



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E

INDUSTRIAL

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

Tema:

**APLICACIÓN WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA
COMUNIDAD ANGAHUANA ALTO DE LA PARROQUIA SANTA ROSA.**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Segundo Carlos Iza Poaquiza

TUTOR: Ing. Félix Oscar Fernández Peña, PhD.

Ambato – Ecuador

septiembre - 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: APLICACIÓN WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA COMUNIDAD ANGAHUANA ALTO DE LA PARROQUIA SANTA ROSA, desarrollado bajo la modalidad Proyecto de Investigación por el señor Segundo Carlos Iza Poaquiza, estudiante de la Carrera de Ingeniería en Sistemas Computaciones e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022

.....

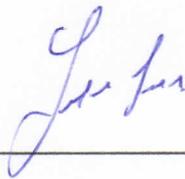
Ing. Félix Oscar Fernández Peña, PhD.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: APLICACIÓN WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA COMUNIDAD ANGAHUANA ALTO DE LA PARROQUIA SANTA ROSA, es absolutamente autentico, original y personal. En tal virtud el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022



Segundo Carlos Iza Poaquiza

C.C. 1805280375

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Segundo Carlos Iza Poaquiza, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA COMUNIDAD ANGAHUANA ALTO DE LA PARROQUIA SANTA ROSA**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, septiembre 2022

.....

Ing. Carlos Sanchez Rosero

PRESIDENTE SUBROGANTE DEL TRIBUNAL

.....

Ing. Marco Guachimboza
PROFESOR CALIFICADOR

.....

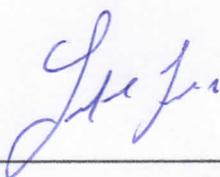
Ing. Ruben Nogales
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022



Segundo Carlos Iza Poaquiza

C.C. 1805280375

AUTOR

DEDICATORIA

El presente trabajo investigativo lo dedico a Dios y a mi madre María Elevación Poaquiza Pomaquiza por ser la mujer que me dio la vida, mi mejor amiga, mi mayor apoyo incondicional y mi gran guía.

Gracias por el apoyo incondicional en cada momento de mi vida y ser la mujer comprensiva y paciente, gracias por el amor que me das todos los días, siempre estaré agradecido por darme la vida maravillosa que tengo junto a ti y mis hermanos.

Segundo Carlos Iza Poaquiza

AGRADECIMIENTO

Agradezco en primer lugar a Dios por guiarme y dar la capacidad intelectual que me permite aprender día con día.

A mi madre por ser el guía y el apoyo incondicional para poder realizar mi carrera profesional.

A la Universidad Técnica de Ambato, por ser mi segunda casa donde pude aprender y desarrollar mis conocimientos.

A mi tutor Ing. Feliz Fernández, por ser el guía en la elaboración de este trabajo, y compartir sus conocimientos en toda mi formación profesional.

Segundo Carlos Iza Poaquiza

ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
APROBACIÓN TRIBUNAL DE GRADO	iv
DERECHOS DE AUTOR	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	x
INDICE DE FIGURAS	xiii
RESUMEN EJECUTIVO	xvi
ABSTRACT	xvii
CAPITULO I. MARCO TEORICO	1
1.1. Tema de investigación.....	1
1.2. Antecedentes Investigativos.....	1
1.3. Contextualización del Problema.....	2
1.4. Fundamentación Teórica.....	3
1.5. Tecnologías de Desarrollo Web.....	7
1.5.1. Tecnologías del frond-End.....	7
1.5.2 Tecnologías del Back-End.....	10
1.6. Estudio comparativo entre tecnologías del frond-end y back-end.....	13
1.6.1. Análisis de framework de desarrollo del frond-end.....	13
1.6.2 Análisis de framework de desarrollo del back-end.....	15
1.7. Sistemas Gestores de Bases de Datos Relacionales y no relacionales.....	17
1.7.1. Sistemas Gestores de Bases de Datos Relacionales.....	17
1.7.2. Base de dato no relacional.....	18
1.8. Estudio comparativo entre la base de datos.....	20
1.9. Metodologías de Desarrollo de Software.....	21
1.9.1 Clasificación de las metodologías.....	22
1.9.2. Comparación de metodologías.....	24
1.9.3. Metodologías tradicionales.....	24
1.9.4. Estudio comparativo entre las metodologías agiles.....	29
1.10. Objetivos.....	31

1.10.1. Objetivo general	31
1.10.2 Objetivos específicos.....	31
CAPITULO II. METODOLOGÍA	32
2.1. Materiales.....	32
2.2 Métodos	34
2.2.1 Modalidad de Investigación	34
2.2.2 Población y Muestra	34
2.2.3 Recolección de Información.....	35
2.2.4 Procesamiento y análisis de datos	36
CAPITULO III. RESULTADOS Y DISCUSION	37
3.1 Análisis y discusión de resultados	37
3.2 Desarrollo de la propuesta	40
3.3 Aplicación de la metodología	40
3.3.1 Arquitectura del sistema web	40
3.3.2 Fase 1: Exploración	41
3.3.3 Fase 2: Planificación del proyecto.....	47
3.3.4 Fase 3: Iteraciones	59
3.3.5 Fase 4: Producción	73
3.3.6 Fase 5: Mantenimiento	81
CAPITULO IV. CONCLUSIONES Y RECOMENDACIONES	81
4.1 Conclusiones.....	81
4.2 Recomendaciones	82
BIBLIOGRAFIA.....	83
ANEXOS	88

ÍNDICE DE TABLAS

Tabla 1. Estudio comparativo entre tecnologías del front-end.....	14
Tabla 2. Estudio comparativo entre tecnologías del back-end.....	16
Tabla 3. Estudio comparativo entre las bases de datos relacional y no relacional.....	21
Tabla 4. Diferencia entre la metodología tradicional y ágil.....	24
Tabla 5. Estudio comparativo entre la metodología ágiles.....	30
Tabla 6. Materiales.....	32
Tabla 7. Entrevista.....	33
Tabla 8. Directivos de la comunidad Angahuana Alto.....	35
Tabla 9. Recolección de información.....	35
Tabla 10. Resultado de la entrevista.....	39
Tabla 11. Historia de usuario-Base de datos.....	41
Tabla 12. Historia de usuario-Página de inicio.....	42
Tabla 13. Historia de usuario-Página de inicio dentro del sistema.....	42
Tabla 14. Historia de usuario-Formulario de ingreso.....	43
Tabla 15. Historia de usuario-Página de inicio dentro del sistema.....	43
Tabla 16. Historia de usuario-Formulario para listar los socios.....	43
Tabla 17. Historia de usuario-Formulario de ingreso del socio.....	44
Tabla 18. Historia de usuario-Formulario la edición del socio.....	44
Tabla 19. Historia de usuario-Gestión de usuario.....	45
Tabla 20. Historia de usuario-Formulario de registro de reunión.....	45
Tabla 21. Historia de usuario-Formulario de registro de asistencia.....	45
Tabla 22. Historia de usuario-Formulario de cobro.....	46
Tabla 23. Historia de usuario-Formulario de cobro.....	46
Tabla 24. Historia de usuario-Formulario de registro de documentos.....	47
Tabla 25. Historia de usuario-Formulario de registro de documentos.....	47
Tabla 26. Historia de usuario-Formulario de registro de documentos.....	48
Tabla 27. Plan de entregas.....	48
Tabla 28. Equipo y roles.....	49
Tabla 29. Tarea - Modelado de la base de datos.....	49
Tabla 30. Tarea – Plantilla para la página de inicio.....	49
Tabla 31. Tarea – Diseño de las secciones de la página de inicio.....	50
Tabla 32. Tarea – Funcionalidad del menú de navegación en la página de inicio.....	50
Tabla 33. Tarea – Diseño del formulario para la administración del banner principal de la página de inicio.....	50

