



UNIVERSIDAD TÉCNICA DE AMBATO

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

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

TEMA:

**SISTEMA AUTOMATIZADO PARA LA GESTIÓN DE LAS COMUNIDADES
DE LA PARROQUIA SALASACA DE LA CIUDAD DE PELILEO.**

Trabajo de Graduación. Modalidad: Proyecto de Investigación. Trabajo Estructurado de Manera Independiente,
presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

**AUTOR: Curichumbi Jerez Oscar Eduardo
TUTOR: Ing. Rubén Eduardo Nogales Portero**

Ambato - Ecuador
Enero, 2020

CERTIFICACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

“SISTEMA AUTOMATIZADO PARA LA GESTIÓN DE LAS COMUNIDADES DE LA PARROQUIA SALASACA DE LA CIUDAD DE PELILEO”, del señor Curichumbi Jerez Oscar Eduardo, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato

Ambato, Enero de 2020



Ing. Rubén Eduardo Nogales Portero

TUTOR

AUTORÍA DEL TRABAJO

El presente trabajo de investigación titulado: “SISTEMA AUTOMATIZADO PARA LA GESTIÓN DE LAS COMUNIDADES DE LA PARROQUIA SALASACA DE LA CIUDAD DE PELILEO”. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Enero de 2020



Oscar Eduardo Curichumbi Jerez

CC: 1805198478

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este trabajo de titulación como un documento disponible para su lectura, consulta y procesos de investigación. Cedo los Derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, Enero de 2020



Oscar Eduardo Curichumbi Jerez

CC: 1805198478

APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. David Guevara e Ing. Hernando Buenaño, revisó y aprobó el Informe Final del trabajo de graduación titulado "SISTEMA AUTOMATIZADO PARA LA GESTIÓN DE LAS COMUNIDADES DE LA PARROQUIA SALASACA DE LA CIUDAD DE PELILEO", presentado por el señor Oscar Eduardo Curichumbi Jerez de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.



Ing. Pilar Urrutia, Mg.

PRESIDENTA ENCARGADA DEL TRIBUNAL



Ing. David Guevara
DOCENTE CALIFICADOR.



Ing. Hernando Buenaño
DOCENTE CALIFICADOR.

DEDICATORIA

El presente proyecto está dedicado a mi madre, por haber sido mi apoyo a lo largo de toda mi vida y a lo largo de mi carrera universitaria, quien me inspiró y dio fuerzas para continuar en este proceso para obtener uno de mis anhelos más deseados.

A mi padre, por su sacrificio en todos estos años, gracias a usted he logrado llegar hasta aquí y convertirme en lo que soy hoy en día.

A mi hermano y hermana, por estar siempre presentes, brindándome apoyo moral a lo largo de esta etapa de mi vida.

A la Universidad Técnica de Ambato por ser mi segundo hogar y la sede de todo el conocimiento adquirido en estos años.

Finalmente, a todas las personas que me han apoyado, en especial a aquellos que me abrieron las puertas y compartieron sus conocimientos para que el trabajo se realice con éxito.

Curichumbi Jerez Oscar Eduardo

AGRADECIMIENTO

Agradezco primeramente a Dios por bendecirme la vida, por acompañarme a lo largo de mi existencia, y guiar mi camino para poder salir adelante logrando cada objetivo propuesto en mi vida.

A mi madre por su apoyo en todo momento, por ser mi pilar fundamental, por infundir en mí su ejemplo de lucha constante, y ser el apoyo en aquellos momentos de dificultad y debilidad.

A mi Tutor de Tesis el Ing. Rubén Nogales, por compartirme sus conocimientos, y por haberme guiado en la elaboración de este proyecto de titulación para desarrollarme profesionalmente.

Agradezco a mi querida Universidad por abrirme sus puertas, y a mis profesores por la orientación, por la paciencia y por infundirme sus conocimientos para permitirme concluir con una etapa de mi vida, y hacer de mí una persona de bien.

Curichumbi Jerez Oscar Eduardo

ÍNDICE

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
APROBACIÓN COMISIÓN CALIFICADORA	v
Dedicatoria	vi
Agradecimiento	vii
Introducción	xxv
CAPÍTULO 1 El problema	1
1.1 Tema de investigación	1
1.2 Planteamiento del problema	1
1.3 Delimitación	2
1.4 Justificación	3
1.5 Objetivos	4
1.5.1 General	4
1.5.2 Específicos	4
CAPÍTULO 2 Marco Teórico	5
2.1 Antecedentes investigativos	5
2.2 Fundamentación filosófica	5
2.3 Fundamentación teórica	6
2.3.1 Gestión de la Calidad	6
2.3.2 Gestión Documental	6
2.3.3 Funciones Administrativas	6
2.3.4 Contabilidad Administrativa	7
2.3.5 Gestión por Procesos	7
2.3.6 Gestión Administrativa	7
2.3.7 Metodologías de desarrollo de Software	7

2.3.8	Metodologías Ágiles	8
2.3.9	Metodologías Tradicionales	8
2.3.10	Aplicaciones Web	8
2.3.11	ASP.NET	8
2.4	Propuesta de solución	9
CAPÍTULO 3 Metodología		10
3.1	Modalidad de la investigación	10
3.1.1	Modalidad de Campo	10
3.1.2	Modalidad Bibliográfica o Documental	10
3.1.3	Modalidad Aplicada	10
3.2	Población y muestra	10
3.3	Recolección de información	10
3.4	Procesamiento y análisis de datos	11
3.5	Desarrollo del proyecto	11
CAPÍTULO 4 Desarrollo de la propuesta		13
4.1	Analizar los procesos manuales de la comunidad para definir las necesidades	13
4.1.1	Identificar y Definir las necesidades y los requerimientos	13
4.2	Investigar las herramientas adecuadas para el desarrollo de la aplicación web	14
4.2.1	Análisis del Gestor de Base de Datos a Aplicar	14
4.2.2	Análisis del Lenguaje de Programación a Aplicar	16
4.2.3	Análisis del Entorno de Desarrollo Integrado (IDE) a Aplicar	19
4.2.4	Análisis de la Metodología de Desarrollo de Software a Aplicar	20
4.2.5	Herramientas de la Metodología XP	25
4.2.5.1	Historias de Usuario	25
4.2.5.2	Tareas de Ingenierías	26
4.2.5.3	Pruebas de Aceptación	27
4.2.5.4	Tarjetas CRC	27
4.2.5.5	Fases de la Metodología XP	28
4.3	Desarrollar el Sistema Informático para la Gestión de las comuni- dades de la parroquia Salasaca	30
4.3.1	Planificación del Proyecto	30
4.3.2	Historias de Usuarios	30
4.3.3	Equipos y Roles	37
4.3.4	Planificación de las Entregas al Cliente	38

4.3.5	Desarrollo de la Primera Iteración	39
4.3.5.1	Estimación de la Primera Iteración	39
4.3.5.2	Bocetos de las Historias de Usuarios 1-2	40
4.3.5.3	Descripción de las Tareas de Ingenierías	43
4.3.5.4	Tarjetas CRC: Historias de usuario 1-2	46
4.3.5.5	Diagrama de datos	47
4.3.5.6	Diseño de Interfaz	48
4.3.5.7	Codificación	51
4.3.5.8	Pruebas de Aceptación	57
4.3.5.9	Resultados de la Primera Iteración	58
4.3.6	Desarrollo de la Segunda Iteración	58
4.3.6.1	Estimación de la Segunda Iteración	58
4.3.6.2	Bocetos de las Historias de Usuarios 3-4-5	59
4.3.6.3	Descripción de las Tareas de Ingenierías	65
4.3.6.4	Tarjetas CRC: Historias de usuario 3-4-5	70
4.3.6.5	Diagrama de datos	72
4.3.6.6	Diseño de Interfaz	72
4.3.6.7	Codificación	80
4.3.6.8	Pruebas de Aceptación	100
4.3.6.9	Resultados de la Segunda Iteración	103
4.3.7	Desarrollo de la Tercera Iteración	103
4.3.7.1	Estimación de la Tercera Iteración	103
4.3.7.2	Bocetos de la Historia de Usuario 6	104
4.3.7.3	Descripción de las Tareas de Ingenierías	106
4.3.7.4	Tarjetas CRC: Historia de usuario 6	109
4.3.7.5	Diagrama de datos	109
4.3.7.6	Diseño de Interfaz	110
4.3.7.7	Codificación	113
4.3.7.8	Pruebas de Aceptación	128
4.3.7.9	Resultados de la Tercera Iteración	130
4.3.8	Desarrollo de la Cuarta Iteración	130
4.3.8.1	Estimación de la Cuarta Iteración	130
4.3.8.2	Bocetos de la Historia de Usuario 7	131
4.3.8.3	Descripción de las Tareas de Ingenierías	140
4.3.8.4	Tarjetas CRC: Historias de usuario 7	144
4.3.8.5	Diagrama de datos	147
4.3.8.6	Diseño de Interfaz	147

4.3.8.7	Codificación	157
4.3.8.8	Pruebas de Aceptación	178
4.3.8.9	Resultados de la Cuarta Iteración	181
4.3.9	Desarrollo de la Quinta Iteración	181
4.3.9.1	Estimación de la Quinta Iteración	182
4.3.9.2	Bocetos de las Historias de Usuarios 8-9-10	182
4.3.9.3	Descripción de las Tareas de Ingenierías	187
4.3.9.4	Tarjetas CRC: Historias de usuario 8-9-10	192
4.3.9.5	Diagrama de datos	194
4.3.9.6	Diseño de Interfaz	194
4.3.9.7	Codificación	200
4.3.9.8	Pruebas de Aceptación	210
4.3.9.9	Resultados de la Quinta Iteración	213
4.3.10	Desarrollo de la Sexta Iteración	213
4.3.10.1	Estimación de la Primera Iteración	214
4.3.10.2	Bocetos de las Historias de Usuarios 11-12-13-14	214
4.3.10.3	Descripción de las Tareas de Ingeniería	222
4.3.10.4	Diagrama de datos	225
4.3.10.5	Diseño de Interfaz	225
4.3.10.6	Codificación	236
4.3.10.7	Pruebas de Aceptación	242
4.3.10.8	Resultados de la Sexta Iteración	243
CAPÍTULO 5 Conclusiones y Recomendaciones		244
5.1	Conclusiones	244
5.2	Recomendaciones	245
Bibliografía		246
ANEXOS		250

ÍNDICE DE TABLAS

4.2	Gestores de Base de Datos.	15
4.3	Cuadro Comparativo - Gestores de Base de Datos.	16
4.5	Lenguajes de Programación.	18
4.6	Cuadro Comparativo - Lenguajes de Programación.	18
4.7	Cuadro de Comparativo - Entorno de Desarrollo Integrado	19
4.9	Cuadro comparativo - Metodologías ágiles	24
4.10	Plantilla - Historia de Usuario.	26
4.11	Plantilla - Tareas de Ingenierías.	26
4.12	Plantilla - Pruebas de Aceptación.	27
4.13	Plantilla - Tarjetas CRC.	28
4.14	Historia de Usuario: Ingreso al sistema.	31
4.15	Historia de Usuario: Creación de cargos y permisos.	31
4.16	Historia de Usuario: Registro de Usuarios.	32
4.17	Historia de Usuario: Asignación de usuarios para la directiva de la comunidad.	32
4.19	Historia de Usuario: Control de asistencia.	33
4.18	Historia de Usuario: Generar reuniones.	33
4.20	Historia de Usuario: Tesorería.	34
4.21	Historia de Usuario: Control de Entrada y Salida de documentos. . . .	34
4.22	Historia de Usuario: Egresos y Otros Ingresos.	35
4.23	Historia de Usuario: Reporte general de Ingresos y Egresos.	35
4.24	Historia de Usuario: Reportes del presidente o vicepresidente. . . .	36
4.25	Historia de Usuario: Reportes del secretario(a).	36
4.26	Historia de Usuario: Reportes de Tesorería.	37
4.27	Historia de Usuario: Reportes de los Vocales.	37
4.28	Equipos y Roles.	38
4.29	Planificación de las Entregas al cliente.	38
4.30	Tareas a Realizar de las Historias de Usuario 1-2.	39
4.31	Horas asignadas para el desarrollo de la Iteración 1.	40
4.32	Tarea 1 de la Historia de Usuario N.-1	43

4.33	Tarea 2 de la Historia de Usuario N.-1	44
4.34	Tarea 3 de la Historia de Usuario N.-1	44
4.35	Tarea 4 de la Historia de Usuario N.-1	44
4.36	Tarea 5 de la Historia de Usuario N.-2	45
4.37	Tarea 6 de la Historia de Usuario N.-2	45
4.38	Tarea 7 de la Historia de Usuario N.-2	45
4.39	Tarea 8 de la Historia de Usuario N.-2	45
4.40	Tarea 9 de la Historia de Usuario N.-2	46
4.41	Tarea 10 de la Historia de Usuario N.-2	46
4.42	Tarjeta CRC: Directivos.	46
4.43	Tarjeta CRC: Tipo Cargo.	47
4.44	Tarjeta CRC: Cargo Menu.	47
4.45	Tarjeta CRC: Menu.	47
4.46	Caso de Prueba: Ingreso al sistema.	57
4.47	Caso de Prueba: Creación de cargos y permisos.	57
4.48	Tareas a Realizar de las Historias de Usuario 3-4-5.	58
4.49	Horas asignadas para el desarrollo de la Iteración 2.	59
4.50	Tarea 11 de la Historia de Usuario N.-3	66
4.51	Tarea 12 de la Historia de Usuario N.-3	66
4.52	Tarea 13 de la Historia de Usuario N.-3	66
4.53	Tarea 14 de la Historia de Usuario N.-3	66
4.54	Tarea 15 de la Historia de Usuario N.-3	67
4.55	Tarea 16 de la Historia de Usuario N.-3	67
4.56	Tarea 17 de la Historia de Usuario N.-4	67
4.57	Tarea 18 de la Historia de Usuario N.-4	67
4.58	Tarea 19 de la Historia de Usuario N.-4	68
4.59	Tarea 20 de la Historia de Usuario N.-4	68
4.60	Tarea 21 de la Historia de Usuario N.-5	68
4.61	Tarea 22 de la Historia de Usuario N.-5	69
4.62	Tarea 23 de la Historia de Usuario N.-5	69
4.63	Tarea 24 de la Historia de Usuario N.-5	69
4.64	Tarea 25 de la Historia de Usuario N.-5	69
4.65	Tarea 26 de la Historia de Usuario N.-5	70
4.66	Tarjeta CRC: Usuarios.	70
4.67	Tarjeta CRC: Directivos.	71
4.68	Tarjeta CRC: Reuniones.	71
4.69	Tarjeta CRC: Tipo Reunión.	72

4.70	Caso de Prueba: Ingreso de usuarios al sistema.	101
4.71	Caso de Prueba: Asignación de usuarios para la directiva de la comunidad.	101
4.72	Caso de Prueba: Generar reuniones.	102
4.73	Tareas a Realizar de la Historia de Usuario 6.	103
4.74	Horas asignadas para el desarrollo de la Iteración 3.	104
4.75	Tarea 27 de la Historia de Usuario N.-6.	107
4.76	Tarea 28 de la Historia de Usuario N.-6.	107
4.77	Tarea 29 de la Historia de Usuario N.-6.	107
4.78	Tarea 30 de la Historia de Usuario N.-6.	108
4.79	Tarea 31 de la Historia de Usuario N.-6.	108
4.80	Tarea 32 de la Historia de Usuario N.-6.	108
4.81	Tarea 33 de la Historia de Usuario N.-6.	109
4.82	Tarjeta CRC: Control de Asistencia.	109
4.83	Caso de Prueba: Control de Asistencia.	129
4.84	Tareas a Realizar de la Historia de Usuario 7.	130
4.85	Horas asignadas para el desarrollo de la Iteración 4.	131
4.86	Tarea 34 de la Historia de Usuario N.-7.	141
4.87	Tarea 35 de la Historia de Usuario N.-7.	141
4.88	Tarea 36 de la Historia de Usuario N.-7.	141
4.89	Tarea 37 de la Historia de Usuario N.-7.	142
4.90	Tarea 38 de la Historia de Usuario N.-7.	142
4.91	Tarea 39 de la Historia de Usuario N.-7.	142
4.92	Tarea 40 de la Historia de Usuario N.-7.	143
4.93	Tarea 41 de la Historia de Usuario N.-7.	143
4.94	Tarea 42 de la Historia de Usuario N.-7.	143
4.95	Tarea 43 de la Historia de Usuario N.-7.	143
4.96	Tarea 44 de la Historia de Usuario N.-7.	144
4.97	Tarjeta CRC: Cobroasistencia.	144
4.98	Tarjeta CRC: Descuentomultas.	145
4.99	Tarjeta CRC: Colaboraciones.	145
4.100	Tarjeta CRC: Cobrocolaboracion.	146
4.101	Tarjeta CRC: Sanciones.	146
4.102	Tarjeta CRC: Cobrosancion.	147
4.103	Caso de Prueba: Tesorería-Registros.	178
4.104	Caso de Prueba: Tesorería-Cobros.	179
4.105	Caso de Prueba: Tesorería-Informes.	180

4.106	Tareas a Realizar de las Historias de Usuario 8-9-10.	181
4.107	Horas asignadas para el desarrollo de la Iteración 5.	182
4.108	Tarea 45 de la Historia de Usuario N.-8.	188
4.109	Tarea 46 de la Historia de Usuario N.-8.	188
4.110	Tarea 47 de la Historia de Usuario N.-8.	188
4.111	Tarea 48 de la Historia de Usuario N.-8.	189
4.112	Tarea 49 de la Historia de Usuario N.-8.	189
4.113	Tarea 50 de la Historia de Usuario N.-9.	189
4.114	Tarea 51 de la Historia de Usuario N.-9.	190
4.115	Tarea 52 de la Historia de Usuario N.-9.	190
4.116	Tarea 53 de la Historia de Usuario N.-9.	190
4.117	Tarea 54 de la Historia de Usuario N.-9.	191
4.118	Tarea 55 de la Historia de Usuario N.-10.	191
4.119	Tarea 56 de la Historia de Usuario N.-10.	191
4.120	Tarea 57 de la Historia de Usuario N.-10.	192
4.121	Tarea 58 de la Historia de Usuario N.-10.	192
4.122	Tarjeta CRC: Tipo Documento.	192
4.123	Tarjeta CRC: Documentos.	193
4.124	Tarjeta CRC: Egresos.	193
4.125	Tarjeta CRC: otros Ingresos.	194
4.126	Caso de Prueba: Control de Entrada y Salida de documentos. . .	211
4.127	Caso de Prueba: Egresos y otros Ingresos.	212
4.128	Caso de Prueba: Reporte general de Ingresos y Egresos.	213
4.129	Tareas a Realizar de las Historias de Usuario 11-12-13-14.	214
4.130	Horas asignadas para el desarrollo de la Iteración 6.	214
4.131	Tarea 59 de la Historia de Usuario N.-11.	223
4.132	Tarea 60 de la Historia de Usuario N.-11.	223
4.133	Tarea 61 de la Historia de Usuario N.-12.	223
4.134	Tarea 62 de la Historia de Usuario N.-12.	224
4.135	Tarea 63 de la Historia de Usuario N.-13.	224
4.136	Tarea 64 de la Historia de Usuario N.-13.	224
4.137	Tarea 65 de la Historia de Usuario N.-14.	224
4.138	Tarea 66 de la Historia de Usuario N.-14.	225
4.139	Caso de Prueba: Reportes del presidente o vicepresidente.	242
4.140	Caso de Prueba: Reportes del Secretario(a).	242
4.141	Caso de Prueba: Reportes de Tesorería.	243
4.142	Caso de Prueba: Reportes de los Vocales.	243

ÍNDICE DE FIGURAS

4.1	Bosquejo de Menú Inicio.	40
4.2	Bosquejo de Inicio de Sesión.	41
4.3	Bosquejo de Cambio de Clave.	41
4.4	Bosquejo de Registrar Cargo.	42
4.5	Bosquejo de Otorgar Permisos.	42
4.6	Bosquejo de Modificar Cargo.	43
4.7	Diagrama de datos: Directivos, Cargos y Menús.	48
4.8	Diseño de Interfaz - Ingreso al sistema.	48
4.9	Diseño de Interfaz - Menú Inicio.	49
4.10	Diseño de Interfaz - Cambiar Clave.	49
4.11	Diseño de Interfaz - Crear Cargos.	50
4.12	Diseño de Interfaz - Otorgar Permisos.	50
4.13	Diseño de Interfaz - Modificar Cargo.	51
4.14	Boceto de Crear Usuarios.	59
4.15	Boceto de Modificar Usuario.	60
4.16	Boceto de Eliminar Usuario.	60
4.17	Boceto de Registrar Directivos.	61
4.18	Boceto de Cambio de Cargo.	61
4.19	Boceto de Periodo Finalizado.	62
4.20	Boceto de Generar Reuniones.	62
4.21	Boceto de Modificar Reuniones.	63
4.22	Boceto de Subir Acta.	63
4.23	Boceto de Modificar Acta.	64
4.24	Boceto de Finalizar Reunión.	64
4.25	Boceto de Activar Reunión.	65
4.26	Boceto de Informe General de Reunión.	65
4.27	Diagrama de datos: Usuarios, Directivos y Reuniones	72
4.28	Diseño de Interfaz-Verificar número de cedula.	73
4.29	Diseño de Interfaz-Registrar Usuario.	73
4.30	Diseño de Interfaz-Modificar Usuario.	74

4.31	Diseño de Interfaz-Eliminar Usuario.	75
4.32	Diseño de Interfaz-Registrar Directivos.	75
4.33	Diseño de Interfaz-Cambio Cargo.	76
4.34	Diseño de Interfaz-Finalizar Periodo.	76
4.35	Diseño de Interfaz-Registrar Reunión.	77
4.36	Diseño de Interfaz-Modificar Reunión.	77
4.37	Diseño de Interfaz-Subir Acta de Reunión.	78
4.38	Diseño de Interfaz-Modificar Acta de Reunión.	78
4.39	Diseño de Interfaz-Finalizar Reunión.	79
4.40	Diseño de Interfaz-Reanudar Reunión.	79
4.41	Diseño de Interfaz-Informe de Reuniones.	80
4.42	Boceto de Control de Asistencia.	104
4.43	Bosquejo de Modificar Asistencia.	105
4.44	Boceto de Informe Asistencia Por Reunión.	105
4.45	Boceto de Informe Asistencia Por Usuario.	106
4.46	Boceto de Informe Asistencia Por Estado.	106
4.47	Diagrama de datos: Control de Asistencia.	110
4.48	Diseño de Interfaz-Control de Asistencia.	110
4.49	Diseño de Interfaz-Control de Asistencia Usuarios.	111
4.50	Diseño de Interfaz-Control de Asistencia Directivos.	111
4.51	Diseño de Interfaz-Modificar Asistencia.	112
4.52	Diseño de Interfaz-Informe General de Asistencia.	112
4.53	Diseño de Interfaz-Informe Asistencia Por Usuario.	113
4.54	Diseño de Interfaz-Informe Asistencia Por Estado.	113
4.55	Boceto de Registro de Colaboraciones.	131
4.56	Boceto de Registro de Sanciones.	132
4.57	Boceto de Cobro de Multas Por Usuario.	132
4.58	Boceto de Cobro de Multas Por Reunión.	133
4.59	Boceto de Cobro de Colaboraciones Por Usuario.	133
4.60	Boceto de Cobro de Colaboraciones Por Reunión.	134
4.61	Boceto de Cobro de Sanción Por Usuario.	134
4.62	Boceto de Informe de Asistencia Por Reunión.	135
4.63	Boceto de Informe de Asistencia Por Usuario.	135
4.64	Boceto de Informe de Asistencia Por Estado.	136
4.65	Boceto de Informe Cobro de Asistencia Por Usuario.	136
4.66	Boceto de Informe Cobro de Asistencia Por Reunión.	137
4.67	Boceto de Informe Colaboración Por Reunión.	137

4.68	Boceto de Informe Colaboración Por Usuario.	138
4.69	Boceto de Informe Cobro de Colaboración Por Usuario.	138
4.70	Boceto de Informe Cobro de Colaboración Por Reunión.	139
4.71	Boceto de Informe General de Sanciones.	139
4.72	Boceto de Informe Cobro de Sanciones.	139
4.73	Boceto de Descuentos de Multas.	140
4.74	Boceto de Informe General de Descuentos.	140
4.75	Diagrama de datos: Tesorería.	147
4.76	Diseño de Interfaz-Registro de Colaboraciones.	148
4.77	Diseño de Interfaz-Registro de Sanciones.	148
4.78	Diseño de Interfaz-Cobro de Multas de Asistencia por Usuario. . .	149
4.79	Diseño de Interfaz-Cobro de Multas de Asistencia por Reunión. .	149
4.80	Diseño de Interfaz-Cobro de Colaboraciones por Usuario.	150
4.81	Diseño de Interfaz-Cobro de Colaboraciones por Reunión.	150
4.82	Diseño de Interfaz-Cobro de Sanciones por Usuario.	151
4.83	Diseño de Interfaz-Informe de Asistencia por Reunión.	151
4.84	Diseño de Interfaz-Informe de Asistencia por Usuario.	152
4.85	Diseño de Interfaz-Informe de Asistencia por Estado.	152
4.86	Diseño de Interfaz-Informe Cobro de Asistencia por Usuario. . . .	153
4.87	Diseño de Interfaz-Informe Cobro de Asistencia por Reunión. . . .	153
4.88	Diseño de Interfaz-Informe de Colaboración por Reunión.	154
4.89	Diseño de Interfaz-Informe Colaboración por Usuario.	154
4.90	Diseño de Interfaz-Informe Cobro de Colaboración por Usuario. .	155
4.91	Diseño de Interfaz-Informe Cobro de Colaboración por Reunión. .	155
4.92	Diseño de Interfaz-Informe de Sanciones.	156
4.93	Diseño de Interfaz-Informe Cobro de Sanciones.	156
4.94	Diseño de Interfaz-Registro de Descuentos.	157
4.95	Diseño de Interfaz-Informe De Descuentos.	157
4.96	Boceto de Documentos que Ingresan a la comunidad.	182
4.97	Boceto de Documentos que Salen de la comunidad.	183
4.98	Boceto de Descarga de Actas.	183
4.99	Boceto de Descarga de Documentos Recibidos.	183
4.100	Boceto de Descarga de Documentos Entregados.	184
4.101	Boceto de Egresos Con Factura.	184
4.102	Boceto de Egresos Sin Factura.	185
4.103	Boceto de Otros Ingresos.	185
4.104	Boceto de Informe General de Egresos Con Factura.	186

4.105	Boceto de Informe General de Egresos Sin Factura.	186
4.106	Boceto de Informe General de Otros Ingresos.	186
4.107	Boceto de Reporte General de Ingresos y Egresos.	187
4.108	Boceto de Reporte General de Valores Faltantes Por Cobrar.	187
4.109	Diagrama de datos: Documentos, Ingresos-Egresos y Reportes.	194
4.110	Diseño de Interfaz-Documentos Recibidos.	195
4.111	Diseño de Interfaz-Documentos Entregados.	195
4.112	Diseño de Interfaz-Descargar Actas.	196
4.113	Diseño de Interfaz-Descargar Documentos Recibidos.	196
4.114	Diseño de Interfaz-Descargar Documentos Entregados.	196
4.115	Diseño de Interfaz-Egresos con Factura.	197
4.116	Diseño de Interfaz-Egresos sin Factura.	197
4.117	Diseño de Interfaz-Otros Ingresos.	198
4.118	Diseño de Interfaz-Informe General de Egresos con Factura.	198
4.119	Diseño de Interfaz-Informe General de Egresos sin Factura.	198
4.120	Diseño de Interfaz-Informe General de Otros Ingresos.	199
4.121	Diseño de Interfaz-Reporte General de Ingresos y Egresos.	199
4.122	Diseño de Interfaz-Reporte General de Valores Faltantes Por Cobrar.	200
4.123	Boceto de Reporte de Usuarios Activos.	215
4.124	Boceto de Reporte de Usuarios No Activos.	215
4.125	Boceto de Reporte de Usuarios.	215
4.126	Boceto de Reporte de Directivos Activos.	216
4.127	Boceto de Reporte de Directivos Con Periodo Finalizado.	216
4.128	Boceto de Reporte de Reuniones Activas.	216
4.129	Boceto de Reporte de Reuniones Finalizadas.	217
4.130	Boceto de Reporte de Asistencia Por Reunión.	217
4.131	Boceto de Reporte de Asistencia Por Usuario.	217
4.132	Boceto de Reporte de Asistencia Por Estado.	218
4.133	Boceto de Reporte Registro de Colaboración Por Reunión.	218
4.134	Boceto de Reporte Registro de Colaboración Por Usuario.	219
4.135	Boceto de Reporte General de Sanciones.	219
4.136	Boceto de Reporte Cobro de Asistencia Por Usuario.	219
4.137	Boceto de Reporte Cobro de Asistencia Por Reunión.	220
4.138	Boceto de Reporte Cobro de Colaboración Por Usuario.	220
4.139	Boceto de Reporte Cobro de Colaboración Por Reunión.	220
4.140	Boceto de Reporte Cobro de Sanciones.	221
4.141	Boceto de Reporte General de Descuentos.	221

4.142	Boceto de Reporte de Egresos Sin Factura.	221
4.143	Boceto de Reporte de Egresos Con Factura.	222
4.144	Boceto de Reporte de Otros Ingresos.	222
4.145	Boceto de Reporte General de Documentos.	222
4.146	Diagrama de datos: Reportes de Directivos.	225
4.147	Diseño de Interfaz-Reporte de Usuarios Activos.	226
4.148	Diseño de Interfaz-Reporte de Usuarios No Activos.	226
4.149	Diseño de Interfaz-Reporte de Usuarios.	227
4.150	Diseño de Interfaz-Reporte de Directivos Activos.	227
4.151	Diseño de Interfaz-Reporte de Directivos Finalizados.	228
4.152	Diseño de Interfaz-Reporte de Reuniones Activas.	228
4.153	Diseño de Interfaz-Reporte de Reuniones Finalizadas.	229
4.154	Diseño de Interfaz-Reporte de Asistencia Por Reunión.	229
4.155	Diseño de Interfaz-Reporte de Asistencia Por Estado.	230
4.156	Diseño de Interfaz-Reporte de Asistencia Por Usuario.	230
4.157	Diseño de Interfaz-Reporte de Registro Colaboraciones Por Reunión.	231
4.158	Diseño de Interfaz-Reporte de Registro Colaboraciones Por Usuario.	231
4.159	Diseño de Interfaz-Reporte General de Sanciones.	232
4.160	Diseño de Interfaz-Reporte de Cobro Asistencia Por Usuario.	232
4.161	Diseño de Interfaz-Reporte de Cobro Asistencia Por Reunión.	233
4.162	Diseño de Interfaz-Reporte de Cobro Colaboraciones Por Usuario.	233
4.163	Diseño de Interfaz-Reporte de Cobro Colaboraciones Por Reunión.	234
4.164	Diseño de Interfaz-Reporte de Cobro Sanciones.	234
4.165	Diseño de Interfaz-Reporte General de Descuentos.	235
4.166	Diseño de Interfaz-Reporte General de Egresos Sin Factura.	235
4.167	Diseño de Interfaz-Reporte General de Egresos Con Factura.	235
4.168	Diseño de Interfaz-Reporte General de Otros Ingresos.	236
4.169	Diseño de Interfaz-Reporte General de Documentos.	236
A.1	Manual Usuario Abrir y Cerrar Sesión.	251
A.2	Manual Usuario Administrar Usuarios.	252
A.3	Manual Usuario Administrar Directivos.	253
A.4	Manual Usuario Administrar Reuniones.	254
A.5	Manual Usuario Administrar Control de Asistencias.	255
A.6	Manual Usuario Tesorería Registros.	256
A.7	Manual Usuario Tesorería Cobros.	257
A.8	Manual Usuario Tesorería Descuentos.	258
A.9	Manual Usuario Egresos.	259

A.10 Manual Usuarios Otros Ingresos.	260
A.11 Manual Usuarios Registro de Documentos.	260
A.12 Manual Usuarios Descargas de Documentos.	261
A.13 Manual Usuarios Reporte Anual.	262
A.14 Manual Usuarios Reporte General.	263
B.1 Cuaderno de Control de Asistencia y Cobros.	264
B.2 Archivero de Documentos.	265

Resumen Ejecutivo

La comunidad de Sanjaloma Bajo de la parroquia Salasaca no cuenta con un sistema informático que gestione los procesos administrativos y financieros, la comunidad registra cada uno de sus procesos de modo manual (cuadernos) sin ningún tipo de organización, estos procesos son efectuados por los Directivos, que a su vez son elegidos por los usuarios de la comunidad por un periodo de tiempo determinado, con el cargo de Presidente, Vicepresidente, Secretario(a), Tesorero(a) y Vocales. Para solucionar este problema se realizó un sistema web para la gestión de los procesos administrativos y financieros de la comunidad.

Para diseñar y desarrollar el sistema web se aplicó la metodología Extreme Programming (XP). Las herramientas informáticas que se aplicaron en el desarrollo son: Lenguaje de programación C Sharp(c#), Framework Asp.Net, IDE Visual Studio .Net (versión Community 2017) y el Sistema gestor de base de datos MySQL.

El sistema tiene como objetivo organizar y controlar las reuniones, las asistencias, las recaudaciones, los egresos y los documentos. Estos procesos administrativos y financieros son vitales para la comunidad, por tal razón fueron automatizados en el sistema.

Los Directivos utilizan el sistema mediante el cargo que poseen, de este modo se controla el manejo de cada proceso que el sistema ejecuta, para evitar problemas por el mal uso del sistema. De igual manera si se otorga un permiso adicional al cargo, el sistema registrará los datos del Directivo que utilice dicho permiso. El sistema genera reportes sobre la gestión de la comunidad, estos reportes visualizan la información que los Directivos necesitan analizar para la toma de decisiones. Los reportes también tienen la finalidad de explicar detalladamente a los usuarios de la comunidad, sus adeudos y pagas realizadas durante el periodo de los Directivos de turno. El sistema puede ser utilizado sin complicación debido a que la interfaz es sencilla de manejar y su uso es fácil de aprender.

Palabras clave: Procesos Administrativos, Procesos Financieros, Gestión de Procesos, Metodologías, Método de Observación, Sistema Web, Directivos.

Abstract

The community of Sanjaloma Bajo of the Salasaca parish does not have a computer system that manages the administrative and financial processes, the community records each of its processes manually (notebooks) without any organization, these processes are carried out by the Directors , which in turn are chosen by the users of the community for a certain period of time, with the position of President, Vice President, Secretary, Treasurer and Vocales. To solve this problem, a web system was created for the management of the administrative and financial processes of the community.

To design and develop the web system, the Extreme Programming (XP) methodology was applied. The computer tools that are applied in the development are: C Sharp programming language (c #), Asp.Net Framework, IDE Visual Studio .Net (Community 2017 version) and the MySQL database management system.

The system aims to organize and control meetings, attendance, collections, expenses and documents. These administrative and financial processes are vital for the community, which is why they were automated in the system.

The Directivos use the system through the charges they have, thus controlling the handling of each process that the system executed, to avoid problems due to the misuse of the system. In the same way, if an additional permit is granted to the position, the system records the data of the Director who uses said permit. The system generates reports on community management, these reports visualize the information that managers need to analyze in order to make decisions. The reports are also intended to explain specifically to the users of the community, their debts and payments made during the period of the Shift Directivos. The system can be used without complications because the interface is easy to operate and its use is easy to learn.

Keywords: Administrative Processes, Financial Processes, Process Management, Methodologies, Observation Method, Web System, Directivos.

Glosario de términos y acrónimos

- IS: Ingeniería de Software.
- INEC: Instituto Nacional de Estadística y Censos.
- GbP: Gestión basada en los Procesos.
- MDS: Metodologías de desarrollo de Software.
- ASP: Active Server Pages. En español sería Páginas del servidor activo.
- CLR: Common Language Runtime. En español sería Entorno en tiempo de ejecución de lenguaje común.
- XP: Extreme Programming. En español sería Programación Extrema.
- IDE: Entorno de desarrollo integrado.
- CRC: Clase-Responsabilidad-Colaboración.
- DDL: Data Definition Language. En español sería Lenguaje de definición de datos.
- DML: Data Manipulation Language. En español sería Lenguaje de Manipulación de Datos.
- SGBD: Sistemas de gestión de bases de datos.

INTRODUCCIÓN

El presente proyecto trata sobre el desarrollo e implementación de un sistema automatizado para la gestión administrativa y financiera de la comunidad de Sanjaloma Bajo de la parroquia Salasaca. Los sistemas automatizados controlan y organizan la información, optimizando los procesos de las empresas o instituciones, mejorando la calidad de atención al cliente.

La comunidad de Sanjaloma Bajo es una organización que registra cada uno de sus procesos administrativos y financieros de modo manual (cuadernos), como resultado de esta práctica los Directivos de la comunidad tienen varios problemas y estos son: análisis incompleto de información, control erróneo de asistencias, registro inadecuado de recaudaciones y documentos sin ninguna organización, por estos problemas los Directivos al final de su periodo no pueden informar con detalle cada proceso que realizaron, a los usuarios de la comunidad.

Ante los problemas planteados inicialmente, se busca desarrollar una solución que permita trasladar los procesos manuales a un sistema informático donde se mantenga la información actualizada y disponible, por esto se plantea el proyecto de investigación denominado “SISTEMA AUTOMATIZADO PARA LA GESTIÓN DE LAS COMUNIDADES DE LA PARROQUIA SALASACA DE LA CIUDAD DE PELILEO”; dividido en los siguientes capítulos:

Capítulo 1: Se plantea el problema de investigación sobre los problemas que existen en las comunidades de la parroquia Salasaca, con la respectiva justificación y los objetivos que se cumplirán a lo largo del proyecto, con la finalidad de dar solución al problema.

Capítulo 2: Se manifiestan los antecedentes investigativos referentes a aplicaciones web o sistemas web que gestionan procesos de una organización o institución, respaldados por un fundamento teórica, aportando al planteamiento de una propuesta para dar solución al problema.

Capítulo 3: Se presentan los diferentes tipos de modalidades de investigación a desarrollarse, especificando el método de recolección de información que se utilizara para el desarrollo del presente proyecto, de igual manera se presenta

un listado de las diferentes actividades necesarias para cumplir con los objetivos planteados.

Capítulo 4: Se presenta el desarrollo de la propuesta, con la planificación de las actividades a realizar y el desarrollo de cada una de ellas, para la elección de la metodología del desarrollo de software se utilizara un cuadro comparativo.

Capítulo 5: En este capítulo se establecerá las conclusiones y recomendaciones a las que se ha llegado, después de haber finalizado todo el proyecto.

CAPÍTULO 1

El problema

1.1. Tema de investigación

SISTEMA AUTOMATIZADO PARA LA GESTIÓN DE LAS COMUNIDADES DE LA PARROQUIA SALASACA DE LA CIUDAD DE PELILEO.

1.2. Planteamiento del problema

Las tendencias tecnológicas aportan significativamente al rendimiento de los recursos informáticos, debido a esto el desarrollo de software con el pasar del tiempo ha tomado importancia dentro de las instituciones, promoviendo la utilización de sistemas informáticos que mejoran la calidad de sus procesos. Tanto instituciones públicas como privadas realizan procesos que involucran información y documentación, por lo tanto, es necesario llevar una adecuada gestión administrativa y financiera de los procesos que se realizan todos los días [1][2].

En el Ecuador el desarrollo de software ha surgido de modo acelerado, esto está ligado a que las instituciones implementan sistemas gestores de información y documentación que mejoran el funcionamiento y la optimización de los procesos. Estas instituciones son de carácter empresarial, educativa, públicas o privadas que trabajan con grandes cantidades de información y documentación, implicando utilizar sistemas informáticos que gestionen adecuadamente, de este modo las instituciones podrán realizar búsquedas de la información con facilidad y rapidez, mejorando el tiempo de atención y la calidad del servicio al cliente [3][4].

Las comunidades del pueblo Salasaca no cuentan con una adecuada gestión de los procesos que administran las reuniones, recaudaciones, contabilidad, registro de asistencia y documentación, provocando que la información se duplique y que se extravíe información, afectando de este modo en la toma de decisiones y en el control de los valores monetarios de las comunidades, es decir la gestión de los

procesos administrativos y financieros que se realizan en las comunidades, son de suma importancia para los Directivos al igual que como para los usuarios. Los Directivos buscan la manera eficiente y rentable de gestionar sus procesos y hacer la administración más flexible.

La información financiera de la comunidad Sanjaloma Bajo son registrados manualmente, esta información se genera por los egresos y por los ingresos de recaudaciones de multas, colaboraciones y sanciones, debido a esto la información registrada es susceptible a extraviarse, este problema afectaría a la contabilidad dando valores erróneos, provocando inconsistencias de capital, es decir la comunidad no tiene un respaldo correcto de la información obtenida de sus procesos financieros. Por la inadecuada gestión financiera los Directivos tendrán graves problemas al generar informes, que justifiquen los ingresos y los egresos.

El registro de asistencia de los usuarios no es llevado con un control adecuado, esto causa problemas al registrar atrasos y ausencias que corresponden a las reuniones y mingas, que se realizan dependiendo de las circunstancias por las que se convocaron. Los Directivos tienen muchos conflictos con los usuarios de la comunidad debido a que no pueden explicar con detalla los atrasos y las ausencias que se registraron de modo manual (cuaderno).

La tecnología mejora los procesos y facilita el trabajo de las personas, pero no son utilizados en algunos lugares; este es el caso de la comunidad de Sanjaloma Bajo, debido a que no tiene un sistema que gestione los procesos administrativos y financieros, obstaculiza la búsqueda de información que los usuarios desean de sus registros. Por estas razones es necesario que se implemente en la comunidad un sistema automatizado que gestione los procesos, para realizar consultas personalizadas y que brinde una manera eficiente de acceder con facilidad a toda la información y documentación de la comunidad.

1.3. Delimitación

- Área Académica: Software.

- Línea de Investigación: Desarrollo de Software.
- Sublínea de investigación: Aplicaciones Web.
- Delimitación espacial: La presente investigación se desarrollará en las comunidades de la parroquia Salasaca de la ciudad de Pelileo.
- Delimitación temporal: La presente investigación se desarrolló a partir del 09 de noviembre de 2018 hasta el 25 de noviembre de 2019.

1.4. Justificación

La importancia de esta investigación se fundamenta en solucionar los problemas que provoca la inadecuada gestión de procesos de la comunidad de Sanjaloma Bajo de la parroquia Salasaca, por tal razón es necesario plantear una alternativa sustentable y escalable, que gestione cada proceso administrativo y financiero, por tanto se ha visto la necesidad de implementar un sistema que automatice los procesos que administren las reuniones, las recaudaciones, la contabilidad, el registro de la asistencia y los documentos de la comunidad.

El presente proyecto pretende ser un aporte significativo a la gestión de los procesos administrativos y financieros de la comunidad. Promoviendo el uso de herramientas de software libre, con el objetivo de organizar y controlar la información de los procesos que se generan en la comunidad, permitiendo a los Directivos que gestionan la comunidad contar con información confiable, rápida y segura.

Este proyecto se puede adaptar a las necesidades de las comunidades del pueblo Salasaca, debido a que mantiene los principios, métodos y conceptos en los que se basan los sistemas automatizados que gestionan los procesos.

Adicionalmente, existe el conocimiento necesario para realizar el presente proyecto, aprovechando la información y recursos disponibles que ofrece la Internet, también el financiamiento del estudio será por parte del investigador.

1.5. Objetivos

1.5.1. General

Implementar un Sistema Automatizado para la Gestión de las comunidades de la parroquia Salasaca de la ciudad de Pelileo.

1.5.2. Específicos

1. Analizar los procesos manuales de la comunidad para definir las necesidades y las prioridades de los usuarios del sistema.
2. Investigar las herramientas adecuadas para el desarrollo de la aplicación web.
3. Desarrollar el Sistema Informático para la Gestión de las comunidades de la parroquia Salasaca.

CAPÍTULO 2

Marco Teórico

2.1. Antecedentes investigativos

Revisando la información existente en la biblioteca de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, en referencia a Base de Datos se ha encontrado los siguientes trabajos:

Según el trabajo Presentado por Edison Javier Freire Vargas de título “sistema informático basado en software libre para la gestión documental en la facultad de ciencia e ingeniería en alimentos” (2014). Este menciona algunos aspectos para tener en cuenta en lo referente al sistema automatizado de gestión de documentos y menciona: “Para un mejor desempeño del Sistema Informático de Gestión Documental se recomienda a los usuarios que utilicen navegadores actualizados y de preferencia el navegador Google Chrome, además es importante habilitar la opción de soporte para JavaScript en el navegador.” “Se recomienda al Jefe de Sistemas que, en su calidad de administrador del Sistema Informático de Gestión Documental, designe a una persona especializada en el área de sistemas para garantizar la integridad y veracidad de los datos” [3].

Según el trabajo Presentado por Miguel Ángel Remache Chicaiza de título “sistema automatizado de gestión de documentos en el área administrativa en la unidad educativa Luis a. Martínez de la ciudad de Ambato, provincia del Tungurahua” (2014). Este menciona algunos aspectos para tener en cuenta en lo referente al sistema automatizado de gestión de documentos y menciona: “Utilizar metodologías que guíen y ejecuten correctamente las actividades para tomar decisiones y lograr el producto informático final deseado” [5].

2.2. Fundamentación filosófica

Para la presente investigación se ha utilizado el paradigma Crítico-Propositivo, misma que permitirá estudiar, analizar e interpretar la realidad que se vive en las

comunidades de la parroquia Salasaca, con la finalidad de mejorar el control de los procesos administrativos y financieros, esto permitirá ejecutar una propuesta que ayude a solucionar los problemas que se manifiestan en las comunidades. El paradigma crítico realizara un análisis crítico del problema y el paradigma propositivo proponera una solución factible. El método de la observación permite realizar registro de datos de lo observado, esto ayuda a producir descripciones de calidad. Los registros de datos obtenidos se producen sobre una realidad, y esto destaca que el escrito obtenido por la observación sea confiable, debido a que la información que se registró tendrá las características necesarias para el análisis de los procesos que se realizan en el lugar investigado. El investigador responde sus cuestionamientos mediante lo observado con sentido de indagación [6].

2.3. Fundamentación teórica

2.3.1. Gestión de la Calidad

La Gestión de la Calidad es un conjunto de acciones planificados y sistematizados, que son necesarios para que el producto o servicio satisfaga los requisitos de la calidad. La gestión de la calidad permite documentar procesos, procedimientos y asignaciones de responsabilidades, para lograr alcanzar los objetivos planteados por la calidad. Este tipo de gestión permite coordinar y dirigir las actividades de una organización, para alcanzar las normativas con la finalidad de mejorar la efectividad y la eficiencia. Existen estándares internacionales que especifican los requerimientos para la gestión de calidad [7].

2.3.2. Gestión Documental

La Gestión Documental controla el procedimiento de entrada y salida de la documentación en las Instituciones o Empresas, este proceso es necesario para tramitar cada documento tanto físico como magnético, con la finalidad de administrar de forma organizada los documentos. El conjunto de técnicas que ofrece la gestión documental tiene como objetivo administrar los documentos producidos y recibidos por los usuarios, para facilitar su utilización y conservación [8].

2.3.3. Funciones Administrativas

Las funciones administrativas son procesos que explican las diversas actividades que se realizan en una Institución o Empresa, involucrando una serie de procedimientos que son: planificación, organización, dirección y control, esto se

debe cumplir de modo eficaz y eficiente, para obtener con detalles cada una de las actividades administrativas que se ejecutan en una organización [9].

2.3.4. Contabilidad Administrativa

La contabilidad administrativa es una herramienta que mejora la capacidad de un ejecutivo para tomar decisiones económicas eficientes, debido a que la contabilidad informa con detalle los ingresos y egresos que se produjeron en la Institución o Empresa, es importante tomar en cuenta que al momento que se genera información contable útil, ayudara a la toma de decisiones económicas para el crecimiento de la Empresa [10].

2.3.5. Gestión por Procesos

La Gestión de Procesos, se orienta a desarrollar la organización, para satisfacer las expectativas de los usuarios que conforman un negocio. La estructura organizativa que aplica un modelo de gestión de procesos, es el resultado de la experiencia de las organizaciones que han evolucionado, debido a que utilizaron técnicas o métodos para que cada proceso se ejecute de manera sincronizada y directa, con el objetivo de aumentar la productividad en las Instituciones o Empresas [11].

2.3.6. Gestión Administrativa

Toda empresa o institución lleva a cabo una serie de procesos que son realizados diariamente, registrados y tramitados en la documentación administrativa. La gestión administrativa controla que los procesos se efectúen de manera eficaz y eficiente, pero para obtener resultados óptimos la gestión debe cumplir estos consejos: el proceso debe ser oportuno, el proceso debe minimizar los costes y el proceso debe ser riguroso y seguro [12].

2.3.7. Metodologías de desarrollo de Software

Metodología de desarrollo de software es el estudio de métodos que determinan la manera optima de desarrollar software, organizando las etapas con sus respectivos procesos para obtener un producto eficiente. Actualmente el término desarrollo es el más utilizado para referirse a las actividades que involucran la creación, fabricación, actualización o modificación de software [13].

2.3.8. Metodologías Ágiles

Las empresas operan en un entorno global y están susceptibles a cambios, por esto deben responder a nuevas oportunidades y mercados, debido al cambio de las condiciones económicas y surgimiento de productos y servicios competitivos. Es por estas razones que es necesario emplear sistemas informáticos, por lo que el software es partícipe en las operaciones empresariales, por lo tanto, se debe desarrollar de manera ágil, para responder con oportunidad y calidad las necesidades de las Instituciones o Empresas. La metodología ágil permite adaptar el modo de trabajo a las condiciones del proyecto, para desarrollar con rapidez y flexibilidad soluciones que estén acorde a las necesidades de los usuarios o clientes [13].

2.3.9. Metodologías Tradicionales

De acuerdo con Pressman (2010), las metodologías tradicionales son llamadas también modelos de proceso prescriptivo, y fueron propuestas originalmente para poner orden en el caos del desarrollo de software que existía cuando se empezó a generar masivamente. La historia indica que estos modelos tradicionales, propuestos en la década de los 60, han dado cierta estructura útil al trabajo de ingeniería de software y constituyen un mapa razonablemente eficaz para los equipos de software [13].

2.3.10. Aplicaciones Web

Las aplicaciones web se ejecuta en Internet, para que la información que se trabaja en la aplicación se almacene permanentemente en lo servidores que se encuentran vinculados a la web, con el objetivo de que los datos que requirieren los usuarios en ese momento sean enviados a sus dispositivos o equipos, dejando una copia temporal dentro del equipo. Estos servidores web que proporcionan el servicio de alojamiento, se encuentran en todo el mundo, por esta razón el servicio es gratuito en la mayoría de los casos y es extremadamente seguro [14].

2.3.11. ASP.NET

ASP.NET es un marco de desarrollo de aplicación web del lado del servidor de código abierto diseñado para el desarrollo de páginas dinámicas. Fue desarrollado por Microsoft para permitir a los programadores crear aplicaciones web, sitios web dinámicos y servicios web. Fue lanzado por primera vez en enero de 2002 con la versión 1.0 de .NET Framework, y es el sucesor de la tecnología ASP (Active

Server Pages) de Microsoft. ASP.NET se basa en Common Language Runtime (CLR), lo que permite a los programadores escribir código ASP.NET utilizando cualquier lenguaje .NET compatible [15].

2.4. Propuesta de solución

En este proyecto se plantea el desarrollo de un sistema web, para automatizar la organización y control de la información de la comunidad de Sanjaloma Bajo de la parroquia Salasaca. Promoviendo el uso de herramientas informáticas, con el objetivo de mejorar la gestión de los procesos administrativos y financieros de la comunidad.

CAPÍTULO 3

Metodología

3.1. Modalidad de la investigación

El presente trabajo tiene las siguientes modalidades: Campo, Bibliográfica o Documental y Aplicada. Las Metodologías ágiles permitirán que el proyecto se adapte a cambios no previstos durante la etapa de desarrollo.

3.1.1. Modalidad de Campo

La Investigación es de campo porque se realizará los estudios y análisis necesarios de las características del problema, en el lugar en donde se está desarrollando y se relacionará directamente con las personas interesadas, y de esta manera formular la propuesta que solucionará la sistematización de los procesos administrativos y financieros de la comunidad de Sanjaloma Bajo.

3.1.2. Modalidad Bibliográfica o Documental

La investigación es bibliográfica porque permite ampliar y profundizar diferentes enfoques, teorías, conceptualizaciones y criterios de diversos autores sobre el problema en estudio, basándose en documentos referentes a base de datos y sistemas automatizados.

3.1.3. Modalidad Aplicada

Por la utilización de los conocimientos adquiridos a lo largo de la carrera universitaria.

3.2. Población y muestra

Por la naturaleza del proyecto no requiere población y muestra.

3.3. Recolección de información

Para la presente investigación se usará la técnica de observación que consiste en observar fenómenos aleatorios, con el fin de obtener información que será

necesaria para la investigación, y se trabajará con la información obtenida durante la observación de los procesos que realizan cada directivo en la comunidad de Sanjaloma Bajo.

3.4. Procesamiento y análisis de datos

Para la investigación de los procesos administrativos y financieros de la comunidad de Sanjaloma Bajo de la parroquia de Salasaca, se utilizará el método de la observación, seguidamente se procederá a realizar el análisis de las herramientas informáticas y de la metodología ágil que se aplicará en el desarrollo del sistema, para cumplir con los objetivos.

El primer paso para comenzar con la observación es planificar reuniones con los Directivos para clasificar los procesos que se realizan dentro de la comunidad. Se planifica y organiza reuniones con un horario acorde a cada uno de los Directivos, para observar detalladamente como ejecutan de inicio a fin los procesos, registrando cada dato de información para su respectivo análisis. Este procedimiento servirá para encontrar las necesidades que los Directivos y Usuarios tienen en la comunidad, también se observará los problemas causados por la inadecuada gestión de los procesos administrativos y financieros.

El análisis de la metodología ágil y de las herramientas informáticas se lo realizará con cuadros comparativos, para establecer cuales de ellas son las indicadas para el desarrollo del sistema web.

3.5. Desarrollo del proyecto

Se llevará a cabo las siguientes actividades para cumplir con los objetivos del proyecto de investigación, también se investigará y analizará una metodología para desarrollar el sistema web.

1. Identificar y Definir las necesidades y los requerimientos.
2. Análisis del Gestor de Base de Datos.
3. Análisis del Lenguaje de Programación.
4. Análisis del Entorno de Desarrollo Integrado (IDE).
5. Análisis de la Metodología de Desarrollo de Software.
6. Diseño de la Base de Datos.
7. Diseño de Interfaz.

