



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

**“SISTEMA PARA EL CONTROL DE APORTACIONES Y PRÉSTAMOS
DE LA CAJA DE AHORROS DE LA EMPRESA TUBASEC C.A.”**

TRABAJO DE TITULACIÓN: PROYECTO TÉCNICO

Para optar al Grado Académico de:

INGENIERA EN SISTEMAS INFORMÁTICOS

AUTORA: DEYSI CAROLINA LATORRE GARCÉS

TUTORA: ING. NATALIA PATRICIA LAYEDRA LARREA, MG.

Riobamba-Ecuador

2018

©2018, Deysi Carolina Latorre Garcés

Se autoriza la reproducción total o parcial, confines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal de Trabajo de Titulación certifica que: El proyecto técnico: “SISTEMA PARA EL CONTROL DE APORTACIONES Y PRÉSTAMOS DE LA CAJA DE AHORROS DE LA EMPRESA TUBASEC C.A.”, de responsabilidad de la señorita Deysi Carolina Latorre Garcés, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE

FIRMA

FECHA

Dr. Julio Santillán C.

VICEDECANO DE LA FACULTAD

DE INFORMÁTICA Y

ELECTRÓNICA

Ing. Patricio Moreno C.

DIRECTOR DE LA ESCUELA

DE INGENIERIA EN SISTEMAS

Ing. Natalia Layedra L.

DIRECTOR DE TRABAJO

DE TITULACIÓN

Ing. Jorge Menéndez V.

MIEMBRO DEL TRIBUNAL

“Yo, Deysi Carolina Latorre Garcés, soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación, y el patrimonio intelectual del mismo a la Escuela Superior Politécnica de Chimborazo”

DEYSI CAROLINA LATORRE GARCÉS

DEDICATORIA

A Dios por darme las fuerzas necesarias para salir adelante y superar todas y cada una de las adversidades encontradas en el transcurso de mi vida, a mi hija Zoe por estar siempre a mi lado y darme su amor incondicional, por darme las fuerzas para no desmayar, por ser mi inspiración y el motivo para luchar cada día.

Deysi

AGRADECIMIENTO

Doy gracias a Dios por cuidarme, protegerme y guiar mi camino en todo momento y por permitirme culminar una de mis metas. A mi madre Mirian, hermanos Grace, Steve y Alexander y mi sobrina Gabriela por su cariño, su apoyo incondicional y estar a mi lado a pesar de las adversidades, a mi esposo por su cariño y apoyo en todo este tiempo, a mis amigas y amigos que siempre han estado dándome ánimos y apoyando en el transcurso de mi carrera.

Deysi

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE GRÁFICOS.....	xi
ÍNDICE DE ANEXOS.....	xii
RESUMEN.....	xiii
SUMMARY.....	xiv
INTRODUCCIÓN.....	1
CAPÍTULO I	
1. MARCO TEÓRICO.....	7
1.1. Java.....	7
<i>1.1.1. Definición.....</i>	<i>7</i>
1.2. Angular JS.....	7
<i>1.2.1. Características.....</i>	<i>8</i>
<i>1.2.2. Funcionamiento.....</i>	<i>10</i>
<i>1.2.2.1. Scopes.....</i>	<i>11</i>
<i>1.2.2.2. Controllers.....</i>	<i>11</i>
<i>1.2.2.3. Data binding.....</i>	<i>11</i>
<i>1.2.2.4. Directives.....</i>	<i>11</i>
<i>1.2.2.5. Filters.....</i>	<i>11</i>
<i>1.2.2.6. Services.....</i>	<i>12</i>
<i>1.2.3. ¿Por qué utilizar AngularJS?.....</i>	<i>12</i>
1.3. Arquitectura MVVM.....	13
<i>1.3.1. Modelo.....</i>	<i>13</i>
<i>1.3.2. Vistas.....</i>	<i>13</i>
<i>1.3.3. Modelo de la vista.....</i>	<i>13</i>
1.4. MySQL.....	14
<i>1.4.1. ¿Por qué usar MySQL?.....</i>	<i>15</i>
1.5. Metodología SCRUM.....	15
1.6. Herramientas software para desarrollo del sistema.....	17
CAPÍTULO II	
2. MARCO METODOLÓGICO.....	18
2.1. Análisis de la situación actual de la Caja de Ahorros.....	18
2.2. Metodología SCRUM.....	20

2.2.1.	<i>Personas y roles del proyecto</i>	21
2.2.2.	<i>Requerimientos</i>	21
2.3.	Planificación del proyecto	23
2.4.	Desarrollo del sistema	26
2.4.1.	<i>Arquitectura del sistema</i>	27
2.4.2.	<i>Base de Datos</i>	27
2.4.2.1.	<i>Diccionario de datos</i>	30
2.4.3.	<i>Definición de la interfaz de usuario</i>	31
2.4.4.	<i>Historias de Usuario</i>	32
2.4.5.	<i>Tareas de ingeniería</i>	34
2.5.	<i>Gestión del proyecto</i>	35
2.6.	<i>Población y muestra</i>	36
CAPÍTULO III		
3.	MARCO DE RESULTADOS	38
3.1.	Demostración de reducción de tiempo	38
3.1.1.	<i>Análisis de resultados</i>	40
CONCLUSIONES		46
RECOMENDACIONES		47
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-1: Herramientas software para el desarrollo del sistema.	17
Tabla 1-2: Roles del proyecto	21
Tabla 2-2: Product Backlog para el proyecto.....	21
Tabla 3-2: Tabla de planificación del proyecto.....	24
Tabla 4-2: Diccionario de Datos	30
Tabla 5-2: Modelo de la Historia de Usuario.....	33
Tabla 6-2: Formato de Tarea de Ingeniería.....	34
Tabla 7-2: Formato de Pruebas de Aceptación	35
Tabla 1-3: Tabla de tiempos manuales	39
Tabla 2-3: Porcentaje de tiempos en una unidad de tiempo.....	39
Tabla 3-3: Tabla de tiempo de préstamo.....	40
Tabla 4-3: Tabla de tiempo de cierre de caja.	41
Tabla 5-3: Tabla de tiempo de reporte aportaciones.	42
Tabla 6-3: Tabla de tiempo de reporte de préstamo.....	43
Tabla 7-3: Tabla de tiempos.....	44

ÍNDICE DE FIGURAS

Figura 1-1: Two-Way Data Binding	9
Figura 2-1: Funcionamiento de AngularJS.....	10
Figura 3-1: Interacción de las capas del patrón MVVM.....	14
Figura 4-1: Fases de la metodología SCRUM.....	17
Figura 1-2: Proceso de Entrega de Préstamo.	19
Figura 2-2: Generación de Reportes.	20
Figura 3-2: Diagrama de casos de usos general de los procesos de la Caja de Ahorros.....	27
Figura 4-2: Arquitectura del sistema	27
Figura 5-2: Modelo de entidad relación de la base de datos.....	29
Figura 6-2: Bosquejo de la interfaz de usuario	32

ÍNDICE DE GRÁFICOS

Gráfico 1-2: Gráfica representativa del avance del desarrollo del proyecto.	36
Gráfico 1-3: Resultado del proceso de préstamo.	40
Gráfico 2-3: Resultado del proceso de cierre de caja.....	41
Gráfico 3-3: Resultado del reporte de aportaciones.....	42
Gráfico 4-3: Resultado del reporte de préstamos.....	43
Gráfico 5-3: Resultado total de los procesos.....	44

ÍNDICE DE ANEXOS

Anexo A: Historias de Usuario

Anexo B: Tareas de Ingeniería.

Anexo C: Pruebas de Aceptación

Anexo D: Diccionario de Datos

RESUMEN

El objetivo del trabajo de titulación fue reducir el tiempo en la entrega de préstamos y generación de informes para la caja de ahorro de la empresa Tubasec C.A., mediante el desarrollo de un sistema, para ello se utilizaron las técnicas como la observación y la entrevista, aplicada a la persona encargada de la caja. El sistema fue realizado en base a la metodología de desarrollo ágil SCRUM la misma que permitió crear un entorno de trabajo óptimo e interactivo entre el cliente y el sistema, para la creación del mismo se siguió el patrón Modelo, Vista, Modelo-Vista (MVVM). Para el desarrollo de la aplicación, se utilizó la tecnología AngularJS para el front end, mientras que para el back end se trabajó con Java en el ambiente de desarrollo NetBeans 8.2 y servicios RestFULL lo que permitió la conexión entre el front end y el back end, y para la gestión de la base de datos se utilizó MySQL. Sobre el sistema se pudo verificar la reducción de tiempo en los procesos de entrega de préstamos y generación de informes mediante el uso de la estadística descriptiva. De esta manera se puede decir que el sistema respondió favorablemente a cada uno de los procesos, ya que se tiene una reducción de tiempo de hasta un 98% en la entrega de préstamos y de un 99.96 % en la generación de los informes. Se recomienda ampliar los estudios de AngularJS para buscar otras maneras de crear código para la generación de reportes, debido a que las librerías con las que trabaja, no permiten generarlos de forma automática.

PALABRAS CLAVES: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, <AUTOMATIZACIÓN DE PROCESOS FINANCIEROS>, <METODOLOGÍA DE DESARROLLO (SCRUM)>, <ANGULARJS>, <MODELO, VISTA, MODELO VISTA (MVVM)>.

SUMMARY

The objective of the present research work was to reduce the time in granting loans and the generation of reports for the savings bank of Tubasec C.A. company, through the development of a system, for which techniques such as observation and interview were used. They were applied to the person in charge of the box. The system was made based on SCRUM agile development methodology, which allowed creating an optimal and interactive work environment between the client and the system. For the creation of the system, it was followed the Model, View, View model (MVVM). For the development of the application, AngularJS technology was used for the front end, while for the back end we worked with Java in the NetBeans 8.2 development environment and RestFULL services, which allowed the connection between the front end and the back end, and MySQL was used to manage the database. On the system it was possible to verify the reduction of time in the processes of granting loans and generation of reports through the use of descriptive statistics. In this way it can be said that the system responded favorably to each of the processes, since there is a reduction of time of up to 98% in the granting of loans and 99.96% in the generation of reports. It is recommended to extend the AngularJS studies to find other ways to create code for the generation of reports, because the libraries with which it works, do not allow generating them automatically.

KEYWORDS: <TECHNOLOGY AND SCIENCE OF ENGINEERING>, <SOFTWARE ENGINEERING>, <AUTOMATION OF FINANCIAL PROCESSES>, <DEVELOPMENT METHODOLOGY (SCRUM)>, <ANGULARJS>, <MODEL, VIEW, VIEW MODEL (MVVM)>.

INTRODUCCIÓN

El presente trabajo de titulación tiene como objetivo presentar el desarrollo de un sistema web que permita el control de aportaciones y préstamos de la caja de ahorro de los empleados de la empresa Tubasec C.A, aplicando la tecnología AngularJS.

El propósito fundamental del presente trabajo de titulación es aplicar la tecnología AngularJS para el desarrollo de un sistema web que permita mejorar los procesos de entrega de préstamos y las aportaciones que se manejan dentro de la caja de ahorros permitiendo que estos procesos se los realice en un menor tiempo que de la forma manual con que se los maneja.

El trabajo de titulación está comprendido por 3 capítulos, dentro de los cuales se tiene el capítulo 1 donde se puede apreciar las tecnologías y herramientas que se utilizan para el desarrollo adecuado de la aplicación, así como para la gestión del desarrollo del mismo.

En el capítulo 2 se describe en base a las fases y la utilización de los diferentes artefactos con los que cuenta la metodología SCRUM el cómo se gestionó el desarrollo del sistema.

Y por último en el capítulo 3 se presentan los resultados obtenidos en la comparación de tiempos entre los procesos llevados de forma manual y los procesos con el sistema implementado.

ANTECEDENTES

Tubasec C.A es una empresa Riobambeña y una de las principales del centro del país, buscando día tras día superar nuevas metas y sobre todo la satisfacción del cliente en todas sus líneas de producción: EUROLIT, ECUATEJA y TECHOLUZ. Dentro de la empresa existe una caja de ahorro con un total de 50 socios, en la cual se realizan aportaciones mensuales por parte de los socios y a su vez entrega préstamos para que los socios puedan cubrir cualquier tipo de necesidad.

La caja de ahorro debido al tiempo de demora en la entrega de los préstamos y reportes que se realizan mensualmente, ha visto la necesidad de implementar un sistema que permita tener un control de la caja y a su vez agilizar los procesos que se realizan tanto en préstamos y reportes, ya que hasta el momento se realizan de modo manual y de esta manera evitar inconformidades de los socios por pagos atrasados en los préstamos, provocando así el retiro de socios y la disminución en las aportaciones.

Para la implementación del sistema se utilizará el gestor de base de datos MySQL que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se acomodan a las diferentes necesidades y requerimientos (DefinicionABC, 2016), además cabe recalcar que en la empresa este gestor es utilizado debido a las diferentes aplicaciones con las que ya cuenta, se integrará el patrón de arquitectura MVC con el Framework AngularJS que permite la utilización de cualquier tipo de back end para trabajar con la aplicación, además reduce la carga del servidor de aplicaciones debido a que el framework construye la vista de forma dinámica y mejora el rendimiento de la aplicación. (Peña Arpi & Cambisaca Sánchez)

FORMULACIÓN DEL PROBLEMA

¿De qué manera el sistema permitirá reducir el tiempo de entrega de préstamos y generación de informes para la caja de ahorros de la empresa Tubasec C.A.?

SISTEMATIZACIÓN DEL PROBLEMA

¿Qué beneficios brinda el desarrollo del sistema al utilizar el Framework AngularJS?

¿Cómo mejorará la aplicación a la entrega de préstamos que se entrega a los socios?

¿Cómo el sistema llevará un control sobre los pagos de los préstamos entregados a los socios?

¿Cómo los reportes de los préstamos y aportaciones de los socios mejorarán el control llevado por la empresa?

JUSTIFICACIÓN

Justificación teórica

En la actualidad debido al cambio tecnológico y la innovación que se da constantemente, el uso AngularJS se puede demostrar que ha sido motivo de investigación ya que se ha realizado tesis con el propósito de comparar este Framework con EmberJS determinando así que AngularJS es más eficiente y que reduce el consumo del hardware del computador como son la tarjeta gráfica, procesador, memoria RAM, Disco Duro y la red, además AngularJS es el más apropiado para aplicaciones web que requieran de un mejor rendimiento. (Peña Arpi & Cambisaca Sanchez, 2015)

AngularJS es un framework de código abierto de JavaScript, mantenido por Google, que ayuda con la gestión de aplicaciones SPA (Single Page Applications aplicaciones de una sola página). Su objetivo es incrementar las aplicaciones basadas en navegador, con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para que el desarrollo y las pruebas sean más fáciles.(MEJÍA, 2015)

Para el desarrollo del sistema se utilizará JAVA ya que podemos programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. (Alvarez, 2001) y el patrón de arquitectura MVC ya que este crea una estructura de diseño que mejora la organización y reutilización de código, el mantenimiento y escalabilidad de una aplicación y facilita el trabajo. (Morocho, 2010)

En la investigación realizada se encontró una tesis de grado que trata de un sistema web para el control de una caja de ahorro el cuál fue desarrollado en MySQL y PHP, usando para el desarrollo del software la metodología RUP y el modelado de datos UML. (Quijada, 2011)

Las razones por las que no se puede adoptar esta solución debido a: No existe librería de código fuente para poderlo adaptar a los procesos que requiere la empresa, además la tecnología utilizada es diferente a la que se maneja, la misma que incurrirá en costos, debido a que se necesita conocer dicha tecnología.

Justificación aplicativa

El sistema se desarrollará enfocándose en la línea de investigación según Escuela Superior Politécnica de Chimborazo en la Tecnología de la información, comunicación, procesos industriales y biotecnológicos, además se apega a la línea de Investigación de la Escuela de Ingeniería en Sistemas en el Desarrollo de aplicaciones informáticas y en el ámbito de análisis y diseño de proyectos software.

El desarrollo del sistema se lo implementará por medio de módulos: en el módulo de autenticación se ingresará los roles de los usuarios con los cuales se darán los permisos según el usuario, el módulo de socios nos permitirá ingresar los datos de los empleados que forman parte de la caja; el módulo de aportaciones que permitirá llevar un control detallado de las

aportaciones que se realizan mensualmente por parte de los socios, el módulo de préstamos el cual agilizará el proceso de préstamos que se entregan a los socios dependiendo del tipo de préstamo que se desee aplicar.

Se tendrá también el módulo de reportes que permitirá tener de manera oportuna los reportes que se realizan mensualmente con el propósito de conocer el estado en el que se encuentra la caja y así evitar que los socios se sientan inconformes con los pagos atrasados por los préstamos, además los socios podrán obtener información exacta y oportuna de las aportaciones y préstamos que se realizan, con esto se pretende atraer a otros empleados para que sean parte de la caja y puedan tener los beneficios que esta brinda a sus socios.

OBJETIVOS

Objetivos General

Desarrollar un sistema para el control de aportaciones y préstamos para la caja de ahorro de la empresa Tubasec C.A.

Objetivos específicos

- Determinar los tiempos que tarda la persona encargada de la caja en la entrega de préstamos de forma manual y a su vez en generar informes contables.
- Recolectar de la persona encargada los requerimientos para automatizar los procesos de entrega de préstamos y generación de informes contables.
- Utilizar AngularJS para la integración de un sistema web con base de datos.
- Diseñar e implementar un sistema web para la automatización de los procesos de entrega de préstamos y generación de informes contables.

