Figura 2.51 Vista para actualizar una sucursal existente del sistema.

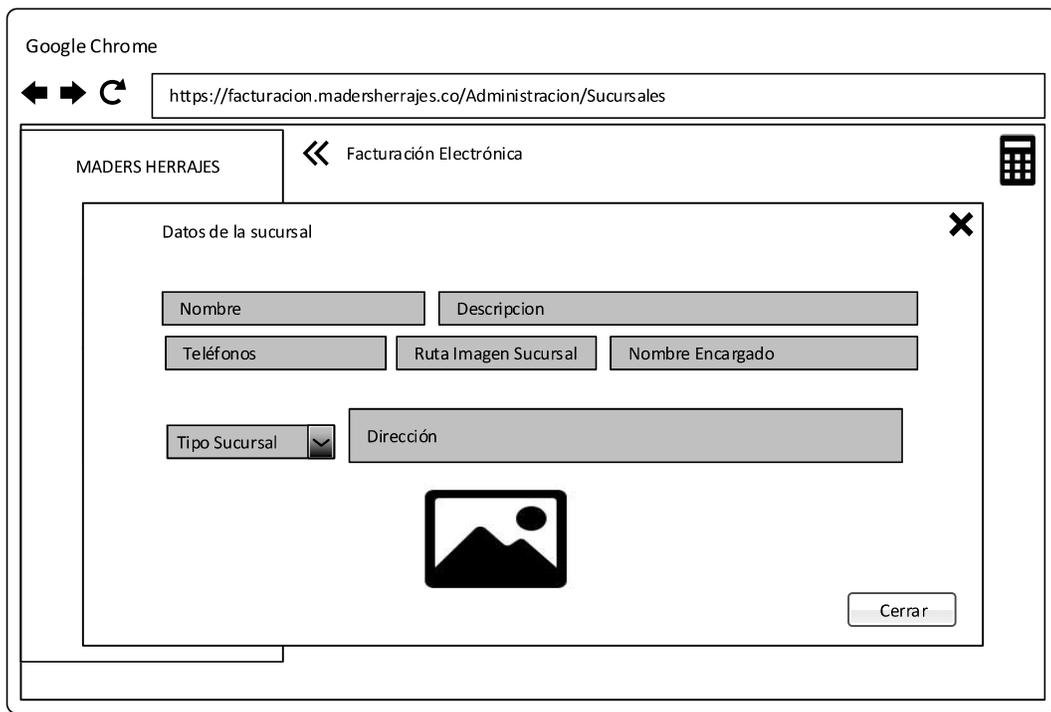


Figura 2.52 Vista para ver información de una sucursal existente del sistema.

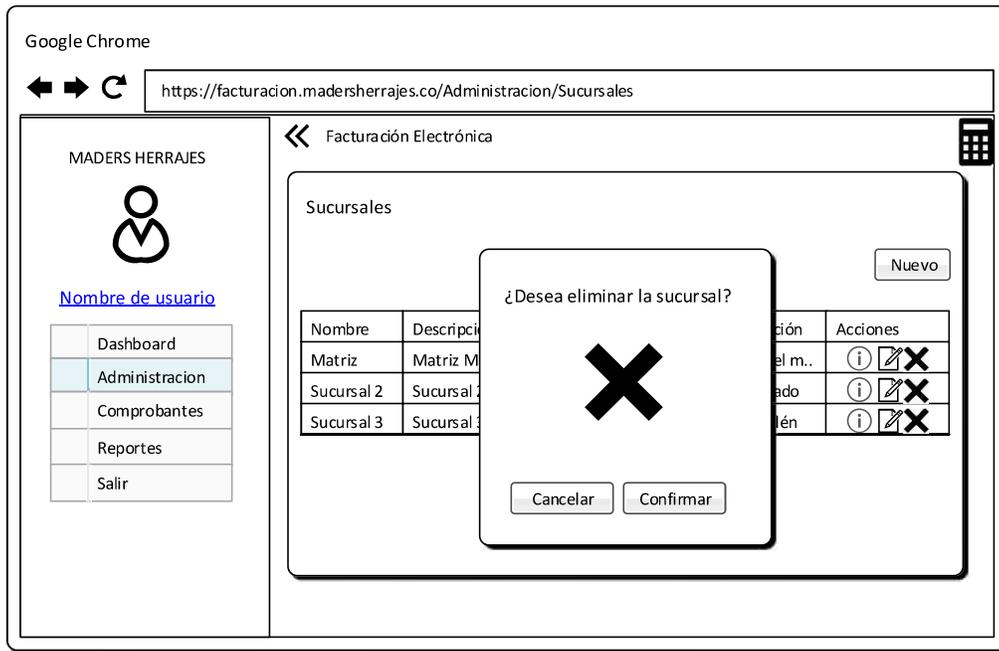


Figura 2.53 Vista para eliminar una sucursal del sistema.

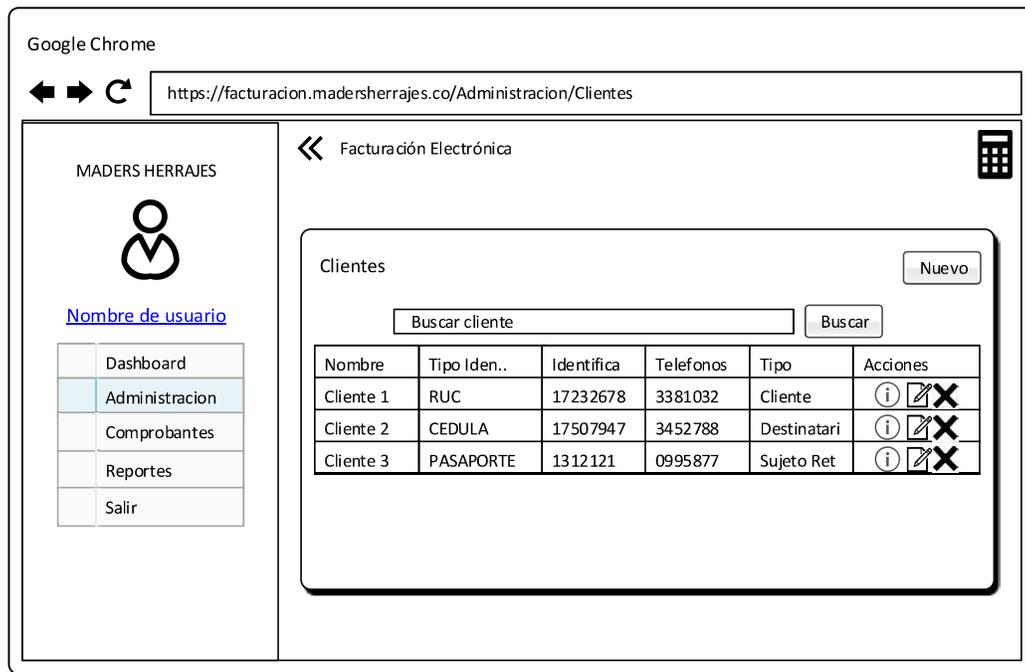


Figura 2.54 Vista general para administrar clientes.

Google Chrome
 https://facturacion.madersherrajes.co/Administracion/Clientes

MADERS HERRAJES Facturación Electrónica

Datos Cliente

Estado

Razón Social / Nombre Ruc Tipo Identificación

Tipo de cliente

Cliente Destinatario Sujeto retenido

Teléfonos Correos

Dirección

Cancelar Agregar

Figura 2.55 Vista para agregar un nuevo cliente al sistema.

Google Chrome
 https://facturacion.madersherrajes.co/Administracion/Clientes

MADERS HERRAJES Facturación Electrónica

Datos Cliente

Estado

Razón Social / Nombre Ruc Tipo Identificación

Tipo de cliente

Cliente Destinatario Sujeto retenido

Teléfonos Correos

Dirección

Cancelar Actualizar

Figura 2.56 Vista para actualizar un cliente existente del sistema.

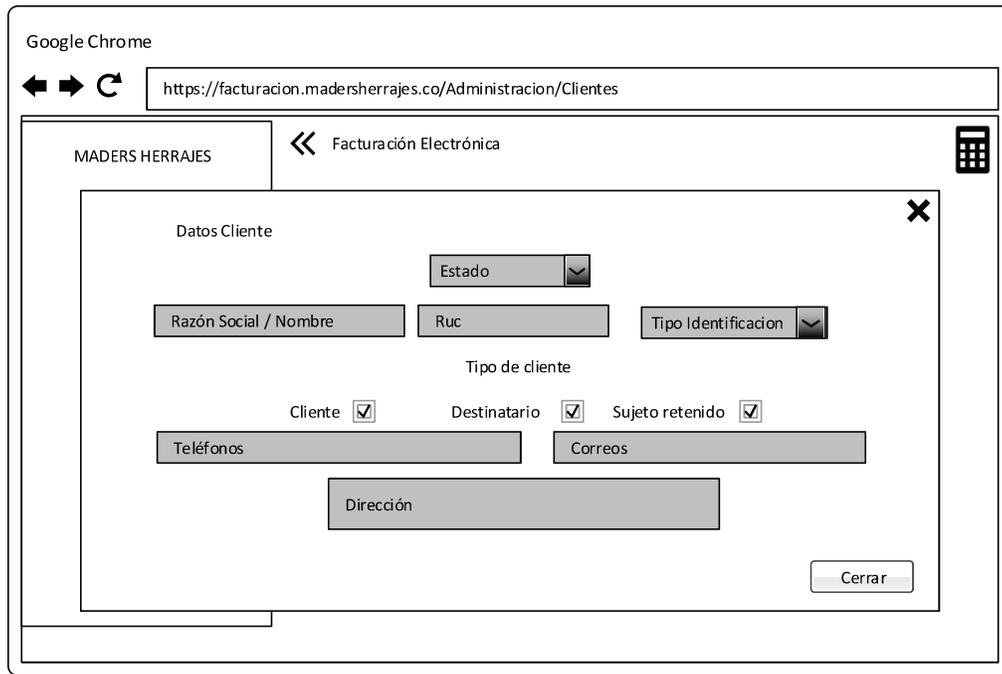


Figura 2.57 Vista para ver información de un cliente existente del sistema.

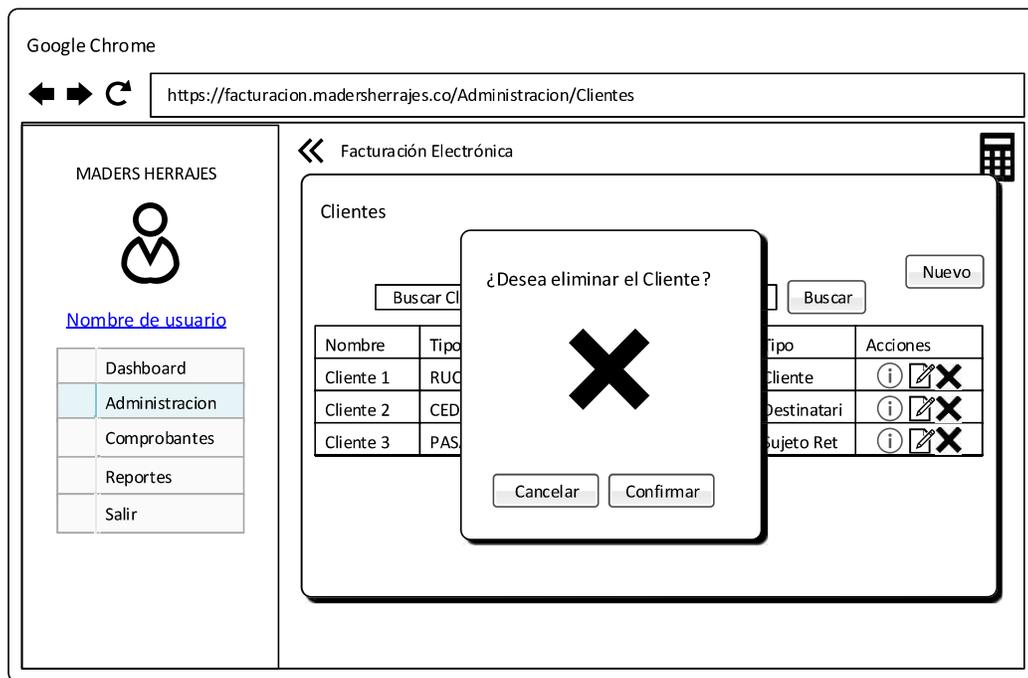


Figura 2.58 Vista para eliminar un cliente del sistema.

2.1.9.2.10. Administración de transportistas

Esta vista permite administrar la lista de transportistas del sistema y es accesible por los roles de administrador, administrador de sucursal y usuarios

La Figura 2.59 muestra la vista general de los transportistas que visualiza el usuario. Se visualiza la información más relevante de los transportistas y las acciones disponibles para cada transportista.

Para transportistas ingresados se han establecido las acciones ver, editar y eliminar.

La Figura 2.60 presenta la vista para agregar un nuevo transportista al sistema.

La vista para eliminar el transportista se presenta en la Figura 2.63. Únicamente se podrán eliminar los transportistas con los que no se hayan generado guías de remisión.

Las vistas para la visualización de los datos extendidos de transportistas y modificación se presentan en la Figura 2.62 y Figura 2.61 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

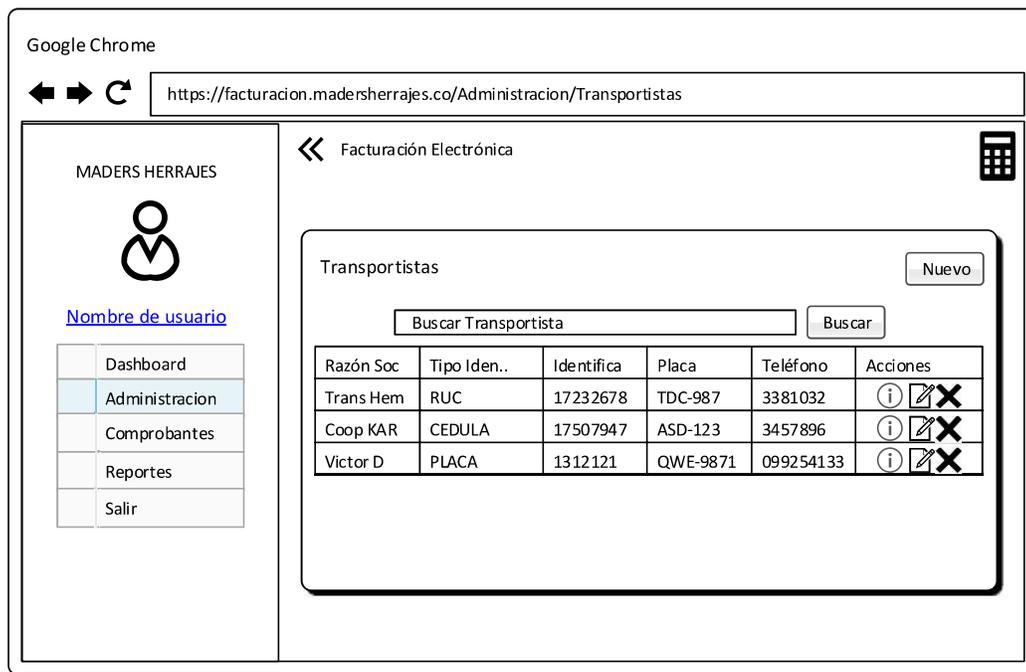


Figura 2.59 Vista general para administrar transportistas.

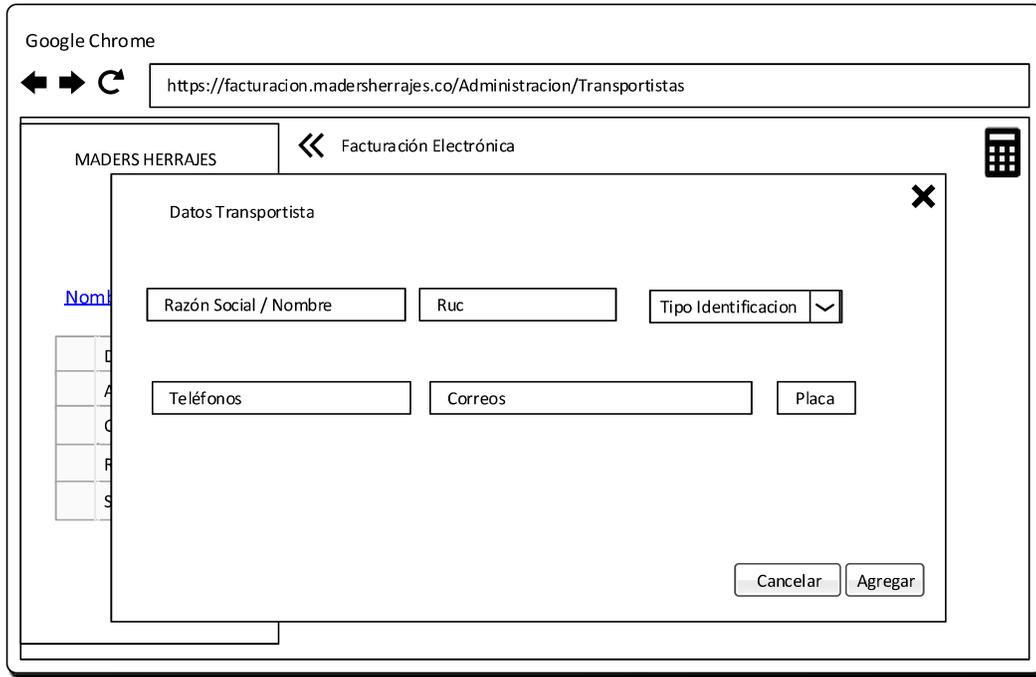


Figura 2.60 Vista para agregar un nuevo transportista al sistema.

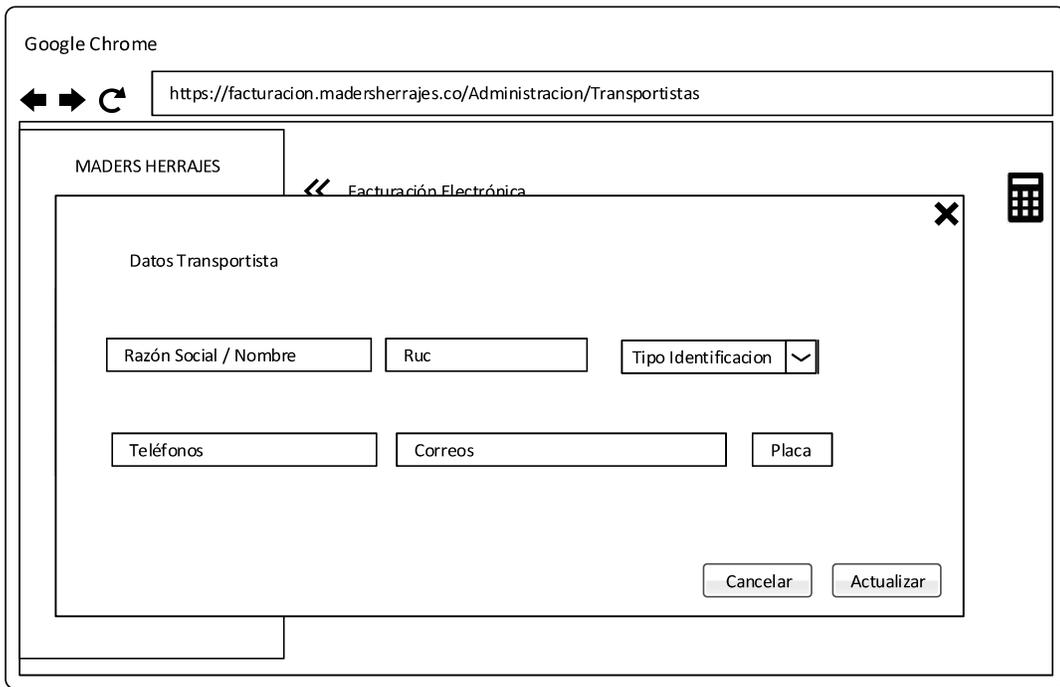


Figura 2.61 Vista para actualizar un transportista existente del sistema.

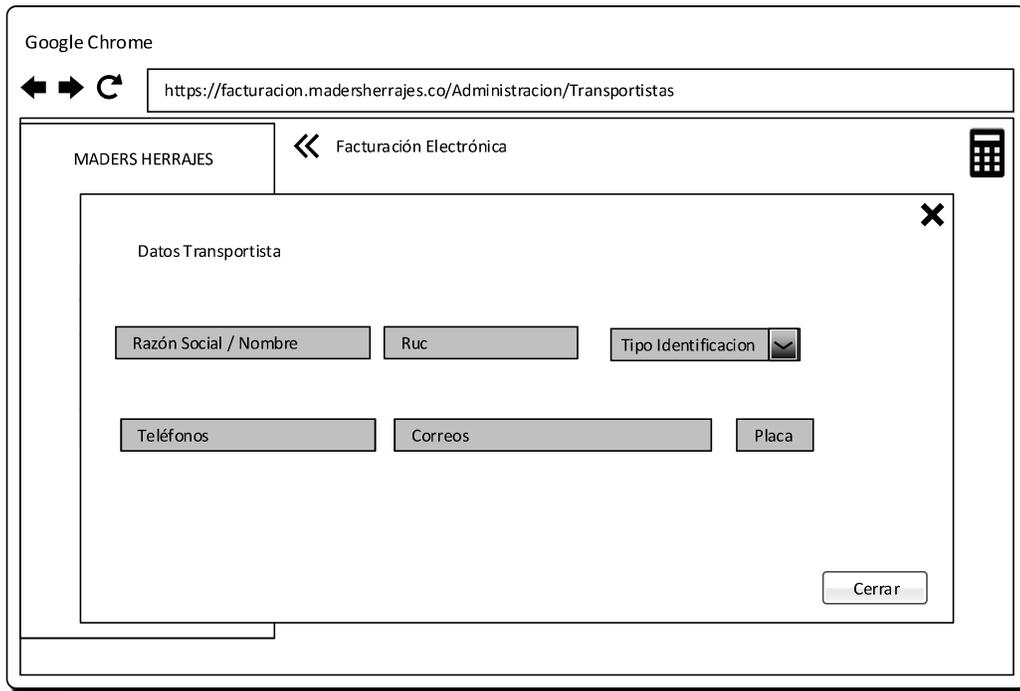


Figura 2.62 Vista para ver información de un transportista existente del sistema.

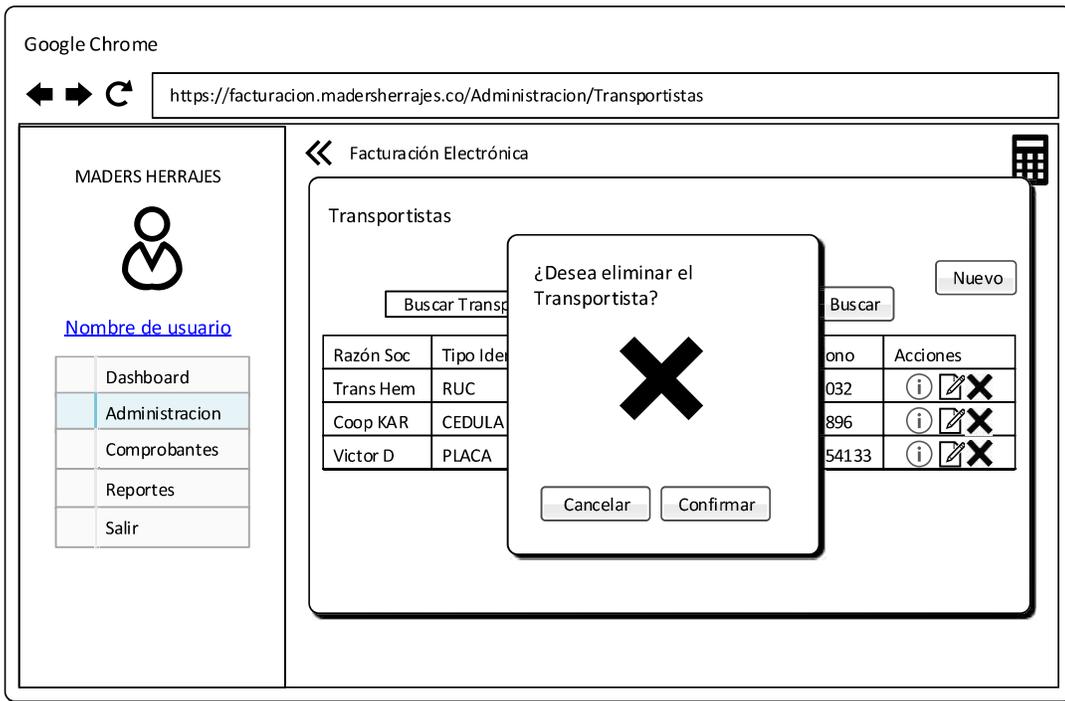


Figura 2.63 Vista para eliminar un transportista del sistema.

2.1.9.2.11. Administración de proveedores

Esta vista permite administrar la lista de proveedores del sistema y es accesible por los roles de administrador, administrador de sucursal, el resto de roles no pueden ver o ingresar a esta vista.

La Figura 2.64 muestra la vista general de los proveedores que visualiza el usuario. Se visualiza la información más relevante de los proveedores y las acciones disponibles para cada proveedor.

Para proveedores ingresados se han establecido las acciones ver, editar y eliminar.

La Figura 2.65 presenta la vista para agregar un nuevo proveedor al sistema.

La vista para eliminar el proveedor se presenta en la Figura 2.68.

Las vistas para la visualización de los datos extendidos de proveedores y modificación se presentan en la Figura 2.67 y Figura 2.66 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.2.12. Administración de categorías de productos

Esta vista permite administrar las categorías de productos del sistema y es accesible por los roles de administrador, administrador de sucursal.

La Figura 2.69 muestra la vista general de las categorías de productos que visualiza el usuario.

Para categorías de productos ingresadas se han establecido las acciones ver, editar y eliminar.

La Figura 2.70 presenta la vista para agregar una nueva categoría al sistema.

La vista para eliminar la categoría de producto se presenta en la Figura 2.73. No se pueden eliminar las categorías de productos que tengan productos agregados en ellas.

Las vistas para la visualización de los datos extendidos de categorías de productos y modificación se presentan en la Figura 2.72 y Figura 2.71 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

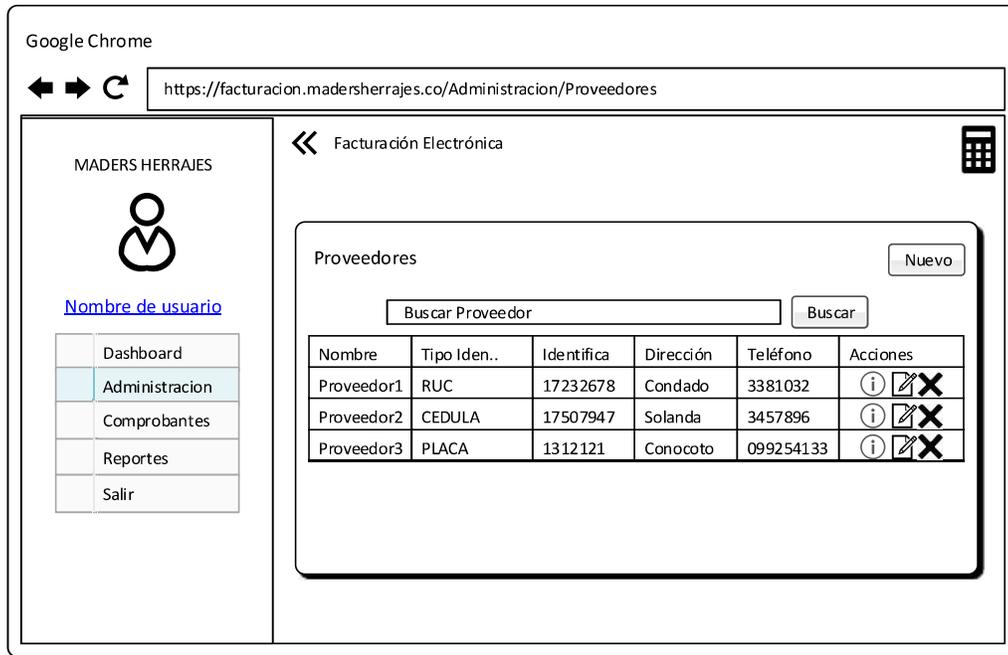


Figura 2.64 Vista general para administrar proveedores.

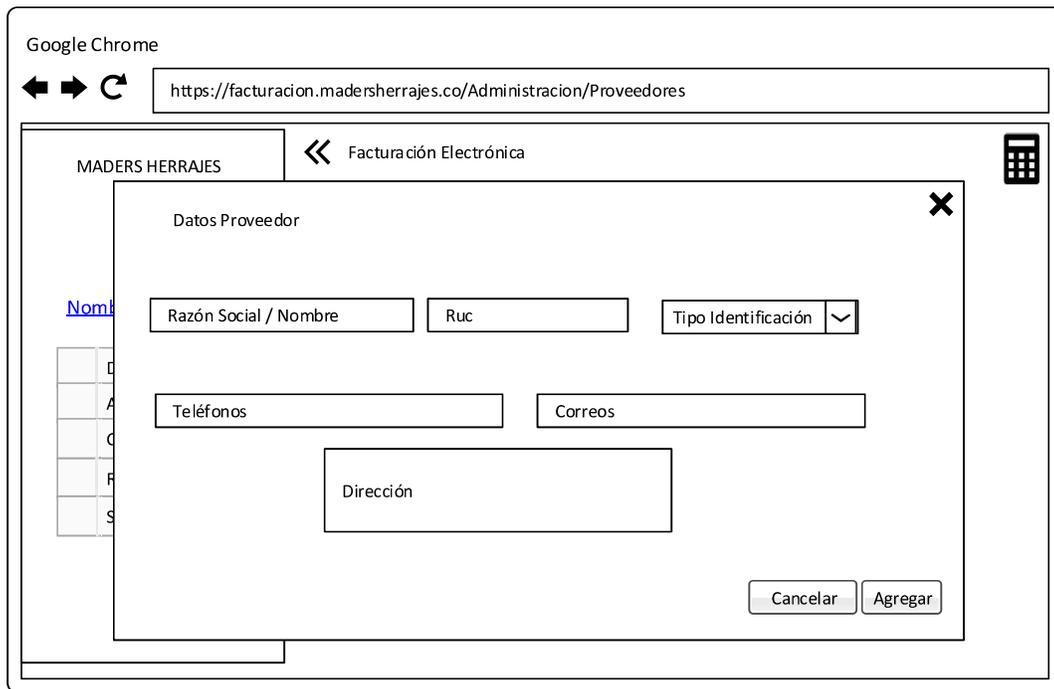


Figura 2.65 Vista para agregar un nuevo proveedor al sistema.

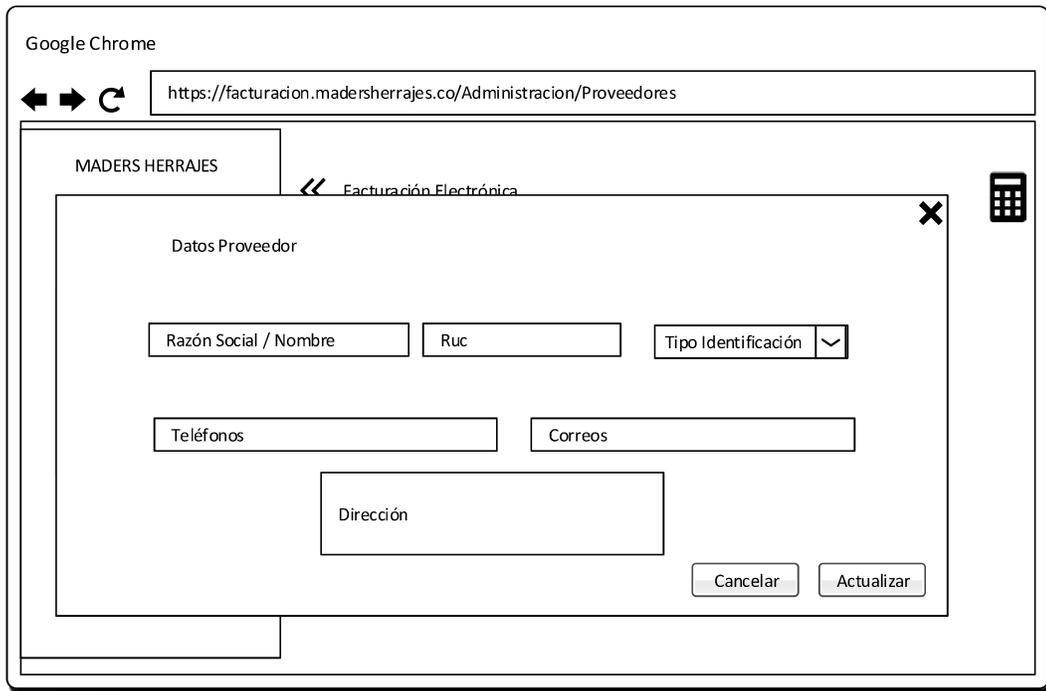


Figura 2.66 Vista para actualizar un proveedor existente del sistema.

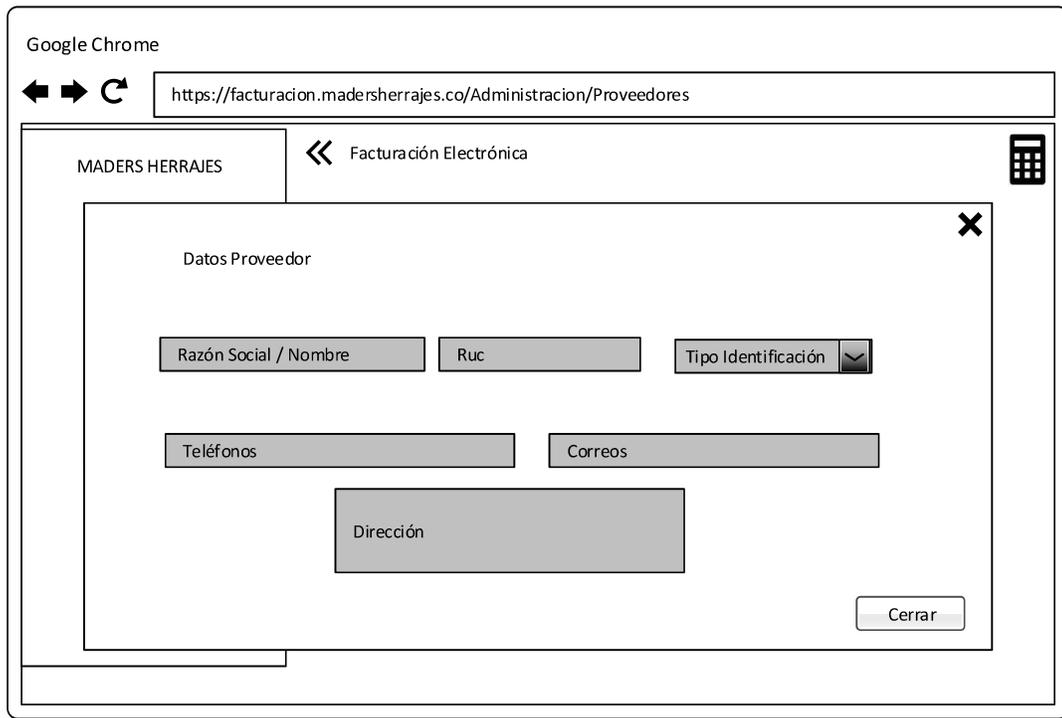


Figura 2.67 Vista para ver información de un proveedor existente del sistema.

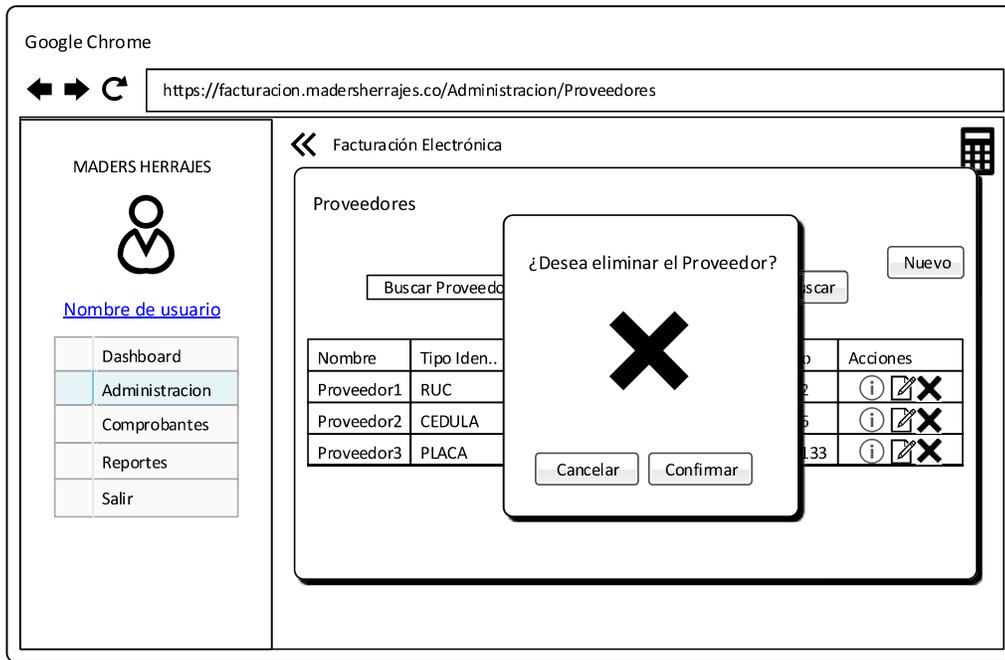


Figura 2.68 Vista para eliminar un proveedor del sistema.

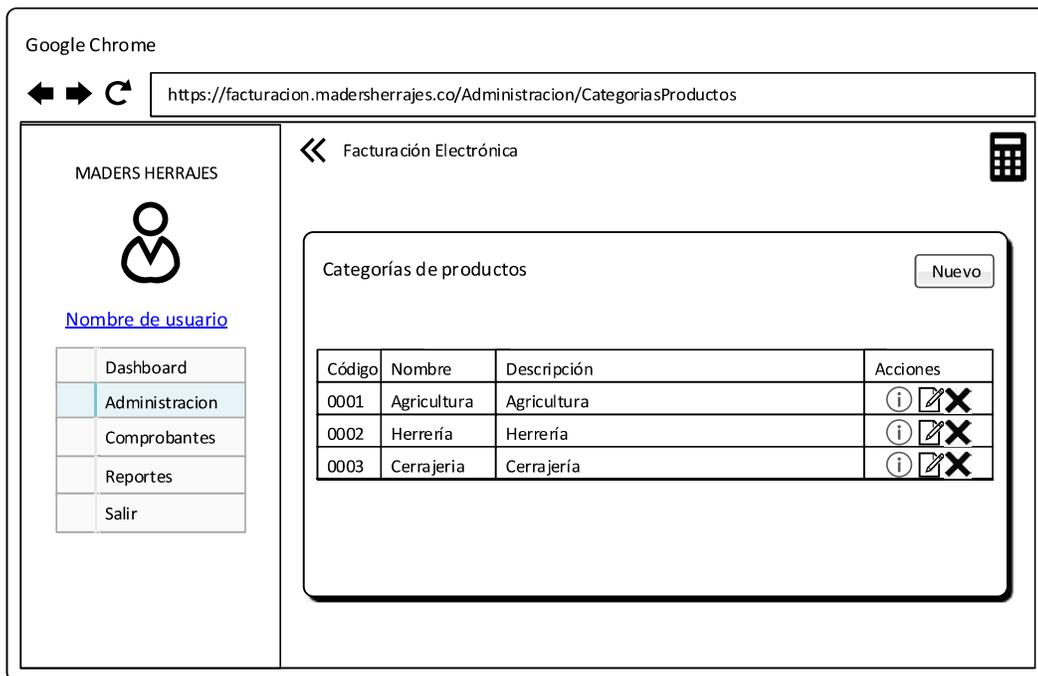


Figura 2.69 Vista general para administrar categorías de productos.

