

**UNIVERSIDAD INTERNACIONAL DE LAS AMERICAS
ESCUELA DE INGENIERÍA INFORMÁTICA**

PROYECTO FINAL DE GRADUACIÓN

Para optar por el grado de Bachillerato en
Ingeniería de Software

**PROTOTIPO FUNCIONAL PARA LA GESTIÓN DE PROYECTOS DE LA
COOPERATIVA SULÁ BATSÚ R.L UBICADA EN BARRIO ESCALANTE**

STEVEN MORERA HERNÁNDEZ

AUTOR

FABIAN RODRIGUEZ SIBAJA

TUTOR

CARLOS CUEVA VILLALOBOS

LECTOR

San José, Costa Rica

Abril, 2019

CONTENIDO

ÍNDICE DE TABLAS	7
ÍNDICE DE FIGURAS	9
CARTA DE AUTORIZACIÓN DE LA DIRECCIÓN DE CARRERA	11
APROBACIÓN DEL TRIBUNAL EXAMINADOR.....	12
CARTA DE APROBACIÓN DEL TUTOR.....	13
CARTA DEL LECTOR	14
DECLARACIÓN JURADA	15
CÓDIGO DE ÉTICA	16
CARTA DE REVISIÓN FILÓLOGICA.....	17
Resumen Ejecutivo.....	18
CAPÍTULO I.....	22
Introducción	22
Planteamiento del Problema.....	22
Objetivos	23
General.	23
Específicos.	23
Justificación.....	23
Viabilidad Operativa.	24
Viabilidad Técnica.	24
Viabilidad Económica.	26
Viabilidad Legal.....	27
Proyecciones.....	28
Proyección funcional.....	28
Proyección tecnológica.....	31

Proyección metodológica	32
CAPÍTULO II	34
Marco Referencial	34
Software	34
Sistema operativo.	34
Aplicación.	35
POO.	36
Módulo.	37
Framework.	38
Materialize.....	39
HTML.....	40
PHP.....	40
JavaScript.	41
JQuery.	42
Ajax.	42
Base de Datos	44
Base de Datos.	44
MySQL.....	45
Tablas.	46
SQL.	46
Servicio de alojamiento web	47
Hosting (alojamiento web).	47
Contrato del nivel de servicio (SLA – Service Level Agreement).....	47
Servidor DNS.	48
Servidor web.	49
Servidor de aplicaciones.....	49

Términos del negocio	50
Proyecto.....	50
Activos.	50
Pasivos.....	51
Costos operativos.	53
Gastos administrativos u overhead.....	53
Excedentes.....	54
Balance de comprobación.	54
Balance de situación.....	55
Cuentas por pagar.....	56
Cuentas por cobrar.	57
CAPÍTULO III	58
Marco Metodológico	58
Enfoque de Investigación	58
Cuantitativa.	58
Cualitativa.	59
Mixto.	60
Método utilizado.	61
Tipos de Investigación	62
Descriptiva.	62
Exploratoria.....	62
Explicativa.....	63
Tipo utilizado.	63
Fuentes de Información.....	63
Primarias.....	63
Secundarias.....	64

Terciarias.....	64
Descripción de Variables.....	64
Variable Conceptual.....	64
Variable Operacional.....	65
Variable Instrumental.....	65
Población.....	68
Muestra.....	68
Instrumentos para la recolección de datos.....	69
Entrevista.....	69
Observación.....	70
Proceso para recolección y análisis de datos.....	70
CAPÍTULO IV.....	71
Análisis de Resultados.....	71
CAPÍTULO V.....	74
Conclusiones.....	74
Recomendaciones.....	75
CAPÍTULO VI.....	77
Propuesta.....	77
Análisis.....	77
Software desarrollado.....	77
Hardware requerido.....	78
Telecomunicaciones requeridas.....	79
Base de datos.....	80
Personal requerido.....	80
Casos de Uso.....	81
Diseño.....	130

Arquitectura del sistema.....	130
Arquitectura del software.....	131
Diseño de interfaces.....	132
Diseño de base de datos.....	137
Diccionario de base de datos.....	138
Diseño de procesos.....	162
Diseño de salidas.....	166
Diagramas UML.....	171
Programación.....	172
Entradas y salidas.....	172
Procesos.....	173
Validaciones.....	175
Módulos señalados.....	176
Pruebas.....	190
REFERENCIAS.....	77
APÉNDICES.....	273
Apéndice A: Guía de Entrevista.....	273
Apéndice B: Guía de Observación.....	274

ÍNDICE DE TABLAS

Tabla 1. Características del equipo de desarrollo utilizado.....	25
Tabla 2. Características del equipo necesario para el uso del prototipo.....	25
Tabla 3. Costos por servicios profesionales	27
Tabla 4. Cuadro de Variables	66
Tabla 5. Caso de uso autenticación de usuario	82
Tabla 6. Caso de uso mantenimiento asociados.....	83
Tabla 7. Caso de uso mantenimiento proyectos	86
Tabla 8. Caso de uso mantenimiento colaboradores.....	91
Tabla 9. Caso de uso mantenimientos clientes.....	94
Tabla 10. Caso de uso mantenimientos proveedores	97
Tabla 11. Caso de uso flujos	100
Tabla 12. Caso de uso cuentas por cobrar	104
Tabla 13. Caso de uso cuentas por pagar	107
Tabla 14. Caso de uso gastos operativos.....	109
Tabla 15. Caso de uso historial de transferencias	113
Tabla 16. Caso de uso asignar proyectos	116
Tabla 17. Caso de uso excedentes	120
Tabla 18. Caso de uso usuarios	123
Tabla 19. Caso de uso roles.....	126
Tabla 20. Caso de uso reportes.....	129
Tabla 21. Interfaz flujos	136
Tabla 22. Entidad usuarios	138
Tabla 23. Entidad roles.....	139
Tabla 24. Entidad asociados.....	139
Tabla 25. Entidad colaboradores	140
Tabla 26. Entidad proyectos.....	142
Tabla 27. Entidad rubros	143
Tabla 28. Entidad categoría rubros.....	143
Tabla 29. Entidad Rubro Proyecto	144
Tabla 30. Entidad excedentes.....	144

Tabla 31. Entidad historial transferencias	147
Tabla 32. Entidad asociado proyecto	148
Tabla 33. Entidad colaborador proyecto	148
Tabla 34. Entidad gastos operativos.....	149
Tabla 35. Entidad gastos operativos categoría.	151
Tabla 36. Entidad flujos ingresos.....	152
Tabla 37. Entidad flujo ingresos	153
Tabla 38. Entidad flujo ingresos proyecto.....	154
Tabla 39. Entidad flujo gastos ejecución.....	154
Tabla 40. Entidad flujo gastos ejecución categoría.....	156
Tabla 41. Entidad flujos gastos ejecución proyecto	157
Tabla 42. Entidad proveedores.....	157
Tabla 43. Entidad clientes	158
Tabla 44. Entidad cliente proyecto.....	159
Tabla 45. Entidad cuentas pagar.....	160
Tabla 46. Entidad cuentas cobrar	161
Tabla 47. Caso de prueba autenticación de usuario.....	190
Tabla 48. Caso de prueba mantenimiento asociados.....	192
Tabla 49. Caso de prueba mantenimiento proyectos.....	196
Tabla 50. Caso de prueba mantenimiento colaboradores.....	204
Tabla 51. Caso de prueba mantenimiento clientes	208
Tabla 52. Caso de prueba mantenimiento proveedores.....	212
Tabla 53. Caso de prueba flujos	216
Tabla 54. Caso de prueba cuentas por cobrar.....	228
Tabla 55. Caso de prueba cuentas por pagar.	233
Tabla 56. Caso de prueba gastos operativos.....	237
Tabla 57. Caso de prueba historial de transferencias.	243
Tabla 58. Caso de prueba asignar proyecto.....	248
Tabla 59. Caso de prueba excedentes.....	253
Tabla 60. Caso de prueba usuarios.....	259
Tabla 61. Caso de prueba roles	262
Tabla 62. Caso de prueba reportes	266

ÍNDICE DE FIGURAS

Figura 1. Etapas del modelo en cascada.....	32
Figura 2. Sistemas Operativos.....	35
Figura 3. Aplicaciones.....	35
Figura 4. Programación orientada a objetos.....	37
Figura 5. Frameworks.....	38
Figura 6. Logo materialize.....	39
Figura 7. Proceso de comunicación de una página web básica.....	43
Figura 8. Proceso de comunicación con Ajax.....	44
Figura 9. Tabla del debe y haber.....	52
Figura 10. Balance de comprobación.....	55
Figura 11. Balance de situación.....	56
Figura 12. Proceso de investigación cuantitativa.....	59
Figura 13. El proceso cualitativo.....	60
Figura 14. Método mixto.....	61
Figura 15. Modelo de casos de uso.....	81
Figura 16. Arquitectura de Sistema.....	131
Figura 17. Arquitectura de Software.....	132
Figura 18. Menú principal.....	133
Figura 19. Página de Inicio.....	134
Figura 20. Interfaz de mantenimiento.....	136
Figura 21. Diagrama Entidad – Relación 1.....	137
Figura 22. Proceso de flujo de ingresos o gastos de ejecución.....	162
Figura 23. Proceso de excedentes parte 1.....	164
Figura 24. Proceso de excedentes parte 2.....	165
Figura 25. Reporte gastos operativos.....	167
Figura 26. Reporte cuentas por pagar.....	168
Figura 27. Reporte de estado de proyecto por periodo.....	169
Figura 28. Reporte de distribución de excedentes.....	170
Figura 29. Diagrama de clases.....	171
Figura 30. Código entrada al insertar.....	172
Figura 31. Código de salida para mostrar registros.....	173

Figura 32. Código proceso flujo gastos ejecución	174
Figura 33. Código proceso excedentes.....	175
Figura 34. Código de validaciones	176
Figura 35. Código de seguridad	177
Figura 36. Código de seguridad 2	178
Figura 37. Código de mantenimientos JavaScript.....	179
Figura 38. Código mantenimientos PHP	179
Figura 39. Código de reportes JavaScript	180
Figura 40. Código reportes PHP	181
Figura 41. Código de consultas JavaScript	182
Figura 42. Código de consultas PHP	182
Figura 43. Código de cálculos financieros	183
Figura 44. Código de cálculos financieros 2	183
Figura 45. Código cuentas por pagar.....	184
Figura 46. Código alertas cuentas por pagar	185
Figura 47. Código cuentas por cobrar.	186
Figura 48. Código alertas cuentas por cobrar.....	187
Figura 49. Código asignaciones	188
Figura 50. Código insertar asignación y excedente.....	189
Figura 51. Código de cálculo excedente anual.....	190

DEDICATORIA

Quiero dedicar este trabajo a todas las personas que me dieron apoyo, tanto psicológico como moral a lo largo de mi carrera como estudiante de ingeniería de software, la cual fue una travesía bastante interesante, estresante en varios puntos y muy gratificante al final.

Sin embargo, en especial quiero dedicarle este proyecto final de graduación a mis padres Haydeé Hernández Pérez y Claudio Morera Ávila, ya que gracias a sus enseñanzas, influencia y constante apoyo de perseverar es como he llegado hasta este punto en mi vida.

Por otro lado, también quiero dedicarle este proyecto a Dios ya que gracias a Él es como todos los proyectos en vida tienen éxito y gracias a Él es que todos estamos aquí un día más.

AGRADECIMIENTO

Quiero agradecerle a toda mi familia por la constante ayuda que me brindaron y el soporte dado tanto emocional como financiero para poder completar todos mis estudios. Por otro lado, también quiero agradecer a mi tutor Fabian Rodriguez Sibaja el cual aclaro todas las dudas que tuve durante la elaboración del proyecto y la disponibilidad dada.

A su vez quiero agradecerle en conjunto a todos los profesores que me enseñaron tantas cosas en los diferentes cursos en la carrera de ingeniería de software, sin ellos no tendría todos los conocimientos que tengo hoy en día, muchas gracias.

Tambien agradecer a la Cooperativa Sulá Batsú por brindarme la oportunidad de realizar el proyecto dentro de su organización y a la atención e información brindada.

RESUMEN EJECUTIVO

En este proyecto se elaboró un proceso de investigación y desarrollo de un prototipo funcional el cual brinda a la Cooperativa Sulá Batsú R.L solucionar los atrasos en elaboración de distintos reportes sobre los proyectos que se realizan, reducir la carga de trabajo del recurso a cargo de éste y, a su vez, disminuir los riesgos posibles en la manipulación incorrecta de los cálculos realizados. Es por ello que se realizaron los siguientes capítulos

En el capítulo I, se inicia con una introducción sobre la Cooperativa, el planteamiento de los problemas a mayor detalle, los objetivos del proyecto, la justificación y las proyecciones se éste. Dentro de las justificaciones brindadas se encuentran las siguientes: operativa, técnica, económica y legal, donde en cada una de ellas se dicen las razones del porqué el proyecto es viable para la Cooperativa, es importante resaltar que al ser un proyecto de graduación la Cooperativa no necesita realizar ninguna especie de gasto para la elaboración del prototipo. Por otro lado, dentro de las proyecciones propuestas se determinan las diferentes funcionalidades que el prototipo tiene, las diferentes herramientas o dispositivos tecnológicos utilizados para su desarrollo, pruebas y puesta en marcha de éste y la metodología de ciclo de vida que éste utiliza, que incluye las fases de análisis, diseño, programación y pruebas.

Dentro del capítulo II y III, se encuentra el marco referencial y el metodológico donde en el primero se buscó definir diferentes términos o conceptos relacionados con el prototipo funcional desarrollado y los procesos dentro de la gestión de proyectos que realiza la Cooperativa Sulá Batsú y en el metodológico se mencionaron las diferentes características metodológicas utilizadas para la elaboración del prototipo donde se estableció una población y una muestra a estudiar dentro de la Cooperativa mediante distintos instrumentos de recolección de datos.

Con respecto del capítulo IV, el análisis de resultados es en donde se realizó el estudio de los distintos datos recolectados y los procesos encontrados mediante las entrevistas y las observaciones con la población determinada en la Cooperativa, dando como resultado los diferentes módulos del prototipo.

Para finalizar con los capítulos V y VI, con las distintas recomendaciones y conclusiones del proyecto y la propuesta realizada, respectivamente. La propuesta es uno de

los apartados con mayor importancia dentro de proyecto donde se realizan las diferentes fases de la metodología de desarrollo utilizado, las cuales se componen de lo siguiente:

- **Análisis:** Fase donde se explican los diferentes módulos del prototipo, el hardware y equipo de telecomunicaciones utilizado, la base de datos, el personal con los conocimientos requeridos y los distintos casos de uso que explican el funcionamiento del prototipo.
- **Diseño:** Fase donde se muestran los diferentes diseños establecidos para el prototipo funcional entre ellos: la arquitectura del sistema, del software, de las interfaces más importantes, de la base de datos con su respectivo diccionario de datos, los procesos más importantes, las salidas de reportes y los diagramas UML extra necesarios.
- **Programación:** En ésta se muestran diferentes ejemplos de código en los lenguajes de programación utilizados para el desarrollo, mostrando lo siguiente: entradas, salidas, procesos más importantes, validaciones realizadas y distintos módulos establecidos.
- **Pruebas:** Fase donde se muestran los diferentes casos de prueba realizados para comprobar la funcionalidad del prototipo.

Como consecuencia se puede determinar que el prototipo funcional elaborado va a resultar de gran ayuda para la Cooperativa permitiendo automatizar diferentes procesos y cálculos, que posibilitará a éste a expandir su capacidad de gestionar distintos proyectos y a reducir los tiempos invertidos en la elaboración de distintos reportes. A su vez, a futuro se podría expandir las funcionalidades del prototipo para asumir más funcionalidades contables con el fin reasignar diferente personal que utilice el prototipo y reducir los costos de un contador externo.

CAPÍTULO I

Introducción

La Cooperativa Sulá Batsú es una empresa de economía social solidaria donde su principal actividad es la realización de diferentes proyectos para promover el desarrollo local, a través del trabajo con diferentes organizaciones, empresas sociales, redes comunitarias y movimientos sociales a nivel regional, nacional y global, junto con todos los asociados que la componen.

Planteamiento del Problema

Sulá Batsú trabaja con un grupo de asociados y asociadas, el cual se encarga de realizar la toma de decisiones sobre la cooperativa y los proyectos que se trabajan o trabajarán denominado Junta Directiva, sin embargo, esta presenta problemas con la toma de decisiones con respecto a los proyectos, ya que los reportes que se brindan respectivamente con la información de estos presenta atrasos, por lo que la Junta Directiva al no tener la información oportunamente no puede decidir de manera precisa y rápida.

Ahora bien, dentro de la cooperativa se encuentra otros asociados y asociadas que trabajan en los diferente proyectos o funciones que tiene la cooperativa, por lo que todos tienen diferentes tareas de las cuales hacerse cargo, pero con respecto a la tarea de administrar proyectos solamente se encuentra una asociada a cargo a dicha tarea, sobrecargando a la misma, ya que a su vez esta tiene otras tareas a realizar como sus compañeros, lo que provoca atrasos y pago de jornadas extra.

Además, todos los datos referentes a los proyectos se encuentran almacenados dentro de hojas de Excel, por lo que la manipulación de los mismos se realiza manualmente. Lo que nos lleva a que, a la hora de realizar diferentes reportes sobre el estado de los proyectos, se realice lentamente ya que la encargada debe recorrer cada una de las hojas de Excel manualmente para extraer la información necesaria, preparar el reporte y entregarlo a los interesados.

Por otro lado, la realización de los cálculos matemáticos necesarios para obtener los estados financieros de los proyectos, así como el pago de excedentes a los asociados se realiza

mediante fórmulas hechas en Excel, las cuales pueden ser modificadas en cualquier momento, esto podría incurrir en la presencia de un riesgo de manipulación de la información, provocando que se puedan presentar pérdidas financieras o pérdida de proyectos por malas decisiones.

Objetivos

General.

Realizar un prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú R.L ubicada en Barrio Escalante.

Específicos.

- Analizar todos los requerimientos necesarios para la realización del prototipo funcional de la Cooperativa Sulá Batsú, mediante entrevistas y observaciones.
- Realizar todos los diseños del prototipo funcional con el fin de completar el prototipo funcional utilizando diagramas entidad – relación, diagrama de flujo y diseño de interfaces.
- Programar la correcta funcionalidad del prototipo funcional mediante las herramientas de desarrollo establecidas.
- Comprobar la correcta funcionalidad del prototipo funcional para asegurar el cumplimiento de las soluciones propuestas mediante pruebas unitarias.

Justificación

La gestión de proyectos es una tarea extensa y de gran importancia para el funcionamiento de la Cooperativa Sulá Batsú, dicho esto, es esencial que esta tarea se desarrolle lo mejor posible, por lo que la automatización de ello traerá beneficios a la Cooperativa.

Esto agilizará los procesos y los cálculos necesarios permitiendo la creación de reportes con la información precisa necesaria, posibilitando la toma de decisiones por parte de la Junta Directiva de manera oportuna y precisa.

A su vez, al tener una sistematización de la gestión de proyectos se podrá determinar con mayor facilidad las diferentes tareas que se realizan, para así distribuir las entre los diferentes asociados que vayan a ser asignados posteriormente.

Viabilidad Operativa.

La Cooperativa, actualmente, cuenta con una plataforma web por lo que los funcionarios están familiarizados con los navegadores webs, así que la viabilidad del desarrollo de un aplicativo web es factible, sin embargo, se necesitará una capacitación del uso del prototipo en caso de una implementación para los asociados o asociadas que tendrán uso de este.

A su vez, las responsabilidades de los asociados o asociadas podrían ser reorganizadas, permitiendo la asignación de más personal a la tarea de la gestión de proyectos, sin embargo, ya que el prototipo agilizará el proceso efectuado actualmente, no sería algo cien por ciento necesario.

Por lo anterior, se concluye que existe la viabilidad operativa para el prototipo realizado.

Viabilidad Técnica.

A continuación, se muestra el apartado sobre la viabilidad técnica que posee el proyecto, mediante las características técnicas de los equipos con los que se disponen para el desarrollo del prototipo como del necesario para la utilización de este y los servicios adquiridos por parte de la Cooperativa.

Tabla 1. Características del equipo de desarrollo utilizado

Hardware	Software
Tarjeta Madre: Gigabyte H110M-H Procesador: Intel i3 6100 3.70 Memoria: 8GB DDR4 Almacenamiento: HDD 500GB Tarjeta de Video: AMD Radeon 560 Strix OC Puerto: Ethernet 100 Mbps Dispositivos: -Teclado -Mouse -Monitor	Windows 10 Pro 64bits Herramientas de desarrollo: -Brackets v1.13 -XAMPP v3.2.2 32bits -Materialize v1.0.0 -PHP7 -jQuery -Ajax -MySQL Workbench 8.0 CE

Nota: Elaboración Propia

Tabla 2. Características del equipo necesario para el uso del prototipo

Hardware	Software
-Computadores de gama baja. -Procesador: 2GHz -Memoria: 4GB. -Almacenamiento: 100 GB. -Adaptador de red: Ethernet o Wifi Dispositivos: -Teclado. -Mouse. -Monitor. -Router con conexión a Internet y una velocidad mínima de 1Mbps.	Sistema Operativo: Windows 7 Navegador: Chrome o Firefox.

Fuente: Elaboración Propia

Estos requerimientos de hardware, actualmente, son las características que poseen las computadoras de gama baja, generalmente. Un equipo que cumple con éstas, podrá soportar el aplicativo sin ningún problema, siempre y cuando solamente se utilicen el hardware y software indicado previamente. Esto debido a que estas características supera los requerimientos mínimos del sistema operativo indicado y, a su vez, permite tener capacidad de extra en memoria y almacenamiento para el navegador y otros aplicativos de oficina.

Todos los asociados de la Cooperativa tienen a su disposición dispositivos o computadoras compatibles que cumplen con los requerimientos mínimos indicados previamente. A su vez, ya poseen, actualmente, un hosting en la nube con servicios de base de datos, CPanel y soporte para php, también posee 1 año de servicio con soporte los 365 días durante 24 horas.

Por otro lado, la Cooperativa, a su vez, tiene un contrato empresarial con el proveedor de internet Tigo por un servicio de Internet de cable coaxial de una velocidad de 25mbps y 3mbps de descarga y carga respectivamente.

Por lo anterior, se concluye que existe la viabilidad técnica para el prototipo realizado

Viabilidad Económica.

Todas las herramientas que se utilizan para desarrollar el prototipo funcional son gratuitas por lo que no será necesario invertir dinero en licenciamientos.

A su vez, la empresa ya cuenta con un plan de hosting para sustentar su propia página web por lo que no será necesario la inversión en uno de estos, para alojar el prototipo funcional, a su vez, este plan posee todo el soporte necesario para el mantenimiento del equipo.

Por otro lado, para efectos informativos se presentará el costo que hubiera tenido el desarrollo de un prototipo como el propuesto. Según el decreto de Salarios Mínimos 2018 del Ministerio de Trabajo un Ingeniero en Software (http://www.mtss.go.cr/temas-laborales/salarios/Documentos-Salarios/lista_ocupacion_2018.pdf) no graduado recibe por hora ₡1,642.67, dando a entender en la siguiente tabla el pago por servicios profesionales sería

Tabla 3. Costos por servicios profesionales

Fase	Duración	Total
Análisis	110 horas	€180,693.7
Diseño	90 horas	€147,840.3
Programación	140 horas	€229,973.8
Pruebas	80 horas	€131,413.6
Total:		€689,921.4

Fuente: elaboración propia.

Sin embargo, para la elaboración del prototipo no se cobra ningún monto ya que será usado como proyecto final de graduación de la carrera del sustentante.

Por lo anterior, se concluye que existe la viabilidad económica para el prototipo realizado.

Viabilidad Legal.

Se debe asegurar el cumplimiento de las siguientes leyes:

- Ley de Protección de la Persona Frente al Tratamiento de sus Datos Personales (Ley N° 8968): la información que se obtendrá, manipulará y utilizará durante el desarrollo del proyecto será únicamente con dicho fin, en el caso de que el prototipo entre en operación la información será de uso exclusivo para los fines propios del mismo y no se compartirá la información con terceros.
- Ley de Procedimientos de Observancia de los Derechos de Propiedad Intelectual (Ley N° 8039): herramientas por utilizar serán gratuitas o se paga el respectivo licenciamiento.
- Ley de Derechos de Autor y Derechos Conexos (Ley N° 6683): protege los autores intelectuales y sus obras ya que se mencionarán en las respectivas citas bibliográficas y se harán referencia de ellos en las referencias; además la organización donde se desarrollará el proyecto, no distribuirá el prototipo sin consentimiento del desarrollador y solamente se utilizará para los fines para los cuales será desarrollado.

Dado lo anterior, se concluye que existe la viabilidad legal para el prototipo realizado.

Proyecciones

Dentro de este apartado se hace mención de los aspectos que se abarcan dentro del proyecto y, a su vez, las distintas funcionalidades que este mismo posee

Proyección funcional.

- Módulo de Seguridad:

Módulo encargado de verificar las credenciales de los usuarios para mantener una seguridad óptima, donde se realizará lo siguiente:

- ❖ Comprobar la coincidencia de las credenciales ingresadas.
- ❖ Verificar los roles de los usuarios que han iniciado sesión.

- Módulo de Mantenimientos:

Módulo encargado de realizar todas las inserciones de datos dentro de cada una de las entidades en la base de datos, estos registros podrán ser modificados, consultados o dados de baja de ser necesario. A continuación, se mencionará cómo trabajará cada entidad:

- ❖ Asociados: Todos los datos relacionados con la información de cada asociado o asociada de la Cooperativa, los cuales cuentan con un porcentaje definido de excedente que les corresponde dependiendo de su tiempo en la Cooperativa y de ser necesario se podrán dar de baja dentro del sistema.
- ❖ Proyectos: Datos de cada uno de los proyectos para la realización de cálculos o reportes, de ser necesario el proyecto podrá darse de baja en el sistema.
- ❖ Asociado Proyecto: Registro referente a los asociados y asociadas designados a cada uno de los proyectos.
- ❖ Colaboradores: Datos de cada uno de los colaboradores y colaboradoras contratadas para la realización de los distintos proyectos dentro de la Cooperativa.

- ❖ Colaboradores Proyecto: Registro referente a los colaboradores y colaboradoras designada a cada uno de los proyectos, junto a su jornada y monto respectivos.
- ❖ Excedentes: Registro del cálculo de excedentes anuales y mensuales de cada uno de los asociados, asociadas, colaboradores y colaboradoras dentro de la Cooperativa, de acuerdo con los proyectos en que fueron asignados.
- ❖ Rubros: Datos almacenados sobre los diferentes rubros que cada uno de los proyectos posee para determinar su presupuesto.
- ❖ Categoría Rubros: Catálogo de categorías para cada uno de los rubros desinadas a cada uno de los proyectos, junto con el total de estos.
- ❖ Gastos Operativos: Datos sobre los diferentes gastos operativos que cada uno de los proyectos posee, registrando sus valores en los distintos meses y un total del periodo.
- ❖ Historial Transferencias: Historial de transferencias bancarias hacia cada una de las cuentas que poseen cada uno de los asociados, asociadas, colaboradores y colaboradoras para el pago de excedentes con respecto de los proyectos en los que trabajan.
- ❖ Clientes: Entidad en donde se almacena toda aquella información necesaria sobre todos los clientes que posee la Cooperativa.
- ❖ Proveedores: Entidad en donde se almacena toda aquella información necesaria sobre todos los proveedores que posee la cooperativa.
- ❖ Cuentas por Pagar: Almacenará todos los datos de las diferentes cuentas por pagar relacionadas con cada uno de los proyectos, de ser necesario podrán darse de baja.
- ❖ Cuentas por Cobrar: Almacenará todos los datos de las diferentes cuentas por cobrar relacionadas con cada uno de los proyectos, de ser necesario podrán darse de baja.
- ❖ Flujo de Gastos: Registro de los diferentes gastos de ejecución que incurre el proyecto dentro de un periodo de un año.
- ❖ Flujo de Ingresos: Registro de los diferentes ingresos que recibe el proyecto dentro del periodo de un año.

- Módulo de Reportes:

Módulo encargado de armar cada uno de los reportes necesarios para la verificación de cada uno de los proyectos de la Cooperativa, los reportes al realizar serán:

- ❖ Reporte general de las cuentas por pagar y/o cobrar permitiendo observar aquellas que están próximas a vencer o en caso de ser necesario las vencidas o aquellas que se encuentren dadas de baja.
- ❖ Reporte del estado del proyecto dando a conocer sus flujos de ingreso y gastos de ejecución durante cada periodo y los saldos resultantes.
- ❖ Reporte de los gastos operativos de la cooperativa mostrando la actividad y los costos de los mismos efectuados durante el periodo.
- ❖ Reporte de distribución de excedentes con respecto al fondo del proyecto y la jornada de cada uno de los colaboradores.

- Módulo de Consultas:

- ❖ Consulta general de cuentas por pagar y cuentas por cobrar activas como inactivas, dependiendo de la necesidad, filtrado por número de cuenta, fecha o monto.
- ❖ Consulta del historial de transferencias realizadas a los asociados para el pago de excedentes o las del propio proyecto.
- ❖ Consulta general de la información de cada uno de los asociados de la Cooperativa.
- ❖ Consulta general de la información de cada uno de los proyectos.
- ❖ Consulta general los costos operativos de cada uno de los proyectos.
- ❖ Consulta general de los flujos de cada proyecto.

- Módulo de Cálculos Financieros:

Módulo encargado de realizar todos los cálculos necesarios para obtener el estado financiero de cada proyecto, teniendo en cuenta las cuentas por pagar y cobrar, activos y los contratos disponibles.

Dichos cálculos se realizarán de forma automatizada permitiendo agilizar el proceso gracias a los recursos computacionales y al acceso rápido a la información necesaria.

Esto al final podrá conseguir la correcta estimación de las pérdidas y/o saldos restantes que se generan de los diferentes proyectos en los que trabaja la Cooperativa Sulá Batsú.

- **Módulo de Cuentas por Cobrar:**

Módulo encargado de mostrar las diferentes cuentas por cobrar dentro de diferentes periodos, éstas serán registradas por medio de una factura o algún gasto efectuado en el momento, donde se elaborarán alertas mensuales para informar las cuentas que están a punto de vencer o el atraso a para ser canceladas.

- **Módulo de Cuentas por Pagar:**

Módulo donde se desplegarán las cuentas por pagar de cada proyecto que posee la Cooperativa con respecto de monto brindado, permitiendo que éstas puedan ser canceladas o abonadas en el sistema y, a su vez se realizarán alertas mensuales para informar el vencimiento o atraso de éstas.

- **Módulo de Excedentes a Asociados:**

Módulo encargado de realizar el cálculo de excedentes de cada proyecto para su distribución correspondiente a cada uno de los asociados que tuvieron participación en éste, cubriendo sus respectivas jornadas asignadas, a su vez se realizará el cálculo del excedente anual de está como pago de aguinaldo cada año.

Proyección tecnológica.

- **Sistema Operativo:** El prototipo funcional es desarrollado en un sistema operativo Windows 10 Pro de 64bits y en el navegador Google Chrome v69.0.3497.100 de 64bits.
- **Herramientas de desarrollo:** Se utiliza la herramienta gratuita Brackets v1.13, el framework de Materilize v1.0.0 y se emplean los lenguajes de HTML5, CSS3, PHP, JQuery, JavaScript y Ajax
- **Motor de Base de Datos:** Para la base de datos se utiliza el motor de base de datos InnoDB y el gestor de base de datos MySQL Worbench 8.0 CE, los cuales son gratuitos.

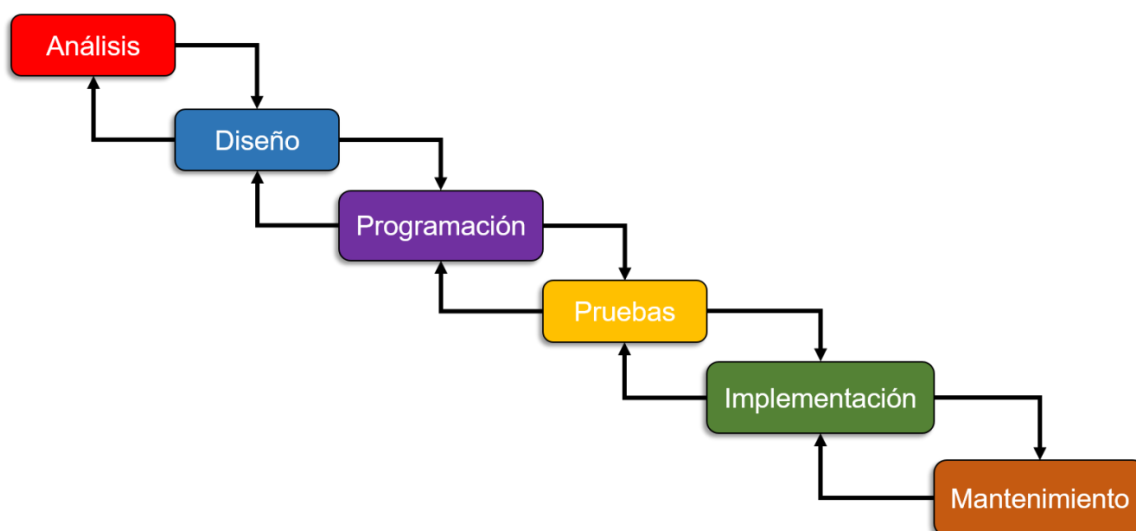
- Hosting: Servicio de hosting de aplicaciones webs por parte de <https://www.siteground.com> con 1 año de servicio, dando soporte los 365 días durante 24 horas.
- Servicio de Internet: Contrato empresarial con Tigo por un servicio de Internet de cable coaxial por una velocidad de 25mbps de bajada y 3mbps de subida.

Proyección metodológica.

Como dice Molina, Zea, Contento y García (2018) “La construcción de software es una actividad que debe ser planificada y sistematizada mediante estrategias que garanticen el éxito del mismo” (p.4), es debido a esto que para el desarrollo de este prototipo funcional se utilizarán las características y etapas que posee el desarrollo en cascada.

El desarrollo en cascada es una de las metodologías más básicas del desarrollo de software ya que cada una de sus fases va ligada obligatoriamente con su sucesiva, ya que si la fase anterior no está presente esto podría comprometer el buen funcionamiento del sistema. Esto, a su vez, es un problema ya que, al presentarse algún retraso en las primeras fases durante la ejecución de las posteriores, esto obligará al proyecto a atrasarse y devolverse fase por fase hasta corregir el error, volviendo a recorrerlos.

Figura 1. Etapas del modelo en cascada



Fuente: Elaboración Propia

Como muestra la Figura 1 Etapas del Modelo en Cascada, se pueden observar las diferentes fases en las que consiste el modelo, sin embargo, para efectos de este proyecto, el cual será un prototipo funcional, no se utilizará la totalidad de las fases, éstas solamente abarcarán las fases de Análisis, Diseño, Codificación o Programación y Pruebas, dejando de lado las demás fases.

CAPÍTULO II

Marco Referencial

En este capítulo se define y aclaran distintos conceptos mencionados o vistos durante la investigación planteada, lo cuales podrían estar no relacionados con el lector, por lo cual se busca facilitarle la comprensión de estos.

Software

A continuación, se definen todos aquellos conceptos relacionados en el software que se mencionarán con el proyecto de investigación, con el fin de esclarecerlos para el lector y así pueda tener un mayor entendimiento.

Sistema operativo.

Según Norton et al (2000):

Un sistema operativo (OS, del inglés operating system) le dice a la computadora la forma en que debe utilizar sus propios componentes. Entre los ejemplos de sistemas operativos se incluyen Windows, el sistema operativo de Macintosh y Linux (véase la figura 1B.15). Un sistema operativo es esencial para cualquier computadora, debido a que actúa como un intérprete entre el hardware, los programas de aplicación y el usuario.

Cuando un programa requiere al hardware para hacer algo, se comunica con él a través del sistema operativo. De manera parecida, cuando usted desea que el hardware haga algo (por ejemplo, copiar o imprimir un archivo), su solicitud es atendida por el sistema operativo (p. 33).

Los sistemas operativos tienen diferentes versiones, que utilizan dependiendo de cómo el sistema almacenará los archivos que se utilicen, estas versiones son las de 32 bits y 64 bits, también conocidas como x86 y x64, respectivamente. Actualmente, muchas aplicaciones utilizan ambos sistemas de ficheros y distintos componentes también, sin embargo, una de las diferencias principales de un sistema de 32 a otro de 64, es que el de 32 no es capaz de soportar más allá de 4GB de memoria RAM, limitando así la capacidad y desempeño del equipo.

Figura 2. Sistemas Operativos



Fuente: Yoelvis, O

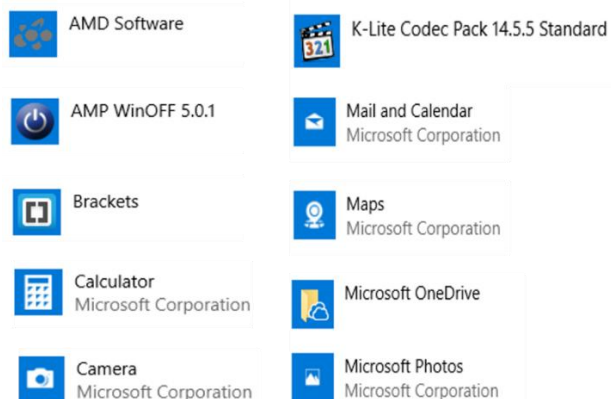
Aplicación.

Como dice Norton et al (2000):

El software de aplicaciones le dice a la computadora la forma como debe llevar a cabo tareas específicas para el usuario, por ejemplo, procesar textos o dibujar. Existen miles de aplicaciones disponibles para muchos propósitos y personas de todas las edades (p. 33).

Algunas de estos se encargan de realizar una tarea específica o un conjunto de tareas, por lo que el catálogo de selección es muy amplio, estas aplicaciones también obedecen las versiones del sistema de ficheros que utiliza el sistema operativo por lo que es importante tener en cuenta cuál, es la versión que se instalará en el equipo.

Figura 3. Aplicaciones



Fuente: Elaboración Propia.

POO.

Según Canchala (2008):

La POO es un paradigma de la programación de computadores; esto hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, proporcionando un medio bien definido para visualizar el dominio del problema e implementar en un lenguaje de programación la solución a ese problema.

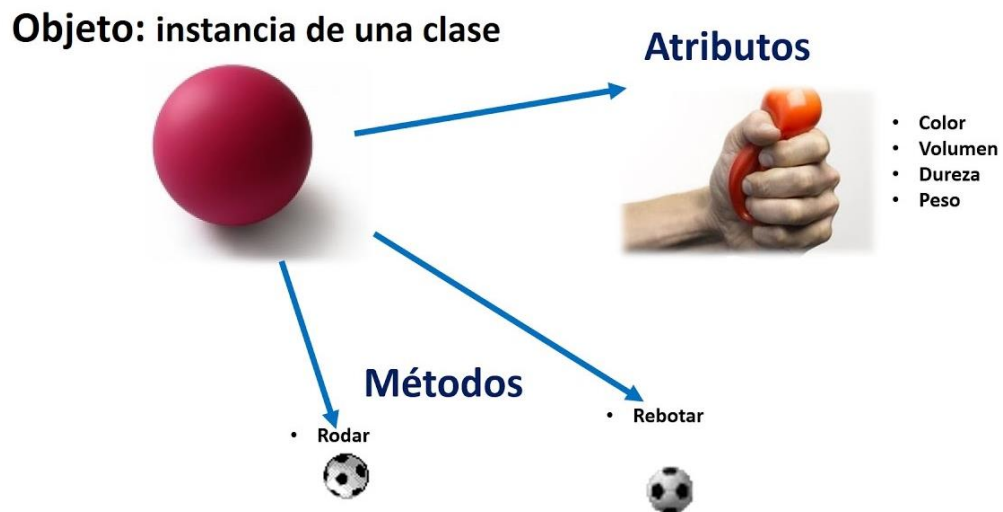
La POO se basa en el modelo objeto donde el elemento principal es el objeto, el cual es una unidad que contiene todas sus características y comportamientos en sí misma, lo cual lo hace como un todo independiente pero que se interrelaciona con objetos de su misma clase o de otras clase [sic], como sucede en el mundo real.

Sus componentes se dividen en:

- **Objeto:** Componente o código de *software* que contiene en sí mismo, tanto sus características (campos) como sus comportamientos (métodos); se accede a través de su interfaz o *signatura*.
- **Campo:** Es una característica de un objeto, que ayuda a definir su estructura y permite diferenciarlo de otros objetos. Se define con un identificador y un tipo, el cual indica los valores que puede almacenar.
- **Método:** Es la implementación de un algoritmo que representa una operación o función que un objeto realiza (párr.2-6).

Actualmente, la gran cantidad de sistemas desarrollados utilizan este paradigma a la hora cuando son desarrollados, debido a la facilidad que permite al abstraer los objetos percibidos en la realidad a la programación, dando un entendimiento mayor sobre cuáles son los atributos que lo componen, como su comportamiento. A la hora de realizar una programación orientada a objetos hay que tener en cuenta que dichos objetos deben ser los más específicos posibles, ya que cualquier ambigüedad que se presente ocasionará confusión al desarrollador y podría determinar el mal funcionamiento del sistema.

Figura 4. Programación orientada a objetos



Fuente: Narvaéz A.

Módulo.

Según Canchala (2008):

Son partes de un todo que se integran perfectamente entre sí para que funcionen por un objetivo general, y a las cuales se les pueden agregar más componentes que se acoplen perfectamente al todo, o extraerle componentes sin afectar su funcionamiento. En el caso que se requiera actualizar un módulo, no hay necesidad de hacer cambios en otras partes del todo. Un ejemplo clásico es un conjunto de módulos que, al integrarlos conforman un armario, el cual puede agregarle más funcionalidad si se le agregan más módulos, o al contrario. También se puede cambiar su finalidad si se acomodan esos módulos para darle otro objetivo: volverlo una mesa (párr. 9).

Los módulos permiten subdividir grandes sistemas en partes más pequeñas, ya que cada uno de ellos se encargaría de realizar una tarea específica, esto posibilita que haya un mejor manejo de errores dentro del código fuente del sistema, ya que al presentarse alguno se podría determinar, específicamente, en qué módulo ocurrió el error y modificarlo.

Framework.

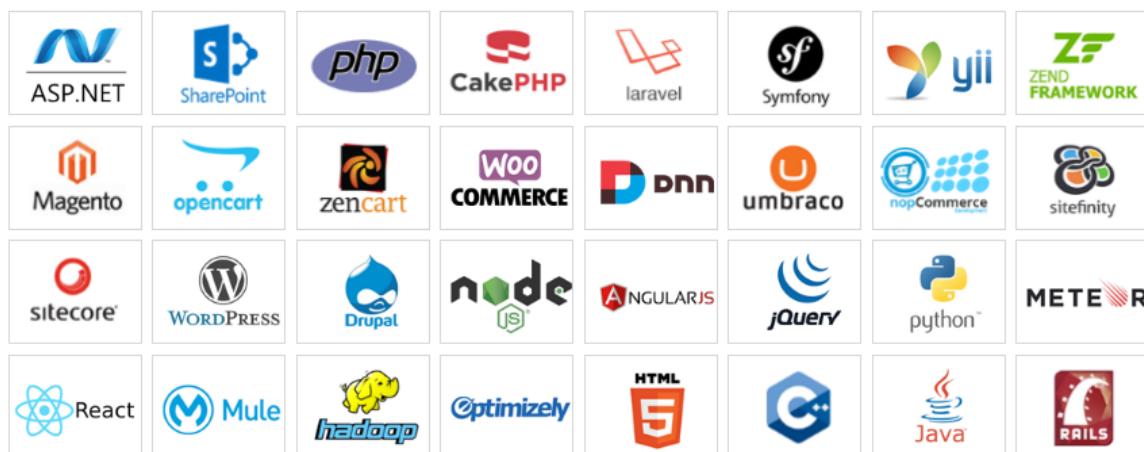
Según Gutiérrez (2014):

... un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones (p. 1).

Los framework juegan un papel muy importante en el desarrollo de diferentes sistemas, ya que estos establecen los cimientos en que los sistemas se construirán, permitiendo establecer una estructura única y comprensible para todos aquellos desarrolladores que trabajen en el sistema, esto es debido a que, en muchos casos en la vida de los desarrolladores, habrá distintos grupos de trabajo que estén desarrollando el sistema, por lo que es necesario que éstas, puedan comunicarse de la manera más clara.