- Verificar la reducción de tiempo en la entrega de préstamos y la generación de los informes con la utilización del sistema desarrollado.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Java

1.1.1. Definición

Es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y es de gran importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre orientado a cubrir las necesidades tecnológicas.(Alvarez, 2001)

Una de las características principales por las que Java se ha hecho muy famoso es que su lenguaje es independiente de la plataforma. Esto significa que si hacemos un programa en Java este podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes se realizaba un programa para cada sistema operativo, por ejemplo, Windows, Linux, Apple, etc. Esto se consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java, posibilita que este último se entienda perfectamente. (Alvarez, 2001)

1.2. Angular JS

AngularJS es un framework MVC de JavaScript para el Desarrollo Web Front End que admite crear aplicaciones SPA Single-Page Applications. Este está dentro de la familia de frameworks como BackboneJS o EmberJS.(Azaustre, 2013)

Angular está diseñado para aplicaciones web dinámicas, sostenido por Google, utiliza HTML como lenguaje de plantillas y la desarrolla para expresar los componentes de su aplicación, luego AngularJS enseña al navegador la nueva sintaxis a lo que llamamos directivas. Los conceptos de enlace de datos e inyección de dependencias permiten disminuir líneas de código y su ejecución es elaborada dentro del navegador. (Peña Arpi & Cambisaca Sanchez, 2015)

Angular es 100% JavaScript del lado del cliente, no es un simple sistema de plantillas, la razón se debe a “bidirectional data binding”, enlace de datos bidireccional, que posee y se ejecuta de forma automática. La plantilla se compila en el navegador y el paso de compilación da como resultado una vista en vivo.(Peña Arpi & Cambisaca Sanchez, 2015)

Angular permite el uso de cualquier tipo de back end para trabajar con la aplicación, existe independencia en la utilización de Java, Python, Ruby, C# o cualquier otro lenguaje. Lo que se necesita es una forma de comunicación de ida y vuelta con el servidor, peticiones HTTP XML o JSON. (Peña Arpi & Cambisaca Sanchez, 2015)

Si el servidor tiene un API REST que provee valores JSON, trabaja muy fácilmente con el front end, aunque se debe tener en cuenta que no es un problema si el servidor no devuelve JSON, porque se le puede enseñar a AngularJS a comunicarse con el servidor XML o con algún otro formato. (Peña Arpi & Cambisaca Sanchez, 2015)

1.2.1. Características

Data Binding

“Enlace de Datos”, Sin duda uno de los puntos fuertes de AngularJS.

El binding es enlazar los datos del Scope con lo que mostramos en el HTML de forma automática, producida en dos sentidos.(Peña Arpi & Cambisaca Sanchez, 2015)

One-waybinding

Los datos van únicamente desde el Scope hacia la parte visual, (modelo hacia la vista). Se consigue mediante la sintaxis de llaves dobles.

```
{{Datos}}
```

Si se modifica el dato del modelo (Scope), se actualiza automáticamente la vista. (Peña Arpi & Cambisaca Sanchez, 2015)

Two-waybinding

Los datos fluyen no solo desde el Scope hacia la vista, sino también desde la vista hacia el Scope, se implementa mediante la directiva ngModel.

```
<input type="text" ng-model="miDato" />(Peña Arpi & Cambisaca Sanchez, 2015)
```

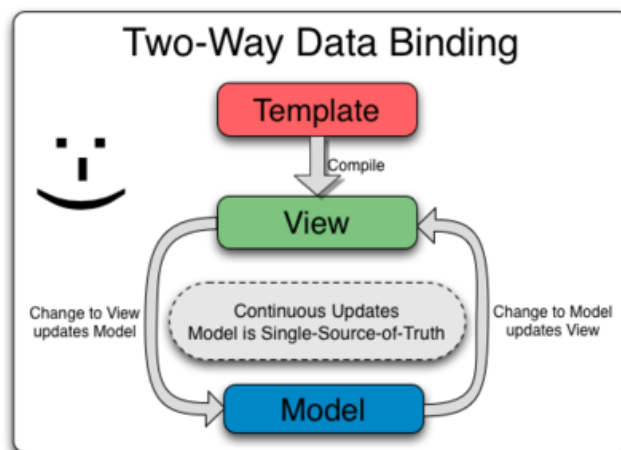


Figura 1-1:Two-Way Data Binding.

Fuente: <https://docs.angularjs.org/guide/databinding>, 2015.

Cuando el modelo cambia, el dato que está dentro de la caja de texto cambiará y cuando el usuario cambie el valor de la caja de texto el Scope se actualiza automáticamente. La **Figura 1-1** presenta el funcionamiento del enlace de datos bidireccional de AngularJS.

```
<div ng-app>
```

```
<input type="text" ng-model="dato" /><!--Doble binding -->
```

```
{{ dato }}                                <!--Simple binding -->
</div>(Peña Arpi & Cambisaca Sanchez, 2015)
```

1.2.2. Funcionamiento

El funcionamiento de AngularJS se lo puede considerar en la **Figura 2-1**, en el cual se puede notar que mediante el uso de servicios REST los mismo que tiene un estándar HTTP, permitiendo que las solicitudes y la comunicación con la base de datos se pueda dar en conjunto con AngularJS desde el navegador del cliente hasta el servidor de la aplicación.

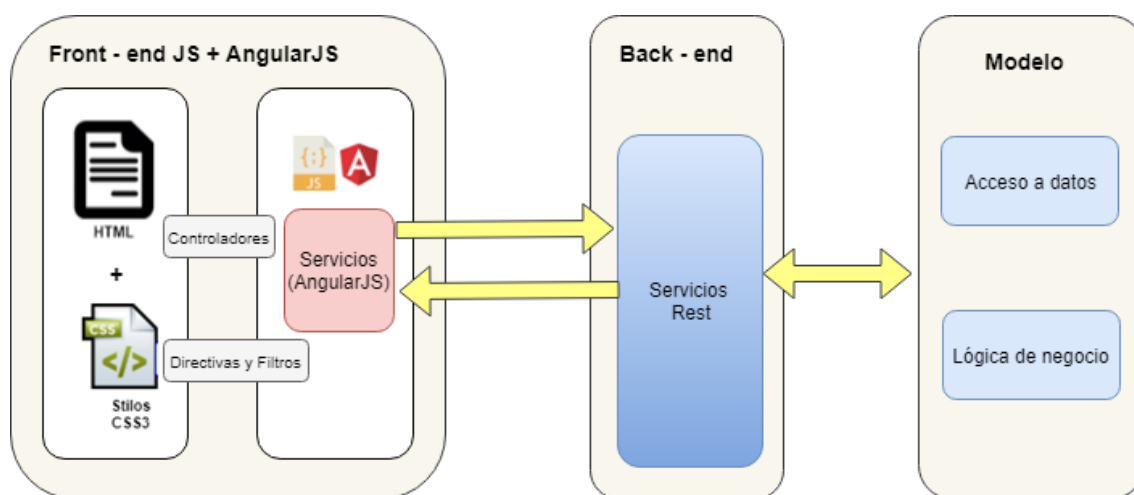


Figura 2-1: Funcionamiento de AngularJS.

Realizado por: Deysi Latorre, 2018

Fuente:(Blanco Rodrigues, 2015, <http://blog.gfi.es/super-heroes-en-la-web-con-angularjs/>).

Pablo Lázaro en su blog personal menciona que angular brinda algunas características principales como son: directivas, data binding, filtros y servicios, pero también es necesario utilizar los scopes y los controladores, su utilización y el propósito de cada una de ellas.

1.2.2.1. Scopes

Los Scopes son las expresiones principales sobre las cuales trabaja AngularJS, el mismo que se referencia o se llama para su utilización la directiva *ng-model*, también el scope es utilizado para poder mostrar los datos en las diferentes vistas de la aplicación y son manejados desde los controladores y las directivas. (Lázaro, 2013)

1.2.2.2. Controllers

Los controller son quienes permiten que se inicialice y se modifique la información de los scope, también se puede crear funciones para ser llamadas desde la vista mediante la declaración del scope. (Lázaro, 2013)

1.2.2.3. Data binding

Mediante la utilización de directivas se puede actualizar las vistas y el modelo ya que estos trabajan en conjunto, permitiendo así que los cambios que se efectúen tanto en la vista como en el modelo se reflejen automáticamente. (Lázaro, 2013)

1.2.2.4. Directives

Las directivas son las que permiten que se extienda la sintaxis del HTML, permitiendo con ello que se modifique el comportamiento del elemento al que afecta la directiva como por ejemplo un botón o un input del HTML. (Lázaro, 2013)

1.2.2.5. Filters

Pablo Lazaro en su bloc personal menciona que los filtros permiten modificar la presentación de la información su forma de utilizar “*{{ expresion | filtro }}*” donde en expresión es el atributo

por el cual se realizara el filtrado y donde filtro es el nombre del filtro con el que se identificará al filtro. (Lázaro, 2013)

1.2.2.6. Services

Los servicios permiten que la intercomunicación con el servidor con el fin de poder obtener información que se pueda manipular con los scopes y los controladores para ser utilizados en las vistas.

También se tiene diferentes tipos de servicios como son services, factories y providers, donde los services son los recursos que permiten encapsular la interacción con servicios RESTful evitando las llamadas http de forma directa. (Lázaro, 2013)

1.2.3. ¿Por qué utilizar AngularJS?

AngularJS consiste en un conjunto de herramientas integradas que ayudan a construir aplicaciones del lado del cliente, estructuradas en un sistema modular, con menos código y más flexibilidad.(Boaventura, 2012)

Extiende HTML, proporcionando directrices que añaden funcionalidad al margen de beneficio y permite crear plantillas dinámicas. También se puede crear directrices, elaborando componentes reusables que completen las necesidades.(Boaventura, 2012)

También implementa binding de datos de dos vías, conectando HTML (vistas) a los objetos de JavaScript (modelos) sin problemas. Esto significa que cualquier actualización en el modelo se reflejará inmediatamente en la vista, sin necesidad de ningún tipo de manipulación en el control de eventos.(Boaventura, 2012)

La mayoría de los marcos de JavaScript, AngularJS permite trabajar con cualquier tecnología de servidor, ya que la aplicación se comunica a través de una API REST Web. También proporciona servicios, los cuales simplifican considerablemente el código y permite separar las API en servicios reusables. Se puede mover el modelo y la lógica del negocio para el front-end y construir aplicaciones web back-end independientes. (Boaventura, 2012)

1.3. Arquitectura MVVM

AngularJS trabaja con el patrón de arquitectura basada en el MVC (Modelo Vista Controlador) tradicional pero debido a que el modelo y la vista trabajan en conjunto, los creadores consideran que el patrón se mejora y ahora es denominado MVVM (Model, View, View -Model), este patrón permite lograr una separación entre la lógica del negocio y la interfaz del usuario y de los procesos que realizan en la aplicación, para entender un poco mejor el patrón MVVM a continuación se va a describir.(Rauh, 2015)

1.3.1. Modelo

En el modelo se realiza la representación de la lógica de negocios permitiendo de esta forma que se realicen las operaciones que se crean convenientes con los datos de la aplicación, generalmente en el modelo se trabaja con los \$Scope y pueden ser llamados desde un controlador. (Minar, 2012)

1.3.2. Vistas

Es la representación de la información para el usuario mediante archivos HTML e inclusive las hojas de estilos CSS, de tal forma que pueda interactuar el usuario con la información presentada.(Microsoft, s.f.)

1.3.3. Modelo de la vista

Esta es una capa intermedia entre la vista ay el modelo que permite acceder a la información o a los datos y transformarlos de tal forma que puedan ser representados en la vista, de la forma en la que mejor convenga.(Microsoft, s.f.)

Para comprender de mejor manera las relaciones entre la vista, el modelo y el modelo de la vista se presenta cómo interactúan entre sí, se puede ver en la **Figura 3-1**.

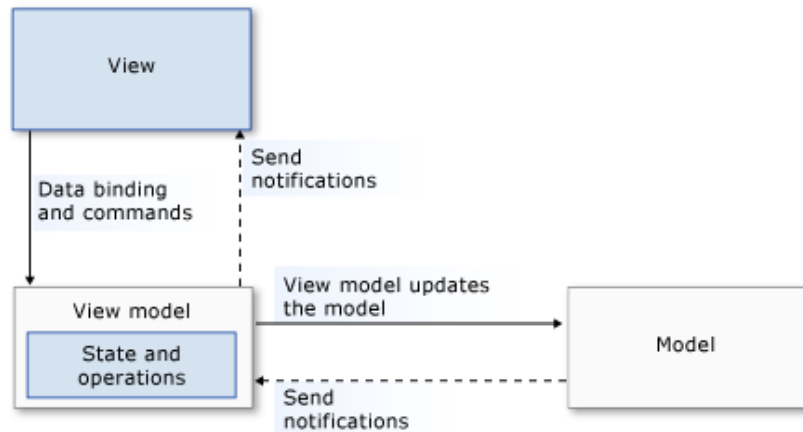


Figura3-1: Interacción de las capas del patrón MVVM

Fuente:(Microsoft, <https://msdn.microsoft.com/es-xl/library/windows/apps/jj160324.aspx>)

1.4. MySQL

MySQL es un sistema de administración de bases de datos relacionales. Una base de datos relacional recopila los datos en tablas separadas en lugar de poner todos los datos en un solo lugar. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se requiera consultar datos. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandarizado para el acceso a bases de datos relacionales. (Bravo, 2007)

El sistema de base de datos operacional MySQL hoy en día uno de los más significativos en lo que hace al diseño y programación de base de datos de tipo relacional. Este cuenta con millones de aplicaciones y emerge en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se utiliza como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo.(Victoria, 2009)

Una de las características principales de MySQL es que admite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adecuan a diferentes necesidades y requerimientos. Por otro lado, MySQL es distinguida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores. Las plataformas que maneja son de variado tipo y entre ellas podemos mencionar LAMP, MAMP, SAMP, BAMP y WAMP (aplicables a Mac, Windows, Linux, BSD, Open Solaris, Perl y Phyton entre otras). (Victoria, 2009)

1.4.1. ¿Por qué usar MySQL?

El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Si eso es lo que se está buscando, se le debe dar una oportunidad a MySQL. Se pueden encontrar comparaciones de desempeño con algunos otros manejadores de bases de datos en la página de MySQL.

El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor MySQL ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet. (Bravo, 2007)

1.5. Metodología SCRUM

SCRUM, es una metodología que permite el trabajo en equipo y que tiene como finalidad el mejorar el desarrollo colaborativo, aplicando un conjunto de buenas prácticas, permitiendo que el rendimiento del grupo de desarrollo tenga un mejor desempeño, conllevando de esta forma que a pesar de cambios o sucesos que se puedan dar durante el desarrollo del producto software este llega su fin, evitando en lo posible aumento de costes o demoras en la entrega del producto. (Albaladejo, 2012)

Una de las características principales de SCRUM, es que se deben realizar entregas de una versión del producto según se ha planificado, cada uno de estos entregables se considera un

sprint. La dimensión de un sprint se considera entre 1 y 4 semanas como el tiempo de duración para el desarrollo.(Albaladejo, 2012)

En la **Figura 4-1**, podemos observar las fases de la metodología SCRUM, en la cual podemos observar que lo primero a realizar es el análisis de los requerimientos, para poder diseñar el product backlog que es en el cual se pondrá en orden según la prioridad del cliente para su posterior desarrollo en los sprints. Donde el equipo a desarrollar selecciona los requerimientos del product backlog que se desarrollaran, estiman el esfuerzo en la realización de los mismos, esta tarea no demora más de 4 horas, para pasar a la ejecución del sprint, durante la ejecución de los sprint se debe realizar reuniones diarias donde se responderán preguntas que se refieren al avance, continuidad y dificultades para el desarrollo de la tarea seleccionada en el sprint, al finalizar el sprint se realiza una reunión en la cual se puede realizar un breve demostración del avance del producto y analizar posibles inconvenientes que se podrían presentar en los sprints posteriores permitiendo de esta forma que el scrum master pueda ayudar a resolverlos y eliminar dichos problemas.(Albaladejo, 2012)

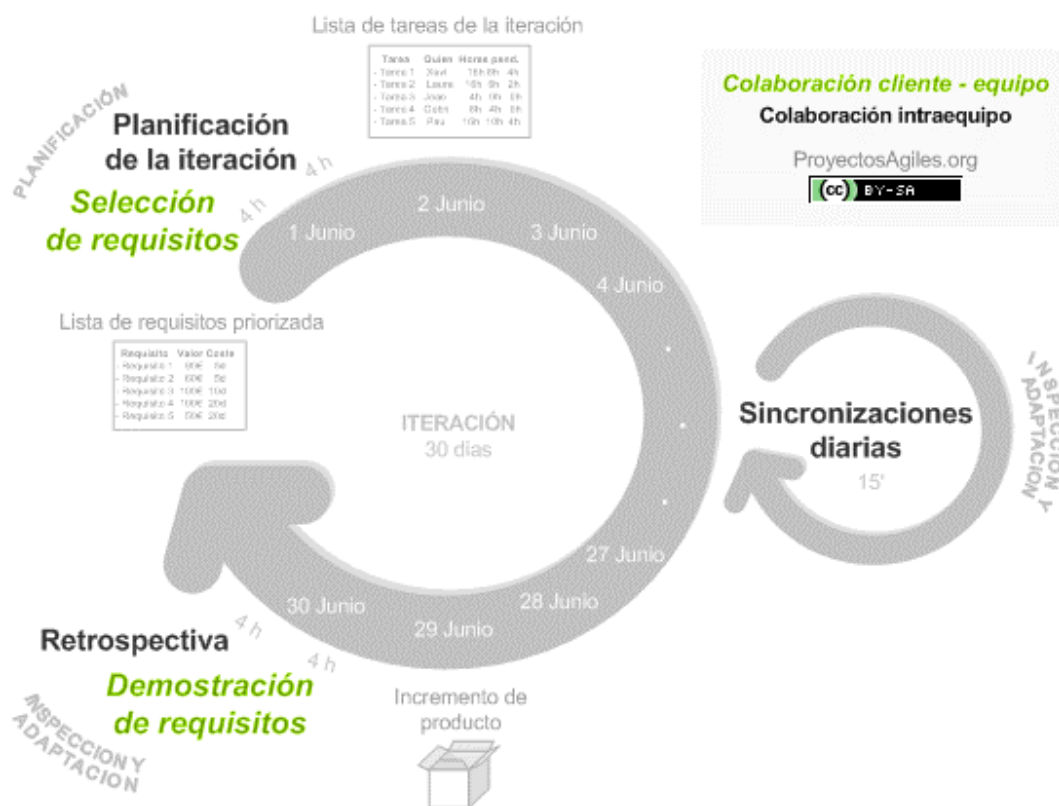


Figura4-1: Fases de la metodología SCRUM.