8. Codificación.

9. Pruebas.

10. Documentación.

CAPÍTULO 4

Desarrollo de la propuesta

4.1. Analizar los procesos manuales de la comunidad para definir las necesidades

4.1.1. Identificar y Definir las necesidades y los requerimientos

Mediante el método de la observación se realizó una recopilación de información y se obtuvo lo siguiente: La Comunidad Sanjaloma Bajo tiene varios procesos administrativos y financieros que son realizadas por los Directivos de modo manual (cuadernos), esto produce inconvenientes al momento de analizar, localizar y organizar la información. Otro de los problemas es la incorrecta forma de archivar los documentos que ingresan y salen de la comunidad. La Comunidad tiene 5 vocales, pero también pueden ser de 4 a 11 vocales, dependiendo de las resoluciones en cuanto a elección de vocales para la comunidad, en cada comunidad puede seleccionar varios vocales, pero ese tema solo se resuelve en la reunión de elecciones de vocales. De los vocales seleccionados se elegirán Presidente, Vicepresidente, Secretario(a), Tesorero(a) y Vocales principales. Los Directivos deben constatar que han cumplido su cargo durante su periodo, para controlar que todos los usuarios puedan ser Directivos por un periodo de tiempo y así fomentar la participación de los usuarios en la comunidad.

El Presidente convoca a los usuarios a Reuniones (Generales, Ordinarias, Extraordinarias, Directivos y Mingas), pero la convocatoria se lo realiza mediante parlante en donde informa a la comuna la fecha para realizar la reunión, cada reunión tendrá su propia acta y en cada reunión se pasará lista sin excepción alguna. Las actas son textualizadas y firmadas por el Presidente y Secretario(a) en un libro de actas. Los certificados, oficios y solicitudes que salen e ingresan a la comunidad son archivados en un folder sin organización.

El control de asistencia es registrado por el Secretario(a), pero en casos especiales al no poder asistir a la reunión el Presidente asignara a un Directivo de turno para realizar el control de asistencia. El control de asistencia se debe realizar dos veces en la reunión, la primera al inicio y la segunda al final, en casos especiales en una reunión puede realizar más de dos veces el control de asistencia. Si el usuario no pasa la lista se lo tomará como Ausente, si el usuario pasa la lista

solamente al final de la reunión se considerará como Atrasado y si el usuario pasa la lista completamente se considerará como Presente. El valor de cobro de la Ausencia y Atraso será asignada y aprobada por los usuarios de la comunidad en la misma reunión. Las colaboraciones y Sanciones son designados y aprobados por los usuarios de la comunidad en una reunión.

El Tesorero(a) registra las recaudaciones en cualquier momento y da constancia de sus registros con documentos o con un informe mensual, sobre los valores monetarios de ingresos y egresos. Los egresos deben constar con facturas o sin facturas, pero se debe registrar la explicación del gasto. El Tesorero(a) puede realizar al final del año un descuento en las multas, este descuento es aprobado por los usuarios de la comunidad en una reunión. Los Directivos sancionan a uno o varios usuarios con un valor monetario y una explicación del motivo de sanción, pero debe ser aprobado en una reunión.

El Vicepresidente puede convocar a reuniones de igual manera que el Presidente, pero las convocatorias se lo realizan mediante el parlante. Puede observar las recaudaciones, control de asistencias y la documentación.

Los Vocales pueden observar las recaudaciones, los egresos y los documentos.

4.2. Investigar las herramientas adecuadas para el desarrollo de la aplicación web

4.2.1. Análisis del Gestor de Base de Datos a Aplicar

El Gestor de Base de Datos que se va a aplicar en este proyecto será analizado mediante la comparación de ellos. Los Gestores de Base de Datos son varios, pero para este proyecto se plantearán cuatro de ellos que se muestran a continuación:

BASE DE DATOS	VENTAJAS	DESVENTAJAS
---------------	----------	-------------

MySql	MySQL software es Open Source. Bajo costo en requerimientos para la elaboración de bases de datos. Velocidad al realizar las operaciones, con mejor rendimiento. Consume menos recursos en el servidor. Soporta un gran volumen de datos sin problemas [16].	Un gran porcentaje de las utilidades de MySQL no están documentadas. No es intuitivo, como otros programas [16].
SQLServer	Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente [17].	Es la enorme cantidad de memoria RAM que utiliza para la instalación y utilización del software. Al pertenecer a Microsoft es necesario adquirir una licencia para el uso [17].
Oracle	Permite almacenar y manipular información que puede ser digitalizada (representada) por Objetos, proporciona una estructura flexible con acceso ágil, rápido con gran capacidad de modificación [18].	Elevado precio para adquirir [18].
PostgreSQL	Maneja sin problema grandes cantidades de datos, además de soportar una alta concurrencia de usuarios [19].	Es lento en inserciones y actualizaciones. No tiene foros oficiales para soporte online. Consume más recursos en el servidor. Existen comandos no comprensibles [19].

Tabla 4.2: Gestores de Base de Datos.

En el siguiente cuadro se muestra las principales características de los cuatro Gestores de Base de Datos, mediante este cuadro se analizará y se decidirá el adecuado gestor de base de datos para el desarrollo del sistema propuesto. A continuación, el cuadro comparativo:

	MySQL	Oracle	SQL Server	PostgreSQL
Tipo de licencia	GPL o Uso comercial	Privativa	Microsoft EULA	License
Multiplataforma	SI	SI	NO	SI
Capacidad de Almacenamiento	1 GB	Ilimitado	2 GB	Ilimitado
Transacciones	SI	SI	SI	SI
Seguridad	SI	SI	SI	SI
Respaldos	SI	SI	SI	SI
Auditoria de Información	SI	SI	SI	SI

Tabla 4.3: Cuadro Comparativo - Gestores de Base de Datos.

Elaborado por: Oscar Curichumbi.

Análisis Elección de Base de Datos Tras analizar las características de cada gestor de base de datos en la Tabla 4.2 y Tabla 4.3, se escoge el gestor de base de datos MySQL, por mantener licencia de tipo open source, que es uno de los indicadores relevantes para el desarrollo del sistema propuesto, adicional a eso se elige el gestor de base de datos MySQL porque permite la construcción, de una manera más sencilla y es eficiente en cuanto a la instalación y configuración, cuenta con velocidad al realizar las operaciones o transacciones, y soporta un gran volumen de datos sin problemas; el gestor de base de datos MySQL se adapta a los requisitos para el desarrollo del Sistema Automatizado para la Gestión de las comunidades de la parroquia Salasaca de la ciudad de Pelileo.

4.2.2. Análisis del Lenguaje de Programación a Aplicar

Existen varios lenguajes de programación, pero fueron escogidos cinco de ellos, porque facilitara el respectivo análisis, a continuación, el cuadro de comparación:

Ventajas	Desventajas
----------	-------------

C++	Es potente en cuanto a lo que se refiere a creación de sistemas complejos y este lenguaje es muy robusto [20].	No es atractivo visualmente. No soporta la creación de páginas web [20].
Visual Basic	Permite usar con facilidad la plataforma de los sistemas Windows. Fácil de encontrar información, documentación y fuentes para los proyectos. Fácilmente extensible mediante librerías [20].	Problema de versionado asociado con varias librerías DLL. Pobre soporte para programación orientada a objetos. Incapacidad para crear aplicaciones multihilo, sin tener que recurrir a llamadas del api de Windows [20].
C#	Se puede definir una o más clases dentro de un mismo espacio de nombres. Rango más amplio y definido de tipos de datos. Atributo de acceso del tipo público, protegido, interno y privado [20].	Se tiene que conseguir una versión reciente de Visual Studio .net. Debe contar con Windows 7 o superior, tener alrededor de 4 gigas de espacio libre para la pura instalación, etc [20].
JAVA	Se pueden realizar aplicaciones como un procesador de palabras, una hoja que sirva para cálculos, una aplicación gráfica, etc. Se puede realizar soluciones empresariales en un entorno web. Soporta el desarrollo de aplicaciones móviles [21].	Esperar la actualización siguiente para que sea más rápido [21].
PHP	Es un lenguaje multiplataforma. Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos [22].	Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código, pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución [22].

Tabla 4.5: Lenguajes de Programación.

En el siguiente cuadro se muestra las principales características de los cinco lenguajes de programación, mediante este cuadro se analizará y se decidirá el adecuado lenguaje de programación para el desarrollo del sistema propuesto. A continuación, el cuadro comparativo:

	C++	VISUAL BASIC	C#	JAVA	PHP
Código libre	NO	NO	NO	SI	SI
Rapidez en la ejecución	SI	REGULAR	SI	SI	SI
Seguridad	SI	NO	SI	SI	SI
Desarrollo rápido de aplicaciones	NO	NO	SI	SI	NO
Multiplataforma	SI	NO	NO	SI	SI
Lenguaje orientado a objetos	SI	SI	SI	SI	REGULAR
Soporte y documentación	REGULAR	NO	REGULAR	SI	SI
Manejo de excepciones	SI	SI	SI	SI	SI
Soporte a aplicaciones móviles	NO	NO	SI	SI	REGULAR
Multihilo	SI	SI	SI	SI	NO

Tabla 4.6: Cuadro Comparativo - Lenguajes de Programación.

Análisis Elección de Lenguaje de Programación Todos los lenguajes de programación tienen ventajas y desventajas al momento de desarrollar sistemas, para decidir el indicado se toma en cuenta que cada uno tiene su propia forma de programar, por esta razón se requiere estudiar todo el lenguaje de programación y esto requiere tiempo, pero el tesista se acomoda al lenguaje de programación C Sharp(c#) por la experiencia y practica en dicho lenguaje, por lo tanto se utilizara este lenguaje de programación para el desarrollarlo del Sistema Automatizado para la Gestión de las comunidades de la parroquia Salasaca.

4.2.3. Análisis del Entorno de Desarrollo Integrado (IDE) a Aplicar

Para el desarrollo del sistema propuesto se necesita de un Entorno de Desarrollo Integrado (IDE), se realiza un análisis para elegir el IDE que se utilizara en el desarrollo del sistema web, a continuación, el cuadro de comparación:

Microsoft Visual Studio	NetBeans	Eclipse
Visual Studio es un IDE para cualquier tipo de desarrollador. La productividad de los desarrolladores mejora. Crea apps para Windows, iOS o Android. La instalación de Visual Studio es más sencilla y contiene las características que se necesitan. Tiempo de arranque más rápido. Para modificar y depurar el código no es necesario cargar cada proyecto. Las notificaciones son en tiempo real. Desarrollo de aplicaciones móviles. Publica rápidamente los archivos y los ajustes de configuración desde las máquinas de desarrollo hasta el sitio final implementado [23].	Integración de MySQL más ajustada y una mejor manera de compartir librerías entre proyectos dependientes. El aclamado soporte para RUBY/JRUBY ha mejorado. Menor consumo de memoria. Es utilizada para desarrollar cualquier aplicación. Es inestable, lento y pesado. El código no se puede manipular al cien por ciento. Son sistemas multiplataforma para 32 Y 64 bits en Windows, Linux y Mac os. Mejor soporte para PHP. Permite importar proyectos de Eclipse y otros IDE's. NetBeans trae por defecto un editor visual para crear pantallas de forma sencilla e intuitiva [24].	Dispone de un editor de texto. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, test y refactorización. Plataforma ligera para componentes de software. Emplea módulos. Es fácil de usar, ligero y estable. Son sistemas multiplataforma para 32 Y 64 bits en Windows, Linux y Mac os. Mas rápido, más flexible y más plugins. Mejor soporte de desarrollo Android. Eclipse no trae por defecto un editor visual con el que se podrá crear pantallas de una forma muy sencilla e intuitiva, pero se le puede añadir como plugins [25].

Tabla 4.7: Cuadro de Comparativo - Entorno de Desarrollo Integrado

Análisis Elección de Entorno de Desarrollo Integrado (IDE) En la Tabla 4.6 se compara los Entorno de desarrollo integrado (IDE), con esta información se analizará el IDE con el que se trabajara para desarrollar el sistema web, pero como se ha escogido el lenguaje de programación C Sharp(c#), el mejor IDE para este lenguaje de programación es Microsoft Visual Studio Community, es un entorno de desarrollo integrado para sistemas operativos Windows, soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET,

F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web. Por estas razones se ha decidido elegir el IDE Microsoft Visual Studio Community, para el desarrollo del Sistema Automatizado para la Gestión de las comunidades de la parroquia Salasaca.

4.2.4. Análisis de la Metodología de Desarrollo de Software a Aplicar

Las metodologías ágiles y tradicionales son parte fundamental del mundo del desarrollo de software y aplicaciones web, estas metodologías mejoran la forma en que se desarrolla un proyecto, porque ayuda a la construcción de software o aplicación web con guías específicas para su elaboración y ejecución.

En las metodologías tradicionales se crea un solo proyecto, de estructura definida y grandes dimensiones. El proceso es rígido, secuencial, sin cambios, con una sola dirección y sin marcha atrás. Los requerimientos del proyecto son definidos una sola vez y para todo el proyecto, esto produce grandes plazos de planeación y poca comunicación con el cliente.

Las metodologías ágiles son flexibles, para que los proyectos se subdividan en proyectos más pequeños con características ordenadas. Cada proyecto se desarrolla independientemente y durante un periodo corto de tiempo, de entre dos y seis semanas. Los proyectos se desarrollan colaborativamente y se adaptan mejor a los cambios, y el producto del desarrollo entregado al cliente es mejorado frecuentemente porque hay una comunicación constante con el cliente.

Para el desarrollo de este proyecto se elegirá utilizar las metodologías ágiles porque son adaptables a los cambios de los requerimientos, la comunicación con el cliente es constante y el producto puede mejorarse frecuentemente.

Las metodologías ágiles se clasifican en varias, pero solamente mencionaremos la metodología XP y la Metodología Scrum para el respectivo análisis, porque son compatibles con el proyecto de desarrollo de la aplicación web. A continuación, se describe las características y las ventajas sobre estas metodologías, con una comparación entre ellas, para analizar y seleccionar la metodología más adecuada para el desarrollo de aplicación web [26].

Metodología Extreme Programming (XP) La metodología XP fue desarrollada por Kent Beck para guiar a equipos de trabajo pequeños, entre dos y diez programadores en ambientes de requerimientos cambiantes. De esta metodología lo más importante son las historias de usuario las cuales corresponden a una técnica de requisitos, las historias de usuarios son formatos en los cuales el cliente describe las características y funcionalidades que el sistema debe poseer. El

usuario o cliente define las historias de usuario y el desarrollador en base a ellas establece las características de la entrega, costos de implementación y número de interacciones que componen una entrega funcional.

La metodología XP se caracteriza en la programación en parejas, cada funcionalidad debe de ser desarrollada por dos programadores, todo el equipo debe obtener conocimiento y es necesario que las parejas deban cambiar con cierta frecuencia para que la información no sea solo de una persona, comienza con una o varias reuniones con el usuario o cliente, en las cuales se da claridad las necesidades del cliente a través de las historias de usuario, donde se tiene presente:

- Reuniones continuas y comunicación efectiva.
- Los requisitos pueden cambiar.
- Prioridades en los requisitos.
- Equipo con formación elevada y capacidad de aprender.
- Metodología basada en prueba y error.
- Grupo pequeño y muy integrado (máximo 12 personas).

Para terminar la metodología XP debe cumplir con una etapa muy importante las cuales son las pruebas de aceptación, una vez que se ha desarrollado una funcionalidad que se ha detallado en una historia de usuario, entra a pruebas por parte del usuario o cliente, para que de su aprobación o rechazo [27].

Metodología Scrum El nombre de la metodología Scrum proviene del deporte rugby, que es una formación requerida para la recuperación rápida del juego ante una infracción menor. La metodología Scrum fue diseñado de tal forma que logra la colaboración eficaz del equipo de desarrollo siguiendo un marco de trabajo, se define roles para generar una estructura de correcto funcionamiento y se emplea un conjunto de reglas.

La metodología Scrum define tres roles, los cuales son: El líder del proyecto, el dueño del producto y el equipo de desarrollo. El líder del proyecto asegura que el equipo cumpla las reglas y procesos de la metodología. El dueño del producto es el representante de los clientes que usan el software. El equipo de desarrollo es el grupo de profesionales encargados de convertir la lista de requerimientos en funcionalidades del software.

La metodología Scrum utiliza el Sprint como un elemento principal, que corresponde a una etapa de trabajo donde se desarrolla una versión utilizable del producto. Un Sprint está compuesto por los siguientes elementos: reunión de planeación del Sprint, reunión diaria, trabajo de desarrollo, revisión y retrospectiva del Sprint. Cada sprint es considerado como un proyecto individual. La metodología Scrum pretende conseguir la satisfacción de todo el personal que participa en el desarrollo, de esta forma se favorece el trabajo en equipo y la visibilidad del producto. Esta metodología interviene inmediatamente si se verifica algún problema oculto para evitar poner en peligro el proyecto. Los equipos se guían por su conocimiento y experiencia al desarrollar proyectos, los equipos no se guían por planes de proyectos formalmente definidos. La planificación se realiza sobre cortos espacios de tiempo lo que permite una retroalimentación que proporciona revisiones constantes y un ciclo de vida adaptable. Así, el desarrollo de productos se produce de forma incremental y con un control permanente del proceso que permite la mejora continua [27].

Comparación entre la Metodología Scrum y XP A continuación, se muestra un cuadro comparativo entre las metodologías Scrum y XP.

	XP	Scrum
Framework	Puede ser adaptable, mayor flexibilidad, dinámico y funcional [28][27].	Esta metodología se basa en la gestión y en el desarrollo de software, y su proceso es incremental e iterativo [28][27].
Revisión de las metodologías	Integrar como mínimo una vez al día, y realizar las pruebas sobre la totalidad del proceso [28][27].	Revisión diaria, se describen 3 cuestiones: 1. El trabajo realizado el día anterior. 2. El trabajo previsto a realizar. 3. Las cosas que pueden realizarse o impedimentos [28][27].

<p>Objetivos que cumplen las Metodologías</p>	<p>Dar prioridad a trabajos con resultado directo. Satisfacer al cliente. Trabajos realizados en grupo (mínimo dos personas). Variables en las que se va actuar: Coste, Tiempo, Calidad y Alcance. Fomenta la comunicación entre los clientes y los desarrolladores. Puede ser aplicada a cualquier lenguaje de programación. Permite ahorrar mucho tiempo y dinero [28][27].</p>	<p>Indicado para Proyectos en entornos complejos: Satisfacer al cliente. Trabajos realizados en grupo (mínimo tres personas). Obtener resultados inmediatos. Requisitos cambiantes. Los problemas se identifican por adelantado en las reuniones diarias y por lo tanto se pueden resolver rápidamente. Puede ser aplicada a cualquier lenguaje de programación [28][27].</p>
<p>Desarrollo de las metodologías</p>	<p>Su desarrollo es liviana y adaptable. Sus fases son: Planificación del proyecto. Diseño. Codificación. Pruebas [28][27].</p>	<p>Su desarrollo es simple, pero requiere trabajo duro. Su control es de forma empírica y adaptable a la evolución del proyecto [28][27].</p>
<p>Facilidad de Uso</p>	<p>Esta metodología es orientada para pequeños o medianos equipos. Para proyectos de riesgo: fecha de entrega. No apto para mucho personal. Requisitos con probabilidad de cambiar dependiendo del usuario o cliente [28][27].</p>	<p>No se basa en el seguimiento de un plan. Modelo adaptable. Construcción incremental basada en iteraciones. No existe trabajo con diseños o abstracciones [28][27].</p>

<p>Diferencias entre las metodologías</p>	<p>Las iteraciones de entrega son de una a tres semanas. Las tareas que se van terminando en las diferentes entregas al cliente son susceptibles a modificaciones durante el transcurso de todo el proyecto.</p> <p>Los miembros programan en parejas.</p> <p>El equipo de trabajo sigue estrictamente el orden de prioridad de las tareas marcadas por el cliente.</p> <p>Se centra en la programación o creación del producto [28][27].</p>	<p>Las iteraciones de entrega son de dos a cuatro semanas y se conocen como sprint.</p> <p>Al finalizar un sprint, las tareas que se han realizado del Sprint y en las que el dueño del producto ha mostrado su conformidad ya no se vuelven a tocar en ningún momento.</p> <p>Cada miembro del equipo trabaja individualmente.</p> <p>El equipo de trabajo puede modificar el orden del desarrollo de las tareas.</p> <p>Está más enfocado en la administración del proyecto [28][27].</p>
---	---	---

Tabla 4.9: Cuadro comparativo - Metodologías ágiles

Análisis Elección de la Metodología Se ha analizado mediante un cuadro comparativo las metodologías Scrum y XP con la finalidad de escoger y acoplar al proyecto de desarrollo web la metodología más indicada. En la comparación de las metodologías Scrum y XP se puede observar los procesos de trabajo y las diferencias entre estas metodologías.

La metodología Scrum se enfoca más en la administración del proyecto, el proyecto se administrará por tareas y cada miembro del equipo trabajará individualmente con el fin de realizar cada tarea exitosamente, para que el cliente quede conforme con el producto para no volver a tocar la tarea o sprint en ningún momento, y esta metodología es apta para grupos grandes.

La metodología XP por lo contrario se centra en la programación o creación del producto, el cliente marcará el orden de prioridad de las tareas y el equipo de trabajo lo seguirá estrictamente, esta metodología no es apta para mucho personal. En esta metodología las tareas que se van entregando al cliente son susceptibles a modificaciones durante el transcurso del proyecto.

Para este proyecto de desarrollo de sistema web se ha escogido la metodología XP, por las características más favorables para el proyecto, encontradas en el cuadro comparativo de metodologías Ágiles (Tabla 4.8), esta metodología puede ayudar a planificar, coordinar y ordenar cada proceso del proyecto. La metodología XP es orientada para pequeños o medianos equipos, y principalmente esta metodología puede modificar las tareas ya entregadas al cliente, esto es primordial en un proyecto de desarrollo porque si hay algún error en una tarea ya entregada es necesario hacer la modificación correspondiente para resolver dicho problema, y así obtener un proyecto exitoso y funcional. Por las razones ya explicadas se ha escogido la metodología XP, para la elaboración del proyecto de desarrollo de sistema web.

4.2.5. Herramientas de la Metodología XP

4.2.5.1. Historias de Usuario

Las Historias de Usuario describen el comportamiento del sistema, estas descripciones son planteadas para solventar las necesidades del cliente, para realizar las historias del usuario se debe realizar reuniones con el cliente, con la finalidad de obtener resultados que cumplan todas las expectativas que desea el cliente en su sistema. Las historias de usuario son utilizadas para realizar estimaciones de tiempo y obtener una planificación de desarrollo, esto ayuda a disminuir costos y documentación, mediante estas historias de usuario se podrán crear las pruebas de aceptación. Las historias de usuario deben ser comprensibles para que los programadores puedan realizar cada una de ellas en el menor tiempo posible [29]. La Plantilla de las historias de usuario que se va a elaborar se muestra en la Tabla 4.9, y se detalla cada uno de sus componentes a continuación:

Historia de Usuario	
Numero de historia: Permite identificar a una historia de usuario.	Nombre de historia: Describe de manera general a una historia de usuario.
Usuario: Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.	
Prioridad en negocio: Grado de importancia que el cliente asigna a una historia de usuario.	Puntos Estimados: Es el tiempo estimado en días para concluir la historia de usuario. Se trabaja solo con días enteros; si una historia de usuario tiene un día y medio, inmediatamente se aproxima a dos días.
Riesgo en desarrollo: Valor de complejidad que una historia de usuario representa al equipo de desarrollo.	Iteración Asignada: Número de iteración, en que el cliente desea que se implemente una historia de usuario.
Programador responsable: Persona encargada de programar cada historia de usuario.	
Descripción: Información detallada de una historia de usuario.	
Observaciones: Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario.	

Tabla 4.10: Plantilla - Historia de Usuario.

Elaborado por: Oscar Curichumbi.

4.2.5.2. Tareas de Ingenierías

Las Historias de Usuario se descomponen en algunas tareas de ingeniería, las cuales describen las actividades que se realizarán en cada uno de ellos, las tareas de ingeniería son realizadas y detalladas por desarrollador, esto permite que el desarrollador pueda darse una idea clara del código [29]. La Plantilla de las tareas de ingenierías que se va a elaborar se muestra en la Tabla 4.10, y se detalla cada uno de sus componentes a continuación:

Tarea de Ingeniería	
Numero de Tarea: Permite identificar a una tarea de ingeniería	Numero de Historia: Número asignado de la historia correspondiente
Nombre de Tarea: Describe de manera general a una tarea de ingeniería.	
Tipo de Tarea: Tipo al que corresponde la tarea de ingeniería.	Puntos Estimados: Número de días que se necesitará para el desarrollo de una tarea de ingeniería.
Fecha Inicio: Fecha inicial para el desarrollo de la tarea de ingeniería.	Fecha Fin: Fecha Final para concluir el desarrollo de la tarea de ingeniería.
Programador Responsable: Persona encargada de programar la tarea de ingeniería.	
Descripción: Información detallada de la tarea de ingeniería.	

Tabla 4.11: Plantilla - Tareas de Ingenierías.

Elaborado por: Oscar Curichumbi.

4.2.5.3. Pruebas de Aceptación

Las Pruebas de aceptación verifican si la iteración se lo realizo con éxito, para poder continuar con el desarrollo de la siguiente iteración, estas pruebas ayudan a que el cliente pueda conocer el avance en el desarrollo del sistema, también ayuda a los programadores a verificar que falta por desarrollar. La prueba de aceptación permite al programador una retroalimentación, para desarrollar las próximas historias de usuario. Estas pruebas verifican si cada historia de usuario cumple con las funcionalidades que desea el cliente [29]. La Plantilla de la Prueba de Aceptación que se va a elaborar se muestra en la Tabla 4.11, y se detalla cada uno de sus componentes a continuación:

CASO DE PRUEBA	
Código: Número Único, permite identificar la prueba de aceptación.	Nº Historia de Usuario: Número único que identifica a la historia de usuario.
Historia de Usuario: Nombre que indica de manera general la descripción de la historia de usuario.	
Condiciones de Ejecución: Condiciones previas que deben cumplirse para realizar la prueba de aceptación.	
Entrada/Pasos de Ejecución: Pasos que siguen los usuarios para probar la funcionalidad de la historia de usuario.	
Resultado Esperado: Respuesta del sistema que el cliente espera, después de haber ejecutado una funcionalidad.	
Evaluación de la Prueba: Nivel de satisfacción del cliente sobre la respuesta del sistema. Los niveles son: Aprobada y No Aprobada.	

Tabla 4.12: Plantilla - Pruebas de Aceptación.

Elaborado por: Oscar Curichumbi.

4.2.5.4. Tarjetas CRC

Las Tarjetas CRC (Clase-Responsabilidades-Colaboradores), son una herramienta de la metodología XP que consiste en dibujar una tarjeta por cada clase u objeto y se divide en: Nombre de la Clase, Responsabilidades y Colaboradores [29]. La Plantilla de las Tarjetas CRC que se va a elaborar se muestra en la Tabla 4.12, y se detalla cada uno de sus componentes a continuación:

TARJETAS CRC	
Nombre de la Clase: Nombre de la clase al cual hace referencia la tarjeta.	
Responsabilidades: Atributos y operaciones de la clase.	Colaboradores: Clases que colaboran con la clase citada en la tarjeta.

Tabla 4.13: Plantilla - Tarjetas CRC.

Elaborado por: Oscar Curichumbi.

4.2.5.5. Fases de la Metodología XP

PLANEACIÓN

La Metodología XP realiza una planificación mediante reuniones continuas entre el cliente y los programadores que estarán involucrados en el proyecto, con la finalidad de recopilar información para junto con el cliente realizar las respectivas historias de usuarios. Una vez obtenidas estas historias de usuarios, los programadores analizarán y plantearán rápidamente el tiempo estimado para el desarrollo de cada una, la planificación se clasifica en:

- **Historias de Usuarios:** Las cuales son descritas por el cliente según sus necesidades, estas descripciones son cortas y deben explicar las funcionalidades que el sistema va a realizar.
- **Plan de Entrega:** El plan de entrega debe establecer que historias de usuario serán agrupadas, para conformar una entrega. Este cronograma será el resultado de las reuniones entre el cliente y los involucrados con el proyecto.
- **Plan de Iteraciones:** Cada Iteración debe contener una o varias historias de usuario, deben ser seleccionadas entre el cliente y los programadores, esto ayudara a que el proyecto se lo realice mediante un orden, con la finalidad de obtener resultados exitosos.
- **Reuniones Diarias de Seguimiento:** La finalidad de estas reuniones es la de mantener la comunicación entre el equipo, esto ayudara a solucionar problemas inmediatamente durante el desarrollo del proyecto [29].

DISEÑO

La Metodología XP propone que los diseños sean simples y claros. Para el diseño se debe tener en cuenta los siguientes conceptos:

- Simplicidad: Un diseño simple ayuda a que el programador lo desarrollo más rápidamente, y también que el cliente pueda manejarlo sin ninguna dificultad.
- Soluciones: Para explorar diferentes soluciones se pueden utilizar pequeños programas de prueba, que ayude a estimar el tiempo que necesitara para la implementación de una historia de usuario. De igual manera estos programas podrán ayudar a encontrar soluciones a problemas técnicos.
- Recodificación: Cuando se necesita recodificar el código es importante tener en cuenta que la metodología XP lo permite, y menciona que se debe tener en cuenta que, al momento de recodificar el código lo realice sin cambiar la funcionalidad, esto ayudara a que el sistema tenga un código más simple y entendible.
- Metáforas: La metodología XP sugiere que este concepto explique de manera sencilla el propósito del proyecto, con la finalidad de hacer comprender la estructura del proyecto de manera sencilla y clara al cliente [29].

CODIFICACIÓN

La metodología XP requiere que el cliente este disponible durante todo el proyecto. Esto ayudara a los programadores a desarrollar un proyecto sin complicaciones, con el cliente y el programador juntos podrán proporcionar las historias de usuario, con el objetivo de desarrollarlos fácilmente, y así lograr que las pruebas den resultados exitosos. La metodología XP propone que se desarrolle en grupos de dos programadores, ambos trabajaran en un mismo ordenador, este método ayudara a minimizan los errores y se lograra mejores diseños. El producto obtenido es de mejor calidad, pero cuando el desarrollo se lo realiza individualmente se obtiene un producto de baja calidad. Para evitar errores, el código se debe integrar solamente por una pareja de desarrolladores. No obstante, todo el equipo puede contribuir con nuevas ideas que ayuden a mejorar el desarrollo del proyecto. De igual manera una pareja de programadores puede cambiar el código cuando sea necesario, esto corregirá futuros problemas, y por último se debe llevar un ritmo constante y razonable, sin sobrecargar al equipo [29].

PRUEBAS

Todos los módulos deben realizar pruebas unitarias, estas pruebas deben ser definidas antes de realizar el código. Cuando se produce un error debe ser corregido inmediatamente, y para evitar errores similares se debe tener

precauciones que eviten que ocurra nuevamente, también se debe generar nuevas pruebas para verificar si el error fue solucionado.

Las Pruebas de Aceptación son creadas en base a las historias de usuarios, en cada iteración el cliente debe mencionar uno o varios escenarios, con la finalidad de comprobar si una historia de usuario fue implementada correctamente. Si los resultados de las pruebas fueron fallidas, se debe indicar una solución para arreglar dicho fallo. Para continuar con el desarrollo de la siguiente historia de usuario se debe obtener resultados exitosos a las pruebas de aceptación [29].

4.3. Desarrollar el Sistema Informático para la Gestión de las comunidades de la parroquia Salasaca

4.3.1. Planificación del Proyecto

En la Fase inicial de la metodología XP, se establece la comunicación continua entre el desarrollador y el cliente, para obtener principalmente los requisitos para desarrollar el sistema web. Además, se establecerá el alcance del proyecto y la fecha de entrega del sistema web, considerando la prioridad y el tiempo estimado para desarrollar cada historia de usuario.

4.3.2. Historias de Usuarios

Las Historias de Usuario son explicados en un lenguaje común, para que sean entendidas por los directivos y el desarrollador, esto ayudara a cumplir con los requerimientos de la aplicación web. Las Historias de Usuarios de la aplicación web para la gestión de la comunidad son las siguientes:

- Ingreso al sistema.
- Creación de cargos y permisos.
- Registro de Usuarios.
- Asignación de usuarios para la directiva de la comunidad.
- Generar reuniones.
- Control de asistencia.
- Tesorería.
- Control de Entrada y Salida de documentos.

- Egresos y otros Ingresos.
- Reporte general de Ingresos y Egresos.
- Reportes del presidente o vicepresidente.
- Reportes del Secretario(a).
- Reportes de Tesorería.
- Reportes de los Vocales.

A continuación, se muestran las historias de usuarios, las cuales fueron utilizadas para llevar a cabo el desarrollo del sistema web.

Historia de Usuario 1	
Numero de historia: 1	Nombre de historia: Ingreso al sistema.
Usuario: Directivos de la comunidad	
Prioridad en negocio: Alta	Puntos Estimados: 6
Riesgo en desarrollo: Medio	Iteración Asignada: 1
Programador responsable: Oscar Curichumbi	
Descripción: Los directivos para ingresar al sistema tendrán un nombre de usuario y clave única con la que podrán ingresar.	
Observaciones: Los directivos deben estar registrados en la base de datos para poder otorgarles acceso al sistema.	

Tabla 4.14: Historia de Usuario: Ingreso al sistema.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 2	
Numero de historia: 2	Nombre de historia: Creación de cargos y permisos.
Usuario: Administrador o Presidente	
Prioridad en negocio: Alta	Puntos Estimados: 8
Riesgo en desarrollo: Alta	Iteración Asignada: 1
Programador responsable: Oscar Curichumbi	
Descripción: La aplicación web permitirá al administrador o presidente asignar el cargo con su respectivo nombre de usuario y clave a los nuevos directivos, los nuevos directivos podrán cambiar su clave en cualquier momento. Solo el administrador podrá otorgar el acceso y manejo de pestañas correspondientes al cargo que tendrá cada directivo.	
Observaciones: El Administrador de la aplicación web será el único usuario que tendrá acceso general a todas las funcionalidades de la aplicación web.	

Tabla 4.15: Historia de Usuario: Creación de cargos y permisos.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 3	
Numero de historia: 3	Nombre de historia: Registro de Usuarios.
Usuario: Presidente	
Prioridad en negocio: Medio	Puntos Estimados: 7
Riesgo en desarrollo: Medio	Iteración Asignada: 2
Programador responsable: Oscar Curichumbi	
Descripción: El Presidente ingresara la cedula, nombres, apellidos, correo electrónico, numero de celular, número de teléfono si lo tiene, el tipo de sexo, fecha de ingreso a la comunidad, el estado del usuario si está o no activo en las actividades que realiza la comunidad y observaciones si es necesario, esta información de un nuevo usuario será registrado en la aplicación web para constatar su participación en la comunidad.	
Observaciones: El usuario si no es activo en la comunidad no se le tomará control de asistencia, también no tendrá asignado los cobros de multas, colaboraciones y sanciones.	

Tabla 4.16: Historia de Usuario: Registro de Usuarios.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 4	
Numero de historia: 4	Nombre de historia: Asignación de usuarios para la directiva de la comunidad.
Usuario: Administrador, Presidente	
Prioridad en negocio: Medio	Puntos Estimados: 7
Riesgo en desarrollo: Medio	Iteración Asignada: 2
Programador responsable: Oscar Curichumbi	
Descripción: El Administrador o Presidente ingresa a los nuevos directivos de la comuna, se especifica la fecha de ingreso y su cargo correspondiente: Presidente, Secretario(a), tesorero(a), Vicepresidente o Vocales, para que el directivo cumpla con su cargo en la comuna.	
Observaciones: EL Expresidente deberá ingresar un nuevo presidente y después finalizar su periodo y también finalizar el periodo de sus respectivos compañeros directivos.	

Tabla 4.17: Historia de Usuario: Asignación de usuarios para la directiva de la comunidad.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 6	
Numero de historia: 6	Nombre de historia: Control de asistencia.
Usuario: Secretario(a)	
Prioridad en negocio: Alta	Puntos Estimados: 14
Riesgo en desarrollo: Alta	Iteración Asignada: 3
Programador responsable: Oscar Curichumbi	
Descripción: Una vez que el señor presidente genera una reunión el secretario realiza el control de asistencia, el secretario registra el control de asistencia con la fecha en la que se realiza la reunión, para asignar a un usuario el estado de asistencia el secretario debe confirmar si el usuario se encuentra Presente, Atrasado o con Ausencia, si el estado del usuario consta con Ausencia o Atraso recibirá una multa. El secretario deberá dar informes a los usuarios sobre sus Atrasos o Ausencias, también deberá dar informes generales del control de asistencia.	
Observaciones: El control de asistencia debe ser realizada con precaución, para que al momento de ingresar en la base de datos no tenga problemas con el usuario, por mal ingreso de datos.	

Tabla 4.19: Historia de Usuario: Control de asistencia.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 5	
Numero de historia: 5	Nombre de historia: Generar reuniones.
Usuario: Presidente o vicepresidente	
Prioridad en negocio: Alta	Puntos Estimados: 7
Riesgo en desarrollo: Medio	Iteración Asignada: 2
Programador responsable: Oscar Curichumbi	
Descripción: El presidente debe generar una reunión donde explica el motivo de la reunión, la fecha en que se realiza la reunión, y si la reunión es General, Extraordinaria, Ordinaria, De Directivos o mingas. La reunión debe ser manifestada por el presidente para que el secretario realice el control de asistencia. Después de haber terminado la reunión, se guarda el Acta de la respectiva reunión con su número de acta.	
Observaciones: El vicepresidente puede generar las reuniones si el presidente no puede por motivos de calamidad doméstica.	

Tabla 4.18: Historia de Usuario: Generar reuniones.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 7	
Numero de historia: 7	Nombre de historia: Tesorería.
Usuario: Tesorero(a)	
Prioridad en negocio: Alta	Puntos Estimados: 56
Riesgo en desarrollo: Alta	Iteración Asignada: 4
Programador responsable: Oscar Curichumbi	
Descripción: El Tesorero(a) registrara las colaboraciones y sanciones, donde especificara el motivo de la colaboración o sanción junto con el valor monetario asignado. El tesorero(a) cobra multas de ausencias, multas de atrasos y sanciones, también se cobra las colaboraciones que son asignadas por un motivo específico. El cobro debe tener la fecha, el nombre de la persona que paga y el valor específico. Se debe generar informes de cobros para especificar como fueron hechos los respectivos cobros en la reunión. Los cobros se realizarán varias veces hasta que las multas y colaboraciones sean canceladas en su totalidad. Las multas de atrasos y ausencias pueden tener un valor de descuento especificando el motivo, pero las colaboraciones y sanciones no tienen descuento.	
Observaciones: EL descuento de los atrasos y ausencias solo se aplica al momento de realizar un cobro general, el valor de los atrasos y ausencias se acumulan en un solo valor, para posteriormente ser cobrados con el respectivo descuento, si es aprobado en una reunión.	

Tabla 4.20: Historia de Usuario: Tesorería.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 8	
Numero de historia: 8	Nombre de historia: Control de Entrada y Salida de documentos.
Usuario: Directivos de la comunidad	
Prioridad en negocio: media	Puntos Estimados: 14
Riesgo en desarrollo: media	Iteración Asignada: 5
Programador responsable: Oscar Curichumbi	
Descripción: El directivo escanea los documentos que ingresan o salen de la comunidad, el directivo registra la fecha, título del documento, el tipo de documento que se está ingresando o saliendo de la comunidad, y especifica si el documento está ingresando o saliendo de la comunidad.	
Observaciones: Todos los directivos pueden ingresar o entregar documentos si es necesario para el beneficio de la comunidad.	

Tabla 4.21: Historia de Usuario: Control de Entrada y Salida de documentos.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 9	
Numero de historia: 9	Nombre de historia: Egresos y otros Ingresos.
Usuario: Tesorero(a)	
Prioridad en negocio: media	Puntos Estimados: 7
Riesgo en desarrollo: media	Iteración Asignada: 5
Programador responsable: Oscar Curichumbi	
Descripción: Los egresos y otros Ingresos son realizados por el tesorero, en cada egreso se especifica el concepto de gasto, el valor de compra y la fecha en la que se hizo el egreso a excepción de los egresos que tienen factura, porque se especificara adicional el número de factura, la fecha de compra, la dirección de compra y el nombre del local donde se realizó la factura. Los otros ingresos son especificados con la misma información que se ingresa en los egresos sin factura.	
Observaciones: El Tesorero debe colocar el valor del egreso o ingreso exacto, para registrar correctamente en la base de datos.	

Tabla 4.22: Historia de Usuario: Egresos y Otros Ingresos.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 10	
Numero de historia: 10	Nombre de historia: Reporte general de Ingresos y Egresos.
Usuario: Tesorero(a)	
Prioridad en negocio: Alta	Puntos Estimados: 14
Riesgo en desarrollo: Alta	Iteración Asignada: 5
Programador responsable: Oscar Curichumbi	
Descripción: EL Tesorero necesita realizar informes mensuales o anuales de los valores monetarios, para informar a los usuarios el valor monetario de egresos y de los ingresos que se han realizado en el año. El informe también debe constar si la comunidad tiene saldo positivo o negativo, este informe dará a conocer si la comunidad puede o no realizar nuevos egresos.	
Observaciones: El Tesorero podrá mediante este informe explicar los valores monetarios de las colaboraciones, sanciones y multas, para informar a los usuarios de la comunidad los valores de ingresos y egreso entre fechas.	

Tabla 4.23: Historia de Usuario: Reporte general de Ingresos y Egresos.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 11	
Numero de historia: 11	Nombre de historia: Reportes del presidente o vicepresidente.
Usuario: Presidente o vicepresidente	
Prioridad en negocio: Medio	Puntos Estimados: 3
Riesgo en desarrollo: Medio	Iteración Asignada: 6
Programador responsable: Oscar Curichumbi	
Descripción: El Presidente debe realizar reportes de las reuniones que se han generado en el año, reportes de los documentos que ha ingresado o ha entregado. Los reportes de los Ingresos y Egresos que se ha hecho en el año el tesorero, reportes del registro de asistencia, reportes de multas de atraso o faltas, al igual que de las colaboraciones y sanciones.	
Observaciones: El vicepresidente puede realizar los mismos reportes del presidente.	

Tabla 4.24: Historia de Usuario: Reportes del presidente o vicepresidente.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 12	
Numero de historia: 12	Nombre de historia: Reportes del Secretario(a).
Usuario: Secretario(a)	
Prioridad en negocio: Medio	Puntos Estimados: 3
Riesgo en desarrollo: Medio	Iteración Asignada: 6
Programador responsable: Oscar Curichumbi	
Descripción: El Secretario(a) realiza reportes de los documentos que ha entregado o ha ingresado a la comunidad, reportes del control de asistencia y reportes de los cobros de las multas de atrasos o ausencias.	
Observaciones:	

Tabla 4.25: Historia de Usuario: Reportes del secretario(a).

Elaborado por: Oscar Curichumbi.

Historia de Usuario 13	
Numero de historia: 13	Nombre de historia: Reportes de Tesorería.
Usuario: Tesorero(a)	
Prioridad en negocio: Medio	Puntos Estimados: 5
Riesgo en desarrollo: Medio	Iteración Asignada: 6
Programador responsable: Oscar Curichumbi	
Descripción: El Tesorero realiza reportes de ingresos o egresos que se ha realizado en el año con sus respectivos conceptos, reportes de los cobros a usuarios y reportes del control de asistencia para verificar si las multas están bien asignadas. El reporte debe mencionar si un usuario con deuda a cancelado o no, esta información es necesaria en las reuniones para mantener actualizado a los usuarios sobre sus deudas y sobre sus pagos.	
Observaciones:	

Tabla 4.26: Historia de Usuario: Reportes de Tesorería.

Elaborado por: Oscar Curichumbi.

Historia de Usuario 14	
Numero de historia: 14	Nombre de historia: Reportes de los Vocales.
Usuario: Directivos	
Prioridad en negocio: Medio	Puntos Estimados: 3
Riesgo en desarrollo: Medio	Iteración Asignada: 6
Programador responsable: Oscar Curichumbi	
Descripción: Los vocales realizan reportes de los documentos que ha ingresado o entregado en la comunidad, reportes de las multas de atrasos o ausencias, colaboraciones y sanciones para informar a los usuarios que deben cancelar sus deudas en un tiempo predefinido, reportes de ingresos y egresos del tesorero para constatar si hay alguna irregularidad y reportes de control de asistencia para informar a los usuarios en que fechas se ha atrasado o no asistió a una reunión.	
Observaciones:	

Tabla 4.27: Historia de Usuario: Reportes de los Vocales.

Elaborado por: Oscar Curichumbi.

4.3.3. Equipos y Roles

En la Tabla 4.27, se muestra la asignación de los roles para el presente proyecto.

MIEMBRO	ROLES	METODOLOGÍA
Oscar Curichumbi	Programador	EXTREME PROGRAMMING (XP)
Directivos de la Comunidad	Cliente	
Oscar Curichumbi Ing. Rubén Nogales	Encargado de pruebas (TESTER)	
Ing. Rubén Nogales	Entrenador (COACH)	
Oscar Curichumbi Ing. Rubén Nogales	Encargado de seguimiento (TRACKER)	
Oscar Curichumbi	Consultor	

Tabla 4.28: Equipos y Roles.

Elaborado por: Oscar Curichumbi.

4.3.4. Planificación de las Entregas al Cliente

La planificación de las entregas al cliente es el resultado de haber establecido las historias de usuario, para continuar con la planificación se debe realizar una negociación con el cliente sobre los alcances de cada historia de usuario, se evalúa las prioridades, los riesgos y la estimación de cada uno de ellos. La planificación de las iteraciones se elaboró con el cliente y se muestra en la Tabla 4.28, las iteraciones fueron agrupadas con sus respectivas historias de usuario. En esta planificación se establece diez semanas de estimación para el desarrollo del proyecto, la fecha de inicio es el 15/04/2019 y la fecha para la culminación total del proyecto es el 22/09/2019. El cliente verifico y aprobó la planificación para la elaboración del sistema web.

ITERACION	N.- Historia	Fecha Inicio	Fecha Final	SEMANAS														
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	15/04/2019	20/04/2019	■														
	2	21/04/2019	28/04/2019	■	■													
2	3	29/04/2019	05/05/2019			■												
	4	06/05/2019	12/05/2019				■											
	5	13/05/2019	19/05/2019					■										
3	6	20/05/2019	02/06/2019					■	■									
4	7	03/06/2019	04/08/2019							■	■	■	■					
5	8	05/08/2019	18/08/2019									■	■	■				
	9	19/08/2019	25/08/2019												■			
	10	26/08/2019	08/09/2019													■		
6	11	09/09/2019	11/09/2019															■
	12	12/09/2019	14/09/2019															■
	13	15/09/2019	19/09/2019															■
	14	20/09/2019	22/09/2019															■

Tabla 4.29: Planificación de las Entregas al cliente.

Elaborado por: Oscar Curichumbi.

Las iteraciones fueron agrupadas de acuerdo a la complejidad en los componentes a desarrollar, los requerimientos del cliente son la base primordial para el desarrollo del sistema web.

4.3.5. Desarrollo de la Primera Iteración

Se realizará la descripción general de las tareas respectivas de las dos primeras historias de usuario, que pertenecen a la primera iteración. Como se indica en la planificación se desarrollará los módulos de ingreso al sistema web, creación de cargos y permisos para el control de pestañas según su cargo, que el directivo podrá manejar durante su periodo. En la Tabla 4.29 se muestra de manera general las tareas de ingeniería correspondientes a la historia de usuario número 1-2 para cumplir con lo planificado.

Numero de Tarea	Numero de Historia	Esfuerzo	Nombre de la Tarea
1	1	1	Diseño de la interfaz para el ingreso al Sistema.
2	1	1	Diseño de base de datos para los directivos.
3	1	2	Control de Usuario y/o Contraseña de directivos para determinar autorización de ingreso al sistema.
4	1	2	Modificar la contraseña de Usuario.
5	2	1	Diseño de la interfaz para registrar cargos.
6	2	1	Diseño de base de datos para el control de cargos.
7	2	2	Grabar Creación de cargos en la Base de Datos.
8	2	1	Diseño de la interfaz para el control de permisos sobre las pestañas que el directivo podrá manejar según su cargo.
9	2	1	Diseño de base de datos para el control sobre las pestañas que el directivo podrá manejar según su cargo.
10	2	2	Grabar los cambios de los permisos de las pestañas que el directivo podrá manejar según su cargo en la Base de Datos.

Tabla 4.30: Tareas a Realizar de las Historias de Usuario 1-2.

Elaborado por: Oscar Curichumbi.

4.3.5.1. Estimación de la Primera Iteración

A continuación, se muestra la estimación de la Primera Iteración:

ITERACIÓN 1				
N.- Historia	Fecha Inicio	Fecha Final	DÍAS	HORAS
1	15/04/2019	20/04/2019	6	24
2	21/04/2019	28/04/2019	8	88
TOTAL ESTIMACIÓN:			14	112

Tabla 4.31: Horas asignadas para el desarrollo de la Iteración 1.

Elaborado por: Oscar Curichumbi.

4.3.5.2. Bocetos de las Historias de Usuarios 1-2

En las siguientes figuras se muestran los bocetos de las Historias de Usuarios 1-2 que forman parte de la Primera Iteración.

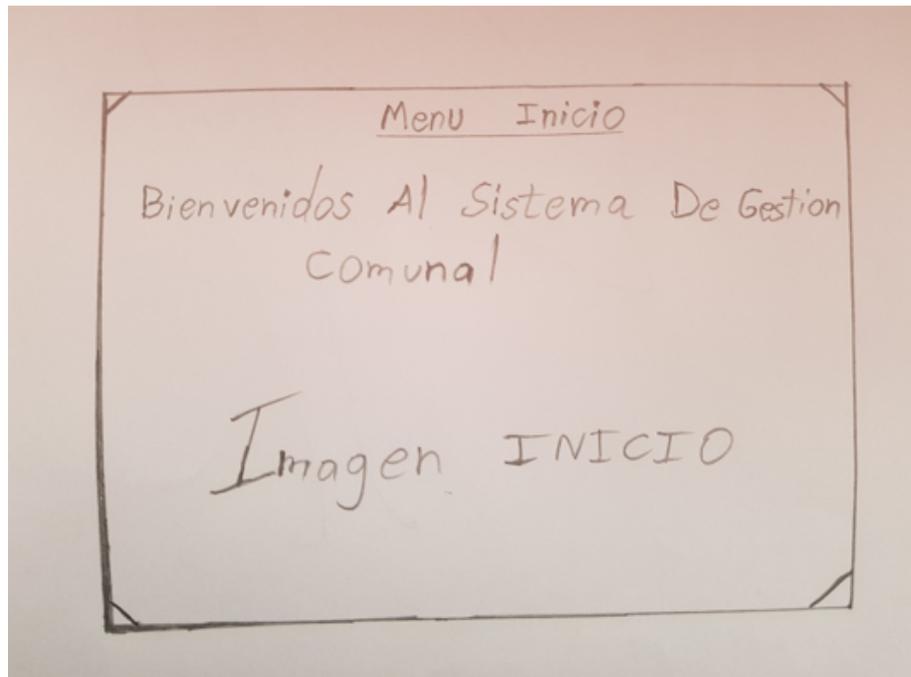


Figura 4.1: Bosquejo de Menú Inicio.
Elaborado por: Oscar Curichumbi.

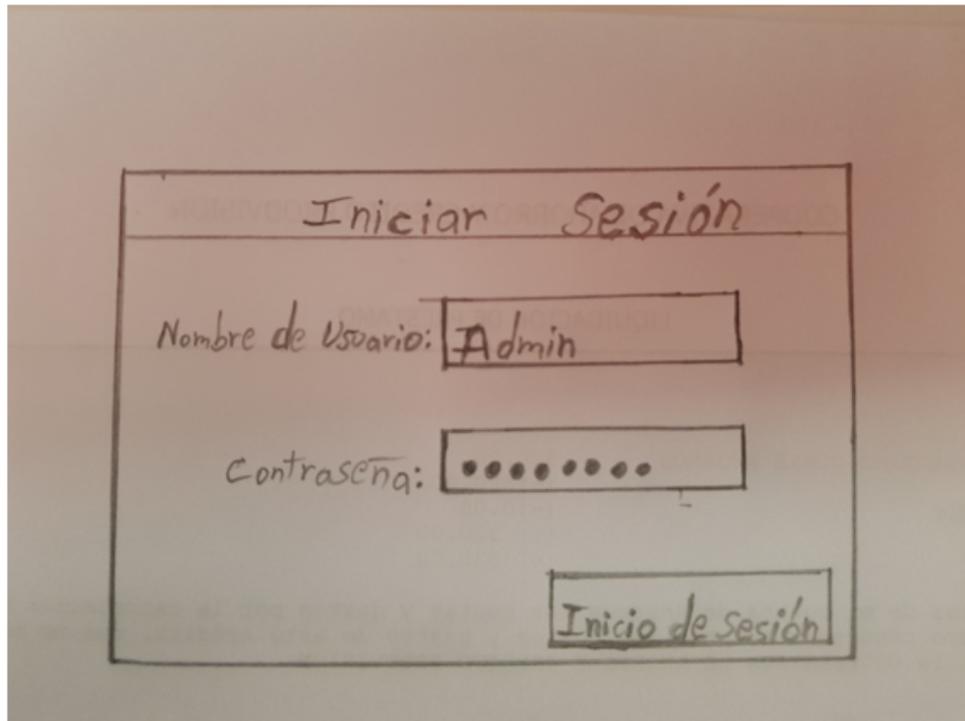


Figura 4.2: Bosquejo de Inicio de Sesión.
Elaborado por: Oscar Curichumbi.

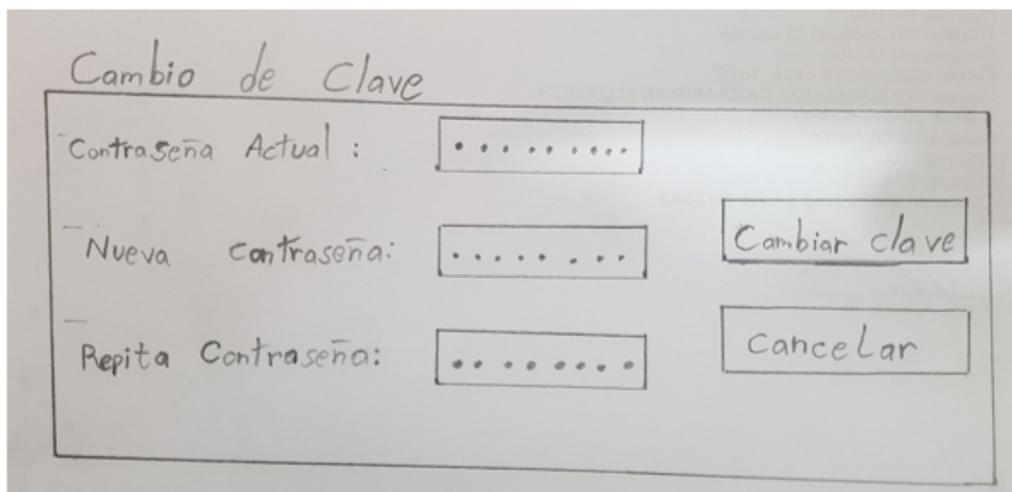


Figura 4.3: Bosquejo de Cambio de Clave.
Elaborado por: Oscar Curichumbi.

Agregar Cargo

Nombre Cargo:

Descripcion:

ID	Nombre de Cargo	Descripcion
datos	Datos	Datos

Figura 4.4: Bosquejo de Registrar Cargo.
Elaborado por: Oscar Curichumbi.

