

AFICOP- APLICACIÓN WEB PARA LOS AFILIADOS DE LAS COOPERATIVAS

**PAOLA ANDREA PANTOJA GALINDRES
CRISTY MARILYN RIOS YELA**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2013**

AFICOP- APLICACIÓN WEB PARA LOS AFILIADOS DE LAS COOPERATIVAS

**PAOLA ANDREA PANTOJA GALINDRES
CRISTY MARILYN RIOS YELA**

**Trabajo de Grado presentado como requisito parcial
para optar al título de Ingenieras de Sistemas**

**Ing. Nelson Jaramillo Enríquez
Director.**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2013**

NOTA DE RESPONSABILIDAD

“Las ideas y conclusiones aportadas en la tesis de grado son responsabilidad exclusiva de los autores”

Artículo 1º del Acuerdo N° 324 de octubre 11 de 1966 emanado del Honorable Consejo Directivo de Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”

Artículo 13, Acuerdo N° 005 de 2010 emanado por el Honorable Consejo Académico.

NOTA DE ACEPTACIÓN

Firma del Jurado

Firma del Jurado

Firma del director del proyecto

San Juan de Pasto, Junio 2013

RESUMEN

En este documento se presenta el desarrollo del trabajo de grado; análisis y diseño de **AFICOP- APLICACIÓN WEB PARA LOS AFILIADOS DE LAS COOPERATIVAS.**

Este proyecto es una aplicación web desarrollada para apoyar la gestión de las cooperativas en la atención de sus afiliados de una forma directa y efectiva; se encuentra dividido en seis módulos:

El módulo de afiliados que permite el registro, consulta, modificación y eliminación de los datos del afiliado, actividades que solo le serán permitidas al usuario tipo administrador.

El módulo de Ingresos Aportes y Pagos que permite el registro, consulta, modificación y eliminación de los datos de los ingresos por cualquier concepto de la cooperativa; datos que serán manejados solo por el tipo de usuario administrador. También permite el registro de los aportes que el afiliado realizara en la cooperativa; este registro como la modificación y eliminación de estos datos solo podrá ser realizada por el tipo de usuario administrador, la consulta de los aportes del afiliado podrá se realizada por el tipo de usuario administrador y afiliado. Y por ultimo en este módulo se manejaran el registro del pago de las cuotas de los prestamos realizados por los afiliados, su modificación y eliminación serán tareas que solo podrá ser realizadas por el usuario administrador.

El módulo de Egresos y Préstamos que permite el registro, consulta, modificación y eliminación de los datos de los egresos por cualquier concepto de la cooperativa; el registro y eliminación de los prestamos realizados por los afiliados son datos que solo podrán ser manejados por el tipo de usuario administrador y el usuario tipo afiliado solo tendrá acceso a la consulta de los datos del o de los prestamos realizados por él.

El módulo de usuarios que permite realizar el registro de los datos de los las personas que podrán tener acceso a la aplicación, asignado el tipo de usuario administrado o afiliado.

El módulo de herramientas, cuenta con las opciones de seguridad para la aplicación como copia y restauración de la base de datos. La implementación de la exportación e importación de la base de datos entre la aplicación y otros sistemas contables, utilidades que solo podrán ser gestionadas por el usuario tipo administrador.

El módulo de configuración que facilita el registro de nuevos tipos de aporte o tipos de préstamo que se podrán realizar en la cooperativa y su respectiva consulta modificación y alineación.

ABSTRACT

This paper presents the development of the thesis, analysis and design of **AFICOP- APLICACIÓN WEB PARA LOS AFILIADOS DE LAS COOPERATIVAS.**

This project is a web application developed to support the management of the cooperative at attention with their members in a more direct and effective and is divided into six modules:

The affiliate module that allows registration, query, modify and delete data affiliate, activities will be permitted only to the user type administrator.

The module Income Contributions and Payments allows registration, query, modify and delete data revenues for any concept of the cooperative data will be handled only by the type of root. It also allows the registration of the member contributions held in the cooperative, this register as modifying and deleting this data can only be done by the user type administrator, consulting affiliate contributions may be conducted by the type of affiliate administrator and user. And finally in this module will handle the payment of registration fees loans made by affiliates, modification, deletion, task that can only be done by root.

The module of Expenditures and Loan allows registration, query, modify and delete data outlays for any concept of the cooperative data will be handled only by the type of user and administrator registration and disposal of loans made by members, that only the user can make the administrator and the user types such affiliate has access only to the data query or loans made by him.

The module allows users to determine the registration of data users can access the application, assigning the type of user that will allow access depending on their type certain activities that can be performed in the application, consultation and Removal of these data.

The toolkit that has security settings for the application as the backup and restore of the database. The implementation of the export and import of the database between the application and other utilities accounting systems that can only be managed by the administrator user type.

The configuration module that facilitates the registration of new types of contribution or loan types that can be made in the cooperative and its respective query modification and deletion.

CONTENIDO

INTRODUCCION	22
1 MARCO TEORICO	29
1.1 ¿QUE ES UNA COOPERATIVA?	29
1.1.1 Características Principales De Una Cooperativa.....	29
1.1.2 Clases De Cooperativa..	29
1.1.3 Organismos De Dirección Y Control	30
1.2 METODOLOGÍA PARA EL DESARROLLO DE AFICOP	31
1.2.1 Definición de Programación Extrema.....	31
1.2.2 Valores de la Programación Extrema	32
1.2.3 Prácticas de La Programación Extrema.....	32
1.2.4 Ciclo De Vida De La Programación Extrema	34
1.3 TECNOLOGIAS PARA EL DESARROLLO DE AFICOP	37
1.3.1 JavaServer Pages (JSP).	37
1.3.2 Apache Tomcat.....	38
1.3.3 PostgreSQL.....	38
2 RESULTADOS DE LA INVESTIGACION.....	40
2.1 FASE DE PLANEACION	40
2.1.1.1 Historia de usuario No 1: Aplicación web.....	40
2.1.1.2 Historia de usuario No 2: Ingreso a la aplicación.....	40
2.1.1.3 Historia de usuario No 3: Módulos de la aplicación.....	41
2.1.1.4 Historia de usuario No 4: Bases de datos de la aplicación.	41
2.1.1.5 Historia de usuario No 5: Cuentas de usuario.....	41
2.1.1.6 Historia de usuario No 6: Perfiles de usuario	42
2.1.1.7 Historia de usuario No 7: Consultar cuentas de usuarios.....	42
2.1.1.8 Historia de usuario No 8: Modificar cuentas de usuario.	42
2.1.1.9 Historia de usuario No 9: Eliminar cuentas de usuario.	43

2.1.1.10	Historia de usuario No 10: Registrar formularios de afiliación.....	43
2.1.1.11	Historia de usuario No 11: Consultar afiliados.....	43
2.1.1.12	Historia de usuario No 12: Modificar afiliados.....	44
2.1.1.13	Historia de usuario No 13: Eliminar afiliados.	44
2.1.1.14	Historia de usuario No 14: Registrar aportes.	44
2.1.1.15	Historia de usuario No 15: Modificar aportes.	45
2.1.1.16	Historia de usuario No 16: Consultar aportes.	45
2.1.1.17	Historia de usuario No 17: Eliminar aportes.....	45
2.1.1.18	Historia de usuario No 18: Registrar préstamos.	45
2.1.1.19	Historia de usuario No 19: Consultar préstamos.	46
2.1.1.20	Historia de usuario No 20: Eliminar préstamos.....	46
2.1.1.21	Historia de usuario No 21: Registrar egresos.....	46
2.1.1.22	Historia de usuario No 22: Modificar egresos.	47
2.1.1.23	Historia de usuario No 23: Consultar egresos.....	47
2.1.1.24	Historia de usuario No 24: Eliminar egresos.	47
2.1.1.25	Historia de usuario No 25: Registrar pagos.....	48
2.1.1.26	Historia de usuario No 26: Modificar pagos.....	48
2.1.1.27	Historia de usuario No 27: Consultar pagos.....	48
2.1.1.28	Historia de usuario No 28: Registrar ingresos.	48
2.1.1.29	Historia de usuario No 29: Modificar ingresos.	49
2.1.1.30	Historia de usuario No 30: Consultar ingresos.	49
2.1.1.31	Historia de usuario No 31: Eliminar ingresos.....	49
2.1.1.32	Historia de usuario No 32: Reporte de afiliados.....	50
2.1.1.33	Historia de usuario No 33: Reporte de aportes.	50
2.1.1.34	Historia de usuario No 34: Reporte de préstamos.	50
2.1.1.35	Historia de usuario No 35: Reporte de pagos.....	51
2.1.1.36	Historia de usuario No 36: Reporte de egresos.....	51

2.1.1.37	Historia de usuario No 37: Reporte de ingresos.	51
2.1.1.38	Historia de usuario No 38: Importar base de datos.....	52
2.1.1.39	Historia de usuario No 39: Exportar base de datos.	52
2.1.1.40	Historia de usuario No 40: Visualizar bitácora.	52
2.1.1.41	Historia de usuario No 41: Generar copia de seguridad.....	53
2.1.1.42	Historia de usuario No 42: Restaurar copia de seguridad.....	53
2.1.1.43	Historia de usuario No 43: Ayuda del sistema.....	53
2.2	FASE DE DISEÑO	56
2.2.1	Metáfora del Sistema.	57
2.2.1.1	Características del módulo Usuarios.	57
2.2.1.2	Características del módulo Afiliados	57
2.2.1.3	Características del módulo Aportes	57
2.2.1.4	Características del módulo Préstamos y Egresos	58
2.2.1.5	Características del módulo Pagos e Ingresos	58
2.2.1.6	Características del módulo Reportes	58
2.2.1.7	Características del módulo Seguridad	58
2.2.2	Tarjetas CRC.....	59
2.2.2.1	Tarjetas CRC No 1: Conexión con la base de datos.	59
2.2.2.2	Tarjetas CRC No 2: Validación de los datos ingresados.	59
2.2.2.3	Tarjetas CRC No 3: Formulario de inicio de la aplicación.	59
2.2.2.4	Tarjetas CRC No 4: Administrar las cuentas de los usuarios.	60
2.2.2.5	Tarjetas CRC No 5: Registrar la cuenta de un usuario.	60
2.2.2.6	Tarjetas CRC No 6: Consultar usuario	61
2.2.2.7	Tarjetas CRC No 7: Modificar los datos del usuario.	61
2.2.2.8	Tarjetas CRC No 8: Establecer los permisos de acceso a un usuario.	62
2.2.2.9	Tarjetas CRC No 9: Registrar la eliminación de un usuario	62
2.2.2.10	Tarjetas CRC No 10: Administrar la información de un afiliado.....	62

2.2.2.11	Tarjetas CRC No 11: Registrar al nuevo afiliado.	63
2.2.2.12	Tarjetas CRC No 12: Formulario de información general del afiliado.	63
2.2.2.13	Tarjetas CRC No 13: Formulario de información económica del afiliado.	63
2.2.2.14	Tarjetas CRC No 14: Formulario de información financiera del afiliado.	64
2.2.2.15	Tarjetas CRC No 15: Formulario de información de los beneficiarios del afiliado.	64
2.2.2.16	Tarjetas CRC No 16: Información del tutor de los beneficiarios del afiliado.	64
2.2.2.17	Tarjetas CRC No 17: Consultar la información del afiliado.	65
2.2.2.18	Tarjetas CRC No 18: Modificar la información del afiliado.	65
2.2.2.19	Tarjetas CRC No 19: Eliminar la información del afiliado.	66
2.2.2.20	Tarjetas CRC No 20: Administrar tipos de aporte.	66
2.2.2.21	Tarjetas CRC No 21: Registrar tipos de aporte.	66
2.2.2.22	Tarjetas CRC No 22: Consultar información de los tipos de aporte.	67
2.2.2.23	Tarjetas CRC No 23: Modificar la información de los tipos de aporte.	67
2.2.2.24	Tarjetas CRC No 24: Eliminar la información de los tipos de aporte.	68
2.2.2.25	Tarjetas CRC No 25: Administrar los aportes del afiliado.	68
2.2.2.26	Tarjetas CRC No 26: Registrar aportes al afiliado.	68
2.2.2.27	Tarjetas CRC No 27: Consultar los aportes del afiliado.	69
2.2.2.28	Tarjetas CRC No 28: Modificar los aportes del afiliado.	69
2.2.2.29	Tarjetas CRC No 29: Eliminar los aportes del afiliado.	70
2.2.2.30	Tarjetas CRC No 30: Administrar la información de los tipos de préstamo.	70
2.2.2.31	Tarjetas CRC No 31: Registrar la información de los tipos de préstamo.	70

2.2.2.32	Tarjetas CRC No 32: Consultar la información de los tipos de préstamo.	71
2.2.2.33	Tarjetas CRC No 33: Modificar la información de los tipos de préstamo.	71
2.2.2.34	Tarjetas CRC No 34: Eliminar la información de los tipos de préstamo.	72
2.2.2.35	Tarjetas CRC No 35: Administrar préstamos del afiliado.	72
2.2.2.36	Tarjetas CRC No 36: Registrar préstamos al afiliado.	72
2.2.2.37	Tarjetas CRC No 37: Consultar préstamos del afiliado.	73
2.2.2.38	Tarjetas CRC No 38: Eliminar préstamos del afiliado.	74
2.2.2.39	Tarjetas CRC No 39: Administrar egresos.	74
2.2.2.40	Tarjetas CRC No 41: Registrar egresos.	74
2.2.2.41	Tarjetas CRC No 41: Consultar egresos.	75
2.2.2.42	Tarjetas CRC No 42: Modificar egresos.	75
2.2.2.43	Tarjetas CRC No 43: Eliminar egreso.	75
2.2.2.44	Tarjetas CRC No 44: Administrar Ingresos.	76
2.2.2.45	Tarjetas CRC No 45: Registrar Ingresos.	76
2.2.2.46	Tarjetas CRC No 46: Consultar Ingresos.	76
2.2.2.47	Tarjetas CRC No 47: Modificar Ingresos.	76
2.2.2.48	Tarjetas CRC No 48: Eliminar ingresos.	77
2.2.2.49	Tarjetas CRC No 49: Administrar pagos	77
2.2.2.50	Tarjetas CRC No 50: Registrar pagos.	78
2.2.2.51	Tarjetas CRC No 51: Consultar pagos.	78
2.2.2.52	Tarjetas CRC No 52: Modificar pagos.	78
2.2.2.53	Tarjetas CRC No 53: Reporte de información del afiliado	78
2.2.2.54	Tarjetas CRC No 54: Reporte de aportes de afiliados.	79
2.2.2.55	Tarjetas CRC No 55: Reporte de préstamos.	79
2.2.2.56	Tarjetas CRC No 56: Reporte de pagos.	79
2.2.2.57	Tarjetas CRC No 57: Reporte de egresos.	80

2.2.2.58	Tarjetas CRC No 58: Reporte de ingresos	80
2.2.2.59	Tarjetas CRC No 59: Reporte de usuarios	80
2.2.2.60	Tarjetas CRC No 60: Reporte de Tipo de aportes.....	81
2.2.2.61	Tarjetas CRC No 61: Reporte de Tipo de préstamos	81
2.2.2.62	Tarjetas CRC N° 62: Bitácora.	81
2.2.2.63	Tarjetas CRC No 63: Reporte Bitácora	82
2.2.2.64	Tarjetas CRC No 64: Crear las copias de las bases de datos.	82
2.2.2.65	Tarjetas CRC No 65: Restaurar las copias de las bases de datos.	82
2.2.2.66	Tarjeta CRC No 66: Importar bases de datos.....	83
2.2.2.67	Tarjetas CRC No 67: Exportar las bases de datos.	83
2.3	FASE DE DESARROLLO.....	84
2.3.1	Unidades de Prueba.....	84
2.3.1.1	Unidad de prueba No 1: Conexión con la base de datos.....	84
2.3.1.2	Unidad de prueba No 2: Validación de los datos	84
2.3.1.3	Unidad de prueba No 3: Página de inicio.....	85
2.3.1.4	Unidad de prueba No 4: Administrar usuario	85
2.3.1.5	Unidad de prueba No 5: Crear usuario.....	86
2.3.1.6	Unidad de prueba No 6: Consultar usuario	87
2.3.1.7	Unidad de prueba No 7: Editar usuario	87
2.3.1.8	Unidad de prueba No 8: Activar cuenta de usuario	88
2.3.1.9	Unidad de prueba No 9: Eliminar usuario	88
2.3.1.10	Unidad de prueba No 10: Administrar afiliado.....	89
2.3.1.11	Unidad de prueba No 11: Crear afiliado	90
2.3.1.12	Unidad de prueba No 12: Formulario de información general	91
2.3.1.13	Unidad de prueba No 13: Formulario de información economica	91
2.3.1.14	Unidad de prueba No 14: Formulario de información financiera	92

2.3.1.15	Unidad de prueba No 15: Formulario de información de de beneficiarios	92
2.3.1.16	Unidad de prueba No 16:Formulario información de Tutores.....	93
2.3.1.17	Unidad de prueba No 17: Consultar afiliado	94
2.3.1.18	Unidad de prueba No 18: Modificar afiliado.....	94
2.3.1.19	Unidad de prueba No 19: Eliminar afiliado	95
2.3.1.20	Unidad de prueba No 20: Administrar tipos de aporte	95
2.3.1.21	Unidad de prueba No 21: Crear tipos de aporte	96
2.3.1.22	Unidad de prueba No 22: Consultar tipos de aporte.....	97
2.3.1.23	Unidad de prueba No 23: Modificar tipos de aporte.....	97
2.3.1.24	Unidad de prueba No 24: Eliminar tipos de aporte	98
2.3.1.25	Unidad de prueba No 25: Administrar aportes del afiliado	99
2.3.1.26	Unidad de prueba No 26: Registrar aportes del afiliado	99
2.3.1.27	Unidad de prueba No 27: Consultar aportes del afiliado.....	100
2.3.1.28	Unidad de prueba No 28: Modificar aportes del afiliado.....	101
2.3.1.29	Unidad de prueba No 29: Eliminar aportes del afiliado	102
2.3.1.30	Unidad de prueba No 30: Administrar tipos de préstamos	102
2.3.1.31	Unidad de prueba No 31: Crear tipos de préstamos	103
2.3.1.32	Unidad de prueba No 32: Consultar información de los tipos de préstamos	104
2.3.1.33	Unidad de prueba No 33: Modificar información de los tipos de préstamos	104
2.3.1.34	Unidad de prueba No 34: Eliminar información de los tipos de préstamos	105
2.3.1.35	Unidad de prueba No 35: Administrar préstamos del afiliado.....	106
2.3.1.36	Unidad de prueba No 36: Registrar préstamo del afiliado.....	106
2.3.1.37	Unidad de prueba No 37: Consultar préstamo del afiliado.....	107
2.3.1.38	Unidad de prueba No 38: Eliminar préstamos del afiliado	108

2.3.1.39	Unidad de prueba No 39: Administrar egresos	108
2.3.1.40	Unidad de prueba No 40: Registrar egreso.....	109
2.3.1.41	Unidad de prueba No 41: Consultar egreso.....	110
2.3.1.42	Unidad de prueba No 42: Modificar egreso	111
2.3.1.43	Unidad de prueba No 43: Eliminar egreso	111
2.3.1.44	Unidad de prueba No 44: Administrar ingreso	112
2.3.1.45	Unidad de prueba No 45: Registrar ingreso	113
2.3.1.46	Unidad de prueba No 46: Consultar ingreso	113
2.3.1.47	Unidad de prueba No 49: Modificar ingreso	114
2.3.1.48	Unidad de prueba No 48: Eliminar ingreso	115
2.3.1.49	Unidad de prueba No 49: Administrar pago	116
2.3.1.50	Unidad de prueba No 50: Registrar pago.....	116
2.3.1.51	Unidad de prueba No 51: Consultar pago.....	117
2.3.1.52	Unidad de prueba No 52: Modificar pago.....	118
2.3.1.53	Unidad de prueba No 53: Reporte de usuarios.....	118
2.3.1.54	Unidad de prueba No 54: Reporte de afiliados	119
2.3.1.55	Unidad de prueba No 55: Reporte de aportes.....	120
2.3.1.56	Unidad de prueba No 56: Reporte de préstamos	120
2.3.1.57	Unidad de prueba No 57: Reporte de pagos	121
2.3.1.58	Unidad de prueba No 58: Reporte de Egresos	122
2.3.1.59	Unidad de prueba No 59: Reporte de Ingresos.....	122
2.3.1.60	Unidad de prueba No 60: Reporte de tipos de aporte.....	123
2.3.1.61	Unidad de prueba No 61: Reporte tipos de préstamo.....	124
2.3.1.62	Unidad de prueba No 62: Bitácora.....	124
2.3.1.63	Unidad de prueba No 63: Reporte de bitácora	125
2.3.1.64	Unidad de prueba No 64: Copia de seguridad de la base de datos	125
2.3.1.65	Unidad de prueba No 65: Restaurar copia de seguridad de la base de datos	126

2.3.1.66	Unidad de prueba No 66: Importar la base de datos.....	126
2.3.1.67	Unidad de prueba No 67: Exportar la base de datos.....	127
2.4	FASE DE DE PRUEBAS	127
2.4.1	Pruebas de Aceptación.	
2.4.1.1	Caso de Prueba de Aceptación No 1: Iniciar la Aplicación	128
2.4.1.2	Caso de Prueba de Aceptación No 2: Iniciar Sesión.....	129
2.4.1.3	Caso de Prueba de Aceptación No 3: Crear Usuario	129
2.4.1.4	Caso de Prueba de Aceptación No 4: Consultar cuentas de usuarios.	131
2.4.1.5	Caso de Prueba de Aceptación No 5: Modificar cuentas de usuarios.	133
2.4.1.6	Caso de Prueba de Aceptación No 6: Eliminar cuentas de usuarios.	134
2.4.1.7	Caso de Prueba de Aceptación No 7: Registrar Formularios de afiliación.....	135
2.4.1.8	Caso de Prueba de Aceptación No 8: Consultar afiliados.....	139
2.4.1.9	Caso de Prueba de Aceptación No 9: Modificar Afiliados.....	140
2.4.1.10	Caso de Prueba de Aceptación No 10: Eliminar Afiliados.....	141
2.4.1.11	Caso de Prueba de Aceptación No 11: Registrar Aportes.....	143
2.4.1.12	Caso de Prueba de Aceptación No 12: Consultar Aportes.....	144
2.4.1.13	Caso de Prueba de Aceptación No 13: Modificar Aportes.	145
2.4.1.14	Caso de Prueba de Aceptación No 14: Eliminar Aportes.	146
2.4.1.15	Caso de Prueba de Aceptación No15: Registrar Préstamos.....	147
2.4.1.16	Caso de Prueba de Aceptación No 16: Consultar préstamo.	148
2.4.1.17	Caso de Prueba de Aceptación No 17: Eliminar préstamos.....	150
2.4.1.18	Caso de Prueba de Aceptación No 18: Registrar Egreso.....	151
2.4.1.19	Caso de Prueba de Aceptación No 19: Consultar egresos.....	152
2.4.1.20	Caso de Prueba de Aceptación No 20: Modificar egresos.	153
2.4.1.21	Caso de Prueba de Aceptación No 21: Eliminar egresos.	154

2.4.1.22	Caso de Prueba de Aceptación No 22: Registrar pagos.....	156
2.4.1.23	Caso de Prueba de Aceptación No 23: Registrar Ingreso.	157
2.4.1.24	Caso de Prueba de Aceptación No 24: Consultar Ingresos.	158
2.4.1.25	Caso de Prueba de Aceptación No 25: Modificar Ingresos.	159
2.4.1.26	Caso de Prueba de Aceptación No 26: Eliminar Ingresos.....	161
2.4.1.27	Caso de Prueba de Aceptación No 27: Reporte de afiliados.....	162
2.4.1.28	Caso de Prueba de Aceptación No 28: Reporte de usuarios.	163
2.4.1.29	Caso de Prueba de Aceptación No 29: Reporte de aportes.	164
2.4.1.30	Caso de Prueba de Aceptación No 30: Reporte de préstamos. .	165
2.4.1.31	Caso de Prueba de Aceptación No 31: Reporte de pagos.....	166
2.4.1.32	Caso de Prueba de Aceptación No 32: Reporte de egresos.	167
2.4.1.33	Caso de Prueba de Aceptación No 33: Reporte de ingresos.	168
2.4.1.34	Caso de Prueba de Aceptación No 34: Importar base de datos.	169
2.4.1.35	Caso de Prueba de Aceptación No 35: Exportar base de datos.	170
2.4.1.36	Caso de prueba de aceptación No 36: Visualizar bitácora.	171
2.4.1.37	Caso de prueba de aceptación No 37: Generar copia de seguridad.....	173
2.4.1.38	Caso de prueba de aceptación No 38: Restaurar copia de seguridad.....	174
2.4.1.39	Caso de prueba de aceptación No 39: Ayuda de la Aplicación.	175
CONCLUSIONES		176
RECOMENDACIONES		177
BIBLIOGRAFIA		178
ANEXOS		179

LISTA DE ILUSTRACIONES

Ilustración 1 – Tablas de la Base de Datos – parte 1.....	54
Ilustración 2 – Tablas de la Base de Datos – parte 2.....	55
Ilustración 3 – Tablas de la Base de Datos – parte 3.....	55
Ilustración 4 – Relación entre todas las bases de datos	56

GLOSARIO

Afiliado: o asociado es una persona que forma parte de una cooperativa y que se caracteriza por tener los mismos derechos que todos lo demás asociados de la cooperativa.

Aplicación web: Una aplicación web es un programa o software al cual se accede a través de Internet mediante un Navegador Web. En otras palabras, una página web que realiza tareas similares a las que realizaría un programa estándar (de escritorio).

Aportes: es un valor entregado por el afiliado por voluntad propia deducible del salario, estipendios y otros, que puede ser de carácter devolutivo cuando así lo considere este.

Beneficiario: la persona a la cual el afiliado asigna porcentajes de beneficio de los aportes en caso de muerte o incapacidad definitiva del afiliado

Curso cooperativo: formación que debe tener el afiliado en los principios básicos del cooperativismo para pertenecer a la entidad.

Egresos: son todas aquellas salidas monetarias que la cooperativa realiza por diferentes conceptos.

Ingresos: son todas aquellas entradas monetarias que la cooperativa recibe por diferentes conceptos como aportes, pagos a créditos entre otros.

Pagos: son cuotas mensuales que el afiliado debe abonar para cubrir la deuda del préstamo solicitado.

Préstamo: Es un servicio que se presta cuando el afiliado se encuentra al día en sus obligaciones con la cooperativas para fines educativos, de turismo y recreación.

Salario Básico: salario que devenga el afiliado por su trabajo.

Servidor (equipo): Un equipo servidor es aquel en el que se ejecutan las Aplicaciones Servidor. Debe tener instaladas las funcionalidades necesarias para que las aplicaciones Servidor puedan ejecutarse.

Tutor: La persona responsable del manejo de los dineros otorgados por el afiliado a él (los) beneficiarios en caso de que estos sean menores de edad.

Usuario: es aquella persona que hace uso de las diferentes funcionalidades de la aplicación web y puede ser tipo de usuario Administrador que tiene la facultad de acceder de forma amplia a todos los módulos y tipo de usuario afiliados cuyo acceso es de forma restringida.

INTRODUCCIÓN

En la actualidad las tecnologías han sido usadas para mejorar la calidad de vida y optimizar los procesos básicos en todo tipo de organizaciones, de esta forma se da lugar a transformación de sistemas llevados de modo mecánico a sistemas con facilidad de almacenamiento búsqueda modificación y actualización continua de la información obtenida en cada una de estas.

De la misma manera el desarrollo orientado a la web desde su aparición hasta el día de hoy, ha ido cambiando de forma progresiva impulsando también un avance en los Sistemas de Información y en la forma en que éstos pueden ser mejorados. Así, mientras que los primeros Sistemas de Información ofrecen información a los usuarios, y se construían fundamentalmente como páginas estáticas o dinámicas e hiperenlaces entre ellas; en la actualidad, muchos de estos se conciben como un medio para ofrecer servicios a través de Internet.

Así mismo, hoy en día muchas de las grandes organizaciones están sustituyendo sus sistemas completamente integrados por redes del negocio en las cuales cada participante provee a los demás de servicios especializados.

Basados en lo anterior, se presenta el sistema de información para cooperativas; software a implementar en la **Cooperativa de Trabajadores de la Regional de Salud Centro Occidental “COOTRESCO” Túquerres**, el cual servirá como una importante herramienta de apoyo que permitirá tener mayor seguridad, veracidad y facilidad de acceso a la información, para mejorar la atención de los afiliados.

Este documento, presenta como trabajo de grado el desarrollo y la implementación del **SISTEMA DE INFORMACION PARA LA ADMINISTRACIÓN FINANCIERA DE LOS AFILIADOS A LAS COOPERATIVAS EN LA REGIÓN.**

En la primera parte de este documento se presenta especificaciones del problema objeto de estudio y el marco teórico que describe generalidades del funcionamiento y administración de una cooperativa, la metodología usada para el desarrollo de la aplicación y las tecnologías que se hicieron necesarias para el desarrollo de este proyecto. En las siguientes secciones se presenta el resultado de la investigación donde se describe las actividades realizadas según la metodología de desarrollo y por último las conclusiones y recomendaciones para un mejor aprovechamiento de esta aplicación.

DESCRIPCIÓN

PLANTEAMIENTO DEL PROBLEMA.

La importancia de la sistematización de procesos en las empresas, organizaciones y en la vida cotidiana, da lugar a solucionar los problemas de desorganización interna, la débil veracidad en la información recolectada y en algunos casos a la deficiencia en la prestación de servicios a la comunidad

Actualmente, las cooperativas buscan implementar nuevas estrategias para mejorar su bajo desarrollo tecnológico y la deficiencia en la atención a los asociados; problemas administrativos que han causado la falta de motivación y la poca participación de estos, creándose así, la necesidad de un sistema que permita obtener una información veraz, oportuna, estructurada, confiable para realizar análisis de información correctos, un medio que ofrezca la posibilidad de estar más en contacto con sus asociados y conocer de primera mano sus necesidades.

Es entonces donde el nuevo sistema de información se ofrece como una herramienta de apoyo a los procesos de gestión administrativa, que además, cuenta con un espacio que permite aprovechar de manera más eficiente recursos tecnológicos como el Internet, para que los asociados puedan en cualquier momento y lugar conocer el estado actual de las actividades que se están llevando a cabo en la cooperativa, el estado de sus cuentas en aportes y/o créditos y así favorecer una participación más activa por parte de sus asociados.

FORMULACIÓN DEL PROBLEMA.

¿Cómo se pueden usar las nuevas tecnologías de la Información y la Comunicación (TIC'S) para mejorar los procesos de administración de los usuarios de la cooperativa de trabajadores de la regional de salud centro occidental "COOTRESCO" Túquerres?

SISTEMATIZACIÓN DEL PROBLEMA

- ¿De qué manera permite el uso de una aplicación Web gestionar los datos personales de cada uno de los afiliados a la cooperativa siguiendo el modelo de COOTRESCO?
- ¿De qué forma se puede gestionar eficientemente la información de aportes económicos que realicen los afiliados?

- ¿Cómo se puede tratar rápida y claramente los estados financieros, para la toma oportuna de decisiones en la asignación de préstamos y egresos generados en la cooperativa?
- ¿Cómo se puede registrar de una manera más fácil los pagos realizados por los afiliados de cada una de las cuotas que deban cancelar por motivo de los préstamos solicitados y los ingresos que tenga la cooperativa por algún otro concepto?
- ¿Cuál es la manera de garantizar la manipulación de la información generada por las actividades realizadas en la cooperativa, dependiendo del usuario que ingrese al sistema?
- ¿Cómo se puede generar eficientemente los reportes de los procesos para la toma de decisiones en la cooperativa?
- ¿Cuál es la manera de garantizar la seguridad e integridad de la información generada por las actividades realizadas en la cooperativa?
- ¿Cuál es la forma de personalizar el sistema para la identificación y reconocimiento de la cooperativa?
- ¿Cuál es la estrategia de empalme con los sistemas contables de cada cooperativa?

ALCANCE Y DELIMITACIÓN

El sistema de Información desarrollado es una Aplicación Web que facilita la organización de los procesos comunes en todas las cooperativas de trabajadores del sector salud en la región.

Dado que son muchas las cooperativas que existen en la región y en cada una de ellas manejan sus procesos con ciertas diferencias, según las políticas implementadas por cada Junta Administrativa, hallar un modelo general para satisfacer las necesidades de todas ellas es un poco difícil. Por ello, con el fin de tomar un modelo de referencia, el proyecto se centrará en la Cooperativa de Trabajadores de la Regional de Salud Centro Occidental "COOTRESCO", de la ciudad de Túquerres, que es una de las más grandes e importantes de la región.