Fuente:<https://proyectosagiles.org/que-es-scrum/>

Una vez terminado todos los sprints en el tiempo establecido se puede proceder a poner en producción el producto software, siendo esa la fase de cierre del desarrollo.

1.6. Herramientas software para desarrollo del sistema

Para el desarrollo del sistema SYSCAP se utiliza algunas herramientas software, las cual se detallan en la **Tabla 1-1**, donde se describe las características y su especificación.

Tabla 1-1: Herramientas software para el desarrollo del sistema.

Nombre	Especificación Técnica	Descripción
MySQL	Motor de base de datos	Sistema de gestión de base de datos.
	Administrador de la base de datos	Aplicación grafica para gestionar el gestor de la base de datos MySQL.
NetBeans 8.2	Entorno de desarrollo integrado	Entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.
Apache Tomcat	Servidor web	Servidor de aplicaciones web
Google Chrome	Navegador	Aplicación que opera a través de Internet, interpretando archivos y sitios web desarrollados.

Realizado por: Deysi Latorre, 2018

CAPÍTULO II

2. MARCO METODOLÓGICO

En este capítulo se presenta el desarrollo del sistema para la Caja de Ahorros de la Empresa Tubasec C.A., utilizando el framework AngularJS y con la tecnología Open Source para el desarrollo, de igual manera para la gestión de la base de datos.

AngularJS permite la utilización de cualquier tipo de back-end para trabajar con la aplicación, es por esto que en el modelo se está trabajando con Java estableciendo la conexión entre el back-end y el front-end con servicios Restfull.

2.1. Análisis de la situación actual de la Caja de Ahorros.

En la Caja de Ahorros de la empresa actualmente los procesos de entrega de préstamos y generación de informes contables hasta el momento se los viene realizando de modo manual.

Un empleado para solicitar un préstamo debe acercarse a la oficina donde se encuentra la persona encargada de la caja.

Proceso de entrega de préstamos: el socio se acerca a la persona encargada a solicitar un préstamo, la encargada revisa los datos del empleado para verificar si es socio, de serlo procede a verificar si es cumplido en las aportaciones y si no ha tenido algún inconveniente en algún préstamo anterior, revisado esto se procede a ver el monto a solicitar y se envía la solicitud. La persona encargada verificará si es posible entregar el préstamo al socio, si se le aprueba se procederá a calcular la tabla de amortización para que el socio conozca las cuotas a pagar por el monto solicitado, si el socio está de acuerdo en las cuotas se procederá a la entrega del dinero conjuntamente con la tabla de amortización, en la cual estará especificado la fecha de pago, el monto, el interés y el saldo restante. En el **Figura 1-2** se puede ver como se realiza el proceso de entrega de préstamos.

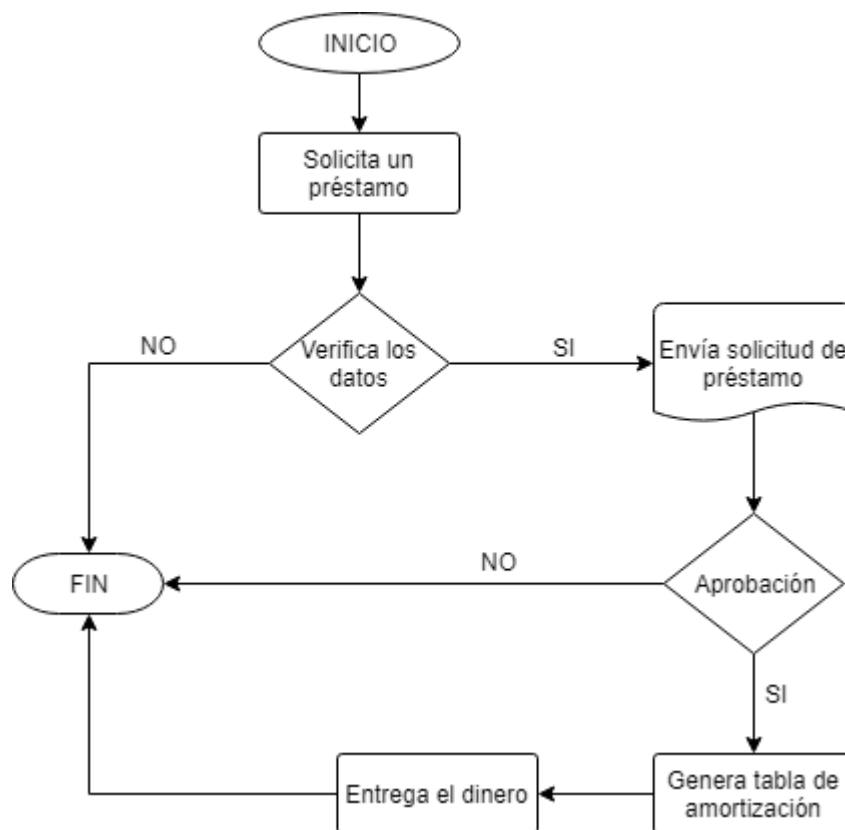


Figura 1-2: Proceso de Entrega de Préstamo.

Realizado por: Deysi Latorre, 2018

Proceso de generación de informes contables: para realizar los informes mensuales sobre las aportaciones, préstamos o cierre de caja, la persona encargada debe revisar cada una de las tarjetas de cada uno de los socios e ir verificando y anotando los datos dependiendo del informe

que se esté realizando. En el **Figura 2-2** se puede ver como se realiza el proceso de generación de informes contables.



Figura2-2: Generación de Reportes.

Realizado por: Deysi Latorre, 2018

Luego de conocer cómo se realizan los procesos de entrega de préstamos y generación de informes contables, se evidencia los inconvenientes que existen dentro de la caja, los mismos que son mencionados en el apartado Antecedentes de este documento.

2.2. Metodología SCRUM

La metodología SCRUM, permite que el sistema SYSCAP se desarrolle mediante Sprint donde se distribuyen cada uno de los requerimientos a desarrollar, teniendo en cuenta que el tamaño mínimo de duración de cada Sprint es de una semana y un máximo de cuatro semanas, una semana de trabajo es de cinco días y cada día tiene un valor de ocho horas de trabajo.

2.2.1. *Personas y roles del proyecto*

Para realizar el Proyecto se cuenta con la participación de dos personas y tres roles cuyos datos se mencionan en la **Tabla 1-2**.

Tabla 1-2: Roles del proyecto

Persona	Rol	Contacto
Dra. Betty Coral	Product Owner	032626218
Ing. Natalia Layedra	Scrum Master	nlayedra@esPOCH.edu.ec
Deysi Latorre	TeamDevelopment	deykrolina13@gmail.com

Realizado por: Deysi Latorre, 2018

2.2.2. *Requerimientos*

Se parte de la lista de tareas o lo que es lo mismo las historias de usuario, estos son un conjunto de requerimientos descritos en un lenguaje no técnico. Se prioriza los requerimientos basándose en objetivos, coste, etc. Cada miembro del equipo se le asigna un número de actividades para realizarlas, de igual manera se determina el número de iteraciones en las cuales se desarrolla el proyecto.

Luego de varias reuniones con el PRODUCT OWNER y SCRUM MASTER se obtuvo los requerimientos necesarios para el desarrollo del sistema, el tiempo de estimación se lo realiza en días, teniendo que una semana de trabajo está comprendida en cinco días sin incluir los fines de semana, cada día estará valorado por 8 horas de trabajo, con una prioridad que se valora en alta, media y baja.

A continuación, en la **Tabla 2-2** se da a conocer cada uno de los requerimientos con su prioridad.

Tabla 2-2: Product Backlog para el proyecto.

Nº	Requerimientos	Prioridad
1	Como administrador del sistema deseo poder tener un inicio de sesión	ALTA
2	Como administrador del sistema deseo poder registrar el ingreso de un	ALTA

	socio	
3	Como administrador del sistema deseo poder actualizar los datos del socio	ALTA
4	Como administrador del sistema deseo poder eliminar los datos de un socio	MEDIA
5	Como administrador del sistema deseo poder registrar el ingreso de una aportación	ALTA
6	Como administrador del sistema deseo poder actualizar los datos de una aportación	ALTA
7	Como administrador del sistema deseo poder eliminar los datos de una aportación	MEDIA
8	Como usuario del sistema deseo poder ingresar la solicitud de un préstamo	MEDIA
9	Como usuario del sistema deseo poder ver el estado de mi préstamo	MEDIA
10	Como administrador del sistema deseo poder registrar el ingreso de un préstamo	ALTA
11	Como administrador del sistema deseo poder generar la tabla de amortización de un préstamo	ALTA
12	Como administrador del sistema deseo poder actualizar los datos de un préstamo	ALTA
13	Como administrador del sistema deseo poder registrar el ingreso de la cuota del préstamo	MEDIA
14	Como administrador del sistema deseo poder eliminar los datos un préstamo	MEDIA
15	Como administrador del sistema deseo poder generar el listado de socios	ALTA
16	Como administrador del sistema deseo poder generar el reporte de las aportaciones del día	ALTA
17	Como administrador del sistema deseo poder generar el reporte de las personas que no han aportando	ALTA
18	Como administrador del sistema deseo poder generar el reporte de los préstamos entregados	ALTA
19	Como administrador del sistema deseo poder generar el reporte de las personas que han tenido atraso en sus préstamos	MEDIA

20	Como administrador del sistema deseo poder ingresar la utilidad de la caja	MEDIA
21	Como administrador del sistema deseo poder ingresar la utilidad del comisariato	MEDIA
22	Como administrador del sistema deseo poder calcular la utilidad de un socio que se retire	ALTA
23	Como administrador del sistema deseo poder generar las utilidades de los socios	ALTA
24	Como usuario del sistema deseo poder consultar el saldo disponible	MEDIA
25	Como usuario del sistema deseo poder ver la fecha de pago del préstamo	MEDIA
26	Diagrama de casos de uso	MEDIA
27	Diseño de la base de datos	ALTA
28	Diseño de la interface de usuario	MEDIA
29	Implementación de la base de datos	ALTA
30	Encriptación de la clave	MEDIA
31	Como administrador del sistema deseo poder realizar el cierre de caja mensual	MEDIA

Realizado por: Deysi Latorre, 2018

El product backlog consta de un total de 31 requerimientos, los mismos son analizados para conocer si son historias de usuario o metáforas del sistema.

2.3. Planificación del proyecto

La planificación del proyecto se lo realiza con el fin de determinar el tiempo en el que se desarrolla cada una de las actividades de las metáforas del sistema planteadas al igual que las historias de usuario, las mismas que se encuentran distribuidas en sprints. Para identificar cada sprint en la **Tabla 3-2**, se puede ver que se empieza enumerando cada uno de estos, seguidamente el identificador la actividad a realizar identificado como MT a las metáforas del sistema y como HU las historias de usuario, seguidamente por el nombre de la historia y su respectiva duración.

Tabla 3-2: Tabla de planificación del proyecto

Sprint	Historia	Nombre de la Historia	Duración
		Investigación de AngularJS	266 días
		Documentación del desarrollo de la Tesis	266 días
		Recolección de requerimientos	5 días
0	MT_01	Diagramas de casos de uso	1 día
	MT_02	Diseño de la base de datos	2 días
	MT_03	Implementación de la base de datos	2 días
	MT_04	Configuración del entorno de desarrollo para AngularJS	5 días
1	MT_05	Encriptación de la clave	2 días
	MT_07	Diseño de la interfaz de usuario	3 días
2	HU_01	Implementación de la funcionalidad para generar el listado de los socios	3 días
	HU_02	Implementación de la funcionalidad para el registro de ingreso de un socio	2 días
3	HU_03	Implementación de la funcionalidad para actualizar los datos del socio	2 días
	HU_04	Implementación de la funcionalidad para eliminar los datos de un socio	3 días
4	HU_05	Implementación de la funcionalidad para el registro el ingreso de una aportación	1 día
	HU_06	Implementación de la funcionalidad para actualizar los datos de una aportación	2 días
	HU_07	Implementación de la funcionalidad para eliminar los datos de una aportación	2 días
5	HU_08	Implementación de la funcionalidad para ingresar la	3 días

		solicitud de un préstamo	
	HU_09	Implementación de la funcionalidad para ver el estado de un préstamo	2 días
6	HU_10	Implementación de la funcionalidad para el registro el ingreso de un préstamo	2 días
	HU_11	Implementación de la funcionalidad para actualizar los datos de un préstamo	2 días
	HU_12	Implementación de la funcionalidad para el registro el ingreso de la cuota del préstamo	1 día
7	HU_13	Implementación de la funcionalidad para generar la tabla de amortización de un préstamo	5 días
8	HU_14	Implementación de inicio de sesión	5 días
9	HU_15	Implementación de la funcionalidad para eliminar los datos un préstamo	3 días
	HU_16	Implementación de la funcionalidad para generar el reporte de las aportaciones del día	2 días
10	HU_17	Implementación de la funcionalidad para generar el reporte de las personas que no han aportando	1 día
	HU_18	Implementación de la funcionalidad para generar el reporte de los préstamos entregados	2 día
	HU_19	Implementación de la funcionalidad para generar el reporte de las personas que han tenido atraso en sus préstamos	2 días
11	HU_20	Implementación de la funcionalidad para ingresar la utilidad de la caja	3 días
	HU_21	Implementación de la funcionalidad para ingresar la utilidad del comisariato	2 días
12	HU_22	Implementación de la funcionalidad para calcular la utilidad de un socio que se retire	2 días
	HU_23	Implementación de la funcionalidad para generar las utilidades de los socios	3 días
13	HU_24	Implementación de la funcionalidad para consultar el saldo disponible	2 día

	HU_25	Implementación de la funcionalidad para ver la fecha de pago del préstamo	1 día
	HU_26	Implementación de la funcionalidad para realizar el cierre de caja mensual	2 días
14	MT_08	Despliegue de la aplicación en el servidor	5 días

Realizado por: Deysi Latorre, 2018

El proyecto de software tiene una duración de 56 semanas, con fecha de inicio 31/10/2016 y fecha de finalización 24/11/2017, dividida en 14 sprints detallados y organizados a como se puede ver en la **Tabla 3-2**. Cada sprint se lo realiza en días, teniendo en cuenta que un día tiene 8 horas de trabajo y un sprint puede tener máximo 5 días de trabajo teniendo así un total de 40 horas a la semana.

2.4. Desarrollo del sistema

El método de investigación inductivo permite la observación como herramienta para conocer los procesos para la resolución del problema previamente planteado, en base a los requerimientos recolectados dentro de los cuales se identifica el registro de aportaciones, entrega de préstamos y la generación de reportes, de los cuales el diagrama de casos de uso permitirá que se visualice de forma apropiada el trabajo de los diferentes actores sobre estos módulos. Como se puede ver en la **Figura 3-2**.

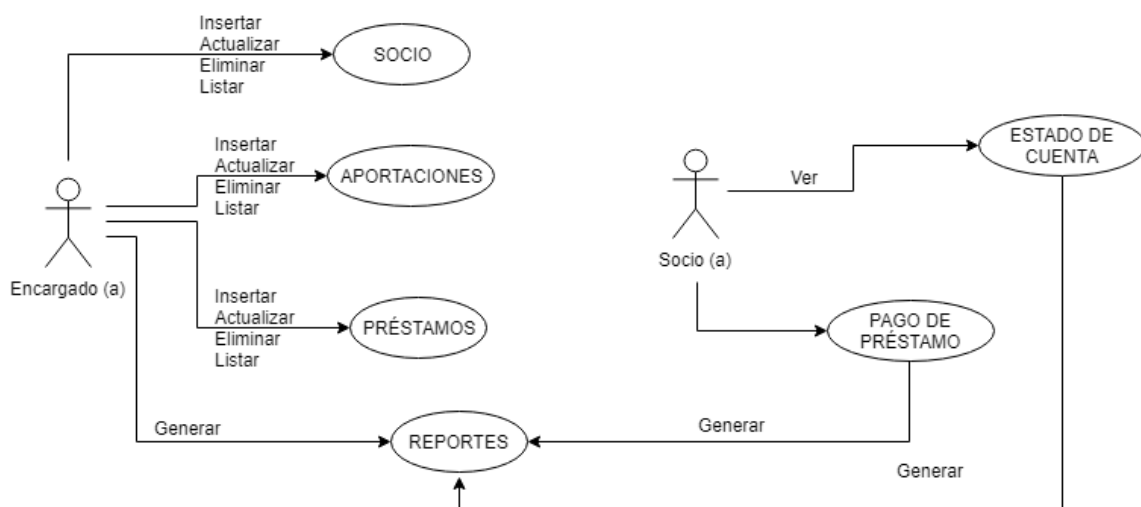


Figura 3-2: Diagrama de casos de usos general de los procesos de la Caja de Ahorros.

Realizado por: Deysi Latorre, 2018

2.4.1. *Arquitectura del sistema*

Para la solución del sistema, se opta por la arquitectura en N-capas, la cual proporciona escalabilidad permitiendo adelantos al sistema sin cambiar la arquitectura, también permite la organización de código, donde el sistema con sus n-capas al igual que la base de datos estarán alojadas en un único servidor.

De esta manera el sistema determina los recursos físicos y lógicos para el desarrollo, y así proveer los servicios necesarios de manera fácil y segura como se puede observar en la **Figura 4-2**.