Permisos

Cargo:

Administración	OTORGADO	<input type="button" value="Cambiar"/>
Usuarios	NEGADO	<input type="button" value="Cambiar"/>
Directivos	OTORGADO	<input type="button" value="Cambiar"/>
Reuniones	NEGADO	<input type="button" value="Cambiar"/>
Asistencias	OTORGADO	<input type="button" value="Cambiar"/>
Tesoreria	NEGADO	<input type="button" value="Cambiar"/>

Figura 4.5: Bosquejo de Otorgar Permisos.
Elaborado por: Oscar Curichumbi.

Modificar Cargo

Ingrese ID cargo:	<input type="text"/>	<input type="button" value="Buscar"/>
Nombre Cargo :	<input type="text"/>	<input type="button" value="Modificar"/>
Descripcion :	<input type="text"/>	<input type="button" value="Cancelar"/>

ID	Nombre de Cargo	Descripcion
Datos	Datos	Datos

Figura 4.6: Bosquejo de Modificar Cargo.
Elaborado por: Oscar Curichumbi.

4.3.5.3. Descripción de las Tareas de Ingenierías

En base a la planificación se ha realizado las tareas de ingeniería, en estas se ha asignado el tiempo de estimación para el desarrollo de cada una de ellas. Si el cliente está satisfecho con el desarrollo de las tareas, que forman parte de las dos historias de usuario, esto cumplirá con las necesidades del cliente y de este modo se culminará la primera iteración. A continuación, se muestra las tareas de ingenierías de cada historia de usuario.

Tarea de Ingeniería	
Numero de Tarea: 1	Numero de Historia: 1
Nombre de Tarea: Diseño de la interfaz para el ingreso al Sistema.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 15/04/2019	Fecha Fin: 15/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la interfaz, donde los Directivos registraran su usuario y/o contraseña para ingresar al sistema.	

Tabla 4.32: Tarea 1 de la Historia de Usuario N.-1

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 2	Numero de Historia: 1
Nombre de Tarea: Diseño de base de datos para los directivos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 16/04/2019	Fecha Fin:16/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la base de datos, donde los Directivos registraran su usuario y/o contraseña para ingresar al sistema.	

Tabla 4.33: Tarea 2 de la Historia de Usuario N.-1

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 3	Numero de Historia: 1
Nombre de Tarea: Control de Usuario/Contraseña de directivos para determinar autorización de ingreso al sistema.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 17/04/2019	Fecha Fin: 18/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se realizará el control del ingreso al sistema mediante el usuario y la contraseña, para la comparación de datos se tomará los campos de la tabla directivos (usuario y/o contraseña). El directivo podrá cambiar su usuario y/o contraseña cuando lo desee. El cierre de sesión tendrá un factor muy importante en la seguridad, el nombre de usuario y contraseña se borrarán con el objetivo de evitar que el sistema vuelva a ingresar con los mismos datos del directivo, esto ayudara a que el directivo no tenga problemas de identificación en sus procesos correspondientes.	

Tabla 4.34: Tarea 3 de la Historia de Usuario N.-1

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 4	Numero de Historia: 1
Nombre de Tarea: Modificar la contraseña de Usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 19/04/2019	Fecha Fin: 20/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se realizará el código para modificar la contraseña del usuario. Si el usuario desea cambiar la contraseña debe ingresar la nueva contraseña y repetir la contraseña, para poder modificar correctamente la contraseña anterior.	

Tabla 4.35: Tarea 4 de la Historia de Usuario N.-1

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 5	Numero de Historia: 2
Nombre de Tarea: Diseño de la interfaz para registrar cargos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 21/04/2019	Fecha Fin: 21/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la interfaz, donde se ingresarán los cargos que tendrán los directivos.	

Tabla 4.36: Tarea 5 de la Historia de Usuario N.-2

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 6	Numero de Historia: 2
Nombre de Tarea: Diseño de base de datos para el control de cargos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 22/04/2019	Fecha Fin: 22/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la base de datos, donde se registrarán los cargos que tendrán los directivos.	

Tabla 4.37: Tarea 6 de la Historia de Usuario N.-2

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 7	Numero de Historia: 2
Nombre de Tarea: Grabar Creación de cargos en la Base de Datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 23/04/2019	Fecha Fin: 24/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se debe guardar en la base de datos la información de los nuevos cargos.	

Tabla 4.38: Tarea 7 de la Historia de Usuario N.-2

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 8	Numero de Historia: 2
Nombre de Tarea: Diseño de la interfaz para el control de permisos sobre las pestañas que el directivo podrá manejar según su cargo.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 25/04/2019	Fecha Fin: 25/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño la interfaz, donde se realizará el control de permisos sobre las pestañas que el directivo podrá manejar según su cargo.	

Tabla 4.39: Tarea 8 de la Historia de Usuario N.-2

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 9	Numero de Historia: 2
Nombre de Tarea: Diseño de base de datos para el control sobre las pestañas que el directivo podrá manejar según su cargo.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 26/04/2019	Fecha Fin: 26/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de base de datos para el control sobre las pestañas que el directivo podrá manejar según su cargo.	

Tabla 4.40: Tarea 9 de la Historia de Usuario N.-2

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 10	Numero de Historia: 2
Nombre de Tarea: Grabar los cambios de los permisos de las pestañas que el directivo podrá manejar según su cargo en la Base de Datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 27/04/2019	Fecha Fin: 28/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se debe guardar en la base de datos el cambio de los permisos sobre las pestañas que el directivo podrá manejar según su cargo en la Base de Datos.	

Tabla 4.41: Tarea 10 de la Historia de Usuario N.-2

Elaborado por: Oscar Curichumbi.

4.3.5.4. Tarjetas CRC: Historias de usuario 1-2

En las siguientes tablas se muestran las tarjetas CRC, estas tarjetas son de gran utilidad para la Primera Iteración.

Directivos	
Responsabilidad	Colaboración
Atributos:	Usuarios
Id_Directivos	Tipo_cargo
Id_Usuario	
Nombre_Usuario	
Contrasenia	
Fecha_Ingreso	
Fecha_Salida	
Id_Cargo	
Estado	
Observaciones	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.42: Tarjeta CRC: Directivos.

Elaborado por: Oscar Curichumbi.

Tipo_cargo	
Responsabilidad	Colaboración
Atributos:	
Id_Cargo	
Nombre_Cargo	
Descripcion_Cargo	
Confirmacion	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.43: Tarjeta CRC: Tipo Cargo.

Elaborado por: Oscar Curichumbi.

Cargo_menu	
Responsabilidad	Colaboración
Atributos:	Tipo_cargo
Id_Cargo_Menu	Menu
Id_Cargo	
Id_Menu	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.44: Tarjeta CRC: Cargo Menu.

Elaborado por: Oscar Curichumbi.

Menu	
Responsabilidad	Colaboración
Atributos:	Menu
Id_Menu	
Nombre	
Id_Padre	
Url	
Descripcion	
Métodos	
Buscar	

Tabla 4.45: Tarjeta CRC: Menu.

Elaborado por: Oscar Curichumbi.

4.3.5.5. Diagrama de datos

A continuación, se muestra el diagrama de datos de la Primera Iteración:

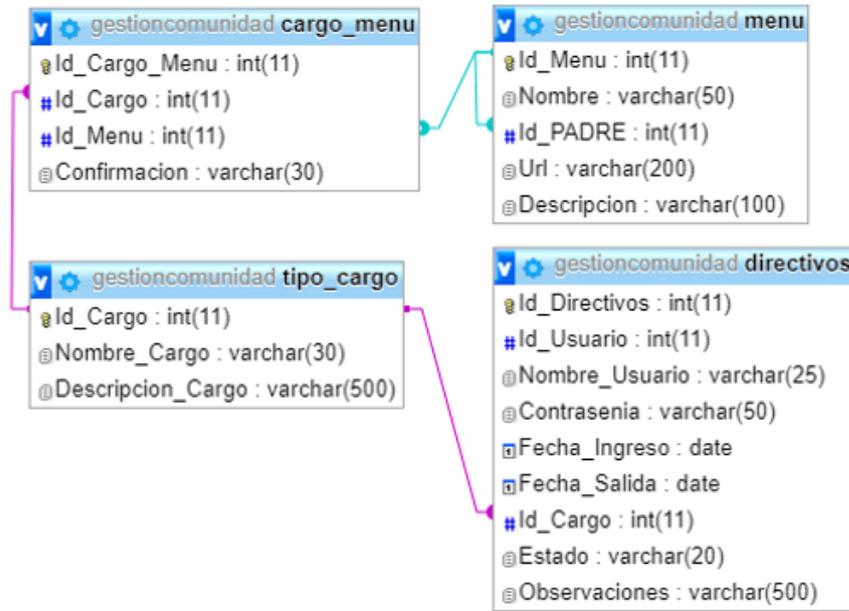


Figura 4.7: Diagrama de datos: Directivos, Cargos y Menús.
Elaborado por: Oscar Curichumbi.

4.3.5.6. Diseño de Interfaz

A continuación, se muestra los diseños de interfaz de la Primera Iteración:

La interfaz muestra un encabezado con el título 'GESTION COMUNAL' y un menú de navegación con las opciones: 'Pagina Principal', 'Administracion', 'Usuarios', 'Asistencia' y 'Cambio Clave'. El formulario de inicio de sesión, titulado 'Iniciar Sesión', contiene los siguientes elementos:

- Campo 'Nombre de Usuario:' con el valor 'admin' ingresado.
- Campo 'Contraseña:' con caracteres ocultos por puntos.
- Una casilla de verificación desactivada con el texto 'Recordármelo la próxima vez.'
- Un botón 'Inicio de sesión' ubicado en la parte inferior derecha del formulario.

Figura 4.8: Diseño de Interfaz - Ingreso al sistema.
Elaborado por: Oscar Curichumbi.



Figura 4.9: Diseño de Interfaz - Menú Inicio.
Elaborado por: Oscar Curichumbi.



Figura 4.10: Diseño de Interfaz - Cambiar Clave.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL

Pagina Principal | Administracion | Usuarios | Asistencia | Cambio Clave

AGREGAR CARGO

Cargo
 Modificar Cargo
 Permisos

Nombre Cargo:

Descripcion:

Agregar

Cancelar

ID	NOMBRE DE CARGO	DESCRIPCION
1	ADMINISTRADOR	ADMINISTRADOR

Figura 4.11: Diseño de Interfaz - Crear Cargos.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
 Cerrar sesión

Pagina Principal | Administracion | Usuarios | Asistencia | Cambio Clave

PERMISOS

Cargo
 Modificar Cargo
 Permisos

CARGO:

Verificar Permisos

Cancelar

MENU	PERMISOS	
Pagina Principal	OTORGADO	Cambiar
Administracion	OTORGADO	Cambiar
Usuarios	OTORGADO	Cambiar
Directivos	NEGADO	Cambiar
Reuniones	NEGADO	Cambiar
Asistencia	OTORGADO	Cambiar

Figura 4.12: Diseño de Interfaz - Otorgar Permisos.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

[Pagina Principal](#)
[Administracion](#)
[Usuarios](#)
[Asistencia](#)
[Cambio Clave](#)

MODIFICAR CARGO

INGRESE ID CARGO:

NOMBRE CARGO:

DESCRIPCION:

ID	NOMBRE DE CARGO	DESCRIPCION
1	ADMINISTRADOR	ADMINISTRADOR

Figura 4.13: Diseño de Interfaz - Modificar Cargo.
Elaborado por: Oscar Curichumbi.

4.3.5.7. Codificación

- Método login

Este método controla el ingreso y el error que se pueda realizar al ingresar: Usuario y la Contraseña.

```

1  protected void Login1_Authenticate(object sender ,
      AuthenticateEventArgs e)
2  {
3  try
4  {
5  if (AutenticacionDAL.Autenticar(Login1.UserName, Login1.
      Password)){
      FormsAuthentication.
      RedirectFromLoginPage(Login1.UserName, Login1.
      RememberMeSet);
      Response.Redirect("
      ~/Default.aspx");
6  }
7  }catch (Exception ex){
8  lblMensajes.Text = "La conexión ha Fallado. Compruebe que
      la base de datos este Activada." + ex.Message;
9  lblMensajes.Visible = true;
10 }
11 }

```

- Método Agregar Cargos

```
1  protected void btnAgregar_Click(object sender , EventArgs e)
    {
2  CargoDAL dal = new CargoDAL();
3  Cargo CARGOS = new Cargo();
4  if (txtNombreCargo.Text == ""){
5  lblMensajes.Text = "Ingrese Nombre de Cargo.";
6  lblMensajes.Visible = true;
7  }
8  else{
9  if (txtDescripcion.Text == ""){
10 lblMensajes.Text = "Ingrese Descripcion de Cargo.";
11 lblMensajes.Visible = true;
12 }
13 else{
14 try{
15 CARGOS.NombreCargo = txtNombreCargo.Text;
16 CARGOS.Descripcion = txtDescripcion.Text;
17 try{
18 dal.Insert(CARGOS);
19 lblMensajes.Text = "Registro Ingresado.";
20 lblMensajes.Visible = true;
21 cargarCargos();
22 limpiar();
23 }
24 catch
25 {
26 lblMensajes.Text = "Problema al ingresar cargo";
27 lblMensajes.Visible = true;
28 }
29 }catch (MySqlException excepcion){
30 Excepciones.Gestionar(excepcion , Plataforma.Web);
31 lblMensajes.Text = "Error de SQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
32 lblMensajes.Visible = true;
33 }
34 catch (Exception excepcion){
35 Excepciones.Gestionar(excepcion , Plataforma.Web);
36 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
```

```

        MensajePersonalizado;
37 lblMensajes.Visible = true;
38 } } } }

```

- Método Modificar Permisos

```

1  protected void gvCargarDatos_RowCommand(object sender,
        GridViewCommandEventArgs e){
2  CargoMenuDAL dal = new CargoMenuDAL();
3  CargoMenuV cargov = new CargoMenuV();
4  if (e.CommandName == "Select"){
5  int rowIndex = Convert.ToInt32(e.CommandArgument);
6  GridViewRow row = gvCargarDatos.Rows[rowIndex];
7  Label Id = row.FindControl("ID") as Label;
8  Label Permiso = row.FindControl("PERMISOS") as Label;
9  Button B = row.FindControl("bntCambiar") as Button;
10 if(Permiso.Text == "NEGADO"){
11 try{
12 cargov.Confirmacion = "OTORGADO";
13 int IdCargoMenu = Convert.ToInt32(Id.Text);
14 cargov.IdCargoMenu = IdCargoMenu;
15 bool correcto = dal.Update(cargov);
16 if (correcto){
17 lblMensajes.Text = "Se cambió el permiso.";
18 lblMensajes.Visible = true;
19 cargardatos();
20 }
21 else{
22 lblMensajes.Text = "Problemas al cambiar permisos";
23 lblMensajes.Visible = true;
24 }
25 }catch (MySqlException excepcion){
26 Excepciones.Gestionar(excepcion, Plataforma.Web);
27 lblMensajes.Text = "Error de SQL Server:" + "<br>" +
        Excepciones.MensajePersonalizado;
28 }
29 catch (Exception excepcion){
30 Excepciones.Gestionar(excepcion, Plataforma.Web);

```

```

31 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
    MensajePersonalizado;
32 }
33 }else{
34 if (Permiso.Text == "OTORGADO"){
35 try{
36 cargov.Confirmacion = "NEGADO";
37 int IdCargoMenu = Convert.ToInt32(Id.Text);
38 cargov.IdCargoMenu = IdCargoMenu;
39 bool correcto = dal.Update(cargov);
40 if (correcto){
41 lblMensajes.Text = "Se cambio el permiso.";
42 lblMensajes.Visible = true;
43 cargardatos();
44 }
45 else{
46 lblMensajes.Text = "Problemas al cambiar permisos";
47 lblMensajes.Visible = true;
48 }
49 }catch (MySqlException excepcion){
50 Excepciones.Gestionar(excepcion, Plataforma.Web);
51 lblMensajes.Text = "Error de SQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
52 }
53 catch (Exception excepcion){
54 Excepciones.Gestionar(excepcion, Plataforma.Web);
55 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
    MensajePersonalizado;
56 } } } } }

```

- Método Modificar Cargos

```

1 protected void btnModificar_Click(object sender, EventArgs
    e){
2 CargoDAL dal = new CargoDAL();
3 Cargo cargos = new Cargo();
4 if (txtIdCargo.Text == ""){
5 lblMensajes.Text = "Ingreso ID CARGO.";

```

```

6 lblMensajes.Visible = true;
7 }
8 else{
9 if (txtNombreCargo.Text == "" || txtDescripcion.Text == ""
    ){
10 lblMensajes.Text = "Para realizar la Actualizacion debe
    ingresar Nombre de cargo y la descripcion.";
11 lblMensajes.Visible = true;
12 }
13 else{
14 try{
15 cargos.NombreCargo = txtNombreCargo.Text;
16 cargos.Descripcion = txtDescripcion.Text;
17 int Idcargo = Convert.ToInt32(txtIdCargo.Text);
18 cargos.IdCargo = Idcargo;
19 bool correcto = dal.Update(cargos);
20 if (correcto){
21 lblMensajes.Text = "Registro actualizado";
22 lblMensajes.Visible = true;
23 cargarCargos();
24 limpiar();
25 }
26 else{
27 lblMensajes.Text = "Problemas al actualizar";
28 lblMensajes.Visible = true;
29 }
30 }catch (MySqlException excepcion){
31 Excepciones.Gestionar(excepcion, Plataforma.Web);
32 lblMensajes.Text = "Error de SQL Server:" + "<br>" +
    Excepciones.MensajePersonalizado;
33 }
34 catch (Exception excepcion){
35 Excepciones.Gestionar(excepcion, Plataforma.Web);
36 lblMensajes.Text = "Error de c#" + "<br>" + Excepciones.
    MensajePersonalizado;
37 } } } }

```

- Método Modificar Contraseña

```

1  protected void bntCambiarClave_Click(object sender ,
      EventArgs e){
2  DirectivosDAL dal = new DirectivosDAL();
3  Directivo clave = new Directivo();
4  if (txtNuevaContrasenia.Text == txtRepitaContrasenia.Text){
5  try{
6  clave.Contrasenia = txtNuevaContrasenia.Text;
7  clave.ContraseniaActual = txtContrasenia.Text;
8  int usuario = Convert.ToInt32(cbUsuario.SelectedItem.Value)
      ;
9  clave.IdDirectivo = usuario;
10 bool correcto = dal.UpdateContrasenia(clave);
11 if (correcto){
12 lblMensajes.Text = "Registro actualizado";
13 lblMensajes.Visible = true;
14 Limpiar();
15 }
16 else{
17 lblMensajes.Text = "Problemas al actualizar. Verifique
      Contraseña Actual.";
18 lblMensajes.Visible = true;
19 }
20 }catch (MySqlException excepcion){
21 Excepciones.Gestionar(excepcion , Plataforma.Web);
22 lblMensajes.Text = "Error de SQL Server:" + "<br>" +
      Excepciones.MensajePersonalizado;
23 }
24 catch (Exception excepcion){
25 Excepciones.Gestionar(excepcion , Plataforma.Web);
26 lblMensajes.Text = "Error de c#" + "<br>" + Excepciones.
      MensajePersonalizado;
27 }
28 }
29 else{
30 lblMensajes.Text = "La Nueva Contraseña no coincide con la
      Contraseña Repetida.";
31 lblMensajes.Visible = true;
32 }
33 }

```

4.3.5.8. Pruebas de Aceptación

A continuación, se muestra las pruebas de aceptación de la Primera Iteración:

CASO DE PRUEBA	
Código: 1	Nº Historia de Usuario: 1
Historia de Usuario: Ingreso al sistema.	
Condiciones de Ejecución: Cada directivo debe contar con un nombre de usuario y su contraseña para poder acceder a las funciones del sistema.	
Entrada/Pasos de Ejecución: Dar clic en Inicio de sesión. Llenar el formulario inicio de sesión introduciendo su nombre de usuario y Contraseña. Luego pulsar el botón INICIO DE SESION	
Resultado Esperado: Acceso a las funcionalidades del sistema dependiendo del tipo de usuario y contraseña.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.46: Caso de Prueba: Ingreso al sistema.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 2	Nº Historia de Usuario: 2
Historia de Usuario: Creación de cargos y permisos.	
Condiciones de Ejecución: Para crear los cargos y permisos debe ingresar como administrador.	
Entrada/Pasos de Ejecución: Crear Cargo y Otorgar Permisos Ingresar como administrador. Dar clic en la pestaña Administración y seleccionar la opción Cargo. Ingresar el nuevo cargo y dar clic en el botón Guardar. Luego Dar clic en la pestaña Permisos, para otorgar al nuevo Cargo los respectivos permisos. Modificar Cargo Seleccionar la pestaña Administración, la opción Modificar Cargo. Introducir el ID de Cargo para buscar la información que desea modificar. Modifique la información del Cargo que se ha ingresado incorrectamente en el sistema. Luego seleccionar la opción Modificar.	
Resultado Esperado: El directivo tendrá acceso mediante su cargo a las funcionalidades o pestañas de la aplicación web.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.47: Caso de Prueba: Creación de cargos y permisos.

Elaborado por: Oscar Curichumbi.

4.3.5.9. Resultados de la Primera Iteración

Como resultado de entrega de la primera iteración que consta de los módulos de Ingreso al sistema y creación de cargos y permisos, el cliente quedó satisfecho con las funcionalidades de los módulos que se desarrollaron, la primera iteración se ha concluido sin ningún problema manifestado por el cliente. A continuación, se realizará la segunda iteración.

4.3.6. Desarrollo de la Segunda Iteración

Se realizará la descripción general de las tareas respectivas de la tercera, cuarta y quinta historias de usuario, que pertenecen a la segunda iteración. Como se indica en la planificación se desarrollará los módulos de Registro de Usuarios, Asignación de usuarios para la directiva de la comunidad y Generación de reuniones. En la Tabla 4.47, se muestra de manera general las tareas de ingeniería correspondientes a las historias de usuario número 3-4-5 para cumplir con la planificación.

Numero de Tarea	Numero de Historia	Esfuerzo	Nombre de la Tarea
11	3	1	Diseño de la base de datos para los usuarios de la comunidad.
12	3	1	Diseño de la interfaz para el ingreso de usuario de la comunidad.
13	3	2	Validación de datos de los usuarios.
14	3	2	Guardar datos de los usuarios en la base de datos.
15	3	2	Modificar la información de la base de datos de los usuarios.
16	3	2	Eliminar la información de la base de datos de los usuarios.
17	4	1	Diseño y adaptación de la base de datos para los directivos.
18	4	1	Diseño de la interfaz de asignación de usuarios para directivos de la comunidad.
19	4	3	Guardar los datos de los directivos en la base de datos.
20	4	3	Modificar la información de la base de datos de los directivos.
21	5	1	Diseño de la base de datos para las Reuniones.
22	5	1	Diseño de la interfaz para Gestión de las Reuniones.
23	5	1	Validación de datos para registrar correctamente las reuniones.
24	5	3	Subir y Bajar actas del servidor.
25	5	2	Guardar la información de las Reuniones en la base de datos.
26	5	2	Modificar la información de la base de datos de las reuniones.

Tabla 4.48: Tareas a Realizar de las Historias de Usuario 3-4-5.

Elaborado por: Oscar Curichumbi.

4.3.6.1. Estimación de la Segunda Iteración

A continuación, se muestra la estimación de la Segunda Iteración:

ITERACIÓN 2				
N.- Historia	Fecha Inicio	Fecha Final	DÍAS	HORAS
3	29/04/2019	05/05/2019	7	56
4	06/05/2019	12/05/2019	7	56
5	13/05/2019	19/05/2019	7	56
TOTAL ESTIMACIÓN:			21	168

Tabla 4.49: Horas asignadas para el desarrollo de la Iteración 2.

Elaborado por: Oscar Curichumbi.

4.3.6.2. Bocetos de las Historias de Usuarios 3-4-5

En las siguientes figuras se muestran los bocetos de las Historias de Usuarios 3-4-5 que forman parte de la Segunda Iteración.

The image shows a hand-drawn wireframe for a 'Crear Usuarios' (Create Users) form and a data table. The form includes the following fields and controls:

- Cedula:** Input field with buttons for 'Verifica Cedula', 'Busca', and 'Cancelar'.
- Nombres:** Input field with an 'Agregar' button.
- Apellidos:** Input field with a 'Cancelar' button.
- E-mail:** Input field.
- Celular:** Input field.
- Telefono:** Input field.
- Sexo:** Dropdown menu with the text 'selecciona sexo' and a checkmark.
- Actividad:** Dropdown menu with the text 'S. Actividad' and a checkmark.
- Fecha:** Input field containing '01-08-2019'.

Below the form is a table with the following columns: ID, Cedula, Nombres, E-mail, celular, Telefono, sexo, and Fecha ingreso.

ID	Cedula	Nombres	E-mail	celular	Telefono	sexo	Fecha ingreso

Figura 4.14: Boceto de Crear Usuarios.

Elaborado por: Oscar Curichumbi.

Modificar Usuario

Ingresar cedula:

Nombres:

Apellidos:

E-mail:

celular:

Telefono:

Sexo:

Actividad:

observaciones:

ID	cedula	Nombres	E-mail	celular	Telefono	SEXO	Actividad

Figura 4.15: Boceto de Modificar Usuario.
Elaborado por: Oscar Curichumbi.

Eliminar Usuario

Ingresar cedula:

Nombres:

Apellidos:

E-mail:

celular:

Telefono:

sexo:

Actividad:

ID	cedula	Nombres	Email	celular	Telefono	SEXO	Activid.

Figura 4.16: Boceto de Eliminar Usuario.
Elaborado por: Oscar Curichumbi.

Ingresar Nuevo Directivo

Cedula:

Persona:

Nombre de usuario:

Contraseña:

Fecha Ingreso:

Cargo:

ID	Nombres	Nombre Usuario	Fecha Ingreso	Cargo	Estado

Figura 4.17: Boceto de Registrar Directivos.
Elaborado por: Oscar Curichumbi.

Cambio de Cargo

Directivos:

Cargo:

observaciones:

ID	Nombres	Nombre Usuario	Fecha Ingreso	Cargo

Figura 4.18: Boceto de Cambio de Cargo.
Elaborado por: Oscar Curichumbi.

Fin Periodo de Directivos

Directivos : Finalizar

Fecha Salida: Cancelar

observaciones:

ID	Nombres	Nombre Usuario	Fecha Ingreso	Cargo	Estado

Figura 4.19: Boceto de Periodo Finalizado.
Elaborado por: Oscar Curichumbi.

Generar Reunión

Tipo de Reunión: Generar

Motivo de Reunión: Cancelar

Fecha de Reunión:

Tipo de Reunión	Motivo Reunión	Fecha	Estado	Representante

Figura 4.20: Boceto de Generar Reuniones.
Elaborado por: Oscar Curichumbi.

Modificar Reuniones

ID Reunión:

Tipo Reunión:

Motivo Reunión:

Fecha Reunión:

Número Acta:

Documento:

observaciones:

ID	Tipo Reunión	Motivo Reunión	Fecha	Número Acta	Estado	Documento

Figura 4.21: Boceto de Modificar Reuniones.
Elaborado por: Oscar Curichumbi.

Subir Acta

Motivo Reunión:

Número Acta:

Documento:

Motivo Reunión	Fecha	Nº Acta	Documento

Figura 4.22: Boceto de Subir Acta.
Elaborado por: Oscar Curichumbi.

Modificar Acta

ID Reunión:

Documento:

Observaciones:

ID	Tipo Reunion	Motivo Reunion	Fecha	Nº Acta	Estado	Documento

Figura 4.23: Boceto de Modificar Acta.
Elaborado por: Oscar Curichumbi.

Finalizar Reunión

Motivo Reunión:

Número Acta:

Documento:

Tipo Reunión	Motivo Reunión	Fecha	Estado	Nº Acta	Documento

Figura 4.24: Boceto de Finalizar Reunión.
Elaborado por: Oscar Curichumbi.

Activar Reunión

ID Reunión:

Motivo Reunión:

Observaciones:

ID	Motivo Reunión	Fecha	Estado

Figura 4.25: Boceto de Activar Reunión.
Elaborado por: Oscar Curichumbi.

Informe General de Reuniones

ID	Tipo Reunión	Motivo Reunión	Fecha	Nº Acta	Estado	Documento	Presidente

Figura 4.26: Boceto de Informe General de Reunión.
Elaborado por: Oscar Curichumbi.

4.3.6.3. Descripción de las Tareas de Ingenierías

En base a la planificación se ha realizado las tareas de ingeniería, en estas se ha asignado el tiempo de estimación para el desarrollo de cada una de ellas. Si el cliente está satisfecho con el desarrollo de las tareas, que forman parte de las tres historias de usuario, esto cumplirá con las necesidades del cliente y de este modo se culminará la segunda iteración. A continuación, se muestra las tareas de ingenierías de cada historia de usuario.

Tarea de Ingeniería	
Numero de Tarea: 11	Numero de Historia: 3
Nombre de Tarea: Diseño de la base de datos para los usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 29/04/2019	Fecha Fin: 29/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se creará la tabla usuarios en la base de datos, para registrar la información de los usuarios de la comunidad.	

Tabla 4.50: Tarea 11 de la Historia de Usuario N.-3

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 12	Numero de Historia: 3
Nombre de Tarea: Diseño de la interfaz para el ingreso de usuario de la comunidad.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 30/04/2019	Fecha Fin: 30/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el presidente ingresará la información de los usuarios de la comunidad.	

Tabla 4.51: Tarea 12 de la Historia de Usuario N.-3

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 13	Numero de Historia: 3
Nombre de Tarea: Validación de datos de los usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 01/05/2019	Fecha Fin: 02/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere validar el número de cedula para continuar ingresando los datos del nuevo usuario de la comunidad, posterior a esto se validará los datos del nuevo usuario antes de ingresar a la base de datos, para que no ingrese datos nulos o datos erróneos.	

Tabla 4.52: Tarea 13 de la Historia de Usuario N.-3

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 14	Numero de Historia: 3
Nombre de Tarea: Guardar datos de los usuarios en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 02/05/2019	Fecha Fin: 03/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere guardar la información de los usuarios de la comunidad, para registrarlos en la base de datos.	

Tabla 4.53: Tarea 14 de la Historia de Usuario N.-3

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 15	Numero de Historia: 3
Nombre de Tarea: Modificar la información de la base de datos de los usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 03/05/2019	Fecha Fin: 04/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere modificar la información de los usuarios de la comunidad, para rectificar los datos mal ingresados en la base de datos. Para obtener la información a modificar se debe ingresar el número de cedula.	

Tabla 4.54: Tarea 15 de la Historia de Usuario N.-3

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 16	Numero de Historia: 3
Nombre de Tarea: Eliminar la información de la base de datos de los usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 04/05/2019	Fecha Fin: 05/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere eliminar la información del usuario que ya no pertenece a la comunidad. Para eliminar al usuario se debe ingresar el número de cedula.	

Tabla 4.55: Tarea 16 de la Historia de Usuario N.-3

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 17	Numero de Historia: 4
Nombre de Tarea: Diseño de la base de datos para los directivos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 06/05/2019	Fecha Fin: 06/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se creará la tabla directivos en la base de datos, para registrar la información de los nuevos directivos de la comunidad.	

Tabla 4.56: Tarea 17 de la Historia de Usuario N.-4

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 18	Numero de Historia: 4
Nombre de Tarea: Diseño de la interfaz de asignación de usuarios para directivos de la comunidad.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 07/05/2019	Fecha Fin: 07/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el Expresidente ingresará la información de los nuevos directivos que administraran la comunidad.	

Tabla 4.57: Tarea 18 de la Historia de Usuario N.-4

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 19	Numero de Historia: 4
Nombre de Tarea: Guardar los datos de los directivos en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 08/05/2019	Fecha Fin: 10/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere guardar la información de los nuevos directivos de la comunidad, para registrarlos en la base de datos. El presidente debe ingresar el número de cedula del usuario de la comunidad que ejercerá como directivo, el dato ingresado dará paso a visualizar los datos que se deben ingresar para guardar la información del nuevo directivo, el nombre de usuario debe ser único si por tal caso repite el sistema no dará paso a seguir con el proceso de guardar la información del nuevo directivo.	

Tabla 4.58: Tarea 19 de la Historia de Usuario N.-4

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 20	Numero de Historia: 4
Nombre de Tarea: Modificar la información de la base de datos de los directivos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 10/05/2019	Fecha Fin: 12/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere modificar la información de los nuevos directivos de la comunidad, para rectificar los datos mal ingresados en la base de datos. El presidente podrá modificar el cargo del directivo que tenga mal ingresado. El presidente podrá finalizar el periodo de los directivos asignando la fecha de finalización.	

Tabla 4.59: Tarea 20 de la Historia de Usuario N.-4

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 21	Numero de Historia: 5
Nombre de Tarea: Diseño de la base de datos para las Reuniones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 13/05/2019	Fecha Fin: 13/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se creará la tabla reuniones en la base de datos, para registrar la información de las reuniones que se realizaran en la comunidad.	

Tabla 4.60: Tarea 21 de la Historia de Usuario N.-5

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 22	Numero de Historia: 5
Nombre de Tarea: Diseño de la interfaz para Gestión de las Reuniones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 14/05/2019	Fecha Fin: 14/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el presidente ingresará la información de las reuniones que se realizaran en la comunidad.	

Tabla 4.61: Tarea 22 de la Historia de Usuario N.-5

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 23	Numero de Historia: 5
Nombre de Tarea: Validación de datos para registrar correctamente las reuniones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 15/05/2019	Fecha Fin: 15/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere validar los datos de las reuniones antes de ingresar a la base de datos, para no ingresar datos nulos o datos erróneos.	

Tabla 4.62: Tarea 23 de la Historia de Usuario N.-5

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 24	Numero de Historia: 5
Nombre de Tarea: Subir y Bajar actas del servidor.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 15/05/2019	Fecha Fin: 17/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Las actas que se generan al final de una reunión deben guardarse en una carpeta dentro del servidor, el acta debe ser subida al servidor y debe generar un link del documento para ingresar a la base de datos y así descargarlo cuando lo desee los directivos o los usuarios de la comunidad.	

Tabla 4.63: Tarea 24 de la Historia de Usuario N.-5

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 25	Numero de Historia: 5
Nombre de Tarea: Guardar la información de las Reuniones en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 17/05/2019	Fecha Fin: 18/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere guardar la información de las nuevas reuniones que se realizaran en la comunidad, para registrarlos en la base de datos.	

Tabla 4.64: Tarea 25 de la Historia de Usuario N.-5

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 26	Numero de Historia: 5
Nombre de Tarea: Modificar la información de la base de datos de las reuniones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 18/05/2019	Fecha Fin: 19/05/2019
Programador Responsable: Oscar Curichumbi	
<p>Descripción: Se requiere modificar la información de las nuevas reuniones que se realizaran en la comunidad, para rectificar los datos mal ingresados en la base de datos, los datos que se modificaran se podrán obtener mediante una consulta con el ID DE REUNION.</p> <p>Para finalizar la reunión el presidente seleccionara la reunión correspondiente y llenar los campos que son número de acta, link del documento y si es necesario las observaciones.</p> <p>Las reuniones deben reanudarse si el presidente finalizo la reunión por accidente, debe ingresar el ID DE REUNION que aparece en la ventana de consulta de reuniones finalizadas, con el dato ingresado el presidente podrá reanudar una reunión.</p>	

Tabla 4.65: Tarea 26 de la Historia de Usuario N.-5

Elaborado por: Oscar Curichumbi.

4.3.6.4. Tarjetas CRC: Historias de usuario 3-4-5

En las siguientes tablas se muestran las tarjetas CRC, estas tarjetas son de gran utilidad para la Segunda Iteración.

Usuarios	
Responsabilidad	Colaboración
Atributos:	
Id_Usuario	
Cedula	
Nombres	
Apellidos	
NoombresCompletos	
E_mail	
Celular	
Telefono	
Sexo	
Estado	
Fecha_Ingreso	
Observaciones	
Métodos	
Agregar	
Editar	
Eliminar	
Buscar	

Tabla 4.66: Tarjeta CRC: Usuarios.

Elaborado por: Oscar Curichumbi.

Directivos	
Responsabilidad	Colaboración
Atributos:	Usuarios
Id_Directivos	Tipo_cargo
Id_Usuario	
Nombre_Usuario	
Contrasenia	
Fecha_Ingreso	
Fecha_Salida	
Id_Cargo	
Estado	
Observaciones	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.67: Tarjeta CRC: Directivos.

Elaborado por: Oscar Curichumbi.

Reuniones	
Responsabilidad	Colaboración
Atributos:	Directivos
Id_Reunion	Tipo_Reunion
Id_TipoReunion	
Motivo_Reunion	
Fecha_Reunion	
Numero_Acta	
Estado	
Id_Directivos	
linkDocumento	
Observaciones	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.68: Tarjeta CRC: Reuniones.

Elaborado por: Oscar Curichumbi.

Tipo_Reunion	
Responsabilidad	Colaboración
Atributos:	
Id_TipoReunion	
Id_NombreReunion	
Descripcion	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.69: Tarjeta CRC: Tipo Reunión.

Elaborado por: Oscar Curichumbi.

4.3.6.5. Diagrama de datos

A continuación, se muestra el diagrama de datos de la Segunda Iteración:

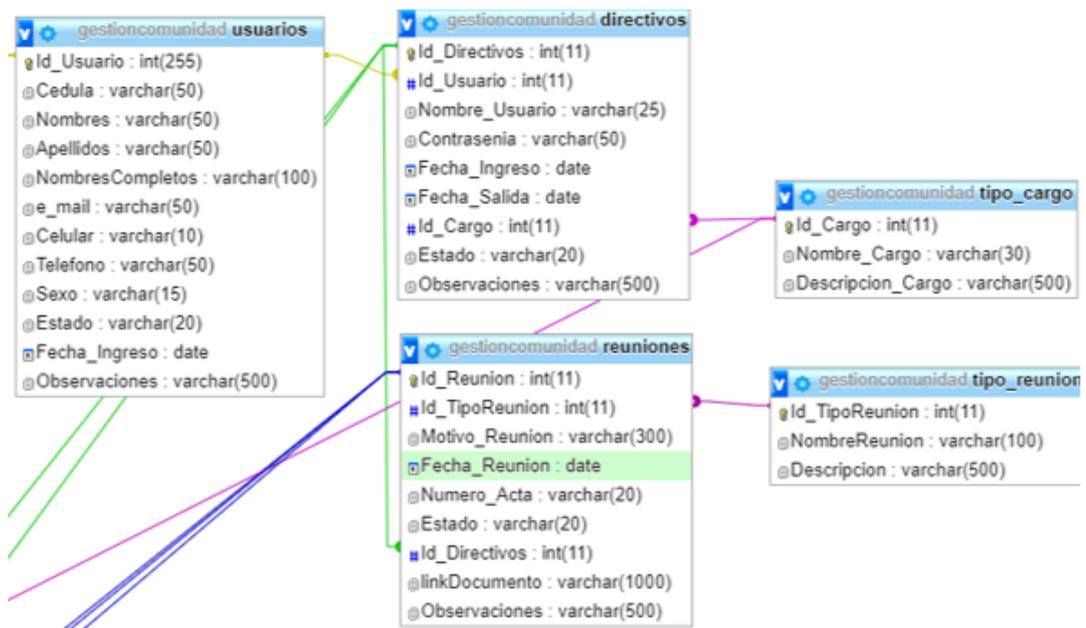


Figura 4.27: Diagrama de datos: Usuarios, Directivos y Reuniones
Elaborado por: Oscar Curichumbi.

4.3.6.6. Diseño de Interfaz

A continuación, se muestra los diseños de interfaz de la Segunda Iteración:

GESTION COMUNAL Bienvenido

Pagina Principal | Administracion | **Usuarios** | Directivos | Reuniones | Asistencia | Tesoreria | Cambio Clave

Crear Usuarios

Registrar Usuario
 Modificar Usuario
 Eliminar Usuario

Cedula:

Nombres:
 Apellidos:
 E_mail:
 Celular:
 Telefono:
 Sexo:
 Actividad:
 Fecha:

ID	Cedula	Nombre Completo	E-MAIL	Celular	Telefono	Sexo	Estado	Fecha Ingres
	1805198478	OSCAR EDUARDO CURICHUMBI				MASCULINO	ACTIVO	19/07/

Figura 4.28: Diseño de Interfaz-Verificar número de cedula.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido

Pagina Principal | Administracion | **Usuarios** | Directivos | Reuniones | Asistencia | Tesoreria | Cambio Clave

Crear Usuarios

Cedula:

Nombres:

Apellidos:

E_mail:

Celular:

Telefono:

Sexo:

Actividad:

Fecha:

Figura 4.29: Diseño de Interfaz-Registrar Usuario.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL

Pagina Principal | Administracion | Usuarios | Directivos | Reuniones | Asistencia | Tesoreria

Registrar Usuario
Modificar Usuario
 Eliminar Usuario

Ingresar Cedula:

Nombres:
Apellidos:
E_mail:
Celular:
Telefono:
Sexo: SELECCIONE SEXO
Actividad: SELECCIONE ACTIVIDAD
Observaciones:

ID	Cedula	Nombre Completo	E-MAIL	Celular	Telefono	Sexo	Estado
uario.aspx		OSCAR ENRIQUE					

Figura 4.30: Diseño de Interfaz-Modificar Usuario.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL

Pagina Principal | Administracion | Usuarios | Directivos | Reuniones | Asistencia | Tesoreria

Eliminar Usuario

Ingresar Cedula:

Nombres:

Apellidos:

E_mail:

Celular:

Telefono:

Sexo:

Actividad:

Observaciones:

ID	Cedula	Nombre Completo	E-MAIL	Celular	Telefono	Sexo	Estado
1	1805198478	OSCAR EDUARDO	oscar@gmail.com	0993552849	099999999999	MASCULINO	ACTIVO

Figura 4.31: Diseño de Interfaz-Eliminar Usuario.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido

Pagina Principal | Administracion | Usuarios | Directivos | Reuniones | Asistencia | Tesoreria | Cambio Clave

Ingresar Nuevo Directivo

Cedula:

Persona:

Nombre de Usuario:

Contraseña:

Fecha_Ingreso:

Cargo:

ID	Nombres y Apellido del Directivo	Nombre Usuario	Fecha de Ingreso	Cargo	Estado
1	OSCAR EDUARDO CURICHUMBI JEREZ	ADMIN	30/04/2019 0:00:00	ADMINISTRADOR	ACTIVO
2	JORGE MANUEL MASAQUIZA MASAQUIZA	oscar	01/07/2019 0:00:00	PRESIDENTE	ACTIVO
3	JORGE MANUEL MASAQUIZA MASAQUIZA	jorge	19/07/2019 0:00:00	VOCAL	ACTIVO
4	EDUARDO JEREZ	edu	19/07/2019 0:00:00	VOCAL	ACTIVO

Figura 4.32: Diseño de Interfaz-Registrar Directivos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido

Pagina Principal | Administracion | Usuarios | Directivos | Reuniones | Asistencia | Tesoreria | Cambio Clave

Registrar Directivos
Cambio Cargo
Finalizar Periodo

Cambio de Cargo

Directivos:

Cargo:

Observaciones:

ID	Nombre Completo	Nombre Usuario	Fecha de Ingreso	Cargo	Estado
1	OSCAR EDUARDO CURICHUMBI JEREZ	ADMIN	30/04/2019 0:00:00	ADMINISTRADOR	ACTIVO
2	JORGE MANUEL MASAQUIZA MASAQUIZA	oscar	01/07/2019 0:00:00	PRESIDENTE	ACTIVO
3	JORGE MANUEL MASAQUIZA MASAQUIZA	jorge	19/07/2019 0:00:00	VOCAL	ACTIVO
4	EDUARDO JEREZ	edu	19/07/2019 0:00:00	VOCAL	ACTIVO

Figura 4.33: Diseño de Interfaz-Cambio Cargo.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido

Pagina Principal | Administracion | Usuarios | Directivos | Reuniones | Asistencia | Tesoreria | Cambio Clave

Registrar Directivos
Cambio Cargo
Finalizar Periodo

Fin Periodo de Directivos

Directivos:

Fecha Salida:

Observaciones:

ID	Nombre Completo	Nombre Usuario	Fecha de Ingreso	Cargo	Estado
1	OSCAR EDUARDO CURICHUMBI JEREZ	ADMIN	30/04/2019 0:00:00	ADMINISTRADOR	ACTIVO
2	JORGE MANUEL MASAQUIZA MASAQUIZA	oscar	01/07/2019 0:00:00	PRESIDENTE	ACTIVO
3	JORGE MANUEL MASAQUIZA MASAQUIZA	jorge	19/07/2019 0:00:00	VOCAL	ACTIVO
4	EDUARDO JEREZ	edu	19/07/2019 0:00:00	VOCAL	ACTIVO

Figura 4.34: Diseño de Interfaz-Finalizar Periodo.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: admin
Cerrar sesión

[Pagina Principal](#) |
 [Administracion](#) |
 [Usuarios](#) |
 [Directivos](#) |
 [Reuniones](#) |
 [Asistencia](#) |
 [Tesoreria](#) |
 [Cambio Clave](#)

[Registrar Reunion](#)
[Finalizar Reunion](#)
[Modificar Reunion](#)
[Reanudar Reunion](#)

Generar Reunion

Tipo de Reunion:
Motivo de la Reunion:
Fecha Reunion:

TIPO DE REUNION	MOTIVO REUNION	FECHA REUNION	ESTADO	REPRESENTANTE
Reunion Ordinaria	REUNION GENERAL	01/01/0001 0:00:00	ACTIVO	OSCAR EDUARDO CURICHUMBI JEREZ

Figura 4.35: Diseño de Interfaz-Registrar Reunión.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: admin
Cerrar sesión

[Pagina Principal](#) |
 [Administracion](#) |
 [Reuniones](#) |
 [Asistencia](#) |
 [Tesoreria](#) |
 [Informes Tesoreria](#) |
 [Documentos](#) |
 [Movimientos](#)

[Reportes](#)

Modificar Reuniones

ID REUNION:

Tipo de Reunion:
Motivo de la Reunion:
Fecha Reunion:
Numero de Acta:
Documento:
Observaciones:

ID REUNION	Tipo Reunion	Motivo Reunion	Fecha Reunion	Numero de Acta	Estado	Link Del Documento
1	Reunion Ordinaria	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	23/04/2019	1	FINALIZADO	ACTA1.docx

Figura 4.36: Diseño de Interfaz-Modificar Reunión.
Elaborado por: Oscar Curichumbi.

Subir Acta

Motivo de Reunion:

Numero de Acta:

Documento: Ningún archivo seleccionado

MOTIVO REUNION	FECHA REUNION	N.- ACTA	DOCUMENTO
REUNION DE DIRECTIVOS	24/05/2019		SIN DOCUMENTO
REUNION GENERAL	01/01/2019		SIN DOCUMENTO
PRUEBA DE REUNIONES	01/07/2019		SIN DOCUMENTO
PARA COMPROBAR DIRECTIVO	07/08/2019		SIN DOCUMENTO

Figura 4.37: Diseño de Interfaz-Subir Acta de Reunión.
Elaborado por: Oscar Curichumbi.

Modificar Acta

ID REUNION:

DOCUMENTO: Ningún archi... eleccionado

OBSERVACIONES:

ID REUNION	Tipo Reunion	Motivo Reunion	Fecha Reunion	Numero de Acta	Estado	Link Del Documento
1	Reunion Ordinaria	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	23/04/2019	1	FINALIZADO	ACTA1.docx
2	Reunion Extraordinaria	REUNION PARA GENERAR IDEAS D CONSULTA	03/05/2019	2	FINALIZADO	ACTA2.docx
3	Reunion Extraordinaria	REUNION SOBRE LA FISCALIZACION 2019	20/05/2019	3	FINALIZADO	ACTA3.docx

Figura 4.38: Diseño de Interfaz-Modificar Acta de Reunión.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: admin
Cerrar sesión

Página Principal | Administración | Reuniones | Asistencia | Tesorería | Informes Tesorería | Documentos | Movimientos

Reportes

Finalizar Reunion

Motivo de Reunion:

Numero de Acta:

Documento:

Tipo Reunion	Motivo Reunion	Fecha Reunion	Estado	N.- ACTA	DOCUMENTO
Reunion Ordinaria	REUNION DE DIRECTIVOS	24/05/2019	ACTIVO		SIN DOCUMENTO
Reunion Ordinaria	REUNION GENERAL	01/01/2019	ACTIVO		SIN DOCUMENTO
Reunion Ordinaria	PRUEBA DE REUNIONES	01/07/2019	ACTIVO		SIN DOCUMENTO
Reunion Ordinaria	PARA COMPROBAR DIRECTIVO	07/08/2019	ACTIVO		SIN DOCUMENTO
Reunion Ordinaria	POR PRUEBA DIRECTIVO2	07/08/2019	ACTIVO		SIN DOCUMENTO
Reunion Ordinaria	POR PRUEBA DIRECTIVO3	07/08/2019	ACTIVO		SIN DOCUMENTO

Figura 4.39: Diseño de Interfaz-Finalizar Reunión.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: admin
Cerrar sesión

Página Principal | Administración | Usuarios | Directivos | Reuniones | Asistencia | Tesorería | Cambio Clave

Reuniones

- Registrar Reunion
- Finalizar Reunion**
- Modificar Reunion
- Reanudar Reunion

ACTIVAR REUNION

ID REUNION:

Motivo de la Reunion:

Observaciones:

ID REUNION	MOTIVO REUNION	FECHA REUNION	ESTADO REUNION
1	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	23/04/2019 0:00:00	FINALIZADO
2	REUNION PARA GENERAR IDEAS D CONSULTA	03/05/2019 0:00:00	FINALIZADO
3	REUNION SOBRE LA FISCALIZACION 2019	20/05/2019 0:00:00	FINALIZADO
4	REUNION DE DIRECTIVOS	24/05/2019 0:00:00	FINALIZADO
6	PRUEBA DE REUNIONES	01/07/2019 0:00:00	FINALIZADO

Figura 4.40: Diseño de Interfaz-Reanudar Reunión.
Elaborado por: Oscar Curichumbi.

INFORME GENERAL DE REUNIONES

ID REUNION	Tipo Reunion	Motivo Reunion	Fecha Reunion	Numero de Acta	Estado	Link Del Documento	PRESIDENTE
1	Reunion Ordinaria	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	23/04/2019	1	FINALIZADO	ACTA1.docx	OSCAR EDUARDO CURICHUMBI JEREZ
2	Reunion Extraordinaria	REUNION PARA GENERAR IDEAS D CONSULTA	03/05/2019	2	FINALIZADO	ACTA2.docx	OSCAR EDUARDO CURICHUMBI JEREZ
3	Reunion Extraordinaria	REUNION SOBRE LA FISCALIZACION 2019	20/05/2019	3	FINALIZADO	ACTA3.docx	OSCAR EDUARDO CURICHUMBI JEREZ
4	Reunion Ordinaria	REUNION DE DIRECTIVOS	24/05/2019		ACTIVO	SIN DOCUMENTO	OSCAR EDUARDO CURICHUMBI JEREZ
5	Reunion Ordinaria	REUNION GENERAL	01/01/2019		ACTIVO	SIN DOCUMENTO	OSCAR EDUARDO CURICHUMBI JEREZ
6	Reunion Ordinaria	PRUEBA DE REUNIONES	01/07/2019		ACTIVO	SIN DOCUMENTO	OSCAR EDUARDO CURICHUMBI JEREZ

Figura 4.41: Diseño de Interfaz-Informe de Reuniones.
Elaborado por: Oscar Curichumbi.