Figura 5. Frameworks



Fuente: Patel, M

Materialize

Siguiendo la definición de Ramírez (2017) Materialize es:

... un framework que hace más fácil el proceso de creación de diseño web. Es un framework moderno y responsivo que se basa en Material Design. Tiene muchos componentes y clases predefinidas, para hacer uso de ellas mucho fácilmente. **Aumenta [sic] el ritmo de desarrollo**

Como framework Materialize es uno muy completo que tiene una gran cantidad de estilos, íconos y animaciones para la creación de un sitio web que sea compatible con navegadores, tanto en dispositivos móviles como en equipos de escritorio, este framework, a su vez, cuenta con diferentes temas ya hecho para poder tomarlos como base a la hora de crear un sitio web, estos temas poseen una explicación de cada uno de los elementos utilizados en el mismo para que los desarrolladores puedan utilizarlo con libertad.

Figura 6. Logo materialize.



Fuente: Materialize.

HTML.

Según MDN web docs et al (2013):

HTML, que significa **Lenguaje de Marcado para Hipertextos (HyperText Markup Language)** es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript).

HTML le da "valor añadido" a un texto estándar en español. Hiper Texto se refiere a enlaces que conectan una página Web con otra, ya sea dentro de una página web o entre diferentes sitios web. los vínculos son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a páginas de otras personas, te haces participante activo de esta Red Mundial.

HTML usa "markup" o marcado para anotar textos, imágenes, y otros contenidos que se muestran en el Navegador Web. El lenguaje de marcado HTML incluye "elementos" especiales tales como `<head>`, `<title>`, `<body>`, `<header>`, `<article>`, `<section>`, `<p>`, `<div>`, ``, ``, y muchos otros más (párr. 1-3).

Es por ello que HTML es esencial siempre que se hable sobre la estructura o desarrollo de una página web. Actualmente, este lenguaje se encuentra en su versión cinco denominado HTML5, la cual tiene nuevas etiquetas y permite integrar nuevas funcionalidades a las páginas webs, por ejemplo, animaciones simples a imágenes y texto sin necesidad de utilizar javascript.

PHP.

Según González (s.f)

PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un

mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc. (p. 1).

El uso común que se le da al lenguaje php es la interacción con los datos alojados dentro de la base de dato de un sitio web, donde se realizan distintas consultas directamente hacia la base de datos o por medio de procedimientos almacenados (consultas pre establecidas dentro de la base de datos). Este lenguaje no es visible desde los navegadores por lo cual es sumamente recomendado dentro de la programación web, ya que todo aquello que tenga que ver con la manipulación de datos es sumamente riesgoso, y necesita seguridad de algún tipo, para asegurar la integridad, manipulación y divulgación de los datos.

JavaScript.

Como menciona Pérez (2007):

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript se puede crear diferentes efectos e interactuar con nuestros usuarios.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros (p.5-6).

Este lenguaje permite realizar diferentes funciones dentro del desarrollo de sitios webs, como animaciones o respuestas automatizadas a acciones, por lo que su funcionalidad es sumamente importante con los desarrollos que se realizan hoy.

Actualmente, existen muchas librerías hechas en javascript en Internet, que pueden ser utilizadas gratuitamente por cualquier desarrollador permitiendo facilitar el desarrollo.

JQuery.

Siguiendo el concepto dado por jQuery Foundation (s.f):

jQuery es una librería de JavaScript rápida, pequeña y con muchas funciones. Hace más fácil el desplazamiento y manipulación de documentos HTML, manejo de eventos, animaciones y Ajax con una API fácil de usar, capaz de funcionar en distintos navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado el modo en que millones de personas escriben JavaScript (párr. 1) - Traducido por sustentante.

Esta librería es una de las más comunes y funcionales que existen actualmente, ya que posee una extensa documentación de todos las funciones y métodos que posee, permitiendo consultarlas y entenderlas mediante distintos ejemplos. Muchas de las animaciones utilizadas para los sitios webs se debe a que utilizan dicha librería, por ejemplo, el framework de Materialize mencionado anteriormente hace gran uso de esta librería.

Ajax.

Según IBM Knowledge Center (s.f):

Ajax (Asynchronous JavaScript and XML) se refiere a un grupo de tecnologías que se utilizan para desarrollar aplicaciones web. Al combinar estas tecnologías, las páginas web parece que son más receptivas puesto que los paquetes pequeños de datos se intercambian con el servidor y las páginas web no se vuelven a cargar cada vez que un usuario realiza un cambio de entrada. Ajax permite que un usuario de la aplicación web interactúe con una página web sin la interrupción que implica volver a cargar la página web. La interacción del sitio web ocurre rápidamente sólo con partes de la página de recarga y renovación.

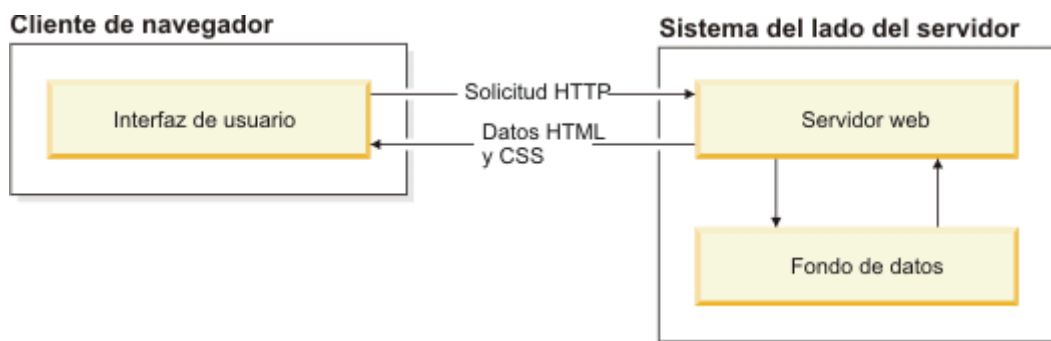
Ajax se compone de las siguientes tecnologías:

- XHTML y CSS para presentar información.

- DOM (Document Object Model - modelo de objetos de documento) para visualizar e interactuar de forma dinámica la información presentada.
- El objeto XMLHttpRequest para manipular los datos de forma asíncrona con el servidor web.
- XML, HTML y XSLT para el intercambio y la manipulación de datos.
- Se visualiza JavaScript para enlazar solicitudes e información de datos.

En una aplicación web tradicional, las solicitudes HTTP, que se inician mediante la interacción del usuario con la interfaz web, se realizan a un servidor web. El servidor web procesa la solicitud y devuelve una página HTML al cliente. Durante el transporte HTTP, el usuario no puede interactuar con la aplicación web.

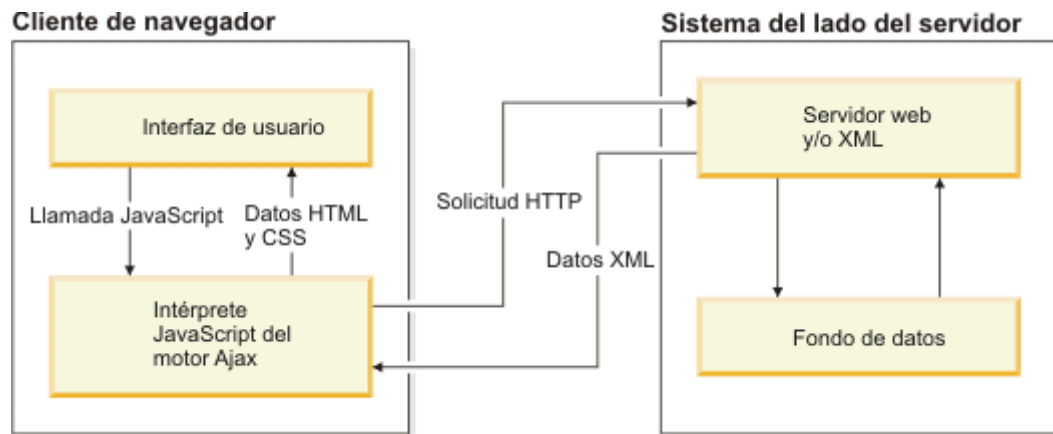
Figura 7. Proceso de comunicación de una página web básica



Fuente: IBM Knowledge Center.

En una aplicación web Ajax, no se interrumpe el usuario en interacciones con la aplicación web. El motor de Ajax o el intérprete JavaScript permite que el usuario interactúe con la aplicación web independientemente del transporte HTTP procedente del servidor o que tenga el servidor como destino representando la interfaz y gestionando las comunicaciones con el servidor en nombre del usuario (párr. 1-4)

Figura 8. Proceso de comunicación con Ajax.



Fuente: IBM Knowledge Center.

Utilizar Ajax dentro del desarrollo web permite que la percepción del usuario final al sistema o a la página web sea mejor, ya que puede trabajar de manera asíncrona, permitiendo que el usuario siga realizando diferentes tareas mientras se hacen peticiones al servidor, esto hace que sea más interactivo y amigable con el usuario.

Base de Datos

En este apartado, se explican algunos términos generales referente a las bases de datos que se utilizan para esta investigación.

Base de Datos.

Según Carmona (2011):

Realmente una base de datos es un conjunto de datos pertenecientes a un mismo contexto que se encuentran almacenados de forma sistemática para un uso posterior. Así puede entenderse que es una biblioteca en una base de datos compuesta, en su mayoría, por documentos y textos impresos en papel que se encuentran indexados para su consulta (p. 9).

Las bases de datos constituyen una parte fundamental en la informática actualmente, ya que todos los sistemas necesitan algún repositorio de datos del cual puedan extraer y

almacenar datos para procesarla y convertirla en información útil, en la actualidad, existen diferentes bases de datos que pueden ser utilizadas de manera gratuita o de pago que pueden ser adaptados a las arquitecturas de software que se requieran.

MySQL.

Según ICTEA (s.f):

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

Esta base de datos como se menciona anteriormente es muy utilizada dentro de los ambientes de desarrollo web, que como las demás bases de datos relacionales que existen, dentro del mercado utiliza el lenguaje SQL para la manipulación de los datos, sin embargo, éste posee tipos de datos que pueden diferir de las más por lo que es importante informarse sobre los tipos de datos que maneja y reconoce, así como sí ésta reconoce la diferencia entre minúsculas y mayúsculas.

Tablas.

Según Microsoft (2017)

Las tablas son objetos de base de datos que contienen todos sus datos. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas, similar al de una hoja de cálculo. Cada fila representa un registro único y cada columna un campo dentro del registro. Por ejemplo, en una tabla que contiene los datos de los empleados de una compañía puede haber una fila para cada empleado y distintas columnas en las que figuren detalles de los mismos, como el número de empleado, el nombre, la dirección, el puesto que ocupa y su número de teléfono particular (párr. 1).

Estas constituyen la parte fundamental de todas las bases de datos ya que en ellas es donde se almacena todos los datos que el sistema o aplicación solicite al usuario y la información que éste procese, muchas bases de datos que se utilizan, actualmente, son relacionales la cuales se fundamentan en las relaciones que se crean entre las diferentes tablas que componen la base de datos.

SQL.

Según Microsoft (s.f):

SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos. Los programas de bases de datos relacionales, como Microsoft Office Access, usan SQL para trabajar con datos. A diferencia de muchos lenguajes de computación, SQL no es difícil de leer y entender, incluso para un usuario inexperto. Al igual que muchos lenguajes de computación, SQL es un estándar internacional reconocido por organismos de estándares como ISO y ANSI.

SQL se usa para describir conjuntos de datos que pueden ayudarle a responder preguntas. Al usar SQL, debe usar la sintaxis correcta. La sintaxis es el conjunto de reglas mediante las que se combinan correctamente los elementos de un idioma. La sintaxis SQL se basa en la sintaxis del idioma inglés y usa muchos de los mismos elementos que la sintaxis de Visual Basic para Aplicaciones (VBA) (párr. 5-6).

Como SQL es un estándar todos los motores de bases de datos relacionales son capaces de interactuar con éstos permitiendo que cualquier persona con conocimientos en

dicho leguaje, pueda ejecutar diferentes instrucciones para manipular o consultar los datos necesarios dentro de la base de datos, siempre y cuando ésta tenga acceso a ellos.

Servicio de alojamiento web

A continuación, se explican diferentes términos y conceptos relacionados con el servicio de alojamiento web que toman en cuenta dentro de esta investigación.

Hosting (alojamiento web).

Según Duplika (2013):

El hosting es la prestación en donde una empresa ofrece a su cliente el espacio físico (dentro de un ordenador conocido como “servidor”) para que almacene los datos de su website de modo que esté siempre online.

Estos servicios de hosting funcionan ofreciéndote **computadoras de grandes prestaciones** (servidores web), utilizando conexiones de alta velocidad además de muchas otras funcionalidades concernientes a seguridad y sistemas (párr. 2-3).

Siempre que se consideren desarrollar aplicaciones webs o páginas webs, es importante tener en cuenta el servicio de hosting por contratar, ya que estos son los que brindarán el posicionamiento de éste dentro de la gran red llamada Internet, a su vez, la nube habilitando que éstas puedan ser encontradas por medio de buscadores o ingresadas directamente por medio de una dirección web (url).

Contrato del nivel de servicio (SLA – Service Level Agreement).

Según Iglesias(2017)

... los SLA no son más que un acuerdo por escrito por el que el proveedor de servicios TIC se compromete con el cliente a un determinado nivel de calidad (entendido en su más amplia concepción, desde la potencia de procesamiento hasta la disponibilidad del mismo, el tiempo de respuesta ante un incidente, el personal a cargo, la gestión del mantenimiento o las medidas de ciberseguridad) y se establecen los mecanismos de control y resolución de

disputas (incluyendo las compensaciones por incumplimiento) en caso de producirse un incidente (párr. 3).

Para el proveedor los SLA no son sólo obligaciones. Estos acuerdos suelen especificar detalles clave como la duración de la prestación del servicio, plazos y modalidades de pago o la posibilidad ya negociada de incorporar nuevos servicios al contrato. Eso sí, en cualquiera de los casos, el proveedor siempre debe tener en cuenta una máxima a la hora de redactar el documento: ser capaz de cumplir con creces con todo lo prometido, si queremos evitar multas (párr. 7).

Los niveles de servicio son fundamentales siempre que se trate incursionar a servicios en Internet, ya que estos permiten respaldar tanto a la empresa contratada y a la entidad que contrata, la cual sería el cliente. Debido a que cada uno de estos acuerdos estipula diferentes puntos de tolerancia para ambos en caso de cualquier falla, uno de estos puede ser las previamente mencionadas multas en caso de que se presente algún inconveniente, posibilitando que la confianza de los distintos clientes no se pierda en su totalidad y así la empresa pueda aún solventar los problemas y mantener a éstos.

Servidor DNS.

Según Molina y Polo (2014) los servidores DNS:

... almacenan tablas de correspondencias entre direcciones de nombres de dominio y direcciones IP. Cuando una estación desea establecer una conexión con una dirección DNS, llama a la rutina del sistema o resolvidor que primero comprueba si puede obtener la dirección IP a través de una tabla local o almacenada de forma temporal de una consulta anterior. Si no la encuentra en esa tabla, envía un mensaje con el protocolo UDP a la dirección del servidor DNS que tenga configurado por defecto (normalmente el más próximo). Este servidor consulta primero en sus registros de recursos de zona la dirección solicitada (en caso de ser un servidor primario o secundario) y devuelve la dirección IP si la encuentra. Si no la encuentra ahí, consultará entonces la tabla local donde están almacenadas temporalmente consultas anteriores. En caso de que tampoco la encuentre ahí, puede consultar otros servidores DNS, operación que se denomina consulta recursiva (véase la figura 3.3). Finalmente, el resolvidor devuelve la dirección IP solicitada a la estación (p. 171).

Es debido a ellos que las personas son capaces de aprender direcciones webs, ya que se presenta una mayor facilidad aprender direcciones por palabras, por ejemplo “google.com”, que aprender una dirección IP directamente. En Internet existen y se crean constantemente una gran infinidad de páginas webs que poseen, a su vez, sus respectivas direcciones IP, por lo cual es sumamente complicado saber cuáles son direcciones de cada una de ellas, sin embargo, gracias a los servicios de hosting que ofrecen diferentes empresas, facilitan que dichas configuraciones se hagan rápidamente.

Servidor web.

Los servidores webs son aquellos que permiten a los usuarios finales ingresar a diferentes páginas web, ya que el proceso es como lo describen Molina y Polo (2014):

Cuando un cliente escribe una dirección de una página WWW o hace click sobre un hipervínculo, el servidor correspondiente recibe una petición a través del protocolo TCP por el puerto 80 solicitando la página concreta. El protocolo que genera (en el lado del cliente) o recibe las respuestas (en el lado del servidor) se llama HTTP (HyperText Transfer Protocol o Protocolo de Transferencia de Hipertexto) y normalmente se usa para que un equipo servidor envíe páginas a un cliente, aunque la comunicación en sentido contrario también es posible (p. 38).

Los navegadores reciben dichas páginas webs y se las muestran al cliente, es por ello que a la hora de desarrollar una página web o una aplicación es importante tener en cuenta que los elementos que se están integrando a éste sean compatibles con los navegadores más usados por los clientes.

Servidor de aplicaciones.

Según IBM Knowledge Center (s.f):

Los servidores de aplicaciones son programas de servidor en una red distribuida que proporciona el entorno de ejecución para un programa de aplicación. Más específicamente, el servidor de aplicaciones es el componente de tiempo de ejecución principal en todas las configuraciones y donde una aplicación se ejecuta. El servidor de

aplicaciones colabora con el servidor web para ofrecer una respuesta dinámica y personalizada a una solicitud de cliente.

Estos servidores son de gran utilidad actualmente, ya que la tendencia actual con las tecnologías es ubicar todas sus aplicaciones en la nube (red de Internet), debido a que esto permite disminuir la utilización de recursos propios para la realización de las tareas diarias de las empresas por adquirir un servicio por la utilización de esto, sin preocuparse en dar mantenimiento a los equipos que las alojan.

Términos del negocio

En este apartado se definen los conceptos de términos del negocio utilizados en esta investigación, con el fin de esclarecer cualquier duda por parte del lector.

Proyecto.

Siguiendo el concepto de Ollé y Cerezuela (2017) se puede determinar que “Con el término proyecto nos referimos a aquella actividad que se realiza para dar respuesta con un producto o servicio único y concreto a una idea, un problema o una oportunidad identificada” (p. 17).

Sin embargo, también estaría bien agregar que PMBOK (2013) citado por Ollé y Cerezuela (2017) determina cómo un proyecto el “esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único” (p. 18)

Es por ello que se puede considerar que un proyecto es aquel esfuerzo cuantificable y medible que se lleva a cabo en el tiempo para obtener una meta específica, ya que los proyectos pueden abarcar múltiples disciplinas es necesario contar con diferentes recursos para llevarlos a cabo.

Activos.

El activo es una gran masa patrimonial conformado por elementos de signo positivo (bienes y derechos) los cuales están conformados por:

- Activo corriente (Circulante): Son aquellos elementos que se renuevan a lo largo del ejercicio económico.
 - ❖ Disponible: Dinero efectivo y cuentas bancarias.
 - ❖ Realizable: Bienes que pueden convertirse en dinero, salvo las existencias.
 - ❖ Existencias: Bienes objeto de la actividad empresarial, como sus productos, subproductos y materiales, incluso los de oficina, para la obtención de estos.
- Activo no corriente (Fijo o inmovilizado): Elementos patrimoniales que permanecen en la empresa a lo largo de varios ejercicios, hasta que se produce un desgaste, deterioro, inutilidad u obsolescencia.
 - ❖ Tangible: Aquellos que pueden ser percibidos físicamente
 - ❖ Intangible: Aquellos con naturaleza inmaterial, como licencias, patentes, concesiones administrativas, aplicaciones informáticas, derechos de traspaso y en general todos aquellos “derechos controlados por la empresa, resultantes de sucesos pasados, de los que se espera que la empresa obtenga beneficios o rendimientos económicos futuros”. Sin incluir inversiones financieras, debido a que son tangibles (Albarrán, 2014, p. 34-36).

Todas las empresas o personas jurídicas poseen activos que pueden ser clasificados de la forma antes mencionada, importante saber que los activos deben ser iguales a los pasivos ($A = P$) debido a que esto indica que hay un balance y el dinero fue utilizado, sin desaparecer sin ningún rastro.

Pasivos.

El pasivo es una gran masa patrimonial conformado por elementos de signo negativo (obligaciones) los cuales se dividen en:

- Exigible: Son aquellos recursos ajenos a la empresa y son por tanto obligaciones con el exterior como los préstamos o las deudas.
 - ❖ Corriente: Deudas a corto plazo (menor a 1 año).
 - ❖ No corriente: Deudas a largo plazo (mayor a 1 año).

- No exigible o Neto: Son aquellos conformados por los recursos propios de la empresa, o deudas y obligaciones internas (capital apartado por los propietarios y beneficios no distribuidos)

Nota: Este coincide con el patrimonio neto el cual es la diferencia: Bienes + Derecho – Obligaciones externas

- ❖ Capital: Aportaciones de los socios.
- ❖ Reservas: Beneficios pendientes a distribuir o no distribuidos (Albarrán, 2014, p. 34-36).

Como obligaciones que son, todos los pasivos deben ser resueltos, de tal manera que los activos pueden subsanar las necesidades los pasivos, esto con el fin de mantener a la empresa a flote y en caso de una persona jurídica, estable.

Es importante tener en cuenta que en las transacciones van a existir diferentes incrementos y decrementos, tanto en los activos como en los pasivos, por lo que generalmente se utiliza una tabla que los indica, la cual está conformada de la siguiente forma:

Figura 9. Tabla del debe y haber

Debe	Haber
+ Activo	+ Pasivo
– Pasivo	– Activo
Gastos	Ingresos
Compras	Ventas
Pérdidas	Ganancias

Fuente: Albarrán J.

Costos operativos.

Según Medianero y Carmen (2016):

Los costos de operación son los gastos que afrontará la institución durante el periodo posterior a la inversión inicial; es decir, después de culminada la ejecución del proyecto... Contablemente, los costos operativos comprenden a los gastos de operación y los gastos de mantenimiento.

Los costos de operación o costos de funcionamiento del proyecto son aquellos que ocurren luego del inicio, construcción o instalación de la nueva capacidad productiva hasta la finalización de su vida útil. Se obtienen a partir de la valorización monetaria de los bienes y servicios que deben adquirirse para mantener la operatividad y los beneficios generados o inducidos por el proyecto.

Los costos de operación o costos de funcionamiento del proyecto son aquellos que ocurren luego del inicio, construcción o instalación de la nueva capacidad productiva hasta la finalización de su vida útil. Se obtienen a partir de la valorización monetaria de los bienes y servicios que deben adquirirse para mantener la operatividad y los beneficios generados o inducidos por el proyecto (párr. 1-3).

Estos costos operativos obedecen dentro de la Cooperativa, aquellos aspectos relacionados con los servicios básicos que necesita la Cooperativa Sulá Batsú, para operar, estos son como: la luz, el agua, el alquiler, el Internet, la limpieza, entre otros... Es importante determinarlos y tenerlos presentes para que el flujo de trabajo no se vea interrumpido y así los proyectos que se vayan a elaborar, cuenten con el costo que refleja el esfuerzo que la Cooperativa realiza para terminar sus proyectos.

Gastos administrativos u overhead.

Según Nuñez (2016) los gastos administrativos:

Son aquellos que tienen que ver directamente con la administración general del negocio y no con sus actividades operativas. No son Gastos de Ventas, no son Costos de Producción. Contienen los salarios del Gerente General, secretarías, contadores, alquileres de oficinas, papelería de oficinas, suministros y equipo de oficinas, etc. No se incluyen en esta categoría los gastos que tienen que ver propiamente con la operación del

giro del negocio ni con el mercadeo. Usualmente se incluyen los gastos de selección y reclutamiento de personal (párr. 5).

Este es un gasto que siempre se presentará en los distintos proyectos que se generen dentro de la Cooperativa, ya que a pesar de que son proyectos sociales los que se realizan mayormente, la tarea de su administración, la contabilidad y la provisión de suministros adecuados para el proyecto debe ser adquirida mediante una inversión realizada.

Excedentes.

Según SUMMON PRESS, S.L (s.f):

El término excedente se utiliza habitualmente en ámbitos económicos, para hacer referencia a las cantidades sobrantes de algo. El excedente tiene su origen tras el desarrollo de la agricultura y ganadería, momento en el que empiezan a producirse más alimentos de los consumidos por el ser humano. Estas cantidades sobrantes o excedentes empiezan a ser utilizados como moneda de cambio en el trueque (párr. 1).

Dentro la Cooperativa este término hace referencia a aquella cantidad de dinero entre las diferencias entre ingresos y gastos de un periodo, los cuales distribuirán entre los diferentes asociados de la Cooperativa de acuerdo con los proyectos en los cuales estos trabajaron y a la jornada en que fueron asignados a cada uno.

Balance de comprobación.

Según Albarrán (2014):

El Balance de comprobación de sumas y saldos es un informe contable que contiene la suma de todas las cantidades anotadas al Debe, de todas las anotadas al Haber, y del saldo de cada una de las cuentas, y el de todas las cuentas en su conjunto, en un momento dado (p. 56).

Es gracias a este balance que se puede determinar el cómo, el cuándo y en qué se gastó el dinero en un determinado tiempo, permitiendo verificar el estado en que se encuentra la empresa o la persona jurídica.

Figura 10. Balance de comprobación

Astilleros del Sur Ayamonte Huelva		Balance de Comprobación de Sumas y Saldos		Del 01/01/2009 al 31/03/2009	
Cuentas	Descripción	SUMAS		SALDOS	
		DEBE	HABER	DEBE	HABER
100	Capital social		100.000		100.000
112	Reserva legal		7.000		7.000
129	Resultado del ejercicio		4.400		4.400
170	Deudas a largo plazo con entidades de cto.		5.000		5.000
210	Terrenos y bienes naturales	39.000		39.000	
211	Construcciones	17.000		17.000	
...
...
...

Fuente: Albarrán J.

Balance de situación.

Según Albarrán (2014):

El Balance de situación es una fotografía de la empresa que muestra la situación patrimonial de la empresa en un momento determinado, en el que se refleja por un lado la estructura económica (Activo) y por otro, la estructura financiera (Pasivo). Por un lado [sic] el destino de los fondos con que cuenta y por otro lado los propios fondos o fuentes de financiación del activo (p. 56-57).

Gracias a este balance se observa en qué situación se encuentra la empresa un tiempo específico, permitiendo a los altos mandos tomar decisiones oportunas, de haber algún tipo de inconveniente en éste.

Figura 11. Balance de situación

Balance de situación	
ACTIVO	PASIVO
Activo no corriente	Patrimonio neto
Intangible	Capital
Tangible Material	Reservas
Tangible Financiero	Pasivo no corriente
Activo corriente	Pasivo corriente
Existencias	/
Realizable	
Disponible	
Total	Total.....

Fuente: Albarrán J.

Cuentas por pagar.

Según Debitoor (2017):

Consiste en una deuda contraída por la empresa relacionada directamente con la actividad económica de la empresa

Una cuenta a pagar es una cuenta deudora en una empresa e indica que ésta tiene que pagar a sus proveedores (u otros acreedores).

Los importes que son contabilizados como cuentas a pagar proceden de la compra de bienes o servicios en términos de crédito. Entonces, cuentas por pagar son similares a créditos con la diferencia de que los bancos no están involucrados (párr. 1-3).

Dentro del área de gestión de proyectos dentro de la Cooperativa se adquieren estas cuentas debido a la administración de fondos en cada uno de sus proyectos o a la adquisición de una deuda para el desarrollo ésta. Estas cuentas deben ser canceladas de manera oportuna ya que al presentarse algún atraso en ellas podría perjudicar, tanto a la Cooperativa como a los proyectos que ésta, realiza.

Cuentas por cobrar.

Según Shopify (2017)

Las cuentas por cobrar consisten en el dinero que se le debe a un negocio por los bienes o servicios que provee. Es un registro de la fecha de venta, a quién se le vendió, la cantidad y cuándo se hará efectivo el pago. Los términos de crédito podrían variar en períodos de días, semanas o incluso años (párr. 1).

Dentro de la Cooperativa estas cuentas se generan debido a la necesidad de realizar algún proyecto, antes de que ingrese dinero para su ejecución. Es debido a esto que ellas deben ser cobradas de manera oportuna para que el capital con que cuenta la Cooperativa no quede sin fondos, impidiendo así el desarrollo de más proyectos.

CAPÍTULO III

Marco Metodológico

En este capítulo se presentan los diferentes procedimientos metodológicos utilizados para la investigación, tomando en cuenta las características respectivas de cada uno de los tipos, como la presentación de las fuentes de información correspondientes junto con las variables necesarias para medir el logro de los objetivos específicos.

Enfoque de Investigación

A continuación, se muestra y explican los diferentes enfoques que existen a la hora de realizar investigaciones, a su vez, se dirá el enfoque seleccionado para esta respectiva investigación y la razón de porqué fue seleccionada.

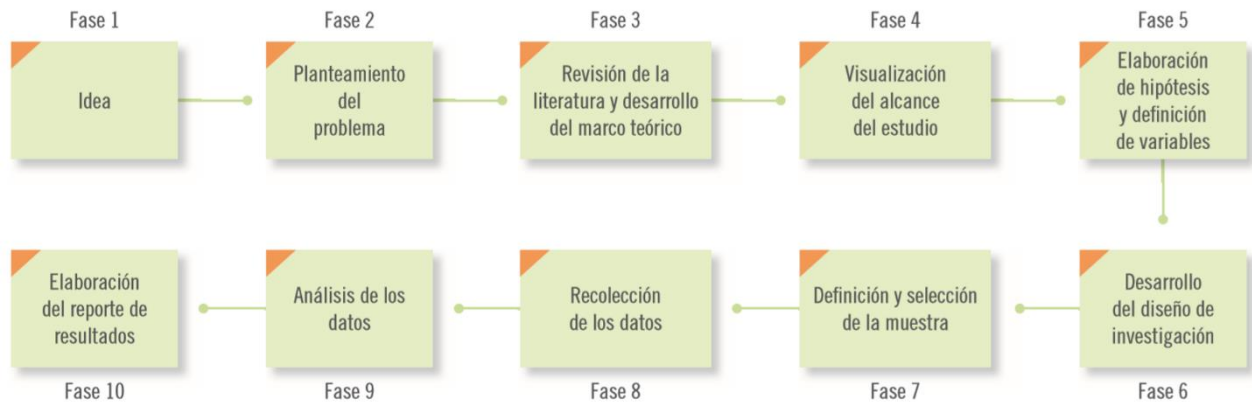
Cuantitativa.

Según Maguiña (2009) la metodología de investigación cuantitativa se puede definir como:

... aquella en la que se recogen y analizan datos cuantitativos sobre variables. Además, trata de determinar la fuerza de asociación o correlación entre variable, la generalización y objetivación de los resultados a través de una muestra para hacer inferencia a una población de la cual toda muestra procede (p. 8).

Debido a esto, se puede deducir que el enfoque cuantitativo se refiere a aquellos aspectos que posee la investigación que sean cuantificables y medibles, mediante cálculos estadísticos respaldados y certificados para obtener información veraz. Estos valores pueden ser adquiridos mediante diferentes herramientas, como las encuestas o cuestionarios, las cuales permiten percibir diferentes respuestas u opiniones, para así cuantificarlas alcanzando porcentajes o números que las respalden.

Figura 12. Proceso de investigación cuantitativa



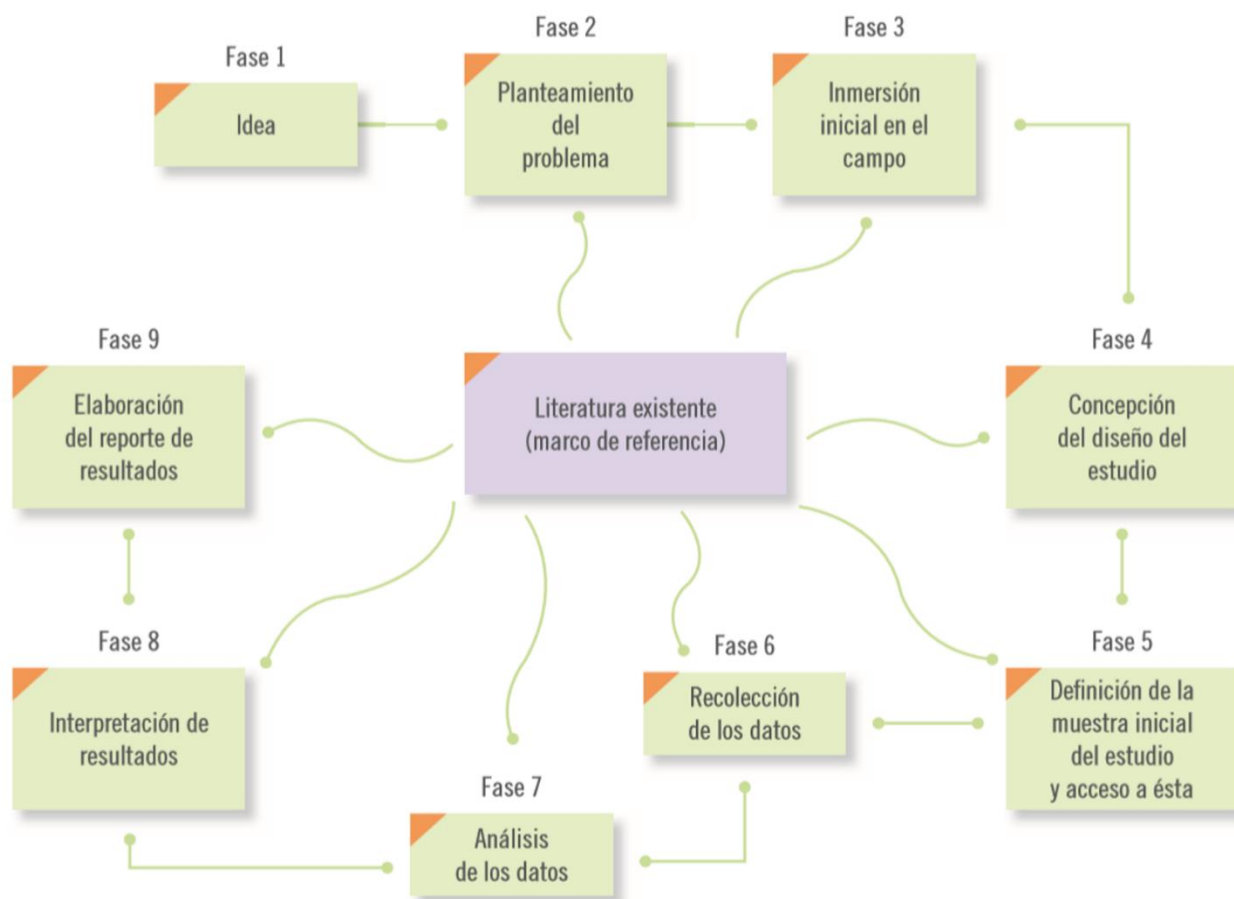
Fuente: Hernández, Fernández y Baptista P (2010).

Cualitativa.

La investigación cualitativa se refiere a todos aquellos aspectos que quieran ser estudiados dentro de la investigación, pero que no puedan ser cuantificados o medibles mediante cálculos, como dice Maguiña (2009) La investigación cualitativa estudia las realidades humanas como estructuras, totalidades, como sistemas, donde el todo no se explica por el estudio analítico de las partes [sic] sino que las partes, por su inserción en un todo, adquieren especial significado y explicación (p.9).

Debido a esto el aspecto cualitativo en las investigaciones sirve de gran medida para completar todas aquellas áreas que no pueden ser explicadas mediante análisis lógicos, ya que ésta permite estudiar el comportamiento de los individuos que se estudiarán, así como sus respuestas voluntarias e involuntarias, o comprender de manera más clara los procesos estudiados. Algunas de las técnicas que se utilizan para poder recolectar información son las técnicas de la observación y las entrevistas.

Figura 13. El proceso cualitativo

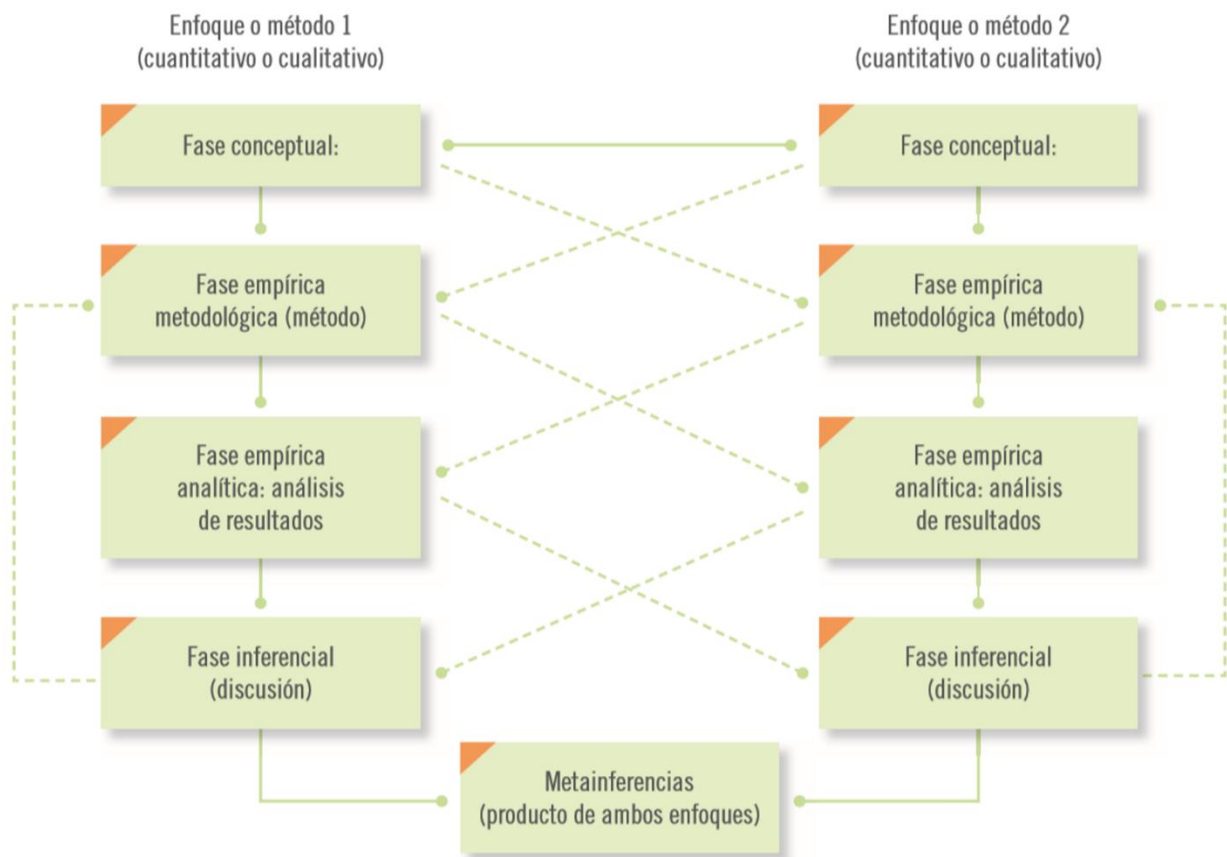


Fuente: Hernández, Fernández y Baptista P (2010).

Mixto.

Teniendo en cuenta las características de los dos métodos anteriormente mencionados, se podría decir que ambos métodos de investigación se complementan de cierta manera, ya que la cuantitativa al verificar todos aquellos aspectos de forma racional y estructurada y la cualitativa, por otro lado, interpreta y observa continuamente los procesos realizados, se puede llegar a obtener un análisis más profundo de lo que se investiga, como lo dice Ernesto y Luis (2013) “El cruce, la mixtura de los modelos cuantitativo y cualitativo aumenta la posibilidad de producir nuevos conocimientos, la formulación de nuevas teorías y el zanjar problemas, que empleando un solo enfoque quedarían sin resolver o no explicados de manera acabada” (p. 44).

Figura 14. Método mixto



Fuente: Hernández, Fernández y Baptista P (2010).

Método utilizado.

Debido a las diferentes características que presentan en cada uno de los enfoques de investigación previamente presentada, se podría concluir que para la elaboración de esta investigación se tomó un enfoque cualitativo, esto debido a que dicho enfoque permite tener una mayor comprensión de los procesos que se van a estudiar, ya que se realizaron entrevistas y observaciones sobre todos aquellos procesos que la asociada ejerce en su día a día, para comprenderlos de manera minuciosa y, así, verificar la validez de éstos para implementarlos en la elaboración del prototipo y a que se posee solamente una persona para la recolección de información referente a la gestión de proyectos dentro de la Cooperativa por lo que no se pudo obtener datos estadísticos para utilizarlos en la investigación.

Tipos de Investigación

Se presentan y explican de manera general los diferentes tipos de investigaciones que existen, donde posteriormente, se define el tipo al que esta investigación pertenece y el porqué de ello.

Descriptiva.

Su propósito es describir la realidad objeto de estudio, un aspecto de ella, sus partes, sus clases, sus categorías o las relaciones que se pueden establecer entre varios objetos, con el fin de esclarecer una verdad, corroborar un enunciado o comprobar una hipótesis. Se entiende como el acto de representar por medio de palabras las características de fenómenos, hechos, situaciones, cosas, personas y demás seres vivos, de tal manera que quien lea o interprete, los evoque en la mente (Niño, 2011, p. 34).

Como se dice anteriormente el tipo descriptivo se encarga de describir todos aquellos aspectos en los que la investigación se desenvuelva, permitiendo que el lector sea capaz de comprenderlos de la manera más clara y precisa posible, por lo que éstas deben realizarse de forma precisa y abundante evidenciándolos todos.

Exploratoria.

Como dice Niño (2011):

La investigación exploratoria tiene como propósito: "... proporcionar una visión general sobre una realidad o un aspecto de ella, de una manera tentativa o aproximativa. Este tipo de estudios es necesario cuando todavía no se dispone de los medios o no hay acceso para abordar una investigación más formal o de mayor exhaustividad" (p. 33).

Es debido a esto que la investigación exploratoria es comúnmente utilizado cuando se dan los primeros pasos de una investigación, ya que permite obtener una comprensión más clara de aquellos aspectos que se van a estudiar y así poder delimitar la investigación por realizar, permitiendo al investigador establecer un punto de partida sólido de la cual apoyarse.

Explicativa.

La explicación también es un instrumento utilizado en muchos tipos de investigación; es casi que el objetivo final, la meta o la exigencia, ya que busca respuesta a una pregunta fundamental, por el deseo de conocer y saber del ser humano: “¿Por qué?”. Averigua las causas de las cosas, hechos o fenómenos de la realidad (Niño, 2011, p.35).

Por lo que se puede deducir que la explicativa es similar a la descriptiva, sin embargo, ésta se enfoca de profundizar más en los procesos que lo componen, las partes que lo forman y las razones por la que se realizan o se comporta. Por lo tanto, la investigación de todos los elementos debe ser de una manera exhaustiva y estructurada para poder obtener resultados determinantes para que el lector pueda comprenderlos con mayor facilidad.

Tipo utilizado.

Para esta investigación se buscó realizar una de tipo descriptiva en la que se identificaron todos las, variables, componentes y procesos que se realizan en la Cooperativa diariamente, referente a la administración de proyectos, para examinarlos paso a paso y verificar la funcionalidad de cada uno de estos, con el fin de corregirlos, de ser necesario, e implementarlos en el prototipo funcional, permitiendo, así la automatización de los procesos y la satisfacción de los usuarios que utilizarán el prototipo día a día.

Fuentes de Información

A continuación, se mencionan las diferentes fuentes de información que serán utilizadas para la elaboración de la investigación y del prototipo final.

Primarias.

Las fuentes primarias de información son aquellas que son directamente obtenidas del objeto a ser estudiado por el sustentante mediante la recolección de información. En este caso la información fue brindada de primera mano por la asociada de la Cooperativa Sulá Batsú,

que tiene como responsabilidad la administración de los proyectos de ésta, para obtener los datos y la información de cada una de las tareas o procesos que fueron implementadas dentro del prototipo.

Secundarias.

