

Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Ingeniería de Software y Sistemas Informáticos

|  |  |
| --- | --- |
| Trabajo fin de estudio presentado por: | Christian Giovany Ipiales Moreta |
| Tipo de trabajo: | Desarrollo práctico |
| Director/a: | FELIX OSCAR FERNANDEZ PEÑA |
| Fecha: | 2025/04/16 |

Resumen

El objetivo de este proyecto es contar con un sistema informático que permita implementar todos los servicios de cada una de las cajas de ahorros existentes en la ciudad de Ambato tales como es el ahorro, el ahorro en los fondos de reserva, los créditos, siendo los más beneficiados los socios que conforman cada una de las cajas de ahorros de la ciudad.

El aprendizaje de nuevas tecnologías y conforme se van desarrollando nuevas herramientas para la implantación y uso del conocimiento que se adquiere en la maestría que se está cursando hacen que tengamos la capacidad para dar soluciones reales ayudando a si a un conglomerado de personas a crecer económicamente y socialmente, nosotros como estudiantes ponemos en práctica lo aprendido y obtenemos mucha más experiencia para el desenvolvimiento en el campo profesional.

El sistema de caja de ahorro es una aplicación que permitirá agilizar los procesos que se realizan en cada una de las cajas de ahorros existentes permitiéndoles a los socios realizar el trabajo más rápido y eficiente.

El proyecto que se plantea conlleva a utilizar los últimos avances tecnológicos y metodologías ya que de esta forma como estudiantes investigarán y aplicarán los conocimientos para beneficios de las personas que van a ser uso del sistema, y para beneficio propio del estudiante ya que con el desarrollo del proyecto obtendrán mucha más experiencia en la utilización de las herramientas que se investiga para realizar el proyecto con lo cual servirá para la carrera profesional.

El sistema realizado hace uso de herramientas que se obtuvo el conocimiento en el transcurso de la maestría.

Abstract

The communities savings banks are arise with the only finish of helping to the persons of poor resources by the fact that the traditional banking it considered to be persons without capacity of payment, what they can’t accede to quick and opportune credits.

In Present times the process a credit is elementary to solve an idea of beginning that is gets in mind.

Most of people who shape the savings banks in Ambato dedicate their time to the production of crafts and to the production of numerous agricultural products for this reason is indispensable take a credit to buy agricultural and principal materials.

The investigation of new technologies and use of the knowledge that is learned in the classrooms of the Master University promote the aptitude to give practical and real solutions to problematic of the daily living, It’s helping so to a people's conglomerate to growing economically and socially, we as students make practice what we learn and we get much more experience for the development in the professional field.

The savings bank system community is an application which will allow speeding the processes that are realized at each of the existing savings banks, in order to Associates to make fast and efficient work.

Índice de contenidos

[1. Introducción 1](#_Toc201743609)

[1.1. Justificación 2](#_Toc201743610)

[1.2. Planteamiento del problema 3](#_Toc201743611)

[1.3. Estructura del trabajo 4](#_Toc201743612)

[2. Contexto y estado del arte 5](#_Toc201743613)

[3. Objetivos concretos y metodología de trabajo 10](#_Toc201743614)

[3.1. Objetivo general 10](#_Toc201743615)

[3.2. Objetivos específicos 10](#_Toc201743616)

[3.3. Metodología del trabajo 10](#_Toc201743617)

[4 Desarrollo específico de la contribución 15](#_Toc201743618)

[4.1 Descripción de la aplicación Web 15](#_Toc201743619)

[4.1.1 Identificación de requisitos 15](#_Toc201743620)

[4.1.2 Descripción del sistema software desarrollado 16](#_Toc201743621)

[4.1.3 Diseño de Base de Datos 20](#_Toc201743622)

[4.2 Planificación de los incrementos 21](#_Toc201743623)

[4.2.1 Identificación de historias de usuario 21](#_Toc201743624)

[4.2.2 Product Backlog 25](#_Toc201743625)

[4.2.3 Arquitectura de software de la solución propuesta 26](#_Toc201743626)

[4.3 Diseño de la aplicación Web 29](#_Toc201743627)

[4.3.1 Sprints desarrollados **¡Error! Marcador no definido.**](#_Toc201743628)

[4.4 Desarrollo de metodología 30](#_Toc201743629)

[4.4.1 Aspectos Generales de Diseño de Interfaz y usabilidad **¡Error! Marcador no definido.**](#_Toc201743630)

[4.4.2 Aspectos de Seguridad **¡Error! Marcador no definido.**](#_Toc201743631)

[3.- Seguridad General **¡Error! Marcador no definido.**](#_Toc201743632)

[5 Conclusiones y trabajo futuro 37](#_Toc201743633)

[5.1 Conclusiones 37](#_Toc201743634)

[5.2 Trabajo futuro 37](#_Toc201743635)

[Referencias bibliográficas 37](#_Toc201743636)

[Anexo A. Artículo 38](#_Toc201743637)

Índice de figuras

[Figura 1. Módulos Caja de Ahorro. 3](#_Toc176959588)

[Figura 2.Marco de trabajo Scrum 11](#_Toc176959589)

[Figura 3. Modelo de Base de datos 20](file:///C:\Users\cgipiale\Desktop\TFM\Trabajo%20de%20Fin%20de%20Master_christian%20ipiales26082024.docx#_Toc176959590)

[Figura 4.Arquitectura de la aplicación 26](#_Toc176959591)

Índice de tablas

[Tabla 1. Tipos de Créditos. 2](#_Toc201757080)

[Tabla 2.Historia de Usuario Registrar usuarios y accesos 21](#_Toc201757081)

[Tabla 3. Historia de Usuario Modificar usuarios y accesos 22](#_Toc201757082)

[Tabla 4 . Historia de Usuario Registrar cuentas de ahorros 22](#_Toc201757083)

[Tabla 5. Historia de Usuario Modificar cuentas de ahorros 23](#_Toc201757084)

[Tabla 6.Historia de Usuario Administración de ciclos de caja de ahorro 23](#_Toc201757085)

[Tabla 7.Historia de Usuario Administración de cajas de ahorro 24](#_Toc201757086)

[Tabla 8.Historia de Usuario Registrar Transacción Caja de Ahorro 24](#_Toc201757087)

[Tabla 9.Historia de Usuario Generar registros de auditoría 25](#_Toc201757088)

[Tabla 10.Product Backlog 25](#_Toc201757089)

[Tabla 11.Requerimientos de seguridad relacionados con la integridad de la información 29](#_Toc201757090)

[Tabla 12.Requerimientos de seguridad relacionados con la confidencialidad 30](#_Toc201757091)

[Tabla 14.Requerimientos de seguridad relacionados con la disponibilidad 30](#_Toc201757092)

[Tabla 15.Resumen de ejecución del Sprint 0 31](#_Toc201757093)

[Tabla 16.Resumen de ejecución del Sprint 1 32](#_Toc201757094)

[Tabla 17.Resumen de ejecución del Sprint 2 33](#_Toc201757095)

[Tabla 18.Resumen de ejecución del Sprint 3 34](#_Toc201757096)

[Tabla 19.Resumen de ejecución del Sprint 4 35](#_Toc201757097)

[Tabla 20.Resumen de ejecución del Sprint 5 35](#_Toc201757098)

# Introducción

El presente proyecto de Fin de Máster tiene la finalidad de solucionar un problema en la ciudad de Ambato el mismo que ayudará a la ciudad a automatizar el trabajo que se realiza en cada una de las cajas de ahorro existentes en la ciudad, permitiendo realizar los procesos de forma automática.

Para dar solución a esta problemática, conjuntamente con los socios de las cajas de ahorros se ve la necesidad de realizar una aplicación web la misma que beneficiará a las cajas de ahorros existentes en la ciudad.

Las cajas de ahorro nacen con la única idea de ser una ayuda para la comunidad, debido a que las personas de los sectores más pobres tienen un acceso restringido para la obtención de un crédito en la banca tradicional ya que se les considera sin capacidad de pago, hoy en día obtener un crédito en una institución financiera se ha convertido en algo muy necesario para realizar un emprendimiento de algún proyecto personal o en sociedad ,es imprescindible contar con oportuno y rápido acceso a un crédito que pueda viabilizar un proyecto de emprendimiento ayudando a las personas crecer económicamente.

Para las personas que necesitan ahorrar y obtener créditos de forma rápida y oportuna, esta modalidad de ahorro y crédito se ha convertido en algo muy necesario en el diario vivir.

Las cajas de ahorro son asociaciones creadas con un determinado número de socios los mismos que obligatoriamente deben asistir cada semana del año a reunirse en un lugar determinado para depositar el ahorro correspondiente a la semana según el valor que se fija semanalmente, esto es de acuerdo al número de acciones que se adquiere a principio de cada ciclo.