4.3.6.7. Codificación

- Método Agregar Usuarios

```

1  protected void btnAgregar_Click(object sender, EventArgs e)
    {
2  PersonasDAL dal = new PersonasDAL();
3  Personas usuarios = new Personas();
4  string cedula = txtCedula.Text;
5  int cant = dal.SelectRepetido(cedula);
6  if (cant >= 1){
7  lblMensajes.Text = "La cedula ingresada ya existe.
    Verifique el numero de cedula.";
8  lblMensajes.Visible = true;
9  }
10 else{
11  if (txtCedula.Text == ""){
12  lblMensajes.Text = "Ingrese un numero de Cedula.";
13  lblMensajes.Visible = true;
14  }
15  else{
16  if (txtNombres.Text == "" || txtApellidos.Text == "" ||
    cbSexo.SelectedIndex == 0 || cbEstado.SelectedIndex ==
    0){

```

```

17 lblMensajes.Text = " Verifique los datos.";
18 lblMensajes.Visible = true;
19 }
20 else{
21 try{
22 usuarios.Cedula = txtCedula.Text;
23 usuarios.Nombres = txtNombres.Text;
24 usuarios.Apellidos = txtApellidos.Text;
25 usuarios.NombresCompletos = txtNombres.Text + " " +
    txtApellidos.Text;
26 usuarios.E_mail = txtEmail.Text;
27 usuarios.Celular = txtCelular.Text;
28 usuarios.Telefono = txtTelefono.Text;
29 usuarios.Sexo = cbSexo.Text;
30 usuarios.Estado = cbEstado.Text;
31 usuarios.FechaIngreso = txtFechaingreso.Text;
32 try{
33 dal.Insert(usuarios);
34 lblMensajes.Text = "La Informacion del Usuario se ha
    ingresado correctamente";
35 lblMensajes.Visible = true;
36 cargarDatos();
37 Limpiar();
38 btnVerificar.Visible = true;
39 btnAgregar.Visible = false;
40 txtNombres.Visible = false;
41 txtApellidos.Visible = false;
42 txtEmail.Visible = false;
43 txtCelular.Visible = false;
44 txtTelefono.Visible = false;
45 cbSexo.Visible = false;
46 cbEstado.Visible = false;
47 txtFechaingreso.Visible = false;
48 btnCancelar.Visible = false;
49 btnCancelar2.Visible = true;
50 txtCedula.Focus();
51 }
52 catch{
53 lblMensajes.Text = "No se ingresado la Informacion del
    Usuario. Ingrese datos validos y Vuelva a Intentarlo";

```

```

54 lblMensajes.Visible = true;
55 }
56 }catch (MySqlException excepcion){
57 Excepciones.Gestionar(excepcion, Plataforma.Web);
58 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
59 lblMensajes.Visible = true;
60 }
61 catch (Exception excepcion){
62 Excepciones.Gestionar(excepcion, Plataforma.Web);
63 lblMensajes.Text = "Error de c#" + "<br>" + Excepciones.
    MensajePersonalizado;
64 lblMensajes.Visible = true;
65 } } } } }

```

- Método Modificar Usuario

```

1  protected void btnModificar_Click1(object sender, EventArgs
    e){
2  PersonasDAL dal = new PersonasDAL();
3  Personas usuario = new Personas();
4  if (txtCedula.Text == ""){
5  lblMensajes.Text = "Ingrese un número de Cedula.";
6  lblMensajes.Visible = true;
7  }
8  else{
9  if (txtNombres.Text == "" || txtApellidos.Text == "" ||
    cbSexo.SelectedIndex == 0 || cbEstado.SelectedIndex ==
    0){
10 lblMensajes.Text = "Para realizar la Actualizacion debe
    ingresar un número de cedula que corresponde a un
    usuarios de la comunidad, para realizar la respectiva
    busqueda de datos.";
11 lblMensajes.Visible = true;
12 Limpiar();
13 }
14 else{
15 try{

```

```

16 usuario.Nombres = txtNombres.Text;
17 usuario.Apellidos = txtApellidos.Text;
18 usuario.NombresCompletos = txtNombres.Text + " " +
    txtApellidos.Text;
19 usuario.E_mail = txtEmail.Text;
20 usuario.Celular = txtCelular.Text;
21 usuario.Telefono = txtTelefono.Text;
22 usuario.Sexo = cbSexo.Text;
23 usuario.Estado = cbEstado.Text;
24 usuario.Observaciones = txtObservaciones.Text;
25 usuario.Cedula = txtCedula.Text;
26 bool correcto = dal.Update(usuario);
27 if (correcto){
28 lblMensajes.Text = "Registro actualizado";
29 lblMensajes.Visible = true;
30 cargarDatos();
31 Limpiar();
32 txtCedula.Focus();
33 }
34 else{
35 lblMensajes.Text = "Problemas al actualizar";
36 lblMensajes.Visible = true;
37 }
38 }catch (MySqlException excepcion){
39 Excepciones.Gestionar(excepcion, Plataforma.Web);
40 lblMensajes.Text = "Error de SQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
41 }
42 catch (Exception excepcion){
43 Excepciones.Gestionar(excepcion, Plataforma.Web);
44 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
    MensajePersonalizado;
45 } } } }

```

- Método Eliminar Usuario

```

1 protected void btnEliminar_Click(object sender, EventArgs e
    ){

```

```

2  PersonasDAL dal = new PersonasDAL();
3  if (txtCedula.Text == ""){
4  lblMensajes.Text = "□Ingrese□un□numero□de□Cedula.";
5  lblMensajes.Visible = true;
6  }
7  else{
8  if (txtNombres.Text == "" && txtApellidos.Text == ""){
9  lblMensajes.Text = "□Para□Eliminar□Usuario□debe□ingresar□un
    □numero□de□cedula";
10 lblMensajes.Visible = true;
11 Limpiar();
12 }
13 else{
14 try{
15 int cedula = Convert.ToInt32(txtCedula.Text);
16 bool correcto = dal.Delete(cedula);
17 if (correcto){
18 lblMensajes.Text = "Registro□eliminado";
19 lblMensajes.Visible = true;
20 cargarDatos();
21 Limpiar();
22 txtCedula.Focus();
23 }
24 else{
25 lblMensajes.Text = "Problemas□al□eliminar";
26 lblMensajes.Visible = true;
27 }
28 }catch (MySqlException excepcion){
29 Excepciones.Gestionar(excepcion, Plataforma.Web);
30 lblMensajes.Text = "Error□de□SQL□Server:□" + "<br>" +
    Excepciones.MensajePersonalizado;
31 }
32 catch (Exception excepcion){
33 Excepciones.Gestionar(excepcion, Plataforma.Web);
34 lblMensajes.Text = "Error□de□c#□" + "<br>" + Excepciones.
    MensajePersonalizado;
35 } } } }

```

- Método Agregar Nuevo Directivo

```
1 protected void btnAgregar_Click(object sender, EventArgs e)
    {
2     DirectivosDAL dal = new DirectivosDAL();
3     Directivo directivo = new Directivo();
4     string nombreUsuario = txtUsuario.Text;
5     int cant = dal.SelectRepetido(nombreUsuario);
6     if (cant >= 1){
7     lblMensajes.Text = "El_Nombre_de_Usuario_ya_existe._
        Porfavor_ingrese_uno_Nuevo_para_continuar.";
8     lblMensajes.Visible = true;
9     }
10    else{
11    if (txtUsuarioComunidad.Text == "" || txtUsuario.Text == ""
        || txtContrasenia.Text == ""){
12    lblMensajes.Text = "Falta_llenar_datos.";
13    lblMensajes.Visible = true;
14    }
15    else{
16    if (cbCargo.SelectedIndex == 0){
17    lblMensajes.Text = "_Seleccionel_Cargo_que_corresponde_al
        _nuevo_directivo.";
18    lblMensajes.Visible = true;
19    }
20    else{
21    try{
22    int idUsuario = Convert.ToInt32(txtIdUsuario.Text);
23    directivo.IdUsuario = idUsuario;
24    directivo.NombreUsuario = txtUsuario.Text;
25    directivo.Contrasenia = txtContrasenia.Text;
26    directivo.FechaIngreso = txtFechaingreso.Text;
27    int idcargo = Convert.ToInt32(cbCargo.SelectedItem.Value);
28    directivo.IdCargo = idcargo;
29    directivo.Estado = "ACTIVO";
30    try{
31    dal.Insert(directivo);
32    lblMensajes.Text = "La_Informacion_del_Nuevo_Directivo_se
        ha_ingresado_correctamente";
33    lblMensajes.Visible = true;
```

```

34 cargarDatos();
35 Limpiar();
36 btnAgregar.Visible = false;
37 btnBuscar.Visible = true;
38 txtCedula.Enabled = true;
39 txtUsuarioComunidad.Visible = false;
40 txtUsuario.Visible = false;
41 txtContrasenia.Visible = false;
42 txtFechaingreso.Visible = false;
43 cbCargo.Visible = false;
44 btnCancelar.Visible = false;
45 btnCancelar2.Visible = true;
46 txtCedula.Focus();
47 }
48 catch{
49 lblMensajes.Text = "No se ha ingresado la Informacion del
        Nuevo Directivo. Ingrese datos validos y Vuelva a
        Intentarlo";
50 lblMensajes.Visible = true;
51 }
52 }catch (MySqlException excepcion){
53 Excepciones.Gestionar(excepcion, Plataforma.Web);
54 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
        Excepciones.MensajePersonalizado;
55 lblMensajes.Visible = true;
56 }
57 catch (Exception excepcion){
58 Excepciones.Gestionar(excepcion, Plataforma.Web);
59 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
        MensajePersonalizado;
60 lblMensajes.Visible = true;
61 } } } } }

```

- Método Modificar Cargo de Directivo

```

1 protected void btnCambiar_Click(object sender, EventArgs e)
    {
2 DirectivosDAL dal = new DirectivosDAL();

```

```

3 Directivo directivo = new Directivo();
4 if (cbDirectivos.SelectedIndex == 0 ){
5 lblMensajes.Text = "Debe seleccionar un directivo.";
6 lblMensajes.Visible = true;
7 }
8 else{
9 if (cbCargo.SelectedIndex == 0){
10 lblMensajes.Text = "Debe seleccionar el nuevo cargo.";
11 lblMensajes.Visible = true;
12 }
13 else{
14 if (txtObservaciones.Text == ""){
15 lblMensajes.Text = "Ingrese observaciones por el cambio de
    cargo.";
16 lblMensajes.Visible = true;
17 }
18 else{
19 try{
20 int idCargo = Convert.ToInt32(cbCargo.SelectedItem.Value);
21 directivo.IdCargo = idCargo;
22 directivo.Observaciones = txtObservaciones.Text;
23 int idDirectivo = Convert.ToInt32(cbDirectivos.SelectedItem
    .Value);
24 directivo.IdDirectivo = idDirectivo;
25 bool correcto = dal.Update(directivo);
26 if (correcto){
27 lblMensajes.Text = "Registro actualizado";
28 lblMensajes.Visible = true;
29 cargarDatos();
30 Limpiar();
31 cbDirectivos.Focus();
32 }
33 else{
34 lblMensajes.Text = "Problemas al actualizar";
35 lblMensajes.Visible = true;
36 }
37 }catch (MySqlException excepcion){
38 Excepciones.Gestionar(excepcion, Plataforma.Web);
39 lblMensajes.Text = "Error de SQL Server:" + "<br>" +
    Excepciones.MensajePersonalizado;

```

```

40 }
41 catch (Exception excepcion){
42     Excepciones.Gestionar(excepcion, Plataforma.Web);
43     lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
        MensajePersonalizado;
44 } } } } }

```

- Método Finalizar Periodo

```

1  protected void btnCambiar_Click(object sender, EventArgs e)
    {
2      DirectivosDAL dal = new DirectivosDAL();
3      Directivo directivo = new Directivo();
4      if (cbDirectivos.SelectedIndex == 0){
5          lblMensajes.Text = "Debe seleccionar un directivo.";
6          lblMensajes.Visible = true;
7      }
8      else{
9          try{
10             directivo.Estado = "FINALIZADO";
11             directivo.FechaSalida = txtFechaSalida.Text;
12             directivo.Observaciones = txtObservaciones.Text;
13             int idDirectivo = Convert.ToInt32(cbDirectivos.SelectedItem
                .Value);
14             directivo.IdDirectivo = idDirectivo;
15             bool correcto = dal.UpdateBaja(directivo);
16             if (correcto){
17                 lblMensajes.Text = "Registro actualizado";
18                 lblMensajes.Visible = true;
19                 cargarDatos();
20                 cargarDirectivos();
21                 Limpiar();
22                 cbDirectivos.Focus();
23             }
24             else{
25                 lblMensajes.Text = "Problemas al actualizar";
26                 lblMensajes.Visible = true;
27             }

```

```

28 }catch (MySqlException excepcion){
29     Excepciones.Gestionar(excepcion , Plataforma.Web);
30     lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
        Excepciones.MensajePersonalizado;
31 }
32 catch (Exception excepcion){
33     Excepciones.Gestionar(excepcion , Plataforma.Web);
34     lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
        MensajePersonalizado;
35 } } }

```

- Método Agregar Reuniones

```

1  protected void bntGenerar_Click(object sender , EventArgs e)
    {
2      ReunionDAL dal = new ReunionDAL();
3      Reunion reunion = new Reunion();
4      if (cbTipoReunion.SelectedIndex ==0){
5          lblMensajes.Text = "Selección de Tipo de Reunion.";
6          lblMensajes.Visible = true;
7          String txtJS = String.Format("<script>alert('{0}');</script>
            >", "Selección de Tipo de Reunion.");
8          ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
            script", txtJS, false);
9      }
10     else{
11         if (txtMotivoReunion.Text==""){
12             lblMensajes.Text = "Ingreso de Motivo de Reunion.";
13             lblMensajes.Visible = true;
14             String txtJS = String.Format("<script>alert('{0}');</script>
                >", "Ingreso de Motivo de Reunion.");
15             ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
                script", txtJS, false);
16         }
17         else{
18             try{
19                 int idTipoReunion = Convert.ToInt32(cbTipoReunion.
                    SelectedItem.Value);

```

```

20 reunion.IdTipoReunion = idTipoReunion;
21 reunion.MotivoReunion = txtMotivoReunion.Text;
22 reunion.FechaReunion = txtFechaReunion.Text;
23 reunion.Estado = "ACTIVO";
24 reunion.linkDocumento = "SIN_DOCUMENTO";
25 int idDirectivo = Convert.ToInt32(cbDEncargado.SelectedItem
    .Value);
26 reunion.IdDirectivos = idDirectivo;
27 try{
28 dal.Insert(reunion);
29 lblMensajes.Text = "La Reunion se ha Generado correctamente
    ";
30 lblMensajes.Visible = true;
31 cargarDatos();
32 Limpiar();
33 cbTipoReunion.Focus();
34 }
35 catch{
36 lblMensajes.Text = "No se ha Generado la Reunion. Ingrese
    datos validos y Vuelva a Intentarlo.";
37 lblMensajes.Visible = true;
38 String txtJS = String.Format("<script>alert('{0}');</script
    >", "No se ha Generado la Reunion. Ingrese datos validos
    y Vuelva a Intentarlo.");
39 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
40 }
41 }catch (MySqlException excepcion){
42 Excepciones.Gestionar(excepcion, Plataforma.Web);
43 lblMensajes.Text = "Error de MySQL Server:" + "<br>" +
    Excepciones.MensajePersonalizado;
44 lblMensajes.Visible = true;
45 }
46 catch (Exception excepcion){
47 Excepciones.Gestionar(excepcion, Plataforma.Web);
48 lblMensajes.Text = "Error de c#:" + "<br>" + Excepciones.
    MensajePersonalizado;
49 lblMensajes.Visible = true;
50 } } } }

```

- Método Modificar Reuniones

```
1 protected void bntModificar_Click(object sender, EventArgs
    e){
2 ReunionDAL dal = new ReunionDAL();
3 Reunion reuniones = new Reunion();
4 if (txtIdReunion.Text == ""){
5 lblMensajes.Text = "□Ingrese□ID□REUNION.";
6 lblMensajes.Visible = true;
7 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese□ID□REUNION.");
8 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
9 txtIdReunion.Focus();
10 }
11 else{
12 if (txtNActa.Text == "" || txtMotivoReunion.Text == "" ||
    cbTipoReunion.SelectedIndex == 0 ){
13 lblMensajes.Text = "□Para□realizar□la□Actualizacion□la□
    reunion□debe□tener□todos□los□campos□ingresados□
    correctamente.";
14 lblMensajes.Visible = true;
15 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Para□realizar□la□Actualizacion□la□reunion□debe□
    tener□todos□los□campos□ingresados□correctamente.");
16 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
17 }
18 else{
19 if (txtObservaciones.Text==""){
20 lblMensajes.Text = "□Para□realizar□la□actualizacion□ingrese
    □Observaciones.";
21 lblMensajes.Visible = true;
22 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Para□realizar□la□actualizacion□ingrese□
    Observaciones.");
23 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
24 txtObservaciones.Focus();
25 }
```

```

26 else{
27 try{
28 int tipoReunion = Convert.ToInt32(cbTipoReunion.
    SelectedItem.Value);
29 reuniones.IdTipoReunion = tipoReunion;
30 reuniones.MotivoReunion = txtMotivoReunion.Text;
31 reuniones.FechaReunion = txtFechaReunion.Text;
32 reuniones.NumeroActa = txtNActa.Text;
33 reuniones.linkDocumento = txtLinkDocumento.Text;
34 reuniones.Observaciones = txtObservaciones.Text;
35 int Idreunion = Convert.ToInt32(txtIdReunion.Text);
36 reuniones.IdReunion = Idreunion;
37 bool correcto = dal.Update(reuniones);
38 if (correcto){
39 lblMensajes.Text = "Registro□actualizado";
40 lblMensajes.Visible = true;
41 cargarDatos();
42 Limpiar();
43 txtIdReunion.Enabled = true;
44 txtIdReunion.Focus();
45 txtIdReunion.Text = "";
46 bntModificar.Visible = false;
47 }
48 else{
49 lblMensajes.Text = "Problemas□al□actualizar";
50 lblMensajes.Visible = true;
51 }
52 }catch (MySqlException excepcion){
53 Excepciones.Gestionar(excepcion, Plataforma.Web);
54 lblMensajes.Text = "Error□de□MYSQL□Server:□" + "<br>" +
    Excepciones.MensajePersonalizado;
55 }
56 catch (Exception excepcion){
57 Excepciones.Gestionar(excepcion, Plataforma.Web);
58 lblMensajes.Text = "Error□de□c#□" + "<br>" + Excepciones.
    MensajePersonalizado;
59 } } } } }

```

- Método Finalizar Reuniones

```

1  protected void bntFinalizar_Click(object sender, EventArgs
    e){
2  ReunionDAL dal = new ReunionDAL();
3  Reunion reuniones = new Reunion();
4  if (cbMotivoReunion.SelectedIndex == 0){
5  lblMensajes.Visible = true;
6  lblMensajes.Text = " Selecciona Motivo de Reunion para
    Finalizar Reunion.";
7  String txtJS = String.Format("<script>alert('{0}')</script
    >", " Selecciona Motivo de Reunion para Finalizar Reunion
    .");
8  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
9  }
10 else{
11 if (txtNumeroActa.Text==""){
12 lblMensajes.Visible = true;
13 lblMensajes.Text = " Para finalizar la reunion debe
    constatar el NUMERO DE ACTA.";
14 String txtJS = String.Format("<script>alert('{0}')</script
    >", " Para finalizar la reunion debe constatar el NUMERO
    DE ACTA.");
15 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
16 }
17 else{
18 if (txtDocumento.Text==""){
19 lblMensajes.Visible = true;
20 lblMensajes.Text = " Para finalizar la reunion debe
    constatar el DOCUMENTO.";
21 String txtJS = String.Format("<script>alert('{0}')</script
    >", " Para finalizar la reunion debe constatar el
    DOCUMENTO.");
22 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
23 }
24 else{
25 try{

```

```

26 reuniones.Estado = "FINALIZADO";
27 int motivoreunion = Convert.ToInt32(cbMotivoReunion.
    SelectedItem.Value);
28 reuniones.IdReunion = motivoreunion;
29 bool correcto = dal.UpdateFinalizar(reuniones);
30 if (correcto){
31 lblMensajes.Text = "Reunion□Finalizada.";
32 lblMensajes.Visible = true;
33 cargarDatos();
34 cargarReunion();
35 cbMotivoReunion.Focus();
36 cbMotivoReunion.SelectedIndex = 0;
37 txtDocumento.Text = "";
38 txtNumeroActa.Text = "";
39 cbMotivoReunion.Enabled = true;
40 bntBuscar.Visible = true;
41 bntFinalizar.Visible = false;
42 }
43 else{
44 lblMensajes.Text = "Problemas□al□Finalizar□Reunion.";
45 lblMensajes.Visible = true;
46 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Problemas□al□Finalizar□Reunion.");
47 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
48 }
49 }catch (MySqlException excepcion){
50 Excepciones.Gestionar(excepcion, Plataforma.Web);
51 lblMensajes.Text = "Error□de□MYSQL□Server:□" + "<br>" +
    Excepciones.MensajePersonalizado;
52 }
53 catch (Exception excepcion){
54 Excepciones.Gestionar(excepcion, Plataforma.Web);
55 lblMensajes.Text = "Error□de□c#□" + "<br>" + Excepciones.
    MensajePersonalizado;
56 } } } } }

```

- Método Activar Reunion

```

1  protected void bntModificar_Click(object sender, EventArgs
    e){
2  ReunionDAL dal = new ReunionDAL();
3  Reunion reuniones = new Reunion();
4  if (txtIdReunion.Text == ""){
5  lblMensajes.Text = "▯Ingrese▯ID▯Reunion.";
6  lblMensajes.Visible = true;
7  String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese▯ID▯Reunion.");
8  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
9  }
10 else{
11 if (txtObservaciones.Text == ""){
12 lblMensajes.Text = "▯Ingrese▯observacion▯para▯activar▯la▯
    reunion.";
13 lblMensajes.Visible = true;
14 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese▯observacion▯para▯activar▯la▯reunion.");
15 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
16 }
17 else{
18 try{
19 reuniones.Estado = "ACTIVO";
20 reuniones.Observaciones = txtObservaciones.Text;
21 int Idreunion = Convert.ToInt32(txtIdReunion.Text);
22 reuniones.IdReunion = Idreunion;
23 bool correcto = dal.UpdateActivar(reuniones);
24 if (correcto){
25 lblMensajes.Text = "Reunion▯activada▯exitosamente.";
26 lblMensajes.Visible = true;
27 cargarDatos();
28 txtIdReunion.Focus();
29 txtIdReunion.Enabled = true;
30 txtIdReunion.Text = "";
31 txtMotivoReunion.Text = "";
32 txtObservaciones.Text = "";
33 }
34 else{

```

```

35 lblMensajes.Text = "Problemas al activar la REUNION.
    Verifique conexion a la base de datos.";
36 lblMensajes.Visible = true;
37 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Problemas al activar la REUNION. Verifique conexion
    a la base de datos.");
38 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
39 }
40 }catch (MySqlException excepcion){
41 Excepciones.Gestionar(excepcion, Plataforma.Web);
42 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
43 }
44 catch (Exception excepcion){
45 Excepciones.Gestionar(excepcion, Plataforma.Web);
46 lblMensajes.Text = "Error de c#" + "<br>" + Excepciones.
    MensajePersonalizado;
47 } } } }

```

■ Método Subir Actas

```

1 protected void bntSubirActa_Click(object sender, EventArgs
    e){
2 ReunionDAL dal = new ReunionDAL();
3 Reunion reuniones = new Reunion();
4 if (cbMotivoReunion.SelectedIndex == 0){
5 lblMensajes.Text = " Seleccione Motivo de Reunion para
    Finalizar Reunion.";
6 lblMensajes.Visible = true;
7 }
8 else{
9 if (txtNumeroActa.Text == ""){
10 lblMensajes.Text = " Ingrese numero de acta.";
11 lblMensajes.Visible = true;
12 }
13 else{
14 if (FileUpload2.HasFile){

```

```

15 string extension = Path.GetExtension(FileUpload2.FileName);
16 if (extension.ToLower() != ".doc" && extension.ToLower() !=
    ".docx" && extension.ToLower() != ".pdf"){
17 lblMensajes.Text = "Selecione otro tipo de archivo.";
18 lblMensajes.Visible = true;
19 }
20 else{
21 int filezise = FileUpload2.PostedFile.ContentLength;
22 if (filezise > 12582912){
23 lblMensajes.Text = "Tamaño máximo de archivo (12MB)
    excedido.";
24 lblMensajes.Visible = true;
25 }
26 else{
27 string documento = FileUpload2.FileName;
28 int cant = dal.SelectComparacionDocumento(documento);
29 if (cant >= 1){
30 lblMensajes.Text = "El nombre del archivo ya se encuentra
    asignado. Porfavor cambiar el nombre del archivo.";
31 lblMensajes.Visible = true;
32 }
33 else{
34 try{
35 FileUpload2.SaveAs(Server.MapPath("/") + "/Archivo/Actas/"
    + FileUpload2.FileName);
36 reuniones.linkDocumento = FileUpload2.FileName;
37 reuniones.NumeroActa = txtNumeroActa.Text;
38 int motivoReunion = Convert.ToInt32(cbMotivoReunion.
    SelectedItem.Value);
39 reuniones.IdReunion = motivoReunion;
40 bool correcto = dal.UpdateActa(reuniones);
41 if (correcto){
42 lblMensajes.Text = "Acta subida al servidor con éxito.";
43 lblMensajes.Visible = true;
44 cargarDatos();
45 cargarReunion();
46 cbMotivoReunion.Focus();
47 cbMotivoReunion.SelectedIndex = 0;
48 txtNumeroActa.Text = "";
49 }

```

```

50 else{
51 lblMensajes.Text = "Problemas al subir acta. Verifique
    conexion al servidor.";
52 lblMensajes.Visible = true;
53 }
54 }catch (MySqlException excepcion){
55 Excepciones.Gestionar(excepcion, Plataforma.Web);
56 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
57 }
58 catch (Exception excepcion){
59 Excepciones.Gestionar(excepcion, Plataforma.Web);
60 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
    MensajePersonalizado;
61 } } } } }
62 else{
63 lblMensajes.Text = "Seleccione archivo.";
64 lblMensajes.Visible = true;
65 } } } }

```

■ Método Modificar Actas

```

1 protected void bntSubirActa_Click(object sender, EventArgs
    e){
2 ReunionDAL dal = new ReunionDAL();
3 Reunion reuniones = new Reunion();
4 if (txtIdReunion.Text==""){
5 lblMensajes.Text = "Ingrese ID REUNION.";
6 lblMensajes.Visible = true;
7 txtIdReunion.Focus();
8 }
9 else{
10 string userName2 = HttpContext.Current.User.Identity.Name;
11 int Idr2 = Convert.ToInt32(txtIdReunion.Text);
12 int cant2 = dal.SelectConteoMoficarActa(Idr2, userName2);
13 if (cant2==0){
14 lblMensajes.Text = "Ingrese ID REUNION que se encuentra en
    la consulta.";

```

```

15 lblMensajes.Visible = true;
16 }
17 else{
18 if (FileUpload2.HasFile){
19 string extension = Path.GetExtension(FileUpload2.FileName);
20 if (extension.ToLower() != ".doc" && extension.ToLower() !=
    ".docx" && extension.ToLower() != ".pdf"){
21 lblMensajes.Text = "Selecione otro tipo de archivo.";
22 lblMensajes.Visible = true;
23 }
24 else{
25 int filezise = FileUpload2.PostedFile.ContentLength;
26 if (filezise > 12582912){
27 lblMensajes.Text = "Tamaño máximo de archivo (12MB)
    excedido.";
28 lblMensajes.Visible = true;
29 }
30 else{
31 if (txtObservaciones.Text == ""){
32 lblMensajes.Text = "Ingrese observaciones por la
    modificacion del archivo.";
33 lblMensajes.Visible = true;
34 }
35 else{
36 try{
37 FileUpload2.SaveAs(Server.MapPath("/") + "/Archivo/Actas/"
    + FileUpload2.FileName);
38 reuniones.linkDocumento = FileUpload2.FileName;
39 reuniones.Observaciones = txtObservaciones.Text;
40 int idreunion = Convert.ToInt32(txtIdReunion.Text);
41 reuniones.IdReunion = idreunion;
42 bool correcto = dal.UpdateArchivoActa(reuniones);
43 if (correcto){
44 lblMensajes.Text = "Acta subida al servidor con éxito.";
45 lblMensajes.Visible = true;
46 cargarDatos();
47 txtIdReunion.Text = "";
48 txtIdReunion.Focus();
49 txtObservaciones.Text = "";
50 }

```

```

51  else{
52  lblMensajes.Text = "Problemas al subir acta. Verifique
        conexion al servidor.";
53  lblMensajes.Visible = true;
54  }
55  }catch (MySqlException excepcion){
56  Excepciones.Gestionar(excepcion, Plataforma.Web);
57  lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
        Excepciones.MensajePersonalizado;
58  }
59  catch (Exception excepcion){
60  Excepciones.Gestionar(excepcion, Plataforma.Web);
61  lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
        MensajePersonalizado;
62  } } } } }
63  else{
64  lblMensajes.Text = "Seleccione archivo.";
65  lblMensajes.Visible = true;
66  } } } }

```

4.3.6.8. Pruebas de Aceptación

A continuación, se muestra las pruebas de aceptación de la Segunda Iteración:

CASO DE PRUEBA	
Código: 3	Nº Historia de Usuario: 3
Historia de Usuario: Registro de Usuarios.	
Condiciones de Ejecución: El Presidente deberá solicitar al usuario la información solicitada por el sistema.	
Entrada/Pasos de Ejecución:	
Crear Usuarios	
Seleccionar la pestaña Usuarios, la opción Crear Usuarios.	
Ingresar la información del usuario para registrar en la base de datos.	
Luego seleccionar la opción Agregar.	
Modificar Usuarios	
Seleccionar la pestaña Usuarios, la opción Modificar Usuarios.	
Introducir la cedula del usuario para buscar la información que desea modificar.	
Modifique la información del usuario que se ha ingresado incorrectamente en el sistema.	
Luego seleccionar la opción Modificar.	
Eliminar Usuarios	
Seleccionar la pestaña Usuarios, la opción Eliminar Usuarios.	
Introducir la cedula del usuario que desea eliminar.	
Confirmar si el usuario a eliminar es correcto.	
Luego seleccionar la opción Eliminar.	
Resultado Esperado: La información del usuario será guardado, modificado o eliminado en la base de datos.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.70: Caso de Prueba: Ingreso de usuarios al sistema.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 4	Nº Historia de Usuario: 4
Historia de Usuario: Asignación de usuarios para la directiva de la comunidad.	
Condiciones de Ejecución: El Expresidente deberá solicitar que los directivos ingresen un nombre de usuario y una contraseña para poder ingresar al sistema.	
Entrada/Pasos de Ejecución:	
Ingresar Nuevo directivo	
Seleccionar la pestaña Directivos, la opción Nuevo Directivo.	
Seleccionar los campos correspondientes para registrar al nuevo directivo en la base de datos.	
Luego seleccionar la opción Agregar.	
Cambio de Cargo	
Seleccionar la pestaña Directivos, la opción Modificar Cargo.	
Seleccionar el directivo y modificar el cargo que se ha ingresado incorrectamente al sistema.	
Luego seleccionar la opción Cambiar.	
Finalizar Periodo	
Seleccionar la pestaña Directivos, la opción Finalizar Periodo.	
Seleccionar el directivo, el estado y asignar fecha de salida, en la caja de observaciones llenarla si hay alguna.	
Confirmar si el directivo ha finalizado su periodo.	
Luego seleccionar la opción Finalizar.	
Resultado Esperado: La información del nuevo directivo será guardado en la base de datos. El cargo del directivo podrá ser modificado. Finalizar el periodo del directivo sin inconsistencias.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.71: Caso de Prueba: Asignación de usuarios para la directiva de la comunidad.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 5	Nº Historia de Usuario: 5
Historia de Usuario: Generar reuniones.	
Condiciones de Ejecución: Solo el presidente o vicepresidente podrán generar una reunión.	
Entrada/Pasos de Ejecución:	
Generar Reunión	
<p>Seleccionar la pestaña Reuniones, la opción Generar Reunión. Seleccionar el tipo de reunión e ingresar el motivo de reunión. Luego dar clic en el botón Generar para registrar la reunión.</p>	
Modificar Reunión	
<p>Seleccionar la pestaña Reuniones, la opción Modificar Reunión. Buscar la reunión a modificar con el id de reunión. Modifique la información de la reunión que se ha ingresado incorrectamente en el sistema. Luego seleccionar la opción Modificar.</p>	
Subir Acta	
<p>Seleccionar la pestaña Reuniones, la opción Subir Acta. El secretario(a) debe subir el archivo del acta e ingresar el número de acta. Luego dar clic en el botón subir acta para registrar los cambios.</p>	
Modificar Acta	
<p>Seleccionar la pestaña Reuniones, la opción Modificar Acta. Ingresar el ID REUNION y subir el nuevo archivo del acta, luego dar clic en el botón Modificar para registrar la modificación.</p>	
Finalizar Reunión	
<p>Seleccionar la pestaña Reuniones, la opción Finalizar Reunión. Seleccionar y llenar los campos necesarios para finalizar la reunión. Luego dar clic en el botón Finalizar.</p>	
Activar Reunión	
<p>Seleccionar la pestaña Reuniones, la opción Activar Reunión. Buscar la reunión con el ID REUNION e ingresamos las observaciones, damos clic en el botón Activar Reunión para reanudar la reunión finalizada.</p>	
Informe Reunión	
<p>Seleccionar la pestaña Reuniones, la opción Informe General de Reunión. El informe mostrara todas las reuniones con su respectiva información.</p>	
Resultado Esperado: La información de la reunión será guardado en la base de datos. Se podrá modificar la información de las reuniones sin ningún problema, pero solo el presidente podrá realizar dichos cambios. Las actas podrán ser subidas al servidor sin ningún problema, y se podrá visualizar la información de las reuniones en el informe.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.72: Caso de Prueba: Generar reuniones.

Elaborado por: Oscar Curichumbi.

4.3.6.9. Resultados de la Segunda Iteración

Como resultado de la entrega de la segunda iteración que consta de los módulos de Registro de Usuarios, Asignación de usuarios para la directiva de la comunidad y Generación de reuniones, el cliente quedó satisfecho con las funcionalidades de los módulos que se desarrollaron, la segunda iteración se ha concluido sin ningún problema manifestado por el cliente. A continuación, se realizará la Tercera iteración.

4.3.7. Desarrollo de la Tercera Iteración

Se realizará la descripción general de las tareas respectivas de la sexta historia de usuario, que pertenece a la tercera iteración. Como se indica en la planificación se desarrollará el módulo de control de asistencia de los usuarios y directivos de la comunidad. En la Tabla 4.72 se muestra de manera general las tareas de ingeniería correspondientes a la historia de usuario número 6 para cumplir con la planificación.

Numero de Tarea	Numero de Historia	Esfuerzo	Nombre de la Tarea
27	6	1	Diseño de la base de datos para el control de asistencia.
28	6	1	Diseño de la interfaz para la Gestión del control de asistencia.
29	6	2	Validación de datos para registrar correctamente el control de asistencia.
30	6	2	Crear Triggers para realizar la acumulación de multas de atraso y faltas de los usuarios en un solo valor acumulativo, para realizar el cobro respectivo de las multas en su totalidad o con el descuento aprobado por los usuarios en la reunión.
31	6	4	Guardar la información del control de asistencia en la base de datos.
32	6	3	Modificar la información de la base de datos de la asistencia.
33	6	2	Informe General, Informe por Usuario e Informe por Estado sobre el control de asistencia.

Tabla 4.73: Tareas a Realizar de la Historia de Usuario 6.

Elaborado por: Oscar Curichumbi.

4.3.7.1. Estimación de la Tercera Iteración

A continuación, se muestra la estimación de la Tercera Iteración:

ITERACIÓN 3				
N.- Historia	Fecha Inicio	Fecha Final	DÍAS	HORAS
6	20/05/2019	02/06/2019	14	112
TOTAL ESTIMACIÓN:			14	112

Tabla 4.74: Horas asignadas para el desarrollo de la Iteración 3.

Elaborado por: Oscar Curichumbi.

4.3.7.2. Bocetos de la Historia de Usuario 6

En las siguientes figuras se muestran los bocetos de la Historia de Usuario 6 que forman parte de la Tercera Iteración.

Control de Asistencia

Reunión:

Valor Multa Atraso:

Valor Multa Ausencia:

Fecha:

ID	Nombres Completos	Estado	
1	Oscar Curichumbi	<input type="radio"/> Presente <input type="radio"/> Atrasado <input type="radio"/> Ausente	<input type="text" value="Pasar Lista"/>
2	Eduardo Jerez	<input type="radio"/> Presente <input type="radio"/> Atrasado <input type="radio"/> Ausente	<input type="text" value="Pasar Lista"/>
3	Luis Guerrero	<input type="radio"/> Presente <input type="radio"/> Atrasado <input type="radio"/> Ausente	<input type="text" value="Pasar Lista"/>

Figura 4.42: Boceto de Control de Asistencia.

Elaborado por: Oscar Curichumbi.

Modificar Asistencia

Reunión:

Usuario:

Valor Atraso:

Valor Ausencia:

Estado:

Figura 4.43: Bosquejo de Modificar Asistencia.
Elaborado por: Oscar Curichumbi.

Informe Asistencia Por Reunión

Reunión:

Usuarios	Reunión	Estado	Multa Atraso	Multa Ausencia	Cobro Multas

Figura 4.44: Boceto de Informe Asistencia Por Reunión.
Elaborado por: Oscar Curichumbi.

Informe Asistencia Por Usuario

Usuario:

Usuario	Reunión	Estado	Valor Multa	Cobro multa

Figura 4.45: Boceto de Informe Asistencia Por Usuario.
Elaborado por: Oscar Curichumbi.

Informe Asistencia Por Estado

Estado:

Usuario	Reunión	Estado	Multa Valor	Cobro Multa	Estado Cobro

Figura 4.46: Boceto de Informe Asistencia Por Estado.
Elaborado por: Oscar Curichumbi.

4.3.7.3. Descripción de las Tareas de Ingenierías

En base a la planificación se ha realizado las tareas de ingeniería, en estas se ha asignado el tiempo de estimación para el desarrollo de cada una de ellas. Si el cliente está satisfecho con el desarrollo de las tareas, que forman parte de la historia de usuario, esto cumplirá con las necesidades del cliente y de este modo se culminará la tercera iteración. A continuación, se muestra el desarrollo de las tareas de la historia de usuario.

Tarea de Ingeniería	
Numero de Tarea: 27	Numero de Historia: 6
Nombre de Tarea: Diseño de la base de datos para el control de asistencia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/05/2019	Fecha Fin: 20/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se creará la tabla asistencias y multas de asistencia en la base de datos, para registrar la información del control de asistencia que se realizarán en la comunidad por cada reunión.	

Tabla 4.75: Tarea 27 de la Historia de Usuario N.-6.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 28	Numero de Historia: 6
Nombre de Tarea: Diseño de la interfaz para la Gestión del control de asistencia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 21/05/2019	Fecha Fin: 21/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el secretario(a) o los directivos asignados podrán ingresar la información del control de asistencia que se realizarán en la comunidad.	

Tabla 4.76: Tarea 28 de la Historia de Usuario N.-6.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 29	Numero de Historia: 6
Nombre de Tarea: Validación de datos para registrar correctamente el control de asistencia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 22/05/2019	Fecha Fin: 23/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere validar los datos del control de asistencia antes de ingresar a la base de datos, para no ingresar datos nulos o datos erróneos.	

Tabla 4.77: Tarea 29 de la Historia de Usuario N.-6.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 30	Numero de Historia: 6
Nombre de Tarea: Crear Triggers para realizar la acumulación de multas de atraso y faltas de los usuarios en un solo valor acumulativo, para realizar el cobro respectivo de las multas en su totalidad o con el descuento aprobado por los usuarios en la reunión.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 24/05/2019	Fecha Fin: 25/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Los Triggers ayudaran a registrar por cada usuario las multas de atrasos y faltas en la tabla multas de asistencia después de ingresar datos en la tabla asistencias, para poder controlar de mejor manera los cobros.	

Tabla 4.78: Tarea 30 de la Historia de Usuario N.-6.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 31	Numero de Historia: 6
Nombre de Tarea: Guardar la información del control de asistencia en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 26/05/2019	Fecha Fin: 29/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información del control de asistencia con las respectivas multas de atrasos o faltas, para registrarlos en la base de datos y realizar el respectivo cobro.	

Tabla 4.79: Tarea 31 de la Historia de Usuario N.-6.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 32	Numero de Historia: 6
Nombre de Tarea: Modificar la información de la base de datos de la asistencia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 29/05/2019	Fecha Fin: 31/05/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se modifica la información de la asistencia, para rectificar los datos mal ingresados en la base de datos.	

Tabla 4.80: Tarea 32 de la Historia de Usuario N.-6.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 33	Numero de Historia: 6
Nombre de Tarea: Informe General, Informe por Usuario e Informe por Estado sobre el control de asistencia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 01/06/2019	Fecha Fin: 02/06/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se realizará la visualización de los Informes Generales, por usuario o por estado para manifestar a los usuarios sus Atrasos o sus Ausencias.	

Tabla 4.81: Tarea 33 de la Historia de Usuario N.-6.

Elaborado por: Oscar Curichumbi.

4.3.7.4. Tarjetas CRC: Historia de usuario 6

En las siguientes tablas se muestran las tarjetas CRC, estas tarjetas son de gran utilidad para la Tercera Iteración.

Asistencia	
Responsabilidad	Colaboración
Atributos:	Usuarios
Id_Asistencia	Reuniones
Id_Usuario	Directivos
Id_Reunion	
Id_Directivos	
Fecha_Asistencia	
Estado	
ValorMulta	
MontoPago	
SaldoTotal	
EstadoDeuda	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.82: Tarjeta CRC: Control de Asistencia.

Elaborado por: Oscar Curichumbi.

4.3.7.5. Diagrama de datos

A continuación, se muestra el diagrama de datos de la Tercera Iteración:

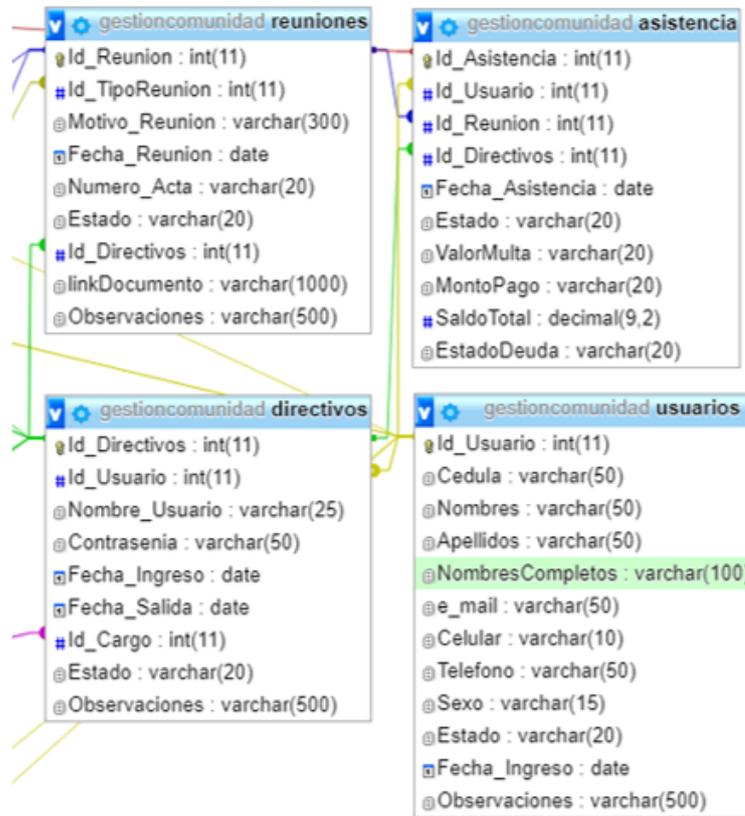


Figura 4.47: Diagrama de datos: Control de Asistencia.
Elaborado por: Oscar Curichumbi.

4.3.7.6. Diseño de Interfaz

A continuación, se muestra los diseños de interfaz de la Tercera Iteración:

Figura 4.48: Diseño de Interfaz-Control de Asistencia.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: osc
Cerrar sesi

[Página principal](#) [Usuarios](#) [Directivos](#) [Reuniones](#) [Asistencia](#)

CONTROL DE ASISTENCIA

Reunion: REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO

Valor Multa Atraso: 0.00

Valor Multa Ausencia: 0.00

Fecha-Hora: 22-05-2019 Cancelar

Verifique si el Usuario o Directivo esta PRESENTE, ATRASADO O AUSENTE

ID	NOMBRE COMPLETO	ESTADO	ASISTENCIA
1	OSCAR EDUARDO CURICHUMBI JEREZ	<input type="radio"/> PRESENTE <input type="radio"/> ATRASADO <input type="radio"/> AUSENTE	<input type="button" value="PASAR LISTA"/>
5	GGGGGGGG GGGGGGGGGG FGFG	<input checked="" type="radio"/> PRESENTE <input checked="" type="radio"/> ATRASADO <input checked="" type="radio"/> AUSENTE	<input type="button" value="PASAR LISTA"/>
6	DYTYYYD HSHHHHHS KAKKAKAKAKAA AKAKAKAKAKA	<input type="radio"/> PRESENTE <input type="radio"/> ATRASADO <input type="radio"/> AUSENTE	<input type="button" value="PASAR LISTA"/>
6	DYTYYYD HSHHHHHS KAKKAKAKAKAA AKAKAKAKAKA	<input type="radio"/> PRESENTE <input type="radio"/> ATRASADO <input type="radio"/> AUSENTE	<input type="button" value="PASAR LISTA"/>

Figura 4.49: Diseño de Interfaz-Control de Asistencia Usuarios.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: osc
Cerrar sesi

[Página principal](#) [Usuarios](#) [Directivos](#) [Reuniones](#) [Asistencia](#)

CONTROL DE ASISTENCIA

Reunion: REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO

Valor Multa Atraso: 0.00

Valor Multa Ausencia: 0.00

Fecha-Hora: 22-05-2019 Cancelar

Verifique si el Usuario o Directivo esta PRESENTE, ATRASADO O AUSENTE

ID	NOMBRE COMPLETO	ESTADO	ASISTENCIA
1	OSCAR EDUARDO CURICHUMBI JEREZ	<input checked="" type="radio"/> PRESENTE <input checked="" type="radio"/> ATRASADO <input checked="" type="radio"/> AUSENTE	<input type="button" value="PASAR LISTA"/>
6	DYTYYYD HSHHHHHS KAKKAKAKAKAA AKAKAKAKAKA	<input type="radio"/> PRESENTE <input type="radio"/> ATRASADO <input type="radio"/> AUSENTE	<input type="button" value="PASAR LISTA"/>

Figura 4.50: Diseño de Interfaz-Control de Asistencia Directivos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Usuarios Directivos Reuniones Asistencia Cambio Clave

MODIFICAR ASISTENCIA

Reunion: SELECCIONE MOTIVO DE REUNION

Usuario: SELECCIONE USUARIO

Valor Multa Atraso:

Valor Multa Ausencia:

Estado: SELECCIONE ESTADO

Figura 4.51: Diseño de Interfaz-Modificar Asistencia.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Usuarios Directivos Reuniones Asistencia Cambio Clave

INFORME ASISTENCIA GENERAL

Reunion: REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO

Usuarios	Reunion	Estado	Multa Atraso	Cobro Atraso	Multa Ausencia	Cobro Ausencia
ADMINISTRADOR	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	PRESENTE	0.00	0.00	0.00	0.00
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	PRESENTE	0.00	0.00	0.00	0.00

Figura 4.52: Diseño de Interfaz-Informe General de Asistencia.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **adm**
[Cerrar sesión](#)

Página Principal Administración Usuarios Directivos Reuniones Asistencia Cambio Clave

INFORME ASISTENCIA POR USUARIO

Usuario: ADMINISTRADOR

Usuario	Reunion	Estado	Multa Atraso	Cobro Atraso	Multa Ausencia	Cobro Ausencia
ADMINISTRADOR	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	PRESENTE	0.00	0.00	0.00	0.00
ADMINISTRADOR	REUNION PARA GENERAR IDEAS D CONSULTA	PRESENTE	0.00	0	0.00	0
ADMINISTRADOR	REUNION SOBRE LA FISCALIZACION 2019	AUSENTE	0.00	0.00	20.00	20

Figura 4.53: Diseño de Interfaz-Informe Asistencia Por Usuario.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **adm**
[Cerrar sesión](#)

Página Principal Administración Usuarios Directivos Reuniones Asistencia Cambio Clave

INFORME ASISTENCIA POR ESTADO

Estado: ATRASADO

Usuario	Reunion	Estado	Multa Atraso	Cobro Atraso	Multa Ausencia	Cobro Ausencia
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	REUNION PARA GENERAR IDEAS D CONSULTA	ATRASADO	2.00	2	0.00	0.00

Figura 4.54: Diseño de Interfaz-Informe Asistencia Por Estado.
 Elaborado por: Oscar Curichumbi.

4.3.7.7. Codificación

- Método Controlar Asistencias

```
1 protected void gvCargarDatos_RowCommand(object sender ,
    GridViewCommandEventArgs e){
```

```

2 AsistenciaDAL dal = new AsistenciaDAL();
3 Asistencias asistencia = new Asistencias();
4 CultureInfo culture = new CultureInfo("en-GB");
5 if (e.CommandName == "Select"){
6 int rowIndex = Convert.ToInt32(e.CommandArgument);
7 GridViewRow row = gvCargarDatos.Rows[rowIndex];
8 Label Id = row.FindControl("ID") as Label;
9 Button B = row.FindControl("bntPasarLista") as Button;
10 RadioButtonList RP = row.FindControl("rbtEstado") as
    RadioButtonList;
11 if (cbReunion.SelectedIndex == 0){
12 lblMensajes.Visible = true;
13 lblMensajes.Text = "Seleccione Reunion para continuar con
    el control de asistencia.";
    String
    txtJS = String.Format("<script>alert('{0}');</script>",
    "Seleccione Reunion para continuar con el control de
    asistencia.");
14 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
15 }
16 else{
17 decimal argument = Convert.ToDecimal(txtValorAtraso.Text,
    culture);
18 int count = BitConverter.GetBytes(decimal.GetBits(argument)
    [3])[2];
19 if (count > 2){
20 lblMensajes.Text = "Ingrese solo dos decimales.";
21 lblMensajes.Visible = true;
22 txtValorAtraso.Focus();
23 }
24 else{
25 decimal argument2 = Convert.ToDecimal(txtValorFalta.Text,
    culture);
26 int count2 = BitConverter.GetBytes(decimal.GetBits(
    argument2)[3])[2];
27 if (count2 > 2){
28 lblMensajes.Text = "Ingrese solo dos decimales.";
29 lblMensajes.Visible = true;
30 txtValorFalta.Focus();
31 }

```

```

32 else{
33 if (RP.SelectedValue == ""){
34 lblMensajes.Visible = true;
35 lblMensajes.Text = "Verifique si el Usuario o Directivo
    esta PRESENTE, ATRASADO O AUSENTE. ";
    String txtJS = String.Format
    ("<script>alert('{0}')</script>", "Verifique si el
    Usuario o Directivo esta PRESENTE, ATRASADO O AUSENTE. ")
    ;
36 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
37 }
38 else{
39 if (RP.SelectedValue == "ATRASADO" && txtValorAtraso.Text
    == "0.00"){
40 lblMensajes.Visible = true;
41 lblMensajes.Text = "Asigne valor de atraso. ";
42 String txtJS = String.Format("<script>alert('{0}')</script
    >", "Asigne valor de atraso. ");
    ScriptManager.
    RegisterClientScriptBlock(Page, GetType(), "script",
    txtJS, false);
    txtValorAtraso.Focus();
43 }
44 else if (RP.SelectedValue == "ATRASADO"){
45 try{
46 int idusuario = Convert.ToInt32(Id.Text);
47 asistencia.IdUsuario = idusuario;
48 int idreunion = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
    asistencia.IdReunion = idreunion;
49 int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
    asistencia.
    IdDirectivos = idDirectivo;
50 asistencia.FechaAsistencia = txtFecha.Text;
51 asistencia.Estado = RP.SelectedValue;
52 asistencia.ValorMulta = txtValorAtraso.Text;
53 asistencia.EstadoDeuda = "PENDIENTE";
54 try{
55 int Idusuario2 = Convert.ToInt32(Id.Text);

```

```

56 int Idreunion2 = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
57 int cant = dal.SelectRepetido(Idusuario2, Idreunion2);
58 if (cant >= 1){
59 lblMensajes.Text = "El usuario ha pasado la lista en la
    reunion seleccionada. Verifique si la reunion es la
    correcta o verifique el reporte de asistencias, para
    comprobar que usuarios han pasado la lista en la reunion
    seleccionada.";
60 String txtJS = String.Format("<script>alert('{0}')</script
    >", "El usuario ha pasado la lista en la reunion
    seleccionada. Verifique si la reunion es la correcta o
    verifique el reporte de asistencias, para comprobar que
    usuarios han pasado la lista en la reunion seleccionada.
    ");
    ScriptManager.RegisterClientScriptBlock(Page, GetType(),
        "script", txtJS, false);
                                                lblMensajes.
        Visible = true;
61 }
62 else{
63 dal.Insert(asistencia);
64 lblMensajes.Text = "El Usuario a Pasado la Lista.";
65 String txtJS = String.Format("<script>alert('{0}')</script
    >", "El Usuario a Pasado la Lista.");
                                                ScriptManager.
        RegisterClientScriptBlock(Page, GetType(), "script",
        txtJS, false);
        lblMensajes.Visible = true;
66 B.Visible = false;
67 }
68 }
69 catch{
70 lblMensajes.Text = "Ocurrio un problema al pasar la lista.
    Verifique la conexion a la base de datos.";
71 String txtJS = String.Format("<script>alert('{0}')</script
    >", "Ocurrio un problema al pasar la lista. Verifique la
    conexion a la base de datos.");
                                                ScriptManager.RegisterClientScriptBlock(Page, GetType(),

```

```

        "script", txtJS, false);
                                                    lblMensajes.
        Visible = true;
72 }
73 }
74 catch (MySqlException excepcion){
75 Excepciones.Gestionar(excepcion, Plataforma.Web);
76 lblMensajes.Text = "Error de MySQL Server: " +
77 "<br>" + Excepciones.MensajePersonalizado;
78 lblMensajes.Visible = true;
79 }
80 catch (Exception excepcion){
81 Excepciones.Gestionar(excepcion, Plataforma.Web);
82 lblMensajes.Text = "Error de c#" +
83 "<br>" + Excepciones.MensajePersonalizado;
84 lblMensajes.Visible = true;
85 }
86 }
87 if (RP.SelectedValue == "AUSENTE" && txtValorFalta.Text ==
        "0.00"){
                                                    lblMensajes
        .Visible = true;
88 lblMensajes.Text = "Asigne valor de Ausencia.";
89 String txtJS = String.Format("<script>alert('{0}');</script
        >", "Asigne valor de Ausencia.");
                                                    ScriptManager.
        RegisterClientScriptBlock(Page, GetType(), "script",
        txtJS, false);
        txtValorFalta.Focus();
90 }
91 else if (RP.SelectedValue == "AUSENTE"){
92 try{
93 int idusuario = Convert.ToInt32(Id.Text);
94 asistencia.IdUsuario = idusuario;
95 int idreunion = Convert.ToInt32(cbReunion.SelectedItem.
        Value);
        asistencia.IdReunion = idreunion;
96 int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
                                                    asistencia.
        IdDirectivos = idDirectivo;
97 asistencia.FechaAsistencia = txtFecha.Text;

```

```

98  asistencia.Estado = RP.SelectedValue;
99  asistencia.ValorMulta = txtValorFalta.Text;
100 asistencia.EstadoDeuda = "PENDIENTE";
101 try{
102 int Idusuario2 = Convert.ToInt32(Id.Text);
103 int Idreunion2 = Convert.ToInt32(cbReunion.SelectedItem.
      Value);
104 int cant = dal.SelectRepetido(Idusuario2, Idreunion2);
105 if (cant >= 1){
106 lblMensajes.Text = "El usuario ha pasado la lista en la
      reunion seleccionada. Verifique si la reunion es la
      correcta o verifique el reporte de asistencias, para
      comprobar que usuarios han pasado la lista en la reunion
      seleccionada.";
107 String txtJS = String.Format("<script>alert('{0}')</script
      >", "El usuario ha pasado la lista en la reunion
      seleccionada. Verifique si la reunion es la correcta o
      verifique el reporte de asistencias, para comprobar que
      usuarios han pasado la lista en la reunion seleccionada.
      ");
      ScriptManager.RegisterClientScriptBlock(Page, GetType(),
      "script", txtJS, false);
      lblMensajes.
      Visible = true;
108 }
109 else{
110 dal.Insert(asistencia);
111 lblMensajes.Text = "El Usuario a Pasado la Lista.";
112 String txtJS = String.Format("<script>alert('{0}')</script
      >", "El Usuario a Pasado la Lista.");
      ScriptManager.
      RegisterClientScriptBlock(Page, GetType(), "script",
      txtJS, false);
      lblMensajes.Visible = true;
113 B.Visible = false;
114 }
115 }
116 catch{
117 lblMensajes.Text = "Ocurrio un problema al pasar la lista.
      Verifique la conexion a la base de datos.";

```

```

118 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ocurrio un problema al pasar la lista. Verifique la
    conexion a la base de datos.");

    ScriptManager.RegisterClientScriptBlock(Page, GetType(),
        "script", txtJS, false);

        lblMensajes.
            Visible = true;
119 }
120 }
121 catch (MySqlException excepcion){
122     Excepciones.Gestionar(excepcion, Plataforma.Web);
123     lblMensajes.Text = "Error de MySQL Server: " +
124     "<br>" + Excepciones.MensajePersonalizado;
125     lblMensajes.Visible = true;
126 }
127 catch (Exception excepcion){
128     Excepciones.Gestionar(excepcion, Plataforma.Web);
129     lblMensajes.Text = "Error de c#" +
130     "<br>" + Excepciones.MensajePersonalizado;
131     lblMensajes.Visible = true;
132 }
133 }
134 if (RP.SelectedValue == "PRESENTE"){
135     try{
136         int idusuario = Convert.ToInt32(Id.Text);
137         asistencia.IdUsuario = idusuario;
138         int idreunion = Convert.ToInt32(cbReunion.SelectedItem.
            Value);
            asistencia.IdReunion = idreunion;
139         int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
            asistencia.
                IdDirectivos = idDirectivo;
140         asistencia.FechaAsistencia = txtFecha.Text;
141         asistencia.Estado = RP.SelectedValue;
142         asistencia.ValorMultas = "0.00";
143         asistencia.EstadoDeuda = "NINGUNA";
144         try{
145             int Idusuario2 = Convert.ToInt32(Id.Text);
146             int Idreunion2 = Convert.ToInt32(cbReunion.SelectedItem.