Las secundarias son aquellas fuentes que son utilizadas para complementar la información brindada en las fuentes primarias, éstas pueden permitir aclarar aún más la información que haya quedado confusa o incompleta. Para el caso de la investigación, se utilizaron libros de consulta referentes a la administración de proyectos y al área de contabilidad, para así aclarar conceptos, procedimientos y cálculos, que son realizados en las diferentes tareas de la asociada dentro de la Cooperativa Sulá Batsú.

Terciarias.

En caso de las terciarias, se refiera a aquellas que son utilizadas como material de consulta en índices, directorios o catálogos de libros que permitan obtener cualquier información adicional que sea breve y precisa. En el caso de esta investigación no se utilizaron fuentes de información terciarias para su realización.

Descripción de Variables

Se definen las diferentes variables que constituyen la investigación planteada para la realización del prototipo funcional para la gestión de proyectos de Sulá Batsú.

Variable Conceptual

“La variable conceptual es aquella explicación o definición sobre la misma variable en sí, permitiendo comprender de mejor manera qué significado posee o como [sic] esta misma funciona, por los fenómenos que la componen.” (Henández, Collado y Baptista, 1997. p.110) Esto permite que el investigador pueda obtenerlas, medirlas y comprobarlas mediante

operaciones e instrumentos, dando así al lector una visión amplia y clara sobre lo que se está definiendo como variable de investigación.

Variable Operacional

Según Hernández, Fernández y Baptista P (2010), citando a Reynolds (1986) se puede definir como "... el conjunto de procedimientos que describe las actividades que un observador debe realizar para recibir las impresiones sensoriales, las cuales indican la existencia de un concepto teórico en mayor o menor grado." (p.111).

Esto quiere decir que en esta variable se establecerán todos aquellos procedimientos o actividades que se realizarán junto con la utilización de los instrumentos para poder conseguir la variable establecida en la investigación.

Variable Instrumental

Como dice Niño (2011) los instrumentos son:

... recursos, medios, útiles, pruebas o herramientas, los instrumentos tienen como función "capturar los datos o la información" requerida con el fin de verificar el logro de los objetivos de la investigación, medir las variables y validar la hipótesis, en caso de que se contemplen (p. 87).

Por lo tanto, en las variables instrumentales se establecerán todos aquellos instrumentos necesarios para recolección información para medir todas las variables establecidas, mediante la realización de las actividades de las variables operacionales.

Cuadro de variables.

Tabla 4. Cuadro de Variables

Objetivo Específico	Variable	Variable Conceptual	Variable Operacional	Variable Instrumental
<p>Analizar todos los requerimientos necesarios para la realización del prototipo funcional de la Cooperativa Sulá Batsú, mediante encuestas, entrevistas y observaciones.</p>	<p>- Requerimientos.</p>	<p>Según Senn (1992) Un requerimiento es una característica que debe incluirse en un nuevo sistema. Ésta puede ser la inclusión de determinada forma para capturar o procesar datos, producir información, controlar una actividad de la empresa o brindar soporte a la gerencia (p. 122).</p>	<p>- Reuniones con la asociada.</p>	<p>- Entrevistas. - Observación.</p>
<p>Realizar todos los diseños del prototipo funcional con el fin de completar el prototipo funcional utilizando diagramas entidad – relación, diagrama de flujo y diseño de interfaces.</p>	<p>- Diagrama de Entidad – Relación - Diseño de interfaces. - Diagramas de Flujo</p>	<p>Según Bravent (2017) el diseño se considera: Es el momento en que los profesionales tienen que aportar sus conocimientos, experiencia y creatividad para llegar a una solución que cumpla con los requerimientos funcionales y no funcionales establecidos</p>	<p>- Creación de los diferentes diagramas.</p>	<p>- Power Point 2016 - Dia</p>

		<p>en la fase de la toma de requisitos. (pp. 1-2).</p>		
<p>Programar la correcta funcionalidad del prototipo funcional mediante las herramientas de desarrollo MySQL Workbench, Materialize. PHP, HTML5, CSS3, JavaScript, JQuery y Ajax</p>	- Prototipo	<p>Como lo dice Universia España (2015) Un prototipo es una simulación del producto final. Es como una maqueta interactiva cuyo objetivo principal es probar si el flujo de interacción es el correcto o si hace falta corregirlo.</p> <p>Los prototipos dan vida a cualquier diseño y proporcionan una gran cantidad de información sobre la interacción del usuario en varios niveles. No sólo nos permiten poner a prueba la viabilidad y la utilidad de nuestros diseños antes de que se comience a programar, sino que también ayudan [sic] a descubrir mejoras e innovaciones inesperadas que pueden hacer nuestro proyecto</p>	- Los diferentes módulos necesarios del prototipo.	<ul style="list-style-type: none"> - Brackets. - My SQL Workbench - Materialize - PHP - HTML5 - CSS3 - JavaScript - JQuery - Ajax

		aún mejor (pp. 4-5).		
Comprobar la correcta funcionalidad del prototipo funcional para asegurar el cumplimiento de las soluciones propuestas mediante pruebas unitarias.	- Pruebas	Como dicen Gutiérrez y Pagés (2008) las pruebas sirven “para garantizar que se han realizado todo tipo de controles y, por lo tanto, el producto obtenido tiene la calidad deseada.” (p. 62).	- Realización de las pruebas establecidas.	- Pruebas unitarias. - Pruebas de caja negra. - Pruebas de caja blanca.

Fuente: elaboración propia.

Población

Según Niño (2011):

Cundo [sic] se trata de especificar el objeto de estudio, es necesario partir de la identificación de la población que se va a estudiar, constituida por una totalidad de unidades, vale decir, por todos aquellos elementos (personas, animales, objetos, sucesos, fenómenos, etcétera) que pueden conformar el ámbito de una investigación (p. 55).

Debido a esto la población, para esta investigación serán aquellas personas que utilizarán el prototipo funcional, por lo que esta se definiría por la asociada encargada de la administración de proyectos, dejando una población de solo un miembro.

Muestra

Según Niño (2011):

Es posible abordar una investigación sobre la totalidad de una población, si esta es pequeña, por ejemplo, los alumnos del grado quinto del Colegio San Martín, matriculados en el año 2011. Pero en realidad, la mayoría de las veces a un investigador le es imposible abordar un estudio de todas y cada una de las unidades de una población, por ser

numerosa, como es el caso de los alumnos y aún de los docentes de la Universidad Nacional. Entonces surge la necesidad de seleccionar y fijar una muestra de la población elegida (p. 55).

Por lo tanto, la muestra en esta investigación, será del mismo tamaño de la población, al estar constituida solamente por un miembro, debido a eso no se utilizará ninguna fórmula para su definición.

Instrumentos para la recolección de datos

Según Sabino (1996) citado por Mota, Guevara, Jiménez, Hernández y Armas (2014) dice que "... un instrumento de recolección de datos es, en principio, cualquier recurso de que pueda valerse el investigador para acercarse a los fenómenos y extraer de ellos información" (p. 3). Por lo que para la realizar esta investigación, la cual tiene un enfoque cualitativo, se utilizarán dos instrumentos, estos son las entrevistas y las observaciones, permitiendo tener un acercamiento mayor a la población previamente mencionada, aclarando, tanto la información recolectada como las tareas y procesos que realiza.

Entrevista.

Siguiendo la definición de Niño (2011):

La entrevista es una técnica, fundamentalmente de tipo oral, basada en preguntas y respuestas entre investigador y participantes, que permite recoger las opiniones y puntos de vista de dichos participantes o, eventualmente, según objetivos, intercambiar con ellos en algún campo (p. 64).

Debido a esto se estableció una guía de entrevista, la cual se presenta en el Apéndice A, para recolectar la información necesaria de la asociada dentro de la Cooperativa, ya que es ella la que posee mayor conocimiento de los diferentes requerimientos que debe poseer el prototipo por realizar.

Observación.

Como lo describe Niño (2011):

Observar es un acto mental bien complejo. Implica mirar atentamente una cosa, una persona o ser vivo, un fenómeno o una actividad, percibir e identificar sus características, formas y cualidades, registrarlas mediante algún instrumento (o al menos en la mente), organizarlas, analizarlas y sintetizarlas. No basta con “ver” las cosas, proceso fisiológico que se genera en los sentidos. Es necesario “mirar”, proceso cognitivo que, aunque se inicia como [sic] ver, exige una actividad de la mente (p. 62).

Por lo que se realizará una guía de observación, la cual se presenta en el Apéndice B, de la manera más detallada posible para llegar a comprender todas aquellas partes que componen las tareas y los procesos ejecutados por la asociada en su día a día, con el fin de automatizar y mejorar aquellos que formen parte del prototipo por realizar.

Proceso para recolección y análisis de datos

El proceso de recolección de información se realiza mediante 5 citas predefinidas con la asociada dentro de la cooperativa, donde se le aplicará la entrevista en la primera cita, en las dos siguientes se hacen las observaciones necesarias de los procesos más importantes, obteniendo así la información requerida, el paso a paso de éstos y notas adicionales sobre el comportamiento de la asociada a la hora de realizar los procesos, después en las dos restantes se verifica la información recibida en caso de existir dudas al respecto, con el fin de obtener los datos de la manera más clara posible.

Durante los días de entrevistas se analiza la información obtenida para obtener los requerimientos funcionales y no funcionales del prototipo, así como los datos que componen el sistema, a su vez puede existir la posibilidad de que se presenten datos ambiguos o confusos de analizar para el sustentante, por lo que se aclararán por las dos citas restantes, como anteriormente se mencionó.

CAPÍTULO IV

Análisis de Resultados

En este capítulo se realiza el análisis de información recolectada mediante el proceso que se estableció previamente, en el cual se utiliza observaciones para validar los diferentes procesos que la asociada realiza en la gestión de proyectos y así definir los requerimientos funcionales y no funcionales necesarios para su respectiva integración en el prototipo, a su vez, mediante la entrevista se especifican diferentes aspectos que permiten el desarrollo del prototipo con las tecnologías seleccionadas y la necesidad de éste.

A continuación, se definirán los diferentes módulos de acuerdo con los requerimientos especificados en la recolección de información:

- Módulo de seguridad:

Según la información recolectada se necesita por parte del sistema un medio para autenticar las credenciales de cada uno de los asociados o asociadas que tengan acceso al prototipo, la cual por el momento es solamente uno, sin embargo, se prevé que éste pueda ser utilizado por más en su implementación.

- Módulo de mantenimientos:

Identificando las diferentes entidades que la asociada utiliza para realizar la gestión de proyectos dentro de la Cooperativa Sulá Batsú, se determinaron trece diferentes entidades que se encargarán de poseer los datos necesarios para formular la información requerida, agilizando el proceso y la realización de las tareas en la gestión de proyectos. Sin embargo, las entidades que se encuentran dentro de los mantenimientos son:

- ❖ Asociados.
- ❖ Colaboradores.
- ❖ Proyectos.
- ❖ Rubros.
- ❖ Categoría Rubros.
- ❖ Categoría Gastos Operativos.
- ❖ Categoría Flujos de Ingreso.
- ❖ Categoría Flujos de Gastos de Ejecución.

- ❖ Gastos Operativos.
- ❖ Clientes.
- ❖ Proveedores.
- Módulo de reportes:

Según las tareas diarias que realiza la asociada para la gestión de proyectos y las necesidades especificadas se determinaron los siguientes reportes:

 - ❖ Un reporte general de las cuentas por pagar y/o cobrar donde se pueda observar aquellas que están próximas a vencer o en caso de ser necesario las vencidas, a su vez, se podrán filtrar para observar las que se encuentren dadas de baja.
 - ❖ Estado financiero del proyecto dando a conocer el estado general de cada uno de los proyectos con respecto de su avance, cuántas pérdidas ha generado, saldos dejados.
 - ❖ Reporte de los costos operativos de cada uno de los proyectos mostrando todos aquellos costos que van ligados al proyecto como el pago de excedentes, los activos asociados a éste, los contratos realizados para su realización y transporte de haberse utilizado.
 - ❖ Reporte de distribución de excedentes con respecto del fondo del proyecto y la jornada de cada uno de los asociados
- Módulo Consulta:

Debido a las distintas entidades que se definieron previamente, de acuerdo con las necesidades de la Cooperativa, se determinaron consultas generales para cada uno de éstas, permitiendo ver los datos guardados. Pero, a su vez, se definieron consultas específicas que permiten filtrar los datos de las entidades para así conseguir la información requerida de cada una si se requiere, además de habilitar una consulta de aquellos registros que podrían estar deshabilitados dentro de la base de datos, esto con las entidades de asociados, proyectos, clientes, proveedores, cuentas por pagar y cuentas por cobrar.

- **Módulo de Cálculos Financieros:**

Se determinó el requerimiento de este módulo debido a que la asociada realiza diferentes operaciones matemáticas dentro de las hojas de Excel, con los diferentes datos almacenados para obtener la información deseada, sin embargo, dichas fórmulas deben ser automatizadas minimizando el riesgo de la manipulación de éstas, es por ello que los flujos de los proyectos se realizarán dentro de este módulo.

- **Módulo de Cuentas por Cobrar:**

Módulo determinado por la necesidad de la asociada, de registrar las diferentes cuentas por cobrar de cada proyecto. Este, a su vez, permitirá enviar alertas sobre los plazos de fechas de cada cuenta.

- **Módulo de Cuentas por Pagar:**

Módulo determinado por la necesidad de la asociada, de registrar las diferentes cuentas por pagar de cada uno de los proyectos dentro de la Cooperativa. Este, a su vez, permitirá enviar alertas sobre los plazos de fechas de cada cuenta.

- **Módulo de Excedentes a Asociados:**

Módulo determinado por la necesidad de la asociada y la Cooperativa de definir el monto de excedente que le corresponde a cada asociado y asociada, con respecto del porcentaje de Excedente asignado a cada uno, a su vez, también se definirá el excedente de cada colaborador y colaboradora, de acuerdo con el contrato realizado, que define la jornada y el monto que les corresponde. Todo esto de acuerdo con los proyectos a los que cada uno haya sido asignado, a su vez es necesario registrar los montos transferidos a las cuentas, como la fecha cuando estos movimientos fueron realizados mediante un historial.

CAPÍTULO V

Conclusiones

Se logró determinar los distintos requerimientos de la Cooperativa para elaborar y establecer los procesos más importantes que componen la gestión de proyectos dentro éste, así como las diferentes entidades que la conforman. Todo esto con el fin de elaborar el prototipo funcional que cumpla con la proyección funcional previamente establecida.

Las entrevistas y las observaciones realizadas clarificaron el proceso que toma la asociada de principio a fin en su labor diaria, determinando los diferentes requerimientos funcionales y aquellos requerimientos no funcionales, pero de importancia que debe tener el prototipo para completar su funcionalidad a la Cooperativa.

El proceso de diagramación del prototipo es un paso esencial ya que permite plantear una guía al desarrollar cada uno de los módulos de manera correcta, es por ello que se estipularon y crearon los distintos diagramas, los cuales son: el diagrama de Entidad – Relación, que permitió crear una base de datos normalizada, los diagramas de flujos, que clarificó y determinó el proceso paso a paso de las tareas más importantes dentro del prototipo funcional y finalmente, los distintos diseños de interfaces permitiendo crear un entorno de simplicidad y familiaridad con el usuario que posibilita una menor resistencia al uso del prototipo funcional.

Los diferentes elementos utilizados para crear el prototipo funcional son exactamente para desarrollo web, por lo que le sirve de gran medida a la Cooperativa ya que ellos no poseen servidores de ningún tipo, pero sí mantienen un contrato de servicio de hosting para su propia página web. A su vez, al utiliza solicitudes Ajax mediante JQuery y JavaScript, para solicitar datos de MySQL mediante diferentes clases en PHP7, permitiendo tener una respuesta asíncrona con las diferentes tablas de la clase JsGrid, que desde la perspectiva del usuario será más simple y fluido, ya que no deberá recargar la página cada vez que interactúe con las tablas.

Gracias a la elaboración de los distintos casos de uso se logró crear diferentes casos de prueba para verificar la funcionalidad del prototipo funcional, donde cada uno de estos fueron verificados y resultaron tener la respuesta esperada. Por lo que se puede determinar que la funcionalidad del prototipo está garantizada.

Recomendaciones

En este apartado, se mencionarán diferentes recomendaciones que brindarán mayor utilidad al prototipo funcional, ya que éstas se centrarán en extender sus funcionalidades para que puedan ayudar en una mejor manera a la Cooperativa.

- Realizar la implementación del prototipo:
La implementación del prototipo traería un gran beneficio para la Cooperativa, ya que éste fue diseñado y desarrollado siguiendo las necesidades específicas de la Cooperativa el cual permite agilizar el proceso de realización de informes y la automatización de distintos procesos que la conforman, la realización de esto tomaría un esfuerzo de unas 28 horas extra, lo que sería una inversión de 45,994.76 colones.
- Incluir un módulo de facturación para la integración de factura electrónica:
Actualmente, la cooperativa se está integrando a las regulaciones de Hacienda para los registros de factura electrónica que por ley se piden, por lo que, ellos utilizan la plataforma proporcionada por el mismo ente. Sin embargo, la Cooperativa desea que desde el prototipo desarrollado se pueda implementar un módulo para el manejo de esto, por lo que se define que tomaría un tiempo de 124 horas, lo que sería una inversión de 203,691.08 colones.
- Incluir un módulo para el registro de vacaciones de los asociados y asociadas:
La Cooperativa opera con distintos asociados que trabajan en diferentes proyectos todos los años, estos miembros, a su vez, requieren tiempos de descanso o recuperación en caso de alguna urgencia de fuerza mayor aparezca por lo que desean implementar un módulo de manejo de vacaciones o faltas para sus miembros, esto tomaría un tiempo de 56 horas siendo un costo de 91,989.52 colones.
- Expandir la funcionalidad actual del prototipo a dispositivos móviles:
Propiciar un diseño totalmente “responsive” traería un beneficio a la Cooperativa ya que la persona encargada del proceso de administración de proyectos y las que, posteriormente, vayan a ser asignadas al mismo, podrán tener un acceso desde cualquier dispositivo electrónico al prototipo,

respondiendo solicitudes de la junta a mayor velocidad y sin dependencia de dispositivos específicos. Esto tomaría un tiempo de 60 horas con una inversión de 98,560.2 colones.

- Agregar un mayor manejo de monedas:

Dentro de la Cooperativa existen ocasiones donde administran proyectos de índole internacional, fuera del territorio costarricense, por lo que les resulta de gran ayuda manejar diferentes divisas para definir los diferentes montos que ellos utilizan, por lo que si el prototipo funciona las implementa aumentaría la usabilidad del mismo, para ello se tomaría un tiempo de 28 horas con un costo de 45,994.76 colones.

- Exportación e importación de formato xls o xlsx:

Como se mencionó anteriormente en la problemática la principal herramienta de trabajo utilizada para la tarea de gestionar los proyectos es Excel, es por ello que les resultaría a los usuarios un alivio tener a su disposición la posibilidad de poder exportar e importar los datos desde estos archivos, para ello, se tomaría un tiempo de 60 horas con un coste de 98,560.2 colones.

- Agregar más parámetros a los reportes:

Existen casos dentro de la Cooperativa donde a la asociada a cargo de la gestión de los proyectos se le hace la solicitud de crear diferentes reportes que poseen características específicas, como el flujo de ingresos de un mes en especial o los gastos operativos efectuados dentro del rango de unos meses e incluso la distribución de excedentes de asociados o colaboradores dentro de un rango de meses, por lo que resultaría de ayuda poderlos incluir dentro del prototipo, para ello se tomaría un tiempo de 20 horas teniendo un costo de 32,853.4 colones.

CAPÍTULO VI

Propuesta

En este capítulo se presentarán las diferentes fases de desarrollo que se realizaron para el prototipo en esta investigación, respondiendo a los requerimientos funcionales y no funcionales sobre la gestión de proyectos dentro de la Cooperativa Sulá Batsú.

Análisis

Apartado donde se determinará los distintos elementos y funcionalidades del prototipo funcional para responder a los requerimientos funcionales y no funcionales obtenidos en el análisis de datos.

Software desarrollado.

- Módulo de seguridad:

En este módulo se comprueban el usuario y contraseña del asociado permitiendo el acceso solamente a aquellos que están registrados dentro del prototipo, devolviendo a la pantalla de login a aquellos que no tengan acceso. A su vez, se consulta el rol del usuario.

- Módulo de mantenimientos:

Acá se realiza el mantenimiento rutinario de cada una de las entidades que constituyen la base de datos del prototipo funcional, en éstas se utilizará la biblioteca de JsGrid, archivos php para las solicitudes de datos y solicitudes Ajax para que la interacción con las tablas mostradas sea más dinámica, a su vez, se desplegarán diferentes mensajes de error en caso de que ocurra alguno.

- Módulo de reportes:

Para este módulo se utiliza la biblioteca de FPDF, la cual permitirá la creación de documentos PDF donde se mostrarán los datos respectivos a los reportes previamente planteados con una distribución de los datos y diseño que facilite su comprensión

- **Módulo Consulta:**

Módulo que permite la consulta de las diferentes entidades planteadas, permitiendo al usuario filtrar los datos que requiera mediante la biblioteca se JsGrid.
- **Módulo de Cálculos Financieros:**

Módulo que utiliza los archivos php bl_entidad y la biblioteca JsGrid para realizar diferentes operaciones matemáticas con los datos almacenados dentro de la base de datos, permitiendo desplegar los diferentes resultados obtenidos al usuario dentro de los flujos, rubros y excedentes.
- **Módulo de Cuentas por Cobrar:**

Módulo que utiliza la biblioteca JsGrid y archivos php para la realización de los mantenimientos de la entidad cuentas por cobrar, donde, a su vez, examina las fechas de vencimiento de éstas y despliega alertas en caso de que esa fecha se encuentre a una semana o menos de vencer.
- **Módulo de Cuentas por Pagar:**

Módulo que utiliza la biblioteca JsGrid y archivos php para la realización de los mantenimientos de la entidad cuentas por pagar, donde, a su vez, examina las fechas de vencimiento de éstas y despliega alertas en caso de que esa fecha se encuentre a una semana o menos de vencer.
- **Módulo de Excedentes a Asociados:**

Módulo que utiliza archivos de php y la biblioteca JsGrid para realizar el mantenimiento de las entidades de Excedentes, Historia Transferencias, Colaborador Proyecto y Asociado Proyecto, donde se tomará en cuenta los datos almacenados dentro de cada una de las entidades para la realización de los cálculos de los excedentes correspondientes a cada uno de los asociados, asociadas, colaboradores y colaboradoras.

Hardware requerido.

Es importante que destacar que los requerimientos de hardware presentados, son aproximaciones del hardware necesario para el desarrollo y posible implementación del prototipo, por lo cual estos pueden variar dependiendo de las circunstancias.

Para el desarrollo del prototipo, se utiliza una computadora de escritorio con las siguientes características técnicas:

- Procesador: Intel i3 6100 3.7GHz.
- Memoria RAM: 8GB.
- Almacenamiento: HDD 500GB.
- Dispositivos como: Monitor, Teclado y Mouse.

En caso de que se realice una implementación del prototipo es necesario un equipo con las siguientes características técnicas:

- Servidor web o de aplicaciones.
- Procesador: Intel o AMD de al menos 3GHz.
- Memoria RAM: al menos 8GB.
- Almacenamiento: al menos 100GB.

Por otro lado, para el uso cotidiano del prototipo por parte de los asociados y asociadas dentro de la cooperativa, sería necesario un equipo con las siguientes características:

- Procesador: Intel o AMD de al menos 1.8GHz.
- Memoria RAM: al menos 4GB.
- Almacenamiento: al menos 100GB.
- Dispositivos: Monitor, Teclado y Mouse.

Estos requerimientos de hardware están estipulados de tal manera que permite el correcto funcionamiento o en su caso, desarrollo, del prototipo, tanto dentro del entorno de hosting contratado para su ubicación web como en el ingreso diario por medio de Google Chrome al mismo.

Telecomunicaciones requeridas.

Ya que el prototipo desarrollado es un aplicativo web, éste, necesita conexión a Internet para poder usarlo, por ello es importante contar con un contrato con algún proveedor de servicio para Internet y otro para el servicio de alojamiento en la nube (hosting).

Las características del servicio a Internet deben ser como mínimo:

- Velocidad mínima de 1mbps, tanto de descarga como de subida.

- Router para conexiones Wifi, ya que la mayoría de equipos por utilizar son laptops.

A su vez, para el servicio de hosting se requerirá las siguientes características:

- Soporte a páginas PHP.
- Soporte de 24 horas los 365 días.
- Base de datos compatible con MySQL.

Estos requerimientos permiten que el prototipo se encuentre disponible en la nube en cualquier momento, a su vez, si ocurre algún imprevisto durante su alojamiento web, debido al contrato de nivel de servicio, éste se resolverá lo más pronto posible.

Base de datos.

Para el correcto funcionamiento del prototipo funcional desarrollado, se utiliza una base de datos relacional compatible dentro del servicio de hosting denominado MySQL, éste permite soportar la carga de datos del prototipo y es el encargado de responder a las solicitudes de éste mediante procedimientos almacenados establecidos. Estos procedimientos permiten agilizar las respuestas de la base de datos y, a su vez, reducir el riesgo de que se presenten datos comprometidos por inyección SQL.

Personal requerido.

Para el prototipo funcional solamente se ha definido un perfil del personal necesario para operar el prototipo en su implementación, el cual es el administrador, éste tendrá acceso total al sistema, éste puede realizar todas las funciones que, actualmente, desempeña la gestión de proyectos, dentro de la Cooperativa Sulá Batsú como: realizar los mantenimientos necesarios de cada una de las entidades de la base de datos, autogenerar reportes y calcular los pagos de excedentes de cada uno de los asociados, asociadas, colaboradores y colaboradoras dentro del mismo.

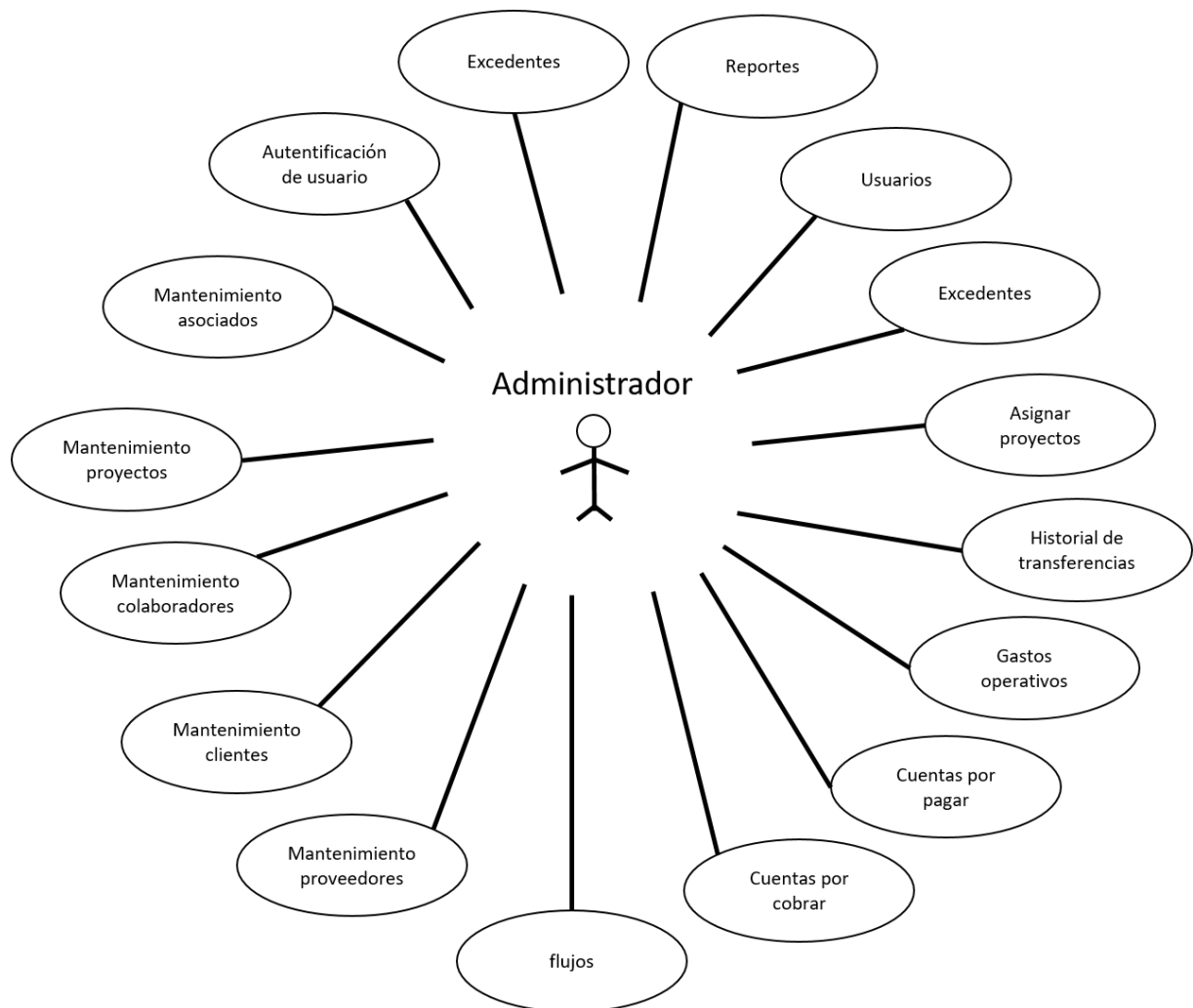
Los conocimientos que necesita la persona encargada de desempeñar esta función son:

- Contabilidad.
- Base de datos.
- Gestión.

Casos de Uso.

Gracias a los distintos requerimientos reunidos dentro del análisis de resultados de la información recolectada, se definieron los siguientes casos de uso:

Figura 15. Modelo de casos de uso.



Fuente: Elaboración propia.

En el modelo anterior se puede observar que el prototipo funcional solamente cuenta con un actor denominado "Administrador", éste es el encargado de realizar todos los casos de uso definidos, los cuales su totalidad conforman 16 diferentes casos de uso.

Tabla 5. Caso de uso autenticación de usuario

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 01	Nombre del Caso de Uso: Autenticación de usuario.
Fecha de elaboración:	26/11/2018
Descripción Caso de Uso:	Como el sistema realizará la autenticación del usuario.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	Asociada encarga de la gestión de proyectos.
Precondiciones:	El usuario tiene que estar registrado en el prototipo.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. La asociada ingresa a la página web del prototipo. 2. El sistema muestra la pantalla de login, con un formulario que pide usuario y contraseña. 3. La asociada digita el usuario y contraseña en el formulario y presiona ingresar. 4. El sistema verifica los campos (FA1 – Validación de usuario) y redirige el usuario a la página de inicio. 5. Fin del caso de uso. 	
Flujos Alternos	
Flujo Alterno No. 1 Validación de usuario.	El sistema validará sí los valores de usuario y contraseña existen dentro de la base de datos, en caso contrario, mostrará un mensaje de error que indica que el usuario o la contraseña son incorrectos.
Post-Condiciones	
La asociada queda autenticada en el prototipo y puede acceder a todas las opciones del mismo.	

Fuente: Elaboración Propia.

Tabla 6. Caso de uso mantenimiento asociados.

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 02	Nombre del Caso de Uso: Mantenimiento Asociados
Fecha de elaboración:	26/11/2018
Descripción Caso de Uso:	En este caso de uso se definen los procesos de consulta, inserción, modificación y eliminación lógica de los registros en la tabla de asociados de la base de datos
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	Usuario administrador.
Precondiciones:	La correcta autenticación del usuario.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Mantenimientos – Asociados. 2. El sistema redirige a la asociada a la pantalla de mantenimiento asociados. 3. El usuario ve la pantalla de mantenimiento con una tabla que muestra los registros habilitados de la entidad y diferentes botones en la misma. <ol style="list-style-type: none"> a. Interruptor “Deshabilitados” desactivado (SF1). b. Botón “+” verde en la última columna al lado de los encabezados (SF2). c. Botón de lupa en la última columna (SF3). d. Botón de limpieza al lado de la lupa en la última columna (SF4). e. Botón de lápiz verde en la última columna de los distintos registros (SF5). f. Botón de basurero rojo en la última columna de los distintos registros (SF6). 4. El usuario digita los datos deseados en la primera fila y presiona Enter. 5. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario. 6. Fin del caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario activa el interruptor “Deshabilitados” 2. El sistema muestra todos los registros que se encuentran dentro de la base de datos, habilitados como deshabilitado, estos últimos con un fondo gris y un botón en la última columna “Restaurar” (SF7).

Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, a su vez la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación de datos asociados) y actualiza la tabla con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario ingresa datos en la primera fila, y presiona Enter o le da click a la al icono de la lupa en la última columna. 2. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario presiona el botón. 2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables y la última columna cambia por con un check verde (SF8) y una “x” roja (SF9).
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la una columna de un registro en la tabla desplegada. 2. El sistema deshabilita el registro de la fila seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón “Restaurar”. 2. El sistema actualiza el registro dentro de la base de datos y cambia a habilitado, refrescando la tabla con el nuevo registro habilitado.
Sub Flujo 8	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del

	<p>check verde en la última columna.</p> <p>2. El sistema modifica los datos (FA1 – Validación de datos asociados) y actualiza la tabla mostrada</p>
Sub Flujo 9	<p>1. El usuario da click al botón de “x” rojo en la última columna de la fila.</p> <p>2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro</p>
Flujos Alternos	
Flujo Alternativo No 1 Validación de datos asociados	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Cédula: No se encuentre vacío y debe tener 11 caracteres. • Nombre: No se encuentre vacío. • Primer Apellido: No se encuentre vacío. • Segundo Apellido: No se encuentre vacío. • Teléfono: No se encuentre vacío y debe tener 8 dígitos. • Cuenta ¢: No se encuentre vacío y debe tener 20 caracteres. • Cuenta \$: Debe tener 20 caracteres y solo números. <p>En caso que alguno de estos se presente, se desplegará como una alerta los errores correspondientes.</p>
Post-Condiciones	
<p>En caso de que el usuario realice una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de datos.</p> <p>En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.</p> <p>En caso de haber eliminado un registro, éste quedará deshabilitado dentro de la base de datos, sin eliminar totalmente el registro.</p>	

Fuente: Elaboración Propia.

Tabla 7. Caso de uso mantenimiento proyectos

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 03	Nombre del Caso de Uso: Mantenimientos Proyectos
Fecha de elaboración:	26/11/2018
Descripción Caso de Uso:	En este caso de uso se definen los procesos de consulta, inserción, modificación y eliminación lógica de los registros en la tabla de proyectos de la base de datos.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador.
Precondiciones:	La correcta autenticación del usuario. Un cliente registrado y habilitado dentro del prototipo.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Mantenimientos – Proyectos. 2. El sistema redirige a la asociada a la pantalla de mantenimiento proyectos. 3. El usuario ve la pantalla de mantenimiento con una tabla que muestra los registros habilitados de la entidad y diferentes botones en la misma. <ol style="list-style-type: none"> a. Interruptor “Deshabilitados” desactivado (SF1). b. Botón “+” verde en la última columna al lado de los encabezados (SF2). c. Botón de lupa en la última columna (SF3). d. Botón de limpieza al lado de la lupa en la última columna (SF4). e. Botón de lápiz verde en la última columna de los distintos registros (SF5). f. Botón de basurero rojo en la última columna de los distintos registros (SF6). 4. El usuario digita los datos deseados en la primera fila y presiona Enter 5. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario. 6. Fin del caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario activa el interruptor “Deshabilitados” 2. El sistema muestra todos los registros que se encuentran dentro de

	la base de datos, habilitados como deshabilitado, estos últimos con un fondo gris y un botón en la última columna “Restaurar” (SF7).
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, excepto las columnas presupuesto donde en cada una se muestra un valor de 0 que no puede ser modificado y a su vez la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación de datos proyecto) y actualiza la tabla con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario ingresa datos en la primera fila, y presiona Enter o le da click a la al icono de la lupa en la última columna. 2. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables, excepto que con Presupuesto (℄), donde se muestra un botón “Ver Rubros” (SF8), y la última columna cambia por los botones check verde (SF9) y una “x” roja (SF10).
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada.

	<ol style="list-style-type: none"> 2. El sistema deshabilita el registro de la fila seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón “Restaurar”. 2. El sistema actualiza el registro dentro de la base de datos y cambia a habilitado, refrescando la tabla con el nuevo registro habilitado.
Sub Flujo 8	<ol style="list-style-type: none"> 1. El usuario presiona el botón “Rubros” de la columna Presupuesto. 2. El sistema muestra un panel con los registros de rubros, el fondo oscurecido y los siguientes botones: <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF11). b. Botón de lupa en la última columna (SF12). c. Botón de limpieza al lado de la lupa en la última columna (SF13). d. Botón de lápiz verde en la última columna de los distintos registros (SF14). e. Botón de basurero rojo en la última columna de los distintos registros (SF15). f. Botón “Cerrar” al final del panel (SF16)
Sub Flujo 9	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 – Validación de datos proyectos) y actualiza la tabla mostrada.
Sub Flujo 10	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro
Sub Flujo 11	<ol style="list-style-type: none"> 1. El usuario da click al botón “+” verde de la tabla. 2. El sistema sustituye la fila de filtrado por una con fondo verde y campos para ingresar datos, donde categoría se muestra una lista de categorías registradas, con una opción que dice “Nueva Categoría” (SF17). 3. El usuario ingresa los datos en cada uno de los campos. 4. El sistema ingresa los datos digitado (FA2 – Validación de datos

	rubros) y refresca la tabla con el nuevo registro.
Sub Flujo 12	<ol style="list-style-type: none"> 1. El usuario ingresa datos en la primera fila y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados.
Sub Flujo 13	<ol style="list-style-type: none"> 1. El usuario da click al botón de limpieza. 2. El sistema limpia los campos de la primera fila y refresca la tabla para mostrar todos los registros de rubros.
Sub Flujo 14	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables, donde la columna categoría muestra una lista de opciones, entre las cuales esta “Nueva Categoría.” (SF17) y la última columna cambia por los botones check verde (SF18) y una “x” roja (SF19).
Sub Flujo 15	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema borra el registro de la fila seleccionada.
Sub Flujo 16	<ol style="list-style-type: none"> 1. El usuario da click al botón “Cerrar”. 2. El sistema cierra el panel, realiza el cálculo de la totalidad de rubros para refrescar los registros de la tabla proyecto con la totalidad como presupuesto del proyecto, donde a su vez se calcula su equivalente en dólares.
Sub Flujo 17	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Nueva categoría”. 2. El sistema despliega un prompt que solicita el que ingrese una nueva categoría. 3. El usuario ingresa el nombre y presiona aceptar. 4. El sistema valida (FA3 – Validación categoría rubros), guarda el registro y refresca la lista mostrada.
Sub Flujo 18	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna.

	2. El sistema modifica los datos (FA2 – Validación datos rubros) y actualiza la tabla mostrada.
Sub Flujo 19	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación datos proyecto	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Nombre: No se encuentre vacío y no este repetido. • Fecha Inicio No se encuentre vacío. • Fecha Final: No se encuentre vacío y después de la fecha de inicio. • Presupuesto: No se encuentre vacío y no sea negativo. • Cuenta ¢: No se encuentre vacío y debe tener 20 caracteres. • Cuenta \$: No se encuentre vacío y debe tener 20 caracteres. <p>En caso que alguno de estos se presente, se desplegará como una alerta los errores correspondientes.</p>
Flujo Alternativo No. 2 Validación datos rubros	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Descripción: No se encuentre vacío. • Monto: No se encuentre vacío y no sea negativo. <p>En caso que alguno de estos se presente, se desplegará como una alerta los errores correspondientes.</p>
Flujo Alternativo No. 3 Validación categoría rubros	<p>El sistema validará si el nombre de categoría ya existe, en caso de que exista se mostrará el siguiente mensaje: La categoría ya existe.</p>
Post-Condiciones	
<p>En caso de que el usuario realice una consulta en la ventana de proyectos el sistema refresca la tabla de proyectos para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción en la tabla proyectos, el registro queda almacenado dentro de la base de datos con un presupuesto 0 en ambos casos.</p>	

En caso de haber realizado una modificación en la tabla proyectos, se pudo dar los siguientes resultados:

- El registro de proyecto se actualiza con los nuevos valores quedando almacenado dentro de la base de datos.
- Consultado algún registro del proyecto.
- Creado nuevos rubros dentro del proyecto.
- Modificado algún registro de los rubros del proyecto.
- Eliminado algún rubro del proyecto.
- Creado una nueva categoría para los rubros.

En caso de haber eliminado un registro en la tabla proyectos, éste quedará deshabilitado dentro de la base de datos, sin eliminar totalmente el registro.

Fuente: Elaboración Propia.

Tabla 8. Caso de uso mantenimiento colaboradores.

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 04	Nombre del Caso de Uso: Mantenimientos Colaboradores.
Fecha de elaboración:	26/11/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso de consultar, editar, insertar y eliminar de la entidad de Colaboradores.
Autor caso de uso:	Steven Morera Hernández.
Actores relacionados:	El usuario administrador.
Precondiciones:	Correcta autenticación del usuario.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Mantenimientos – Colaboradores. 2. El sistema redirige a la asociada a la pantalla de mantenimiento colaboradores. 3. El usuario ve la pantalla de mantenimiento con una tabla que muestra los registros habilitados de la entidad y diferentes botones en la misma. <ol style="list-style-type: none"> a. Interruptor “Deshabilitados” desactivado (SF1). b. Botón “+” verde en la última columna al lado de los encabezados (SF2). 	

<ul style="list-style-type: none"> c. Botón de lupa en la última columna (SF3). d. Botón de limpieza al lado de la lupa en la última columna (SF4). e. Botón de lápiz verde en la última columna de los distintos registros (SF5). f. Botón de basurero rojo en la última columna de los distintos registros (SF6). <ul style="list-style-type: none"> 4. El usuario digita los datos deseados en la primera fila y presiona Enter 5. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario. 6. Fin del caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ul style="list-style-type: none"> 1. El usuario activa el interruptor “Deshabilitados” 2. El sistema muestra todos los registros que se encuentran dentro de la base de datos, habilitados como deshabilitado, estos últimos con un fondo gris y un botón en la última columna “Restaurar” (SF7).
Sub Flujo 2	<ul style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos y a su vez la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación de datos colaboradores) y actualiza la tabla con los datos ingresados.
Sub Flujo 3	<ul style="list-style-type: none"> 1. El usuario ingresa datos en la primera fila y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados.
Sub Flujo 4	<ul style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.

Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables y la última columna cambia por los botones check verde (SF8) y una “x” roja (SF9).
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema deshabilita el registro de la fila seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón “Restaurar”. 2. El sistema actualiza el registro dentro de la base de datos y cambia a habilitado, refrescando la tabla con el nuevo registro habilitado.
Sub Flujo 8	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 – Validación de datos colaboradores) y actualiza la tabla mostrada.
Sub Flujo 9	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación de datos colaboradores.	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Cédula: No se encuentre vacío y debe tener 11 caracteres. • Nombre: No se encuentre vacío • Primer Apellido: No se encuentre vacío. • Segundo Apellido: No se encuentre vacío. • Teléfono: No se encuentre vacío y debe tener 8 dígitos. • Cuenta ¢: No se encuentre vacío y debe tener 20 caracteres. • Cuenta \$: Debe tener 20 caracteres y solo números. <p>En caso que alguno de estos se presente, se desplegará como una alerta los errores correspondientes.</p>

Post-Condiciones
En caso de que el usuario realizara una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo en los datos ingresados por el usuario.
En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de datos.
En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.
En caso de haber eliminado un registro, éste quedará deshabilitado dentro de la base de datos, sin eliminar totalmente el registro.

Fuente: Elaboración propia

Tabla 9. Caso de uso mantenimientos clientes

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 05	Nombre del Caso de Uso: Mantenimientos Clientes
Fecha de elaboración:	26/11/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso de consultar, editar, insertar y eliminar registros de la entidad Clientes.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador.
Precondiciones:	La correcta autenticación del usuario.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Mantenimientos – Clientes. 2. El sistema redirige a la asociada a la pantalla de mantenimiento clientes. 3. El usuario ve la pantalla de mantenimiento con una tabla que muestra los registros habilitados de la entidad y diferentes botones en la misma. <ol style="list-style-type: none"> a. Interruptor “Deshabilitados” desactivado (SF1). b. Botón “+” verde en la última columna al lado de los encabezados (SF2). c. Botón de lupa en la última columna (SF3). d. Botón de limpieza al lado de la lupa en la última columna (SF4). e. Botón de lápiz verde en la última columna de los distintos registros (SF5). 	