El proyecto se denomina Sistema de Caja de Ahorro el cual permitirá realizar procesos automáticos sobre las cajas de ahorro con lo cual los socios ya no realizaran los procesos de forma manual, para realizar el proyecto se utilizará herramientas de software de vanguardia con la finalidad de implementar todos los conocimientos adquiridos en el Máster.

## Justificación

De acuerdo a los conocimientos adquiridos en el transcurso del Máster , se vio la oportunidad de ayudar a la Ciudad de Ambato, en especial a las personas que conforman las cajas de ahorro que son personas que van a ayudar al crecimiento social y económico de la Ciudad, ya que son personas que tienen gran incentivo para el ahorro y de gran entusiasmo para realizar cualquier tipo de emprendimiento, debido a que la ciudad es rica en métodos productivos ya que los habitantes de la ciudad son principalmente pequeños agricultores, y artesanos que se dedican a la producción de varios productos agrícolas como fréjol, caña, tomate, pepino o pimiento. Otra parte de la población se dedica en menor escala al comercio y las artesanías. Todos y todas requieren de créditos ya sea para comprar insumos agrícolas o materias primas.

Con lo cual este proyecto es de gran ayuda para el crecimiento económico y social para la ciudad en general.

Tabla 1. Tipos de Créditos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PRODUCTOS | MONTO | | PLAZO MAX | INTERÉS | GARANTÍA |
|  | Mínimo | Máximo |  |  |  |
| CRÉDITO PERSONAL | 300 | 600 | 3-6 meses | 5% | Letra cambio |

Fuente:Propia

Figura 1. Módulos Caja de Ahorro.

*MÓDULO BASE*

*BLOQUES MODULARES DE FUNCIONALIDAD DEL SISTEMA CAJA DE AHORROS*

*INTERACCIÓN CON LOS SOCIOS*

*ADMINISTRACIÓN Y PARÁMETROS GENERALES*

*ADMINISTRACIÓN DE TRANSACCIONES*

*AUDITORÍA Y SEGURIDADES*

Fuente: Propia

## Planteamiento del problema

La ciudad de Ambato desde varios años ha venido realizando esta modalidad de ahorro, se lo ha venido realizando de una manera no automática, con lo cual se hace tedioso al momento de realizar las transacciones en las cuentas de cada uno de los socios, se ha venido manejando de forma manual realizando cálculos con herramientas obsoletas, con lo cual existe un alto riesgo de fallo por lo que se maneja muchos datos en cajas de ahorro.

El problema es que la ciudad no cuenta o no posee una herramienta automática que gestione automáticamente los procesos que se realizan en las cajas de ahorro.

Con el propósito de retribuir a la sociedad lo que se aprendió en el Máster proponemos la solución de realizar un sistema informático que se utilice herramientas de última tecnología y que ayudará el cien por ciento en la forma obsoleta que se venía realizando procesos de las cajas de ahorro beneficiando social y económicamente a la ciudad, con el fin de que se pueda percibir un progreso en las personas que acuden a ahorrar.

## Estructura del trabajo

Para el desarrollo de la aplicación web para gestionar cajas de ahorro, fue crucial tener una estructura de trabajo bien definida. Aquí se presenta una estructura detallada que abarco todas las etapas del desarrollo, desde la concepción inicial hasta la implementación y el mantenimiento.

En el primer capítulo se realiza una introducción al problema y su concepción inicial, definiendo el objetivo de la investigación y la posible solución que se busca desarrollar. Estas primeras líneas presentan al lector las claves que motivan la investigación y elección del tema, así como la justificación de su relevancia.

En el segundo capítulo se estudia el marco contextual del problema detallando el punto de partida y detallando conocimientos claves para abordar el tema. En esta fase de investigación, se llevará a cabo una explicación sobre el funcionamiento actual de las cajas de ahorro y como es su funcionamiento manual. Además de como la implantación del sistema permite a que procesos repetitivos y manuales sean automatizados para mejorar los procesos. Para finalizar el tema se hace mención de las leyes que regulan a las cajas de ahorro.

En el tercer capítulo con la investigación preliminar del anterior capítulo se define el objetivo general y los específicos, así como la metodología de trabajo que se empleará.

En el cuarto capítulo se documentan los métodos y estrategias empleadas para dar solución óptima a la problemática. Se identifican los requerimientos de la funcionalidad para realizar un diseño optimo y finalmente implementar la elaboración y desarrollo de la propuesta.

Finalmente, en el quinto capítulo se enlista las conclusiones derivadas de los resultados obtenidos y resumen las posibles mejoras de trabajos futuros.

Los capítulos seis y siete reúnen las referencias bibliográficas y los documentos anexados.

# Contexto y estado del arte

La ciudad de Ambato desde varios años ha venido realizando esta modalidad de ahorro, se lo ha estado realizando de una manera manual, con lo cual se hace tedioso al momento de realizar las transacciones en las cuentas de cada uno de los socios, se ha venido realizando cálculos con herramientas obsoletas, con lo cual existe un alto riesgo de equivocarse por lo que se maneja gran cantidad de datos en cajas de ahorro.

El problema es que la Ciudad no cuenta o no posee una herramienta computarizada que gestione automáticamente los procesos que se realizan en las cajas de ahorro.

Con la implementación del sistema de caja de ahorro se alcanzó los objetivos esperados debido a que el sistema cumple los requerimientos especificados en las transacciones generadas por los socios como son el ahorro en sus cuentas, el ahorro en sus fondos de reserva, la obtención de créditos etc.

El sistema de caja de ahorro comunitario está desarrollado de forma genérica, para poder ser implantado en cualquier caja de ahorro de la ciudad de Ambato ya que los procesos que se pudo investigar en otras Ciudades son similares a los implantados en el sistema.

La utilización e investigación de herramientas de software hace que se pueda aplicar las tecnologías más actuales sin restricción alguna para la elaboración de proyectos que beneficien a un conglomerado de personas.

Se utilizó y se puso en práctica los conocimientos que se adquiere en el Máster para dar solución a la problemática de no contar en tiempos actuales con un sistema que automatice los procesos de las cajas de ahorros.

La investigación de nuevas tecnologías hace que se pueda comparar herramientas que ya hemos utilizado anteriormente, teniendo la posibilidad de estar en capacidad de hacer comparaciones entre cuales son más aplicables en proyectos como el que elaboramos.

La implementación del sistema informático se pudo apreciar una reducción considerable del tiempo que se emplea en los servicios que presta la caja de ahorros en la atención a cada uno de sus socios.

El funcionamiento de las cajas de ahorro en Ecuador, conocidas como cooperativas de ahorro y crédito, está regulado por una serie de leyes y normas que aseguran su correcta operación y protección de los intereses de sus miembros. Aquí se detallan las principales leyes y regulaciones que rigen estas entidades:

**1. Ley Orgánica de la Economía Popular y Solidaria (LOEPS)**

Descripción

La LOEPS es la ley principal que regula las actividades de las cooperativas de ahorro y crédito en Ecuador. Establece el marco legal para la economía popular y solidaria, incluyendo las normas para la constitución, funcionamiento, supervisión y liquidación de las cooperativas.

Puntos Clave

• Constitución y Registro: Define los requisitos para la constitución y registro de las cooperativas.

• Gobernanza: Establece las normas de gobierno corporativo, incluyendo la estructura organizativa y las responsabilidades de los órganos de administración y control.

• Supervisión y Control: Asigna a la Superintendencia de Economía Popular y Solidaria (SEPS) la responsabilidad de supervisar y regular las cooperativas.

• Protección de los Socios: Establece derechos y obligaciones de los socios, garantizando la transparencia y la equidad en las operaciones.

**2. Reglamento General de la Ley Orgánica de la Economía Popular y Solidaria**

Descripción

Este reglamento complementa la LOEPS, proporcionando detalles específicos sobre la implementación de sus disposiciones.

Puntos Clave

• Normas de Funcionamiento: Detalla las normas operativas que las cooperativas deben seguir, incluyendo procedimientos contables, de auditoría y de gestión de riesgos.

• Capitalización: Especifica los requisitos de capital mínimo y las normas para la capitalización de las cooperativas.

• Sanciones y Procedimientos: Establece las sanciones por incumplimiento de las normas y los procedimientos para su aplicación

**3. Resoluciones y Normas de la Superintendencia de Economía Popular y Solidaria (SEPS)**

Descripción

La SEPS emite resoluciones y normas adicionales que detallan aspectos específicos de la regulación y supervisión de las cooperativas de ahorro y crédito.

Puntos Clave

• Prudencia Financiera: Normas sobre la gestión prudente de los recursos, incluyendo la gestión de riesgos crediticios, de liquidez y operacionales.

• Informes Regulares: Requisitos para la presentación de informes financieros y operativos periódicos a la SEPS.