Tabla 34. Tarea – Funcionalidad para el formulario de la administración del banner de la página de inicio.....	51
Tabla 35. Tarea – Diseño del formulario para la administración del banner principal de la página de inicio.	51
Tabla 36. Tarea – Funcionalidad para el formulario de la administración del banner de la página de inicio.....	51
Tabla 37. Tarea – Diseño del formulario para la administración del banner principal de la página de inicio.	52
Tabla 38. Tarea – Funcionalidad para el formulario de la administración del banner de la página de inicio.....	52
Tabla 39. Tarea - Modelado de la base de datos.	52
Tabla 40. Tarea – Funcionalidad en el formulario de ingreso al sistema.	53
Tabla 41. Tarea – Definición de plantilla para la página de inicio dentro del sistema.	53
Tabla 42. Tarea – Menú de navegación acorde a los perfiles.	53
Tabla 43. Tarea – Funcionalidad del menú de navegación.	54
Tabla 44. Tarea - Modelado de la base de datos.	54
Tabla 45. Tarea – Formulario para añadir un nuevo socio.	54
Tabla 46. Tarea – Funcionalidad del formulario para añadir un nuevo socio.....	55
Tabla 47. Tarea – Formulario para la edición del socio.....	55
Tabla 48. Tarea – Funcionalidad del formulario para la edición del socio.....	55
Tabla 49. Tarea – Formulario de creación de usuario.....	56
Tabla 50. Tarea – Funcionalidad del formulario de creación de usuario.....	56
Tabla 51. Tarea – Formulario para la edición del usuario.	56
Tabla 52. Tarea – Formulario de registro de la reunión.....	56
Tabla 53. Tarea – Funcionalidad de registro de la reunión.....	57
Tabla 54. Tarea – Formulario de registro de asistencia.	57
Tabla 55. Tarea – Funcionalidad para el registro de asistencia.	57
Tabla 57. Tarea – Funcionalidad del formulario para cobro del rubro.	58
Tabla 58. Tarea – Formulario para el ingreso de un rubro.....	58
Tabla 59. Tarea – Funcionalidad para el ingreso de un nuevo rubro.	58
Tabla 60. Tarea – Formulario para el ingreso de documento.	58
Tabla 61. Tarea – Funcionalidad para el registro del documento.	59
Tabla 62. Tarea – Funcionalidad para el registro de asistencia.	59
Tabla 63. Tarea – Presentación de informes.	59
Tabla 65. Tarea – Historias de usuario iteración 2.....	65
Tabla 66. Tarea – Historias de usuario iteración 3.....	67
Tabla 67. Tarea – Historias de usuario iteración 4.....	68

Tabla 68. Tarea – Historias de usuario iteración 5.....	72
Tabla 69. Prueba de aceptación – Página de inicio administrativo.....	73
Tabla 70. Prueba de aceptación – Administración del banner de la página de inicio.....	74
Tabla 71. Prueba de aceptación – Administración de la página de inicio.....	74
Tabla 72. Prueba de aceptación – Administración de la página de inicio.....	75
Tabla 73. Prueba de aceptación – Administración de la página de inicio.....	75
Tabla 74. Prueba de aceptación – Ingreso al sistema web.	75
Tabla 75. Prueba de aceptación – Página de inicio dentro del sistema.....	76
Tabla 76. Prueba de aceptación – Lista de socios.....	76
Tabla 77. Prueba de aceptación – Creación de usuario.....	77
Tabla 78. Prueba de aceptación – Editar la información del socio.	77
Tabla 79. Prueba de aceptación – Registro de un nuevo usuario.....	77
Tabla 80. Prueba de aceptación – Actualización del usuario.....	78
Tabla 81. Prueba de aceptación – Inactivar un usuario.....	78
Tabla 82. Prueba de aceptación – Registro de una nueva reunión.....	78
Tabla 83. Prueba de aceptación – Registro de asistencia.....	79
Tabla 84. Prueba de aceptación – Cobro de rubros.....	79
Tabla 85. Prueba de aceptación – Registro de un nuevo rubro.....	80
Tabla 86. Prueba de aceptación – Registro de documentos.....	80
Tabla 87. Prueba de aceptación – Generación de informe.....	80

INDICE DE FIGURAS

Figura 1. Arquitectura de AngularJS	8
Figura 2. Diagrama del motor de plantilla Jade [24].	10
Figura 3. Arquitectura de ASP .NET Framework.....	11
Figura 4. Arquitectura de NodeJS	12
Figura 5. Esquema de la metodología Scrum [46].	26
Figura 6. Esquema de la metodología Extreme programing.....	29
Figura 7. Diagrama de proceso del sistema.	40
Figura 8. Arquitectura de sistema web para la automatización de procesos.....	41
Figura 9. Modelo relacional de la base de datos.	60
Figura 10. Clase principal del proyecto	61
Figura 11. Conexión a la base de datos.....	61
Figura 12. Página de inicio del sistema administrativo.....	62
Figura 13. Formulario administración de las categorías de la página.	62
Figura 14. Formulario del contenido principal de la página de bienvenida.	63
Figura 15. Formulario de administración del banner principal.....	63
Figura 16. Formulario de ingreso al sistema.....	64
Figura 17. Librería de hashing	64
Figura 18. Página de inicio dentro del sistema web.....	65
Figura 19. Formulario para listar socios.	66
Figura 20. Formulario para crear un nuevo socio.	66
Figura 21. Formulario para la actualización del socio.	67
Figura 22. Formulario para la actualización del socio.	68
Figura 23. Formulario para la actualización del socio.	68
Figura 24. Formulario de registro de reunión.	69
Figura 25. Formulario para el registro de asistencia.	69
Figura 26. Listado de reuniones.	70
Figura 27. Formulario de socios por cobrar.	70
Figura 28. Formulario para el cobro del rubro por socio.	71
Figura 29. Recibo de cobro.	71
Figura 30. Formulario para ingresar un nuevo rubro.	72
Figura 31. Formulario para guardar documentos.	72
Figura 32. Informe del listado de socios.	73
Figura 33. Diseño de la cabecera de la página de inicio.	88
Figura 34. Carga del banner y el contenido principal.	88

Figura 35. Función principal para la carga de la página de inicio.	89
Figura 36. Diseño para el formulario del contenido principal en HTLM.	89
Figura 37. Función principal para la administración de la categoría	90
Figura 38. Diseño de para la administración del contenido principal.....	90
Figura 39. Función para el guardado de contenido principal.....	91
Figura 40. Función para dar de baja el contenido principal.....	91
Figura 41. Diseño del formulario del banner principal.....	92
Figura 42. Función para subir el banner principal.	92
Figura 43. Función para dar de baja un banner.....	93
Figura 44. Método validación de datos	93
Figura 45. Diseño del menú.	94
Figura 46. Modelo del usuario.	94
Figura 47. Función de control de roles.....	95
Figura 48. Función para el resumen del dashboard.....	95
Figura 49. Diseño del formulario para listar socios.	96
Figura 50. Modelo del esquema socio.....	96
Figura 51. Función para la obtención de los socios.	97
Figura 52. Modelo del esquema socio.....	97
Figura 53. Función para el guardado de socio	98
Figura 54. Modelo del esquema socio.....	98
Figura 55. Función para obtener un socio específico.....	99
Figura 56. Función para actualizar el socio.	99
Figura 57. Modelo de esquema usuario	100
Figura 58. Función para la obtención de todos los usuarios.	100
Figura 59. Función para guardar un nuevo usuario.....	101
Figura 60. Función para editar un usuario	101
Figura 61. Función para editar el usuario.....	101
Figura 62. Modelo para el registro de asistencia.....	102
Figura 63. Función para la obtención de las asistencias de los socios	102
Figura 64. Función para guardar la asistencia.....	103
Figura 65. Función para obtener el detalle de rubros del cliente.	103
Figura 66. Función para guardar la factura.	104
Figura 67. Función para detallar los rubros	104
Figura 68. Función para obtener la factura	105
Figura 69. Modelo del nuevo rubro.....	105
Figura 70. Función para obtener los rubros.	106