Puesto que AFICOP es una Aplicación Web (WebApp), su instalación, configuración y uso puede realizarse en sistemas operativos de la familia Windows

y en los basados en la plataforma UNIX. La WebApp estará compuesta por varios módulos que se describen a continuación:

Módulo de Afiliados: Desde esta sección de la aplicación se puede administrar la 'materia prima' del software: las personas que están afiliadas a la cooperativa. Todos aquellos que pertenezcan a la entidad y reciban sus beneficios deben ser registrados en el sistema, utilizando esta sección.

Además, se puede realizar las respectivas modificaciones de los datos de los afiliados en caso de ser necesario, cada vez que se realicen cambios en sus estados. Por otro lado, cuando una persona se retira del sistema y deja de pertenecer a la cooperativa, se la podrá eliminar. En cuyo caso, no se borrarán totalmente los datos sino que se desactivará a la persona, para poderla activar si resulta necesario.

Los Afiliados no tienen la posibilidad de uso de los módulos del sistema como los administradores y demás personas que lo manipulen, pero sí tendrán acceso a él. Al momento de ingresarlos al sistema, se les asignará una cuenta de usuario y ellos activarán su registro asignando una clave de acceso a su cuenta de usuario, con la cual podrán hacer consultas de su estado financiero, de los pagos efectuados y del estado de los préstamos realizados.

Módulo de Aportes: Este módulo permite gestionar los aportes económicos que realicen los afiliados. El sistema permitirá que los aportes solo sean ingresados. En el caso que se desee modificar algún registro, la aplicación solo le permite hacerlo al Administrador.

En este módulo se puede realizar consultas relacionadas con los ingresos diarios, semanales, mensuales y trimestrales que llegan a la cooperativa. También es posible consultar los aportes de cada afiliado, aportando sumatorias y datos resumidos que ayuden a conocer el estado actual financiero de la persona.

Módulo de Préstamos y Egresos: Uno de los beneficios que obtiene la persona que está afiliada a cooperativa, es la posibilidad de gestionar y recibir préstamos de una cantidad de dinero a un plazo determinado, con el compromiso de pagar cuotas periódicas a las que se les aumenta una tasa de interés considerablemente baja.

Mediante las funcionalidades de este módulo se puede gestionar la asignación de estos préstamos. Primero se debe realizar una solicitud, que es ingresada en el sistema, esta es analizada por la Junta Directiva y aprobada o denegada, según sean los criterios de evaluación que apliquen.

También es posible modificar los datos del préstamo, o eliminarlo, si es necesario, antes de que se confirme que se ha realizado la entrega de éste al afiliado solicitante. Por cierto, este módulo es manipulado por usuarios Administradores.

Algo más que permite gestionar este módulo son los egresos. Cada que la cooperativa realice compras, haga pagos de facturas o actividades que impliquen la salida de dinero de la entidad, se registra en el sistema y desde este módulo se gestiona estas acciones.

De esta última, y de las demás actividades mencionadas del módulo, se puede realizar consultas especializadas, relacionadas con los préstamos y el estado financiero de los afiliados, así como de los gastos que la cooperativa tenga en determinado periodo.

Módulo de Pagos e Ingresos: En el anterior módulo se plantea la opción de realizar préstamos a los afiliados. Ahora, la misión para éste módulo consiste en gestionar los pagos que se realicen los afiliados de cada una de las cuotas que deban cancelar por motivo de los préstamos que hayan solicitado.

Como suele suceder, es posible que alguno de los afiliados se atrase con el pago de sus cuotas. En este caso, el módulo se encarga de aplicar los incrementos por mora que defina la Junta Directiva. Por otro lado, en caso de que el afiliado solo aporte una parte del valor de alguna cuota, el sistema gestiona esta acción para que se pueda conocer cuánto pagó de dicha cuota y cuánto adeuda de ella.

De forma adicional, el sistema permite la gestión de los ingresos que tenga la cooperativa por algún otro concepto. Por ejemplo, la cooperativa puede recibir dinero de fuentes como estas: entidades del estado o personas naturales que quizá hagan alguna donación, la venta de algún producto, mueble o inmueble que ésta posea, o incluso por la prestación de algún servicio; de nuevo, dependiendo de las políticas que hayan sido establecidas en ella.

Finalmente, tal como en el módulo anterior, este permite realizar consultas de los pagos efectuados por los afiliados y de los ingresos llevados a cabo durante un periodo de tiempo.

Módulo de Usuarios: Con este módulo se realiza la gestión de las personas que usarán la WebApp. Como no todos los usuarios tienen acceso al sistema para realizar las mismas actividades, se realizan perfiles de acceso que les permitan manipular algunas secciones y otras no.

La información de los usuarios es almacenada en el sistema, y podrá ser actualizada en cualquier momento. También es posible eliminar a cualquier usuario que ya no deba tener acceso al sistema. Estas labores, así como la manipulación del resto del módulo, son llevadas a cabo por el Administrador principal de la Aplicación.

Además, puesto que las actividades de los usuarios son monitoreadas por el sistema, este módulo permite observar un registro en forma de bitácora de cada una de las acciones realizadas por cada usuario.

Módulo de Reportes: Aunque en cada módulo se puede realizar consultas e imprimir reportes de los datos presentes en ellos, en este módulo la Junta Directiva puede encontrar otro tipo de reportes serán de mucha utilidad para la toma de decisiones en la cooperativa.

Estos reportes, con características de asientos contables (reporte Diario de ingresos y gastos, resúmenes similares al libro Mayor y similares al Balance general, entre otros), contribuirán a la administración financiera de la cooperativa.

Los reportes son generados en formato PDF para evitar su manipulación una vez generados.

Módulo de Seguridad: En cuanto a la seguridad de los datos del sistema, la aplicación provee las funcionalidades que permita crear copias de seguridad de forma sencilla, para evitar que los administradores del sistema tengan que ejecutar sentencias de código SQL directamente en la base de datos.

Posteriormente, las copias de seguridad generadas por la aplicación, puede ser restaurada de forma simple, llamándolas desde este módulo. Así recuperar la información en caso de presentarse fallas en el sistema de almacenamiento donde se encuentre la base de datos.

ANTECEDENTES

En las respectivas instituciones educativas universitarias públicas y privadas del departamento de Nariño en el área de Ingeniería de Sistemas no se encontró proyectos de grado que presentaran sistemas de información en cooperativas.

A continuación se referencia software para cooperativas que si bien realizan algunas de las actividades que proponemos en este proyecto, el impedimento radica principalmente en los altos costos generados en cuanto a la adquisición, las respectivas licencias que se generan y el mantenimiento de estos sistemas puesto que los representantes o encargados piden gastos tanto de transporte,

alojamiento y de alimentación sin tener en cuenta el tiempo que se demoran en trasladarse para atender al llamado de la empresa; tiempo importante que se pierde presentándose así deficiencias en la prestación del servicio.

ZEUS OPEN BANK

Es un software especializado para: cooperativas, fondos de empleados de ahorro y crédito, multiactivas, financieras. Permite llevar el control integrado de todas las áreas de la entidad, generando los documentos soportes tales como: Solicitudes de crédito, comprobantes de egreso, notas contables, impresión de cheques, cdts, y en general cualquier documento que haya sido definido. Cumple con las normas de control y genera informes para reportar a la supersolidaria,

LINUX SOFTWARE

Sistema de información integrado para el manejo especializado de operaciones Comerciales, Administrativas y Financieras de cooperativas y fondos de empleados de ahorro y crédito, multiactivas, comerciales y de servicios, generación de documentos soportes; facilidad y capacidad de manejar grandes volúmenes de información con eficiencia y alto grado de seguridad e integridad. El sistema se compone de módulos como contabilidad, tesorería, crédito, cartera, aportes, convenios, asociados, cuentas de ahorro, nómina, auxilios, servicios, tarjeta debito/crédito, compras, ventas, gerencial y servicios web para asociados.

XEO XSOFTWARE

Sistema de información especializado para cooperativas, fondos de empleados y entidades de finanzas, en las áreas de: Cartera financiera, ahorros, solicitud y aprobación de crédito, diferentes modelos de extractos para los asociados, reporte de novedades de los asociados, aportes, contabilidad, caja, tesorería, reportes a Entidades de Control: súper solidaria, centrales de riesgo, asociados realicen consultas de saldos, solicitud de crédito y actualización de datos de la hoja de vida, vía Internet, generación de reportes tiene cubrimiento en el Oriente Colombiano, Bogotá, Costa Atlántica; Eje Cafetero, Valle del Cauca y Antioquia.

JUSTIFICACIÓN

Hoy en día la información hace parte de los principales recursos con los que cuenta cualquier organización u empresa, es un aspecto fundamental que se debe llevar de forma clara y precisa. En la actualidad la cooperativa lleva los procesos de obtención de datos manualmente es decir en papel; que en ciertas ocasiones presenta deficiencias en la calidad y seguridad de la información, lo que ha ocasionado un mal ambiente administrativo para los empleados, ya que por estos problemas de organización requieren de más tiempo para realizar algún reporte para los administrativos, o alguna otra información que es requerida por parte de los asociados, quienes ante éstas situaciones presentan quejas sobre el ineficiente sistema de atención. Es así que el desarrollo de un sistema de

información se convierte en una herramienta necesaria para el manejo eficiente de la información, éste permite obtener y acceder de forma más fácil a los datos y la optimización de los procesos inmersos, lo que va a favorecer el mejoramiento de las actividades realizadas a nivel administrativo y apoyar así, la toma oportuna de decisiones en asuntos vitales de la cooperativa, lo que permitirá la consolidación y desarrollo de la misma. Por otra parte la incorporación de los nuevos servicios tecnológicos ofrecidos a los asociados, proporcionará una comunicación más directa del ente administrativo con el asociado quién, podrá identificar claramente las necesidades del asociado y ser atendidas oportunamente, creando así, un ambiente favorable que va a motivar una participación más activa de estos.

OBJETIVOS

OBJETIVO GENERAL

Aplicar las nuevas tecnologías de la información y las comunicación (TIC'S) mediante el desarrollo un Sistema de Información que mejore los procesos de administración de los usuarios de la cooperativa de trabajadores de la regional de salud centro occidental COOTRESCO de la ciudad de Túquerres

OBJETIVOS ESPECÍFICOS

- Administrar la información de cada uno de los afiliados de la cooperativa siguiendo un patrón semejante a la cooperativa COOTRESCO.
- Desarrollar un módulo que permita gestionar los aportes económicos realizados por los afiliados.
- Desarrollar un módulo para la administración y control de la asignación de préstamos y egresos generados en la cooperativa
- Desarrollar un módulo que registre los ingresos por pago de cuotas que deban cancelar por motivo de los préstamos y los ingresos que se generen por algún otro concepto.
- Desarrollar un módulo que permita la administración de los usuarios que tienen los perfiles de acceso al sistema.
- Desarrollar un módulo que nos presente la información de forma ágil, fácil y eficiente en reportes debidamente organizados.
- Desarrollar un módulo que nos permita garantizar la seguridad de la información del sistema de forma sencilla, la configuración de la información general de la empresa y la exportación e importación los datos con los sistemas contables que estén funcionando en la cooperativa.

1 MARCO TEORICO

1.1 ¿QUE ES UNA COOPERATIVA?

El artículo 4 de la Ley 79 de 1988, establece que las cooperativas son aquellas empresas asociativas "sin ánimo de lucro en las cuales los trabajadores o los usuarios, según el caso, son simultáneamente los aportantes y los gestores de las empresas, creadas con el objeto de producir o distribuir conjunta y eficientemente bienes o servicios para satisfacer las necesidades de sus asociados y de la comunidad en general"¹.

1.1.1 Características Principales De Una Cooperativa. Es una empresa económica de interés social que funciona de conformidad con el principio de la participación democrática, la empresa es de propiedad colectiva. Los asociados tienen la posibilidad de asociarse y retirarse de ella libre y voluntariamente y se reparten los excedentes de forma equitativa.

1.1.2 Clases De Cooperativa. Las cooperativas, en razón del desarrollo de sus actividades, podrán ser especializadas, multiáctivas e integrales.

Cooperativas especializadas: Son cooperativas especializadas que se organizan para atender una necesidad específica, correspondiente a una sola rama de actividad económica, social o cultural. Ejemplo: Cooperativa de vivienda: cuyo objetivo único es la consecución de una vivienda.

Estas cooperativas pueden ofrecer servicios diferentes a los establecidos en su objeto social, mediante la suscripción de convenios con otras entidades cooperativas.

Cooperativas multiáctivas: Son cooperativas que se organizan para atender varias necesidades, mediante concurrencia de servicios en una sola entidad jurídica. Los servicios deberán ser organizados en secciones independientes, de acuerdo con las características de cada tipo especializado de cooperativa. Ejemplo: Cooperativa de transporte urbano donde se prestan los servicios de taxi colectivo e individual a la comunidad en general, a los socios (dueños de los taxis) se les presta los servicios de crédito además de la venta de productos y accesorios para automóviles.

Cooperativas Integrales: Son cooperativas que en el desarrollo su objeto social, realizan dos o más actividades conexas y complementarias entre sí, de producción

¹Sitio web: <http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=9211>

distribución, consumo y prestación de servicios. Ejemplo: Cooperativa de Productos Lácteos donde se fabrican sus productos, se encargan del abastecimiento de la materia prima y de insumos para la venta y distribución de estos.

1.1.3 Organismos De Dirección Y Control “La Cooperativa organiza sus recursos en forma colectiva con la participación de todos los socios y con igualdad de derechos y obligaciones. Todas las actividades que se desarrollan en ella, son dirigidas, coordinadas y controladas por los socios a través de organismos y cargos que tienen funciones específicas”². Los organismos de Dirección que conforman la estructura básica de una Cooperativa son:

La Asamblea General de Socios: es el órgano máximo de la administración su función es la toma de decisiones, constituida por los asociados hábiles o sus delegados y es elegida democráticamente.

El Concejo de Administración: Sus funciones son la planeación la organización, el control y la evaluación de las actividades económicas y sociales de la cooperativa. También deben presentar a la asamblea general el informe de actividades, el balance y la distribución de excedentes. Según los estatutos reglamentar:

Las sesiones del Consejo.

Los servicios de la Cooperativa.

Las ventas a crédito.

La inversión de fondos.

Las medidas de seguridad para los trabajadores socios o no socios.

La inversión de fondos.

La fiscalización económica por parte de los socios.

Nombrar a: Gerente, Secretario, Tesorero, Contador y demás empleados de la Cooperativa.

La Junta de Vigilancia y el Fiscal: Es un organismo de control social, conformado por un número máximo de tres asociados hábiles con sus respectivos suplentes, debe ser elegida por la Asamblea General, a quien reportara los resultados de su ejercicio .y en general según especifiquen términos de los estatutos de cada cooperativa esta junta debe Velar por el correcto funcionamiento y el cumplimiento de las normas legales y reglamentarias de la Cooperativa.

²Sitio web: <http://www.banrepcultural.org/>

El fiscal quien debe ser contador público con matrícula vigente y en ningún caso podrá ser socio de la cooperativa en la cual se desempeña como revisor fiscal, es el responsable del control en el aspecto contable y financiero de la cooperativa. Las funciones serán las señaladas en los estatutos y reglamentos de cada cooperativa.

Los Comités: Son órganos de apoyo que facilitan a los directivos, la gestión social y empresarial. Estos permiten lograr una mayor participación de los asociados en el desarrollo de los diferentes programas de la cooperativa. Se pueden conformar diferentes comités de acuerdo con el objeto económico y social, así de educación, de créditos, de compras, de vivienda, de solidaridad, bienestar social, técnico, etc. Pero hacen parte de la estructura administrativa los de educación y los de solidaridad.

Comité de Educación: Esta en la obligación de realizar de forma permanente actividades que tiendan a la formación de sus asociados y trabajadores en el ámbito de la economía solidaria y para los entes administrativos en gestión empresarial. Estas actividades por ley cuentan con un 20% de los excedentes.

Comité de solidaridad: Es el órgano encargado de desarrollar programas de promoción, previsión, de salud y asistenciales para atender eventos de calamidad y enfermedad de sus asociados. Cuenta con en un mínimo de un 10% de los excedentes.

1.2 METODOLOGÍA PARA EL DESARROLLO DE AFICOP

La metodología utilizada para el desarrollo de la **Aplicación Web Para Los Afiliados De Las Cooperativas- AFICOP** fue la programación extrema que es una de las metodologías donde una de sus principales características es la interacción continua con el cliente, permitiendo así responder con seguridad y facilidad al cambio de los requisitos del sistema planteados por este ya que se los hace a medida que se va desarrollando el programa y no al final donde tocaría cambiar un gran parte del programa. es especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes

1.2.1 Definición de Programación Extrema. La Programación Extrema (PX), mejor conocida por su nombre en inglés Extreme Programming (XP), es una de las llamadas metodologías ágiles de desarrollo de software; basada en una serie de valores y **prácticas** que intenta definir rápidamente un plan global del sistema,

desarrollar y liberar el software para posteriormente revisarlo continuamente e incorporarle características adicionales.

1.2.2 Valores de la Programación Extrema

- **Comunicación:** Es importante para el entendimiento del grupo de trabajo con el cliente quien siempre debe estar disponible para solucionar dudas igual para que los programadores puedan resolver rápidamente cualquier problema con el código.
- **Simplicidad:** En el diseño para facilitar el desarrollo y el mantenimiento, en la codificación para agilizar los cambios que se requieran en el código y realizar lo necesario y lo pedido por el cliente.
- **Retroalimentación:** Del sistema, del cliente, y del equipo, retroalimentación del sistema (mediante pruebas), los clientes o usuarios, y del grupo de trabajo; así en caso de haber algún error, poder corregirlo antes de que sea muy tarde.
- **Valentía:** Le permite a los desarrolladores que se sientan cómodos con reconstruir su código cuando sea necesario, de saber cuando se debe desechar un código; programar para hoy y no para mañana.
- **Respeto:** El equipo debe trabajar como uno, lo importante es sacar a delante el proyecto con la mejor calidad y desempeño no se pueden tomar decisiones repentinas sin consultar o comunicarlo al resto del equipo; cada uno debe respetar el trabajo que el otro realiza.

1.2.3 Prácticas de La Programación Extrema. La programación extrema especifica ciertas prácticas concretas de programación que deben llevarse a cabo al implementar este modelo.

- **Planificación:** El cliente y el equipo de desarrollo deben comunicarse claramente para que el programador estime el esfuerzo que requiere la implementación de las historias de usuarios, y el cliente decida el alcance y los tiempos de las entregas.
- **Entregas e iteraciones pequeñas:** Son partes del sistema que serán revisadas por el cliente; aunque estas no tengan todas las funcionalidades requeridas solo las mas importantes.
- **Metáfora:** Es una explicación simple y sencilla del sistema para que tanto los integrantes del grupo de trabajo y el cliente entiendan el funcionamiento y

finalidad del sistema. Cada proyecto es guiado por una historia simple de cómo funciona el sistema en general.

- **Diseño Simple:** Es la realización de un diseño de la forma más sencilla para que el sistema pueda ser implementado de forma rápida, contar con las funcionalidades estrictamente necesarias, todo el equipo de trabajo debe entenderlo.
- **Pruebas:** Toda característica en el programa debe ser probada, los programadores deben escribir pruebas para chequear el correcto funcionamiento del programa es decir las pruebas que se le realizaran al código y el cliente realizara las pruebas funcionales o del sistema.
- **Refactorización:** Se refiere a una reestructuración del código, eliminando la duplicación, simplificando funciones; modificar el código para mejorarlo sin cambiar su funcionamiento. Es el proceso de modificar el código de un desarrollo para mejorar su estructura interna sin alterar la funcionalidad que ofrece el desarrollo externamente⁴.
- **Programación en Parejas:** La programación debe hacerse en parejas de programadores que deben codificar en un solo ordenador, para aumentar la calidad del software ya que permite detectar más rápido la presencia de errores en el código.
- **Código Colectivo:** El código no es de propiedad individual sino grupal ya que todos participaron en la construcción y mejoramiento de este además esto permite que ningún programador sea imprescindible a la hora de cambios en el código pues cualquier persona puede hacerlo.
- **Integración Continúa:** Es el proceso de integración continua de código nuevo al sistema. Todos los casos de prueba se deben pasar antes y después de la integración.
- **Semana laboral de 40 horas:** Cada Trabajador labora no más de 40 horas por semana, esto hace que se incentive la creatividad, la motivación ya que evita el cansancio y permite también mejora de la calidad del producto.
- **Cliente en el Sitio:** El cliente debe estar siempre disponible para responder a las preguntas, fijar prioridades y sobre todo para la realización de las pruebas funcionales.

⁴Sitio web <http://www.programación.com>

- **Estándares de Codificación:** Se debe establecer un estándar de codificación aceptado e implantado por todo el equipo o grupo de trabajo.

1.2.4 Ciclo De Vida De La Programación Extrema

El ciclo de vida de XP se enfatiza en el carácter iterativo e incremental del desarrollo, una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas que en el caso de XP corresponden a un conjunto de historias de usuarios.

Las iteraciones deben ser cortas ya que entre mas rápido se entreguen los avances a los clientes más retroalimentación se va a obtener; lo que representa a largo plazo una mejor calidad en el producto. En cada iteración se realiza el análisis, diseño, desarrollo y pruebas; fases superpuestas de tal manera que no se separen en el tiempo.

■ Fase de Planeación

Aquí es necesario realizar las Historias de usuario; que son planteadas por el cliente y el primer paso a realizar, definir un plan de entregas que contiene unas fechas donde se entregaran las versiones del programa, las iteraciones y las parejas de programadores.

En esta fase el equipo de trabajo también deberá establecer las herramientas y las tecnologías que serán usadas para el desarrollo del trabajo.

- **Historias de usuario:** Son una descripción breve realizada por el cliente que contiene las características del sistema a desarrollar. Y serán la base para desarrollar las pruebas funcionales del sistema.
- **Plan de entregas:** Una vez definidas las historias de usuario los desarrolladores y el cliente determinaran un plan de publicaciones donde se indicaran la fechas de publicación de cada versión del programa y las historias de usuario que serán implementadas en cada versión.
- **Velocidad del proyecto:** Es la medida de la rapidez con que se desarrolla el proyecto; contando el número de historias de usuario que se pueden implementar en una iteración; para saber cuántas historias se desarrollan por iteración; y controlar el tiempo de desarrollo de cada iteración. Si se encuentra que no se está cumpliendo con las fechas del plan de entregas; replantear con el cliente un nuevo plan.

- **Programación en pareja:** Es una técnica que sugiere que el código producido en un proyecto de desarrollo de software sea implementado por dos personas en una misma computadora, de esta forma mientras uno puede desarrollar código el otro realizaría una revisión buscando errores así es menos probable producir un mal diseño.
- **Reuniones diarias:** Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

■ Fase de diseño

En esta fase se hace especial énfasis en los diseños simples y claros se usa una metáfora de sistema, no se incorporan funcionalidades extras; se dan soluciones puntuales y solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual.

- **Metáfora del sistema:** El concepto de metáfora del sistema se refiere a una manera sencilla de explicar el propósito del proyecto, tanto para el cliente como para el equipo de desarrolladores del proyecto.
- **Diseños simples:** Debido a que los requerimientos pueden cambiar; la metodología X.P sugiere que hay que realizar los diseños más simples posibles y centrarse en el cumplimiento de los requerimientos actuales.
- **Funcionalidad extra:** Solo se analiza lo que se debe desarrollar en la iteración actual, no se debe añadir funcionalidad extra al programa aunque se piense que se puede utilizar en un futuro.
- **Refactorizar:** Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad.
- **Tarjetas C.R.C:** El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

■ Fase de desarrollo

Esta fase de desarrollo se realiza de forma paralela con el diseño. Destacan la disponibilidad del cliente, la unidad de pruebas, la programación por parejas y la integración.

- ***El cliente está siempre disponible:*** Una de las premisas de la programación extrema es tener siempre disponible al cliente; quien con la ayuda de los desarrolladores realizara las historias de usuarios, seleccionara las que se realizaran en cada iteración.
- ***Unidad de pruebas:*** La producción de código está dirigida por las pruebas unitarias o pruebas que se realizan al código y se ejecutan constantemente para cada modificación del sistema y las pruebas funcionales que se escriben para la evaluación de las historias de usuario.
- ***Programación en parejas:*** Se propone desarrollar el código por parejas, en un mismo ordenador. Para favorecer la detección temprana de errores, obtener un código de mayor calidad que posibilite un mejor entendimiento por parte de los otros integrantes del grupo en caso de realizar cambios.
- ***Integración secuencial:*** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente además esta Es una forma de que todo el mundo esté trabajando con casi la última versión

■ Fase de pruebas

Uno de los pilares de la programación extrema es el proceso de pruebas, se debe probar tanto como sea posible y sólo se podrá liberar una nueva versión si ésta ha superado el cien por ciento de la totalidad de ellas. De no ser así, se buscará el error y se solucionará. En XP las pruebas del sistema se dividen en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

- **Pruebas unitarias:** Todos los módulos deben pasar las pruebas unitarias antes de ser liberados o publicados. El sistema y el conjunto de pruebas deben ser guardados junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.
Estas pruebas deben ser construidas por los programadores con el empleo de algún mecanismo que permita automatizarlas de modo que su implementación y ejecución consuman el menor tiempo posible.
- **Pruebas de aceptación:** También se conocen como pruebas funcionales. Estas pruebas son supervisadas por el cliente, quien comprueba que se cumple lo especificado en las historias de usuario. Por tanto, en todas las iteraciones, las historias de usuario que se lleven a cabo deberán pasar una o varias pruebas de aceptación.
Estas pruebas se diferencian totalmente de las pruebas unitarias en cuanto al papel que toma el cliente, ya que en las unitarias el cliente no tiene ningún tipo de intervención puesto que son de total competencia del equipo de desarrollo.
Los resultados de estas pruebas deben publicarse para que todo el equipo esté al tanto.
- **Detección y corrección de errores:** Cuando se encuentra un error es importante corregirlo lo antes posible y evitar que vuelvan a ocurrir errores similares. Para corregir el error es necesario crear una nueva prueba y así comprobar que se ha solucionado.

1.3 TECNOLOGIAS PARA EL DESARROLLO DE AFICOP

1.3.1 JavaServer Pages (JSP). Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

“Como parte de la familia de tecnologías Java, JSP tecnología permite el desarrollo rápido de aplicaciones basadas en Web que son independientes de la plataforma. La tecnología JSP separa la interfaz de usuario de generación de contenidos, permitiendo a los diseñadores cambiar el diseño general de la página, sin alterar el contenido dinámico subyacente.”⁵

⁵Sitio web <http://www.encyclopedia.us.es>

La tecnología JSP permite extender el lenguaje JSP creando etiquetas simples y reutilizables que ayudan a reducir la cantidad de código, facilitan el mantenimiento y la modularidad.

Una página JSP es archivo de texto simple que consiste en contenido HTML o XML con elementos JSP. Cuando un cliente pide una página JSP del sitio web y no se ha ejecutado antes, la página es inicialmente pasada al motor de JSP, el cual compila la página convirtiéndola en Servlet, la ejecuta y devuelve el contenido de los resultados al cliente.

La tecnología JSP hace que sea más rápido y más fácil que nunca para construir aplicaciones basadas en Web ya que se separa la lógica de la página de su diseño y visualización.

Una ventaja de JSP es que multiplataforma se puede utilizar en cualquier computador siempre que exista una máquina virtual de java para él.

1.3.2 Apache Tomcat. Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es un servidor web con soporte de servlets y jsp, es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Tomcat es un software desarrollado con java por lo tanto funciona en cualquier sistema operativo que disponga de la máquina virtual java correspondiente.

1.3.3 PostgreSQL. Es un sistema gestor de base de datos relacional orientado a objetos basado en el proyecto POSTGRES, de la universidad de Berkeley.

PostgreSQL, es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

Algunas de las características que posee PostgreSQL, son:

- Corre en la mayoría de los Sistemas Operativos más utilizados Linux, varias versiones de UNIX y Windows.

- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Alta concurrencia haciendo que varios usuarios accedan a una misma base de datos sin que se presenten bloqueos.
- Cumple la prueba ACID (Atomicity, Consistency, Integrity, Durability)
- Uso de índices avanzados; usa varios índices para resolver consultas sobre una misma tabla.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.

2 RESULTADOS DE LA INVESTIGACION

2.1 FASE DE PLANEACION

En esta fase se realizaron las entrevistas con el gerente de la cooperativa COOTRESCO, para la obtención de la información que fue registrada en las historias de usuario que sirvieron como base para que el equipo de trabajo; determinara los requerimientos de la aplicación, el planteamiento de las tareas y el cronograma de entregas.

2.1.1 HISTORIAS DE USUARIO. Las historias de usuario son escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles, representan los requisitos de la aplicación web, contienen no más de dos frases. Las historias de usuario son usadas para estimar los tiempos de desarrollo de la parte de aplicación. También se utilizan en la fase de pruebas para verificar si el programa cumple con lo que especifica la historia de usuario.

2.1.1.1 Historia de usuario No 1: Aplicación web.

Historia de usuario	
Número: 001	Desarrollador: Paola Pantoja
Nombre de historia: Aplicación en la web	
Fecha de entrega prevista: Nov /18	Iteración asignada: 1
Descripción: El programa que se desea desarrollar debe ser una aplicación web que se podrá acceder desde cualquier computador conectado a Internet.	
Observaciones:	

2.1.1.2 Historia de usuario No 2: Ingreso a la aplicación.

Historia de usuario	
Número: 002	Desarrollador: Paola Pantoja
Nombre de historia: Ingresar al sistema	
Fecha de entrega prevista: May /27	Iteración asignada: 1
Descripción: En la pantalla de bienvenida se cargaran los módulos y la conexión a la base de datos, después el usuario digitará su login (o nombre de usuario) y contraseña que le permitirán el acceso a las secciones del sistema dependiendo qué permisos le hayan sido concedidos.	
Observaciones:	

2.1.1.3 Historia de usuario No 3: Módulos de la aplicación.

Historia de usuario	
Número: 003	Desarrollador: Cristy Ríos
Nombre de historia: Menú de Módulos	
Fecha de entrega prevista: Oct /7	Iteración asignada: 1
Descripción: El sistema mostrara un menú que permitirá identificar la actividad que se quiere realizar es decir el ingreso a los diferentes módulos.	
Observaciones: Los módulos son Afiliados, Aportes, Préstamos y Egresos, Pagos e Ingresos, Usuarios, Reportes y Seguridad.	

2.1.1.4 Historia de usuario No 4: Bases de datos de la aplicación.

Historia de usuario	
Número: 004	Desarrollador: Cristy Ríos
Nombre de historia: Base de datos	
Fecha de entrega prevista: Jun/17	Iteración asignada: 1
Descripción: La base de datos que contendrá la información vital para el funcionamiento de la cooperativa estará centralizada en un equipo servidor.	
Observaciones:	

2.1.1.5 Historia de usuario No 5: Cuentas de usuario.

Historia de usuario	
Número: 005	Desarrollador: Paola Pantoja
Nombre de historia: Crear cuentas de usuario	
Fecha de entrega prevista: Jun/3	Iteración asignada: 1
Descripción: El sistema debe permitir la creación de cuentas de usuario. De cada usuario se registrará la siguiente información: documento de identidad, nombres y apellidos, login (o nombre de usuario), password (o contraseña) y nivel de acceso.	
Observaciones: Esta es una funcionalidad de módulo usuarios.	

2.1.1.6 Historia de usuario No 6: Perfiles de usuario

Historia de usuario	
Número: 006	Desarrollador: Paola Pantoja
Nombre de historia: Perfiles de usuario	
Fecha de entrega prevista: Jun/3	Iteración asignada: 2
Descripción: Los perfiles de usuario que se deben permitir crear son los siguientes: Administrador del Sistema: que debe tener acceso a todos los módulos del sistema y permiso para crear eliminar y modificar en cualquier sección de este. Auxiliar Administrativo: este usuario podrá utilizar todos los módulos del sistema, excepto el de Seguridad. Se encargará de ingresar datos relacionados con todos los movimientos de la cooperativa. Podrá modificar, pero no eliminar registros. Afiliado: este podrá tener el acceso a todos los módulos excepto al módulo de seguridad (de uso exclusivo del administrador) pero también en algunos módulos como los de aportes, préstamos y egresos, pagos e ingresos, únicamente podrá realizar consultas no se le permitirá ingresar datos, ni modificar ni eliminar.	
Observaciones: Esta es una funcionalidad de módulo usuarios.	

2.1.1.7 Historia de usuario No 7: Consultar cuentas de usuarios.

Historia de usuario	
Número: 007	Desarrollador: Cristy Ríos
Nombre de historia: Consultar cuentas de usuario	
Fecha de entrega prevista: Jun/3	Iteración asignada: 2
Descripción: El sistema debe permitir que se realicen las consultas de las cuentas de usuario.	
Observaciones: Esta es una funcionalidad de módulo usuarios.	