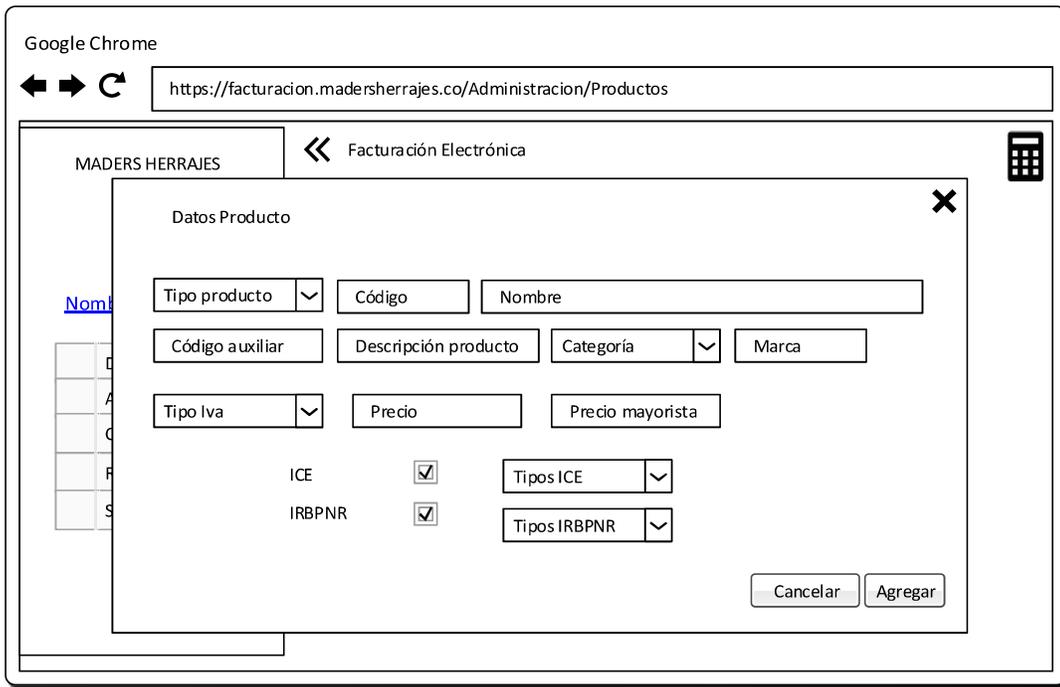


Figura 2.70 Vista para agregar una categoría de productos al sistema.

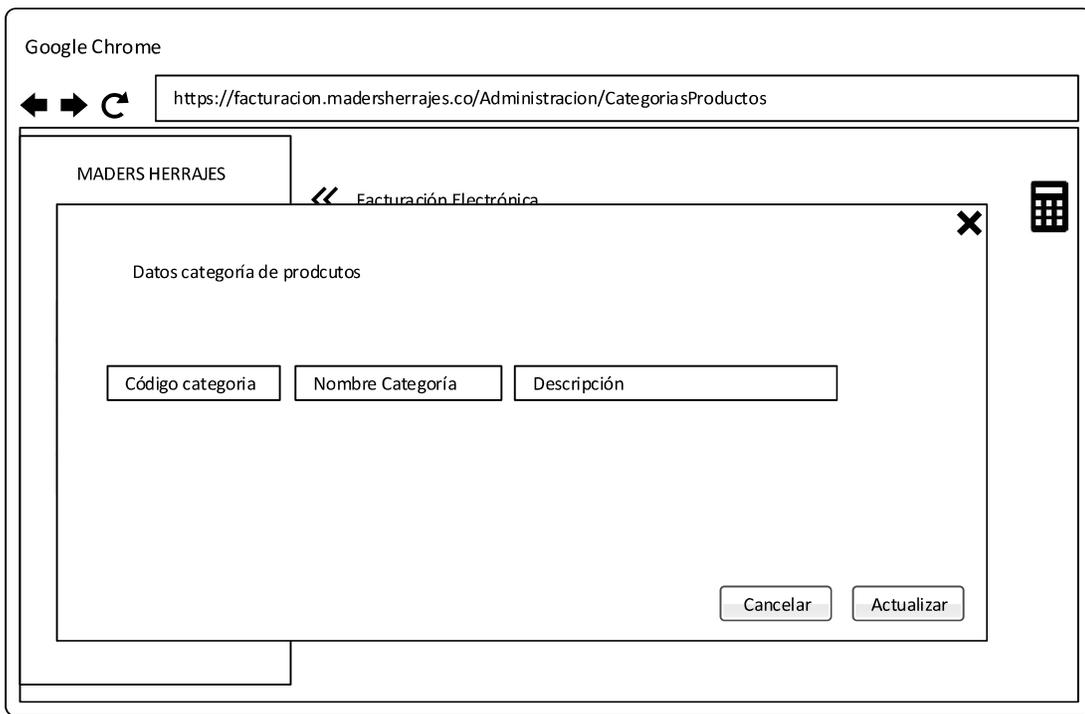


Figura 2.71 Vista para actualizar una categoría de productos existente del sistema.

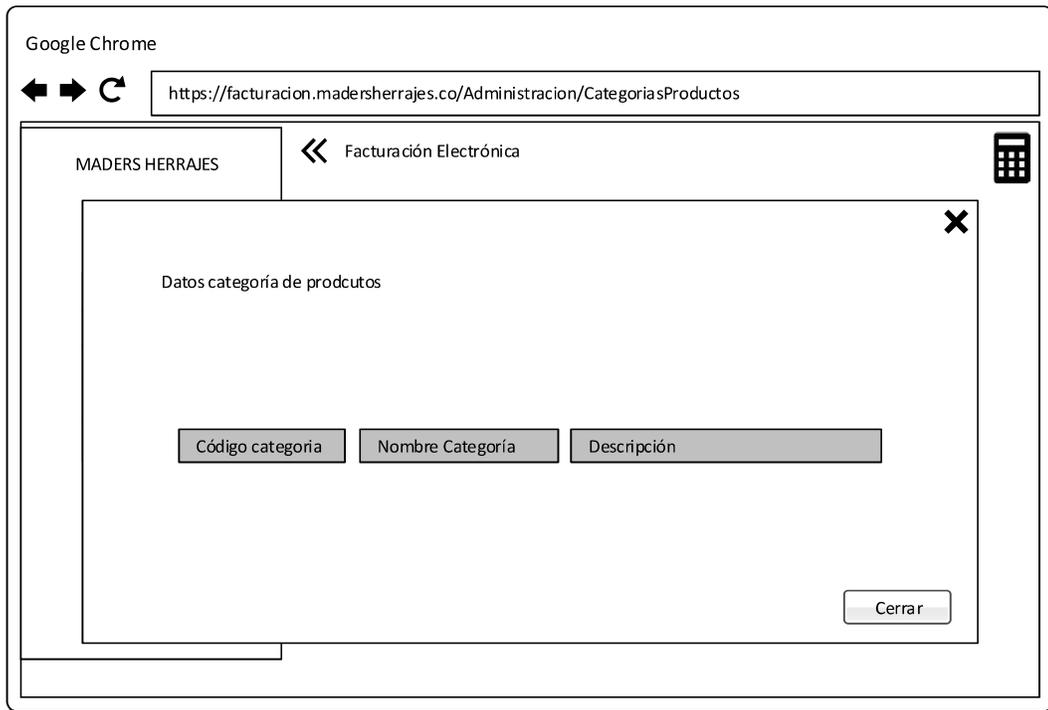


Figura 2.72 Vista para ver información de una categoría de productos existente del sistema.

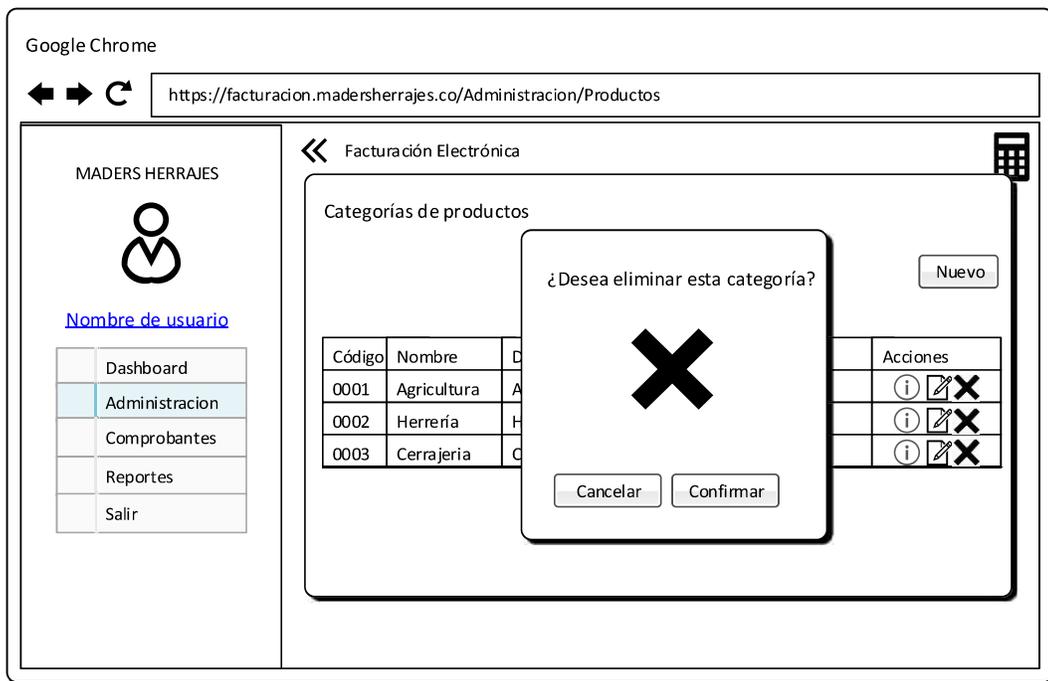


Figura 2.73 Vista para eliminar una categoría de productos del sistema.

2.1.9.2.13. Administración de productos

Esta vista permite administrar los productos del sistema y es accesible por los roles de administrador, administrador de sucursal y usuario.

La Figura 2.74 muestra la vista general de productos que visualiza el usuario.

Para productos ingresados se han establecido las acciones ver, editar y eliminar.

La Figura 2.75 presenta la vista para agregar un nuevo producto al sistema.

La vista para eliminar el producto se presenta en la Figura 2.78.

Las vistas para la visualización de los datos extendidos de productos y modificación se presentan en la Figura 2.77 y Figura 2.76 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.2.14. Administración de costos

Esta vista permite administrar los costos de cada sucursal del sistema y es accesible por los roles de administrador y administrador de sucursal, el resto de roles no pueden ver ni acceder a dicha vista.

La Figura 2.79 muestra la vista general de costos ingresados que visualiza el usuario.

Para costos ingresados se han establecido las acciones ver, editar.

La Figura 2.80 presenta la vista para el ingreso de un nuevo costo al sistema.

Las vistas para la visualización de los datos extendidos de costos y modificación se presentan en la Figura 2.82 y Figura 2.81 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.2.15. Administración de gastos

Esta vista permite administrar los gastos de cada sucursal del sistema y es accesible por los roles de administrador y administrador de sucursal, el resto de roles no pueden ver ni acceder a dicha vista.

La Figura 2.83 muestra la vista general de gastos ingresados que visualiza el usuario.

Para costos ingresados se han establecido las acciones ver y editar.

La Figura 2.84 presenta la vista para el ingreso de un nuevo gasto al sistema.

Las vistas para la visualización de los datos extendidos de gastos y modificación se presentan en la Figura 2.86 y Figura 2.85 respectivamente.

Para el ingreso se valida que los campos estén llenos correctamente.

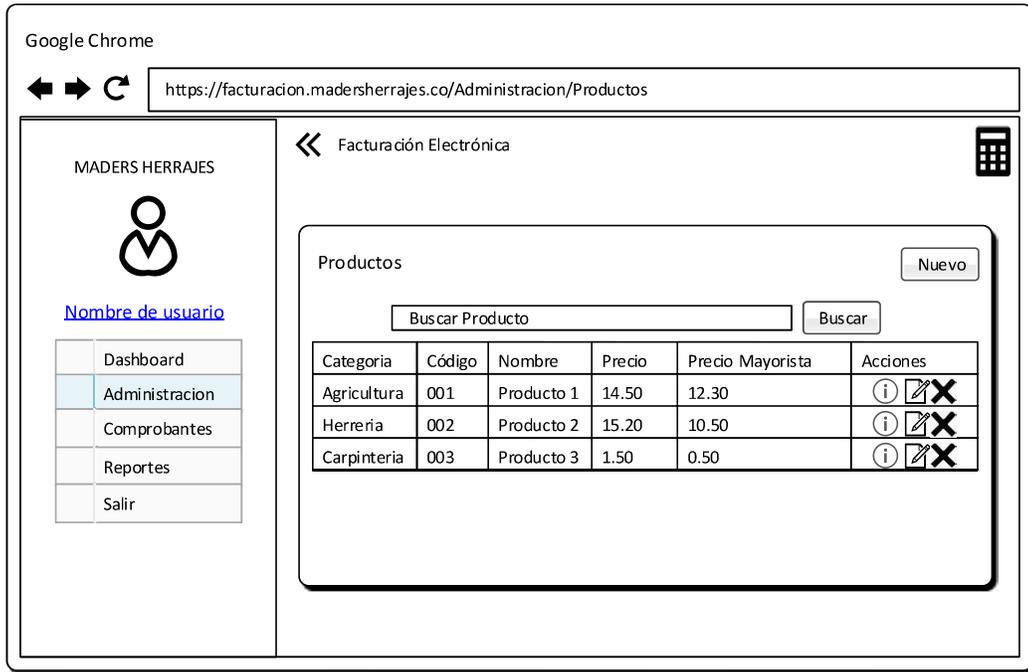


Figura 2.74 Vista general para administrar productos.

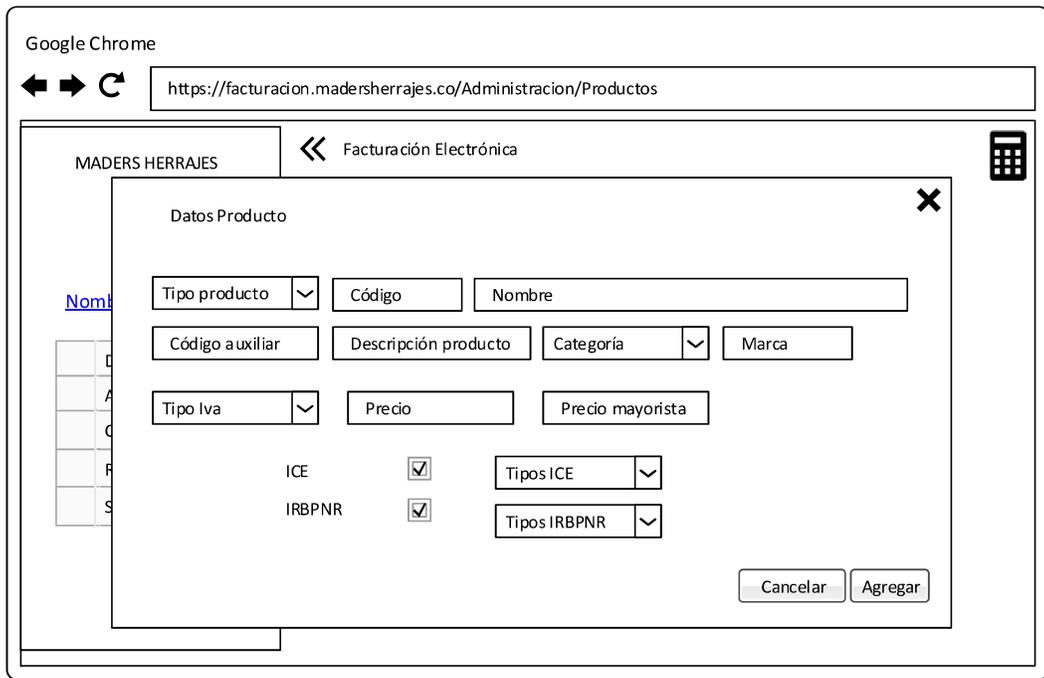


Figura 2.75 Vista para agregar un producto al sistema.

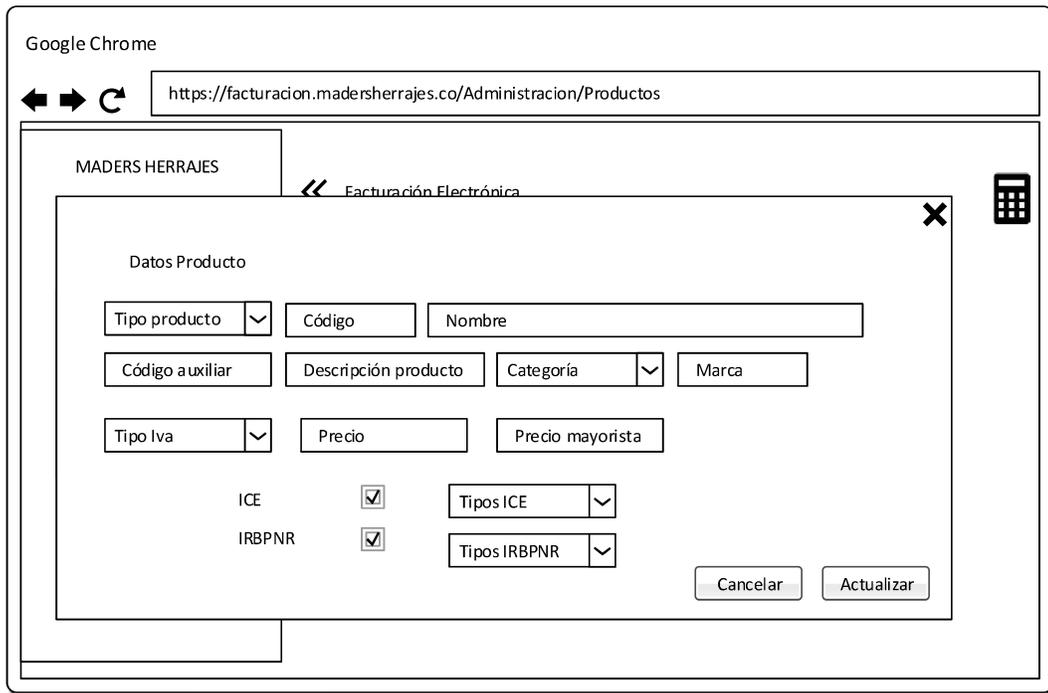


Figura 2.76 Vista para actualizar un producto existente del sistema.

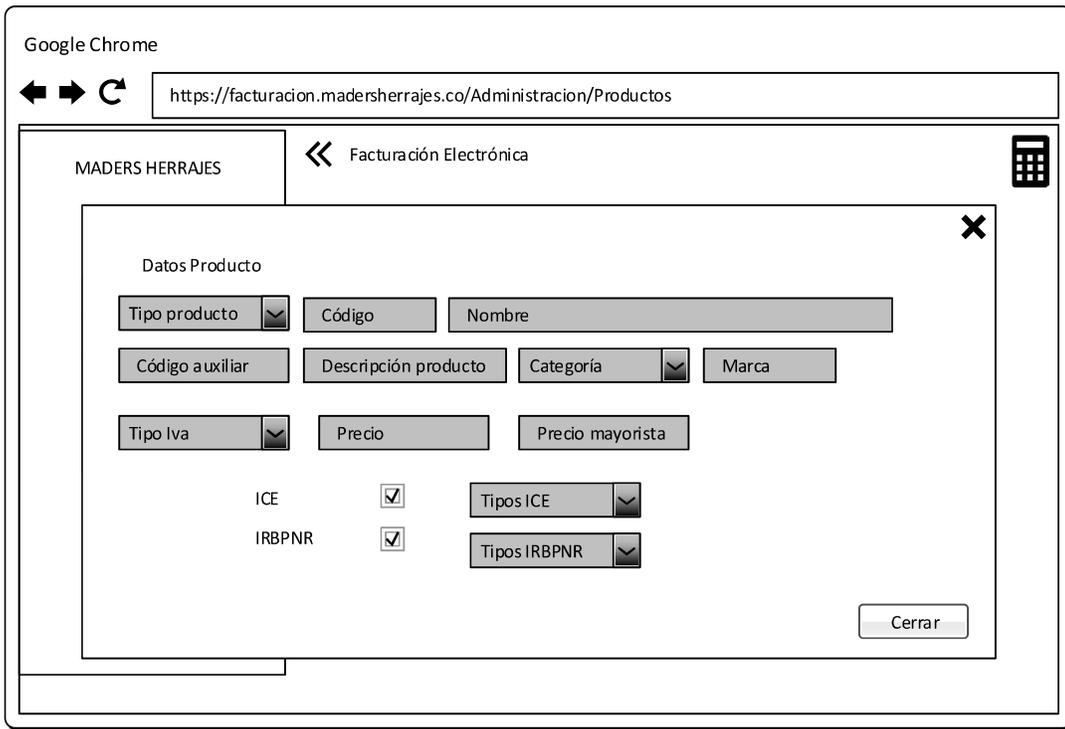


Figura 2.77 Vista para ver información de un producto existente del sistema.

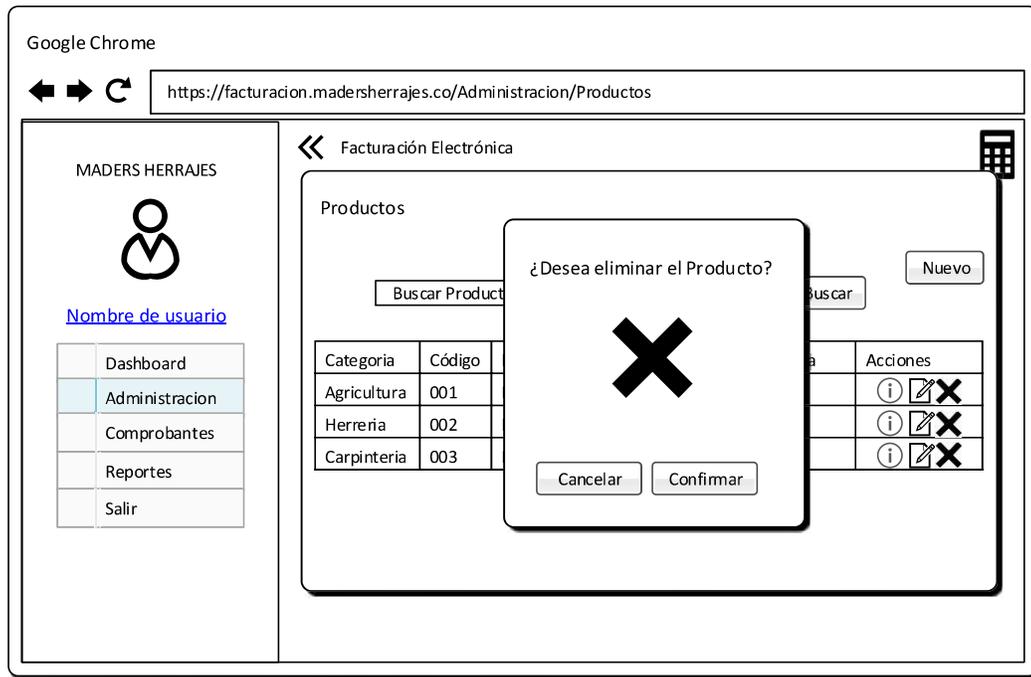


Figura 2.78 Vista para eliminar un producto del sistema.

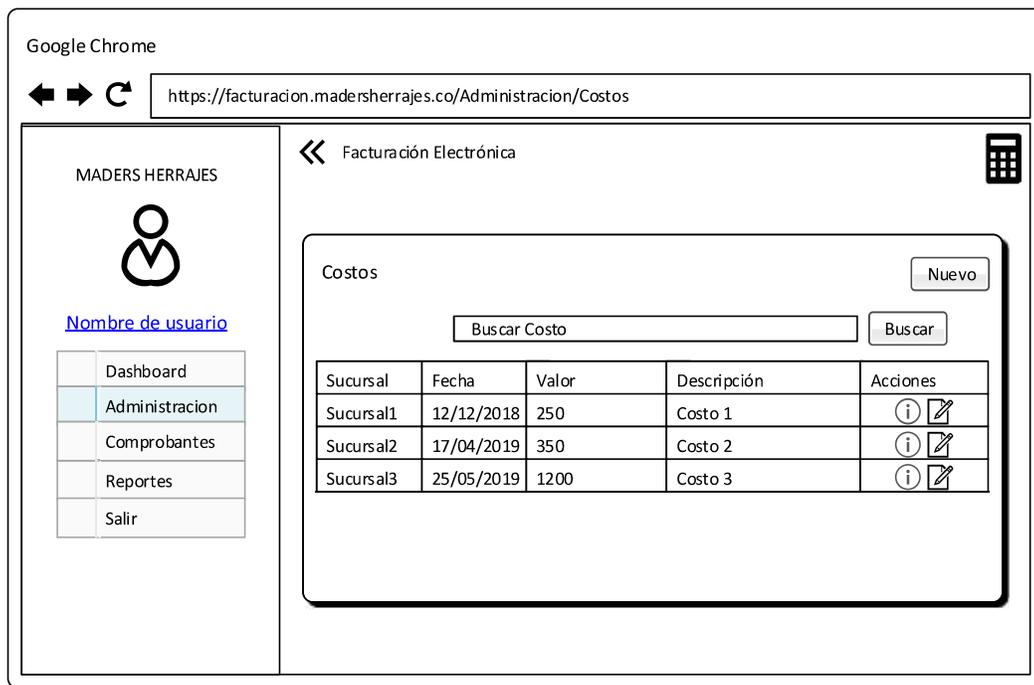


Figura 2.79 Vista general para administrar costos.

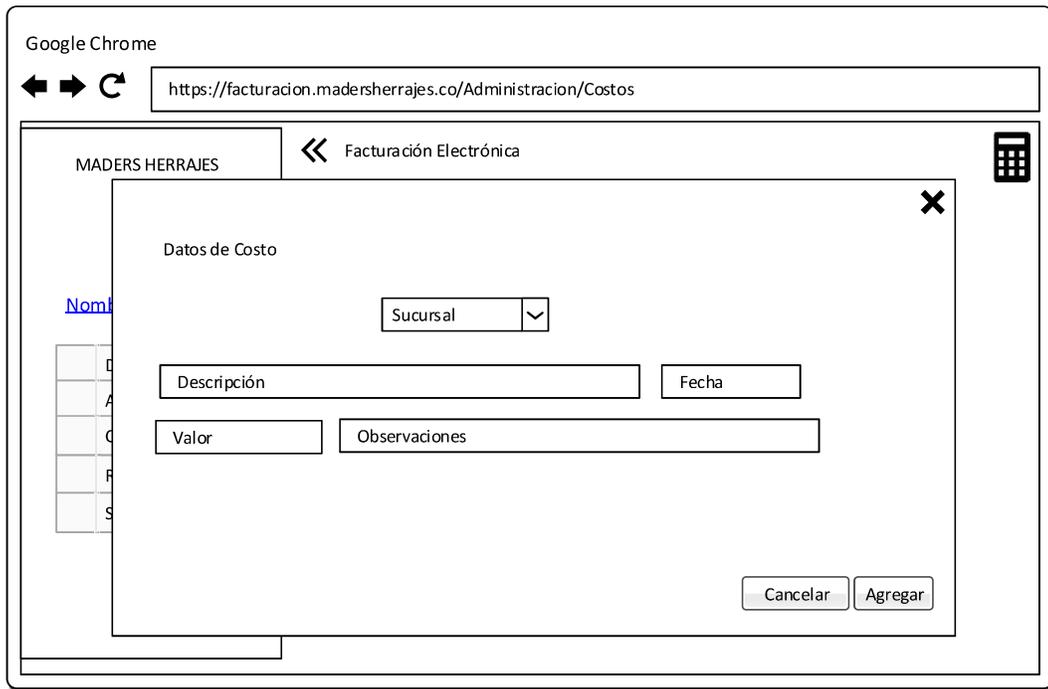


Figura 2.80 Vista para agregar un costo al sistema.

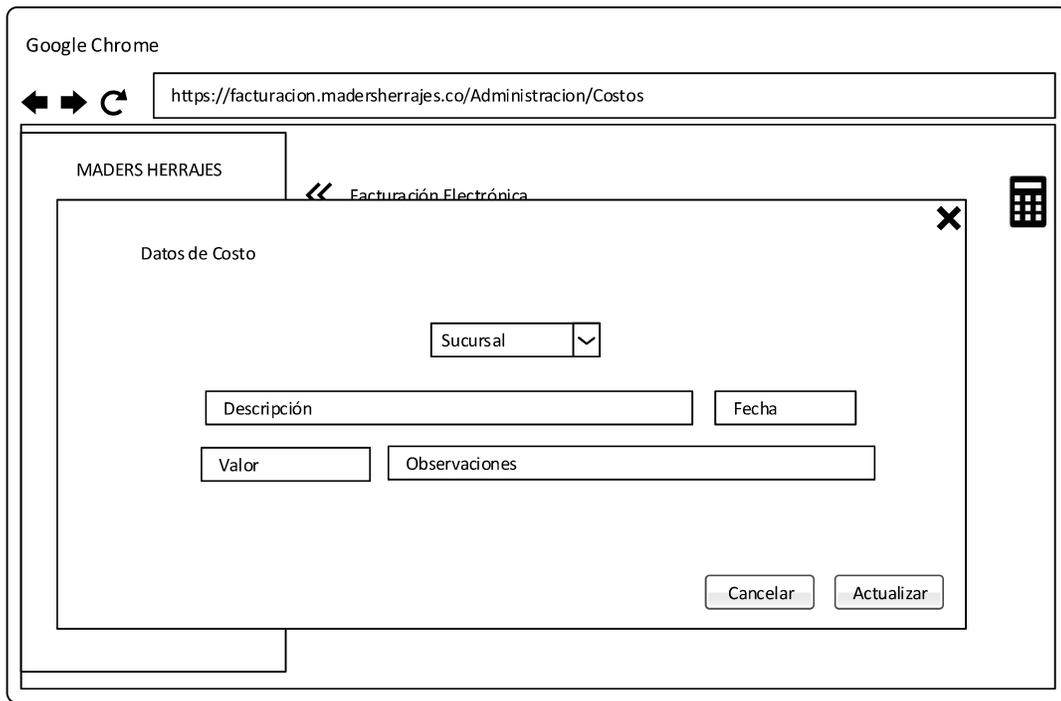


Figura 2.81 Vista para actualizar un costo existente del sistema.

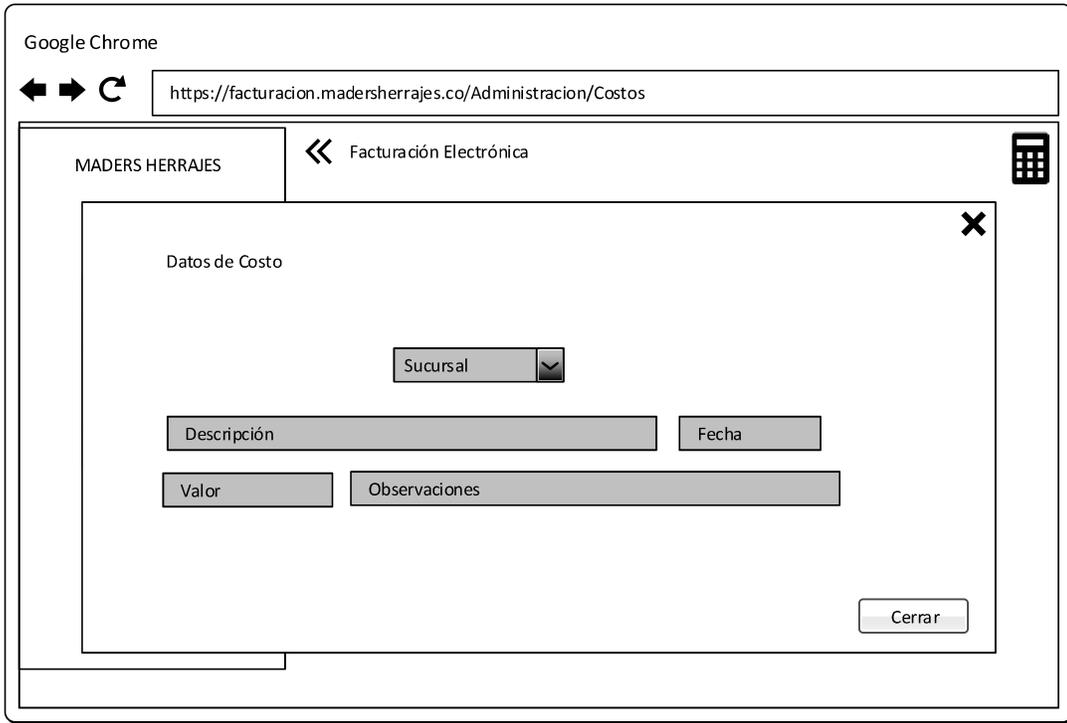


Figura 2.82 Vista para ver información de un costo existente del sistema.

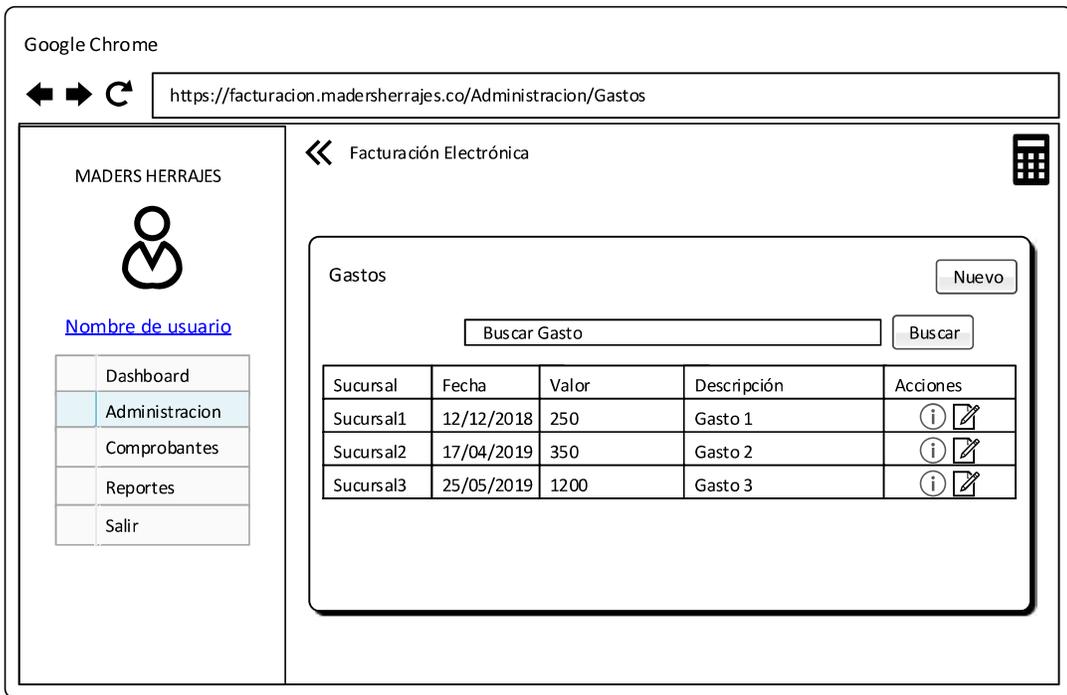


Figura 2.83 Vista general para administrar gastos.

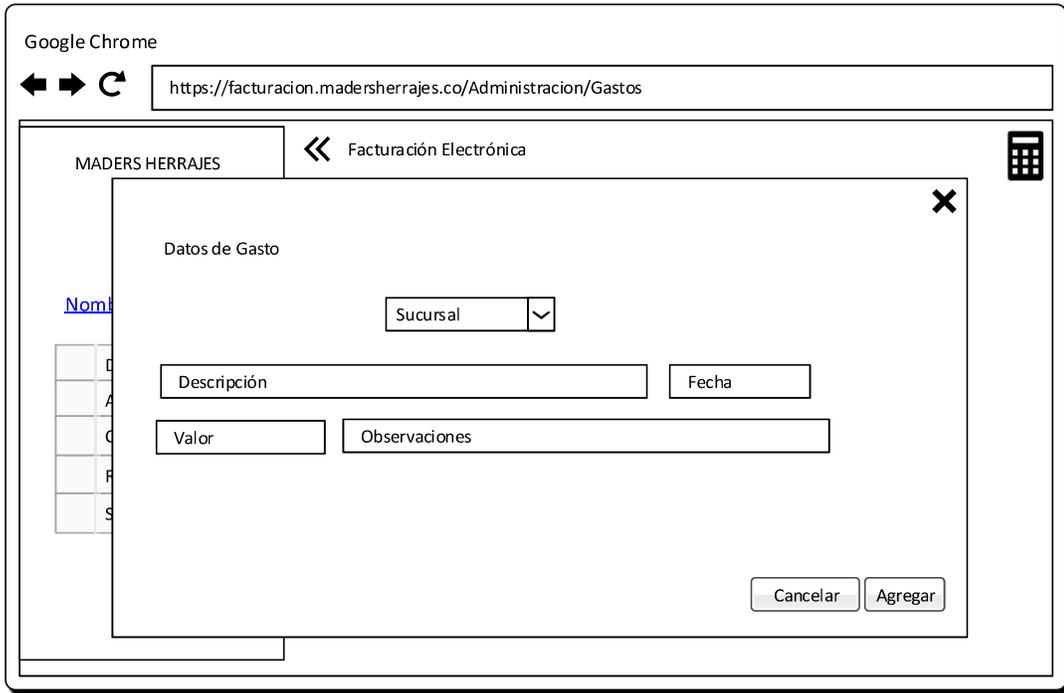


Figura 2.84 Vista para agregar un gasto al sistema.