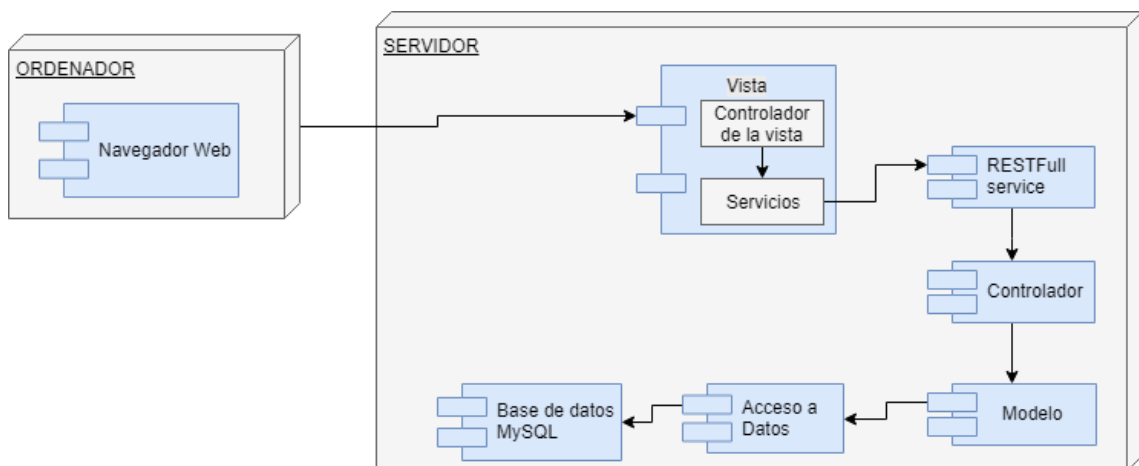


Figura 4-2: Arquitectura del sistema

Realizado por: Deysi Latorre, 2018

2.4.2. *Base de Datos*

Para el diseño de la base de datos se realiza un análisis sobre el funcionamiento actual de la caja, conjuntamente con los requerimientos establecidos por el usuario; y de esta manera diseñar una base de datos que cumpla con las necesidades, tratando de garantizar un mejor uso de los datos que van hacer almacenados.

Para la solución del sistema se utiliza el motor de base de datos MySQL, sobre la cual se aloja la base de datos cajatub y su respectivo diagrama entidad relación ampliado, se puede ver en la **Figura 5-2** que un socio (t_socio) tiene una o varias solicitudes de préstamo (t_solicitud), las cuales después de ser aprobadas se procede a la creación de un único préstamo (t_prestamo), del cual se genera una tabla de amortización (t_amortizacion), además el préstamo se lo deposita de acuerdo a la disponibilidad del efectivo en la caja de ahorro, los mismo depósitos son guardados en la tabla t_abonoprestamo. El socio puede realizar varias transacciones como depósitos, retiros pagos de préstamo y la transacción propia de la caja como es el desembolso del préstamo, cada una de estas transacciones son almacenadas en la tabla t_cuenta de acuerdo al tipo que se tiene en t_tipotransaccion. Se maneja tablas adicionales que se encuentran sueltas sin ninguna relación ya que permite gestionar información de importancia para la caja como son el monto del interés y el número de cuenta que se genera para cada socio.

Para todo socio que se registre en la caja de ahorros se crea un usuario para su posterior acceso al sistema, estos datos se encuentran almacenados en las tablas t_usuario, t_rol y t_usuariorol.

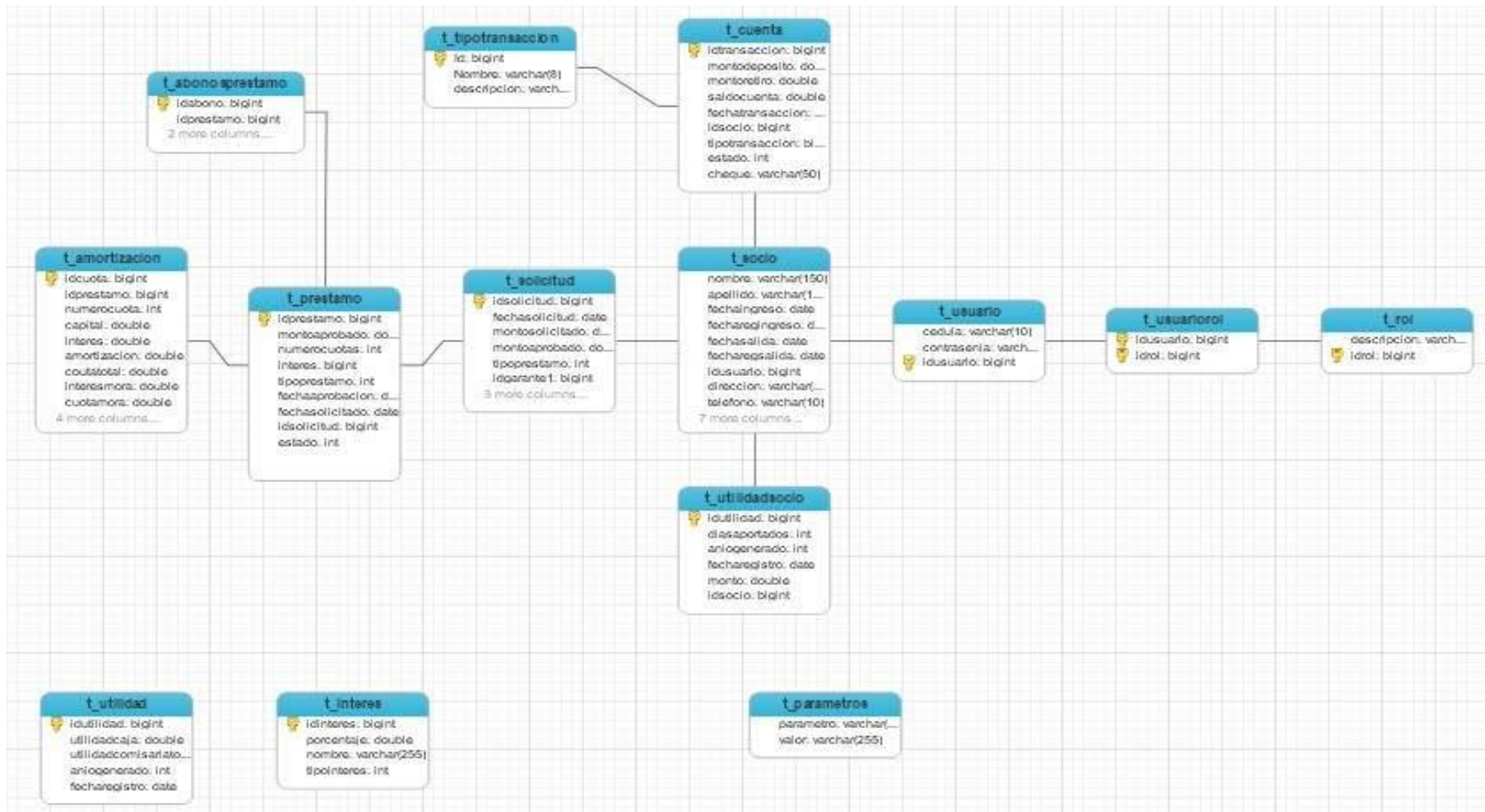


Figura 5-2: Modelo de entidad relación de la base de datos.

Realizado por: Deysi Latorre, 2018

La base datos cuenta con un total de 14 tablas de las cuales se tiene un total de 18 funciones y 26 procedimientos almacenadas los mismos que permite tener la disponibilidad de la información ya recopilada para su posterior uso en el sistema mediante el acceso a datos implementado.

2.4.2.1. Diccionario de datos

El diccionario de datos es una herramienta que se utiliza por parte de la persona encargada de su gestión y administración, para conocer cada una de las tablas de la base de datos con sus respectivos campos, tipo de dato asignada a cada campo, longitud del dato, aceptación de nulo entre otras. Que permitirán entender las características lógicas con las que cuenta el sistema para la gestión de préstamos y generación de informe contables para la caja de ahorros.(IBM, s.f.)

En la **Tabla 4-2** se puede ver que el diccionario de datos de la tabla socio, se debe definir primero el nombre de la tabla, la base de datos a la que pertenece, la fecha de creación del diccionario de datos, en campo irá el nombre de los campos que tiene la tabla, tamaño, tipo de dato que recibe, si acepta o no nulidad, los valores por defecto del campo y una descripción breve de la utilidad. El resto del diccionario de datos sigue el mismo formato y se lo visualiza en el **Anexo D**.

Tabla 4-2: Diccionario de Datos

TABLASOCIO					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de los socios					
FECHA DE CREACIÓN: 29 de enero del 2018					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
Nombre	150	varchar	NO		Nombre del socio
Apellido	150	varchar	NO		Apellido del socio
direccion	255	varchar	SI		Dirección actual donde reside el socio
Teléfono	10	varchar	SI		Número telefónico del socio

estadocivil	11	int	SI	Estado civil del socio
beneficiario	255	varchar	SI	Nombre de la persona que será beneficiaria
fechaingreso		date	SI	Fecha de ingreso a la caja
fechareingreso		date	SI	Fecha de ingreso
fechasalida		date	SI	Fecha de salida de la caja
fecharegsalida		date	SI	Fecha de ingreso
numcuenta	12	varchar	SI	Número de cuenta
estado	11	int	NO	1 Estado de la cuenta (activa/ inactiva)
idusuario	20	bigint	SI	Identificador del socio
idsocio	20	bigint	NO	Número de socio

Realizado por: Deysi Latorre, 2018

El diccionario de datos tiene un total de 14 tablas en las cuales se describe cada uno de los atributos que recibe y los formatos para su uso, permitiendo tener así información detallada de los datos aumentando así su integridad.

2.4.3. Definición de la interfaz de usuario

Determinar el bosquejo inicial de la interface de usuario del sistema SYSCAP, se realiza con el fin de tener un diseño estándar que se utiliza para la implementación del sistema para la caja de ahorro y crédito Tubasec C.A., permitiendo de esta manera trabajar con un diseño de interfaz definida y lograr con ello que el sistema sea uniforme.

Los colores serán similares a la de la página web de la empresa, teniendo como color principal el blanco y como secundario el navy, logrando contrastar con color naranja para los botones secundarios, la apariencia es distinta ya que se trata de la caja de ahorros de la empresa, como se puede ver en la **Figura 6-2** la propuesta presentada contiene el logo de la empresa, este se encuentra al lado izquierdo de la vista, el nombre se encuentra al lado derecho conformando así

la cabecera, la cual será genérica para las demás vistas, en el pie estará los derechos y los sitios como Facebook, Twitter y YouTube donde se puede encontrar información de la empresa. Seguido del encabezado en la parte central se tiene un menú con las diferentes opciones de acuerdo al rol con el que se ingrese y en la parte céntrica de la página estará el contenido generado por las funcionalidades implementadas.



Figura 6-2: Bosquejo de la interfaz de usuario

Realizado por: Deysi Latorre, 2018

Para el administrador del sistema se tiene un total de 28 vistas, en formato HTML, y para el socio se tiene un total de 8 vistas, ya que la funcionalidad implementada para este rol es menor a la del administrador. Teniendo un total de 36 vistas implementadas para todo el sistema.

2.4.4. Historias de Usuario

Las historias de usuario se utilizan en todas las metodologías ágiles como Scrum ya que son un instrumento principal para identificar los requerimientos, como una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

Las historias de usuario permiten responder rápidamente a los requisitos cambiantes, estos son el resultado de la colaboración del cliente y el desarrollador del proyecto los mismos que

plasmaron los requerimientos que debe cumplir el sistema SYSCAP. Cada Historia de usuario esta descrita por:

- **Número:** Es el identificador de la Historia de Usuario está descrita con el prefijo HU_ seguido de una numeración.
- **Nombre de la Historia:** Es el nombre descriptivo de la Historia de Usuario.
- **Usuario:** Quién utilizará la funcionalidad de la historia de usuario desarrollada.
- **Sprint:** Número del sprint.
- **Descripción:** Es una descripción resumida de la Historia de Usuario.
- **Observaciones:** Son indicaciones sobre el funcionamiento del sistema.

En la **Tabla 5-2** se presenta la historia de usuario realizada en el primer sprint, todas las historias de usuario realizadas en esta y en el resto de los Sprint siguen el mismo formato y se las visualiza en el **Anexo A**.

Tabla 5-2: Modelo de la Historia de Usuario.

HISTORIA DE USUARIO	
Número: HU_01	Listado de los socios
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 2
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo generar el listado de los socios de la caja.	
Observaciones:	
<ul style="list-style-type: none"> • Mostrará el listado de todos los socios que existen en la caja. • Se podrá buscar un socio por su apellido. 	

Realizado por: Deysi Latorre, 2018

Administración del proyecto o gestión del proyecto

Para emplear las historias de usuario correspondientes en cada Sprint para el desarrollo del mismo se toman en cuenta las siguientes consideraciones.

- Las Historias se convierten en tareas al comienzo del Sprint.
- La historia de usuario sirve para comprobar la funcionalidad descrita
- La realización de las tareas de ingeniería se estima en días ideales.

El proyecto cuenta con un total 26 historias de usuario, los cuales tienen requerimientos funcionales, así como metáforas del sistema. De las cuales se han generado un total 42 tareas de ingeniería las mismas que describen en el punto 2.4.5.

2.4.5. Tareas de ingeniería

Las tareas de ingeniería son actividades que se realizan para cumplir con el desarrollo correcto del requerimiento especificado de la historia de usuario, permitiendo distribuir el trabajo de las historias en tiempos más pequeños y fáciles de realizar.

En la **Tabla 6-2** se describe la tarea de ingeniería TI_1 correspondiente a la MT_02 para mostrar el formato que se siguió para cada una de las tareas adjuntas en el **Anexo B**.

Tabla 6-2: Formato de Tarea de Ingeniería

TAREA DE INGENIERÍA	
Historia de Usuario: MT_02 Diseño de la base de datos	
Número de la Tarea: TI_1	Nombre de Tarea: Diccionario de datos correspondiente
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Realizar el respectivo diccionario de datos de la base SYSCAP	
Prueba de aceptación: Verificar que el diccionario de datos cuente con los campos: nombre tabla, base de datos que pertenece, fecha de creación, campo, tamaño, tipo de dato, nulidad, valor por defecto y descripción.	

Realizado por: Deysi Latorre, 2018

Se tiene un total de 42 tareas de ingeniería de las historias de usuario, la tarea de ingeniería más larga tiene una duración de 3 días y la tarea de ingeniera más corta tiene una duración de 1 día, cada una de estas tareas se cumplen al 100% con lo especificado en las mismas.

Pruebas de Aceptación. – son condiciones que debe cumplir la tarea de ingeniería para considerarse correctamente desarrollada.

La **Tabla 7-2**, muestra una prueba de aceptación con los campos que incluyen las tareas de ingeniería del **Anexo B**.

Tabla 7-2: Formato de Pruebas de Aceptación

Prueba de Aceptación	
Código: PA_01	Historia de Usuario: Diseño de Base de Datos
Nombre: Que el diccionario de datos tenga las columnas descritas.	
Responsable: Deysi Latorre	
Descripción: El diccionario de datos debe tener documentadas todas las tablas de la base de datos, con las columnas especificadas.	
Condiciones de Ejecución: El responsable revisará visualmente que cada una de las tablas estén documentadas en el diccionario de datos.	
Pasos de ejecución: Revisar si están el número de tablas descritas. Luego revisar si todas las tablas cuentan con las columnas descritas.	

Realizado por: Deysi Latorre, 2018

Las pruebas de aceptación propuestas en las tareas de ingeniería dan un total de 44 pruebas de aceptación **Anexo C**, cada una de estas pruebas se cumplen al 100% con lo especificado, dando así un 100% de cumplimiento de todas las pruebas y la funcionalidad descrita que debe tener.

2.5. *Gestión del proyecto*

Según la planificación elaborada antes de iniciar el sistema, este cuenta con catorce Sprints los cuales se asignaron varias tareas a desarrollar, estos se avanzaron con éxito, aunque los tiempos que fueron especificados variaron un poco, tomando en consideración la curva de aprendizaje de la tecnología a implementarse. En la **Gráfica 1-2**, se puede ver de forma paralela el avance del proyecto en cuanto a su desarrollo.

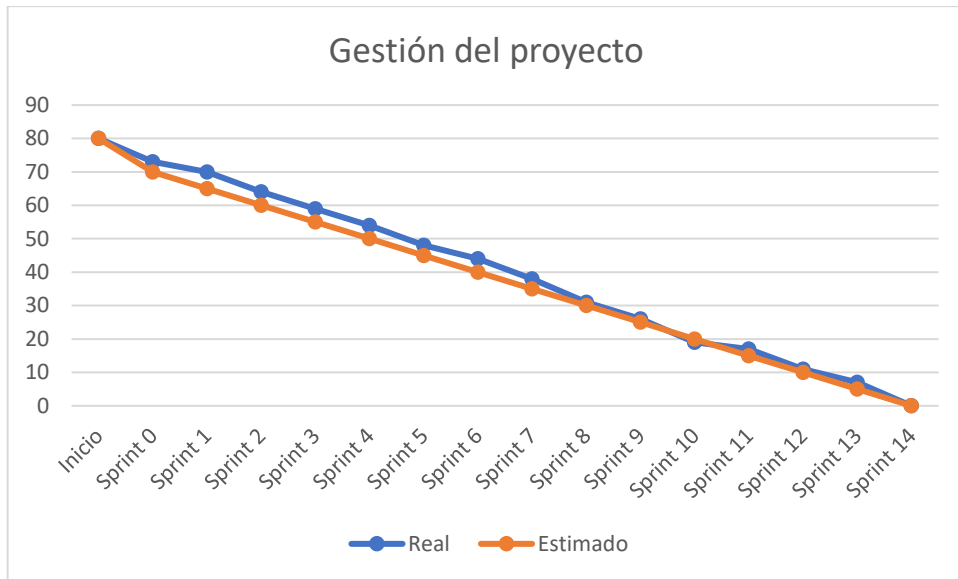


Gráfico 1-2: Gráfica representativa del avance del desarrollo del proyecto.

Realizado por: Deysi Latorre, 2018

En el gráfico de la gestión del proyecto, muestra que en el sprint 1, 4, 6 y 11, se toma un día más del tiempo estimado, mientras que en los sprint 7, 9 y 13 el tiempo de la ejecución de las tareas lleva dos días adicionales a lo planificado, sin embargo se recupera estos tiempos en los sprint 0, 5, 10 y 12, ya cada uno de estos se toma tres, dos y cuatro días de los cinco planificados, mientras que en los sprint 2, 3 y 8 las tareas toman el tiempo estimado.