<p>f. Botón de basurero rojo en la última columna de los distintos registros (SF6).</p> <p>4. El usuario digita los datos deseados en la primera fila y presiona Enter</p> <p>5. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario.</p> <p>6. Fin del caso de uso.</p>	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario activa el interruptor “Deshabilitados” 2. El sistema muestra todos los registros que se encuentran dentro de la base de datos, habilitados como deshabilitado, estos últimos con un fondo gris y un botón en la última columna “Restaurar” (SF7).
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos y a su vez la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación de datos clientes) y actualiza la tabla con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario ingresa datos en la primera fila y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los

	valores mostrados previamente, pero editables y la última columna cambia por los botones check verde (SF8) y una “x” roja (SF9).
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema deshabilita el registro de la fila seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón “Restaurar”. 2. El sistema actualiza el registro dentro de la base de datos y cambia a habilitado, refrescando la tabla con el nuevo registro habilitado.
Sub Flujo 8	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 – Validación de datos clientes) y actualiza la tabla mostrada.
Sub Flujo 9	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación de datos clientes	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Cédula: No se encuentre vacío y debe tener un máximo de 12 caracteres. • Nombre: No se encuentre vacío. • Teléfono1: No se encuentre vacío y debe tener 8 dígitos. • Teléfono2: Debe tener a 8 dígitos. • Dirección: No se encuentre vacío. • Tipo: No se seleccione ningún tipo. <p>En caso que alguno de estos se presente, se desplegará como una alerta los errores correspondientes.</p>
Post-Condiciones	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo a los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de</p>	

datos.

En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.

En caso de haber eliminado un registro, este quedará deshabilitado dentro de la base de datos, sin eliminar totalmente el registro.

Fuente: Elaboración propia.

Tabla 10. Caso de uso mantenimientos proveedores

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 06	Nombre del Caso de Uso: Mantenimientos Proveedores
Fecha de elaboración:	26/11/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso de consultar, editar, insertar y eliminar registros de la entidad Proveedores.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador.
Precondiciones:	La correcta autenticación del usuario.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Mantenimientos – Proveedores. 2. El sistema redirige a la asociada a la pantalla de mantenimiento proveedores. 3. El usuario ve la pantalla de mantenimiento con una tabla que muestra los registros habilitados de la entidad y diferentes botones en la misma. <ol style="list-style-type: none"> a. Interruptor “Deshabilitados” desactivado (SF1). b. Botón “+” verde en la última columna al lado de los encabezados (SF2). c. Botón de lupa en la última columna (SF3). d. Botón de limpieza al lado de la lupa en la última columna (SF4). e. Botón de lápiz verde en la última columna de los distintos registros (SF5). f. Botón de basurero rojo en la última columna de los distintos registros (SF6). 4. El usuario digita los datos deseados en la primera fila y presiona Enter 5. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario. 	

6. Fin del caso de uso.	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario activa el interruptor “Deshabilitados” 2. El sistema muestra todos los registros que se encuentran dentro de la base de datos, habilitados como deshabilitado, estos últimos con un fondo gris y un botón en la última columna “Restaurar” (SF7).
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos y a su vez la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación de datos proveedores) y actualiza la tabla con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario ingresa datos en la primera fila y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables y la última columna cambia por los botones check verde (SF8) y una “x” roja (SF9).
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada.

	2. El sistema deshabilita el registro de la fila seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón “Restaurar”. 2. El sistema actualiza el registro dentro de la base de datos y cambia a habilitado, refrescando la tabla con el nuevo registro habilitado.
Sub Flujo 8	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 – Validación de datos proveedores) y actualiza la tabla mostrada.
Sub Flujo 9	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación de datos proveedores	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Cédula: No se encuentre vacío y debe como máximo 12 caracteres. • Nombre: No se encuentre vacío. • Teléfono1: No se encuentre vacío y debe tener 8 dígitos. • Teléfono2: Debe tener 8 dígitos. • Dirección: No se encuentre vacío. • Tipo: No se seleccione algún tipo. <p>En caso que alguno de estos se presente, se desplegará como una alerta los errores correspondientes.</p>
Post-Condiciones	
<p>En caso de que el usuario realizará una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo a los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de datos.</p> <p>En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.</p>	

En caso de haber eliminado un registro, este quedará deshabilitado dentro de la base de datos, sin eliminar totalmente el registro.

Fuente: Elaboración propia.

Tabla 11. Caso de uso flujos

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 07	Nombre del Caso de Uso: Flujos
Fecha de elaboración:	11/12/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso del manejo de flujos de ingresos y gastos de ejecución para cada uno de los proyectos dentro de un periodo de tiempo.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario y al menos un proyecto debe existir en sistema.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario ingresa a la opción Flujos del menú principal 2. El sistema carga la pantalla de flujos del primer proyecto habilitado con su título (SF1), mostrando dos tablas, una de ingresos y otra de gastos de ejecución, donde se presenta una fila azul la cual muestra totales de columna, en cada una se puede ingresar datos y poseen los siguientes botones: <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF2). b. Botón de lupa en la última columna (SF3). c. Botón de limpieza al lado de la lupa en la última columna (SF4). d. Botón de lápiz verde en la última columna de los distintos registros (SF5). e. Botón de basurero rojo en la última columna de los distintos registros (SF6). 3. El usuario selecciona una categoría y/o digita datos dentro de cada uno de los campos de alguna de las dos tablas y presiona Enter. 4. El sistema refresca la tabla donde fueron ingresados los datos y muestra los registros que coinciden únicamente con los datos ingresados. 	

5. Fin caso de uso.	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario da click al título de flujo de proyectos. 2. El sistema abre un panel superpuesto y oscurece el fondo, mostrando una lista de los diferentes proyectos. 3. El usuario selecciona el proyecto que quiere abrir y presiona “Cerrar”. 4. El sistema recarga la página con los flujos del proyecto seleccionado.
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla seleccionada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, donde la penúltima columna aparecerá una lista de categorías, con su última opción “Nueva Categoría” (SF7) y en la ultima un botón “+”, definiendo la categoría del gasto o ingreso a insertar, a su vez la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación flujo ingresos o gastos) y actualiza la tabla seleccionada con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario selecciona y/o ingresa datos en la primera fila de una de las tablas y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla seleccionada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla seleccionada, con todos ellos.

Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click a la fila de algún registro de las tablas o presiona el botón de lápiz verde en la última columna de alguna fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente pero editables, donde en la última columna cambia por los botones check verde (SF8) y una “x” roja (SF9).
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema elimina el registro de la fila seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario selecciona “Nueva Categoría” de las opciones desplegadas. 2. El sistema despliega un “prompt” que solicita el ingreso de una nueva categoría. 3. El usuario ingresa el nombre de la nueva categoría y presiona aceptar. 4. El sistema valida (FA3 – Validación categoría flujo ingreso o gasto), guarda el registro y refresca la lista mostrada.
Sub Flujo 8	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 - Validación flujo ingresos o gastos) y actualiza la tabla seleccionada.
Sub Flujo 9	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación flujo ingresos o gastos	<p>El sistema validará los campos a editar o insertar, de la tabla de ingresos y gastos ejecutivos de la siguiente forma:</p> <ul style="list-style-type: none"> • Descripción: No se encuentre vacío. • Presupuesto ¢: No se encuentre vacío y no puede ser

	<p>negativo.</p> <ul style="list-style-type: none"> • Presupuesto \$: No se encuentre vacío y no puede ser negativo. • Enero: No puede ser negativo. • Febrero: No puede ser negativo. • Marzo: No puede ser negativo. • Abril: No puede ser negativo. • Mayo: No puede ser negativo. • Junio: No puede ser negativo. • Julio: No puede ser negativo. • Agosto: No puede ser negativo. • Septiembre: No puede ser negativo. • Octubre: No puede ser negativo. • Noviembre: No puede ser negativo. • Diciembre: No puede ser negativo. <p>En caso que alguno de estos se presente, se desplegará como una alerta los errores correspondientes.</p>
<p>Flujo Alterno No. 3 Validación categoría flujo ingreso o gasto</p>	<p>El sistema validará el campo de la categoría de flujo de ingresos o gastos ejecución, de la siguiente forma:</p> <ul style="list-style-type: none"> • Nombre: No se encuentre vacío y tampoco repetido. <p>En caso que se cumpla esta condición desplegará el mensaje de error en forma de una alerta.</p>
<p>Post-Condiciones</p>	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo en los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de datos.</p> <p>En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.</p> <p>En caso de haber eliminado un registro, este quedará eliminado totalmente dentro de la base de datos.</p>	

En caso de haber ingresado una nueva categoría, esta se guardará dentro de la base de datos.

Fuente: Elaboración propia.

Tabla 12. Caso de uso cuentas por cobrar

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 08	Nombre del Caso de Uso: Cuentas por Cobrar
Fecha de elaboración:	11/12/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso de consultar, registrar, editar y eliminar las diferentes cuentas por cobrar que se realizan en la gestión de proyectos de la Cooperativa Sulá Batsú.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Cuentas por Cobrar. 2. El sistema redirige a la asociada a la pantalla de cuentas por cobrar. 3. El usuario ve la pantalla de mantenimiento con una tabla que muestra los registros de la entidad y diferentes botones en la misma. <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF1). b. Botón de lupa en la última columna (SF2). c. Botón de limpieza al lado de la lupa en la última columna (SF3). d. Botón de lápiz verde en la última columna de los distintos registros (SF4). e. Botón de basurero rojo en la última columna de los distintos registros (SF5). 4. El usuario digita los datos y/o selecciona un cliente en la primera fila y presiona Enter 5. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario. 6. Fin del caso de uso. 	

Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos y seleccionar un cliente, a su vez la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación datos cuentas por cobrar) y actualiza la tabla con los datos ingresados.
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario ingresa datos y/o selecciona un cliente en la primera fila y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados o el cliente seleccionado.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables, a su vez en la columna clientes muestra una lista de los que se encuentran registrados en el prototipo y la última columna cambia por los botones check verde (SF6) y una “x” roja (SF7).
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema deshabilita el registro de la fila seleccionada.
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna.

	2. El sistema modifica los datos (FA1- Validación datos cuentas por cobrar) y actualiza la tabla mostrada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación datos cuentas por cobrar	<p>El sistema validará los campos a editar o insertar, de la siguiente forma:</p> <ul style="list-style-type: none"> • Fecha de emisión: No se encuentre vacío. • Cliente: No se seleccione un cliente. • Tipo: No se encuentre vacío. • Descripción: No se encuentre vacío. • Vencimiento: No se encuentre vacío y no puede ser anterior a la fecha de emisión. • Monto: No se encuentre vacío y no puede ser negativo • Estado: No se seleccione un estado. <p>En caso que se cumpla esta condición desplegará el mensaje de error en forma de una alerta.</p>
Post-Condiciones	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de datos.</p> <p>En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.</p> <p>En caso de haber eliminado un registro, éste quedará eliminado totalmente dentro de la base de datos.</p>	

Fuente: Elaboración propia.

Tabla 13. Caso de uso cuentas por pagar

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 09	Nombre del Caso de Uso: Cuentas por Pagar
Fecha de elaboración:	10/12/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso de consultar, registrar, editar y eliminar las diferentes cuentas por pagar que se realizan en la gestión de proyectos de la Cooperativa Sulá Batsú.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario.
Flujo Básico del caso de uso	
<p>6. El usuario presiona en el menú principal la opción Cuentas por Pagar.</p> <p>7. El sistema redirige a la asociada a la pantalla de cuentas por pagar.</p> <p>8. El usuario ve la pantalla de mantenimiento con una tabla cargando</p> <p>9. El sistema muestra los registros de la entidad dentro la tabla y diferentes botones en la misma.</p> <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF1). b. Botón de lupa en la última columna (SF2). c. Botón de limpieza al lado de la lupa en la última columna (SF3). d. Botón de lápiz verde en la última columna de los distintos registros (SF4). e. Botón de basurero rojo en la última columna de los distintos registros (SF5). <p>10. El usuario digita los datos y/o selecciona un cliente en la primera fila y presiona Enter</p> <p>11. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario.</p> <p>7. Fin del caso de uso.</p>	
Sub Flujos	
Sub Flujo 1	<p>5. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada.</p> <p>6. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos y seleccionar un</p>

	<p>proveedor, a su vez la última columna se remplazan los botones por un botón con el símbolo “+” verde en la última columna.</p> <p>7. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna.</p> <p>8. El sistema ingresa datos (FA1 - Validación datos cuentas por pagar) y actualiza la tabla con los datos ingresados.</p>
Sub Flujo 2	<p>3. El usuario ingresa datos y/o selecciona un cliente en la primera fila y presiona Enter o da click al botón de la lupa.</p> <p>4. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados o el cliente seleccionado.</p>
Sub Flujo 3	<p>3. El usuario da click el botón de limpieza.</p> <p>4. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.</p>
Sub Flujo 4	<p>3. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila.</p> <p>4. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables, a su vez en la columna proveedor muestra una lista de los que se encuentran registrados en el prototipo y la última columna cambia por los botones check verde (SF6) y una “x” roja (SF7).</p>
Sub Flujo 5	<p>3. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada.</p> <p>4. El sistema deshabilita el registro de la fila seleccionada.</p>
Sub Flujo 6	<p>3. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna.</p> <p>4. El sistema modifica los datos (FA1 - Validación datos cuentas por pagar) y actualiza la tabla mostrada.</p>
Sub Flujo 7	<p>3. El usuario da click al botón de “x” roja en la última columna de la fila.</p> <p>4. El sistema revertirá la fila a su estado anterior ocultando los</p>

	campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación datos cuentas por cobrar	<p>El sistema validará los campos a editar o insertar, de la siguiente forma:</p> <ul style="list-style-type: none"> • Fecha de emisión: No se encuentre vacío. • Proveedor: No se seleccione un proveedor. • Tipo: No se encuentre vacío. • Descripción: No se encuentre vacío. • Vencimiento: No se encuentre vacío y no puede ser anterior a la fecha de emisión. • Monto: No se encuentre vacío y no puede ser negativo • Estado: No se seleccione un estado. <p>En caso que se cumpla esta condición desplegará el mensaje de error en forma de una alerta.</p>
Post-Condiciones	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de datos.</p> <p>En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.</p> <p>En caso de haber eliminado un registro, éste quedará eliminado totalmente dentro de la base de datos.</p>	

Fuente: Elaboración propia.

Tabla 14. Caso de uso gastos operativos

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 10	Nombre del Caso de Uso: Gastos Operativos
Fecha de elaboración:	11/12/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso de consultar,

	registrar, editar y eliminar los diferentes gastos operativos que se realizan en los diferentes proyectos.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario y debe existir al menos un proyecto en el sistema.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Gastos Operativos. 2. El sistema redirige a la asociada a la pantalla de gastos operativos. 3. El usuario ve la pantalla de mantenimiento con una tabla que muestra los registros de la entidad y diferentes botones en la misma. <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF1). b. Botón de lupa en la última columna (SF2). c. Botón de limpieza al lado de la lupa en la última columna (SF3). d. Botón de lápiz verde en la última columna de los distintos registros (SF4). e. Botón de basurero rojo en la última columna de los distintos registros (SF5). 4. El usuario digita los datos y/o selecciona un cliente en la primera fila y presiona Enter 5. El sistema consultará los datos en la base de datos y mostrará solamente aquellos que coincidan con los ingresados por el usuario. 6. Fin del caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla desplegada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, donde la penúltima columna aparecerá una lista de categorías, con su última opción “Nueva Categoría” (SF6) y a su vez en la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+”

	<p>verde en la última columna.</p> <p>4. El sistema ingresa datos (FA1 – Validación datos gastos operativos) y actualiza la tabla con los datos ingresados.</p>
Sub Flujo 2	<p>1. El usuario ingresa datos y/o selecciona una categoría en la primera fila y presiona Enter o da click al botón de la lupa.</p> <p>2. El sistema consulta con la base de datos y muestra los datos que coincidan con los datos ingresados o la categoría seleccionada.</p>
Sub Flujo 3	<p>1. El usuario da click el botón de limpieza.</p> <p>2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla mostrada con todos ellos.</p>
Sub Flujo 4	<p>1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila.</p> <p>2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente, pero editables, donde la primera columna tendrá una lista para seleccionar una categoría o un botón “+” para agregar alguna y a su vez en la columna proveedor muestra una lista de los que se encuentran registrados en el prototipo y la última columna cambia por los botones check verde (SF7) y una “x” roja (SF8).</p>
Sub Flujo 5	<p>1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada.</p> <p>2. El sistema elimina el registro de la fila seleccionada.</p>
Sub Flujo 6	<p>1. El usuario selecciona “Nueva Categoría” de las opciones desplegadas.</p> <p>2. El sistema despliega un prompt que solicita el ingreso de una nueva categoría.</p> <p>3. El usuario ingresa el nombre de la nueva categoría y presiona aceptar.</p> <p>4. El sistema valida (FA2 – Validación categoría gastos operativos), guarda el registro y refresca la lista mostrada.</p>

Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 - Validación datos gastos operativos) y actualiza la tabla mostrada.
Sub Flujo 8	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación datos gastos operativos	<p>El sistema validará los campos a editar o insertar, de la siguiente forma:</p> <ul style="list-style-type: none"> • Actividad: No se encuentre vacío. • Total: No puede ser negativo. • Enero: No puede ser negativo. • Febrero: No puede ser negativo. • Marzo: No puede ser negativo. • Abril: No puede ser negativo. • Mayo: No puede ser negativo. • Junio: No puede ser negativo. • Julio: No puede ser negativo. • Agosto: No puede ser negativo. • Septiembre: No puede ser negativo. • Octubre: No puede ser negativo. • Noviembre: No puede ser negativo. • Diciembre: No puede ser negativo. <p>En caso que se cumpla esta condición desplegará el mensaje de error en forma de una alerta.</p>
Flujo Alternativo No. 2 Validación categoría gastos operativos	<p>El sistema validará el campo de la categoría de gastos operativos, de la siguiente forma:</p> <ul style="list-style-type: none"> • Nombre: No se encuentre vacío y tampoco repetido. <p>En caso que se cumpla esta condición desplegará el mensaje de error</p>

	en forma de una alerta.
Post-Condiciones	
En caso de que el usuario realizara una consulta el sistema refresca la tabla para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.	
En caso de haber realizado una inserción, el registro queda almacenado dentro de la base de datos.	
En caso de haber realizado una modificación, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.	
En caso de haber eliminado un registro, éste quedará eliminado totalmente dentro de la base de datos.	
En caso de haber ingresado una nueva categoría para los gastos operativos esta quedará guardada en la base de datos	

Fuente: Elaboración Propia.

Tabla 15. Caso de uso historial de transferencias

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 11	Nombre del Caso de Uso: Historial de transferencias
Fecha de elaboración:	11/12/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso crear, editar y eliminar el historial de transferencias dentro del sistema
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario dentro del prototipo funcional y al menos un proyecto y asociado o colaborador registrados dentro del sistema.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción de Excedentes y luego en Historial de Transferencias 2. El sistema redirige al usuario a la pantalla de historial. 3. El usuario ve una tabla cargando. 	

<ol style="list-style-type: none"> 4. El sistema muestra la tabla con los distintos registros que se encuentran almacenados de usuarios, la cual tiene lo siguientes botones: <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF1). b. Botón de lupa en la última columna (SF2). c. Botón de limpieza al lado de la lupa en la última columna (SF3). d. Botón de lápiz verde en la última columna de los distintos registros (SF4). e. Botón de basurero rojo en la última columna de los distintos registros (SF5). 5. El usuario selecciona un proyecto o una persona. 6. El sistema automáticamente cambia la lista de cuenta de envió y cuenta recibe por las cuentas en colones y dólares del proyecto o persona seleccionada. 7. El usuario digita otros datos dentro de cada uno de los campos y presiona Enter 8. El sistema refresca la tabla y muestra los registros que coinciden únicamente con los datos ingresados o seleccionados. 9. Fin caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla seleccionada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, en la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario selecciona un proyecto o una persona. 4. El sistema automáticamente cambia la lista de cuenta de envió y cuenta recibe por las cuentas en colones y dólares del proyecto o persona seleccionada. 5. El usuario ingresa los demás datos y da click al botón de símbolo “+” verde en la última columna. 6. El sistema ingresa datos (FA1 – Validación Historial) y actualiza la tabla seleccionada con los datos ingresados.
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario selecciona y/o ingresa datos en la primera fila de una de las tablas y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los registros

	que coincidan con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla seleccionada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla seleccionada, con todos ellos.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click a la fila de algún registro de las tablas o presiona el botón de lápiz verde en la última columna de alguna fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente pero editables, donde en la última columna cambia por los botones check verde (SF6) y una “x” roja (SF7).
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema elimina el registro de la fila seleccionada.
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 – Validación Historial) y actualiza la tabla seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del cambio a la fila.
Flujos Alternos	
Flujo Alternativo No. 1 Validación Historial	<p>El sistema validara los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Proyecto: No se ha seleccionado un proyecto. • Persona: No se ha seleccionado una persona. • Fecha: No se ingresado una fecha valida o no se cumple con el formato d/m/y. • N° Cuenta Envio: No se ha seleccionado un proyecto.

	<ul style="list-style-type: none"> • N° Cuenta Recibe: No se ha seleccionado una persona. • Monto: No se ha ingresado un valor, no se han ingresado solamente números. • Descripción: No se ha ingresado una descripción. <p>En caso de que alguna de estas se cumplan, se desplegará en forma de alerta el mensaje mencionado.</p>
Post-Condiciones	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla seleccionada para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción en cualquier de las tablas, el registro queda almacenado dentro de la base de datos.</p> <p>En caso de haber realizado una modificación en cualquiera de las tablas, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.</p> <p>En caso de haber eliminado un registro en cualquiera de las tablas, este quedará eliminado totalmente dentro de la base de datos.</p>	

Fuente: Elaboración propia.

Tabla 16. Caso de uso asignar proyectos

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 12	Nombre del Caso de Uso: Asignar Proyectos
Fecha de elaboración:	11/12/2018
Descripción Caso de Uso:	En este caso de uso se define el proceso asignar asociados, asociadas, colaboradores y colaboradoras a los distintos proyectos registrados.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario, debe existir al menos un proyecto en el sistema y debe haber al menos un asociado, asociada, colaborador o colaboradora registrada.

Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción de Excedentes y luego en Asignaciones 2. El sistema redirige al usuario a la pantalla de asignaciones. 3. El usuario ve cargando dos tablas. 4. El sistema muestra una de asociados y asociadas a proyectos, donde se pueden ingresar datos y posee los siguientes botones: <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF1). b. Botón de lupa en la última columna (SF2). c. Botón de limpieza al lado de la lupa en la última columna (SF3). d. Botón de basurero rojo en la última columna de los distintos registros (SF4). <p>Y otra de colaboradores y colaboradoras a proyecto, de también se puede ingresar datos y poseen botones similares:</p> <ol style="list-style-type: none"> e. Botón “+” verde en la última columna al lado de los encabezados (SF1). f. Botón de lupa en la última columna (SF2). g. Botón de limpieza al lado de la lupa en la última columna (SF3). h. Botón de lápiz verde en la última columna de los distintos registros (SF5). i. Botón de basurero rojo en la última columna de los distintos registros (SF4). 5. El usuario selecciona un dato y/o digita datos dentro de cada uno de los campos de alguna de las dos tablas y presiona Enter. 6. El sistema refresca la tabla donde fueron ingresados los datos y muestra los registros que coinciden únicamente con los datos ingresados. 7. Fin caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla seleccionada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, en la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna.

	<p>4. El sistema ingresa los datos (FA1 – Validación Asignar Asociado en caso de la tabla Asociados - Proyecto o FA2 – Validación Asignar Colaborador en caso de la tabla Colaboradores - Proyecto), crea los excedentes correspondientes y actualiza la tabla seleccionada con los datos ingresados.</p>
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario selecciona y/o ingresa datos en la primera fila de una de las tablas y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los registros que coincidan con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla seleccionada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla seleccionada, con todos ellos.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema muestra un mensaje para confirmar su eliminación 3. El usuario procede a eliminar, sí presiona cancelar se termina el sub flujo. 4. El sistema elimina los excedentes correspondientes y el registro de la fila seleccionada, actualizando la tabla con los registros restantes.
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click a la fila de algún registro de la tabla Colaborador - Proyecto o presiona el botón de lápiz verde en la última columna de alguna fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente pero editables, donde en la última columna cambia por los botones check verde (SF6) y una “x” roja (SF7).
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario modifica los campos editables requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA2 – Validación Asignar Colaborador en caso de la tabla Colaboradores - Proyecto),

	modifica, inserta o elimina los excedentes correspondientes y actualiza la tabla seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del cambio a la fila.
Flujos Alternos	
Flujo Alternativo No. 1 Validación Asignar Asociado	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Asociado/a: No se ha seleccionado un asociado o asociada. • Proyecto: No se ha seleccionado un proyecto. <p>En caso de que alguna de éstas se cumplan, se desplegará en forma de alerta el mensaje mencionado.</p>
Flujo Alternativo No. 2 Validación Asignar Colaborador	<p>El sistema validará los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Colaborador/a: No se ha seleccionado un colaborador o colaboradora. • Proyecto: No se ha seleccionado un proyecto. • Jornada: No se ha seleccionado un tipo de jornada. • Monto Jornada: No se encuentre vacío y solo puede ingresar números. • Fecha Inicio: No puede estar vacío y el formato debe ser dd/mm/YYYY. • Fecha Final: No puede estar vacío, el formato debe ser dd/mm/YYYY, debe ser posterior a la fecha de inicio y <p>En caso que se cumpla esta condición desplegará el mensaje de error en forma de una alerta.</p>
Post-Condiciones	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla seleccionada para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción en cualquier de las tablas, el registro queda almacenado dentro de la base de datos junto con el o los excedentes correspondientes.</p>	

En caso de haber realizado una modificación en cualquiera de las tablas, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos y se actualizará, insertará o eliminará el o los excedentes correspondientes.

En caso de haber eliminado un registro en cualquiera de las tablas, se eliminarán los excedentes correspondientes y el registro quedará eliminado totalmente dentro de la base de datos.

Fuente: Elaboración Propia.

Tabla 17. Caso de uso excedentes

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 13	Nombre del Caso de Uso: Excedentes
Fecha de elaboración:	11/12/2018
Descripción Caso de Uso:	En este caso de uso se describe el proceso de consulta, modificación o eliminación de los excedentes de los diferentes asociados, asociadas, colaboradores y colaboradoras que hayan sido asignados a un proyecto
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario y debe haber al menos un asociado, asociada, colaborador o colaboradora asignado a algún proyecto.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción Excedentes. 2. El sistema redirige al usuario a la pantalla de Mes y Anual. 3. El usuario ve en la pantalla una tabla cargando. 4. El sistema muestra una tabla con todos los registros (SF1) de repartición de excedente anual y mensual de los distintos asociados, asociadas, colaboradores y colaboradoras, donde la columna anual puede tener los siguientes colores: <ul style="list-style-type: none"> • Rojo: Indica que el valor mensual total es mayor al anual. • Verde: Indica que el valor mensual total es igual al anual. 	

<ul style="list-style-type: none"> • Azul: Indica que el valor mensual total es menor al anual. <ol style="list-style-type: none"> 5. El usuario selecciona y/o ingresa los datos que requiera buscar. 6. El sistema muestra dentro de la tabla solo el o los registros del o los datos seleccionado y/o ingresado por el usuario. 7. Fin del caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario da click a una de las filas de los registros. 2. El sistema despliega un panel con una tabla que muestra los distintos registros del excedente en ese periodo, con el fondo oscurecido, los mismos colores de la columna anual y los siguientes botones: <ol style="list-style-type: none"> a. Botón de lupa en la última columna (SF2). b. Botón de limpieza al lado de la lupa en la última columna (SF3). c. Botón de lápiz verde en la última columna de los distintos registros (SF4). d. Botón de basurero rojo en la última columna de los distintos registros (SF5). 3. El usuario da click a “Cerrar”. 4. El panel se cierra y la tabla se refresca.
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario selecciona un proyecto y/o ingresa datos en la primera fila de la tabla y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los datos en la tabla que coincidan con los datos seleccionados o ingresados,
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla mostrada, realiza un consulta general de todos los registros de la entidad para mostrarlos en la tabla.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click a la fila de un registro de la tabla o presiona el botón de lápiz verde en la última columna de una fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo y

	diferentes campos editables para modificar los datos previos, además en la última columna se remplazan los botones por un check verde (SF6) y una “x” roja (SF7).
Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema muestra un mensaje para confirmar su eliminación 3. El usuario procede a eliminar, si presiona cancelar se termina el sub flujo. 4. El sistema elimina el excedente correspondiente.
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario da click al check verde. 2. El sistema realiza la modificación de los datos (FA1 – Validación excedentes) y refresca la tabla con los cambio del registro seleccionado.
Sub Flujo 7	<ol style="list-style-type: none"> 3. El usuario da click al botón de “x” roja en la última columna de la fila. 1. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del registro.
Flujos Alternos	
Flujo Alternativo No. 1 Validación Excedentes	<p>El sistema validara los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Anual: No puede estar vacío o ser negativo. • Total: No puede estar vacío o ser negativo. • Enero: No puede estar vacío o ser negativo. • Febrero: No puede estar vacío o ser negativo. • Marzo: No puede estar vacío o ser negativo. • Abril: No puede estar vacío o ser negativo. • Mayo: No puede estar vacío o ser negativo. • Junio: No puede estar vacío o ser negativo. • Julio: No puede estar vacío o ser negativo. • Agosto: No puede estar vacío o ser negativo. • Septiembre: No puede estar vacío o ser negativo. • Octubre: No puede estar vacío o ser negativo.

	<ul style="list-style-type: none"> • Noviembre: No puede estar vacío o ser negativo. • Diciembre: No puede estar vacío o ser negativo. <p>En caso de que alguna de estas se cumplan, se desplegará en forma de alerta el mensaje mencionado.</p>
Post-Condiciones	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla principal o la del panel para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una modificación en la tabla del panel, el registro se actualizará con los nuevos valores y quedará almacenado dentro de la base de datos.</p> <p>En caso de haber eliminado un registro en la tabla del panel, éste quedará eliminado totalmente dentro de la base de datos.</p>	

Fuente: Elaboración Propia.

Tabla 18. Caso de uso usuarios

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 14	Nombre del Caso de Uso: Usuarios
Fecha de elaboración:	19/2/2019
Descripción Caso de Uso:	En este caso de uso se define el proceso crear, editar y eliminar usuarios dentro del sistema
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario dentro del prototipo funcional.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción de Configuraciones y luego en Usuarios 2. El sistema redirige al usuario a la pantalla de usuarios. 3. El usuario ve una tabla cargando. 4. El sistema muestra la tabla con los distintos registros que se encuentran 	

<p>almacenados de usuarios, la cual tiene lo siguientes botones:</p> <ol style="list-style-type: none"> Botón “+” verde en la última columna al lado de los encabezados (SF1). Botón de lupa en la última columna (SF2). Botón de limpieza al lado de la lupa en la última columna (SF3). Botón de lápiz verde en la última columna de los distintos registros (SF4). Botón de basurero rojo en la última columna de los distintos registros (SF5). <ol style="list-style-type: none"> El usuario selecciona una categoría y/o digita datos dentro de cada uno de los campos de la tabla y presiona Enter. El sistema refresca la tabla con los datos ingresados y muestra los registros que coinciden únicamente con los datos ingresados. Fin caso de uso. 	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> El usuario presiona el botón con símbolo “+” verde, en la tabla seleccionada. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, en la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. El sistema ingresa datos (FA1 – Validación Usuario) y actualiza la tabla seleccionada con los datos ingresados.
Sub Flujo 2	<ol style="list-style-type: none"> El usuario selecciona y/o ingresa datos en la primera fila de una de las tablas y presiona Enter o da click al botón de la lupa. El sistema consulta con la base de datos y muestra los registros que coincidan con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> El usuario da click el botón de limpieza. El sistema limpia los campos de la primera fila de la tabla seleccionada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla seleccionada, con todos ellos.
Sub Flujo 4	<ol style="list-style-type: none"> El usuario da click a la fila de algún registro de las tablas o presiona el botón de lápiz verde en la última columna de alguna

	<p>fila.</p> <p>2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente pero editables, donde en la última columna cambia por los botones check verde (SF6) y una “x” roja (SF7).</p>
Sub Flujo 5	<p>1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada.</p> <p>2. El sistema muestra un mensaje para confirmar su eliminación</p> <p>3. El usuario procede a eliminar, si presiona cancelar se termina el sub flujo.</p> <p>4. El sistema elimina el registro de la fila seleccionada.</p>
Sub Flujo 6	<p>1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna.</p> <p>2. El sistema modifica los datos (FA1 – Validación Usuario) y actualiza la tabla seleccionada.</p>
Sub Flujo 7	<p>1. El usuario da click al botón de “x” roja en la última columna de la fila.</p> <p>2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del cambio a la fila.</p>
Flujos Alternos	
Flujo Alternativo No. 1 Validación Usuario	<p>El sistema validara los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Usuario: No se ha ingresado un usuario válido o no cumple con el formato ejemplo@ejemplo.com. • Contraseña: No se ha ingresado una contraseña, no debe tener caracteres especiales. • Rol: No se ha seleccionado un rol. <p>En caso de que alguna de estas se cumplan, se desplegará en forma de alerta el mensaje mencionado.</p>
Post-Condiciones	
En caso de que el usuario realizara una consulta el sistema refresca la tabla seleccionada	

para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.

En caso de haber realizado una inserción en cualquier de las tablas, el registro queda almacenado dentro de la base de datos.

En caso de haber realizado una modificación en cualquiera de las tablas, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.

En caso de haber eliminado un registro en cualquiera de las tablas, éste quedará eliminado totalmente dentro de la base de datos.

Fuente: Elaboración propia.

Tabla 19. Caso de uso roles

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 15	Nombre del Caso de Uso: Roles
Fecha de elaboración:	19/2/2019
Descripción Caso de Uso:	En este caso de uso se define el proceso crear, editar y eliminar roles dentro del sistema
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario dentro del prototipo funcional.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción de Configuraciones y luego en Roles. 2. El sistema redirige al usuario a la pantalla de roles. 3. El usuario ve una tabla cargando. 4. El sistema muestra la tabla con los distintos registros que se encuentran almacenados de roles, la cual tiene lo siguientes botones: <ol style="list-style-type: none"> a. Botón “+” verde en la última columna al lado de los encabezados (SF1). b. Botón de lupa en la última columna (SF2). c. Botón de limpieza al lado de la lupa en la última columna (SF3). d. Botón de lápiz verde en la última columna de los distintos registros (SF4). 	

<p>e. Botón de basurero rojo en la última columna de los distintos registros (SF5).</p> <p>5. El usuario selecciona una categoría y/o digita datos dentro de cada uno de los campos de alguna de las dos tablas y presiona Enter.</p> <p>6. El sistema refresca la tabla donde fueron ingresados los datos y muestra los registros que coinciden únicamente con los datos ingresados.</p> <p>7. Fin caso de uso.</p>	
Sub Flujos	
Sub Flujo 1	<ol style="list-style-type: none"> 1. El usuario presiona el botón con símbolo “+” verde, en la tabla seleccionada. 2. El sistema reemplaza la primera fila mostrada para filtrar datos, por una con fondo verde para ingresar datos, en la última columna se reemplazan los botones por un botón con el símbolo “+” verde en la última columna. 3. El usuario ingresa los datos y da click al botón de símbolo “+” verde en la última columna. 4. El sistema ingresa datos (FA1 – Validación Roles) y actualiza la tabla seleccionada con los datos ingresados.
Sub Flujo 2	<ol style="list-style-type: none"> 1. El usuario selecciona y/o ingresa datos en la primera fila de una de las tablas y presiona Enter o da click al botón de la lupa. 2. El sistema consulta con la base de datos y muestra los registros que coincidan con los datos ingresados.
Sub Flujo 3	<ol style="list-style-type: none"> 1. El usuario da click el botón de limpieza. 2. El sistema limpia los campos de la primera fila de la tabla seleccionada, realiza un consulta general de todos los registros de la entidad y actualiza la tabla seleccionada, con todos ellos.
Sub Flujo 4	<ol style="list-style-type: none"> 1. El usuario da click a la fila de algún registro de las tablas o presiona el botón de lápiz verde en la última columna de alguna fila. 2. El sistema reemplaza la fila, por otra con fondo amarillo con los valores mostrados previamente pero editables, donde en la última columna cambia por los botones check verde (SF6) y una “x” roja (SF7).

Sub Flujo 5	<ol style="list-style-type: none"> 1. El usuario da click al botón con el basurero rojo en la última columna de un registro en la tabla desplegada. 2. El sistema muestra un mensaje para confirmar su eliminación 3. El usuario procede a eliminar, sí presiona cancelar se termina el sub flujo. 4. El sistema elimina el registro de la fila seleccionada.
Sub Flujo 6	<ol style="list-style-type: none"> 1. El usuario modifica los campos requeridos y da click a botón del check verde en la última columna. 2. El sistema modifica los datos (FA1 – Validación Roles) y actualiza la tabla seleccionada.
Sub Flujo 7	<ol style="list-style-type: none"> 1. El usuario da click al botón de “x” roja en la última columna de la fila. 2. El sistema revertirá la fila a su estado anterior ocultando los campos editables y mostrando los valores anteriores del cambio a la fila.
Flujos Alternos	
Flujo Alternativo No. 1 Validación Roles	<p>El sistema validara los campos de la siguiente forma:</p> <ul style="list-style-type: none"> • Descripción: No se encuentre vacío o no puede estar repetido. <p>En caso de que alguna de estas se cumplan, se desplegará en forma de alerta el mensaje mencionado.</p>
Post-Condicion	
<p>En caso de que el usuario realizara una consulta el sistema refresca la tabla seleccionada para mostrar los datos consultados de acuerdo con los datos ingresados por el usuario.</p> <p>En caso de haber realizado una inserción en cualquier de las tablas, el registro queda almacenado dentro de la base de datos.</p> <p>En caso de haber realizado una modificación en cualquiera de las tablas, el registro se actualizará con los nuevos valores quedará almacenado dentro de la base de datos.</p> <p>En caso de haber eliminado un registro en cualquiera de las tablas, éste quedará eliminado totalmente dentro de la base de datos.</p>	

Fuente: Elaboración propia.

Tabla 20. Caso de uso reportes.

Prototipo: Prototipo funcional para la gestión de proyectos de la Cooperativa Sulá Batsú ubicada en Barrio Escalante.	
Número Caso de Uso: 16	Nombre del Caso de Uso: Reportes
Fecha de elaboración:	09/03/2019
Descripción Caso de Uso:	En este caso de uso se define el proceso para generar un reporte.
Autor caso de uso:	Steven Morera Hernández
Actores relacionados:	El usuario administrador
Precondiciones:	La correcta autenticación del usuario dentro del prototipo.
Flujo Básico del caso de uso	
<ol style="list-style-type: none"> 1. El usuario presiona en el menú principal la opción de Reportes y luego en Cuentas Cobrar y Pagar. 2. El sistema redirige al usuario a la pantalla de reportes. 3. El usuario ve dos formularios de parámetros de reporte para cuentas por cobrar y para cuentas por pagar con distintos campos como: <ol style="list-style-type: none"> a. Estado de las cuentas b. ¿Incluir deshabilitados? 4. El usuario selecciona un estado y/o activa los deshabilitados, finalizando con un click al botón de “Generar PDF”. (FA1 – No hay registros) 5. El sistema redirige al usuario a una nueva ventana donde se autogenera el archivo PDF. 6. Fin caso de uso. 	
Flujos Alternos	
Flujo Alternativo No. 1 No hay registros	El sistema al solicitar la información no encuentra los datos que coincidan con los parámetros seleccionados por el usuario, por lo que envía un mensaje diciendo “Reporte Vacío, no hay registros”.
Post-Condiciones	
La generación de un archivo PDF que puede ser guardado por el usuario.	

Fuente: Elaboración propia.

Diseño

Es este apartado se encuentran todos los diseños referentes al desarrollo del prototipo funcional para la gestión de proyectos.

Arquitectura del sistema.

En esta arquitectura se compone de diferentes capas:

- Capa de servicio de hosting:

Esta capa muestra el servicio de web hosting que actualmente Sulá Batsú tienen contratado para su página web, en esta se muestran los servidores webs donde está alojado el prototipo funcional a desarrollar, y la base de datos MySQL que se emplea el almacenamiento de datos.

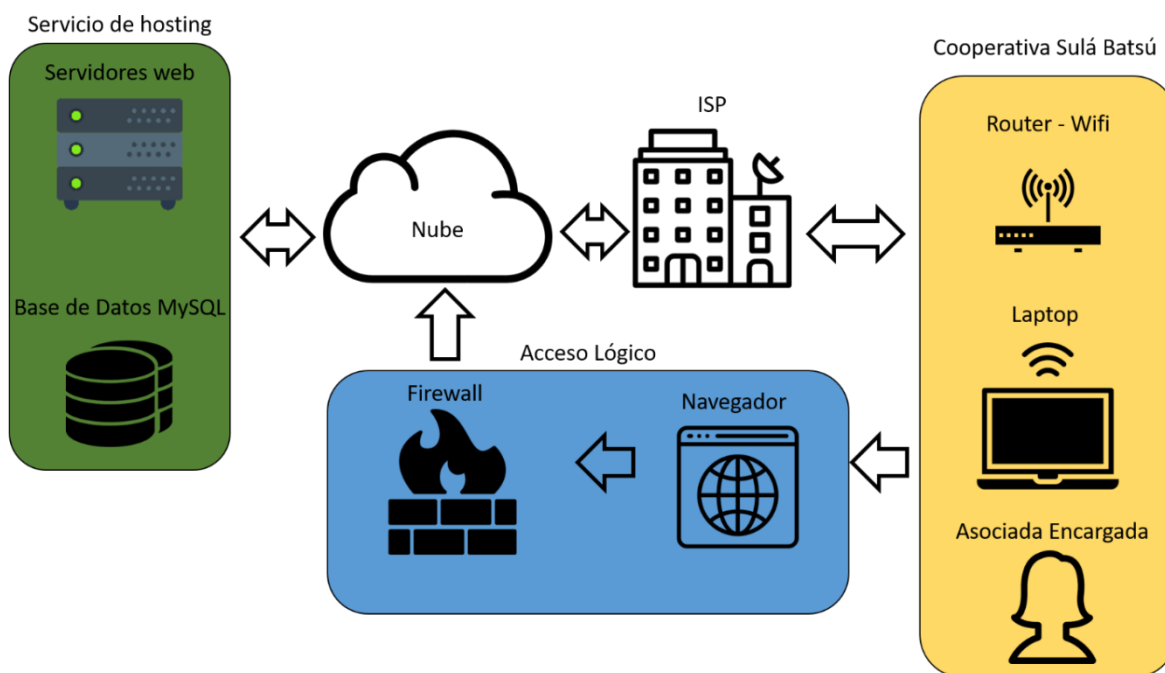
- Capa de cooperativa Sulá Batsú:

En esta capa se muestra la estructura informática que la Cooperativa Sulá Batsú, utiliza, la cual cuenta con un servicio de Internet wifi, de par trenzado, del ISP Tigo por una velocidad de 25mbps de bajada y 3mbps de subida. A su vez se muestra que el equipo de cómputo utilizado por la asociada, la cual tiene a su cargo la gestión de proyectos, el cual es una laptop con conexión inalámbrica.

- Capa de acceso lógico:

Esta capa sirve para mostrar de tal manera cómo se realiza la comunicación entre la capa de Sulá Batsú y el servicio de hosting, con el fin de ingresar y utilizar el prototipo funcional, mediante el navegador web.

Figura 16. Arquitectura de Sistema



Fuente: Elaboración propia.

Arquitectura del software.

La arquitectura de software diseñada, se muestra el recorrido lógico que se realiza desde el cliente, en este caso el equipo de cómputo utilizado por la asociada, hasta la base de datos, el cual almacena todos los datos.

El cliente utiliza el navegador web de Google Chrome, para acceder a Internet a través del cortafuegos del propio equipo, que a su vez, cruza el cortafuegos del lado del servidor web, para ingresar finalmente al prototipo funcional para la gestión de proyectos.