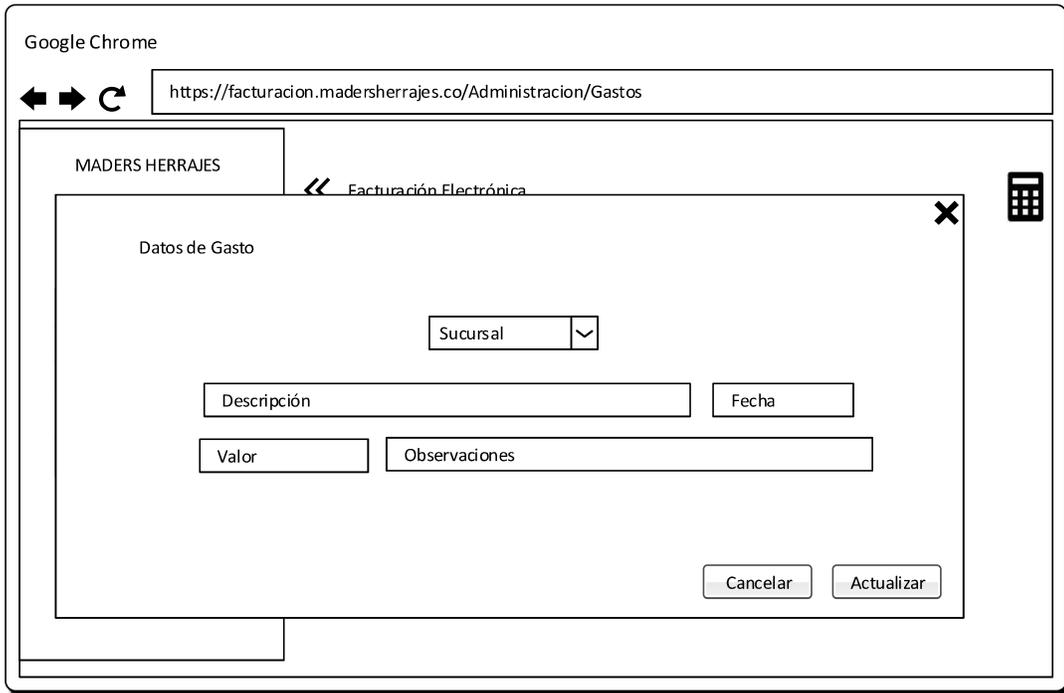


Figura 2.85 Vista para actualizar un gasto existente del sistema.

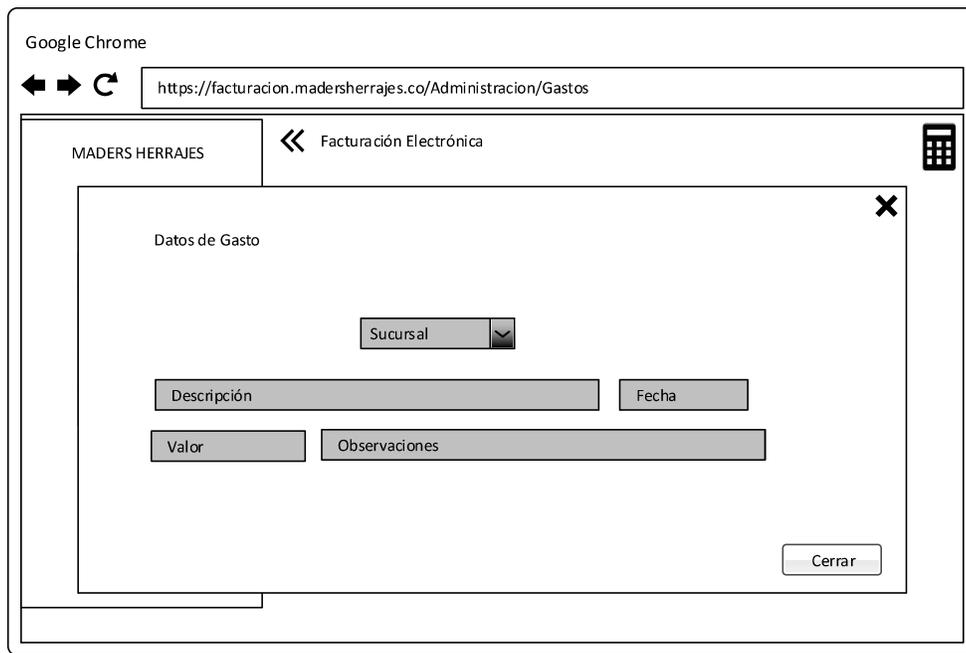


Figura 2.86 Vista para ver información de un gasto existente del sistema.

2.1.9.3. MÓDULO DE COMPROBANTES

Permite la emisión, autorización, visualización y envío de comprobantes electrónicos del sistema.

La Figura 2.87 presenta el diagrama de secuencia para el proceso de la generación de comprobantes electrónicos.

En la Figura 2.88 se presenta la vista general para la visualización todos los comprobantes electrónicos generados en el sistema.

2.1.9.3.1. Cotizaciones

Esta vista permite generar y visualizar cotizaciones y es accesible por los roles de administrador, administrador de sucursal y usuario.

La Figura 2.89 presenta la vista para la generación de cotizaciones.

La Figura 2.90 muestra la vista de cotizaciones ingresadas en un menú desplegable que visualiza el usuario.

La única acción disponible para cotizaciones generadas en el sistema es mostrar el comprobante.

Para el ingreso se valida que los campos estén llenos correctamente.

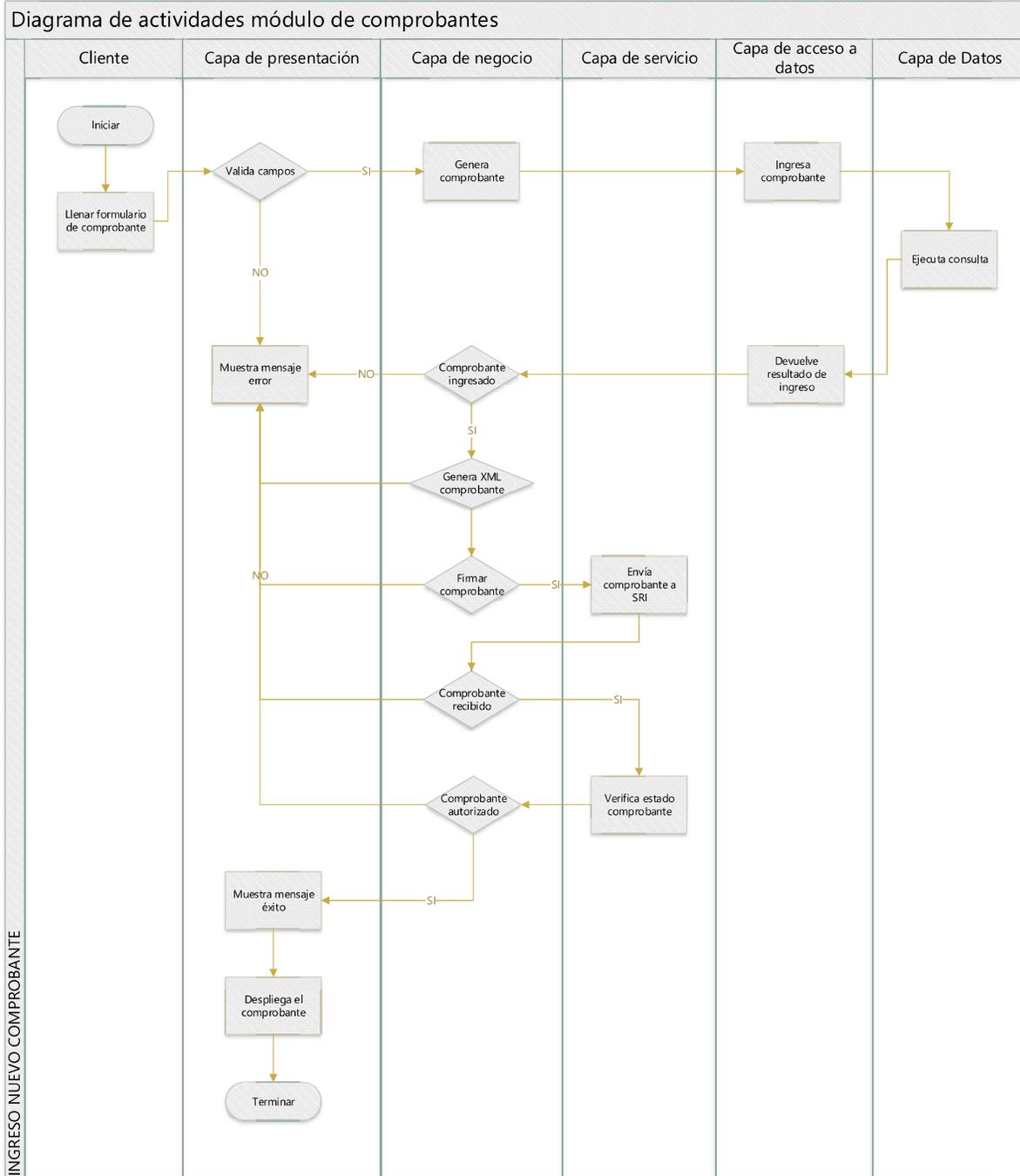


Figura 2.87 Diagrama de actividades para el ingreso de un nuevo comprobante electrónico.

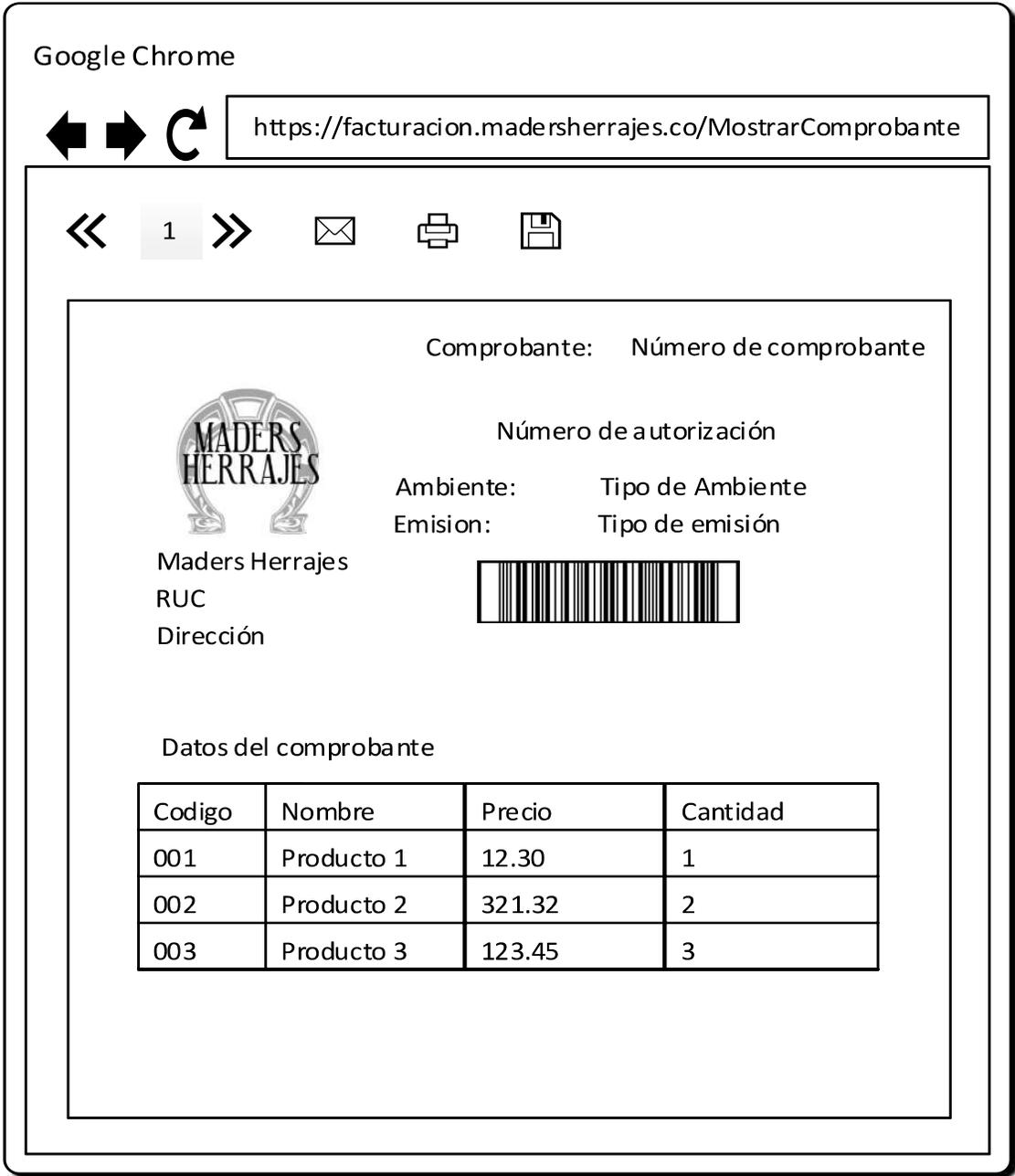


Figura 2.88 Vista para visualizar un comprobante electrónico.

Google Chrome
 https://facturacion.madersherrajes.co/Comprobantes/Cotizaciones

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

Dashboard
Comprobantes
Comprobantes
Reportes
Salir

Cotizaciones generadas

Generar nueva cotización

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matriz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del comprador

Consumidor Final Cliente

Identificación Razón social

Correo electrónico Dirección

Buscar cliente

Detalle de cotización

Cantidad	Código	Producto	Precio	Iva	Ice	Total	
1	001	Producto 1	17.50	2.50	0	19.50	✘
2	002	Producto 2	13.20	0	0	13.20	✘
3	003	Producto 3	45.60	12.37	2.30	60.27	✘

Buscar producto

Total Cotización

Subtotal	0.00
Subtotal 12	0.00
Total IVA	0.00
Total ICE	0.00
Total	0.00

Generar cotización Cancelar

Figura 2.89 Vista para generar una cotización.

Google Chrome
 https://facturacion.madersherrajes.co/Comprobantes/Cotizaciones

MADERS HERRAJES



[Nombre de usuario](#)

Dashboard
Comprobantes
Comprobantes
Reportes
Salir

Facturación Electrónica

Cotizaciones generadas

Sucursal	Nro.	Cliente	Subtotal	Iva	Total	Acciones
Sucursal 1	001001	Francisco G	12.50	2.30	14.80	
Sucursal 1	002002	Daniel H	32.41	4.60	37.01	
Sucursal 1	003003	Maria Jose F	122.00	36.0	158.00	
Sucursal 2	004004	Juan Viallaci	65410	0.00	65410	
Sucursal 2	005005	Martha Lucr	65.20	21.2	76.40	
Sucursal 2	006006	Victor Ramir	98.45	14.4	112.95	
Sucursal 3	007007	Marco Flores	45.32	32.2	77.42	
Sucursal 3	008008	Jose Fierro	981.23	123	1089.60	

[1](#) [2](#) [3](#)

Generar nueva cotización

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matriz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del comprador

Consumidor Final Cliente

Identificación Razón social

Detalle de cotización

Código	Producto	Precio	Iva	Ice	Total	
001	Producto 1	17.50	2.50	0	19.50	
002	Producto 2	13.20	0	0	13.20	
003	Producto 3	45.60	12.37	2.30	60.27	

Figura 2.90 Vista desplegada con lista de cotizaciones generadas.

2.1.9.3.2. Facturas

Esta vista permite generar y visualizar facturas y es accesible por los roles de administrador, administrador de sucursal y usuario.

La Figura 2.91 muestra la vista para generar una nueva factura en el sistema.

La Figura 2.92 muestra la vista de facturas ingresadas en un menú desplegable que visualiza el usuario.

Las acciones disponibles para facturas generadas en el sistema es mostrar el comprobante y verificar la autorización del comprobante a través de los servicios web del SRI.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.3.3. Retenciones

Esta vista permite generar y visualizar retenciones y es accesible por los roles de administrador, administrador de sucursal.

La Figura 2.93 presenta la vista para la generación de un nuevo comprobante de retención.

La Figura 2.94 muestra la vista de retenciones ingresadas en un menú desplegable que visualiza el usuario.

Las acciones disponibles para retenciones generadas en el sistema es mostrar el comprobante y verificar la autorización del comprobante a través de los servicios web del SRI.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.3.4. Notas de crédito

Esta vista permite generar y visualizar notas de crédito y es accesible por los roles de administrador y administrador de sucursal.

La Figura 2.95 presenta la vista para la generación de una nueva nota de crédito.

La Figura 2.96 muestra la vista de notas de crédito ingresadas en un menú desplegable que visualiza el usuario.

Las acciones disponibles para notas de crédito generadas en el sistema es mostrar el comprobante y verificar la autorización del comprobante a través de los servicios web del SRI.

Para el ingreso se valida que los campos estén llenos correctamente.

Google Chrome
 https://facturacion.madersherrajes.co/Comprobantes/Facturas

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

Dashboard
Comprobantes
Comprobantes
Reportes
Salir

Facturas generadas

Generar nueva factura

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matriz

Datos del comprobante

Número de comprobante Clave de acceso

Fecha de emisión Guía de remisión

Datos del comprador

Consumidor Final Cliente

Identificación Razón social

Correo electrónico Dirección

Buscar cliente

Detalle de la factura

Cantidad	Código	Producto	Precio	Iva	Ice	Total	
1	001	Producto 1	17.50	2.50	0	19.50	X
2	002	Producto 2	13.20	0	0	13.20	X
3	003	Producto 3	45.60	12.37	2.30	60.27	X

Buscar producto

Forma de pago

Subtotal	0.00	Forma de Pago	Total	
Subtotal 12	0.00	Efectivo	123.32	X
Total IVA	0.00			
Total ICE	0.00			
Total	0.00			

Buscar forma pago

Generar factura Cancelar

Figura 2.91 Vista para generar una factura.

Google Chrome
 https://facturacion.madersherrajes.co/Comprobantes/Facturas

MADERS HERRAJES

Nombre de usuario

- Dashboard
- Comprobantes
- Comprobantes
- Reportes
- Salir

Facturación Electrónica

Facturas generadas

Buscar factura

Sucursal	Nro.	Cliente	Estado	Subtot..	Total	Acciones
Sucursal 1	001001	Francisco G	Autorizado	2.30	14.80	
Matríz	002002	Daniel H	Autorizado	4.60	37.01	
Sucursal 1	003003	Maria Jose F	Autorizado	36.0	158.00	
Sucursal 2	004004	Juan Viallaci	Autorizado	0.00	65410	
Sucursal 2	005005	Martha Lucr	Autorizado	21.2	76.40	
Matríz	006006	Victor Ramir	Autorizado	14.4	112.95	
Sucursal 3	007007	Marco Flores	Autorizado	32.2	77.42	
Sucursal 3	008008	Jose Fierro	No Auto..	123	1089.60	

1 2 3

Generar nueva factura

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matríz

Datos del comprobante

Número de comprobante Clave de acceso

Fecha de emisión Guía de remisión

Datos del comprador

Consumidor Final Cliente

Identificación Razón social

Detalle de factura

Código	Producto	Precio	Iva	Ice	Total	
001	Producto 1	17.50	2.50	0	19.50	X
002	Producto 2	13.20	0	0	13.20	X
003	Producto 3	45.60	12.37	2.30	60.27	X

Figura 2.92 Vista desplegada con lista de facturas generadas.

Google Chrome

https://facturacion.madersherrajes.co/Comprobantes/Retenciones

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

- Dashboard
- Comprobantes
- Comprobantes
- Reportes
- Salir

Comprobantes de retención generados

Generar nueva retención

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matriz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del sujeto retenido

Mes Año

Identificación Razón social

Correo electrónico Dirección

Buscar cliente

Detalle del comprobante de retención

Impuesto	Emisión	Tipo	%	Base	Valor	
0001	24/04/2019	IVA	8%	123	19.50	X
0002	25/05/2018	RENTA	10%	321	13.12	X

Agregar Impuesto

Generar comprobante de retención Cancelar

Figura 2.93 Vista para generar un comprobante de retención.

Google Chrome

https://facturacion.madersherrajes.co/Comprobantes/Retenciones

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

- Dashboard
- Comprobantes
- Comprobantes
- Reportes
- Salir

Comprobantes de retención generados

Buscar retención

Sucursal	Nro.	Cliente	Estado	Periodo	Total Ret	Acciones
Sucursal 1	001001	Francisco G	Autorizado	01/19	14.80	
Matríz	002002	Daniel H	Autorizado	02/19	37.01	
Sucursal 1	003003	María Jose F	Autorizado	03/19	158.00	
Sucursal 2	004004	Juan Viallaci	Autorizado	04/19	65410	
Sucursal 2	005005	Martha Lucr	Autorizado	05/19	76.40	
Matríz	006006	Victor Ramir	Autorizado	06/19	112.95	
Sucursal 3	007007	Marco Flores	Autorizado	07/19	77.42	
Sucursal 3	008008	Jose Fierro	No Auto..	08/19	1089.60	

1 2 3

Generar nueva retención

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matríz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del sujeto retenido

Mes Año

Identificación Razón social

Correo electrónico Dirección

Figura 2.94 Vista desplegada con lista de comprobantes de retención generados.

Google Chrome
 https://facturacion.madersherrajes.co/Comprobantes/NotasDeCredito

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

Dashboard
Comprobantes
Comprobantes
Reportes
Salir

Notas de crédito generadas

Generar nueva nota de crédito

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matriz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del comprador

Identificación Razón social

Correo electrónico Dirección

Buscar cliente

Detalle de la nota de crédito

Cantidad	Código	Producto	Precio	Iva	Ice	Total	
1	001	Producto 1	17.50	2.50	0	19.50	X
2	002	Producto 2	13.20	0	0	13.20	X
3	003	Producto 3	45.60	12.37	2.30	60.27	X

Forma de pago

Subtotal 0.00 Subtotal 12 0.00 Total IVA 0.00 Total ICE 0.00 Total 0.00

Buscar producto

Número comprobante Fecha Emisión

Motivo nota de crédito

Buscar factura

Generar nota de crédito Cancelar

Figura 2.95 Vista para generar una nota de crédito.

Google Chrome

← → ↻ <https://facturacion.madersherrajes.co/Comprobantes/NotasDeCredito>

MADERS HERRAJES



[Nombre de usuario](#)

Dashboard
Comprobantes
Comprobantes
Reportes
Salir

Facturación Electrónica

Notas de crédito generadas

Buscar nota de crédito

Sucursal	Nro.	Cliente	Estado	Subtot..	Total	Acciones
Sucursal 1	001001	Francisco G	Autorizado	2.30	14.80	<input type="checkbox"/> <input type="checkbox"/>
Matríz	002002	Daniel H	Autorizado	4.60	37.01	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 1	003003	Maria Jose F	Autorizado	36.0	158.00	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 2	004004	Juan Viallaci	Autorizado	0.00	65410	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 2	005005	Martha Lucr	Autorizado	21.2	76.40	<input type="checkbox"/> <input type="checkbox"/>
Matríz	006006	Victor Ramir	Autorizado	14.4	112.95	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 3	007007	Marco Flores	Autorizado	32.2	77.42	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 3	008008	Jose Fierro	No Auto..	123	1089.60	<input type="checkbox"/> <input type="checkbox"/>

[1](#) [2](#) [3](#)

Generar nueva nota de crédito

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matríz

Datos del comprobante

Número de comprobante Clave de acceso

Fecha de emisión

Datos del comprador

Identificación Razón social

Correo electrónico Dirección

Detalle de nota de crédito

Código	Producto	Precio	Iva	Ice	Total	
001	Producto 1	17.50	2.50	0	19.50	✘
002	Producto 2	13.20	0	0	13.20	✘
003	Producto 3	45.60	12.37	2.30	60.27	✘

Figura 2.96 Vista desplegada con lista de notas de crédito generadas.

2.1.9.3.5. Notas de débito

Esta vista permite generar y visualizar notas de débito y es accesible por los roles de administrador y administrador de sucursal.

La Figura 2.97 presenta la vista para la generación de una nota de débito

La Figura 2.98 muestra la vista general de notas de débito ingresadas en un menú desplegable que visualiza el usuario.

Las acciones disponibles para notas de débito generadas en el sistema es mostrar el comprobante y verificar la autorización del comprobante a través de los servicios web del SRI.

Para el ingreso se valida que los campos estén llenos correctamente.

2.1.9.3.6. Guías de remisión

Esta vista permite generar y visualizar guías de remisión y es accesible por los roles de administrador, administrador de sucursal y usuario.

La Figura 2.99 presenta la vista para la generación de una guía de remisión.

La Figura 2.100 muestra la vista de guías de remisión ingresadas en un menú desplegable que visualiza el usuario.

Las acciones disponibles para guías de remisión generadas en el sistema es mostrar el comprobante y verificar la autorización del comprobante a través de los servicios web del SRI.

Para el ingreso se valida que los campos estén llenos correctamente.

Google Chrome
 https://facturacion.madersherrajes.co/Comprobantes/NotasDeDebito

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

Dashboard
Comprobantes
Comprobantes
Reportes
Salir

Notas de débito generadas

Generar nueva nota de débito

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matriz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del cliente

Identificación Razón social

Correo electrónico Dirección

Detalle de la nota de débito

Buscar cliente

Número comprobante Fecha Emisión Ice:

Tipos ICE Valor \$

ID	Motivo	Valor	
1	No agregado a factura	19.50	X

Buscar factura Agregar detalle

Forma de pago

Tipo de IVA

Subtotal	0.00	Forma de Pago	Total	
Subtotal 12	0.00	Efectivo	123.32	X
Total IVA	0.00			
Total ICE	0.00			
Total	0.00			

Buscar forma pago

Generar nota de débito Cancelar

Figura 2.97 Vista para generar una nota de débito.

Google Chrome

https://facturacion.madersherrajes.co/Comprobantes/NotasDeDebito

MADERS HERRAJES

Facturación Electrónica

Notas de débito generadas

Buscar nota de débito

Sucursal	Nro.	Cliente	Estado	Subtot..	Total	Acciones
Sucursal 1	001001	Francisco G	Autorizado	2.30	14.80	
Matríz	002002	Daniel H	Autorizado	4.60	37.01	
Sucursal 1	003003	Maria Jose F	Autorizado	36.0	158.00	
Sucursal 2	004004	Juan Viallaci	Autorizado	0.00	65410	
Sucursal 2	005005	Martha Lucr	Autorizado	21.2	76.40	
Matríz	006006	Victor Ramir	Autorizado	14.4	112.95	
Sucursal 3	007007	Marco Flores	Autorizado	32.2	77.42	
Sucursal 3	008008	Jose Fierro	No Auto..	123	1089.60	

1 2 3

Generar nueva nota de débito

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matríz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del cliente

Identificación Razón social

Correo electrónico Dirección

Buscar cliente

Detalle de la nota de débito

Número comprobante Fecha Emisión Ice:

Tipos ICE Valor \$

Figura 2.98 Vista desplegada con lista de notas de débito generadas.

Google Chrome

https://facturacion.madersherrajes.co/Comprobantes/GuiasRemision

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

Dashboard
Comprobantes
Comprobantes
Reportes
Salir

Guías de remisión generadas

Generar nueva guía de remisión

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matriz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del transportista y traslado

Identificación Razón social Placa

Correo electrónico Dirección partida

Fecha inicio transporte Fecha fin transporte

Buscar transportista

Detalle de destinatarios

Destinatari	Factura	Destino	F. Emisión	Motivo	Cant.	
Francisco G	001001	Condado	19/04/19	Trasla	10	✘
Giovanny A	002002	Quicentro N	21/05/19	Movimi	20	✘

Agregar destinatario

Generar guía de remisión Cancelar

Figura 2.99 Vista para generar una guía de remisión.

Google Chrome

https://facturacion.madersherrajes.co/Comprobantes/GuiasRemision

Facturación Electrónica

MADERS HERRAJES

Nombre de usuario

- Dashboard
- Comprobantes
- Comprobantes
- Reportes
- Salir

Guías de remisión generadas

Buscar guía de remisión

Sucursal	Nro.	Transportista	Estado	Placa	Destino	Acciones
Sucursal 1	001001	Francisco G	Autorizad	QWE321	Condado	<input type="checkbox"/> <input type="checkbox"/>
Matríz	002002	Daniel H	Autorizad	ASD123	Quicentr	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 1	003003	Maria Jose F	Autorizad	ZXC321	Sur	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 2	004004	Juan Viallaci	Autorizad	EWQ321	Norte	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 2	005005	Martha Lucre	Autorizad	TYU123	Cumbay	<input type="checkbox"/> <input type="checkbox"/>
Matríz	006006	Victor Ramir	Autorizad	VBN123	Floresta	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 3	007007	Marco Flores	Autorizad	TYU321	Mariana	<input type="checkbox"/> <input type="checkbox"/>
Sucursal 3	008008	Jose Fierro	No Auto..	NBV321	Calderas	<input type="checkbox"/> <input type="checkbox"/>

1 2 3

Generar nueva guía de remisión

Datos de emisor

Ruc Razón Social Nombre comercial

Dirección establecimiento Dirección matríz

Datos del comprobante

Fecha de emisión

Número de comprobante Clave de acceso

Datos del transportista y traslado

Identificación Razón social Placa

Correo electrónico Dirección partida

Fecha inicio transporte Fecha fin transporte

Figura 2.100 Vista desplegada con lista de guías de remisión generadas.

2.1.9.4. MÓDULO DE REPORTE

Permite la generación de reportes en base a la información generada por los comprobantes electrónicos generados y los costos y gastos registrados en el sistema.

La Figura 2.101 presenta el diagrama de actividades para el módulo de reportes y la interacción entre las diferentes capas del sistema.

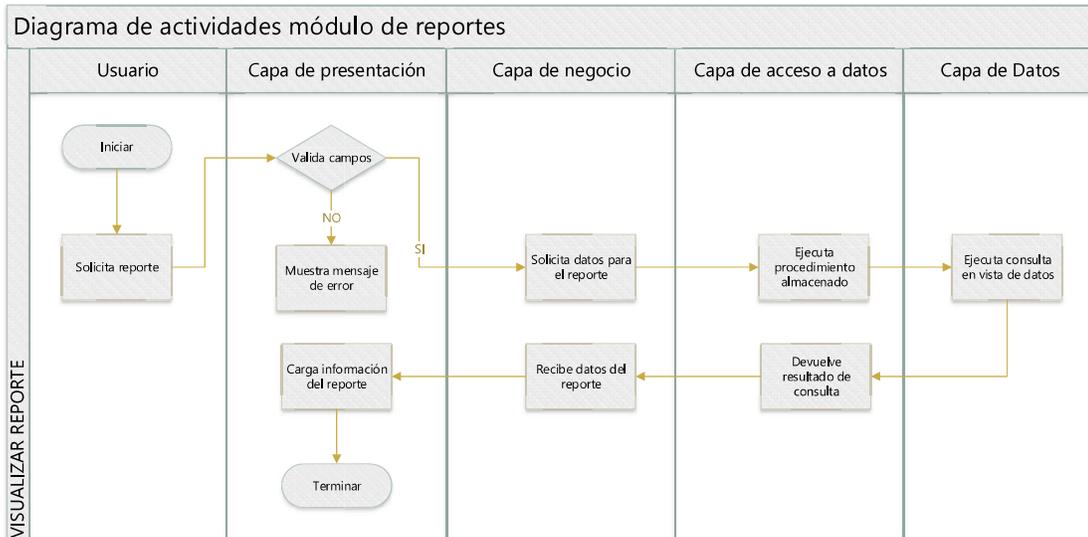


Figura 2.101 Diagrama de actividades del módulo de reportes.

2.1.9.4.1. Gerencial de ventas

La Figura 2.102 muestra la vista de gerencial de ventas presentada al usuario.

Esta vista permite generar y visualizar reportes consolidados de ventas y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte gerencial de ventas son obtener reportes concernientes a las ventas generadas en el año seleccionado e imprimir los resultados.

En base al rol del usuario se puede presentar información consolidada de la sucursal o información global del negocio.

2.1.9.4.2. Gerencial de compras

La Figura 2.103 muestra la vista de gerencial de compras presentada al usuario.

Esta vista permite generar y visualizar reportes consolidados de compras y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte gerencial de compras son obtener reportes concernientes a las compras generadas en el año seleccionado e imprimir los resultados.

En base al rol del usuario se puede presentar información consolidada de la sucursal o información global del negocio.

2.1.9.4.3.Compras vs ventas

La Figura 2.104 muestra la vista de gerencial de compras vs ventas presentada al usuario.

Esta vista permite generar y visualizar reportes consolidados de compras en comparadas con ventas y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte gerencial de compras son obtener reportes concernientes a las compras comparadas con las ventas generadas en el año seleccionado e imprimir los resultados.

En base al rol del usuario se puede presentar información consolidada de la sucursal o información global del negocio.

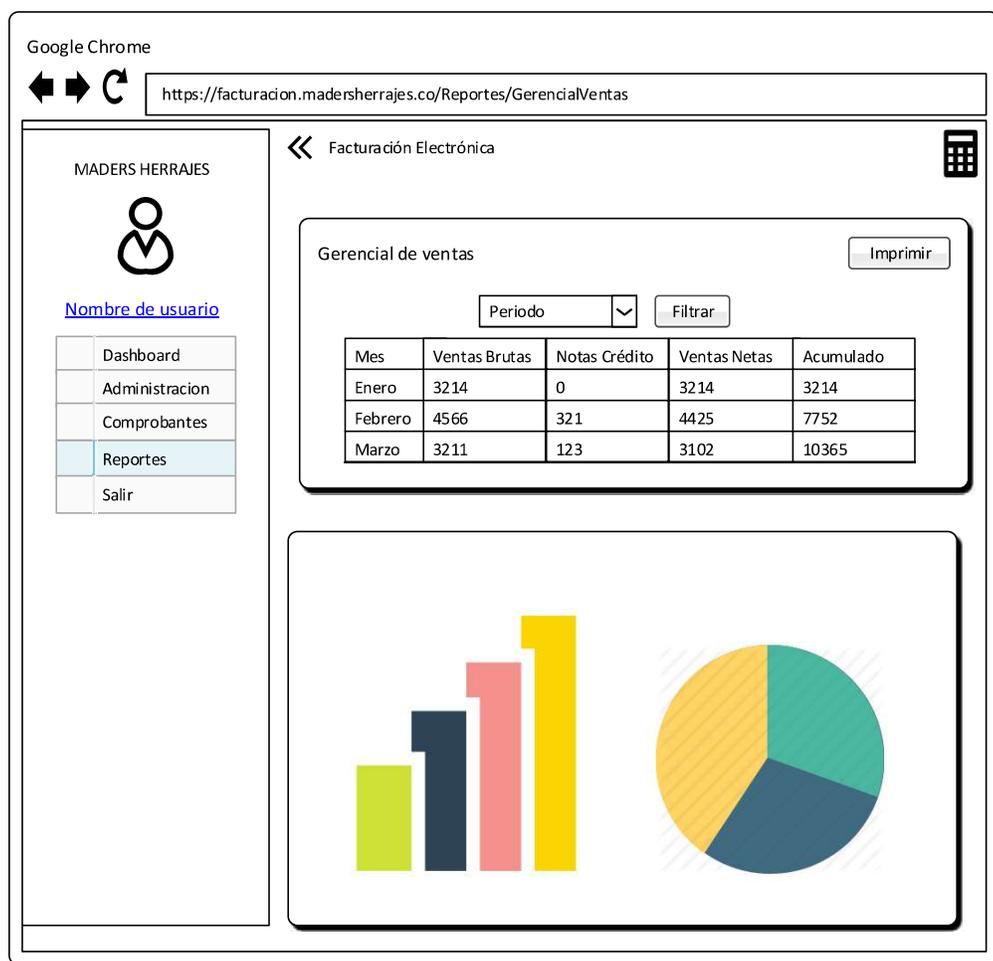


Figura 2.102 Vista de reporte gerencial de ventas.

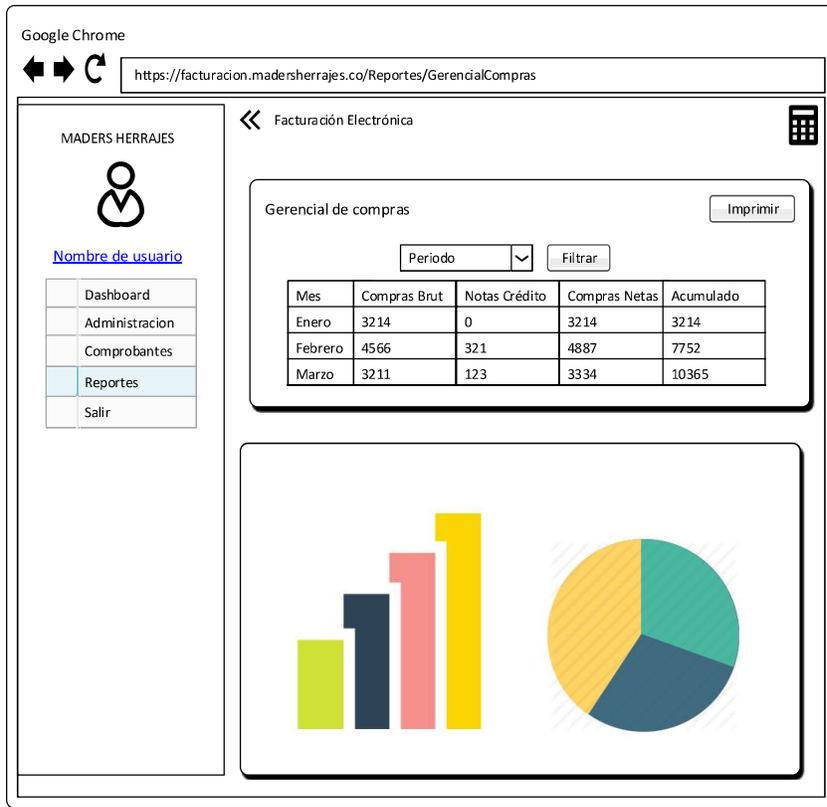


Figura 2.103 Vista de reporte gerencial de compras.

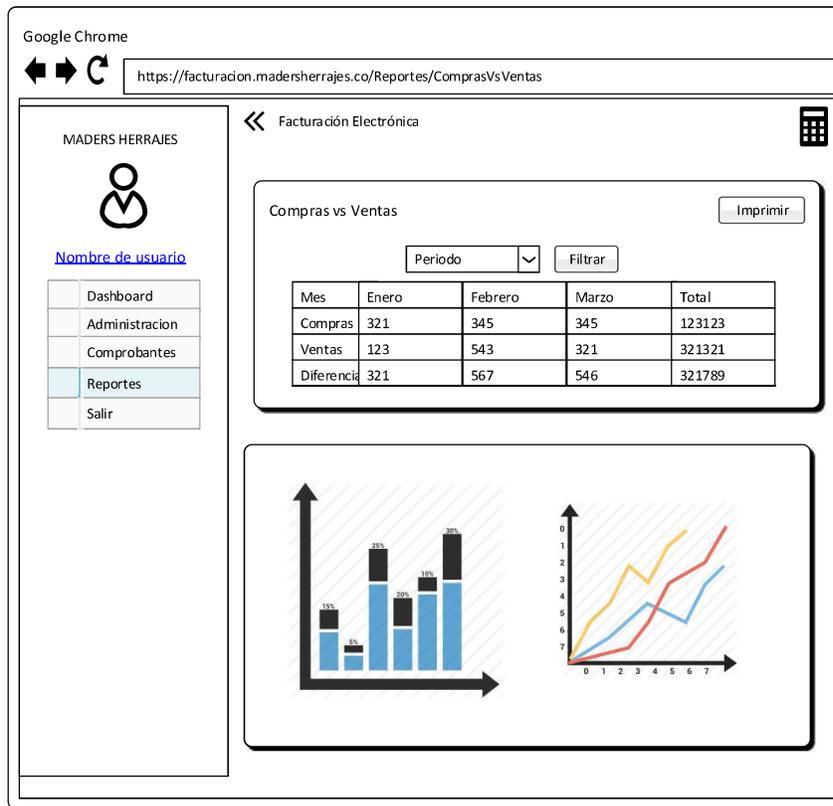


Figura 2.104 Vista de reporte de compras versus ventas.

2.1.9.4.4. Productos más vendidos

La Figura 2.105 muestra la vista de productos más vendidos presentada al usuario.

Esta vista permite generar y visualizar reportes detallados de productos más vendidos y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte de productos más vendidos son obtener reportes concernientes a los productos vendidos en un rango de fechas seleccionado e imprimir los resultados.