2.1.1.8 Historia de usuario No 8: Modificar cuentas de usuario.

Historia de usuario	
Número: 008	Desarrollador: Cristy Ríos
Nombre de historia: Modificar cuentas de usuario	
Fecha de entrega prevista: Jun/3	Iteración asignada: 2

Descripción: El sistema debe permitir que se modifique cualquiera de los datos de las cuentas de usuario.
Observaciones: Esta es una funcionalidad de módulo usuarios.

2.1.1.9 Historia de usuario No 9: Eliminar cuentas de usuario.

Historia de usuario	
Número: 009	Desarrollador: Paola Pantoja
Nombre de historia: Eliminar cuentas de usuario	
Fecha de entrega prevista: Jun/10	Iteración asignada: 2
Descripción: El sistema debe permitir que se realicen eliminaciones de las cuentas de usuario. Esta funcionalidad será permitida solo para el usuario administrador.	
Observaciones: Esta es una funcionalidad de módulo usuarios.	

2.1.1.10 Historia de usuario No 10: Registrar formularios de afiliación.

Historia de usuario	
Número: 010	Desarrollador: Paola Pantoja
Nombre de historia: Registrar Formularios de afiliación	
Fecha de entrega prevista: Jun/24	Iteración asignada: 3
Descripción: El sistema debe permitir la Registro de formularios de afiliación en donde se ingresará la Información general, económica, financiera, de beneficiarios, de tutores y cursos cooperativos.	
Observaciones: Esta es una funcionalidad del módulo afiliados.	

2.1.1.11 Historia de usuario No 11: Consultar afiliados.

Historia de usuario	
Número: 011	Desarrollador: Cristy Ríos
Nombre de historia: Consultar afiliados	
Fecha de entrega prevista: Jun/17	Iteración asignada: 3
Descripción: El sistema debe permitir realizar consultas de la información registrada	

en los formularios de afiliación.

Observaciones:

Esta es una funcionalidad del módulo afiliados.

2.1.1.12 Historia de usuario No 12: Modificar afiliados.

Historia de usuario	
Número: 012	Desarrollador: Cristy Ríos
Nombre de historia: Modificar afiliados	
Fecha de entrega prevista: Jul/1	Iteración asignada: 3
Descripción: El sistema debe permitir realizar cualquier modificación de la información registrada en los formularios de afiliación.	
Observaciones: Esta es una funcionalidad del módulo afiliados.	

2.1.1.13 Historia de usuario No 13: Eliminar afiliados.

Historia de usuario	
Número: 013	Desarrollador: Paola Pantoja
Nombre de historia: Eliminar afiliados	
Fecha de entrega prevista: Jun/24	Iteración asignada: 3
Descripción: El sistema debe permitir realizar la eliminación de información registrada en los formularios de afiliación.	
Observaciones: Esta es una funcionalidad del módulo afiliados.	

2.1.1.14 Historia de usuario No 14: Registrar aportes.

Historia de usuario	
Número: 014	Desarrollador: Paola Pantoja
Nombre de historia: Registrar aportes	
Fecha de entrega prevista: Jul/2	Iteración asignada: 4
Descripción: El sistema debe permitir el registro de aportes de los afiliados.	
Observaciones: Esta es una funcionalidad del módulo aportes.	

2.1.1.15 Historia de usuario No 15: Modificar aportes.

Historia de usuario	
Número: 015	Desarrollador: Cristy Ríos
Nombre de historia: Modificar aportes	
Fecha de entrega prevista: Jul/15	Iteración asignada: 4
Descripción: El sistema debe permitir realizar cualquier modificación de los aportes de los afiliados.	
Observaciones: Esta es una funcionalidad del módulo aportes.	

2.1.1.16 Historia de usuario No 16: Consultar aportes.

Historia de usuario	
Número: 016	Desarrollador: Cristy Ríos
Nombre de historia: Consultar aportes	
Fecha de entrega prevista: Jul/8	Iteración asignada: 4
Descripción: El sistema debe permitir realizar consultas de los aportes de afiliados.	
Observaciones: Esta es una funcionalidad del módulo aportes.	

2.1.1.17 Historia de usuario No 17: Eliminar aportes.

Historia de usuario	
Número: 017	Desarrollador: Paola Pantoja
Nombre de historia: Eliminar aportes	
Fecha de entrega prevista: Jul/15	Iteración asignada: 4
Descripción: El sistema debe permitir realizar la eliminación de los aportes realizados por el afiliado.	
Observaciones: Esta es una funcionalidad del módulo aportes.	

2.1.1.18 Historia de usuario No 18: Registrar préstamos.

Historia de usuario	
Número: 018	Desarrollador: Paola Pantoja
Nombre de historia: Registrar préstamos	

Fecha de entrega prevista: Agt/5	Iteración asignada: 5
Descripción: El sistema debe permitir el registro de los préstamos asignados a los afiliados.	
Observaciones: Esta es una funcionalidad del módulo préstamos y egresos.	

2.1.1.19 Historia de usuario No 19: Consultar préstamos.

Historia de usuario	
Número: 019	Desarrollador: Cristy Ríos
Nombre de historia: Consultar préstamos	
Fecha de entrega prevista: Jul/22	Iteración asignada: 5
Descripción: El sistema debe permitir realizar consultas de la información registrada.	
Observaciones: Esta es una funcionalidad del módulo préstamos y egresos.	

2.1.1.20 Historia de usuario No 20: Eliminar préstamos.

Historia de usuario	
Número: 020	Desarrollador: Paola Pantoja
Nombre de historia: Eliminar préstamos	
Fecha de entrega prevista: Jul/29	Iteración asignada: 5
Descripción: El sistema debe permitir realizar la eliminación de los préstamos realizados por el afiliado.	
Observaciones: Esta es una funcionalidad del módulo préstamos y egresos.	

2.1.1.21 Historia de usuario No 21: Registrar egresos.

Historia de usuario	
Número: 021	Desarrollador: Paola Pantoja
Nombre de historia: Registrar egresos	
Fecha de entrega prevista: Agt/5	Iteración asignada: 6
Descripción: El sistema debe permitir el registro de los egresos realizados en la cooperativa por cualquier concepto.	

Observaciones:

Esta es una funcionalidad del módulo préstamos y egresos.

2.1.1.22 Historia de usuario No 22: Modificar egresos.

Historia de usuario	
Número: 022	Desarrollador: Cristy Ríos
Nombre de historia: Modificar egresos	
Fecha de entrega prevista: Agt/5	Iteración asignada: 6
Descripción: El sistema debe permitir realizar cualquier modificación de egresos realizados en la cooperativa por cualquier concepto.	
Observaciones: Esta es una funcionalidad del módulo préstamos y egresos.	

2.1.1.23 Historia de usuario No 23: Consultar egresos.

Historia de usuario	
Número: 023	Desarrollador: Cristy Ríos
Nombre de historia: Consultar egresos	
Fecha de entrega prevista: Agt/5	Iteración asignada: 6
Descripción: El sistema debe permitir realizar consultas de egresos realizados en la cooperativa por cualquier concepto.	
Observaciones: Esta es una funcionalidad del módulo préstamos y egresos.	

2.1.1.24 Historia de usuario No 24: Eliminar egresos.

Historia de usuario	
Número: 024	Desarrollador: Paola Pantoja
Nombre de historia: Eliminar egresos	
Fecha de entrega prevista: Agt/5	Iteración asignada: 6
Descripción: El sistema debe permitir la eliminación de egresos realizados en la cooperativa por cualquier concepto.	
Observaciones: Esta es una funcionalidad del módulo préstamos y egresos.	

2.1.1.25 Historia de usuario No 25: Registrar pagos.

Historia de usuario	
Número: 025	Desarrollador: Paola Pantoja
Nombre de historia: Registrar pagos	
Fecha de entrega prevista: Agt/26	Iteración asignada: 7
Descripción: El sistema debe permitir el registro de los pagos de las cuotas de los préstamos realizados por los afiliados.	
Observaciones: Esta es una funcionalidad del módulo pagos e ingresos.	

2.1.1.26 Historia de usuario No 26: Modificar pagos.

Historia de usuario	
Número: 026	Desarrollador: Cristy Ríos
Nombre de historia: Modificar pagos	
Fecha de entrega prevista: Agt/12	Iteración asignada: 7
Descripción: El sistema debe permitir realizar cualquier modificación de los pagos de las cuotas de los préstamos realizados por los afiliados.	
Observaciones: Esta es una funcionalidad del módulo pagos e ingresos.	

2.1.1.27 Historia de usuario No 27: Consultar pagos.

Historia de usuario	
Número: 027	Desarrollador: Cristy Ríos
Nombre de historia: Consultar pagos	
Fecha de entrega prevista: Agt/19	Iteración asignada: 7
Descripción: El sistema debe permitir realizar consultas de los pagos de las cuotas de los préstamos realizados por los afiliados.	
Observaciones: Esta es una funcionalidad del módulo pagos e ingresos.	

2.1.1.28 Historia de usuario No 28: Registrar ingresos.

Historia de usuario	
Número: 028	Desarrollador: Paola Pantoja

Nombre de historia: Registrar ingresos	
Fecha de entrega prevista: Sep/2	Iteración asignada: 8
Descripción: El sistema debe permitir el registro de los ingresos realizados en la cooperativa por cualquier concepto.	
Observaciones: Esta es una funcionalidad del módulo pagos e ingresos.	

2.1.1.29 Historia de usuario No 29: Modificar ingresos.

Historia de usuario	
Número: 029	Desarrollador: Cristy Ríos
Nombre de historia: Modificar ingresos	
Fecha de entrega prevista: Sep/9	Iteración asignada: 8
Descripción: El sistema debe permitir realizar cualquier modificación de ingresos realizados en la cooperativa por cualquier concepto.	
Observaciones: Esta es una funcionalidad del módulo pagos e ingresos.	

2.1.1.30 Historia de usuario No 30: Consultar ingresos.

Historia de usuario	
Número: 030	Desarrollador: Cristy Ríos
Nombre de historia: Consultar ingresos	
Fecha de entrega prevista: Sep/9	Iteración asignada: 8
Descripción: El sistema debe permitir realizar consultas de los ingresos realizados en la cooperativa por cualquier concepto.	
Observaciones: Esta es una funcionalidad del módulo pagos e ingresos.	

2.1.1.31 Historia de usuario No 31: Eliminar ingresos.

Historia de usuario	
Número: 031	Desarrollador: Paola Pantoja
Nombre de historia: Eliminar ingresos	
Fecha de entrega prevista: Sep/16	Iteración asignada: 8
Descripción: El sistema debe permitir realizar la eliminación de ingresos realizados en	

la cooperativa por cualquier concepto.

Observaciones:

Esta es una funcionalidad del módulo pagos e ingresos.

2.1.1.32 Historia de usuario No 32: Reporte de afiliados.

Historia de usuario	
Número: 032	Desarrollador: Paola Pantoja
Nombre de historia: Reporte de afiliados	
Fecha de entrega prevista: Sep/23	Iteración asignada: 9
Descripción: El sistema permitirá al usuario configurar el reporte de afiliados según se desee y ofrecer las opciones de impresión y almacenamiento del mismo.	
Observaciones: Esta es una funcionalidad de módulo de reportes. Los reportes se generarán con base a la información existente en la base de datos.	

2.1.1.33 Historia de usuario No 33: Reporte de aportes.

Historia de usuario	
Número: 033	Desarrollador: Cristy Ríos
Nombre de historia: Reporte de aportes	
Fecha de entrega prevista: Sep/30	Iteración asignada: 9
Descripción: La aplicación permitirá al usuario configurar el reporte de aportes según se desee y ofrecer las opciones de impresión y almacenamiento del mismo.	
Observaciones: Esta es una funcionalidad de módulo de reportes. Los reportes se generarán con base a la información existente en la base de datos.	

2.1.1.34 Historia de usuario No 34: Reporte de préstamos.

Historia de usuario	
Número: 034	Desarrollador: Cristy Ríos
Nombre de historia: Reporte de préstamos	
Fecha de entrega prevista: Oct/7	Iteración asignada: 9
Descripción: El sistema permitirá configurar al usuario el reporte de préstamos según	

desea y ofrecer las opciones de impresión y almacenamiento del mismo.

Observaciones:

Esta es una funcionalidad de módulo de reportes.
Los reportes se generarán con base a la información existente en la base de datos.

2.1.1.35 Historia de usuario No 35: Reporte de pagos.

Historia de usuario	
Número: 035	Desarrollador: Paola Pantoja
Nombre de historia: Reporte de pagos	
Fecha de entrega prevista: Oct/21	Iteración asignada: 9
Descripción: El sistema permitirá al usuario la configuración del reporte según sus requerimientos, la visualización, impresión y almacenamiento.	
Observaciones: Esta es una funcionalidad de módulo de reportes. Los reportes se generarán con base a la información existente en la base de datos.	

2.1.1.36 Historia de usuario No 36: Reporte de egresos.

Historia de usuario	
Número: 036	Desarrollador: Paola Pantoja
Nombre de historia: Reporte de egresos	
Fecha de entrega prevista: Oct/14	Iteración asignada: 9
Descripción: El sistema permitirá al usuario la configuración del reporte según sus requerimientos, la visualización, impresión y almacenamiento.	
Observaciones: Esta es una funcionalidad de módulo de reportes. Los reportes se generarán con base a la información existente en la base de datos.	

2.1.1.37 Historia de usuario No 37: Reporte de ingresos.

Historia de usuario	
Número: 037	Desarrollador: Paola Pantoja
Nombre de historia: Reporte de ingresos	
Fecha de entrega prevista: Oct/17	Iteración asignada: 9

Descripción: El sistema permitirá al usuario la configuración del reporte según sus requerimientos, la visualización, impresión y almacenamiento.
Observaciones: Esta es una funcionalidad de módulo de reportes. Los reportes se generarán con base a la información existente en la base de datos.

2.1.1.38 Historia de usuario No 38: Importar base de datos.

Historia de usuario	
Número: 038	Desarrollador: Paola Pantoja
Nombre de historia: Importar bases de datos	
Fecha de entrega prevista: Oct/17	Iteración asignada: 9
Descripción: El sistema permitirá la importación de bases de datos externas que vienen en archivos planos además debe realizar las debidas comprobaciones de los datos encontrados en estos archivos para cargarlos a la base de datos del sistema.	
Observaciones: Esta es una funcionalidad de módulo de herramientas	

2.1.1.39 Historia de usuario No 39: Exportar base de datos.

Historia de usuario	
Número: 039	Desarrollador: Paola Pantoja
Nombre de historia: Exportar bases de datos	
Fecha de entrega prevista: Oct/17	Iteración asignada: 9
Descripción: El sistema permitirá la exportación de sus bases de datos en archivos planos para permitir la compatibilidad con otros sistemas.	
Observaciones: Esta es una funcionalidad de módulo de herramientas	

2.1.1.40 Historia de usuario No 40: Visualizar bitácora.

Historia de usuario	
Número: 040	Desarrollador: Cristy Ríos
Nombre de historia: Visualizar bitácora	
Fecha de entrega prevista: Nov/4	Iteración asignada: 10

<p>Descripción: El sistema permitirá realizar la consulta de la bitácora o el movimiento diario de la aplicación, o ver movimiento de días anteriores, la bitácora no podrá ser alterada o editada. A esta funcionalidad solo podrá acceder el administrador del sistema.</p>
<p>Observaciones: Esta es una funcionalidad de módulo de seguridad.</p>

2.1.1.41 Historia de usuario No 41: Generar copia de seguridad.

Historia de usuario	
Número: 041	Desarrollador: Paola Pantoja
Nombre de historia: Generar copia de seguridad	
Fecha de entrega prevista: Nov/11	Iteración asignada: 10
<p>Descripción: El sistema permitirá realizar una copia de seguridad de la bitácora o de la base de datos completa para su posterior revisión según la fecha requerida. A esta funcionalidad solo podrá acceder el administrador del sistema.</p>	
<p>Observaciones: Esta es una funcionalidad de módulo seguridad.</p>	

2.1.1.42 Historia de usuario No 42: Restaurar copia de seguridad.

Historia de usuario	
Número: 042	Desarrollador: Cristy Ríos
Nombre de historia: Restaurar copia de seguridad	
Fecha de entrega prevista: Nov/11	Iteración asignada: 10
<p>Descripción: El sistema permitirá restaurar una copia de seguridad de la bitácora o de la base de datos completa según la fecha requerida. A esta funcionalidad solo podrá acceder el Administrador del sistema.</p>	
<p>Observaciones: Esta es una funcionalidad de módulo seguridad. Si se realiza una restauración completa de la base de datos a un estado anterior los datos guardados en fechas posteriores se perderán.</p>	

2.1.1.43 Historia de usuario No 43: Ayuda del sistema.

Historia de usuario	
Número: 043	Desarrollador: Cristy Ríos

Nombre de historia: Ayuda del sistema	
Fecha de entrega prevista: Nov/11	Iteración asignada: 10
Descripción: El sistema brindara una ayuda al usuario para permitirle mayor facilidad de manejo y uso de la aplicación.	
Observaciones:	

2.1.2 Modelo de Datos. Un conjunto de normas que determinan las estructuras en las cuales se van a almacenar los datos de una base de datos y los tipos de operaciones que se van a realizar. El modelo de datos utilizado para la base de datos de AFICOP es el modelo de base de datos relacional, que es un modelo de datos basado en la teoría de conjuntos

2.1.2.1 Modelo de Datos Relacional. El modelo de base de datos relacional utiliza una relación, que es una estructura compuesta por filas llamadas tuplas y columnas llamadas atributos; los atributos se les denomina campos y son propiedades o características del concepto que se va a almacenar y las tuplas se les denomina registros y son los datos del concepto. En esta metodología no se utiliza la descripción de la base de datos, sin embargo en anexos, se presenta las principales tablas utilizadas en la base de datos de la aplicación.

2.1.2.2 Tablas de la base de datos

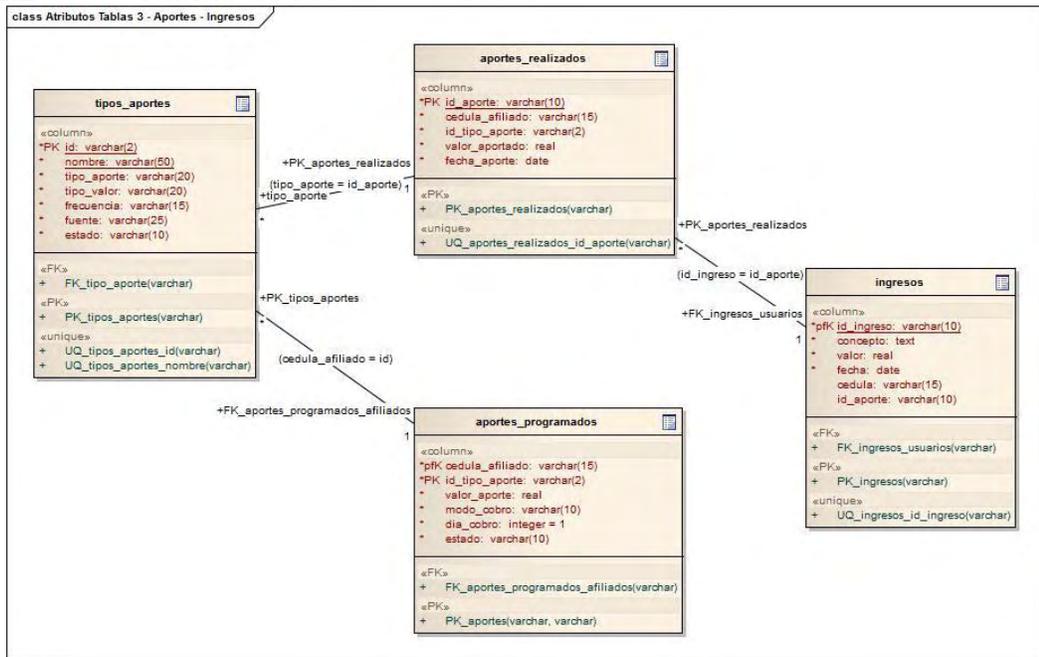


Ilustración 1 – Tablas de Base de Datos – Parte 1

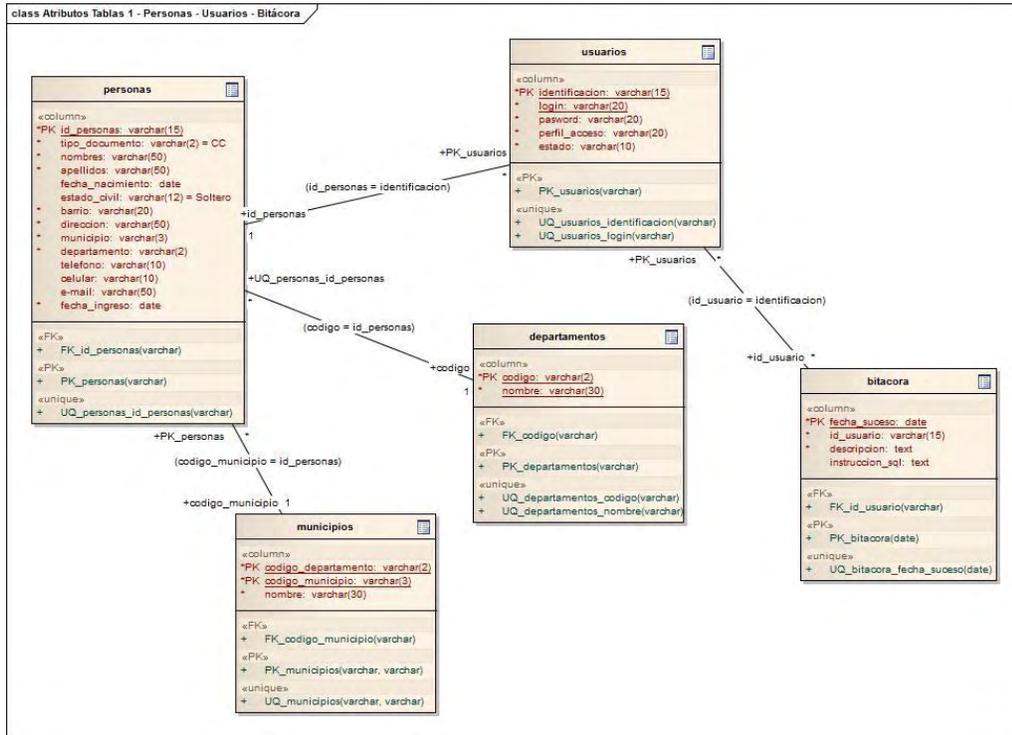


Ilustración 2 – Tablas de Base de Datos – Parte 2

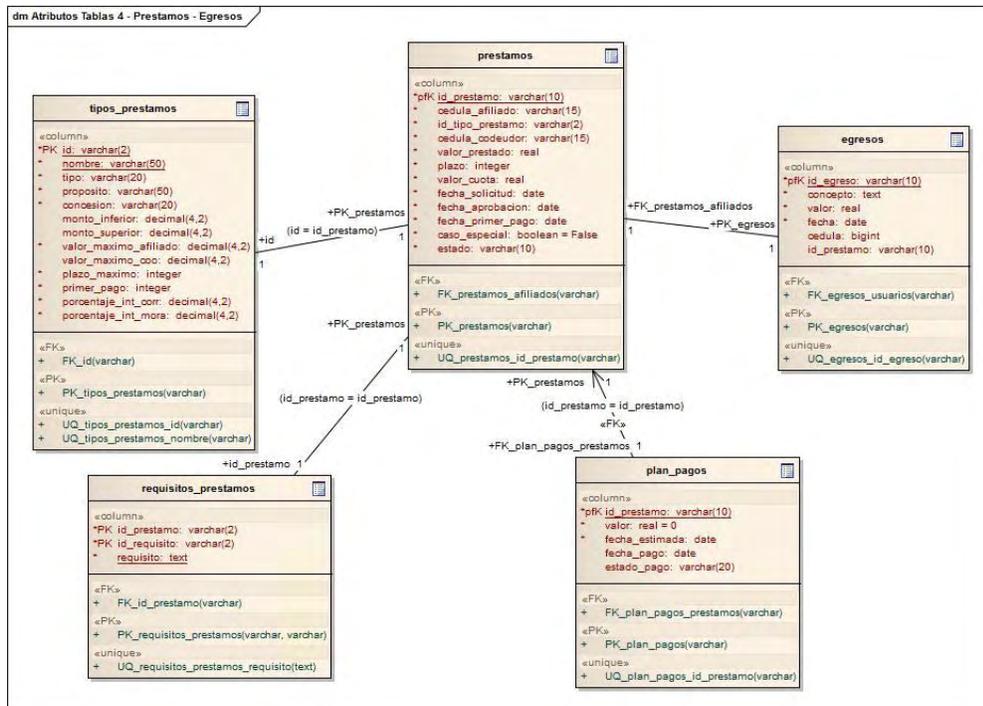


Ilustración 3 – Tablas de Base de Datos – Parte 3

2.1.2.3 Relación entre todas las tablas de la base de datos

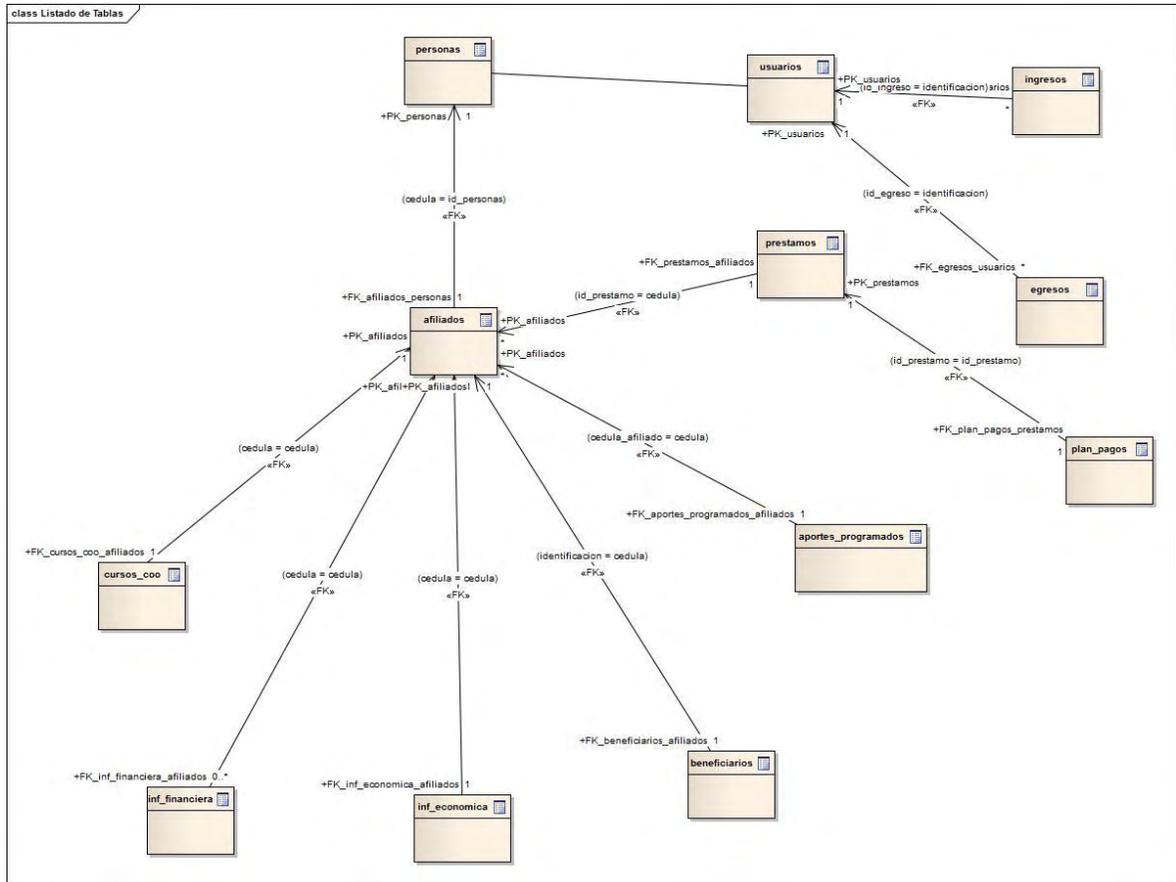


Ilustración 4 – Relación entre todas las Tablas.

2.2 FASE DE DISEÑO

En esta parte para la realización del diseño se hace necesario recurrir a una metáfora del sistema, que describe o cuenta acerca de cómo funciona el sistema para cualquier persona, tanto para usuarios como para el equipo de trabajo de desarrollo del sistema puedan entender el objeto del software que se va a desarrollar.

También se hace uso de las tarjetas CRC (Clase, Responsabilidad y Colaboración) que representa la descripción de una clase, en la parte superior se

ubica el nombre de la clase a modo de título en la tarjeta, las responsabilidades y las clases que implican colaboración.

2.2.1 Metáfora del Sistema. Estas son características generales que describen como será el funcionamiento de la aplicación, definiendo tanto las características generales como las de cada módulo.

- Es una aplicación orientada a la web que puede ser accedida desde cualquier computador que esté conectado a internet.
- Funciona para cualquier cooperativa de pequeño o mediano tamaño.
- Es necesario un equipo servidor para la instalación de la aplicación y de la base de datos.
- Para la seguridad del manejo del uso de sistema se debe crear un control de acceso a los usuarios especificando las restricciones a ciertas funciones del sistema según el caso.
- Para todos los usuarios del sistema se presenta una ayuda que le servirá como guía para hacer un uso adecuado de la funciones del sistema.

2.2.1.1 Características del módulo Usuarios.

- Se permite la creación de usuarios dl sistema, asignando los permisos respectivos para el ingreso y uso de la aplicación.
- Los usuarios pueden hacer consulta de los datos que ingresaron.
- Se hace un control sobre las modificaciones de los datos y si se requiere la eliminación de los mismos.

2.2.1.2 Características del módulo Afiliados

- En este módulo se crea un formulario que permita la obtención de la información del afiliado.
- Se puede modificar los datos de afiliado ya registrado.
- Usando este módulo también se puede hacer la consulta de la información ya registrada de cada afiliado por parte de los usuarios.
- Es posible eliminar a un afiliado ya registrado con anterioridad que ya no desee pertenecer a la cooperativa.

2.2.1.3 Características del módulo Aportes

- Este módulo permite el registro de aportes realizados por los afiliados a la cooperativa.
- De ser necesario se podrá realizar cualquier modificación a los aportes de los afiliados.

- Se puede realizar la consulta sobre los aportes de los afiliados por parte de los usuarios del sistema.
- También se puede eliminar los aportes del afiliado que desee retirarse.

2.2.1.4 Características del módulo Préstamos y Egresos

- Este módulo permite el registro de préstamos realizados a los afiliados de la cooperativa y los egresos generados por cualquier concepto.
- Se puede realizar cualquier modificación a la información sobre los egresos generados en la cooperativa.
- Se puede realizar la consulta sobre los préstamos a los afiliados y los egresos generados en la cooperativa.
- También se puede eliminar los préstamos cancelados por parte de los afiliados.

2.2.1.5 Características del módulo Pagos e Ingresos

- Este módulo permite el registro de pago de las cuotas de los préstamos realizados por los afiliados de la cooperativa y los ingresos generados por cualquier concepto.
- Se puede realizar modificaciones a la información sobre los pagos de las cuotas de los préstamos de los afiliados e ingresos generados en la cooperativa.
- Se puede realizar la consulta sobre los pagos de las cuotas de los préstamos de los afiliados e ingresos generados en la cooperativa.

2.2.1.6 Características del módulo Reportes

- Desde este módulo se permite al usuario extraer reportes de los resultados de la información obtenida de cada uno de los módulos del sistema por parte de cualquier usuario.
- También se dispone de ciertos parámetros que determinan la búsqueda según la necesidad del usuario.
- Todos los reportes son exportados en formato (*.PDF) para mayor seguridad evitando la manipulación de la información generada.
- Se permite imprimir los reportes consultados si el usuario lo desea.

2.2.1.7 Características del módulo Seguridad

- En este módulo se puede realizar la consulta de la bitácora del sistema del día que se requiera.

- Se puede realizar copia de seguridad tanto de la bitácora como de lavase de datos para su posterior revisión.
- Se puede realizar la restauración de copia de seguridad tanto de la bitácora como de la base de datos.

2.2.2 Tarjetas CRC. Las Tarjetas CRC que se describen a continuación, representan las principales clases que sirvieron para implementación de la aplicación web.

2.2.2.1 Tarjetas CRC No 1: Conexión con la base de datos.

TARJETA CRC N° 001	
NOMBRE DE LA CLASE	ConexionDB
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos e informar al usuario si se pudo establecer o no la conexión. • Ejecutar las consultas que el usuario desee sobre la base de datos. • Informar sobre la operación realizada. 	