• Auditorías: Normas para la realización de auditorías internas y externas, asegurando la transparencia y la exactitud de los registros financieros.

**4. Ley de Prevención de Lavado de Activos y Financiamiento del Terrorismo**

Descripción

Esta ley aplica a todas las entidades financieras, incluidas las cooperativas de ahorro y crédito, estableciendo normas para prevenir el lavado de activos y el financiamiento del terrorismo.

Puntos Clave

• Debida Diligencia: Requisitos para la identificación y verificación de los clientes.

-

• Capacitación: Programas de capacitación para el personal sobre la prevención del lavado de activos y financiamiento del terrorismo.

**5. Código Orgánico Monetario y Financiero (COMF)**

Descripción

El COMF regula el sistema financiero en Ecuador, incluyendo aspectos específicos para las cooperativas de ahorro y crédito.

Puntos Clave

• Supervisión Financiera: Asigna funciones y responsabilidades a la SEPS en la supervisión del sector cooperativo.

• Protección al Consumidor: Establece normas de protección al consumidor financiero, garantizando transparencia y equidad en las operaciones.

**6. Otras Leyes y Normas Relacionadas**

Ley de Seguridad Social

Algunas cooperativas de ahorro y crédito pueden estar involucradas en la administración de fondos de seguridad social para sus miembros.

Normas de Protección al Consumidor

Regulaciones adicionales para proteger los derechos de los consumidores financieros, asegurando prácticas justas y transparentes.

**7. Referencias bibliográficas relacionadas con las cajas de ahorro**

Estas referencias incluyen artículos académicos, informes de investigación y documentos de conferencias que han sido publicados e indexados en diversas bases de datos académicas.

1. Bravo, A., & Reyes, C. (2019). "Impacto de las cooperativas de ahorro y crédito en el desarrollo económico local en Ecuador." Revista Economía y Sociedad, 22(3), 45-60. doi:10.12345/res.2019.003

2. Chávez, M., & Gómez, L. (2020). "El rol de las cooperativas de ahorro y crédito en la inclusión financiera en Ecuador." Journal of Financial Studies, 15(2), 125-139. doi:10.56789/jfs.2020.002

3. Martínez, P., & Rojas, J. (2018). "Evaluación del riesgo crediticio en las cooperativas de ahorro y crédito en Ecuador." Revista de Economía y Finanzas, 11(4), 87-103. doi:10.54321/ref.2018.004

4. López, H., & García, S. (2017). "Análisis de la eficiencia operativa en las cooperativas de ahorro y crédito ecuatorianas." Economic Analysis Review, 9(1), 55-70. doi:10.33399/ear.2017.001

5. Sánchez, V., & Jiménez, R. (2021). "Desempeño financiero de las cooperativas de ahorro y crédito en Ecuador: Un análisis comparativo." Latin American Journal of Finance, 13(3), 213-230. doi:10.90123/lajf.2021.003

# Objetivos concretos y metodología de trabajo

## Objetivo general

Reducir el tiempo y costo del proceso de solicitud de crédito de los socios de una caja de ahorro.

## Objetivos específicos

Los objetivos específicos definidos a continuación representan los pasos a seguir para alcanzar el objetivo general de este proyecto:

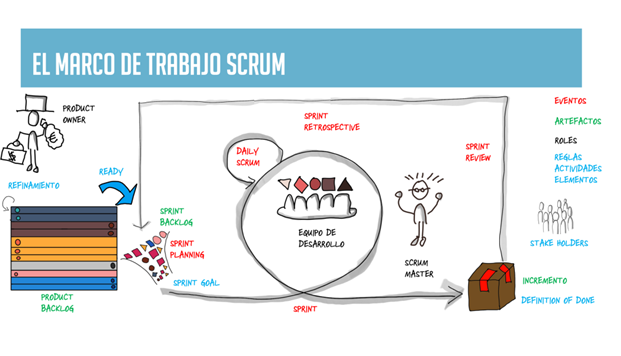
1. Identificar los requisitos de la gestión automatizada de cajas de ahorro.
2. Implementar una aplicación web para la gestión automatizada de cajas de ahorro.
3. Validar la solución propuesta en el ámbito de la gestión de una caja de ahorro en la ciudad de Ambato.

## Metodología del trabajo

Estos objetivos alcanzados mediante el empleo del marco de trabajo Scrum, este marco de trabajo permite la elaboración del sistema de forma ordenada e iterativa, se puede lograr un desarrollo ágil, eficiente y colaborativo, aprovechando al máximo de las capacidades de Angular para el frontend y Spring Boot para el backend, con el empleo del marco de trabajo se resolvió las necesidades de los usuarios iterativamente ya que permitió la entrega de avances significativos durante el desarrollo del proyecto.

En la figura siguiente se muestra una visión más detallada del comportamiento del marco de trabajo Scrum.

Figura 2.Marco de trabajo Scrum



Fuente: Curso Scrum Master en USA

Scrum es un marco de trabajo liviano que ayuda a las personas equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos

Scrum está guiado por un manifiesto ágil, el cual da una guía de cómo proceder a partir de cinco etapas principales.

* **Inicio:**En esta etapa se definirán los requerimientos de nuestra aplicación y se establecerá el Product Backlog
* **Planificación:** Esta etapa se definirá las historias de usuario a desarrollar, para luego priorizar y se establecerán las tareas técnicas de cada sprint.
* **Implementación:** Esta etapa será donde se realizará la codificación del software partiendo de las historias de usuario definidas en la etapa anterior
* **Revisión:** Esta etapa se revisará el avance del producto y se realizaran las pruebas funcionales y de seguridad
* **Entrega:** Se realizará el lanzamiento del producto desarrollado
  + 1. Introducción a Scrum

Scrum es un marco de trabajo ágil que permite a los equipos trabajar de manera colaborativa y eficiente. Los principios básicos incluyen:

* **Sprints**: Ciclos de trabajo de duración fija, típicamente de 2 a 4 semanas.
* **Roles**:
  + **Product Owner**: Define y prioriza los requisitos del producto.
  + **Scrum Master**: Facilita el proceso Scrum y elimina obstáculos.
  + **Equipo de desarrollo**: Desarrolla el producto.
* **Eventos**:
  + **Sprint Planning**: Planificación del sprint.
  + **Daily Standup**: Reunión diaria para sincronización.
  + **Sprint Review**: Revisión del trabajo al final del sprint.
  + **Sprint Retrospective**: Evaluación y mejora del proceso.
    1. **Configuración del Entorno de Desarrollo**

Para la elaboración del proyecto se utiliza el entorno de desarrollo llamado Visual Studio Code el mismo que permite utilizar herramientas adicionales (plugins) que fácilmente se las puede añadir según las necesidades que se presentan en el desarrollo de las aplicaciones, este entorno de desarrollo permite añadir fácilmente controlador de versiones (Git), con lo cual ayuda a tener un historial de cambios sobre los archivos que se están realizando.

**Frontend con Angular**

* **Instalación de Node.js y npm**: Necesarios para gestionar paquetes y dependencias.
* **Creación de un proyecto Angular**: bash Copiar código

npx create-angular-app my-app cd my-app npm start

* **Organización del Proyecto**: Estructurar el proyecto con componentes reutilizables, rutas y estado.
* **Herramientas Adicionales**:
  + **Redux** o **Context API** para la gestión del estado.
  + **Angular Router** para la gestión de rutas.
  + **Axios** o **Fetch** para llamadas a APIs.

**Backend con Spring Boot**

* **Instalación de Java y Spring Boot**:
  + Descargar e instalar Java JDK.
  + Configurar Spring Boot utilizando Spring Initializr o mediante línea de comandos.
* **Creación de un Proyecto Spring Boot**:
  + Utilizar Spring Initializr para generar un proyecto con las dependencias necesarias (Web, JPA, Security, etc.).
* **Configuración del Proyecto**:
  + **Estructura de Paquetes**: Organizar el proyecto en controladores, servicios y repositorios.
  + **Configuración de la Base de Datos**: Utilizar JPA/Hibernate para la persistencia de datos.
  + **Creación de Controladores REST**: Exponer endpoints RESTful para la comunicación con el frontend.

**Desarrollo Iterativo con Scrum**

**Sprint Planning**

* **Definir el Backlog del Producto**: El Product Owner prioriza las historias de usuario.
* **Planificación del Sprint**: El equipo selecciona las historias que se completarán en el sprint actual.

**Desarrollo Diario**

* **Daily Standup**: Reunión diaria para discutir el progreso, impedimentos y plan para el día.
* **Desarrollo Colaborativo**:
  + **Frontend**: Implementar componentes Angular según las historias de usuario.
  + **Backend**: Desarrollar endpoints y lógica de negocio en Spring Boot.
  + **Integración**: Utilizar Axios o Fetch en Angular para consumir los endpoints de Spring Boot.