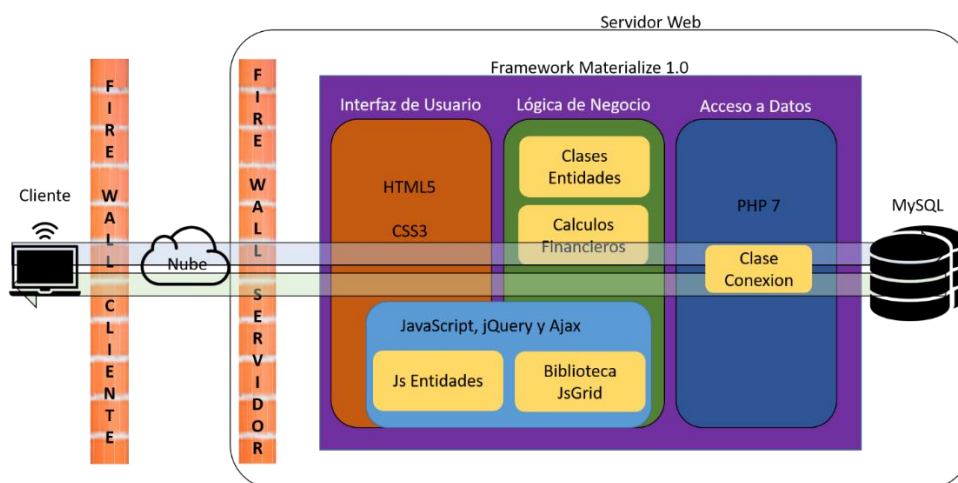
El prototipo funcional será dividido de la siguiente forma:

- Interfaz de usuario: Capa, donde se encuentra el diseño de las interfaces y el estilo de éstas, realizadas y definidas por el framework de Materialize y, a su vez, por la biblioteca de jQuery y Ajax, JsGrid.
- Lógica de negocio: Capa donde se realiza el procesamiento de los datos solicitados de la base de datos para presentarlos a la asociada y viceversa, a su vez, es donde se realiza los cálculos financieros automatizados propuestos para el funcionamiento del prototipo.

- Acceso a datos: Capa encargada de realizar la conexión de base de datos y las solicitudes de ingreso o extracción de datos con MySQL Workbench.

Todos estos se encuentran englobados por el framework seleccionado, Materialize que define el estándar utilizado para la utilización de clases y manejo de eventos en JavaScript.

Figura 17. Arquitectura de Software



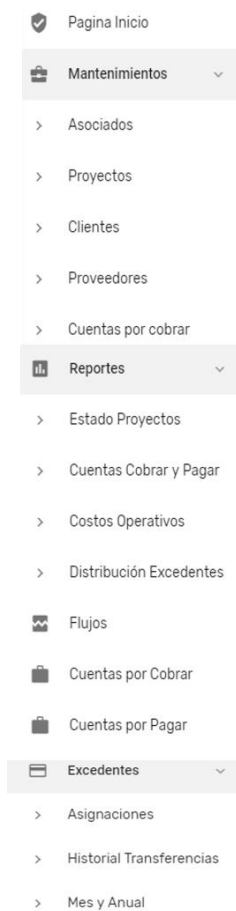
Fuente: Elaboración propia.

Diseño de interfaces.

Las interfaces son sumamente importantes a la hora del desarrollo de cualquier sistema o aplicación, ya que ésta es la que permite que los usuarios sean capaces de interactuar con éstas, es por ello que el diseño que éstas llevan deben ser simples, funcionales e intuitivas en la mayor medida de lo posible.

Las distintas interfaces diseñadas para el prototipo funcional, poseen elementos que se comparten entre ellos, como un menú en el costado izquierdo de la pantalla, que posee las diferentes pantallas del prototipo dentro de subdivisiones y una barra de superior de color naranja que dispone de una campana que indica las alertas del prototipo y una imagen de avatar para cerrar la sesión.

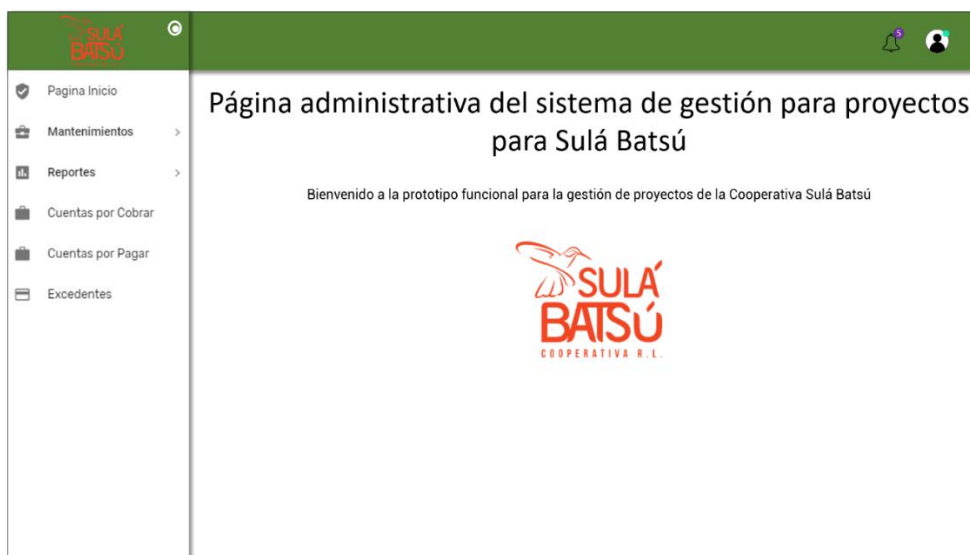
Figura 18. Menú principal



Fuente: Elaboración propia.

A continuación, se muestra la pantalla de inicio que da una bienvenida al usuario dentro del prototipo funcional:

Figura 19. Página de Inicio



Fuente: Elaboración propia.

Para el caso de los distintos mantenimientos dentro del prototipo, se define una interface simple con el título de la entidad a la cual se van a realizar los procesos de consultar, insertar, modificar y/o eliminar. Dicho proceso se realiza mediante una tabla diseñada por la biblioteca jsGrid, la cual usa Ajax, esta tabla como se puede ver a continuación utiliza diferentes filas:

- Fila de encabezado:

Son las que muestran el encabezado de cada columna que se despliegue en la tabla, estas columnas pueden ser seleccionadas para ordenar los registros (filas) en orden ascendente o descendente, a su vez, en la última columna se muestra un botón con el símbolo “+” verde, el cual al darle click despliega la siguiente fila.
- Fila de inserción:

Fila con el fondo de color verde claro, ésta permite ingresar registros dentro de la entidad en la base de datos, en la última columna se muestra el botón con el símbolo “+” verde el cual, ésta valida los datos, procede con su inserción a la base de datos y refresca la tabla para mostrar el nuevo registro.

Esta fila es invisible y solamente aparece reemplazando la fila de filtrado, cuando se da click al “+” verde de la fila de encabezado, a su vez, se

sustituye el botón “+” por uno de una lupa, para revertir el cambio y cancelar la inserción de datos.

- Fila de filtrado:

Fila predeterminada la cual se utiliza para consultar datos dentro de la tabla en sus respectivas columnas, en la última columna se muestran dos botones:

- ❖ La lupa: Encargado de procesar los campos filtrados y refresca la tabla para mostrar los registros que coincidan.
- ❖ El limpiador: Encargado de limpiar los campos de la fila de filtrado y refrescar la tabla con todos los registros de la entidad.

- Fila de registros:

Filas predeterminadas para mostrar los distintos registros de la entidad de la base de datos, en la última columna se muestran dos botones:

- ❖ El lápiz: Botón para modificar los datos del registro, al darle click muestra la fila de edición.
- ❖ El basurero: Botón encargado de la eliminación del registro, en este caso lo que en realidad hace, es una eliminación lógica del registro actualizando su estado dentro de la base de datos.

Al darle click sobre el registro la fila de edición siempre se muestra, sin importar si se dio click al botón del lápiz de la fila de registro correspondiente.

- Fila de edición:

Fila que reemplaza la fila del registro respectivo y sustituye sus campos por unos con fondo amarillo claro y la opción de editar sus valores, en la última columna sus botones también serán reemplazados por los siguientes:

- ❖ Check: Encargado de validar los datos modificados, realizar la modificación dentro de la base de datos y refrescar la tabla con los nuevos datos del registro.
- ❖ Equis: Cancelar la edición del registro y revertir el cambio de la fila a la fila de registro predeterminada.

Figura 20. Interfaz de mantenimiento

[Column1]	[Column2]	[Column3]	[Column4]	[Column5]	[Column6]	[Column7]	[Column8]	[Column9]	
_____	_____	_____	_____	_____	_____	_____	_____	_____	+
_____	_____	_____	_____	_____	_____	_____	_____	_____	🔍
[Valor1]	[Valor2]	[Valor3]	[Valor4]	[Valor5]	[Valor6]	[Valor7]	[Valor8]	[Valor9]	🗑️
[Valor1]	[Valor2]	[Valor3]	[Valor4]	[Valor5]	[Valor6]	[Valor7]	[Valor8]	[Valor9]	✓ ✖

Fuente: Elaboración propia.

En la gestión de proyectos realizada dentro de la cooperativa Sulá Batsú los proyectos poseen diferentes flujos que permiten determinar los diferentes ingresos que éste adquiere o aquellos gastos a los que éste incurre, durante un periodo determinado, ya que al registrarlos mensualmente se puede evaluar el estado de éstos. Es por ello que se estableció la siguiente interface:

Tabla 21. Interfaz flujos

Flujos/[Proyecto]													
[Proyecto]													
Ingresos												Periodo: [Año]	
Descripcion	Presupuesto €	Presupuesto \$	Total Ingresos	Supuesto	Enero	Febrero	Marzo	Abril	Mayo	Diciembre	
_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	+
_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	🔍
[Total Pres €]	[Total Pres \$]	[Total Ingresos]	[Total Supuest]	[Total Enero]	[Total Febrero]	[Total Marzo]	[Total Abril]	[Total Mayo]	[.....]	[.....]	[Total Diciembre]		
[Valor1]	[Valor2]	[Valor3]	[Valor4]	[Valor5]	[Valor6]	[Valor7]	[Valor8]	[Valor9]	[Valor 10]	[.....]	[.....]	[Valor16]	🗑️
[Valor1]	[Valor2]	[Valor3]	[Valor4]	[Valor5]	[Valor6]	[Valor7]	[Valor8]	[Valor9]	[Valor 10]	[.....]	[.....]	[Valor16]	✓ ✖

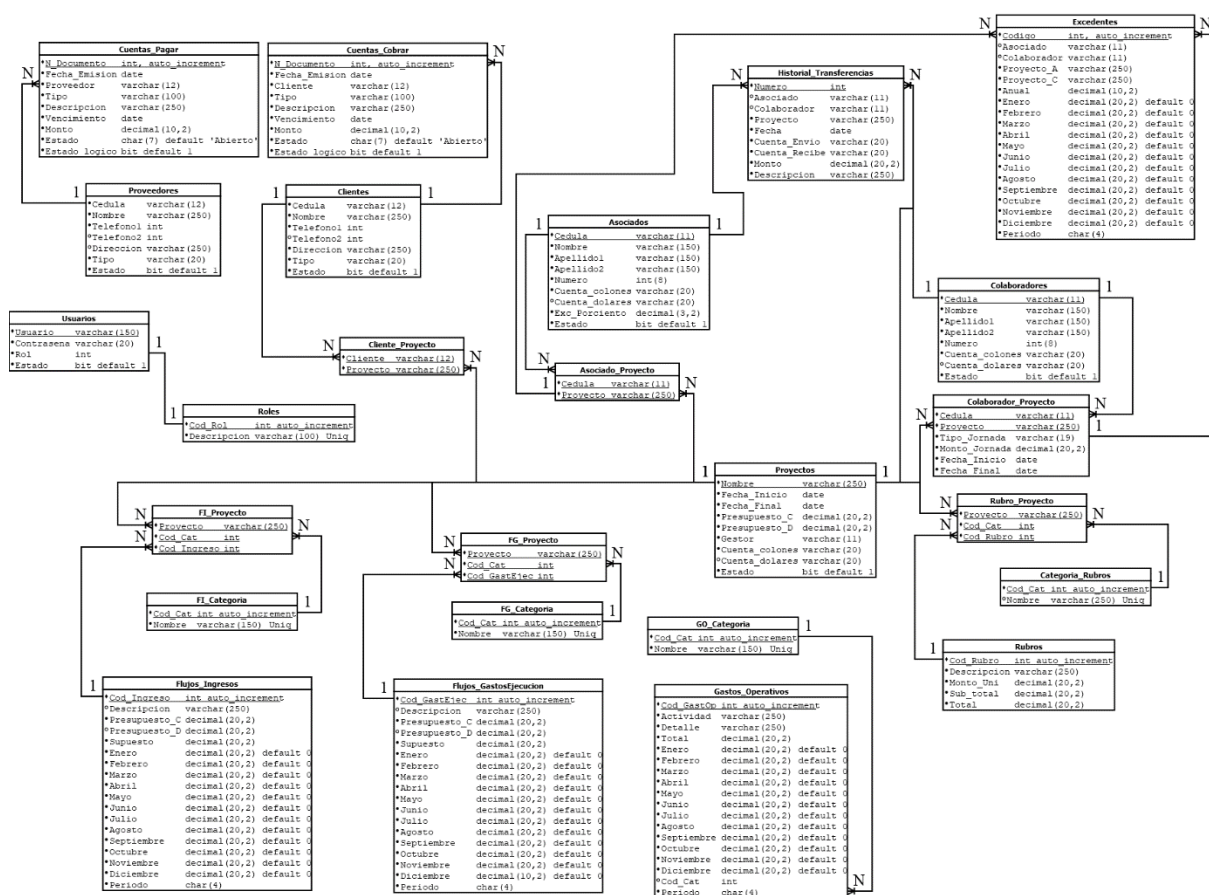
Gastos de Ejecución													
Descripcion	Presupuesto €	Presupuesto \$	Total Gastos	Supuesto	Enero	Febrero	Marzo	Abril	Mayo	Diciembre	
_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	+
_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	_____	🔍
[Total Pres €]	[Total Pres \$]	[Total Gast]	[Total Supuest]	[Total Enero]	[Total Febrero]	[Total Marzo]	[Total Abril]	[Total Mayo]	[.....]	[.....]	[Total Diciembre]		
[Valor1]	[Valor2]	[Valor3]	[Valor4]	[Valor5]	[Valor6]	[Valor7]	[Valor8]	[Valor9]	[Valor 10]	[.....]	[.....]	[Valor16]	🗑️
[Valor1]	[Valor2]	[Valor3]	[Valor4]	[Valor5]	[Valor6]	[Valor7]	[Valor8]	[Valor9]	[Valor 10]	[.....]	[.....]	[Valor16]	✓ ✖

Fuente: Elaboración propia.

Diseño de base de datos.

Para la creación de la base de datos del prototipo funcional desarrollado, se utiliza el motor de base de datos InnoDB y el sistema gestor de base de datos MySQL Workbench 8.0 CE, por lo cual las diferentes entidades de la base de datos poseen columnas con tipos de datos referentes al mismo.

Figura 21. Diagrama Entidad – Relación 1



Fuente: Elaboración propia.

Como se puede ver en el diagrama anterior se definieron 24 diferentes entidades las cuales se encargarán de registrar todos los datos referentes a la gestión de proyectos en el prototipo funcional de esta investigación.

Diccionario de base de datos.

En este apartado se muestran los diferentes campos que poseen las entidades mostradas en el diagrama de entidad – relación, donde cada una de ellas posee un nombre, tipo de dato, longitud, nulidad, un valor por defecto si es necesario, una regla de validación realizada directamente dentro de MySQL y una descripción que define que significa dicho campo.

Usuarios.

Tabla 22. Entidad usuarios

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Usuario	Varchar	150	Not null			Llave primaria que indica el nombre del usuario.
Contrasena	Varchar	20	Not null			Contraseña del usuario.
Rol	Int		Not null			Llave foránea que indica el rol del usuario.
Estado	Bit		Not null	1		Estado lógico del usuario, sí se encuentra habilitado o deshabilitado.

Fuente: Elaboración propia.

Roles.

Tabla 23. Entidad roles

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_Rol	Int		Not null	Auto increment		Llave primaria que indica el código del rol.
Descripción	Varchar	100	Not null			Nombre del rol.

Fuente: Elaboración propia.

Asociados.

Tabla 24. Entidad asociados

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cedula	Varchar	11	Not null			Llave primaria que indica la cédula del asociado o asociada.
Nombre	Varchar	150	Not null			Nombre del asociado o asociada.
Apellido1	Varchar	150	Not null			Primer apellido del asociado o asociada.
Apellido2	Varchar	150	Not null			Segundo apellido del asociado o asociada.
Numero	Int	8	Not null			Número telefónico

						del asociado o asociada.
Cuenta_colones	Varchar	20	Not null			Cuenta en colones del asociado o asociada.
Cuenta_dolares	Varchar	20	Null			Cuenta en dólares del asociado o asociada.
Exc_porcentaje	Decimal	3,2	Not null			Porcentaje de excedente correspondiente a cada asociado o asociada.
Estado	Bit		Not null	1		Estado lógico del asociado o asociada que indica si está habilitado o deshabilitado.

Fuente: Elaboración propia.

Colaboradores

Tabla 25. Entidad colaboradores

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cedula	Varchar	11	Not null			Llave primaria que indica la cédula del colaborador o colaboradora.

Nombre	Varchar	150	Not null			Nombre del colaborador o colaboradora.
Apellido1	Varchar	150	Not null			Primer apellido del colaborador o colaboradora.
Apellido2	Varchar	150	Not null			Segundo apellido del colaborador o colaboradora.
Numero	Int	8	Not null			Número telefónico del colaborador o colaboradora.
Cuenta_colones	Varchar	20	Not null			Cuenta en colones del colaborador o colaboradora.
Cuenta_dolares	Varchar	20	Null			Cuenta en dólares del colaborador o colaboradora
Estado	Bit		Not null			Estado lógico del colaborador o colaboradora, habilitado o deshabilitado.

Fuente: Elaboración propia.

Proyectos.

Tabla 26. Entidad proyectos

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Nombre	Varchar	250	Not null			Llave primaria que indica el nombre del proyecto.
Fecha_Inicio	Date		Not null			Fecha de inicio del proyecto.
Fecha_Final	Date		Not null			Fecha de finalización del proyecto.
Presupuesto_C	Decimal	20,2	Not null			Presupuesto en colones del proyecto.
Presupuesto_D	Decimal	20,2	Null			Presupuesto en dólares del proyecto.
Gestor	Varchar	11	Not null		Unique	Llave foránea que indica el asociado o asociada con el papel de gestor del proyecto,
Cuenta_colones	Varchar	20	Not null			Cuenta en colones del proyecto.
Cuenta_dolares	Varchar	20	Null			Cuenta en dólares del proyecto
Estado	Bit		Not null	1		Estado lógico del proyecto,

						habilitado o deshabilitado.
--	--	--	--	--	--	-----------------------------

Fuente: Elaboración propia.

Rubros.

Tabla 27. Entidad rubros

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_Rubro	Int		Not null	Auto increment		Llave primaria que indica el código del rubro.
Descripcion	Varchar	250	Not null			Nombre del rubro.
Monto_Uni	Decimal	20,2	Not null	0		Monto unitario del rubro.
Sub_Total	Decimal	20,2	Not null			Sub total del rubro
Total	Decimal	20,2	Not null			Total del rubro

Fuente: Elaboración propia.

Categoría de rubros.

Tabla 28. Entidad categoría rubros

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_Cat	Int		Not null	Auto increment		Llave primaria que indica el código de la categoría de rubros.
Nombre	Varchar	250	Not null			Nombre de la categoría de

						rubro.
--	--	--	--	--	--	--------

Fuente: Elaboración propia.

Rubro Proyecto.

Tabla 29. Entidad Rubro Proyecto

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Proyecto	Varchar	250	Not null			Llave primaria que indica a qué proyecto está ligado el rubro.
Cod_Cat	Int		Not null	Auto increment		Llave primaria que indica el código de la categoría de rubros.
Cod_Rubro	Int		Not null			Llave primaria que indica el código del rubro.

Fuente: Elaboración propia.

Excedentes.

Tabla 30. Entidad excedentes

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Codigo	Int		Not null	Auto increment		Llave primaria que indica el código de excedente

Asociado	Varchar	11	Null			Llave foránea que indica a qué Asociado de asignación de Asociado Proyecto corresponde el excedente..
Colaborador	Varchar	11	Null			Llave foránea que indica a qué Colaborador de asignación de Colaborador Proyecto corresponde el excedente.
Proyecto_A	Varchar	250	Null			Llave foránea que indica a qué proyecto de asignación de Asociado Proyecto corresponde el excedente.
Proyecto_C	Varchar	250	Null			Llave foránea que indica a qué proyecto de asignación de Colaborador Proyecto corresponde el excedente.

Anual	Decimal	20,2	Not null			Excedente anual que le corresponde.
Enero	Decimal	20,2	Not null	0		Excedente del mes de Enero.
Febrero	Decimal	20,2	Not null	0		Excedente del mes de Febrero.
Marzo	Decimal	20,2	Not null	0		Excedente del mes de Marzo.
Abril	Decimal	20,2	Not null	0		Excedente del mes de Abril.
Mayo	Decimal	20,2	Not null	0		Excedente del mes de Mayo.
Junio	Decimal	20,2	Not null	0		Excedente del mes de Junio.
Julio	Decimal	20,2	Not null	0		Excedente del mes de Julio.
Agosto	Decimal	20,2	Not null	0		Excedente del mes de Agosto.
Septiembre	Decimal	20,2	Not null	0		Excedente del mes de Septiembre
Octubre	Decimal	20,2	Not null	0		Excedente del mes de Octubre.
Noviembre	Decimal	20,2	Not null	0		Excedente del mes de Noviembre.
Diciembre	Decimal	20,2	Not null	0		Excedente del mes de Diciembre.
Periodo	Char	4	Not null			Periodo (año) de los excedentes.

Fuente: Elaboración propia.

Historial de transferencias.

Tabla 31. Entidad historial transferencias

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Numero	Int			Auto_increment		Llave primaria que indica el número de la transferencia
Asociado	Varchar	11	Null			Llave foránea que indica el asociado.
Colaborador	Varchar	11	Null			Llave foránea que indica el colaborador
Proyecto	Varchar	250	Not null			Llave foránea que indica el nombre del proyecto.
Fecha	Date		Not null			Fecha cuando se realizó la transferencia.
Cuenta_Envio	Varchar	20	Not null			Cuenta que envió el monto transferido.
Cuenta_Recibe	Varchar	20	Not null			Cuenta que recibió el monto transferido.
Monto	Decimal	20,2	Not null			Monto transferido.
Descripcion	Varchar	250	Not null			Detalles de la

						transferencia.
--	--	--	--	--	--	----------------

Fuente: Elaboración propia.

Asociado proyecto.

Tabla 32. Entidad asociado proyecto

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cedula	Varchar	11	Not null			Llave primaria y foránea que indica la cédula del asociado o asociada.
Proyecto	Varchar	250	Not null			Llave primaria y foránea que indica el nombre del proyecto.

Fuente: Elaboración propia.

Colaborador proyecto.

Tabla 33. Entidad colaborador proyecto

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cedula	Varchar	11	Not null			Llave primaria y foránea que indica la cédula del colaborador o colaboradora.
Proyecto	Varchar	250	Not null			Llave primaria y foránea que indica

						el nombre del proyecto
Tipo_Jornada	Varchar	19	Not null			Tipo de jornada que posee el colaborador o colabora al proyecto asignado.
Monto_Jornada	Decimal	20,2	Not null			Monto acordado por la jornada.
Fecha_Inicio	Date		Not null			Fecha de inicio del periodo de contrato.
Fecha_Final	Date		Not null			Fecha final del periodo de contrato.

Fuente: Elaboración propia.

Gastos operativos.

Tabla 34. Entidad gastos operativos

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_GastOp	Int		Not null	Auto increment		Llave primaria que indica el código del gasto operativo.
Actividad	Varchar	250	Not null			Nombre del gasto operativo.
Detalle	Varchar	250	Null			Detalle adicional del gasto

						operativo
Total	Decimal	20,2	Not null			Suma total de todos los gastos mensuales.
Enero	Decimal	20,2	Not null	0		Gasto operativo del mes de Enero.
Febrero	Decimal	20,2	Not null	0		Gasto operativo del mes de Febrero.
Marzo	Decimal	20,2	Not null	0		Gasto operativo del mes de Marzo.
Abril	Decimal	20,2	Not null	0		Gasto operativo del mes de Abril.
Mayo	Decimal	20,2	Not null	0		Gasto operativo del mes de Mayo.
Junio	Decimal	20,2	Not null	0		Gasto operativo del mes de Junio.
Julio	Decimal	20,2	Not null	0		Gasto operativo del mes de Julio.
Agosto	Decimal	20,2	Not null	0		Gasto operativo del mes de Agosto.
Septiembre	Decimal	20,2	Not null	0		Gasto operativo del mes de Septiembre.
Octubre	Decimal	20,2	Not null	0		Gasto operativo del mes de Octubre.
Noviembre	Decimal	20,2	Not null	0		Gasto operativo del mes de Noviembre.

Diciembre	Decimal	20,2	Not null	0		Gasto operativo del mes de Diciembre.
Cod_Cat	Int		Not null			Llave foránea que indica la categoría del gasto operativo.
Periodo	Char	4	Not null			Periodo (año) de los gastos operativos.

Fuente: Elaboración propia.

Categoría de gastos operativos.

Tabla 35. Entidad gastos operativos categoría.

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_Cat	Int		Not null	Auto increment		Llave primaria que indica la categoría de los gastos operativos.
Nombre	Varchar	150	Not null			Nombre de categoría.

Fuente: Elaboración propia.

Flujos de ingresos

Tabla 36. Entidad flujos ingresos.

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_Ingreso	Int		Not null	Auto increment		Llave primaria que indica el código del ingreso.
Descripcion	Varchar	250	Not null			Descripción o nombre del ingreso.
Presupuesto_C	Decimal	20,2	Not null			Presupuesto en colones del ingreso.
Presupuesto_D	Decimal	20,2	Null			Presupuesto en dólares del ingreso.
Supuesto	Decimal	20,2	Not null			Supuesto restante del ingreso.
Enero	Decimal	20,2	Not null	0		Ingreso en el mes de Enero.
Febrero	Decimal	20,2	Not null	0		Ingreso en el mes de Febrero.
Marzo	Decimal	20,2	Not null	0		Ingreso en el mes de Marzo.
Abril	Decimal	20,2	Not null	0		Ingreso en el mes de Abril.
Mayo	Decimal	20,2	Not null	0		Ingreso en el mes de Mayo
Junio	Decimal	20,2	Not null	0		Ingreso en el mes

						de Junio.
Julio	Decimal	20,2	Not null	0		Ingreso en el mes de Julio.
Agosto	Decimal	20,2	Not null	0		Ingreso en el mes de Agosto.
Septiembre	Decimal	20,2	Not null	0		Ingreso en el mes de Septiembre.
Octubre	Decimal	20,2	Not null	0		Ingreso en el mes de Octubre.
Noviembre	Decimal	20,2	Not null	0		Ingreso en el mes de Noviembre.
Diciembre	Decimal	20,2	Not null	0		Ingreso en el mes de Diciembre.
Periodo	Decimal	4	Not null			Periodo (año) del flujo de ingresos.

Fuente: Elaboración propia.

Categoría de flujo de ingresos.

Tabla 37. Entidad flujo ingresos

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_Cat	Int		Not null	Auto increment		Llave primaria que indica el código de la categoría de los ingresos del flujo.
Nombre	Varchar	150	Not null			Nombre de la categoría.

Fuente: Elaboración propia.

Flujo de ingresos proyecto.

Tabla 38. Entidad flujo ingresos proyecto

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Proyecto	Varchar	250	Not null			Llave primaria y foránea que indica el nombre del proyecto.
Cod_Cat	Int		Not null			Llave primaria y foránea que indica la categoría de los ingresos del flujo
Cod_Ingreso	Int		Not null			Llave primaria y foránea que indica los ingresos.

Fuente: Elaboración propia.

Flujo de gastos de ejecución.

Tabla 39. Entidad flujo gastos ejecución

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_GastEjec	Int		Not null	Auto increment		Llave primaria que indica el código del gasto de ejecución.
Descripcion	Varchar	250	Not null			Descripción o nombre del gasto de ejecución.
Presupuesto_C	Decimal	20,2	Not null			Presupuesto en

						colones del gasto de ejecución.
Presupuesto_D	Decimal	20,2	Null			Presupuesto en dólares del gasto de ejecución.
Supuesto	Decimal	20,2	Not null			Supuesto restante del gasto de ejecución.
Enero	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Enero.
Febrero	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Febrero.
Marzo	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Marzo.
Abril	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Abril.
Mayo	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Mayo
Junio	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Junio.
Julio	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Julio.
Agosto	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Agosto.
Septiembre	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Septiembre.

Octubre	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Octubre.
Noviembre	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Noviembre.
Diciembre	Decimal	20,2	Not null	0		Gasto de ejecución en el mes de Diciembre.
Periodo	Decimal	4	Not null			Periodo (año) del flujo de gastos de ejecución.

Fuente: Elaboración propia.

Categoría de flujo de gastos de ejecución.

Tabla 40. Entidad flujo gastos ejecución categoría

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cod_Cat	Int		Not null	Auto increment		Llave primaria que indica el código de la categoría de los gastos de ejecución.
Nombre	Varchar	150	Not null			Nombre de categoría.

Fuente: Elaboración propia.

Flujos de gastos de ejecución proyecto.

Tabla 41. Entidad flujos gastos ejecución proyecto

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Proyecto	Varchar	250	Not null			Llave primaria y foránea que indica el nombre del proyecto.
Cod_Cat	Int		Not null			Llave primaria y foránea que indica el código de la categoría de los gastos de ejecución de los flujos.
Cod_GastEjec	Int		Not null			Llave primaria y foránea que indica los gastos de ejecución.

Fuente: Elaboración propia.

Proveedores.

Tabla 42. Entidad proveedores

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cedula	Varchar	12	Not null			Llave primaria que indica la cédula o cédula jurídica del proveedor.

Nombre	Varchar	250	Not null			Nombre del proveedor.
Telefono1	Int		Not null			Número telefónico primario del proveedor.
Telefono2	Int		Null			Número telefónico secundario del proveedor.
Direccion	Varchar	250	Null			Dirección del proveedor.
Tipo	Varchar	20	Not null			Tipo de proveedor, físico, jurídico o físico extranjero.
Estado	Bit		Not null	1		Estado lógico del proveedor, habilitado o deshabilitado.

Fuente: Elaboración propia.

Cientes.

Tabla 43. Entidad clientes

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cedula	Varchar	12	Not null			Llave primaria que indica la cédula o cédula jurídica del cliente.
Nombre	Varchar	250	Not null			Nombre del cliente.
Telefono1	Int		Not null			Número telefónico primario del cliente.
Telefono2	Int		Null			Número telefónico

						secundario del cliente.
Direccion	Varchar	250	Null			Dirección del cliente.
Tipo	Varchar	20	Not null			Tipo de cliente, físico, jurídico o físico extranjero.
Estado	Bit		Not null	1		Estado lógico del cliente, habilitado o deshabilitado.

Fuente: Elaboración propia.

Cientes proyecto

Tabla 44. Entidad cliente proyecto

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
Cliente	Varchar	12	Not null			Llave primaria y foránea que indica el cliente asociado al proyecto.
Proyecto	Varchar	250	Not null			Llave primaria y foránea que indica el nombre del proyecto.

Fuente: Elaboración Propia.

Cuentas por pagar

Tabla 45. Entidad cuentas pagar

Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
N_Documento	Int		Not null	Auto increment		Llave primaria que indica el número de cuenta por pagar.
Fecha_Emision	Date		Not null			Fecha cuando se abrió la cuenta por pagar.
Proveedor	Varchar	12	Not null			Llave foránea que indica el proveedor al que se debe cancelar la cuenta por pagar.
Tipo	Varchar	100	Not null			Tipo de documento de la cuenta por pagar.
Descripcion	Varchar	250	Not null			Descripción de la cuenta por pagar.
Vencimiento	Date		Not null			Fecha de vencimiento para la cuenta por pagar.
Monto	Decimal	10,2	Not null			Monto a pagar.
Estado	Char	7	Not null	'Abierto'		Estado de la cuenta por pagar.
Estado_logico	Bit		Not null	1		Estado lógico de las cuentas por pagar.

Fuente: Elaboración propia.

Cuentas por cobrar.

Tabla 46. Entidad cuentas cobrar

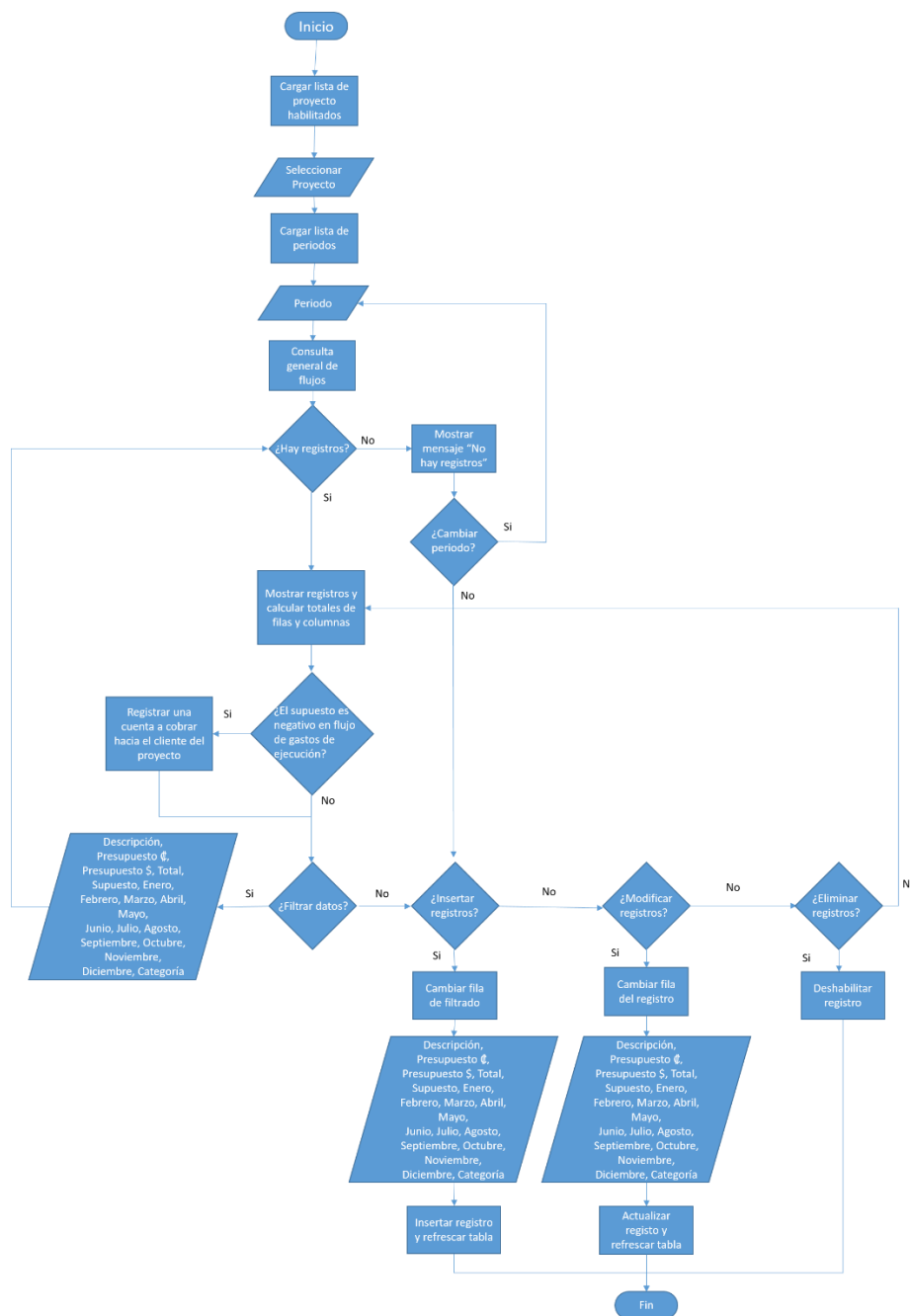
Nombre	Tipo de dato	Longitud	Nulidad	Por defecto (default)	Regla de validación	Descripción
N_Documento	Int		Not null	Auto increment		Llave primaria que indica el número de cuenta por cobrar.
Fecha_Emision	Date		Not null			Fecha cuando se abrió la cuenta por cobrar.
Cliente	Varchar	12	Not null			Llave foránea que indica el cliente al que se debe cancelar la cuenta por cobrar.
Tipo	Varchar	100	Not null			Tipo de documento de la cuenta por cobrar.
Descripcion	Varchar	250	Not null			Descripción de la cuenta por cobrar.
Vencimiento	Date		Not null			Fecha de vencimiento para la cuenta por cobrar.
Monto	Decimal	10,2	Not null			Monto a cobrar.
Estado	Char	7	Not null	'Abierto'		Estado de la cuenta por cobrar.
Estado_logico	Bit		Not null	1		Estado lógico de las cuentas por cobrar.

Fuente: Elaboración propia.

Diseño de procesos.

En este apartado se muestran los distintos diseños planteados de los principales procesos que se realizan dentro de prototipo funcional para la gestión de proyectos obedeciendo los diferentes elementos que se utilizan para los diagramas de flujo, estos son:

Figura 22. Proceso de flujo de ingresos o gastos de ejecución.



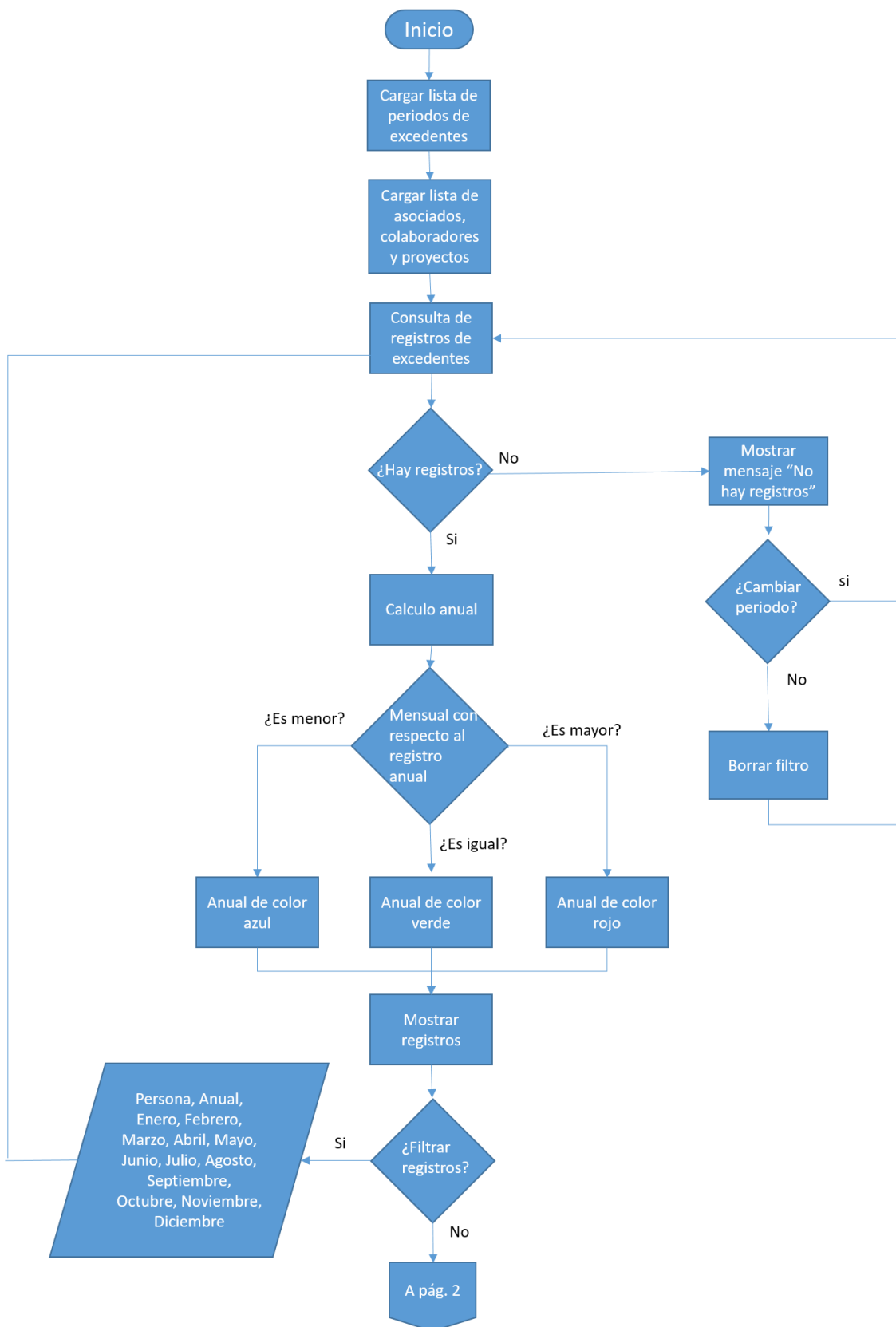
Fuente: Elaboración propia.

Dentro de este proceso se abarca el registro de los diferentes ingresos y gastos de ejecución para cada proyecto, es importante que éste, inicia una vez que el usuario comienza el caso de uso de flujos previamente establecido.

Primeramente, la opción de flujos no estará disponible si no hay ningún proyecto registrado y habilitado dentro de inicio, este proceso consulta si existen proyectos registrados dentro del prototipo. Pero al cumplir dicho requisito previo se muestra una consulta general de los diferentes registros de las entidades de flujos de ingreso y de gastos de ejecución del proyecto y el periodo seleccionado, de manera predetermina será el más actual.

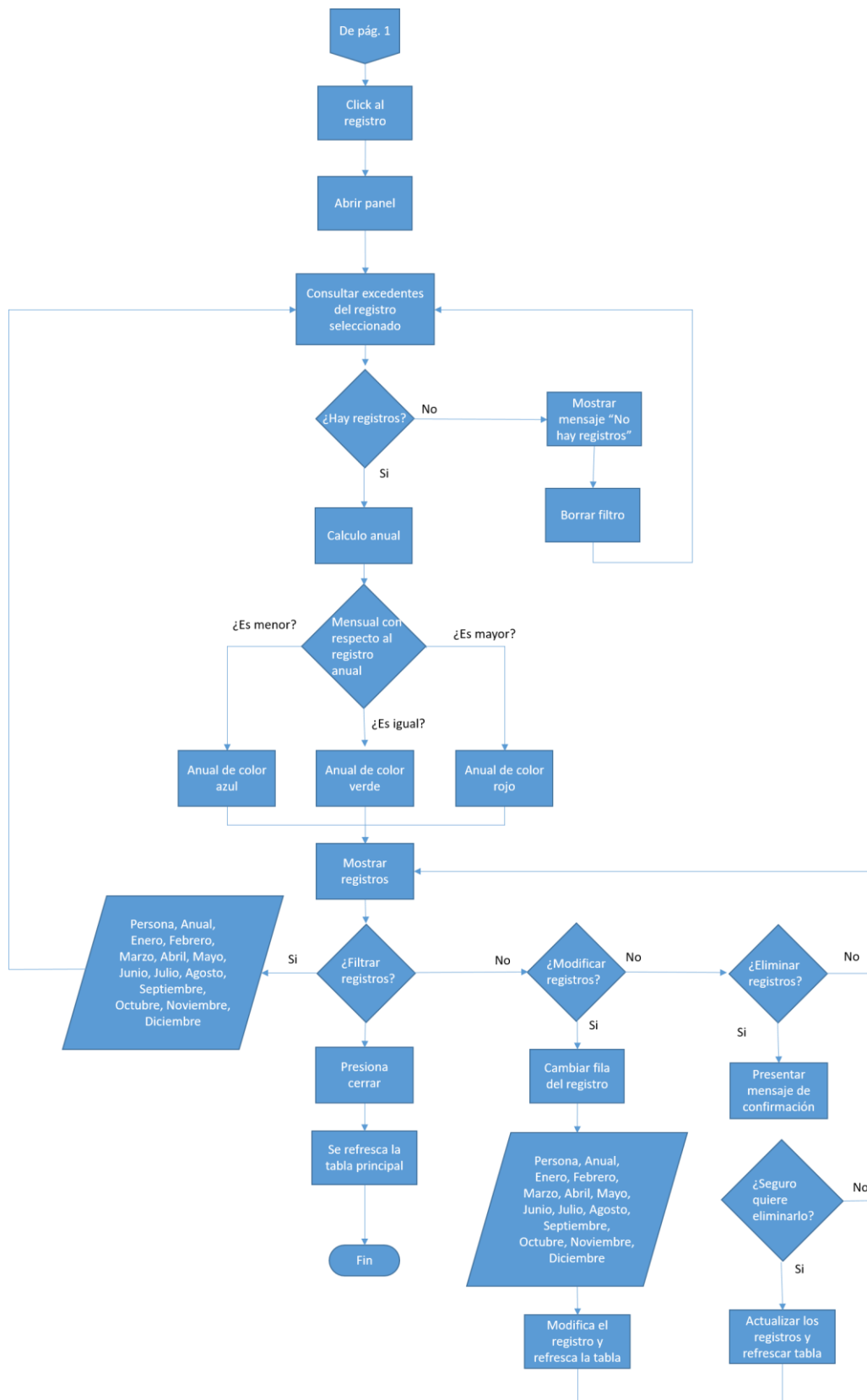
Al realizar la carga de estos registros se mostrarán las diferentes totalidades, dentro de la fila azul de las tablas, a su vez, dentro de las columnas total gastos o total ingresos se mostrarán la sumatoria de los diferentes montos de cada mes del periodo y se realizará una suma (ingreso) o una resta (gasto) con las totalidades y el presupuesto, para dar como resultado el supuesto, permitiendo ver si estos son positivos o negativos. En caso que el supuesto de gastos de ejecución de un resultado negativo, automáticamente se abrirá una cuenta por cobrar al cliente del correspondiente proyecto, el cual finalizará 6 meses posteriores a la primera incidencia del mismo.