En base al rol del usuario se puede presentar información consolidada de la sucursal o información global del negocio.

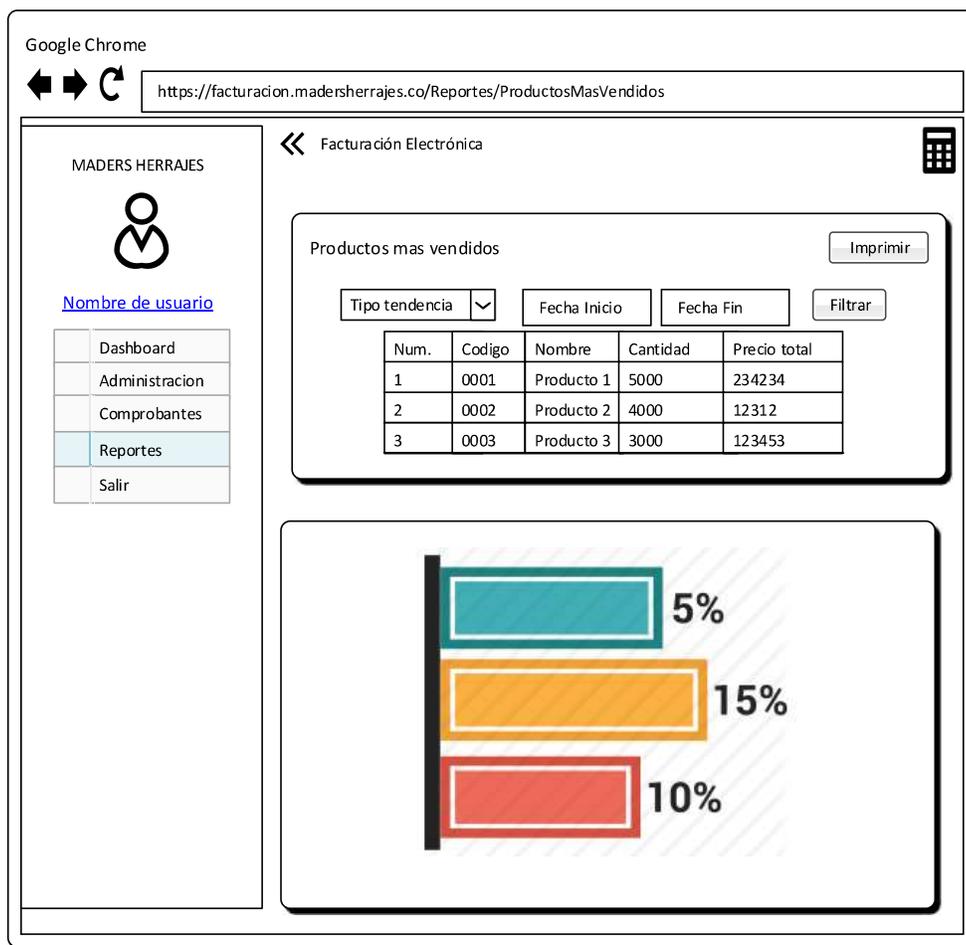


Figura 2.105 Vista de reporte de productos más vendidos.

2.1.9.4.5. Tendencia de ventas

La Figura 2.106 muestra la vista de tendencia de ventas presentada al usuario.

Esta vista permite generar y visualizar reportes detallados de tendencia de ventas y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte gerencial de compras son obtener reportes concernientes a la tendencia de ventas por mes en un rango de fechas seleccionado e imprimir los resultados.

En base al rol del usuario se puede presentar información consolidada de la sucursal o información global del negocio.

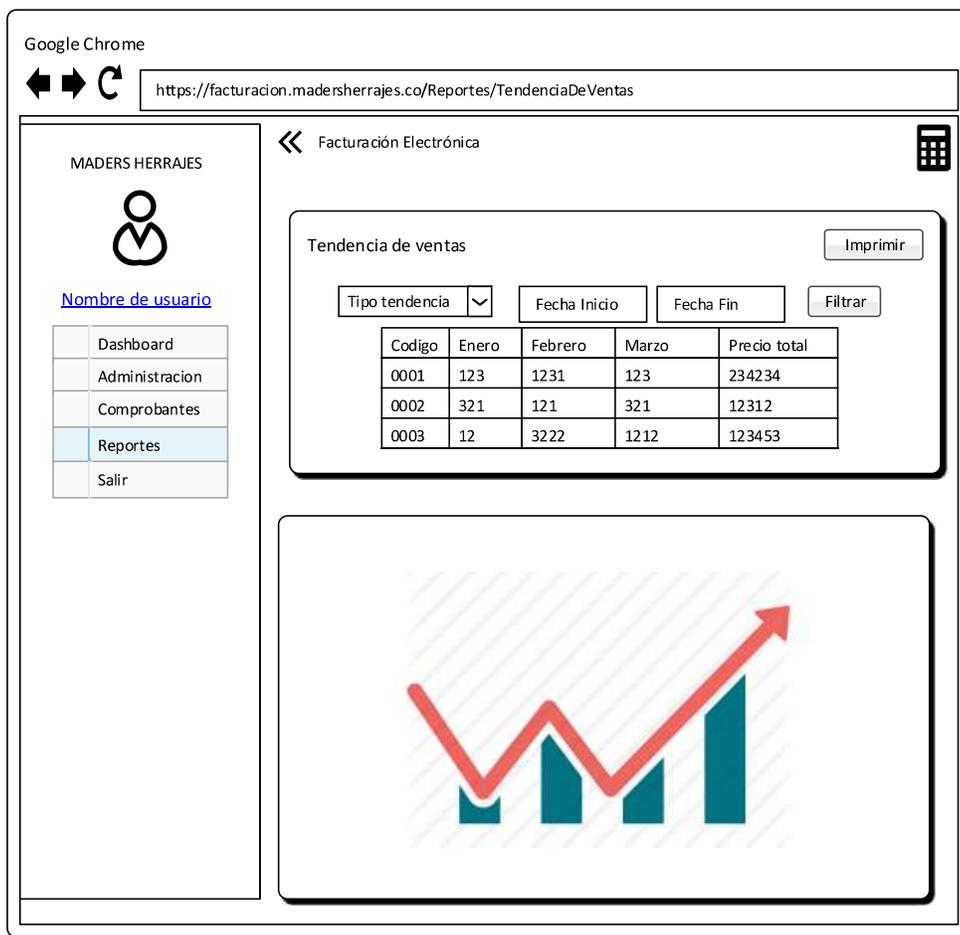


Figura 2.106 Vista de reporte gerencial de tendencia de ventas.

2.1.9.4.6. Análisis de gastos

La Figura 2.107 muestra la vista de análisis de gastos presentada al usuario.

Esta vista permite generar y visualizar reportes detallados de gastos y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte de gastos son obtener reportes concernientes a los gastos generados en el año seleccionado e imprimir los resultados.

En base al rol del usuario se puede presentar información consolidada de la sucursal o información global del negocio.

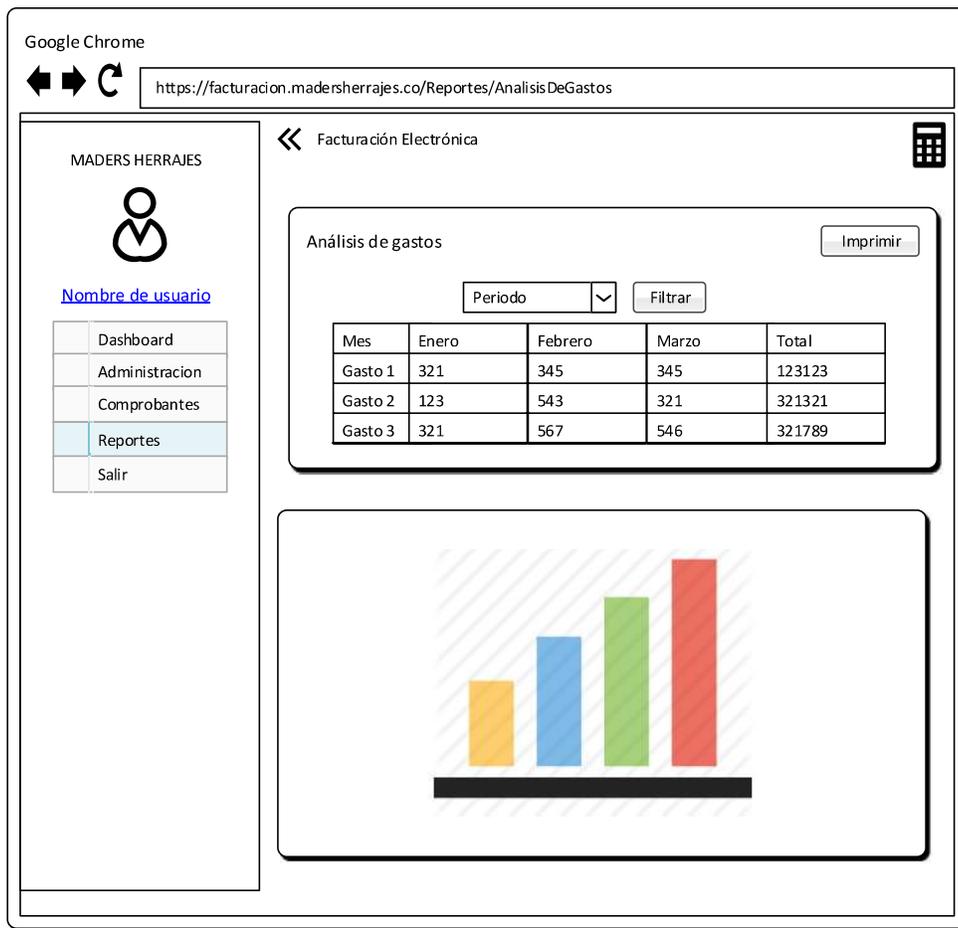


Figura 2.107 Vista de reporte de análisis de gastos.

2.1.9.4.7. Análisis de costos

La Figura 2.108 muestra la vista de análisis de costos presentada al usuario.

Esta vista permite generar y visualizar reportes detallados de costos y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte costos son obtener reportes concernientes a los costos generados en el año seleccionado e imprimir los resultados.

En base al rol del usuario se puede presentar información consolidada de la sucursal o información global del negocio.

2.1.9.4.8. Resumen de movimientos (KARDEX)

La Figura 2.109 muestra la vista de resumen de movimientos presentada al usuario.

Esta vista permite generar y visualizar reportes de movimientos de comprobantes electrónicos y es accesible por los roles de administrador y administrador de sucursal.

Las acciones disponibles para el reporte de movimientos son obtener todos los movimientos generados para un comprobante específico e imprimir los resultados.

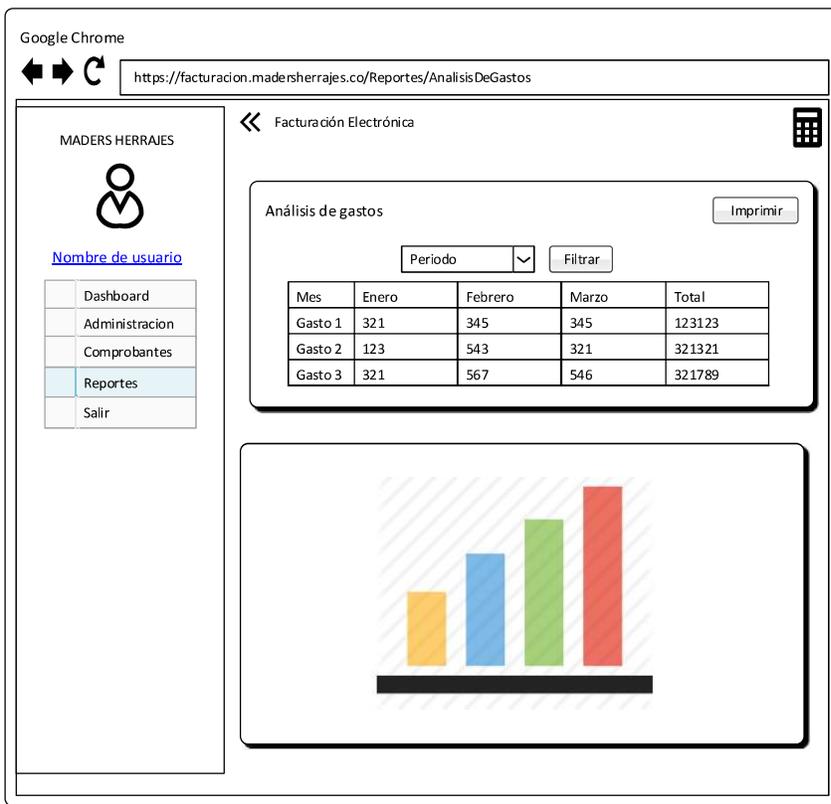


Figura 2.108 Vista de reporte de análisis de costos.

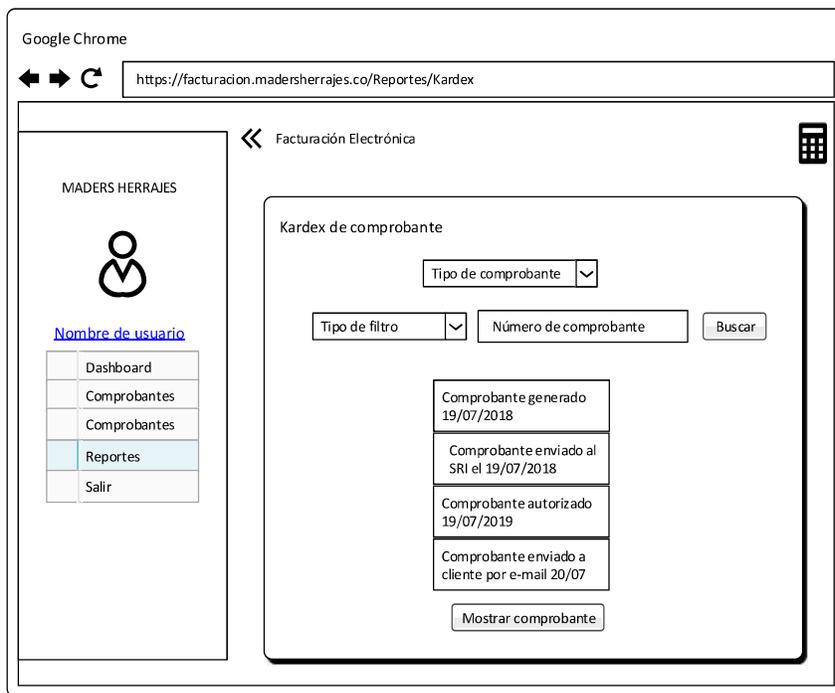


Figura 2.109 Vista de reporte de movimientos de comprobantes.

2.1.9.5. MÓDULO DE CLIENTES

Permite a los usuarios con rol cliente la visualización de comprobantes electrónicos generados en el sistema.

La

Figura 2.110 presenta el diagrama de actividades del módulo de clientes en donde se observa la interacción entre el usuario y las diferentes capas del sistema.

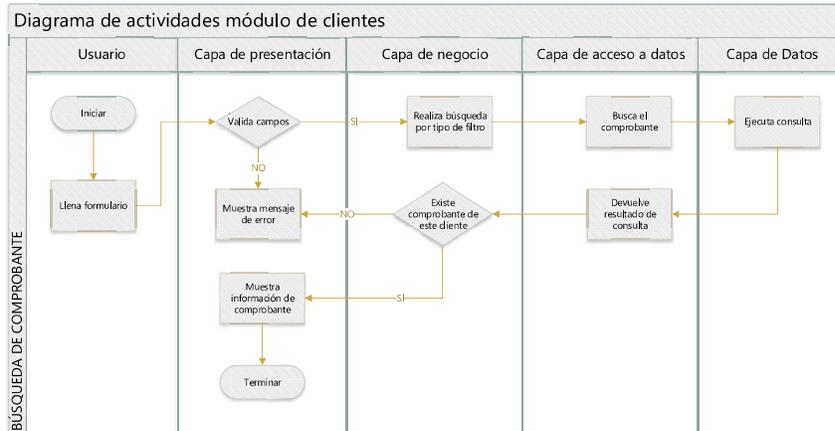


Figura 2.110 Diagrama de actividades del módulo de clientes.

2.1.9.5.1. Búsqueda de comprobantes

Esta vista permite al cliente realizar una búsqueda de comprobantes generados por el sistema.

El cliente debe conocer el tipo de comprobante y el número del mismo o a su vez la clave de acceso o número de autorización.

La Figura 2.111 muestra la vista en la que el cliente podrá realizar la búsqueda de los comprobantes electrónicos generados en el sistema.

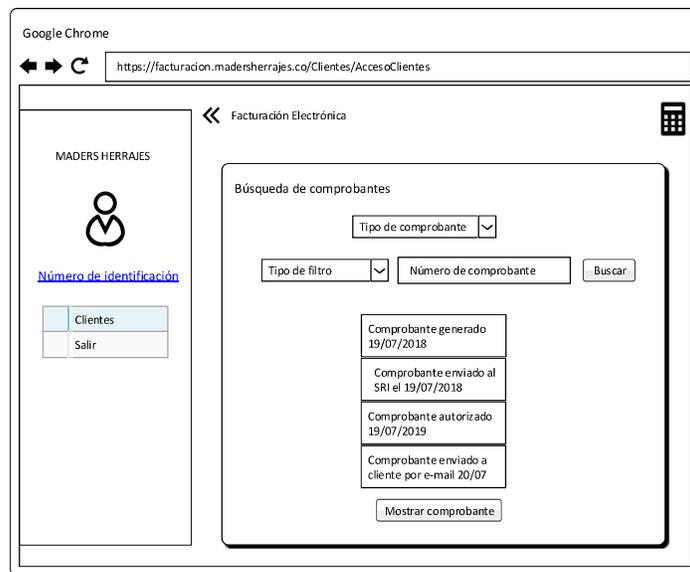


Figura 2.111 Vista de búsqueda de comprobantes de clientes.

2.2. CODIFICACIÓN DEL SISTEMA

En esta sección se detalla la planificación y desarrollo de cada uno de los prototipos definidos según la metodología de desarrollo seleccionada.

2.2.4. CODIFICACIÓN DE LA CAPA DE DATOS

Para el desarrollo de la capa de datos se utilizó Microsoft SQL Server 2017 Developer Edition mediante el IDE¹⁷ SQL Server Management Studio.

Para el desarrollo del prototipo se utilizó tablas, vistas, *triggers* y procedimientos almacenados. A continuación, se muestra un ejemplo de la creación de cada uno de los objetos utilizados de la base de datos.

En el Segmento de código 2.1 se muestra la instrucción necesaria para la creación de la tabla *Usuarios*, mediante la sentencia `CREATE TABLE`, como atributos de esta se han definido la columna *Id_Usuario* como clave primaria, entero auto incremental iniciando en 1 y no nulo, la columna *Id_Sucursal* se ha definido como una clave foránea de la tabla *Sucursales*, las columnas *Nombre*, *Username*, *Correo_Electronico* y *Telefono* se han establecido como columnas de tipo `VARCHAR` de un límite de caracteres especificados, las columnas *Password* y *Path_Image* se han definido como `VARCHAR` con límite máximo de caracteres debido a que la contraseña se almacena como un `HASH`¹⁸ y la ruta a la imagen de la sucursal almacenada en el servidor puede tener una extensa longitud. Adicionalmente la columna *Estado_Usuario* se ha definido de tipo bit ya que el usuario puede tener un estado activo (1) o inactivo (0), la columna *Tipo_Usuario* se ha definido de tipo `TINYINT` porque se ha definido un número pequeño de roles en el sistema. Además, la columna es de tipo `SMALLDATETIME` ya que almacenará la fecha y hora de registro del usuario y se tomará por defecto la fecha del sistema. Finalmente, con la instrucción `CREATE UNIQUE NONCLUSTERED INDEX` se ha definido una restricción en la base de datos para que no sea posible ingresar usuarios con el mismo nombre de usuario en estado activo.

En el Segmento de código 2.2 se presenta la instrucción necesaria para la creación de una vista a través de la instrucción `CREATE VIEW` que devuelve un conjunto de columnas agrupadas de las tablas *Comprobantes* y *Respuestas_Web_Service* haciendo uso de un alias para facilitar la legibilidad de la consulta, adicionalmente se emplea el uso de la instrucción `LEFT JOIN` para realizar una juntura de las tablas en cuestión.

¹⁷ **Integrated Development Environment (IDE)**: es un paquete de *software* que consolida las herramientas básicas que los desarrolladores necesitan para escribir y probar *software*.

¹⁸ Hash: para cifrar

Para la creación del procedimiento almacenado se emplea la instrucción `CREATE PROCEDURE` y se declaran los parámetros de entrada `@FechaInicio` y `@FechaFin` de tipo `DATE` para poder realizar un filtro de los datos de la vista `Vw_DatosProductos`. Además, se generaliza la fecha ya que solo se necesita determinar el mes mediante la instrucción `DATEFROMPARTS`. Finalmente se filtra en el rango de fechas seleccionado y se ordena en base a la cantidad y precio tal como se muestra en el Segmento de código 2.3.

```
--/Creación Tabla Usuario/
CREATE TABLE dbo.Usuarios
(
    Id_Usuario INT NOT NULL PRIMARY KEY IDENTITY(1,1),
    Id_Sucursal INT FOREIGN KEY REFERENCES Sucursales(Id_Sucursal),
    Nombre VARCHAR(250),
    Username VARCHAR(50) NOT NULL,
    Password VARCHAR(MAX) NOT NULL,
    Tipo_Usuario TINYINT NOT NULL,
    Correo_Electronico VARCHAR(150),
    Telefono VARCHAR(40),
    Fecha_Creacion SMALLDATETIME DEFAULT(GETDATE()),
    Path_Image VARCHAR(MAX),
    Estado_Usuario BIT NOT NULL DEFAULT (1)
)
GO
CREATE UNIQUE NONCLUSTERED INDEX UNIQUE_USERNAME
ON Usuarios(Username)
WHERE Estado_Usuario=1
GO
```

Segmento de código 2.1 Script para la creación de la tabla usuarios.

```
1 --/Creación Vista Vw_DatosEnvioComprobantes/
2 CREATE VIEW Vw_DatosEnvioComprobantes
3 AS
4 SELECT c.Id_Comprobante, c.Clave_Acceso, c.Numero_Autorizacion,
c.Nro_Comprobante, c.Id_Tipo_Comprobante [Tipo_Comprobante], c.Fecha_Emision,
c.Fecha_Generacion [Fecha_Ingreso], c.Fecha_Envio_Comprobante [Fecha_Envio],
c.Comprobante_Enviado_Cliente [EstadoEnvioCliente], c.Estado_Comprobante,
r.Estado [Estado_Envio_SRI], r.Fecha [Fecha_Envio_SRI]
5 FROM Comprobantes c
6 LEFT JOIN Respuestas_Web_Service r on r.Id_Comprobante = c.Id_Comprobante
7 GO
```

Segmento de código 2.2 Script para la creación de la tabla usuarios.

```

1 --/Creación Procedimiento almacenado Sp_GetDatosTendenciaVentas/
2 CREATE PROCEDURE Sp_GetDatosTendenciaVentas
3 @FechaInicio DATE,
4 @FechaFin DATE
5 AS
6 BEGIN
7 SELECT Precio, Cantidad,Codigo_Principal, Nombre_Producto, Id_Sucursal,
DATEFROMPARTS(DATEPART(Year, Fecha_Emision), DATEPART(Month, Fecha_Emision),
1) AS Periodo
8 FROM Vw_DatosProductos
9 WHERE Fecha_Emision BETWEEN @FechaInicio AND @FechaFin
10 ORDER BY Cantidad DESC, Precio DESC
11 END
12 GO

```

Segmento de código 2.3 Script para la creación del procedimiento almacenado Sp_GetDatosTendenciaVentas.

El segmento de código 2.4 muestra la generación del *trigger* Guardar_Emisor_Logs mediante la instrucción **CREATE TRIGGER** que almacena la información de los cambios realizados en la tabla Emisores en la tabla genérica Change_Log_Table cada vez que se genera una instrucción **Update**, **Insert** o **Delete**

```

--Crear desencadenador para guardar logs de tabla emisor
CREATE TRIGGER dbo.Guardar_Emisor_Logs
ON dbo.Emisores
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    DECLARE @operation CHAR(6)
    SET @operation = CASE
        WHEN EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT *
FROM deleted)
            THEN 'Update'
        WHEN EXISTS(SELECT * FROM inserted)
            THEN 'Insert'
        WHEN EXISTS(SELECT * FROM deleted)
            THEN 'Delete'
        ELSE NULL
    END
    IF @operation = 'Delete'
        INSERT INTO Change_Log_Table(Id_Registro, Fecha_Actualizacion,
Nombre_Tabla, Command_Type, Nombre_Campo_1, Antiguo_Valor_Campo_1, Nombre_Campo_2,
Antiguo_Valor_Campo_2, Nombre_Campo_3, Antiguo_Valor_Campo_3, UserName)
        SELECT d.Id_Emisor, GETDATE(), 'Emisor', @operation, 'RUC',
d.RUC, 'Razon_Social', d.Razon_Social, 'Direccion_Establecimiento',
d.Direccion_Establecimiento, USER_Name()
        FROM deleted d

    IF @operation = 'Insert'
        INSERT INTO Change_Log_Table(Id_Registro, Fecha_Actualizacion,
Nombre_Tabla, Command_Type, Nombre_Campo_1, Nuevo_Valor_Campo_1, Nombre_Campo_2,
Nuevo_Valor_Campo_2, Nombre_Campo_3, Nuevo_Valor_Campo_3, UserName)
        SELECT i.Id_Emisor, GETDATE(), 'Emisor', @operation, 'RUC',
i.RUC, 'Razon_Social', i.Razon_Social, 'Direccion_Establecimiento',
i.Direccion_Establecimiento, USER_Name()
        FROM inserted i

    IF @operation = 'Update'
        INSERT INTO Change_Log_Table(Id_Registro, Fecha_Actualizacion,
Nombre_Tabla, Command_Type, Nombre_Campo_1, Antiguo_Valor_Campo_1,
Nuevo_Valor_Campo_1, Nombre_Campo_2, Antiguo_Valor_Campo_2, Nuevo_Valor_Campo_2,
Nombre_Campo_3, Antiguo_Valor_Campo_3, Nuevo_Valor_Campo_3, UserName)
        SELECT d.Id_Emisor, GETDATE(), 'Emisor', @operation, 'RUC',
d.RUC,i.RUC, 'Razon_Social', d.Razon_Social,i.Razon_Social,
'Direccion_Establecimiento', d.Direccion_Establecimiento,
i.Direccion_Establecimiento, USER_Name()
        FROM deleted d, inserted i
END
GO

```

Segmento de código 2.4 Script para la creación del trigger Guardar_Emisor_Logs.

Los scripts utilizados para el desarrollo de tablas, vistas, *triggers* y procedimientos almacenados de la base de datos se encuentran disponibles en el Anexo II.

2.2.5. CODIFICACIÓN DE LA CAPA DE ACCESO A DATOS

Para el desarrollo de las capas del sistema se utilizó el entorno de desarrollo integrado Visual Studio 2017, donde se creó una solución en blanco y se agregó un nuevo proyecto por cada capa del sistema.

La Figura 2.112 muestra el diseño de la arquitectura n-capas en una solución Visual Studio.

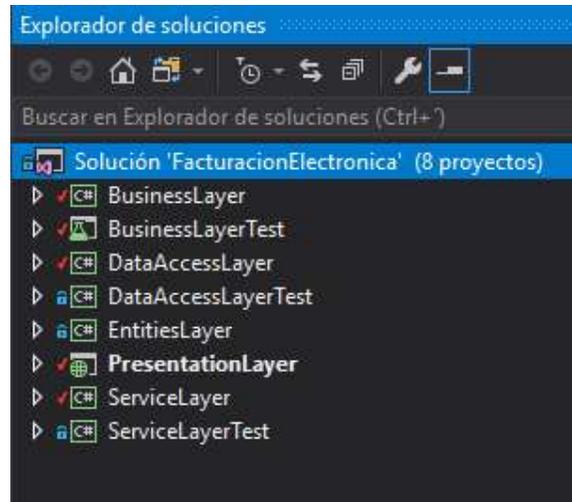


Figura 2.112 Estructura de una aplicación n-capas en Visual Studio 2017.

Para la capa de acceso a datos se utilizó un proyecto de tipo Biblioteca de Clases (.Net Framework). La Figura 2.113 muestra el tipo de proyecto en el IDE Visual Studio

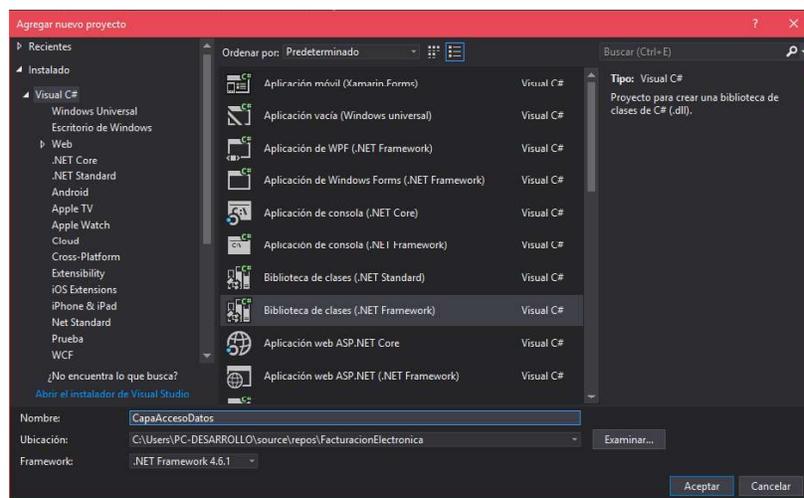


Figura 2.113 Creación de un proyecto de biblioteca de clases (.NET Framework) en Visual Studio 2017.

Para el acceso a la capa de datos se utilizó Entity Framework 6. La Figura 2.114 muestra la forma de agregar un componente entity framework a un proyecto.

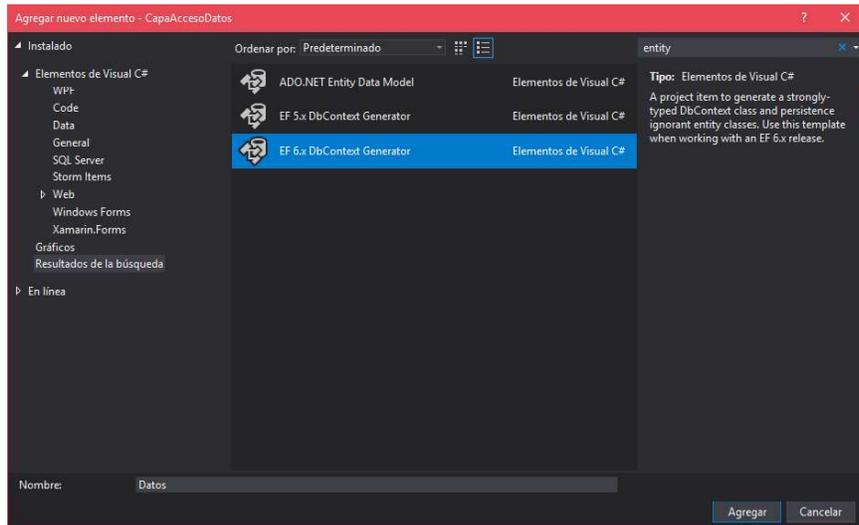


Figura 2.114 Creación de un elemento Entity Framework en Visual Studio 2017.

Haciendo uso del patrón de diseño repositorio para el consumo de las acciones de la capa de datos se definió la interfaz `IRepository` para exponer las acciones de la base de datos. El Segmento de código 2.5 muestra el código de la interfaz `IRepository`

```

public interface IRepository : IDisposable
{
    /// Crea un objeto de tipo generico
    TEntity Create(TEntity objCreate) where TEntity : class;
    /// Elimina un objeto de tipo genérico
    bool Eliminar(TEntity objEliminar) where TEntity : class;
    /// Actualiza un objeto de tipo generico
    bool Actualizar(TEntity objActualizar) where TEntity :
class;
    /// Devuelve una lista de objetos genéricos filtrados por una
expresión
    List<TEntity> Buscar<TEntity>(Expression<Func<TEntity, bool>>
criterio, List<Expression<Func<TEntity, object>>> entidades) where TEntity :
class;
    /// Devuelve una lista de entidades de tipo generico.
    List<TEntity> Listar<TEntity>(List<Expression<Func<TEntity,
object>>> entidades) where TEntity : class;
    /// Ejecuta un procedimiento de la BDD de forma genérica
    List<TEntity> EjecutarProcedimiento<TEntity>(string
nombreProcedimiento, List<SqlParameter> parametros) where TEntity : class;
}

```

Segmento de código 2.5 Definición de la interfaz `IRepository`.

Se trabajó con objetos genéricos para reutilizar la mayor parte del código mejorando el tiempo de desarrollo del sistema.

Siguiendo los lineamientos de la arquitectura se implementó la interfaz en la clase `Repository` que permite el acceso al contexto de datos mediante *entity framework*, así

mismo haciendo uso del patrón *Factory* se creó la clase `RepositoryFactory` para realizar la llamada al contexto de datos seleccionado. El Segmento de código 2.6 muestra un ejemplo de la implementación de la interfaz `IRepository`

```
public class Repository : IRepository
{
    DbContext Context;
    public Repository(DbContext dbContext)
    {
        this.Context = dbContext;
    }

    public TEntitydad Crear<TEntidad>(TEntidad objCrear) where TEntitydad :
class
    {
        TEntitydad Result = default(TEntidad);
        try
        {
            Result = Context.Set<TEntidad>().Add(objCrear);
            Context.SaveChanges();
        }
        catch (Exception)
        { }
        return Result;
    }
}
```

Segmento de código 2.6 Ejemplo de implementación de la interfaz `IRepository`.

```
public class RepositoryFactory
{
    public static IRepository CreateRepository()
    {
        var context = new FacturacionElectronicaDatabaseEntities();
        context.Configuration.ProxyCreationEnabled = false;
        return new Repository(context);
    }
}
```

Segmento de código 2.7 Ejemplo de implementación del patrón de diseño `Factory`.

Siguiendo la metodología de desarrollo se realizaron pruebas del prototipo desarrollado para lo cual se codificó un proyecto de tipo Aplicación de consola (.NET Framework). La Figura 2.115 muestra la estructura del proyecto de prueba para la capa de acceso a datos. Los resultados de las pruebas del prototipo se discuten en el siguiente capítulo del presente trabajo de titulación.

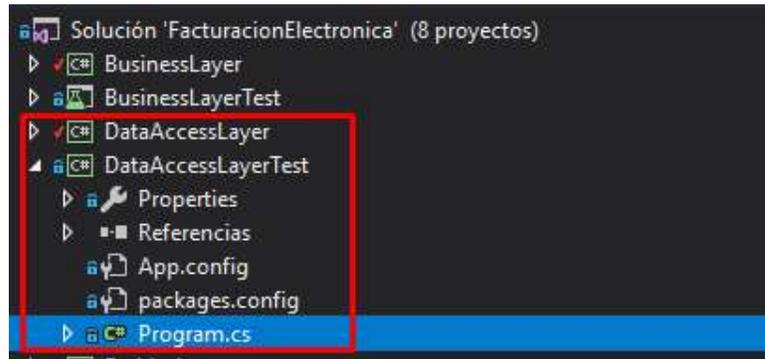


Figura 2.115 Estructura proyecto DataAccessLayerTest.

2.2.6. CODIFICACIÓN DE LA CAPA DE ENTIDADES

Para la codificación de la capa de Entidades al igual que la capa de acceso de datos se hizo uso de la herramienta Entity Framework ya que al realizar el mapeo de las entidades de la base de datos a clases para ser utilizadas en el sistema.

Para realizar la capa de Entidades se utilizó un proyecto de tipo librería de clases (.Net Framework) y se realizó el desacople de las entidades de la capa de acceso a datos mediante la configuración del archivo .tt del contexto de datos generado por *entity framework*. La Figura 2.116 muestra la estructura del contexto de datos generado con la herramienta.

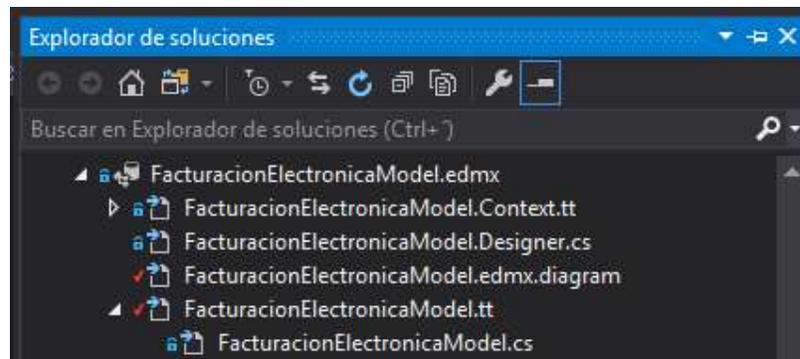


Figura 2.116 Estructura del contexto de datos generado por Entity Framework.

La Figura 2.117 muestra las entidades desacopladas de la capa de acceso a datos y presentes en la capa de entidades junto al resto de entidades utilizadas por el sistema.

2.2.7. CODIFICACIÓN DE LA CAPA DE NEGOCIO

Para el desarrollo de la capa de negocio se utilizó un proyecto de tipo biblioteca de clases (.NET Framework).

La capa de negocio consume las capas de acceso a datos y entidades como se muestra en la Figura 2.118, la capa de negocio tiene referencias a los proyectos mencionados.

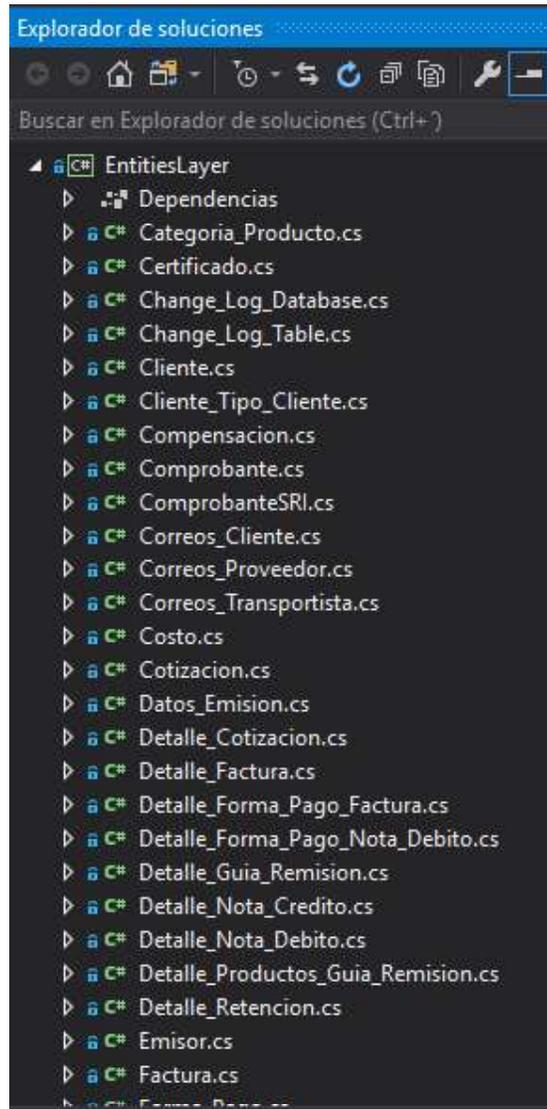


Figura 2.117 Estructura del proyecto EntitiesLayer.