Figura 71. Función para guardar el rubro.....	106
Figura 72. Función para la obtención del documento.....	107

RESUMEN EJECUTIVO

Las Tecnologías de la Información y la Comunicación (TIC) han transformado la manera de gestionar los recursos de las organizaciones. Debido a esto, la implementación de herramientas tecnológicas ha tomado mucha importancia en la actualidad, llegando a promover la utilización de los sistemas informáticos en las diferentes instituciones donde realizan procesos que involucran el manejo de la información y la documentación. De acuerdo a ello, el presente trabajo tuvo como objetivo implementar una aplicación web para la gestión administrativa en la comunidad Angahuana Alto de la parroquia Santa Rosa. Para el desarrollo de la aplicación se utilizó una arquitectura MVC (Modelo Vista Controlador) y la metodología XP, la cual permitió presentar pruebas unitarias, con el fin de prevenir y corregir los errores, también obtener productos tangibles en periodos cortos. Las herramientas aplicadas en el desarrollo permitieron crear una aplicación web responsiva y dinámica para el usuario, gracias a ello pueden utilizar en cualquier dispositivo llegando a satisfacer la experiencia del usuario. Finalmente se concluyó que el sistema web para la automatización de los procesos permite que los directivos y los socios lleven de una mejor manera las actividades que realiza la comunidad con las métricas generadas, tanto al inicio del mes y fin de año.

Palabras clave: Sistema web, herramientas tecnológicas, MongoDB, XP, administración.

ABSTRACT

Information and Communication Technologies (ICT) have transformed the way of managing the resources of organizations, due to this, the implementation of technological tools has become very important today, even promoting the use of computer systems in the different institutions where they carry out processes that involve the handling of information and documentation. Accordingly, this work aimed to implement a web application for administrative management in the Angahuana Alto community of the Santa Rosa parish. For the development of the application, an MVC (Model View Controller) architecture and the XP methodology were obtained, which demonstrated unit tests, in order to prevent and correct errors, as well as obtain tangible products in short periods. The tools applied in the development allowed the creation of a responsive and dynamic web application for the user, thanks to which they can be used on any device, satisfying the user experience. Finally, it was concluded that the web system for the automation of processes allows managers and partners to better carry out the activities carried out by the community with the metrics generated, both at the beginning of the month and at the end of the year.

Keywords: Web system, technological tools, MongoDB, XP, administration.

CAPITULO I. MARCO TEORICO

1.1. Tema de investigación

APLICACIÓN WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA COMUNIDAD ANGAHUANA ALTO DE LA PARROQUIA SANTA ROSA.

1.2. Antecedentes Investigativos

Revisando la información existente en diversos repositorios, incluyendo la biblioteca de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, se han encontrado trabajos que sirvieron de apoyo en el trabajo de investigación.

El trabajo investigativo desarrollado por Bermeo [1] con el tema: “Sistema web para el control de procesos administrativos en la Escuela Particular de Educación Básica Triunfadores del Futuro” tuvo como objetivo implementar un sistema web para el control de los procesos administrativos de la institución educativa Triunfadores del Futuro. Se integraron los respectivos módulos y se llevaron a cabo las pruebas en las cuales se obtuvieron resultados satisfactorios, logrando el adecuado control de registro y asistencia del personal docente, de los administrativos, alumnos cobros de pensiones, entre otros procesos, por lo que finalmente se obtuvo un sistema web que permitió automatizar los procesos de la institución.

La investigación desarrollada por Zurita [2] con el tema: “Sistema web para la gestión académica y administrativa de la empresa de capacitación profesional DIENAV” tuvo como finalidad implementar un sistema web para la empresa de capacitaciones profesionales DIENAV, para brindar un servicio automático de procesos a través de la web, utilizando la metodología ágil para el desarrollo del software que en este caso fue la XP (Extreming Programming). Para el diseño web de páginas dinámicas se empleó PHP y como gestor de base de datos MySQL. En conclusión, el proyecto facilitó significativamente los procesos de asignación de

usuarios, matrículas, cursos o convocatorias y emisión de reportes con calidad y eficiencia en la prestación de servicios educativos hacia la comunidad en general.

El trabajo investigativo desarrollado por Hurtado [3] con el tema: “Aplicación web administrativa para reserva de servicios de transporte y envío de encomiendas para la empresa Romero y Asociados (AMBASEUR) de la ciudad de Ambato” enmarcó su objetivo en implementar una aplicación web que gestione de manera automática los procesos actuales que son realizados de forma manual. La metodología empleada fue la XP (Extreme Programming), ya que permitió integrar los nuevos requerimientos suscitados durante el desarrollo, de manera ordenada y fácil. El sistema web permitió reservar servicios de forma ordenada y ágil, así también realizar el seguimiento de las encomiendas de sus clientes.

La investigación desarrollada por Curichumbi [4] con el tema: “Sistema automatizado para la gestión de las comunidades de la parroquia Salasaca de la ciudad de Pelileo” tuvo como objetivo implementar un sistema automatizado para la gestión de las comunidades de la parroquia Salasaca organizando y controlado las reuniones, las asistencias, las recaudaciones, los egresos y los documentos. Para diseñar y desarrollar el sistema web se aplicó la metodología XP. Las herramientas informáticas que se aplicaron en el desarrollo fueron el lenguaje de programación C Sharp (C #), Framework Asp.Net, IDE Visual Studio.Net y MySQL como gestor de base de datos. La aplicación web permitió que los directivos lleven un mayor control de los procesos de la comunidad, agilizando el registro y búsqueda de información.

1.3. Contextualización del Problema

Las tendencias tecnológicas aportan significativamente al rendimiento de los recursos humanos, debido a ello han desarrollado herramientas dinámicas y fácil de usar, con el fin de promover la utilización de los sistemas informáticos en las diferentes instituciones, tanto públicas como privadas; donde realizan procesos que involucran el manejo de la información y la documentación [4] [5].

En Ecuador la forma de administrar la información y la documentación varía según el lugar y región; las diferentes normas, formas y lineamientos generan un

inadecuado manejo tanto en la institución pública como privadas [5]. La necesidad de manejar grandes cantidades de información permitió a las instituciones implementar sistemas para la gestión de información y documentación. En la actualidad, las empresas han optado en automatizar los procesos y llevar la documentación en forma digital y trasladarlos a la nube (Cloud), con el fin de tener acceso desde cualquier lugar que lo requiera [6].

En Tungurahua el uso de la tecnología ha ido creciendo notablemente, gracias a ello muchas instituciones públicas como privadas han optado por automatizar los procesos ya que la rapidez y la confiabilidad de un sistema ha llevado a tener mayor acogida y reconocimiento en el mercado, estas empresas públicas como privadas han preferido adquirir aplicaciones para facilitar y reducir los tiempos en la gestión de la información, algunas instituciones ya constan con sistemas informáticos de facturación e inventarios, con ello han agilizado el servicio y disminuido el tiempo de atención a sus clientes [7].

En la actualidad la comunidad Angahuana Alto no cuenta con una adecuada gestión en el manejo de los procesos, como reuniones, recaudaciones, registro de asistencia y el manejo de la documentación, provocando así, el malestar en los socios y dificultando el manejo para los directivos.

La información de la comunidad es registrada manualmente, así como también las recaudaciones, debido a esto los problemas administrativos al momento de justificar los ingresos y egresos han crecido, llegando a causar malestar en los socios y no poder informar el estado financiero. El registro de asistencia se lo llevado de una manera manual e inadecuada, ya que los registros se almacenan en documentos físico, el cual al momento de acceder provoca retraso y malestar y con ello pérdida de información y duplicidad.