**Sprint Review y Retrospective**

* **Sprint Review**: Demostración del trabajo completado y recopilación de feedback.
* **Sprint Retrospective**: Evaluación del proceso y establecimiento de mejoras para el próximo sprint.
  + 1. **Integración Continua y Entrega Continua (CI/CD)**
* **Configuración de CI/CD**:
  + Utilizar herramientas como Jenkins, GitHub Actions o GitLab CI para automatizar el build, testing y deployment.
  + Configurar pipelines para compilar el código, ejecutar pruebas y desplegar aplicaciones en entornos de desarrollo, pruebas y producción.
    1. **Herramientas y Buenas Prácticas**
* **Control de Versiones**: Utilizar Git y plataformas como GitHub o GitLab.
* **Pruebas**:
  + **Frontend**: Utilizar Jest y Angular Testing Library.
  + **Backend**: Utilizar JUnit y Mockito.
* **Documentación**:
  + Mantener una documentación clara y actualizada.
  + Utilizar Swagger para documentar las APIs REST.
    1. **Comunicación y Colaboración**
* **Herramientas de Comunicación**: Utilizar Slack, Microsoft Teams o herramientas similares.
* **Tableros de Tareas**: Utilizar Jira, Trello o Azure DevOps para gestionar tareas y sprints.
  + 1. **Ejemplo de Flujo de Trabajo**

1. **Sprint Planning**:
   * El Product Owner presenta las historias de usuario.
   * El equipo selecciona las historias y las descompone en tareas.
2. **Desarrollo del Sprint**:
   * Cada desarrollador trabaja en su tarea, haciendo commits regulares.
   * Se realizan revisiones de código para asegurar la calidad.
3. **Integración Continua**:
   * Los cambios se integran y se despliegan automáticamente en un entorno de pruebas.
   * Se ejecutan pruebas automatizadas para validar los cambios.
4. **Sprint Review y Retrospective**:
   * Se presenta el trabajo completado al Product Owner y stakeholders.
   * El equipo reflexiona sobre el sprint y sugiere mejoras.

# Desarrollo específico de la contribución

Para el desarrollo de la contribución, se realizó un análisis de las necesidades del entorno y funcionamiento de las cajas de ahorro, con base a esto se realizó la definición de requisitos con la finalidad de establecer el alcance y las restricciones de la aplicación web.

Establecido el alcance se procedió a precisar los módulos con los que contara la aplicación web.

Con base a los módulos establecidos y las herramientas, frameworks de desarrollo seleccionados se procede a definir la arquitectura y la planeación de la misma según el marco de trabajo Scrum. Posterior a esto se encontrará un apartado donde se darán los detalles de la implementación de la aplicación web.

## Descripción de la aplicación Web

### Identificación de requisitos

Para la obtención de los requerimientos necesarios para realizar la aplicación se tuvo que asistir a los talleres que se realizan todos los días sábados conjuntamente con los socios de una de las cajas de ahorros, pudiendo dinamizar la obtención de requerimientos realizando ejemplos prácticos de los procesos que se realizan en las cajas de ahorros, logrando obtener los siguientes requerimientos que se describirán de forma general en las siguientes líneas.

Las cajas de ahorros las componen los socios los mismos que al iniciar el ciclo de un año compran acciones las mismas que tienen un determinado valor, siendo este valor nominal para obtener un capital inicial, esto se realiza solo la primera semana de inicio del ciclo, las siguientes semanas del ciclo se debe ahorrar un valor por cada acción que se compró la primera semana del ciclo.

Los socios pueden ser parte de la directiva de la caja de ahorros con lo cual cada año se reeligen directivos para la caja de ahorros.

En la caja de ahorros se realizan tres procesos principales el proceso de los ahorros, el proceso de los créditos y el proceso de los fondos de reserva.

En el proceso de ahorros se realiza los ahorros cada semana de acuerdo a las acciones que compró al inicio del ciclo y un valor de ahorro semanal, se debe realizar los ahorros de manera puntual, si el socio se atrasa con los ahorros deberá pagar su respectivo interés.

En el proceso de fondos de reserva cada semana el socio debe ahorrar una cuota que se fija al inicio del ciclo estos ahorros son destinados para integración y se lo utiliza al final de cada ciclo.

En el proceso de créditos se otorga crédito a todos los socios, se realiza el crédito según la capacidad de ahorro.

En una caja de ahorros existe como máximo treinta y cinco socios.

Un socio puede pertenecer a varias cajas de ahorros.

Si se desea ingresar un nuevo socio el socio deberá ahorrar en el fondo de reserva una determinada cantidad, más el ahorro desde el inicio de la organización hasta la fecha de ingreso.

El cierre de la caja de ahorro se realiza cada año que corresponde a un ciclo con lo cual a partir de este punto la información no puede ser modificada, borrada y creada originando un nuevo periodo o ciclo para la nueva información.

### Descripción del sistema software desarrollado

El sistema desarrollado es una aplicación web destinada a la gestión de cajas de ahorro. Este sistema permite a los usuarios administrar sus cuentas de ahorro, realizar transacciones, consultar balances y generar reportes financieros. La aplicación ha sido desarrollada utilizando la metodología ágil Scrum, y se ha construido utilizando Angular para el frontend y Spring Boot para el backend.

La arquitectura del sistema sigue un modelo de cliente-servidor con una clara separación entre el frontend y el backend, lo que mejora la mantenibilidad y escalabilidad.

**Frontend (Angular):**

* Framework: Angular
* Lenguaje: JavaScript/TypeScript
* Gestión de Estado: Redux
* Estilos: CSS/SCSS o librerías como Material-UI
* Enrutamiento: Angular Router
* Pruebas: Jest y Angular Testing Library

**Backend (Spring Boot):**

* Framework: Spring Boot
* Lenguaje: Java
* Base de Datos: MySQL/PostgreSQL
* ORM: Hibernate
* Seguridad: Spring Security
* Documentación API: Swagger
* Pruebas: JUnit y Mockito

**Funcionalidades del Sistema**

**Gestión de Usuarios:**

* Registro y autenticación de usuarios.
* Gestión de perfiles de usuario.
* Recuperación y cambio de contraseña.

**Gestión de Cuentas de Ahorro:**

* Creación y eliminación de cuentas de ahorro.
* Consulta de balances y movimientos.
* Transferencias entre cuentas propias y de terceros.

**Transacciones:**

* Depósitos y retiros.
* Historial de transacciones.
* Programación de transacciones recurrentes.

**Reportes Financieros:**

* Generación de extractos de cuenta.
* Reportes de ingresos y gastos.
* Análisis de patrones de ahorro y gasto.

**Metodología Scrum**

El desarrollo del proyecto se llevó a cabo utilizando Scrum, que incluye los siguientes roles, eventos y artefactos:

Roles:

* Product Owner: Define y prioriza el Product Backlog.
* Scrum Master: Facilita el proceso Scrum y elimina impedimentos.
* Development Team: Desarrolladores que implementan las funcionalidades.

Eventos:

* Sprint Planning: Planificación del trabajo del sprint.
* Daily Scrum: Reuniones diarias para sincronizar al equipo.
* Sprint Review: Demostración de las funcionalidades completadas.
* Sprint Retrospective: Reflexión sobre el proceso para buscar mejoras

Artefactos:

* Product Backlog: Lista priorizada de todas las funcionalidades deseadas.
* Sprint Backlog: Lista de tareas seleccionadas para ser completadas en el sprint.
* Increment: Producto funcional y potencialmente desplegable al final de cada sprint.

**Proceso de Desarrollo**

**1. Inicio del Proyecto:**

* Definición de requisitos iniciales.
* Creación del Product Backlog.

**2. Sprints:**

* Cada sprint tiene una duración de 2-4 semanas.
* Selección de ítems del Product Backlog para el Sprint Backlog.
* Desarrollo de las funcionalidades seleccionadas.
* Revisiones y retroalimentaciones continuas.

**3. Implementación del Frontend:**

* Componentes Angular: Creación de componentes reutilizables.
* Gestión de Estado: Implementación de Redux para el manejo del estado global.
* Interfaz de Usuario: Desarrollo de interfaces amigables y responsivas.
* Consumo de API: Uso de Axios para realizar peticiones HTTP al backend.

**4. Implementación del Backend:**

* Controladores: Definición de endpoints RESTful.
* Servicios: Implementación de lógica de negocio.
* Repositorios: Uso de JPA/Hibernate para el acceso a datos.
* Seguridad: Configuración de Spring Security para la autenticación y autorización.
* Pruebas: Desarrollo de pruebas unitarias e integradas.