```

```

        Value);
147 int cant = dal.SelectRepetido(Idusuario2 , Idreunion2);
148 if (cant >= 1){
149 lblMensajes.Text = "El usuario ha pasado la lista en la reunion seleccionada. Verifique si la reunion es la correcta o verifique el reporte de asistencias , para comprobar que usuarios han pasado la lista en la reunion seleccionada.";
150 String txtJS = String.Format("<script>alert('{0}');</script>", "El usuario ha pasado la lista en la reunion seleccionada. Verifique si la reunion es la correcta o verifique el reporte de asistencias , para comprobar que usuarios han pasado la lista en la reunion seleccionada.");
        ScriptManager.RegisterClientScriptBlock(Page, GetType(), "script", txtJS, false);
                                lblMensajes.
        Visible = true;
151 }
152 else{
153 dal.Insert(asistencia);
154 lblMensajes.Text = "El Usuario a Pasado la Lista.";
155 String txtJS = String.Format("<script>alert('{0}');</script>>", "El Usuario a Pasado la Lista.");
                                ScriptManager.
        RegisterClientScriptBlock(Page, GetType(), "script", txtJS, false);
        lblMensajes.Visible = true;
156 B.Visible = false;
157 }
158 }
159 catch{
160 lblMensajes.Visible = true;
161 lblMensajes.Text = "Ocurrio un problema al pasar la lista. Verifique la conexion a la base de datos.";
162 String txtJS = String.Format("<script>alert('{0}');</script>>", "Ocurrio un problema al pasar la lista. Verifique la conexion a la base de datos.");
163 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "script", txtJS, false);

```

```

164 }
165 }
166 catch (MySqlException excepcion){
167     Excepciones.Gestionar(excepcion, Plataforma.Web);
168     lblMensajes.Text = "Error de MySQL Server: " +
169     "<br>" + Excepciones.MensajePersonalizado;
170     lblMensajes.Visible = true;
171 }
172 catch (Exception excepcion){
173     Excepciones.Gestionar(excepcion, Plataforma.Web);
174     lblMensajes.Text = "Error de c# " +
175     "<br>" + Excepciones.MensajePersonalizado;
176     lblMensajes.Visible = true;
177 } } } } } } } }

```

- Método Modificar Asistencias

```

1  protected void btnModificar_Click(object sender, EventArgs
    e){
2  AsistenciaDAL dal = new AsistenciaDAL();
3  Asistencias asistenciaa = new Asistencias();
4  if (txtMontoPago.Text == "0.00" || txtMontoPago.Text == "0"
    ){
5  if (cbReunion.SelectedIndex == 0){
6  lblMensajes.Text = " Seleccione Reunion.";
7  lblMensajes.Visible = true;
8  String txtJS = String.Format("<script>alert('{0}');</script
    >", " Seleccione Reunion.");
9  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
10 }
11 else{
12 if (txtNombre.Text==""){
13 lblMensajes.Text = " Seleccione Usuario.";
14 lblMensajes.Visible = true;
15 String txtJS = String.Format("<script>alert('{0}');</script
    >", " Seleccione Usuario.");
16 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
    }

```

```

17 else{
18 if (cbEstado.SelectedIndex == 0){
19 lblMensajes.Text = " Seleccione Estado.";
20 lblMensajes.Visible = true;
21 String txtJS = String.Format("<script>alert('{0}');</script
    >", " Seleccione Estado.");
22 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
23 }
24 else{
25 if (cbEstado.SelectedValue == "PRESENTE" && txtValorAtraso.
    Text != "0.00" || cbEstado.SelectedValue == "PRESENTE"
    && txtValorFalta.Text != "0.00"){
26 lblMensajes.Text = " Los Valores Multa ATRASO y Multa
    AUSENTE deben ser 0.00 si el Estado del Usuario es
    PRESENTE.";
27 lblMensajes.Visible = true;
28 String txtJS = String.Format("<script>alert('{0}');</script
    >", " Los Valores Multa ATRASO y Multa AUSENTE deben ser
    0.00 - si el Estado del Usuario es PRESENTE.");
29 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
30 }
31 else{
32 if (cbEstado.SelectedValue == "ATRASADO" && txtValorAtraso.
    Text == "0.00" || cbEstado.SelectedIndex == 2 &&
    txtValorFalta.Text != "0.00"){
33 lblMensajes.Text = " Modificar Valor Multa ATRASO. VERIFIQUE
    QUE EL VALOR DE AUSENCIA sea 0.00 ";
    lblMensajes.Visible = true;
34 String txtJS = String.Format("<script>alert('{0}');</script
    >", " Modificar Valor Multa ATRASO. VERIFIQUE QUE EL
    VALOR DE AUSENCIA sea 0.00 ");
35 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
    }
36 else{
37 if (cbEstado.SelectedValue == "AUSENTE" && txtValorFalta.
    Text == "0.00" || cbEstado.SelectedIndex == 3 &&
    txtValorAtraso.Text != "0.00"){

```

```

38 lblMensajes.Text = "Modificar Valor Multa AUSENTE.
    VERIFIQUE QUE EL VALOR DE ATRASO sea 0.00 ";
                                                                    lblMensajes.
        Visible = true;
39 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Modificar Valor Multa AUSENTE. VERIFIQUE QUE EL
    VALOR DE ATRASO sea 0.00 ");
40 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
                                                                    }
41 else{
42 if (cbEstado.SelectedValue == "AUSENTE"){
43 try{
44 asistenciaa.ValorMulta = txtValorFalta.Text;
45 asistenciaa.Estado = cbEstado.SelectedValue;
46 asistenciaa.EstadoDeuda = "PENDIENTE";
47 asistenciaa.IdReunion = Convert.ToInt32(cbReunion.
    SelectedValue);
        asistenciaa.IdUsuario = Convert.ToInt32(txtUsuarioID.
        Text);
48 bool correcto = dal.Update(asistenciaa);
49 if (correcto){
50 lblMensajes.Text = "Registro actualizado";
        lblMensajes.Visible = true;
51 Limpiar();
52 cbReunion.Enabled = false;
53 btnBuscarUsuario.Focus();
54 int ReunionID3 = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
                                                                    int
        IdUsuario3 = Convert.ToInt32(txtUsuarioID.Text);
55 gvDatos.DataSource = dal.SelectBuscarvalor(ReunionID3,
    IdUsuario3);
        gvDatos.DataBind();
56 }
57 else{
58 lblMensajes.Text = "Problemas al actualizar. Verifique

```

```

        conexion_al_servidor.";

        lblMensajes.
        Visible = true;
59 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Problemas_al_actualizar.Verifique_conexion_al_
        servidor.");
60 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
        }
61 }
62 catch (MySqlException excepcion){

        Excepciones.Gestionar(excepcion, Plataforma.Web);

        lblMensajes.Text = "Error_de_MySqlServer:" +

        "<br>" + Excepciones.MensajePersonalizado;
63 }
64 catch (Exception excepcion){
65 Excepciones.Gestionar(excepcion, Plataforma.Web);

        lblMensajes.Text = "Error_de_c#:" +
66 "<br>" + Excepciones.MensajePersonalizado;
67 }
68 }
69 if (cbEstado.SelectedValue == "ATRASADO"){
70 try{
71 asistenciaa.ValorMulta = txtValorAtraso.Text;
72 asistenciaa.Estado = cbEstado.SelectedValue;
73 asistenciaa.EstadoDeuda = "PENDIENTE";
74 asistenciaa.IdReunion = Convert.ToInt32(cbReunion.
        SelectedValue);
75 asistenciaa.IdUsuario = Convert.ToInt32(txtUsuarioID.Text);
76 bool correcto = dal.Update(asistenciaa);
77 if (correcto){
78 lblMensajes.Text = "Registro_actualizado";

        lblMensajes.Visible = true;
79 Limpiar();
80 cbReunion.Enabled = false;

```

```

81 btnBuscarUsuario.Focus();
82 int ReunionID3 = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
                                                    int
    IdUsuario3 = Convert.ToInt32(txtUsuarioID.Text);
83 gvDatos.DataSource = dal.SelectBuscarvalor(ReunionID3,
    IdUsuario3);

    gvDatos.DataBind();
84 gvDatos.Visible = true;
85 }
86 else{
87 lblMensajes.Text = "Problemas al actualizar. Verifique
    conexion al servidor.";

    lblMensajes.Visible = true;
88 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Problemas al actualizar. Verifique conexion al
    servidor.");
89 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
                                                    }
90 }
91 catch (MySqlException excepcion){

    Excepciones.Gestionar(excepcion, Plataforma.Web);

    lblMensajes.Text = "Error de MySQLServer: " +

    "<br>" + Excepciones.MensajePersonalizado;
92 }
93 catch (Exception excepcion){
94 Excepciones.Gestionar(excepcion, Plataforma.Web);

    lblMensajes.Text = "Error de c# " +
95 "<br>" + Excepciones.MensajePersonalizado;
96 }
97 }
98 if (cbEstado.SelectedValue == "PRESENTE"){
99 try{

```

```

100 asistenciaa.ValorMulta = "0.00";
101 asistenciaa.Estado = cbEstado.SelectedValue;

        asistenciaa.EstadoDeuda = "NINGUNA";
102 asistenciaa.IdReunion = Convert.ToInt32(cbReunion.
        SelectedValue);

        asistenciaa.IdUsuario = Convert.ToInt32(txtUsuarioID.
        Text);
103 bool correcto = dal.Update(asistenciaa);
104 if (correcto){
105 lblMensajes.Text = "Registro▯actualizado";

        lblMensajes.Visible = true;
106 Limpiar();
107 cbReunion.Enabled = false;
108 btnBuscarUsuario.Focus();
109 int ReunionID3 = Convert.ToInt32(cbReunion.SelectedItem.
        Value);

                                                                 int
        IdUsuario3 = Convert.ToInt32(txtUsuarioID.Text);
110 gvDatos.DataSource = dal.SelectBuscarvalor(ReunionID3,
        IdUsuario3);

        gvDatos.DataBind();
111 }
112 else{
113 lblMensajes.Text = "Problemas▯al▯actualizar.▯Verifique▯
        conexion▯al▯servidor.";

                                                                 lblMensajes.
        Visible = true;
114 String txtJS = String.Format("<script>alert('{0}');</script
        >", "Problemas▯al▯actualizar.▯Verifique▯conexion▯al▯
        servidor.");
115 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);

                                                                 }
116 }
117 catch (MySqlException excepcion){

```

```

        Excepciones.Gestionar(excepcion , Plataforma.Web);

        lblMensajes.Text = "Error de MySQLServer: " +
            "<br>" + Excepciones.MensajePersonalizado;
118 }
119 catch (Exception excepcion){
120 Excepciones.Gestionar(excepcion , Plataforma.Web);

        lblMensajes.Text = "Error de c#" +
121 "<br>" + Excepciones.MensajePersonalizado;
122 } } } } } } } } } }
123 else{
124 lblMensajes.Text = "NO SE PUEDE MODIFICAR PORQUE SE HA
        ECHO COBRO DE ESTA MULTA. ";
        lblMensajes.
        Visible = true;
125 String txtJS = String.Format("<script>alert ('{0}');</script
        >", "NO SE PUEDE MODIFICAR PORQUE SE HA ECHO COBRO DE
        ESTA MULTA. ");
126 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
127 }
128 }

```

- Método Generar Informe

```

1 protected void btnGenerarInforme_Click(object sender ,
        EventArgs e){
2 AsistenciaDAL dal = new AsistenciaDAL();
3 if (txtIdreunion.Text==""){
4 lblMensajes.Text = " Seleccione Reunion. ";
5 lblMensajes.Visible = true;
6 String txtJS = String.Format("<script>alert ('{0}');</script
        >", " Seleccione Reunion. ");
7 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
8 }
9 else{

```

```

10 int ReunionID1 = Convert.ToInt32(txtIdreunion.Text);
11 int cant = dal.SelectconteoInformeGeneral(ReunionID1);
12 if (cant==0){
13     lblMensajes.Text = "La reunion seleccionada no tiene
        asistencias.";
14     lblMensajes.Visible = true;
15     String txtJS = String.Format("<script>alert('{0}');</script
        >", "La reunion seleccionada no tiene asistencias.");
16     ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);                               btnBuscar.
        Focus();
17     btnGenerarInforme.Visible = false;
18 }
19 else{
20     int ReunionID = Convert.ToInt32(txtIdreunion.Text);
21     gvDatos.DataSource = dal.SelectAsistenciaGeneral(ReunionID)
        ;
22     gvDatos.DataBind();
23     gvDatos.Visible = true;
24     btnGenerarInforme.Visible = false;
25 }
26 }
27 }

```

4.3.7.8. Pruebas de Aceptación

A continuación, se muestra las pruebas de aceptación de la Tercera Iteración:

CASO DE PRUEBA	
Código: 6	Nº Historia de Usuario: 6
Historia de Usuario: Control de Asistencia.	
Condiciones de Ejecución: Solo el cargo de secretario podrá realizar el control de asistencia.	
<p>Entrada/Pasos de Ejecución:</p> <p>Control de Asistencia Seleccionar la pestaña Asistencia, la opción Control de Asistencia. Seleccionar la Reunión Activa generada anteriormente por el presidente, dar clic en el botón Listar Usuarios si la reunión corresponde a los Usuarios, caso contrario dar clic en el botón Listar Directivos si la reunión corresponde para los Directivos. Para continuar con el control de asistencia se debe seleccionar el estado correcto del usuario, con el cual se podrá asignar al usuario el estado de Presente si llego puntual, Atrasado si llego después del tiempo que se estableció para la reunión o Ausencia si el usuario no asistió a la reunión. El control de asistencia podrá identificar si el usuario paso la lista anteriormente, esto servirá para el correcto control de asistencia, también el secretario deberá asignar los valores de multa de atraso y ausencia con la fecha de la reunión. Luego seleccionar la opción Pasar Lista, este paso finalizara el control de asistencia. En la base de datos se podrá observar que los valores de atraso y ausencia se sumarian en un valor acumulado para posteriormente realizar el cobro de las multas por separado o realizar el cobro del valor acumulado de las multas.</p> <p>Modificar la Asistencia Seleccionar la pestaña Asistencia, la opción Modificar Asistencia. Seleccionar la reunión y el usuario y dar clic en el botón Buscar Asistencia para buscar la información de la asistencia mal ingresada. Modificar los valores de multa y el estado de la asistencia mal ingresada. Por último, seleccionar la opción modificar para finalizar el proceso de modificar asistencia. Si la asistencia tiene valores de cobro en el caso de Atraso o Ausencia, la aplicación web no le permitirá realizar la modificación para evitar problemas monetarios.</p> <p>Informes del Control de Asistencia 1) Seleccionar la pestaña Asistencia, la opción Informe General. Seleccionamos la reunión y damos clic en el botón Generar Informe. Este proceso mostrara la información del control de asistencia que se realizó en la reunión seleccionada. 2) Seleccionar la pestaña Asistencia, la opción Informe de Asistencia Por Usuario. Seleccionamos el usuario y damos clic en el botón Generar Informe. Este proceso mostrara la información de todas las asistencias que le corresponden al usuario seleccionado. 3) Seleccionar la pestaña Asistencia, la opción Informe Por Estado. Seleccionamos el estado Atraso o Ausencia y damos clic en el botón Generar Informe. Este proceso mostrara la información de todos los usuarios Atrasados, Ausentes o Presentes.</p>	
<p>Resultado Esperado: La información del control de asistencia será guardada en la base de datos. La información de la asistencia del usuario podrá ser modificada a excepción de las asistencias que tienen valores de cobro. Finalizar el control de asistencia sin inconsistencias.</p>	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.83: Caso de Prueba: Control de Asistencia.

Elaborado por: Oscar Curichumbi.

4.3.7.9. Resultados de la Tercera Iteración

Como resultado de la entrega de la tercera iteración que consta del módulo de Control de Asistencia, el cliente quedó satisfecho con las funcionalidades del módulo que se desarrolló, la tercera iteración se ha concluido sin ningún problema manifestado por el cliente. A continuación, se realizará la Cuarta iteración.

4.3.8. Desarrollo de la Cuarta Iteración

Se realizará la descripción general de las tareas respectivas de la séptima historia de usuario, que pertenece a la cuarta iteración. Como se indica en la planificación se desarrollará el módulo de Tesorería. En la Tabla 4.83 se muestra de manera general las tareas de ingeniería correspondientes a la historia de usuario número 7 para cumplir con la planificación.

Numero de Tarea	Numero de Historia	Esfuerzo	Nombre de la Tarea
34	7	4	Diseño de la base de datos para la Tesorería.
35	7	5	Diseño de la interfaz para la Gestión de la Tesorería.
36	7	3	Validación de datos para registrar correctamente la información de la tesorería.
37	7	6	Guardar la información del registro de las colaboraciones en la base de datos.
38	7	4	Guardar la información del registro de las sanciones en la base de datos.
39	7	6	Crear Triggers para el control de cobros de las multas de asistencia, colaboraciones y sanciones de los usuarios.
40	7	5	Guardar la información de los cobros de las multas de asistencia en la base de datos.
41	7	7	Guardar la información de los cobros de las colaboraciones en la base de datos
42	7	8	Guardar la información de los cobros de las sanciones en la base de datos
43	7	8	Guardar la información del descuento en las multas de la asistencia.
44	7	7	Informes de los cobros de las multas de asistencia, colaboraciones y sanciones.

Tabla 4.84: Tareas a Realizar de la Historia de Usuario 7.

Elaborado por: Oscar Curichumbi.

4.3.8.1. Estimación de la Cuarta Iteración

A continuación, se muestra la estimación de la Cuarta Iteración:

ITERACIÓN 4				
N.- Historia	Fecha Inicio	Fecha Final	DÍAS	HORAS
7	03/06/2019	04/08/2019	63	252
TOTAL ESTIMACIÓN:			63	252

Tabla 4.85: Horas asignadas para el desarrollo de la Iteración 4.

Elaborado por: Oscar Curichumbi.

4.3.8.2. Bocetos de la Historia de Usuario 7

En las siguientes figuras se muestran los bocetos de la Historia de Usuario 7 que forman parte de la Cuarta Iteración.

The wireframe is titled "Registrar Colaboraciones" and consists of two main parts:

Form Fields:

- Reunión: [] [v]
- Motivo colaboración: []
- Valor colaboración: [0.00]
- Fecha: [18-10-2019]

Buttons:

- Listar usuarios
- confirmar
- cancelar

User List Table:

ID	Nombre completo	
1	Oscar Curichumbi	confirmar
2	Jorge Masagviza	confirmar

Figura 4.55: Boceto de Registro de Colaboraciones.

Elaborado por: Oscar Curichumbi.

Registrar Sanciones

Usuario:	<input type="text"/>	<input type="button" value="Buscar Usuario"/>
Reunión:	<input type="text"/>	<input type="button" value="Confirmar"/>
Motivo Sanción:	<input type="text"/>	<input type="button" value="Cancelar"/>
Valor Sanción:	0.00	
Fecha:	18-10-2019	

ID	Motivo Reunión	Motivo Sanción	Saldo Deuda	Estado

Figura 4.56: Boceto de Registro de Sanciones.
Elaborado por: Oscar Curichumbi.

Cobro Multas Por Usuario

Usuario:	<input type="text"/>	<input type="button" value="Buscar usuario"/>
Nombre de quien cancela:	<input type="text"/>	<input type="button" value="Buscar Multas"/>
Monto Pago:	0.00	<input type="button" value="Confirmar"/>
Fecha:	18-10-2019	<input type="button" value="Nuevo Cobro"/>

Reunión	Estado	Saldo Deuda	
Por colaboración	Atrasado	1,00	<input type="button" value="Cobrar"/>

Figura 4.57: Boceto de Cobro de Multas Por Usuario.
Elaborado por: Oscar Curichumbi.

Cobro Multas Por Reunión

Reunión:	<input type="text"/>	Buscar Reunión
Nombre de quien cancela:	<input type="text"/>	Buscar Multas
Monto Pago:	0.00	Confirmar
Fecha:	18-10-2019	Nuevo

Usuarios	Estado	Saldo Deuda	
Oscar Eduardo Curichumbi	Atrasado	1,00	Cobrar

Figura 4.58: Boceto de Cobro de Multas Por Reunión.
Elaborado por: Oscar Curichumbi.

Cobro colaboración Por Usuario

Usuario:	<input type="text"/>	Buscar Usuario
Nombre de quien cancela:	<input type="text"/>	Buscar Cobberación
Monto Pago:	0.00	Confirmar
Fecha:	10-08-2019	Nuevo

Motivo colaboración	Saldo Deuda	
Compra Muebles	5,00	Cobrar

Figura 4.59: Boceto de Cobro de Colaboraciones Por Usuario.
Elaborado por: Oscar Curichumbi.

Cobro Colaboracion Por Reunion

Reunion:	<input type="text"/>	Buscar Reunion
Nombre de quien cancela:	<input type="text"/>	Buscar Colaboracion
Monto Pago:	0.00	confirmar
Fecha:	10-08-2019	Nuevo

usuarios	Motivo Colaboracion	Saldo	
oscar curichumbi	Para, Comprar cuadernos	5,00	Cobrar

Figura 4.60: Boceto de Cobro de Colaboraciones Por Reunion.
Elaborado por: Oscar Curichumbi.

Cobro Sancion Por Usuario

Usuario:	<input type="text"/>	Buscar Usuario
Nombre de quien cancela:	<input type="text"/>	Buscar Sancion
Monto Pago:	0.00	confirmar
Fecha:	10-08-2019	Nuevo

Motivo Sancion	Saldo	
Por Mal ingreso	20,00	Cobrar

Figura 4.61: Boceto de Cobro de Sancion Por Usuario.
Elaborado por: Oscar Curichumbi.

Informe Asistencia Por Reunión

Reunión:

Buscar Reunión

Generar Informe

cancelar

Usuarios	Reunión	Fecha	Estado	Saldo	Estado Deuda

Figura 4.62: Boceto de Informe de Asistencia Por Reunión.
Elaborado por: Oscar Curichumbi.

Informe Asistencia Por Usuario

Usuario:

Buscar Reunión

Generar Informe

Nueva búsqueda

Reunión	Fecha	Estado	Saldo	Estado Deuda

Figura 4.63: Boceto de Informe de Asistencia Por Usuario.
Elaborado por: Oscar Curichumbi.

Informe Asistencia Por Estado

Estado:

Usuario	Reunión	Fecha	Estado	Saldo	Estado Deuda

Figura 4.64: Boceto de Informe de Asistencia Por Estado.
Elaborado por: Oscar Curichumbi.

Informe Cobro Asistencia Por Usuario

Usuario:

ID	Pagado por	Cobrado por	Valor cobro	Fecha Cobro	Reunion	Estado

Figura 4.65: Boceto de Informe Cobro de Asistencia Por Usuario.
Elaborado por: Oscar Curichumbi.

Informe cobro Asistencia Por Reunión

Reunión:

usuario Deudor	Pagado por	cobrado por	Valor	Fecha	Reunión multa	Estado

Figura 4.66: Boceto de Informe Cobro de Asistencia Por Reunión.
Elaborado por: Oscar Curichumbi.

Informe Colaboración Por Reunión

Reunión:

Usuarios	Motivo Colaboración	Fecha	Saldo	Estado Deuda

Figura 4.67: Boceto de Informe Colaboración Por Reunión.
Elaborado por: Oscar Curichumbi.

Informe colaboración Por Usuario

Usuario:

Buscar Usuario

Generar Informe

Nueva Búsqueda

Reunión	Motivo Colaboración	Fecha	Saldo	Estado Deuda

Figura 4.68: Boceto de Informe Colaboración Por Usuario.
Elaborado por: Oscar Curichumbi.

Informe Cobro colaboración por Usuario

Usuario:

Buscar Usuario

Generar Informe

Nueva Búsqueda

ID	Pagado por	Cobrado por	Valor	Fecha	Reunión	Estado Deuda

Figura 4.69: Boceto de Informe Cobro de Colaboración Por Usuario.
Elaborado por: Oscar Curichumbi.

Informe Cobro Colaboración Por Reunión

Reunión:

Buscar Reunión

Generar Informe

Nueva Búsqueda

Usuario Deudor	Pagado por	Cobrado por	Valor	Fecha	Reunión	Estado Deuda

Figura 4.70: Boceto de Informe Cobro de Colaboración Por Reunión.
Elaborado por: Oscar Curichumbi.

Informe General de Sanciones

ID	Usuarios	Reuniones	Motivo, Sanción	Saldo Deuda	Estado Deuda

Figura 4.71: Boceto de Informe General de Sanciones.
Elaborado por: Oscar Curichumbi.

Informe Cobro de Sanciones

Usuario Deudor	Pagado por	Cobrado por	Valor	Fecha	Reunión	Sanción	Estado Deuda

Figura 4.72: Boceto de Informe Cobro de Sanciones.
Elaborado por: Oscar Curichumbi.

Descuento De Multas

Usuario:	<input type="text"/>	<input type="button" value="Buscar Usuario"/>
Valor Total Deuda:	18.00	<input type="button" value="Calcular Multa"/>
Reunión:	<input type="text"/>	<input type="button" value="Buscar Reunión"/>
Valor Descuento:	0.00	<input type="button" value="Generar Descuento"/>
Motivo Descuento:	<input type="text"/>	<input type="button" value="Nuevo Registro"/>
Fecha:	25-10-2019	

Reunión	Saldo Deuda	Estado

Figura 4.73: Boceto de Descuentos de Multas.
Elaborado por: Oscar Curichumbi.

Informe General de Descuentos

Usuarios	Reunión	Motivo Descuento	Valor Descuento	Deuda Total	Fecha	Ingresado Por

Figura 4.74: Boceto de Informe General de Descuentos.
Elaborado por: Oscar Curichumbi.

4.3.8.3. Descripción de las Tareas de Ingenierías

En base a la planificación se ha realizado las tareas de ingeniería, en estas se ha asignado el tiempo de estimación para el desarrollo de cada una de ellas. Si el cliente está satisfecho con el desarrollo de las tareas, que forman parte de la historia de usuario, esto cumplirá con las necesidades del cliente y de este modo

se culminará la cuarta iteración. A continuación, se muestra el desarrollo de las tareas de la historia de usuario.

Tarea de Ingeniería	
Numero de Tarea: 34	Numero de Historia: 7
Nombre de Tarea: Diseño de la base de datos para la Tesorería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 03/06/2019	Fecha Fin: 06/06/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se creará las tablas de la tesorería en la base de datos, para registrar la información de los procesos de la tesorería.	

Tabla 4.86: Tarea 34 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 35	Numero de Historia: 7
Nombre de Tarea: Diseño de la interfaz para la Gestión de la Tesorería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Fecha Inicio: 07/06/2019	Fecha Fin: 11/06/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el tesorero(a) o los directivos asignados podrán ingresar la información de la tesorería de la comunidad.	

Tabla 4.87: Tarea 35 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 36	Numero de Historia: 7
Nombre de Tarea: Validación de datos para registrar correctamente la información de la tesorería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 12/06/2019	Fecha Fin: 14/06/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere validar los datos de la tesorería antes de ingresar a la base de datos, para no ingresar datos nulos o datos erróneos.	

Tabla 4.88: Tarea 36 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 37	Numero de Historia: 7
Nombre de Tarea: Guardar la información del registro de las colaboraciones en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6
Fecha Inicio: 15/06/2019	Fecha Fin: 20/06/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información de las colaboraciones, para registrarlos en la base de datos y realizar el respectivo cobro.	

Tabla 4.89: Tarea 37 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 38	Numero de Historia: 7
Nombre de Tarea: Guardar la información del registro de las sanciones en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 21/06/2019	Fecha Fin: 24/06/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información de las sanciones, para registrarlos en la base de datos y realizar el respectivo cobro.	

Tabla 4.90: Tarea 38 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 39	Numero de Historia: 7
Nombre de Tarea: Crear Triggers para el control de cobros de las multas de asistencia, colaboraciones y sanciones de los usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6
Fecha Inicio: 25/06/2019	Fecha Fin: 30/06/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Los Triggers ayudaran a controlar los pagos después de ingresar datos en las tablas de cobros, para poder controlar de mejor manera los pagos.	

Tabla 4.91: Tarea 39 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 40	Numero de Historia: 7
Nombre de Tarea: Guardar la información de los cobros de las multas de asistencia en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Fecha Inicio: 01/07/2019	Fecha Fin: 05/07/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información de los cobros de las multas de asistencia.	

Tabla 4.92: Tarea 40 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 41	Numero de Historia: 7
Nombre de Tarea: Guardar la información de los cobros de las colaboraciones en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7
Fecha Inicio: 06/07/2019	Fecha Fin: 12/07/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información de los cobros de las colaboraciones.	

Tabla 4.93: Tarea 41 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 42	Numero de Historia: 7
Nombre de Tarea: Guardar la información de los cobros de las sanciones en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8
Fecha Inicio: 13/07/2019	Fecha Fin: 20/07/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información de los cobros de las sanciones.	

Tabla 4.94: Tarea 42 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 43	Numero de Historia: 7
Nombre de Tarea: Guardar la información del descuento en las multas de la asistencia.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8
Fecha Inicio: 21/06/2019	Fecha Fin: 28/07/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información del descuento de las multas de la asistencia. Este descuento será únicamente si los usuarios de la comunidad lo aprueban, esta aprobación debe constar en un acta, se detallará en un campo la información de la reunión y el número de acta en donde se confirma la aprobación del descuento.	

Tabla 4.95: Tarea 43 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 44	Numero de Historia: 7
Nombre de Tarea: Informes de los registros y cobros de las multas de asistencia, colaboraciones y sanciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7
Fecha Inicio: 29/07/2019	Fecha Fin: 04/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se realizará la visualización de los informes de los registros del control de asistencia, colaboraciones y sanciones, también se realizará informes que permitan visualizar los cobros de las multas de asistencia, colaboraciones y sanciones, estos informes serán por usuario y reuniones.	

Tabla 4.96: Tarea 44 de la Historia de Usuario N.-7.

Elaborado por: Oscar Curichumbi.

4.3.8.4. Tarjetas CRC: Historias de usuario 7

En las siguientes tablas se muestran las tarjetas CRC, estas tarjetas son de gran utilidad para la Cuarta Iteración.

Cobroasistencia	
Responsabilidad	Colaboración
Atributos:	Asistencia
Id_CobroAsistencia	Usuarios
Id_Asistencia	Directivos
Id_Usuario	
Id_Directivos	
NombrePersonaPaga	
MontoPago	
FechaPago	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.97: Tarjeta CRC: Cobroasistencia.

Elaborado por: Oscar Curichumbi.

Descuentomultas	
Responsabilidad	Colaboración
Atributos:	Usuarios
Id_DescuentoMultas	Directivos
Id_Usuario	Reuniones
Id_Directivos	
Id_Reunion	
MotivoReunion	
ValorDescuento	
ValorDeudaTotal	
FechaDescuento	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.98: Tarjeta CRC: Descuentomultas.

Elaborado por: Oscar Curichumbi.

Colaboraciones	
Responsabilidad	Colaboración
Atributos:	Reuniones
Id_Colaboracion	Usuarios
Id_Reunion	
Id_Usuario	
MotivoColaboracion	
ValorColaboracion	
MontoPago	
SaldoTotal	
Estado	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.99: Tarjeta CRC: Colaboraciones.

Elaborado por: Oscar Curichumbi.

Cobrocolaboracion	
Responsabilidad	Colaboración
Atributos:	Colaboraciones
Id_CobroColaboracion	Directivos
Id_Colaboracion	
Id_Directivos	
NombrePersonaPaga	
MontoPago	
FechaPago	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.100: Tarjeta CRC: Cobrocolaboracion.

Elaborado por: Oscar Curichumbi.

Sanciones	
Responsabilidad	Colaboración
Atributos:	Reuniones
Id_Sancion	Usuarios
Id_Reunion	
Id_Usuario	
MotivoSancion	
ValorSancion	
MontoPago	
SaldoTotal	
Estado	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.101: Tarjeta CRC: Sanciones.

Elaborado por: Oscar Curichumbi.

Cobrosancion	
Responsabilidad	Colaboración
Atributos:	Reuniones
Id_CobroSancion	Usuarios
Id_Sancion	
Id_Directivos	
NombrePersonaPaga	
MontoPago	
FechaPago	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.102: Tarjeta CRC: Cobrosancion.

Elaborado por: Oscar Curichumbi.

4.3.8.5. Diagrama de datos

A continuación, se muestra el diagrama de datos de la Cuarta Iteración:

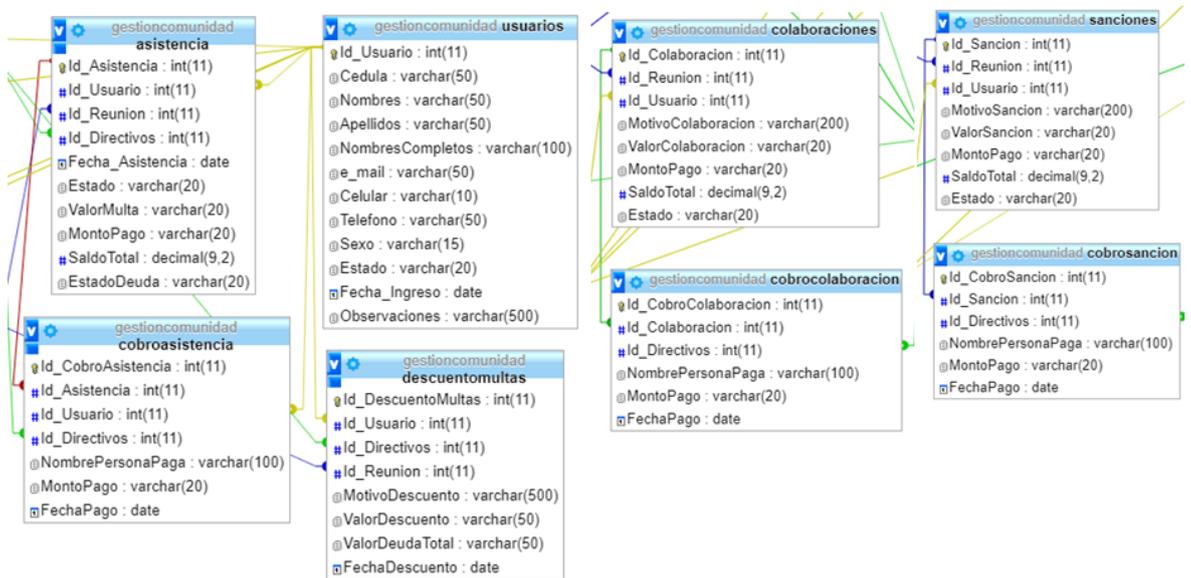


Figura 4.75: Diagrama de datos: Tesorería.

Elaborado por: Oscar Curichumbi.

4.3.8.6. Diseño de Interfaz

A continuación, se muestra los diseños de interfaz de la Cuarta Iteración:

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

REGISTRAR COLABORACIONES

Reunion: REUNION DE DIRECTIVOS
Motivo Colaboracion:
Valor Colaboracion: 0.00
Fecha: 18-10-2019

ID	NOMBRE COMPLETO	COLABORACION
1	OSCAR EDUARDO CURICHUMBI JEREZ	
2	JORGE MANUEL MASAQUIZA MASAQUIZA	
3	EDUARDO JEREZ	

Figura 4.76: Diseño de Interfaz-Registro de Colaboraciones.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

REGISTRAR SANCIONES

Usuario: OSCAR EDUARDO CURICHUMBI JEREZ
Reunion: REUNION GENERAL
Motivo Sancion:
Valor Sancion: 0.00
Fecha: 18-10-2019

Figura 4.77: Diseño de Interfaz-Registro de Sanciones.
 Elaborado por: Oscar Curichumbi.

COBRO MULTAS POR USUARIO

USUARIO:	<input type="text" value="EDUARDO JEREZ"/>
NOMBRE DE QUIEN CANCELA:	<input type="text"/>
MONTO PAGO:	<input type="text" value="0.00"/>
FECHA:	<input type="text" value="18-10-2019"/>
<input type="button" value="Nuevo Cobro"/>	

REUNION	ESTADO	SALDO DEUDA	COBRAR
POR PRUEBA DIRECTIVO4	ATRASADO	1,00	<input type="button" value="Cobrar"/>

Figura 4.78: Diseño de Interfaz-Cobro de Multas de Asistencia por Usuario.
 Elaborado por: Oscar Curichumbi.

COBRO MULTAS POR REUNION

REUNION:	<input type="text" value="POR PRUEBA DIRECTIVO4"/>
NOMBRE DE QUIEN CANCELA:	<input type="text"/>
MONTO PAGO:	<input type="text" value="0.00"/>
FECHA:	<input type="text" value="18-10-2019"/>
<input type="button" value="Nuevo"/>	

USUARIOS	ESTADO	SALDO DEUDA	COBRAR
EDUARDO JEREZ	ATRASADO	1,00	<input type="button" value="Cobrar"/>

Figura 4.79: Diseño de Interfaz-Cobro de Multas de Asistencia por Reunión.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Documentos Movimientos
 Reportes

COBRO COLABORACION POR USUARIO

USUARIO:
 NOMBRE DE QUIEN CANCELA:
 MONTO PAGO:
 FECHA:

MOTIVO COLABORACION	SALDO DEUDA	COBRAR
PARA COMPRAR VIVERES	2,00	<input type="button" value="Cobrar"/>
PARA COMPRAR COMIDA PARA EL EVENTO DE LA REINA	10,00	<input type="button" value="Cobrar"/>

Figura 4.80: Diseño de Interfaz-Cobro de Colaboraciones por Usuario.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Documentos Movimientos
 Reportes

COBRO COLABORACIONES POR REUNION

REUNION:
 NOMBRE DE QUIEN CANCELA:
 MONTO PAGO:
 FECHA:

USUARIOS	MOTIVOCOLABORACION	SALDO DEUDA	COBRAR
EDUARDO JEREZ	PARA COMPRAR COMIDA PARA EL EVENTO DE LA REINA	10,00	<input type="button" value="Cobrar"/>

Figura 4.81: Diseño de Interfaz-Cobro de Colaboraciones por Reunión.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Pagina Principal | Administracion | Reuniones | Asistencia | Tesoreria | Informes Tesoreria | Documentos | Movimientos

Reportes

COBRO SANCION POR USUARIO

USUARIO:

NOMBRE DE QUIEN CANCELA:

MONTO PAGO:

FECHA:

MOTIVO SANCION	SALDO DEUDA	COBRAR
POR MAL INGRESO DE FONDOS	20,00	<input type="button" value="Cobrar"/>

Figura 4.82: Diseño de Interfaz-Cobro de Sanciones por Usuario.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Pagina Principal | Administracion | Reuniones | Asistencia | Tesoreria | Informes Tesoreria | Documentos | Movimientos

Reportes

INFORME ASISTENCIA GENERAL

Reunion:

USUARIOS	REUNION	FECHA	ESTADO	SALDO DEUDA	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	POR PRUEBA DIRECTIVO4	21/08/2019	ATRASADO	0,00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	POR PRUEBA DIRECTIVO4	21/08/2019	ATRASADO	0,00	CANCELADO
EDUARDO JEREZ	POR PRUEBA DIRECTIVO4	21/08/2019	ATRASADO	1,00	PENDIENTE

Figura 4.83: Diseño de Interfaz-Informe de Asistencia por Reunión.
Elaborado por: Oscar Curichumbi.

INFORME ASISTENCIA POR USUARIO

Usuario:

USUARIO	REUNION	FECHA	ESTADO	SALDO DEUDA	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION PARA GENERAR IDEAS D CONSULTA	14/08/2019	ATRASADO	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION SOBRE LA FISCALIZACION 2019	15/08/2019	ATRASADO	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION DE DIRECTIVOS	20/08/2019	ATRASADO	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION GENERAL	20/08/2019	ATRASADO	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	PRUEBA DE REUNIONES	20/08/2019	ATRASADO	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	POR ELECCIONES DEL NUEVO PRESIDENTE	02/08/2019	ATRASADO	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	PRUEBA DE COBROS	05/08/2019	AUSENTE	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION PRUEBA 2	05/08/2019	AUSENTE	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	POR PRUEBA DIRECTIVO4	21/08/2019	ATRASADO	0,00	CANCELADO

Figura 4.84: Diseño de Interfaz-Informe de Asistencia por Usuario.
 Elaborado por: Oscar Curichumbi.

INFORME ASISTENCIA POR ESTADO

Estado:

USUARIO	REUNION	FECHA	ESTADO	SALDO DEUDA	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	PRUEBA DE COBROS	05/08/2019	AUSENTE	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION PRUEBA 2	05/08/2019	AUSENTE	0,00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	POR ELECCIONES DEL NUEVO PRESIDENTE	02/08/2019	AUSENTE	0,00	CANCELADO

Figura 4.85: Diseño de Interfaz-Informe de Asistencia por Estado.
 Elaborado por: Oscar Curichumbi.

INFORME COBRO ASISTENCIA POR USUARIO

USUARIO:

ID	PAGADO POR	COBRADO POR	VALOR COBRADO	FECHA COBRO	REUNION	ESTADO ASISTENCIA	MULTA	ESTADO DEUDA
1	NOGALES	OSCAR EDUARDO CURICHUMBI JEREZ	1.00	05/08/2019	POR ELECCIONES DEL NUEVO PRESIDENTE	ATRASADO	1.00	CANCELADO
4	NOGALES	OSCAR EDUARDO CURICHUMBI JEREZ	3.00	05/08/2019	PRUEBA DE COBROS	AUSENTE	10.00	CANCELADO
4	NOGALES	OSCAR EDUARDO CURICHUMBI JEREZ	7.00	05/08/2019	PRUEBA DE COBROS	AUSENTE	10.00	CANCELADO
7	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	3.00	05/08/2019	REUNION PRUEBA 2	AUSENTE	10.00	CANCELADO
7	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	2.00	07/08/2019	REUNION PRUEBA 2	AUSENTE	10.00	CANCELADO
7	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	2.50	06/08/2019	REUNION PRUEBA 2	AUSENTE	10.00	CANCELADO

Figura 4.86: Diseño de Interfaz-Informe Cobro de Asistencia por Usuario.
Elaborado por: Oscar Curichumbi.

INFORME COBRO ASISTENCIA POR REUNION

REUNION:

USUARIO DEUDOR	PAGADO POR	COBRADO POR	VALOR COBRADO	FECHA COBRO	REUNION	ESTADO ASISTENCIA	MULTA	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	NOGALES	OSCAR EDUARDO CURICHUMBI JEREZ	1.00	05/08/2019	POR ELECCIONES DEL NUEVO PRESIDENTE	ATRASADO	1.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	NOGALES	OSCAR EDUARDO CURICHUMBI JEREZ	2.00	05/08/2019	POR ELECCIONES DEL NUEVO PRESIDENTE	AUSENTE	10.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	NOGALES	OSCAR EDUARDO CURICHUMBI JEREZ	3.00	05/08/2019	POR ELECCIONES DEL NUEVO PRESIDENTE	AUSENTE	10.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	NOGALES	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	05/08/2019	POR ELECCIONES DEL NUEVO PRESIDENTE	AUSENTE	10.00	CANCELADO

Figura 4.87: Diseño de Interfaz-Informe Cobro de Asistencia por Reunión.
Elaborado por: Oscar Curichumbi.

INFORME COLABORACION POR REUNION

REUNION:

USUARIOS	REUNION	FECHA	SALDO DEUDA	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION DE DIRECTIVOS	24/05/2019	0,00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	REUNION DE DIRECTIVOS	24/05/2019	0,00	CANCELADO
EDUARDO JEREZ	REUNION DE DIRECTIVOS	24/05/2019	10,00	PENDIENTE

Figura 4.88: Diseño de Interfaz-Informe de Colaboración por Reunión.
Elaborado por: Oscar Curichumbi.

INFORME COLABORACIONES POR USUARIO

USUARIO:

USUARIO	REUNION	FECHA	SALDO DEUDA	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION PARA GENERAR IDEAS D CONSULTA	03/05/2019	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION SOBRE LA FISCALIZACION 2019	20/05/2019	0,00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	REUNION DE DIRECTIVOS	24/05/2019	0,00	CANCELADO

Figura 4.89: Diseño de Interfaz-Informe Colaboración por Usuario.
Elaborado por: Oscar Curichumbi.

INFORME COBRO COLABORACIONES POR USUARIO

USUARIO:

ID	PAGADO POR	COBRADO POR	VALOR COBRADO	FECHA COBRO	REUNION	COLABORACION	ESTADO DEUDA
1	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	15/08/2019	REUNION PARA GENERAR IDEAS D CONSULTA	5.00	CANCELADO
4	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	1.55	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO
4	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	3.25	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO
4	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	0.20	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO
7	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	10.00	21/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
7	MANUEL	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	21/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO

Figura 4.90: Diseño de Interfaz-Informe Cobro de Colaboración por Usuario.
 Elaborado por: Oscar Curichumbi.

INFORME COBRO COLABORACION POR REUNION

REUNION:

USUARIO DEUDOR	PAGADO POR	COBRADO POR	VALOR COBRADO	FECHA COBRO	REUNION	COLABORACION	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	1.55	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	0.20	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	3.25	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	1.55	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	0.20	15/08/2019	REUNION SOBRE LA FISCALIZACION 2019	5.00	CANCELADO

Figura 4.91: Diseño de Interfaz-Informe Cobro de Colaboración por Reunión.
 Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Documentos Movimientos

Reportes

INFORME GENERAL DE SANCIONES

ID	USUARIOS	REUNION	MOTIVO SANCION	SALDO DEUDA	ESTADO DEUDA
1	OSCAR EDUARDO CURICHUMBI JEREZ	REUNION DE DIRECTIVOS	POR MAL INGRESO DE FONDOS	0,00	CANCELADO
2	JORGE MANUEL MASAQUIZA MASAQUIZA	REUNION DE DIRECTIVOS	POR MAL INGRESO DE FONDOS	0,00	CANCELADO
3	EDUARDO JEREZ	REUNION DE DIRECTIVOS	POR MAL INGRESO DE FONDOS	20,00	PENDIENTE

Figura 4.92: Diseño de Interfaz-Informe de Sanciones.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Documentos Movimientos

Reportes

INFORME COBRO DE SANCIONES

USUARIO DEUDOR	PAGADO POR	COBRADO POR	VALOR COBRADO	FECHA COBRO	REUNION	SANCION	ESTADO DEUDA
OSCAR EDUARDO CURICHUMBI JEREZ	OSCAR	OSCAR EDUARDO CURICHUMBI JEREZ	10.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	10.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	OSCAR	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
OSCAR EDUARDO CURICHUMBI JEREZ	OSCAR	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	2.35	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	1.50	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	0.15	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	1.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO

Figura 4.93: Diseño de Interfaz-Informe Cobro de Sanciones.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Documentos Movimientos
Reportes

DESCUENTO DE MULTAS

USUARIO: JORGE MANUEL MASAQUIZA MASAQUIZA
 VALOR TOTAL DEUDA: 1.00
 REUNION: PRUEBA DE COBROS
 VALOR DESCUENTO: 0.00
 MOTIVO DESCUENTO: Registrar Descuento
 FECHA: 25-10-2019 Nuevo Registro

Ingrese valor del descuento.

REUNION	SALDO DEUDA	ESTADO
COBROS 3	1,00	PENDIENTE

Figura 4.94: Diseño de Interfaz-Registro de Descuentos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Documentos Movimientos
Reportes

INFORME GENERAL DE DESCUENTOS

USUARIOS	REUNION	MOTIVO DESCUENTO	VALOR DE DESCUENTO	DEUDA TOTAL	FECHA	INGRESADO POR
EDUARDO JEREZ	COBROS 3	PRIMER DESCUENTO	2.50	2.50	24/10/2019	OSCAR EDUARDO CURICHUMBI JEREZ

Figura 4.95: Diseño de Interfaz-Informe De Descuentos.
Elaborado por: Oscar Curichumbi.