El Segmento de código 2.8 muestra un ejemplo del consumo de la capa de acceso a datos desde la capa de negocio.

Una vez finalizado el prototipo siguiendo la metodología de desarrollo se realizaron las pruebas de funcionamiento del prototipo. Dichas pruebas se realizaron mediante el desarrollo de un proyecto de tipo Proyecto de Prueba Unitaria (.NET Framework), los resultados de las pruebas de funcionamiento se discuten en el tercer capítulo del presente trabajo.

La Figura 2.119 muestra la estructura del proyecto de prueba para la capa de negocio.

2.2.8. CODIFICACIÓN DE LA CAPA DE SERVICIO

Para el desarrollo de la capa de servicio se utilizó un proyecto de tipo biblioteca de clases (.NET Framework).

Como se muestra en la Figura 2.120 la capa de servicio realiza el consumo de las capas de negocio y entidades.

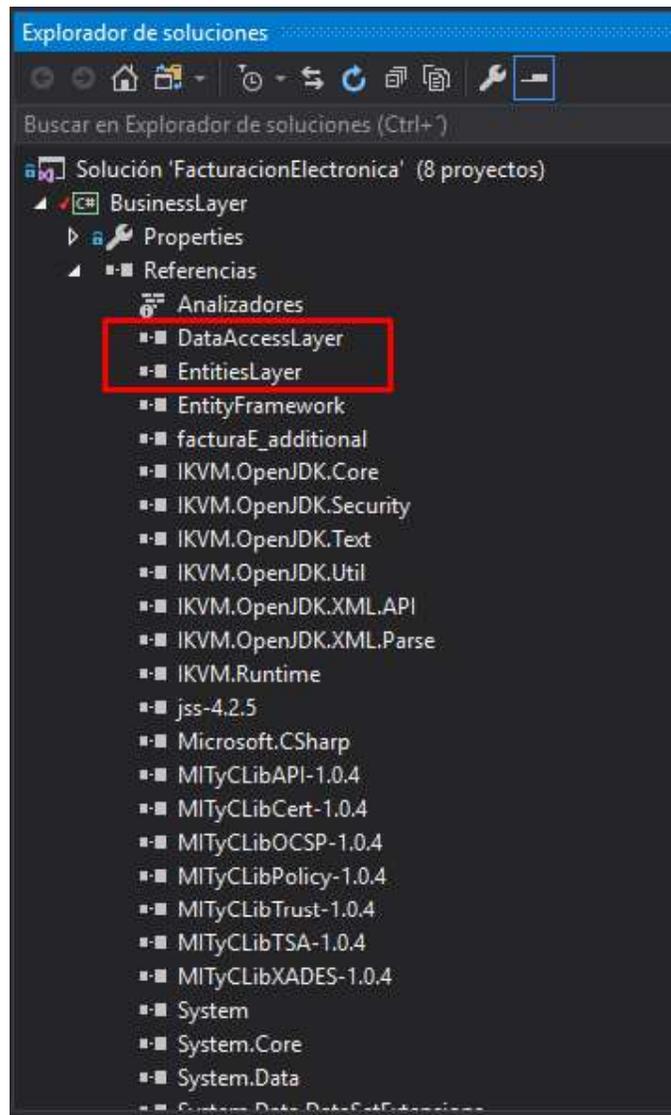


Figura 2.118 Referencias del proyecto BusinessLayer.

El consumo de servicios web en .NET se puede hacer de dos maneras diferentes descritas a continuación:

- Generando el mensaje HTTP y realizando la solicitud directamente. En este caso se debe generar todo el mensaje de la solicitud a través de código. El Segmento de código 2.9 muestra la forma de consumir un servicio web realizando la solicitud HTTP desde código.
- Agregando la referencia al servicio web requerido. En este caso el compilador genera la solicitud HTTP y solo se debe realizar el consumo de los métodos expuestos por el servicio web. La Figura 2.121 muestra la estructura de la referencia a servicios web.

```

public Categoria_Producto AgregarCategoria_Producto(string codigo, string
nombre, string descripcion)
{
    Categoria_Producto Result = null;
    using (var r = RepositoryFactory.CreateRepository())
    {
        Categoria_Producto newCategoria_Producto = new
Categoria_Producto
        {
            Nombre_Categoria = nombre,
           Codigo_Categoria = codigo,
           Descripcion_Categoria = descripcion
        };
        r.Crear(newCategoria_Producto);
        Result = newCategoria_Producto;
    }
    return Result;
}
}

```

Segmento de código 2.8 Método para agregar una categoría

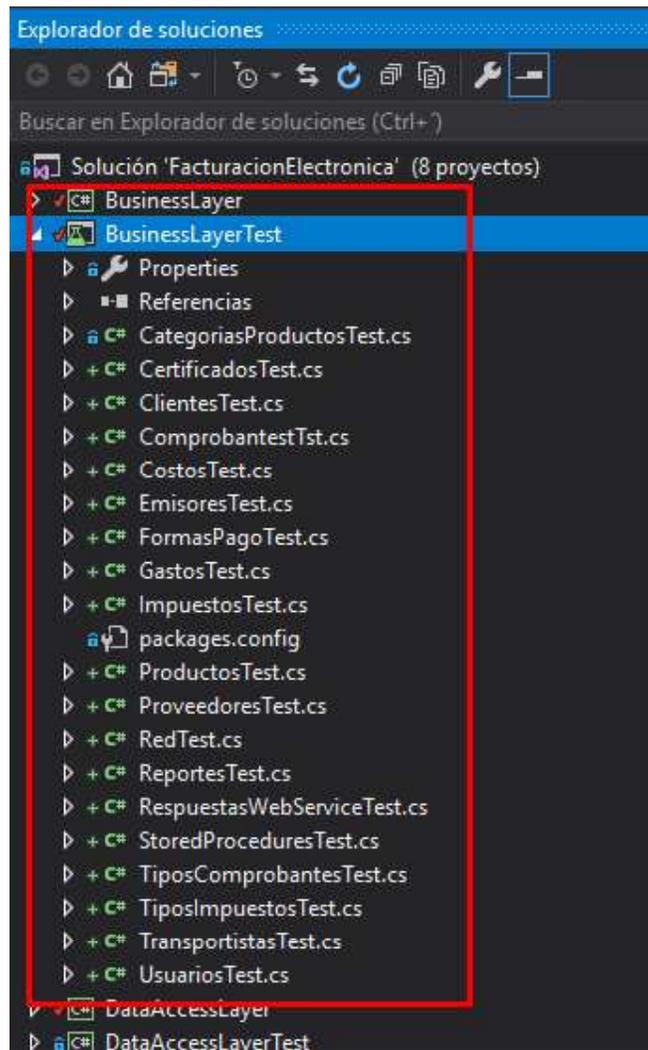


Figura 2.119 Estructura proyecto BusinessLayerTest.

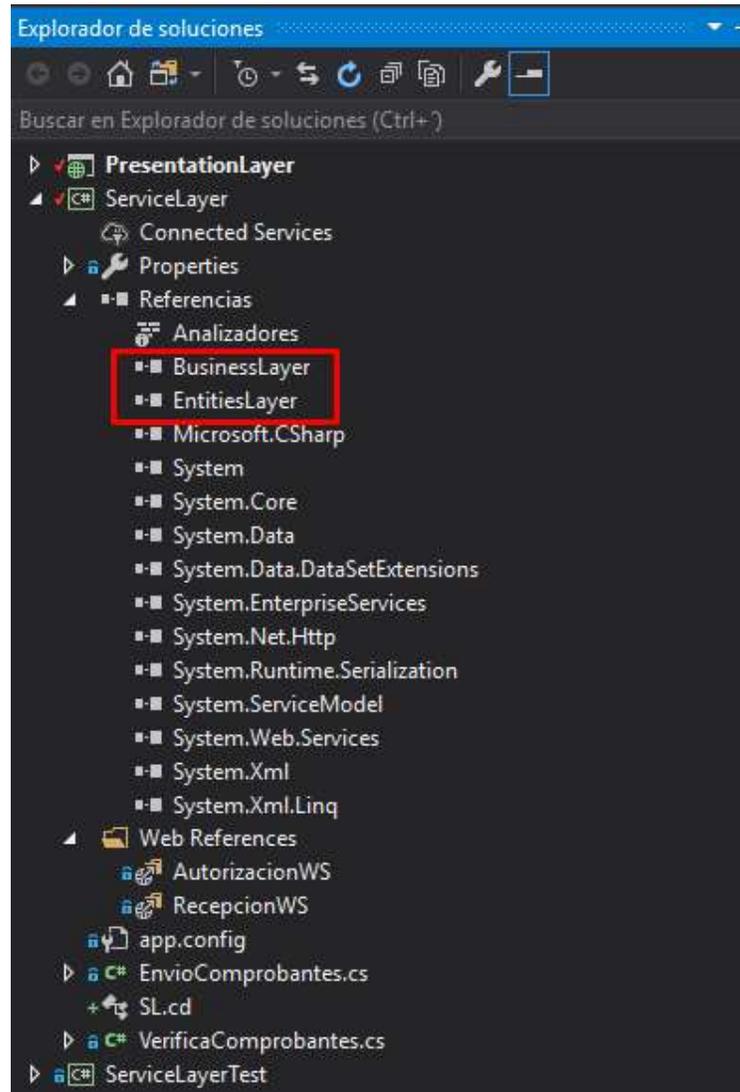


Figura 2.120 Estructura capa de servicio.

Una vez finalizado el prototipo siguiendo la metodología de desarrollo se realizó el proyecto para realizar las pruebas de funcionamiento del prototipo. Dicho proyecto es de tipo Aplicación de consola (.NET Framework), los resultados de las pruebas de funcionamiento se discuten en el tercer capítulo del presente trabajo.

La Figura 2.122 muestra la estructura del proyecto de prueba para la capa de servicio.

```

string xml = "<soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:ec=\"http://ec.gob.sri.ws.recepcion/\">";
xml = xml + "<soapenv:Header/>";
xml = xml + "<soapenv:Body>";
xml = xml + "<ec:validarComprobante>";
xml = xml + "<xml>" + encodedData + "</xml>";
xml = xml + "</ec:validarComprobante>";
xml = xml + "</soapenv:Body>";
xml = xml + "</soapenv:Envelope>";

byte[] bytes = System.Text.Encoding.ASCII.GetBytes(xml);
HttpRequest request =
(HttpWebRequest)WebRequest.Create(UriBase + UriAction);

request.Method = "POST";
request.ContentLength = bytes.Length;
request.ContentType = "text/xml";

Stream requestStream = request.GetRequestStream();
requestStream.Write(bytes, 0, bytes.Length);
requestStream.Close();

HttpWebResponse response =
(HttpWebResponse)request.GetResponse();
if (response.StatusCode == HttpStatusCode.OK)
{
    Stream responseStream = response.GetResponseStream();
    StreamReader reader = new StreamReader(responseStream);
    resultado = reader.ReadToEnd();
}
resultado = WebUtility.HtmlDecode(resultado);
response.Close();

```

Segmento de código 2.9 Generación solicitud HTTP desde código C#.

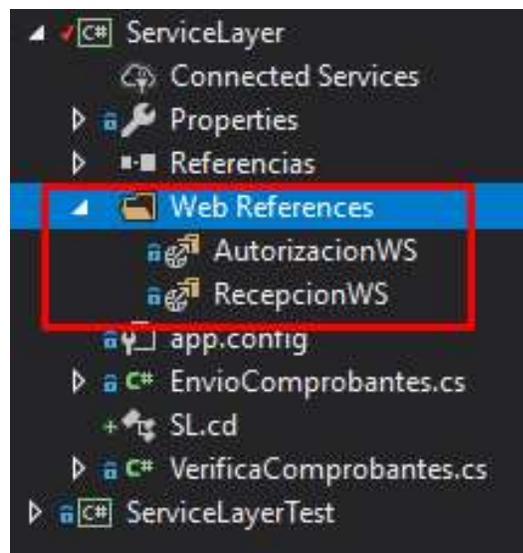


Figura 2.121 Estructura referencias a servicios web

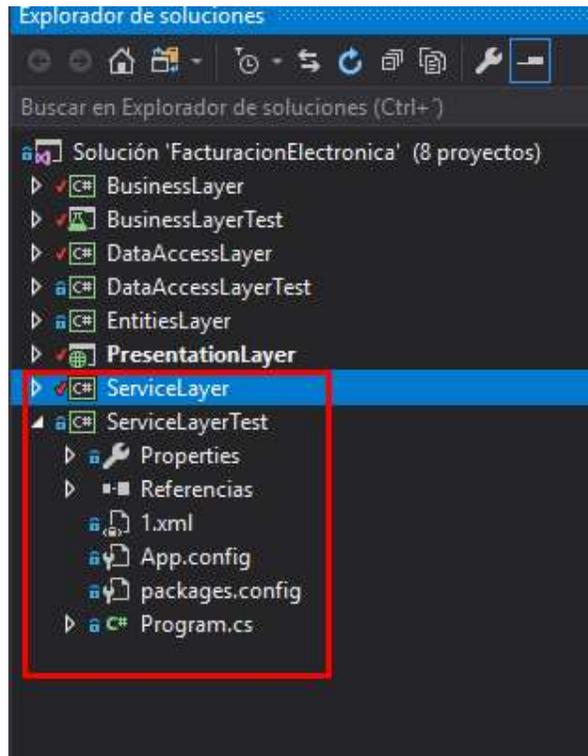


Figura 2.122 Estructura proyecto ServiceLayerTest.

2.2.9. CODIFICACIÓN DE LA CAPA DE PRESENTACIÓN

La capa de presentación se desarrolló teniendo en cuenta las vistas de usuario y diseño de procesos realizados previamente en la sección 2.10.

Para el desarrollo de la interfaz se utilizó el lenguaje HTML5 en conjunto con CSS3, JavaScript y el *framework* Bootstrap con el fin de proporcionar una interfaz atractiva y de utilidad al usuario independientemente del dispositivo con el que se acceda al sistema.

Para la capa de presentación se utilizó un proyecto de tipo Aplicación Web ASP.NET *Web Forms*.

Como se indica en la Figura 2.123, la capa de presentación consume la capa de negocio, servicio y entidades cumpliendo con la arquitectura del sistema.

La codificación de la capa de presentación se realizó en base a los módulos definidos para el sistema.

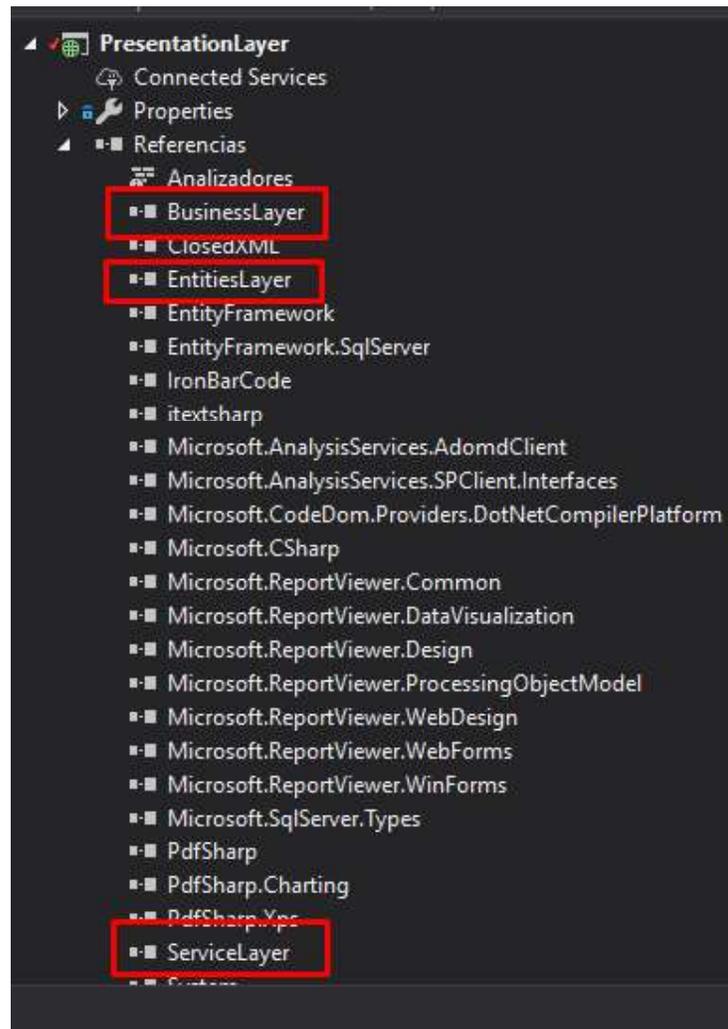


Figura 2.123 Referencias del proyecto PresentationLayer.

2.2.9.1. Módulo de autenticación

Para el módulo de autenticación se ha definido el formulario `wfrm_Login.aspx` en conjunto con la página maestra `Main.aspx` que permiten el acceso al sistema en base al rol del usuario y muestra únicamente los menús a los que el usuario tiene acceso.

La página maestra establece la estructura general y los elementos comunes a todos los formularios hijos. Por lo tanto, cada página hija reutilizará el código HTML establecido para la página maestra agregando su propio código definido en un elemento `ContentPlaceholder`.

Para la validación de los campos de ingreso de credenciales en el formulario `wfrm_Login.aspx` se utiliza controles de tipo `RequiredFieldValidator` y `RegularExpressionValidator` permitiendo realizar una validación en el lado del cliente optimizando el rendimiento del sistema. El Segmento de código 2.10 muestra un ejemplo de uso de los validadores del lado del cliente.

```

<asp:TextBox runat="server" CssClass="form-control" ID="txtUsername" />

<asp:RequiredFieldValidator runat="server" ControlToValidate="txtUsername"
ValidationGroup="ValidarLogin" Text="Debe ingresar el usuario" CssClass="text-
danger" />

<asp:RegularExpressionValidator runat="server" ValidationGroup="ValidarLogin"
ErrorMessage="Se debe ingresar solo letras y numeros" CssClass="text-danger"
ControlToValidate="txtUsername" ValidationExpression="^[a-zA-Z0-
9]*$" ></asp:RegularExpressionValidator>

```

Segmento de código 2.10 Validación de campos del lado del cliente.

La validación del usuario se realiza en la capa de negocio en donde se realiza el HASH ¹⁹a partir de la contraseña y se valida la existencia del usuario a través de la capa de acceso a datos. El Segmento de código 2.11 muestra la clase utilizada para calcular el HASH a partir de un *string*.

```

public class Encoding
{
    public const string ClaveCifradoSQL = "f@ctUr4c10N$2019";

    public static string GetHash(string input)
    {
        HashAlgorithm hashAlgorithm = new SHA256CryptoServiceProvider();

        byte[] byteValue = System.Text.Encoding.UTF8.GetBytes(input);

        byte[] byteHash = hashAlgorithm.ComputeHash(byteValue);

        return Convert.ToBase64String(byteHash);
    }
}

```

Segmento de código 2.11 Definición clase Encoding.

Para el almacenamiento seguro de cookies y credenciales se ha utilizado la autenticación y autorización en formularios ASP.NET. El Segmento de código 2.12 muestra la configuración en el archivo `web.config` para establecer el nivel de seguridad requerido.

```

<authentication mode="Forms">
    <forms name="Auth" path="/" loginUrl="Account/Wfrm_Login.aspx"
defaultUrl="Wfrm_Dashboard.aspx" protection="All" timeout="525600" />
</authentication>

```

Segmento de código 2.12 Configuración de seguridad mediante formularios

¹⁹ **HASH:** Es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija.

La Figura 2.124 muestra la cookie almacenada por el sitio cifrada por los formularios de ASP.NET.

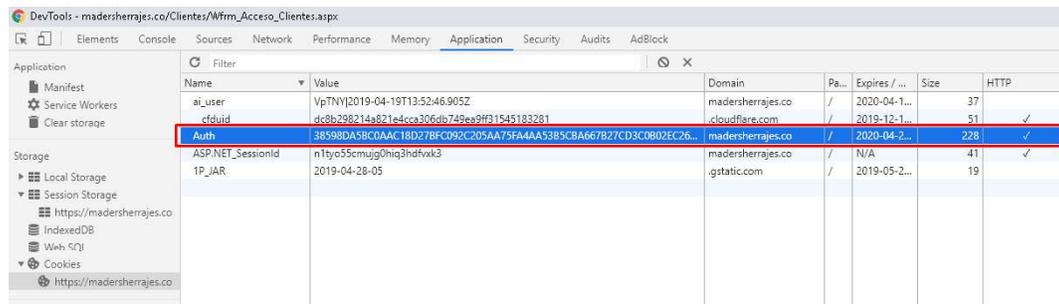


Figura 2.124 Cookie del usuario almacenada en el navegador

2.2.9.2. Módulo de administración

Para el módulo de administración se ha codificado los formularios descritos en la Tabla 2.28.

Tabla 2.28 Formularios para el módulo de administración

Formulario	Descripción
Wfrm_Perfil.aspx	Formulario que permite administrar el perfil personal del usuario
Wfrm_Usuarios.aspx	Formulario para administrar los usuarios del sistema
Wfrm_Categorias_Productos.aspx	Formulario para administrar las categorías de productos del sistema
Wfrm_Certificados.aspx	Formulario para administrar certificados digitales
Wfrm_Clientes.aspx	Formulario para administrar clientes del sistema
Wfrm_Emisores.aspx	Formulario para administrar emisores del sistema
Wfrm_Formas_Pago.aspx	Formulario para administrar formas de pago del sistema
Wfrm_Gestios_Costos.aspx	Formulario para administrar costos del sistema
Wfrm_Gestion_Gastos.aspx	Formulario para administrar gastos del sistema
Wfrm_Impuestos.aspx	Formulario para administrar impuestos del sistema
Wfrm_Productos.aspx	Formulario para administrar productos del sistema
Wfrm_Proveedores.aspx	Formulario para administrar proveedores del sistema
Wfrm_Red.aspx	Formulario para administrar parámetros de configuración del sistema
Wfrm_Sucursales.aspx	Formulario para administrar sucursales del sistema
Wfrm_Tipos_Comprobantes.aspx	Formulario para administrar tipos de comprobantes del sistema
Wfrm_Transportistas.aspx	Formulario para administrar transportistas del sistema

Adicionalmente se ha almacenado en directorios particulares los formularios de cada módulo para mejor administración y seguridad. La Figura 2.125 muestra la estructura del módulo de administración en la capa de presentación.

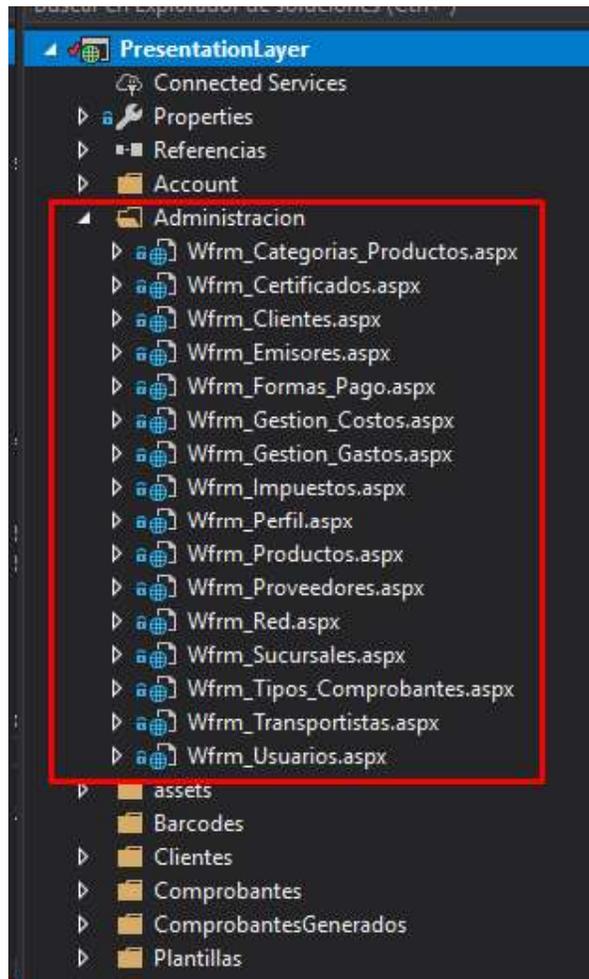


Figura 2.125 Directorio de almacenamiento del módulo de administración

2.2.9.3. Módulo de comprobantes

Para el módulo de comprobantes se han definido los formularios descritos en la Tabla 2.29.

Tabla 2.29 Formularios para el módulo de comprobantes

Formulario	Descripción
Wfrm_Cotizaciones.aspx	Permite generar cotizaciones
Wfrm_Facturas.aspx	Permite generar facturas y verificar su estado en línea
Wfrm_Retenciones.aspx	Permite generar comprobantes de retención y verificar su estado en línea
Wfrm_Notas_Credito.aspx	Permite generar notas de crédito y verificar su estado en línea
Wfrm_Notas_Debito.aspx	Permite generar notas de débito y verificar su estado en línea
Wfrm_Mostrar_Comprobante.aspx	Permite visualizar un comprobante seleccionado y enviarlo por correo electrónico o descargarlo.

Para la generación del código de barras se utilizó la librería Open Source IronBarcode disponible a través de paquetes NuGet de Visual Studio. El código de barras correspondiente a la clave de acceso se almacena en el directorio Barcodes del servidor web. La Figura 2.126 muestra el administrador de paquetes NuGet.

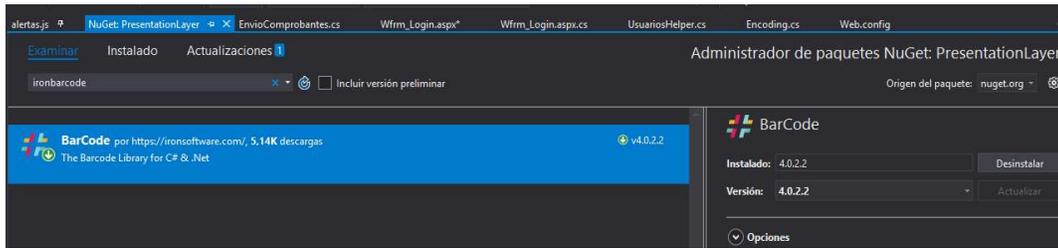


Figura 2.126 Administrador de paquetes NuGet

- Proceso de generación de comprobantes

Para la generación de comprobantes electrónicos se deben pasar por los siguientes subprocesos

1. *Generación de comprobante*: se genera el comprobante y se almacena en el sistema
2. *Generación archivo XML*: en base a la información del comprobante ingresado se genera el archivo XML en el servidor web en el directorio ComprobantesGenerados.
3. *Búsqueda de certificado digital*: en base al emisor que generó el comprobante se busca el archivo del certificado digital almacenado en el servidor web.
4. *Descifrar la clave del certificado*: se recupera la contraseña almacenada en la capa de datos y se descifra para firmar el archivo XML.
5. *Firmar el archivo XML*: se firma el archivo XML del directorio ComprobantesGenerados y se reemplaza el archivo anterior.
6. *Envío del comprobante*: se envía el archivo XML firmado del comprobante haciendo uso de los servicios web del SRI consumiendo la capa de servicio. El tiempo de espera del servicio web está definido en el emisor del comprobante.
7. *Comprobación de autorización del comprobante*: se comprueba el estado del comprobante en el SRI a través de los servicios web.

En la Figura 2.127 se muestra la estructura del módulo de comprobantes dentro de la capa de presentación.

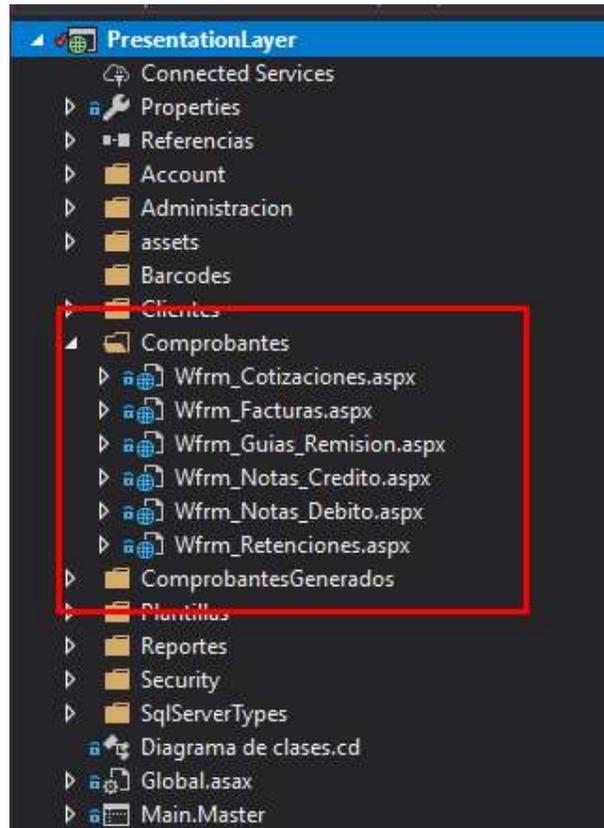


Figura 2.127 Módulo de comprobantes en capa de presentación

Para la visualización de comprobantes se utilizó la herramienta Microsoft ReportViewer pudiendo definir plantillas para los comprobantes del sistema y cargar dinámicamente la información del comprobante. A continuación, la Figura 2.128 muestra la plantilla generada para facturas del sistema.

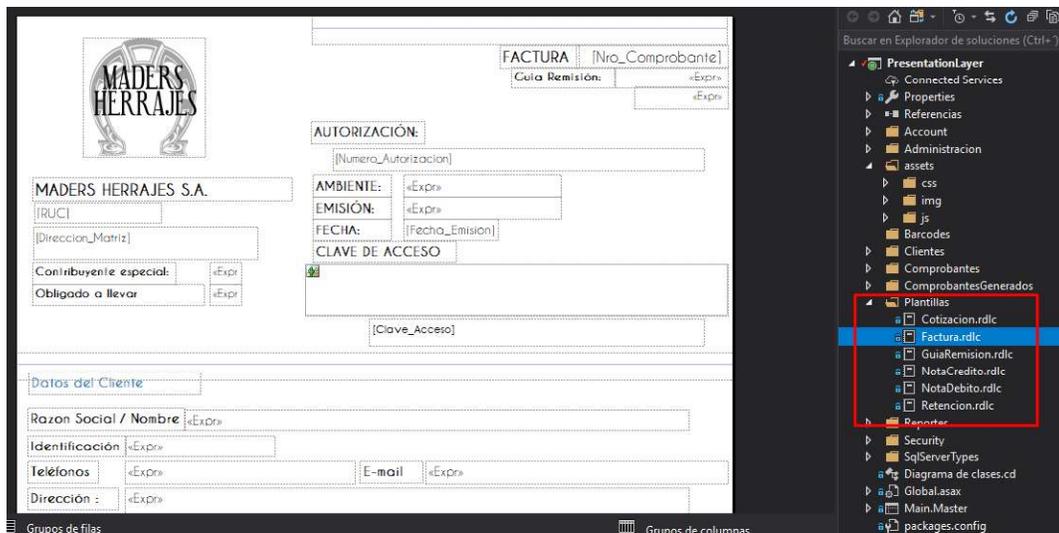


Figura 2.128 Plantillas de comprobantes electrónicos.

2.2.9.4. Módulo de reportes

Para la generación del módulo de reportes se codificaron los formularios descritos en la Tabla 2.30.

Tabla 2.30 Formularios para el módulo de reportes

Formulario	Descripción
Wfrm_Analisis_Costos.aspx	Permite generar reportes de análisis de costos
Wfrm_Analisis_Gastos.aspx	Permite generar reportes de análisis de gastos
Wfrm_Compras_vs_Ventas.aspx	Permite generar reportes de compras vs ventas
Wfrm_Kardex.aspx	Permite visualizar todos los movimientos de un comprobante
Wfrm_Productos_Vendidos.aspx	Permite generar reportes de productos vendidos
Wfrm_Tendencia_Ventas.aspx	Permite generar reportes de tendencia de ventas
Wfrm_Dashboard.aspx	Es el formulario mostrado al ingresar al sistema permite tener una vista general del negocio o sucursal presentando estadísticas de comprobantes.

Para la obtención de datos para los reportes se hace uso de las vistas definidas en la capa de datos, además para la aplicación de filtros se utilizan procedimientos almacenados sobre las vistas generadas permitiendo reducir la complejidad de las consultas y mejorar el rendimiento del sistema.

Para la llamada de los métodos definidos en el formulario web se utilizó AJAX²⁰. El Segmento de código 2.13 muestra un ejemplo del uso de AJAX en JavaScript.

Para la renderización de los gráficos para los reportes se utilizó la librería JavaScript ChartJS que permite realizar gráficos mediante el elemento canvas de HTML5. A continuación, la Figura 2.129 presenta un ejemplo de las estadísticas generadas con las librerías ChartJS.

2.2.9.5. Módulo de clientes

Para la codificación del módulo de cliente se requirió únicamente el formulario Wfrm_Acceso_Clientes.aspx ya que la única acción del usuario del sistema con rol cliente es buscar los comprobantes electrónicos generados por el sistema.

El cliente puede visualizar el estado del comprobante, así mismo verificar todos los movimientos que se han registrado con este desde su generación hasta el envío al correo electrónico del cliente.

La Figura 2.130 muestra el módulo de clientes en la capa de presentación.

²⁰ **Asynchronous JavaScript and XML (AJAX):** Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano

```

function cargarInfoVentas() {
    var labels = [];
    var chartData = [];
    var chartData2 = [];
    var data = [];
    var async1 = $.ajax({
        type: "POST",
        url: "/Reportes/Wfrm_Gerencial_Ventas.aspx/GetVentasData",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        success: function (result) {
            $.each(result.d, function (key, value) {
                labels.push(value.Mes);
                chartData.push(value.Valor);
                chartData2.push(value.Valor2);
            });
            data.push(labels, chartData, chartData2);
            console.log(data);
            ventasCharts(data);
        },
        error: function () {
            console.log("error");
        }
    });

    $.when(async1).done(function () {
    });
}

```

Segmento de código 2.13 Llamada a método C# desde JavaScript (AJAX).

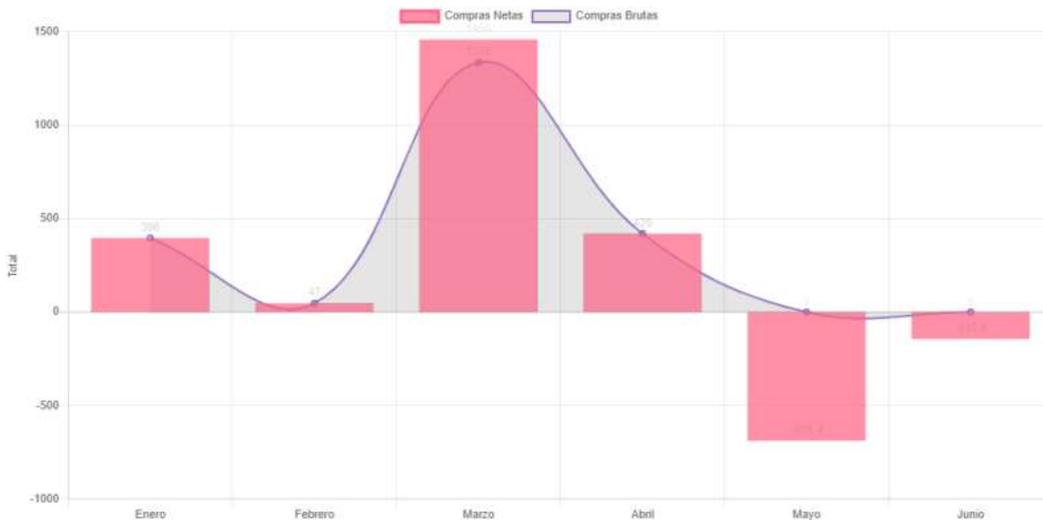


Figura 2.129 Gráfico generado con ChartJS.

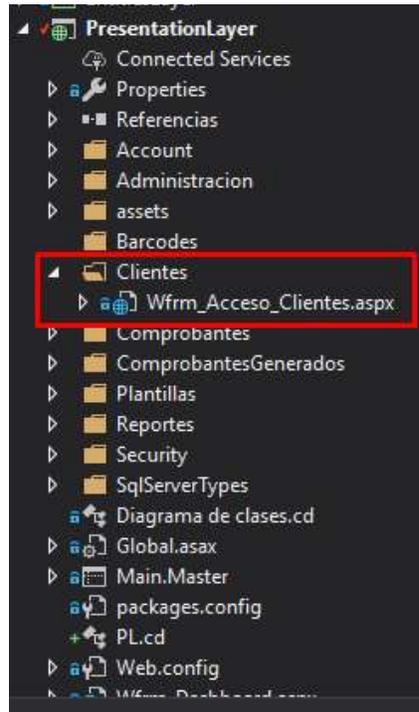


Figura 2.130 Módulo de clientes en la capa de presentación.

2.3. IMPLEMENTACIÓN DEL SISTEMA EN SERVIDOR WEB

Una vez finalizados todos los prototipos de las capas de la arquitectura se realizó la integración del sistema.

Para la implementación y despliegue del sistema en Internet se utilizó un servidor propio y se realizó la configuración del servidor.

2.3.4. PREPARACIÓN DEL SERVIDOR

Previo a la preparación del servidor se adquirió el dominio madersherrajes.co a través de la plataforma en línea GoDaddy. La Figura 2.131 muestra la configuración del servicio DNS en la plataforma GoDaddy.

Administración de DNS MADERSHERRAJES.CO

Tipo	Nombre	Valor	TTL
A	@	190.57.191.211	1 hora
CNAME	facturacion	@	1 hora
CNAME	www	@	1 hora
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com	1 hora

Figura 2.131 Administración de DNS de dominio en plataforma GoDaddy.

La Figura 2.132 presenta la configuración de la IP pública manejada por el router, la misma dirección IP a la cual el dominio debe redirigirse.

WAN	
MAC Address:	14-CC-20-AE-11-D3
IP Address:	190.57.191.211 Static IP
Subnet Mask:	255.255.255.248
Default Gateway:	190.57.191.209
DNS Server:	200.105.225.4 , 200.105.225.8

Figura 2.132 Configuración IP pública.

La Figura 2.133 presenta la configuración de la intranet del router.

LAN	
MAC Address:	14-CC-20-AE-11-D2
IP Address:	192.168.0.1
Subnet Mask:	255.255.255.0

Figura 2.133 Configuración LAN.

Para el ambiente de producción el servidor fue configurado con la dirección IPv4 192.168.0.50 para la intranet tal como se muestra en la Figura 2.134 al cual se realiza la redirección de puertos desde la IP pública, por motivos de seguridad se ha redirigido únicamente el puerto 443 válido para conexiones SSL²¹ tal configuración se muestra en la Figura 2.135.

<input type="radio"/>	Obtener una dirección IP automáticamente
<input checked="" type="radio"/>	Usar la siguiente dirección IP:
Dirección IP:	192 . 168 . 0 . 50
Máscara de subred:	255 . 255 . 255 . 0
Puerta de enlace predeterminada:	192 . 168 . 0 . 1
<input type="radio"/>	Obtener la dirección del servidor DNS automáticamente
<input checked="" type="radio"/>	Usar las siguientes direcciones de servidor DNS:
Servidor DNS preferido:	200 . 105 . 225 . 4
Servidor DNS alternativo:	200 . 105 . 225 . 8

Figura 2.134 Configuración IPv4 del servidor de producción

²¹ **Secure Sockets Layer (SSL):** Es un protocolo de seguridad que hace permite a los datos viajar de manera segura por Internet a través de claves digitales.