2.2.2.2 Tarjetas CRC No 2: Validación de los datos ingresados.

TARJETA CRC N° 002	
NOMBRE DE LA CLASE	ValidarDatos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Conectarse con la base de datos. • Validar que la información ingresada en el formulario corresponda con las exigencias de la base de datos • Verificar si los datos ingresados por el usuario corresponden a los datos registrados en la base de datos. • Informar sobre el resultado de la operación. 	ConexiónDB

2.2.2.3 Tarjetas CRC No 3: Formulario de inicio de la aplicación.

TARJETA CRC N° 003	
NOMBRE DE LA CLASE	PaginalInicio
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Mostrar la interfaz principal de la aplicación. • Conectarse con la base de datos. 	ConexiónDB ValidarDatos

<ul style="list-style-type: none"> • Solicitar los datos para el control de acceso al usuario al sistema. • Validar los datos. • Estar a la espera de las solicitudes y abrir la sección que el usuario desee. • Informar sobre el resultado de la operación. 	
---	--

2.2.2.4 Tarjetas CRC No 4: Administrar las cuentas de los usuarios.

TARJETA CRC N° 004	
NOMBRE DE LA CLASE	AdministrarCuentaUsuario
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Conectarse con la base de datos. • Mostrar en pantalla el listado de los usuarios registrados. • Permitir la actualización del listado de usuarios en pantalla. • Permitir la selección de un usuario del listado en pantalla. • Mostrar las opciones de registro, modificación, eliminación • Permitir la creación de una nueva cuenta de usuario. • Ofrecer la opción de cerrar para cancelar toda operación. • Estar a la espera de la solicitud del usuario. • Informar el resultado de la operación realizada. 	ConexiónDB ValidarDatos

2.2.2.5 Tarjetas CRC No 5: Registrar la cuenta de un usuario.

TARJETA CRC N° 005	
NOMBRE DE LA CLASE	RegCuentaUsuario
RESPONSABILIDADES	COLABORADORES

<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar al usuario actual los datos de ingreso a la aplicación. • Permitir asignar los permisos de acceso, para la creación del nuevo usuario. • Ingresar a la base de datos al nuevo usuario. • Informar sobre el resultado de la operación. 	ConexionDB ValidarDatos AdministrarCuentaUsuario
---	---

2.2.2.6 Tarjetas CRC No 6: Consultar usuario

TARJETA CRC N° 006	
NOMBRE DE LA CLASE	ConsultarUsuario
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de un parámetro de búsqueda • Realizar las debidas valoraciones de la información ingresada • Mostrar listado de usuarios según el parámetro de búsqueda ingresado • Informar sobre el resultado de la operación. 	ConexionDB ValidarDatos AdministrarCuentaUsuario

2.2.2.7 Tarjetas CRC No 7: Modificar los datos del usuario.

TARJETA CRC N° 007	
NOMBRE DE LA CLASE	ModificarCuentaUsuario
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar al usuario actual los datos de ingreso a la aplicación. • Permitir la modificación de los datos del usuario como su password, login. • Ingresar a la base de datos los cambios realizados. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarCuentaUsuario RegCuentaUsuario

2.2.2.8 Tarjetas CRC No 8: Establecer los permisos de acceso a un usuario.

TARJETA CRC N° 008	
NOMBRE DE LA CLASE	ActivarCuentaUsuario
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar al usuario actual los datos de ingreso a la aplicación. • Ofrecer la opción de activar los permisos de acceso. • Ingresar a la base de datos los cambios realizados. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.9 Tarjetas CRC No 9: Registrar la eliminación de un usuario

TARJETA CRC N° 009	
NOMBRE DE LA CLASE	EliminarCuentaUsuario
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar al usuario los datos de identificación. • Mostrar en pantalla el tipo de usuario y el estado. • Realizar las validaciones necesarias para la eliminación. • Ingresar a la base de datos los cambios realizados. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarCuentaUsuario RegCuentaUsuario

2.2.2.10 Tarjetas CRC No 10: Administrar la información de un afiliado.

TARJETA CRC N° 009	
NOMBRE DE LA CLASE	AdministrarAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer la conexión con la base de datos. • Ofrecer las opciones de registro, consulta modificación y eliminación de los datos del afiliado a la cooperativa. • Estar a la espera de la operación que desee realizar el usuario. • Informar sobre el resultado de la operación realizada. 	ConexionDB ValidarDatos

2.2.2.11 Tarjetas CRC No 11: Registrar al nuevo afiliado.

TARJETA CRC N° 011	
NOMBRE DE LA CLASE	RegAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar el ingreso de la información necesaria (la información general, económica, financiera, de beneficiarios) para realizar la afiliación a la cooperativa. • Ofrecer la opción de modificar e eliminar la información de tutores y cónyuges. • Realizar las debidas validaciones de la información ingresada. • Registrar en la base de datos la información del nuevo afiliado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarAfiliado

2.2.2.12 Tarjetas CRC No 12: Formulario de información general del afiliado.

TARJETA CRC N° 012	
NOMBRE DE LA CLASE	FormInfGeneral
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar el ingreso de los datos información general del afiliado. • Realizar las debidas validaciones de la información ingresada. • Registrar en la base de datos la información. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.13 Tarjetas CRC No 13: Formulario de información económica del afiliado.

TARJETA CRC N° 013	
NOMBRE DE LA CLASE	FormInfEconomica
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar el ingreso de los datos información 	ConexionDB ValidarDatos

<p>económica del afiliado.</p> <ul style="list-style-type: none"> • Realizar las debidas validaciones de la información ingresada por el afiliado. • Registrar en la base de datos la información ingresada. • Informar sobre el resultado de la operación en curso. 	
---	--

2.2.2.14 Tarjetas CRC No 14: Formulario de información financiera del afiliado.

TARJETA CRC N° 014	
NOMBRE DE LA CLASE	FormInfFinanciera
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar el ingreso de los datos información financiera del afiliado. • Realizar las debidas validaciones de la información ingresada por el afiliado. • Registrar en la base de datos la información ingresada. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.15 Tarjetas CRC No 15: Formulario de información de los beneficiarios del afiliado.

TARJETA CRC N° 015	
NOMBRE DE LA CLASE	FormInfBeneficiarios
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar el ingreso de los datos información de los beneficiarios del afiliado. • Realizar las debidas validaciones de la información ingresada por el afiliado. • Registrar en la base de datos la información ingresada. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.16 Tarjetas CRC No 16: Información del tutor de los beneficiarios del afiliado.

TARJETA CRC N° 016

NOMBRE DE LA CLASE	FormInfTutores
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar los datos del tutor (nombres y apellidos, identificación, dirección, ciudad y teléfono) • Realizar las debidas validaciones de la información ingresada por el afiliado. • Realizar la acción que se le envió por parámetros. • Registrar en la base de datos la información ingresada. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.17 Tarjetas CRC No 17: Consultar la información del afiliado.

TARJETA CRC N° 017	
NOMBRE DE LA CLASE	ConsultarAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de un parámetro de búsqueda (nombre o cedula). • Realizar las debidas validaciones de la información ingresada. • Mostrar el listado de afiliados según el parámetro de búsqueda ingresado. • Permitir la selección del afiliado. • Mostrar la información de los datos básicos del afiliado seleccionado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos RegAfiliado FormCursos FormTutores FormInfBeneficiarios FormInfFinanciera FormInfEconomica FormInfGeneral

2.2.2.18 Tarjetas CRC No 18: Modificar la información del afiliado.

TARJETA CRC N° 018	
NOMBRE DE LA CLASE	ModificarAfiliado
RESPONSABILIDADES	COLABORADORES

<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Ofrecer al usuario el tipo de información (General, económica, financiera, tutores) que desea modificar. • Mostrar en pantalla el tipo de información que el usuario administrador eligió. • Permitir la modificación de esa información. • Registrar en la base de datos la información modificada. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos RegAfiliado FormCursos FormTutores FormInfBeneficiarios FormInfFinanciera FormInfEconomica FormInfgeneral ConsultarAfiliado
---	--

2.2.2.19 Tarjetas CRC No 19: Eliminar la información del afiliado.

TARJETA CRC N° 019	
NOMBRE DE LA CLASE	EliminarAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la eliminación de un afiliado. • Ofrecer la opción de confirmar eliminación de afiliado. • Registrar en la base de datos los cambios realizados. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos ConsultarAfiliado

2.2.2.20 Tarjetas CRC No 20: Administrar tipos de aporte.

TARJETA CRC N° 020	
NOMBRE DE LA CLASE	AdmonTiposDeAportes
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Ofrecer las opciones de crear, modificar, consultar y eliminar de los tipos de aporte que se realizaran en la cooperativa. • Estar a la espera de la opción seleccionada por el usuario administrador. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.21 Tarjetas CRC No 21: Registrar tipos de aporte.

TARJETA CRC N° 021

NOMBRE DE LA CLASE	RegTipoDeAportes
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el registro de los tipos de aporte que se realizaran en la cooperativa. • Permitir establecer los topes de los porcentajes mínimos y máximos de cada tipo de aporte en base al salario mínimo mensual legal vigente. • Registrar la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonTiposDeAportes

2.2.2.22 Tarjetas CRC No 22: Consultar información de los tipos de aporte.

TARJETA CRC N° 022	
NOMBRE DE LA CLASE	ConsultarInftAportes
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de un parámetro de búsqueda para realizar la consulta. • Mostrar en pantalla la información de los tipos de aporte registrados según el parámetro de búsqueda ingresado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonTiposDeAportes

2.2.2.23 Tarjetas CRC No 23: Modificar la información de los tipos de aporte.

TARJETA CRC N° 023	
NOMBRE DE LA CLASE	ModificarInftAportes
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla la información de los tipos de aportes. • Permitir la modificación de la información de los tipos de aporte. • Registrar en la base de datos las modificaciones realizadas. 	ConexionDB ValidarDatos AdmonTiposDeAportes ConsultarInftAportes

<ul style="list-style-type: none"> • Informar sobre el resultado de la operación en curso. 	
---	--

2.2.2.24 Tarjetas CRC No 24: Eliminar la información de los tipos de aporte.

TARJETA CRC N° 024	
NOMBRE DE LA CLASE	EliminartAportes
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la eliminación de un tipo de aporte. • Ofrecer la opción de confirmar eliminación. • Registrar en la base de datos los cambios realizados. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonTiposDeAportes ConsultarInftAportes

2.2.2.25 Tarjetas CRC No 25: Administrar los aportes del afiliado.

TARJETA CRC N° 025	
NOMBRE DE LA CLASE	AdmonAportesAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla la lista de los afiliados. • Permitir la selección de un afiliado. • Ofrecer las opciones de registrar, consultar, modificar y eliminar el aporte o aportes del afiliado. • Estar a la espera de la opción seleccionada por el usuario administrador. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.26 Tarjetas CRC No 26: Registrar aportes al afiliado.

TARJETA CRC N° 026	
NOMBRE DE LA CLASE	RegAportesAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla los tipos de aporte que se pueden realizar en la cooperativa. 	ConexionDB ValidarDatos AdmonAportesAfiliado

<ul style="list-style-type: none"> • Permitir la selección del tipo de aporte o aportes que puede realizar el afiliado (como mínimo debe quedar uno registrado). • Permitir el ingreso del valor del porcentaje en base al SMLV del aporte o los aportes que realizara el afiliado. • Permitir establecer las fechas que se realizara el recaudo del aporte o de los tipos de aportes. • Permitir establecer la forma de recepción del aporte. • Informar sobre el valor total de la cantidad a aportar por cada tipo de aporte. • Registrar la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	
---	--

2.2.2.27 Tarjetas CRC No 27: Consultar los aportes del afiliado.

TARJETA CRC N° 027	
NOMBRE DE LA CLASE	ConsultarAportesAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de un parámetro de búsqueda para realizar la consulta. • Mostrar en pantalla los datos del aporte según el parámetro de búsqueda ingresado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonAportesAfiliado

2.2.2.28 Tarjetas CRC No 28: Modificar los aportes del afiliado.

TARJETA CRC N° 028	
NOMBRE DE LA CLASE	ModificarAportesAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la modificación de la información de los tipos de aporte. • Registrar en la base de datos las modificaciones realizadas. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonAportesAfiliado ConsultarAportesAfiliado

2.2.2.29 Tarjetas CRC No 29: Eliminar los aportes del afiliado.

TARJETA CRC N° 029	
NOMBRE DE LA CLASE	EliminarAportesDelAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la eliminación de la información de un tipo de aporte realizado por el afiliado (mínimo se debe dejar el registro de un tipo de aporte). • Ofrecer la opción de confirmar eliminación. • Registrar en la base de datos los cambios realizados. • Informar sobre el resultado de la operación realizada. 	ConexionDB ValidarDatos AdmonAportesAfiliado ConsultarAportesAfiliado

2.2.2.30 Tarjetas CRC No 30: Administrar la información de los tipos de préstamo.

TARJETA CRC N° 030	
NOMBRE DE LA CLASE	AdmonTiposDePrestamos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla las opciones de registro, consulta, modificación y eliminación de la información de los préstamos que la cooperativa ofrece a los afiliados. • Estar a la espera de la opción seleccionada por el usuario administrador. • Informar sobre el resultado de la operación realizada. 	ConexionDB ValidarDatos

2.2.2.31 Tarjetas CRC No 31: Registrar la información de los tipos de préstamo.

TARJETA CRC N° 031	
NOMBRE DE LA CLASE	RegInftPréstamos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el registro de los tipos de préstamo que se pueden realizar en la cooperativa. 	ConexionDB ValidarDatos AdmonTiposDePrestamos

<ul style="list-style-type: none"> • Permitir el registro del monto para cada tipo de préstamo. • Permitir el registro del plazo para cada tipo de préstamo. • Permitir el registro del porcentaje del interés corriente y el interés por mora para cada tipo de préstamo. • Registrar la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	
---	--

2.2.2.32 Tarjetas CRC No 32: Consultar la información de los tipos de préstamo.

TARJETA CRC N° 032	
NOMBRE DE LA CLASE	ConsultarInftPrestamos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de un parámetro de búsqueda. • Mostrar en pantalla la información de los tipos de préstamo ofrecidos por la cooperativa según el parámetro de búsqueda ingresado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonTiposDePrestamos

2.2.2.33 Tarjetas CRC No 33: Modificar la información de los tipos de préstamo.

TARJETA CRC N° 033	
NOMBRE DE LA CLASE	ModificarInftPrestamos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla una lista de los tipos de préstamo ofrecidos por la cooperativa. • Permitir la selección de un tipo de préstamo para la modificación de sus datos. • Permitir la modificación de la información de los tipos de préstamo. • Registrar en la base de datos las modificaciones 	ConexionDB ValidarDatos ConsultarInftPrestamos AdmonTiposDePrestamos

realizadas.	
<ul style="list-style-type: none"> • Informar sobre el resultado de la operación en curso. 	

2.2.2.34 Tarjetas CRC No 34: Eliminar la información de los tipos de préstamo.

TARJETA CRC N° 034	
NOMBRE DE LA CLASE	EliminarInftPrestamos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la eliminación de un tipo de préstamo. • Ofrecer la opción de confirmar eliminación. • Registrar en la base de datos los cambios realizados. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos ConsultarInftPrestamos AdmonTiposDePrestamos

2.2.2.35 Tarjetas CRC No 35: Administrar préstamos del afiliado.

TARJETA CRC N° 035	
NOMBRE DE LA CLASE	AdmonPrestamoAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla la lista de los afiliados ya registrados. • Permitir la selección de un afiliado. • Ofrecer las opciones de registrar, consultar, modificar y eliminar el préstamo(s) del afiliado. • Estar a la espera de la opción seleccionada. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.36 Tarjetas CRC No 36: Registrar préstamos al afiliado.

TARJETA CRC N° 036	
NOMBRE DE LA CLASE	RegPrestamoAfiliado
RESPONSABILIDADES	COLABORADORES

<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla los tipos de préstamos que se pueden realizar en la cooperativa. • Permitir la selección del tipo préstamo que realizara el afiliado. • Permitir establecer la fecha de aprobación del préstamo. • Permitir la selección del monto del préstamo. • Permitir la selección del plazo del préstamo. • Permitir establecer las fechas que se realizara el pago de las cuotas del préstamo. • Permitir establecer la forma de recepción del pago de las cuotas del préstamo. • Informar sobre la cantidad del valor del préstamo. • Informar sobre el valor de las cuotas a pagar con interés corriente y por mora. • Informar sobre el número de cuotas a pagar. • Registrar la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonPrestamoAfiliado
--	--

2.2.2.37 Tarjetas CRC No 37: Consultar préstamos del afiliado.

TARJETA CRC N° 037	
NOMBRE DE LA CLASE	ConsultarPrestamosAfiliado
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de un parámetro de búsqueda. • Mostrar en pantalla la información correspondiente al registro del tipo de préstamo efectuado por el afiliado en la cooperativa según el parámetro de búsqueda ingresado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonPrestamoAfiliado

2.2.2.38 Tarjetas CRC No 38: Eliminar préstamos del afiliado.

TARJETA CRC N° 038	
NOMBRE DE LA CLASE	EliminarPrestamo
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la eliminación del préstamo del afiliado. • Ofrecer la opción de confirmar eliminación. • Registrar los cambios realizados en la base de datos. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdmonPrestamoAfiliado ConsultarPrestamosAfiliado

2.2.2.39 Tarjetas CRC No 39: Administrar egresos.

TARJETA CRC N° 039	
NOMBRE DE LA CLASE	AdministrarEgreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla las opciones de registro, consulta, modificación e eliminación de la información correspondiente a los egresos realizados en la cooperativa. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.40 Tarjetas CRC No 41: Registrar egresos.

TARJETA CRC N° 040	
NOMBRE DE LA CLASE	RegistrarEgreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar los datos necesarios para el registro del egreso como la fecha, la cantidad y el concepto por el cual se realice el egreso y el responsable que autorizo el egreso. • Validar los datos suministrados por el usuario administrador. • Registrar la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarEgreso

2.2.2.41 Tarjetas CRC No 41: Consultar egresos.

TARJETA CRC N° 041	
NOMBRE DE LA CLASE	ConsultarEgreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos • Permitir el ingreso de un parámetro de búsqueda. • Mostrar en pantalla los datos del egreso (La fecha, la cantidad, el concepto por el cual se realizo y el responsable que autorizo el egreso) según el parámetro de búsqueda ingresado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarEgreso

2.2.2.42 Tarjetas CRC No 42: Modificar egresos.

TARJETA CRC N° 042	
NOMBRE DE LA CLASE	ModificarEgreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla los datos de los egresos de la cooperativa. • Permitir la selección de un egreso. • Permitir la modificación de los datos del egreso seleccionado. • Registrar los cambios realizados a la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarEgreso ConsultarEgreso

2.2.2.43 Tarjetas CRC No 43: Eliminar egreso.

TARJETA CRC N° 043	
NOMBRE DE LA CLASE	EliminarEgreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la eliminación del egreso. • Ofrecer la opción de confirmar eliminación. • Registrar los cambios realizados a la información en la base de datos. • Informar sobre el resultado de la operación realizada. 	ConexionDB ValidarDatos AdministrarEgreso ConsultarEgreso

2.2.2.44 Tarjetas CRC No 44: Administrar Ingresos.

TARJETA CRC N° 044	
NOMBRE DE LA CLASE	AdministrarIngreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla las opciones de registro, consulta, modificación e eliminación de la información correspondiente a los ingresos realizados. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.45 Tarjetas CRC No 45: Registrar Ingresos.

TARJETA CRC N° 045	
NOMBRE DE LA CLASE	RegistrarIngreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Solicitar los datos necesarios para el registro del ingreso como fecha, cantidad y concepto por el cual se realice en ingreso. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarIngreso

2.2.2.46 Tarjetas CRC No 46: Consultar Ingresos.

TARJETA CRC N° 046	
NOMBRE DE LA CLASE	ConsultarIngreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de un parámetro de búsqueda. • Mostrar en pantalla los datos del ingreso (fecha el concepto y la cantidad y el responsable que autorizo el egreso) según el parámetro de búsqueda ingresado. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos RegistrarIngreso AdministrarIngreso

2.2.2.47 Tarjetas CRC No 47: Modificar Ingresos.

TARJETA CRC N° 047	
NOMBRE DE LA CLASE	ModificarIngreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. 	ConexionDB

<ul style="list-style-type: none"> • Permitir el ingreso de la fecha que requiere modificar como parámetro de búsqueda. • Mostrar en pantalla los ingresos realizados en esa fecha. • Permitir la selección de un ingreso. • Permitir la modificación de los datos del ingreso. • Registrar los cambios realizados en la base de datos. • Informar sobre el resultado de la operación en curso. 	validarDatos RegistrarIngreso AdministrarIngreso ConsultarIngreso
---	--

2.2.2.48 Tarjetas CRC No 48: Eliminar ingresos.

TARJETA CRC N° 048	
NOMBRE DE LA CLASE	EliminarIngreso
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de la fecha como parámetro de búsqueda. • Mostrar en pantalla los ingresos realizados en esa fecha. • Permitir la selección de un ingreso. • Permitir la eliminación de la información correspondiente a ese ingreso. • Validar los datos suministrados por el usuario. • Registrar los cambios realizados en la base de datos. • Informar sobre el resultado de la operación realizada. 	ConexionDB ValidarDatos RegistrarIngreso AdministrarIngreso ConsultarIngreso

2.2.2.49 Tarjetas CRC No 49: Administrar pagos

TARJETA CRC N° 049	
NOMBRE DE LA CLASE	AdministrarPago
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla un listado de los afiliados • Permitir la selección de un afiliado del listado y ofrecer las opciones de registrar, Modificar, consultar e eliminar la información correspondiente a los pagos de las cuotas de los préstamos. • Registrar los cambios realizados en la base de datos. • Validar los datos suministrados por el usuario. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.50 Tarjetas CRC No 50: Registrar pagos.

TARJETA CRC N° 050	
NOMBRE DE LA CLASE	RegistrarPago
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer la conexión a la base de datos. • Ingresar la cantidad que se recibió del afiliado como pago de la cuota del préstamo. • Validar los datos suministrados por el usuario. • Registrar la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos AdministrarPago

2.2.2.51 Tarjetas CRC No 51: Consultar pagos.

TARJETA CRC N° 051	
NOMBRE DE LA CLASE	ConsultarPago
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla los datos del pago (fechas y cantidades recibidas). • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos RegistrarPago AdministrarPago

2.2.2.52 Tarjetas CRC No 52: Modificar pagos.

TARJETA CRC N° 052	
NOMBRE DE LA CLASE	ModificarPago
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla los datos del pago. • Permitir modificar los datos del pago. • Registrar los cambios realizados a la información en la base de datos. • Informar sobre el resultado de la operación en curso. 	ConexionDB ValidarDatos RegistrarPago AdministrarPago ConsultarPago

2.2.2.53 Tarjetas CRC No 53: Reporte de información del afiliado

TARJETA CRC N° 053	
NOMBRE DE LA CLASE	ReporteAfiliados
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. 	ConexionDB

<ul style="list-style-type: none"> • Permite la selección de varios criterios para la configuración del reporte de la información de afiliación de los afiliados. • Validar los datos suministrados por el usuario. • Mostrar en pantalla los datos de afiliación según la búsqueda realizada por el usuario. • Mostrar al usuario la opción de imprimir, guardar en formato pdf, el reporte seleccionado. • Informar el estado de la operación en curso. 	ValidarDatos
--	---------------------

2.2.2.54 Tarjetas CRC No 54: Reporte de aportes de afiliados.

TARJETA CRC N° 056	
NOMBRE DE LA CLASE	ReportAportesAfiliados
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla la información de los aportes realizados por los afiliados • Mostrar al usuario la opción de imprimir, guardar en formato pdf, el reporte. • Informar el estado de la operación en curso. 	ConexionDB ValidarDatos ConsultarAfiliado ConsultarAportes AdmonReportAportes

2.2.2.55 Tarjetas CRC No 55: Reporte de préstamos.

TARJETA CRC N° 058	
NOMBRE DE LA CLASE	AdmonReportprestamos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Mostrar en pantalla la información de los prestamos realizados por los afiliados • Permitir la selección de un tipo de reporte. • Mostrar en pantalla la opción que seleccionó el usuario. • Informar el estado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.56 Tarjetas CRC No 56: Reporte de pagos.

TARJETA CRC N° 056	
NOMBRE DE LA CLASE	AdmonReportPagos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. 	ConexionDB

<ul style="list-style-type: none"> • Permite mostrar en pantalla la información de los pagos de las cuotas de los préstamos realizados por los afiliados • Permitir la selección de un tipo de reporte. • Mostrar en pantalla la opción que seleccionó el usuario. • Informar el estado de la operación en curso. 	ValidarDatos
---	---------------------

2.2.2.57 Tarjetas CRC No 57: Reporte de egresos.

TARJETA CRC N° 057	
NOMBRE DE LA CLASE	ReportEgresos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos • Permitir mostrar en pantalla la información de los egresos de la cooperativa • Mostrar en pantalla la información requerida. • Mostrar al usuario la opción de imprimir, guardar en formato pdf, el reporte. • Informar el estado de la operación en curso. 	ConexionDB ValidarDatos ConsultarEgresos

2.2.2.58 Tarjetas CRC No 58: Reporte de ingresos

TARJETA CRC N° 058	
NOMBRE DE LA CLASE	ReportIngresos
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos • Permitir mostrar en pantalla la información de los egresos de la cooperativa. • Mostrar en pantalla la información requerida. • Mostrar al usuario la opción de imprimir, guardar en formato pdf, el reporte. • Informar el estado de la operación en curso. 	ConexionDB ValidarDatos ConsultarIngresos

2.2.2.59 Tarjetas CRC No 59: Reporte de usuarios

TARJETA CRC N° 059	
NOMBRE DE LA CLASE	ReportUsuarios
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos 	ConexionDB

<ul style="list-style-type: none"> • Permitir mostrar en pantalla la información de los usuarios de la aplicación. • Mostrar en pantalla la información requerida. • Mostrar al usuario la opción de imprimir, guardar en formato pdf, el reporte. • Informar el estado de la operación en curso. 	ValidarDatos ConsultarEgresos
---	--

2.2.2.60 Tarjetas CRC No 60: Reporte de Tipo de aportes

TARJETA CRC N° 060	
NOMBRE DE LA CLASE	ReportTipoaporte
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir mostrar en pantalla la información de los tipos de aporte que se pueden realizar en la cooperativa. • Mostrar en pantalla la información requerida. • Mostrar al usuario la opción de imprimir, guardar en formato pdf, el reporte. 	ConexionDB

2.2.2.61 Tarjetas CRC No 61: Reporte de Tipo de préstamos

TARJETA CRC N° 061	
NOMBRE DE LA CLASE	ReportTipoprestamo
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir mostrar en pantalla la información de los tipos de préstamo que se pueden realizar en la cooperativa. • Mostrar en pantalla la información requerida. • Mostrar al usuario la opción de imprimir, guardar en formato pdf, el reporte. 	ConexionDB

2.2.2.62 Tarjetas CRC N° 62: Bitácora.

TARJETA CRC N° 064	
NOMBRE DE LA CLASE	Bitacora
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Debe registrar los movimientos realizados en cada uno de los módulos. 	ConexionDB

<ul style="list-style-type: none"> • Mostrar en pantalla todos los movimientos registrados. • Permitir la elección de la unidad donde desea guardar el registro de los movimientos realizados. • Informar el estado de la operación en curso. 	
--	--

2.2.2.63 Tarjetas CRC No 63: Reporte Bitácora

TARJETA CRC N° 063	
NOMBRE DE LA CLASE	CopiaBd
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir el ingreso de las fechas como parámetro de búsqueda para generar el reporte de la bitácora de la aplicación • Permitir mostrar en pantalla todos los movimientos registrados según el parámetro de búsqueda ingresado • Permitir la elección de la unidad donde desea guardar el registro de los movimientos realizados. • Informar el estado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.64 Tarjetas CRC No 64: Crear las copias de las bases de datos.

TARJETA CRC N° 064	
NOMBRE DE LA CLASE	CopiaBd
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la realización de copias de respaldo de la base de datos. • Permitir la elección de la unidad donde desea guardar la copia. • Informar el estado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.65 Tarjetas CRC No 65: Restaurar las copias de las bases de datos.

TARJETA CRC N° 065	
NOMBRE DE LA CLASE	ImportarBd
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Presentar al usuario en pantalla las opciones que le permitan escoger el tipo de base de datos que desea importar. 	ConexionDB ValidarDatos

<ul style="list-style-type: none"> • Mostrar al usuario pantalla las opciones necesarias para seleccionar un archivo de texto con las especificaciones del tipo de base de datos seleccionado. • Cargar el archivo indicado por el usuario y validar que su formato corresponda con el tipo de base de datos selecciona. • Separar los registros que se encuentren en el archivo, en registros válidos e inválidos. • Mostrar los registros válidos e inválidos en listados diferentes. • informar sobre el resultado de la operación en curso. 	
--	--

2.2.2.66 Tarjeta CRC No 66: Importar bases de datos.

TARJETA CRC N° 066	
NOMBRE DE LA CLASE	RestCopiaBd
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Permitir la eliminación actual de la base de datos. • Ofrecer al usuario la opción de restaurar los datos de la copia de seguridad de la base de datos. • Informar el estado de la operación en curso. 	ConexionDB ValidarDatos

2.2.2.67 Tarjetas CRC No 67: Exportar las bases de datos.

TARJETA CRC N° 068	
NOMBRE DE LA CLASE	ExportarBd
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> • Establecer conexión con la base de datos. • Presentar al usuario en pantalla las opciones que le permitan escoger el tipo de base de datos que desea exportar. • Mostrar una opción que permita visualizar en pantalla un listado de los registros deseados, según el tipo de base de datos a exportar y según el criterio de búsqueda aplicado. • Exportar a un archivo de Excel los registros listados en pantalla. • Exportar a un archivo de texto plano los registros 	ConexionDB ValidarDatos

listados en pantalla.	
-----------------------	--

2.3 FASE DE DESARROLLO

Entre los elementos más importantes que menciona XP referente al desarrollo, es la realización de las pruebas unitarias, que como se dijo anteriormente es una actividad que debe realizarse antes y durante la implementación de un módulo. Terminada la implementación del mismo las pruebas unitarias deben ser todas correctas, asegurando así el buen funcionamiento del módulo, es necesario que todas las pruebas sean correctas para poder integrar el módulo nuevo al sistema existente

2.3.1 Unidades de Prueba. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema, la producción de código está dirigida por las pruebas unitarias.

2.3.1.1 Unidad de prueba No 1: Conexión con la base de datos

UNIDAD DE PRUEBA N° 001	
Nombre de la Clase	ConexionDB
Código Tarjeta CRC	001
¿Cumple la clase con los siguientes requisitos?	
• ¿Permite la clase establecer la conexión con la base de datos?	Si
• ¿Informa la clase sobre el resultado de la conexión con la base de datos?	Si
• ¿Implementa la clase un método que permita ejecutar instrucciones SQL de tipo Select, Insert, Update y Delete?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación con un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.2 Unidad de prueba No 2: Validación de los datos

UNIDAD DE PRUEBA N° 002	
Nombre de la Clase	ValidarDatos
Código Tarjeta CRC	002
¿Cumple la clase con los siguientes requisitos?	
• ¿Permite la clase establecer la conexión con la base de datos?	Si

• ¿Informa la clase sobre el resultado de la conexión con la base de datos?	Si
• ¿Implementa la clase un método que permita Validar la información ingresada en el formulario que corresponda con las exigencias de la base de datos?	Si
• ¿Implementa la clase un método que permita Verificar si los datos ingresados por el usuario corresponden a los datos registrados en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación en la base de datos por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.3 Unidad de prueba No 3: Página de inicio

UNIDAD DE PRUEBA N° 003	
Nombre de la Clase	PáginaInicio
Código Tarjeta CRC	003
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para el ingreso de la información de inicio de sesión?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Da paso a la ejecución de la página Principal en caso que todo esté correcto?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.4 Unidad de prueba No 4: Administrar usuario

UNIDAD DE PRUEBA N° 004	
Nombre de la Clase	AdministrarUsuario
Código Tarjeta CRC	004
¿Cumple la clase con los siguientes requisitos?	

• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método que permita extraer de la base de datos un listado de los usuarios registrados y mostrarlos en pantalla?	Si
• ¿Instancia un objeto de la clase CrearCuentaUsuario que permita acceder a la sección que posibilita agregar un nuevo usuario?	Si
• ¿Instancia un objeto de la clase EditarCuentaUsuario que permita acceder a la sección que posibilita modificar un usuario?	Si
• ¿Instancia un objeto de la clase EliminarCuentaUsuario que permita acceder a la sección que posibilita eliminar al usuario de la aplicación?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.5 Unidad de prueba No 5: Crear usuario

UNIDAD DE PRUEBA N° 005	
Nombre de la Clase	RegUsuario
Código Tarjeta CRC	005
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si

• ¿Realiza una instancia de la clase ActivarCuentaUsuario con la que pueda establecer el estado del usuario ingresado?	Si
• ¿Contiene un método para el ingreso de los datos del nuevo usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.6 Unidad de prueba No 6: Consultar usuario

UNIDAD DE PRUEBA N° 006	
Nombre de la Clase	ConsultarUsuario
Código Tarjeta CRC	006
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de un parámetro de búsqueda?	Si
• ¿Contiene un método para mostrar en pantalla la información de los usuarios?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.7 Unidad de prueba No 7: Editar usuario

UNIDAD DE PRUEBA N° 007	
Nombre de la Clase	ModificarCuentaUsuario
Código Tarjeta CRC	007
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda	Si

realizar las gestiones sobre la base de datos?	
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Realiza una instancia de la clase ActivarCuentaUsuario con la que pueda establecer el estado del usuario ingresado?	Si
• ¿Contiene un método para la edición de los datos del usuario ya registrado?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.8 Unidad de prueba No 8: Activar cuenta de usuario

UNIDAD DE PRUEBA N° 008	
Nombre de la Clase	ActivarCuentaUsuario
Código Tarjeta CRC	008
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para cambiar el estado (activo e inactivo) del usuario ya registrado?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.9 Unidad de prueba No 9: Eliminar usuario

UNIDAD DE PRUEBA N° 009	
Nombre de la Clase	EliminarUsuario
Código Tarjeta CRC	009
¿Cumple la clase con los siguientes requisitos?	

• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Realiza una instancia de la clase ActivarCuentaUsuario con la que pueda establecer el estado del usuario ingresado?	Si
• ¿Contiene un método para la eliminación de los datos del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.10 Unidad de prueba No 10: Administrar afiliado

UNIDAD DE PRUEBA N° 010	
Nombre de la Clase	AdministrarAfiliado
Código Tarjeta CRC	010
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método que permita extraer de la base de datos un listado de los usuarios registrados y mostrarlos en pantalla?	Si
• ¿Instancia un objeto de la clase RegAfiliado que permita acceder a la sección que posibilita agregar un nuevo afiliados?	Si
• ¿Instancia un objeto de la clase ConsultarAfiliado que permita acceder a la sección que posibilita realizar búsqueda de los afiliados?	Si
• ¿Instancia un objeto de la clase ModificarAfiliado que permita acceder a la sección que posibilita modificarlos datos de un afiliado?	Si

• ¿Instancia un objeto de la clase EliminarAfiliado que permita acceder a la sección que posibilita eliminar los datos del afiliado de la aplicación?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.11 Unidad de prueba No 11: Crear afiliado

UNIDAD DE PRUEBA N° 011	
Nombre de la Clase	RegAfiliado
Código Tarjeta CRC	011
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Realiza una instancia de la clase FormInfGeneral con la que pueda ingresar la información general del Afiliado?	Si
• ¿Realiza una instancia de la clase FormInfEconomico con la que pueda ingresar la información económica del Afiliado?	Si
• ¿Realiza una instancia de la clase FormInfFinanciera con la que pueda ingresar la información Financiera del Afiliado?	Si
• ¿Realiza una instancia de la clase FormInfBeneficiarios con la que pueda ingresar la información de los beneficiarios del Afiliado?	Si
• ¿Realiza una instancia de la clase FormInfTutores con la que pueda ingresar la información de los tutores asignados por el Afiliado?	Si
• ¿Realiza una instancia de la clase FormInfCursos con la que pueda ingresar la información de los cursos cooperativos realizados por el Afiliado?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.12 Unidad de prueba No 12: Formulario de información general

UNIDAD DE PRUEBA N° 012	
Nombre de la Clase	FormInfGeneral
Código Tarjeta CRC	012
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Realiza una instancia de la clase ActivarCuentaUsuario con la que pueda establecer el estado del usuario ingresado?	Si
• ¿Contiene un método para la edición de los datos del Afiliado?	Si
• ¿La clase permite el registro de la información general del afiliado en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.13 Unidad de prueba No 13: Formulario de información económica

UNIDAD DE PRUEBA N° 013	
Nombre de la Clase	FormInfEconomica
Código Tarjeta CRC	013
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para la edición de los datos de información	Si

económica del afiliado?	
• ¿La clase permite el registro de la información económica del afiliado en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.14 Unidad de prueba No 14: Formulario de información financiera

UNIDAD DE PRUEBA N° 014	
Nombre de la Clase	FormInfFinanciera
Código Tarjeta CRC	014
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para la edición de los datos de información financiera del afiliado?	Si
• ¿La clase permite el registro de la información financiera del afiliado en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.15 Unidad de prueba No 15: Formulario de información de de beneficiarios

UNIDAD DE PRUEBA N° 015	
Nombre de la Clase	FormInfBeneficiarios
Código Tarjeta CRC	015
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si

• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para la edición de los datos del beneficiario(s) asignado por el afiliado?	Si
• ¿La clase permite el registro de la información del beneficiario(s) asignado por el afiliado en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.16 Unidad de prueba No 16:Formulario información de Tutores

UNIDAD DE PRUEBA N° 016	
Nombre de la Clase	FormInfTutores
Código Tarjeta CRC	016
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para la edición de los datos del tutor asignado por el afiliado?	Si
• ¿La clase permite el registro de la información del tutor asignado por el afiliado en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.17 Unidad de prueba No 17: Consultar afiliado

UNIDAD DE PRUEBA N° 017	
Nombre de la Clase	ConsultarAfiliado
Código Tarjeta CRC	017
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de los parámetros de búsqueda de información?	Si
• ¿Contiene un método que permite listar los datos del afiliado de acuerdo a los parámetros de búsqueda ingresados?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.18 Unidad de prueba No 18: Modificar afiliado

UNIDAD DE PRUEBA N° 018	
Nombre de la Clase	ModificarAfiliado
Código Tarjeta CRC	018
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para la edición de la información del afiliado?	Si
• ¿La clase permite el registro de la información del afiliado en la	Si

base de datos?	
<ul style="list-style-type: none"> ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla? 	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.19 Unidad de prueba No 19: Eliminar afiliado

UNIDAD DE PRUEBA N° 019	
Nombre de la Clase	EliminarAfiliado
Código Tarjeta CRC	019
¿Cumple la clase con los siguientes requisitos?	
<ul style="list-style-type: none"> ¿Implementa la clase las características de una página web? 	Si
<ul style="list-style-type: none"> ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación? 	Si
<ul style="list-style-type: none"> ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos? 	Si
<ul style="list-style-type: none"> ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario? 	Si
<ul style="list-style-type: none"> ¿Contiene un método para la eliminación de la información del afiliado? 	Si
<ul style="list-style-type: none"> ¿La clase permite la actualización en la base de datos? 	Si
<ul style="list-style-type: none"> ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla? 	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.20 Unidad de prueba No 20: Administrar tipos de aporte

UNIDAD DE PRUEBA N° 020	
Nombre de la Clase	AdmonTiposDeAportes
Código Tarjeta CRC	020
¿Cumple la clase con los siguientes requisitos?	
<ul style="list-style-type: none"> ¿Implementa la clase las características de una página web? 	Si
<ul style="list-style-type: none"> ¿Incluye los componentes gráficos necesarios para la administración de los afiliados registrados en la aplicación? 	Si
<ul style="list-style-type: none"> ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos? 	Si

• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método que permita extraer de la base de datos un listado de los usuarios registrados y mostrarlos en pantalla?	Si
• ¿Instancia un objeto de la clase CrearTipoDeAportes que permita acceder a la sección que posibilita agregar un nuevo afiliados?	Si
• ¿Instancia un objeto de la clase ConsultarTipoDeAportes que permita acceder a la sección que posibilita realizar búsqueda de los afiliados?	
• ¿Instancia un objeto de la clase EditarTipoDeAportes que permita acceder a la sección que posibilita modificarlos datos de un afiliado?	Si
• ¿Instancia un objeto de la clase EliminarTipoDeAportes que permita acceder a la sección que posibilita eliminar los datos del afiliado de la aplicación?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.21 Unidad de prueba No 21: Crear tipos de aporte

UNIDAD DE PRUEBA N° 021	
Nombre de la Clase	RegTipoDeAportes
Código Tarjeta CRC	021
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Implementa las características de los tipos de aporte que se pueden realizar en la cooperativa?	Si
• ¿La clase permite establecer la los topes de los porcentajes mínimos y máximos de cada tipo de aporte en base al salario	Si

mínimo mensual legal vigente	
• ¿La clase permite registrar la forma de recepción del aporte (por descuento de nomina, por ventanilla)?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.22 Unidad de prueba No 22: Consultar tipos de aporte

UNIDAD DE PRUEBA N° 022	
Nombre de la Clase	ConsultarInfAportes
Código Tarjeta CRC	022
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite mostrar en pantalla los tipos de aportes que se pueden realizar en la cooperativa?	Si
• ¿La clase permite registrar la forma de recepción del aporte (por descuento de nomina, por ventanilla)?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.23 Unidad de prueba No 23: Modificar tipos de aporte

UNIDAD DE PRUEBA N° 023	
Nombre de la Clase	ModificarInfAportes
Código Tarjeta CRC	023
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si

• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite mostrar en pantalla los tipos de aportes que se pueden realizar en la cooperativa?	Si
• ¿La clase Contiene un método para la edición de los datos de tipos de aportes que se pueden realizar en la cooperativa?	Si
• ¿La clase permite la actualización de la información de los tipos de aportes en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.24 Unidad de prueba No 24: Eliminar tipos de aporte

UNIDAD DE PRUEBA N° 024	
Nombre de la Clase	EliminarInfAportes
Código Tarjeta CRC	024
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite mostrar en pantalla los tipos de aportes que se pueden realizar en la cooperativa?	Si
• ¿La clase Contiene un método para la eliminación de los datos de tipos de aportes que se pueden realizar en la cooperativa?	Si
• ¿La clase permite la actualización de la información de los tipos de aportes en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.25 Unidad de prueba No 25: Administrar aportes del afiliado

UNIDAD DE PRUEBA N° 025	
Nombre de la Clase	AdmonAportesAfiliado
Código Tarjeta CRC	025
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método que permita extraer de la base de datos un listado de los Aportes registrados y mostrarlos en pantalla?	Si
• ¿Instancia un objeto de la clase RegAporteAfiliado que permita acceder a la sección que posibilita agregar un nuevo Aporte realizado por el afiliados?	Si
• ¿Instancia un objeto de la clase ConsultarAporteAfiliado que permita acceder a la sección que posibilita realizar búsqueda de los Aporte realizado por el afiliados?	
• ¿Instancia un objeto de la clase EditarAporteAfiliado que permita acceder a la sección que posibilita modificarlos datos de Aporte realizado por un afiliado?	Si
• ¿Instancia un objeto de la clase EliminarAporteAfiliado que permita acceder a la sección que posibilita eliminar los datos de los aportes realizados por el afiliado a la aplicación?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.26 Unidad de prueba No 26: Registrar aportes del afiliado

UNIDAD DE PRUEBA N° 026	
Nombre de la Clase	RegAportesAfiliado
Código Tarjeta CRC	026

¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Realiza una instancia de la clase CrearTipoDeAporte con la que permita escoger el tipo de aporte a realizar el afiliado?	Si
• ¿La clase implementa un método que permite el ingreso del valor del porcentaje en base al SMLV del aporte o los aportes que realizara el afiliado?	Si
• ¿La clase implementa un método que permite establecer las fechas que se realizara el recaudo del aporte o de los tipos de aportes?	Si
• ¿La clase implementa un método que permite establecer la forma de recepción del aporte?	Si
• ¿La clase implementa un método que Informe sobre el valor total de la cantidad a aportar por cada tipo de aporte?	Si
• Registrar la información en la base de datos	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.27 Unidad de prueba No 27: Consultar aportes del afiliado

UNIDAD DE PRUEBA N° 027	
Nombre de la Clase	ConsultarAportesAfiliado
Código Tarjeta CRC	027
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda	Si

realizar las validaciones correspondientes a los datos ingresados por el usuario?	
<ul style="list-style-type: none"> ¿La clase permite el ingreso de las fechas como parámetro de búsqueda para realizar la consulta del valor del aporte o los aportes realizados por el afiliado? 	Si
<ul style="list-style-type: none"> ¿La clase permite mostrar en pantalla la información requerida? 	Si
<ul style="list-style-type: none"> ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla? 	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.28 Unidad de prueba No 28: Modificar aportes del afiliado

UNIDAD DE PRUEBA N° 028	
Nombre de la Clase	ModificarAportesAfiliado
Código Tarjeta CRC	028
¿Cumple la clase con los siguientes requisitos?	
<ul style="list-style-type: none"> ¿Implementa la clase las características de una página web? 	Si
<ul style="list-style-type: none"> ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación? 	Si
<ul style="list-style-type: none"> ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos? 	Si
<ul style="list-style-type: none"> ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario? 	Si
<ul style="list-style-type: none"> ¿La clase permite mostrar en pantalla los tipos de aportes registrados al afiliado? 	Si
<ul style="list-style-type: none"> ¿La clase Permitir la selección de un tipo de aporte? 	Si
<ul style="list-style-type: none"> ¿La clase Permitir la modificación de la información de los tipos de aporte? 	Si
<ul style="list-style-type: none"> ¿La clase Contiene un método para la edición de los datos de tipos de aportes que se pueden realizar en la cooperativa? 	Si
<ul style="list-style-type: none"> ¿La clase permite la actualización de las modificaciones realizadas en la base de datos? 	Si
<ul style="list-style-type: none"> ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla? 	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.29 Unidad de prueba No 29: Eliminar aportes del afiliado

UNIDAD DE PRUEBA N° 029	
Nombre de la Clase	EliminarAportesDelAfiliado
Código Tarjeta CRC	029
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para la eliminación de la información del aporte realizado por el afiliado?	Si
• ¿La clase permite mostrar en pantalla la información de los tipos de aportes realizados por el afiliado?	Si
• ¿La clase ofrece la opción de confirmar eliminación?	
• ¿La clase permite la actualización en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.30 Unidad de prueba No 30: Administrar tipos de préstamos

UNIDAD DE PRUEBA N° 030	
Nombre de la Clase	AdmonTiposDePrestamos
Código Tarjeta CRC	030
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si

• ¿Contiene un método que permita extraer de la base de datos un listado de los afiliados con préstamos vigentes y mostrarlos en pantalla?	Si
• ¿Instancia un objeto de la clase CrearPrestamo que permita acceder a la sección que posibilita agregar un nuevo tipo de préstamo ofrecido a los afiliados?	Si
• ¿Instancia un objeto de la clase ConsultarInfPrestamo que permita acceder a la sección que posibilita realizar búsqueda de los tipos préstamos ofrecidos a los afiliados?	
• ¿Instancia un objeto de la clase ModificarInfPrestamo que permita acceder a la sección que posibilita modificarlos tipos de préstamos ofrecidos a los afiliados?	Si
• ¿Instancia un objeto de la clase EliminarInfPrestamo que permita acceder a la sección que posibilita eliminar los tipos de préstamos ofrecidos a los afiliados?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.31 Unidad de prueba No 31: Crear tipos de préstamos

UNIDAD DE PRUEBA N° 031	
Nombre de la Clase	RegInfPréstamos
Código Tarjeta CRC	031
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase Permite el registro de los tipos de préstamo que se pueden realizar en la cooperativa?	Si
• ¿La clase permite registrar el monto de cada tipo de préstamo?	Si
• ¿La clase permite registrar el plazo para cada tipo de préstamo?	Si
• ¿La clase permite el registro del porcentaje de interés corriente y el	Si

interés por mora parta cada tipo de préstamo?	
• ¿La clase permite la actualización de la información de los tipos de préstamos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.32 Unidad de prueba No 32: Consultar información de los tipos de préstamos

UNIDAD DE PRUEBA N° 032	
Nombre de la Clase	ConsultarInfPrestamos
Código Tarjeta CRC	032
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite mostrar en pantalla los tipos de préstamos que se pueden realizar en la cooperativa?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.33 Unidad de prueba No 33: Modificar información de los tipos de préstamos

UNIDAD DE PRUEBA N° 033	
Nombre de la Clase	ModificarInfPrestamos
Código Tarjeta CRC	033
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si

• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite mostrar en pantalla los tipos de préstamos que se pueden realizar en la cooperativa?	Si
• ¿La clase Contiene un método para la edición de los datos de tipos de préstamos que se pueden realizar en la cooperativa?	Si
• ¿La clase permite la actualización de la información de los tipos de préstamos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.34 Unidad de prueba No 34: Eliminar información de los tipos de préstamos

UNIDAD DE PRUEBA N° 034	
Nombre de la Clase	EliminarInfPrestamos
Código Tarjeta CRC	034
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite mostrar en pantalla los tipos de préstamos que se pueden realizar en la cooperativa?	Si
• ¿La clase Contiene un método para la eliminación de los datos de tipos de préstamos que se pueden realizar en la cooperativa?	Si
• ¿La clase permite realizar la confirmación de eliminación a un tipo de préstamo?	Si
• ¿La clase permite la actualización de la información de los tipos de préstamos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de	Si

la operación por medio de un mensaje en pantalla?	
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.35 Unidad de prueba No 35: Administrar préstamos del afiliado

UNIDAD DE PRUEBA N° 035	
Nombre de la Clase	AdmonPrestamoAfiliado
Código Tarjeta CRC	035
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método que permita extraer de la base de datos un listado de los Afiliados y mostrarlos en pantalla?	Si
• ¿Instancia un objeto de la clase RegPréstamoAfiliado que permita acceder a la sección que posibilita agregar un nuevo préstamo realizado por el afiliado?	Si
• ¿Instancia un objeto de la clase ConsultarPréstamoAfiliado que permita acceder a la sección que posibilita realizar búsqueda de los préstamos realizado por los afiliados?	Si
• ¿Instancia un objeto de la clase EditarPréstamoAfiliado que permita acceder a la sección que posibilita modificarlos datos del préstamo(s) realizado por un afiliado?	Si
• ¿Instancia un objeto de la clase EliminarPréstamoAfiliado que permita acceder a la sección que posibilita eliminar los datos de los préstamos cancelados por el afiliado?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.36 Unidad de prueba No 36: Registrar préstamo del afiliado

UNIDAD DE PRUEBA N° 036

Nombre de la Clase	RegPrestamoAfiliado	
Código Tarjeta CRC	036	
¿Cumple la clase con los siguientes requisitos?		
• ¿Implementa la clase las características de una página web?		Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?		Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?		Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?		Si
• ¿Realiza una instancia de la clase CrearTipoPréstamo con la que permita escoger el tipo de préstamo a realizar el afiliado?		Si
• ¿La clase permite establecer la fecha de aprobación del préstamo?		Si
• ¿La clase permite seleccionar el monto y plazo del préstamo?		Si
• ¿La clase permite establecer las fechas que se realizara el pago de las cuotas del préstamo?		Si
• ¿La clase permite establecer la forma de recepción del pago de las cuotas del préstamo?		Si
• ¿La clase implementa un método que informa la cantidad del valor prestado?		Si
• ¿La clase implementa un método que informa sobre el valor de las cuotas a pagar con interés corriente y por mora?		Si
• ¿La clase permite Registrar la información en la base de datos?		Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?		Si
Calificación a la Prueba	Implementación correcta de la clase	

2.3.1.37 Unidad de prueba No 37: Consultar préstamo del afiliado

UNIDAD DE PRUEBA N° 037		
Nombre de la Clase	ConsultarPréstamosAfiliado	
Código Tarjeta CRC	037	
¿Cumple la clase con los siguientes requisitos?		
• ¿Implementa la clase las características de una página web?		Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?		Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda		Si

realizar las gestiones sobre la base de datos?	
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase muestra en pantalla la información correspondiente al registro del préstamo (tipo de préstamo, Monto, Plazo, número de cuotas a pagar, la fecha de aprobación del préstamo, la cantidad total del préstamo) efectuado por el afiliado en la cooperativa?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.38 Unidad de prueba No 38: Eliminar préstamos del afiliado

UNIDAD DE PRUEBA N° 038	
Nombre de la Clase	EliminarPrestamo
Código Tarjeta CRC	038
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método para la eliminación del registro del préstamo realizado por el Afiliado?	Si
• ¿La clase permite la actualización de los datos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.39 Unidad de prueba No 39: Administrar egresos

UNIDAD DE PRUEBA N° 039

Nombre de la Clase	AdministrarEgreso
Código Tarjeta CRC	039
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Contiene un método que permita extraer de la base de datos un listado de los afiliados con préstamos vigentes y mostrarlos en pantalla?	Si
• ¿Instancia un objeto de la clase RegistrarEgreso 5") que permita acceder a la sección que posibilita agregar un nuevo egreso?	Si
• ¿Instancia un objeto de la clase ConsultarEgreso 6") que permita acceder a la sección que posibilita realizar búsqueda de los egresos ejecutados en la cooperativa?	Si
• ¿Instancia un objeto de la clase ModificarEgreso 6") que permita acceder a la sección que posibilita modificar los egresos ejecutados en la cooperativa?	Si
• ¿Instancia un objeto de la clase EliminarEgreso 7") que permita acceder a la sección que posibilita eliminar los egresos ejecutados en la cooperativa?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.40 Unidad de prueba No 40: Registrar egreso

UNIDAD DE PRUEBA N° 040	
Nombre de la Clase	RegistrarEgreso
Código Tarjeta CRC	040
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la	Si

administración de los usuarios registrados en la aplicación?	
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase Solicita los datos necesarios para el registro del egreso como la fecha, la cantidad y el concepto por el cual se realizo el egreso y el responsable que autorizo el egreso?	Si
• ¿La clase permite la actualización en la base de datos de la información de los egresos ejecutados en la cooperativa?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.41 Unidad de prueba No 41: Consultar egreso

UNIDAD DE PRUEBA N° 041	
Nombre de la Clase	ConsultarEgreso
Código Tarjeta CRC	041
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de la fecha que se desee consultar como parámetro de búsqueda?	
• ¿La clase permite mostrar en pantalla los datos del egreso (fecha, cantidad, concepto por el cual se realizo y el responsable que autorizo) realizado en la cooperativa?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.42 Unidad de prueba No 42: Modificar egreso

UNIDAD DE PRUEBA N° 042	
Nombre de la Clase	ModificarEgreso
Código Tarjeta CRC	042
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de la fecha como parámetro de búsqueda?	Si
• ¿La clase permite mostrar en pantalla los egresos realizados en esa fecha con su respectiva información?	Si
• ¿La clase permite seleccionar un egreso realizado resultado de la búsqueda anterior?	Si
• ¿La clase Contiene un método para la modificación de los datos del egreso realizado en la cooperativa?	Si
• ¿La clase permite la actualización de la información de egresos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.43 Unidad de prueba No 43: Eliminar egreso

UNIDAD DE PRUEBA N° 043	
Nombre de la Clase	EliminarEgreso
Código Tarjeta CRC	043
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si

• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de la fecha como parámetro de búsqueda?	Si
• ¿La clase permite mostrar en pantalla los egresos realizados en esa fecha con su respectiva información?	Si
• ¿La clase permite seleccionar un egreso realizado resultado de la búsqueda anterior?	Si
• ¿La clase Contiene un método para la eliminación de los datos del egreso realizado en la cooperativa?	Si
• ¿La clase permite la actualización de la información de egresos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.44 Unidad de prueba No 44: Administrar ingreso

UNIDAD DE PRUEBA N° 044	
Nombre de la Clase	AdministrarIngreso
Código Tarjeta CRC	044
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Instancia un objeto de la clase RegistrarIngreso 5”) que permita acceder a la sección que posibilita agregar un nuevo Ingreso?	Si
• ¿Instancia un objeto de la clase ConsultarIngreso 6”) que permita acceder a la sección que posibilita realizar búsqueda de los Ingresos ejecutados en la cooperativa?	

• ¿Instancia un objeto de la clase ModificarIngreso 6”) que permita acceder a la sección que posibilita modificar los ingresos ejecutados en la cooperativa?	Si
• ¿Instancia un objeto de la clase EliminarIngreso 7”) que permita acceder a la sección que posibilita eliminar los Ingresos ejecutados en la cooperativa?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.45 Unidad de prueba No 45: Registrar ingreso

UNIDAD DE PRUEBA N° 045	
Nombre de la Clase	RegistrarIngreso
Código Tarjeta CRC	045
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase Solicita los datos necesarios para el registro del Ingreso como la fecha, la cantidad y el concepto por el cual se realizo el ingreso?	Si
• ¿La clase permite la actualización en la base de datos de la información de los ingresos ejecutados en la cooperativa?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.46 Unidad de prueba No 46: Consultar ingreso

UNIDAD DE PRUEBA N° 046	
Nombre de la Clase	ConsultarIngreso

Código Tarjeta CRC	046
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de la fecha que se desee consultar como parámetro de búsqueda?	
• ¿La clase permite mostrar en pantalla los datos del ingreso (fecha, cantidad, concepto por el cual se realizo) realizado en la cooperativa?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.47 Unidad de prueba No 49: Modificar ingreso

UNIDAD DE PRUEBA N° 047	
Nombre de la Clase	ModificarIngreso
Código Tarjeta CRC	047
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de la fecha como parámetro de búsqueda?	Si
• ¿La clase permite mostrar en pantalla mostrar en pantalla los ingresos realizados en esa fecha con su respectiva información?	Si

• ¿La clase permite seleccionar un ingreso realizado resultado de la búsqueda anterior?	Si
• ¿La clase Contiene un método para la modificación de los datos del ingreso realizado en la cooperativa?	Si
• ¿La clase permite la actualización de la información de ingresos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.48 Unidad de prueba No 48: Eliminar ingreso

UNIDAD DE PRUEBA N° 048	
Nombre de la Clase	EliminarIngreso
Código Tarjeta CRC	048
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso de la fecha como parámetro de búsqueda?	Si
• ¿La clase permite mostrar en pantalla los ingresos realizados en esa fecha con su respectiva información?	Si
• ¿La clase permite seleccionar un ingreso realizado resultado de la búsqueda anterior?	Si
• ¿La clase Contiene un método para la eliminación de los datos del ingreso realizado en la cooperativa?	Si
• ¿La clase permite la actualización de la información de ingresos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.49 Unidad de prueba No 49: Administrar pago

UNIDAD DE PRUEBA N° 049	
Nombre de la Clase	AdministrarPago
Código Tarjeta CRC	051
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los pagos realizados por el afiliado?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿Instancia un objeto de la clase RegistrarPagos que permita acceder a la sección que posibilita agregar un nuevo pago?	Si
• ¿Instancia un objeto de la clase ConsultarPagos que permita acceder a la sección que posibilita realizar búsqueda de los de los pagos realizados por los afiliados?	
• ¿Instancia un objeto de la clase ModificarPagos que permita acceder a la sección que posibilita modificar los pagos realizados por los afiliados?	Si
• ¿Instancia un objeto de la clase EliminarPagos que permita acceder a la sección que posibilita eliminar los pagos realizados por los afiliados?	Si
• ¿La clase permite estar a la espera de la solicitud del usuario?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.50 Unidad de prueba No 50: Registrar pago

UNIDAD DE PRUEBA N° 050	
Nombre de la Clase	RegistrarPago
Código Tarjeta CRC	050
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si

• ¿Incluye los componentes gráficos necesarios para el registro de los pagos realizados por el afiliado?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase Solicita los datos necesarios para el registro del pago como cedula o nombre del afiliado?	Si
• ¿La clase permite la actualización en la base de datos de la información del pago realizado por el afiliado?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.51 Unidad de prueba No 51: Consultar pago

UNIDAD DE PRUEBA N° 051	
Nombre de la Clase	ConsultarPago
Código Tarjeta CRC	051
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los usuarios registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite el ingreso del número de identificación o el nombre del afiliado?	Si
• ¿La clase permite mostrar en pantalla los pagos realizados por el afiliado?	Si
• ¿La clase permite seleccionar un pago y mostrar en pantalla los datos del pago como la fecha y cantidad?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.52 Unidad de prueba No 52: Modificar pago

UNIDAD DE PRUEBA N° 052	
Nombre de la Clase	ModificarPago
Código Tarjeta CRC	052
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la modificación de los datos del pago realizado por el afiliado?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la seleccion del número de identificación o el nombre del afiliado como parámetro de búsqueda?	Si
• ¿La clase permite mostrar en pantalla los pagos realizados por ese afiliado?	Si
• ¿La clase permite seleccionar un pago realizado por el afiliado?	Si
• ¿La clase Contiene un método para la modificar los datos del pago realizado por el afiliado?	Si
• ¿La clase permite la actualización de la información de pagos en la base de datos?	Si
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.53 Unidad de prueba No 53: Reporte de usuarios

UNIDAD DE PRUEBA N° 053	
Nombre de la Clase	ReporteUsuarios
Código Tarjeta CRC	053
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los usuarios registrados en la aplicación?	Si

• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección de varios parámetros para que el usuario configure el reporte que desea consultar?	Si
• ¿La clase permite mostrar en pantalla los datos seleccionados por el usuario?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.54 Unidad de prueba No 54: Reporte de afiliados

UNIDAD DE PRUEBA N° 054	
Nombre de la Clase	ReporteAfiliados
Código Tarjeta CRC	054
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección de varios parámetros para que el usuario configure el reporte que desea consultar?	Si
• ¿La clase permite mostrar en pantalla los datos seleccionados por el usuario?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.55 Unidad de prueba No 55: Reporte de aportes

UNIDAD DE PRUEBA N° 055	
Nombre de la Clase	ReporteAportes
Código Tarjeta CRC	055
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los afiliados registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección de varios parámetros para que el usuario configure el reporte que desea consultar?	Si
• ¿La clase permite mostrar en pantalla los datos seleccionados por el usuario?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.56 Unidad de prueba No 56: Reporte de préstamos

UNIDAD DE PRUEBA N° 056	
Nombre de la Clase	ReportePrestamos
Código Tarjeta CRC	056
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los préstamos registrados a los afiliados?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si

• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección de varios parámetros para que el usuario configure el reporte que desea consultar?	Si
• ¿La clase permite mostrar en pantalla los datos seleccionados por el usuario?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.57 Unidad de prueba No 57: Reporte de pagos

UNIDAD DE PRUEBA N° 057	
Nombre de la Clase	ReportePagos
Código Tarjeta CRC	057
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de pagos registrados a los afiliados?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección de varios parámetros para que el usuario configure el reporte que desea consultar?	Si
• ¿La clase permite mostrar en pantalla los datos seleccionados por el usuario?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.58 Unidad de prueba No 58: Reporte de Egresos

UNIDAD DE PRUEBA N° 058	
Nombre de la Clase	ReportEgresos
Código Tarjeta CRC	058
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los egresos registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección de varios parámetros para que el usuario configure el reporte que desea consultar?	Si
• ¿La clase permite mostrar en pantalla los datos seleccionados por el usuario?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.59 Unidad de prueba No 59: Reporte de Ingresos

UNIDAD DE PRUEBA N° 059	
Nombre de la Clase	ReportIngresos
Código Tarjeta CRC	059
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los ingresos registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda	Si

realizar las validaciones correspondientes a los datos ingresados por el usuario?	
• ¿La clase permite la selección de varios parámetros para que el usuario configure el reporte que desea consultar?	Si
• ¿La clase permite mostrar en pantalla los datos seleccionados por el usuario?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.60 Unidad de prueba No 60: Reporte de tipos de aporte

UNIDAD DE PRUEBA N° 060	
Nombre de la Clase	ReportTipoAporte
Código Tarjeta CRC	060
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los ingresos registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección de la opción generar reporte de tipos de aporte?	Si
• ¿La clase permite mostrar en pantalla el reporte de tipos de aporte que se pueden realizar en la cooperativa?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.61 Unidad de prueba No 61: Reporte tipos de préstamo

UNIDAD DE PRUEBA N° 061	
Nombre de la Clase	ReportIngresos
Código Tarjeta CRC	061
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de los datos de los ingresos registrados en la aplicación?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿Realiza una instancia de la clase ValidarDatos con la que pueda realizar las validaciones correspondientes a los datos ingresados por el usuario?	Si
• ¿La clase permite la selección la opción generar reporte de tipos de préstamos?	Si
• ¿La clase permite mostrar en pantalla el reporte de tipos de préstamos que se pueden realizar en la cooperativa?	Si
• ¿Implementa la clase un método que permita la impresión o almacenamiento del reporte generado?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.62 Unidad de prueba No 62: Bitácora

UNIDAD DE PRUEBA N° 062	
Nombre de la Clase	Bitacora
Código Tarjeta CRC	062
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de la bitácora?	Si
• ¿Implementa la clase un método que permite el registro de todos los movimientos realizados en la aplicación?	Si
• ¿Implementa la clase un método que permita imprimir o guardar o en pantalla un reporte de todos los movimientos realizados en la aplicación?	Si

• ¿Implementa la clase un método que muestre en pantalla todos los movimientos realizados en la aplicación?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.63 Unidad de prueba No 63: Reporte de bitácora

UNIDAD DE PRUEBA N° 063	
Nombre de la Clase	ReporteBitacora
Código Tarjeta CRC	063
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿incluye los componentes gráficos necesarios para la administración de la bitácora?	Si
• ¿La clase permite el ingreso de un parámetro de búsqueda?	Si
• ¿Implementa la clase un método que muestre en pantalla todos los movimientos realizados en la aplicación según el parámetro de búsqueda?	Si
• ¿Implementa la clase un método que permita imprimir o guardar o en pantalla un reporte de todos los movimientos realizados en la aplicación?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.64 Unidad de prueba No 64: Copia de seguridad de la base de datos

UNIDAD DE PRUEBA N° 064	
Nombre de la Clase	CopiaBd
Código Tarjeta CRC	064
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de la base de datos?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda	Si

realizar las gestiones sobre la base de datos?	
• ¿La clase permite la creación de una copia de seguridad de la base de datos?	
• ¿Implementa la clase un método que permita el almacenamiento de la copia de seguridad generada?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.65 Unidad de prueba No 65: Restaurar copia de seguridad de la base de datos

UNIDAD DE PRUEBA N° 065	
Nombre de la Clase	RestCopiaBd
Código Tarjeta CRC	065
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de la base de datos?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿La clase permite la restauración de una copia de seguridad de la base de datos?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.66 Unidad de prueba No 66: Importar la base de datos

UNIDAD DE PRUEBA N° 066	
Nombre de la Clase	ImportarBd
Código Tarjeta CRC	066
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de la base de datos?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda	Si

realizar las gestiones sobre la base de datos?	
• ¿La clase permite la importación de la base de datos?	
• ¿Implementa la clase un método que permita la validación de los datos importados a la base de datos?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.3.1.67 Unidad de prueba No 67: Exportar la base de datos

UNIDAD DE PRUEBA N° 067	
Nombre de la Clase	ExportarBd
Código Tarjeta CRC	067
¿Cumple la clase con los siguientes requisitos?	
• ¿Implementa la clase las características de una página web?	Si
• ¿Incluye los componentes gráficos necesarios para la administración de la base de datos?	Si
• ¿Realiza una instancia de la clase ConexionDB con la que pueda realizar las gestiones sobre la base de datos?	Si
• ¿La clase permite la exportación de la base de datos?	
• ¿Implementa la clase un método que informe sobre el resultado de la operación por medio de un mensaje en pantalla?	Si
Calificación a la Prueba	Implementación correcta de la clase

2.4 FASE DE DE PRUEBAS

Un desarrollo mediante programación extrema esta compuesta por una serie de las iteraciones cortas. Cada iteración concluye ejecutando un conjunto de pruebas: primero las pruebas unitarias y por último las de aceptación y pruebas funcionales que son creadas y usadas por los clientes para comprobar si se esta de acuerdo o no con el resultado. El efecto de estas pruebas es un programa mas seguro que soporta cambios en el tiempo.