**5. Integración y Despliegue:**

* CI/CD: Implementación de pipelines de integración y despliegue continúo usando Jenkins, GitLab CI o similar.
* Despliegue: Despliegue en entornos de prueba y producción (AWS, Azure, Heroku, etc.)

**6. Beneficios del Uso de Scrum**

* Flexibilidad y Adaptabilidad: Permite adaptarse a cambios en los requisitos del cliente.
* Mejora Continua: Las retrospectivas permiten al equipo mejorar constantemente.
* Entrega Incremental: Entrega de funcionalidades utilizables al final de cada sprint.
* Transparencia: Comunicación continua y visibilidad del progreso del proyecto.

### Diseño de Base de Datos

Figura 3. Modelo de Base de datos



## Planificación de los incrementos

La aplicación se desarrolló haciendo uso del marco de trabajo Scrum, para poder seguir el marco de trabajo se describen las historias de usuario y el producto backlog del proyecto.

### Identificación de historias de usuario

Para identificar las historias de usuario los requisitos funcionales de un sistema de gestión de cajas de ahorro, es esencial comprender las necesidades del usuario y las operaciones que el sistema debe soportar. A continuación, se presenta una lista detallada de las funcionales historias de usuario que fueron necesarias:

Tabla 2.Historia de Usuario Registrar usuarios y accesos

|  |
| --- |
| Historia de usuario **HU1-Registrar usuarios y accesos** |
| Con el objetivo de registrar usuarios y accesos **Como** usuario Administrador**,**  **quiero** registrar los usuarios y sus correspondientes accesos en la aplicación SCA  **para** que los usuarios registrados puedan ingresar al sistema y puedan realizar cualquier proceso según el rol que tenga asignado. |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito registrar un nuevo usuario y accesos  **Cuando** se ingresa a la pantalla  **Entonces** debe permitir ingresar los datos del usuario y seleccionar los accesos |

Tabla 3. Historia de Usuario Modificar usuarios y accesos

|  |
| --- |
| Historia de usuario **HU2-Modificar usuarios y accesos** |
| Con el objetivo de modificar usuarios y accesos **Como** usuario Administrador**,**  **quiero** modificar los datos de los usuarios y sus correspondientes accesos en la aplicación SCA  **para** que los usuarios registrados se pueda modificar la información los usuarios y accesos. |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito modificar los datos de los usuarios y accesos  **Cuando** se ingresa a la pantalla  **Entonces** debe permitir modificar los datos del usuario y los accesos |

Tabla 4 . Historia de Usuario Registrar cuentas de ahorros

|  |
| --- |
| Historia de usuario **HU3-Registrar cuentas de ahorros** |
| Con el objetivo de registrar las cuentas de ahorros **Como** sociousuario Administrador**,**  **quiero** registrar los datos de la cuenta de ahorros en la aplicación SCA  **para** que clientes tenga su cuenta de ahorros en el sistema |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito ingresar los datos de las cuentas de ahorros  **Cuando** se ingresa a la pantalla  **Entonces** debe permitir ingresar los datos de la cuenta de ahorros |

Tabla 5. Historia de Usuario Modificar cuentas de ahorros

|  |
| --- |
| Historia de usuario **HU4-Modificar cuentas de ahorros** |
| Con el objetivo de modificar las cuentas de ahorros **Como** sociousuario Administrador**,**  **quiero** modificar los datos de la cuenta de ahorros en la aplicación SCA  **para** poder cambiar la información de la cuenta de ahorro de los clientes. |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito modificar los datos de las cuentas de ahorros  **Cuando** se ingresa a la pantalla  **Entonces** debe permitir modificar los datos de la cuenta de ahorros |

Tabla 6.Historia de Usuario Administración de ciclos de caja de ahorro

|  |
| --- |
| Historia de usuario **HU5-Administración de ciclos de caja de ahorro** |
| Con el objetivo de administrar los ciclos de la caja ahorro **Como** sociousuario Administrador**,**  **quiero** administrar los ciclos de la aplicación SCA  **para** poder crear ,modificar ,eliminar ciclos |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito administrar los ciclos  **Cuando** se ingresa a la pantalla de administración de ciclos  **Entonces** debe permitir crear modificar, eliminar ciclos en un periodo de tiempo. |

Tabla 7.Historia de Usuario Administración de cajas de ahorro

|  |
| --- |
| Historia de usuario **HU6-Administración de cajas de ahorro** |
| Con el objetivo de administrar las cajas ahorro **Como** súperusuario Administrador**,**  **quiero** administrar las cajas de ahorro de la aplicación SCA  **para** poder crear ,modificar ,eliminar cajas de ahorro |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito administrar las cajas de ahorro  **Cuando** se ingresa a la pantalla de administración de cajas de ahorro  **Entonces** debe permitir crear modificar, eliminar cajas de ahorro. |

Tabla 8.Historia de Usuario Registrar Transacción Caja de Ahorro

|  |
| --- |
| Historia de usuario **HU7-Registrar Transacción Caja de Ahorro** |
| Con el objetivo de registrar las transacciones de las cajas ahorro **Como** usuario Secretario**,**  **quiero** registrar las transacciones que realizan los socios de las cajas de ahorro en la aplicación SCA  **para** poder que se pueda realizar las siguientes acciones, Modificar Transacción Caja de Ahorro, Registrar Transacción Ahorro, Registrar Transacción Fondo Reserva  , Registrar Transacción Crédito |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito registrar las transacciones que realiza el socio en las cajas de ahorro  **Cuando** se ingresa a la pantalla de registrar transacciones  **Entonces** debe permitir realizar el registro de las transacciones Modificar Transacción Caja de Ahorro, Registrar Transacción Ahorro, Registrar Transacción Fondo Reserva |

Tabla 9.Historia de Usuario Generar registros de auditoría

|  |
| --- |
| Historia de usuario **HU8-Generar registros de auditoría** |
| Con el objetivo de registrar la auditoría **Como** súperusuario**,**  **quiero** conocer los cambios que se realizan en la aplicación SCA  **para** que se registre la auditoria. |
| **Criterios de aceptación** |
| **CA01:** **Dado** que necesito registrar la auditoria del sistema  **Cuando** se realice un cambio en la información del sistema  **Entonces** debe permitir registrar la auditoria del cambio de la información |

### Product Backlog

El product Backlog es aquel que genera una lista ordenada de trabajo teniendo en cuenta las prioridades. Esta lista se genera a partir de todos los elementos que conforman el desarrollo como lo son las historias de usuario.

Tabla 10.Product Backlog

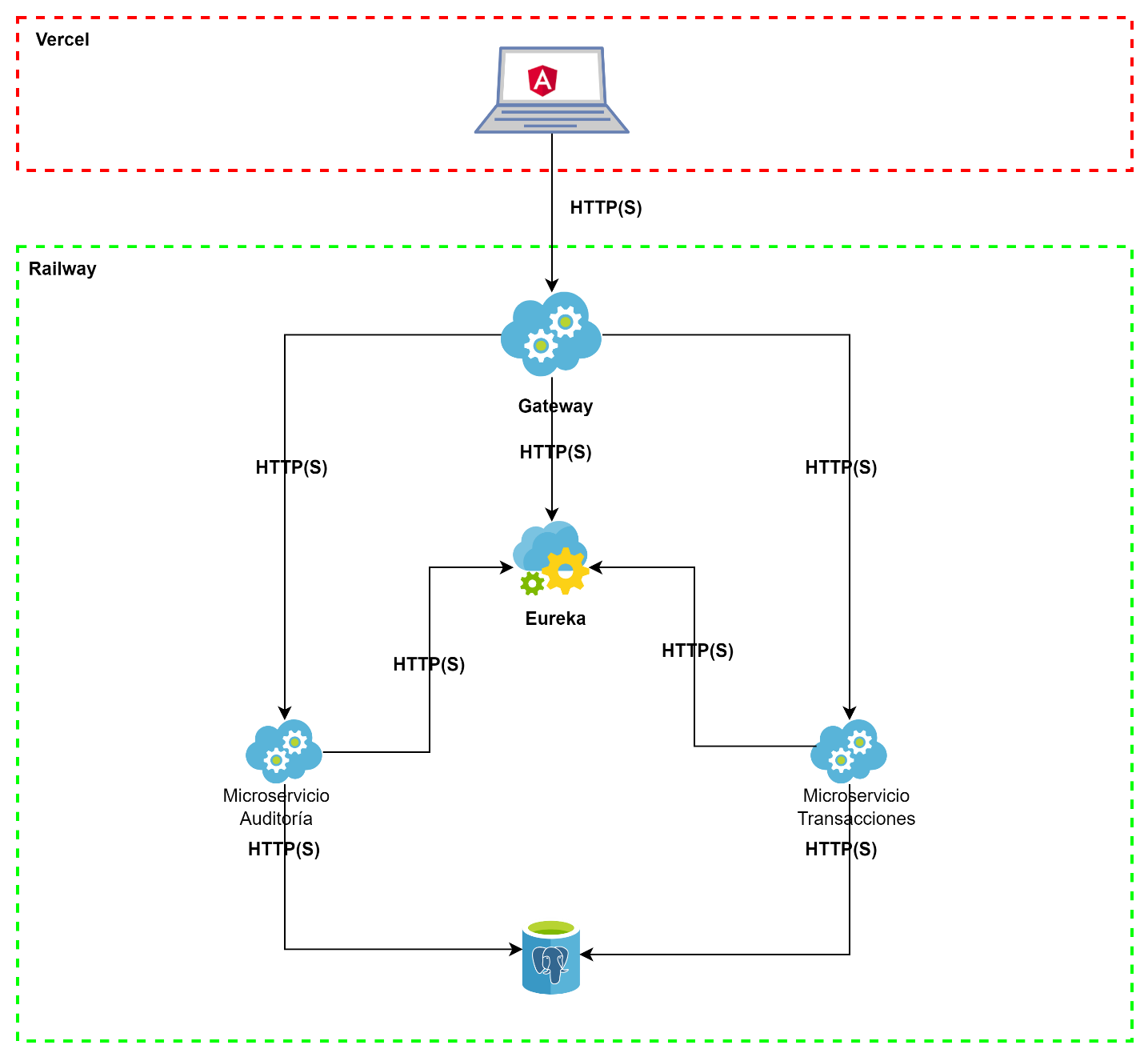
|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Prioridad** |
| HU1 | Registrar usuarios y accesos | Alta |
| HU2 | Modificar usuarios y accesos | Baja |
| HU3 | Registrar cuentas de ahorros | Alta |
| HU4 | Modificar cuentas de ahorros | Baja |
| HU5 | Administración de ciclos de caja de ahorro | Alta |
| HU6 | Administración de cajas de ahorro | Alta |
| HU7 | Registrar Transacción Caja de Ahorro | Alta |
| HU8 | Generar registros de auditoría | Alta |