4.3.8.7. Codificación

- Método Registrar Colaboraciones

```

1 protected void gvCargarDatos_RowCommand(object sender ,
      GridViewCommandEventArgs e){
2 ColaboracionDAL dal = new ColaboracionDAL();
3 Colaboracion colaboracion = new Colaboracion();
4 if (e.CommandName == "Select"){
5 int rowIndex = Convert.ToInt32(e.CommandArgument);

```

```

6 GridViewRow row = gvCargarDatos.Rows[rowIndex];
7 Label Id = row.FindControl("ID") as Label;
8 Button B = row.FindControl("bntColaboracion") as Button;
9 if (cbReunion.SelectedIndex == 0){
10 lblMensajes.Visible = true;
11 lblMensajes.Text = "Seleccione la Reunion para continuar
    con el proceso de asignar las colaboraciones.";
12 String txtJS = String.Format("<script>alert('{0}')</script
    >", "Seleccione la Reunion para continuar con el proceso
    de asignar las colaboraciones.");
13 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
14 }
15 else{
16 CultureInfo culture = new CultureInfo("en-GB");
17 decimal argument = Convert.ToDecimal(txtValorColaboracion.
    Text, culture);
18 int count = BitConverter.GetBytes(decimal.GetBits(argument)
    [3])[2];
19 if (count > 2){
20 lblMensajes.Text = "Ingrese solo dos decimales.";
21 lblMensajes.Visible = true;
22 txtValorColaboracion.Focus();
23 }
24 else{
25 if (txtMotivoColaboracion.Text == ""){
26 lblMensajes.Visible = true;
27 lblMensajes.Text = "Asigne Motivo de la Colaboracion.";
28 String txtJS = String.Format("<script>alert('{0}')</script
    >", "Asigne Motivo de la Colaboracion.");
29 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
30 }
31 else{
32 if (txtValorColaboracion.Text == "0.00"){
33 lblMensajes.Visible = true;
34 lblMensajes.Text = "Asigne Valor de la Colaboracion.";
35 String txtJS = String.Format("<script>alert('{0}')</script
    >", "Asigne Valor de la Colaboracion.");
36 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "

```

```

        script", txtJS, false);
37 }
38 else{
39 try{
40 int idusuario = Convert.ToInt32(Id.Text);
41 colaboracion.IdUsuario = idusuario;
42 int idreunion = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
43 colaboracion.IdReunion = idreunion;
44 colaboracion.MotivoColaboracion = txtMotivoColaboracion.
    Text;
45 colaboracion.ValorColaboracion = txtValorColaboracion.Text;
46 colaboracion.MontoPago = "0.00";
47 colaboracion.SaldoTotal = txtValorColaboracion.Text;
48 colaboracion.Estado = "PENDIENTE";
49 try{
50 int Idusuario2 = Convert.ToInt32(Id.Text);
51 int Idreunion2 = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
52 int cant = dal.SelectRepetido(Idusuario2, Idreunion2);
53 if (cant >= 1){
54 lblMensajes.Text = "Este Usuario ya tiene Asignado la
    Colaboracion correspondiente a la Reunion Seleccionada .
    ";
55 lblMensajes.Visible = true;
56 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Este Usuario ya tiene Asignado la Colaboracion
    correspondiente a la Reunion Seleccionada.");
57 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
58 }
59 else{
60 dal.Insert(colaboracion);
61 lblMensajes.Text = "La Colaboracion fue Asignada al Usuario
    ";
62 lblMensajes.Visible = true;
63 B.Visible = false;
64 }
65 }catch{
66 lblMensajes.Text = "Ocurrio un problema al Asignar

```

```

        Colaboracion.VerifiqueConexionalServidor.";
67 lblMensajes.Visible = true;
68 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ocurrio un problema al Asignar Colaboracion.
        VerifiqueConexionalServidor.");
69 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
70 }
71 }catch (MySqlException excepcion){
72 Excepciones.Gestionar(excepcion, Plataforma.Web);
73 lblMensajes.Text = "Error de MySQL Server:" + "<br>" +
        Excepciones.MensajePersonalizado;
74 lblMensajes.Visible = true;
75 }
76 catch (Exception excepcion){
77 Excepciones.Gestionar(excepcion, Plataforma.Web);
78 lblMensajes.Text = "Error de c#:" + "<br>" + Excepciones.
        MensajePersonalizado;
79 lblMensajes.Visible = true;
80 } } } } } } } }

```

- Método Registrar Sanción

```

1  protected void btnConfirmarSancion_Click(object sender,
        EventArgs e){
2  SancionesDAL dal = new SancionesDAL();
3  Sanciones sancioness = new Sanciones();
4  if (txtIdUsuario.Text==""){
5  lblMensajes.Visible = true;
6  lblMensajes.Text = "Seleccione usuario.";
7  String txtJS = String.Format("<script>alert('{0}');</script
        >", "Seleccione Usuario.");
8  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
9  }
10 else{
11 if (cbReunion.SelectedIndex == 0){
12 lblMensajes.Visible = true;

```

```

13 lblMensajes.Text = "Seleccione la reunion.";
14 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Seleccione la reunion.");
15 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
16 }
17 else{
18 CultureInfo culture = new CultureInfo("en-GB");
19 decimal argument = Convert.ToDecimal(txtValorSancion.Text,
    culture);
20 int count = BitConverter.GetBytes(decimal.GetBits(argument)
    [3])[2];
21 if (count > 2){
22 lblMensajes.Text = "Ingrese solo dos decimales.";
23 lblMensajes.Visible = true;
24 txtValorSancion.Focus();
25 }
26 else{
27 if (txtMotivoSancion.Text == ""){
28 lblMensajes.Visible = true;
29 lblMensajes.Text = "Asigne Motivo de la Sancion.";
30 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Asigne Motivo de la Sancion.");
31 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
32 txtMotivoSancion.Focus();
33 }
34 else{
35 if (txtValorSancion.Text == "0.00"){
36 lblMensajes.Visible = true;
37 lblMensajes.Text = "Asigne Valor de la Sancion.";
38 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Asigne Valor de la Sancion.");
39 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
40 txtValorSancion.Focus();
41 }
42 else{
43 try{
44 int idusuario = Convert.ToInt32(txtIdUsuario.Text);

```

```

45 sancioness.IdUsuario = idusuario;
46 int idreunion = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
47 sancioness.IdReunion = idreunion;
48 sancioness.MotivoSancion = txtMotivoSancion.Text;
49 sancioness.ValorSancion = txtValorSancion.Text;
50 sancioness.MontoPago = "0.00";
51 sancioness.SaldoTotal = txtValorSancion.Text;
52 sancioness.Estado = "PENDIENTE";
53 try{
54 int Idusuario2 = Convert.ToInt32(txtIdUsuario.Text);
55 int Idreunion2 = Convert.ToInt32(cbReunion.SelectedItem.
    Value);
56 int cant = dal.SelectRepetido(Idusuario2, Idreunion2);
57 if (cant >= 1){
58 lblMensajes.Text = "Este Usuario ya tiene Asignado la
    Sancion correspondiente a la Reunion Seleccionada.";
59 lblMensajes.Visible = true;
60 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Este Usuario ya tiene Asignado la Sancion
    correspondiente a la Reunion Seleccionada.");
61 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
62 }
63 else{
64 dal.Insert(sancioness);
65 lblMensajes.Text = "La Sancion fue Asignada al Usuario
    correctamente.";
66 lblMensajes.Visible = true;
67 gvDatos.DataSource = dal.SelectSancionesInsert(idusuario);
68 gvDatos.DataBind();
69 gvDatos.Visible = true;
70 btnConfirmarSancion.Visible = false;
71 txtIdUsuario.Text = "";
72 txtMotivoSancion.Text = "";
73 txtNombreUsuario.Text = "";
74 txtValorSancion.Text = "0.00";
75 cbReunion.SelectedIndex = 0;
76 txtMotivoSancion.Visible = false;
77 txtValorSancion.Visible = false;

```

```

78 txtFecha.Visible = false;
79 Session["nombrec"] = "";
80 Session["codigoc"] = "";
81 }
82 }catch{
83 lblMensajes.Text = "Ocurrio un problema al Asignar Sancion.
      Verifique conexion al servidor.";
84 lblMensajes.Visible = true;
85 String txtJS = String.Format("<script>alert('{0}');</script
      >", "Ocurrio un problema al Asignar Sancion. Verifique
      conexion al servidor.");
86 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
      script", txtJS, false);
87 }
88 }catch (MySqlException excepcion){
89 Excepciones.Gestionar(excepcion, Plataforma.Web);
90 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
      Excepciones.MensajePersonalizado;
91 lblMensajes.Visible = true;
92 }
93 catch (Exception excepcion){
94 Excepciones.Gestionar(excepcion, Plataforma.Web);
95 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
      MensajePersonalizado;
96 lblMensajes.Visible = true;
97 } } } } } } }

```

- Método Cobro de Asistencias

```

1 protected void gvCargarDatos_RowCommand(object sender,
      GridViewCommandEventArgs e){
2 CobroDAL dal = new CobroDAL();
3 Cobro cobroo = new Cobro();
4 if (e.CommandName == "Select"){
5 int rowIndex = Convert.ToInt32(e.CommandArgument);
6 GridViewRow row = gvCargarDatos.Rows[rowIndex];
7 Label Id = row.FindControl("ID") as Label;
8 Button B = row.FindControl("bntCobrar") as Button;

```

```

 9  if (txtIDUsuario.Text==""){
10  lblMensajes.Visible = true;
11  lblMensajes.Text = "Seleccione␣Usuario.";
12  String txtJS = String.Format("<script>alert('{0}');</script
    >", "Seleccione␣Usuario.");
13  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
14  }
15  else{
16  if (txtNombre.Text == ""){
17  lblMensajes.Visible = true;
18  lblMensajes.Text = "Debe␣escribir␣el␣nombre␣de␣la␣persona␣
    que␣realizara␣el␣pago.";
19  String txtJS = String.Format("<script>alert('{0}');</script
    >", "Debe␣escribir␣el␣nombre␣de␣la␣persona␣que␣realizara
    ␣el␣pago.");
20  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
21  }
22  else{
23  CultureInfo culture = new CultureInfo("en-GB");
24  decimal argument = Convert.ToDecimal(txtMontoPago.Text,
    culture);
25  int count = BitConverter.GetBytes(decimal.GetBits(argument)
    [3])[2];
26  if (count > 2){
27  lblMensajes.Text = "Ingrese␣solo␣dos␣decimales.";
28  lblMensajes.Visible = true;
29  txtMontoPago.Focus();
30  }
31  else{
32  if (txtMontoPago.Text == "0.00"){
33  lblMensajes.Visible = true;
34  lblMensajes.Text = "Debe␣poner␣el␣monto␣de␣pago";
35  String txtJS = String.Format("<script>alert('{0}');</script
    >", "Debe␣poner␣el␣monto␣de␣pago.");
36  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
37  }
38  else{

```

```

39 try{
40 int idAsistencia = Convert.ToInt32(Id.Text);
41 cobroo.IdAsistencia = idAsistencia;
42 int idUsuario = Convert.ToInt32(txtIDUsuario.Text);
43 cobroo.IdUsuario = idUsuario;
44 int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
45 cobroo.IdDirectivo = idDirectivo;
46 cobroo.NombrePago = txtNombre.Text;
47 cobroo.MontoPago = txtMontoPago.Text;
48 cobroo.FechaPago = txtFecha.Text;
49 try{
50 dal.InsertAsistencia(cobroo);
51 lblMensajes.Text = "El cobro se realizo correctamente.";
52 lblMensajes.Visible = true;
53 B.Visible = false;
54 txtMontoPago.Text = "0.00";
55 txtMontoPago.Focus();
56 int idusuario = Convert.ToInt32(txtIDUsuario.Text);
57 gvCargarDatos.DataSource = dal.SelectAsistencia(idusuario);
58 gvCargarDatos.DataBind();
59 btnConfirmar.Visible = true;
60 }
61 catch{
62 lblMensajes.Text = "Ocurrio un problema al realizar el
        cobro. El valor ha cancelar es mayor al valor de deuda."
        ;
63 lblMensajes.Visible = true;
64 String txtJS = String.Format("<script>alert('{0}');</script
        >", "Ocurrio un problema al realizar el cobro. El valor
        ha cancelar es mayor al valor de deuda.");
65 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
66 }
67 }catch (MySqlException excepcion){
68 Excepciones.Gestionar(excepcion, Plataforma.Web);
69 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
        Excepciones.MensajePersonalizado;
70 lblMensajes.Visible = true;
71 }
72 catch (Exception excepcion){

```

```

73 Excepciones.Gestionar(excepcion, Plataforma.Web);
74 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
    MensajePersonalizado;
75 lblMensajes.Visible = true;
76 } } } } } } } }

```

- Método Cobro Colaboraciones

```

1  protected void gvCargarDatos_RowCommand(object sender,
    GridViewCommandEventArgs e){
2  CobroDAL dal = new CobroDAL();
3  Cobro cobro = new Cobro();
4  if (e.CommandName == "Select"){
5  int rowIndex = Convert.ToInt32(e.CommandArgument);
6  GridViewRow row = gvCargarDatos.Rows[rowIndex];
7  Label Id = row.FindControl("ID") as Label;
8  Label Id2 = row.FindControl("IDUSUARIO") as Label;
9  Button B = row.FindControl("bntCobrar") as Button;
10 if (txtIDReunion.Text == ""){
11 lblMensajes.Visible = true;
12 lblMensajes.Text = "Selecione Reunion.";
13 String txtJS = String.Format("<script>alert('{0}');</script>
    >", "Selecione Reunion.");
14 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
15 }
16 else{
17 if (txtNombre.Text == ""){
18 lblMensajes.Visible = true;
19 lblMensajes.Text = "Debe escribir el nombre de la persona
    que realizara el pago.";
20 String txtJS = String.Format("<script>alert('{0}');</script>
    >", "Debe escribir el nombre de la persona que realizara
    el pago.");
21 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
22 }
23 else{

```

```

24 CultureInfo culture = new CultureInfo("en-GB");
25 decimal argument = Convert.ToDecimal(txtMontoPago.Text,
    culture);
26 int count = BitConverter.GetBytes(decimal.GetBits(argument)
    [3])[2];
27 if (count > 2){
28 lblMensajes.Text = "Ingrese_solo_dos_decimales.";
29 lblMensajes.Visible = true;
30 txtMontoPago.Focus();
31 }
32 else{
33 if (txtMontoPago.Text == "0.00"){
34 lblMensajes.Visible = true;
35 lblMensajes.Text = "Debe_poner_el_monto_de_pago";
36 String txtJS = String.Format("<script>alert('{0}');</script>
    >", "Debe_poner_el_monto_de_pago.");
37 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
38 }
39 else{
40 try{
41 int idColaboracion = Convert.ToInt32(Id.Text);
42 cobroo.IdColaboracion = idColaboracion;
43 int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
44 cobroo.IdDirectivo = idDirectivo;
45 cobroo.NombrePago = txtNombre.Text;
46 cobroo.MontoPago = txtMontoPago.Text;
47 cobroo.FechaPago = txtFecha.Text;
48 try{
49 dal.InsertColaboracion(cobroo);
50 lblMensajes.Text = "El_cobro_se_realizo_correctamente.";
51 lblMensajes.Visible = true;
52 B.Visible = false;
53 txtMontoPago.Text = "0.00";
54 txtNombre.Text = "";
55 txtNombre.Focus();
56 }
57 catch{
58 lblMensajes.Text = "Ocurrio_un_problema_al_realizar_el
    cobro.El_valor_ha_cancelar_es_mayor_al_valor_de_deuda."

```

```

    ;
59 lblMensajes.Visible = true;
60 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ocurrio un problema al realizar el cobro. El valor
    ha cancelar es mayor al valor de deuda.");
61 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
62 }
63 }catch (MySqlException excepcion){
64 Excepciones.Gestionar(excepcion, Plataforma.Web);
65 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
66 lblMensajes.Visible = true;
67 }
68 catch (Exception excepcion){
69 Excepciones.Gestionar(excepcion, Plataforma.Web);
70 lblMensajes.Text = "Error de c#" + "<br>" + Excepciones.
    MensajePersonalizado;
71 lblMensajes.Visible = true;
72 } } } } } } } }

```

- Método Cobro Sanciones

```

1  protected void gvCargarDatos_RowCommand(object sender,
    GridViewCommandEventArgs e){
2  CobroDAL dal = new CobroDAL();
3  Cobro cobro = new Cobro();
4  if (e.CommandName == "Select"){
5  int rowIndex = Convert.ToInt32(e.CommandArgument);
6  GridViewRow row = gvCargarDatos.Rows[rowIndex];
7  Label Id = row.FindControl("ID") as Label;
8  Button B = row.FindControl("bntCobrar") as Button;
9  if (txtIDusuario.Text == ""){
10 lblMensajes.Visible = true;
11 lblMensajes.Text = "Seleccione Usuario.";
12 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Seleccione Usuario.");
13 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);

```

```

14 }
15 else{
16 if (txtNombre.Text == ""){
17 lblMensajes.Visible = true;
18 lblMensajes.Text = "Debe escribir el nombre de la persona
    que realizara el pago.";
19 String txtJS = String.Format("<script>alert('{0}')</script
    >", "Debe escribir el nombre de la persona que realizara
    el pago.");
20 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
21 }
22 else{
23 CultureInfo culture = new CultureInfo("en-GB");
24 decimal argument = Convert.ToDecimal(txtMontoPago.Text,
    culture);
25 int count = BitConverter.GetBytes(decimal.GetBits(argument)
    [3])[2];
26 if (count > 2){
27 lblMensajes.Text = "Ingrese solo dos decimales.";
28 lblMensajes.Visible = true;
29 txtMontoPago.Focus();
30 }
31 else{
32 if (txtMontoPago.Text == "0.00"){
33 lblMensajes.Visible = true;
34 lblMensajes.Text = "Debe poner el monto de pago.";
35 String txtJS = String.Format("<script>alert('{0}')</script
    >", "Debe poner el monto de pago.");
36 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
37 }
38 else{
39 try{
40 int idSancion = Convert.ToInt32(Id.Text);
41 cobroo.IdSancion = idSancion;
42 int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
43 cobroo.IdDirectivo = idDirectivo;
44 cobroo.NombrePago = txtNombre.Text;
45 cobroo.MontoPago = txtMontoPago.Text;

```

```

46 cobroo.FechaPago = txtFecha.Text;
47 try{
48 dal.InsertSancion(cobroo);
49 lblMensajes.Text = "El cobro se realizo correctamente.";
50 lblMensajes.Visible = true;
51 B.Visible = false;
52 txtMontoPago.Text = "0.00";
53 txtMontoPago.Focus();
54 int idusuario = Convert.ToInt32(txtIDusuario.Text);
55 gvCargarDatos.DataSource = dal.SelectSanciones(idusuario);
56 gvCargarDatos.DataBind();
57 btnConfirmar.Visible = true;
58 confirmar();
59 }
60 catch{
61 lblMensajes.Text = "Ocurrio un problema al realizar el
        cobro. El valor ha cancelar es mayor al valor de deuda."
        ;
62 lblMensajes.Visible = true;
63 String txtJS = String.Format("<script>alert('{0}');</script
        >", "Ocurrio un problema al realizar el cobro. El valor
        ha cancelar es mayor al valor de deuda.");
64 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
65 }
66 }catch (MySqlException excepcion){
67 Excepciones.Gestionar(excepcion, Plataforma.Web);
68 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
        Excepciones.MensajePersonalizado;
69 lblMensajes.Visible = true;
70 }
71 catch (Exception excepcion){
72 Excepciones.Gestionar(excepcion, Plataforma.Web);
73 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
        MensajePersonalizado;
74 lblMensajes.Visible = true;
75 } } } } } } } }

```

- Método Descuento de Multas

```
1 protected void btnRegistrarDescuento_Click(object sender ,
    EventArgs e){
2 CobroDAL dal = new CobroDAL();
3 Cobro cobro = new Cobro();
4 CultureInfo culture = new CultureInfo("en-GB");
5 if (txtIDUsuario.Text==""){
6 lblMensajes.Text = "Seleccione Usuario.";
7 lblMensajes.Visible = true;
8 String txtJS = String.Format("<script>alert('{0}');</script>
    >", "Seleccione Usuario.");
9 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
10 }
11 else{
12 if (txtValorAcumulado.Text=="0" || txtValorAcumulado.Text=="
    0.00"){
13 lblMensajes.Text = "El usuario no tiene deudas.";
14 lblMensajes.Visible = true;
15 String txtJS = String.Format("<script>alert('{0}');</script>
    >", "El usuario no tiene deudas.");
16 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
17 }
18 else{
19 if (txtIdReunion.Text==""){
20 lblMensajes.Text = "Seleccionar Reunion.";
21 lblMensajes.Visible = true;
22 String txtJS = String.Format("<script>alert('{0}');</script>
    >", "Seleccionar Reunion.");
23 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
24 }
25 else{
26 if (txtValorDescuento.Text=="0.00"){
27 lblMensajes.Text = "Ingrese valor del descuento.";
28 lblMensajes.Visible = true;
29 txtValorDescuento.Focus();
30 String txtJS = String.Format("<script>alert('{0}');</script>
```

```

        >", "Ingrese_valor_del_descuento.");
31 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
32 }
33 else{
34 decimal argument = Convert.ToDecimal(txtValorDescuento.Text
    , culture);
35 int count = BitConverter.GetBytes(decimal.GetBits(argument)
    [3])[2];
36 if (count > 2){
37 lblMensajes.Text = "Ingrese_solo_dos_decimales.";
38 lblMensajes.Visible = true;
39 txtValorDescuento.Focus();
40 }
41 else{
42 decimal numero1 = Convert.ToDecimal(txtValorAcumulado.Text ,
    culture);
43 decimal numero2 = Convert.ToDecimal(txtValorDescuento.Text ,
    culture);
44 if (numero2 > numero1){
45 lblMensajes.Text = "El_valor_descuento_es_mayor_al_valor
    total_deuda.";
46 lblMensajes.Visible = true;
47 txtValorDescuento.Focus();
48 String txtJS = String.Format("<script>alert('{0}');</script
    >", "El_valor_descuento_es_mayor_al_valor_total_deuda.");
    ;
49 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
50 }
51 else{
52 if (txtMotivoDescuento.Text == ""){
53 lblMensajes.Text = "Ingrese_el_motivo_del_descuento.";
54 lblMensajes.Visible = true;
55 txtMotivoDescuento.Focus();
56 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese_el_motivo_del_descuento.");
57 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
58 }

```

```

59 else{
60 if (txtFecha.Text == ""){
61 lblMensajes.Text = "Seleccione□Fecha.";
62 lblMensajes.Visible = true;
63 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Seleccione□Fecha.");
64 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
65 }
66 else{
67 try{
68 int idUsuario2 = Convert.ToInt32(txtIDUsuario.Text);
69 cobroo.IdUsuario = idUsuario2;
70 int idDirectivo2 = Convert.ToInt32(cbID.SelectedItem.Value)
    ;
71 cobroo.IdDirectivo = idDirectivo2;
72 int idReunion2 = Convert.ToInt32(txtIdReunion.Text);
73 cobroo.IdReunion = idReunion2;
74 cobroo.MotivoDescuento = txtMotivoDescuento.Text;
75 cobroo.ValorDescuento = txtValorDescuento.Text;
76 cobroo.ValorDeudaTotal = txtValorAcumulado.Text;
77 cobroo.FechaDescuento = txtFecha.Text;
78 try{
79 dal.InsertDescuentos(cobroo);
80 }
81 catch{
82 lblMensajes.Text = "Ocurrio□un□problema□al□realizar□el□
    Descuento.□Contacte□al□administrador.";
83 lblMensajes.Visible = true;
84 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ocurrio□un□problema□al□realizar□el□cobro.□Contacte□
    al□administrador.");
85 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
86 }
87 }catch (MySqlException excepcion){
88 Excepciones.Gestionar(excepcion, Plataforma.Web);
89 lblMensajes.Text = "Error□de□MYSQL□Server:□" + "<br>" +
    Excepciones.MensajePersonalizado;
90 lblMensajes.Visible = true;

```

```

91 }
92 catch (Exception excepcion){
93     Excepciones.Gestionar(excepcion, Plataforma.Web);
94     lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
        MensajePersonalizado;
95     lblMensajes.Visible = true;
96     int Idusuario = Convert.ToInt32(txtIDUsuario.Text);
97     decimal descuento = Convert.ToDecimal(txtValorDescuento.
        Text, culture);
98     decimal descuentoaux = 0;
99     foreach (DataRow fila in dal.SelectAcumuladoPendiente(
        Idusuario).Rows){
100     if (descuento != 0){
101     if (descuento >= Convert.ToDecimal(fila [2], culture)){
102     descuento = descuento - Convert.ToDecimal(fila [2], culture;
103     try{
104     int idAsistencia = Convert.ToInt32(fila [0]);
105     cobroo.IdAsistencia = idAsistencia;
106     int idUsuario = Convert.ToInt32(txtIDUsuario.Text);
107     cobroo.IdUsuario = idUsuario;
108     int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
109     cobroo.IdDirectivo = idDirectivo;
110     cobroo.NombrePago = "DESCUENTO";
111     string pago = Convert.ToString(fila [2], culture);
112     cobroo.MontoPago = pago;
113     cobroo.FechaPago = txtFecha.Text;
114     try{
115     dal.InsertAsistencia(cobroo);
116     lblMensajes.Text = "El Descuento se realizo correctamente."
        ;
117     lblMensajes.Visible = true;
118     gvCargarDatos.DataSource = dal.SelectAcumuladoPendiente(
        Idusuario);
119     gvCargarDatos.DataBind();
120     btnBuscarReunion.Visible = false;
121     btnBuscarUsuario.Visible = false;
122     btnRegistrarDescuento.Visible = false;
123     btnCalcularDeuda.Visible = false;
124     txtValorDescuento.Enabled = false;
125     txtFecha.Enabled = false;

```

```

126 txtMotivoDescuento.Enabled = false;
127 }
128 catch{
129 lblMensajes.Text = "Ocurrio un problema al realizar el
        Descuento. Contacte al administrador.";
130 lblMensajes.Visible = true;
131 String txtJS = String.Format("<script>alert('{0}');</script
        >", "Ocurrio un problema al realizar el cobro. Contacte
        al administrador.");
132 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
        script", txtJS, false);
133 }
134 }catch (MySqlException excepcion){
135 Excepciones.Gestionar(excepcion, Plataforma.Web);
136 lblMensajes.Text = "Error de MySQL Server:" + "<br>" +
        Excepciones.MensajePersonalizado;
137 lblMensajes.Visible = true;
138 }
139 catch (Exception excepcion){
140 Excepciones.Gestionar(excepcion, Plataforma.Web);
141 lblMensajes.Text = "Error de c#:" + "<br>" + Excepciones.
        MensajePersonalizado;
142 lblMensajes.Visible = true;
143 }
144 }else{
145 descuentoaux = descuento;
146 descuento = 0;
147 try{
148 int idAsistencia = Convert.ToInt32(fila[0]);
149 cobroo.IdAsistencia = idAsistencia;
150 int idUsuario = Convert.ToInt32(txtIDUsuario.Text);
151 cobroo.IdUsuario = idUsuario;
152 int idDirectivo = Convert.ToInt32(cbID.SelectedItem.Value);
153 cobroo.IdDirectivo = idDirectivo;
154 cobroo.NombrePago = "DESCUENTO";
155 string pago = Convert.ToString(descuentoaux, culture);
156 cobroo.MontoPago = pago;
157 cobroo.FechaPago = txtFecha.Text;
158 try{
159 dal.InsertAsistencia(cobroo);

```



```

1  protected void btnGenerarInforme_Click(object sender ,
      EventArgs e){
2  CobroDAL dal = new CobroDAL();
3  if (txtNombreUsuario.Text == ""){
4  lblMensajes.Text = "▯Seleccione▯Usuario.";
5  lblMensajes.Visible = true;
6  String txtJS = String.Format("<script>alert('{0}')</script
      >", "Seleccione▯Usuario.");
7  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
      script", txtJS, false);
8  }
9  else{
10 int IDUsuario = Convert.ToInt32(txtIDusuario.Text);
11 int cant = dal.
      SelectconteoInformeCobroColaboracionesUsuario(IDUsuario)
      ;
12 if (cant == 0){
13 lblMensajes.Text = "El▯usuario▯seleccionado▯no▯tiene▯ningun
      ▯cobro.";
14 lblMensajes.Visible = true;
15 String txtJS = String.Format("<script>alert('{0}')</script
      >", "El▯usuario▯seleccionado▯no▯tiene▯ningun▯cobro.");
16 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
      script", txtJS, false);
17 btnBuscarUsuario.Focus();
18 btnGenerarInforme.Visible = false;
19 }
20 else{
21 int UsuarioID = Convert.ToInt32(txtIDusuario.Text);
22 gvDatos.DataSource = dal.SelectCobroColaboracionesUsuario(
      UsuarioID);
23 gvDatos.DataBind();
24 gvDatos.Visible = true;
25 btnGenerarInforme.Visible = false;
26 } } }

```

4.3.8.8. Pruebas de Aceptación

A continuación, se muestra las pruebas de aceptación de la Cuarta Iteración:

CASO DE PRUEBA	
Código: 7	Nº Historia de Usuario: 7
Historia de Usuario: Tesorería.	
Condiciones de Ejecución: Solo el cargo de Tesorero(a) podrá realizar los procesos de tesorería.	
Entrada/Pasos de Ejecución:	
Registrar Colaboraciones	
<p>Seleccionar la pestaña Tesorería, la opción Registrar Colaboraciones.</p> <p>Seleccionar la Reunión que debe estar Activa generada anteriormente por el presidente, dar clic en el botón Listar Usuarios, ingresar el motivo de la Colaboración, el valor de colaboración y la fecha para luego dar clic en el botón confirmar Colaboración, inmediatamente aparecerá un listado donde se debe dar clic en el botón confirmar, para registrar la colaboración a los usuarios.</p> <p>El botón Nuevo limpia los campos de reunión, motivo de colaboración, el valor de colaboración, la fecha y el listado de usuarios.</p>	
Registrar sanción	
<p>Seleccionar la pestaña Tesorería, la opción Registrar sanción.</p> <p>Seleccionar la Reunión que debe estar Activa generada anteriormente por el presidente y el usuario que se le aplicara la sanción, ingresar el motivo de la sanción, el valor de sanción y la fecha, luego dar clic en el botón confirmar sanción para registrar la información.</p> <p>El botón Nuevo limpia los campos de motivo de sanción, el valor de sanción, la fecha, la reunión y el usuario.</p>	
Generar Descuentos en las Multas	
<p>Seleccionar la pestaña Tesorería, la opción Generar Descuento.</p> <p>Para generar un descuento se debe buscar el usuario y la reunión en donde se aprueba el descuento por los usuarios, después dar clic en el botón Calcular Multa Total para verificar el valor total de pago y descontar lo que se aprobó en reunión, si el usuario no tiene ninguna deuda no se generara el descuento, se debe también tener en cuenta si el valor a descontar es mayor al valor de deuda total no se generara el descuento, después de verificar que los campos están bien ingresados se debe dar clic en el botón Generar Descuento para registrar y realizar el descuento de la multa.</p> <p>El botón Nuevo Registro limpia los campos de Usuario, Reunión, Valor Descuento, Motivo del Descuento y la fecha.</p>	
Resultado Esperado: La información de la Tesorería será guardada en la base de datos. El módulo de Tesorería facilita el control de cobros, registros de colaboraciones y sanciones, este módulo también genera informes. Los informes de los cobros y registros, serán de gran ayuda para verificar si el usuario ha cancelado todas sus deudas y colaboraciones, con estos informes el usuario podrá verificar si tienen algún problema en sus cobros y en sus registros. Finalizo la prueba del módulo de Tesorería sin inconsistencias.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.103: Caso de Prueba: Tesorería-Registros.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 8	Nº Historia de Usuario: 7
Historia de Usuario: Tesorería.	
Condiciones de Ejecución: Solo el cargo de Tesorero(a) podrá realizar los procesos de tesorería.	
Entrada/Pasos de Ejecución:	
<p>Cobro de Multas de Asistencia Por Usuario Seleccionar la pestaña Tesorería, la opción Cobro Asistencia por Usuario. Dar clic en el botón Buscar Usuario y Seleccionar al usuario, luego dar clic en el botón Buscar Multa para que inmediatamente aparezca un listado de todas las multas del usuario, luego dar clic en el botón Confirmar Cobro para ingresar el nombre de quien cancela, el monto de pago y la fecha, después de ingresar la información se da clic en el botón Cobrar para registrar el cobro. El botón Nuevo Cobro limpia los campos de usuario, nombre de quien cancela, monto pago y la fecha.</p> <p>Cobro de Multas de Asistencia Por Reunión Seleccionar la pestaña Tesorería, la opción Cobro Asistencia por Reunión. Dar clic en el botón Buscar Reunión y Seleccionar la reunión, luego dar clic en el botón Buscar Multa para que inmediatamente aparezca un listado de todas las multas que se encuentran registradas en la reunión seleccionada, luego dar clic en el botón Confirmar Cobro para ingresar el nombre de quien cancela, el monto de pago y la fecha, después de ingresar la información se da clic en el botón Cobrar para registrar el cobro. El botón Nuevo Cobro limpia los campos de reunión, nombre de quien cancela, monto pago y la fecha.</p> <p>Cobro de Colaboraciones Por Usuario Seleccionar la pestaña Tesorería, la opción Cobro Colaboraciones por Usuario. Dar clic en el botón Buscar Usuario y Seleccionar al usuario, luego dar clic en el botón Buscar Colaboración para que inmediatamente aparezca un listado de todas las colaboraciones registradas al usuario, luego dar clic en el botón Confirmar Cobro para ingresar el nombre de quien cancela, el monto de pago y la fecha, después de ingresar la información se da clic en el botón Cobrar para registrar el cobro. El botón Nuevo Cobro limpia los campos de usuario, nombre de quien cancela, monto pago y la fecha.</p> <p>Cobro de Colaboraciones Por Reunión Seleccionar la pestaña Tesorería, la opción Cobro Colaboraciones por Reunión. Dar clic en el botón Buscar Reunión y Seleccionar la reunión, luego dar clic en el botón Buscar Colaboración para que inmediatamente aparezca un listado de todas las colaboraciones que se encuentran registradas en la reunión seleccionada, luego dar clic en el botón Confirmar Cobro para ingresar el nombre de quien cancela, el monto de pago y la fecha, después de ingresar la información se da clic en el botón Cobrar para registrar el cobro. El botón Nuevo Cobro limpia los campos de reunión, nombre de quien cancela, monto pago y la fecha.</p> <p>Cobro de Sanciones Por Usuario Seleccionar la pestaña Tesorería, la opción Cobro de Sanciones Por Usuario. Dar clic en el botón Buscar Usuario y Seleccionar al usuario, luego dar clic en el botón Buscar Sanción para que inmediatamente aparezca un listado de todas las sanciones registradas al usuario, luego dar clic en el botón Confirmar Cobro para ingresar el nombre de quien cancela, el monto de pago y la fecha, después de ingresar la información se da clic en el botón Cobrar para registrar el cobro. El botón Nuevo Cobro limpia los campos de usuario, nombre de quien cancela, monto pago y la fecha.</p>	
<p>Resultado Esperado: La información de la Tesorería será guardada en la base de datos. El módulo de Tesorería facilita el control de cobros, registros de colaboraciones y sanciones, este módulo también genera informes. Los informes de los cobros y registros, serán de gran ayuda para verificar si el usuario ha cancelado todas sus deudas y colaboraciones, con estos informes el usuario podrá verificar si tienen algún problema en sus cobros y en sus registros. Finalizo la prueba del módulo de Tesorería sin inconsistencias.</p>	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.104: Caso de Prueba: Tesorería-Cobros.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 9	Nº Historia de Usuario: 7
Historia de Usuario: Tesorería.	
Condiciones de Ejecución: Solo el cargo de Tesorero(a) podrá realizar los procesos de tesorería.	
Entrada/Pasos de Ejecución:	
Informes de Tesorería	
1) Seleccionar la pestaña Asistencia, la opción Informe General. Seleccionamos la reunión y damos clic en el botón Generar Informe. Este proceso mostrara la información del control de asistencia que se realizó en la reunión seleccionada.	
2) Seleccionar la pestaña Asistencia, la opción Informe de Asistencia Por Usuario. Seleccionamos el usuario y damos clic en el botón Generar Informe. Este proceso mostrara la información de todas las asistencias que le corresponden al usuario seleccionado.	
3) Seleccionar la pestaña Asistencia, la opción Informe Por Estado. Seleccionamos el estado Atraso o Ausencia y damos clic en el botón Generar Informe. Este proceso mostrara la información de todos los usuarios Atrasados, Ausentes o Presentes.	
4) Seleccionar la pestaña Informes Tesorería, la opción Informe Colaboraciones Por Usuario. Seleccionamos el usuario y damos clic en el botón Generar Informe. Este proceso mostrara la información de todas las Colaboraciones que le corresponden al usuario seleccionado.	
5) Seleccionar la pestaña Informes Tesorería, la opción Informe Colaboraciones Por Reunión. Seleccionamos la reunión y damos clic en el botón Generar Informe. Este proceso mostrara la información de las Colaboraciones que corresponden a la reunión seleccionada.	
6) Seleccionar la pestaña Informes Tesorería, la opción Informe General De Sanciones. Este informe mostrara inmediatamente la información de las Sanciones que se han registrado.	
7) Seleccionar la pestaña Informes Tesorería, la opción Informe General De Descuentos. Este informe mostrara inmediatamente la información de los descuentos que se han registrado.	
8) Seleccionar la pestaña Tesorería, la opción Informe Cobro Asistencia Por Usuario. Seleccionamos el usuario y damos clic en el botón Generar Informe. Este proceso mostrara la información de todos los cobros de asistencias que le corresponden al usuario seleccionado.	
9) Seleccionar la pestaña Tesorería, la opción Informe Cobro Asistencia Por Reunión. Seleccionamos la reunión y damos clic en el botón Generar Informe. Este proceso mostrara la información de los cobros de asistencia que se realizaron en la reunión seleccionada.	
10) Seleccionar la pestaña Tesorería, la opción Informe Cobro Colaboraciones Por Usuario. Seleccionamos el usuario y damos clic en el botón Generar Informe. Este proceso mostrara la información de todos los cobros de colaboraciones que le corresponden al usuario seleccionado.	
11) Seleccionar la pestaña Tesorería, la opción Informe Cobro Colaboraciones Por Reunión. Seleccionamos la reunión y damos clic en el botón Generar Informe. Este proceso mostrara la información de los cobros de colaboraciones que se realizaron en la reunión seleccionada.	
12) Seleccionar la pestaña Informes Tesorería, la opción Informe Cobro De Sanciones. Este informe mostrara inmediatamente la información de los cobros de sanciones que se han registrado.	
Resultado Esperado: La información de la Tesorería será guardada en la base de datos. El módulo de Tesorería facilita el control de cobros, registros de colaboraciones y sanciones, este módulo también genera informes. Los informes de los cobros y registros, serán de gran ayuda para verificar si el usuario ha cancelado todas sus deudas y colaboraciones, con estos informes el usuario podrá verificar si tienen algún problema en sus cobros y en sus registros.	
Finalizo la prueba del módulo de Tesorería sin inconsistencias.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.105: Caso de Prueba: Tesorería-Informes.

Elaborado por: Oscar Curichumbi.

4.3.8.9. Resultados de la Cuarta Iteración

Como resultado de entrega de la cuarta iteración que consta del módulo de Tesorería, el cliente quedó satisfecho con las funcionalidades del módulo que se desarrolló, la cuarta iteración se ha concluido sin ningún problema manifestado por el cliente. A continuación, se realizará la Quinta Iteración.

4.3.9. Desarrollo de la Quinta Iteración

Se realizará la descripción general de las tareas respectivas de la octava, novena y décima historias de usuario, que pertenecen a la quinta iteración. Como se indica en la planificación se desarrollarán los módulos de Control de Entrada y Salida de documentos, Egresos y otros Ingresos y el módulo de Reporte general de Ingresos y Egresos. En la Tabla 4.105 se muestra de manera general las tareas de ingeniería correspondientes a las historias de usuario número 8-9-10 para cumplir con lo planificado.

Número de Tarea	Número de Historia	Esfuerzo	Nombre de la Tarea
45	8	2	Diseño de base de datos para el Control de Entrada y Salida de documentos.
46	8	2	Diseño de la interfaz para el Control de Entrada y Salida de documentos.
47	8	3	Validación de datos para registrar correctamente el Control de Entrada y Salida de documentos.
48	8	4	Subir y Bajar Actas, Documentos Entregados y Documentos recibidos del servidor.
49	8	3	Guardar la información del Control de Entrada y Salida de documentos en la base de datos.
50	9	1	Diseño de base de datos para el control de Egresos y otros Ingresos.
51	9	1	Diseño de la interfaz para el control de Egresos y otros Ingresos.
52	9	1	Validación de datos para registrar correctamente los Egresos y otros Ingresos.
53	9	3	Guardar la información de Egresos y otros Ingresos en la base de datos.
54	9	1	Informe de Egresos con factura y sin factura. Informe de otros ingresos.
55	10	2	Diseño de la interfaz para el Reporte General De Ingresos y Egresos.
56	10	8	Generar reporte con parámetros de fechas para verificar ingresos y egresos.
57	10	1	Diseño de la interfaz para el Reporte General De Valores Faltantes Por Cobrar.
58	10	3	Generar reporte para verificar Valores Faltantes Por Cobrar.

Tabla 4.106: Tareas a Realizar de las Historias de Usuario 8-9-10.

Elaborado por: Oscar Curichumbi.

4.3.9.1. Estimación de la Quinta Iteración

A continuación, se muestra la estimación de la Quinta Iteración:

ITERACIÓN 5				
N.- Historia	Fecha Inicio	Fecha Final	DÍAS	HORAS
8	05/08/2019	18/08/2019	14	56
9	19/08/2019	25/08/2019	7	28
10	26/08/2019	08/09/2019	14	56
TOTAL ESTIMACIÓN:			35	140

Tabla 4.107: Horas asignadas para el desarrollo de la Iteración 5.

Elaborado por: Oscar Curichumbi.

4.3.9.2. Bocetos de las Historias de Usuarios 8-9-10

En las siguientes figuras se muestran los bocetos de las Historias de Usuarios 8-9-10 que forman parte de la Quinta Iteración.

Documentos que Ingresan a la Comunidad

Tipo de Documento: | | Registrar Documento

Titulo del documento: | Nuevo

Fecha Registro: 10-08-2019 | Cancelar

Documento: Seleccionar Archivo.....

Observaciones:

Tipo Documento	Titulo Documento	Fecha	Ingresado por	Documento	Observacion

Figura 4.96: Boceto de Documentos que Ingresan a la comunidad.

Elaborado por: Oscar Curichumbi.

Documento que Sale de la Comunidad

Tipo de Documento:	<input type="text"/>	<input type="text"/>
Título del Documento:	<input type="text"/>	<input type="text"/>
Fecha Registro:	10-08-2019	<input type="text"/>
Documento:	Seleccionar Archivo	<input type="text"/>
Observaciones:	<input type="text"/>	<input type="text"/>

Tipo Documento	Título Documento	Fecha	Ingresado Por	Documento	observación

Figura 4.97: Boceto de Documentos que Salen de la comunidad.
Elaborado por: Oscar Curichumbi.

Descargar Actas

Motivo Reunión	Fecha	Nº Acta	Descargar
Reunión sobre elección Directiva	20/09/19	1	Acta1.docx
Reunión sobre campeonato	29/10/19	2	Acta2.docx

Figura 4.98: Boceto de Descarga de Actas.
Elaborado por: Oscar Curichumbi.

Descargar Documentos Recibidos

Tipo Documento	Título Documento	Fecha	Descargar
Oficio	Documento Consejo Gobierno	04/09/2019	Oficio1.pdf
certificado	Oficio Municipio	10/09/2019	Oficio2.pdf

Figura 4.99: Boceto de Descarga de Documentos Recibidos.
Elaborado por: Oscar Curichumbi.

Descargar Documentos Entregados

Tipo Documento	Titulo Documento	Fecha	Descargar
oficio	Documento municipio	5/09/2019	oficio3.pdf
oficio	Documento Junta parroquial	7/10/2019	oficio4.pdf

Figura 4.100: Boceto de Descarga de Documentos Entregados.
Elaborado por: Oscar Curichumbi.

Egreso con factura

concepto del Egreso:	<input type="text"/>	<input type="button" value="Ingresar Egreso"/>
valor del Egreso:	0.00	<input type="button" value="Nuevo"/>
Fecha Egreso:	10-11-2019	<input type="button" value="Cancelar"/>
Comprado por:	<input type="text"/>	
Número de factura:	<input type="text" value="000"/> - <input type="text" value="000"/> - <input type="text"/>	
Nombre Local:	<input type="text"/>	
Fecha compra:	<input type="text"/>	
Dirección Local:	<input type="text"/>	

concepto del egreso	valor	Fecha	comprado por	Número factura	Local

Figura 4.101: Boceto de Egresos Con Factura.
Elaborado por: Oscar Curichumbi.

Egreso sin Factura

Concepto Egreso:	<input type="text"/>	Ingresar Egreso
Valor Egreso:	0.00	Nuevo
Fecha Egreso:	10-11-2019	Cancelar
Comprado por:	<input type="text"/>	
Dirección compra:	<input type="text"/>	

concepto Egreso	valor Egreso	Fecha	comprado por	Dirección

Figura 4.102: Boceto de Egresos Sin Factura.
Elaborado por: Oscar Curichumbi.

Otros Ingresos

Concepto Ingreso:	<input type="text"/>	Ingresar Ingreso
Valor Ingreso:	0.00	Nuevo
Fecha Ingreso:	10-11-2019	Cancelar

Concepto Ingreso	Valor Ingreso	Fecha

Figura 4.103: Boceto de Otros Ingresos.
Elaborado por: Oscar Curichumbi.

Informe General de Egresos con Factura

concepto Egreso	valor	Fecha	comprado por	Nº factura	Local	Dirección

Figura 4.104: Boceto de Informe General de Egresos Con Factura.
Elaborado por: Oscar Curichumbi.

Informe General de Egresos sin Factura

concepto Egreso	valor	Fecha	comprado Por	Dirección	Ingresado Por

Figura 4.105: Boceto de Informe General de Egresos Sin Factura.
Elaborado por: Oscar Curichumbi.

Informe General de otros Ingresos

concepto Ingreso	Valor Ingreso	Fecha	Ingresado Por

Figura 4.106: Boceto de Informe General de Otros Ingresos.
Elaborado por: Oscar Curichumbi.

Reporte General de Ingresos y Egresos

Fecha Inicio:	20-11-2018	Generar Reporte
Fecha Fin:	20-11-2019	Nueva Busqueda

Ingresos

Multas:	20,00
Colaboraciones:	20,00
Sanciones:	20,00
Otros Ingreso:	100,00
Total Ingresos:	160,00

Egresos

Egresos sin Factura:	60,00
Egresos con Factura:	10,00
Descuentos:	90,00
Total Egresos:	160,00

Total Ingresos-Egresos: 0,00

Figura 4.107: Boceto de Reporte General de Ingresos y Egresos.
Elaborado por: Oscar Curichumbi.

Reporte General de Valores Faltantes Por Cobrar

Valores	
Multas:	20,00
Colaboraciones:	20,00
Sanciones:	20,00
Total Deuda Por Cobrar:	60,00

Figura 4.108: Boceto de Reporte General de Valores Faltantes Por Cobrar.
Elaborado por: Oscar Curichumbi.

4.3.9.3. Descripción de las Tareas de Ingenierías

En base a la planificación se ha realizado las tareas de ingeniería, en estas se ha asignado el tiempo de estimación para el desarrollo de cada una de ellas. Si el

cliente está satisfecho con el desarrollo de las tareas, que forman parte de las tres historias de usuario, esto cumplirá con las necesidades del cliente y de este modo se culminará la quinta iteración. A continuación, se muestra el desarrollo de las tareas de cada historia de usuario.

Tarea de Ingeniería	
Número de Tarea: 45	Número de Historia: 8
Nombre de Tarea: Diseño de base de datos para el Control de Entrada y Salida de documentos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 05/08/2019	Fecha Fin:06/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se creará las tablas para el control de Entrada y Salida de documentos en la base de datos, para registrar la información de los procesos que se generan.	

Tabla 4.108: Tarea 45 de la Historia de Usuario N.-8.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Número de Tarea: 46	Número de Historia: 8
Nombre de Tarea: Diseño de la interfaz para el Control de Entrada y Salida de documentos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 07/08/2019	Fecha Fin:08/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el secretario(a) o los directivos asignados podrán ingresar la información del control de Entrada y Salida de documentos de la comunidad.	

Tabla 4.109: Tarea 46 de la Historia de Usuario N.-8.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Número de Tarea: 47	Número de Historia: 8
Nombre de Tarea: Validación de datos para registrar correctamente el Control de Entrada y Salida de documentos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 09/08/2019	Fecha Fin: 11/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere validar los datos de Entrada y Salida de documentos antes de ingresar a la base de datos, para no ingresar datos nulos o datos erróneos.	

Tabla 4.110: Tarea 47 de la Historia de Usuario N.-8.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Número de Tarea: 48	Número de Historia: 8
Nombre de Tarea: Subir y Bajar Actas, Documentos Entregados y Documentos recibidos del servidor.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 12/08/2019	Fecha Fin: 15/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Los documentos recibidos y entregados deben guardarse en una carpeta dentro del servidor, los documentos deben ser subidos al servidor y deben generar un link del documento para ingresar a la base de datos y así descargarlos cuando lo desee los directivos de la comunidad.	

Tabla 4.111: Tarea 48 de la Historia de Usuario N.-8.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Número de Tarea: 49	Número de Historia: 8
Nombre de Tarea: Guardar la información del Control de Entrada y Salida de documentos en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 16/08/2019	Fecha Fin: 18/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información del Control de Entrada y Salida de documentos, para registrarlos en la base de datos.	

Tabla 4.112: Tarea 49 de la Historia de Usuario N.-8.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Número de Tarea: 50	Número de Historia: 9
Nombre de Tarea: Diseño de base de datos para el control de Egresos y otros Ingresos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 19/08/2019	Fecha Fin: 19/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se creará las tablas para el control de Egresos y otros Ingresos en la base de datos, para registrar la información de los procesos que se generan.	

Tabla 4.113: Tarea 50 de la Historia de Usuario N.-9.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Número de Tarea: 51	Número de Historia: 9
Nombre de Tarea: Diseño de la interfaz para el control de Egresos y otros Ingresos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/08/2019	Fecha Fin: 20/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el Tesorero(a) o los directivos asignados podrán ingresar la información del control de Egresos y otros Ingresos de la comunidad.	

Tabla 4.114: Tarea 51 de la Historia de Usuario N.-9.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 52	Número de Historia: 9
Nombre de Tarea: Validación de datos para registrar correctamente los Egresos y otros Ingresos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 21/08/2019	Fecha Fin: 21/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se requiere validar los datos de Egresos y otros Ingresos antes de ingresar a la base de datos, para no ingresar datos nulos o datos erróneos.	

Tabla 4.115: Tarea 52 de la Historia de Usuario N.-9.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 53	Número de Historia: 9
Nombre de Tarea: Guardar la información de Egresos y otros Ingresos en la base de datos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 22/08/2019	Fecha Fin: 24/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se guardará la información del Control de Egresos y otros Ingresos, para registrarlos en la base de datos.	

Tabla 4.116: Tarea 53 de la Historia de Usuario N.-9.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 54	Numero de Historia: 9
Nombre de Tarea: Informe de Egresos con factura y sin factura. Informe de otros ingresos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 25/08/2019	Fecha Fin: 25/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se realizará la visualización de los informes de los registros de Egresos con Factura y Egresos sin Factura, también se realizará informes que permitan visualizar otros ingresos.	

Tabla 4.117: Tarea 54 de la Historia de Usuario N.-9.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 55	Numero de Historia: 10
Nombre de Tarea: Diseño de la interfaz para el Reporte General De Ingresos y Egresos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 26/08/2019	Fecha Fin: 27/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el Tesorero(a) visualizar la información general de ingresos y egresos que la comunidad ha registrado.	

Tabla 4.118: Tarea 55 de la Historia de Usuario N.-10.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 56	Numero de Historia: 10
Nombre de Tarea: Generar reporte con parámetros de fechas para verificar ingresos y egresos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 8
Fecha Inicio: 28/08/2019	Fecha Fin: 04/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: El reporte se generará mediante la selección de la fecha inicio y la fecha fin, este reporte visualizara los ingresos totales y los egresos totales que la comunidad registro, para verificar si la comunidad tiene saldo positivo o negativo. Este reporte ayudara al Tesorero(a) a informa a los usuarios de la comunidad sobre los ingresos y los egresos que se han registrado durante el tiempo de los directivos actuales.	

Tabla 4.119: Tarea 56 de la Historia de Usuario N.-10.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 57	Numero de Historia: 10
Nombre de Tarea: Diseño de la interfaz para el Reporte General De Valores Faltantes Por Cobrar.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 05/08/2019	Fecha Fin: 05/08/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Se Diseñará la interfaz, donde el Tesorero(a) visualizará el total de los valores faltantes por cobrar.	

Tabla 4.120: Tarea 57 de la Historia de Usuario N.-10.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 58	Numero de Historia: 10
Nombre de Tarea: Generar reporte para verificar Valores Faltantes Por Cobrar.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 06/08/2019	Fecha Fin: 08/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: El reporte se generará sin parámetros, este reporte visualizará los valores faltantes por cobrar, para que el Tesorero(a) informe a todos los usuarios de la comunidad sobre el total de deuda que aún deben cancelar los usuarios deudores.	

Tabla 4.121: Tarea 58 de la Historia de Usuario N.-10.

Elaborado por: Oscar Curichumbi.

4.3.9.4. Tarjetas CRC: Historias de usuario 8-9-10

En las siguientes tablas se muestran las tarjetas CRC, estas tarjetas son de gran utilidad para la Quinta Iteración.

Tipodocumento	
Responsabilidad	Colaboración
Atributos:	
Id_TipoDocumento	
NombreDocumento	
DescripcionDocumento	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.122: Tarjeta CRC: Tipo Documento.

Elaborado por: Oscar Curichumbi.

Documentos	
Responsabilidad	Colaboración
Atributos:	Tipodocumento
Id_Documento	Directivos
Id_TipoDocumento	
TituloDocumento	
FechaRegistro	
linkDocumento	
Documento	
Id_Directivos	
Observaciones	
ObservacionesCambio	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.123: Tarjeta CRC: Documentos.

Elaborado por: Oscar Curichumbi.

Egresos	
Responsabilidad	Colaboración
Atributos:	Directivos
Id_Egreso	
Id_Directivos	
ConceptoEgreso	
ValorEgreso	
FechaEgreso	
PersonaCompra	
DireccionCompra	
TipoEgreso	
NumeroFactura	
NombreLocalCompra	
FechaCompra	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.124: Tarjeta CRC: Egresos.

Elaborado por: Oscar Curichumbi.

Ingresos	
Responsabilidad	Colaboración
Atributos:	Directivos
Id_Ingreso	
Id_Directivos	
ConceptoIngreso	
ValorIngreso	
FechaIngreso	
Métodos	
Agregar	
Editar	
Buscar	

Tabla 4.125: Tarjeta CRC: otros Ingresos.

Elaborado por: Oscar Curichumbi.

4.3.9.5. Diagrama de datos

A continuación, se muestra el diagrama de datos de la Quinta Iteración:

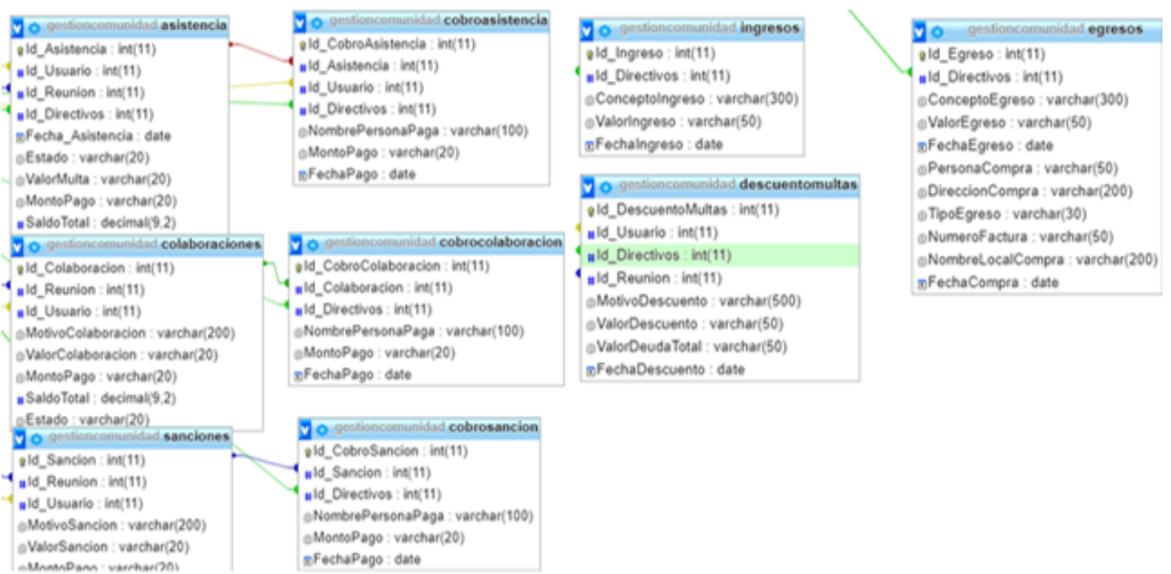


Figura 4.109: Diagrama de datos: Documentos, Ingresos-Egresos y Reportes.

Elaborado por: Oscar Curichumbi.

4.3.9.6. Diseño de Interfaz

A continuación, se muestra los diseños de interfaz de la Quinta Iteración:

DOCUMENTOS QUE INGRESAN A LA COMUNIDAD

Tipo De Documento:

Titulo Del Documento:

Fecha Registro:

Documento: Ningún archivo seleccionado

Observaciones:

El documento se ingreso correctamente.

TIPO DE DOCUMENTO	TITULO DE DOCUMENTO	FECHA REGISTRO	INGRESADO POR	DOCUMENTO	Observaciones
OFICIO	OFICIO DEL CONSEJO DE GOBIERNO DEL PUEBLO SALASAKA	10/11/2019	OSCAR EDUARDO CURICHUMBI JEREZ	Oficio Consejo de gobierno5.pdf	NINGUNA
OFICIO	PARA PRUEBAS 4	11/09/2019	OSCAR EDUARDO CURICHUMBI JEREZ	62161.pdf	NINGUNA
CERTIFICADO	PARA PRUEBAS 3	11/09/2019	OSCAR EDUARDO CURICHUMBI JEREZ	RecoBajo - copia.docx	NINGUNA
OFICIO	DOCUMENTO DE RECIVO POR PARTE DEL MUNICIPIO	04/09/2019	OSCAR EDUARDO CURICHUMBI JEREZ	oficiorecivido.docx	NINGUNA

Figura 4.110: Diseño de Interfaz-Documentos Recibidos.
Elaborado por: Oscar Curichumbi.

DOCUMENTOS QUE SALEN DE LA COMUNIDAD

Tipo De Documento:

Titulo Del Documento:

Fecha Registro:

Documento: Ningún archivo seleccionado

Observaciones:

El documento se ingreso correctamente.

TIPO DE DOCUMENTO	TITULO DE DOCUMENTO	FECHA REGISTRO	INGRESADO POR	DOCUMENTO	Observaciones
OFICIO	OFICIO RESPUESTA AL CONSEJO DE GOBIERNO	10/11/2019	OSCAR EDUARDO CURICHUMBI JEREZ	Oficio Respuesta Consejo De Gobierno5.pdf	NINGUNA
SOLICITUD	PARA PRUEBAS 2	11/09/2019	OSCAR EDUARDO CURICHUMBI JEREZ	RecoBajo - 7.docx	NINGUNA
CERTIFICADO	PARA TERRENOS	04/09/2019	OSCAR EDUARDO CURICHUMBI JEREZ	certificadorecivido.docx	NINGUNA

Figura 4.111: Diseño de Interfaz-Documentos Entregados.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

DESCARGAR ACTAS

MOTIVO REUNION	FECHA	N° ACTA	DESCARGAR
REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	23/04/2019	1	ACTA1.docx
REUNION PARA GENERAR IDEAS D CONSULTA	03/05/2019	2	ACTA2.docx
REUNION SOBRE LA FISCALIZACION 2019	20/05/2019	3	ACTA3.docx
POR ELECCIONES DEL NUEVO PRESIDENTE	03/08/2019	7	ACTA4.docx
PRUEBA DE COBROS	05/08/2019	8	ACTA5.docx
REUNION PRUEBA 2	05/08/2019	9	ACTA6.docx

Figura 4.112: Diseño de Interfaz-Descargar Actas.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

DESCARGAR DOCUMENTOS RECIBIDOS

TIPO DE DOCUMENTO	TITULO DOCUMENTO	FECHA	DESCARGAR
OFICIO	DOCUMENTO DE RECIVO POR PARTE DEL MUNICIPIO	04/09/2019	oficiorecivido.docx
CERTIFICADO	PARA PRUEBAS 3	11/09/2019	RecoBajo - copia.docx
OFICIO	PARA PRUEBAS 4	11/09/2019	62161.pdf
OFICIO	OFICIO DEL CONSEJO DE GOBIERNO DEL PUEBLO SALASAKA	10/11/2019	Oficio Consejo de gobierno5.pdf

Figura 4.113: Diseño de Interfaz-Descargar Documentos Recibidos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

DESCARGAR DOCUMENTOS ENTREGADOS

TIPO DE DOCUMENTO	TITULO DOCUMENTO	FECHA	DESCARGAR
CERTIFICADO	PARA TERRENOS	04/09/2019	certificadorecivido.docx
SOLICITUD	PARA PRUEBAS 2	11/09/2019	RecoBajo - 7.docx
OFICIO	OFICIO RESPUESTA AL CONSEJO DE GOBIERNO	10/11/2019	Oficio Respuesta Consejo De Gobierno5.pdf

Figura 4.114: Diseño de Interfaz-Descargar Documentos Entregados.
Elaborado por: Oscar Curichumbi.