1.4. Fundamentación Teórica

1.4.1. Servicio al cliente.

El servicio al cliente es una actividad que se lleva a cabo mediante la interacción entre el cliente y el empleado y que tiene como finalidad satisfacer las dudas o necesidades del consumidor, dicho servicio puede tener grandes ventajas si se realiza

de manera adecuada. Aplicar un adecuado servicio al cliente es fundamental para el éxito de las empresas, debido a que los clientes tienen diferentes opciones de compra en el mercado, por lo cual, si no están de acuerdo en el trato de una empresa pueden buscar otra posibilidad [8].

Una de las cosas que debe considerar cualquier organización o entidad empresarial es una mejora del servicio que impactará en el trabajo de mejora empresarial, ya que, con una comunicación efectiva, la empresa es capaz de mantener buenas relaciones y armonía con los consumidores. De esta manera la atención al cliente es una de las actividades importantes porque involucra el rumbo futuro de la empresa, por lo tanto, el papel del servicio de atención al cliente es fidelizar a los clientes proporcionando un servicio de calidad, esto puede ser la base para adquirir nuevos clientes, cuidarlos, mantenerlos y desarrollarlos [9].

1.4.2. Desarrollo de software.

El proceso de desarrollo de software es un conjunto estructurado de las actividades que se requieren para la construcción de un sistema. Dicho proceso se utiliza con la finalidad de mejorar la comprensión del problema que se tiene que resolver, la comunicación entre los actores del proyecto y el mantenimiento de un sistema complejo. Las principales actividades del proceso de desarrollo de software son: especificación de requerimientos, diseño, codificación, validación y mantenimiento [10].

Para el desarrollo de un producto de software se llevan a cabo una serie de tareas entre la idea inicial con la especificación del sistema y el producto final, en el cual se incluye también el mantenimiento del éste después de utilizarlo. Actualmente la mayoría de los métodos de desarrollo de software emplean modelos, pero lo que varía de un método a otro es la clase de modelos que deben construirse, la forma en que se los representa y manipula [11].

Al incorporar metodologías de desarrollo de software garantizan diseñar software de alta calidad que cumplan con los requerimientos, planificación y presupuesto establecido, puesto que involucra un análisis de riesgo, cubre todo el ciclo de vida del producto, soporta un enfoque de desarrollo iterativo e incremental, además

proporciona iteraciones tempranas que se centran en validar y producir una arquitectura de software, y un ciclo de desarrollo inicial que le permite tomar la forma de un prototipo ejecutable que va evolucionando de manera gradual hasta convertirse en el producto final y además implica un proceso de evaluación continua de la calidad [12].

1.4.3. Gestión de documentos

La gestión de documentos es un método de gestión empresarial que se orienta a aprovechar los documentos y la información contenida en ellos por parte de las organizaciones, ya sean públicas o privadas, desde que estos se producen o reciben en la organización, hasta su eliminación final o conservación permanente en un archivo histórico [13].

De igual manera la gestión documental hace referencia al conjunto de normas técnicas y prácticas para la administración del flujo de documentos de cualquier tipo en una empresa. Dicha gestión facilita la recuperación de información, determina el tiempo que los documentos deben archivarse y conservarse, evita los documentos no esenciales u obsoletos y asegura la conservación de los documentos más valiosos.

En definitiva, la gestión documental permite abarcar toda la información de un documento o grupo de documentos de manera organizada con el fin de que la búsqueda sea más fácil, rápida, efectiva y aligere los procesos de una empresa según sus necesidades, brindando mayor seguridad a la información por medio de la definición de reglas para el acceso de los usuarios. La gestión documental permite la reducción de costos, espacio físico y colabora con el medio ambiente, debido a que disminuye considerablemente el uso y gasto de papel [14].

De esta manera la implementación de un sistema de gestión de documentos para el control y manipulación eficaz de documentos digitales brinda estabilidad en el registro, consulta, actualización, búsqueda y conservación de evidencias, que respaldan las actividades que se efectúan en las áreas funcionales de las organizaciones [15].

1.4.4. Gestión Administrativa

La gestión administrativa es la manera más eficiente y económica de utilizar los recursos, trata de escoger las acciones en función de los resultados que se esperan obtener y de los medios que se dispone. La gestión administrativa se refiere a la puesta en práctica de cada uno de los procesos de la administración, estos son: la planificación, organización, dirección, coordinación, y el control de actividades de la organización; es decir, hace referencia a la toma de decisiones y acciones oportunas para dar cumplimiento de los objetivos preestablecidos de la empresa [16].

La gestión administrativa es el conjunto de acciones y mecanismos que permite utilizar los recursos humanos, materiales y financieros de una organización, que siempre tendrán las características de ser escasos, por lo que las decisiones acertadas provocan en las empresas un impacto positivo y por el contrario al no ser lo suficientemente adecuada y en el tiempo oportuno su impacto será negativo [17].

1.4.5. Gestión de calidad.

La gestión de calidad hace referencia al conjunto de normas interrelacionadas de una empresa u organización por los cuales se administra de manera ordenada la calidad de la misma, con el fin de lograr la satisfacción de las necesidades y expectativas de sus clientes [18]. De este modo la gestión de la calidad es un medio que permite a las organizaciones mantenerse ancladas en el mercado global, por medio de la búsqueda y mejora continua en la calidad del servicio que brinda a los usuarios [19].

Para la aplicación de una gestión de calidad se considera los siguientes elementos [18]:

- **Estructura de la organización:** se refiere al organigrama de los sistemas de la empresa donde se jerarquiza los niveles directivos y de gestión.
- **Estructura de responsabilidades:** involucra a las personas y departamentos.
- **Procedimientos:** Hace referencia al plan permanente de pautas detalladas para el control de las acciones de la organización.

- **Procesos:** Se refiere a la sucesión completa de las operaciones que se enfocan en conseguir los objetivos.
- **Recursos:** Se refiere a los recursos económicos, humanos y técnicos, los cuales deben estar definidos de manera estable y circunstancial.

1.5. Tecnologías de Desarrollo Web.

Un framework es aquel que agrega funcionalidad extendida a un lenguaje de programación y tiene como objetivo automatizar muchos de los patrones de programación para inclinarlos a un determinado propósito, facilitando una estructura al código, mejorándolo y haciéndolo más entendible y sostenible, y permite separar en capas la aplicación [20].

La aplicación se divide en tres capas que son [20]:

- La lógica de presentación, la que se basa en la administración de las interacciones entre el usuario y el software.
- La Lógica de datos, la cual permite el acceso a un agente de almacenamiento persistente u otros.
- La lógica de dominio o de negocio, que desarrolla los modelos de datos a través de los comandos recibidos mediante la presentación.

1.5.1. Tecnologías del front-End.

Angular

Es un framework utilizado para el desarrollo web basado en TypeScript, de código abierto, utilizado para crear y mantener aplicaciones web, es mantenido por Google, que ayuda a la gestión de lo que se llama aplicaciones de una página (SPA) y extiende el tradicional HTML con etiquetas propias. Es una solución completa, ya que dispone de todas las herramientas para llevar a cabo el desarrollo de una aplicación web, sin tener que preocuparse en la manera de implementar las funcionalidades [21] [22].

Este framework no está ligado a un solo tipo de arquitectura es beneficioso, debido a que es de libre elección de la persona o equipo que va a desarrollar la aplicación. Las aplicaciones angulares se componen de NgModules, que son contenedores para

un bloque de código, esta es una plataforma construida sobre el runtime de JavaScript de Google Chrome diseñada para construir aplicaciones en la red de manera rápida y escalable [23].