La Figura 2.135 presenta la configuración utilizada para la redirección de puertos de las solicitudes realizadas al enrutador a través de la dirección IP pública al servidor en la red interna.

Virtual Servers						
Note: Make sure the nat is enable if you want the Virtual Servers configuration take effect						
ID	Service Port	Internal Port	IP Address	Protocol	Status	Modify
1	443	443	192.168.0.50	TCP	Enabled	Modify Delete

Figura 2.135 Configuración para redirección de puertos

2.3.5. MIGRACIÓN DE BASE DE DATOS

El servidor de base de datos para el prototipo desplegado fue SQL Server 2017 en su version Express Edition que cuenta con una capacidad máxima de almacenamiento de información de 10GB.

Para la migración de la base de datos se utilizó el IDE SQL Server Management Studio mediante el almacenamiento y carga de un archivo .bak de la base de datos de pruebas.

La Figura 2.136 presenta la migración de la base de datos mediante la restauración de un archivo .bak.

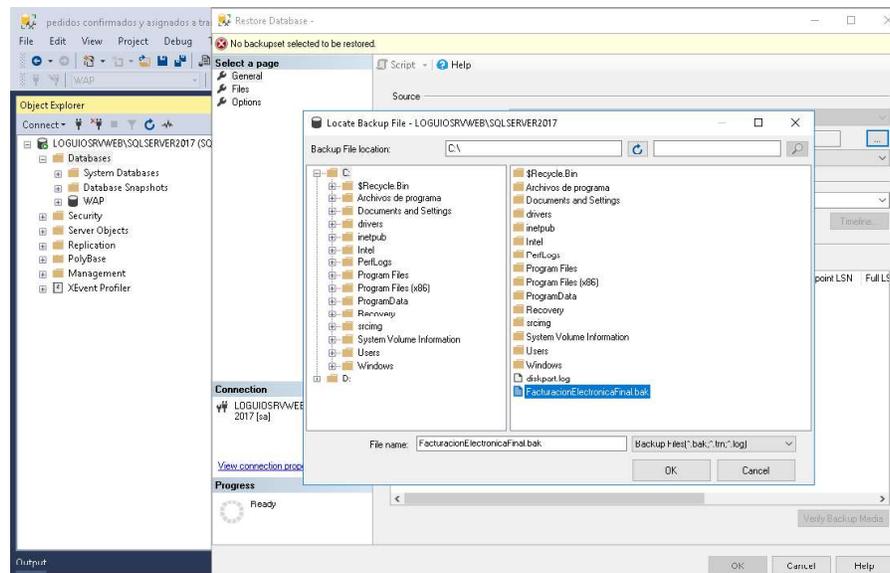


Figura 2.136 Carga de base de datos en SQL Server

La Figura 2.137 muestra la base de datos migrada al servidor de producción. Se verificó la carga de todas las tablas, vistas, procedimientos almacenados y triggers.

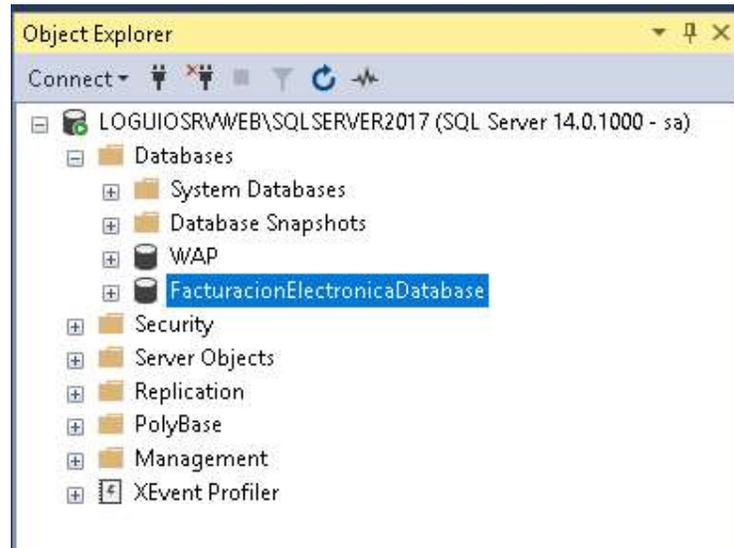


Figura 2.137 Base de datos migrada a servidor de producción.

2.3.6. CONFIGURACIÓN IIS

Para la configuración de la aplicación web se configuró el servidor de aplicaciones web IIS. Una vez creada la aplicación web se configuró los enlaces a los que debe responder la aplicación web. La Figura 2.138 presenta la configuración de los enlaces del sistema.

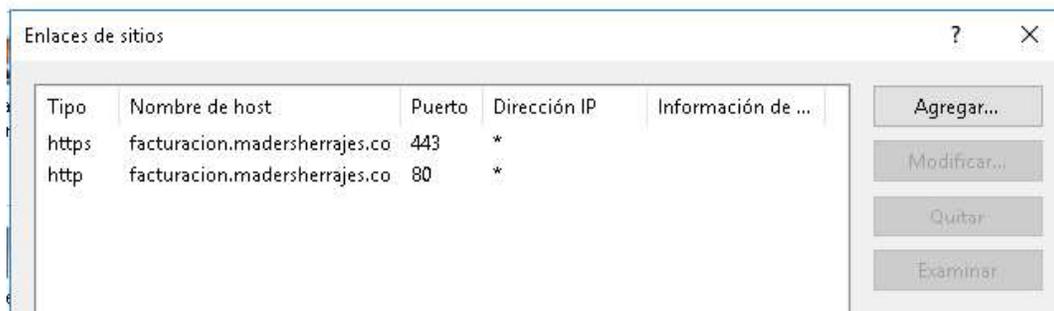


Figura 2.138 Configuración de enlaces en IIS.

Para la implementación de aplicaciones web a través de IIS puede realizarse a través de varios métodos los cuales son:

- Carga de archivos mediante FTP
- Carga de ficheros directamente al servidor
- Carga a través de Web Deploy

De los métodos posibles para el despliegue de aplicaciones web en IIS el más utilizado es el método que hace uso de Web Deploy debido a que permite un manejo sencillo y seguro de carga de ficheros a través de listas de control de acceso al servidor web.

La Figura 2.139 muestra la configuración realizada de Web Deploy para el despliegue e implementación del sistema en el servidor web.

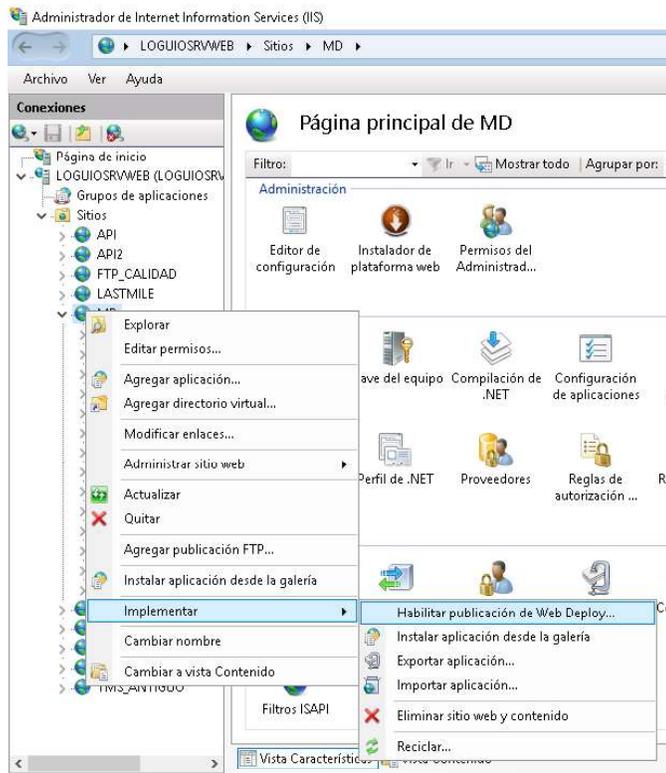


Figura 2.139 Habilitar publicación Web Deploy en IIS

Una vez habilitada la carga de archivos del sistema a través de Web Deploy se obtuvo el perfil de publicación válido para la implementación del sistema en el servidor web, este archivo se debe importar a través del Visual Studio al momento de la publicación del proyecto. La Figura 2.140 muestra el perfil de publicación generado para la aplicación web.

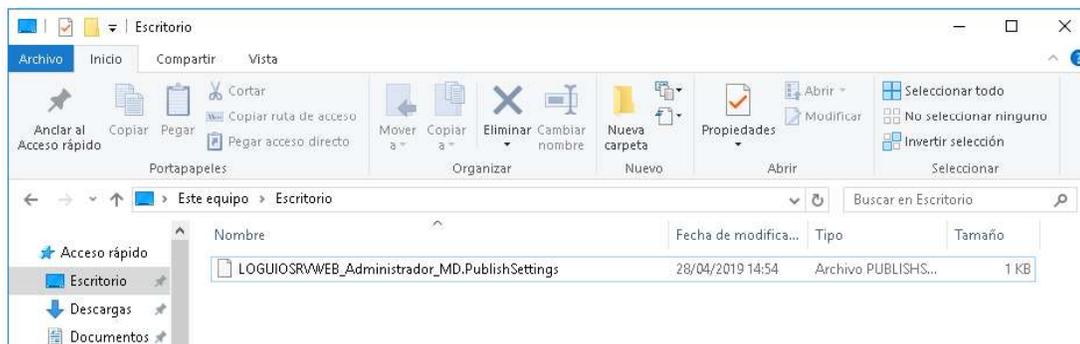


Figura 2.140 Perfil de publicación generado con herramienta Web Deploy.

2.3.7. DESPLIEGUE DEL SISTEMA EN EL SERVIDOR WEB

Para el despliegue del sistema desde el ambiente de depuración al servidor web se publicó el proyecto a través de Visual Studio para lo cual se ingresó al menú del proyecto web en la opción publicar tal como se indica en la Figura 2.141.

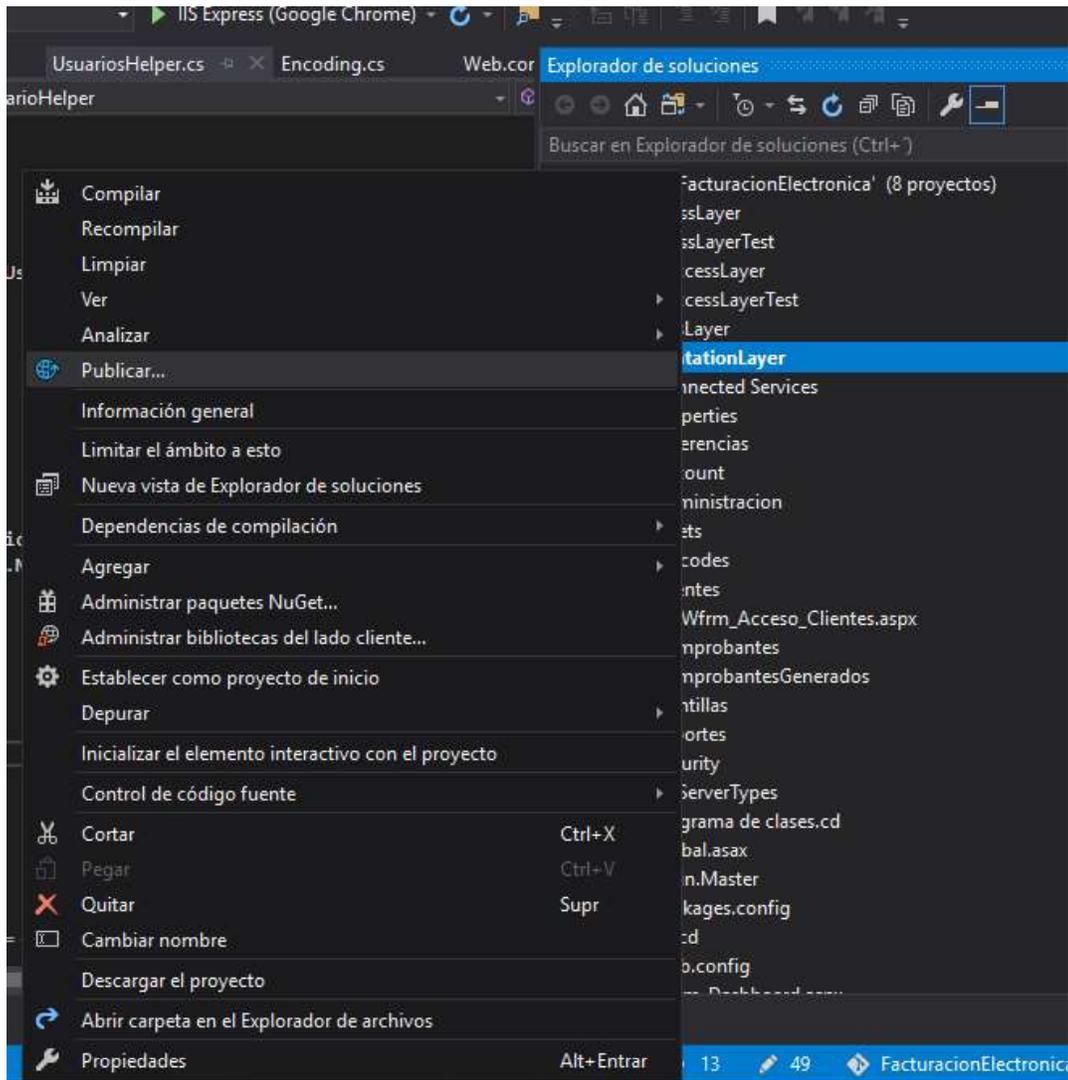


Figura 2.141 Menú de capa presentación.

En el menú de publicación se seleccionó el despliegue a través del método Web Deploy y se importó el perfil de publicación generado anteriormente. La Figura 2.142 muestra la configuración importada para la publicación del sistema en el servidor web con las credenciales del servidor.

Una vez configurado el perfil de publicación se procedió a publicar el proyecto. Resultado del proceso de publicación Visual Studio indica todos los logs²² a través del panel de salida tal como se muestra en la Figura 2.143.

Una vez finalizado el proceso de despliegue se pudo visualizar el sistema implementado en el servidor web a través de la URL <https://facturacion.madersherrajes.co> como se indica en la Figura 2.144.

²² **Logs:** Son todos los acontecimientos que afectan a un proceso particular.

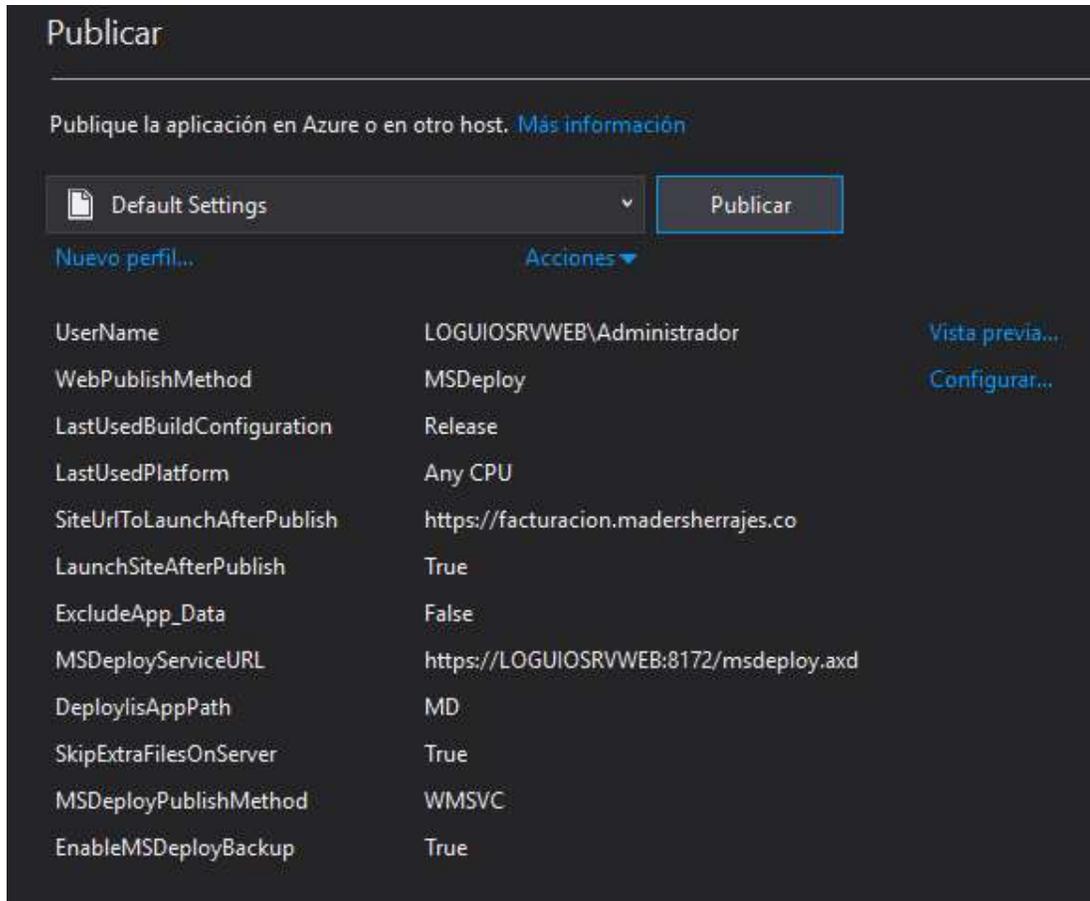


Figura 2.142 Perfil de publicación del proyecto para el servidor web

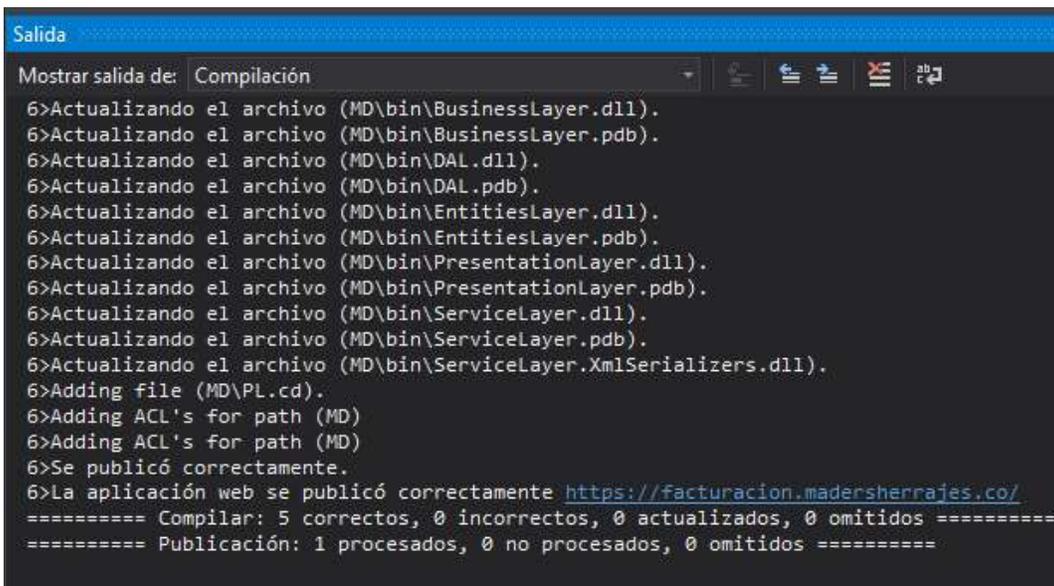


Figura 2.143 Salida del proceso de publicación del sistema en el servidor

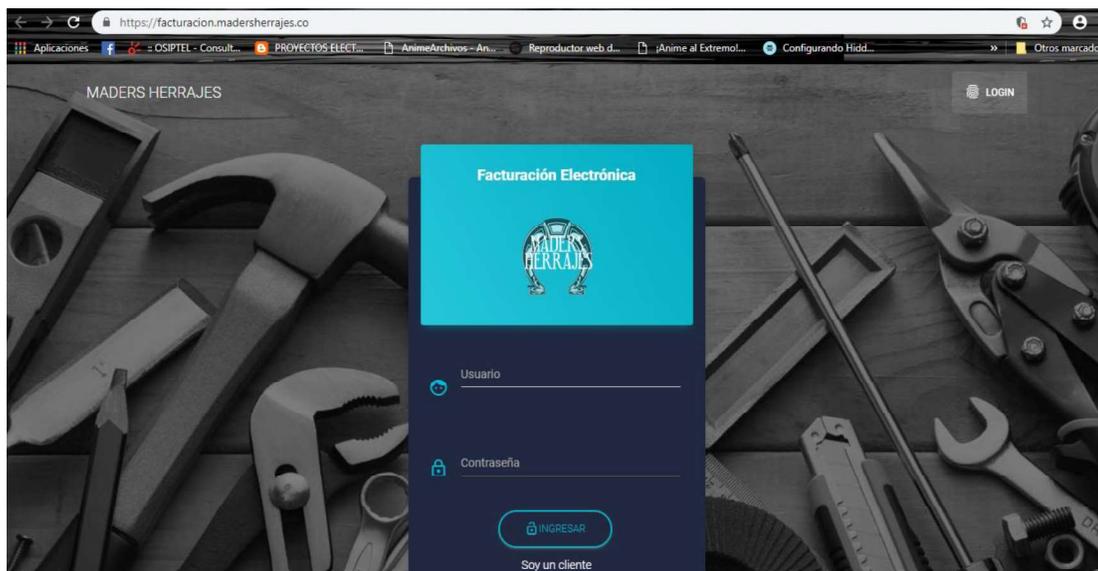


Figura 2.144 Sistema publicado en servidor web

3. RESULTADOS Y DISCUSIÓN

En este capítulo se indican los resultados más relevantes de las pruebas realizadas al sistema. Siguiendo los lineamientos de la metodología de desarrollo RAD se realizaron pruebas de cada prototipo después de su codificación. Posteriormente, finalizado el sistema prototipo se realizaron pruebas de integración de todo el sistema, y el comportamiento entre los prototipos de las diferentes capas.

3.1. CONTROL DE VERSIONES DE LA APLICACIÓN

Tal como se especificó en el capítulo uno del presente trabajo de titulación para el control de versiones de la aplicación web se utilizó GitHub.

En base a la metodología de desarrollo utilizada después de la codificación de los diferentes prototipos de las capas del sistema, se realizó un *commit* para registrar el progreso en el desarrollo del sistema. Así mismo se realizaron *commits* en la fase de pruebas y correcciones de cada prototipo del sistema.

La Figura 3.1 muestra el panel de administración de los diferentes *commits* realizados en el desarrollo del sistema.

La Figura 3.2 muestra los archivos agregados, actualizados o eliminados de los diferentes prototipos de las capas codificadas.

La Figura 3.3 presenta los cambios realizados en un componente específico del sistema, mostrando los cambios a realizados en el mismo marcando con color rojo las líneas de código eliminadas y con color verde las líneas de código agregadas.

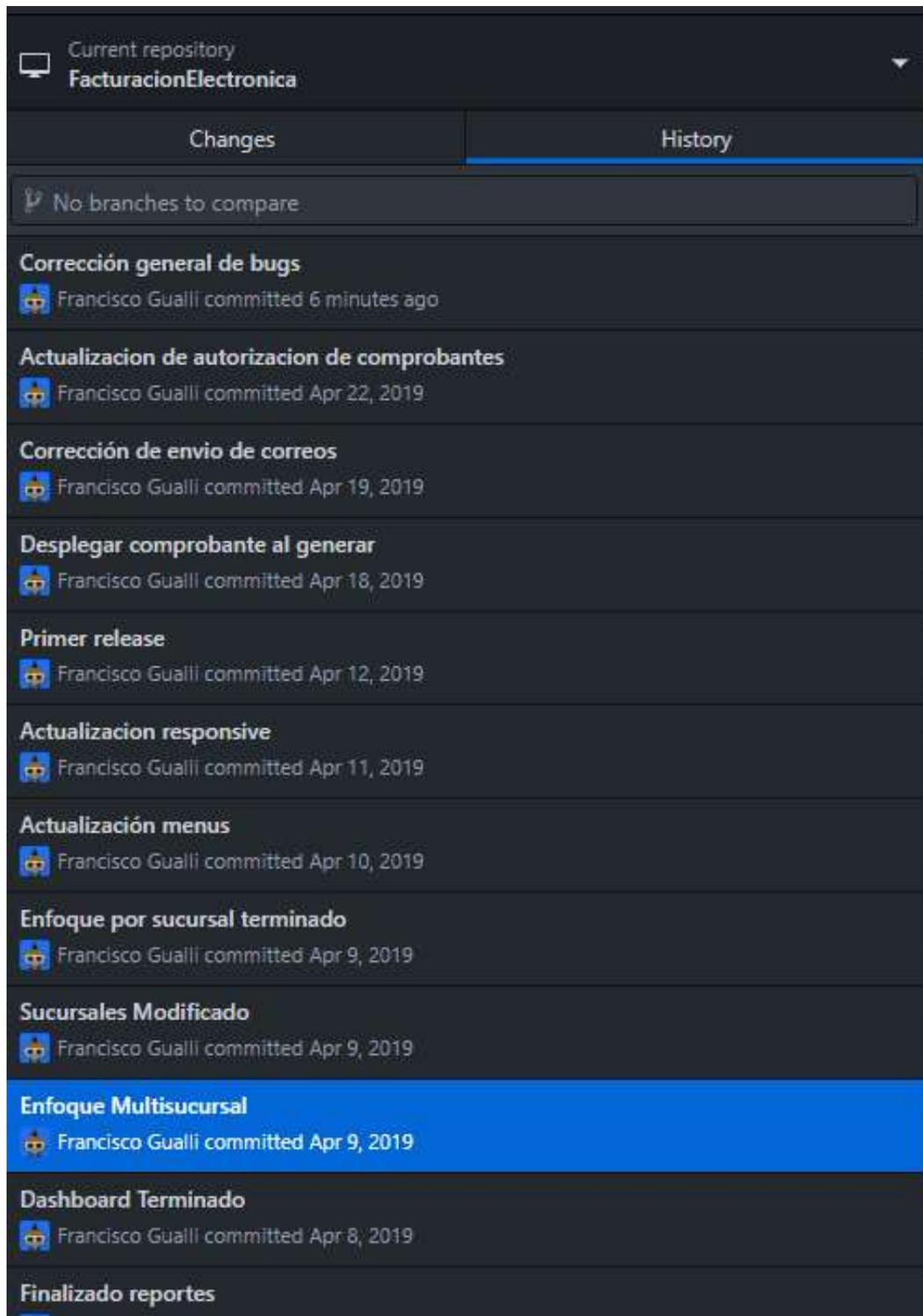


Figura 3.1 Panel de administración de control de versiones GitHub.

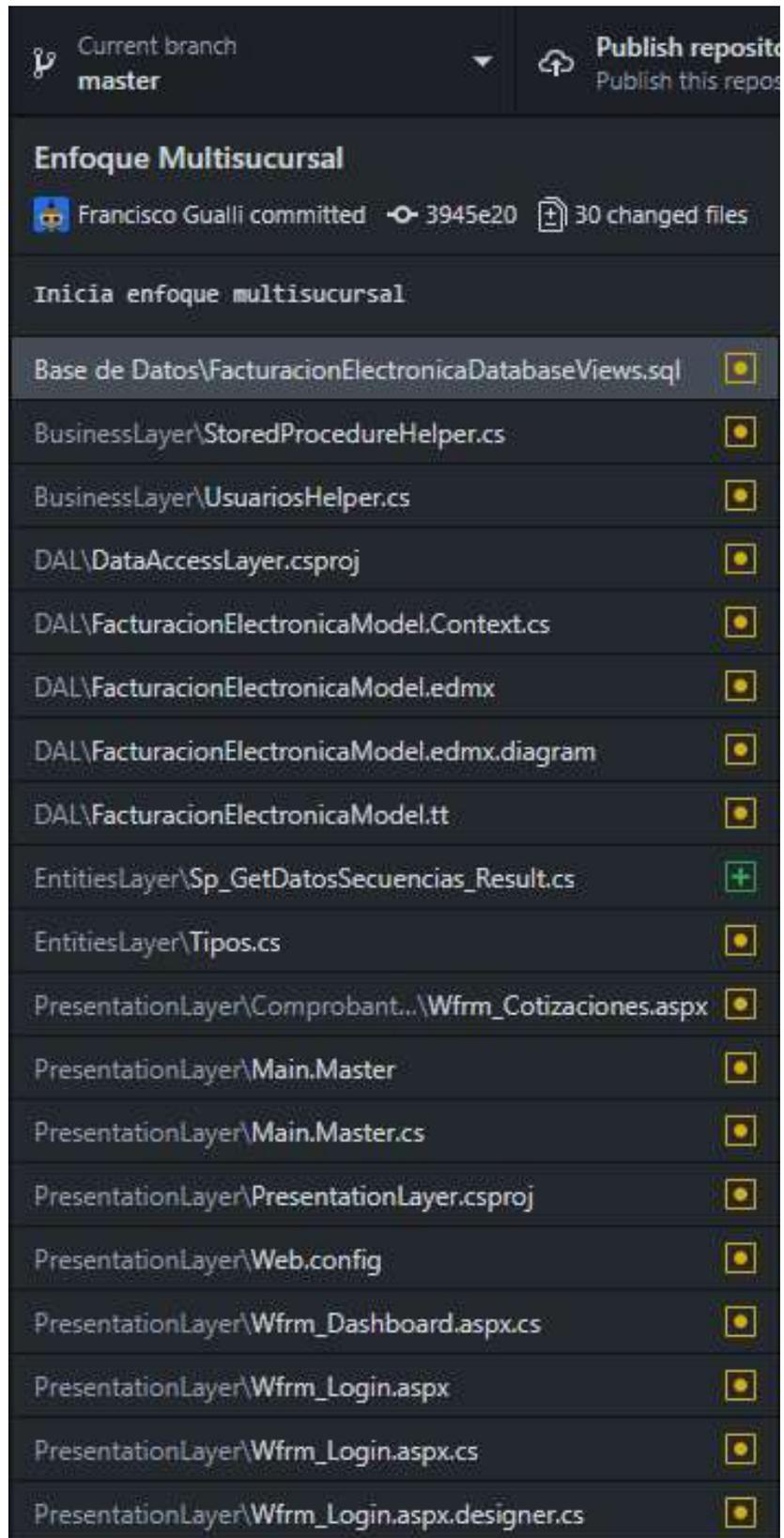


Figura 3.2 Panel de archivos actualizados en GitHub.

```

@@ -215,11 +215,19 @@ WHERE cl.Numero_Identificacion = @Identificacion_Cliente
215      215      GO
216      216
217      217      CREATE PROCEDURE Sp_GetDatosComprobantes
218      218      +@Periodo DATE
219      219      AS
220      220      BEGIN
221      221      -SELECT * FROM Vw_DatosEnvioComprobantes
222      222      -ORDER BY Id_Comprobante
223      223      -END
224      224      +SELECT v.Total, v.Tipo_Comprobante, v.Numero_Comprobantes, v.Periodo, s.Id_Sucursal, s.Nombre_Sucursal, s.Descripcion_Sucursal, s.Tipo_Sucursal FROM Vw_DatosComprobantes v
225      225      +INNER JOIN Sucursales s ON v.Id_Sucursal = s.Id_Sucursal
226      226      +WHERE v.Periodo = @Periodo
227      227      +END
228      228      +GO
229      229      +
230      230      +CREATE PROCEDURE Sp_GetDatosSecuencias
231      231      +AS
232      232      +BEGIN
233      233      +SELECT * FROM Vw_DatosSecuencias ORDER BY Secuencia DESC
234      234      +END
235      235      GO
236      236
237      237      -select * from Vw_DatosComprobantes where Periodo = '01/03/2019'

```

Figura 3.3 Panel de verificación de cambios realizados en archivos de GitHub.

3.2. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DATOS

Para la validación de la capa de datos se ejecutaron consultas a través del IDE Management Studio de SQL Server y se validó el correcto funcionamiento de vistas, procedimientos almacenados y *triggers*.

La Figura 3.4 muestra el registro de todas las transacciones en la tabla de auditoria Change_Log_Table auditoria

Fecha_Actualizacion	Nombre_Tabla	Command_Type	Nombre_Campo_1	Antiguo_Valor_Campo_1	Nuevo_Valor_Campo_1	Nombre_Campo_2	Antiguo_Valor_Campo_2	Nuevo_Valor_Campo_2	Nombre_Campo_3	Antiguo_Valor_Campo_3
2019-04-29 22:54:00	Cientes	Delete	Nombre_Razon_Social	Prueba de actualización	NULL	Numero_Identificacion	0987645321	NULL	Tipo_Identificacion	5
2019-04-29 22:50:00	Cientes	Update	Nombre_Razon_Social	Prueba de actualización	Prueba de actualización	Numero_Identificacion	0987645321	0987645321	Tipo_Identificacion	5
2019-04-29 22:46:00	Cientes	Insert	Nombre_Razon_Social	NULL	Prueba de ingreso	Numero_Identificacion	NULL	12134567890	Tipo_Identificacion	NULL

Figura 3.4 Ejemplo de tabla de auditoría.

3.3. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE ACCESO A DATOS

Las pruebas de funcionalidad del prototipo de la capa de acceso a datos se realizaron a través de la implementación de un proyecto de consola que consume directamente la capa de acceso a datos para determinar la funcionalidad de la capa.

Para las pruebas se realizó la prueba de cada uno de los métodos expuestos por la interfaz IRepository listados a continuación.

- Agregar Entidad

La Figura 3.5 presenta la clase utilizada para la prueba de funcionamiento del ingreso de entidades consumiendo directamente la capa de acceso a datos.

```
class Program
{
    static void Main(string[] args)
    {
        TestIngreso();
    }

    static void TestIngreso()
    {
        using (var r = RepositoryFactory.CreateRepository())
        {
            r.Crear<Cliente>(new Cliente
            {
                Numero_Identificacion = "12134567890",
                Tipo_Identificacion = (byte)Tipos_Identificacion.Cedula,
                Nombre_Razon_Social = "Prueba de ingreso",
                Direccion = "Prueba dirección",
                Fecha_Creacion = DateTime.Now,
                Estado = true
            });
        }
    }
}
```

Figura 3.5 Método para *testing* de ingreso de una entidad usando capa de acceso a datos

Tal como se puede ver en la Figura 3.6 se puede ver el correcto funcionamiento de la capa de acceso a datos al insertar la entidad en la base de datos.

	Id_Cliente	Nombre_Razon_Social	Tipo_Identificacion	Numero_Identificacion	Direccion	Fecha_Creacion	Estado
1	4008	Prueba de ingreso	5	12134567890	Prueba dirección	2019-04-29 22:46:00	1
2	2002	Francisco Gualli	5	1723267819	Zabala	2019-02-24 17:54:00	1
3	1000	Harold Guerra	5	3257932023	708 North First St.	2019-02-24 17:54:00	1
4	999	Summer Walker	4	1086468043430	902 East Second Parkway	2019-02-24 17:54:00	1

Figura 3.6 Entidad ingresada en la capa de datos.

- Actualizar Entidad

La Figura 3.7 presenta la clase utilizada para la actualización de una entidad a través de la capa de acceso a datos.

```

class Program
{
    static void Main(string[] args)
    {
        TestActualizar();
    }

    static void TestActualizar()
    {
        using (var r = RepositoryFactory.CreateRepository())
        {
            r.Actualizar<Cliente>(new Cliente
            {
                Id_Cliente = 4008,
                Numero_Identificacion = "0987645321",
                Tipo_Identificacion = (byte)Tipos_Identificacion.Cedula,
                Nombre_Razon_Social = "Prueba de actualización",
                Direccion = "Prueba dirección",
                Estado = false
            });
        }
    }
}

```

Figura 3.7 Método para *testing* de actualización de una entidad usando capa de acceso a datos.

Tal como se muestra en la Figura 3.8 se verificó el correcto funcionamiento de la capa de acceso a datos al actualizar una entidad de la base de datos.

	Id_Cliente	Nombre_Razon_Social	Tipo_Identificacion	Numero_Identificacion	Direccion	Fecha_Creacion	Estado
1	4008	Prueba de actualización	5	0987645321	Prueba dirección	2019-04-29 22:50:00	0
2	2002	Francisco Gualli	5	1723267819	Zabala	2019-02-24 17:54:00	1
3	1000	Harold Guerra	5	3257932023	708 North First St.	2019-02-24 17:54:00	1

Figura 3.8 Entidad actualizada en la capa de datos.

- Eliminar Entidad

La Figura 3.9 presenta la clase utilizada para realizar pruebas de funcionamiento para la eliminación de una entidad del sistema a través de la capa de acceso a datos.

```

class Program
{
    static void Main(string[] args)
    {
        TestEliminar();
    }

    static void TestEliminar()
    {
        using (var r = RepositoryFactory.CreateRepository())
        {
            r.Eliminar<Cliente>(new Cliente
            {
                Id_Cliente = 4008
            });
        }
    }
}

```

Figura 3.9 Método para *testing* de eliminación de una entidad usando capa de acceso a datos.

Tal como se puede ver en la Figura 3.10 se verificó el correcto funcionamiento de la capa de acceso a datos al eliminar una entidad de la base de datos.

	Id_Cliente	Nombre_Razon_Social	Tipo_Identificacion	Numero_Identificacion	Direccion	Fecha_Creacion	Estado
1	2002	Francisco Gualli	5	1723267819	Zabala	2019-02-24 17:54:00	1
2	1000	Harold Guerra	5	3257932023	708 North First St.	2019-02-24 17:54:00	1
3	999	Summer Walker	4	1086468043430	902 East Second Parkway	2019-02-24 17:54:00	1
4	998	Irene Buross	4	4913856233399	56 South Rockv Haque Parkwav	2019-02-24 17:54:00	1

Figura 3.10 Entidad eliminada de la capa de datos.

- Listar Entidades

La Figura 3.11 presenta la clase utilizada para realizar las pruebas de funcionamiento para listar las entidades almacenadas en el sistema a través de la capa de acceso a datos.

```

class Program
{
    static void Main(string[] args)
    {
        TestListar();
    }

    static void TestListar()
    {
        using (var r = RepositoryFactory.CreateRepository())
        {
            var listaClientes = r.Listar<Cliente>(null);
        }
    }
}

```

Figura 3.11 Método para *testing* de listado de entidades usando la capa de acceso a datos.

Tal como se puede ver en la Figura 3.12 se verificó el correcto funcionamiento de la capa de acceso a datos para listar un conjunto de entidades del sistema.

listaClientes	Count = 925
[0]	{EntitiesLayer.Cliente}
[1]	{EntitiesLayer.Cliente}
[2]	{EntitiesLayer.Cliente}
[3]	{EntitiesLayer.Cliente}
[4]	{EntitiesLayer.Cliente}
Cliente_Tipo_Cliente	Count = 0
Correos_Clientes	Count = 0
Cotizaciones	Count = 0
Detalle_Guia_Remision	Count = 0
Direccion	"13 Clarendon Freeway"
Estado	true
Facturas	Count = 0
Fecha_Creacion	{24/02/2019 17:54:00}
Id_Cliente	7
Nombre_Razon_Social	"Lisa Aguilar"
Notas_Credito	Count = 0
Notas_Debito	Count = 0
Numero_Identificacion	"4597106369556"
Retenciones	Count = 0
Telefonos_Clientes	Count = 0
Tipo_Identificacion	4
[5]	{EntitiesLayer.Cliente}
[6]	{EntitiesLayer.Cliente}
[7]	{EntitiesLayer.Cliente}
[8]	{EntitiesLayer.Cliente}
[9]	{EntitiesLayer.Cliente}

Figura 3.12 Lista de entidades devueltas por la capa de acceso a datos.