Figura 23. Proceso de excedentes parte 1



Fuente: Elaboración propia.

Figura 24. Proceso de excedentes parte 2



Fuente: Elaboración propia.

Dentro del proceso se muestran los registros de los excedentes anuales y mensuales totales que le corresponde a cada uno de los asociados, asociadas, colaboradores y colaboradoras que trabajan en el desarrollo de los proyectos en el mismo periodo (año).

Este inicia consultando los registros de los distintos asociados, asociadas, colaboradores, colaboradoras y los distintos proyectos que estén habilitados dentro del prototipo funcional, en caso de que alguno de los estos no presente registros el sistema despliega un mensaje para notificar al usuario de ello y llenará la lista con los registros disponibles.

Posteriormente, el sistema realizará una consulta con el excedente anual y mensual total de cada uno de los registros dentro de la base de datos, donde luego se realiza un cálculo de los meses de cada uno de los registros para determinar si se igualó, superó o no superó el monto establecido del excedente anual correspondiente, esto es mostrado por medio de colores al usuario.

Sí el usuario quiere ver con más detalle los excedentes y/o modificarlos procederá hacer click a alguna de las filas de los registros de la tabla mostrada, por lo que el prototipo por medio de un evento abrirá un panel con una nueva tabla que muestra el detalle de los excedentes que corresponden a la persona del registro seleccionado anteriormente, esta vez dividido por sus proyectos. En esta nueva tabla se podrá filtrar, modificar o eliminar los datos dependiendo de la necesidad del usuario.


Diseño de salidas.

A continuación, se muestran los diferentes diseños de los reportes que se estipularon dentro de la proyección funcional y que son utilizados dentro del prototipo funcional para mostrar información de relevancia hacia el usuario, entre ellos están los siguientes:

- Reporte de los distintos costos operativos de la Cooperativa divididos por sus categorías (año) en que estos se dieron, donde se muestra el nombre de la actividad, el monto total y su detalle.

Figura 25. Reporte gastos operativos

Pág [Número de página].



Reporte de Gastos Operativos en el Periodo [Periodo]

Reporte sobre los diferentes gastos operativos de la Cooperativa Sulá Batsú R.L durante el [Periodo], en este se muestran los datos generales sobre cada uno de las actividades que constituyen dichos gastos, como su total mensual y el detalle agregado.

Gastos Operativos

Categoría: [Nombre de la Categoría1] (Total: ₡[Total de la categoría1])

Código	Actividad	Total
[Código1]	[Actividad1]	[Total1]
Detalle: [Detalle1]		
[Código2]	[Actividad2]	[Total2]
Detalle: [Detalle2]		
[Código3]	[Actividad3]	[Total3]
Detalle: [Detalle3]		

Categoría: [Nombre de la Categoría2] (Total: ₡[Total de la categoría2])

Código	Actividad	Total
[Código1]	[Actividad1]	[Total1]
Detalle: [Detalle1]		
[Código2]	[Actividad2]	[Total2]
Detalle: [Detalle2]		
[Código3]	[Actividad3]	[Total3]
Detalle: [Detalle3]		


Fecha: [Fecha de emisión del reporte dd/mm/YYYY]

Fuente: Elaboración propia.

- Reporte de las cuentas por cobrar y pagar divididos por los clientes o proveedores a los que éstas van dirigidas, dentro de éstos se podrán ver los distintos montos, fechas de emisión y vencimiento y el número de documento único de los mismos.

Figura 26. Reporte cuentas por pagar

Pág [Número de página].



Reporte de Cuentas por Pagar

Reporte sobre las diferentes cuentas por pagar de la Cooperativa Sulá Batsú R.L., en este se muestran los datos generales sobre cada una de las cuentas con sus diferentes clientes

Cuentas por pagar

Proveedor: [Nombre del proveedor1].

N°	Emisión	Tipo	Descripción	Vencimiento	Monto
[Código1]	[Fecha1]	[Tipo1]	[Descripción1]	[Vecimiento1]	[Monto1]
[Código2]	[Fecha2]	[Tipo2]	[Descripción2]	[Vecimiento2]	[Monto2]

Proveedor: [Nombre del proveedor2].

N°	Emisión	Tipo	Descripción	Vencimiento	Monto
[Código1]	[Fecha1]	[Tipo1]	[Descripción1]	[Vecimiento1]	[Monto1]
[Código2]	[Fecha2]	[Tipo2]	[Descripción2]	[Vecimiento2]	[Monto2]

Proveedor: [Nombre del proveedor2].

N°	Emisión	Tipo	Descripción	Vencimiento	Monto
[Código1]	[Fecha1]	[Tipo1]	[Descripción1]	[Vecimiento1]	[Monto1]
[Código2]	[Fecha2]	[Tipo2]	[Descripción2]	[Vecimiento2]	[Monto2]


Fecha: [Fecha de emisión del reporte dd/mm/YYYY]

Fuente: Elaboración propia.

- Reporte estado de los proyectos, tanto en general como entre los distintos periodos permitiendo un amplio margen de observación y supervisión

Figura 27. Reporte de estado de proyecto por periodo

Pág [Número de página].



Reporte del estado del proyecto [Nombre del Proyecto] por período

Reporte sobre los diferentes flujos de ingreso y de gastos de ejecución del proyecto [Nombre proyecto] de la Cooperativa Sulá Batsú R.L durante el [Periodo], donde se muestran los nombres, presupuesto establecido, total mensual y el total restante de cada uno de los datos, subdivididos por sus categorías.

Flujo de Ingresos

Código	Nombre	Presupuesto	Total Mensual	Total
	[Categoría1]	[Presupuesto Total Cat1]	[Mensual Total Cat1]	[Total Cat1]
[Código1]	[Nombre1]	[Presupuesto1]	[Mensual1]	[Total1]
[Código2]	[Nombre2]	[Presupuesto2]	[Mensual2]	[Total2]
	[Categoría2]	[Presupuesto Total Cat2]	[Mensual Total Cat2]	[Total Cat2]
[Código3]	[Nombre3]	[Presupuesto3]	[Mensual3]	[Total3]
[Código4]	[Nombre4]	[Presupuesto4]	[Mensual4]	[Total4]
Total de flujos de ingreso				[Sumatoria]

Flujo de Gastos de Ejecución

Código	Nombre	Presupuesto	Total Mensual	Total
	[Categoría1]	[Presupuesto Total Cat1]	[Mensual Total Cat1]	[Total Cat1]
[Código1]	[Nombre1]	[Presupuesto1]	[Mensual1]	[Total1]
[Código2]	[Nombre2]	[Presupuesto2]	[Mensual2]	[Total2]
	[Categoría2]	[Presupuesto Total Cat2]	[Mensual Total Cat2]	[Total Cat2]
[Código3]	[Nombre3]	[Presupuesto3]	[Mensual3]	[Total3]
[Código4]	[Nombre4]	[Presupuesto4]	[Mensual4]	[Total4]
Total de flujos de gastos de ejecución				[Sumatoria]


Fecha: [Fecha de emisión del reporte dd/mm/YYYY]

Fuente: Elaboración propia.

- Reporte de la distribución de excedentes entre los distintos asociados y colaboradores asignados a los distintos proyectos que tiene la Cooperativa, dentro éste se puede ver la distribución mensual de los excedentes correspondientes al asociado o colaborador dividido entre los proyectos de donde se tomó.

Figura 28. Reporte de distribución de excedentes.

Pág [Número de página].



Reporte de distribución de excedentes del [Nombre del Asociado o Colaborador]

Reporte sobre la manera en que se distribuyó los excedentes del asociado en el periodo [Periodo], en el cual se muestra los diferentes proyectos de donde se distribuyó el pago de cada mes.

Proyecto: [Nombre del Proyecto1]

Código	Total Mensual	Anual
[Código1]	[TotalMen1]	[Anual1]
Distribución		
Enero		[MontoEnero1]
Febrero		[MontoFebrero1]
Marzo		[MontoMarzo1]
Abril		[MontoAbril1]
Mayo		[MontoMayo1]
Junio		[MontoJunio1]
Julio		[MontoJulio1]
Agosto		[MontoAgosto1]
Septiembre		[MontoSeptiembre1]
Octubre		[MontoOcupre1]
Noviembre		[MontoNoviembre1]
Diciembre		[MontoDiciembre1]

Proyecto: [Nombre del Proyecto2]

Código	Total Mensual	Anual
[Código2]	[TotalMen2]	[Anual2]
Distribución		
Enero		[MontoEnero2]
Febrero		[MontoFebrero2]
.....	
Noviembre		[MontoNoviembre2]
Diciembre		[MontoDiciembre2]

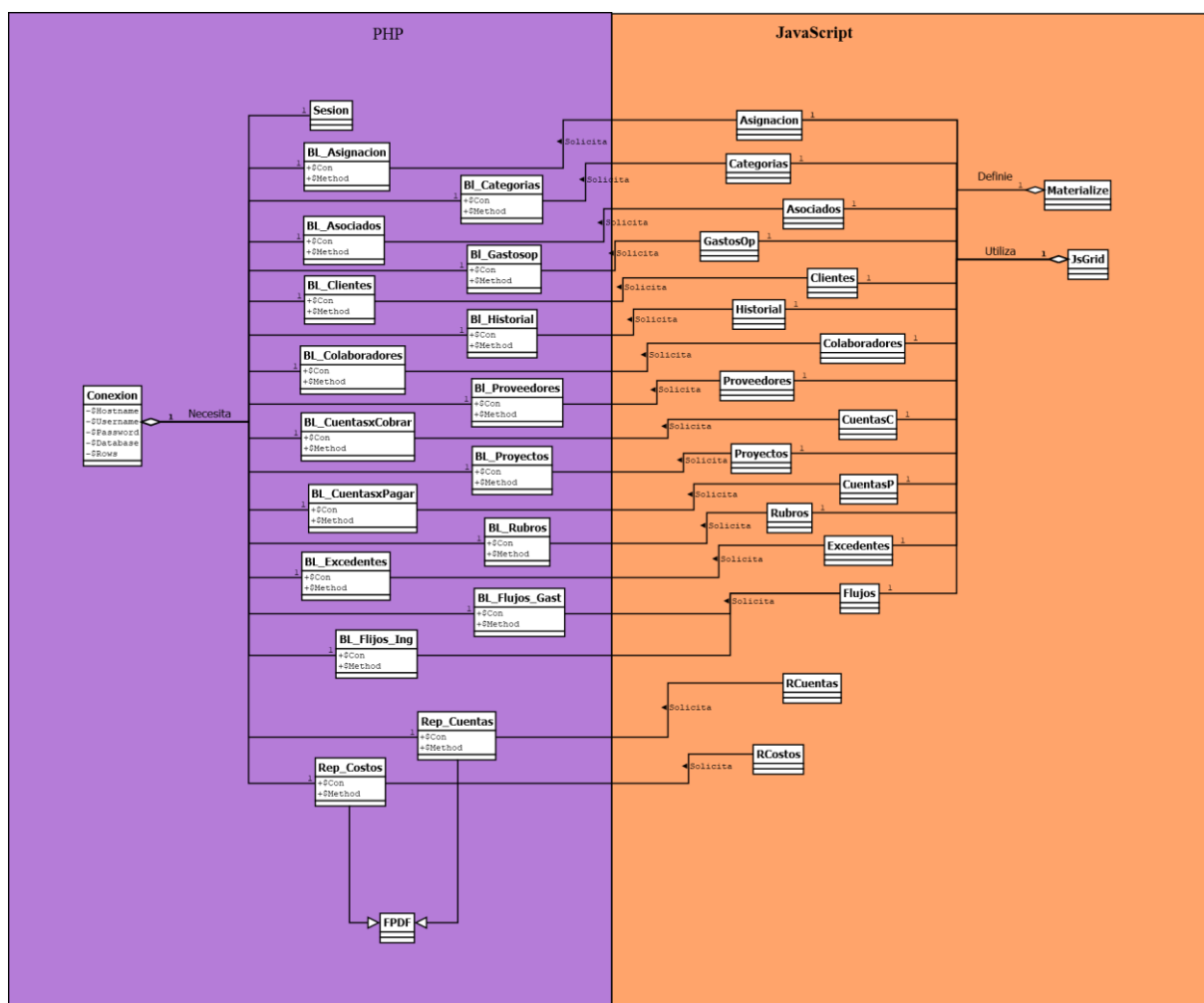
Fecha: [Fecha de emisión del reporte dd/mm/YYYY]

Fuente: Elaboración propia.

Diagramas UML.

Dentro del prototipo elaborado se utiliza una arquitectura de programación POO (Programación Orientada a Objetos) se determina que la manera más eficiente para ejemplificar la estructura de éste es mediante un diagrama de clases, en este diagrama de clases se presentan dos grandes grupos, aquellos que están elaborados por medio del lenguaje de PHP y lenguaje JavaScript. Estos dos grupos se relacionan mediante solicitudes Ajax de JQuery por lo que la omisión de la clase JQuery se debe a que éste engloba todo el prototipo.

Figura 29. Diagrama de clases



Fuente: Elaboración propia.

Programación

En este apartado se hace mención a diferentes procesos que se encuentran codificados dentro del prototipo funcional, donde se presenta una parte del código al lector para que éste pueda comprenderlo en una mayor medida.

Entradas y salidas.

Las entradas de las distintas ventanas del prototipo funcional son realizadas mediante solicitudes Ajax en JavaScript y JQuery, la cuales son ejecutadas por medio de distintos métodos dentro de la clase de JsGrid.

Figura 30. Código entrada al insertar

```

insertItem: function(item){
    return $.ajax({
        type: "POST",
        url: "classes/bl_asociados.php",
        data:item,
        success: function(result){
            var json = result;
            if(json.response.status == 'error') {
                if(json.response.result.code == 1062){
                    swal("MySQL Error "+json.response.result.code,"Ya existe un asociado/a con esa cédula","error");
                }else{
                    swal("MySQL Error "+json.response.result.code,json.response.result.message,"error");
                }
            }
            StartGridAsociados(disable);
        },
        error: function (xhr, ajaxOptions, thrownError) {
            swal("Insert asociados","Error "+xhr.status+": "+thrownError+ " en ajax", "error");
        }
    });
},

```

Fuente: Elaboración propia.

En el código anterior se puede observar que mediante la función insertItem, se ejecuta y retorna una solicitud Ajax hacia la clase bl_asociados definida en PHP, mediante una petición POST, la cual recibe los datos ingresados por el usuario para procesarlos y enviarlos para su inserción a la base de datos.

Las salidas, por otro lado, son muy similares a las entradas ya que éstas también utilizan los mismos componentes, sin embargo, dentro de la clase JsGrid utiliza otro método como se puede observar en el siguiente ejemplo.

Figura 31. Código de salida para mostrar registros

```

loadData: function(filter){
  var deferred = $.Deferred();
  filter.disable = disable;
  $.ajax({
    type: "GET",
    url: "classes/bl_asociados.php",
    data:filter,
    success: function(result){
      var json = result;
      if(json.response.status != 'error') {
        swal("MySQL Error "+json.response.result.code, json.response.result.message, "error");
      }else{
        deferred.resolve(json.response.result);
      }
    },
    error: function (xhr, ajaxOptions, errorThrown) {
      swal("Consulta asociado/as","Error "+xhr.status+": "+thrownError+ " en ajax","error");
    }
  });
  return deferred.promise();
},

```

Fuente: Elaboración propia.

Como se observa en esta función se ejecuta una solicitud Ajax hacia la clase bl_asociados una vez más, pero dependiendo de la respuesta que se obtenga del servidor se desplegará un mensaje de error o se retornará los datos solicitados.

Procesos.

A continuación, se mostrará fragmentos de código de los procesos más importantes utilizados dentro del prototipo funcional, para aclarar al lector sobre cómo fueron estos elaborados.

Flujos de ingresos y gastos de ejecución.

En este proceso se utilizan la clase de JsGrid para la elaboración de una tabla dinámica tomando un diseño familiar para el usuario, dentro se utiliza código en JavaScript y JQuery permitiendo validar y realizar diferentes acciones que finalizarán con solicitudes Ajax a clases en PHP para manipular o consultar los datos.

Figura 32. Código proceso flujo gastos ejecución

```

if(tsup<0){
$.ajax({
type: "GET",
url: "classes/bl_flujos_gast.php",
data: {"CC":true, "Proyecto":proyecto, "Periodo":periodo, "Cliente":cliente},
success: function(data){
var json = data;
if(json.response.status == 'error') {
swal("MySQL Error "+json.response.result.code, json.response.result.message, "error");
}else{
if(json.response.result.length==0){
swal({
title: "El supuesto total de los flujos de gastos de ejecución del proyecto "+proyecto+" en el periodo "+periodo+"
es negativo",
text: "¿Desea ingresar un cuenta por cobrar?",
icon: "warning",
buttons: {
cancel: 'No',
save: 'Si'
}
});
}).then(function(value){
if(value){
var monto = (tsup*-1);
var hoy = new Date();
var emision = hoy.getFullYear()+"-"+(hoy.getUTCMonth()+1)+"-"+hoy.getUTCDate();
hoy.setUTCMonth(hoy.getUTCMonth()+8);
var vencimiento = hoy.getFullYear()+"-"+(hoy.getUTCMonth()+1)+"-"+hoy.getUTCDate();
$.ajax({
type:"POST",
url: "classes/bl_flujos_gast.php",
data: {"CC":true, "Fecha_emision":emision, "Vencimiento":vencimiento, "Proyecto":proyecto,
"Periodo":periodo, "Cliente":cliente, "Monto":monto},
success: function(data){
var res = data;
if(res.response.status == 'error') {
alert("MySQL Error "+res.response.result.code+" "+res.response.result.message);
}else{

```

Fuente: Elaboración propia.

El código anterior se realiza justamente después de cargar los registros de la tabla de flujos de gastos de ejecución mediante el método `OnDataLoaded` de `JsGrid`, acá se verifica si el supuesto de los gastos de ejecución total da un valor menor a 0, en cuyo caso se procede a comprobar si existe una cuenta por cobrar ya abierta del proyecto, si es así se ajustará el monto con respecto del supuesto negativo, en caso contrario, se procederá preguntar al usuario si desea abrir una cuenta por cobrar al cliente del proyecto por la cantidad del supuesto negativo.

Durante la realización de este proceso se hacen varias comprobaciones, donde se determina si existe la cuenta por cobrar, si existen dos o más cuentas por cobrar del mismo proyecto y periodo, evitando así que se pueda determinar la cuenta por cobrar correcta por lo que se envía un mensaje al usuario de dicho error o si la cuenta no existe del todo.

Excedentes.

Este proceso está, también, realizado por medio de la clase `JsGrid` para la elaboración de la dos tablas utilizadas y el comportamiento que tienen éstas, utilizan código JavaScript y eventos de `JQuery` para definir el diseño y el comportamiento del prototipo, para posteriormente, por medio de solicitudes Ajax a clases en PHP se manipularán o solicitarán datos.

Figura 33. Código proceso excedentes

```

cess: function(result){
var res = result;
if(res.response.status1=="error" || res.response.status2=="error" || res.response.status3=="error"){
var error = "";
if(res.response.status1=="error"){
error += "(Asociados) MySQL Error "+res.response.result1.code+" "+res.response.result1.message+"\n";
}
if(res.response.status2=="error"){
error += "(Colaboradores) MySQL Error "+res.response.result2.code+" "+res.response.result2.message+"\n";
}
if(res.response.status3=="error"){
error += "(Proyectos) MySQL Error "+res.response.result3.code+" "+res.response.result3.message+"\n";
}
swal("Error al cargar las listas", error, "error");
}else{
var asociados = res.response.result1;
var colaboradores = res.response.result2;
var proyectos = res.response.result3;
proyectos.unshift({"Nombre":null,"Cuenta_colones":null,"Cuenta_dolares":null});
var list = [{Cedula: null, NombreCompleto: "", Cuenta_colones: "", Cuenta_dolares:""}];
if(asociados.length == 0 && colaboradores.length == 0){
swal("No hay datos", "No hay registros disponibles en asociados o colaboradores", "error");
}else{
$.each(asociados, function(key, val){
var Nombre = val.NombreCompleto;
val.NombreCompleto = "As - "+Nombre;
list.push(val);
});
$.each(colaboradores, function(key, val){
var Nombre = val.NombreCompleto;
val.NombreCompleto = "Co - "+Nombre;
list.push(val);
});
}
StartExcedentes(list,$("#Select-Periodo").val(), proyectos);
}
$("#Select-Periodo").change(function(){
console.log($("#Select-Periodo").val());
StartExcedentes(list,$("#Select-Periodo").val(), proyectos);
});
$('.tooltipped').tooltip({
delay: 50,
tooltip: "Ingresar dato a buscar"
});
}
}

```

Fuente: Elaboración propia.

Este código es ejecutado cuando se finaliza de cargar la página, en éste por medio de una solicitud Ajax solicitará los datos de asociados, colaboradores y proyectos disponibles dentro del prototipo funcional para llenar las listas necesarias para los excedentes, para posteriormente, invocar el método de StartExcedentes que inicia las configuraciones necesarias para desplegar los registros de excedentes dentro de la tabla.

Validaciones.

En el caso de las validaciones, éstas son realizadas mediante los métodos y funciones de la clase de JsGrid dentro de las cuales se ejecuta el código JavaScript y JQuery para realizar las validaciones, retornando en caso de ser necesario el mensaje de error.

Figura 34. Código de validaciones

```

validate: [{
  validator: "required",
  message: function(value, item){
    return "Ingrese un Número de Cuenta en Colones";
  }
},{
  message: "Cuenta colones debe tener 20 caracteres",
  validator: function(value){
    if(value.length<=20){
      return true;
    }else if(value == null){
      return true;
    }else{
      return false;
    }
  }
},{
  message: "Ingrese solamente números en Cuenta Colones",
  validator: function(value){
    if(/^([0-9]+)$/.test(value)){
      return true;
    }else if(value==null||value==''){
      return true;
    }else{
      return false;
    }
  }
}]

```

Fuente: Elaboración propia.

Como se muestra en el código anterior, el campo se valida con respecto de:

- No se encuentre vacío, ya que se establece un validador de “required” que significa requerido.
- No debe tener más de 20 caracteres dentro del campo.
- Solamente se pueden ingresar números dentro del campo, esto excluye los símbolos especiales o el símbolo de negativo “-”.

La función “validate” se encargará de enviar el arreglo con las funciones y los mensajes correspondientes a la clase de JsGrid, la cual se encargará de presentar al usuario los errores si estos se presentan.

Módulos señalados.

Los módulos planteados dentro del documento están realizados por medio de métodos y funciones de PHP y JavaScript, estos se interconectan y realizan distintas tareas, determinadas por el módulo que afectan su comportamiento.

Seguridad.

Dentro de este módulo, mediante PHP se comprueban las credenciales de usuario y contraseña que el usuario ingrese dentro del prototipo, donde si estos están correctamente identificados dentro de la base de datos procederá a redirigirlo a la página de inicio, en caso contrario, retornará al login y mostrará un mensaje de error.

Figura 35. Código de seguridad

```

<?php
  if (isset($_GET["log"])){
    if($_GET["log"]=="out"){
      include("classes/sesion.php");
      $Se = new Session();
      $Se->Start();
      $Se->End();
    }
  }
  if (isset($_GET["error"])){
    $error= $_GET["error"];
  }else{
    $error=null;
  }
  if($error == 'forbidden'){
    echo "<script>alert('No tiene acceso al sistema');</script>";
  }
  if (isset($_POST["User"]) Or isset($_POST["Password"])){
    include("classes/conexion.php");
    $Con= new Connection();
    $user=$_POST["User"];
    $pass=$_POST["Password"];
    $array= $Con->Login($user,$pass);
    if (empty($array)) {
      $error="wrong";
    }else{
      include("classes/sesion.php");
      $Se = new Session();
      $Se->Start();
      $Se->set("Usuario",$array[0]["Usuario"]);
      $Se->set("Rol",$array[0]["Rol"]);
      header("Location: inicio.php");
    }
  }
}
?>

```

Fuente: Elaboración propia.

Figura 36. Código de seguridad 2

```
<?php
    Include("classes/sesion.php");
    Include("classes/conexion.php");
    date_default_timezone_set('America/Costa_Rica');
    $Se = new Session();
    $Se->Start();
    $Con = new Connection();
    if($_SESSION["Rol"]!='Administrador'){
        header("Location: index.php?error=forbidden");
    }
?>
```

Fuente: Elaboración propia.

Dentro del módulo, a su vez, se comprueba que el rol que haya sido autenticado dentro del prototipo sea el que cumpla el rol de Administrador, ya que éste es el único rol que tiene acceso al mismo.

Mantenimientos.

En este módulo se utiliza la clase de JavaScript y JQuery para tablas dinámicas llamada JsGrid, que permiten realizar los diferentes procesos de consultar, insertar, actualizar y eliminar, los cuales son ejecutados mediante distintos botones que se muestran en las diferentes tablas que componen el prototipo funcional.

Figura 37. Código de mantenimientos JavaScript

```

controller:{
  loadData: function(filter){
    loadData: function(filter){
      var deferred = $.Deferred();
      filter.disable = disable;
      $.ajax({
        type: "GET",
        url: "clases/bl_asociados.php",
        data:filter,
        success: function(result){
          var json = result;
          if(json.response.status == 'error') {
            swal("MySQL Error "+json.response.result.code, json.response.result.message, "error");
          }else{
            deferred.resolve(json.response.result);
          }
        },
        error: function (xhr, ajaxOptions, thrownError) {
          swal("Consulta asociado/as","Error "+xhr.status+": "+thrownError+ " en ajax","error");
        }
      });
      return deferred.promise();
    },
    insertItem: function(item){
      return $.ajax({
        type: "POST",
        url: "clases/bl_asociados.php",
        data:item,
        success: function(result){
          var json = result;
          if(json.response.status == 'error') {
            if(json.response.result.code == 1062){
              swal("MySQL Error "+json.response.result.code,"Ya existe un asociado/a con esa cédula","error");
            }else{
              swal("MySQL Error "+json.response.result.code,json.response.result.message,"error");
            }
          }
          StartGridAsociados(disable);
        },
        error: function (xhr, ajaxOptions, thrownError) {
          swal("Inserción asociados","Error "+xhr.status+": "+thrownError+ " en ajax", "error");
        }
      });
    },
    updateItem: function(item){
      return $.ajax({
        type: "PUT",
        url: "clases/bl_asociados.php",
        data:item,
        success: function(result){
          var json = result;
          if(json.response.status == 'error') {
            swal("MySQL Error "+json.response.result.code,json.response.result.message,"error");
          }
          StartGridAsociados(disable);
        },
        error: function (xhr, ajaxOptions, thrownError) {
          swal("Edición asociados","Error "+xhr.status+": "+thrownError+ " en ajax","error");
        }
      });
    },
    deleteItem: function(item){
      item.Estado = 0;
      return $.ajax({
        type: "DELETE",
        url: "clases/bl_asociados.php",

```

Fuente: Elaboración propia.

Figura 38. Código mantenimientos PHP

```

<?php
include("conexion.php");
$con = new Connection();
$method = $_SERVER['REQUEST_METHOD'];
if($method=="GET"){
  if($_GET["disable"]=="false"){
    $estado = 1;
  }else{
    $estado = 0;
  }
  $numero = $_GET["numero"];
  if($numero==null){
    $numero = 'null';
  }else if($numero == 'N/A' || $numero=="n/a"){
    $numero = 1;
  }else{
    $numero = (int)$_GET["numero"];
  }
  $cuentaD = $_GET["Cuenta_dolares"];
  if($cuentaD == null){
    $cuentaD = 'null';
  }else if($cuentaD == 'N/A' || $cuentaD=="n/a"){
    $cuentaD = "1";
  }else{
    $cuentaD = "%".$_GET["Cuenta_dolares"]."%" ;
  }
  $excendente = isset($_GET["Exc_porcentaje"]) ? ((int)$_GET["Exc_porcentaje"]/100):'null';
  $query = $con->Select_G_Asociados("%".$_GET["Cedula"]."%", "%".$_GET["Nombre"]."%", "%".$_GET["Apellido1"]."%", "%".$_GET["Apellido2"]."%", $numero, "%".$_GET["Cuenta_colones"]."%", $cuentaD, $excendente, $estado);
}
if($method=="POST"){
  $numero = $_POST["numero"]!=null? (int)$_POST["numero": 'null';
  $cuentaD = $_POST["Cuenta_dolares"]!=null? $_POST["Cuenta_dolares": 'null';
  $query = $con->Insert_Asociados($_POST["Cedula"], $_POST["Nombre"], $_POST["Apellido1"], $_POST["Apellido2"], $numero, $_POST["Cuenta_colones"], $cuentaD, ((int)$_POST["Exc_porcentaje"]/100));
}
if($method=="PUT"){
  parse_str(file_get_contents("php://input"), $PUT);
  $numero = $_PUT["numero"]!=null? (int)$_PUT["numero": 'null';
  $cuentaD = $_PUT["Cuenta_dolares"]!=null? $_PUT["Cuenta_dolares": 'null';
  $query = $con->Update_Asociados($_PUT["Cedula"], $_PUT["Nombre"], $_PUT["Apellido1"], $_PUT["Apellido2"], $numero, $_PUT["Cuenta_colones"], $cuentaD, ((int)$_PUT["Exc_porcentaje"]/100), 1);
}
if($method=="DELETE"){
  parse_str(file_get_contents("php://input"), $DELETE);
  $numero = $_DELETE["numero"]!=null? (int)$_DELETE["numero": 'null';
  $cuentaD = $_DELETE["Cuenta_dolares"]!=null? $_DELETE["Cuenta_dolares": 'null';
  $query = $con->Update_Asociados($_DELETE["Cedula"], $_DELETE["Nombre"], $_DELETE["Apellido1"], $_DELETE["Apellido2"], $numero, $_DELETE["Cuenta_colones"], $cuentaD, ((int)$_DELETE["Exc_porcentaje"]/100), $_DELETE["Estado"]);
}
header("Content-Type: application/json");
echo json_encode($query);
?>

```

Fuente: Elaboración propia.

Como se puede observar en las imágenes anteriores la clase de JsGrid brinda las funciones necesarias para realizar los distintos mantenimientos de las diferentes entidades del prototipo, como:

- **loadData:** Función que se encarga de hacer una solicitud Ajax a la clase PHP, que consulta y/o filtra los distintos registros de la base de datos.
- **insertItem:** Función que realiza una solicitud Ajax a la clase PHP con los datos brindados por el usuario para su inserción dentro de la base de datos.
- **updateItem:** Función que realiza una solicitud Ajax a la clase PHP para guardar los cambios de los datos que el usuario haya editado.
- **deleteItem:** Función para eliminar o deshabilitar los registros dentro de la base de datos mediante una solicitud Ajax a la clase PHP. Se eliminan o deshabilitan los registros dependiendo de la entidad donde se ejecute la función.

Reportes.

Módulo en el cual se utiliza código PHP, Javascript y la biblioteca de FPDF para crear y solicitar los archivos PDF con los datos correspondientes a cada uno de los reportes planteados dentro del prototipo funcional.

Figura 39. Código de reportes JavaScript

```
function PDFCuentasCobrar(){
    var estado = $("#Select-Estado-CCobrar").val();
    var disable = $('#DisableCCobrar').is(':checked');
    var reporte = "CuentasCobrar"
    $.ajax({
        type:"Get",
        url: "classes/rep_cuentas.php",
        data: {"Consulta":true,"reporte":reporte,"estado":estado,"disable":disable},
        success: function(result){
            var res = JSON.parse(result);
            if(res.response.status=="error"){
                alert("MySQL Error: "+res.response.result.code+": "+res.response.result.message);
            }else{
                if(res.response.result.length == 0){
                    alert("Reporte vacio");
                }else{
                    window.open("classes/rep_cuentas.php?reporte=CuentasCobrar&estado="+estado+"&disable="+disable);
                }
            }
        }
    });
}
```

Fuente: Elaboración propia.

Figura 40. Código reportes PHP

```

class PDF extends FPDF{
    var $widths;
    var $aligns;
    function Header(){ }
    function Footer(){ }
    function SetWidths($w){ }
    function SetAligns($a){ }
    function Row($data,$state){ }
    function CheckPageBreak($h){ }
    function NbLines($w,$txt){ }

    function CuentasCobrarTabla($header,$ccobrar,$clientes){
        $flag = false;
        $this->SetFont('Arial','B',18);
        $this->SetX(10);
        $this->Cell(0,0,"Cuentas por cobrar");
        $this->Ln(10);
        $this->SetWidths(array(10,30,30,50,35,40));
        $this->SetAligns(array("C","C","C","L","C","C"));
        //Recorrer array para imprimir diferentes tablas
        for($j=0;$j<count($clientes);$j++){
            $this->SetFont('', 'B', 14);
            if(!$flag){
                $this->SetX(20);
                $this->Cell(10,4,"Cliente: ".utf8_decode($clientes[$j]["Cliente"]).".");
                $this->Ln(10);
                for($i=0;$i<count($header);$i++){
                    $this->Cell($this->widths[$i],7,utf8_decode($header[$i]),1,0,'C');
                }
                $this->Ln();
                $flag = true;
            }
            foreach($ccobrar as $row){
                if($row["Cliente"]==$clientes[$j]["Cliente"]){
                    $array=array();
                    array_push($array,$row["N_Documento"], $row["Fecha_emision"],$row["Tipo"], $row["Descripcion"], $row["Vencimiento"], $row["Monto"]);
                    $this->Row($array,$row["Estado_logico"]);
                }else{
                    $flag = false;
                }
            }
        }
        $this->Ln(10);
    }
}

```

Fuente: Elaboración propia.

Dentro del código se realizan solicitudes Ajax mediante JavaScript por medio de eventos de JQuery con los distintos botones. Al ejecutarse la parte de PHP es donde se utiliza la biblioteca FPDF con una herencia de ésta a la clase PDF para crear todos los apartados del documento final.

Consultas.

Las consultas utilizadas dentro del prototipo funcional son efectuadas por medio de JavaScript, PHP y la clase de JsGrid, la cual permite una forma más sencilla y natural de consultar diferentes datos dentro de los registros de la base de datos.

Figura 41. Código de consultas JavaScript

```

loadData: function(filter){
    var deferred = $.Deferred();
    filter.disable = disable;
    $.ajax({
        type: "GET",
        url: "classes/bl_cuentasxcobrar.php",
        data:filter,
        success: function(result){
            var json = result;
            if(json.response.status == 'error') {
                if(json.response.result.code == 1292){
                    swal("MySQL Error "+json.response.result.code, "El formato de fecha es incorrecto", "error");
                }else{
                    swal("MySQL Error "+json.response.result.code, json.response.result.message, "error");
                }
            }
            $(''.tooltipped').tooltip("close");
            StartGridCuentasxCobrar(clientes,disable);
        }else{
            deferred.resolve(json.response.result);
        }
    },
    error: function (xhr, ajaxOptions, thrownError) {
        swal("Consulta cuentas por cobrar", "Error "+xhr.status+": "+thrownError+ " en ajax", "error");
    }
    });
    return deferred.promise();
},

```

Fuente: Elaboración propia.

Figura 42. Código de consultas PHP

```

if($method=="GET"){
    if ($_GET["disable"]=="false"){
        $estado = 1;
    }else{
        $estado = 0;
    }
    $documento = $_GET["N_Documento"]!=null? $_GET["N_Documento"]: 'null';
    $femision = $_GET["Fecha_emision"]!=null? "".$_GET["Fecha_emision"]."": 'null';
    $vencimiento = $_GET["Vencimiento"]!=null? "".$_GET["Vencimiento"]."": 'null';
    $monto = $_GET["Monto"]!=null? $_GET["Monto"]: 'null';
    $query = $Con->Select_G_CuentasCobrar($documento, $femision, "%".$_GET["Cliente"]."%", "%".$_GET["Tipo"]."%",
        "%".$_GET["Descripcion"]."%", $vencimiento, $monto, "%".$_GET["Estado"]."%", $estado);
}

```

Fuente: Elaboración propia.

Mediante la fila de filtrado que brinda JsGrid dentro de sus tablas dinámicas, se pueden filtrar los datos con respecto de los datos que el usuario ingrese, estos inicialmente se encontrarán vacíos permitiendo realizar una consulta general de todos los registros de la entidad, pero una vez que el usuario ingrese algún dato éste lo consultará automáticamente y mostrará aquellos que coincidan con la búsqueda, en caso que no se encuentre nada se desplegará un mensaje de registros no encontrados.

Cálculos financieros.

La funcionalidad de este módulo está dispersa en distintas funciones y operaciones dentro del prototipo funcional, siendo éstas ejecutadas por código PHP dentro de distintas clases.

Figura 43. Código de cálculos financieros

```

if(args.data.length!=0){
    var total = 0;
    var tene = 0;
    var tfeb = 0;
    var tmar = 0;
    var tabr = 0;
    var tmay = 0;
    var tjun = 0;
    var tjul = 0;
    var tago = 0;
    var tsep = 0;
    var toct = 0;
    var tnov = 0;
    var tdec = 0;
    args.data.forEach(function(element){
        if(element.Cod_GastOp == undefined){
            total += parseFloat(element.Total);
            tene += parseFloat(element.Enero);
            tfeb += parseFloat(element.Febrero);
            tmar += parseFloat(element.Marzo);
            tabr += parseFloat(element.Abril);
            tmay += parseFloat(element.Mayo);
            tjun += parseFloat(element.Junio);
            tjul += parseFloat(element.Julio);
            tago += parseFloat(element.Agosto);
            tsep += parseFloat(element.Septiembre);
            toct += parseFloat(element.Octubre);
            tnov += parseFloat(element.Noviembre);
            tdec += parseFloat(element.Diciembre);
        }
    });
    args.data.unshift({"Cod_GastOp":"Totales","Actividad":"","Detalle":"","Total":total, "Enero":tene, "Febrero":tfeb, "Marzo":tmar,
    "Abril":tabr, "Mayo":tmay, "Junio":tjun, "Julio":tjul, "Agosto":tago, "Septiembre":tsep, "Octubre":toct, "Noviembre":tnov,
    "Diciembre":tdec});
    $("#Grid-GastOp").jsGrid("refresh");
}
}

```

Fuente: Elaboración propia.

Dentro de éste se realiza una sumatoria de todos los distintos registros para pasar a obtener la totalidad del total mensual y el total de cada uno de los meses para facilitar al usuario la información general de cómo se encuentra los gastos operativos de la Cooperativa.

Figura 44. Código de cálculos financieros 2

```

$total = ($_POST["Enero"]+$_POST["Febrero"]+$_POST["Marzo"]+$_POST["Abril"]+$_POST["Mayo"]+$_POST["Junio"]+$_POST["Julio"]+
$_POST["Agosto"]+$_POST["Septiembre"]+$_POST["Octubre"]+$_POST["Noviembre"]+$_POST["Diciembre"]);
$supuesto = ($total + $_POST["Presupuesto_C"]);
$query = $Con->Insert_FlujoIngreso($_POST["Descripción"], $_POST["Presupuesto_C"],($_POST["Presupuesto_C"]/$_POST["Cambio"]),
$total, $supuesto, $_POST["Enero"], $_POST["Febrero"], $_POST["Marzo"], $_POST["Abril"], $_POST["Mayo"], $_POST["Junio"],
$_POST["Julio"], $_POST["Agosto"], $_POST["Septiembre"], $_POST["Octubre"], $_POST["Noviembre"], $_POST["Diciembre"],
$_POST["Periodo"], $_POST["Proyecto"], $_POST["Cod_Cat"]);

```

Fuente: Elaboración propia.

Este se encarga de determinar flujo de ingreso total al sumar los distintos montos de los meses y al dar el supuesto de éste al sumar dicho monto con el supuesto establecido, a su vez, por otro lado, se encarga de realizar la conversión a dólares con el tipo de cambio.

Cuentas por pagar.

Este módulo como los anteriores está constituido por código en JavaScript, PHP y la clase de JsGrid, dentro de éste se establecen los mismos métodos y funciones a un mantenimiento, donde se verifica los datos que el usuario ingresa y se realiza el proceso determinado.

Figura 45. Código cuentas por pagar.

```

jsGrid.locale("es");
$("#Grid-CuentasxPagar").jsGrid({
  width: "100%",
  filtering: true,
  inserting: true,
  editing: true,
  sorting: true,
  paging: true,
  autoload: true,
  rowClass: function(item) { ... },
  pageSize: 10,
  pageButtonCount: 5,
  confirmDeleting: false,
  onItemDeleting: function(args) { ... },
  invalidNotify: function(args) { ... },
  rowClick: function(args) { ... },
  controller: {
    loadData: function(filter) { ... },
    insertItem: function(item) { ... },
    updateItem: function(item) { ... },
    deleteItem: function(item) { ... }
  },
  fields: [
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { name: "Estado", type: "select", width: 100, align: "center", title: "Estado",
      items: [{"Estado": ""}, {"Estado": "Abierto"}, {"Estado": "Cerrado"}], valueField: "Estado", textField: "Estado",
      filterTemplate: function() { ... },
      filterValue: function() { ... },
      insertTemplate: function() { ... },
      insertValue: function() { ... },
      validate: [ ... ]
    },
    { ... }
  ]
});
$('input').on('apply.daterangepicker', function(ev, picker) {
  $(this).val(picker.startDate.format('DD/MM/YYYY'));
  if($(this).parent().is(".dateFilter")){
    $("#Grid-CuentasxPagar").jsGrid("search");
  }
});

```

Fuente: Elaboración propia.

Dentro del código en JavaScript se puede ver cómo está compuesto la tabla dinámica de los distintos registros de las cuentas por pagar, éstas poseen sus propias validaciones, funciones y métodos para poder registrarlas, editarlas, consultarlas y deshabilitarlas como el usuario así lo requiera.

Figura 46. Código alertas cuentas por pagar

```

foreach($vencp as $row){
    $fecha = new DateTime($row["Vencimiento"], new DateTimeZone("America/Costa_Rica"));
    $diff = $hoy->diff($fecha);
    if(($diff->days<=31&&$diff->days>=28)&&$diff->invert){
        $wcpmes = true;
        array_push($mescp,$row["N_Documento"]);
        $notifications = true;
    }else if($diff->days<=14&&$diff->days>7&&$diff->invert){
        $wcp2semana = true;
        array_push($cp2seman,$row["N_Documento"]);
        $notifications = true;
    }

    if($wcpmes){
        $txt = '';
        $l = count($mescp);
        for($i=0;$i<$l;$i++){
            if($l==1){
                $txt .= $mescp[$i];
            }else if($i == ($l-1)&&$l>=2){
                $txt .= " y ".$mescp[$i];
            }else if($i == ($l-2)&&$l>=2){
                $txt .= $mescp[$i];
            }else{
                $txt .= $mescp[$i].", ";
            }
        }
        echo '<li>
            <a href="cuentasp.php" class="grey-text text-darken-2">
                <span class="material-icons icon-bg-circle dark-yellow small">warning</span>
                Las cuentas por pagar: '.$txt.' vencerán en un mes o menos
            </a>
        </li>';
    }
}

```

Fuente: Elaboración propia.