2.6. Población y muestra

Utilizando la estadística descriptiva se procede al cálculo de la muestra de un infinito cualitativo para determinar el número de veces a tomarlos de cada uno de los procesos automatizados mediante el sistema.

El tamaño de la muestra se calcula en base al universo infinito cualitativo aplicando la fórmula, dado por;

$$n = \frac{Z^2 p(1 - p)}{e^2}$$

Donde los valores aplicados a la fórmula se tienen de:

Z= 1.96 nivel de confianza que es un 95%

e= con un error del 5%

p= 0.5 que identifica la proporción a encontrar

Se procede al cálculo de la muestra utilizando la fórmula planteada con los valores especificados, dando como resultado una muestra de 384 tiempos a recoger de cada uno de los procesos.

$$n = \frac{0.95^2 * 0.5(1 - 0.5)}{0.05^2} = 384$$

Con el cálculo de la media nos permite hacer un análisis y mediante la representación de los datos demostrar la reducción del tiempo en cada uno de los procesos, ya que el objetivo de la estadística descriptiva es **describir** las características y comportamientos mediante **medidas de resumen, tablas o gráficos**.

CAPÍTULO III

3. MARCO DE RESULTADOS

En el marco de resultados se presentó un análisis de los resultados obtenidos al aplicar la prueba de reducción de tiempo sobre el sistema desarrollado para la Caja de Ahorros, con la utilización del framework AngularJS, además el resultado nos dio una idea clara de lo que el usuario piensa sobre la aplicación.

Para la realización de la prueba de reducción de tiempo se utilizó la estadística descriptiva la cual facilitó la comparación entre el tiempo actual y con el sistema, de esta manera se observó si hubo reducción del tiempo en los diferentes procesos que se realizan en la caja.

3.1. Demostración de reducción de tiempo

Dada la recolección de tiempo que toma en entregar un préstamo, realizar en cierre de caja y la generación los reportes de aportaciones y préstamos mensuales, al generarlo de forma manual se puede ver en la **Tabla 1-3**, los tiempos se tomaron en conjunto con la encargada de la caja los cuales nos sirvieron para poder hacer una comparación entre estos y con el sistema implementado, los mismo que fueron tomados mediante la observación en la realización del proceso y medidos con un cronómetro.

De igual manera se obtuvo los tiempos al momento de la generación de reportes los cuales la encargada supo manifestar que, el tiempo que conlleva en generar un informe tanto de aportaciones como de préstamos es de 1 día ya que debe revisar las tarjetas de cada uno de los socios dependiendo del informe que se esté realizando, y el tiempo que tarda en realizar el cierre de caja va de 1 a 6 meses, ya que la encargada no se dedica exclusivamente a realizarla ya que tiene que cumplir diferentes actividades.

Para poder realizar la comparación se sacó un promedio en el proceso de entrega de préstamos y en los informes contables como son cierre de caja, reporte de aportaciones y reporte de préstamos.

Tabla 1-3: Tabla de tiempos manuales

Proceso	Porcentaje
Entrega de Préstamo	2 horas
Cierre de Caja	480 horas
Reporte Aportaciones	8 horas
Reporte Préstamos	8 horas

Realizado por: Deysi Latorre, 2018

Con el sistema se han realizado los mismos procesos y el tiempo que se ha tomado en realizar dichos procesos con las mismas condiciones, los tiempos se midieron con la ayuda de un cronómetro al momento en que la encargada realizaba los procesos para entregar el préstamo y de igual forma para generar los reportes.

Una vez que se determina el tiempo promedio al ejecutar cada uno de los procesos con ayuda del sistema, estos valores se los deja en una unidad de tiempo igual a la que se tiene en los tiempos de los procesos manuales, con el fin de poder realizar las comparaciones entre ambos tiempos, esta se puede ver en la **Tabla 2-3**.

Tabla 2-3: Porcentaje de tiempos en una unidad de tiempo.

Proceso	Porcentaje
Entrega de Préstamo	0.04 horas (144 seg)
Cierre de Caja	0,0029 horas (10.48 seg)
Reporte Aportaciones	0,0033 horas (12.02 seg)
Reporte Préstamos	0,0035 horas (12.53 seg)

Realizado por: Deysi Latorre, 2018

3.1.1. Análisis de resultados

Para obtener un porcentaje de la reducción de tiempo que se ha obtenido al implementar el sistema, se aplicó una regla de tres sobre los valores de tiempo manuales con los valores de tiempo del sistema, donde se tomó el valor manual como el 100% del tiempo y se calcula el porcentaje del tiempo del sistema en relación al tiempo manual.

- Proceso de Préstamo

Tabla 3-3: Tabla de tiempo de préstamo.

Tiempo	Promedio
Manual	2 horas
Sistema	0,04 horas
Porcentaje	$P = \frac{0.04h * 100\%}{2h} = 2\%$

Realizado por: Deysi Latorre, 2018

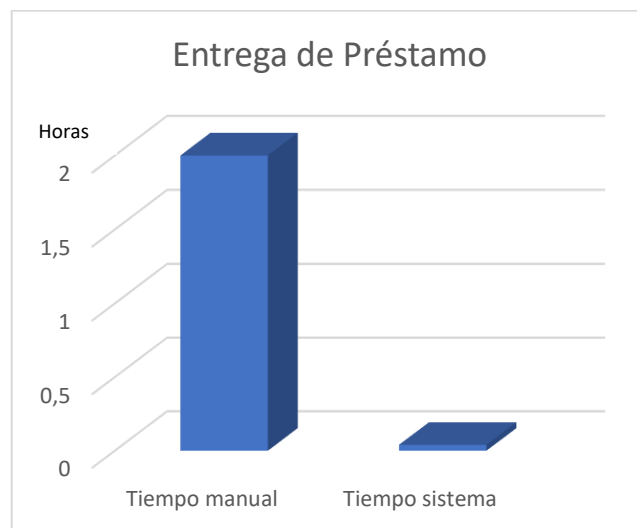


Gráfico 1-3: Resultado del proceso de préstamo.

Realizado por: Deysi Latorre, 2018

Como se observa en la **Tabla 3-3** y en el **Gráfico 1-3**, el tiempo de entrega del préstamo a un socio ha disminuido notablemente, teniendo en cuenta que al realizarlo de forma manual le

llevaba 2 horas en el proceso mientras que con el sistema se tiene un porcentaje del 98% de reducción de tiempo.

- Proceso Cierre de Caja

Tabla 4-3: Tabla de tiempo de cierre de caja.

Tiempo	Promedio
Manual	480 horas
Sistema	0.0029 horas
Porcentaje	$P = \frac{0.0029h * 100\%}{480h} = 0.00006\%$

Realizado por: Deysi Latorre, 2018

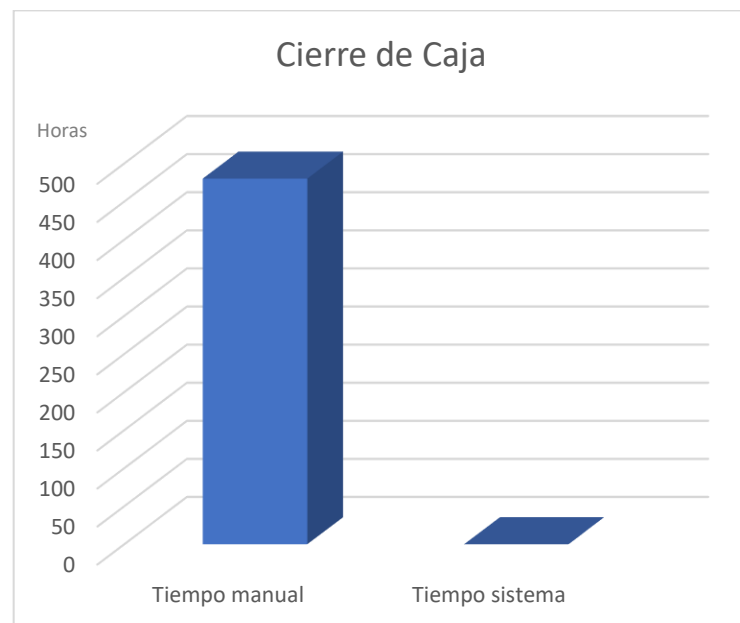


Gráfico 2-3: Resultado del proceso de cierre de caja.

Realizado por: Deysi Latorre, 2018

En la **Tabla 4-3** y en el **Gráfico 2-3**, el tiempo al realizar el cierre de caja mensual ha variado notablemente, teniendo en cuenta que al realizarlo de forma manual le llevaba 480 horas (3

mes) como promedio mientras que con el sistema se tiene un porcentaje del 99.99% de reducción de tiempo.

- Proceso Reporte Aportaciones

Tabla 5-3: Tabla de tiempo de reporte aportaciones.

Tiempo	Promedio
Manual	8 horas
Sistema	0.0033 horas
Porcentaje	$P = \frac{0.0033h * 100\%}{8h} = 0.04125\%$

Realizado por: Deysi Latorre, 2018



Gráfico 3-3: Resultado del reporte de aportaciones.

Realizado por: Deysi Latorre, 2018

Como se puede ver en la **Tabla 5-3** y en el **Gráfico 3-3**, el tiempo al generar el reporte de aportaciones disminuyó significativamente, ya que al realizarlo manualmente la encargada se

demoraba 1 día, mientras que con el sistema se tiene un porcentaje de 99.95% de reducción de tiempo.

- Proceso Reporte de Préstamo

Tabla 6-3: Tabla de tiempo de reporte de préstamo.

Tiempo	Valor
Manual	8 horas
Sistema	0.0035 horas
Porcentaje	$PCr = \frac{0.0035h * 100\%}{8h} = 0.04375\%$

Realizado por: Deysi Latorre, 2018



Gráfico 4-3: Resultado del reporte de préstamos.

Realizado por: Deysi Latorre, 2018

Como se observa en la **Tabla 6-3** y en el **Gráfico 4-3**, el tiempo al generar el reporte de préstamos se redujo notablemente, teniendo en cuenta que tardaba 1 día en realizarlo de forma manual, mientras que con el sistema se tiene un porcentaje de 99.95% de reducción de tiempo al realizar este proceso.

Para obtener un porcentaje final de los tiempos en los que la administradora de la caja tarda en realizar los procesos de manera manual y con el sistema, se suman y promedian los tiempos obtenidos para totalizar y tener un tiempo general en el que todos los procesos se realizan, con esto se demuestra que al utilizar el sistema existe una reducción de tiempo en relación al tiempo manual, este cálculo se puede apreciar en la **Tabla 7-3**.

Tabla 7-3: Tabla de tiempos.

Proceso	Tiempo Manual	Tiempo sistema
Entrega de Préstamo	2 horas	0.04 horas (144 seg)
Cierre de Caja	480 horas	0,0029 horas (10.48 seg)
Reporte Aportaciones	8 horas	0,0033 horas (12.02 seg)
Reporte Préstamos	8 horas	0,0035 horas (12.53 seg)
Total	498 horas	0,0497 horas
Promedio	124.5	0.012
Porcentaje	100 %	0.0096 %

Realizado por: Deysi Latorre, 2018

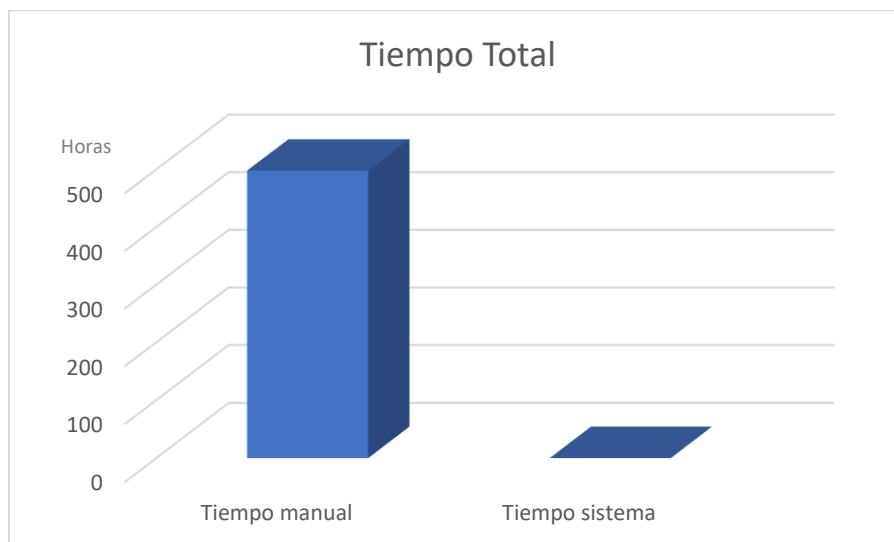


Gráfico 5-3: Resultado total de los procesos.

Realizado por: Deysi Latorre, 2018

Se aprecia en el **Grafico 5-3**, que el tiempo para la realización de los procesos ha reducido hasta un 99.9904 % con la implementación del sistema, cabe recalcar que la realización de los procesos de forma manual se los realiza secuencialmente ya que debe culminar cada una de las actividades para empezar la siguiente, mientras que con la ayuda del sistema los procesos automatizados se los pueden realizar en forma paralela o de forma secuencial, según como la encargada de la caja requiera.

CONCLUSIONES

- El sistema SYSCAP reduce el tiempo en cada uno de los procesos, estos tiempos han disminuido de forma considerable ya que se tiene una reducción del 98% en la entrega de préstamos y del 99.96 % en la generación de los informes contables de forma general.
- Para la entrega de un préstamo se determina que el tiempo en que este proceso es de alrededor de 2 horas realizándolo de forma manual, mientras que para la generación de los informes contables estos pueden tardar hasta seis meses.
- Para la elaboración del sistema se obtiene un total de 26 requerimientos, los mismos que fueron otorgados por parte de la administradora de la caja de ahorros de los empleados de la empresa Tubasec C.A.
- La utilización de AngularJS para el desarrollo e implementación de páginas web permite que la aplicación sea de una sola página disminuyendo con ello el tiempo de carga y de respuesta del servidor, mejorando con ello la productividad de la aplicación con cada uno de los procesos con los que cuenta.
- El sistema web para la caja de ahorros de la empresa Tubasec C.A. tiene automatizado los procesos de entrega de préstamos con una reducción de tiempo del 98%, cierre de caja con una reducción de tiempo del 99.99% y en los reportes tanto de aportaciones como de préstamos una la reducción de tiempo del 99.95%.

RECOMENDACIONES

- Se debe ampliar los estudios de AngularJS para buscar otras maneras de crear código para la generación de reportes, ya que las librerías con las que trabaja, no permiten generarlos de forma automática.
- Se debe considerar la información con la que cuenta una tecnología antes de implementarla o utilizarla, sin dejar de lado el investigar el avance de la información y de la tecnología que se genera día a día.
- Desarrollar los requerimientos recolectados inicialmente para evitar inconvenientes futuros al momento del desarrollo del sistema, ya que puede existir aumentos de requerimientos que dificulten el tiempo de entrega.

BIBLIOGRAFÍA

ALBALADEJO, Javier. *¿Qué es SCRUM?* [En línea]. ProyectosAgiles.com, 2012. [Consulta: 15 marzo 2017]. Disponible en: <https://proyectosagiles.org/que-es-scrum/>.

ALVAREZ, Cecilio. *Angular.js módulos y arquitectura* [En línea], 2015. [Consulta: 10 junio 2017]. Disponible en: <https://www.genbetadev.com/desarrollo-web/angular-js-modulos-y-arquitectura>.

ALVAREZ, Miguel Angel. *¿Qué es Java?* [En línea]. DesarrolloWeb.com, 2001. [Consulta: 10 mayo 2016]. Disponible en: <https://desarrolloweb.com/articulos/497.php>.

AZAUSTRE, Carlos. *¿Qué es AngularJS? Primeros pasos para aprenderlo* [Blog]. Madrid-España, 2013. [Consulta: 16 mayo 2017]. Disponible en: <https://carlosazaustre.es/blog/empezando-con-angular-js/>.

BOAVENTURA, Raoni. *Un tutorial pasoa paso para tu primera aplicación AngularJS* [En línea]. Toptal, 2012. [Consulta: 20 octubre 2017]. Disponible en: <https://www.toptal.com/angular-js/un-tutorial-paso-a-paso-para-tu-primera-aplicaci%C3%B3n-angularjs/es>.

BRAVO MARTINEZ, Indira. *¿Qué es MySQL?* [Blog]. Informatica,2007. [Consulta: 05 junio 2017]. Disponible en: <http://indira-informatica.blogspot.com/2007/09/que-es-mysql.html>.

GAJARDO RIFFO, Sandra. *Estadística Descriptiva* [En línea]. SlidePlaye, 2015. [Consulta: 21 mayo 2017]. Disponible en: <http://slideplayer.es/slide/1729731/>

GONZALEZ, Lorenzo. *Arquitectura en 3 capas y modelos ricos en AngularJS* [En línea], 2015 [Consulta: 13 junio 2017]. Disponible ent: <http://slides.com/logongas/arquitectura-3-capas-modelos-ricos#/>.

GRIMALDE DE QUIJADA, Yenny Verónica. *Sistema web para el control de la caja de ahorros de SENECA (CATSENE)*. (Tesis de pregrado) [En línea]. Universidad Nacional Abierta, Área de Ingeniería, Ingeniería de Sistemas. (La Asunción - Venezuela), 2011. pp.1-222.

[Consulta: 25 enero 2017]. Disponible en: <http://biblo.una.edu.ve/docu.7/bases/marc/texto/t37721b.pdf>.

Introducción a Java [En línea]. Aprende a Programar.com, 2015. [Consulta: 12 junio 2017]. Disponible en: <http://www.aprendeaprogramar.com/mod/resource/view.php?id=204>

LAZARO, Pablo. *¿Qué es AngularJS? Una breve introducción* [Blog], 2013. [Consulta: 12 mayo 2017]. Disponible en: <http://pablolazarodev.blogspot.com/2013/05/que-es-angularjs-una-breve-introduccion.html>.