Las pruebas de aceptación o funcionales sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario, para asegurar el funcionamiento final de una historia de usuario, pruebas que son creadas y usadas por los clientes para comprobar que estas cumplen su cometido

2.4.1 Pruebas de Aceptación. Las pruebas de aceptación tienen varios aspectos, se registra el nombre de las historias de usuario evaluadas, las

condiciones de ejecución, por ejemplo para el *Caso de Prueba de Aceptación No3: Crear usuario* una de las condiciones de ejecución, son los pasos que se debe seguir para mirar el resultado de nuestra prueba, los resultados esperados, lo que se espera realice el sistema y por último la evaluación de la prueba que es lo que el sistema realizó una vez realizados los pasos de ejecución.

2.4.1.1 Caso de Prueba de Aceptación No 1: Iniciar la Aplicación

Caso de prueba de aceptación		Iniciar la Aplicación
Nombre y código de HU	Aplicación web_001 Ingresar al sistema_002	
Descripción	Iniciar la aplicación web AFICOP desde un navegador web	
Condiciones de Ejecución	<ul style="list-style-type: none"> En el equipo servidor debe estar ejecutándose el servidor web, el motor de bases de datos y la aplicación debidamente instalada. 	
Entrada / Pasos de ejecución		
Caso 1 Flujo Básico	<ul style="list-style-type: none"> Se ingresa la URL de la aplicación web en la barra de dirección del navegador web que se esté utilizando. Se espera hasta que el navegador web cargue la aplicación. 	
Caso 2 Dirección web incorrecta	<ul style="list-style-type: none"> Se ingresa la URL de la aplicación web en la barra de dirección del navegador web que se esté utilizando. Se espera hasta que el navegador web cargue la aplicación. 	
Resultado Esperado		
Caso 1 Flujo Básico	<ul style="list-style-type: none"> Se establece la conexión con la aplicación y se conecta con las bases de datos. Debe abrir la ventana principal donde se solicite el ingreso del login y la clave de usuario. 	
Caso 2 Dirección web incorrecta	<ul style="list-style-type: none"> La aplicación web AFICOP no debe ejecutarse. (por que se encuentran errores de escritura o por fallas en la conexión de red). El navegador web mostrara una página diferente en caso de haber digitado mal la dirección de la aplicación. 	
Evaluación de la prueba		
Caso 1 Flujo Básico	<ul style="list-style-type: none"> La aplicación inició y mostró la ventana principal; donde se solicita el ingreso del login y clave. 	
Caso 2 Dirección	<ul style="list-style-type: none"> La aplicación no se ejecuto. 	

web incorrecta	
Prueba aprobada	Si

2.4.1.2 Caso de Prueba de Aceptación No 2: Iniciar Sesión

Caso de prueba de aceptación		Iniciar Sesión
Nombre y código de las HU	Ingreso a la aplicación_002 Menú de módulos_003	
Descripción	Hacer uso de la aplicación ingresando el login y clave de usuario	
Condiciones de Ejecución	<ul style="list-style-type: none"> PDA – Iniciar la Aplicación – Caso 1. 	
Entrada / Pasos de ejecución		
Caso 1 Inicio de sesión	<ul style="list-style-type: none"> Se ingresa el login y clave de usuario para acceder a la aplicación. 	
Caso 2 Login y clave incorrectos	<ul style="list-style-type: none"> Se ingresa el login y clave de usuario para acceso a la aplicación. 	
Resultado esperado		
Caso 1 Inicio de sesión	<ul style="list-style-type: none"> Se debe mostrar en pantalla la página principal de la aplicación. 	
Caso 2 Login y clave incorrectos	<ul style="list-style-type: none"> Se debe mostrar un mensaje de error indicando que el login y/o la clave de usuario son incorrectos. Se debe solicitar el ingreso correcto de login y de la clave. 	
Evaluación de la prueba		
Caso 1 Inicio de sesión	<ul style="list-style-type: none"> La aplicación inició y mostró la página principal de la aplicación con sus diferentes módulos. 	
Caso 2 Login y clave incorrectos	<ul style="list-style-type: none"> La aplicación mostró en pantalla un mensaje de error indicando que el login y clave son inválidos y solicitando el reingreso de los datos. 	
Prueba aprobada	Si	

2.4.1.3 Caso de Prueba de Aceptación No 3: Crear Usuario

Caso de prueba de aceptación		Crear Usuario
Nombre y código de las HU	<ul style="list-style-type: none"> Crear cuenta de Usuario 005 	

	<ul style="list-style-type: none"> Perfiles de usuario 006
Descripción	El usuario administrador creara una cuenta de usuario adicionando los datos personales (Identificación, Tipo de documento, Nombres, Apellidos, Fecha de Nacimiento, Sexo, Estado Civil, Cargo, Barrio, Dirección, Departamento, Municipio, Teléfono, Celular, Correo Electrónico, Fecha de Ingreso) y de cuenta de usuario (Login, Perfil de acceso, Estado, Clave, Confirmar Clave) donde se asigna los permisos para acceder a la aplicación.
Condiciones de Ejecución	<ul style="list-style-type: none"> PDA Iniciar la Aplicación – Caso 1 PDA Iniciar Sesión _Caso 1. El administrador deberá seleccionar desde el menú principal el submenú de usuarios y escoger la opción Nuevo Usuario.
Entrada / Pasos de ejecución	
Caso 1 Crear usuario	<ul style="list-style-type: none"> Se solicitan los datos personales del nuevo usuario Se solicitan los datos de la cuenta del nuevo usuario Se asignan el perfil de acceso y los permisos de operación para el nuevo usuario. Se registra el nuevo usuario
Caso 2 Número de identificación ya registrado	<ul style="list-style-type: none"> Se ingresa un número de identificación ya registrado.
Caso 3 Login incorrecto	<ul style="list-style-type: none"> Se ingresa un login ya registrado. Se establecen el perfil de acceso y los permisos de operación para el nuevo usuario.
Caso 4 Confirmación de clave incorrecta	<ul style="list-style-type: none"> Se ingresa la confirmación de clave incorrecta. Se establecen el nivel de acceso y los permisos de operación para el nuevo usuario.
Resultado esperado	
Caso 1 Crear usuario	<ul style="list-style-type: none"> La aplicación debe permitir el registro del nuevo usuario Debe mostrar un mensaje, indicando que se registró al nuevo usuario.
Caso 2	<ul style="list-style-type: none"> La aplicación muestre un mensaje en pantalla indicando

Número de identificación ya registrado	que ya se encuentra un usuario registrado con esa identificación.
Caso 3 Login incorrecto	<ul style="list-style-type: none"> • La aplicación muestre en pantalla un mensaje de error indicando que el login ingresado ya está registrado. • La aplicación solicite nuevamente el login correcto.
Caso 4 Confirmación de clave incorrecta	<ul style="list-style-type: none"> • La aplicación muestre en pantalla un mensaje de error indicando que la confirmación de la clave del nuevo usuario es errónea. • La aplicación solicite nuevamente la confirmación de clave correcta.
Evaluación de la prueba	
Caso 1 Crear usuario	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que el usuario se almaceno correctamente.
Caso 2 Número de identificación ya registrado	<ul style="list-style-type: none"> • La aplicación mostró un mensaje en pantalla indicando que ya existe un usuario registrado con esa identificación.
Caso 3 Login incorrecto	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que ya existe un usuario con un login igual. • Se permitió el ingreso de un nuevo login. • La aplicación mostró un mensaje en pantalla indicando que el usuario se almaceno correctamente.
Caso 4 Confirmación de clave incorrecta	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que la clave y confirmación de la clave no coinciden. • Se permitió el ingreso de la clave y confirmación de clave correctas. • La aplicación mostró un mensaje en pantalla indicando que el usuario se almaceno correctamente.
Prueba aceptada	Si

2.4.1.4 Caso de Prueba de Aceptación No 4: Consultar cuentas de usuarios.

Caso de prueba de aceptación Consultar cuentas de usuarios	
Nombre y código de las HU	<ul style="list-style-type: none"> • Consultar cuenta de usuario 007
Descripción	Se realiza por parte del administrador la consulta de los datos personales como número de cedula, nombres, apellidos y

	datos de la cuenta de usuario como el login, perfil de acceso y clave del usuario.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación – Caso 1. • PDA Iniciar Sesión _Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de usuarios y escoger la opción Gestión de Usuarios.
Entrada / Pasos de ejecución	
Caso 1 Parámetro de búsqueda correcto	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla los datos de la cuenta de usuario según el parámetro de búsqueda seleccionado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda.
Resultado esperado	
Caso 1 Parámetro de búsqueda correcto	<ul style="list-style-type: none"> • La aplicación muestre en pantalla los datos de la cuenta de usuario según el parámetro de búsqueda seleccionado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • La aplicación muestre en pantalla un mensaje indicando que no se encontraron registros que coincidan con el parámetro de búsqueda seleccionado.
Evaluación de la prueba	
Caso 1 Parámetro de búsqueda correcto	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de la cuenta de usuario según el parámetro de búsqueda seleccionado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que no se encontraron resultados para esa búsqueda. • La aplicación permitió la selección de nuevos parámetros de búsqueda.
Prueba aprobada	Si

2.4.1.5 Caso de Prueba de Aceptación No 5: Modificar cuentas de usuarios.

Caso de prueba de aceptación		Modificar cuentas de usuarios
Nombre y código de las HU	<ul style="list-style-type: none"> • Modificar cuenta de usuario 008 	
Descripción	<p>El administrador realiza la modificación de los datos personales (Identificación, Tipo de documento, Nombres, Apellidos, Fecha de Nacimiento, Sexo, Estado Civil, Cargo, Barrio, Dirección, Departamento, Municipio, Teléfono, Celular, Correo Electrónico, Fecha de Ingreso) y de cuenta de usuario (Login, Perfil de acceso, Estado, Clave, Confirmar Clave).</p>	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de usuarios y escoger la opción Gestión de Usuarios. 	
Entrada / Pasos de ejecución		
Caso 1 Edición de cuenta	<ul style="list-style-type: none"> • Se escoge un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla una lista con los datos de las cuentas de usuario según el parámetro de búsqueda seleccionado. • Se selecciona el registro a modificar. • Se escoge la opción de editar. • Se ingresan los datos que se desea cambiar • Se espera que la aplicación muestre en pantalla un mensaje indicando que se registraron los cambios realizados. 	
Resultado esperado		
Caso 1 Edición de cuenta	<ul style="list-style-type: none"> • La aplicación muestre en pantalla los datos de la cuenta de usuario según el parámetro de búsqueda seleccionado. • Se permita el ingreso de los datos de la cuenta de usuario que se desea cambiar. • La aplicación muestre en pantalla un mensaje indicando que se registraron los cambios realizados. 	
Evaluación de la prueba		

Caso 1 Edición de cuenta	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de la cuenta de usuario según el parámetro de búsqueda seleccionado. • Se seleccionó la opción de editar. • Se permitió el ingreso de los datos a modificar. • La aplicación mostró en pantalla un mensaje de éxito indicando que el usuario se actualizó correctamente.
Prueba aprobada	Si

2.4.1.6 Caso de Prueba de Aceptación No 6: Eliminar cuentas de usuarios.

Caso de prueba de aceptación		Eliminar cuentas de usuarios
Nombre y código de las HU		<ul style="list-style-type: none"> • Eliminar cuenta de usuario 008
Descripción		El administrador realiza la eliminación de los datos personales y datos de la cuenta de usuario.
Condiciones de Ejecución		<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de usuarios y escoger la opción Gestión de Usuarios.
Entrada / Pasos de ejecución		
Caso 1 Cuenta eliminada	<ul style="list-style-type: none"> • Se escoge un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla los datos de las cuentas de usuario según el parámetro de búsqueda seleccionado. • Se selecciona el registro a eliminar. • Se escoge la opción de eliminar. • Se espera que la aplicación muestre un mensaje de confirmación de la eliminación. • Se oprime continuar del mensaje de confirmación de la eliminación. 	
Caso 2 Cuenta no eliminada	<ul style="list-style-type: none"> • Se escoge un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla los datos de las cuentas de usuario según el parámetro de búsqueda seleccionado. • Se selecciona el registro a eliminar. • Se escoge la opción de eliminar. 	

	<ul style="list-style-type: none"> • Se espera que la aplicación muestre en pantalla un mensaje de confirmación de la eliminación. • Se oprime continuar del mensaje de confirmación de la eliminación.
Resultado esperado	
Caso 1 Cuenta eliminada	<ul style="list-style-type: none"> • Se muestre en pantalla los datos según el parámetro de búsqueda. • Se muestre en pantalla un mensaje que permita confirmar la eliminación del usuario. • Se muestre en pantalla un mensaje que indique que la cuenta de usuario ha sido eliminada.
Caso 2 Cuenta no eliminada	<ul style="list-style-type: none"> • Se muestre en pantalla los datos según el parámetro de búsqueda. • Se muestre en pantalla un mensaje que permita confirmar la eliminación del usuario. • Se muestre en pantalla un mensaje que indique por que la cuenta de usuario no ha sido eliminada.
Evaluación de la prueba	
Caso 1 Cuenta eliminada	<ul style="list-style-type: none"> • Se mostró en pantalla los datos según el parámetro de búsqueda. • Se seleccionó el registro de la cuenta a eliminar la opción eliminar. • Se seleccionó la opción continuar del mensaje de confirmación de eliminación.
Caso 2 Cuenta no eliminada	<ul style="list-style-type: none"> • Se mostró en pantalla los datos según el parámetro de búsqueda • Se seleccionó la opción eliminar. • Se seleccionó la opción cancelar del mensaje de confirmación de eliminación. • La cuenta no pudo ser eliminada.
Prueba aprobada	Si

2.4.1.7 Caso de Prueba de Aceptación No 7: Registrar Formularios de afiliación.

Caso de prueba de aceptación	Registrar Formularios de afiliación
Nombre y código de las HU	<ul style="list-style-type: none"> • Registrar Formularios de afiliación 010
Descripción	<ul style="list-style-type: none"> • El usuario administrador registrara a un nuevo afiliado adicionando los

	datos de información general, económica, financiera, beneficiarios, tutores, cursos cooperativos y de cuenta de usuario.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1.. • El administrador deberá seleccionar desde el menú principal el submenú de afiliados y escoger la opción Formulario Afiliación.
Entrada / Pasos de ejecución	
Caso 1 Registrar Formularios de afiliación	<ul style="list-style-type: none"> • Se ingresan los datos de información general del nuevo afiliado. • Se oprime el botón registrar. • Se espera que la aplicación registre los datos de información general ingresados del nuevo afiliado. • Se ingresan los datos de información económica del nuevo afiliado. • Se oprime el botón registrar. • Se espera que la aplicación registre los datos de información económica ingresados del nuevo afiliado. • Se ingresan los datos de información financiera del nuevo afiliado. • Se oprime el botón registrar. • Se espera que la aplicación registre los datos de información financiera ingresados del nuevo afiliado. • Se ingresan los datos de los beneficiarios del nuevo afiliado. • Se oprime el botón registrar. • Se espera que la aplicación registre los datos de información de los beneficiarios ingresados del nuevo afiliado. • Se ingresan los datos de la información de tutores del nuevo afiliado. • Se oprime el botón registrar. • Se espera que la aplicación registre los datos de información de tutores ingresados del nuevo afiliado.
Caso 2 Datos de información	<ul style="list-style-type: none"> • Se ingresan los datos de información general del nuevo afiliado.

general no registrados	<ul style="list-style-type: none"> • Se oprime el botón registrar. • Se espera que la aplicación registre los datos de información general del nuevo afiliado. • La aplicación no permita el ingreso de los demás datos de afiliación.
Resultado esperado	
Caso 1 Registrar Formularios de afiliación	<ul style="list-style-type: none"> • La aplicación registre los datos de información general • La aplicación muestre un mensaje en pantalla, indicando que se registraron los datos de información general. • Se permita el ingreso de los datos de la información económica. • La aplicación registre los datos de información económica. • Se muestre un mensaje en pantalla, indicando que se registraron los datos de información económica. • La aplicación permita el ingreso de los datos de información financiera. • La aplicación registre los datos de la información financiera. • Se muestre en pantalla un mensaje indicando que se registraron los datos de la información financiera. • La aplicación permita el ingreso de los datos de información de beneficiarios. • La aplicación registre los datos de la información de beneficiarios. • Se muestre en pantalla un mensaje indicando que se registraron los datos de la información de beneficiarios. • La aplicación permita el ingreso de los datos de información de tutores. • La aplicación registre los datos de la información de tutores. • Se muestre en pantalla un mensaje indicando que se registraron los datos de la información de tutores.
Caso 2 Datos de información general no registrados	<ul style="list-style-type: none"> • La aplicación permita el ingreso de los datos de información general. • La aplicación registre los datos de información general. • La aplicación debe mostrar en pantalla un mensaje de error indicando por que no se registraron los datos de información general del nuevo afiliado.
Evaluación de la prueba	
Caso 1	<ul style="list-style-type: none"> • Se permitió el ingreso de los datos de información general.

Registrar Formularios de afiliación	<ul style="list-style-type: none"> • La aplicación mostró un mensaje en pantalla indicando que la información general fue almacenada correctamente. • Se permitió el ingreso de los datos de información económica. • La aplicación mostró un mensaje en pantalla indicando que la información económica fue almacenada correctamente. • Se permitió el ingreso de los datos de información financiera. • La aplicación mostró un mensaje en pantalla indicando que la información financiera fue almacenada correctamente. • Se permitió el ingreso de los datos de información beneficiarios. • La aplicación mostró un mensaje en pantalla indicando que la información beneficiarios fue almacenada correctamente. • Se permitió el ingreso de los datos de información de tutores. • La aplicación mostró un mensaje en pantalla indicando que la información de tutores fue almacenada correctamente.
Caso 2 Datos de información general no registrados	<ul style="list-style-type: none"> • Se permitió el ingreso de los datos de información general del nuevo afiliado. • La aplicación mostró un mensaje en pantalla indicando que el número de identificación del afiliado ya estaba registrado. • La aplicación permitió el ingreso de un nuevo número de identificación.
Prueba aprobada	Si

2.4.1.8 Caso de Prueba de Aceptación No 8: Consultar afiliados.

Caso de prueba de aceptación		Consultar afiliados
Nombre y código de las HU	Consultar afiliados 011	
Descripción	Se realiza por parte del administrador la consulta de los datos registrados en los formularios de afiliación.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación – Caso 1. • PDA Iniciar Sesión _Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de Afiliados y escoger la opción Gestión de Afiliados. 	
Entrada / Pasos de ejecución		
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda • Se espera que la aplicación muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. 	
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • Se selecciona el parámetro de búsqueda. 	
Resultado esperado		
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • La aplicación muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. 	
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • La aplicación muestre un mensaje en pantalla indicando que no se encontraron registros que coincidan con el parámetro de búsqueda seleccionado. 	
Evaluación de la prueba		
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. 	
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • La aplicación mostró un mensaje en pantalla indicando que no se encontraron registros que coincidan con el parámetro de búsqueda seleccionado. 	
Prueba aprobada	Si	

2.4.1.9 Caso de Prueba de Aceptación No 9: Modificar Afiliados.

Caso de prueba de aceptación		Modificar Afiliados
Nombre y código de las HU	<ul style="list-style-type: none"> • Modificar afiliados 012 	
Descripción	<ul style="list-style-type: none"> • El administrador realiza la modificación de los datos del afiliado. 	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA registrar formulario de afiliación – Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de afiliados y escoger la opción Gestión de Afiliados. 	
Entrada / Pasos de ejecución		
Caso 1 Edición registrada	<ul style="list-style-type: none"> • Se escoge un el parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla una lista con los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado a modificar. • Se escoge la opción de editar del registro seleccionado. • Se espera que la aplicación muestre en pantalla los datos que se pueden cambiar. • Se oprime el botón actualizar datos 	
Caso 2 Edición no registrada	<ul style="list-style-type: none"> • Se escoge un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se escoge el registro a modificar. • Se escoge la opción editar del registro seleccionado. . • Se espera que la aplicación muestre en pantalla los datos de los afiliados que se pueden cambiar. • Se oprime el botón actualizar datos del registro seleccionado. • Se espera que la aplicación indique en pantalla por que no se registraron los cambios realizados. 	
Resultado esperado		
Caso 1 Edición	<ul style="list-style-type: none"> • se muestre en pantalla los datos de los afiliados según el parámetro de búsqueda. 	

registrada	<ul style="list-style-type: none"> • Se muestre en pantalla los datos del afiliado que se pueden cambiar. • Se actualicen los datos del afiliado según los cambios realizados.
Caso 2 Edición no registrada	<ul style="list-style-type: none"> • Se muestre en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se muestre en pantalla los datos del afiliado que se pueden cambiar. • Se muestra en pantalla un mensaje indicando por que no se puede registrar la modificación de los datos del afiliado.
Evaluación de la prueba	
Caso 1 Edición registrada	<ul style="list-style-type: none"> • Se mostró en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del afiliado a modificar. • Se ingresaron los datos a modificar. • Se mostró en pantalla un mensaje de éxito indicando que los datos del afiliado se actualizaron correctamente.
Caso 2 Edición no registrada	<ul style="list-style-type: none"> • Se mostró en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del afiliado a modificar. • Se ingresó el número de identificación del afiliado. • La aplicación mostró un mensaje en pantalla indicando que existe otra persona con ese número de identificación.
Prueba aprobada	Si

2.4.1.10 Caso de Prueba de Aceptación No 10: Eliminar Afiliados.

Caso de prueba de aceptación		Eliminar afiliados
Nombre y código de las HU	<ul style="list-style-type: none"> • Eliminar afiliados 013 	
Descripción	El administrador realiza la eliminación de los datos del afiliado.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _ Caso 1. • PDA Registrar formulario de afiliación _ Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de afiliados y escoger la opción Gestión de Afiliados. 	
Entrada / Pasos de ejecución		

Caso 1 Eliminación registrada	<ul style="list-style-type: none"> • Se escoge un parámetro de búsqueda. • Se espera que la aplicación muestre los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado a eliminar • Se escoge la opción de eliminar. • Se espera que la aplicación muestre en pantalla un mensaje de confirmación de eliminación. • Se escoge la opción de continuar del mensaje de confirmación.
Caso 2 Cancelar eliminación	<ul style="list-style-type: none"> • Se escoge un parámetro de búsqueda. • Se espera que la aplicación muestre los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado a eliminar • Se escoge la opción de eliminar. • Se espera que la aplicación muestre en pantalla un mensaje de confirmación de eliminación. • Se escoge la opción de cancelar del mensaje de confirmación de eliminación del afiliado.
Resultado esperado	
Caso 1 Eliminación registrada	<ul style="list-style-type: none"> • Se muestren los datos según el parámetro de búsqueda ingresado. • Se permita la selección del registro a eliminar. • Se muestre en pantalla un mensaje que permita confirmar la eliminación del afiliado.
Caso 2 Eliminación cancelada	<ul style="list-style-type: none"> • Se muestren los datos según el parámetro de búsqueda ingresado. • Se permita la selección del registro a eliminar. • Se muestre en pantalla un mensaje que permita confirmar la eliminación del afiliado.
Evaluación de la prueba	
Caso 1 Eliminación registrada	<ul style="list-style-type: none"> • Se mostró en pantalla los datos según el parámetro de búsqueda. • Se seleccionó el registro a eliminar. • Se mostró en pantalla un mensaje de confirmación de eliminar. • Se escogió la opción continuar del mensaje de confirmación.
Caso 2 Eliminación cancelada	<ul style="list-style-type: none"> • Se mostró en pantalla los datos según el parámetro de búsqueda. • Se seleccionó el registro a eliminar.

	<ul style="list-style-type: none"> • Se mostró en pantalla un mensaje de confirmación de eliminar. • Se escogió la opción cancelar del mensaje de confirmación.
Prueba aprobada	Si

2.4.1.11 Caso de Prueba de Aceptación No 11: Registrar Aportes.

Caso de prueba de aceptación		Registrar aporte
Nombre y código de las HU	<ul style="list-style-type: none"> • Registrar Aportes 011 	
Descripción	El usuario administrador registrara un nuevo aporte, realizado por un afiliado de la cooperativa	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la aplicación Caso 1. • PDA Iniciar sesión Caso 1 • El usuario administrador deberá ubicar desde el menú principal Ingresos, el submenú Aportes y escoger la opción Gestión de Aportes 	
Entrada / Pasos de ejecución		
Caso 1 Asignar Aporte	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera que aplicación muestre en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado. • Se escoge la opción aportes del registro seleccionado. • Se selecciona el tipo de aporte. • Se ingresa la fecha de inicio y fecha final del aporte. • Se espera que la aplicación genere un plan de aportes. 	
Resultado esperado		
Caso 1 Asignar Aporte	<ul style="list-style-type: none"> • Se muestre los datos según el parámetro de búsqueda. • La aplicación genere el plan de aportes según las fechas ingresadas. 	
Evaluación de la prueba		
Caso 1 Asignar Aporte	<ul style="list-style-type: none"> • Se mostraron los datos del afiliado según el parámetro de búsqueda ingresado. • Se seleccionó el registro del afiliado. • Se seleccionó la opción aportes del afiliado seleccionado. • La aplicación mostró en pantalla un menú desplegable con 	

	<p>los tipos de aportes que se pueden realizar en la cooperativa.</p> <ul style="list-style-type: none"> • Se seleccionó un tipo de aporte • Se ingresó las fechas de inicio y fin del plan de aportes. • La aplicación mostró en pantalla un plan de aportes con las fechas de recaudo del aporte y el valor del aporte de cada fecha según los datos ingresados
Prueba aprobada	Si

2.4.1.12 Caso de Prueba de Aceptación No 12: Consultar Aportes.

Caso de prueba de aceptación		Consultar aportes
Nombre y código de las HU	Consultar Aportes 016	
Descripción	Se realiza por parte del usuario la consulta de los aportes registrados a los afiliados de la cooperativa	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la aplicación Caso 1 • PDA Iniciar sesión Caso 1 • PDA Registrar afiliado Caso 1 • PDA Registrar aporte Caso 1 • El usuario (Administrador, Auxiliar Administrativo, Afiliado) deberá seleccionar desde el menú principal el submenú de aportes y escoger la opción gestión de aportes. 	
Entrada / Pasos de ejecución		
Caso 1 Consultar Aporte	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda • Se espera que la aplicación muestre en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado. • Se selecciona la opción aportes del afiliado. • Se escoge un tipo de aporte de los registrados al afiliado. • Se oprime el botón generar reporte. • Se espera que la aplicación muestre en pantalla el reporte con el plan de aportes del tipo de aporte seleccionado. 	
Resultado esperado		
Caso 1 Consultar Aporte	<ul style="list-style-type: none"> • Se muestren los datos del afiliado según el parámetro de búsqueda ingresado. • Se muestre en pantalla el reporte con el plan de aportes 	

	del aporte seleccionado.
Evaluación de la prueba	
Caso 1 Consultar Aporte	<ul style="list-style-type: none"> • Se mostró en pantalla los datos del afiliado según el parámetro de búsqueda. • Se seleccionó el registro del afiliado. • Se seleccionó la opción aportes del afiliado. • Se seleccionó un tipo de aporte. • Se oprimió el botón generar reporte. • La aplicación mostró en pantalla el reporte en formato pdf con los datos del tipo de aporte seleccionado y el plan de aportes del mismo.
Prueba aprobada	Si

2.4.1.13 Caso de Prueba de Aceptación No 13: Modificar Aportes.

Caso de prueba de aceptación		Modificar Aportes
Nombre y código de las HU		<ul style="list-style-type: none"> • Modificar aportes 015
Descripción		<ul style="list-style-type: none"> • El administrador realiza la modificación del estado del aporte.
Condiciones de Ejecución		<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _ Caso 1. • PDA registrar formulario de afiliación – Caso 1. • PDA Registrar Aporte _ Caso 1 • El usuario administrador deberá seleccionar desde el menú principal el submenú de Aportes y escoger la opción gestionar Aportes.
Entrada / Pasos de ejecución		
Caso 1 Cambiar estado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera que se muestre en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado. • Se escoje la opción aportes del registro seleccionado • Se espera que la aplicación muestre el plan de aportes. • Del plan de aportes se selecciona la cuota para registrar su pago. • Se selecciona la opción pagar del aporte seleccionado. 	
Resultado esperado		

Caso 1 Cambiar estado	<ul style="list-style-type: none"> • Se muestre en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se registre el cambio del estado de la cuota de pagar a pagado.
Evaluación de la prueba	
Caso 1 Cambiar estado	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • La aplicación mostró en pantalla el plan de aportes con el estado de la cuota en pagado.
Prueba aprobada	Si

2.4.1.14 Caso de Prueba de Aceptación No 14: Eliminar Aportes.

Caso de prueba de aceptación		Eliminar aportes
Nombre y código de las HU	<ul style="list-style-type: none"> • Eliminar aportes 017 	
Descripción	El administrador realiza la eliminación de los aportes realizados por el afiliado.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _ Caso 1. • PDA Registrar formulario de afiliación _ Caso 1. • PDA Registrar Aporte _ Caso 1 • El usuario administrador deberá seleccionar desde el menú principal el submenú de aportes y escoger la opción Gestión de Aportes. 	
Entrada / Pasos de ejecución		
Caso 1 Eliminar aporte	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera que se muestre en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado. • Se escoge la opción eliminar del registro seleccionado • Se espera que la aplicación muestre un mensaje de confirmación de la eliminación. • Se selecciona la opción continuar del mensaje de confirmación de la eliminación.. 	
Caso 2 Aporte no eliminado	<ul style="list-style-type: none"> • Se escoge un parámetro de búsqueda. • Se espera que se muestre en pantalla los datos del afiliado según el parámetro de búsqueda seleccionado. • Se selecciona el registro del afiliado. 	