EGRESOS CON FACTURA

Concepto Del Egreso:	<input type="text"/>	Ingresar Egreso
Valor Egreso:	<input type="text" value="0.00"/>	Nuevo
Fecha Egreso:	<input type="text" value="10-11-2019"/>	Cancelar
Comprado Por:	<input type="text"/>	
Numero De Factura:	<input type="text" value="000"/> - <input type="text" value="000"/> - <input type="text"/>	
Nombre Del Local:	<input type="text"/>	
Fecha Compra:	<input type="text"/>	
Direccion Compra:	<input type="text"/>	

CONCEPTO DEL EGRESO	VALOR DEL EGRESO	FECHA EGRESO	COMPRADO POR	NUMERO DE FACTURA	LOCAL	FECHA COMPRA	DIRECCION
COMPRA DE HOJAS PARA LA IMPRESORA	5.00	11/09/2019	JOSE MARIA	001-001-00229	PAPELERIA JUANES	02/09/2019	SALASAKA
COMPRA DE MUEBLES	30.00	11/09/2019	MASAQUIZA EDUARDO	001-001-9992393293939	MUEBLERIA JEREZ	01/09/2019	SALASAKA AV129
COMPRA DE 2 SILLAS PARA LA COMUNIDAD	20.00	11/09/2019	SUSANA MASAQUIZA	001-001-00026361	SILLAS MILAN	01/09/2019	PELILEO

Figura 4.115: Diseño de Interfaz-Egresos con Factura.
Elaborado por: Oscar Curichumbi.

EGRESOS SIN FACTURA

Concepto Del Egreso:	<input type="text"/>	Ingresar Egreso
Valor Egreso:	<input type="text" value="0.00"/>	Nuevo
Fecha Egreso:	<input type="text" value="10-11-2019"/>	Cancelar
Comprado Por:	<input type="text"/>	
Direccion Compra:	<input type="text"/>	

CONCEPTO DEL EGRESO	VALOR DEL EGRESO	FECHA EGRESO	COMPRADO POR	DIRECCION
COMPRA DE ESCOBA	1.00	11/09/2019	MANUEL JEREZ	SALASAKA
COMPRA DE ALMUERZO	2.50	11/09/2019	MANUEL JEREZ	SALASAKA

Figura 4.116: Diseño de Interfaz-Egresos sin Factura.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

OTROS INGRESOS

Concepto Del Ingreso:	<input type="text"/>	Ingresar Ingreso
Valor Ingreso:	<input type="text" value="0.00"/>	Nuevo
Fecha Ingreso:	<input type="text" value="10-11-2019"/>	Cancelar

CONCEPTO DEL INGRESO	VALOR DEL INGRESO	FECHA INGRESO
DONACION DE LA JUNTA PARROQUIAL	100.00	12/09/2019

Figura 4.117: Diseño de Interfaz-Otros Ingresos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

INFORME GENERAL DE EGRESOS CON FACTURA

CONCEPTO DEL EGRESO	VALOR DEL EGRESO	FECHA EGRESO	COMPRADO POR	NUMERO DE FACTURA	LOCAL	FECHA COMPRA	DIRECCION	INGRESADO POR
COMPRA DE HOJAS PARA LA IMPRESORA	5.00	11/09/2019	JOSE MARIA	001-001-00229	PAPELERIA JUANES	02/09/2019	SALASAKA	OSCAR EDUARDO CURICHUMBI JEREZ
COMPRA DE MUEBLES	30.00	11/09/2019	MASAQUIZA EDUARDO	001-001-9992393293939	MUEBLERIA JEREZ	01/09/2019	SALASAKA AV129	OSCAR EDUARDO CURICHUMBI JEREZ
COMPRA DE 2 SILLAS PARA LA COMUNIDAD	20.00	11/09/2019	SUSANA MASAQUIZA	001-001-00026361	SILLAS MILAN	01/09/2019	PELILEO	OSCAR EDUARDO CURICHUMBI JEREZ

Figura 4.118: Diseño de Interfaz-Informe General de Egresos con Factura.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
[Cerrar sesión](#)

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Documentos](#)
[Movimientos](#)

[Reportes](#)

INFORME GENERAL DE EGRESOS SIN FACTURA

CONCEPTO DEL EGRESO	VALOR DEL EGRESO	FECHA EGRESO	COMPRADO POR	DIRECCION	INGRESADO POR
COMPRA DE ESCOBA	1.00	11/09/2019	MANUEL JEREZ	SALASAKA	OSCAR EDUARDO CURICHUMBI JEREZ
COMPRA DE ALMUERZO	2.50	11/09/2019	MANUEL JEREZ	SALASAKA	OSCAR EDUARDO CURICHUMBI JEREZ

Figura 4.119: Diseño de Interfaz-Informe General de Egresos sin Factura.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL				Bienvenido: admin Cerrar sesión			
Pagina Principal	Administracion	Reuniones	Asistencia	Tesoreria	Informes Tesoreria	Documentos	Movimientos
Reportes							
INFORME GENERAL DE OTROS INGRESOS							
CONCEPTO DEL INGRESO	VALOR DEL INGRESO	FECHA INGRESO	INGRESADO POR				
DONACION DE LA JUNTA PARROQUIAL	100.00	12/09/2019	OSCAR EDUARDO CURICHUMBI JEREZ				

Figura 4.120: Diseño de Interfaz-Informe General de Otros Ingresos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL				Bienvenido: admin Cerrar sesión			
Pagina Principal	Administracion	Reuniones	Asistencia	Tesoreria	Informes Tesoreria	Reporte Tesoreria Anual	Reportes
Documentos	Movimientos						
REPORTE GENERAL DE INGRESOS Y EGRESOS							
FECHA INICIO:	<input type="text" value="16-11-2018"/>						
FECHA FIN:	<input type="text" value="16-11-2019"/>	<input type="button" value="Nueva Consulta"/>					
<p>1 < 1 de 1 > 1 <input type="button" value="Refresh"/> <input type="button" value="Zoom"/> 100% <input type="button" value="Print"/> <input type="button" value="Lock"/></p>							
Reporte General De Ingresos Y Egresos							
Del: 16-11-2018		Al: 16-11-2019					
INGRESOS							
Multas:	1,00						
Colaboraciones:	12,00						
Sanciones:	50,00						
Otros Ingresos:	100,00						
Total Ingresos:	163,00						
EGRESOS							
Egresos Con Factura:	55,00						
Egresos Sin Factura:	3,50						
Descuentos:	2,50						
Total Egresos:	61,00						
TOTAL INGRESOS-EGRESOS:	102,00						

Figura 4.121: Diseño de Interfaz-Reporte General de Ingresos y Egresos.
Elaborado por: Oscar Curichumbi.

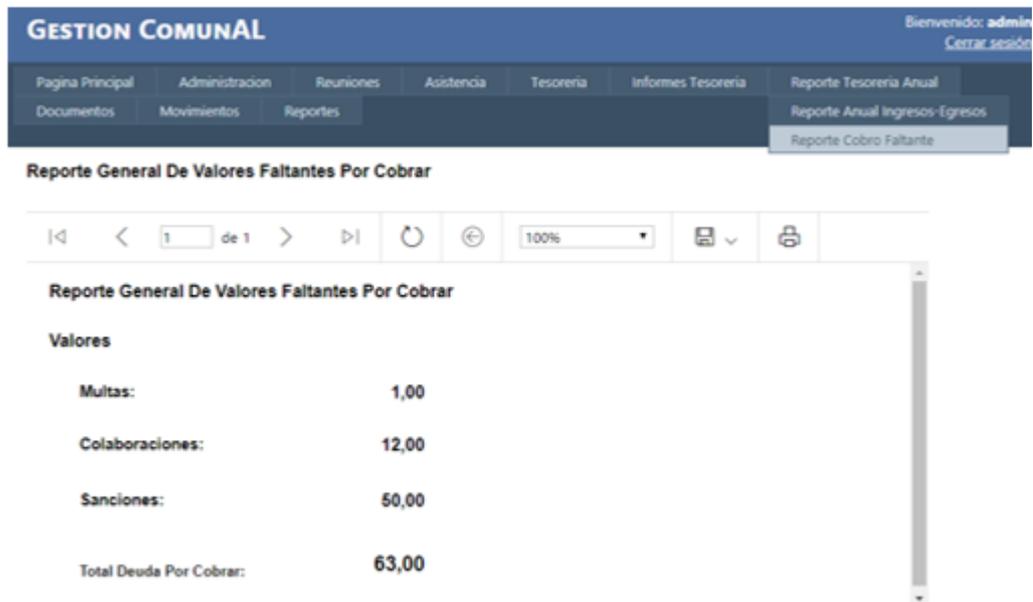


Figura 4.122: Diseño de Interfaz-Reporte General de Valores Faltantes Por Cobrar.

Elaborado por: Oscar Curichumbi.

4.3.9.7. Codificación

- Método Registrar Documentos

```

1  protected void bntSubirDocumento_Click(object sender ,
      EventArgs e){
2  DocumentosDAL dal = new DocumentosDAL();
3  Documento documetoo = new Documento();
4  if (cbTipoDocumento.SelectedIndex == 0){
5  lblMensajes.Text = "Seleccion_Tipo_De_Documento.";
6  lblMensajes.Visible = true;
7  cbTipoDocumento.Focus();
8  }
9  else{
10 if (txtTituloDocumento.Text == ""){
11 lblMensajes.Text = "Ingrese_Titulo_De_Documento.";
12 lblMensajes.Visible = true;
13 txtTituloDocumento.Focus();
14 }
15 else{
16 if (FileUpload2.HasFile){
17 string extension = Path.GetExtension(FileUpload2.FileName);

```

```

18  if (extension.ToLower() != ".doc" && extension.ToLower() !=
        ".docx" && extension.ToLower() != ".pdf"){
19  lblMensajes.Text = "Selecione otro tipo de archivo.";
20  lblMensajes.Visible = true;
21  FileUpload2.Focus();
22  }
23  else{
24  int filezise = FileUpload2.PostedFile.ContentLength;
25  if (filezise > 12582912){
26  lblMensajes.Text = "Tamaño máximo de archivo (12MB)
        excedido.";
27  lblMensajes.Visible = true;
28  FileUpload2.Focus();
29  }
30  else{
31  string documento = FileUpload2.FileName;
32  int cant = dal.SelectConteoDocumentoRecivido(documento);
33  if (cant >= 1){
34  lblMensajes.Text = "El nombre del archivo ya se encuentra
        asignado. Por favor cambiar el nombre del archivo.";
        lblMensajes.Visible = true;
35  FileUpload2.Focus();
36  }
37  else{
38  try{
39  FileUpload2.SaveAs(Server.MapPath("/") + "/Archivo/
        Recividos/" + FileUpload2.FileName);
40  int idTipoDocumento = Convert.ToInt32(cbTipoDocumento.
        SelectedItem.Value);
41  documetoo.IdTipoDocumento = idTipoDocumento;
42  documetoo.TituloDocumento = txtTituloDocumento.Text;
43  documetoo.FechaRegistro = txtFechaRegistro.Text;
44  documetoo.linkDocumento = FileUpload2.FileName;
45  documetoo.DocumentoEstado = "DOCUMENTO_RECIBIDO";
46  int idDirectivo = Convert.ToInt32(cbDEncargado.SelectedItem
        .Value);
47  documetoo.IdDirectivos = idDirectivo;
48  documetoo.Observaciones = txtObservaciones.Text;
49  try{
50  dal.Insert(documetoo);

```

```

51 lblMensajes.Text = "El documento se ingresó correctamente."
    ;
52 lblMensajes.Visible = true;
53 cargarDatos();
54 gvCarga.Visible = true;
55 cbTipoDocumento.Focus();
56 bntSubirDocumento.Visible = false;
57 cbTipoDocumento.SelectedIndex = 0;
58 txtTituloDocumento.Text = "";
59 txtFechaRegistro.Text = DateTime.Now.ToString("dd-MM-yyyy")
    ;
60 txtObservaciones.Text = "";
61 }
62 catch{
63 lblMensajes.Text = "Problemas al ingresar el documento.
    Revisar conexión al servidor.";
64 lblMensajes.Visible = true;
65 } }
66 catch (MySqlException excepcion){
67 Excepciones.Gestionar(excepcion, Plataforma.Web);
68 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
69 }
70 catch (Exception excepcion){
71 Excepciones.Gestionar(excepcion, Plataforma.Web);
72 lblMensajes.Text = "Error de c#" + "<br>" + Excepciones.
    MensajePersonalizado;}
73 } } } }
74 else{
75 lblMensajes.Text = "Seleccione archivo.";
76 lblMensajes.Visible = true;
77 FileUpload2.Focus();
78 } } } }

```

- Método Descargar Documentos

```

1 protected void gvCargarDatos_RowCommand(object sender,
    GridViewCommandEventArgs e){

```

```

2  if (e.CommandName == "Select"){
3  Response.Clear();
4  Response.ClearContent();
5  Response.ClearHeaders();
6  Response.ContentType = "application/octet-stream";
7  Response.AppendHeader("Content-Disposition", "filename=" +
    e.CommandArgument);
    Response.
    TransmitFile(Server.MapPath("~/Archivo/Entregados/") + e
    .CommandArgument);
    Response.End();
8  }
9  }

```

- Método Registrar Egresos

```

1  protected void btnIngresarEgreso_Click(object sender ,
    EventArgs e){
2  EgresosIngresosDAL dal = new EgresosIngresosDAL();
3  EgresosIngresos egresoo = new EgresosIngresos();
4  if (txtConceptoCompra.Text==""){
5  lblMensajes.Text = "Ingrese Concepto de Egreso.";
6  lblMensajes.Visible = true;
7  txtConceptoCompra.Focus();
8  String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese Concepto de Egreso.");
9  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
10 }
11 else{
12 if (txtValorEgreso.Text=="0.00"){
13 lblMensajes.Text = "Asigne Valor De Egreso.";
14 lblMensajes.Visible = true;
15 txtValorEgreso.Focus();
16 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Asigne Valor De Egreso.");
17 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
18 }
19 else{

```

```

20 if (txtPersonaCompra.Text==""){
21 lblMensajes.Text = "Ingrese el nombre de la persona quien
    hizo la compra.";
22 lblMensajes.Visible = true;
23 txtPersonaCompra.Focus();
24 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese el nombre de la persona quien hizo la
    compra.");
25 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
26 }
27 else{
28 if (txtEstablecimiento.Text=="000"){
29 lblMensajes.Text = "Asigne el numero del establecimiento.";
30 lblMensajes.Visible = true;
31 txtEstablecimiento.Focus();
32 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Asigne el numero del establecimiento.");
33 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
34 }
35 else{
36 if (txtPuntoDeEmision.Text=="000"){
37 lblMensajes.Text = "Asigne el numero del punto de emision."
    ;
38 lblMensajes.Visible = true;
39 txtPuntoDeEmision.Focus();
40 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Asigne el numero del punto de emision.");
41 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
42 }
43 else{
44 if (txtNumeracionSecuencial.Text==""){
45 lblMensajes.Text = "Asigne el numero de la Factura.";
46 lblMensajes.Visible = true;
47 txtNumeracionSecuencial.Focus();
48 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Asigne el numero de la Factura.");
49 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "

```

```

        script", txtJS, false);
50 }
51 else{
52 if (txtNombreLocal.Text==""){
53 lblMensajes.Text = "Ingrese el Nombre Del Local.";
54 lblMensajes.Visible = true;
55 txtNombreLocal.Focus();
56 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese el Nombre Del Local.");
57 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
58 }
59 else{
60 if (txtFechaCompra.Text == ""){
61 lblMensajes.Text = "Ingrese la Fecha de la Compra.";
62 lblMensajes.Visible = true;
63 txtFechaCompra.Focus();
64 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese la Fecha de la Compra.");
65 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
66 }
67 else{
68 if (txtDireccionCompra.Text == ""){
69 lblMensajes.Text = "Ingrese la Direccion de la Compra.";
70 lblMensajes.Visible = true;
71 txtDireccionCompra.Focus();
72 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Ingrese la Direccion de la Compra.");
73 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
74 }
75 else{
76 try{
77 int idDirectivos = Convert.ToInt32(cbDEncargado.
    SelectedItem.Value);
78 egresoo.IdDirectivos = idDirectivos;
79 egresoo.ConceptoEgreso = txtConceptoCompra.Text;
80 egresoo.ValorEgreso = txtValorEgreso.Text;
81 egresoo.FechaEgreso = txtFechaEgreso.Text;

```

```

82 egresoo.PersonaCompra = txtPersonaCompra.Text;
83 egresoo.DireccionCompra = txtDireccionCompra.Text;
84 egresoo.TipoEgreso = "CON_FACTURA";
85 egresoo.NumeroFactura = txtEstablecimiento.Text + "-" +
    txtPuntoDeEmision.Text + "-" +txtNumeracionSecuencial.
    Text;
86 egresoo.NombreLocalCompra = txtNombreLocal.Text;
87 egresoo.FechaCompra = txtFechaCompra.Text;
88 try{
89 dal.InsertConFactura(egresoo);
90 lblMensajes.Text = "El Egreso se ha ingresado correctamente
    .";
91 lblMensajes.Visible = true;
92 cargarDatos();
93 txtConceptoCompra.Text = "";
94 txtConceptoCompra.Focus();
95 txtValorEgreso.Text = "0.00";
96 txtFechaEgreso.Text = DateTime.Now.ToString("dd-MM-yyyy");
97 txtPersonaCompra.Text = "";
98 txtEstablecimiento.Text = "000";
99 txtPuntoDeEmision.Text = "000";
100 txtNumeracionSecuencial.Text = "";
101 txtNombreLocal.Text = "";
102 txtFechaCompra.Text = "";
103 txtDireccionCompra.Text = "";
104 }
105 catch{
106 lblMensajes.Text = "Problemas al ingresar el egreso.
    Verifique Conexion al servidor.";
107 lblMensajes.Visible = true;
108 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Problemas al ingresar el egreso. Verifique Conexion
    al servidor.");
109 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
110 } }
111 catch (MySqlException excepcion){
112 Excepciones.Gestionar(excepcion, Plataforma.Web);
113 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;

```



```

23 txtFechaIngreso.Focus();
24 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Selecione_fecha.");
25 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
26 }
27 else{
28 try{
29 int idDirectivos = Convert.ToInt32(cbDEncargado.
    SelectedItem.Value);
30 ingresoss.IdDirectivos = idDirectivos;
31 ingresoss.ConceptoIngreso = txtConceptoIngreso.Text;
32 ingresoss.ValorIngreso = txtValoIngreso.Text;
33 ingresoss.FechaIngreso = txtFechaIngreso.Text;
34 try{
35 dal.InsertIngreso(ingresoss);
36 lblMensajes.Text = "El ingreso se ha ingresado
    correctamente.";
37 lblMensajes.Visible = true;
38 cargarDatos();
39 txtConceptoIngreso.Text = "";
40 txtConceptoIngreso.Focus();
41 txtValoIngreso.Text = "0.00";
42 txtFechaIngreso.Text = DateTime.Now.ToString("dd-MM-yyyy");
43 }
44 catch{
45 lblMensajes.Text = "Problemas al ingresar el ingreso.
    Verifique Conexion al servidor.";
46 lblMensajes.Visible = true;
47 String txtJS = String.Format("<script>alert('{0}');</script
    >", "Problemas al ingresar el ingreso. Verifique
    Conexion al servidor.");
48 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
49 } }
50 catch (MySqlException excepcion){
51 Excepciones.Gestionar(excepcion, Plataforma.Web);
52 lblMensajes.Text = "Error de MySQL Server: " + "<br>" +
    Excepciones.MensajePersonalizado;
53 lblMensajes.Visible = true;

```

```

54 }
55 catch (Exception excepcion){
56 Excepciones.Gestionar(excepcion, Plataforma.Web);
57 lblMensajes.Text = "Error de c# " + "<br>" + Excepciones.
    MensajePersonalizado;
58 lblMensajes.Visible = true;
59 } } } } }

```

- Método Generar Reporte con Parametros

```

1  protected void btnGenerarInforme_Click(object sender,
    EventArgs e){
2  reportesDAL dal = new reportesDAL();
3  if (txtFechaInicio.Text == ""){
4  lblMensajes.Text = " Seleccione Fecha Inicio.";
5  lblMensajes.Visible = true;
6  String txtJS = String.Format("<script>alert('{0}');</script>
    >", " Seleccione Fecha Inicio.");
7  ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
8  }
9  else{
10 if (txtFechaFin.Text == ""){
11 lblMensajes.Text = " Seleccione Fecha Fin.";
12 lblMensajes.Visible = true;
13 String txtJS = String.Format("<script>alert('{0}');</script>
    >", " Seleccione Fecha Fin.");
14 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
    script", txtJS, false);
15 }
16 else{
17 ReporteAnual.Reset();
18 ReportDataSource repor = new ReportDataSource("
    tbReporteGeneral", GetData());
19 ReporteAnual.LocalReport.DataSources.Clear();
20 ReporteAnual.LocalReport.DataSources.Add(repor);
21 ReporteAnual.LocalReport.ReportEmbeddedResource = "
    CapaPresentacion.Reportes.Tesoreria.ReporteGeneralAnual.
    rdlc";

```

```

22 ReportParameter [] parameters = new ReportParameter [2];
23 parameters [0] = new ReportParameter ("FechaInicial",
    txtFechaInicio.Text);
24 parameters [1] = new ReportParameter ("FechaFin", txtFechaFin
    .Text);
    ReporteAnual.LocalReport.
    SetParameters(parameters);
25 ReporteAnual.LocalReport.Refresh();
26 btnGenerarInforme.Visible = false;
27 txtFechaInicio.Enabled = false;
28 txtFechaFin.Enabled = false;
29 lblMensajes.Visible = false;
30 } } }
31 private DataTable GetData() {
32 DataTable dt = new DataTable();
33 reportesDAL dal = new reportesDAL();
34 DateTime fechaDato1 = Convert.ToDateTime(txtFechaInicio.
    Text);
35 string fecha1 = fechaDato1.ToString("yyyy-MM-dd");
36 DateTime fechaDato2 = Convert.ToDateTime(txtFechaFin.Text);
37 string fecha2 = fechaDato2.ToString("yyyy-MM-dd");
38 return dt = dal.SelectReporteGeneralIngresosEgresos(fecha1,
    fecha2);
39 }

```

4.3.9.8. Pruebas de Aceptación

A continuación, se muestra las pruebas de aceptación de la Quinta Iteración:

CASO DE PRUEBA	
Código: 10	Nº Historia de Usuario: 8
Historia de Usuario: Control de Entrada y Salida de documentos.	
Condiciones de Ejecución: Solo el Secretario(a) o directivos asignados podrán realizar el control de entrada y salida de documentos.	
<p>Entrada/Pasos de Ejecución:</p> <p>Documentos Recibidos Seleccionar la pestaña Documentos, la opción Ingresar Documentos Recibidos. Seleccionar el tipo de documento, el archivo a subir al servidor, ingresar el título del documento, ingresar observaciones si es necesario y seleccionar la fecha. Como paso final damos clic en el botón subir documento para registrarlo en la base de datos. Si el documento se registró correctamente se visualizará inmediatamente el documento ingresado. El botón Nuevo limpia los campos: tipo de documento, título de documento, el archivo ya seleccionado, observaciones y la fecha. El botón cancelar concluirá con el proceso de registrar documentos recibidos.</p> <p>Documentos Entregados Seleccionar la pestaña Documentos, la opción Ingresar Documentos Entregados. Seleccionar el tipo de documento, el archivo a subir al servidor, ingresar el título del documento, ingresar observaciones si es necesario y seleccionar la fecha. Como paso final damos clic en el botón subir documento para registrarlo en la base de datos. Si el documento se registró correctamente se visualizará inmediatamente el documento ingresado. El botón Nuevo limpia los campos de tipo de documento, título de documento, el archivo ya seleccionado, observaciones y la fecha. El botón cancelar concluirá con el proceso de registrar documentos entregados.</p> <p>Descargar Actas, Documentos Recibidos y Entregados 1) Seleccionar la pestaña Documentos, la opción Descargar Actas. Para descargar actas lo único que se debe realizar es buscar el documento que desea descargar y dar clic. Su documento inmediatamente se descargará. 2) Seleccionar la pestaña Documentos, la opción Descargar Documentos Recibidos. Para descargar documentos recibidos lo único que se debe realizar es buscar el documento que desea descargar y dar clic. Su documento inmediatamente se descargará. 3) Seleccionar la pestaña Documentos, la opción Descargar Documentos Entregados. Para descargar documentos entregados lo único que se debe realizar es buscar el documento que desea descargar y dar clic. Su documento inmediatamente se descargará.</p>	
Resultado Esperado: La información del Control de Entrada y Salida de documentos será guardado en la base de datos. Los documentos registrados podrán ser descargados inmediatamente solo dando un clic en el documento deseado.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.126: Caso de Prueba: Control de Entrada y Salida de documentos.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 11	Nº Historia de Usuario: 9
Historia de Usuario: Egresos y otros Ingresos.	
Condiciones de Ejecución: Solo el Tesorero(a) o directivos asignados podrán realizar el control de Egresos y otros Ingresos.	
Entrada/Pasos de Ejecución:	
Egresos con Factura	
<p>Seleccionar la pestaña Movimientos, la opción Egresos con Factura.</p> <p>Para realizar el egreso con factura ingresamos el concepto del egreso, valor de egreso, fecha de egreso, el nombre de la persona que hizo la compra, numero de factura, nombre del local en donde se realizó la compra, fecha de compra y la dirección del local. Para terminar con el proceso se debe dar clic en el botón Ingresar Egreso para registrarlo en la base de datos. Si el egreso con factura se registró correctamente se visualizará inmediatamente el egreso ingresado.</p> <p>El botón Nuevo limpia los campos: concepto del egreso, valor de egreso, fecha de egreso, el nombre de la persona que hizo la compra, numero de factura, nombre del local en donde se realizó la compra, fecha de compra y la dirección del local.</p> <p>El botón cancelar concluirá con el proceso de registrar egresos con factura.</p>	
Egresos Sin Factura	
<p>Seleccionar la pestaña Movimientos, la opción Egresos sin Factura.</p> <p>Para realizar el egreso sin factura ingresamos el concepto del egreso, valor de egreso, fecha de egreso, el nombre de la persona que hizo la compra y la dirección donde se realizó la compra. Para terminar con el proceso se debe dar clic en el botón Ingresar Egreso para registrarlo en la base de datos. Si el egreso sin factura se registró correctamente se visualizará inmediatamente el egreso ingresado.</p> <p>El botón Nuevo limpia los campos: concepto del egreso, valor de egreso, fecha de egreso, el nombre de la persona que hizo la compra y la dirección donde se realizó la compra.</p> <p>El botón cancelar concluirá con el proceso de registrar egresos sin factura.</p>	
Otros Ingresos	
<p>Seleccionar la pestaña Movimientos, la opción Otros Ingresos.</p> <p>Para realizar otros ingresos ingresamos el concepto del ingreso, valor del ingreso y la fecha del ingreso. Para terminar con el proceso se debe dar clic en el botón Ingresar Ingreso para registrarlo en la base de datos. Si el ingreso se registró correctamente se visualizará inmediatamente el ingreso registrado.</p> <p>El botón Nuevo limpia los campos: concepto del ingreso, valor del ingreso y fecha del ingreso.</p> <p>El botón cancelar concluirá con el proceso de registrar otros ingresos.</p>	
Informes de Egresos y Otros Ingresos	
<p>1) Seleccionar la pestaña Movimientos, la opción Informe de Egresos con Factura. Inmediatamente se mostrará la información de los egresos con factura que fueron registrados.</p> <p>2) Seleccionar la pestaña Movimientos, la opción Informe de Egresos sin Factura. Inmediatamente se mostrará la información de los egresos sin factura que fueron registrados.</p> <p>3) Seleccionar la pestaña Movimientos, la opción Informe de Otros Ingresos. Inmediatamente se mostrará la información de Otros ingresos que fueron registrados.</p>	
Resultado Esperado: La información de los Egresos y otros Ingresos serán guardados en la base de datos.	
Los Egresos y otros Ingresos registrados podrán ser visualizados en los informes.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.127: Caso de Prueba: Egresos y otros Ingresos.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 12	Nº Historia de Usuario: 10
Historia de Usuario: Reporte general de Ingresos y Egresos.	
Condiciones de Ejecución: Solo el Tesorero(a) podrá Generar el Reporte General de Ingresos y Egresos.	
Entrada/Pasos de Ejecución: Reporte General de Ingresos y Egresos Seleccionar la pestaña Reporte Tesorería Movimientos, la opción Reporte Anual Ingresos-Egresos. Seleccionamos la Fecha Inicio y la Fecha Fin, después damos clic en el botón Generar Reporte para visualizar los Ingresos y Egresos por Fechas. Este reporte informa el valor total de Ingresos y el valor total de Egresos que se realizaron entre dos fechas, este reporte dará a conocer si la comunidad tiene o no fondos monetarios. El botón Nueva Consulta permitirá realizar un nuevo reporte con sus respectivas fechas. Reporte de Valores Faltantes Por Cobrar Seleccionar la pestaña Reporte Tesorería Movimientos, la opción Reporte Cobro Faltante. Este reporte se generará inmediatamente y mostrara el saldo total de los valores que faltan por cobrar. Los valores son las multas de asistencia, las colaboraciones y las sanciones, estos son los valores que la comunidad tiene para obtener fondos monetarios.	
Resultado Esperado: Los reportes visualizaran la información de los Ingresos y Egresos que se han realizado en la comunidad entre fechas. También la información el total de la deuda que falta por cobrar de las multas, colaboraciones y sanciones.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.128: Caso de Prueba: Reporte general de Ingresos y Egresos.

Elaborado por: Oscar Curichumbi.

4.3.9.9. Resultados de la Quinta Iteración

Como resultado de entrega de la quinta iteración que consta de los módulos: Control de Entrada y Salida de documentos, Egresos y otros Ingresos y Reporte general de Ingresos y Egresos, el cliente quedó satisfecho con las funcionalidades de los módulos que se desarrollaron, la quinta iteración se ha concluido sin ningún problema manifestado por el cliente. A continuación, se realizará la sexta iteración.

4.3.10. Desarrollo de la Sexta Iteración

Se realizará la descripción general de las tareas respectivas de las cuatro historias de usuario, que pertenecen a la sexta iteración. Como se indica en la planificación se desarrollará los módulos: Reportes del presidente o vicepresidente, Reportes del Secretario(a), Reportes de Tesorería y Reportes de los Vocales. En la Tabla 4.128 se muestra de manera general las tareas de ingeniería correspondientes a la historia de usuario número 11-12-13-14 para cumplir con lo planificado.

Numero de Tarea	Numero de Historia	Esfuerzo	Nombre de la Tarea
59	11	1	Diseño de la interfaz para los Reportes del presidente o vicepresidente.
60	11	2	Generar Reportes del presidente o vicepresidente.
61	12	1	Diseño de la interfaz para los Reportes del Secretario(a).
62	12	2	Generar Reportes del Secretario(a).
63	13	2	Diseño de la interfaz para los Reportes de Tesorería.
64	13	3	Generar Reportes de Tesorería.
65	14	1	Diseño de la interfaz para los Reportes de los Vocales.
66	14	2	Generar Reportes de los Vocales.

Tabla 4.129: Tareas a Realizar de las Historias de Usuario 11-12-13-14.

Elaborado por: Oscar Curichumbi.

4.3.10.1. Estimación de la Primera Iteración

A continuación, se muestra la estimación de la Sexta Iteración:

ITERACIÓN 6				
N.- Historia	Fecha Inicio	Fecha Final	DÍAS	HORAS
11	09/09/2019	11/09/2019	3	24
12	12/09/2019	14/09/2019	3	24
13	15/09/2019	19/09/2109	5	40
14	20/09/2019	22/09/2019	3	24
TOTAL ESTIMACIÓN:			14	112

Tabla 4.130: Horas asignadas para el desarrollo de la Iteración 6.

Elaborado por: Oscar Curichumbi.

4.3.10.2. Bocetos de las Historias de Usuarios 11-12-13-14

En las siguientes figuras se muestran los bocetos de las Historias de Usuarios 11-12-13-14 que forman parte de la Sexta Iteración.

Reporte De Usuarios Activos

ID	cedula	Nombres y Apellidos	email	celular	sexo	Fecha Ingreso

Figura 4.123: Boceto de Reporte de Usuarios Activos.
Elaborado por: Oscar Curichumbi.

Reporte De Usuarios NO Activos

ID	cedula	Nombres y Apellidos	email	celular	sexo	Fecha Ingreso

Figura 4.124: Boceto de Reporte de Usuarios No Activos.
Elaborado por: Oscar Curichumbi.

Reporte Usuarios General

ID	cedula	Nombres y Apellidos	e-mail	celular	sexo	Fecha Ingreso

Figura 4.125: Boceto de Reporte de Usuarios.
Elaborado por: Oscar Curichumbi.

Reporte Directivos Activos			
Nombres y Apellidos	Nombre Usuario	Fecha Inicio	Cargo

Figura 4.126: Boceto de Reporte de Directivos Activos.
Elaborado por: Oscar Curichumbi.

Reporte Periodo Finalizado					
Nombres y Apellidos	Nombre Usuario	Fecha Inicio	Fecha FIN	Cargo	Observacion

Figura 4.127: Boceto de Reporte de Directivos Con Periodo Finalizado.
Elaborado por: Oscar Curichumbi.

Reporte Reuniones Activas			
Tipo De Reunion	Motivo Reunion	Fecha Reunion	

Figura 4.128: Boceto de Reporte de Reuniones Activas.
Elaborado por: Oscar Curichumbi.

Reporte De Reuniones Finalizadas

Tipo Reunion	Reunion	Fecha Reunion	Nº Acta	Nombre Archivo	Generado Por

Figura 4.129: Boceto de Reporte de Reuniones Finalizadas.
Elaborado por: Oscar Curichumbi.

Reporte de Asistencia Por Reunion

Motivo Reunion:

Usuarios	Reunion	Fecha Asistencia	Estado	Saldo	Estado Deuda

Figura 4.130: Boceto de Reporte de Asistencia Por Reunión.
Elaborado por: Oscar Curichumbi.

Reporte de Asistencia Por Usuario

Usuario:

Reunion	Fecha Asistencia	Estado	Saldo	Estado Deuda

Figura 4.131: Boceto de Reporte de Asistencia Por Usuario.
Elaborado por: Oscar Curichumbi.

Reporte de Asistencia Por Estado

Estado:

Usuarios	Reunion	Fecha Asistencia	Estado	saldo	Estado Devda

Figura 4.132: Boceto de Reporte de Asistencia Por Estado.
Elaborado por: Oscar Curichumbi.

Reporte Registro Colaboración Por Reunion

Reunion:

Motivo Colaboración:

Usuarios	Fecha Colaboración	saldo Total	Estado

Figura 4.133: Boceto de Reporte Registro de Colaboración Por Reunión.
Elaborado por: Oscar Curichumbi.

Reporte Registro Colaboración Por Usuario

Usuario:

Buscar Usuario

Generar Reporte

Nueva consulta

Reunion	Motivo Colaboracion	Fecha Colaboración	Saldo	Estado

Figura 4.134: Boceto de Reporte Registro de Colaboración Por Usuario.
Elaborado por: Oscar Curichumbi.

Reporte General De Sanciones

ID	Nombres Completos	Reunion	Motivo Sancion	Saldo	Estado Deuda

Figura 4.135: Boceto de Reporte General de Sanciones.
Elaborado por: Oscar Curichumbi.

Reporte Cobro Asistencia Por Usuario

Usuario:

Buscar Usuario

Generar Reporte

Nueva consulta

Pagado Por	Cobrado Por	Monto Pago	Fecha Pago	Reunion	Estado multa	Estado Deuda

Figura 4.136: Boceto de Reporte Cobro de Asistencia Por Usuario.
Elaborado por: Oscar Curichumbi.

Reporte Cobro de Asistente Por Reunion

Reunion:

Usuario Deudor	Pagado Por	cobrado Por	monto Pago	Fecha Pago	Estado	Multa

Figura 4.137: Boceto de Reporte Cobro de Asistencia Por Reunión.
 Elaborado por: Oscar Curichumbi.

Reporte Cobro Colaboración Por Usuario

Usuario:

Pagado Por	cobrado por	monto Pago	Fecha Pago	motivo Colaboración

Figura 4.138: Boceto de Reporte Cobro de Colaboración Por Usuario.
 Elaborado por: Oscar Curichumbi.

Reporte Cobro Colaboración Por Reunion

Reunion:

Usuario Deudor	Pagado Por	cobrado Por	Monto Pago	Fecha Pago	motivo Colaboración

Figura 4.139: Boceto de Reporte Cobro de Colaboración Por Reunión.
 Elaborado por: Oscar Curichumbi.

Reporte Cobro De Sanciones

Usuario Deudor	Pagado Por	Cobrado Por	Monto Pago	Fecha Pago	Reunion	Valor

Figura 4.140: Boceto de Reporte Cobro de Sanciones.
Elaborado por: Oscar Curichumbi.

Reporte General de Descuentos

Usuarios	Reunion	Motivo Descuento	Valor	Deudo Total	Fecha	Ingresado

Figura 4.141: Boceto de Reporte General de Descuentos.
Elaborado por: Oscar Curichumbi.

Reporte de Egreso Sin Factura

Concepto Egreso	Valor Egreso	Fecha Egreso	Comprado Por	Dirección Compra

Figura 4.142: Boceto de Reporte de Egresos Sin Factura.
Elaborado por: Oscar Curichumbi.

Reporte de Egreso Con Factura

Concepto Egreso	Valor	Fecha	Comprado Por	Numero Factura	Fecha compra	Dirección

Figura 4.143: Boceto de Reporte de Egresos Con Factura.
Elaborado por: Oscar Curichumbi.

Reporte de Otros Ingresos

Concepto de Ingreso	Valor Ingreso	Fecha Ingreso	Ingresado Por

Figura 4.144: Boceto de Reporte de Otros Ingresos.
Elaborado por: Oscar Curichumbi.

Reporte General de Documentos

Ingresado por	Tipo Documento	Titulo Documento	Fecha registro	Nombre Archivo	Documento

Figura 4.145: Boceto de Reporte General de Documentos.
Elaborado por: Oscar Curichumbi.

4.3.10.3. Descripción de las Tareas de Ingeniería

En base a la planificación se ha realizado las tareas de ingeniería, en estas se ha asignado el tiempo de estimación para el desarrollo de cada una de ellas. Si el cliente está satisfecho con el desarrollo de las tareas, que forman parte de las

cuatro historias de usuario, esto cumplirá con las necesidades del cliente y de este modo se culminará la sexta iteración. A continuación, se muestra el desarrollo de las tareas de cada historia de usuario.

Tarea de Ingeniería	
Numero de Tarea: 59	Numero de Historia: 11
Nombre de Tarea: Diseño de la interfaz para los Reportes del presidente o vicepresidente.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 09/09/2019	Fecha Fin:09/04/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la interfaz, donde el Presidente o Vicepresidente podrán visualizar la información de sus respectivos reportes, con la finalidad de informar la situación actual de la comunidad a los usuarios.	

Tabla 4.131: Tarea 59 de la Historia de Usuario N.-11.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 60	Numero de Historia: 11
Nombre de Tarea: Generar Reportes del presidente o vicepresidente.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 10/09/2019	Fecha Fin:11/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Estos reportes serán generados con parámetros o sin parámetros, estos reportes ayudaran al presidente o al vicepresidente con la información de las actividades que han realizado en la comunidad.	

Tabla 4.132: Tarea 60 de la Historia de Usuario N.-11.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 61	Numero de Historia: 12
Nombre de Tarea: Diseño de la interfaz para los Reportes del Secretario(a).	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 12/09/2019	Fecha Fin:12/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la interfaz, donde el Secretario(a) podrá visualizar la información de sus respectivos reportes, con la finalidad de informar la situación actual de la comunidad a los usuarios.	

Tabla 4.133: Tarea 61 de la Historia de Usuario N.-12.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 62	Numero de Historia: 12
Nombre de Tarea: Generar Reportes del Secretario(a).	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 13/09/2019	Fecha Fin:14/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Estos reportes serán generados con parámetros o sin parámetros, estos reportes ayudaran al Secretario(a) con la información de las actividades que ha realizado en la comunidad.	

Tabla 4.134: Tarea 62 de la Historia de Usuario N.-12.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 63	Numero de Historia: 13
Nombre de Tarea: Diseño de la interfaz para los Reportes de Tesorería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 15/09/2019	Fecha Fin:16/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la interfaz, donde el Tesorero(a) podrá visualizar la información de sus respectivos reportes, con la finalidad de informar la situación actual de la comunidad a los usuarios.	

Tabla 4.135: Tarea 63 de la Historia de Usuario N.-13.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 64	Numero de Historia: 13
Nombre de Tarea: Generar Reportes de Tesorería.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 17/09/2019	Fecha Fin:19/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Estos reportes serán generados con parámetros o sin parámetros, estos reportes ayudaran al Tesorero(a) con la información de las actividades que ha realizado en la comunidad.	

Tabla 4.136: Tarea 64 de la Historia de Usuario N.-13.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 65	Numero de Historia: 14
Nombre de Tarea: Diseño de la interfaz para los Reportes de los Vocales.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/09/2019	Fecha Fin:20/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Diseño de la interfaz, donde los Vocales podrán visualizar la información de sus respectivos reportes, con la finalidad de informar la situación actual de la comunidad a los usuarios.	

Tabla 4.137: Tarea 65 de la Historia de Usuario N.-14.

Elaborado por: Oscar Curichumbi.

Tarea de Ingeniería	
Numero de Tarea: 66	Numero de Historia: 14
Nombre de Tarea: Generar Reportes de los Vocales.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 21/09/2019	Fecha Fin: 22/09/2019
Programador Responsable: Oscar Curichumbi	
Descripción: Estos reportes serán generados con parámetros o sin parámetros, estos reportes ayudaran a los Vocales con la información de las actividades que han realizado en la comunidad.	

Tabla 4.138: Tarea 66 de la Historia de Usuario N.-14.

Elaborado por: Oscar Curichumbi.

4.3.10.4. Diagrama de datos

A continuación, se muestra el diagrama de datos general para realizar los respectivos reportes:

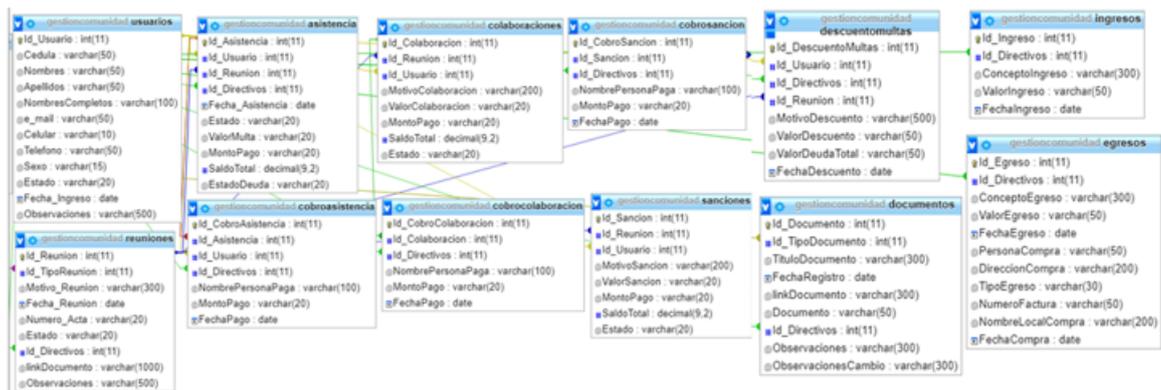


Figura 4.146: Diagrama de datos: Reportes de Directivos.

Elaborado por: Oscar Curichumbi.

4.3.10.5. Diseño de Interfaz

A continuación, se muestra los diseños de interfaz de la Sexta Iteración:

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos

Documentos Movimientos Reportes

REPORTE DE USUARIOS ACTIVOS

⏪ < de 1 > ⏩
🔄 🏠 📄 🖨️

REPORTE DE USUARIOS ACTIVOS

ID	CEDULA	NOMBRES Y APELLIDOS	E_MAIL	CELULAR	SEXO	FECHA INGRESO
1	1805198478	OSCAR EDUARDO CURICHUMBI JEREZ	oscar@gmail.com	0993552849	MASCULINO	19/07/2019
2	2000039434	JORGE MANUEL MASAQUIZA MASAQUIZA	oscar@gmail.com	0999999999	MASCULINO	19/07/2019
3	1303753618	EDUARDO JEREZ	oscar@gmail.com	0999999999	MASCULINO	19/07/2019

Figura 4.147: Diseño de Interfaz-Reporte de Usuarios Activos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos

Documentos Movimientos Reportes

REPORTE DE USUARIOS NO ACTIVOS

⏪ < de 1 > ⏩
🔄 🏠 📄 🖨️

REPORTE DE USUARIOS NO ACTIVOS

ID	CEDULA	NOMBRES Y APELLIDOS	E_MAIL	CELULAR	SEXO	FECHA INGRESO
4	1804864401	WILMER RICHARD MASAUIZA MASAQUIZA	mskz97wil@gmail.com	0997588202	MASCULINO	15/08/2019

Figura 4.148: Diseño de Interfaz-Reporte de Usuarios No Activos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos

Documentos Movimientos Reportes

REPORTE USUARIOS

⏪ < 1 de 1 > ⏩
↻ ⏮ 100% ⏭ 🖨

REPORTE USUARIOS

ID	CEDULA	NOMBRES Y APELLIDOS	E_MAIL	CELULAR	SEXO	ESTADO	FECHA INGRESO
1	1805198478	OSCAR EDUARDO CURICHUMBI JEREZ	oscar@gmail.com	0993552849	MASCULINO	ACTIVO	19/07/2019
2	2000039434	JORGE MANUEL MASAQUIZA MASAQUIZA	oscar@gmail.com	0999999999	MASCULINO	ACTIVO	19/07/2019
3	1303753618	EDUARDO JEREZ	oscar@gmail.com	0999999999	MASCULINO	ACTIVO	19/07/2019
4	1804864401	WILMER RICHARD MASAUIZA MASAQUIZA	mskz97wil@gmail.com	0997588202	MASCULINO	NO ACTIVO	15/08/2019

Figura 4.149: Diseño de Interfaz-Reporte de Usuarios.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos

Documentos Movimientos Reportes

DIRECTIVOS ACTIVOS

⏪ < 1 de 1 > ⏩
↻ ⏮ 100% ⏭ 🖨

DIRECTIVOS ACTIVOS

NOMBRES Y APELLIDOS	NOMBRE USUARIO	FECHA INICIO	CARGO
OSCAR EDUARDO CURICHUMBI JEREZ	ADMIN	30/04/2019	ADMINISTRADOR
JORGE MANUEL MASAQUIZA MASAQUIZA	oscar	01/07/2019	PRESIDENTE
EDUARDO JEREZ	edu	19/07/2019	PRESIDENTE

Figura 4.150: Diseño de Interfaz-Reporte de Directivos Activos.
Elaborado por: Oscar Curichumbi.

REPORTE PERIODO FINALIZADO

⏪
<
1 de 1
>
⏩
🔄
⏮
100%
📄
🖨

REPORTE PERIODO FINALIZADO

ID	NOMBRES Y APELLIDOS	NOMBRE USUARIO	FECHA INICIO	FECHA FIN	CARGO	OBSERVACION
3	JORGE MANUEL MASAQUIZA MASAQUIZA	jorge	19/07/2019	07/08/2019	VOCAL	
5	OSCAR EDUARDO CURICHUMBI JEREZ	oscar2	23/07/2019	23/07/2019	PRESIDENTE	
6	OSCAR EDUARDO CURICHUMBI JEREZ	oscar3	23/07/2019	23/07/2019	VOCAL	

Figura 4.151: Diseño de Interfaz-Reporte de Directivos Finalizados.
Elaborado por: Oscar Curichumbi.

REPORTE DE REUNIONES ACTIVAS

⏪
<
1 de 1
>
⏩
🔄
⏮
100%
📄
🖨

REPORTE DE REUNIONES ACTIVAS

TIPO DE REUNION	MOTIVO DE LA REUNION	FECHA REUNION
Reunion Ordinaria	REUNION DE DIRECTIVOS	24/05/2019
Reunion Ordinaria	REUNION GENERAL	01/01/2019
Reunion Ordinaria	PRUEBA DE REUNIONES	01/07/2019
Reunion Ordinaria	PARA COMPROBAR DIRECTIVO	07/08/2019
Reunion Ordinaria	POR PRUEBA DIRECTIVO2	07/08/2019
Reunion Ordinaria	POR PRUEBA DIRECTIVO3	07/08/2019
Reunion Ordinaria	POR PRUEBA DIRECTIVO4	07/08/2019
Reunion Ordinaria	POR PRUEBA DIRECTIVO2	20/09/2019
Reunion Ordinaria	COBROS	24/10/2019
Reunion Extraordinaria	COBROS 2	24/10/2019
Reunion Ordinaria	COBROS 3	24/10/2019

Figura 4.152: Diseño de Interfaz-Reporte de Reuniones Activas.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Reporte Tesorería Movimientos](#)

[Documentos](#)
[Movimientos](#)
[Reportes](#)

REPORTE DE REUNIONES FINALIZADAS

1 de 1
100%

Tipo Reunion	Reunion	Fecha Reunion	N.- Acta	Nombre Archivo	Generado Por	Observaci
Reunion Ordinaria	REUNION SOBRE ELECCIONES DEL CONSEJO DE GOBIERNO	23/04/2019	1	ACTA1.docx	OSCAR EDUARDO CURICHUMBI JEREZ	
Reunion Extraordinaria	REUNION PARA GENERAR IDEAS D CONSULTA	03/05/2019	2	ACTA2.docx	OSCAR EDUARDO CURICHUMBI JEREZ	
Reunion Extraordinaria	REUNION SOBRE LA FISCALIZACION 2019	20/05/2019	3	ACTA3.docx	OSCAR EDUARDO CURICHUMBI JEREZ	
Reunion Ordinaria	POR ELECCIONES DEL NUEVO PRESIDENTE	03/08/2019	7	ACTA4.docx	OSCAR EDUARDO CURICHUMBI JEREZ	POR MAL INGRESO ARCHIVO LA SECRETA
Reunion Ordinaria	PRUEBA DE COBROS	05/08/2019	8	ACTA5.docx	OSCAR EDUARDO CURICHUMBI JEREZ	
Reunion Ordinaria	REUNION PRUEBA 2	05/08/2019	9	ACTA6.docx	OSCAR EDUARDO CURICHUMBI JEREZ	MODIFICA PRUEBA

Figura 4.153: Diseño de Interfaz-Reporte de Reuniones Finalizadas.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

[Pagina Principal](#)
[Administración](#)
[Reuniones](#)
[Asistencia](#)
[Tesorería](#)
[Informes Tesorería](#)
[Reporte Tesorería Movimientos](#)

[Documentos](#)
[Movimientos](#)
[Reportes](#)

REPORTE DE ASISTENCIA POR REUNION

MOTIVO DE REUNION:

1 de 1
100%

Usuarios	Reunion	Fecha Asistencia	Estado	Saldo Total	Estado Deuda
OSCAR EDUARDO CURICHUMBI JEREZ	COBROS 3	24/10/2019	ATRASADO	0.00	CANCELA
JORGE MANUEL MASAQUIZA MASAQUIZA	COBROS 3	24/10/2019	ATRASADO	1.00	PENDIENT
EDUARDO JEREZ	COBROS 3	24/10/2019	ATRASADO	0.00	CANCELA

Figura 4.154: Diseño de Interfaz-Reporte de Asistencia Por Reunión.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal | Administración | Reuniones | Asistencia | Tesorería | Informes Tesorería | Reporte Tesorería Movimientos

Documentos | Movimientos | Reportes

REPORTE DEUDA POR ESTADO

ESTADO:

1 de 1

100%

REPORTE MULTAS POR ESTADO

Usuarios	Reunion	Fecha Asistencia	Estado	Saldo Total	Estado Deuda
JORGE MANUEL MASAQUIZA MASAQUIZA	COBROS 3	24/10/2019	ATRASADO	1.00	PENDIEN

Figura 4.155: Diseño de Interfaz-Reporte de Asistencia Por Estado.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal | Administración | Reuniones | Asistencia | Tesorería | Informes Tesorería | Reporte Tesorería Movimientos

Documentos | Movimientos | Reportes

REPORTE DE ASISTENCIA POR USUARIO

Usuario:

1 de 1

100%

REPORTE DE ASISTENCIA POR USUARIO

Reunion	Fecha Asistencia	Estado	Saldo Total	Estado Deuda
COBROS 3	24/10/2019	ATRASADO	0.00	CANCELADO
COBROS 2	24/10/2019	AUSENTE	0.00	CANCELADO
COBROS	24/10/2019	ATRASADO	0.00	CANCELADO
POR PRUEBA DIRECTIVO2	23/10/2019	AUSENTE	0.00	CANCELADO
POR PRUEBA DIRECTIVO4	21/08/2019	ATRASADO	0.00	CANCELADO
POR PRUEBA DIRECTIVO3	19/10/2019	ATRASADO	0.00	CANCELADO
PARA COMPROBAR DIRECTIVO	23/10/2019	AUSENTE	0.00	CANCELADO
REUNION PRUEBA 2	05/08/2019	AUSENTE	0.00	CANCELADO

Figura 4.156: Diseño de Interfaz-Reporte de Asistencia Por Usuario.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos
Documentos Movimientos Reportes

REPORTE REGISTRO COLABORACIONES POR REUNION

REUNION: REUNION SOBRE LA FISCALIZACION 2019

Nueva Consulta

1 de 1

100%

REPORTE REGISTRO DE COLABORACIONES POR REUNION

MOTIVO COLABORACION: PARA COMPRAR VIVERES

Usuarios	Fecha Colaboracion	Saldo Total	Estado
OSCAR EDUARDO CURICHUMBI JEREZ	20/05/2019	0,00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	20/05/2019	0,00	CANCELADO

Figura 4.157: Diseño de Interfaz-Reporte de Registro Colaboraciones Por Reunión.

Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos
Documentos Movimientos Reportes

REPORTE COLABORACION POR USUARIO

Usuario: OSCAR EDUARDO CURICHUMBI JEREZ

Nueva Consulta

1 de 1

100%

REPORTE COLABORACION POR USUARIO

Reunion	Fecha Colaboracion	Motivo Colaboracion	Saldo Total	Estado
REUNION DE DIRECTIVOS	24/05/2019	PARA COMPRAR COMIDA PARA EL EVENTO DE LA REINA	0,00	CANCELADO
REUNION SOBRE LA FISCALIZACION 2019	20/05/2019	PARA COMPRAR VIVERES	0,00	CANCELADO
REUNION PARA GENERAR IDEAS D CONSULTA	03/05/2019	PARA PINTURA	0,00	CANCELADO

Figura 4.158: Diseño de Interfaz-Reporte de Registro Colaboraciones Por Usuario.

Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal | Administración | Reuniones | Asistencia | Tesorería | Informes Tesorería | Reporte Tesorería Movimientos

Documentos | Movimientos | Reportes

REPORTE GENERAL DE SANCIONES

1 de 1

Reporte General De Sanciones

ID	NOMBRES COMPLETOS	REUNION	MOTIVO SANCION	SALDO DEUDA	ESTADO DEUDA
1	OSCAR EDUARDO CURICHUMBI JEREZ	REUNION DE DIRECTIVOS	POR MAL INGRESO DE FONDOS	0,00	CANCELADO
2	JORGE MANUEL MASAQUIZA MASAQUIZA	REUNION DE DIRECTIVOS	POR MAL INGRESO DE FONDOS	0,00	CANCELADO
3	EDUARDO JEREZ	REUNION DE DIRECTIVOS	POR MAL INGRESO DE FONDOS	20,00	PENDIENTE
4	OSCAR EDUARDO CURICHUMBI JEREZ	REUNION GENERAL	HOLAA	20,00	PENDIENTE
5	EDUARDO JEREZ	PRUEBA DE REUNIONES	PRUEBA	10,00	PENDIENTE

Figura 4.159: Diseño de Interfaz-Reporte General de Sanciones.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal | Administración | Reuniones | Asistencia | Tesorería | Informes Tesorería | Reporte Tesorería Movimientos

Documentos | Movimientos | Reportes

REPORTE COBRO ASISTENCIA POR USUARIO

USUARIO:

1 de 1

REPORTE DE COBROS DE MULTAS POR ASISTENCIA

Pagado Por	Cobrado Por	Monto Pago	Fecha Pago	Reunion	Estado	Multa	Estado Deuda
DESCUENTO	OSCAR EDUARDO CURICHUMBI JEREZ	1.75	24/10/2019	COBROS 3	ATRASADO	5.00	CANCELAD
DESCUENTO	OSCAR EDUARDO CURICHUMBI JEREZ	3.25	24/10/2019	COBROS 3	ATRASADO	5.00	CANCELAD
DESCUENTO	OSCAR EDUARDO CURICHUMBI JEREZ	9.25	24/10/2019	COBROS 2	AUSENTE	20.00	CANCELAD
DESCUENTO	OSCAR EDUARDO CURICHUMBI JEREZ	10.75	24/10/2019	COBROS 2	AUSENTE	20.00	CANCELAD
DESCUENTO	OSCAR EDUARDO CURICHUMBI JEREZ	2.00	24/10/2019	COBROS	ATRASADO	2.00	CANCELAD
OSCAR	OSCAR EDUARDO CURICHUMBI JEREZ	0.50	23/10/2019	PARA COMPROBAR DIRECTIVO	AUSENTE	20.00	CANCELAD
DESCUENTO	OSCAR EDUARDO CURICHUMBI JEREZ	1.50	23/10/2019	PARA COMPROBAR DIRECTIVO	AUSENTE	20.00	CANCELAD
DESCUENTO	OSCAR EDUARDO CURICHUMBI JEREZ	17.00	23/10/2019	PARA COMPROBAR	AUSENTE	20.00	CANCELAD

Figura 4.160: Diseño de Interfaz-Reporte de Cobro Asistencia Por Usuario.
Elaborado por: Oscar Curichumbi.

REPORTE COBRO ASISTENCIA POR REUNION

REUNION:

REPORTE COBRO DE ASISTENCIA POR REUNION

Usuario Deudor	Pagado Por	Cobrado Por	Monto Pago	Fecha Pago	Estado	Multa	Estado Deuda
OSCAR EDUARDO CURICHUMBI JEREZ	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	2.00	15/08/2019	ATRASADO	2.00	CANCELAI
JORGE MANUEL MASAQUIZA MASAQUIZA	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	2.00	19/08/2019	ATRASADO	2.00	CANCELAI
EDUARDO JEREZ	OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	2.00	19/08/2019	ATRASADO	2.00	CANCELAI

Figura 4.161: Diseño de Interfaz-Reporte de Cobro Asistencia Por Reunión. Elaborado por: Oscar Curichumbi.

REPORTE COBRO COLABORACIONES POR USUARIO

USUARIO:

REPORTE COBRO COLABORACION

Pagado Por	Cobrado Por	Monto Pago	Fecha Pago	Motivo Colaboracion	Valor Colaboracion	Estado Deuda
OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	21/08/2019	PARA COMPRAR COMIDA PARA EL EVENTO DE LA REINA	20.00	CANCELADO
MANUEL	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	21/08/2019	PARA COMPRAR COMIDA PARA EL EVENTO DE LA REINA	20.00	CANCELADO
OSCAR CURICHUMBI	OSCAR EDUARDO CURICHUMBI JEREZ	10.00	21/08/2019	PARA COMPRAR COMIDA PARA EL EVENTO DE LA REINA	20.00	CANCELADO
JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	0.20	15/08/2019	PARA COMPRAR VIVERES	5.00	CANCELADO

Figura 4.162: Diseño de Interfaz-Reporte de Cobro Colaboraciones Por Usuario. Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos
Documentos Movimientos Reportes

REPORTE COBRO COLABORACION POR REUNION

REUNION: REUNION SOBRE LA FISCALIZACION 2019

Nueva Consulta

1 de 1 100%

REPORTE COBRO COLABORACION POR REUNION

Usuario Deudor	Pagado Por	Cobrado Por	Monto Pago	Fecha Pago	Motivo Colaboracion	Valor Colaboracion	Estado Deuda
OSCAR EDUARDO CURICHUMBI JEREZ	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	1.55	15/08/2019	PARA COMPRAR VIVERES	5.00	CANCE
OSCAR EDUARDO CURICHUMBI JEREZ	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	3.25	15/08/2019	PARA COMPRAR VIVERES	5.00	CANCE
OSCAR EDUARDO CURICHUMBI JEREZ	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI JEREZ	0.20	15/08/2019	PARA COMPRAR VIVERES	5.00	CANCE
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE MASAQUIZA	OSCAR EDUARDO CURICHUMBI	3.25	15/08/2019	PARA COMPRAR VIVERES	5.00	CANCE

Figura 4.163: Diseño de Interfaz-Reporte de Cobro Colaboraciones Por Reunión. Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Página Principal Administración Reuniones Asistencia Tesorería Informes Tesorería Reporte Tesorería Movimientos
Documentos Movimientos Reportes

REPORTE COBRO DE SANCIONES

1 de 1 100%

REPORTE COBRO DE SANCIONES

Usuario Deudor	Pagado Por	Cobrado Por	Monto Pago	Fecha Pago	Reunion	Valor Sancion	Estado Deuda
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	5.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	1.50	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	2.35	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	1.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO
JORGE MANUEL MASAQUIZA MASAQUIZA	JORGE	OSCAR EDUARDO CURICHUMBI JEREZ	10.00	23/08/2019	REUNION DE DIRECTIVOS	20.00	CANCELADO

Figura 4.164: Diseño de Interfaz-Reporte de Cobro Sanciones. Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Pagina Principal | Administracion | Reuniones | Asistencia | Tesoreria | Informes Tesoreria | Reporte Tesoreria Movimientos

Documentos | Movimientos | Reportes

REPORTE GENERAL DE DESCUENTOS

1 de 1 100%

REPORTE GENERAL DE DESCUENTOS

Usuarios	Reunion	Motivo Descuento	Valor Descuento	Deuda Total	Fecha	Ingresado Por
EDUARDO JEREZ	COBROS 3	PRIMER DESCUENTO	2.50	2.50	24/10/2019	OSCAR EDUARDO CURICHUMBI JEF

Figura 4.165: Diseño de Interfaz-Reporte General de Descuentos.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Pagina Principal | Administracion | Reuniones | Asistencia | Tesoreria | Informes Tesoreria | Reporte Tesoreria Movimientos

Documentos | Movimientos | Reportes

REPORTE GENERAL DE EGRESOS SIN FACTURA

1 de 1 100%

REPORTE DE EGRESOS SIN FACTURA

Concepto Egreso	Valor Egreso	Fecha Egreso	Comprado Por	Direccion Compra	Ingresado Por
COMPRA DE ESCOBA	1.00	11/09/2019	MANUEL JEREZ	SALASAKA	OSCAR EDUARDO CURICHUMBI JEREZ
COMPRA DE ALMUERZO	2.50	11/09/2019	MANUEL JEREZ	SALASAKA	OSCAR EDUARDO CURICHUMBI JEREZ

Figura 4.166: Diseño de Interfaz-Reporte General de Egresos Sin Factura.
Elaborado por: Oscar Curichumbi.

GESTION COMUNAL Bienvenido: **admin**
Cerrar sesión

Pagina Principal | Administracion | Reuniones | Asistencia | Tesoreria | Informes Tesoreria | Reporte Tesoreria Movimientos

Documentos | Movimientos | Reportes

REPORTE GENERAL DE EGRESOS CON FACTURA

1 de 1 100%

REPORTE DE EGRESOS CON FACTURA

Concepto Egreso	Valor Egreso	Fecha Egreso	Comprado Por	Numero Factura	Nombre Local	Fecha Compra	Direccion Compra	Ingresado Por
COMPRA DE HOJAS PARA LA IMPRESORA	5.00	11/09/2019	JOSE MARIA	001-001-00229	PAPELERIA JUANES	02/09/2019	SALASAKA	OSCAR EDUARDO CURICHUMBI JEREZ
COMPRA DE MUEBLES	30.00	11/09/2019	MASAQUIZA EDUARDO	001-001-9992393293939	MUEBLERIA JEREZ	01/09/2019	SALASAKA AV129	OSCAR EDUARDO CURICHUMBI JEREZ
COMPRA DE 2 SILLAS PARA LA COMUNIDAD	20.00	11/09/2019	SUSANA MASAQUIZA	001-001-00026361	SILLAS MILAN	01/09/2019	PELILEO	OSCAR EDUARDO CURICHUMBI JEREZ

Figura 4.167: Diseño de Interfaz-Reporte General de Egresos Con Factura.
Elaborado por: Oscar Curichumbi.



Figura 4.168: Diseño de Interfaz-Reporte General de Otros Ingresos.
Elaborado por: Oscar Curichumbi.

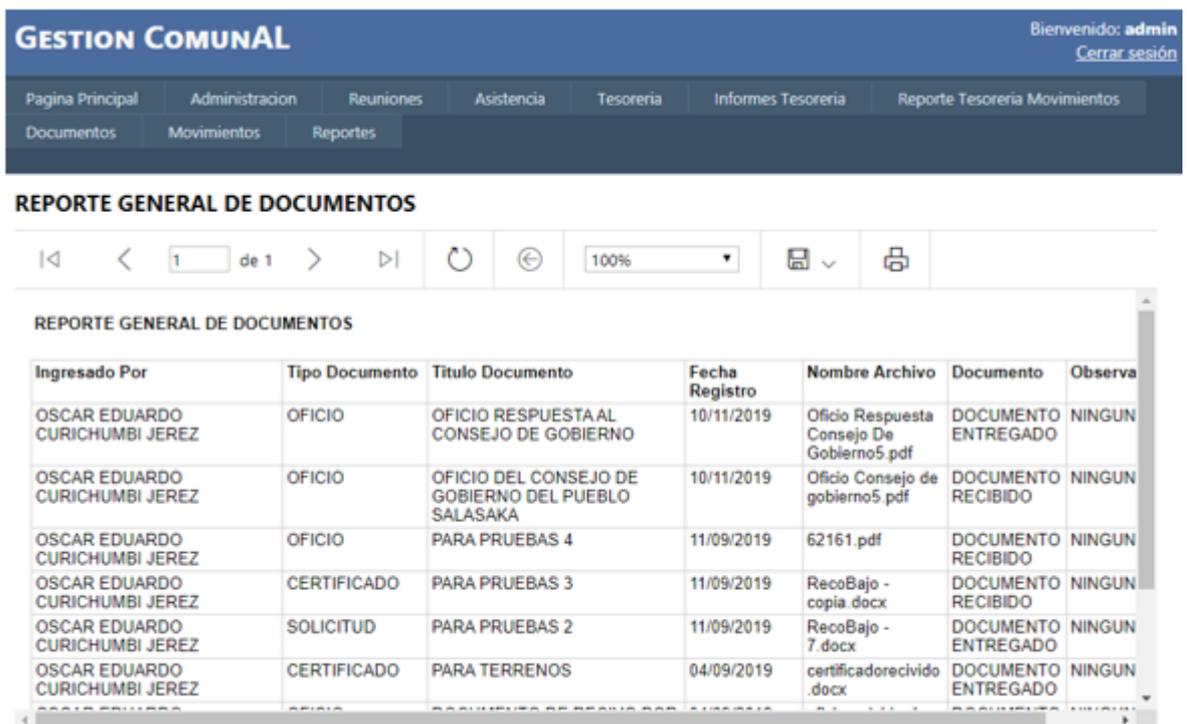


Figura 4.169: Diseño de Interfaz-Reporte General de Documentos.
Elaborado por: Oscar Curichumbi.

4.3.10.6. Codificación

- Método Generar Reporte Asistencias

```
1 protected void btnGenerarInforme_Click(object sender,
    EventArgs e){
```

```

2 reportesDAL dal = new reportesDAL();
3 if (txtIdReunion.Text == ""){
4 lblMensajes.Text = "□Seleccione□Reunion.";
5 lblMensajes.Visible = true;
6 String txtJS = String.Format("<script>alert('{0}')</script
  >", "Seleccione□Reunion.");
7 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
  script", txtJS, false);
8 }
9 else{
10 int ReunionID1 = Convert.ToInt32(txtIdReunion.Text);
11 int cant = dal.SelectconteoInformeGeneral(ReunionID1);
12 if (cant == 0){
13 lblMensajes.Text = "La□reunion□seleccionada□no□tiene□
  asistencias.";
14 lblMensajes.Visible = true;
15 String txtJS = String.Format("<script>alert('{0}')</script
  >", "La□reunion□seleccionada□no□tiene□asistencias.");
16 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
  script", txtJS, false);
  btnGenerarInforme.
  Visible = false;
17 }
18 else{
19 reporteAsistencia.Reset();
20 ReportDataSource repor = new ReportDataSource("tbAsistencia
  ", GetData());
  reporteAsistencia.
  LocalReport.DataSources.Clear();
  reporteAsistencia.LocalReport.DataSources.Add(repor);
  reporteAsistencia.LocalReport.
  ReportEmbeddedResource = "CapaPresentacion.Reportes.
  Asistencias.ReporteAsistenciaReunion7.rdlc";
  reporteAsistencia.LocalReport.
  Refresh();
21 btnGenerarInforme.Visible = false;
22 }
23 }
24 }
25 private DataTable GetData(){
26 DataTable dt = new DataTable();

```

```

27 reportesDAL dal = new reportesDAL();
28 int ReunionID = Convert.ToInt32(txtIdReunion.Text);
29 return dt = dal.SelectAsistenciaGeneral(ReunionID);
30 }

```

- Método Generar Reporte Directivos

```

1 public void cargarDatos() {
2     reporteDirectivos.Reset();
3     ReportDataSource repor = new ReportDataSource("tbDirectivos
    ", GetData());
    reporteDirectivos.LocalReport
    .DataSources.Clear();
    reporteDirectivos
    .LocalReport.DataSources.Add(repor);
    reporteDirectivos.LocalReport.ReportEmbeddedResource = "
    CapaPresentacion.Reportes.Directivos
    .ReporteDirectivoActivos.rdlc";
    reporteDirectivos.LocalReport.Refresh();
4 }
5 private DataTable GetData() {
6     DataTable dt = new DataTable();
7     reportesDAL dal = new reportesDAL();
8     return dt = dal.SelectDirectivosActivos();
9 }

```

- Método Generar Reporte Documentos

```

1 public void cargarDatos() {
2     ReporteGeneralDocumentos.Reset();
3     ReportDataSource repor = new ReportDataSource("tbDocumentos
    ", GetData());
    ReporteGeneralDocumentos
    .LocalReport.DataSources.Clear();
    ReporteGeneralDocumentos.LocalReport.DataSources.Add(
    repor);
    ReporteGeneralDocumentos.LocalReport
    .ReportEmbeddedResource = "CapaPresentacion.Reportes
    .Documentos.ReporteDocumentosGeneral.rdlc";
    ReporteGeneralDocumentos.LocalReport.Refresh();

```

```

4 }
5 private DataTable GetData() {
6     DataTable dt = new DataTable();
7     reportesDAL dal = new reportesDAL();
8     return dt = dal.SelectReporteDocumentos();
9 }

```

- Método Generar Reporte Egresos

```

1 public void cargarDatos() {
2     ReporteConFacturaEgresos.Reset();
3     ReportDataSource repor = new ReportDataSource("tbEgresos",
4         GetData());
5     ReporteConFacturaEgresos.LocalReport.DataSources.Clear();
6     ReporteConFacturaEgresos.LocalReport.DataSources.Add(
7         repor);
8     ReporteConFacturaEgresos.LocalReport.ReportEmbeddedResource = "CapaPresentacion.Reportes.
9         EgresosIngresos.ReporteEgresosConFactura.rdlc";
10    ReporteConFacturaEgresos.LocalReport.Refresh();
11 }
12 private DataTable GetData() {
13     DataTable dt = new DataTable();
14     reportesDAL dal = new reportesDAL();
15     return dt = dal.SelectInformeConFactura();
16 }

```

- Método Generar Reporte Usuarios

```

1 public void cargarDatos() {
2     reportUsuarioA.Reset();
3     ReportDataSource repor = new ReportDataSource("tbUsuarios",
4         GetData());
5     reportUsuarioA.LocalReport.DataSources.Clear();
6     reportUsuarioA.LocalReport.DataSources.Add(repor);
7     reportUsuarioA.LocalReport.Refresh();

```

```

        ReportEmbeddedResource = "CapaPresentacion.Reportes.
        ReportesUsuarios.ReporteUsuarioGeneral.rdlc";
        reportUsuarioA.LocalReport.Refresh();
5 }
6 private DataTable GetData(){
7 DataTable dt = new DataTable();
8 reportesDAL dal = new reportesDAL();
9 return dt = dal.SelectUsuariosGeneral();
10 }

```

- Método Generar Reporte Reuniones

```

1 public void cargarDatos(){
2 reporteReuniones.Reset();
3 ReportDataSource repor = new ReportDataSource("tbReunion",
        GetData());
        reporteReuniones.LocalReport.
        DataSources.Clear();
        reporteReuniones.
        LocalReport.DataSources.Add(repor);
        reporteReuniones.LocalReport.ReportEmbeddedResource = "
        CapaPresentacion.Reportes.Reuniones.
        ReporteReunionFinalizado8.rdlc";
        reporteReuniones.LocalReport.Refresh();
4 }
5 private DataTable GetData(){
6 DataTable dt = new DataTable();
7 reportesDAL dal = new reportesDAL();
8 return dt = dal.SelectReunionFinalizada();
9 }

```

- Método Generar Reporte Tesorería

```

1 protected void btnGenerarInforme_Click(object sender,
        EventArgs e){
2 CobroDAL dal = new CobroDAL();
3 if (txtIDreunion.Text == ""){
4 lblMensajes.Text = "□ Seleccione □ Reunion.";

```

```

5 lblMensajes.Visible = true;
6 String txtJS = String.Format("<script>alert('{0}');</script
  >", "Seleccione Reunion.");
7 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
  script", txtJS, false);
8 }
9 else{
10 int ReunionID1 = Convert.ToInt32(txtIDreunion.Text);
11 int cant = dal.SelectconteoInformeCobroColaboracionReunion(
  ReunionID1);
12 if (cant == 0){
13 lblMensajes.Text = "La reunion seleccionada no tiene ningun
  cobro.";
14 lblMensajes.Visible = true;
15 String txtJS = String.Format("<script>alert('{0}');</script
  >", "La reunion seleccionada no tiene ningun cobro.");
16 ScriptManager.RegisterClientScriptBlock(Page, GetType(), "
  script", txtJS, false);
  btnBuscarReunion.Focus();
17 btnGenerarInforme.Visible = false;
18 }
19 else{
20 ReporteColaboracionCobroReunion.Reset();
21 ReportDataSource repor = new ReportDataSource("
  tbCobroColaboracion", GetData());
  ReporteColaboracionCobroReunion.LocalReport.DataSources.
  Clear();
  ReporteColaboracionCobroReunion.LocalReport.DataSources.
  Add(repor);
  ReporteColaboracionCobroReunion.LocalReport.
  ReportEmbeddedResource = "CapaPresentacion.Reportes.
  Tesoreria.ReporteCobroColaboracionReunion.rdlc";
  ReporteColaboracionCobroReunion.
  LocalReport.Refresh();
22 btnGenerarInforme.Visible = false;
23 }
24 }
25 }
26 private DataTable GetData(){
27 DataTable dt = new DataTable();

```

```

28 reportesDAL dal = new reportesDAL();
29 int ReunionID = Convert.ToInt32(txtIDreunion.Text);
30 return dt = dal.SelectCobroColaboracionReunion(ReunionID);
31 }

```

4.3.10.7. Pruebas de Aceptación

A continuación, se muestra las pruebas de aceptación de la Sexta Iteración:

CASO DE PRUEBA	
Código: 13	Nº Historia de Usuario: 11
Historia de Usuario: Reportes del presidente o vicepresidente.	
Condiciones de Ejecución: El Presidente o Vicepresidente podrán visualizar sus reportes dependiendo de los permisos que tengan otorgados.	
Entrada/Pasos de Ejecución: Seleccionar la pestaña Reportes, seleccionar los reportes que deseen visualizar o imprimir. Los reportes deben informar al presidente o vicepresidente sobre sus actividades realizadas en la comunidad. Estos reportes podrán ser imprimidos cuando sean necesarios.	
Resultado Esperado: Los reportes serán generados inmediatamente, también podrán ser imprimidos o descargados con gran facilidad.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.139: Caso de Prueba: Reportes del presidente o vicepresidente.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 14	Nº Historia de Usuario: 12
Historia de Usuario: Reportes del Secretario(a).	
Condiciones de Ejecución: El Secretario(a) podrá visualizar sus reportes dependiendo de los permisos que tengan otorgados.	
Entrada/Pasos de Ejecución: Seleccionar la pestaña Reportes, seleccionar los reportes que deseen visualizar o imprimir. Los reportes deben informar al Secretario(a) sobre sus actividades realizadas en la comunidad. Estos reportes podrán ser imprimidos cuando sean necesarios.	
Resultado Esperado: Los reportes serán generados inmediatamente, también podrán ser imprimidos o descargados con gran facilidad.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.140: Caso de Prueba: Reportes del Secretario(a).

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 15	Nº Historia de Usuario: 13
Historia de Usuario: Reportes de Tesorería.	
Condiciones de Ejecución: El Tesorero(a) podrá visualizar sus reportes dependiendo de los permisos que tengan otorgados.	
Entrada/Pasos de Ejecución: Seleccionar la pestaña Reportes, seleccionar los reportes que deseen visualizar o imprimir. Los reportes deben informar al Tesorero(a) sobre sus actividades realizadas en la comunidad. Estos reportes podrán ser imprimidos cuando sean necesarios.	
Resultado Esperado: Los reportes serán generados inmediatamente, también podrán ser imprimidos o descargados con gran facilidad.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.141: Caso de Prueba: Reportes de Tesorería.

Elaborado por: Oscar Curichumbi.

CASO DE PRUEBA	
Código: 16	Nº Historia de Usuario: 14
Historia de Usuario: Reportes de los Vocales.	
Condiciones de Ejecución: Los Vocales podrán visualizar sus reportes dependiendo de los permisos que tengan otorgados.	
Entrada/Pasos de Ejecución: Seleccionar la pestaña Reportes, seleccionar los reportes que deseen visualizar o imprimir. Los reportes deben informar a los Vocales sobre sus actividades realizadas en la comunidad. Estos reportes podrán ser imprimidos cuando sean necesarios.	
Resultado Esperado: Los reportes serán generados inmediatamente, también podrán ser imprimidos o descargados con gran facilidad.	
Evaluación de la Prueba: La prueba finalizó correctamente.	

Tabla 4.142: Caso de Prueba: Reportes de los Vocales.

Elaborado por: Oscar Curichumbi.

4.3.10.8. Resultados de la Sexta Iteración

Como resultado de entrega de la sexta iteración que consta de los módulos: Reportes del presidente o vicepresidente, Reportes del Secretario(a), Reportes de Tesorería y Reportes de los Vocales, el cliente quedó satisfecho con las funcionalidades de los módulos que se desarrollaron. Los reportes serán necesarios para los directivos con la finalidad de analizar la información de la comunidad, para tomar decisiones y dar informes a los usuarios de la comunidad, sobre las actividades que realizaron en su periodo. La sexta iteración se ha concluido sin ningún problema manifestado por el cliente.

CAPÍTULO 5

Conclusiones y Recomendaciones

5.1. Conclusiones

- Las técnicas que aporta la metodología Extreme Programming (XP), fueron de vital importancia para el desarrollo del sistema web, estas técnicas ayudaron a planificar y cumplir en forma ordenada cada una de las tareas a desarrollar por el programador, cumpliendo de este modo la entrega del producto conforme lo planificado. Las pruebas que se realizaron para la verificación del producto, tuvieron como respuesta por parte del cliente una afirmación positiva, concluyendo que el sistema cumple con las necesidades requeridas por el cliente.
- Al implementar el sistema web en la comunidad de Sanjaloma Bajo se automatizó los procesos administrativos y financieros, evitando registrar de modo manual (cuadernos) información valiosa, esta será registrada en una base de datos, mejorando la búsqueda de información que requieren los usuarios en cualquier momento, omitiendo búsquedas visuales o manuales que requiere mucho tiempo. Los Directivos con la ayuda del sistema pueden explicar con detalle las reuniones, los registros de asistencia, las recaudaciones, los egresos y los documentos emitidos y recibidos a los usuarios de la comunidad.
- El sistema genera los reportes que visualizan la información que los Directivos necesitan para realizar el análisis correspondiente para la toma de decisiones. Con los reportes la comunidad verifica su estado financiero, para indicar si cuenta con capital positivo o negativo, de igual manera si tiene o no valores monetarios por cobrar, esto mejora el proceso financiero de la comunidad.
- Finalmente se puede concluir que el proyecto alcanzó todos los objetivos propuestos inicialmente, desarrollando un sistema que automatiza los procesos de la comunidad, cumpliendo con las necesidades que requerían tanto Directivos como usuarios.

5.2. Recomendaciones

Para que el sistema web funcione adecuadamente se recomienda:

- Brindar capacitación a los Directivos sobre el manejo del sistema web.
- Establecer políticas sobre el respaldo de la información en caso de fallar el software o el hardware, de igual manera en casos de mala manipulación del sistema.
- Realizar el mantenimiento del sistema por lo menos cada seis meses.
- Los Directivos deben cambiar de contraseña cada mes, para evitar que personas no autorizadas accedan al sistema web.
- Solo los Directivos autorizados y elegidos por los usuarios de la comunidad podrán manipularán cada proceso del sistema.
- Ejecutar el sistema web en el navegador Chrome, porque carga las páginas de manera rápida y también tienen la agilidad a la hora de abrir/cerrar pestañas.

Bibliografía

- [1] L. del Rosario Guachamín Chiluisa, “La gestión de archivos incide en la imagen institucional del Área administrativa de la unidad metropolitana de salud centro de la ciudad de Quito,” *mathesis*, UNIVERSIDAD TÉCNICA DE AMBATO, 2014.
- [2] I. M. L. Álvarez Freire, “Sistema de gestión de calidad y la satisfacción de los usuarios del gobierno autónomo descentralizado municipal del cantón saquisilí,” *mathesis*, UNIVERSIDAD TÉCNICA DE AMBATO, 2012.
- [3] E. J. F. Vargas, “Sistema informático basado en software libre para la gestión documental en la facultad de ciencia e ingeniería en alimentos,” *mathesis*, UNIVERSIDAD TÉCNICA DE AMBATO, Feb. 2014.
- [4] J. F. M. BASTIDAS, “Implementar un sistema de gestión de solicitudes TI para el departamento de sistemas de una compañía dedicada a la fabricación y comercialización de productos para la construcción y la industria,” *mathesis*, ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL, 2015.
- [5] M. Ángel Remache Chicaiza, “Sistema automatizado de gestión de documentos en el Área administrativa en la unidad educativa Luis A. Martínez de la ciudad de Ambato, provincia del Tungurahua,” *mathesis*, UNIVERSIDAD TÉCNICA DE AMBATO, 2014.
- [6] L. Martínez, “La observación y el diario de campo en la definición de un tema de investigación,” *Revista perfiles libertadores*, vol. 4, no. 80, pp. 73–80, 2007.
- [7] C. Camisón, S. Cruz, and T. González, *Gestión de la calidad: conceptos, enfoques, modelos y sistemas*. Pearson, 2007.
- [8] I. N. de Estadística y Censos, “Manual de archivo general y gestión documental.” Disponible: https://www.ecuadorencifras.gob.ec/LOTAIP/2015/DIJU/enero/LA3ENE_DIJU_ArchivoO
- [9] L. Fossi, L. Castro, W. Guerrero, and L. Vera, “Funciones administrativas y la participación comunitaria,” *Orbis. Revista Científica Ciencias Humanas*, vol. 9, no. 25, pp. 47–63, 2013.

- [10] C. T. Horngren, G. L. Sundem, and W. O. Stratton, *Contabilidad administrativa*. Pearson Educación, 2007.
- [11] M. Á. Mallar, “La gestión por procesos: un enfoque de gestión eficiente,” *Revista Científica "Visión de Futuro"*, vol. 13, no. 1, 2010.
- [12] M. E. Caldas, R. Carrión, and A. J. Heras, *Gestión administrativa (Empresa e iniciativa emprendedora)*. Editex, 2017.
- [13] C. I. Rivas, V. P. Corona, J. F. Gutiérrez, and L. Hernández, “Metodologías actuales de desarrollo de software,” *Revista de Tecnología e Innovación*, vol. 2, no. 5, pp. 980–986, 2015.
- [14] A. Ramos, *Aplicaciones Web (Novedad 2011)*. Editorial Paraninfo, 2011.
- [15] P. Day, “Asp. net multitenant applicationsfor newbies,” 2016.
- [16] G. J. L. C. . Y. V. L. ALTAMIRANO, “Sistema informático de inventario y facturación de mercadería con entorno web en la imprenta y gráficas chone,” mathesis, ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ, Mar. 2013.
- [17] J. L. Q. Medina, “Sistema informático que permita la administración contable de los activos fijos que posee la pucese,” mathesis, PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE ESMERALDAS, July 2016.
- [18] I. M. Zabala Villarraga, “Comparativa oracle y sql server,” *Revista de Tecnología e Innovación*, 2018.
- [19] D. C. C. C. . C. X. Z. OVIEDO, “Desarrollo de un sistema web para el manejo de inventario de multibodegas, gestión contable, y creación de un repositorio digital de informes técnicos, utilizando dspace, tecnología primefaces y herramientas open source, para la empresa crosstronik cia. ltda.,” mathesis, ESCUELA POLITÉCNICA DE EJERCITO, June 2013.
- [20] L. P. E. CASAS, “Guía para la elección del lenguaje de desarrollo para proyectos,” mathesis, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, Oct. 2011.
- [21] A. T. E. B. . T. S. N. IVÁN, “Diseño de un sistema de mantenimiento preventivo y correctivo con el uso de interfaces gráficas con software,” mathesis, ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, 2013.

- [22] D. R. T. CHICAIZA, “Análisis, desarrollo e implementación de una aplicación web orientada para el registro de Órdenes de compra, clientes y productos para la empresa ferreteria tobar nolivos cia. ltda,” mathesis, PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR, 2014.
- [23] MundoInsider, “Así es visual studio 2017, el ide perfecto para crear apps en windows, ios, android y la nube.” Disponible: <https://www.mundoinsider.com/121003/asi-visual-studio-2017-ide-perfecto-crear-apps-windows-ios-android-la-nube/>, Mar. 2017.
- [24] NetBeans, “Netbeans ide.” Disponible: <https://netbeans.org/features/index.html>, Aug. 2019.
- [25] Eclipse, “Eclipse ide.” Disponible: <https://www.eclipse.org/ide/>, Aug. 2019.
- [26] A. N. Cadavid, J. D. F. Martínez, and J. M. Vélez, “Revisión de metodologías ágiles para el desarrollo de software,” *Prospectiva*, vol. 11, no. 2, pp. 30–39, 2013.
- [27] J. G. . M. V. TIPÁN COYAGO, DARWIN MAURICIO, “Estudio, selección y aplicación de un modelo de desarrollo de software y su acoplamiento en la cooperativa de ahorro y crédito policia nacional ltda., demostración de su validez en un módulo básico,” mathesis, ESPE UNIVERSIDAD DE LAS FUERZAS ARMADAS, SANGOLQUÍ, 2017.
- [28] J. G. A. CAMACHO, “Estudio comparativo de metodologías de desarrollo de software,” mathesis, UNIVERSIDAD DE NARIÑO, SAN JUAN DE PASTO, 2014.
- [29] M. E. G. . N. N. P. R. SINTYA MILENA MELÉNDEZ VALLADAREZ, “Sistema web de evaluación al desempeño docente unanmanagua, empleando la metodología Ágil programación extrema, en el ii semestre del 2015,” mathesis, UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA, Jan. 2016.

Anexos y Apéndices

Anexo A

Manual de usuario

Manual de Usuario El sistema web podrá ser administrado por los Directivos de la comunidad de una forma rápida y sencilla. A continuación, se detalla los procesos que permite realizar el sistema web:

1. Abrir y Cerrar Sesión.
2. Administrar Usuarios (Ingresar, Actualizar, Eliminar, Buscar).
3. Administrar Directivos (Ingresar, Actualizar, Buscar).
4. Administrar Reuniones (Ingresar, Actualizar, Buscar).
5. Administrar Control de Asistencias (Ingresar, Actualizar).
6. Administrar Tesorería (Ingresar, Actualizar, Buscar).
7. Administrar Egresos (Ingresar).
8. Administrar Documentos (Ingresar).
9. Descargar Documentos.
10. Reportes.

A.1. Abrir y Cerrar Sesión

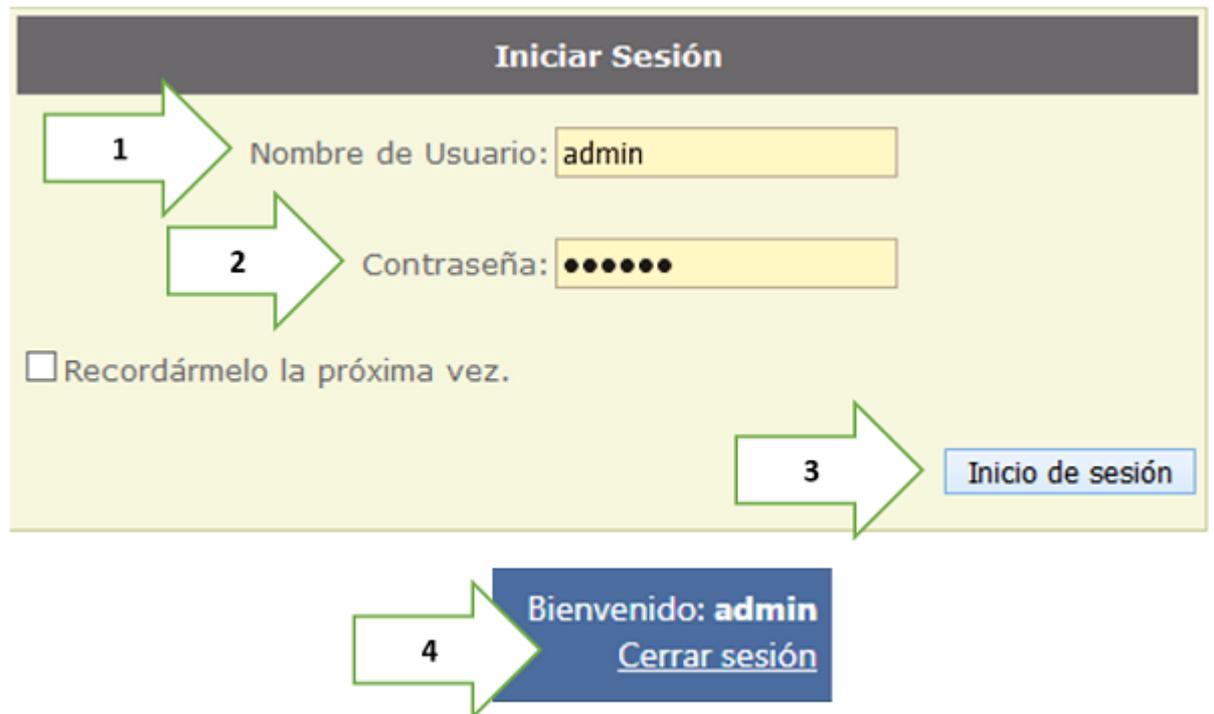


Figura A.1: Manual Usuario Abrir y Cerrar Sesión.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Ingresar y salir del sistema:

1. Primero debe ingresar el Nombre de Y Usuario.
2. Segundo ingresar su contraseña.
3. Dar clic sobre el botón Inicio de sesión, se ingresará al sistema web si el nombre de usuario y contraseña son correctos, caso contrario se presentará un mensaje de error.
4. Dar clic sobre el botón Cerrar sesión para salir del sistema web.

A.2. Administrar Usuarios

The screenshot shows a web form titled "Crear Usuarios". At the top, there is a "Cedula:" field containing the number "1805198478". To its right is a search icon. Below this is a large form containing several fields: "Nombres:", "Apellidos:", "E_mail:", "Celular:", "Telefono:", "Sexo:" (with a dropdown menu showing "SELECCIONE SEXO"), "Actividad:" (with a dropdown menu showing "SELECCIONE ACTIVIDAD"), and "Fecha:" containing "01-08-2019". To the right of these fields are two buttons: "Agregar" and "Cancelar". Five green arrows with numbers 1 through 5 point to the following elements: 1. The "Cedula:" field; 2. The search icon; 3. The "Nombres:" field; 4. The "Agregar" button; 5. The "Cancelar" button.

Figura A.2: Manual Usuario Administrar Usuarios.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Agregar usuarios:

1. Primero debe ingresar el Numero de cedula para verificar si es válido o no.
2. Este botón ayudara a buscar usuarios, para verificar si se ingresó anteriormente o no los datos del nuevo usuario.
3. Se debe ingresar la información de Nombres, Apellidos, E_mail, Celular, Teléfono, Tipo de Sexo, Actividad del usuario si está activo y por último la Fecha de Ingreso.
4. Dar clic sobre el botón Agregar para registrar la información en la base de datos.
5. Dar clic sobre el botón Cancelar para volver a ingresar la información en los campos correspondientes.

A.3. Administrar Directivos

Ingresar Nuevo Directivo

1 2

Cedula:

4

3

5

6

Persona: OSCAR EDUARDO CURICHUMBI JEREZ

Nombre de Usuario: admin

Contraseña:

Fecha_Ingreso: 05-12-2019

Cargo: SELECCIONE CARGO ▾

ID	Nombres y Apellido del Directivo	Nombre Usuario	Fecha de Ingreso	Cargo	Estado
1	OSCAR EDUARDO CURICHUMBI JEREZ	ADMIN	30/04/2019	ADMINISTRADOR	ACTIVO
2	JORGE MANUEL MASAQUIZA MASAQUIZA	oscar	01/07/2019	PRESIDENTE	ACTIVO
3	JORGE MANUEL MASAQUIZA MASAQUIZA	jorge	19/07/2019	VOCAL	FINALIZADO

Figura A.3: Manual Usuario Administrar Directivos.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Agregar usuarios:

1. Primero debe ingresar el Numero de cedula y dar clic en el botón Buscar, para buscar la información del usuario que será el nuevo Directivo.
2. Este botón cancelar sirve para buscar nuevos usuarios.
3. Se debe ingresar la información de Nombre de Usuario, Contraseña, Fecha de Ingreso y el respectivo Cargo.
4. Dar clic sobre el botón Agregar para registrar la información en la base de datos.
5. Dar clic sobre el botón Cancelar para volver a ingresar la información en los campos correspondientes.
6. Se visualizará la información de los nuevo Directivos.

A.4. Administrar Reuniones

Generar Reunion

The form contains the following fields and buttons:

- Tipo de Reunion:
- Motivo de la Reunion:
- Fecha Reunion:
- Buttons: Generar, Cancelar

TIPO DE REUNION	MOTIVO REUNION	FECHA REUNION	ESTADO
Reunion Ordinaria	COBROS 3	24/10/2019	ACTIVO
Reunion Extraordinaria	COBROS 2	24/10/2019	ACTIVO
Reunion Ordinaria	COBROS	24/10/2019	ACTIVO
Reunion Ordinaria	POR PRUEBA DIRECTIVO2	20/09/2019	ACTIVO
Reunion Ordinaria	POR PRUEBA DIRECTIVO4	07/08/2019	ACTIVO

Figura A.4: Manual Usuario Administrar Reuniones.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Agregar Reuniones:

1. Primero se debe ingresar la información correspondiente para la nueva reunión.
2. Dar clic en el botón Generar para registra la información de la nueva reunión.
3. Dar clic sobre el botón Cancelar para volver a ingresar la información en los campos correspondientes.
4. Se visualizará la información de la nueva reunión.

A.5. Administrar Control de Asistencias

CONTROL DE ASISTENCIA

1

2

6

Reunion: REUNION DE DIRECTIVOS

Valor Multa Atraso: 0.00

Valor Multa Ausencia: 0.00

Fecha: 05-12-2019

Listar Usuarios

Listar Directivos

Nuevo

4

3

!!! Antes de pasar la lista Verifique si el Usuario esta PRESENTE, ATRASADO O AUSENTE !!!

5

ID	NOMBRE COMPLETO	ESTADO	ASISTENCIA
1	OSCAR EDUARDO CURICHUMBI JEREZ	<input type="radio"/> PRESENTE <input type="radio"/> ATRASADO <input type="radio"/> AUSENTE	PASAR LISTA
2	JORGE MANUEL MASAQUIZA MASAQUIZA	<input type="radio"/> PRESENTE <input type="radio"/> ATRASADO <input type="radio"/> AUSENTE	PASAR LISTA

Figura A.5: Manual Usuario Administrar Control de Asistencias.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Controla Asistencias:

1. Dar clic en el botón Listar Usuarios o Listar Directivos para visualizar la información de todos los Usuarios o de Todos los Directivo que están Activos.
2. Segundo debemos ingresar la información que corresponde al control de asistencia.
3. Dar clic sobre el botón Pasar Lista para registra la información de la asistencia en la base de datos.
4. Dar clic en el botón Nuevo para volver a ingresar la información en los campos correspondientes.
5. Se visualizará la información de todo los Usuarios o Directivos.
6. La reunión debe estar Activa para visualizarla y poder realizar el proceso de control de asistencia.

A.6. Administrar Tesorería

The image shows a web interface for recording collaborations. The top part is a form titled "REGISTRAR COLABORACIONES" with fields for "Reunion:" (REUNION DE DIRECTIVOS), "Motivo Colaboracion:", "Valor Colaboracion:" (0.00), and "Fecha:" (05-12-2019). There are buttons for "Listar Usuarios" and "Nuevo". A table below lists three users with "Confirmar" buttons. Numbered callouts 1-6 indicate the sequence of actions: 1 (Listar Usuarios), 2 (form fields), 3 (Confirmar), 4 (Nuevo), 5 (table), and 6 (Reunion dropdown).

ID	NOMBRE COMPLETO	COLABORACION
1	OSCAR EDUARDO CURICHUMBI JEREZ	Confirmar
2	JORGE MANUEL MASAQUIZA MASAQUIZA	Confirmar
3	EDUARDO JEREZ	Confirmar

Figura A.6: Manual Usuario Tesorería Registros.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Registrar Colaboraciones y Sanciones:

1. Dar clic en el botón Listar Usuarios para visualizar la información de todos los Usuarios que están Activos.
2. Segundo debemos ingresar la información que corresponde para registrar las colaboraciones.
3. Dar clic sobre el botón Confirmar para registra la información de las colaboraciones en la base de datos.
4. Dar clic en el botón Nuevo para volver a ingresar la información en los campos correspondientes.
5. Se visualizará la información de todo los Usuarios o Directivos.
6. La reunión debe estar Activa para visualizarla y poder realizar el proceso de Tesorería.

COBRO MULTAS POR USUARIO

USUARIO:	<input type="text" value="JORGE MANUEL MASAQUIZA MASAQUIZA"/>	1 ↓ <input type="button" value="Buscar Usuario"/>
NOMBRE DE QUIEN CANCELA:	<input type="text"/>	<input type="button" value="Buscar Multas"/>
MONTO PAGO:	<input type="text" value="0.00"/>	2 →
FECHA:	<input type="text" value="05-12-2019"/>	5 →
		<input type="button" value="Nuevo Cobro"/>

REUNION	ESTADO	SALDO DEUDA	COBRAR
COBROS 3	ATRASADO	1,00	4 ↓ <input type="button" value="Cobrar"/>

6 →

Figura A.7: Manual Usuario Tesorería Cobros.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Cobrar las Multas, Colaboraciones y Sanciones:

1. Dar clic en el botón Buscar Usuarios para visualizar la información de todos los Usuarios que están Activos.
2. Damos clic en el botón Buscar Multas para verificar si el usuario seleccionado tiene o no deudas.
3. Tercero debemos ingresar la información que corresponde para registrar los cobros.
4. Dar clic sobre el botón Cobrar para registra la información de los cobros en la base de datos.
5. Dar clic en el botón Nuevo Cobro para volver a ingresar la información en los campos correspondientes.
6. Se visualizará la información de todas las deudas.

DESCUENTO DE MULTAS

1

USUARIO: JORGE MANUEL MASAQUIZA MASAQUIZA Buscar Usuario

VALOR TOTAL DEUDA: 1.00 2 Calcular Multa Total

REUNION: 3 Buscar Reunion

4 VALOR DESCUENTO: 0.00

MOTIVO DESCUENTO: 5 Generar Descuento

FECHA: 05-12-2019 6 Nuevo Registro

7

REUNION	SALDO DEUDA	ESTADO
COBROS 3	1,00	PENDIENTE

Figura A.8: Manual Usuario Tesorería Descuentos.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Generar Descuentos:

1. Dar clic en el botón Buscar Usuarios para visualizar la información de todos los Usuarios que están Activos.
2. Damos clic en el botón Calcular Multa Total para verificar el valor total de deuda sobre multas que el usuario seleccionado tiene.
3. Damos clic en el botón Buscar Reunión para seleccionar la reunión en donde se ha aprobado el descuento.
4. Cuarto debemos ingresar la información que corresponde para registrar el descuento.
5. Dar clic sobre el botón Generar Descuento para registra la información del descuento en la base de datos.
6. Dar clic en el botón Nuevo Registro para volver a ingresar la información en los campos correspondientes.
7. Se visualizará la información del descuento registrado.

A.7. Administrar Egresos

2 ↓

EGRESOS CON FACTURA

Concepto Del Egreso: Ingresar Egreso

Valor Egreso: Nuevo

Fecha Egreso: Cancelar

Comprado Por:

Numero De Factura: - -

Nombre Del Local:

Fecha Compra:

Direccion Compra:

4 →

1 →

CONCEPTO DEL EGRESO	VALOR DEL EGRESO	FECHA EGRESO	COMPRADO POR	NUMERO DE FACTURA	LOCAL	FECHA COMPRA	DIRECCION
COMPRA DE HOJAS PARA LA IMPRESORA	5.00	11/09/2019	JOSE MARIA	001-001-00229	PAPELERIA JUANES	02/09/2019	SALASAKA
COMPRA DE MUEBLES	30.00	11/09/2019	MASAQUIZA EDUARDO	001-001-9992393293939	MUEBLERIA JEREZ	01/09/2019	SALASAKA AV129
COMPRA DE 2 SILLAS PARA LA COMUNIDAD	20.00	11/09/2019	SUSANA MASAQUIZA	001-001-00026361	SILLAS MILAN	01/09/2019	PELILEO

3 →

Figura A.9: Manual Usuario Egresos.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Generar Descuentos:

1. Primero se debe ingresar toda la información del egreso.
2. Damos clic en el botón Ingresar Egreso para registrar la información en la base de datos.
3. Se visualizará la información del Egreso registrado.
4. Dar clic en el botón Nuevo o cancelar para volver a ingresar la información en los campos correspondientes.

OTROS INGRESOS

Concepto Del Ingreso:

Valor Ingreso:

Fecha Ingreso:

Ingresar Ingreso

Nuevo

Cancelar

CONCEPTO DEL INGRESO	VALOR DEL INGRESO	FECHA INGRESO
DONACION DE LA JUNTA PARROQUIAL	100.00	12/09/2019

Figura A.10: Manual Usuarios Otros Ingresos.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Ingresar Otros Ingresos:

1. Primero se debe ingresar toda la información del ingreso.
2. Damos clic en el botón Ingresar Ingreso para registrar la información en la base de datos.
3. Se visualizará la información del Ingreso registrado.
4. Dar clic en el botón Nuevo o cancelar para volver a ingresar la información en los campos correspondientes.

A.8. Administrar Documentos

DOCUMENTOS QUE INGRESAN A LA COMUNIDAD

Tipo De Documento:

Título Del Documento:

Fecha Registro:

Documento: Ningún archivo seleccionado

Observaciones:

Subir Documento

Nuevo

Cancelar

Figura A.11: Manual Usuarios Registro de Documentos.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Ingresar Documentos:

1. Primero se debe ingresar toda la información de Documento.

2. Dar clic en el botón Seleccionar Archivo para seleccionar el documento para subir al servidor.
3. Damos clic en el botón Subir Documento para registrar la información en la base de datos.
4. Dar clic en el botón Nuevo o cancelar para volver a ingresar la información en los campos correspondientes.

A.9. Descargar Documentos



DESCARGAR DOCUMENTOS RECIBIDOS			
TIPO DE DOCUMENTO	TITULO DOCUMENTO	FECHA	DESCARGAR
OFICIO	DOCUMENTO DE RECIVO POR PARTE DEL MUNICIPIO	04/09/2019	oficiorecivido.docx
CERTIFICADO	PARA PRUEBAS 3	11/09/2019	RecoBajo - copla.docx
OFICIO	PARA PRUEBAS 4	11/09/2019	62161.pdf
OFICIO	OFICIO DEL CONSEJO DE GOBIERNO DEL PUEBLO SALASAKA	10/11/2019	Oficio Consejo de gobierno5.pdf

Figura A.12: Manual Usuarios Descargas de Documentos.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Ingresar Documentos:

1. Para descargar documento debemos dar clic en la pestaña descarga documentos e inmediatamente aparecerá todos los documentos ingresados al servidor.
2. Dar clic en el botón Descargar para bajar el documento seleccionado del servidor.

1. Para Generar el Reporte se debe seleccionar la Fecha de Inicio y la Fecha Fin para realizar la consulta entre fechas.
2. Dar clic en el botón Generar Reporte para realizar el proceso de reporte.
3. Se visualizará el reporte y se podrá imprimir sin ninguna dificultad.
4. Dar clic en el botón Nuevo Consulta para volver a realizar el reporte entre fechas.

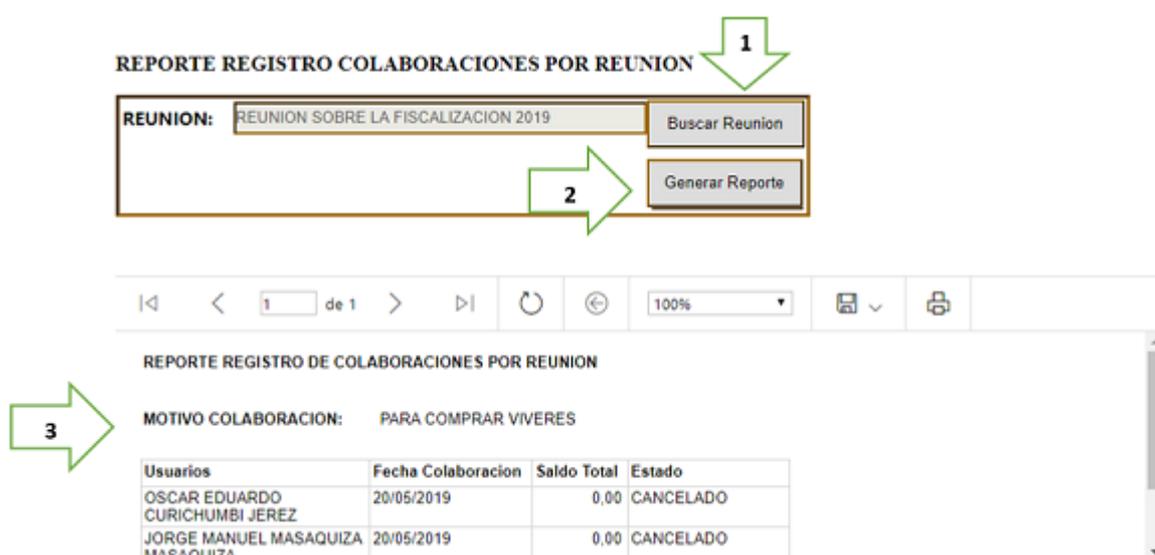


Figura A.14: Manual Usuarios Reporte General.
Elaborado por: Oscar Curichumbi.

Detalle sobre el proceso que se realiza para Reportes:

1. Para Generar el Reporte se debe Buscar la reunión del cual se desea el reporte.
2. Dar clic en el botón Generar Reporte para realizar el proceso de reporte.
3. Se visualizará el reporte y se podrá imprimir sin ninguna dificultad.

Anexo B

Archivero Sanjaloma Bajo

El archivero de la comunidad contiene los cuadernos en los cuales están registrados el control de asistencia y los cobros, de igual manera contiene los documentos.

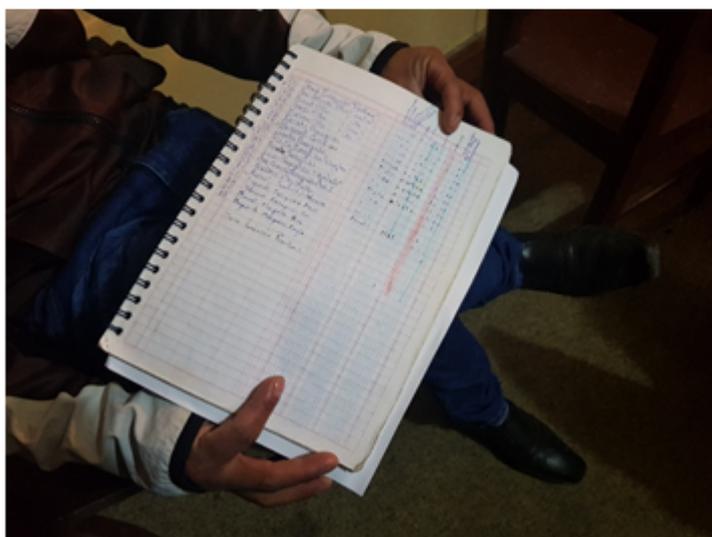
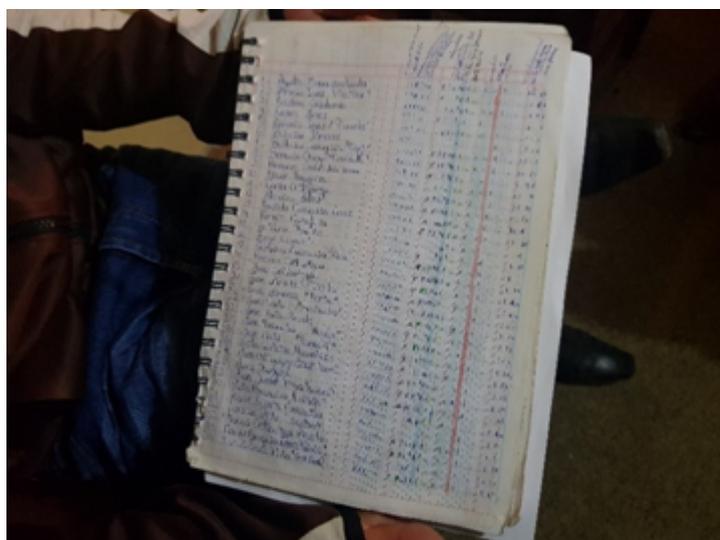


Figura B.1: Cuaderno de Control de Asistencia y Cobros.
Elaborado por: Oscar Curichumbi.



Figura B.2: Archivero de Documentos.
Elaborado por: Oscar Curichumbi.