- Buscar Entidades

La Figura 3.14 presenta la clase utilizada para realizar las pruebas de funcionamiento de búsqueda de entidades a través de la capa de acceso a datos.

```

class Program
{
    static void Main(string[] args)
    {
        TestBuscar();
    }

    static void TestBuscar()
    {
        using (var r = RepositoryFactory.CreateRepository())
        {
            var listaEncontrados = r.Buscar<Cliente>(x => x.Tipo_Identificacion == (byte)Tipos_Identificacion.Cedula, null);
        }
    }
}

```

Figura 3.13 Método para *testing* de búsqueda de entidades usando la capa de acceso a datos.

Tal como se muestra en la Figura 3.14 se verificó el correcto funcionamiento de la capa de acceso a datos para la búsqueda de entidades en base a un criterio de filtrado.

listaEncontrados	Count = 251
[0]	{EntitiesLayer.Ciente}
[1]	{EntitiesLayer.Ciente}
[2]	{EntitiesLayer.Ciente}
[3]	{EntitiesLayer.Ciente}
Cliente_Tipo_Cliente	Count = 0
Correos_Clientes	Count = 0
Cotizaciones	Count = 0
Detalle_Guia_Remision	Count = 0
Direccion	"771 West Green Hague Parkway"
Estado	true
Facturas	Count = 0
Fecha_Creacion	{24/02/2019 17:54:00}
Id_Cliente	8
Nombre_Razon_Social	"Vicki Payne"
Notas_Credito	Count = 0
Notas_Debito	Count = 0
Numero_Identificacion	"1838136837"
Retenciones	Count = 0
Telefonos_Clientes	Count = 0
Tipo_Identificacion	5
[4]	{EntitiesLayer.Ciente}
[5]	{EntitiesLayer.Ciente}
[6]	{EntitiesLayer.Ciente}
[7]	{EntitiesLayer.Ciente}
[8]	{EntitiesLayer.Ciente}
[9]	{EntitiesLayer.Ciente}
[10]	{EntitiesLayer.Ciente}

Figura 3.14 Lista de entidades encontradas devuelta por la capa de acceso a datos.

- Ejecutar Procedimiento

La Figura 3.15 presenta la clase utilizada para realizar la prueba de funcionamiento para la ejecución de un procedimiento almacenado de la base de datos a través de la capa de acceso a datos.

```

class Program
{
    static void Main(string[] args)
    {
        TestProcedure();
    }

    static void TestProcedure()
    {
        using (var r = RepositoryFactory.CreateRepository())
        {
            List<SqlParameter> parametros = new List<SqlParameter>
            {
                new SqlParameter { ParameterName = "3077" }
            };
            var resultadoProcedimiento = r.EjecutarProcedimiento<Sp_ObtenerDatosFactura_Result>("Sp_ObtenerDatosFactura", parametros);
        }
    }
}

```

Figura 3.15 Método para *testing* de ejecución de un procedimiento almacenado usando la capa de acceso a datos.

Tal como se muestra en la Figura 3.16 se verificó el correcto funcionamiento para la ejecución de un procedimiento almacenado en la base de datos mediante la capa de acceso a datos devolviendo el resultado de la ejecución.

resultadoProcedimiento		Count = 3
▶ [0]		{EntitiesLayer.Sp_ObtenerDatosFactura_Result}
▶ [1]		{EntitiesLayer.Sp_ObtenerDatosFactura_Result}
▶ [2]		{EntitiesLayer.Sp_ObtenerDatosFactura_Result}
Cantidad		2
Clave_Acceso		"2204201901172286819500110010030000000031234567810"
Codigo_Principal		"44171"
Contribuyente_Especial		false
Correo_Cliente		"aeaokwr5@Indodt.com, porhdc.zdyg@-tkcin.org"
Descuento		0.00
Direccion		"766 Cowley St."
Direccion_Establecimiento		"El condado"
Direccion_Matriz		"Av del maestro y Av de la Prensa"
Estado_Comprobante		2
Fecha_Emision		{22/04/2019 0:00:00}
Guia_Remision		null
IRBPNR		0.00
IVA		0.00
Llevar_Contabilidad		false
Nombre_Producto		"CADENA PLASTICA BLANCA FIERON 5 MM"
Nombre_Razon_Social		"Faith Juarez"
Nro_Comprobante		"001-003-000000003"
Numero_Autorizacion		"2204201901100100300000000317228681959"
Numero_Identificacion		"00549560433"
Observacion		"OTROS CON UTILIZACION DEL SISTEMA FINANCIERO"
Plazo		0
Precio_Total		148.40
Precio_Unitario		74.20

Figura 3.16 Resultado de la ejecución del procedimiento almacenado devuelto por la capa de acceso a datos.

3.4. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE NEGOCIO

Para las pruebas de funcionalidad de la capa de negocio se utilizó un proyecto de pruebas unitarias (.NET Framework). La Figura 3.17 muestra la implementación de una clase de prueba para realizar pruebas unitarias sobre los componentes de la clase.

```
[TestClass]
public class ClientesTest
{
    [TestMethod]
    public void AgregarClienteTest()
    {
        ClientesHelper clientesHelper = new ClientesHelper();
        Cliente cliente = clientesHelper.AgregarCliente
        (
            "Prueba BL",
            Tipos_Identificacion.Cedula,
            "1234567890",
            "Dirección prueba",
            new List<Telefonos_Cliente>(),
            new List<Correos_Cliente>(),
            new List<Cliente_Tipo_Cliente>()
        );
        Assert.IsNotNull(cliente);
    }
}
```

Figura 3.17 Implementación clase de prueba

El resultado de las pruebas unitarias de la capa de negocio se puede visualizar a través del explorador de pruebas tal como se muestra en la Figura 3.18

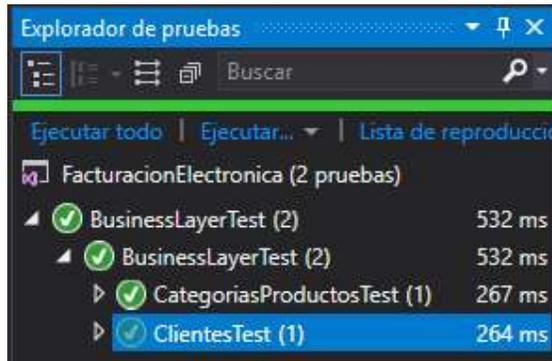


Figura 3.18 Explorador de pruebas de Visual Studio

3.5. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE SERVICIO

Para las pruebas del prototipo de la capa de servicio se utilizó una aplicación de consola (.NET Framework). La capa de servicio expone dos principales funcionalidades:

- Envío de comprobantes

La Figura 3.19 presenta la clase implementada para la realización de pruebas de funcionamiento del envío de comprobantes electrónicos haciendo uso de la capa de servicio para lo cual se debe enviar como parámetros el arreglo de bytes del comprobante generado, el tipo de ambiente del SRI y el tiempo de espera para el servicio web en segundos.

```

class Program
{
    static void Main(string[] args)
    {
        TestEnvio();
    }

    private static void TestEnvio()
    {
        string path = $"1.xml";
        var bytes = File.ReadAllBytes(path);
        EnvioComprobantes autorizacion = new EnvioComprobantes();
        var respuesta = autorizacion.EnvioComprobanteWS(bytes, EntitiesLayer.Tipos_Ambientes.Pruebas, 5);
    }
}

```

Figura 3.19 Método para el *testing* del envío de comprobantes usando la capa de servicio.

Tal como se indica en la Figura 3.20 la capa de presentación realiza la llamada al servicio web y devuelve la respuesta de la solicitud.

respuesta	{EntitiesLayer.EnvioComprobanteResponse}
ComprobanteWS	{EntitiesLayer.ComprobanteWS}
ClaveAcceso	"0503201901179235684900110010010000000091234567811"
Mensaje	{EntitiesLayer.Mensaje}
Identificador	43
InformacionAdicional	null
Message	"CLAVE ACCESO REGISTRADA"
Tipo	"ERROR"
Estado	"DEVUELTA"

Figura 3.20 Respuesta de llamada al servicio web devuelta por la capa de servicio.

- Verificación de comprobantes

La Figura 3.21 presenta la clase implementada para la realización de pruebas de funcionamiento para la verificación de comprobantes haciendo uso de la capa de servicio para lo cual se debe enviar como argumento la clave de acceso del comprobante y el tipo de ambiente del SRI.

```

class Program
{
    static void Main(string[] args)
    {
        TestVerificacion();
    }

    private static void TestVerificacion()
    {
        string clave = "2204201901172286819500110010030000000051234567811";
        VerificaComprobantes verificacion = new VerificaComprobantes();
        var respuesta = verificacion.VerificarComprobanteWS(clave, EntitiesLayer.Tipos_Ambientes.Pruebas);
    }
}

```

Figura 3.21 Método para el testing de la verificación de un comprobante usando la capa de servicio.

Tal como se muestra en la Figura 3.22 se verificó el correcto funcionamiento de la capa de servicio para la verificación de comprobantes ya que la plataforma del SRI devolvió el resultado AUTORIZADO para el comprobante consultado.

respuesta	{EntitiesLayer.VerificarComprobanteResponse}
Autorizacion	{EntitiesLayer.Autorizacion}
Comprobante	{Element, Name="comprobante"}
Estado	"AUTORIZADO"
FechaAutorizacion	{22/04/2019 9:46:10}
Date	{22/04/2019 0:00:00}
Day	22
DayOfWeek	Monday
DayOfYear	112
Hour	9
InternalKind	9223372036854775808
InternalTicks	636915231700000000
Kind	Local
Millisecond	0
Minute	46
Month	4
Second	10
Ticks	636915231700000000
TimeOfDay	{09:46:10}
Year	2019
dateData	9860287268554775808
Miembros estáticos	
Mensaje	"62IDENTIFICACION DEL RECEPTOR MAL CONFORMADAEl número de cédula 1920447481"
NumeroAutorizacion	"2204201901100100300000000517228681958"
TipoAmbiente	Pruebas

Figura 3.22 Resultado verificación del comprobante con el servicio web usando la capa de servicio.

3.6. PRUEBAS DE FUNCIONALIDAD DE LA CAPA DE PRESENTACIÓN

Para las pruebas correspondientes a la capa de presentación se realizaron pruebas de cada módulo definido para el sistema.

3.6.4. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE AUTENTICACIÓN

La autenticación en el sistema se realiza mediante la verificación del nombre de usuario y contraseña en caso de que el rol sea usuario y con el número de identificación en caso de que el rol sea cliente.

En la primera prueba se intentó ingresar al sistema sin ingresar las credenciales de autenticación, por lo que el sistema arrojó un mensaje indicando el ingreso de las credenciales.

La Figura 3.23 presenta la prueba de intento de acceso al sistema sin el ingreso de credenciales, en donde el sistema mostró al usuario que es obligatorio el ingreso de las credenciales para la validación pertinente.

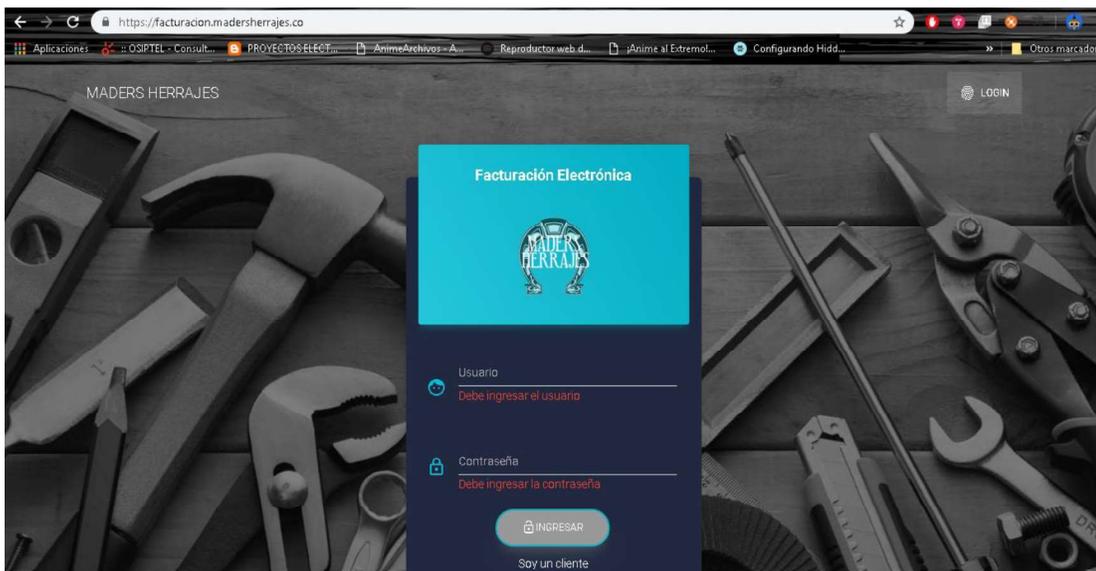


Figura 3.23 Intento de autenticación sin credenciales

Como una segunda prueba se ingresó caracteres especiales en el nombre de usuario tal como se muestra en la Figura 3.24, en donde se puede ver que el sistema notifica al usuario que no es posible el ingreso de caracteres especiales para el nombre de usuario.

Así mismo, se realizó las pruebas de ingreso mencionadas con el ingreso de tipo cliente tal como se muestra en la Figura 3.25, en donde se puede ver que el sistema notifica al usuario que el ingreso de la identificación es obligatorio para la validación correspondiente.

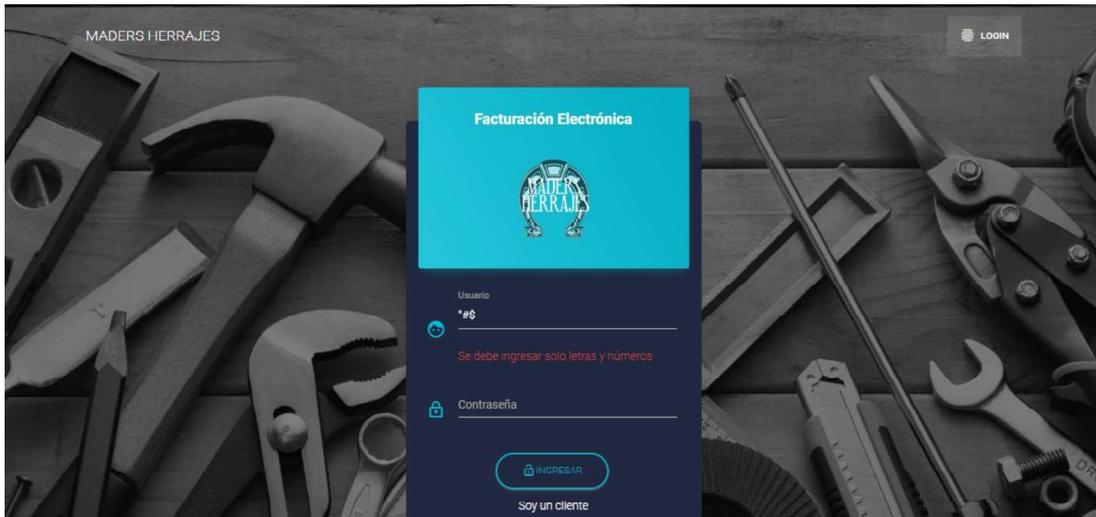


Figura 3.24 Ingreso de caracteres especiales



Figura 3.25 Intento de ingreso sin credenciales como cliente

Posteriormente, se intentó la autenticación con credenciales incorrectas. La Figura 3.26 ilustra el resultado de la autenticación fallida.

Al ingresar con un perfil de tipo administrador el sistema permite elegir el tipo de ingreso que se intenta realizar tal como se muestra en la Figura 3.27.

Cabe recalcar que en base al rol del usuario autenticado se visualizará el acceso a los diferentes módulos del sistema. La Figura 3.28 muestra la pantalla principal del usuario con rol administrador al ingresar al sistema.

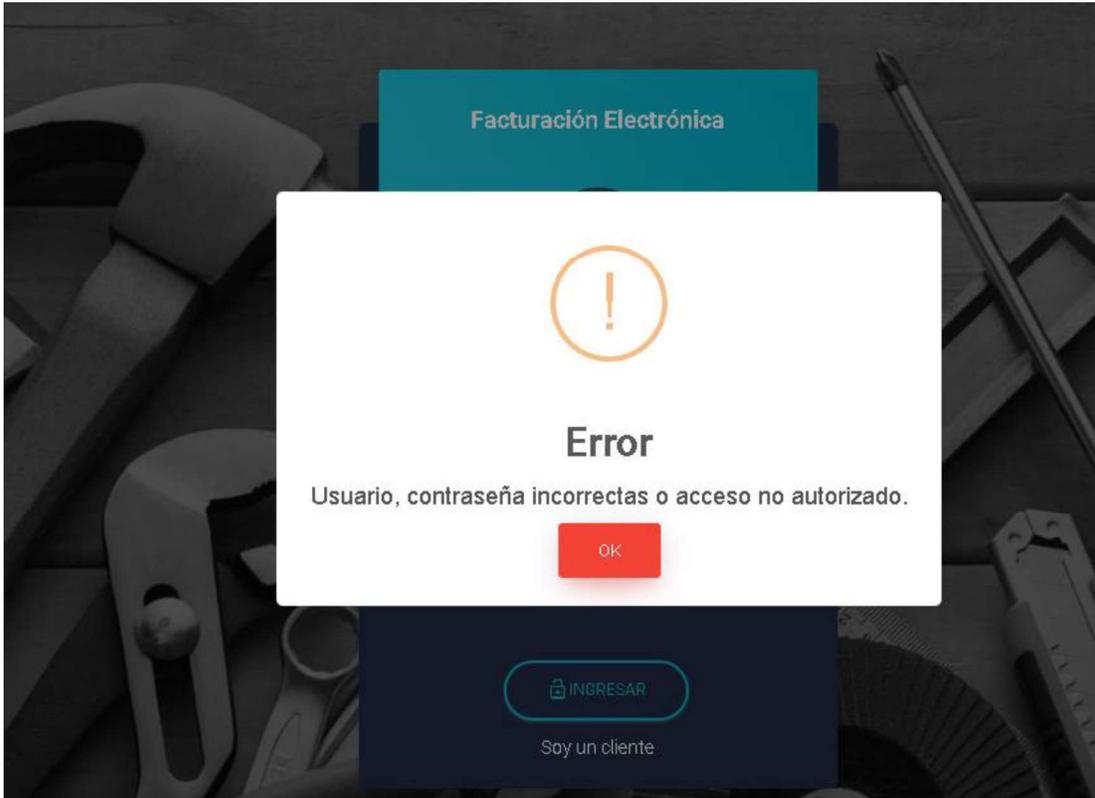


Figura 3.26 Intento de autenticación con credenciales incorrectas.

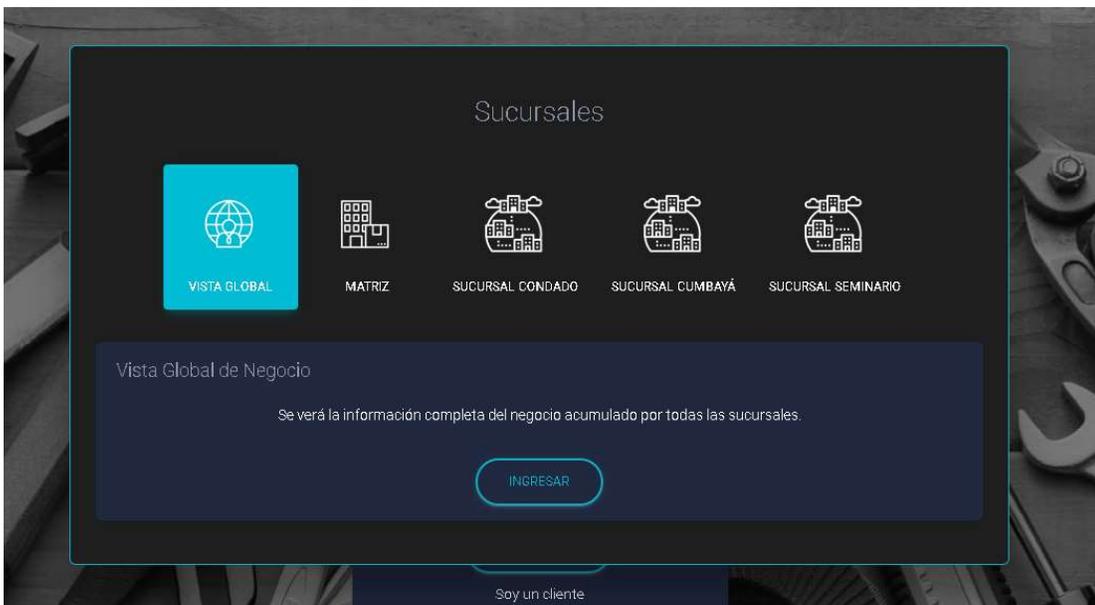


Figura 3.27 Panel de selección de sucursal.

Como se puede ver en la Figura 3.29 el usuario con rol administrador de sucursal tiene acceso limitado a la sucursal a la cual se encuentra asignado.

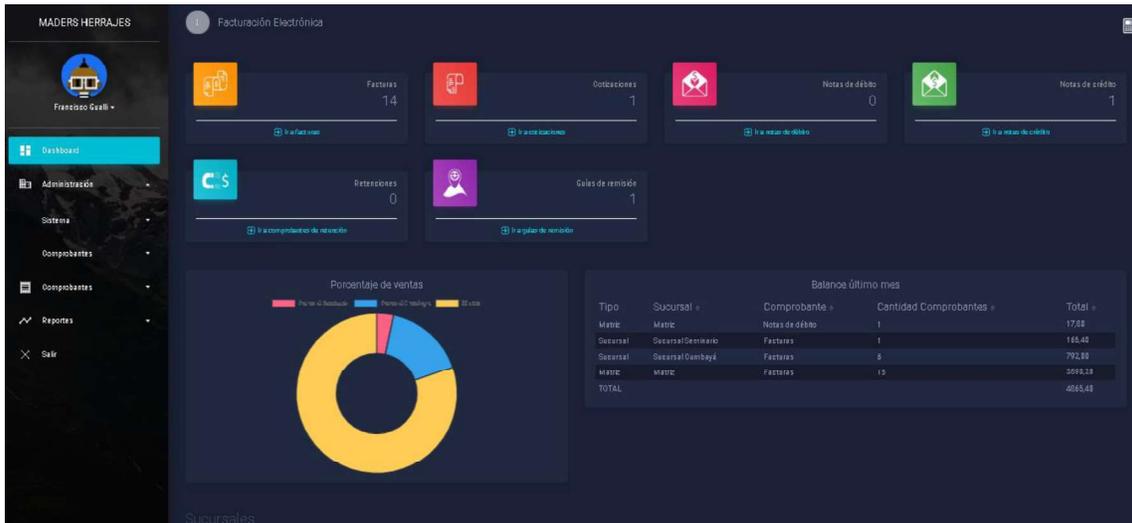


Figura 3.28 Ingreso como administrador.

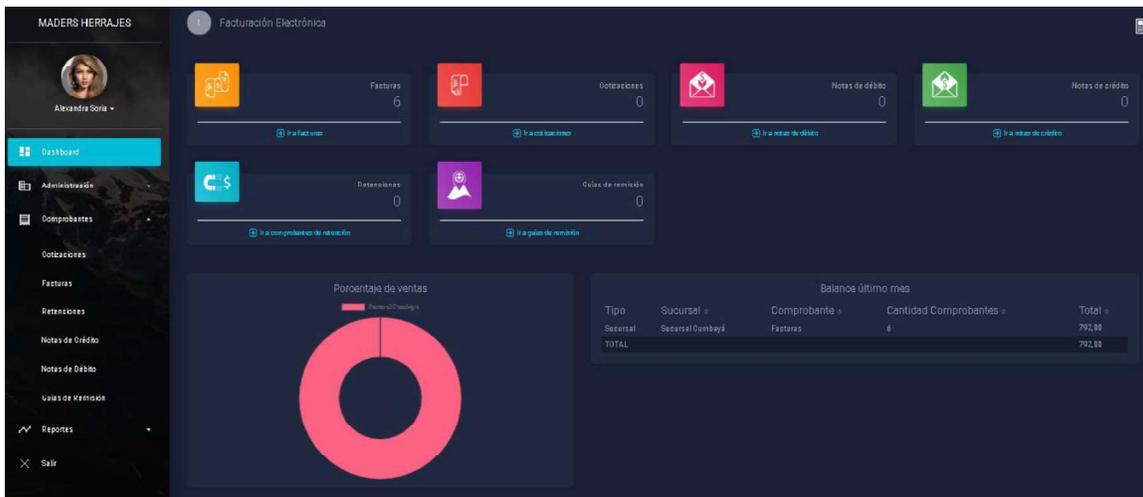


Figura 3.29 Ingreso como administrador de sucursal.

Tal como se muestra en la Figura 3.30 el usuario Juan Lopez tiene el rol Usuario por lo que está limitado únicamente a administrar clientes, transportistas, productos y generar cotizaciones, facturas y guías de remisión.

Tal y como se presenta en la Figura 3.31 el usuario tiene acceso únicamente al panel de búsqueda de comprobantes electrónicos.

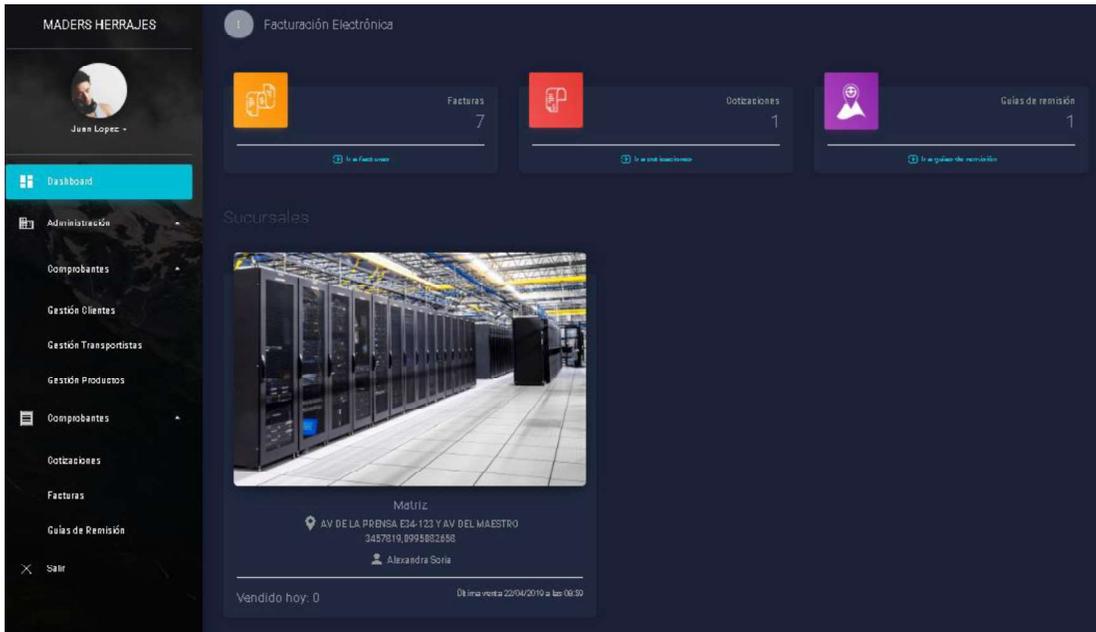


Figura 3.30 Ingreso como usuario

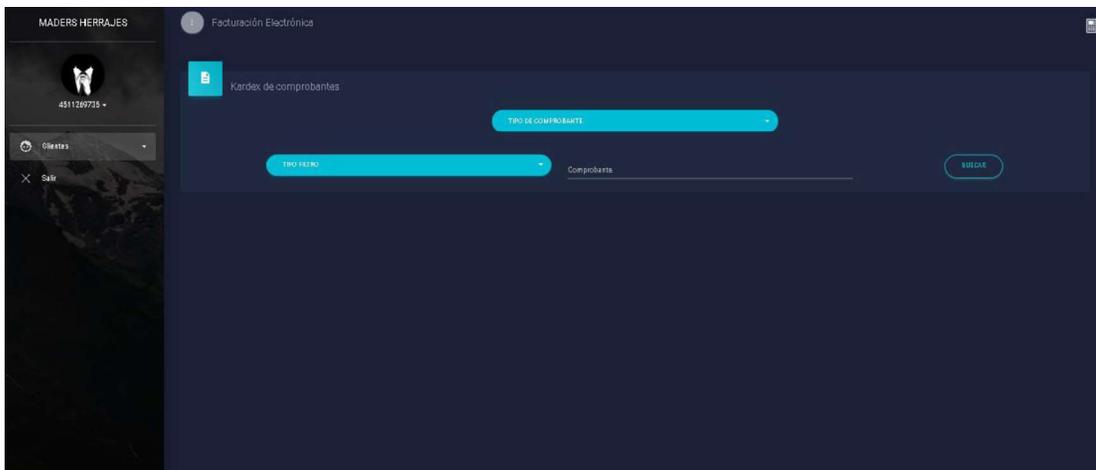


Figura 3.31 Ingreso como cliente

3.6.5. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE ADMINISTRACIÓN

Para realizar las pruebas de funcionalidad del módulo de administración se tomó en cuenta el ingreso y actualización de información

Como primera prueba se intentó registrar un usuario sin ningún campo ingresado con lo que el sistema validó que los campos obligatorios sean ingresados tal como se muestra se la Figura 3.32.

The screenshot shows a web form titled "Configuración de Usuario" with a close button (X) in the top right corner. Under the heading "Datos de usuario", there is a dropdown menu set to "ACTIVO". Below this, there are two red error messages: "Debe ingresar el nombre" and "Debe ingresar el usuario". The form includes several input fields: a dropdown for "SUCURSAL" (set to "SUCURSAL"), a text field for "Nombre", a dropdown for "TIPO USUARIO" (set to "TIPO USUARIO"), a text field for "Usuario", a text field for "Contraseña", a text field for "Teléfono", and a text field for "E-mail". At the bottom right, there are two buttons: "CERRAR" (grey) and "AGREGAR" (blue).

Figura 3.32 Intento de ingreso de nuevo usuario con campos vacíos.

A continuación, tal como se muestra en la Figura 3.33 se ingresó los datos para un nuevo usuario.

The screenshot shows the same "Configuración de Usuario" form, but now with data entered. The "ACTIVO" dropdown remains. The "SUCURSAL" dropdown is now set to "SUCURSAL CUMBAYÁ". The "Nombre" field contains "Martin Flores". The "TIPO USUARIO" dropdown is set to "USUARIO". The "Usuario" field contains "mflores" and the "Contraseña" field contains "****". The "Teléfono" field contains "3453454" and the "E-mail" field contains "mflores@yahoo.com". The "CERRAR" and "AGREGAR" buttons are still present at the bottom right.

Figura 3.33 Ingreso de nuevo usuario.

Tal como se muestra en la Figura 3.34 se agregó el usuario correctamente al sistema, se muestra el mensaje de confirmación y se muestra el mismo en la lista de usuarios.



Figura 3.34 Mensaje de confirmación de registro de nuevo usuario.

Posteriormente, se intentó ingresar al sistema con las credenciales del nuevo usuario ingresado tal como se muestra en la Figura 3.35.

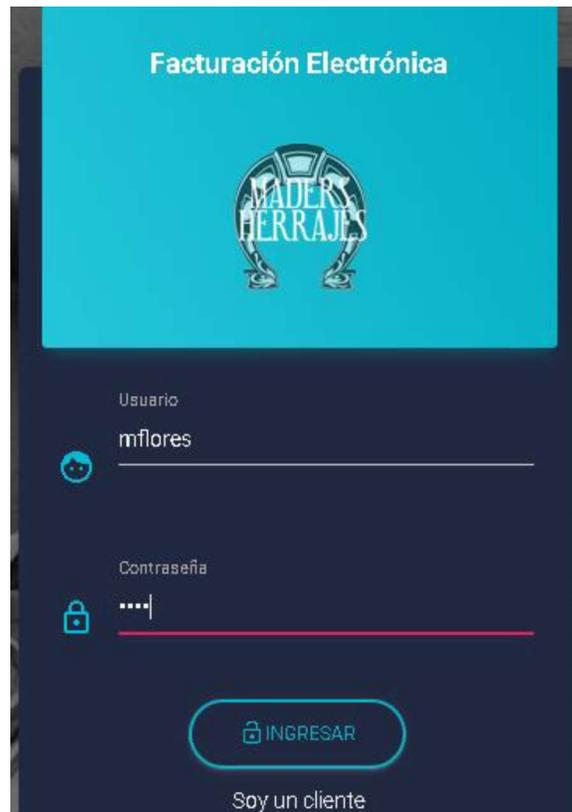


Figura 3.35 Intento de ingreso con credenciales de nuevo usuario.

Tal como se muestra en la Figura 3.36 el usuario mfloros registrado anteriormente tiene el acceso a la sucursal asignada y permisos con rol de usuario.

Después de la validación del usuario registrado se procedió a actualizar los datos del usuario tal como se muestra en la Figura 3.37.

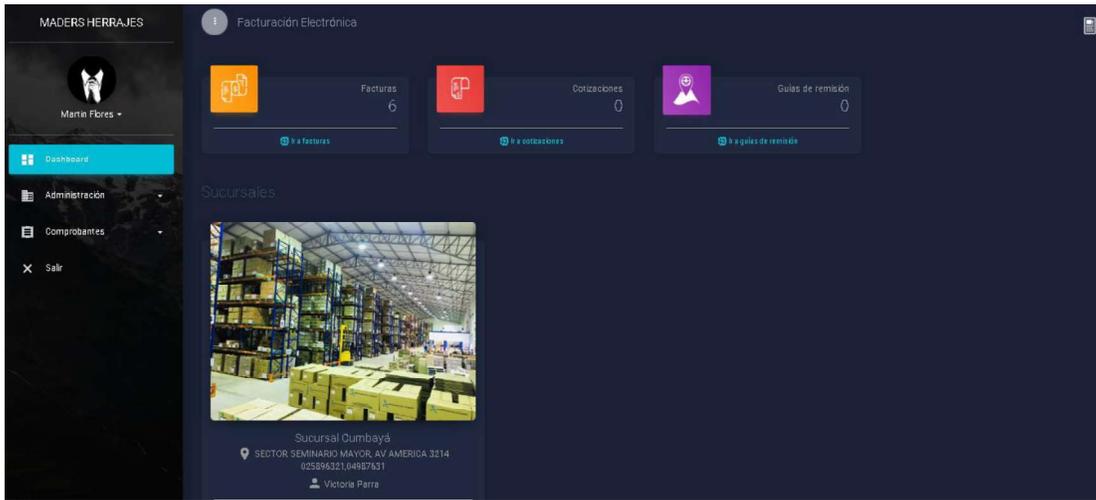


Figura 3.36 Ingreso al sistema con nuevo usuario.

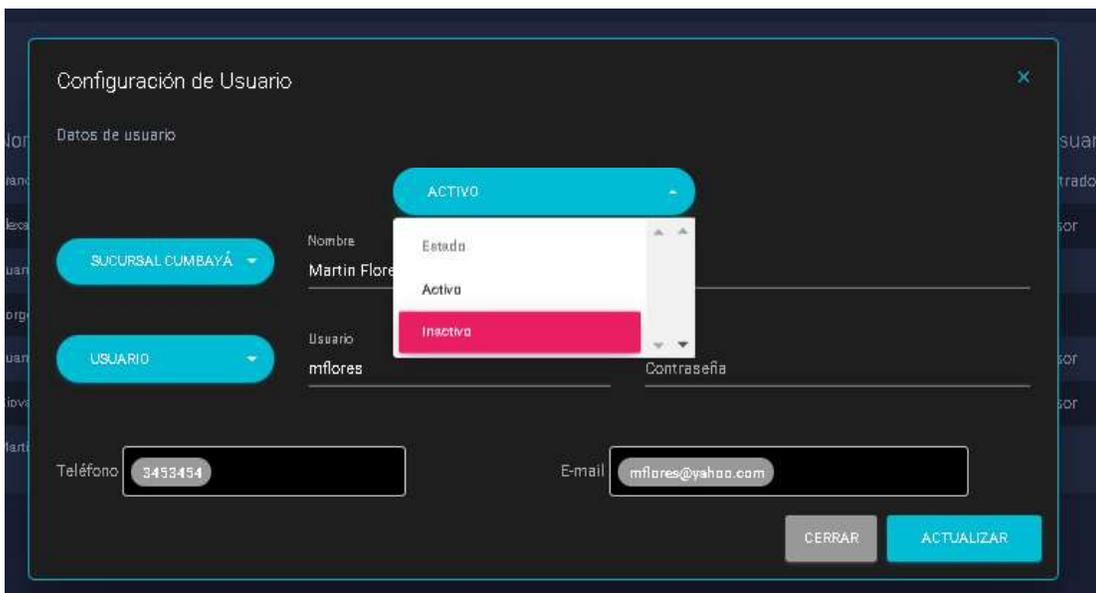


Figura 3.37 Actualización de información de usuario.

Así mismo se pudo verificar la actualización de los datos del usuario en el panel principal de usuarios tal como se muestra en la Figura 3.38.

Finalmente, tal como se muestra en la Figura 3.39 se intentó ingresar al sistema con el usuario mflores en estado inactivo por lo cual el sistema devolvió el mensaje de error mostrado en la Figura 3.40.