	<ul style="list-style-type: none"> • Se escoge la opción eliminar del registro seleccionado • Se espera que la aplicación muestre un mensaje de confirmación de la eliminación. • Se selecciona la opción cancelar del mensaje de confirmación de la eliminación.
Resultado esperado	
Caso 1 Eliminar aporte	<ul style="list-style-type: none"> • Se muestren los datos según el parámetro de búsqueda ingresado. • Se permita seleccionar el registro del afiliado. • Se permita seleccionar la opción eliminar del registro del afiliado. • Se muestre el mensaje de confirmación de la eliminación.
Caso 2 Aporte no eliminado	<ul style="list-style-type: none"> • Se muestre en pantalla los datos del afiliado según el parámetro de búsqueda. • Se permita seleccionar el registro del afiliado. • Se permita seleccionar la opción eliminar del registro del afiliado. • Se muestre el mensaje de confirmación de la eliminación.
Evaluación de la prueba	
Caso 1 Eliminar aporte	<ul style="list-style-type: none"> • Se mostro en pantalla los datos según el parámetro de búsqueda. • Se selecciono el registro del afiliado • Se selecciono la opción eliminar del registro del afiliado. • Se selecciono la opción continuar del mensaje de confirmación de eliminación.
Caso 2 Aporte no eliminado	<ul style="list-style-type: none"> • Se mostró en pantalla los datos según el parámetro de búsqueda. • Se seleccionó la opción eliminar. • Se seleccionó la opción cancelar del mensaje de confirmación de eliminación.
Prueba aprobada	Si

2.4.1.15 Caso de Prueba de Aceptación No15: Registrar Préstamos.

Caso de prueba de aceptación Registrar préstamos	
Nombre y código de las HU	• Registrar prestamos 018
Descripción	El usuario administrador registrara un nuevo préstamo a un afiliado de la cooperativa.
Condiciones de Ejecución	• PDA Iniciar la Aplicación _ Caso 1.

	<ul style="list-style-type: none"> • PDA Iniciar Sesión _Caso 1. • PDA Registrar formulario afiliación_Caso 1 • El administrador deberá seleccionar desde el menú principal el submenú de préstamos y escoger la opción Registrar Préstamo.
Entrada / Pasos de ejecución	
Caso 1 Registrar préstamo	<ul style="list-style-type: none"> • Se ingresa el código de registro del préstamo del afiliado. • Se ingresa el número de cedula del afiliado. • Se selecciona el tipo de préstamo solicitado por el afiliado. • Se ingresa el número de cedula del codeudor. • Se ingresa el valor del préstamo solicitado por el afiliado. • Se ingresa la fecha de solicitud del préstamo del afiliado. • Se ingresa la fecha de aprobación del préstamo del afiliado. • Se ingresa la fecha del pago de la primera cuota del préstamo del afiliado. • Se oprime el botón registrar.
Resultado esperado	
Caso 1 Registrar préstamo	<ul style="list-style-type: none"> • La aplicación registre el préstamo al afiliado en la base de datos. • Se muestre un mensaje en pantalla, indicando que se registró el préstamo al afiliado.
Evaluación de la prueba	
Caso 1 Registrar préstamo	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que el préstamo se ha realizado correctamente.
Prueba aprobada	Si

2.4.1.16 Caso de Prueba de Aceptación No 16: Consultar préstamo.

Caso de prueba de aceptación Consultar préstamo	
Nombre y código de las HU	Consultar préstamo 019
Descripción	Se realiza por parte del administrador la consulta de los prestamos registrados a los afiliados de la cooperativa
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación – Caso 1. • PDA Iniciar Sesión _Caso 1.

	<ul style="list-style-type: none"> • PDA registrar Afiliado _ Caso 1 • PDA Registrar Aporte _ Caso 1 • PDA Registrar Préstamo _ Caso 1 • El usuario (Administrador, Auxiliar Administrativo, Afiliado) deberá seleccionar desde el menú principal el submenú de préstamos y escoger la opción Gestión de Préstamos.
Entrada / Pasos de ejecución	
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda • Se espera que la aplicación muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se selecciona el registro del préstamo del afiliado a consultar. • Se selecciona la opción ver del afiliado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado.
Resultado esperado	
Caso 1 Consultar préstamo	<ul style="list-style-type: none"> • Se muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se muestre en pantalla los datos del préstamo realizado por el afiliado.
Caso 2 Consultar préstamo	<ul style="list-style-type: none"> • Se muestre en pantalla un mensaje indicando que no se encontraron registros que coincidan con el parámetro de búsqueda.
Evaluación de la prueba	
Caso 1 Consultar préstamo	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del préstamo del afiliado. • Se seleccionó la opción ver del afiliado. • La aplicación mostró en pantalla los datos del préstamo realizado por el afiliado.
Caso 2 Consultar préstamo	<ul style="list-style-type: none"> • Se mostró en pantalla un mensaje indicando que no se encontraron registros que coincidan con el parámetro de búsqueda.

Prueba aprobada	Si
------------------------	----

2.4.1.17 Caso de Prueba de Aceptación No 17: Eliminar préstamos.

Caso de prueba de aceptación		Eliminar prestamos
Nombre y código de las HU	<ul style="list-style-type: none"> • Eliminar aportes 021 	
Descripción	El administrador realiza la eliminación de los préstamos realizados a un afiliado.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. y Caso 4 • PDA Iniciar Sesión _Caso 1. • PDA Registrar formulario de afiliación _Caso 1. • PDA Registrar Aporte _ Caso 1 • PDA Registrar Préstamo • El usuario administrador deberá seleccionar desde el menú principal el submenú de préstamos y escoger la opción Gestión de Préstamos. 	
Entrada / Pasos de ejecución		
Caso 1 Eliminar préstamo	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se selecciona el registro del préstamo del afiliado a eliminar • Se escoge la opción de eliminar del afiliado. • Se escoge la opción de confirmar eliminación del préstamo asignado a un afiliado. 	
Caso 2 Eliminación no registrada	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera que la aplicación muestre en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se selecciona el registro del préstamo del afiliado a eliminar • Se escoge la opción de eliminar del afiliado. 	
Resultado esperado		
Caso 1 Eliminar prestamos	<ul style="list-style-type: none"> • Muestren los datos según el parámetro de búsqueda. • Se muestre en pantalla un mensaje que permita confirmar la eliminación del préstamo realizado por el afiliado 	

	<ul style="list-style-type: none"> • Se muestre en pantalla un mensaje indicando que la eliminación se ha realizado.
Caso 2 Eliminación no registrada	<ul style="list-style-type: none"> • Muestren los datos según el parámetro de búsqueda ingresado. • Se muestre en pantalla un mensaje que indique por que no se realizo la eliminación del préstamo del afiliado.
Evaluación de la prueba	
Caso 1 Eliminar prestamos	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los afiliados según el parámetro de búsqueda. • Se selecciono el registro del préstamo del afiliado. • Se selecciono la opción de eliminar del afiliado. • La aplicación mostró un mensaje de confirmación de la eliminación. • Se selecciono el botón continuar del mensaje de confirmación de la eliminación.
Caso 2 Eliminar prestamos	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los afiliados según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del préstamo del afiliado. • Se seleccionó la opción de eliminar del afiliado. • La aplicación mostró un mensaje de confirmación de la eliminación. • Se seleccionó el botón continuar del mensaje de confirmación de la eliminación, • La aplicación mostró en pantalla un mensaje indicando que el préstamo no se puede eliminar porque aun hay cuotas sin pagar.
Prueba aprobada	Si

2.4.1.18 Caso de Prueba de Aceptación No 18: Registrar Egreso.

Caso de prueba de aceptación	Registrar Egreso
Nombre y código de las HU	<ul style="list-style-type: none"> • Registrar Egreso 022
Descripción	El usuario administrador registrara un nuevo egreso realizado en la cooperativa.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación – Caso 1 • PDA Iniciar Sesión _Caso 1. • El administrador deberá seleccionar desde el menú principal el submenú

	de egresos y escoger la opción Registrar Egresos.
Entrada / Pasos de ejecución	
Caso 1 Registrar Egreso	<ul style="list-style-type: none"> • Se ingresan los datos de del egreso. • Se oprime el botón registrar.
Caso 2 Egreso no registrado	<ul style="list-style-type: none"> • Se ingresan los datos del egreso • Se oprime el botón registrar.
Resultado esperado	
Caso 1 Registrar Egreso	<ul style="list-style-type: none"> • La aplicación muestre un mensaje en pantalla indicando que se ha registrado el nuevo egreso.
Caso 2 Egreso no registrado	<ul style="list-style-type: none"> • La aplicación muestre en pantalla un mensaje indicando por qué no se registro el egreso.
Evaluación de la prueba	
Caso 1 Registrar Egreso	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que el egreso se almaceno correctamente.
Caso 2 Egreso no registrado	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que la identificación del responsable no esta registrada.
Prueba aprobada	Si

2.4.1.19 Caso de Prueba de Aceptación No 19: Consultar egresos.

Caso de prueba de aceptación Consultar egresos	
Nombre y código de las HU	<ul style="list-style-type: none"> • Consultar egresos 023
Descripción	Se realiza por parte del administrador la consulta de los egresos registrados en la cooperativa
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Registrar Egreso_ Caso 1 • El usuario deberá seleccionar desde el menú principal el submenú de egresos y escoger la opción Gestión de Egresos.
Entrada / Pasos de ejecución	

Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda • Se espera que la aplicación muestre los datos según el parámetro de búsqueda ingresado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda • Se espera que la aplicación muestre los datos según el parámetro de búsqueda ingresado.
Resultado esperado	
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • La aplicación muestre en pantalla los datos de los egresos realizados según el parámetro de búsqueda seleccionado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • La aplicación muestre en pantalla un mensaje indicando que no se encontraron registros que coincidan con el parámetro de búsqueda seleccionado. • La aplicación permita la selección de nuevos parámetros de búsqueda.
Evaluación de la prueba	
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los egresos realizados según el parámetro de búsqueda seleccionado.
Caso 2 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que no se existen resultados • La aplicación permitió la selección de nuevos parámetros de búsqueda.
Prueba aprobada	Si

2.4.1.20 Caso de Prueba de Aceptación No 20: Modificar egresos.

Caso de prueba de aceptación		Modificar egresos
Nombre y código de las HU	<ul style="list-style-type: none"> • Modificar egresos 024 	
Descripción	<ul style="list-style-type: none"> • El administrador realiza la 	

	modificación de los datos del egreso.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _Caso 1. • PDA Iniciar Sesión _Caso 1.. • PDA Registrar Egreso _Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de egresos y escoger la opción Gestión de Egresos.
Entrada / Pasos de ejecución	
Caso 1 Modificar egreso	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se selecciona el registro del egreso a modificar. • Se escoge la opción de editar. • Se espera que la aplicación muestre en pantalla los datos delos egresos que se pueden cambiar. • Se ingresan los datos que se desea cambiar. • Se oprime el botón Actualizar datos.
Resultado esperado	
Caso 1 Modificar egreso	<ul style="list-style-type: none"> • La aplicación muestra en pantalla los datos de los egresos realizados según el parámetro de búsqueda seleccionado. • La aplicación muestre los datos de los egresos que se pueden modificar. • La aplicación muestre un mensaje indicando que los datos de los egresos han sido modificados.
Evaluación de la prueba	
Caso 1 Modificar egreso	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los egresos realizados según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del egreso a modificar. • Se seleccionó la opción editar del egreso, • La aplicación mostró en pantalla un mensaje de éxito indicando que los datos del egreso se actualizaron correctamente.
Prueba aprobada	Si

2.4.1.21 Caso de Prueba de Aceptación No 21: Eliminar egresos.

Caso de prueba de aceptación		Eliminar egresos
Nombre y código de las HU	<ul style="list-style-type: none"> • Eliminar egresos 024 	
Descripción	El administrador realiza la eliminación	

	de los egresos realizados en la cooperativa.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1.. • PDA Registrar Egreso_Caso 1. • El usuario administrador deberá seleccionar desde el menú principal el submenú de egresos y escoger la opción Gestión de Egresos.
Entrada / Pasos de ejecución	
Caso 1 Eliminar egreso	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera hasta que la aplicación muestre los datos según el parámetro de búsqueda seleccionado. • Se selecciona el registro del egreso a eliminar. • Se selecciona la opción eliminar del mensaje de confirmación de eliminación.
Caso 2 Eliminación no registrada	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se espera hasta que la aplicación muestre los datos según el parámetro de búsqueda seleccionado. • Se selecciona la opción cancelar del mensaje de confirmación de eliminación.
Resultado esperado	
Caso 1 Eliminar egreso	<ul style="list-style-type: none"> • Se muestren los datos según el parámetro de búsqueda seleccionado. • Se muestre en pantalla un mensaje que permita confirmar la eliminación del egreso.
Caso 2 Eliminación no registrada	<ul style="list-style-type: none"> • La aplicación muestre los datos según el parámetro de búsqueda seleccionado. • Se muestre en pantalla un mensaje que permita confirmar la eliminación del egreso.
Evaluación de la prueba	
Caso 1 Eliminar egreso	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos del egreso según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del egreso a eliminar. • Se seleccionó la opción de eliminar. • La aplicación mostró un mensaje en pantalla de confirmación de la eliminación. • Se seleccionó la opción continuar del mensaje.
Caso 2	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos del egreso

Eliminación no registrada	<p>según el parámetro de búsqueda seleccionado.</p> <ul style="list-style-type: none"> • Se seleccionó el registro del egreso a eliminar. • Se seleccionó la opción de eliminar. • La aplicación mostró un mensaje en pantalla de confirmación de la eliminación. • Se seleccionó la opción cancelar del mensaje de confirmación de la eliminación.
Prueba aprobada	Si

2.4.1.22 Caso de Prueba de Aceptación No 22: Registrar pagos.

Caso de prueba de aceptación Registrar Pagos	
Nombre y código de las HU	<ul style="list-style-type: none"> • Registrar pagos 025
Descripción	El usuario administrador registrara un nuevo pago, realizado por un afiliado de la cooperativa
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Registrar Prestamos _Caso 1. • El administrador deberá seleccionar desde el menú principal el submenú de pagos y escoger la opción Gestión de Pagos.
Entrada / Pasos de ejecución	
Caso 1 Registrar pago	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se selecciona el registro del préstamo del afiliado. • Se selecciona lo opción editar del afiliado. • Se ingresa la fecha de pago • Se oprime el botón calcular. • Se oprime el botón realizar pago.
Caso 2 Registrar pago	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se selecciona el registro del préstamo del afiliado. • Se selecciona lo opción editar del afiliado. • Se ingresa la fecha de pago • Se oprime el botón calcular. • Se oprime el botón realizar pago.
Resultado esperado	
Caso 1 Registrar pago	<ul style="list-style-type: none"> • La aplicación muestre los datos del afiliado según el parámetro de búsqueda seleccionado.

	<ul style="list-style-type: none"> • La aplicación muestre un mensaje en pantalla indicando que se ha registrado el pago.
Caso 2 Pago no registrado	<ul style="list-style-type: none"> • La aplicación muestre los datos del afiliado según el parámetro de búsqueda ingresado. • La aplicación muestre un mensaje en pantalla indicando por que no se pudo registrar el pago.
Evaluación de la prueba	
Caso 1 Registrar pago	<ul style="list-style-type: none"> • La aplicación mostro los datos según el parámetro de búsqueda seleccionado. Se ingreso la fecha de pago a cancelar. • Se oprimió el botón calcular. • La aplicación mostró en pantalla el valor de la cuota a cancelar. • Se oprimió el botón realizar pago. • La aplicación mostró un mensaje en pantalla indicando que el pago de la cuota ha sido registrado.
Caso 2 Pago no registrado	<ul style="list-style-type: none"> • La aplicación mostró los datos según el parámetro de búsqueda seleccionado. • Se ingreso la fecha de pago a cancelar. • Se oprimió el botón calcular. • La aplicación mostró en pantalla el valor de la cuota a cancelar. • Se oprimió el botón realizar pago. • La aplicación mostró un mensaje en pantalla indicando fallo en la conexión.
Prueba aprobada	Si

2.4.1.23 Caso de Prueba de Aceptación No 23: Registrar Ingreso.

Caso de prueba de aceptación Registrar Ingreso	
Nombre y código de las HU	<ul style="list-style-type: none"> • Registrar Ingreso 022
Descripción	El usuario administrador registrara a un nuevo ingreso realizado en la cooperativa.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _ Caso 1. • El administrador deberá seleccionar desde el menú principal el submenú de ingresos y escoger la opción Registrar Ingresos.

Entrada / Pasos de ejecución	
Caso 1 Registrar Ingreso	<ul style="list-style-type: none"> • Se ingresan los datos de del ingreso. • Se oprime el botón registrar.
Caso 2 Ingreso no registrado	<ul style="list-style-type: none"> • Se ingresan los datos del ingreso. • Se oprime el botón registrar.
Resultado esperado	
Caso 1 Registrar Ingreso	<ul style="list-style-type: none"> • La aplicación muestre un mensaje en pantalla indicando que se ha registrado el nuevo ingreso.
Caso 2 Ingreso no registrado	<ul style="list-style-type: none"> • La aplicación debe mostrar en pantalla un mensaje de error indicando por qué no se registro el ingreso.
Evaluación de la prueba	
Caso 1 Registrar Ingreso	<ul style="list-style-type: none"> • La aplicación mostró en pantalla un mensaje indicando que el ingreso se almaceno correctamente.
Prueba aprobada	Si

2.4.1.24 Caso de Prueba de Aceptación No 24: Consultar Ingresos.

Caso de prueba de aceptación		Consultar Ingresos
Nombre y código de las HU	<ul style="list-style-type: none"> • Consultar Ingresos 023 	
Descripción	Se realiza por parte del administrador la consulta de los ingresos registrados en la cooperativa.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Registrar Ingreso_ Caso 1 • El usuario deberá seleccionar desde el menú principal el submenú de préstamos y escoger la opción Gestión de Ingresos. 	
Entrada / Pasos de ejecución		
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda 	
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda 	

Resultado esperado	
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> La aplicación muestre en pantalla los datos de los ingresos realizados según el parámetro de búsqueda seleccionado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> La aplicación muestre en pantalla un mensaje indicando que no se encontraron registros que coincidan con el parámetro de búsqueda seleccionado. La aplicación permita la selección de nuevos parámetros de búsqueda.
Evaluación de la prueba	
Caso 1 Parámetro de búsqueda encontrado	<ul style="list-style-type: none"> La aplicación mostró en pantalla los datos de los ingresos realizados según el parámetro de búsqueda seleccionado.
Caso 2 Parámetro de búsqueda no encontrado	<ul style="list-style-type: none"> La aplicación mostró en pantalla un mensaje indicando que no se encontraron registros que coincidan La aplicación permitió la selección de nuevos parámetros de búsqueda.
Prueba aprobada	Si

2.4.1.25 Caso de Prueba de Aceptación No 25: Modificar Ingresos.

Caso de prueba de aceptación		Modificar Ingresos
Nombre y código de las HU	<ul style="list-style-type: none"> Modificar Ingresos 024 	
Descripción	<ul style="list-style-type: none"> El administrador realiza la modificación de los datos del ingreso realizado. 	
Condiciones de Ejecución	<ul style="list-style-type: none"> PDA Iniciar la Aplicación _ Caso 1. PDA Iniciar Sesión _ Caso 1. PDA Registrar Ingreso _ Caso 1. El usuario administrador deberá seleccionar desde el menú principal el submenú de préstamos y escoger la opción Gestión de Ingresos. 	
Entrada / Pasos de ejecución		
Caso 1 Editar Ingreso	<ul style="list-style-type: none"> Se selecciona un parámetro de búsqueda. Se selecciona el registro del Ingreso a modificar. Se escoge la opción de editar. 	

	<ul style="list-style-type: none"> • Se espera que la aplicación muestre en pantalla los datos de los ingresos que se pueden cambiar. • Se ingresan los datos que se desea cambiar. • Se oprime el botón Actualizar datos.
Caso 2 Edición no registrada	<ul style="list-style-type: none"> • Se selecciona un parámetro de búsqueda. • Se escoge la opción de editar. • Se espera que la aplicación muestre en pantalla los datos de los ingresos que se pueden cambiar. • Se ingresan los datos que se desea cambiar. • Se oprime el botón de Actualizar datos.
Resultado esperado	
Caso 1 Modificar Ingreso	<ul style="list-style-type: none"> • La aplicación muestra en pantalla los datos de los ingresos realizados según el parámetro de búsqueda seleccionado. • La aplicación muestre los datos de los ingresos que se pueden modificar. • La aplicación muestre un mensaje indicando que los datos de los Ingresos han sido modificados.
Caso 2 Edición no registrada	<ul style="list-style-type: none"> • La aplicación muestre en pantalla los datos de los ingresos realizados según el parámetro de búsqueda seleccionado. • La aplicación muestre los datos de los ingresos que se pueden modificar. • Se muestre en pantalla un mensaje indicando por qué no se puede registrar la modificación de los datos del ingreso.
Evaluación de la prueba	
Caso 1 Modificar Ingreso	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los ingresos realizados según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del ingreso a modificar. • Se seleccionó la opción editar del ingreso, • La aplicación mostró en pantalla un mensaje indicando que los datos del ingreso se actualizaron correctamente.
Caso 2 Edición no registrada	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos de los ingresos realizados según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del ingreso a modificar. • Se seleccionó la opción editar del ingreso,

	<ul style="list-style-type: none"> La aplicación mostró un mensaje en pantalla indicando que hay fallo en la conexión.
Prueba aprobada	Si

2.4.1.26 Caso de Prueba de Aceptación No 26: Eliminar Ingresos.

Caso de prueba de aceptación		Eliminar Ingresos
Nombre y código de las HU	<ul style="list-style-type: none"> Eliminar Ingresos 024 	
Descripción	El administrador realiza la eliminación de los ingresos realizados en la cooperativa.	
Condiciones de Ejecución	<ul style="list-style-type: none"> PDA Iniciar la Aplicación _ Caso 1. PDA Iniciar Sesión _Caso 1. PDA Registrar Ingreso _Caso 1 El usuario administrador deberá seleccionar desde el menú principal el submenú de ingresos y escoger la opción Gestión de Ingresos. 	
Entrada / Pasos de ejecución		
Caso 1 Eliminar Ingreso	<ul style="list-style-type: none"> Se selecciona un parámetro de búsqueda. Se selecciona el registro del ingreso a eliminar. Se selecciona la opción eliminar del ingreso. 	
Caso 2 Eliminación no registrada	<ul style="list-style-type: none"> Se selecciona un parámetro de búsqueda. Se espera hasta que la aplicación muestre los datos según el parámetro de búsqueda seleccionado. Se selecciona la opción eliminar. 	
Resultado esperado		
Caso 1 Eliminar Ingreso	<ul style="list-style-type: none"> Se muestren los datos según el parámetro de búsqueda seleccionado. Se muestre en pantalla un mensaje que permita confirmar la eliminación del ingreso. Se muestre en pantalla un mensaje de éxito indicando que la eliminación ha sido realizada. 	
Caso 2 Eliminación no registrada	<ul style="list-style-type: none"> La aplicación muestre los datos según el parámetro de búsqueda seleccionado. Se muestre en pantalla un mensaje que informe por que no se ha eliminado el ingreso. 	

Evaluación de la prueba	
Caso 1 Eliminar Ingreso	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos del ingreso según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del ingreso a eliminar. • Se seleccionó la opción de eliminar. • La aplicación mostró un mensaje en pantalla de confirmación de la eliminación. • Se seleccionó la opción continuar del mensaje. • La aplicación mostró en pantalla un mensaje indicando que se ha realizado la eliminación del ingreso.
Caso 2 Eliminación no registrada	<ul style="list-style-type: none"> • La aplicación mostró en pantalla los datos del ingreso según el parámetro de búsqueda seleccionado. • Se seleccionó el registro del ingreso a eliminar. • Se seleccionó la opción de eliminar. • La aplicación mostró un mensaje en pantalla de confirmación de la eliminación. • Se seleccionó la opción continuar del mensaje. • La aplicación mostró un mensaje de error indicando fallo en la conexión.
Prueba aprobada	Si

2.4.1.27 Caso de Prueba de Aceptación No 27: Reporte de afiliados.

Caso de prueba de aceptación		Reporte de afiliados
Nombre y código de las HU	• Reporte de afiliados 034	
Descripción	El usuario (Administrador, Auxiliar Administrativo o Afiliado) realiza la consulta de los datos de los afiliados.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Registro de Formularios de afiliación _Caso 1. • El usuario deberá seleccionar desde el menú principal afiliados la opción Reporte de Afiliados. 	
Entrada / Pasos de ejecución		
Caso 1 Configurar el reporte de afiliados	<ul style="list-style-type: none"> • Se selecciona la información que se desea consultar • Se oprime el botón generar reporte. • Se espera que la aplicación muestra en pantalla el reporte según los datos seleccionados 	

	<ul style="list-style-type: none"> • Se selecciona la opción de imprimir. • se selecciona la opción de guardar reporte.
Resultado esperado	
Caso 1 Configurar el reporte de afiliados	<ul style="list-style-type: none"> • La aplicación muestre el reporte según los datos seleccionados. • La aplicación permita seleccionar la opción de impresión del reporte. • La aplicación permita la selección de un dispositivo de almacenamiento para guardar el reporte.
Evaluación de la prueba	
Caso 1 Configurar el reporte de afiliados	<ul style="list-style-type: none"> • La aplicación permitió la selección de un tipo de información como parámetro de búsqueda. • Se oprimió el botón generar reporte • La aplicación mostró en pantalla el reporte en formato pdf según los parámetros de búsqueda. • Se seleccionó la opción de impresión del reporte y se realizó la impresión del reporte. • Se seleccionó la opción de guardar el reporte y se guardó una copia del reporte en una memoria usb.
Prueba aprobada	Si

2.4.1.28 Caso de Prueba de Aceptación No 28: Reporte de usuarios.

Caso de prueba de aceptación		Reporte de usuarios
Nombre y código de las HU	<ul style="list-style-type: none"> • Reporte de aportes 035 	
Descripción	El usuario (Administrador, Auxiliar Administrativo) realiza la consulta de los datos de los usuarios de la aplicación.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Crear Usuario _Caso 1. • El usuario deberá seleccionar desde el menú principal usuarios; la opción Reporte de Usuarios. 	
Entrada / Pasos de ejecución		
Caso 1 Configurar el reporte de aportes	<ul style="list-style-type: none"> • Se oprime el botón generar reporte. • Se espera que la aplicación muestre en pantalla el reporte. • Se selecciona la opción de imprimir. 	

	<ul style="list-style-type: none"> Se selecciona la opción de guardar reporte.
Resultado esperado	
Caso 1 Configurar el reporte de aportes	<ul style="list-style-type: none"> La aplicación muestre en pantalla el reporte Se permita la impresión del reporte. Se permita la selección de un dispositivo de almacenamiento para guardar el reporte.
Evaluación de la prueba	
Caso 1 Configurar el reporte de aportes	<ul style="list-style-type: none"> Se oprimió el botón generar reporte. La aplicación mostró en pantalla el reporte en formato pdf. Se imprimió el reporte. Se guardó una copia del reporte en una memoria usb.
	<ul style="list-style-type: none">
Prueba aprobada	Si

2.4.1.29 Caso de Prueba de Aceptación No 29: Reporte de aportes.

Caso de prueba de aceptación		Reporte de aportes
Nombre y código de las HU	<ul style="list-style-type: none"> Reporte de aportes 035 	
Descripción	El usuario (Administrador, Auxiliar Administrativo o Afiliado) realiza la consulta de los aportes de los afiliados.	
Condiciones de Ejecución	<ul style="list-style-type: none"> PDA Iniciar la Aplicación _ Caso 1. PDA Iniciar Sesión _Caso 1. PDA Registro de Aportes _Caso 1. El usuario deberá seleccionar desde el menú principal ingresos; el submenú de aportes y de este la opción Reporte de Aportes. 	
Entrada / Pasos de ejecución		
Caso 1 Configurar el reporte de aportes	<ul style="list-style-type: none"> Se ingresa la fecha de inicio como parámetro de búsqueda. Se ingresa la fecha de fin como parámetro de búsqueda. Se selecciona el tipo de aporte como parámetro de búsqueda. Se oprime el botón generar reporte. Se espera que la aplicación muestra en pantalla el 	

	<p>reporte según los parámetros de búsqueda.</p> <ul style="list-style-type: none"> • Se selecciona la opción de imprimir. • Se selecciona la opción de guardar reporte.
Resultado esperado	
Caso 1 Configurar el reporte de aportes	<ul style="list-style-type: none"> • La aplicación muestre en pantalla el reporte según los parámetros de búsqueda. • Se permita la impresión del reporte. • Se permita la selección de un dispositivo de almacenamiento para guardar el reporte.
Evaluación de la prueba	
Caso 1 Configurar el reporte de aportes	<ul style="list-style-type: none"> • Se ingreso la fecha de inicio y fin. • Se seleccionó un tipo de aporte. • Se oprimió el botón generar reporte. • La aplicación mostró en pantalla el reporte en formato pdf. • Se imprimió el reporte. • Se guardó una copia del reporte en una memoria Usb.
Prueba aprobada	Si

2.4.1.30 Caso de Prueba de Aceptación No 30: Reporte de préstamos.

Caso de prueba de aceptación		Reporte de prestamos
Nombre y código de las HU	<ul style="list-style-type: none"> • Reporte de prestamos 036 	
Descripción	El usuario (Administrador, Auxiliar Administrativo) realiza la consulta de los préstamos realizados por los afiliados.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Registrar Préstamos _Caso 1. • El usuario deberá seleccionar desde el menú principal egresos; el submenú de prestamos y de este la opción Reporte de Prestamos. 	
Entrada / Pasos de ejecución		
Caso 1 Configurar el reporte de prestamos	<ul style="list-style-type: none"> • Se ingresa la fecha de inicio como parámetro de búsqueda. • Se ingresa la fecha de fin como parámetro de búsqueda. • Se selecciona el tipo de préstamo como parámetro de búsqueda. 	