Las características y ventajas de Angular son [22]:

- Es ligero y tiene buena gestión de dependencias.
- Posee un potente sistema de plantillas, y extiende vocabulario de HTML básico.
- El concepto de directivas, este permite crear tags customizados que tienen ligado la funcionalidad como capa visual.
- Tiene un potente enlace entre las interfaces (UI-Binding).
- Tiene un desenganche del DOM de Java Script.
- Tiene buenas herramientas que pueden hacer debug.

Arquitectura

Angular maneja una arquitectura básica modelo vista controlador, tal como se muestra en la figura 1, la que dispone de diferentes componentes para la vista, enrutadores para la capa de control y servicios para el backend [21] .

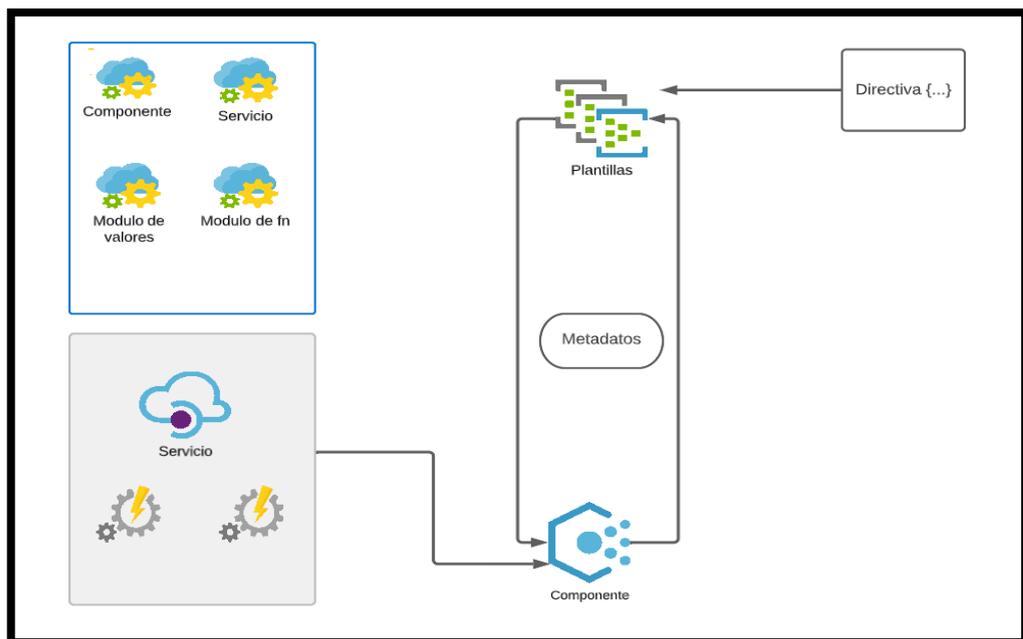


Figura 1. Arquitectura de AngularJS

Fuente: Elaborado por el autor.

Jade

Es un lenguaje nuevo y simplificado que compila en HTML y es extremadamente útil para los desarrolladores web. Jade está diseñado principalmente para plantillas del lado del servidor en Node.js, pero también se puede usar en una variedad de otros entornos para producir documentos similares a XML, como HTML y RSS. Además de las mejoras sintácticas, le permite simplificar marcado redundante con lógica programada. También, te permite crear plantillas que pueden tomar y mostrar datos [24].

Algunas características de Jade son [24]:

- **Función reutilizable:** Esta puede compilar su plantilla en la función reutilizable, la que se puede ejecutar en el servidor o en el lado del cliente.
- **Herencia de plantilla:** La gestión de plantillas es aquella que se asemejan a las técnicas de programación orientada a objetos, por lo tanto, se puede heredar el contenido de una plantilla padre a las plantillas nuevas o hijas mediante la palabra `extends`.
- **Productividad de la plantilla:** Debido a que el motor de plantillas de Jade viene con muchos módulos de manejo, el resultado final puede tener una excelente apariencia.

Diagrama

El diagrama del motor de plantilla Jade es el que se muestra a continuación:

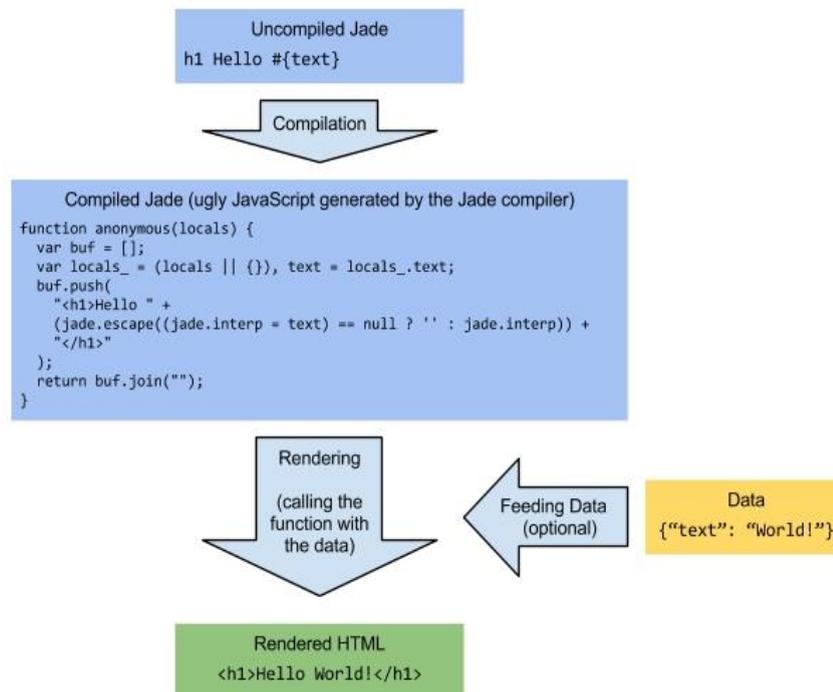


Figura 2. Diagrama del motor de plantilla Jade [24].

1.5.2 Tecnologías del Back-End.

ASP .NET.

NET es una plataforma desarrollada por Microsoft, la que permite a distintos desarrolladores crear varios tipos de aplicaciones multiplataforma y de código abierto facilitándoles un conjunto de herramientas, tecnologías y servicios. Este ofrece soporte a distintos lenguajes de programación como Visual Basic, C++, F#, C#, entre otros [25].

Para el desarrollo del back-end se ofrece las siguientes características que son [25]:

- **Multiplataforma:** Permite a los desarrolladores la implementación de aplicaciones que pueden ser ejecutadas en los sistemas operativos como Windows, Linux y MacOS.
- **Código abierto:** Proporciona una gran transferencia mediante el proceso de la aplicación.
- **Modular:** Es aquel que accede un modularidad total gracias a los paquetes NuGet.

- **Autenticación:** La que incluye bibliotecas, una base de datos y páginas de plantillas que son para manejar inicios de sesión, y está contenida la autenticación multifactorial y la autenticación externa con Google, Twitter y más.
- **Extensiones:** Asp .net permite extender de bibliotecas que ayudan al desarrollo del software, por ejemplo, el resaltando de la sintaxis, la finalización de código y distintas funciones que son particularmente para el desarrollo de páginas web.