MEJÍA MOYANO, Gustavo Wladimir. *Desarrollo de un sistema web para administración de eventos y control de entrada y salida de empleados caso de estudio: centro de formación social "BETHANIA"* [En línea]. Pontificia Universidad Católica del Ecuador, Facultad de Ingeniería, Escuela de Sistemas ,(Quito - Ecuador) 2015. pp. 37-38. [Consulta: 10 mayo 2016]. Disponible en: <http://repositorio.puce.edu.ec/bitstream/handle/22000/8105/Tesis-GWMM-Entrega-vFinal.pdf?sequence=1>.

MINAR, Igor. *MVC vs MVVM vs MVP*[En línea], 2012. [Consulta: 13 junio 2017]. Disponible en: <https://plus.google.com/+AngularJS/posts/aZNVhj355G2>.

MOROCHO GUEVARA, Juan Carlos & FUERTES ORTEGA,Victor Miguel. *Análisis del patrón modelo vista cotrolador implementado en lenguajes se Software libre para el desarrollo de aplicaciones web. Caso práctico: Liceo de talentos STEPHEN HAWKING.* [En línea]. Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería en Sistemas, (Riobamba - Ecuador), 2010. pp.20-24. [Consulta: 12 febrero 2017]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/550/1/18T00441.pdf>

PEÑA ARPI, Jorge Favián & CAMBISACA SANCHEZ, Miltón Geovanny. *Análisis comparativo entre los frameworks javascript mvc, angularjs y ember js para el desarrollo de aplicaciones web. Caso práctico: " Sistema de control de botiquin veterinario para el magap, Morona Santiago".* [En línea]. Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería en Sistemas, (Riobamba - Ecuador), 2015. pp.16-21. [Consulta: 10 junio 2017]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/4583>.

¿Qué es un diccionario de datos? [En línea]. IBM. [Consulta: 03 octubre 2017]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/SSCVKV_9.1.2/Campaign/DatabaseTableAdmin/What_is_a_data_dictionary.html.

RAUH, Stephan. *Model-View-Whatever* [En línea]. Beyond Java, 2015. [Consulta: 15 julio 2017]. Disponible en: <https://www.beyondjava.net/blog/model-view-whatever/>.

Using the Model-View-ViewModel (MVVM) [En línea], Microsoft. [Consulta: 13 junio 2017]. Disponible en: <https://msdn.microsoft.com/es-xl/library/windows/apps/jj160324.aspx>.

VALBUENA APONTE, Angel María. *Guía comparativa de frameworks para los lenguajes html 5,css y javascript para el desarrollo de aplicaciones web.* (Tesis de pregrado) [En línea]. Universidad Tecnológica de Pereira, Facultad de Ingenierías, Ingeniería de Sistemas y Computación. (Pereira - Colombia), 2014. pp.38-40. [Consulta:15 junio 2017]. Disponible en: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/4577/00676V865.pdf?sequence=1&isAllowed=y>.

Victoria, *Definición de MySQL* [En línea], DefiniciónABC, 2009. [Consulta: 13 mayo 2017]. Disponible en: <https://www.definicionabc.com/tecnologia/mysql.php>.

ANEXOS

Anexo A: Historias de Usuario

HISTORIA DE USUARIO	
Número: MT_02	Diseño la base de datos
Modificación de la historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 0
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 día
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo diseñar una base de datos para mantener los mismos disponibles en el transcurso del tiempo	
Observaciones: <ul style="list-style-type: none">• Se utilizará el motor de base de datos MySQL.• Se requiere el diccionario de datos correspondiente.	

HISTORIA DE USUARIO	
Número: MT_03	Implementación la base de datos
Modificación de la historia de usuario:	
Usuario: Administrador	Sprint Asignada: 0
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 día
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como usuario deseo tener mis datos disponibles en el tiempo para su posterior gestión y consulta.	
Observaciones:	

HISTORIA DE USUARIO	
Número: MT_05	Encriptación de la clave
Modificación de la historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 1
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo tener encriptada la clave de acceso al sistema.	
Observaciones:	
<ul style="list-style-type: none"> • La clave de usuario será ocultada con puntos. 	

HISTORIA DE USUARIO	
Número: MT_07	Diseño de la interfaz de usuario
Modificación de la historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 1
Prioridad en el Negocio: Baja	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo diseñar las interfaces a utilizar en el sistema para la interacción con el usuario	
Observaciones:	
<ul style="list-style-type: none"> • Se utilizará interfaces de página simple (SPA). • Se rediseñarán las interfaces del sistema SYSCAP en caso que sean requeridas. 	

HISTORIA DE USUARIO	
Número: HU_01	Listado de los socios
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 2
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 3 días
Descripción: Como desarrollador deseo generar el listado de los socios de la caja.	
Observaciones:	
<ul style="list-style-type: none"> • Mostrará el listado de todos los socios que existen en la caja. • Se podrá buscar un socio por su apellido. 	

HISTORIA DE USUARIO	
Número: HU_02	Registro de ingreso de un socio
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 2
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 3 días
Descripción: Como secretaria deseo poder ingresar los datos de un socio al sistema mediante el consumo de una base existente	
Observaciones:	
<ul style="list-style-type: none"> • Se ingresará la cédula del socio sin guion. • Si el socio se insertó correctamente mostrará un mensaje. 	

HISTORIA DE USUARIO	
Número: HU_03	Actualizar los datos del socio
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 3
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como secretaria deseo poder actualizar los datos de un socio para mantener datos correctos	
Observaciones:	
<ul style="list-style-type: none"> • Se puede editar toda la información de un socio excepto la cédula. • Al guardar los cambios mostrara un mensaje. 	

HISTORIA DE USUARIO	
Número: HU_04	Eliminar los datos de un socio
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 3
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 3 días
Descripción: Como desarrollador deseo poder eliminar los datos de un socio para que no sean tomados en cuenta al momento de un reporte	
Observaciones:	
<ul style="list-style-type: none"> • Al eliminar un socio deberá registrar la fecha de salida y mostrará un mensaje. 	

HISTORIA DE USUARIO	
Número: HU_05	Registro de ingreso de una aportación
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 4
Prioridad en el Negocio: Alta	Tiempo Estimado: 1 día
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo registrar el ingreso de las aportaciones de un socio para tener un control de los ingresos	
Observaciones:	
<ul style="list-style-type: none"> Las aportaciones se harán según el monto que el socio decida poner. Al guardar la aportación se generará automáticamente un recibo del mismo. 	

HISTORIA DE USUARIO	
Número: HU_06	Actualizar los datos de una aportación
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 4
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo actualizar los datos de una aportación de un socio para tener correcto los ingresos	
Observaciones:	
<ul style="list-style-type: none"> Se podrá actualizar la información de la aportación. 	

HISTORIA DE USUARIO	
Número: HU_07	Eliminar los datos de una aportación
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 4
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo eliminar una aportación de un socio.	
Observaciones:	

HISTORIA DE USUARIO	
Número: HU_08	Ingresar la solicitud de un préstamo
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 5
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 día
Descripción: Como desarrollador deseo poder ingresar la solicitud de préstamo para tener un control de estos.	
Observaciones:	
<ul style="list-style-type: none"> • Al ingresar la solicitud mostrara un mensaje. • La solicitud será aprobada o rechazada por la persona encargada. 	

HISTORIA DE USUARIO	
Número: HU_09	Estado de un préstamo
Modificación de la historia de usuario:	
Usuario: Socio	Sprint Asignada: 5
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 4 día
Descripción: Como socio deseo poder ver el estado de mi préstamo	
Observaciones:	
<ul style="list-style-type: none"> • El socio solo podrá ver el estado de préstamo sin ninguna opción. 	

HISTORIA DE USUARIO	
Número: HU_10	Registro el ingreso de un préstamo
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 6
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo poder ingresar el préstamo de un socio, para tener un control de los mismos.	
Observaciones:	
<ul style="list-style-type: none"> • Si el préstamo es aprobado se imprimirá la solicitud para que el socio firme. 	

HISTORIA DE USUARIO	
Número: HU_11	Actualizar los datos de un préstamo
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 6
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo poder actualizar los datos de un préstamo para tener los datos correctos.	
Observaciones:	

HISTORIA DE USUARIO	
Número: HU_12	Registro del ingreso de la cuota del préstamo
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 6
Prioridad en el Negocio: Alta	Tiempo Estimado: 1 día
Riesgo en el Desarrollo: Media	Tiempo Real: 2 día
Descripción: Como desarrollador deseo poder registrar una cuota de un préstamo para tener un control de estos.	
Observaciones:	

HISTORIA DE USUARIO	
Número: HU_13	Generar la tabla de amortización de un préstamo
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 7
Prioridad en el Negocio: Alta	Tiempo Estimado: 5 días
Riesgo en el Desarrollo: Media	Tiempo Real: 6 días
Descripción: Como desarrollador deseo poder generar la tabla de amortización de un préstamo para dar a conocer la cuota mensual.	
Observaciones:	
<ul style="list-style-type: none"> Se generará la tabla de amortización dependiendo del método que se escoja y el plazo que desee. 	

HISTORIA DE USUARIO	
Número: HU_14	Inicio de sesión del usuario
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 8
Prioridad en el Negocio: Alta	Tiempo Estimado: 5 días
Riesgo en el Desarrollo: Media	Tiempo Real: 7 días
Descripción: Como desarrollador deseo poder ingresar al sistema para utilizar la información existente	

HISTORIA DE USUARIO	
Número: HU_15	Eliminar los datos un préstamo
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 9
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo poder eliminar los datos de un préstamo para tener un control de los mismos.	
Observaciones:	
<ul style="list-style-type: none"> • Se eliminará el préstamo que no haya sido entregado el dinero. 	

HISTORIA DE USUARIO	
Número: HU_16	Reporte de las aportaciones del día
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 10
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 3 días
Descripción: Como desarrollador deseo generar un reporte de las aportaciones realizadas en un día.	
Observaciones:	
<ul style="list-style-type: none"> • Mostrará las aportaciones de un día permitiendo descargarlo. 	

HISTORIA DE USUARIO	
Número: HU_17	Reporte de las personas que no han aportando
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 10
Prioridad en el Negocio: Alta	Tiempo Estimado: 1 día
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo conocer los socios que no han aportado en un mes	
Observaciones:	
<ul style="list-style-type: none"> Mostrará los socios que no aportaron en un mes permitiendo descargarlo. 	

HISTORIA DE USUARIO	
Número: HU_18	Reporte de los préstamos entregados
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 10
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2días
Descripción: Como desarrollador deseo conocer los préstamos entregados en un mes para tener un control de estos.	
Observaciones:	
<ul style="list-style-type: none"> Mostrará los préstamos entregados en un mes permitiendo descargarlo. 	

HISTORIA DE USUARIO	
Número: HU_19	Reporte de las personas que han tenido atraso en sus préstamos
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 11
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 3 días
Descripción: Como desarrollador deseo poder conocer a los socios que han tenido retraso en sus préstamos.	
Observaciones:	
<ul style="list-style-type: none"> Mostrará los préstamos entregados en un mes permitiendo descargarlo 	

HISTORIA DE USUARIO	
Número: HU_20	Ingresar la utilidad de la caja
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 11
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo poder ingresar la utilidad de la caja.	
Observaciones:	
<ul style="list-style-type: none"> Se tendrá un campo donde se ingresará la utilidad y mostrará un mensaje. 	

HISTORIA DE USUARIO	
Número: HU_21	Ingresar la utilidad del comisariato
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 12
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo poder ingresar la utilidad que existe en el comisariato.	
Observaciones:	
<ul style="list-style-type: none"> Se tendrá un campo donde se ingresará la utilidad del comisariato y mostrará un mensaje. 	

HISTORIA DE USUARIO	
Número: HU_22	Calcular la utilidad de un socio que se retire
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 12
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 4 días
Descripción: Como desarrollador deseo poder calcular la utilidad de un socio que desee retirarse de la caja.	
Observaciones:	
<ul style="list-style-type: none"> Mostrará la utilidad de un socio. 	

HISTORIA DE USUARIO	
Número: HU_23	Generar las utilidades de los socios
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 13
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo poder generar las utilidades de los socios.	
Observaciones:	
<ul style="list-style-type: none"> • Mostrará el cálculo de las utilidades de los socios. 	

HISTORIA DE USUARIO	
Número: HU_24	Consultar el saldo disponible
Modificación de la historia de usuario:	
Usuario: Socia (o)	Sprint Asignada: 13
Prioridad en el Negocio: Alta	Tiempo Estimado: 2 días
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo poder conocer mi saldo disponible.	
Observaciones:	
<ul style="list-style-type: none"> • El socio podrá ver su saldo disponible, digitando su número de cuenta. 	

HISTORIA DE USUARIO	
Número: HU_25	Fecha de pago del préstamo
Modificación de la historia de usuario:	
Usuario: Socia (o)	Sprint Asignada: 13
Prioridad en el Negocio: Alta	Tiempo Estimado: 1 día
Riesgo en el Desarrollo: Media	Tiempo Real: 1 día
Descripción: Como desarrollador deseo poder conocer la fecha de pago de préstamo.	
Observaciones:	
<ul style="list-style-type: none"> • Mostrará la fecha de pago de préstamo digitando su número de cuenta. 	

HISTORIA DE USUARIO	
Número: HU_26	Cierre de caja mensual
Modificación de la historia de usuario:	
Usuario: Secretaria (o)	Sprint Asignada: 14
Prioridad en el Negocio: Alta	Tiempo Estimado: 3 días
Riesgo en el Desarrollo: Media	Tiempo Real: 2 días
Descripción: Como desarrollador deseo realizar el cierre de caja mensual para tener un control de este.	
Observaciones:	
<ul style="list-style-type: none"> Al seleccionar cierre de caja mostrará las transacciones que se realizaron en el mes y se podrá descargar. 	

HISTORIA DE USUARIO	
Número: MT_08	Despliegue de la aplicación en el servidor
Modificación de la historia de usuario:	
Usuario: Administrador	Sprint Asignada: 14
Prioridad en el Negocio: Alta	Tiempo Estimado: 5 días
Riesgo en el Desarrollo: Media	Tiempo Real: 7 días
Descripción: Como administrador deseo tener la aplicación disponible a través de internet en cualquier ubicación geográfica para poder utilizarla.	
Observaciones:	
<ul style="list-style-type: none"> Utilizará un servidor existente en la empresa. 	

Anexo B: Tareas de Ingeniería.

TAREA DE INGENIERÍA	
Historia de Usuario: MT_02 Diseño de la base de datos	
Número de la Tarea: TI_1	Nombre de Tarea: Diccionario de datos correspondiente
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Realizar el respectivo diccionario de datos de la base SYSCAP	
Prueba de aceptación: Verificar que el diccionario de datos cuente con los campos: nombre tabla, base de datos que pertenece, fecha de creación, campo, tamaño, tipo de dato, nulidad, valor por defecto y descripción.	

TAREA DE INGENIERÍA	
Historia de Usuario: MT_05 Encriptación de la clave	
Número de la Tarea: TI_2	Nombre de Tarea: La clave de usuario será ocultada con puntos
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Se encrestará la clave de usuario	
Prueba de aceptación: Se mostrarán puntos en la clave del usuario.	