	<ul style="list-style-type: none"> • Se oprime el botón generar reporte. • Se espera que la aplicación muestra en pantalla el reporte los parámetros de búsqueda. • Se selecciona la opción de imprimir reporte. • Se selecciona la opción de guardar reporte.
Resultado esperado	
Caso 1 Configurar el reporte de prestamos	<ul style="list-style-type: none"> • La aplicación muestre en pantalla el reporte según el periodo de tiempo ingresado y el tipo de préstamo seleccionado • Se permita la impresión del reporte. • Se permita la selección de un dispositivo de almacenamiento para guardar el reporte.
Evaluación de la prueba	
Caso 1 Configurar el reporte de prestamos	<ul style="list-style-type: none"> • Se ingresó la fecha de inicio y fin. • Se seleccionó un tipo de préstamo. • Se oprimió el botón generar reporte. • La aplicación mostró en pantalla el reporte en formato pdf. • Se imprimió el reporte. • Se guardó una copia del reporte en una memoria usb.
Prueba aprobada	Si

2.4.1.31 Caso de Prueba de Aceptación No 31: Reporte de pagos.

Caso de prueba de aceptación		Reporte de pagos
Nombre y código de las HU	<ul style="list-style-type: none"> • Reporte de pagos 037 	
Descripción	El usuario (Administrador, Auxiliar Administrativo o Afiliado) realiza la consulta de los pagos de las cuotas de los préstamos de los afiliados.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación – Caso 1 • PDA Iniciar Sesión _Caso 1. • PDA Registrar Pagos _Caso 1. • El usuario deberá seleccionar desde el menú principal ingresos; el submenú de pagos y de este la opción Reporte de Pagos. 	
Entrada / Pasos de ejecución		
Caso 1 Configurar	<ul style="list-style-type: none"> • Se ingresa la fecha de inicio como parámetro de búsqueda. 	

el reporte de pagos	<ul style="list-style-type: none"> • Se ingresa la fecha de fin como parámetro de búsqueda. • Se selecciona el tipo de préstamo como parámetro de búsqueda. • Se oprime el botón generar reporte. • Se espera que la aplicación muestra en pantalla el reporte según los parámetros de búsqueda. • Se selecciona la opción de imprimir reporte. • Se selecciona la opción de guardar reporte.
Resultado esperado	
Caso 1 Configurar el reporte de pagos	<ul style="list-style-type: none"> • La aplicación muestre en pantalla el reporte según los parámetros de búsqueda. • Se permita la impresión del reporte. • Se permita la selección de un dispositivo de almacenamiento para guardar el reporte.
Evaluación de la prueba	
Caso 1 Configurar el reporte de pagos	<ul style="list-style-type: none"> • Se ingreso la fecha de inicio y fin. • Se seleccionó un tipo de préstamo. • Se oprimió el botón generar reporte. • La aplicación mostró en pantalla el reporte en formato pdf. • Se imprimió el reporte. • Se guardó una copia del reporte en una memoria usb.
Prueba aprobada	Si

2.4.1.32 Caso de Prueba de Aceptación No 32: Reporte de egresos.

Caso de prueba de aceptación		Reporte de egresos
Nombre y código de las HU	<ul style="list-style-type: none"> • Reporte de egresos 038 	
Descripción	El usuario (Administrador, Auxiliar Administrativo) realiza la consulta de los egresos.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Registrar Egresos _Caso 1. • El usuario deberá seleccionar desde el menú principal egresos; el submenú egresos varios y de este la opción Reporte de Egresos Varios. 	
Entrada / Pasos de ejecución		

Caso 1 Configurar el reporte de egresos	<ul style="list-style-type: none"> • Se ingresa la fecha de inicio como parámetro de búsqueda. • Se ingresa la fecha de fin como parámetro de búsqueda. • Se oprime el botón Generar Reporte. • Se espera que la aplicación muestre en pantalla el reporte según los parámetros de búsqueda ingresados
Resultado esperado	
Caso 1 Configurar el reporte de egresos	<ul style="list-style-type: none"> • La aplicación muestre en pantalla el reporte según los parámetros de búsqueda. • Se permita la impresión del reporte. • Se permita la selección de un dispositivo de almacenamiento para guardar el reporte.
Evaluación de la prueba	
Caso 1 Configurar el reporte de egresos	<ul style="list-style-type: none"> • Se ingresó la fecha de inicio y fin. • Se oprimió el botón generar reporte. • La aplicación mostró en pantalla el reporte en formato pdf. • Se imprimió el reporte. • Se guardó una copia del reporte en una memoria usb.
Prueba aprobada	Si

2.4.1.33 Caso de Prueba de Aceptación No 33: Reporte de ingresos.

Caso de prueba de aceptación		Reporte de ingresos
Nombre y código de las HU	<ul style="list-style-type: none"> • Reporte de ingresos 038 	
Descripción	El usuario (Administrador, Auxiliar Administrativo o Afiliado) realiza la consulta de los ingresos.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • PDA Registrar Ingresos _Caso 1 • El Administrador deberá seleccionar desde el menú principal ingresos; el submenú ingresos varios y de este la opción Reporte de Ingresos Varios. 	
Entrada / Pasos de ejecución		
Caso 1 Configurar	<ul style="list-style-type: none"> • Se ingresa la fecha de inicio como parámetro de búsqueda. 	

el reporte de egresos	<ul style="list-style-type: none"> • Se ingresa la fecha de fin como parámetro de búsqueda. • Se oprime el botón Generar Reporte. • Se espera que la aplicación muestre en pantalla el reporte según los parámetros de búsqueda ingresados.
	•
Resultado esperado	
Caso 1 Configurar el reporte de egresos	<ul style="list-style-type: none"> • La aplicación muestre en pantalla el reporte según los parámetros de búsqueda. • Se permita la impresión del reporte. • Se permita la selección de un dispositivo de almacenamiento para guardar el reporte.
	•
Evaluación de la prueba	
Caso 1 Configurar el reporte de egresos	<ul style="list-style-type: none"> • Se ingresó la fecha de inicio y fin. • Se oprimió el botón generar reporte. • La aplicación mostró en pantalla el reporte en formato pdf. • Se imprimió el reporte. • Se guardó una copia del reporte en una memoria usb.
Prueba aprobada	Si

2.4.1.34 Caso de Prueba de Aceptación No 34: Importar base de datos.

Caso De Prueba De Aceptación		Importar bases de datos
Nombre y código de las HU		• Importar bases de datos 040
Descripción		El usuario administrador importara una base de datos.
Condiciones de Ejecución		<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • El administrador deberá seleccionar desde el menú principal herramientas; el submenú bases de datos y de este seleccionar la opción Importar Bases De Datos.
Entrada / Pasos de ejecución		
Caso 1 Importar bases de		<ul style="list-style-type: none"> • Se selecciona el archivo que desea importar. • Se selecciona el nombre de la tabla que contendrá los datos del archivo que desea importar.

datos	<ul style="list-style-type: none"> • Se oprime el botón importar • Se espera que la aplicación inicie la importación.
Caso 2 Formato del archivo incorrecto	<ul style="list-style-type: none"> • Se selecciona el archivo que desea importar. • Se selecciona el nombre de la tabla que contendrá los datos del archivo que desea importar. • Se oprime el botón importar. • Se espera que la aplicación inicie la importación.
Resultado esperado	
Caso 1 Importar bases de datos	<ul style="list-style-type: none"> • La aplicación permita la selección del archivo a importar. • Se muestre un mensaje en pantalla indicando el resultado de la operación.
Caso 2 Formato del archivo incorrecto	<ul style="list-style-type: none"> • La aplicación permita la selección del archivo a importar. • La aplicación muestre un mensaje en pantalla indicando que el formato del archivo debe ser xls.
Evaluación de la prueba	
Caso 1 Importar bases de datos	<ul style="list-style-type: none"> • La aplicación mostró un cuadro de dialogo seleccionar archivo para ubicar el archivo a importar. • Se realizo la selección del archivo. • Se seleccionó el nombre de la tabla que ira a contener los datos del archivo seleccionado. • Se oprimió el botón importar • Mostró en pantalla un mensaje indicando el número de registros ingresados a la base de datos y el número de registros que no se pudieron ingresar.
Caso 2 Formato del archivo incorrecto	<ul style="list-style-type: none"> • La aplicación mostró un cuadro de dialogo seleccionar archivo para ubicar el archivo a importar. • Se realizo la selección del archivo. • Se seleccionó el nombre de la tabla que ira a contener los datos del archivo seleccionado. • Se oprimió el botón importar. • Mostró en pantalla un mensaje indicando que el formato del archivo a importar debe ser xls.
Prueba aprobada	Si

2.4.1.35 Caso de Prueba de Aceptación No 35: Exportar base de datos.

Caso De Prueba De Aceptación	Exportar bases de datos
-------------------------------------	--------------------------------

Nombre y código de las HU	<ul style="list-style-type: none"> • Exportar bases de datos 041
Descripción	El usuario administrador exportara una base de datos.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • El administrador deberá seleccionar desde el menú principal herramientas; el submenú bases de datos y de este seleccionar la opción Exportar Bases De Datos.
Entrada / Pasos de ejecución	
Caso 1 Exportar bases de datos	<ul style="list-style-type: none"> • Se ubica la unidad donde será guardado el archivo a exportar. • Se ingresa un nombre para el archivo. • Se selecciona el formato del archivo. • Se selecciona la tabla a exportar. • Se oprime el botón exportar. • Se espera que la aplicación realice la exportación de la base de datos.
Resultado esperado	
Caso 1 Exportar bases de datos	<ul style="list-style-type: none"> • La aplicación permita la ubicación de la unidad donde será guardado el archivo de la base de datos. • La aplicación genere el archivo en el formato seleccionado.
Evaluación de la prueba	
Caso 1 Exportar bases de datos	<ul style="list-style-type: none"> • La aplicación permitió el ingreso de un nombre para el archivo a exportar. • Se seleccionó el nombre de la tabla a exportar. • Se dio clic al botón Exportar. • El archivo generado se guardó en la carpeta de descargas del navegador.
Prueba aprobada	Si

2.4.1.36 Caso de prueba de aceptación No 36: Visualizar bitácora.

Caso De Prueba De Aceptación		Visualizar bitácora
Nombre y código de las HU	<ul style="list-style-type: none"> • Visualizar bitácora 041 	
Descripción	El usuario administrador revisara la	

	bitácora.
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • El administrador deberá seleccionar desde el menú principal herramientas; la opción Bitácora.
Entrada / Pasos de ejecución	
Caso 1 Ver bitácora	<ul style="list-style-type: none"> • Se ingresa la fecha de inicio como parámetro de búsqueda. • Se ingresa la fecha de fin como parámetro de búsqueda • Se da click sobre la lupa. • Se espera que la aplicación muestre en pantalla los datos según los parámetros de búsqueda ingresados.
Caso 2 Generar Reporte	<ul style="list-style-type: none"> • Se ingresa la fecha de inicio como parámetro de búsqueda. • Se ingresa la fecha de fin como parámetro de búsqueda • Se oprime el botón Generar reporte • Se espera que la aplicación muestre en pantalla los datos según los parámetros de búsqueda ingresados.
Resultado esperado	
Caso 1 Ver bitácora	<ul style="list-style-type: none"> • La aplicación muestre en pantalla los datos según los parámetros de búsqueda.
Caso 2 Generar Reporte	<ul style="list-style-type: none"> • La aplicación muestre en pantalla el reporte según los parámetros de búsqueda. • Se permita la impresión del reporte. • Se permita la selección de un dispositivo de almacenamiento para guardar el reporte.
Evaluación de la prueba	
Caso 1 Ver bitácora	<ul style="list-style-type: none"> • Se ingresó la fecha de inicio y fin. • Se oprimió la lupa. • La aplicación mostro en pantalla el reporte según los parámetros de búsqueda. • Se imprimió el reporte • Se guardó una copia del reporte en una memoria usb.
Caso 2 Generar Reporte	<ul style="list-style-type: none"> • Se ingresó la fecha de inicio y fin. • Se oprimió el botón generar reporte. • La aplicación mostró en pantalla el reporte según los parámetros de búsqueda.

	<ul style="list-style-type: none"> • Se imprimió el reporte • Se guardo una copia del reporte en una memoria usb.
Prueba aprobada	Si

2.4.1.37 Caso de prueba de aceptación No 37: Generar copia de seguridad.

Caso De Prueba De Aceptación		Generar copia de seguridad
Nombre y código de las HU		<ul style="list-style-type: none"> • Generar copia de seguridad 043
Descripción	El usuario administrador ingresara a la sección de generar una copia de seguridad.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • El administrador deberá seleccionar desde el menú principal herramientas; el submenú copias de seguridad y de este seleccionar la opción Crear Copia de Seguridad. 	
Entrada / Pasos de ejecución		
Caso 1 Generar copia de seguridad	<ul style="list-style-type: none"> • La aplicación debe generar automáticamente el nombre del archivo. • Se oprime el botón crear copia de seguridad. • Se esperar que la aplicación cree la copia de seguridad. 	
Resultado esperado		
Caso 1 Generar copia de seguridad	<ul style="list-style-type: none"> • La aplicación genere automáticamente el nombre del archivo. • La aplicación cree una copia de seguridad. • La aplicación muestre un mensaje indicando que la copia se ha creado correctamente. 	
Evaluación de la prueba		
Caso 1 Generar copia de seguridad	<ul style="list-style-type: none"> • Al ingresar la aplicación suministro automáticamente un nombre para la copia de seguridad • Se oprimió el botón crear copia • La aplicación mostró una barra de progreso donde indica el nivel de generación de la copia. 	
Prueba aprobada	Si	

2.4.1.38 Caso de prueba de aceptación No 38: Restaurar copia de seguridad.

Caso De Prueba De Aceptación		Restaurar copia de seguridad
Nombre y código de las HU	<ul style="list-style-type: none"> Restaurar copia de seguridad 044 	
Descripción	El usuario administrador realizara la restauración de la base de datos.	
Condiciones de Ejecución	<ul style="list-style-type: none"> PDA Iniciar la Aplicación _ Caso 1. PDA Iniciar Sesión _Caso 1. El administrador deberá seleccionar desde el menú principal herramientas el submenú copias de seguridad y luego de este seleccionar la opción Restaurar Copia de Seguridad. 	
Entrada / Pasos de ejecución		
Caso 1 Restaurar copia de seguridad	<ul style="list-style-type: none"> Se selecciona el archivo de la copia de seguridad. Se oprime el botón restaurar copia. Se espera que la aplicación restaure la copia de seguridad. 	
Resultado esperado		
Caso 1 Restaurar copia de seguridad	<ul style="list-style-type: none"> La aplicación permita ubicar y seleccionar el archivo que contiene la copia de seguridad. La aplicación restaure la copia de seguridad. La aplicación informe que la restauración de la copia de seguridad se ha realizado correctamente. 	
Evaluación de la prueba		
Caso 1 Restaurar copia de seguridad	<ul style="list-style-type: none"> La aplicación mostro un cuadro de dialogo donde permitió la ubicación y selección del archivo que contenía la copia de seguridad La aplicación mostró en pantalla un mensaje de confirmación para realizar la copia. Se mostró en pantalla un mensaje indicando que se cerrara el sistema para aplicar los cambios realizados. 	
Prueba aprobada	Si	

2.4.1.39 Caso de prueba de aceptación No 39: Ayuda de la Aplicación.

Caso De Prueba De Aceptación		Ayuda de la Aplicación
Nombre y código de las HU	<ul style="list-style-type: none"> • Ayuda de la aplicación 045 	
Descripción	El usuario seleccionara la opción de ayuda para consultar la forma de realizar alguna actividad en la aplicación.	
Condiciones de Ejecución	<ul style="list-style-type: none"> • PDA Iniciar la Aplicación _ Caso 1. • PDA Iniciar Sesión _Caso 1. • El administrador deberá seleccionar desde el menú principal el submenú de herramientas la opción Ayuda. 	
Entrada / Pasos de ejecución		
Consultar tema	<ul style="list-style-type: none"> • Dar click en el tema que desea consultar. • Esperar que la aplicación muestre el tema seleccionado. 	
Resultado esperado		
Consultar tema	<ul style="list-style-type: none"> • La aplicación muestre los temas que se pueden consultar. • Que la aplicación muestre el tema seleccionado. 	
Evaluación de la prueba		
Consultar tema	<ul style="list-style-type: none"> • La aplicación mostró la lista de temas que se podían consultar • La aplicación permitió la selección de un tema • La aplicación mostró en pantalla la información del tema seleccionado 	
Prueba aprobada	Si	

3 CONCLUSIONES

- Haciendo uso del módulo de afiliados, es posible sistematizar el registro manual de los afiliados, para mayor facilidad de búsqueda, modificación y eliminación de la información de cada uno de ellos.
- La sección de aportes de la aplicación permite la administración adecuada de la información de los aportes económicos de los afiliados.
- La aplicación en cada uno de sus módulos genera reportes que facilitan la consulta adecuada, ágil y clara de la información de los afiliados en la cooperativa.
- Se han establecido perfiles de usuario de tal manera que puedan acceder solo a las secciones que les corresponde, con lo que se garantiza un uso seguro de la aplicación web, permitiendo así al afiliado que las actividades de manejo sean fáciles de realizar.
- El módulo de herramientas cuenta con una sección que garantiza la seguridad de los datos. ya que permite generar copias de seguridad que luego pueden ser restauradas en caso que se presenten errores.
- La aplicación tiene la facilidad de ser compatible con otros sistemas contables con los que cuenta la cooperativa por medio de la sección de importación y exportación de bases de datos, para el desarrollo ágil de los procesos

4 RECOMENDACIONES

- Realizar periódicamente las copias de seguridad de manera adecuada informando el día y hora en que fueron realizadas, para evitar retraso o inconvenientes en los procesos que realice la cooperativa.
- Asignar adecuadamente el perfil de usuario al personal que trabaje en la cooperativa, para garantizar la seguridad de la información que maneja la aplicación.
- Realizar mantenimiento periódico al equipo que será servidor para evitar fallas en la prestación de servicio a los afiliados.
- Suministrar a los afiliados, los datos correctos de la cuenta del sistema para el acceso a la aplicación esta debe ser labor del personal encargado del registro.

BIBLIOGRAFIA

BECK, KENT. Una Explicación de la Programación Extrema: Aceptar el Cambio. Addison-Wesley. 2002.

J.NEWKIRK/R. C. MARTIN, La Programación Extrema en la Práctica. Addison Wesley. 2001.

ENTREVISTA con Carlos Antonio Delgado gerente de la cooperativa COOTRESCO, el 2 de Junio de 2011

Estatutos de la COOPERATIVA DE TRABAJADORES DE LA REGIONAL DE SALUD CENTRO OCCIDENTAL. "COOTRESCO"

Formularios de afiliación de la cooperativa COOTRESCO.

Formularios de afiliación de la COOPERATIVA DEL MAGISTERIO DE TUQUERRES LTDA. "COOACREMAT.LTDA"

Formularios de afiliación de la COOPERATIVA DE ASOCIACION DE MADRES COMUNITARIAS." COASOANDES"

Formularios de afiliación de la COOPERATIVA DE EDUCADORES DE LA SABANA LTDA. "COOESA".

<http://www.oracle.com/technetwork/java/overview-138580.html>

<http://www.programacionextrema.org>

LEY 79 DE 1988 NIVEL NACIONAL.

<http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=9211>

Búsqueda realizada el 17 de mayo de 2011

Estructura Básica de la Cooperativa

Publicación digital en la página web de la Biblioteca Luis Ángel Arango del Banco de la República.

<<http://www.banrepcultural.org/blaavirtual/ciencias/sena/cooperativismo/estructura-basica-de-la-cooperativa/estructura1.htm>>

Búsqueda realizada el 17 de mayo de 2011.

ANEXOS

Tabla de afiliados

Nombre de la Tabla				afiliados
Descripción		Guardar el registro de los datos personales de los afiliados de la cooperativa		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
cedula	varchar(15)	No	Si	Es el número de cedula del afiliado
lugar_nacimiento	varchar(25)	No	No	Es el lugar de nacimiento del afiliado
tipo_sangre	varchar(4)	No	No	Es el tipo de sangre del afiliado
nivel_educativo	varchar(15)	Si	No	Es el nivel de estudios realizado por el afiliado que (Ninguno, Bachiller, Técnico, Tecnológico, Universitario)
titulo_academico	varchar(50)	Si	No	Es el grado académico del afiliado
fecha_afiliacion	Date	No	No	Es la fecha de afiliación a la cooperativa
Índices únicos	Cedula			
Llave Primaria	Cedula			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de aportes

Nombre de la Tabla				aportes
Descripción		Almacenar la información correspondiente a los tipos de aporte que realizara el afiliado.		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
cedula_afiliado	varchar(15)	No	No	El número de cedula del afiliado que realizara el aporte.

id_tipo_aporte	varchar(2)	No	No	El código de identificación del tipo de aporte que realizara el afiliado.
valor_aporte	real (53)	No	No	La cantidad total a aportar.
dia_cobro	Integer	No	No	El día del mes en el que se debe realizar el aporte.
Estado	varchar(10)	No	No	Corresponde al estado del aporte pagado o sin pagar
Índices únicos				
Llave primaria	cedula_afiliado id_tipo_aporte			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	
id_tipo_aporte	tipos_aporte	tipo_aporte	Acceder a los tipos de aporte ya establecidos en la cooperativa.	

Tabla de beneficiarios

Nombre de la Tabla		beneficiarios		
Descripción		Guardar un registro de los datos de los beneficiarios del afiliado		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
identificación	varchar(15)	No	Si	Es el número de identificación del beneficiario
cedula_afiliado	varchar(15)	No	No	Es el número de cedula del afiliado
parentesco_afiliado	varchar(20)	No	No	Es el parentesco del afiliado y el beneficiario
porcentaje_beneficio	decimal	No	No	Es el porcentaje del total de aportes del afiliado que corresponde para el beneficiario
Índices únicos	Cedula			
Llave primaria	Cedula			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de bitácora

Nombre de la Tabla				bitacora
Descripción		Guardar un registro de todos los suceso ocurridos en la aplicación ingresos, salidas, registro de datos, modificaciones, consultas, eliminaciones etc.		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
fecha_suceso	Date	No	Si	La fecha y hora de los sucesos ocurridos en el sistema.
id_usuario	varchar(15)	No	No	El código del usuario; quien realizo alguna operación en la aplicación.
descripcion	Text	No	No	Es un comentario corto de lo ocurrido en ese instante con la aplicación.
Índices únicos	fecha_suceso			
Llave primaria	fecha_suceso			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	
id_usuario	usuarios	Identificación	Acceder al número de la identificación de los usuarios registrados.	

Tabla de cuentas

Nombre de la Tabla				cuentas
Descripción		Almacenar la información de los usuarios que accedan a la aplicación (personal que labora en la cooperativa , afiliados y publico en general).		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
identificacion	varchar(15)	No	Si	El número del documento de identificación del usuario.
login	varchar(20)	No	Si	El nombre que el Usuario utilizara para identificarse en la aplicación

password	varchar(20)	No	No	La contraseña de acceso a la aplicación
perfil_acceso	varchar(20)	No	No	El nivele de acceso a la aplicación (administrador, afiliado, visitante). Administrador del Sistema, Afiliado, Beneficiario, Empleado, Consultor
estado	varchar(10)	No	No	El estado del usuario (activo inactivo)
Índices únicos	Identificación			
Llave primaria	Identificación			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de cursos cooperativos

Nombre de la Tabla		 cursos_coo		
Descripción		Guardar un registro de los cursos cooperativos realizados por el afiliado		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
identificación	varchar(15)	No	No	Es el número de identificación del afiliado
nombre_curso	varchar(50)	No	No	Es el nombre del curso de cooperativismo realizado por el afiliado
entidad_educativa	varchar(50)	No	No	Es la entidad que certifica el curso de cooperativismo.
duracion_horas	Integer	No	No	Es el número de horas de duración del curso de cooperativismo.
fecha	Date	No	No	Fecha de realización del curso de cooperativismo
Índices únicos	Cedula			
Llave primaria	Cedula nombre_curso			

Llaves foráneas			
Atributo	Tabla referenciada	Atributo	Descripción

Tabla de departamentos

Nombre de la Tabla		departamentos		
Descripción		En este campo se almacena el nombre y el código respectivo de cada departamento.		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
codigo	varchar(2)	No	Si	El código del departamento.
nombre	varchar(30)	No	Si	El nombre del departamento.
Índices únicos	Código nombre			
Llave primaria	Código			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de egresos

Nombre de la Tabla		egresos		
Descripción		Guardar la información correspondiente a los egresos realizados por la cooperativa.		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
id_egreso	varchar(10)	No	Si	El código de identificación del egreso.
concepto	Text	No	No	Descripción breve del concepto por el cual se realizo el egreso.
valor	real(53)	No	No	Es la cantidad que sale de la cooperativa.
fecha	Date	No	No	Es la fecha que se realizo el egreso.
cedula	varchar(15)	Si	No	Es el número de la cedula de

				ciudadanía de la persona responsable por el egreso realizado.
id_prestamo	varchar(10)	si	No	Es el código de identificación del préstamo realizado por el afiliado
Índices únicos	id_egreso			
Llave primaria	id_egreso			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	
id_prestamo	prestamos	id_prestamo	Acceder a los préstamos realizados por los afiliados.	

Tabla de información económica

Nombre de la Tabla		inf_economica		
Descripción		Guardar un registro de todos los datos económicos de los afiliados de la cooperativa		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
cedula	varchar(15)	No	Si	Es el número de cedula del afiliado
salario	real(53)	No	No	Es el ingreso básico en pesos del afiliado
otros_ingresos	real(53)	Si	No	Es el total de otros ingresos en pesos obtenidos por el afiliado
Índices únicos	Cedula			
Llave primaria	Cedula			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de información financiera

Nombre de la Tabla	inf_financiera
--------------------	----------------

Descripción		Guardar un registro de todos los datos financieros de los afiliados de la cooperativa		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
cedula	varchar(15)	No	Si	Es el número de cedula del afiliado
entidad_financiera	varchar(20)	Si	No	Es el nombre del banco donde el afiliado tiene una cuenta
número_cuenta	varchar(50)	Si	No	Es el número de la cuenta del banco del afiliado
departamento	varchar(30)	Si	No	Es el nombre del departamento donde esta la entidad financiera del afiliado
municipio	varchar(30)	Si	No	Es el nombre del municipio donde esta la entidad financiera del afiliado
Índices únicos	Cedula			
Llave primaria	Cedula			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de ingresos

Nombre de la Tabla		ingresos		
Descripción		Guardar la información correspondiente a los ingresos realizados a la cooperativa.		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
id_ingreso	varchar(10)	No	Si	El código de identificación del ingreso.
concepto	Text	No	No	Descripción breve del concepto por el cual se realizo el ingreso.
valor	real(53)	No	No	Es la cantidad que ingresa a la cooperativa.
fecha	Date	No	No	Es la fecha que se realizo el

				ingreso.
cedula	varchar(15)	Si	No	Es el número de la cedula de ciudadanía de la persona responsable por el ingreso realizado.
id_aporte	varchar(10)	si	No	Es el código de identificación del aporte que se realizo en la cooperativa.
Id_pago_Prestamo	varchar(10)	Si	No	Si el ingreso es por concepto de pago de una cuota de un préstamo, se ingresa el código del pago.
Índices únicos	id_ingreso			
Llave Primaria	id_ingreso			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	
id_aporte	aportes_realizados	id_aporte	Acceder a los aportes realizados por los afiliados.	
Id_pago_Prestamo	plan_pagos		Acceder a los pagos realizados por los afiliados.	

Tabla de municipios

Nombre de la Tabla		municipios		
Descripción		En este campo se almacena el nombre y el código respectivo de cada Municipio de Colombia		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
codigo_departamento	varchar(2)	No	Si	El código del departamento.
codigo_municipio	varchar(3)	No	Si	El código del municipio.
nombre	varchar(30)	No	No	El nombre del municipio.
Índices únicos	codigo_departamento codigo_municipio			
Llave primaria	codigo_departamento codigo_municipio			

Llaves foráneas			
Atributo	Tabla referenciada	Atributo	Descripción

Tabla de parentesco

Nombre de la Tabla		parentesco		
Descripción		Almacena la información correspondiente al grado de consanguinidad entre personas.		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
Nombre	varchar(30)	No	Si	Tipo de parentesco
Índices únicos		Nombre		
Llave Primaria		Nombre		
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de personas

Nombre de la Tabla		personas		
Descripción		Almacenar la información de personas que tienen alguna relación con la cooperativa (personal que labora en la cooperativa, afiliados y público en general).		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
identificacion	varchar (15)	No	Si	El número del documento de identidad (No de Cédula, No Tarjeta de Identidad, No Cédula de extranjería) de la persona.
tipo_documento	varchar(2)	No	No	El nombre del documento de identidad (si es cédula, tarjeta de Identidad, o cédula de extranjería).

nombres	varchar(50)	No	No	El nombre de la persona.
apellidos	varchar(50)	No	No	El apellido de la persona.
fecha_nacimiento	date	No	No	La fecha de nacimiento de la persona.
estado_civil	varchar(12)	Si	No	El estado civil (Soltero, Casado, Divorciado, Viudo, Unión Libre) de la persona.
Barrio	varchar(50)	Si	No	El nombre del barrio donde reside la persona.
direccion	varchar(50)	No	No	La dirección donde reside la persona.
municipio	varchar(3)	No	No	El código del municipio donde reside la persona.
departamento	varchar(2)	No	No	El código del departamento donde reside la persona.
telefono	varchar(10)	Si	No	El número de teléfono donde reside la persona.
Celular	varchar(10)	Si	No	El número de celular de la persona.
e-mail	varchar(50)	Si	No	La dirección del correo electrónico de la persona.
fecha_ingreso	date	No	No	La fecha de ingreso a la aplicación.
Índices únicos	Identificación			
Llave primaria	Identificación			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de plan de pagos

Nombre de la Tabla				plan_pagos
Descripción		Guardar la información correspondiente a los pagos de las cuotas de los prestamos realizados por los afiliados		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
id_pago	varchar(10)	No	Si	El código de identificación de la

				cuota a pagar
Id_prestamo	varchar(10)	No	No	Es el código de identificación del préstamo realizado por el afiliado
Valor	real(53)	No	No	Es el valor de la cuota a pagar
fecha_estimada	Date	No	No	Es plazo en el que se debe pagar la cuota.
fecha_pago	Date	Si	No	Es la fecha en que se realizo el pago de la cuota
estado_pago	varchar(20)	Si	No	Es el estado de la cuota (pagada o no pagada)
Índices únicos	id_pago			
Llave primaria	id_pago			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	
id_prestamo	prestamos	id_prestamo	Acceder a los préstamos realizados por los afiliados.	

Tabla de prestamos

Nombre de la Tabla		prestamos		
Descripción		Guardar la información correspondiente a los prestamos realizados por los afiliados		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
id_prestamo	varchar(10)	No	Si	El código de identificación del préstamo realizado por el afiliado
cedula_afiliado	varchar(15)	No	No	Es el número de cedula del afiliado que realiza el préstamo.
id_tipo_prestamo	varchar(2)	No	No	Es el código de identificación de los tipos de prestamos que se pueden realizar en la cooperativa
cedula_codeudor	varchar(15)	No	No	Es el número de cedula del codeudor del préstamo que realiza el afiliado
valor_prestado	real(53)	No	No	Es el valor total aprobado del préstamo

Plazo	integer	No	No	Es el número de meses max que se da para el pago total del préstamo
valor_cuota	real(53)	No	No	Es el valor total a pagar de una cuota del préstamo realizado por el afiliado ((Valor Prestado / Número de meses o Plazo) + % de Interés Corriente)
fecha_solicitud	date	No	No	Es la fecha de recepción de la solicitud del préstamo.
fecha_aprobacion	date	No	No	Es la fecha de aprobación del préstamo solicitado por el afiliado
fecha_primer_pago	date	No	No	Es la fecha de pago de la primera cuota del préstamo solicitado por el afiliado
caso_especial	boolean	No	No	Si el préstamo es extraordinario (ya tiene aprobado y pagado hasta el 50% de un préstamo y solicita otro)
estado	varchar(10)			Si esta activo (esta en proceso de pago) inactivo(el valor total del préstamo ya esta cancelado)
Índices únicos	idprestamo			
Llave primaria	idprestamo			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	
id_tipo_prestamos	tipos_prestamos	id_aporte	Acceder a los tipos de prestamos registrados por los cooperativa	

Tabla de tipo de aporte

Nombre de la Tabla				tipos_aportes
Descripción		Almacenar la información correspondiente a los tipos de aporte que se pueden realizar los afiliados en la cooperativa.		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
id	varchar (2)	No	Si	El código que identifica al tipo de

				aporte.
nombre	varchar(50)	No	Si	Son los nombres de los aportes definidos por la cooperativa (educación de recreación y turismo, vivienda,..).
tipo_aporte	varchar(20)	No	No	Son los tipos de aporte (ordinario, extraordinario) que se pueden realizar en la cooperativa.
tipo_valor	varchar(20)	No	No	Es un porcentaje del SMLV que será aportado por el afiliado a la cooperativa.
frecuencia	varchar(15)	No	No	Es el periodo de tiempo de cobro del aporte (Mensual, Bimensual, Trimestral, Semestral, Ocasional)
estado	varchar(10)	No	No	Si el aporte esta vigente o no (Activo o inactivo).
Índices únicos	Id nombre			
Llave primaria	Id			
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	

Tabla de tipo de prestamos

Nombre de la Tabla		tipos_prestamos		
Descripción		Guardar un registro de todos los tipos de préstamos que se pueden realizar en la cooperativa		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
id	varchar(2)	No	Si	Es el código de identificación del tipo de préstamo
nombre	varchar(50)	No	Si	Es el nombre del préstamo (educación universitaria, vivienda, vacaciones)
tipo	varchar(20)	No	No	Es la clase de préstamo ordinario o extraordinario

proposito	varchar(50)	Si	No	Es la razón por la cual el afiliado solicita un préstamo(para vivienda, compra de electrodomésticos, vacaciones, e educación superior de familiares)
consecion	varchar(20)	No	No	Plazo de Concesión (Corto Plazo, Mediano Plazo, Largo Plazo)
monto_inferior	decimal(4)	No	No	Valor del Monto Inferior a prestar, en SMLMV. Ejemplo: 0.2
monto_superior	decimal(4)	Si	No	Valor del Monto Inferior a prestar, en SMLMV. Ejemplo: 1.8
valor_maximo_afiliado	decimal(4)	No	No	Porcentaje máximo a prestar según los aportes ordinarios del solicitante. (Ejemplo: Hasta un 500%. Del saldo de los aportes ordinarios)
valor_maximo_coo	decimal(4)	Si	No	Máximo valor que la cooperativa puede desembolsar para este préstamo. (Ejemplo: 5 SMLMV)
plazo_maximo	integer	No	No	Número de Meses máximo para pagar el préstamo.
primer_pago	integer	No	No	Número de días para que el afiliado realice el pago de la primera cuota del préstamo contraído (Ejemplo: en 30 días)
porcentaje_int_corr	decimal(4)	No	No	Es el valor del porcentaje de interés corriente a pagar
porcentaje_int_mora	decimal(4)	No	No	Es el valor del porcentaje de interés por mora a pagar en caso de incumplimiento del pago oportuno de la cuota del préstamo realizado por el afiliado
tipo_calculo_cuota	varchar(20)	No	No	Es el tipo de calculo que se realiza para definir el valor de las cuotas a pagar (puede ser el método Francés o Libre Ingreso)
Índices únicos	Id nombre			
Llave primaria	Id			
Llaves foráneas				

Atributo	Tabla referenciada	Atributo	Descripción

Tabla de tutores

Nombre de la Tabla		tutores		
Descripción		Almacena la información correspondiente a las personas que tendrán a su cuidado a las personas que hayan sido determinados como beneficiarios		
Atributos				
Nombre	Tipo de dato	Nulo	Único	Descripción
id_persona	integer	No	Si	Es el código de identificación que lo determina como persona
id_afiliado	integer	No	No	Es el código de identificación del afiliado
Índices únicos		Nombre		
Llave Primaria		Nombre		
Llaves foráneas				
Atributo	Tabla referenciada	Atributo	Descripción	
id_persona	afiliados	id_afiliado	Acceder a la tabla afiliados para relacionar el tutor con el afiliado.	