Como se puede observar mediante código PHP las alertas de las cuentas por pagar son determinadas por la fecha de vencimiento de éstas, hechas para alertar al usuario de que se encuentran próximas a vencer en un rango de un mes, dos semanas, una semana o de su vencimiento total.

Cuentas por cobrar.

Este módulo es muy similar al de cuentas por pagar ya que está constituido por los mismos elementos donde se establecen métodos y funciones similares, pero aún se difieren debido a que éstas obedecen a la entidad de cuentas por cobrar.

Figura 47. Código cuentas por cobrar.

```

function StartGridCuentasxCobrar(clientes,disable){
  jsGrid.locale("es");
  $("#Grid-CuentasxCobrar").jsGrid({
    width: "100%",
    filtering:true,
    inserting: true,
    editing: true,
    sorting: true,
    paging: true,
    autoload: true,
    rowClass: function(item){ ... },
    pageSize: 10,
    pageButonCount: 5,
    confirmDeleting: false,
    onItemDeleting: function(args){ ... },
    invalidNotify: function(args) { ... },
    rowClick: function(args){ ... },
    controller:{
      loadData: function(filter){ ... },
      insertItem: function(item){ ... },
      updateItem: function(item){ ... },
      deleteItem: function(item){ ... }
    },
  },

  fields: [
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { name: "Estado", type: "select", width: 100, align:"center", title:"Estado",
      items:[{"Estado":""}, {"Estado":"Abierto"}, {"Estado":"Cerrado"}], valueField:"Estado", textField: "Estado",
      filterTemplate: function(){ ... },
      filterValue: function(){ ... },
      insertTemplate: function(){ ... },
      insertValue: function(){ ... },
      validate: [{ ... }]
    },
  ],
  { type: "control",
    itemTemplate: function(value, item){ ... }
  }
]
});

```

Fuente: Elaboración propia.

Dentro del código en JavaScript se puede ver de igual manera cómo está compuesto la tabla dinámica de los distintos registros de las cuentas por cobrar, pero obedeciendo las características de los propios campos de esta entidad, así como sus validaciones, funciones y métodos para poder registrarlas, editarlas, consultarlas y deshabilitarlas como el usuario así lo requiera.

Figura 48. Código alertas cuentas por cobrar.

```

each($venc as $row){
    $fecha = new DateTime($row["Vencimiento"], new DateTimeZone("America/Costa_Rica"));
    $diff = $hoy->diff($fecha);
    if(($diff->days<=31&&$diff->days>=28)&&!$diff->invert){
        $wccmes = true;
        array_push($mescc,$row["N_Documento"]);
        $notifications = true;
    }else if($diff->days<=14&&$diff->days>7&&!$diff->invert){
        $wcc2semana = true;
        array_push($scc2seman,$row["N_Documento"]);
        $notifications = true;
    }

    if($wccmes){
        $txt = '';
        $l = count($mescc);
        for($i=0;$i<$l;$i++){
            if($l==1){
                $txt .= $mescc[$i];
            }else if($i == ($l-1)&&$l>=2){
                $txt .= " y ".$mescc[$i];
            }else if($i == ($l-2)&&$l>=2){
                $txt .= $mescc[$i];
            }else{
                $txt .= $mescc[$i].", ";
            }
        }
        echo '<li>
            <a href="cuentasc.php" class="grey-text text-darken-2">
                <span class="material-icons icon-bg-circle dark-yellow small">warning</span>
                Las cuentas por cobrar: '.$txt.' vencerán en un mes o menos
            </a>
        </li>';
    }
}

```

Fuente: Elaboración propia,

Como se puede observar mediante código PHP las alertas de las cuentas por cobrar son igualmente determinadas por la fecha de vencimiento de ésta, hechas para alertar al usuario de que están próximas a vencer en un rango de un mes, dos semanas, una semana o de su vencimiento total.

Excedentes a asociados.

Último módulo propuesto en el alcance se utiliza, como en los anteriores, código PHP, JavaScript y la clase de JsGrid para mostrar los distintos registros, tanto para las asignaciones a los proyectos como a la muestra de los distintos excedentes.

Figura 49. Código asignaciones

```

$(function() {
  $.ajax({
    type:"Get",
    url: "classes/bl_asignacion.php",
    data:{"Load":true},
    success: function(result){
      var res = result;
      if(res.response.status1=="error" || res.response.status2=="error" || res.response.status3=="error"){
        var error = "";
        if(res.response.status1=="error"){
          error += "(Proyectos) MySQL Error "+res.result1.code+" "+res.result1.message+"\n";
        }
        if(res.response.status2=="error"){
          error += "(Asociados) MySQL Error "+res.result2.code+" "+res.result2.message+"\n";
        }
        if(res.response.status3=="error"){
          error += "(Colaboradores) MySQL Error "+res.result3.code+" "+res.result3.message;
        }
        swal("Error al cargar las listas",error,"error");
      }else{
        var proyectos = res.result1;
        var asociados = res.result2;
        var colaboradores = res.result3;
        proyectos.unshift({Nombre:null});
        asociados.unshift({Cedula:null, NombreCompleto:""});
        colaboradores.unshift({Cedula:null, NombreCompleto:""});
        StartGridAsigAsociados(proyectos, asociados);
        StartGridAsigColaboradores(proyectos, colaboradores);
      }
      $('#.tooltipped').tooltip({
        delay: 50,
        tooltip: "Ingresar dato a buscar"
      });
    }
  });
});

```

Fuente: Elaboración propia.

Código donde se cargan las diferentes listas de proyectos, asociados y colaboradores para las tablas dinámicas de JsGrid de asignaciones de asociados y asociadas a proyectos y colaborador y colaboradores a proyectos hechas con la clase de JsGrid.

Figura 51. Código de cálculo excedente anual

```

[Renderer: function(value, item){
var val = parseFloat(value).toFixed(2);
var num = ""
if(val > 999){
    num = "$"+val.replace(/\B(?=(\d{3})+(?!\d))/g, ",");
}else{
    num = "$"+val;
}
}
var td = $(" ").addClass("jsgrid-cell jsgrid-align-center").text(num); var total = (parseFloat(item.Enero) + parseFloat(item.Febrero) + parseFloat(item.Marzo) + parseFloat(item.Abril) + parseFloat(item.Mayo) + parseFloat(item.Junio) + parseFloat(item.Julio) + parseFloat(item.Agosto) + parseFloat(item.Septiembre) + parseFloat(item.Octubre) + parseFloat(item.Noviembre) + parseFloat(item.Diciembre)); var mensual = total; if(this.inserting || this.editing){     if(val < mensual){         return td.addClass("red");     }else if(val > mensual){         return td.addClass("blue");     }else if(val == mensual){         return td.addClass("green");     } } } } |
```

Fuente: Elaboración propia.

En este fragmento se calcula el monto total de la sumatoria de los montos de los diferentes meses para comparar si el monto obtenido es igual, menor o mayor al monto anual registrado dentro de la base de datos, mostrando al usuario en distintos colores el estado de la siguiente forma: Rojo (mensual mayor al anual), Verde (mensual igual al anual) y Azul (mensual menor al anual).

Pruebas

Tabla 47. Caso de prueba autenticación de usuario.

Inicio de Sesión			
ID Caso de Prueba: 01		Módulo a probar: Módulo de Seguridad	
Autor: Steven Morera Hernández		Fecha de elaboración: 20/2/2019	
Descripción			
En este proceso de prueba se comprueba el manejo de flujos de ingresos y gastos de ejecución para cada uno de los proyectos dentro de un periodo determinado.			
Precondiciones:		Usuario registrado dentro de la base de datos.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados	Resultados

		Esperados	Obtenidos
Se ingresa al enlace index	_____	El prototipo muestra la pantalla de login.	El prototipo muestra la pantalla de login.
El usuario digita su usuario	steven95.mh@gmail.com	_____	_____
El usuario digita su contraseña	smorera	_____	_____
El usuario presiona el botón de ingresar	_____	El usuario es correcto y es redirigido a la pantalla de inicio del prototipo.	El usuario es correcto y es redirigido a la pantalla de inicio del prototipo.
El usuario da click a la opción de log out en el menú superior	_____	El prototipo cierra la sesión y redirige el usuario a la pantalla de login.	El prototipo cierra la sesión y redirige el usuario a la pantalla de login.
El usuario digita un usuario o deja el campo en blanco	xqpr	_____	_____
El usuario digita una contraseña o deja el campo en blanco	pe1341	_____	_____
El usuario presiona el botón de ingresar	_____	El prototipo muestra un mensaje de que el usuario o la contraseña son incorrectos	El prototipo muestra un mensaje de que el usuario o la contraseña son incorrectos

Estado	Correcto.
Fin de caso de prueba,	

Fuente: Elaboración propia.

Tabla 48. Caso de prueba mantenimiento asociados

Mantenimiento Asociados			
ID Caso de Prueba: 02		Módulo a probar: Módulo de Mantenimientos	
Autor: Steven Morera Hernández		Fecha de elaboración: 20/2/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar y deshabilitar asociados			
Precondiciones:		Correcta autenticación del usuario.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Mantenimientos – Asociados en el menú principal	_____	El prototipo muestra la pantalla de usuarios, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de usuarios, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Cédula: 546456651 Nombre: Shirley Primer Apellido: Salazar Segundo Apellido: Alpízar	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.

	<p>Teléfono: 8435564</p> <p>Cuenta ¢: 1122003355446677</p> <p>Cuenta \$: 1122003355446678</p> <p>% Excedente: 30</p>		
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	<p>Cédula: 2425224B</p> <p>Nombre: _____</p> <p>Primer Apellido: Pérez</p> <p>Segundo Apellido: Alpízar</p> <p>Teléfono: 843546564</p> <p>Cuenta ¢: 11220033SDX46677</p> <p>Cuenta \$: 11ASDAFAAF46678</p> <p>% Excedente: SAD</p>	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	<p>Cédula: _____</p> <p>Nombre: Shirley</p> <p>Primer Apellido: Salazar</p> <p>Segundo Apellido: _____</p> <p>Teléfono: 8435564</p> <p>Cuenta ¢: 1122003355446677</p> <p>Cuenta \$: _____</p> <p>% Excedente: 30</p>	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	<p>Cédula: aszxa</p> <p>Nombre: vserw</p> <p>Primer Apellido: adasd</p> <p>Segundo Apellido: qqrq</p> <p>Teléfono: ____</p> <p>Cuenta ¢: ____</p>	El prototipo muestra un mensaje indicando que no se encontraron registros.	El prototipo muestra un mensaje indicando que no se encontraron registros.

	Cuenta \$: fassas % Excedente: ____		
El usuario deja en blanco los campos de la fila de filtrado.	Cédula: _____ Nombre: _____ Primer Apellido: _____ Segundo Apellido: _____ Teléfono: _____ Cuenta ¢: _____ Cuenta \$: _____ % Excedente: _____	El prototipo realiza una consulta general de los registros de asociados,	El prototipo realiza una consulta general de los registros de asociados,
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Cédula: 546456651 Nombre: Shirley Primer Apellido: Salazar Segundo Apellido: Rojas Teléfono: 8435564 Cuenta ¢: 1122003355446677 Cuenta \$: 1122003355446678 % Excedente: 30	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no válidos	Cédula: 546456651 Nombre: Shirley Primer Apellido: Salazar Segundo Apellido: Rojas	El prototipo muestra un mensaje de error correspondiente al	El prototipo muestra un mensaje de error correspondiente al

o dejando los campos obligatorios vacíos y da click al botón de confirmar edición	Teléfono: 8435564 Cuenta ¢: _____ Cuenta \$: 1122003355446678 % Excedente: 30	error realizado en cada uno de los campos y refresca la tabla.	error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de deshabilitar registro	_____	El prototipo manda un mensaje de confirmación para deshabilitar el registro.	El prototipo manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.	_____	El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Activado.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.

El usuario activa el interruptor de deshabilitados	Deshabilitados: Desactivado.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.
Estado		Correcto.	
Fin de caso de prueba,			

Fuente: Elaboración propia.

Tabla 49. Caso de prueba mantenimiento proyectos

Mantenimiento Proyectos			
ID Caso de Prueba: 03		Módulo a probar: Módulo de Mantenimientos	
Autor: Steven Morera Hernández		Fecha de elaboración: 28/2/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar y deshabilitar proyectos			
Precondiciones:		Correcta autenticación del usuario.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Mantenimientos – Proyectos en el menú principal	_____	El prototipo muestra la pantalla de proyectos, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de proyectos, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar	_____	El prototipo reemplaza la fila de filtrado por el verde de inserción	El prototipo reemplaza la fila de filtrado por el verde de inserción de

		de registros.	registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	<p>Nombre: TIC-as</p> <p>Fecha Inicio: 01/01/2014</p> <p>Fecha Final: 31/11/2014</p> <p>Presupuesto ¢: 0</p> <p>Presupuesto \$: 0</p> <p>Gestor: Shirley Salazar Rojas</p> <p>Cuenta ¢: 21548354831854864561</p> <p>Cuenta \$: _____</p> <p>Cliente: Instituto Nacional de Aprendizaje</p>	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	<p>Nombre: TIC-as</p> <p>Fecha Inicio: 01/01/2014</p> <p>Fecha Final: 31/11/2014</p> <p>Presupuesto ¢: 0</p> <p>Presupuesto \$: 0</p> <p>Gestor: Shirley Salazar Rojas</p> <p>Cuenta ¢: 21548354831854864561</p> <p>Cuenta \$: _____</p> <p>Cliente: Instituto Nacional de Aprendizaje</p>	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	<p>Nombre: TIC-as</p> <p>Fecha Inicio: 01/01/2006</p> <p>Fecha Final: _____</p> <p>Presupuesto \$: _____</p> <p>Gestor: Shirley Salazar Rojas</p> <p>Cuenta ¢: _____</p>	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.

	21548354831854864561 Cuenta \$: _____ Cliente: _____		
El usuario digita datos que no coincidan con los registros en la fila de filtrado.	Nombre: asdafw Fecha Inicio: 15/2/1998 Fecha Final: _____ Presupuesto ¢: -25,000 Presupuesto \$: _____ Gestor: _____ Cuenta ¢: n/a Cuenta \$: _____ Cliente: _____	El prototipo muestra un mensaje indicando que no se encontraron registros.	El prototipo muestra un mensaje indicando que no se encontraron registros.
El usuario deja en blanco los campos de la fila de filtrado.	Nombre: _____ Fecha Inicio: _____ Fecha Final: _____ Presupuesto ¢: _____ Presupuesto \$: _____ Gestor: _____ Cuenta ¢: _____ Cuenta \$: _____ Cliente: _____	El prototipo realiza una consulta general de los registros de proyectos.	El prototipo realiza una consulta general de los registros de proyectos.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior y un botón llamado Ver Rubros.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior y un botón llamado Ver Rubros.
El usuario da click	_____	El prototipo	El prototipo

al botón Ver Rubros		oscurece el fondo y abre un panel superpuesto con una tabla y los registros disponibles de Rubros.	oscurece el fondo y abre un panel superpuesto con una tabla que muestra los registros de Rubros.
El usuario presiona el botón de ingresar en la tabla del panel Rubros	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros en el panel Rubros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros en el panel Rubros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar en el panel Rubros.	<p>Categoría: Transporte</p> <p>Descripción: Viaje en taxi</p> <p>Monto Unitario: 5,000</p> <p>Sub Total por persona: 15,000</p> <p>Total por días: 30,000</p>	El prototipo despliega un mensaje para recibir la cantidad de personas y de días para automatizar el cálculo de Sub Total y Total, donde posteriormente se ingresan los datos dados por el usuario y se refresca con el nuevo registro en el panel Rubros.	El prototipo despliega un mensaje para recibir la cantidad de personas y de días para automatizar el cálculo de Sub Total y Total, donde posteriormente se ingresan los datos dados por el usuario y se refresca con el nuevo registro en el panel Rubros.
El usuario digita y selecciona los	<p>Categoría: _____</p> <p>Descripción:</p>	El prototipo muestra un mensaje	El prototipo muestra un mensaje

datos incorrectos o deja campos en blanco obligatorios para ingresar un nuevo registro y da click al botón de ingresar el panel Rubros.	<p>_____</p> <p>Monto Unitario: safasf</p> <p>Sub Total por persona: _____</p> <p>Total por días: _____</p>	de error correspondiente al error realizado en cada uno de los campos en el panel Rubros.	de error correspondiente al error realizado en cada uno de los campos en el panel Rubros.
El usuario da click a botón de editar o a un registro dentro de la tabla en el panel Rubros	<p>_____</p>	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición en el panel Rubros.	<p>Categoría: Transporte</p> <p>Descripción: Viaje en taxi</p> <p>Monto Unitario: 5,500</p> <p>Sub Total por persona: 16,500</p> <p>Total por días: 33,000</p>	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados en el panel Rubros.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados en el panel Rubros.
El usuario edita los campos del registro ingresando formatos no válidos o dejando los campos	<p>Categoría: Transporte</p> <p>Descripción: _____</p> <p>Monto Unitario: -5000</p> <p>Sub Total por persona: dasfafsafs</p>	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los

obligatorios vacíos y da click al botón de confirmar edición en el panel Rubros	Total por días: _____	campos en el panel Rubros.	campo en el panel Rubros.
El usuario da click al botón de cancelar edición en el panel Rubros.	_____	El prototipo revierte el cambio de la fila de edición del registro en el panel Rubros, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro en el panel Rubros, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar registro en el panel Rubros	_____	El prototipo manda un mensaje de confirmación para eliminar el registro en el panel Rubros.	El prototipo manda un mensaje de confirmación para eliminar el registro en el panel Rubros.
El usuario interactúa con el mensaje de confirmación para eliminar el registro en el panel Rubros.	_____	El prototipo elimina el registro y refresca la tabla en el panel Rubros.	El prototipo deshabilita el registro y refresca la tabla en el panel Rubros.
El usuario digita o selecciona los datos en la fila de filtrado en el panel Rubros.	Categoría: _____ Descripción: Viaje en taxi Monto Unitario: _____ Sub Total por persona: _____	El prototipo muestra los registros que coinciden con los datos ingresados en el panel Rubros.	El prototipo muestra los registros que coinciden con los datos ingresados en el panel Rubros.

	Total por días: 33,000		
El usuario digita o selecciona datos inválidos en la fila de filtrado en el panel Rubros.	Categoría: ____ Descripción: asfasfas Monto Unitario: 123456 Sub Total por persona: 123456 Total por días: _____	El prototipo muestra un mensaje diciendo que no se encontraron registros en el panel Rubros.	El prototipo muestra un mensaje diciendo que no se encontraron registros en el panel Rubros.
El usuario da click al botón Cerrar en la esquina inferior derecha del panel	_____	El prototipo cierra el panel y refresca la tabla mostrada en la ventana de Proyectos, con el cálculo del presupuesto en colones y su conversión a dólares.	El prototipo cierra el panel y refresca la tabla mostrada en la ventana de Proyectos, con el cálculo del presupuesto en colones y su conversión a dólares.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Nombre: TIC-as Fecha Inicio: 25/01/2014 Fecha Final: 15/12/2014 Gestor: Shirley Salazar Rojas Cuenta ¢: 21548354831854864561 Cuenta \$: ____ Cliente: Instituto Nacional de Aprendizaje	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no validos	Nombre: TIC-as Fecha Inicio: asfasfasf Fecha Final: afasfsa Gestor: Shirley Salazar	El prototipo muestra un mensaje de error correspondiente al	El prototipo muestra un mensaje de error correspondiente al

o dejando los campos obligatorios vacíos y da click al botón de confirmar edición.	Rojas Cuenta ¢: 21548354831854864561 Cuenta \$: ____ Cliente: Instituto Nacional de Aprendizaje	error realizado en cada uno de los campos.	error realizado en cada uno de los campo.
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de deshabilitar registro	_____	El prototipo manda un mensaje de confirmación para deshabilitar el registro.	El prototipo manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.	_____	El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Activado.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.

El usuario activa el interruptor de deshabilitados	Deshabilitados: Desactivado.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.
Estado		Correcto.	
Fin de caso de prueba,			

Fuente: Elaboración propia.

Tabla 50. Caso de prueba mantenimiento colaboradores.

Mantenimiento Colaboradores			
ID Caso de Prueba: 04		Módulo a probar: Módulo de Mantenimientos	
Autor: Steven Morera Hernández		Fecha de elaboración: 30/2/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar, editar y deshabilitar colaboradores.			
Precondiciones:		Correcta autenticación del usuario.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Mantenimientos – Colaboradores en el menú principal	_____	El prototipo muestra la pantalla de colaboradores, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de colaboradores, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar	_____	El prototipo reemplaza la fila de filtrado por una	El prototipo reemplaza la fila de filtrado por una

		verde de inserción de registros.	verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Cédula: 21485356483 Nombre: Maria Primer Apellido: Eugenia Segundo Apellido: Herrera Teléfono: 64562548 Cuenta ¢: 30045213548489124563 Cuenta \$: 30125613548489482274	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	Cédula: 214853AC483 Nombre: _____ Primer Apellido: _____ Segundo Apellido: _____ Teléfono: 421SDFA Cuenta ¢: 11220033SDX46677 Cuenta \$: n/a	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	Cédula: 21485356483 Nombre: María Primer Apellido: Eugenia Segundo Apellido: _____ Teléfono: 64562548 Cuenta ¢: _____ Cuenta \$: 30125613548489482274	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	Cédula: wef52ys Nombre: vs2342f Primer Apellido: afsafwq Segundo Apellido: vzsdf Teléfono: 2502	El prototipo muestra un mensaje señalando que no se encontraron	El prototipo muestra un mensaje señalando que no se encontraron

	<p>Cuenta ¢: 30045213548489124563</p> <p>Cuenta \$: n/a</p>	registros.	registros.
El usuario deja en blanco los campos de la fila de filtrado.	<p>Cédula: _____</p> <p>Nombre: _____</p> <p>Primer Apellido: _____</p> <p>Segundo Apellido: _____</p> <p>Teléfono: _____</p> <p>Cuenta ¢: _____</p> <p>Cuenta \$: _____</p>	El prototipo realiza una consulta general de los registros de asociados.	El prototipo realiza una consulta general de los registros de asociados.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	<p>Cédula: 21485356483</p> <p>Nombre: María</p> <p>Primer Apellido: Eugenia</p> <p>Segundo Apellido: Herrera</p> <p>Teléfono: 64562548</p> <p>Cuenta ¢: 30045213548489124563</p> <p>Cuenta \$: _____</p>	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando	<p>Cédula: 21485356483</p> <p>Nombre: María</p> <p>Primer Apellido: Eugenia</p>	El prototipo muestra un mensaje de error	El prototipo muestra un mensaje de error

formatos no validos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición	Segundo Apellido: Herrera Teléfono: 64AFAS548 Cuenta ¢: 3004521ADCS24563 Cuenta \$: _____	correspondiente al error realizado en cada uno de los campos y refresca la tabla.	correspondiente al error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de deshabilitar registro	_____	El prototipo manda un mensaje de confirmación para deshabilitar el registro.	El prototipo manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.	_____	El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Activado.	El prototipo refresca la tabla y muestra los registros	El prototipo refresca la tabla y muestra los registros

		habilitados y deshabilitados.	habilitados y deshabilitados.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Desactivado.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.
Estado		Correcto.	
Fin de caso de prueba,			

Fuente: Elaboración propia.

Tabla 51. Caso de prueba mantenimiento clientes

Mantenimiento Clientes			
ID Caso de Prueba: 05		Módulo a probar: Módulo de Mantenimientos	
Autor: Steven Morera Hernández		Fecha de elaboración: 30/2/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar, editar y deshabilitar clientes.			
Precondiciones:		Correcta autenticación del usuario.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Mantenimientos – Clientes en el menú principal	_____	El prototipo muestra la pantalla de colaboradores, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de colaboradores, donde una tabla es cargada con los registros disponibles.
El usuario presiona	_____	El prototipo	El prototipo

el botón de ingresar		reemplaza la fila de filtrado por la verde de inserción de registros.	reemplaza la fila de filtrado por la verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Cédula: 400004512731 Nombre: Instituto Nacional de Aprendizaje 1° Teléfono: 22525378 2° Teléfono: _____ Dirección: San José Tipo: Jurídico	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	Cédula: 4000AF512731 Nombre: _____ 1° Teléfono: 22525378 2° Teléfono: _____ Dirección: San José Tipo: Jurídico	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	Cédula: 400004512731 Nombre: Instituto Nacional de Aprendizaje 1° Teléfono: _____ 2° Teléfono: N/a Dirección: _____ Tipo: Jurídico	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	Cédula: sad1314sdad Nombre: Instituto Nacional de Aprendizaje 1° Teléfono: _____ 2° Teléfono: N/a Dirección: _____ Tipo: Jurídico	El prototipo muestra un mensaje indicando que no se encontraron registros.	El prototipo muestra un mensaje indicando que no se encontraron registros.

El usuario deja en blanco los campos de la fila de filtrado.	Cédula: _____ Nombre: _____ 1° Teléfono: _____ 2° Teléfono: _____ Dirección: _____ Tipo: _____	El prototipo realiza una consulta general de los registros de clientes.	El prototipo realiza una consulta general de los registros de clientes.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Cédula: 400004512731 Nombre: Instituto Nacional de Aprendizaje (INA) 1° Teléfono: 22525378 2° Teléfono: _____ Dirección: San José Tipo: Jurídico	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no válidos o dejando los campos obligatorios vacíos	Cédula: 400004512731 Nombre: Instituto Nacional de Aprendizaje (INA) 1° Teléfono: 22FC5378 2° Teléfono: BSDAD Dirección: San José Tipo: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca

y da click al botón de confirmar edición		la tabla.	la tabla.
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de deshabilitar registro	_____	El prototipo manda un mensaje de confirmación para deshabilitar el registro.	El prototipo manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.	_____	El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Activado.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.
El usuario activa el interruptor de	Deshabilitados: Desactivado.	El prototipo refresca la tabla y	El prototipo refresca la tabla y

deshabilitados		muestra los registros habilitados únicamente.	muestra los registros habilitados únicamente.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 52. Caso de prueba mantenimiento proveedores.

Mantenimientos Proveedores			
ID Caso de Prueba: 06		Módulo a probar: Módulo de mantenimientos	
Autor: Steven Morera Hernández		Fecha de elaboración: 30/3/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar, editar y deshabilitar proveedores.			
Precondiciones:		Correcta autenticación del usuario.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Mantenimientos – Proveedores en el menú principal	_____	El prototipo muestra la pantalla de colaboradores, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de colaboradores, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.

El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Cédula: 3101054596 Nombre: Tracopa 1° Teléfono: 22458465 2° Teléfono: _____ Dirección: Golfito, Puntarenas Tipo: Jurídico	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	Cédula: 31AFA4596 Nombre: _____ 1° Teléfono: 22458465 2° Teléfono: _____ Dirección: Golfito, Puntarenas Tipo: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	Cédula: 3101054596 Nombre: Tracopa 1° Teléfono: _____ 2° Teléfono: n/a Dirección: Golfito, Puntarenas Tipo: _____	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	Cédula: sad1314sdad Nombre: asfq25d 1° Teléfono: _____ 2° Teléfono: N/a Dirección: _____ Tipo: _____	El prototipo muestra un mensaje señalando que no se encontraron registros.	El prototipo muestra un mensaje señalando que no se encontraron registros.
El usuario deja en blanco los campos de la fila de filtrado.	Cédula: _____ Nombre: _____ 1° Teléfono: _____ 2° Teléfono: _____ Dirección: _____	El prototipo realiza una consulta general de los registros de proveedores.	El prototipo realiza una consulta general de los registros de proveedores.

	Tipo: _____		
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Cédula: 3101054596 Nombre: Tracopa (Buses) 1° Teléfono: 22458465 2° Teléfono: 22438435 Dirección: Golfito, Puntarenas Tipo: Jurídico	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no validos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición	Ceédula: 3101054596 Nombre: _____ 1° Teléfono: 22FC5378 2° Teléfono: BSDAD Dirección: San José Tipo: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de	_____	El prototipo revierte el cambio	El prototipo revierte el cambio

cancelar edición		de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de deshabilitar registro		El prototipo manda un mensaje de confirmación para deshabilitar el registro.	El prototipo manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.		El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Activado.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Desactivado.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.

Estado	Correcto.
Fin de caso de prueba.	

Fuente: Elaboración propia

Tabla 53. Caso de prueba flujos

Flujos			
ID Caso de Prueba: 07		Módulo a probar: Módulo de Cálculos Financieros.	
Autor: Steven Morera Hernández		Fecha de elaboración: 4/3/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar, editar y eliminar los flujos de ingreso y gastos de ejecución de los proyectos, así como los cálculos automatizados dentro de los mismos.			
Precondiciones:		Correcta autenticación del usuario. Tener al menos un proyecto registrador y habilitado dentro del prototipo	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Flujos en el menú principal	_____	El prototipo muestra la pantalla de flujos del primer proyecto disponible, donde dos tablas son cargadas con los registros de flujos de ingreso y flujos de gastos de ejecución del periodo con sus	El prototipo muestra la pantalla de flujos del primer proyecto disponible, donde dos tablas son cargadas con los registros de flujos de ingreso y flujos de gastos de ejecución del periodo con sus

		respectivas totalidades.	respectivas totalidades
El usuario presiona el botón de ingresar de ambas tablas.	_____	El prototipo reemplaza la fila de filtrado de ambas tabla por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado de ambas tabla por la verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar en la tabla de flujo de ingreso.	<p>Descripción: Donación Shirley.</p> <p>Presupuesto ₡: 0</p> <p>Enero: 20,000</p> <p>Febrero: 30,000</p> <p>Marzo: 15,000</p> <p>Abril: 15,000</p> <p>Mayo: 17,000</p> <p>Junio: 23,000</p> <p>Julio: 10,000</p> <p>Agosto: 5,000</p> <p>Septiembre: 5,000</p> <p>Octubre: 10,000</p> <p>Noviembre: 25,000</p> <p>Diciembre: 25,000</p> <p>Categoría: Donaciones</p>	El prototipo ingresa los datos dados por el usuario calculando la conversión del presupuesto en dólares, el total de ingresos mensual y el supuesto del mismo, donde posteriormente refresca la tabla de flujo de ingresos calculando la totalidad de cada uno de los registros mostrados, así como un total por cada categoría.	El prototipo ingresa los datos dados por el usuario calculando la conversión del presupuesto en dólares, el total de ingresos mensual y el supuesto del mismo, donde posteriormente refresca la tabla de flujo de ingresos calculando la totalidad de cada uno de los registros mostrados, así como un total por cada categoría.
El usuario ingresa datos incorrectos o deja campos obligatorios en	<p>Descripción: _____</p> <p>Presupuesto ₡: 0</p> <p>Enero: -20,000</p> <p>Febrero: 30,000</p>	El prototipo muestra un mensaje de error correspondiente al	El prototipo muestra un mensaje de error correspondiente al

blanco y da click al botón ingresar en la tabla de flujos de ingreso.	Marzo: 15,000 Abril: 15,000 Mayo: -17,000 Junio: 23,000 Julio: 10,000 Agosto: -5,000 Septiembre: 5,000 Octubre: -10,000 Noviembre: 25,000 Diciembre: 25,000 Categoría: ____	error realizado en cada uno de los campos.	error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado en la tabla de flujo de ingreso.	Código Ingreso: ____ Descripción: Donación Shirley. Presupuesto ¢: 0 Presupuesto \$: 0 Total Ingresos: 200,000 Supuesto: 200,000 Enero: 20,000 Febrero: 30,000 Marzo: 15,000 Abril: 15,000 Mayo: 17,000 Junio: 23,000 Julio: 10,000 Agosto: 5,000 Septiembre: 5,000 Octubre: 10,000 Noviembre: 25,000 Diciembre: 25,000 Categoría: Donaciones	El prototipo muestra los registros que coinciden con los datos ingresados calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.	El prototipo muestra los registros que coinciden con los datos ingresados calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.
El usuario digita	Código Ingreso: ____	El prototipo	El prototipo

<p>datos que no coinciden con los registros en la fila de filtrado en la tabla de flujo de ingreso.</p>	<p>Descripción: asdasfrt Presupuesto ¢: -20,000 Presupuesto \$: -20,000 Total Ingresos: ____ Supuesto: _____ Enero: _____ Febrero: -28,000 Marzo: -11,111 Abril: -22,222 Mayo: 22,222 Junio: -30,222 Julio: 35,354 Agosto: 150 Septiembre: 35 Octubre: _____ Noviembre: _____ Diciembre: _____ Categoría: Ingresos</p>	<p>muestra un mensaje diciendo que no se encontraron registros.</p>	<p>muestra un mensaje diciendo que no se encontraron registros.</p>
<p>El usuario deja en blanco los campos de la fila de filtrado de la tabla de flujos de ingreso.</p>	<p>Código Ingreso: _____ Descripción: _____ Presupuesto ¢: _____ Presupuesto \$: _____ Total Ingresos: ____ Supuesto: _____ Enero: _____ Febrero: _____ Marzo: _____ Abril: _____ Mayo: _____ Junio: _____ Julio: _____ Agosto: _____</p>	<p>El prototipo realiza una consulta general de los registros de ingresos calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.</p>	<p>El prototipo realiza una consulta general de los registros de ingresos calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.</p>

	Septiembre: _____ Octubre: _____ Noviembre: _____ Diciembre: _____ Categoría: _____		
El usuario da click a botón de editar o a un registro dentro de la tabla de flujos de ingreso.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición de la tabla de flujos de ingreso.	Descripción: Donación Shirley. Presupuesto ₡: 30,000 Enero: 20,000 Febrero: 30,000 Marzo: 15,000 Abril: 15,000 Mayo: 17,000 Junio: 23,000 Julio: 10,000 Agosto: 5,000 Septiembre: 5,000 Octubre: 10,000 Noviembre: 25,000 Diciembre: 25,000 Categoría: Donaciones	El prototipo modifica los campos editados dados por el usuario dentro de la tabla calculando la conversión del presupuesto en dólares, el total de ingresos mensual y el supuesto del mismo, donde posteriormente refresca la tabla para calcular la totalidad de cada uno de los registros	El prototipo modifica los campos editados dados por el usuario dentro de la tabla calculando la conversión del presupuesto en dólares, el total de ingresos mensual y el supuesto del mismo, donde posteriormente refresca la tabla para calcular la totalidad de cada uno de los registros

		mostrados, así como un total por cada categoría.	mostrados, así como un total por cada categoría..
El usuario edita los campos del registro ingresando formatos no validos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición de la tabla de flujos de ingreso.	<p>Descripción: Donación Shirley.</p> <p>Presupuesto ¢: -20,000</p> <p>Enero: 20,000</p> <p>Febrero: 30,000</p> <p>Marzo: -5,000</p> <p>Abril: 15,000</p> <p>Mayo: -18,000</p> <p>Junio: 23,000</p> <p>Julio: 10,000</p> <p>Agosto: 5,000</p> <p>Septiembre: 5,000</p> <p>Octubre: 10,000</p> <p>Noviembre: 25,000</p> <p>Diciembre: 25,000</p> <p>Categoría: _____</p>	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de cancelar edición de la tabla de flujos de ingreso.	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar registro	_____	El prototipo manda un mensaje de confirmación	El prototipo manda un mensaje de confirmación

de la tabla de flujos de ingreso.		para eliminar el registro.	para eliminar el registro.
El usuario interactúa con el mensaje de confirmación para eliminar el registro de la tabla de flujos de ingreso.	_____	El prototipo elimina el registro y refresca la tabla.	El prototipo elimina el registro y refresca la tabla.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar en la tabla de flujo de gastos de ejecución.	<p>Descripción: Viaje SJ - Alajuelita.</p> <p>Presupuesto ₡: 30,000</p> <p>Enero: 0</p> <p>Febrero: 0</p> <p>Marzo: 0</p> <p>Abril: 0</p> <p>Mayo: 5,000</p> <p>Junio: 5,000</p> <p>Julio: 7,850</p> <p>Agosto: 5,000</p> <p>Septiembre: 5,000</p> <p>Octubre: 7,850</p> <p>Noviembre: 0</p> <p>Diciembre: 0</p> <p>Categoría: Transporte</p>	El prototipo ingresa los datos dados por el usuario calculando la conversión del presupuesto en dólares, el total de gastos mensual y el supuesto del mismo, donde posteriormente refresca la tabla de flujos de gastos de ejecución calculando la totalidad de cada uno de los registros mostrados, así como un total por cada categoría.	El prototipo ingresa los datos dados por el usuario calculando la conversión del presupuesto en dólares, el total de gastos mensual y el supuesto del mismo, donde posteriormente refresca la tabla de flujos de gastos de ejecución calculando la totalidad de cada uno de los registros mostrados, así como un total por cada categoría.
El usuario ingresa datos incorrectos o	<p>Descripción: _____</p> <p>Presupuesto ₡: -30,000</p>	El prototipo muestra un	El prototipo muestra un

<p>deja campos obligatorios en blanco y da click al botón ingresar en la tabla de flujos de gastos de ejecución.</p>	<p>Enero: 0 Febrero: 0 Marzo: 0 Abril: 0 Mayo: 5,000 Junio: -5,000 Julio: 7,850 Agosto: 5,000 Septiembre: -5,000 Octubre: 7,850 Noviembre: 0 Diciembre: 0 Categoría: Transporte</p>	<p>mensaje de error correspondiente al error realizado en cada uno de los campos.</p>	<p>mensaje de error correspondiente al error realizado en cada uno de los campos.</p>
<p>El usuario decide crear una cuenta por cobrar al cliente del proyecto, por el total gasto de ejecución total negativo.</p>	<p>_____</p>	<p>El prototipo ingresa una nueva cuenta por cobrar del a nombre del cliente del proyecto con el período del flujo y el monto en positivo, mostrando al usuario el N° del documento.</p>	<p>El prototipo ingresa una nueva cuenta por cobrar del a nombre del cliente del proyecto con el período del flujo y el monto en positivo, mostrando al usuario el N° del documento.</p>
<p>El usuario decide no crear una cuenta por cobrar al cliente del proyecto, por el gasto de ejecución total negativo.</p>	<p>_____</p>	<p>El prototipo no crear una nueva cuenta por cobrar a nombre del cliente del proyecto.</p>	<p>El prototipo no crear una nueva cuenta por cobrar a nombre del cliente del proyecto.</p>

<p>El usuario digita los datos en la fila de filtrado en la tabla de flujos de gastos de ejecución.</p>	<p>Código Gasto: ____ Descripción: ____ Presupuesto ¢: 30,000 Presupuesto \$: 49.12 Total Gastos Ejecutivos: 35,700 Supuesto: 0 Enero: 0 Febrero: 0 Marzo: 0 Abril: 0 Mayo: 5,000 Junio: 5,000 Julio: 7,850 Agosto: 5,000 Septiembre: 5,000 Octubre: 7,850 Noviembre: 0 Diciembre: 0 Categoría: Transporte</p>	<p>El prototipo muestra los registros que coinciden con los datos ingresados calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.</p>	<p>El prototipo muestra los registros que coinciden con los datos ingresados calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.</p>
<p>El usuario digita datos que no coinciden con los registros en la fila de filtrado en la tabla de flujos de gastos de ejecución.</p>	<p>Código Gasto: ____ Descripción: dasafwq Presupuesto ¢: 100 Presupuesto \$: ____ Total Gastos Ejecutivos: -550 Supuesto: ____ Enero: -20,000 Febrero: 30,000 Marzo: ____ Abril: -18,000 Mayo: ____</p>	<p>El prototipo muestra un mensaje diciendo que no se encontraron registros.</p>	<p>El prototipo muestra un mensaje diciendo que no se encontraron registros.</p>

	Junio: -500 Julio: -10,000 Agosto: 700 Septiembre: 100 Octubre: _____ Noviembre: _____ Diciembre: _____ Categoría: _____		
El usuario deja en blanco los campos de la fila de filtrado de la tabla de flujos de gastos de ejecución.	Código Gasto: _____ Descripción: _____ Presupuesto ¢: _____ Presupuesto \$: _____ Total Gastos Ejecutivos: _____ Supuesto: _____ Enero: _____ Febrero: _____ Marzo: _____ Abril: _____ Mayo: _____ Junio: _____ Julio: _____ Agosto: _____ Septiembre: _____ Octubre: _____ Noviembre: _____ Diciembre: _____ Categoría: _____	El prototipo realiza una consulta general de los registros de gastos de ejecución calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.	El prototipo realiza una consulta general de los registros de gastos de ejecución calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.
El usuario da click a botón de editar o a un registro dentro de la tabla	_____	El prototipo reemplaza la fila del registro por uno amarillo claro,	El prototipo reemplaza la fila del registro por uno amarillo claro,

de flujos de gastos de ejecución.		el cual posee distintos campos editables con la información anterior.	el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición de la tabla de flujos de gastos de ejecución.	<p>Descripción: Viaje SJ - Alajuelita.</p> <p>Presupuesto ₡: 40,000</p> <p>Enero: 0</p> <p>Febrero: 0</p> <p>Marzo: 5,000</p> <p>Abril: 5,000</p> <p>Mayo: 5,000</p> <p>Junio: 5,000</p> <p>Julio: 7,850</p> <p>Agosto: 5,000</p> <p>Septiembre: 5,000</p> <p>Octubre: 7,850</p> <p>Noviembre: 0</p> <p>Diciembre: 0</p> <p>Categoría: Transporte</p>	El prototipo modifica los campos editados dados por el usuario dentro de la tabla calculando la conversión del presupuesto en dólares, el total de gastos mensual y el supuesto del mismo, donde posteriormente refresca la tabla para calcular la totalidad de cada uno de los registros mostrados, así como un total por cada categoría.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla calculando la conversión del presupuesto en dólares, el total de gastos mensual y el supuesto del mismo, donde posteriormente refresca la tabla para calcular la totalidad de cada uno de los registros mostrados, así como un total por cada categoría..
El usuario edita los campos del registro ingresando formatos no validos o dejando los campos	<p>Descripción: _____</p> <p>Presupuesto ₡: 40,000</p> <p>Enero: 0</p> <p>Febrero: 0</p> <p>Marzo: 5,000</p> <p>Abril: _____</p>	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los

obligatorios vacíos y da click al botón de confirmar edición de la tabla de flujos de gastos de ejecución.	<p>Mayo: 5,000 Junio: 5,000 Julio: ____ Agosto: -3,000 Septiembre: 5,000 Octubre: -7,850 Noviembre: 0 Diciembre: 0 Categoría: Transporte</p>	campos y refresca la tabla.	campos y refresca la tabla.
El usuario da click al botón de cancelar edición de la tabla de flujos de gastos de ejecución.	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar registro de la tabla de flujos de gastos de ejecución.	_____	El prototipo manda un mensaje de confirmación para eliminar el registro.	El prototipo manda un mensaje de confirmación para eliminar el registro.
El usuario interactúa con el mensaje de confirmación para eliminar el registro de la tabla de flujos de gastos de	_____	El prototipo elimina el registro y refresca la tabla.	El prototipo elimina el registro y refresca la tabla.

ejecución.			
El usuario selecciona un periodo sin registros disponibles.	_____	El sistema muestra un mensaje de que no se han encontrado registros disponibles en ambas tablas.	El sistema muestra un mensaje de que no se han encontrado registros disponibles en ambas tablas
El usuario selecciona un periodo con registros disponibles.	_____	El sistema muestra todos los registros del mismo periodo en ambas tablas.	El sistema muestra todos los registros del mismo periodo en ambas tablas.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 54. Caso de prueba cuentas por cobrar.