TAREA DE INGENIERÍA	
Historia de Usuario: MT_07 Diseño de la interfaz de usuario	
Número de la Tarea: TI_3	Nombre de Tarea: Definición del estándar de diseño de interfaces.
Tipo de Tarea: Desarrollo	Tiempo Estimado: 3 día
Programador Responsable: Deysi Latorre	
Descripción: Definir las características, colores, fuentes, funcionalidades del sistema.	
Prueba de aceptación: Verificar que el diseño cumpla con la identidad de la empresa.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_01 Listado de los socios	
Número de la Tarea: TI_4	Nombre de Tarea: Mostrar el listado de los socios
Tipo de Tarea: Desarrollo	Tiempo Estimado: 3 día
Programador Responsable: Deysi Latorre	
Descripción: Tabla se comunica con el controller correspondiente para mostrar un ArrayList con todos los socios existentes en la base.	
Prueba de aceptación: La tabla se cargará al abrir la pestaña de socio. Muestra la cantidad de registros que existen en la base de datos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_02 Registro de ingreso de un socio	
Número de la Tarea: TI_5	Nombre de Tarea: Mostrar diálogo de nuevo socio
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para ingresar la información del nuevo socio.	
Prueba de aceptación: Al dar clic en el botón nuevo se despliega el dialogo con los campos vacíos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_02 Registro de ingreso de un socio	
Número de la Tarea: TI_6	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función insertar, recibe por parámetro un objeto de la clase socio y retorna un mensaje si fue o no insertado el socio, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_03 Actualizar los datos del socio	
Número de la Tarea: TI_7	Nombre de Tarea: Mostrar diálogo del socio seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener los datos del socio correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de actualizar. Al dar clic en el botón se despliega un diálogo con los datos del socio correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_03 Actualizar los datos del socio	
Número de la Tarea: TI_8	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 1 día
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función actualizar, recibe por parámetro un objeto de la clase socio y retorna un mensaje si fue o no modificados los datos, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_04 Eliminar los datos de un socio	
Número de la Tarea: TI_9	Nombre de Tarea: Mostrar diálogo del socio seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener los datos del socio correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de eliminar. Al dar clic en el botón se despliega un diálogo con los datos del socio correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_04 Eliminar los datos del socio	
Número de la Tarea: TI_10	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función eliminar, recibe por parámetro un entero correspondiente al código y retorna un mensaje si fue o no registrada la salida del socio, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_05 Registro de ingreso de una aportación	
Número de la Tarea: TI_11	Nombre de Tarea: Mostrar el diálogo de la nueva aportación
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para ingresar la información de la nueva aportación.	
Prueba de aceptación: Al dar clic en el botón nueva aportación se despliega el diálogo con los campos vacíos. Se debe ingresar el apellido del socio para verificar sus datos y registrar su aportación. Al guardar la aportación se descargará automáticamente un recibo.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_05 Registro de ingreso de una aportación	
Número de la Tarea: TI_12	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función insertar, recibe por parámetro un objeto de la clase cuenta y retorna un mensaje si fue o no registrada la transacción, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_06 Actualizar los datos de una aportación	
Número de la Tarea: TI_13	Nombre de Tarea: Mostrar diálogo de aportación seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener los datos de la aportación correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de actualizar. Al dar clic en el botón se despliega un diálogo con los datos de la aportación correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_06 Actualizar los datos de una aportación	
Número de la Tarea: TI_14	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función actualizar, recibe por parámetro un objeto de la clase cuenta y retorna un mensaje si fue o no modificado los datos, esta función se encargará de la inyección SQL	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_07 Eliminar los datos de una aportación	
Número de la Tarea: TI_15	Nombre de Tarea: Mostrar diálogo de la aportación seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener los datos de la aportación correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de eliminar. Al dar clic en el botón se despliega un diálogo con los datos de la aportación correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_07 Eliminar los datos de una aportación	
Número de la Tarea: TI_16	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función eliminar, recibe por parámetro un entero correspondiente al código y retorna un mensaje si fue o no eliminada la aportación, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_08 Ingresar la solicitud de un préstamo	
Número de la Tarea: TI_17	Nombre de Tarea: Mostrar el diálogo de la nueva solicitud
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para ingresar la información de la nueva solicitud de préstamo.	
Prueba de aceptación: Al dar clic en el botón nueva solicitud se despliega el diálogo con los campos vacíos. Se debe ingresar el monto a solicitar. Al guardar la solicitud esta deberá ser aprobada o negada por la persona encargada.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_9 Ver el estado de un préstamo	
Número de la Tarea: TI_18	Nombre de Tarea: Mostrar diálogo de préstamo seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener el estado del préstamo correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de estado del préstamo. Al dar clic en el botón se despliega un diálogo con los datos del préstamo correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_11 Actualizar los datos de un préstamo	
Número de la Tarea: TI_19	Nombre de Tarea: Mostrar diálogo de préstamo seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función actualizar, recibe por parámetro un objeto de la clase préstamo y retorna un mensaje si fue o no modificado los datos, esta función se encargará de la inyección SQL	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de actualizar. Al dar clic en el botón se despliega un diálogo con los datos de la aportación correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_12 Registro el ingreso de una cuota del préstamo	
Número de la Tarea: TI_20	Nombre de Tarea: Mostrar diálogo de préstamo seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener la cuota del préstamo correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de cuota. Al dar clic en el botón se despliega un diálogo con los datos de la cuota correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_12 Registro el ingreso de una cuota del préstamo	
Número de la Tarea: TI_21	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función insertar, recibe por parámetro un objeto de la clase préstamo y retorna un mensaje si fue o no registrada la cuota, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_13 Generar la tabla de amortización de un préstamo	
Número de la Tarea: TI_22	Nombre de Tarea: Mostrar diálogo de préstamo seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener el estado del préstamo correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de estado del préstamo. Al dar clic en el botón se despliega un diálogo con los datos del préstamo correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_15 Eliminar los datos de un préstamo	
Número de la Tarea: TI_23	Nombre de Tarea: Mostrar diálogo de la aportación seleccionado
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: En cada fila de la tabla crear una columna extra al extremo derecho la cual contiene el botón para mostrar el diálogo, el cual debe contener los datos del préstamo correspondiente.	
Prueba de aceptación: Todas las filas tienen el botón para mostrar el diálogo de eliminar. Al dar clic en el botón se despliega un diálogo con los datos del préstamo correspondiente.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_15 Eliminar los datos de un préstamo	
Número de la Tarea: TI_24	Nombre de Tarea: Crear función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función eliminar, recibe por parámetro un entero correspondiente al código y retorna un mensaje si fue o no eliminado el préstamo, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_16 Generar el reporte de las aportaciones del día	
Número de la Tarea: TI_25	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función listar, recibe por parámetro un entero correspondiente al identificador y retorna una lista de las aportaciones del día.	
Prueba de aceptación: Probar el Script en MySQL.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_16 Generar el reporte de las aportaciones del día	
Número de la Tarea: TI_26	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Permite generar un archivo pdf en el que contendrá las aportaciones que se registraron en un día.	
Prueba de aceptación: Generar un archivo con extensión pdf.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_17 Generar el reporte de las personas que no han aportando	
Número de la Tarea: TI_27	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función listar, recibe por parámetro un entero correspondiente al identificador y retorna una lista de los socios que no han aportado.	
Prueba de aceptación: Probar el Script en MySQL.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_17 Generar el reporte de las personas que no han aportado	
Número de la Tarea: TI_28	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Permite generar un archivo pdf en el que contendrá los datos de los socios que no han aportado.	
Prueba de aceptación: Generar un archivo con extensión pdf.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_18 Generar el reporte de los préstamos entregados	
Número de la Tarea: TI_29	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función listar, recibe por parámetro un entero correspondiente al identificador y retorna una lista de los préstamos entregados.	
Prueba de aceptación: Probar el Script en MySQL.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_18 Generar el reporte de los préstamos entregados	
Número de la Tarea: TI_30	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Permite generar un archivo pdf en el que contendrá los préstamos entregados	
Prueba de aceptación: Generar un archivo con extensión pdf.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_19 Generar el reporte de las personas que han tenido atraso en sus préstamos.	
Número de la Tarea: TI_31	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función listar, recibe por parámetro un entero correspondiente al identificador y retorna una lista de los socios que han tenido atraso en sus préstamos.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_19 Generar el reporte de las personas que han tenido atraso en sus préstamos.	
Número de la Tarea: TI_32	Nombre de Tarea: Generar .pdf
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Permite generar un archivo pdf en el que contendrá los datos de los socios que han tenido atrasos en sus préstamos.	
Prueba de aceptación: Generar un archivo con extensión pdf.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_20 Ingresar la utilidad de la caja	
Número de la Tarea: TI_33	Nombre de Tarea: Mostrar el diálogo de la utilidad
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para ingresar la utilidad de la caja.	
Prueba de aceptación: Al dar clic en el botón utilidad se despliega el diálogo con el campo vacío.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_20 Ingresar la utilidad de la caja	
Número de la Tarea: TI_34	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función insertar, recibe por parámetro un objeto de la clase utilidad y retorna un mensaje si fue o no insertado la utilidad, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_21 Ingresar la utilidad del comisariato	
Número de la Tarea: TI_35	Nombre de Tarea: Mostrar el diálogo de la utilidad del comisariato
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para ingresar la utilidad del comisariato.	
Prueba de aceptación: Al dar clic en el botón utilidad del comisariato se despliega el diálogo con el campo vacío.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_21 Ingresar la utilidad del comisariato	
Número de la Tarea: TI_36	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función insertar, recibe por parámetro un objeto de la clase utilidad y retorna un mensaje si fue o no insertado la utilidad del comisariato, esta función se encargará de la inyección SQL.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_22 Calcular la utilidad de un socio que se retire	
Número de la Tarea: TI_37	Nombre de Tarea: Mostrar el diálogo para calcular la utilidad
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para el cálculo de la utilidad.	
Prueba de aceptación: Al dar clic en el botón se despliega un diálogo con los datos del socio y su utilidad.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_22 Calcular la utilidad de un socio que se retire	
Número de la Tarea: TI_38	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función calcular, recibe por parámetro un entero correspondiente al identificador y retorna el cálculo de la utilidad.	
Prueba de aceptación: Probar el Script en MySQL. Probar la función con datos reales correctos. Probar la función con datos incorrectos.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_23 Generar las utilidades de los socios	
Número de la Tarea: TI_39	Nombre de Tarea: Crear la función
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: Dentro del paquete persistencia. impl se crea la función listar, recibe por parámetro un entero correspondiente al identificador y retorna una lista de los socios que han tenido atraso en sus préstamos.	
Prueba de aceptación: Probar el Script en MySQL.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_24 Consultar el saldo disponible	
Número de la Tarea: TI_40	Nombre de Tarea: Mostrar el diálogo para consultar el saldo
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para la consulta del saldo.	
Prueba de aceptación: Al dar clic en el botón se despliega un diálogo con los datos del socio y su saldo disponible.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_25 Fecha de pago del préstamo	
Número de la Tarea: TI_41	Nombre de Tarea: Mostrar el diálogo para conocer la fecha de pago de préstamo
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para conocer la fecha de pago de préstamo.	
Prueba de aceptación: Al dar clic en el botón se despliega un diálogo con los datos del socio con los datos del préstamo.	

TAREA DE INGENIERÍA	
Historia de Usuario: HU_26 Cierre de caja mensual	
Número de la Tarea: TI_42	Nombre de Tarea: Mostrar el diálogo para realizar el cierre de caja
Tipo de Tarea: Desarrollo	Tiempo Estimado: 2 días
Programador Responsable: Deysi Latorre	
Descripción: El diálogo se comunica con el controller correspondiente para realizar el cierre de caja mensual.	
Prueba de aceptación: Al dar clic en el botón se despliega una ventana con las transacciones realizadas en el mes.	

Anexo C: Pruebas de Aceptación

Prueba de Aceptación	
Código: PA_01	Historia de Usuario: Diseño de Base de Datos
Nombre: Que el diccionario de datos tenga las columnas descritas.	
Responsable: Deysi Latorre	
Descripción: El diccionario de datos debe tener documentadas todas las tablas de la base de datos, con las columnas especificadas.	
Condiciones de Ejecución: El responsable revisará visualmente que cada una de las tablas estén documentadas en el diccionario de datos.	
Pasos de ejecución: Revisar si están el número de tablas descritas. Luego revisar si todas las tablas cuentan con las columnas descritas.	

Prueba de Aceptación	
Código: PA_02	Historia de Usuario: Encriptación de la clave
Nombre: Que la clave este encriptada	
Responsable: Deysi Latorre	
Descripción: En la base de datos la clave se encontrará encriptada	
Condiciones de Ejecución: El responsable revisará visualmente que cada al ingresar la clave esta esté oculta, y en la base de datos estará encriptada	
Pasos de ejecución: Ir a la tabla usuario. Verificar que la clave esta encriptada.	

Prueba de Aceptación	
Código: PA_03	Historia de Usuario: Diseño de la interfaz de usuario
Nombre: Que la interfaz cumpla con la identidad de la empresa	
Responsable: Deysi Latorre	
Descripción: La interfaz de usuario tendrá los colores que identifica a la empresa	
Condiciones de Ejecución: El responsable revisará visualmente que cumpla con lo especificado en el diseño	
Pasos de ejecución: Ejecutar el sistema Observar que cumpla con la identidad de la empresa, que tenga los colores que la identifica	

Prueba de Aceptación	
Código: PA_04	Historia de Usuario: Listado de los socios
Nombre: La tabla se cargará al abrir la pestaña de socio	
Responsable: Deysi Latorre	
Descripción: Al escoger la pestaña socio esta cargará automáticamente la lista de los socios existentes en la caja	
Condiciones de Ejecución: El responsable revisará visualmente que los datos se carguen	
Pasos de ejecución: Se escoge la pestaña socio Al dar clic se mostrará el listado de los socios existentes en la caja.	

Prueba de Aceptación	
Código: PA_05	Historia de Usuario: Listado de los socios
Nombre: Total de registros que existen en la base	
Responsable: Deysi Latorre	
Descripción: Al cargar los datos de los socios en la parte final mostrara el total de socios registrados	
Condiciones de Ejecución: El responsable revisará visualmente que el número de registros sea igual a los de la base de datos	
Pasos de ejecución: Se escoge la pestaña socio Al dar clic se mostrará el listado de los socios. En la parte inferior mostrará el total de socios.	

Prueba de Aceptación	
Código: PA_06	Historia de Usuario: Registro de ingreso de un socio
Nombre: Despliegue del dialogo para el ingreso de un nuevo socio	
Responsable: Deysi Latorre	
Descripción: Al escoger nuevo socio mostrará un dialogo con los datos a ingresar	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren todos los campos deseados para el registro de un socio	
Pasos de ejecución: Dar clic sobre nuevo socio. Aquí se podrá observar si se encuentran todos los campos para llenar.	

Prueba de Aceptación	
Código: PA_07	Historia de Usuario: Registro de ingreso de un socio
Nombre: Probar script en MySQL	
Responsable: Deysi Latorre	
Descripción: Verificar si el script está bien realizado	
Condiciones de Ejecución: El responsable revisará visualmente que el script funcione correctamente	
Pasos de ejecución: En MySQL hacer correr el script. Verificar que muestre los datos deseados.	

Prueba de Aceptación	
Código: PA_08	Historia de Usuario: Registro de ingreso de un socio
Nombre: Probar la función con datos correctos	
Responsable: Deysi Latorre	
Descripción: Verificar si la función ingresa los datos correctos	
Condiciones de Ejecución: El responsable revisará visualmente que la función ingrese los datos correctamente	
Pasos de ejecución: Ingresar un socio con datos correctos mediante código. Verificar se ingresa.	

Prueba de Aceptación	
Código: PA_09	Historia de Usuario: Registro de ingreso de un socio
Nombre: Probar la función con datos incorrectos	
Responsable: Deysi Latorre	
Descripción: Verificar si la función ingresa los datos incorrectos	
Condiciones de Ejecución: El responsable revisará visualmente que la función no ingrese los datos si están incorrectos	
Pasos de ejecución: Ingresar un socio con datos incorrectos mediante código. Verificar se ingresa o no.	

Prueba de Aceptación	
Código: PA_10	Historia de Usuario: Actualizar los datos del socio
Nombre: Las filas tengan un botón para actualizar los datos	
Responsable: Deysi Latorre	
Descripción: Cada una de las filas con los datos de los socios debe existir un botón para actualizar los mismo	
Condiciones de Ejecución: El responsable revisará visualmente que este botón exista en todas las filas	
Pasos de ejecución: Dar clic en socio. Verificar si en la fila de cada socio existe un botón que diga actualizar.	

Prueba de Aceptación	
Código: PA_11	Historia de Usuario: Actualizar los datos del socio
Nombre: Despliegue del botón actualizar	
Responsable: Deysi Latorre	
Descripción: Al dar clic en el botón se desplegará un diálogo	
Condiciones de Ejecución: El responsable revisará visualmente que al dar clic en el botón este cargue los datos del socio	
Pasos de ejecución: Dar clic en el botón actualizar. Verificar si carga todos los datos del socio seleccionado.	

Prueba de Aceptación	
Código: PA_12	Historia de Usuario: Actualizar los datos del socio
Nombre: Probar script en MySQL	
Responsable: Deysi Latorre	
Descripción: Verificar si el script está bien realizado	
Condiciones de Ejecución: El responsable revisará visualmente que el script funcione correctamente	
Pasos de ejecución: En MySQL hacer correr el script. Verificar que muestre los datos deseados.	

Prueba de Aceptación	
Código: PA_13	Historia de Usuario: Actualizar los datos del socio
Nombre: Probar la función con datos correctos	
Responsable: Deysi Latorre	
Descripción: Verificar si la función actualiza los datos correctamente	
Condiciones de Ejecución: El responsable revisará visualmente que la función actualiza los datos correctamente	
Pasos de ejecución: Ingresar un socio con datos correctos mediante código. Verificar se ingresa.	

Prueba de Aceptación	
Código: PA_14	Historia de Usuario: Actualizar los datos del socio
Nombre: Probar la función con datos incorrectos	
Responsable: Deysi Latorre	
Descripción: Verificar si la función actualiza los datos incorrectos	
Condiciones de Ejecución: El responsable revisará visualmente que la función no actualice los datos si están incorrectos	
Pasos de ejecución: Ingresar un socio con datos incorrectos mediante código. Verificar se ingresa o no.	

Prueba de Aceptación	
Código: PA_15	Historia de Usuario: Eliminar los datos de un socio
Nombre: Las filas tengan un botón para eliminar los datos del socio	
Responsable: Deysi Latorre	
Descripción: Cada una de las filas con los datos de los socios debe existir un botón para eliminar los mismo	
Condiciones de Ejecución: El responsable revisará visualmente que este botón exista en todas las filas	
Pasos de ejecución: Dar clic en socio. Verificar si en la fila de cada socio existe un botón que diga eliminar.	

Prueba de Aceptación	
Código: PA_16	Historia de Usuario: Eliminar los datos de un socio
Nombre: Despliegue del botón eliminar	
Responsable: Deysi Latorre	
Descripción: Al dar clic en el botón se desplegará un diálogo	
Condiciones de Ejecución: El responsable revisará visualmente que al dar clic en el botón este cargue los datos del socio a eliminar	
Pasos de ejecución: Dar clic en el botón eliminar. Verificar si carga todos los datos del socio seleccionado para su posterior eliminación.	

Prueba de Aceptación	
Código: PA_17	Historia de Usuario: Registro de ingreso de una aportación
Nombre: Despliegue del dialogo para el ingreso de una nueva aportación	
Responsable: Deysi Latorre	
Descripción: Al escoger nueva aportación mostrará un dialogo con los datos a ingresar	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren todos los campos deseados para el registro de la aportación	
Pasos de ejecución: Dar clic sobre nueva aportación. Aquí se podrá observar si se encuentran todos los campos para llenar.	

Prueba de Aceptación	
Código: PA_18	Historia de Usuario: Registro de ingreso de una aportación
Nombre: Buscar socio por apellido para registrar la aportación	
Responsable: Deysi Latorre	
Descripción: Al escribir el apellido del socio a buscar, este cargará los datos	
Condiciones de Ejecución: El responsable revisará visualmente que al buscar el socio este cargue los datos correctos	
Pasos de ejecución: Ingresar el apellido del socio. Verificar si los datos del socio son los correctos para realizar la aportación	

Prueba de Aceptación	
Código: PA_19	Historia de Usuario: Registro de ingreso de una aportación
Nombre: Descarga del recibo	
Responsable: Deysi Latorre	
Descripción: Al guardar la aportación se descargará el recibo	
Condiciones de Ejecución: El responsable revisará visualmente que se descargue el recibo luego de guardar la aportación	
Pasos de ejecución: Dar clic sobre guardar. Aquí se podrá observar si se descarga el recibo automáticamente.	