Arquitectura

La arquitectura ASP .NET es la que se muestra a continuación:

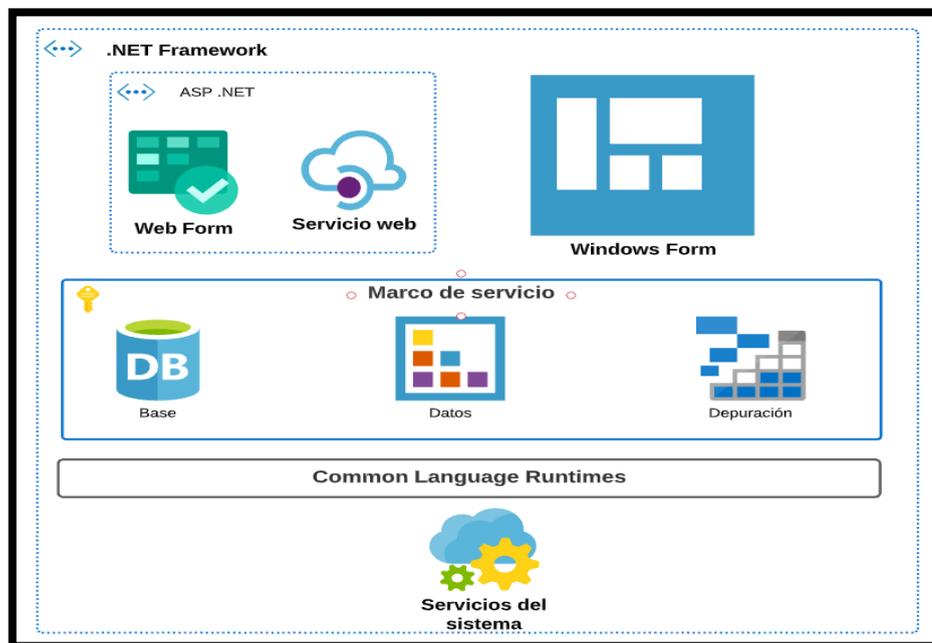


Figura 3. Arquitectura de ASP .NET Framework

Fuente: Elaborado por el autor.

NodeJS.

Node.js es una plataforma que su entorno de ejecución se basa en javascript de Chrome, con la cual se puede construir aplicaciones de red. Node.js rápidas y escalables. Este utiliza un modelo de E/S no bloqueante dirigido por eventos para hacerlo ligero y eficiente, excelente para las aplicaciones data-insensitive en tiempo real [26]. Es decir, Node.js utiliza una menor cantidad de hilos o subprocesos, y de

igual manera menos recursos del computador, lo que ayuda a la fácil y rápida ejecución de las tareas [27].

Debido al gran impacto de Node.js se describe varias características que son [28]:

- **Javascript de extremo a extremo.** – Es una de las ventajas principales de nodeJS debido a que esta le permite escribir scripts del lado del servidor y del cliente en JavaScript.
- **Escalabilidad basada en eventos:** Node.js posee una lógica única para manejar la solicitud web, ya que en lugar de tener múltiples elementos esperando para procesar la solicitud web, con el nodeJS se puede procesar en el mismo modelo de evento básico.
- **Extensibilidad:** Node.js posee una gran comunidad de seguidores activos, debido a que la gente proporciona nuevos módulos para extender la funcionalidad del nodeJS todo el tiempo.
- **Implementación rápida:** Para configurar nodeJS y desarrollar en él no requiere de gran esfuerzo y la instalación requiere solo de minutos para tener un servidor web en funcionamiento.

Arquitectura

La arquitectura de Node.js es la que se muestra a continuación:

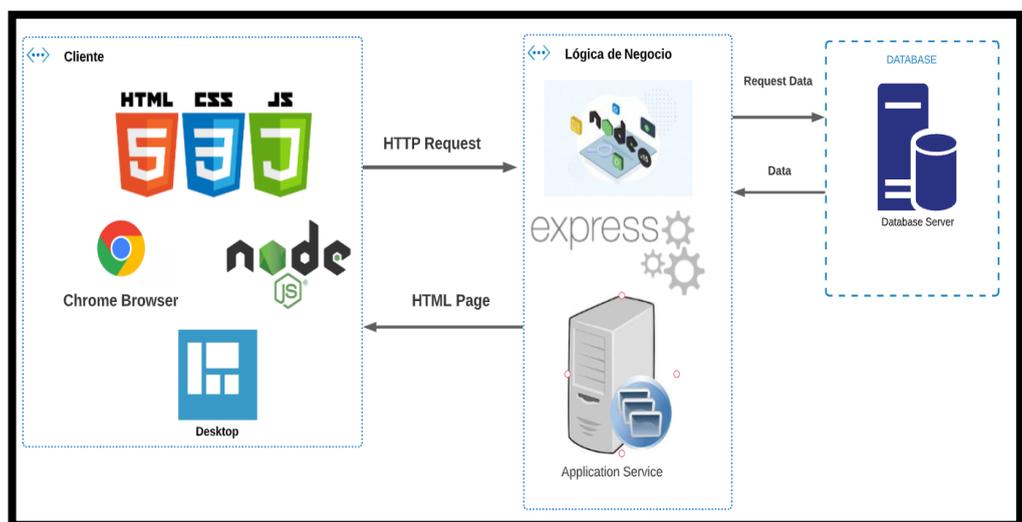


Figura 4. Arquitectura de NodeJS

Fuente: Elaborado por el autor.

1.6. Estudio comparativo entre tecnologías del front-end y back-end

1.6.1. Análisis de framework de desarrollo del front-end

Se procede a analizar 2 framework de desarrollo para el front-end, Angular y Jade, las cuales se enfocarán en sus ventajas, desventajas y sus características principales; la cual nos servirá para la respectiva toma de decisiones.

Herramienta	Característica	Ventaja	Desventaja
Angular	<ul style="list-style-type: none">• Framework de desarrollo para JavaScript• Creación de aplicaciones multiplataforma• Lenguaje de programación el TypeScript	<ul style="list-style-type: none">• El modelo de componentes de Angular ofrece una fuerte encapsulación y estructuración• Está diseñado para la velocidad en la web.• Cliente HTTP sólido.• Documentación detallada y alcance del público	<ul style="list-style-type: none">• La curva de aprendizaje pronunciada para los nuevos desarrolladores• Opciones limitadas de SEO y la poca accesibilidad para los motores de búsqueda
Jade	<ul style="list-style-type: none">• Las líneas de comando son más legibles.• El código está codificado de forma predeterminada	<ul style="list-style-type: none">• Fácil de lectura.• Evita errores.• Bloques reutilizables.• Permite utilizar	<ul style="list-style-type: none">• Nueva sintaxis.• Jade tampoco distingue entre las etiquetas y las variables.• Difícil hacer algunas cosas

	para mejorar la seguridad. <ul style="list-style-type: none"> • Informe de errores de contexto de compilación y tiempo de ejecución. • Modo HTML5. • Clase de marcador estático dinámico combinado. • Adaptabilidad con nodeJS. 	cualquier etiqueta HTML. <ul style="list-style-type: none"> • Compatible con muchas tecnologías. • Sin etiquetas de cierre. • Plantillas maestras. 	simples en Jade.
--	---	---	------------------

Tabla 1. Estudio comparativo entre tecnologías del frond-end.

Fuente: Elaborado por autor.

La tabla 1 muestra las características, ventajas y desventajas entre las tecnologías del frond-end, gracias a ello se ha optado por utilizar el motor de HTML Jade, ya que las ventajas que se describe ayudaron a encaminar a la tecnología mencionada.

Entre las ventajas se encuentran:

- Fácil de lectura.
- Evita errores.
- Bloques reutilizables.
- Permite utilizar cualquier etiqueta HTML.
- Compatible con muchas tecnologías.
- Sin etiquetas de cierre.
- Plantillas maestras.
- Compatible con el framework nodeJS
- Documentación extensa

- Comunidad y blogs

1.6.2 Análisis de framework de desarrollo del back-end

Se procede a analizar 2 frameworks de desarrollo para el back-end: .NET Core y NodeJS, los cuales se enfocarán en sus ventajas, desventajas y sus características principales; la cual nos servirá para la respectiva toma de decisiones.