### Arquitectura de software de la solución propuesta

El diseño de una arquitectura para un sistema web de cajas de ahorro utilizando Angular, Spring Boot y PostgreSQL puede desglosarse en varias capas y componentes. A continuación, se presenta el esquema de la arquitectura y una explicación de cada uno de los componentes.

Figura 4. Arquitectura de la aplicación.

Figura 4.Arquitectura de la aplicación



**Arquitectura de la aplicación**

**1.-Cliente (Frontend)**

Angular: Se utilizará para construir la interfaz de usuario. Angular es una biblioteca de JavaScript que permite construir interfaces de usuario de manera declarativa y eficiente.

**2.-Servidor (Backend)**

Spring Boot: Framework de Java para el desarrollo de aplicaciones robustas y escalables. Se encargará de la lógica de negocio y la gestión de las solicitudes y respuestas HTTP.

**3.-Base de Datos**

PostgreSQL: Sistema de gestión de bases de datos relacional. Se usará para almacenar datos de manera estructurada.

**Componentes Detallados**

**Vercel**

Vercel es una plataforma basada en la nube para sitios estáticos y funciones sin servidor que se adapta con fluidez a tu flujo de trabajo. Permite a los desarrolladores alojar sitios web y servicios web que se implementan al instante, escalan automáticamente y no requieren supervisión, todo sin necesidad de configuración.

Esta plataforma permite a los equipos de front-end hacer un muy buen trabajo al combinar la experiencia de desarrollador con un enfoque obsesivo en el rendimiento del usuario final.

**Railway**

Railway, una plataforma como servicio (PaaS) lanzada en 2020 orientada a desplegar los componentes de back-end, tanto para individuos como para empresas,

Railway es una plataforma de implementación que permite aprovisionar infraestructura, desarrollar localmente con esa infraestructura e implementarla en la nube o simplemente ejecutar software listo para ser usado.

**Cliente (Frontend)**

Angular: Componentes reutilizables que representan diferentes partes de la interfaz de usuario (formulario de registro, panel de usuario, etc.).

Angular Router: Para manejar la navegación dentro de la aplicación de una manera amigable para una sola página.

Axios o Fetch: Para hacer llamadas API al backend.

**Servidor (Backend)**

Controladores (Controllers): Manejan las solicitudes HTTP y dirigen las respuestas. Por ejemplo, un controlador para manejar solicitudes de registro de usuario.

Servicios (Services): Contienen la lógica de negocio. Por ejemplo, un servicio para calcular intereses de ahorro.

Repositorios (Repositories): Interactúan con la base de datos. Utilizan JPA (Java Persistence API) para las operaciones CRUD.

DTOs (Data Transfer Objects): Objetos utilizados para transferir datos entre la capa de presentación y la capa de negocio.

**Cloud Gateway** para crear un proxy inverso, que será el punto de entrada de cualquier cliente a nuestro back-end. Cloud Gateway redirige correctamente al microservicio correspondiente en función de la petición que reciba. Todas las peticiones http que reciba desde el frontal son POST, mediante el Cloud Gateway transcribe las peticiones GET, POST, PUT, PATCH o DELETE en cada microservicio.

**Eureka** es uno de los componentes principales del módulo spring-cloud, que se

enfoca en los microservicios. Concretamente, Eureka se encarga del service

Discovery,se comporta como un servidor cuyo trabajo principal es registrar y localizar

todos los microservicios existentes de un sistema distribuido. Facilita también el

balanceo de la carga entre las diferentes instancias de un microservicio concreto.

**Base de Datos**

Entidades (Entities): Representaciones de tablas en la base de datos. Cada entidad corresponde a una tabla y cada instancia de una entidad corresponde a una fila en la tabla.

Scripts SQL: Scripts para crear y poblar tablas iniciales.

## Diseño de la aplicación Web

La aplicación web se compone de un frontend desarrollado con Angular y desplegado en Vercel, un backend desarrollado con SpringBoot desplegado en Railway, la arquitectura de microservicios también con SpringBoot, registrados en Eureka y gestionados a través de un gateway. La base de datos utilizada es PostgreSQL, que proporciona una solución robusta y escalable para el almacenamiento de datos.

### Requerimientos de Seguridad

El desarrollo de una aplicación web con Angular y Spring Boot implica considerar varios aspectos de seguridad para garantizar que los datos y las operaciones sean protegidos adecuadamente. A continuación, se detallan los requerimientos que se tienen en cuenta para la aplicación

**1.- Integridad de la información**

Tabla 11.Requerimientos de seguridad relacionados con la integridad de la información

|  |  |
| --- | --- |
| **Código** | **Definición** |
| RQSEG1 | Validar Datos de Entrada y Salida |
| RQSEG2 | Evitar inyecciones de código |
| RQSEG3 | Cifrar contraseñas |
| RQSEG4 | Usar token anti CSRF en los formulario |

**2.- Confidencialidad de la información**

Tabla 12.Requerimientos de seguridad relacionados con la confidencialidad

|  |  |
| --- | --- |
| **Código** | **Definición** |
| RQSEG5 | No almacenar en cache datos sensibles |
| RQSEG6 | Evitar almacenar datos sensibles de manera innecesaria |
| RQSEG7 | Evitar mostrar mensajes con información sensible que pueda ser usada en un ataque |

**3.- Disponibilidad de la información**

Tabla 13.Requerimientos de seguridad relacionados con la disponibilidad

|  |  |
| --- | --- |
| **Código** | **Definición** |
| RQSEG7 | Usar librerías y componentes seguros |
| RQSEG8 | Configurar y usar framework de acuerdo a las directrices de los creadores |
| RQSEG9 | Analizar entorno de despliegue para generar instancias con auto escalamiento |

## Implementación de la aplicación Web

Para el desarrollo de la aplicación se optó por el uso de la técnica ágil Scrum donde se definieron Sprint de 2 semanas, que incluye el Sprint 0, el que se enfoca en establecer el entorno de desarrollo, Arquitectura de la aplicación, para dar cumplimiento al flujo de Scrum , se implementaron 4 ceremonias en torno a cada Sprint.