Cuentas por cobrar			
ID Caso de Prueba: 08		Módulo a probar: Módulo de Cuentas por Cobrar.	
Autor: Steven Morera Hernández		Fecha de elaboración: 09/03/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar, editar y deshabilitar las cuentas por cobrar, así como las alertas al usuario de las mismas..			
Precondiciones:		Correcta autenticación del usuario. Tener al menos un cliente registrado y habilitado dentro del prototipo	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos

Se ingresa a la opción Cuentas por Cobrar en el menú principal	_____	El prototipo muestra la pantalla de cuentas por cobrar, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de cuentas por cobrar, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar en la tabla	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Fecha Emisión: 09/03/2019 Cliente: Instituto Nacional de Aprendizaje Tipo de documento: Factura Descripción: Capacitación. Vencimiento: 09/09/2019 Monto: 250,000 Estado: Abierto	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	Fecha Emisión: asfsafasf Cliente: _____ Tipo de documento: ____ Descripción: _____ Vencimiento: asfsafasf Monto: -250,000 Estado: Abierto	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	N° Documento: _____ Fecha Emisión: 09/03/2019 Cliente: Instituto Nacional de Aprendizaje	El prototipo muestra los registros que coinciden con los	El prototipo muestra los registros que coinciden con los

	<p>Tipo de documento: Factura</p> <p>Descripción: Capacitación.</p> <p>Vencimiento: 09/09/2019</p> <p>Monto: 250,000</p> <p>Estado: Abierto</p>	datos ingresados.	datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	<p>N° Documento: _____</p> <p>Fecha Emisión: 20/10/2019</p> <p>Cliente: Instituto Nacional de Seguros</p> <p>Tipo de documento: Contable</p> <p>Descripción: _____</p> <p>Vencimiento: _____</p> <p>Monto: asfsafasf</p> <p>Estado: _____</p>	El prototipo muestra un mensaje diciendo que no se encontraron registros.	El prototipo muestra un mensaje diciendo que no se encontraron registros.
El usuario deja en blanco los campos de la fila de filtrado.	<p>N° Documento: _____</p> <p>Fecha Emisión: _____</p> <p>Cliente: _____</p> <p>Tipo de documento: _____</p> <p>Descripción: _____</p> <p>Vencimiento: _____</p> <p>Monto: _____</p> <p>Estado: _____</p>	El prototipo realiza una consulta general de los registros de cuentas por cobrar.	El prototipo realiza una consulta general de los registros de cuentas por cobrar.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información

		anterior.	anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Fecha Emisión: 15/02/2019 Cliente: Instituto Nacional de Aprendizaje Tipo de documento: Factura Descripción: Capacitación. Vencimiento: 09/08/2019 Monto: 253,200 Estado: Cerrado	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no válidos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición	Fecha Emisión: gxcs Cliente: _____ Tipo de documento: ____ Descripción: _____ Vencimiento: cxvge Monto: -253,200 Estado: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click	_____	El prototipo	El prototipo

al botón de deshabilitar registro		manda un mensaje de confirmación para deshabilitar el registro.	manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.	_____	El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Activado.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.	El prototipo refresca la tabla y muestra los registros habilitados y deshabilitados.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Desactivado.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.
El usuario recarga la página o navega a cualquier otra página dentro del prototipo	_____	El prototipo en la barra superior mostrará una campana con la cantidad de alertas que haya, donde se informa al usuario si una o varias cuentas a cobrar están a un mes,	El prototipo en la barra superior mostrará una campana con la cantidad de alertas que haya, donde se informa al usuario í una o varias cuentas a cobrar están a un mes,

		dos semanas, una semana de vencer o si éstas ya están vencidas.	dos semanas, una semana de vencer o si éstas ya están vencidas.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 55. Caso de prueba cuentas por pagar.

Flujos			
ID Caso de Prueba: 09		Módulo a probar: Módulo de Cuentas por Pagar.	
Autor: Steven Morera Hernández		Fecha de elaboración: 09/03/2019	
Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar, editar y deshabilitar las cuentas por pagar, así como las alertas al usuario de las mismas..			
Precondiciones:		Correcta autenticación del usuario. Tener al menos un proveedor registrado y habilitado dentro del prototipo	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Cuentas por Pagar en el menú principal	_____	El prototipo muestra la pantalla de cuentas por pagar, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de cuentas por pagar, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar en la tabla	_____	El prototipo reemplaza la fila de filtrado por la	El prototipo reemplaza la fila de filtrado por la

		verde de inserción de registros.	verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Fecha Emisión: 05/01/2019 Proveedor: Tracopa Tipo de documento: Factura Descripción: Pago viaje. Vencimiento: 25/01/2019 Monto: 320,000 Estado: Abierto	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	Fecha Emisión: asfsafsf Cliente: _____ Tipo de documento: ____ Descripción: _____ Vencimiento: asfsafsf Monto: -320,000 Estado: Abierto	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	N° Documento: _____ Fecha Emisión: 05/01/2019 Proveedor: Tracopa Tipo de documento: Factura Descripción: Pago viaje. Vencimiento: 25/01/2019 Monto: 320,000 Estado: Abierto	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	N° Documento: _____ Fecha Emisión: asdasfasf Proveedor: Panini Tipo de documento: _____ Descripción: asdasdw. Vencimiento: cvxcbfd	El prototipo muestra un mensaje diciendo que no se encontraron registros.	El prototipo muestra un mensaje diciendo que no se encontraron registros.

	Monto: 320,000 Estado: Cerrado		
El usuario deja en blanco los campos de la fila de filtrado.	N° Documento: _____ Fecha Emisión: _____ Proveedor: _____ Tipo de documento: _____ Descripción: _____ Vencimiento: _____ Monto: _____ Estado: _____	El prototipo realiza una consulta general de los registros de cuentas por pagar.	El prototipo realiza una consulta general de los registros de cuentas por pagar.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Fecha Emisión: 05/01/2019 Proveedor: Tracopa Tipo de documento: Factura Descripción: Pago viaje. Vencimiento: 25/01/2019 Monto: 320,000 Estado: Cerrado.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando	Fecha Emisión: asfasf Proveedor: Tracopa Tipo de documento: _____	El prototipo muestra un mensaje de error	El prototipo muestra un mensaje de error

formatos no válidos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición	Descripción: _____ Vencimiento: asfsafa Monto: 320,000 Estado: _____	correspondiente al error realizado en cada uno de los campos y refresca la tabla.	correspondiente al error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de deshabilitar registro	_____	El prototipo manda un mensaje de confirmación para deshabilitar el registro.	El prototipo manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.	_____	El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Activado.	El prototipo refresca la tabla y muestra los registros	El prototipo refresca la tabla y muestra los registros

		habilitados y deshabilitados.	habilitados y deshabilitados.
El usuario activa el interruptor de deshabilitados	Deshabilitados: Desactivado.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.	El prototipo refresca la tabla y muestra los registros habilitados únicamente.
El usuario recarga la página o navega a cualquier otra página dentro del prototipo	_____	El prototipo en la barra superior mostrará una campana con la cantidad de alertas que haya, donde se informa al usuario si una o varias cuentas por pagar están a un mes, dos semanas, una semana de vencer o si éstas ya están vencidas.	El prototipo en la barra superior mostrará una campana con la cantidad de alertas que haya, donde se informa al usuario si una o varias cuentas a pagar están a un mes, dos semanas, una semana de vencer o si éstas ya están vencidas.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 56. Caso de prueba gastos operativos

Gastos Operativos	
ID Caso de Prueba: 10	Módulo a probar: Módulo de Cálculos Financieros.
Autor: Steven Morera Hernández	Fecha de elaboración: 9/3/2019

Descripción			
Este proceso de prueba verificará el proceso de consultar, insertar, editar y eliminar los gastos operativos de la cooperativa, así como los cálculos automatizados dentro de los mismos.			
Precondiciones:		Correcta autenticación del usuario.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Gastos Operativos en el menú principal	_____	El prototipo muestra la pantalla de gastos operativos del primer proyecto disponible, con una tabla con los registros disponibles, así como el cálculo de la totalidad de los registros y de cada categoría.	El prototipo muestra la pantalla de gastos operativos del primer proyecto disponible, con una tabla con los registros disponibles, así como el cálculo de la totalidad de los registros y de cada categoría.
El usuario presiona el botón de ingresar de la tabla.	_____	El prototipo reemplaza la fila de filtrado de la tabla por una verde de inserción de registros.	El prototipo reemplaza la fila de filtrado de la tabla por una verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click	Actividad: Alquiler Detalle: _____ Enero: 250,000 Febrero: 250,000	El prototipo ingresa los datos dados por el usuario calculando	El prototipo ingresa los datos dados por el usuario calculando

al botón ingresar en la tabla.	Marzo: 250,000 Abril: 250,000 Mayo: 250,000 Junio: 250,000 Julio: 250,000 Agosto: 250,000 Septiembre: 0 Octubre: 0 Noviembre: 0 Diciembre: 0 Categoría: Básico	el total de gasto mensual, donde posteriormente refresca la tabla de gastos operativos calculando la totalidad de cada uno de los registros mostrados.	el total de gasto mensual, donde posteriormente refresca la tabla de gastos operativos calculando la totalidad de cada uno de los registros mostrados.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar en la tabla.	Actividad: Alquiler Detalle: ____ Enero: 250,000 Febrero: 250,000 Marzo: 250,000 Abril: ____ Mayo: ____ Junio: -250,000 Julio: -250,000 Agosto: ____ Septiembre: 0 Octubre: 0 Noviembre: 0 Diciembre: ____ Categoría: ____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado en la tabla.	Código Gasto Operativo: _____ Actividad: Alquiler Detalle: ____ Total Gastos: 2,000,000 Enero: 250,000	El prototipo muestra los registros que coinciden con los datos ingresados calculando la	El prototipo muestra los registros que coinciden con los datos ingresados calculando la

	Febrero: 250,000 Marzo: 250,000 Abril: 250,000 Mayo: 250,000 Junio: 250,000 Julio: 250,000 Agosto: 250,000 Septiembre: 0 Octubre: 0 Noviembre: 0 Diciembre: 0 Categoría: Básico	totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.	totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.
El usuario digita datos que no coinciden con los registros en la fila de filtrado en la tabla.	Código Gasto Operativo: _____ Actividad: sasfasqa Detalle: qwrq Total Gastos: -2,000 Enero: _____ Febrero: 250 Marzo: 452,235 Abril: 3,542 Mayo: _____ Junio: _____ Julio: 250,000 Agosto: 250,000 Septiembre: 0 Octubre: 0 Noviembre: 0 Diciembre: 0 Categoría: Otro	El prototipo muestra un mensaje indicando que no se encontraron registros.	El prototipo muestra un mensaje indicando que no se encontraron registros.
El usuario deja en blanco los campos	Código Gasto Operativo: _____	El prototipo realiza una	El prototipo realiza una

de la fila de filtrado de la tabla.	Actividad: _____ Detalle: _____ Total Gastos: _____ Enero: _____ Febrero: _____ Marzo: _____ Abril: _____ Mayo: _____ Junio: _____ Julio: _____ Agosto: _____ Septiembre: _____ Octubre: _____ Noviembre: _____ Diciembre: _____ Categoría: _____	consulta general de los registros calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.	consulta general de los registros calculando la totalidad de cada uno de los registros mostrados, así como un total de la categoría al que pertenezcan los registros.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición de la tabla.	Actividad: Alquiler Detalle: Pago por edificio Enero: 250,000 Febrero: 250,000 Marzo: 250,000 Abril: 250,000 Mayo: 250,000	El prototipo modifica los campos editados dados por el usuario dentro de la tabla calculando el total de gastos	El prototipo modifica los campos editados dados por el usuario dentro de la tabla calculando el total de gastos

	<p>Junio: 250,000 Julio: 250,000 Agosto: 250,000 Septiembre: 250,000 Octubre: 250,000 Noviembre: 250,000 Diciembre: 250,000 Categoría: Básico</p>	<p>operativos mensual, donde posteriormente refresca la tabla para calcular la totalidad de cada uno de los registros mostrados, así como un total por cada categoría.</p>	<p>operativos mensual, donde posteriormente refresca la tabla para calcular la totalidad de cada uno de los registros mostrados, así como un total por cada categoría.</p>
<p>El usuario edita los campos del registro ingresando formatos no validos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición de la tabla.</p>	<p>Actividad: _____ Detalle: Pago por edificio Enero: ____ Febrero: -250,000 Marzo: 250,000 Abril: -250,000 Mayo: 250,000 Junio: qwrwqrq Julio: -250,000 Agosto: asfsafasf Septiembre: 250,000 Octubre: _____ Noviembre: _____ Diciembre: qwrqwr Categoría: _____</p>	<p>El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.</p>	<p>El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.</p>
<p>El usuario da click al botón de cancelar edición de la tabla.</p>	<p>_____</p>	<p>El prototipo revierte el cambio de la fila de edición del registro, al predeterminado</p>	<p>El prototipo revierte el cambio de la fila de edición del registro, al predeterminado</p>

		restableciendo los valores los campos a los anteriores al cambio.	restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar registro de la tabla.	_____	El prototipo manda un mensaje de confirmación para eliminar el registro.	El prototipo manda un mensaje de confirmación para eliminar el registro.
El usuario interactúa con el mensaje de confirmación para eliminar el registro de la tabla.	_____	El prototipo elimina el registro y refresca la tabla.	El prototipo elimina el registro y refresca la tabla.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 57. Caso de prueba historial de transferencias.

Flujos	
ID Caso de Prueba: 11	Módulo a probar: Módulo de Excedentes.
Autor: Steven Morera Hernández	Fecha de elaboración: 09/03/2019
Descripción	
Este caso de prueba verifica los procesos de consultar, insertar, editar y eliminar el historial de transferencias	
Precondiciones:	Correcta autenticación del usuario. Tener al menos un proyecto, asociado o colaborador registrado y habilitado dentro del prototipo

Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Excedentes – Historial de Transferencias en el menú principal	_____	El prototipo muestra la pantalla de historial de transferencias, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de historial de transferencias, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar en la tabla	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Proyecto: TIC-as Persona: Shirley Salazar Rojas Fecha: 04/03/2019 N° Cuenta Envió: 21548354831854864561 N° Cuenta Recibe: 1122003355446677 Monto: 250,000 Descripción: Pago mensual.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click	Proyecto: ____ Persona: Shirley Salazar Rojas Fecha: fsafasfas N° Cuenta Envió:	El prototipo muestra un mensaje de error correspondiente al error realizado en	El prototipo muestra un mensaje de error correspondiente al error realizado en

	Monto: _____ Estado: _____		
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Proyecto: TIC-as Persona: Shirley Salazar Rojas Fecha: 04/03/2019 N° Cuenta Envió: 21548354831854864561 N° Cuenta Recibe: 1122003355446677 Monto: 320,000 Descripción: Pago mensual del periodo 2018.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no validos o dejando los campos obligatorios vacíos y da click al botón de confirmar	Proyecto: _____ Persona: Shirley Salazar Rojas Fecha: 04/03/2019 N° Cuenta Envió: _____ N° Cuenta Recibe: 1122003355446677 Monto: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.

edición	Descripción: _____		
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar registro	_____	El prototipo manda un mensaje de confirmación para eliminar el registro.	El prototipo manda un mensaje de confirmación para eliminar el registro.
El usuario interactúa con el mensaje de confirmación para eliminar el registro.	_____	El prototipo elimina el registro y refresca la tabla.	El prototipo elimina el registro y refresca la tabla.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 58. Caso de prueba asignar proyecto

Asignar proyectos			
ID Caso de Prueba: 12		Módulo a probar: Módulo de Excedentes.	
Autor: Steven Morera Hernández		Fecha de elaboración: 09/03/2019	
Descripción			
Este caso de prueba se verifican los procesos de consultar, insertar, editar y eliminar registros de asignación asociados, asociadas, colaborador o colaboradora a un proyecto, como el registro de excedentes de los mismos			
Precondiciones:		Correcta autenticación del usuario. Tener al menos un proyecto, asociado o colaborador registrado y habilitado dentro del prototipo	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Excedentes – Asignaciones en el menú principal	_____	El prototipo muestra la pantalla de asignaciones, donde dos tablas cargan con los registros disponibles de asociado a proyecto y colaborador a proyecto,.	El prototipo muestra la pantalla de asignaciones, donde dos tablas cargan con los registros disponibles de asociado a proyecto y colaborador a proyecto,
El usuario presiona el botón de ingresar en cualquiera de las tablas.	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.

El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar de la tabla asociado – proyecto.	Proyecto: TIC-as Asociado: Adriana Ulate Ramírez.	El prototipo ingresa los datos dados por el usuario, ingresa el excedente correspondiente a los periodos del proyecto y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario, ingresa el excedente correspondiente a los periodos del proyecto y se refresca con el nuevo registro.
El usuario deja campos obligatorios en blanco y da click al botón ingresar de la tabla asociado – proyecto.	Proyecto: _____ Asociado: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado de la tabla asociado – proyecto.	Proyecto: TIC-as. Asociado: Adriana Ulate Ramírez	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado de la tabla asociado – proyecto.	Proyecto: TIC-as. Asociado: Steven Salas Cajina.	El prototipo muestra un mensaje diciendo que no se encontraron registros.	El prototipo muestra un mensaje diciendo que no se encontraron registros.
El usuario deja en blanco los campos	Proyecto: _____ Asociado: _____	El prototipo realiza una	El prototipo realiza una

de la fila de filtrado de la tabla asociado – proyecto.		consulta general de los registros de asociado - proyecto.	consulta general de los registros de asociado - proyecto.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar de la tabla colaborador – proyecto.	Proyecto: TIC-as Colaborador: Fernanda Molina Vega Jornada: Media Monto de Jornada: 25.000 Fecha Inicio: 01/01/2019 Fecha Final: 01/06/2019	El prototipo ingresa los datos dados por el usuario, ingresa el excedente correspondiente a la fecha de inicio y final, luego se refresca la tabla con el nuevo registro.	l prototipo ingresa los datos dados por el usuario, ingresa el excedente correspondiente a la fecha de inicio y final, luego se refresca la tabla con el nuevo registro..
El usuario ingresa datos erróneos o deja campos obligatorios en blanco y da click al botón ingresar de la tabla asociado – proyecto.	Proyecto: TIC-as Colaborador: _____ Jornada: Media Monto de Jornada: -25.000 Fecha Inicio: asfasfa Fecha Final: asfasfs	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado de la tabla colaborador – proyecto.	Proyecto: TIC-as Colaborador: Fernanda Molina Vega Jornada: _____ Monto de Jornada: 25.000 Fecha Inicio: 01/01/2019 Fecha Final: 01/05/2019	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita	Proyecto: asfsafa	El prototipo	El prototipo

datos que no coinciden con los registros en la fila de filtrado de la tabla colaborador – proyecto.	Colaborador: _____ Jornada: Tres cuartos. Monto de Jornada: -25.000 Fecha Inicio: 01/06/2019 Fecha Final: 01/05/2019	muestra un mensaje diciendo que no se encontraron registros.	muestra un mensaje diciendo que no se encontraron registros.
El usuario deja en blanco los campos de la fila de filtrado de la tabla colaborador – proyecto.	Proyecto: _____ Colaborador: _____ Jornada: _____ Monto de Jornada: _____ Fecha Inicio: _____ Fecha Final: _____	El prototipo realiza una consulta general de los registros de colaborador - proyecto.	El prototipo realiza una consulta general de los registros de colaborador - proyecto.
El usuario da click a botón de editar o a un registro dentro de la tabla colaborador - proyecto.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición dentro de la tabla colaborador - proyecto.	Jornada: Completa Monto de Jornada: 55,000 Fecha Final: 01/05/2019	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los	Jornada: _____	El prototipo	El prototipo

campos del registro ingresando formatos no válidos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición en la tabla colaborador - proyecto	Monto de Jornada: asfasfas Fecha Final: gyfbh	muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.	muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de cancelar edición en la tabla colaborador proyecto	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar registro	_____	El prototipo manda un mensaje de confirmación para eliminar el registro.	El prototipo manda un mensaje de confirmación para eliminar el registro.
El usuario interactúa con el mensaje de confirmación para eliminar el registro.	_____	El prototipo elimina los excedentes asociados al registro, el registro de asignación y	El prototipo elimina los excedentes asociados al registro, el registro de asignación y

		refresca la tabla.	refresca la tabla.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 59. Caso de prueba excedentes

Excedentes			
ID Caso de Prueba: 13		Módulo a probar: Módulo de Excedentes.	
Autor: Steven Morera Hernández		Fecha de elaboración: 09/03/2019	
Descripción			
Este caso de prueba se verifican los procesos de consultar, insertar, editar y eliminar registros de asignación asociados, asociadas, colaborador o colaboradora a un proyecto, como el registro de excedentes de los mismos			
Precondiciones:		Correcta autenticación del usuario. Debe haber al menos una asignación registrada dentro del prototipo.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Excedentes – Mes y Anual en el menú principal	_____	El prototipo muestra la pantalla de excedentes, donde una tabla carga con los registros disponibles del primer periodo encontrado donde se calcula el total de excedente	El prototipo muestra la pantalla de excedentes, donde una tabla carga con los registros disponibles del primer periodo encontrado donde se calcula el total de excedente

		mensual para determinar si es menor, mayor o igual valor al excedente anual.	mensual para determinar si es menor, mayor o igual valor al excedente anual.
El usuario digita los datos en la fila de filtrado de la tabla.	Persona: Adriana Ulate Ramírez. Anual: 0 Enero: 0 Febrero: 0 Marzo: 0 Abril: 0 Mayo: 0 Junio: 0 Julio: 0 Agosto: 0 Septiembre: 0 Octubre: 0 Noviembre 0: Diciembre: 0	El prototipo muestra los registros que coinciden con los datos ingresados y se calcula el total de excedente mensual para determinar si es menor, mayor o igual valor al excedente anual.	El prototipo muestra los registros que coinciden con los datos ingresados y se calcula el total de excedente mensual para determinar si es menor, mayor o igual valor al excedente anual.
El usuario digita datos que no coinciden con los registros en la fila de filtrado.	Persona: Adriana Ulate Ramírez. Anual: 500,000 Enero: 0 Febrero: -20,000 Marzo: aaojsf Abril: 0 Mayo: 0 Junio: 0 Julio: 0 Agosto: 0	El prototipo muestra un mensaje indicando que no se encontraron registros.	El prototipo muestra un mensaje indicando que no se encontraron registros.

	<p>Septiembre: 0</p> <p>Octubre: 0</p> <p>Noviembre 0:</p> <p>Diciembre: 0</p> <p>.</p>		
<p>El usuario deja en blanco los campos de la fila de filtrado.</p>	<p>Persona: _____</p> <p>Anual: _____</p> <p>Enero: _____</p> <p>Febrero: _____</p> <p>Marzo: _____</p> <p>Abril: _____</p> <p>Mayo: _____</p> <p>Junio: _____</p> <p>Julio: _____</p> <p>Agosto: _____</p> <p>Septiembre: _____</p> <p>Octubre: _____</p> <p>Noviembre _____</p> <p>Diciembre: _____</p>	<p>El prototipo realiza una consulta general de los registros de excedentes, donde se calcula el total de excedente mensual para determinar si es menor, mayor o igual valor al excedente anual.</p>	<p>El prototipo realiza una consulta general de los registros de excedentes, donde se calcula el total de excedente mensual para determinar si es menor, mayor o igual valor al excedente anual.</p>
<p>El usuario da click a un registro dentro de la tabla de excedentes</p>	<p>_____</p>	<p>El prototipo oscurece la pantalla y despliega un panel con la tabla de excedente persona, donde se muestra el detalle de los diferentes excedentes que conforman el</p>	<p>El prototipo oscurece la pantalla y despliega un panel con la tabla de excedente persona, donde se muestra el detalle de los diferentes excedentes que conforman el</p>

		excedente general, de la persona del registro.	excedente general, de la persona del registro.
El usuario digita los datos en la fila de filtrado en el panel de excedente persona.	Código: ____ Proyecto: TIC-as Anual: 0 Enero: 0 Febrero: 0 Marzo: 0 Abril: 0 Mayo: 0 Junio: 0 Julio: 0 Agosto: 0 Septiembre: 0 Octubre: 0 Noviembre 0 Diciembre: 0	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado en el panel de excedente persona.	Código: ____ Proyecto: TIC-as Anual: 200,000 Enero: -50,000 Febrero: 0 Marzo: 0 Abril: 0 Mayo: 30 Junio: 500 Julio: 0 Agosto: 600 Septiembre: 0 Octubre: 710 Noviembre 0	El prototipo muestra un mensaje señalando que no se encontraron registros.	El prototipo muestra un mensaje señalando que no se encontraron registros.

	Diciembre: 0		
El usuario deja en blanco los campos de la fila de filtrado en el panel de excedente persona.	Código: ____ Proyecto: ____ Anual: ____ Enero: ____ Febrero: ____ Marzo: ____ Abril: ____ Mayo: ____ Junio: ____ Julio: ____ Agosto: ____ Septiembre: ____ Octubre: ____ Noviembre ____ Diciembre: ____	El prototipo realiza una consulta general de los registros de excedentes que correspondan a la persona del registro seleccionado previamente.	El prototipo realiza una consulta general de los registros de excedentes que correspondan a la persona del registro seleccionado previamente.
El usuario da click a botón de editar o a un registro dentro del panel excedente persona.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición dentro del panel excedente persona.	Anual: 520,000 Enero: 85,000 Febrero: 85,000 Marzo: 80,000 Abril: 60,000 Mayo: 70,000 Junio: 0	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se

	<p>Julio: 0</p> <p>Agosto: 0</p> <p>Septiembre: 0</p> <p>Octubre: 0</p> <p>Noviembre 0</p> <p>Diciembre: 0</p>	refresca con los cambios realizados.	refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no válidos o dejando los campos obligatorios vacíos y da click al botón de confirmar edición en el panel excedente persona.	<p>Anual: -520,000</p> <p>Enero: asfaf</p> <p>Febrero: asfaf00</p> <p>Marzo: 80,000</p> <p>Abril: asfaf</p> <p>Mayo: 70,000</p> <p>Junio: asfaf</p> <p>Julio: 0</p> <p>Agosto: _____</p> <p>Septiembre: 0</p> <p>Octubre: 0</p> <p>Noviembre 0</p> <p>Diciembre: 0</p>	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.
El usuario da click al botón de cancelar edición en el panel excedente persona.	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar registro	_____	El prototipo manda un mensaje de confirmación	El prototipo manda un mensaje de confirmación

en el panel excedente persona		para eliminar el registro.	para eliminar el registro.
El usuario interactúa con el mensaje de confirmación para eliminar el registro de excedente persona	_____	El prototipo elimina el registro y refresca la tabla.	El prototipo elimina el registro y refresca la tabla.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 60. Caso de prueba usuarios

Usuarios			
ID Caso de Prueba: 14		Módulo a probar: Módulo de Seguridad.	
Autor: Steven Morera Hernández		Fecha de elaboración: 09/03/2019	
Descripción			
Este caso de prueba verifica los procesos de consultar, insertar, editar y eliminar usuarios.			
Precondiciones:		Correcta autenticación del usuario administrador. Debe haber al menos un asociado registrado y habilitado dentro del prototipo.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Configuración – Usuarios en el menú principal	_____	El prototipo muestra la pantalla de usuarios, donde una tabla es cargada con los registros	El prototipo muestra la pantalla de usuarios, donde una tabla es cargada con los registros

		disponibles.	disponibles.
El usuario presiona el botón de ingresar en la tabla	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.
El usuario digita los datos para ingresar un nuevo registro y da click al botón ingresar	Usuario: shirley@sulabatsu.com Contraseña: ssalazar Asociado: Shirley Salazar Rojas.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.	El prototipo ingresa los datos dados por el usuario y se refresca con el nuevo registro.
El usuario ingresa datos incorrectos o deja campos obligatorios en blanco y da click al botón ingresar	Usuario: Shirley Contraseña: sd!\$%1das\$” Asociado: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos.
El usuario digita los datos en la fila de filtrado.	Usuario: shirley@sulabatsu.com Contraseña: ssalazar Asociado: Shirley Salazar Rojas.	El prototipo muestra los registros que coinciden con los datos ingresados.	El prototipo muestra los registros que coinciden con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	Usuario: shirley@gmail.com Contraseña: fq24151 Asociado: Shirley Salazar Rojas.	El prototipo muestra un mensaje señalando que no se encontraron registros.	El prototipo muestra un mensaje señalando que no se encontraron registros.
El usuario deja en	Usuario: ____	El prototipo	El prototipo

blanco los campos de la fila de filtrado.	Contraseña: _____ Asociado: _____	realiza una consulta general de los registros de usuarios.	realiza una consulta general de los registros de usuarios.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita los campos del registro y da click al botón de confirmar edición.	Usuario: shirley@sulabatsu.com Contraseña: sh1rl3ysalazar Asociado: Shirley Salazar Rojas.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.	El prototipo modifica los campos editados dados por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios realizados.
El usuario edita los campos del registro ingresando formatos no válidos o dejando los campos obligatorios vacíos y da click al botón de confirmar	Usuario: shirley@sulabatsu.com Contraseña: sh1rl3y.salazar Asociado: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.	El prototipo muestra un mensaje de error correspondiente al error realizado en cada uno de los campos y refresca la tabla.

edición			
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de deshabilitar el registro	_____	El prototipo manda un mensaje de confirmación para deshabilitar el registro.	El prototipo manda un mensaje de confirmación para deshabilitar el registro.
El usuario interactúa con el mensaje de confirmación para deshabilitar el registro.	_____	El prototipo deshabilita el registro y refresca la tabla.	El prototipo deshabilita el registro y refresca la tabla.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 61. Caso de prueba roles

Roles	
ID Caso de Prueba: 14	Módulo a probar: Módulo de Seguridad.
Autor: Steven Morera Hernández	Fecha de elaboración: 09/03/2019

Descripción			
Este caso de prueba verifica los procesos de consultar, insertar, editar y eliminar roles.			
Precondiciones:		Correcta autenticación del usuario administrador.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Configuración – Roles en el menú principal	_____	El prototipo muestra la pantalla de roles, donde una tabla es cargada con los registros disponibles.	El prototipo muestra la pantalla de roles, donde una tabla es cargada con los registros disponibles.
El usuario presiona el botón de ingresar en la tabla	_____	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.	El prototipo reemplaza la fila de filtrado por la verde de inserción de registros.
El usuario digita el dato para ingresar un nuevo registro y da click al botón ingresar	Descripción: Administrador	El prototipo ingresa el dato dado por el usuario y se refresca con el nuevo registro.	El prototipo ingresa el dato dado por el usuario y se refresca con el nuevo registro.
El usuario deja el campo obligatorio en blanco y da click al botón ingresar	Descripción: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en el campo.	El prototipo muestra un mensaje de error correspondiente al error realizado en el campo.
El usuario digita	Código: _____	El prototipo	El prototipo

los datos en la fila de filtrado.	Descripción: Administrador	muestra el registro que coincide con los datos ingresados.	muestra el registro que coincide con los datos ingresados.
El usuario digita datos que no coinciden con los registros en la fila de filtrado	Código: _____ Descripción: Contador	El prototipo muestra un mensaje diciendo que no se encontraron registros.	El prototipo muestra un mensaje diciendo que no se encontraron registros.
El usuario deja en blanco los campos de la fila de filtrado.	Código: _____ Descripción: _____	El prototipo realiza una consulta general de los registros de usuarios.	El prototipo realiza una consulta general de los registros de usuarios.
El usuario da click a botón de editar o a un registro dentro de la tabla.	_____	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.	El prototipo reemplaza la fila del registro por uno amarillo claro, el cual posee distintos campos editables con la información anterior.
El usuario edita el campo del registro y da click al botón de confirmar edición.	Descripción: Contador	El prototipo modifica el campo editado dado por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios	El prototipo modifica el campo editado dado por el usuario dentro de la tabla, la cual posteriormente se refresca con los cambios

		realizados.	realizados.
El usuario edita el campo del registro dejándolo vacío y da click al botón de confirmar edición	Descripción: _____	El prototipo muestra un mensaje de error correspondiente al error realizado en el campo y refresca la tabla.	El prototipo muestra un mensaje de error correspondiente al error realizado en el campo y refresca la tabla.
El usuario da click al botón de cancelar edición	_____	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.	El prototipo revierte el cambio de la fila de edición del registro, al predeterminado restableciendo los valores los campos a los anteriores al cambio.
El usuario da click al botón de eliminar el registro	_____	El prototipo manda un mensaje de confirmación para eliminar el registro.	El prototipo manda un mensaje de confirmación para eliminar el registro.
El usuario interactúa con el mensaje de confirmación para eliminar el registro.	_____	El prototipo eliminar el registro y refresca la tabla.	El prototipo eliminar el registro y refresca la tabla.
Estado		Correcto.	
Fin de caso de prueba.			

Fuente: Elaboración propia.

Tabla 62. Caso de prueba reportes

Reportes			
ID Caso de Prueba: 15		Módulo a probar: Módulo de Reportes.	
Autor: Steven Morera Hernández		Fecha de elaboración: 09/03/2019	
Descripción			
Este caso de prueba verifica proceso de generar reportes.			
Precondiciones:		Correcta autenticación del usuario administrador.	
Ejecución del caso de prueba			
Pasos	Datos	Resultados Esperados	Resultados Obtenidos
Se ingresa a la opción Reportes – Cuentas Cobrar y Pagar en el menú principal.	_____	El prototipo muestra la pantalla de reportes, dos formularios son cargados con un botón “Generar PDF” cada uno.	El prototipo muestra la pantalla de reportes, dos formularios son cargados con un botón “Generar PDF” cada uno.
El usuario no cambia ningún valor del formulario de cuentas por pagar y da click al botón de “Generar PDF”.	Estado de cuentas: Abierto ¿Incluir deshabilitados?: No	El prototipo redirige al usuario a una nueva ventana donde está el PDF autogenerado de cuentas por pagar, con los parámetros predeterminados.	El prototipo redirige al usuario a una nueva ventana donde está el PDF autogenerado de cuentas por pagar, con los parámetros predeterminados.
El usuario cambia los parámetros del formulario de cuentas por pagar y da click al botón	Estado de cuentas: Cerrado ¿Incluir deshabilitados?: Si	El prototipo redirige al usuario a una nueva ventana donde está el PDF	El prototipo redirige al usuario a una nueva ventana donde está el PDF

de “Generar PDF”		autogenerado de cuentas por pagar, con los nuevos parámetros seleccionados por el usuario.	autogenerado de cuentas por pagar, con los nuevos parámetros seleccionados por el usuario.
El usuario cambia los parámetros del formulario de cuentas por pagar, los cuales no tienen ningún registro relacionado y da click al botón de “Generar PDF”	_____	El prototipo envía un mensaje al usuario señalando que el reporte de cuentas por pagar está vacío.	El prototipo envía un mensaje al usuario señalando que el reporte de cuentas por pagar está vacío.
El usuario no cambia ningún valor del formulario de cuentas por cobrar y da click al botón de “Generar PDF”.	Estado de cuentas: Abierto ¿Incluir deshabilitados?: No	El prototipo redirige al usuario a una nueva ventana donde está el PDF autogenerado de cuentas por cobrar, con los parámetros predeterminados.	El prototipo redirige al usuario a una nueva ventana donde está el PDF autogenerado de cuentas por cobrar, con los parámetros predeterminados.
El usuario cambia los parámetros del formulario de cuentas por cobrar y da click al botón de “Generar PDF”	Estado de cuentas: Cerrado ¿Incluir deshabilitados?: Si	El prototipo redirige al usuario a una nueva ventana donde está el PDF autogenerado de	El prototipo redirige al usuario a una nueva ventana donde está el PDF autogenerado de

		cuentas por cobrar, con los nuevos parámetros seleccionados por el usuario.	cuentas por cobrar, con los nuevos parámetros seleccionados por el usuario.
El usuario cambia los parámetros del formulario de cuentas por cobrar, los cuales no tienen ningún registro relacionado y da click al botón de "Generar PDF"		El prototipo envía un mensaje al usuario señalando que el reporte de cuentas por cobrar esta vacío.	El prototipo envía un mensaje al usuario señalando que el reporte de cuentas por cobrar esta vacío.
Estado	Correcto.		
Fin de caso de prueba.			

Fuente: Elaboración propia.

REFERENCIAS

- Albarrán, J (2014) Técnica contable. [Versión PDF] Recuperado de: <http://ebookcentral.proquest.com/lib/bibliouiasp/detail.action?docID=3229702>
- Bravent (2017) La importancia de un buen diseño del software. Recuperado de: <https://www.bravent.net/la-importancia-de-un-buen-diseno-del-software>
- Canchala, A. (2008) Fundamentos de la poo. Recuperado de: <https://msdn.microsoft.com/es-es/library/bb972232.aspx>
- Carmona, G (2011) Aplicaciones informáticas de bases de datos relacionales (UF0322). [Versión PDF]. Recuperado de: <http://ebookcentral.proquest.com/lib/bibliouiasp/detail.action?docID=3211898>.
- Debitoor. (2017) ¿Qué son las cuentas a pagar? Recuperado de: <https://debitoor.es/glosario/definicion-de-cuentas-pagar>
- Duplika (2013) ¿Qué es el hosting?. Recuperado de: <https://duplika.com/que-es-el-hosting/>
- Yoelvis, O (2013) Archivo: foto de sistema operativo.jpg. Recuperado de: https://www.ecured.cu/Archivo:Foto_de_sistema_operativo.JPG
- Ernesto, S y Luis, S (2013) Metodología de la investigación. [Versión PDF] Recuperado de: <http://ebookcentral.proquest.com/lib/bibliouiasp/detail.action?docID=3216667>
- González, E (s.f). ¿Qué es php? ¿para qué sirve php? un potente lenguaje de programación para crear páginas web (cu00803b). Recuperado de: <https://www.aprenderaprogramar.com/attachments/article/492/CU00803B%20Que%20es%20PHP%20para%20sirve%20potente%20lenguaje%20programacion%20paginas%20web.pdf>

Gutiérrez, Javier (s.f) ¿Qué es un framework web? [Versión PDF] Recuperado de:
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

Gutiérrez, José y Pagés C (2008) Planificación y gestión de proyectos informáticos [Versión PDF] Recuperado de:
<http://ebookcentral.proquest.com/lib/bibliouiasp/detail.action?docID=3176931>

Hernández, R, Fernández, C y Baptista, P (1997) Metodología de la investigación 5° edición. México: McGraw-Hill.

IBM Knowledge Center (s.f) ¿Qué es ajax? Recuperado de:
https://www.ibm.com/support/knowledgecenter/es/SS8PJ7_9.6.1/com.ibm.etools.webtoolscore.doc/topics/cajax.html

ICTEA (s.f). Base de conocimientos ¿qué es MySQL? Recuperado de:
<http://www.ictea.com/cs/index.php?rp=/knowledgebase/3500/Che-cosa-e-MySQL.html>

Iglesias, A (2017). Qué es un sla y para qué sirve. Recuperado de:
<http://www.ticbeat.com/tecnologias/que-es-un-sla-y-para-que-sirve/>

jQuery Foundation (s.f) jQuery. Recuperado de: <https://jquery.com/>

Maguiña, R. (2009) Semejanzas, diferencias y complementariedad de las perspectivas metodológicas cuantitativas y cualitativas y su aplicación a la investigación administrativa. [Versión PDF] Recuperado de:
<http://ebookcentral.proquest.com/lib/bibliouiasp/detail.action?docID=3182274>

MDN web docs et al (2013) Html. Recuperado de:
<https://developer.mozilla.org/es/docs/Web/HTML>

Medianero, D y Carmen, M (2016) Diseño de proyectos sociales. 3° Edición [Versión Web] Recuperado de: <https://sites.google.com/site/disenodeproyectossociales/capitulo-xii>

- Microsoft (s.f) Access sql: conceptos básicos, vocabulario y sintaxis. Recuperado de: <https://support.office.com/es-es/article/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671>
- Microsoft (2017): Tablas. Recuperado de: <https://docs.microsoft.com/es-es/sql/relational-databases/tables/tables?view=sql-server-2017>
- Ministerio de Trabajo (diciembre, 2017) Lista de salarios por ocupación. Página web del Ministerio de Trabajo de Costa Rica. [Versión PDF] Recuperado de: http://www.mtss.go.cr/temas-laborales/salarios/Documentos-Salarios/lista_ocupacion_2018.pdf
- Molina, F, Polo, E (2014) Servicios en red (648) [Versión PDF] Recuperado de: <http://ebookcentral.proquest.com/lib/biblioui/asp/detail.action?docID=3229687>
- Molina, J, Zea, M, Contenido, M y García, F (Marzo – Junio, 2018) Comparación De Metodologías en Aplicaciones Web. 3C Tecnología. [Versión PDF] Recuperado de: <https://doi.org/10.17993/3ctecno.2018.v7n1e25>
- Mota, A, Guevara, D, Jiménez, D, Hernández, L y Armas, Y (2014) Análisis y diseños de sistemas, instrumentos y herramientas de recolección de información. [Versión PDF] Recuperado de: <https://en.calameo.com/books/003734647cd87b2530691>
- Narváez, A (2017) Programación orientada a objetos en java - #1: creación de clases y objetos. Recuperado de: <https://www.youtube.com/watch?v=y7eyMeEZswg>
- Niño, V (2011) Metodología de la investigación: diseño y ejecución. [Versión PDF] Recuperado de: <http://ebookcentral.proquest.com/lib/biblioui/asp/detail.action?docID=3198784>

Norton, P, Fuentes, C, Esqueda, H y Cantón, M (2000). Introducción a la computación 6^o edición. México: McGraw-Hill.

Nuñez, E (abril, 2016) Diferencia entre gasto administrativo y gastos financiero. Recuperado de: <https://www.fundapymes.com/cual-es-la-diferencia-entre-un-gasto-administrativo-y-un-gasto-financiero/>

Ollé, C y Cerezuela, B (2017) Gestión de proyectos paso a paso. [Versión PDF] Recuperado de: <http://ebookcentral.proquest.com/lib/bibliouiasp/detail.action?docID=5214305>

Patel, M (2017) Top 7 web development frameworks. Recuperado de: <https://www.concettolabs.com/blog/top-7-web-development-frameworks/>

Pérez, D (2007) ¿Qué es javascript? Recuperado de: <http://www.maestrosdelweb.com/que-es-javascript/>

Ramírez, P (2017). Diseño web materialize css: ¿qué es y para qué se usa? Recuperado de: <https://fireosoft.com.co/blogs/materialize-css-se-usa/>

Senn, J (1992) Análisis y diseño de sistemas de información. [Versión PDF] Recuperado de: <http://ebookcentral.proquest.com/lib/bibliouiasp/detail.action?docID=3194990>

Shopify (s.f) Cuentas por cobrar. Recuperado de: <https://es.shopify.com/enciclopedia/cuentas-por-cobrar>

SUMMON PRESS, S.L (s.f) Definición de Excedente. Recuperado de: <https://www.economiasimple.net/glosario/excedente>.

Universia España (29 septiembre, 2015). ¿Qué es un prototipo y para qué sirve? Universia España. Recuperado de: <http://noticias.universia.es/consejos-profesionales/noticia/2015/09/29/1131645/prototipo-sirve.html>

APÉNDICES

Apéndice A: Guía de Entrevista.

En la aplicación de esta entrevista se busca identificar cómo se encuentra la cooperativa actualmente y, a su vez, la familiaridad que tiene la misma para el uso de tecnologías que serán utilizadas en la realización del prototipo. De antemano, se agradece toda la información brindada por la asociada y se hace constar que toda la información aquí brindada será de uso exclusivo para la realización del proyecto final de graduación.

1. ¿Anteriormente ha utilizado un sistema para la gestión de proyectos dentro de la cooperativa?
2. ¿Se encuentra familiarizada con el uso de navegadores web?
3. ¿Se han presentado errores a la hora de realizar reportes? Sí es así ¿Con que frecuencia se presentan?
4. ¿Cree usted que la automatización de los procesos le ayudará a ahorrar tiempo?
5. ¿Cuáles son los procesos más críticos o importantes que realiza, en la gestión de los proyectos?
6. ¿Cuántos asociados y asociadas pueden trabajar en un proyecto?
7. ¿Qué tipo de características poseen los proyectos?
8. ¿Utilizan contrataciones externas para la realización de proyectos? Sí es así ¿Cómo administran al personal extra?

Apéndice B: Guía de Observación.

Nombre del Observador:		
Puesto del Observado/a:		
Proceso a realizar:		
Datos utilizados		
		Comportamiento:
Paso a paso del proceso		
Resultado obtenido:		