Herramienta	Característica	Ventaja	Desventaja
ASP .NET	Plataforma de desarrollo compuesta por herramientas, lenguajes de programación y bibliotecas	<ul style="list-style-type: none"> • ASP.Net para organizaciones grandes y medianas. • Sistema de autenticación robustas. • Altamente estables y seguros. 	<ul style="list-style-type: none"> • Plataforma específica de desarrollo • Elevado consumo de recursos. • Elevado tiempo en desarrollo y ejecución de programa.
Nodejs	<ul style="list-style-type: none"> • JavaScript de extremo a extremo • Escalabilidad basada en eventos • Extensibilidad • Capacidad de comunicación en tiempo real. • Su capacidad para integrarse a diferentes stacks • Modelo de un solo subproceso 	<ul style="list-style-type: none"> • Permite escribir scripts del lado del servidor y del cliente en JavaScript. • Lógica única para manejar la solicitud web. • Gran comunidad de seguidores y muy activa. • Muy simple de instalar e incluir nuevos módulos en nodejs puede 	<ul style="list-style-type: none"> • Reduce el rendimiento cuando se manejan tareas informáticas pesadas. • El modelo de programación asincrónica de Node.js dificulta el mantenimiento del código.

	con bucle de eventos	extender un proyecto de nodejs para incluir nueva funcionalidad en minutos. <ul style="list-style-type: none"> • Configurar nodejs y desarrollar en él es muy fácil. 	
--	----------------------	--	--

Tabla 2. Estudio comparativo entre tecnologías del back-end.

Fuente: Elaborado por autor.

La tabla 2 muestra las características, ventajas y desventajas entre las tecnologías del back-end, gracias a ello se ha optado por el nodeJS, ya que las ventajas que se describe nos ayudaron a encaminar a la tecnología mencionada.

Entre las ventajas se encuentran:

- Permite escribir scripts del lado del servidor y del cliente en JavaScript.
- Lógica única para manejar la solicitud web.
- Gran comunidad de seguidores y muy activa.
- Muy simple de instalar e incluir nuevos módulos en nodeJS puede extender un proyecto de nodeJS para incluir nueva funcionalidad en minutos.
- Configurar nodeJS y desarrollar en él es muy fácil.
- Conexión directa con la base de dato no relacional MongoDB.
- Actualización de módulos constantemente

1.7. Sistemas Gestores de Bases de Datos Relacionales y no relacionales

1.7.1. Sistemas Gestores de Bases de Datos Relacionales

SQL Server

Es el sistema de base de base de datos profesional de Microsoft, la cual posee características y herramientas que se pueden utilizar en el desarrollo y administración de bases de datos y soluciones de todo tipo basadas en ellas. Con un entorno integrado de SQL Server Management Studio tiene como función la obtención de acceso, configuración y administración de varios componentes de SQL Server. Management Studio está diseñado con el fin de desarrollar y administrar objetos de bases de datos y para administrar y configurar los objetos existentes de Analysis Services [29].

Algunas características de Microsoft SQL Server son [30]:

- Posee soporte de transacciones.
- Tiene una gran escalabilidad, estabilidad y seguridad.
- Soporta procedimientos que son almacenados.
- Tiene un potente entorno gráfico para administrar, que admite el uso de comandos DDL y DML de manera gráfica.
- Puede trabajar en modo cliente-servidor, en la cual la información y datos se alojan en el servidor y los clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos.

Microsoft SQL Server establece la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos que pueden ser Oracle, Sybase ASE, PostgreSQL o MySQL [30].

MYSQL.

MySQL es una base de datos relacional que utiliza Lenguaje de Consulta Estructurado (SQL). Se basa en un SBD de código abierto, es popular y ampliamente usado debido a la simplicidad e importante rendimiento [31].

MySQL ofrece distintos protocolos de comunicación entre cliente y el servidor. Bajo Unix, es el protocolo para la comunicación que está dado por defecto para la comunicación locales Unix Socket. Sin embargo, se puede hacer uso al protocolo TCP/IP en local, el cual es el único disponible para un cliente remoto [32].

Las estructuras de bases de datos se pueden organizar en archivos físicos que están optimizados en la velocidad. Los modelos lógicos, que tiene objetos tales como bases de datos, tablas, vistas filas y columnas, brinda una solución flexible entorno a la programación. Se configura de acuerdo a las reglas que rigen las relaciones entre los distintos datos, campos, uno a uno, uno a muchos, único, obligatorio u opcional, y punteros entre las distintas tablas [33].

Posee características que son [31]:

- Está optimizado para los equipos con múltiples procesadores.
- Es destacable su velocidad de respuesta.
- Puede usar como cliente-servidor o incrustado en aplicaciones.
- Posee varios conjuntos de tipos de datos.
- Resiste múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento distintos lo que permite optimizar el SGBD a cada caso concreto.
- Su administración se basa en usuarios y privilegios.
- Posee constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- Sus opciones de conectividad comprenden TCP/IP, sockets UNIX y sockets NT, también puede soportar completamente ODBC.
- No incluye características de objetos como tipos de datos estructurados definidos por el usuario, herencia, etc.

1.7.2. Base de dato no relacional

Mongodb.

MongoDB es una base de datos de documentos de código abierto que proporciona alto rendimiento, alta disponibilidad y escalado automático. Al ser una base de

datos NoSQL basada en documentos, redujo la necesidad de uniones y brinda la posibilidad de crear esquemas flexibles y dinámicos, afirma alto rendimiento, alta disponibilidad y escalabilidad horizontal [34].

Al permitir el uso de matrices y documentos anidados, el enfoque orientado a documentos permite representar relaciones jerárquicas complejas con un solo registro. Esto encaja naturalmente en la forma en que los desarrolladores que trabajan con lenguajes modernos orientados a objetos ven sus datos. Tampoco hay esquemas predefinidos: las claves y valores de un documento no tienen tipos ni tamaños fijos. Cuando no hay un esquema fijo, es más fácil agregar o eliminar campos según sea necesario. Esto generalmente acelera el desarrollo ya que los desarrolladores pueden iterar rápidamente [35].

Características

De acuerdo la página oficial de MongoDB Inc., las principales características de la base de datos son las siguientes [36]:

- MongoDB almacena datos en documentos flexibles similares a JSON, por lo que los campos pueden variar entre documentos y la estructura de datos puede cambiarse con el tiempo.
- El modelo de documento se asigna a los objetos en el código de su aplicación para facilitar el trabajo con los datos.
- Las consultas ad hoc, la indexación y la agregación en tiempo real ofrecen maneras potentes de acceder a los datos y analizarlos.
- MongoDB es una base de datos distribuida en su núcleo, por lo que la alta disponibilidad, la escalabilidad horizontal y la distribución geográfica están integradas y son fáciles de usar.
- MongoDB es de uso gratuito. Las versiones lanzadas antes del 16 de octubre de 2018 se publican bajo licencia AGPL. Todas las versiones posteriores al 16 de octubre de 2018, incluidas los parches lanzados para versiones anteriores.
- Alta disponibilidad con replicación y respaldo integrados.
- Escalabilidad horizontal con fragmentación nativa.

1.8. Estudio comparativo entre la base de datos

Base de datos – Relacional y no Relacional	Características	Ventajas	Desventajas
SQL Server	<p>Inteligencia en todos tus datos.</p> <p>Mejor Rendimiento.</p> <p>La plataforma de datos más segura.</p> <p>BI móvil integral.</p>	<p>Soporte de transacciones.</p> <p>Escalabilidad, estabilidad y seguridad.</p> <p>Incluye también un potente entorno gráfico.</p> <p>Uso de comandos DDL y DML gráficamente</p>	<p>Utiliza mucho la memoria RAM.</p> <p>Tiene restricciones.</p> <p>Bloqueos a nivel de página.</p> <p>Precio de licencia elevada.</p>
MySQL	<p>Permite escoger múltiples motores de almacenamiento.</p> <p>Conectividad segura.</p> <p>Replicación</p> <p>Tablas hash en memorias temporales</p> <p>Uso de multihilos mediante hilos de kernel.</p>	<p>MySQL es de uso libre y gratuito.</p> <p>Bajo costo en requerimientos para la elaboración.</p> <p>Facilidad de instalación y configuración.</p> <p>Soporte en casi el 100% de los sistemas operativos actuales.</p>	<p>No presentan documentación oficial.</p> <p>Monitorizar el rendimiento de las aplicaciones en busca de fallos.</p> <p>No es tan eficaz en aplicaciones que requieran de una constante modificación de escritura en</p>

			BD.
MongoDB	Son de fácil lectura para los usuarios. Se guarda en formato JSON. Consultas no estructuradas.	Validación de documentos. Motores de almacenamiento integrado. Menor tiempo de recuperación ante fallas. Muy utilizado para guardar los historiales de sistema complejo.	NO se recomienda para aplicaciones con transacciones complejas. Remplazo para la solución de herencia. Tecnología joven.