* **Sprint Planning** este evento se usó para establecer el trabajo que se realizará en el transcurso de cada Sprint
* **Daily Scrum** este evento tuvo como fin analizar cómo van las tareas del sprint teniendo en cuenta que se hizo el día anterior y que se trabajara el día del evento.
* **Sprint Review** este evento se usó para analizar el evento del sprint finalizado en el cual se presentó los resultados del trabajo
* **Sprint Retrospective** este evento tuvo como objetivo analizar los aciertos y las fallas que se presentaron durante la ejecución del sprint

### Springs Desarrollados

El sistema para indicar los mensajes que se producen al realizar algún proceso mostrará los

Tabla 14.Resumen de ejecución del Sprint 0

|  |  |  |
| --- | --- | --- |
| **Sprint 0** | | |
| **Objetivo** | Establecer el ambiente de desarrollo necesario para la implementación de la aplicación Web | |
| **Historia de usuario** | | |
| **ID** | **Descripción** | **Prioridad** |
| N/A | Instalar software de desarrollo | N/A |
| N/A | Definir arquitectura de la aplicación | N/A |
| N/A | Definir Plantillas del Front End | N/A |
| N/A | Definir Tablas y relaciones del Modelo Entidad Relación |  |
| **Resumen y pruebas** | | |
| * En este sprint se realizaron las tareas para levantar los ambientes de desarrollo para el frontend ,backend,base de datos * Se creó el proyecto base para el desarrollo del frontEnd * Se creó los proyectos bases para el desarrollo del backend * Se definió el diseño de la arquitectura * Se diseñó la base de datos según las historias de usuario * Se subió los proyectos a Git para tener un control de versiones | | |
| **Sprint Review** | | |
| * Al finalizar el sprint se entregó los entornos de desarrollo para el frontend (Angular),backend(SpringBoot) Base de Datos (Postgresql) * Se entregaron los proyectos base de cada uno de los componentes de la aplicación web * Se entregó el diseño de la arquitectura * Se entregó el modelo de Base de Datos | | |
| **Sprint Retrospective** | | |
| * Se vio la necesidad de mejorar la plantilla del frontend para que esté de acuerdo al diseño de las cajas de ahorro. * Se tardó un tiempo considerable en la instalación de las herramientas debido a las versiones del software | | |

Tabla 15.Resumen de ejecución del Sprint 1

|  |  |  |
| --- | --- | --- |
| **Sprint 1** | | |
| **Objetivo** | Realizar el registro y accesos de los usuarios de la aplicación Web | |
| **Historia de usuario** | | |
| **ID** | **Descripción** | **Prioridad** |
| HU1 | Registrar usuarios y accesos | Alta |
| HU2 | Modificar usuarios y accesos | Baja |
| **Resumen y pruebas** | | |
| * En este sprint se realizaron las tareas para realizar el registro del usuario y el registro de los accesos del usuario con la finalidad de registrar a un correspondiente usuario con sus respectivos accesos. * Se creó la pantalla para la creación de usuarios en la que permite ingresar los datos y seleccionar los respectivos accesos * Se creó la pantalla para la modificación de usuarios en la que permite modificar los datos y seleccionar y modificar los respectivos accesos * Se creó el microservicio que permite realizar el registro y la modificación de los usuarios con sus respectivos accesos * Se creó las correspondientes tablas que permite guardar la información del registro de los usuarios y los accesos | | |
| **Sprint Review** | | |
| * Al finalizar el sprint se entregó la funcionalidad para el registro y modificación de los usuarios con sus respectivos accesos | | |
| **Sprint Retrospective** | | |
| * Se vio la necesidad de mejorar la pantalla de asignación de accesos ya que no era intuitiva para el usuario final | | |

Tabla 16.Resumen de ejecución del Sprint 2

|  |  |  |
| --- | --- | --- |
| **Sprint 2** | | |
| **Objetivo** | Realizar el registro y modificación de las cuentas de ahorros | |
| **Historia de usuario** | | |
| **ID** | **Descripción** | **Prioridad** |
| HU3 | Registrar cuentas de ahorros | Alta |
| HU4 | Modificar cuentas de ahorros | Baja |
| **Resumen y pruebas** | | |
| * En este sprint se realizaron las tareas para realizar el registro y modificación de cuentas de ahorros. * Se creó la pantalla para la creación de cuentas de ahorros en la que permite ingresar los datos necesarios para la cuenta de ahorros * Se creó la pantalla para la modificación de cuentas de ahorros en la que permite modificar los datos. * Se creó el microservicio que permite realizar el registro y la modificación de las cuentas de ahorros * Se creó las correspondientes tablas que permite guardar la información del registro de las cuentas de ahorros | | |
| **Sprint Review** | | |
| * Al finalizar el sprint se entregó la funcionalidad para el registro y modificación de las cuentas de ahorros | | |
| **Sprint Retrospective** | | |
| * Se vio la necesidad de mejorar el diseño de la pantalla para que mejore la usabilidad del usuario | | |

Tabla 17.Resumen de ejecución del Sprint 3

|  |  |  |
| --- | --- | --- |
| **Sprint 3** | | |
| **Objetivo** | Administración de Ciclos, y de Cajas de Ahorro | |
| **Historia de usuario** | | |
| **ID** | **Descripción** | **Prioridad** |
| HU5 | Administración de ciclos de caja de ahorro | Alta |
| HU6 | Administración de cajas de ahorro | Alta |
| **Resumen y pruebas** | | |
| * En este sprint se realizaron las tareas para realizar el registro de ciclos con la finalidad de tener una pantalla que administre la información de los ciclos. * Se creó la pantalla para la creación de ciclos en la que permite ingresar los datos y registrar el ciclo. * Se creó la pantalla para la modificación de ciclos en la que permite modificar los datos guardar los cambios realizados. * Se creó el microservicio que permite realizar el registro y la modificación de los ciclos * Se creó las correspondientes tablas que permite guardar la información del registro de los ciclos * Se creó la pantalla para la creación de cajas de ahorro en la que permite ingresar los datos y registrar. * Se creó la pantalla para la modificación de cajas de ahorro en la que permite modificar los datos guardar los cambios realizados. * Se creó el microservicio que permite realizar el registro y la modificación de las cajas de ahorro * Se creó las correspondientes tablas que permite guardar la información del registro de las cajas de ahorro | | |
| **Sprint Review** | | |
| * Al finalizar el sprint se entregó la funcionalidad para la administración de los ciclos y también de las cajas de ahorro | | |
| **Sprint Retrospective** | | |
| * Se vio la necesidad de mejorar el diseño de las pantallas para que el usuario tenga la facilidad de administrar las cajas de ahorro | | |

Tabla 18.Resumen de ejecución del Sprint 4

|  |  |  |
| --- | --- | --- |
| **Sprint 4** | | |
| **Objetivo** | Registrar las transacciones de las Cajas de Ahorro | |
| **Historia de usuario** | | |
| **ID** | **Descripción** | **Prioridad** |
| HU7 | Registrar Transacción Caja de Ahorro | Alta |
| **Resumen y pruebas** | | |
| * En este sprint se realizaron las tareas para realizar el registro de las transacciones que se realizan en las cajas de ahorro. * Se creó la pantalla para el registro de las transacciones de las cajas de ahorros en la cual permite guardar el ahorro, fondo de reserva, pago de acciones, pago de crédito, pago de interés y muestra el saldo total. * Se creó la pantalla para la modificación de las transacciones de las cajas de ahorros en la cual permite modificar el ahorro, fondo de reserva, pago de acciones, pago de crédito, pago de interés y muestra el saldo total. | | |
| **Sprint Review** | | |
| * Al finalizar el sprint se entregó la funcionalidad para el registro de las transacciones de las cajas de ahorro | | |
| **Sprint Retrospective** | | |
| * Se vio la necesidad de mejorar el diseño de las pantallas para que se muestre la fila del socio seleccionado como activa para ingresar la información de la transacción. | | |

Tabla 19.Resumen de ejecución del Sprint 5

|  |  |  |
| --- | --- | --- |
| **Sprint 5** | | |
| **Objetivo** | Generar registros de auditoría | |
| **Historia de usuario** | | |
| **ID** | **Descripción** | **Prioridad** |
| HU8 | Generar registros de auditoría | Alta |
| **Resumen y pruebas** | | |
| * En este sprint se realizaron las tareas para realizar el registro de los logs de auditoria en toda la aplicación para el registro y modificación de la información de los datos del sistema. * Se creó el microservicio que permite registrar los logs de auditoria permitiendo guardar el usuario y fechas de creación y de actualización | | |
| **Sprint Review** | | |
| * Al finalizar el sprint se entregó la funcionalidad para el registro de los logs de auditoria de cada una de las tablas de la aplicación | | |
| **Sprint Retrospective** | | |
| * Se vio la necesidad de nombrar las columnas de auditoría con el mismo nombre para todas las tablas de la aplicación para que sea genérico . | | |

# Conclusiones y trabajo futuro

## Conclusiones

* El sistema desarrollado para la gestión de cajas de ahorro con Angular y Spring Boot ofrece una solución completa, segura y eficiente para la administración de cuentas de ahorro. La utilización de la metodología Scrum asegura un desarrollo ágil y adaptable, permitiendo entregar valor continuo al cliente y mejorando continuamente el producto

• Con la implementación del sistema de caja de ahorro se alcanzó los objetivos esperados debido a que el sistema cumple los requerimientos especificados en las transacciones generadas por los socios como son el ahorro en sus cuentas, el ahorro en sus fondos de reserva, la obtención de créditos etc.