Prueba de Aceptación	
Código: PA_20	Historia de Usuario: Actualizar los datos de una aportación
Nombre: Las filas tengan un botón para actualizar los datos	
Responsable: Deysi Latorre	
Descripción: Cada una de las filas con los datos de la aportación debe existir un botón para actualizar los mismo	
Condiciones de Ejecución: El responsable revisará visualmente que este botón exista en todas las filas	
Pasos de ejecución: Dar clic en aportación. Verificar si en la fila de cada aportación existe un botón que diga actualizar.	

Prueba de Aceptación	
Código: PA_21	Historia de Usuario: Actualizar los datos de una aportación
Nombre: Despliegue del botón actualizar	
Responsable: Deysi Latorre	
Descripción: Al dar clic en el botón se desplegará un diálogo	
Condiciones de Ejecución: El responsable revisará visualmente que al dar clic en el botón este cargue los datos de la aportación	
Pasos de ejecución: Dar clic en el botón actualizar. Verificar si carga todos los datos de la aportación seleccionada.	

Prueba de Aceptación	
Código: PA_22	Historia de Usuario: Eliminar los datos de una aportación
Nombre: Que las filas tengan un botón para eliminar los datos de la aportación	
Responsable: Deysi Latorre	
Descripción: Cada una de las filas con los datos de la aportación debe existir un botón para eliminar los mismo	
Condiciones de Ejecución: El responsable revisará visualmente que este botón exista en todas las filas	
Pasos de ejecución: Dar clic en aportación. Verificar si en la fila de cada aportación existe un botón que diga eliminar.	

Prueba de Aceptación	
Código: PA_23	Historia de Usuario: Eliminar los datos de una aportación
Nombre: Despliegue del botón eliminar	
Responsable: Deysi Latorre	
Descripción: Al dar clic en el botón se desplegará un diálogo con los datos de la aportación	
Condiciones de Ejecución: El responsable revisará visualmente que al dar clic en el botón este cargue los datos de la aportación a eliminar	
Pasos de ejecución: Dar clic en el botón eliminar. Verificar si carga todos los datos de la aportación seleccionada para su posterior eliminación.	

Prueba de Aceptación	
Código: PA_24	Historia de Usuario: Ingresar la solicitud de un préstamo
Nombre: Despliegue del diálogo para el ingreso de la nueva solicitud	
Responsable: Deysi Latorre	
Descripción: Al escoger nueva solicitud mostrará un dialogo con los datos a ingresar	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren todos los campos deseados para el registro de una solicitud	
Pasos de ejecución: Dar clic sobre nueva solicitud. Aquí se podrá observar si se encuentran todos los campos para llenar.	

Prueba de Aceptación	
Código: PA_25	Historia de Usuario: Ingresar la solicitud de un préstamo
Nombre: Llenar datos de solicitud	
Responsable: Deysi Latorre	
Descripción: Al escoger nueva solicitud se debe ingresar los datos de monto y garante.	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren el monto, garante y que la solicitud tenga el formato definido.	
Pasos de ejecución: Dar clic sobre nueva solicitud. Mostrará los campos a llenar como son monto y garante Al dar clic en guardar mostrara un mensaje si se guarda correctamente.	

Prueba de Aceptación	
Código: PA_26	Historia de Usuario: Registro el ingreso de una cuota del préstamo
Nombre: Las filas tendrán un botón para cuota del préstamo	
Responsable: Deysi Latorre	
Descripción: Cada una de las filas con los datos de los prestamos debe existir un botón para pago de la cuota	
Condiciones de Ejecución: El responsable revisará visualmente que este botón exista en todas las filas	
Pasos de ejecución: Dar clic cuota. Verificar si en la fila de cada préstamo existe un botón que diga cuota.	

Prueba de Aceptación	
Código: PA_27	Historia de Usuario: Generar la tabla de amortización de un préstamo
Nombre: Las filas tendrán un botón para generar la amortización	
Responsable: Deysi Latorre	
Descripción: La fila de los préstamos tendrá el botón amortización	
Condiciones de Ejecución: El responsable revisará visualmente que las filas de los préstamos tengan este botón	
Pasos de ejecución: Dar clic en préstamo Revisar que cada fila de los préstamos tenga este botón.	

Prueba de Aceptación	
Código: PA_28	Historia de Usuario: Generar la tabla de amortización de un préstamo
Nombre: Despliegue del diálogo	
Responsable: Deysi Latorre	
Descripción: La seleccionar el botón debe mostrar los datos para el cálculo de la amortización	
Condiciones de Ejecución: El responsable revisará visualmente que estén los campos necesarios para el cálculo de la amortización como son monto, interés, meses y tipo	
Pasos de ejecución: Dar clic en amortización Revisar si tiene los campos antes mencionados	

Prueba de Aceptación	
Código: PA_29	Historia de Usuario: Generar el reporte de las aportaciones del día
Nombre: Probar script en MySQL	
Responsable: Deysi Latorre	
Descripción: Verificar si el script está bien realizado	
Condiciones de Ejecución: El responsable revisará visualmente que el script funcione correctamente	
Pasos de ejecución: En MySQL hacer correr el script. Verificar que muestre los datos deseados.	

Prueba de Aceptación	
Código: PA_30	Historia de Usuario: Generar el reporte de las aportaciones del día
Nombre: Generar un archivo con extensión pdf.	
Responsable: Deysi Latorre	
Descripción: Verificar si se genera el documento con .pdf	
Condiciones de Ejecución: El responsable revisará visualmente que genere el archivo .pdf	
Pasos de ejecución: Al dar clic en imprimir. Se descargará automáticamente el archivo con .pdf.	

Prueba de Aceptación	
Código: PA_31	Historia de Usuario: Generar el reporte de las personas que no han aportando
Nombre: Probar script en MySQL	
Responsable: Deysi Latorre	
Descripción: Verificar si el script está bien realizado	
Condiciones de Ejecución: El responsable revisará visualmente que el script funcione correctamente	
Pasos de ejecución: En MySQL hacer correr el script. Verificar que muestre los datos deseados.	

Prueba de Aceptación	
Código: PA_32	Historia de Usuario: Generar el reporte de las personas que no han aportando
Nombre: Generar un archivo con extensión pdf.	
Responsable: Deysi Latorre	
Descripción: Verificar si se genera el documento con .pdf	
Condiciones de Ejecución: El responsable revisará visualmente que genere el archivo .pdf	
Pasos de ejecución: Al dar clic en imprimir. Se descargará automáticamente el archivo con .pdf.	

Prueba de Aceptación	
Código: PA_33	Historia de Usuario: Generar el reporte de los préstamos otorgados
Nombre: Probar script en MySQL	
Responsable: Deysi Latorre	
Descripción: Verificar si el script está bien realizado	
Condiciones de Ejecución: El responsable revisará visualmente que el script funcione correctamente	
Pasos de ejecución: En MySQL hacer correr el script. Verificar que muestre los datos deseados.	

Prueba de Aceptación	
Código: PA_34	Historia de Usuario: Generar el reporte de los préstamos otorgados
Nombre: Generar un archivo con extensión pdf.	
Responsable: Deysi Latorre	
Descripción: Verificar si se genera el documento con .pdf	
Condiciones de Ejecución: El responsable revisará visualmente que genere el archivo .pdf	
Pasos de ejecución: Al dar clic en imprimir. Se descargará automáticamente el archivo con .pdf.	

Prueba de Aceptación	
Código: PA_35	Historia de Usuario: Generar el reporte de las personas que han tenido atraso en sus préstamos.
Nombre: Probar script en MySQL	
Responsable: Deysi Latorre	
Descripción: Verificar si el script está bien realizado	
Condiciones de Ejecución: El responsable revisará visualmente que el script funcione correctamente	
Pasos de ejecución: En MySQL hacer correr el script. Verificar que muestre los datos deseados.	

Prueba de Aceptación	
Código: PA_36	Historia de Usuario: Generar el reporte de las personas que han tenido atraso en sus préstamos.
Nombre: Generar un archivo con extensión pdf.	
Responsable: Deysi Latorre	
Descripción: Verificar si se genera el documento con .pdf	
Condiciones de Ejecución: El responsable revisará visualmente que genere el archivo .pdf	
Pasos de ejecución: Al dar clic en imprimir. Se descargará automáticamente el archivo con .pdf.	

Prueba de Aceptación	
Código: PA_37	Historia de Usuario: Ingresar la utilidad de la caja
Nombre: Despliegue del diálogo para el ingreso de la utilidad	
Responsable: Deysi Latorre	
Descripción: Al escoger utilidad mostrará un diálogo con los datos a ingresar	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren los campos deseados para el registro de la utilidad	
Pasos de ejecución: Dar clic sobre utilidad. Aquí se podrá observar si se encuentran todos los campos para llenar.	

Prueba de Aceptación	
Código: PA_38	Historia de Usuario: Ingresar la utilidad de la caja
Nombre: Probar script en MySQL	
Responsable: Deysi Latorre	
Descripción: Verificar si el script está bien realizado	
Condiciones de Ejecución: El responsable revisará visualmente que el script funcione correctamente	
Pasos de ejecución: En MySQL hacer correr el script. Verificar funciona correctamente.	

Prueba de Aceptación	
Código: PA_39	Historia de Usuario: Ingresar la utilidad de la caja
Nombre: Probar la función con datos correctos	
Responsable: Deysi Latorre	
Descripción: Verificar si la función ingresa los datos correctos	
Condiciones de Ejecución: El responsable revisará visualmente que la función ingrese los datos correctamente	
Pasos de ejecución: Ingresar la utilidad con datos correctos mediante código. Verificar si ingresa.	

Prueba de Aceptación	
Código: PA_40	Historia de Usuario: Ingresar la utilidad de la caja
Nombre: Probar la función con datos incorrectos	
Responsable: Deysi Latorre	
Descripción: Verificar si la función ingresa los datos incorrectos	
Condiciones de Ejecución: El responsable revisará visualmente que la función no ingrese los datos si están incorrectos	
Pasos de ejecución: Ingresar la utilidad con datos incorrectos mediante código. Verificar si ingresa o no.	

Prueba de Aceptación	
Código: PA_41	Historia de Usuario: Ingresar la utilidad del comisariato
Nombre: Despliegue del diálogo para el ingreso de la utilidad del comisariato	
Responsable: Deysi Latorre	
Descripción: Al escoger utilidad del comisariato mostrará un diálogo con los datos a ingresar	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren los campos deseados para el registro de la utilidad del comisariato	
Pasos de ejecución: Dar clic sobre utilidad del comisariato. Aquí se podrá observar si se encuentran todos los campos para llenar.	

Prueba de Aceptación	
Código: PA_42	Historia de Usuario: Calcular la utilidad de un socio que se retire
Nombre: Despliegue del diálogo para el cálculo de la utilidad	
Responsable: Deysi Latorre	
Descripción: Al escoger utilidad mostrará un diálogo que se deberá escoger las fechas para el cálculo	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren los campos deseados para el cálculo de la utilidad	
Pasos de ejecución: Dar clic sobre utilidad. Aquí se podrá observar si se encuentran todos los campos a escoger. Al dar clic en calcular se mostrará la utilidad del socio.	

Prueba de Aceptación	
Código: PA_43	Historia de Usuario: Fecha de pago del préstamo
Nombre: Despliegue del diálogo para ver la fecha de pago	
Responsable: Deysi Latorre	
Descripción: Al escoger préstamo me mostrará la fecha de pago de la cuota	
Condiciones de Ejecución: El responsable revisará visualmente que se encuentren los datos del préstamo con la fecha a cancelar	
Pasos de ejecución: Dar clic sobre préstamo. Mostrará la tabla de amortización con la fecha de pago.	

Prueba de Aceptación	
Código: PA_44	Historia de Usuario: Cierre de caja mensual
Nombre: Despliegue del diálogo para el cierre de caja	
Responsable: Deysi Latorre	
Descripción: Al escoger cierre de caja este me mostrará el total en cada uno de los procesos	
Condiciones de Ejecución: El responsable revisará visualmente que el cálculo del cierre de caja sea correcto	
Pasos de ejecución: Dar clic sobre cierre de caja. Aquí se podrá observar los valores totales de las transacciones realizadas en el mes.	

Anexo D: Diccionario de Datos

TABLA SOCIO					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de los socios					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
Nombre	150	varchar	NO		Nombre del socio
Apellido	150	varchar	NO		Apellido del socio
direccion	255	varchar	SI		Dirección actual donde reside el socio
Teléfono	10	varchar	SI		Número telefónico del socio
estadocivil	11	int	SI		Estado civil del socio
beneficiario	255	varchar	SI		Nombre de la persona que será beneficiaria
fechaingreso		date	SI		Fecha de ingreso a la caja
fecharegistro		date	SI		Fecha de ingreso
fechasalida		date	SI		Fecha de salida de la caja
fecharegsalida		date	SI		Fecha de ingreso
numcuenta	12	varchar	SI		Número de cuenta
Estado	11	int	NO	1	Estado de la cuenta (activa/ inactiva)
idusuario	20	bigint	SI		Identificador del socio
Idsocio	20	bigint	NO		Número de socio

TABLA AMORTIZACIÓN					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de la amortización					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idcuota	20	bigint	NO		Identificador de la cuota
idprestamo	20	bigint	SI	NULL	Identificador del préstamo
numerocuota	11	int	SI	NULL	Número de la cuota
capital		double	SI	NULL	Monto de dinero
interés		double	SI	NULL	Interés del préstamo
amortizacion		double	SI	NULL	Cuota a cancelar
cuotatotal		double	SI	NULL	Cuota total a cancelar
interesmora		double	SI	NULL	Interés por mora
cuotamora		double	SI	NULL	Cuota con mora
estadocuota	11	int	SI	NULL	Estado de la cuota
saldo		double	SI	NULL	Saldo que queda
fechapago		date	SI	NULL	Fecha de pago de la cuota
fecharealpago		date	SI	NULL	Fecha real de pago de la cuota

TABLA CUENTA					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de la cuenta					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idtransaccion	20	bigint	NO		Identificador de la transacción
montodeposito		double	SI	NULL	Valor que deposita
montoretiro		double	SI	NULL	Valor que retira
saldocuenta		double	NO		Saldo total existente

					en la cuenta
fechatransaccion		datetime	NO	0000-00-00 00:00:00	Fecha que se realiza la transacción
idsocio	20	bigint	SI	NULL	Identificador del socio
tipotransaccion	20	bigint	SI	NULL	Tipo de transacción que se realizo
estado	11	int	SI	NULL	Estado de a transacción
cheque	50	varchar	SI	NULL	Número de cheque en caso se ser depositado

TABLA INTERÉS					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos del interés					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idinteres	20	bigint	NO		Identificador del interés
porcentaje		double	SI	NULL	Porcentaje de interés
nombre	255	varchar	SI	NULL	Nombre del interés
tipointeres	11	int	SI	NULL	Tipo de interés

TABLA PARÁMETROS					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de los parámetros					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
parametro	255	varchar	SI	NULL	Parámetro
valor	255	varchar	SI	NULL	Valor

TABLA PRÉSTAMO					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos del préstamo					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idprestamo	20	bigint	NO		Identificador del préstamo
montoaprobado	255	double	SI	NULL	Monto que se aprueba
numerocuotas	11	int	SI	NULL	Número de cuotas
interes	20	bigint	SI	NULL	Interés del préstamo
tipoprestamo	255	int	SI	NULL	Tipo de préstamo
fechaaprobacion		date	SI	NULL	Fecha que se aprueba el préstamo
fechasolicitud		date	SI	NULL	Fecha que se envía la solicitud
idsolicitud	20	bigint	SI	NULL	Identificador de la solicitud
estado	11	int	SI	1	Estado del préstamo

TABLA ROL					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos del rol					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
descripcion	50	varchar	SI	NULL	Descripción del rol
idrol	20	bigint	NO		Identificador del rol

TABLA SOLICITUD					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de la solicitud					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idsolicitud	20	bigint	NO		Identificador de la solicitud
fechasolicitud		date	NO		Fecha en la cual ingreso la solicitud
montosolicitado		double	NO		Monto solicitado
montoaprobado	255, 0	double	SI	NULL	Monto aprobado
tipoprestamo	11	int	NO		Tipo de préstamo
idsocio	12	bigint	SI	NULL	Identificador del socio
estado	11	int	SI	1	Estado de la solicitud

TABLA TRANSACCIÓN					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de la transacción					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
id	20	bigint	NO		Identificador de la transacción
nombre	8	varchar	SI	NULL	Nombre de la transacción
descripcion	255	varchar	SI	NULL	Descripción de la transacción

TABLA USUARIO					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
cedula	10	varchar	SI	NULL	Cédula del usuario
contrasenia	150	varchar	SI	NULL	Contraseña
idusuario	20	bigint	NO		Identificador del usuario

TABLA USUARIOROL					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos del usuario y el rol					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idusuario	20	bigint	NO		Identificador del usuario
idrol	20	bigint	NO		Identificador del rol

TABLA UTILIDAD					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de la utilidad					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idutilidad	20	bigint	NO		Identificador de la utilidad
utilidadcaja		double	SI	NULL	Utilidad de la caja
utilidadcomisariato		double	SI	NULL	Utilidad del comisariato
aniogenerado	11	int	SI	NULL	Año de generación

fecharegistro		date	SI	NULL	Fecha de registro
---------------	--	------	----	------	-------------------

TABLA UTILIDAD SOCIO					
NOMBRE: BDCajaTub					
DESCRIPCIÓN: Tabla con los campos que contendrán los datos de la utilidad del socio					
FECHA DE CREACIÓN:					
CAMPO	TAMAÑO	TIPO DE DATO	NULIDAD	DEFAULT	DESCRIPCIÓN
idutilidad	20	bigint	NO		Identificador de la utilidad
diasaportados	11	int	SI	NULL	Número de días aportados
aniogenerado	11	int	SI	NULL	Año que genera
fecharegistro		date	SI	NULL	Fecha de registro
monto		double	SI	NULL	Monto que tiene el socio
idsocio	20	bigint	SI	NULL	Identificador del socio