Sucursal	Nombre	Usuario	E-mail	Teléfono	Estado	Tipo Usuario	
Matriz	Francisco Guallí	francixt	francixt@live.com,fguallí@logisticas.com.ec	3381 032,0999082658	Activo	Administrador	
Sucursal Cumbayá	Alexandra Soira	alexa	alexa@hotmail.com	099236541	Activo	Supervisor	
Matriz	Juan Lopez	jlopez	julopez1331@hotmail.com	0998201 020	Activo	Usuario	
Sucursal Seminario	Jorge Luis Zapata	jzapata	jzapata@hotmail.com	3381 032	Activo	Usuario	
Sucursal Condado	Juan Carlos Díaz	jdiaz	as.d@hotmail.com	3381 032	Activo	Supervisor	
Sucursal Seminario	Giovanny Salazar	gsalazar			Activo	Supervisor	
Sucursal Cumbayá	Martin Flores	mflores	mflores@yahoo.com	3452454	Inactivo	Usuario	

Figura 3.38 Lista de usuarios registrados en el sistema

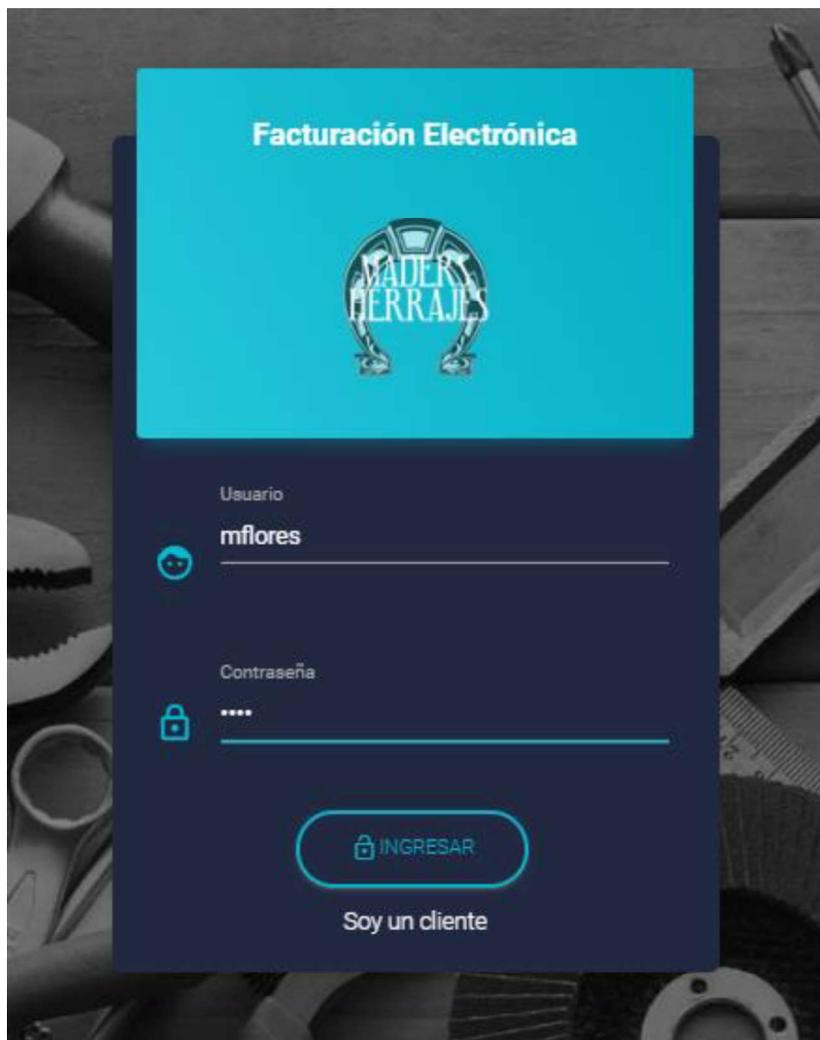


Figura 3.39 Intento de autenticación con credenciales de usuario inactivo.

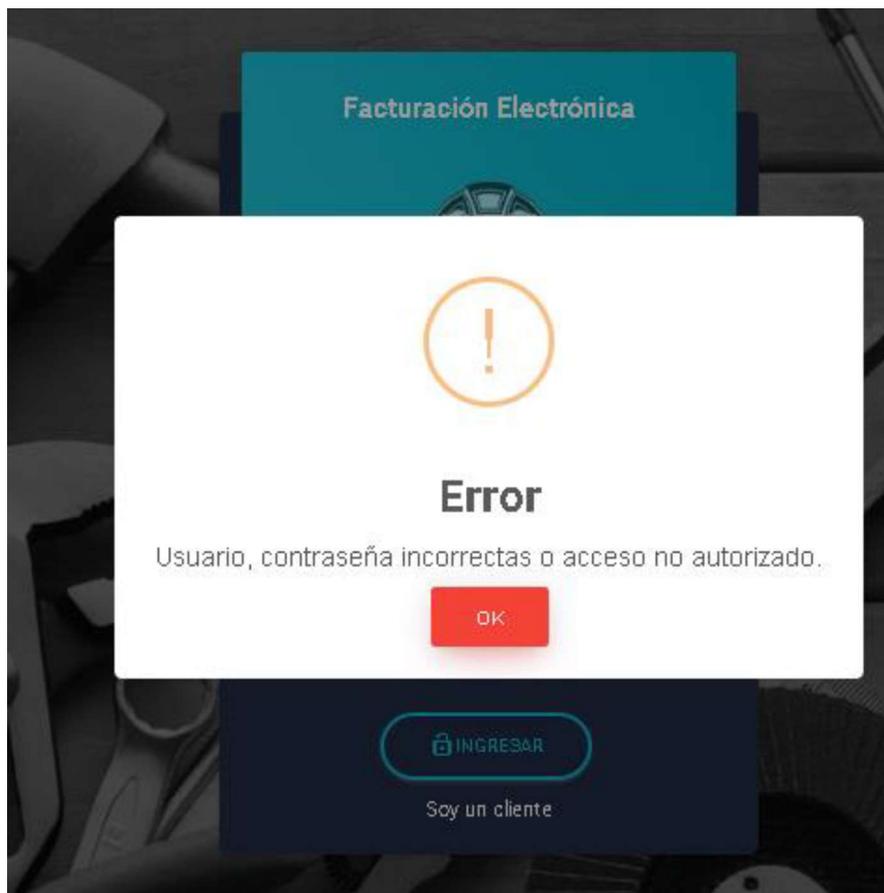


Figura 3.40 Mensaje de error en inicio de sesión.

3.6.6. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE COMPROBANTES

Para realizar las pruebas de funcionalidad del módulo de comprobantes se tomó en cuenta la generación y visualización de comprobantes, así mismo la sincronización con el SRI a través de los servicios web.

Como primera prueba se generó un nuevo comprobante y se validó la correcta generación del mismo en el sistema tal y como se muestra en la Figura 3.41.

Tal como se muestra en la Figura 3.42 el comprobante Nro. 001-004-000000009 se generó correctamente y su estado es autorizado.

Así mismo, se verificó el estado del mismo en la plataforma del SRI a través de los servicios web y los mismos devolvieron el mensaje autorizado tal como se muestra en la Figura 3.43.

A continuación, se visualizó el comprobante y se verificó los datos tal como se muestra en la Figura 3.44.

Mostrar Facturas Generadas
▼

+
Nueva Factura

DATOS DEL EMISOR

<small>RUC</small> 1722868195001	<small>Razón Social</small> ALEXANDRA CARINA SORIA ALVAREZ	<small>Nombre Comercial</small> MADERS HERRAJES
<small>Dirección Establecimiento</small> El condado	<small>Dirección Matriz</small> Av del maestro y Av de la Prensa	
<small>Código Establecimiento</small> 001	<small>Código Punto Emisión</small> 004	<small>Resolución</small>
		<small>Obligación Contabilidad</small> Contribuyente Especial <input type="checkbox"/>

DATOS DEL COMPROBANTE

<small>Número de Comprobante</small> 001-004-00000008	<small>Clave de Acceso</small> 0505201901172286819500110010040000000081234567819
<small>Fecha de emisión</small> 05-05-2019	<small>Guía de remisión</small>

DATOS DEL COMPRADOR

Consumidor Final
 Datos

Tenga en cuenta que facturas superiores a \$200.00 no pueden ser como Consumidor Final

<small>Identificación</small> 99999999999999	<small>Razón Social</small> CONSUMIDOR FINAL	<small>Correo Electrónico</small>
<small>Dirección</small>		

BUSCAR CLIENTE

DETALLE DE LA FACTURA

Cantidad	Código	Producto	Precio	Descuento	Iva	Ice	IRBPNR	Total	Eliminar
1	L1710	CARTUCHO P/ESTUFA LINMEX 710	7,00	0,00	0	0,00	0,00	7,00	✕
1	0889	CUCHILLO 0889 LION TOOLS PAVONADO RUB	22,20	0,00	0	0,00	0,00	22,20	✕

BUSCAR PRODUCTO

FORMA DE PAGO

<small>Subtotal Sin Impuestos</small> 29.20	<small>Subtotal 12</small> 0.00	<small>Subtotal 0</small> 29.20	<small>IVA 12%</small> 0.00	<small>Total ICE</small> 0.00	<small>Total IRBPNR</small> 0.00	<small>Total Descuento</small> 0.00	<small>Propina</small> 0.00	<small>Total</small> 29.20
		<small>Forma de Pago</small>	<small>Valor</small>	<small>Plazo</small>	<small>Tiempo</small>	<small>Eliminar</small>		
		SIN UTILIZACION DEL SISTEMA FINANCIERO	29,20	0	Ninguno	✕		

BUSCAR FORMA DE PAGO

GENERAR FACTURA
CANCELAR

Figura 3.41 Generación de un nuevo comprobante electrónico.

Matriz	001-004-00000005	18/04/2019	Autorizado	Erik Chaney	1146,80	137,62	1284,42		
Matriz	001-004-00000006	18/04/2019	Autorizado	Natasha Farmer	52,60	0,00	52,60		
Sucursal Cumbayá	001-003-00000001	19/04/2019	Autorizado	Summer Walker	52,60	0,00	52,60		
Matriz	001-004-00000007	22/04/2019	Autorizado	Kelvin Hoffman	44,20	0,00	44,20		
Sucursal Cumbayá	001-003-00000002	22/04/2019	Autorizado	Roberta Hurley	210,00	2,04	212,04		
Sucursal Cumbayá	001-003-00000003	22/04/2019	Autorizado	Faith Juarez	221,00	0,00	221,00		
Sucursal Cumbayá	001-003-00000004	22/04/2019	Autorizado	Consumidor Final	10,40	0,00	10,40		
Sucursal Cumbayá	001-003-00000005	22/04/2019	Autorizado	Karen Jimenez	188,00	2,69	190,69		
Sucursal Cumbayá	001-003-00000006	22/04/2019	En procesamiento	Joan Levy	110,00	3,60	113,60		
Matriz	001-004-00000008	05/05/2019	Autorizado	Consumidor Final	29,20	0,00	29,20		

Figura 3.42 Listado de comprobantes generados.

Sucursal Cumbayá	001-003-00000005	22/04/2019	Autorizado	Karen Jimenez	188,00	2,69	190,69		
Sucursal Cumbayá	001-003-00000006	22/04/2019	En procesamiento	Joan Levy	110,00	3,60	113,60		
Matriz	001-004-00000008	05/05/2019	Autorizado	Consumidor Final	29,20	0,00	29,20		

Comprobante autorizado.

Figura 3.43 Mensaje de verificación de comprobante.

MADERS HERRAJES S.A.
1722868195001
Av del mozaño y Av de la Prensa

Contribuyente especial: NO
Obligado a llevar contabilidad: NO

FACTURA 001-004-000000008
Guía Remisión: -
MAYO 5, 2019

AUTORIZACIÓN
0505201901100100400000000817228681951

AMBIENTE: PRUEBAS
EMISIÓN: NORMAL
FECHA: 05/05/2019
CLAVE DE ACCESO

0505201901172286819500110010040000000081234567819

Datos del Cliente

Razon Social / Hombre: Consumidor Final

Identificación: -
Teléfonos: -
Dirección: -
E-mail: -

Detalle de productos

Cantidad	Código	Descripción	Precio Unitario	Descuento	Subtotal	
1	11710	CARTUCHO PUESTA LINMEX 710	7,00	0,00	7,00	
1	0889	CUCHILLO 0889 LION TOOLS PAVONADO RUB	22,20	0,00	22,20	
					SUBTOTAL 12%	\$0,00
					SUBTOTAL De	\$29,20
					SUBTOTAL SIN IVEST	\$29,20
					IVA	\$0,00
					ICE	\$0,00
					DESCUENTO	\$0,00
					IBRRIR	\$0,00
					PROPIA	\$0,00
					VALOR TOTAL	\$29,20

Figura 3.44 Visualización de comprobante generado.

Después, se realizó pruebas del envío del comprobante a través de correo electrónico a las direcciones de correo ingresadas tal como se muestra en la Figura 3.45.



Figura 3.45 Ventana de ingreso de destinatarios de correo electrónico.

Tras la prueba de envío el sistema devolvió el mensaje de éxito tal como se muestra en la Figura 3.46.



Figura 3.46 Mensaje de verificación de envío de comprobante.

Así mismo como se indica en la Figura 3.47 se validó la recepción del comprobante electrónico en la bandeja de entrada del correo ingresado.

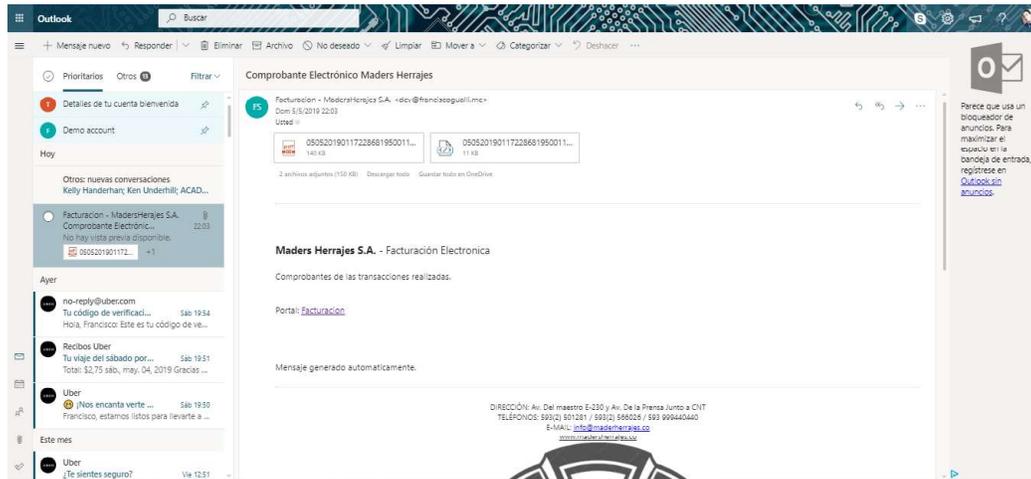


Figura 3.47 Comprobante recibido mediante correo electrónico.

Tal como se muestra en la Figura 3.48 y la Figura 3.49 se verificó la recepción tanto del archivo PDF y el archivo XML firmados electrónicamente por el emisor.

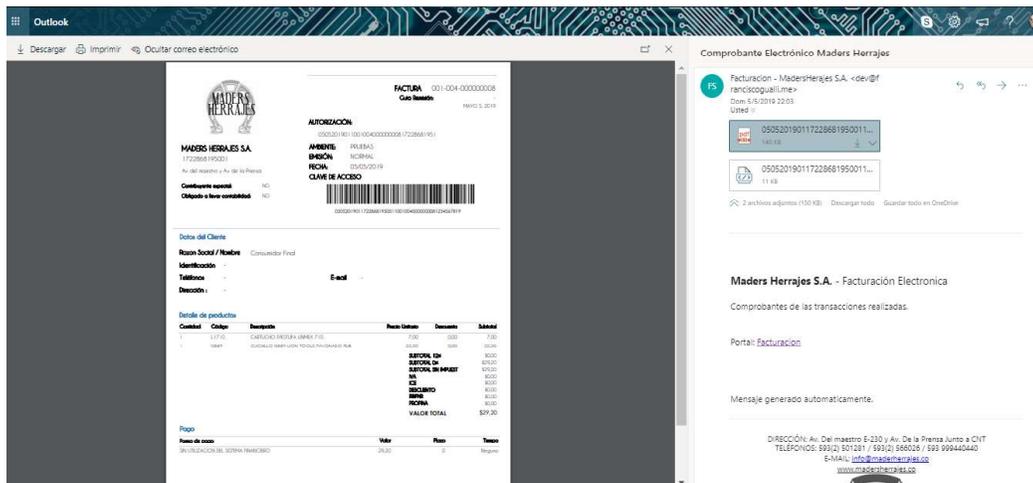


Figura 3.48 Archivo PDF del comprobante electrónico recibido.

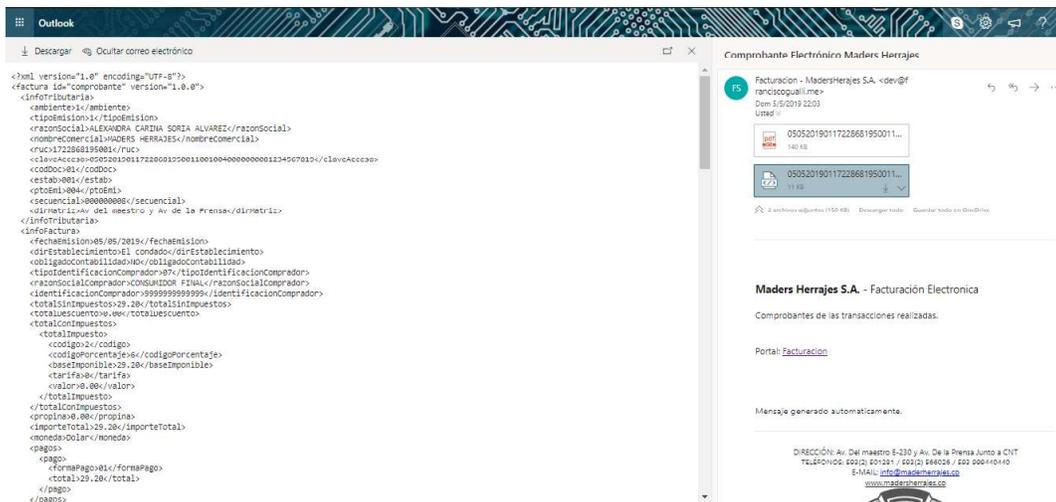


Figura 3.49 Archivo XML del comprobante electrónico recibido.

3.6.7. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE REPORTES

Para realizar las pruebas de funcionamiento del módulo de reportes se tomó en cuenta la visualización correcta de la información del negocio a través de un usuario con rol administrador.

Como prueba se tomó el año 2019 para la generación del reporte de ventas que toma en cuenta la información de facturas, notas de crédito y notas de débito generadas en el sistema.

Tal como se muestra en la Figura 3.50 se comprobó la correcta visualización del reporte gerencia de ventas del año 2019 según los valores indicados.

Así mismo se verificó la correcta impresión del reporte generado tal como se muestra en la Figura 3.51.

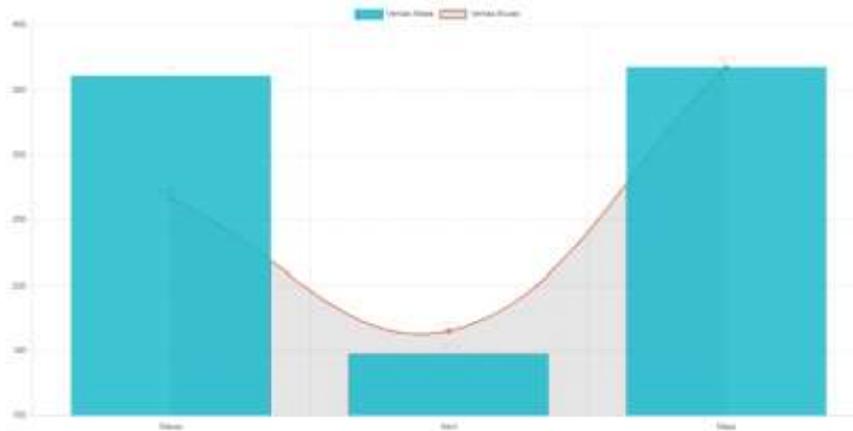
Para validar los datos del comprobante se visualizó a través del IDE Microsoft SQL Server Management Studio los datos generados por la vista `vw_DatosVentas` en donde se agrupa por cada cliente el número de comprobantes del mismo tipo, la sucursal de generación y el valor total de los comprobantes agrupados tal como se muestra en la Figura 3.52.

Tal como se muestra en la Figura 3.53 el procedimiento almacenado `Sp_GetDatosVentas` agrupa por período y sucursal los valores presentados por la vista.



Figura 3.50 Visualización de reporte gerencial de ventas.

Gerencial de Ventas (2019)



Mes	Ventas Brutas	Notas Crédito	Notas Débito	Ventas Netas	Total Acumulado
Marzo	267,00	173,00	267,00	94,00	94,00
Abril	165,40	17,80	0	147,60	241,60
Mayo	367,60	0	0	367,60	609,20
TOTAL	800,00	190,80	267,00	876,20	1745,80

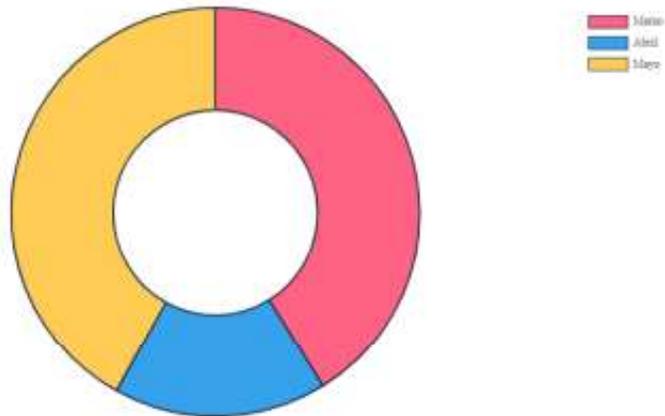


Figura 3.51 Reporte exportado como archivo PDF.

FacturacionElectron...aDatabase (sa (58))* SQLQuery1.sql - WS...aDatabase (sa (56))* ~vs786E.sql - WS201...17.master (sa (5

```
select * from Vw_DatosVentas
```

121 %

	Total	Numero_Identificacion	Nombre_Razon_Social	Periodo	Id_Sucursal	Tipo_Comprobante	Numero_Comprobantes
1	14.00	1086468043430	Summer Walker	2019-03-01	3	2	1
2	17.80	080045793940	Joey Barr	2019-04-01	1	3	1
3	23.00	2127802891661	Freddie Waters	2019-03-01	3	2	1
4	35.00	30910238643	Adam Weaver	2019-03-01	3	2	1
5	75.00	1838136837	Vicki Payne	2019-03-01	3	2	5
6	120.00	267142894683	Pedro Bryant	2019-03-01	3	2	1
7	165.40	NULL	NULL	2019-04-01	4	1	1
8	173.00	0248034999	Toby Davis	2019-03-01	3	3	1
9	367.60	NULL	NULL	2019-05-01	1	1	2
10	792.00	NULL	NULL	2019-04-01	3	1	6
11	1062.20	1723267819	Francisco Gualli	2019-04-01	1	1	1
12	2828.00	NULL	NULL	2019-04-01	1	1	14

Figura 3.52 Resultado de vista Vw_DatosVentas.

FacturacionElectron...aDatabase (sa (58))* SQLQuery1.sql - WS...aDatabase (sa (56))*

```
exec Sp_GetDatosVentas 2019
```

121 %

	Total	Periodo	Id_Sucursal	Tipo_Comprobante	Numero_Comprobantes
1	267.00	2019-03-01	3	2	9
2	173.00	2019-03-01	3	3	1
3	3890.20	2019-04-01	1	1	15
4	17.80	2019-04-01	1	3	1
5	792.00	2019-04-01	3	1	6
6	165.40	2019-04-01	4	1	1
7	367.60	2019-05-01	1	1	2

Figura 3.53 Resultado del procedimiento almacenado Sp_GetDatosVentas.

3.6.8. PRUEBAS DE FUNCIONALIDAD DEL MÓDULO DE CLIENTES

Para realizar las pruebas de funcionamiento del módulo de clientes se verificó el acceso al sistema de un cliente mediante el ingreso de la identificación para la generación de comprobantes electrónicos tal como se muestra en la Figura 3.54.

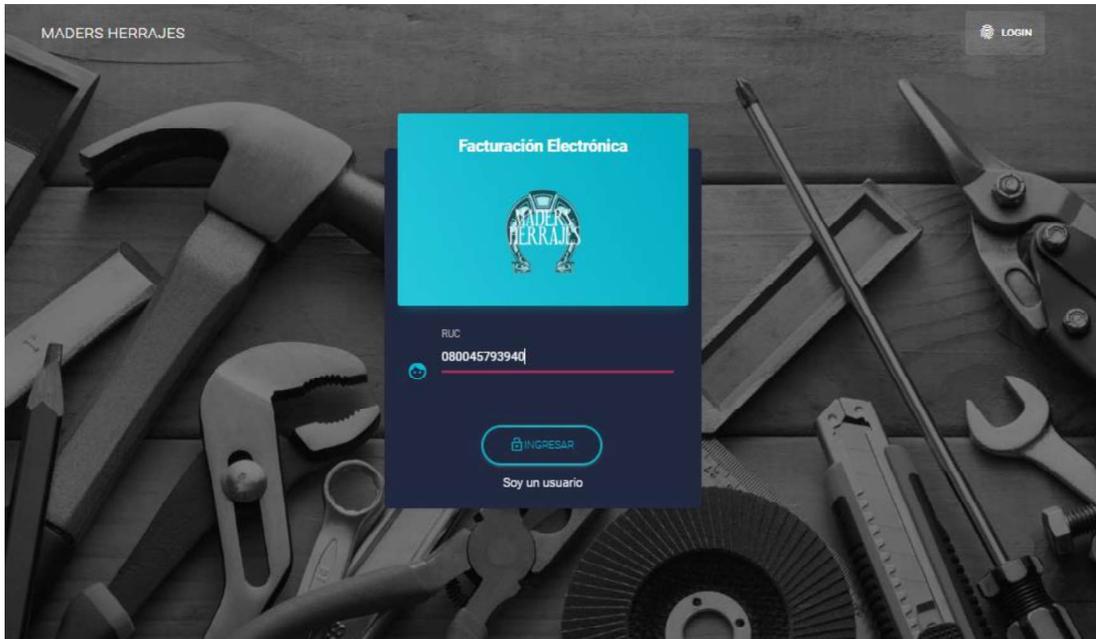


Figura 3.54 Ingreso de identificación del cliente.

Se verificó los permisos en el sistema que tiene el cliente como son la visualización de comprobantes electrónicos generados con su número de cédula, también debe proporcionar el número del comprobante tal como se muestra en la Figura 3.55.

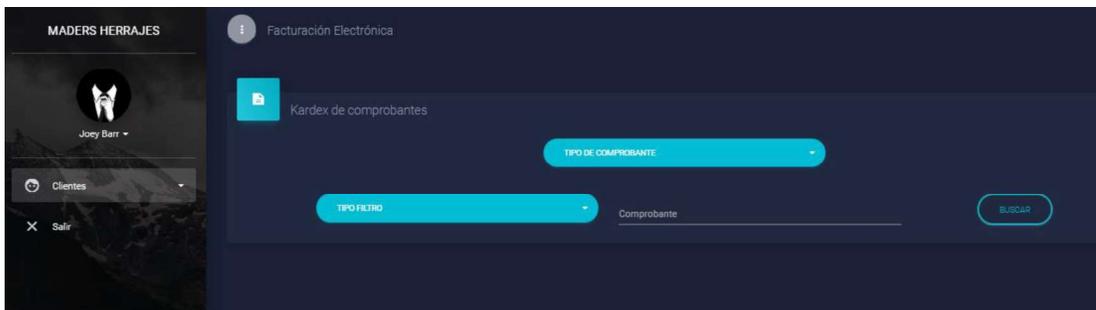


Figura 3.55 Vista principal del cliente.

Después, se validó la búsqueda y visualización del resumen de movimientos para el comprobante buscado tal como se muestra en la Figura 3.56.



Figura 3.56 Visualización de KARDEX del comprobante.

Así mismo, tal como se muestra en la Figura 3.57 se pudo visualizar el comprobante generado y los datos del mismo.

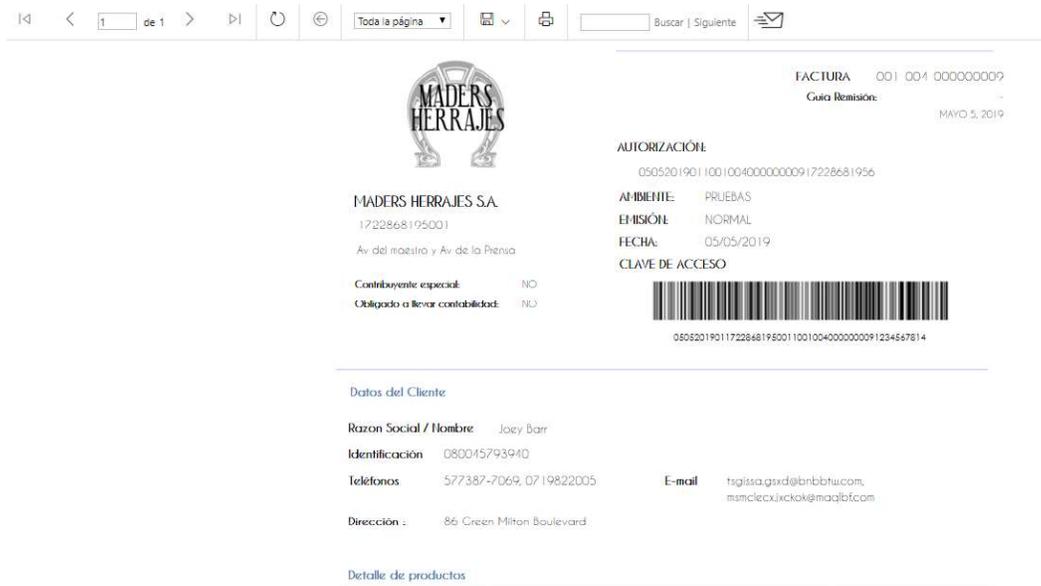


Figura 3.57 Visualización de comprobante.

3.7. CORRECCIÓN DE ERRORES EN LOS COMPONENTES DEL PROTOTIPO

Durante el desarrollo del prototipo los principales fallos encontrados fueron en la codificación correcta de los parámetros del archivo XML correspondiente al comprobante electrónico generado para ser validado por los servicios web del SRI, para lo cual se utilizó la herramienta gratuita para generación de comprobantes electrónicos proporcionada por el SRI para la comparación del archivo generado en ambos sistemas.

Además de los fallos encontrados en la generación de los comprobantes electrónicos, otro de los problemas de gran relevancia fue en el despliegue del prototipo en el servidor en Internet, debido a los permisos de acceso a los directorios del servidor lo cual fue solventado otorgando permisos de lectura y escritura al usuario utilizado por IIS para los directorios en los que se cargan archivos.

Finalmente, todos los errores e inconvenientes presentados en la etapa de implementación fueron solventados permitiendo un funcionamiento óptimo del prototipo en el servidor web.

3.7.4. RESUMEN DE CORRECCIONES

A continuación, en la Tabla 3.1 se presenta un resumen de los cambios realizados en el prototipo.

Tabla 3.1 Tabla de resumen de correcciones.

Falla	Capa	Módulo	Solución
Manejar la secuencia de comprobantes para los diferentes emisores del sistema	Datos	Administración	Creación de tabla relacionada para almacenar la secuencia de comprobantes y tipos de comprobantes.
Error al obtener datos de comprobantes electrónicos emitidos como consumidor final	Datos	Comprobantes	Modificación en procedimiento almacenado que devuelve los datos del comprobante (Cambio de INNER JOIN por LEFT JOIN en tabla Clientes).
Error en el cálculo de valores del reporte de ventas por selección de comprobantes generados	Datos	Reportes	Se seleccionó los comprobantes en estado Autorizado para la generación del reporte.
Error en separación de entidades de la capa de acceso datos y entidades	Entidades	Todos	Se modificó el archivo FacturacionElectronicaModel.tt indicando el path de la capa de Entidades
Inclusión de tablas relacionadas en el mapeo de las entidades	Acceso a datos	Todos	En la clase repository en los métodos de búsqueda o listado se agregó como parámetro de entrada las entidades

			relacionadas a la tabla requerida.
Envío de parámetros para la ejecución de procedimientos almacenados	Acceso a datos	Todos	Se envió el nombre del procedimiento como string y los parámetros que recibe de entrada como una lista de parámetros SQL.
Cuando se tenía contraseñas iguales el hash generado era el mismo	Datos y Negocio	Autenticación	Se utilizó para la generación del hash la contraseña concatenada a la fecha y hora de registro del usuario.
Carga de certificados digitales	Presentación	Administración	Agregar archivos de tipo .pfx a los tipos admitidos por la aplicación web en IIS.
Almacenamiento de archivos en el servidor	Presentación	Administración y Comprobantes	Conceder permisos de escritura y lectura al usuario IIS_IUSRS en los directorios destinados para almacenar archivos cargados.
Error en el archivo web.config en servidor web	Presentación	Autenticación	Instalación del complemento para soporte de reescritura de reglas en IIS.
Error en la respuesta del servicio web	Servicio	Comprobantes	Se modificó el archivo Reference.cs generado por la referencia al servicio web cambiando el tipo de objeto devuelto.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Según los resultados encontrados se concluye que la implementación del sistema web prototipo para facturación electrónica permite mejorar la administración de acuerdo a la información que genera el sistema como es la lista de comprobantes de ventas, notas de débito, notas de crédito, retenciones y guías de remisión.
- Una de las etapas más importantes en el desarrollo de la aplicación web fue el análisis de requerimientos, puesto que ayudó a la comprensión del funcionamiento del proceso de facturación de la ferretería y permitió desarrollar una herramienta para automatizar uno de los procesos principales de esta, también fue de gran utilidad la entrevista con el cliente, ya que permitió que el proceso de análisis de requerimientos se realice en el menor tiempo posible.
- El prototipo desarrollado es altamente escalable, en corto plazo sería posible el desarrollo de nuevas funcionalidades que automaticen otros procesos relacionados con el proceso de facturación como por ejemplo: manejo de inventarios y nómina.
- El uso de la metodología ágil RAD se acopló perfectamente a la arquitectura N-Capas del sistema debido a la codificación de cada una de las capas en el ciclo iterativo del desarrollo de prototipos planteado por la metodología.
- El uso de la herramienta Entity Framework permitió agilizar el proceso de consumo de la capa de datos. Así mismo gracias a la herramienta se pudo desacoplar la capa de acceso de datos del origen de datos pudiéndose utilizar otras bases de datos como por ejemplo Oracle, MySQL, etc.
- Con el uso de GitHub se pudo tener una visión clara de todo el proceso de desarrollo y tiempo utilizado en la codificación del prototipo.
- La configuración del archivo web.config permitió realizar la administración de la aplicación web desde aspectos como la seguridad hasta la optimización y renderizado del sistema.
- Es importante tener en cuenta tener habilitados los dos esquemas de emisión de comprobantes uno electrónico y uno tradicional en caso de contingencia por falla del SRI o del sistema de facturación electrónica.
- Gracias a las pruebas de funcionalidad de cada capa desarrollada para el prototipo fue posible depurar y corregir las fallas de los módulos en una etapa temprana del proceso de desarrollo.

4.2. RECOMENDACIONES

- A pesar de que el sistema es totalmente responsivo para la generación de comprobantes electrónicos se recomienda el uso de un computador para poder verificar los datos del comprobante a emitirse.
- Dado que el sistema está implementado con tecnologías Microsoft para el lanzamiento a un ambiente de producción se recomienda el uso de la plataforma Azure ya que al ser parte de Microsoft se presenta optimizada para el despliegue de este tipo de aplicaciones.
- Dado que el sistema utiliza los servicios web del SRI se recomienda que en la parametrización del emisor se establezca un intervalo de al menos 3 segundos de espera por la respuesta de la plataforma debido a problemas que pudieren presentarse como por ejemplo alta latencia.
- Para el consumo de los servicios web del SRI se recomienda el uso del método de referencia como servicio web ya que la solicitud HTTP se genera automáticamente y se optimizan los tiempos de respuesta.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] El telégrafo, "Pequeñas empresas emitirán factura electrónica en 2019," 2019, [Online]. Available: <https://www.eltelegrafo.com.ec/noticias/economia/4/pequeñas-empresas-emitiran-factura-electronica-en-2019> [Accessed: 23-May-2019].
- [2] Lucidchart, "4 Phases of Rapid Application Development Methodology," 2019, [Online]. Available: <https://www.lucidchart.com/blog/rapid-application-development-methodology> [Accessed: 23-May-2019].
- [3] M. Muñoz "Introducción al desarrollo de aplicaciones N-Capas con tecnologías Microsoft," Ti-capacitación, México, 2018. pp. 6-8, 17-20.
- [4] D. E. Avison y G. Fitzgerald, "Information system development," Maidenhead: McGraw-Hill Education, 2006.
- [5] Powell A, "Rapid application development: What is this and how to used it?," 2016, [Online] Available: <https://airbrake.io/blog/sdlc/rapid-application-development> [Accessed: 23-May-2019].
- [6] M. Ebert, "The Repository Pattern," 2013, [Online] Available: <https://8thlight.com/blog/mike-ebert/2013/03/23/the-repository-pattern.html> [Accessed: 23-May-2019].
- [7] A. Oppel y R. Sheldon, "SQL A Beginner's Guide," Third Edition. McGraw-Hill, 2009.
- [8] J. Groff y P. Weinberg, "SQL: The Complete Reference," Second Edition. McGraw-Hill, 2002.
- [9] T. Thuan y L. Hoang, ".NET Framework Essentials," Second Edition. O'Reilly, 2002.
- [10] M. Muñoz, "Programador Jr. de aplicaciones ASP.NET MVC," Ti-capacitación, México, 2018. pp. 14-17, 32-33.
- [11] J.D. Gauchat, "El gran libro de HTML5, CSS3 y Javascript," Primera Edición, España: marcombo, 2012, pp. 18, 32.
- [12] J. Perez, A. Gardey, "DEFINICIÓN DE HTML," 2012, [Online] Available: <https://definicion.de/html/> [Accessed: 23-May-2019].
- [13] W3C, "Cascading Style Sheets," 2018, [Online] Available: <https://www.w3.org/Style/CSS/Overview.en.html>. [Último acceso: 16-05-2018].
- [14] Firefox, "JavaScript," [Online], Available: <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Accessed: 23-May-2019].

- [15] M. Muñoz “BootStrap,” Ti-capacitación, México, 2018, pp. 6-10.
- [16] Bootstrap, “Grid System BootStrap Documentation,” 2019, [Online] Available: <https://getbootstrap.com/docs/4.0/layout/grid/> [Accessed: 23-May-2019].
- [17] Maslinux, “Introducción a GitHub,” 2017, [Online] Available: <https://maslinux.es/introduccion-a-github/> [Accessed: 23-May-2019].
- [18] SRI, “Comprobantes Electrónicos | Información básica,” 2019, [Online] Available: http://www.sri.gob.ec/documents/156146/0/pdf+FACTURACION+ELECTRONICA+V1_out_03_03_2015.pdf/489fb78d-5e8d-4a01-808f-b4417d1842dc [Accessed: 23-May-2019].
- [19] SRI, “FICHA TÉCNICA: MANUAL DE USUARIO, CATÁLOGO Y ESPECIFICACIONES TÉCNICAS SOBRE EL PROCESO DE AUTORIZACIÓN Y EMISIÓN DE DOCUMENTOS ELECTRÓNICOS,” 2019 [Online] Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjUutPjkrHiAhWqpFkKHS_xA4MQFjAAegQIAxAC&url=http%3A%2F%2Fwww.sri.gob.ec%2FDocumentosAlfrescoPortlet%2Fdescargar%2Fbb05cf5-32f2-4cc3-bc8a-b539c9e0ed74%2FFICHA%2BTECNICA%2BCOMPROBANTES%2BELECTRO%25B4NICOS%2Boffline.pdf&usq=AOvVaw04J2B-BrQBsPohEP89sDfe [Accessed: 23-May-2019].
- [20] Microsoft, “Entity Framework 6,” 2019, [Online] Available: <https://docs.microsoft.com/en-us/ef/ef6/> [Accessed: 24-May-2019].

ANEXOS

Los anexos se incluyen en el CD adjunto al presente documento.

ANEXO A. Encuesta para la obtención de requerimientos.

ANEXO B. Diagramas utilizados para el desarrollo del sistema.

ANEXO C. *Scripts* para la creación de la base de datos, procedimientos almacenados, vistas y triggers.

ANEXO D. Pruebas del sistema.

ANEXO E. Código fuente del sistema.

ANEXO F. Manual de usuario.

ORDEN DE EMPASTADO