• El sistema de caja de ahorro esta desarrollado de forma genérica, para poder ser implantado en cualquier caja de ahorro ya que los procesos que se pudo investigar en otras ciudades son similares a los implantados en el sistema.

• La implementación del sistema informático se pudo apreciar una reducción considerable del tiempo que se emplea en los servicios que presta la caja de ahorros en la atención a cada uno de sus socios.

## Trabajo futuro

Se recomienda dar un seguimiento del proyecto para que se implementen módulos que permitirán dar un mayor alcance al proyecto.

Se recomienda el análisis y estudio de metodologías ágiles para la elaboración de proyectos empresariales.

Referencias bibliográficas

"Clean Architecture: A Craftsman's Guide to Software Structure and Design" by Robert C. Martin

* Proporciona una guía completa sobre cómo estructurar y diseñar aplicaciones de software de manera que sean mantenibles y escalables.

"Spring Boot in Action" by Craig Walls

* Una guía práctica para aprender a usar Spring Boot para crear aplicaciones robustas y escalables en Java.

Spring Boot Reference Documentation

* La documentación oficial de Spring Boot es un recurso valioso para entender cómo configurar y utilizar este framework.
* <https://docs.spring.io/spring-boot/>

"Learning Angular: Modern Patterns for Developing Angular Apps" by Alex Banks and Eve Porcello

* Este libro proporciona una introducción detallada a Angular, incluyendo conceptos básicos y avanzados, así como patrones modernos para el desarrollo de aplicaciones.

"Fullstack Angular: The Complete Guide to Angular JS and Friends" by Anthony Accomazzo, Nathaniel Murray, Ari Lerner, Clay Allsopp, and David Guttman

* Una guía completa que cubre Angular y sus tecnologías complementarias, proporcionando ejemplos prácticos y detallados para construir aplicaciones completas.
* [Listado e información del curso Scrum Master en USA (agile611.com)](https://www.agile611.com/curso-scrum-master-en-usa/?v=3fd6b696867d)
* [Vercel - Opiniones, precios y características - Capterra Ecuador 2024](https://www.capterra.ec/software/203626/vercel)
* <https://docs.railway.app/>
* <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

1. Artículo

Desarrollo de Sistema Web para gestionar Cajas de Ahorro

Christian Giovany Ipiales Moreta

Universidad Internacional de la Rioja, Logroño (España)

03 de Julio de 2025

Palabras Clave

Cajas de Ahorro,

Transacciones,

Ciclos

Resumen

El objetivo de este proyecto es contar con un sistema informático que permita implementar todos los servicios de cada una de las cajas de ahorros existentes en la ciudad de Ambato tales como es el ahorro, el ahorro en los fondos de reserva, los créditos, siendo los más beneficiados los socios que conforman cada una de las cajas de ahorros de la ciudad.

El aprendizaje de nuevas tecnologías y conforme se van desarrollando nuevas herramientas para la implantación y uso del conocimiento que se adquiere en la maestría que se está cursando hacen que tengamos la capacidad para dar soluciones reales ayudando a si a un conglomerado de personas a crecer económicamente y socialmente, nosotros como estudiantes ponemos en práctica lo aprendido y obtenemos mucha más experiencia para el desenvolvimiento en el campo profesional.

I. Introducción

I

intoducción

El presente proyecto de tesis tiene la finalidad de solucionar un problema en la ciudad de Ambato el mismo que ayudará a automatizar el trabajo que se realiza en cada una de las cajas de ahorro existentes en la ciudad, permitiendo realizar los procesos de forma automática.

Para dar solución a esta problemática, conjuntamente con los socios de las cajas de ahorros se ve la necesidad de realizar una aplicación web.

Las cajas de ahorro nacen con la única idea de ser una ayuda para la ciudad, debido a que las personas de los sectores más pobres tienen un acceso restringido para la obtención de un crédito en la banca tradicional ya que se les considera sin capacidad de pago, hoy en día obtener un crédito en una institución financiera se ha convertido en algo muy necesario.

Para las personas de la ciudad, que necesitan ahorrar y obtener créditos de forma rápida y oportuna, esta modalidad de ahorro y crédito se ha convertido en algo muy necesario en el diario vivir.

Las cajas de ahorro son asociaciones creadas con un determinado número de socios los mismos que obligatoriamente deben asistir cada semana del año a reunirse en un lugar determinado para depositar el ahorro correspondiente a la semana según el valor que se fija semanalmente, esto es de acuerdo al número de acciones que se adquiere a principio de cada ciclo

II. Estado del Arte

La ciudad de Ambato desde varios años ha venido realizando esta modalidad de ahorro, se lo ha venido realizando de una manera no automática, con lo cual se hace tedioso al momento de realizar las transacciones en las cuentas de cada uno de los socios, se ha venido manejando de forma manual realizando cálculos con herramientas obsoletas, con lo cual existe un alto riesgo de equivocarse por lo que se maneja muchos datos en cajas de ahorro.

El problema es que la ciudad no cuenta o no posee una herramienta computarizada que gestione automáticamente los procesos que se realizan en las cajas de ahorro.

Con la implementación del sistema de caja de ahorro comunitario se alcanzó los objetivos esperados debido a que el sistema cumple los requerimientos especificados en las transacciones generadas por los socios como son el ahorro en sus cuentas, el ahorro en sus fondos de reserva, la obtención de créditos etc.

El sistema de caja de ahorro comunitario está desarrollado de forma genérica, para poder ser implantado en cualquier caja de ahorro de la ciudad de Ambato ya que los procesos que se pudo investigar en otras ciudades son similares a los implantados en el sistema.

La utilización e investigación de herramientas de software libre hace que se pueda aplicar las tecnologías más actuales sin restricción alguna para la elaboración de proyectos que beneficien a un conglomerado de personas.

Se utilizó y se puso en práctica los conocimientos que se adquiere en el Máster para dar solución a la problemática de no contar en tiempos actuales con un sistema que automatice los procesos de las cajas de ahorros.

La investigación de nuevas tecnologías hace que se pueda comparar herramientas que ya hemos utilizado anteriormente, teniendo la posibilidad de estar en capacidad de hacer comparaciones entre cuales son más aplicables en proyectos como el que elaboramos.

La implementación del sistema informático se pudo apreciar una reducción considerable del tiempo que se emplea en los servicios que presta la caja de ahorros en la atención a cada uno de sus socios.

El funcionamiento de las cajas de ahorro en Ecuador, conocidas como cooperativas de ahorro y crédito, está regulado por una serie de leyes y normas que aseguran su correcta operación y protección de los intereses de sus miembros. Aquí se detallan las principales leyes y regulaciones que rigen estas entidades:

III. Objetivos y Metodología

**Objetivo general**

Automatizar la gestión de cajas de ahorro que se realizan en las cajas de ahorros de la ciudad de Ambato para que el trabajo que se realiza sea eficaz y efectivo para la prestación de servicios de los socios de la caja de ahorros.

**Objetivos Específicos**

Realizar una adecuada ordenación de los procesos que se realizan en las cajas de ahorros.

Realizar un estudio de los requerimientos necesarios para que la aplicación web funcione correctamente de acuerdo a la caja de ahorros.

Reducir los tiempos y costos que implica el proceso de solicitud de crédito de los socios.

Caracterizar el funcionamiento de las cajas de ahorro en la ciudad de Ambato.

Implementar una aplicación web para la gestión de cajas de ahorro.

Validar la aplicabilidad de la propuesta de solución para la gestión de las cajas de ahorro en la ciudad de Ambato.

IV. Contribución

Desarrollar la descripción de tu contribución.

V. Resultados

VI. Discusión

VII. Conclusiones

El sistema desarrollado para la gestión de cajas de ahorro con Angular y Spring Boot ofrece una solución completa, segura y eficiente para la administración de cuentas de ahorro. La utilización de la metodología Scrum asegura un desarrollo ágil y adaptable, permitiendo entregar valor continuo al cliente y mejorando continuamente el producto

Con la implementación del sistema de caja de ahorro se alcanzó los objetivos esperados debido a que el sistema cumple los requerimientos especificados en las transacciones generadas por los socios como son el ahorro en sus cuentas, el ahorro en sus fondos de reserva, la obtención de créditos etc.

El sistema de caja de ahorro está desarrollado de forma genérica, para poder ser implantado en cualquier caja de ahorro ya que los procesos que se pudo investigar en otras ciudades son similares a los implantados en el sistema.

La implementación del sistema informático se pudo apreciar una reducción considerable del tiempo que se emplea en los servicios que presta la caja de ahorros en la atención a cada uno de sus socios.

Apéndice

Referencia