Tabla 3. Estudio comparativo entre las bases de datos.

Fuente: Elaborado por autor

La tabla 3 muestra las características, ventajas y desventajas entre las diferentes bases de datos, gracias al impacto que mantiene la base de datos MongoDB, se ha optado por integrar a nuestra aplicación web, ya que la conexión con nodeJS es más intuitivo, puesto que las librerías vienen integradas en su instalación.

Las principales ventajas son:

- Es multiplataforma.
- Lenguaje query es JavaScript.
- La escalabilidad en de forma horizontal.
- Modo de replicación es de maestro – esclavo.
- Recomendado para aplicaciones en la nube.
- Recuperación de datos automática, rápida e instantánea.
- La fragmentación viene incorporada.

1.9. Metodologías de Desarrollo de Software

La metodología de desarrollo de software es definida como un conjunto de técnicas y métodos que ayuda a abordar de manera homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. En definitiva, se indica que la metodología es un proceso de software detallado y completo. Generalmente las metodologías se basan en la combinación de los modelos de proceso genéricos, definiendo artefactos, roles y actividades, además con prácticas y técnicas recomendadas [37].

Una metodología para el desarrollo de software es un enfoque estructurado para desarrollar el software, la misma que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos. De esta manera se reconoce a la metodología para el desarrollo de software como un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito [38].

Si esto se aplica a la ingeniería del software, se destaca que una metodología permite:

- Optimizar el proceso y el producto software.
- Proporciona métodos que guían en la planificación y en el desarrollo del software.
- Definir qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

Una metodología define una estrategia global para enfrentarse con el proyecto. Entre los elementos que forman parte de una metodología se destaca [37]:

- Fases: tareas a realizar en cada etapa.
- Productos: E/S de cada fase, documentos.
- Procedimientos y herramientas: sirven de apoyo a la realización de cada tarea.
- Criterios de evaluación: del proceso y del producto. Saber si se ha alcanzado los objetivos.

1.9.1 Clasificación de las metodologías

Frente a la gran necesidad de que los proyectos lleguen al éxito y se obtenga un producto con gran valor para los clientes es importante escoger una metodología robusta que ajustada en un equipo cumpla con sus metas. El éxito del producto depende en gran medida de la metodología escogida por el equipo, ya sea tradicional o ágil, donde los equipos maximicen su potencial, incrementen la calidad del producto con los recursos y el tiempo que se ha establecido [39].

Metodologías Tradicionales.

En las metodologías tradicionales se concibe a un proyecto como uno solo de grandes dimensiones y estructura definida, el proceso se lo realiza de forma secuencial, en una sola dirección, es rígido y no cambia, se establece los requerimientos al inicio y para todo el proyecto, lo cual demanda grandes plazos de planeación previa y poca comunicación con el cliente una vez terminada dicha planeación [40].

Las metodologías tradicionales no se adaptan de manera adecuada a los cambios, por lo que no son métodos apropiados cuando se trabaja en un entorno donde los requisitos no pueden predecirse o pueden variar mientras se va desarrollando. Otra de las características importantes dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil [37].

Metodologías Ágiles.

Las metodologías ágiles son aquellas que aportan gran flexibilidad a los cambios, conservando las condiciones del proyecto, para lograr una respuesta más rápida y afrontar los nuevos retos de manera eficaz, lo cual reduce costes e incrementa la productividad. Esta se caracteriza porque los equipos de proyecto son multidisciplinares y se organizan de acuerdo con las necesidades de cada proyecto. Cada etapa del proyecto es llevada por un equipo diferente y va evolucionando según las necesidades de este. Para que esta manera de trabajar sea exitosa es fundamental que exista comunicación entre todos los miembros del proyecto [41].

Un modelo de desarrollo ágil es un proceso incremental (entregas frecuentes con ciclos rápidos), cooperativo (clientes y desarrolladores trabajan en constante comunicación), sencillo (es fácil de aprender y modificar) y finalmente adaptativo (permite cambios de último momento). Además, proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hace que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para el equipo de trabajo. Evita los caminos burocráticos de las metodologías tradicionales, generando poca documentación y no haciendo uso de métodos formales [37].

1.9.2. Comparación de metodologías

En la tabla se muestra la comparación de las metodologías tradicionales y ágiles, con el fin de ver las ventajas que posee cada uno de ellos.

Metodologías Tradicionales	Metodologías Ágiles
Cierta resistencia a los cambios	Preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente por el equipo
Contrato prefijado	Contrato flexible e incluso inexistente
Grupos grandes	Grupos pequeños (<10)
La arquitectura del software es esencial	Menor Énfasis en la arquitectura del software

Tabla 4. Diferencia entre la metodología tradicional y ágil.

Fuente: [42].

1.9.3. Metodologías ágiles

Scrum.

La metodología Scrum es diseñada para atender a los cambios repentinos y repetitivos que se pueden generar en el desarrollo de un proyecto. Esta metodología se fundamenta en la fácil documentación del desarrollo y en la relación directa entre el cliente y el grupo desarrollador. Es aquí donde se presentan los sprints, que es un periodo corto de tiempo donde se obtienen resultados sobre tareas planeadas desde el inicio del proyecto [43].

Comparado con el tradicional método como Cascada, Espiral e Iterativo, SCRUM es mejor preparado para manejar los cambios en el medio ambiente, y puede responder más fácilmente a los cambios en los requisitos, el cronograma y otras actualizaciones definidas externa o internamente.

Fase de planificación. - Sprint Planning reúne a todo el Equipo Scrum. El equipo de desarrollo pronostica lo que se puede desarrollar de manera realista durante el Sprint. El propietario del producto analiza el objetivo que el equipo Scrum espera lograr al final del Sprint (objetivo del Sprint), así como los elementos del Product Backlog que conducen allí. Luego, el equipo de desarrollo desarrolla el Sprint, teniendo siempre en cuenta la meta del Sprint, que a su vez afecta la funcionalidad y el uso de la tecnología [44].

Fase de Sprint. -El desarrollo real del proyecto ocurre en la fase Sprint.

Un Sprint es un ciclo iterativo de trabajo de desarrollo y generalmente dura entre 1 y 4 semanas. Al comienzo de un Sprint, un subconjunto de características del Product Backlog inicial es asignado para ser completado durante el Sprint. Durante el curso de un Sprint, no se pueden agregar otras características, pero las características dentro del Backlog del Sprint se pueden actualizar o cambiar dependiendo de las variables del entorno [45].

Fase de Cierre. -Una vez que un producto ha cumplido con las expectativas del desarrollo equipo, gerencia y cliente, el producto está preparado para liberación general como parte de la fase de Cierre. En este punto, el producto es final y tareas tales como preparar materiales de capacitación, agregar documentación de usuario y preparar material de marketing se completan [45].

Sprints SCRUM. -Los sprints comienzan reuniéndose con cada equipo de desarrollo para determinar cuál será la duración del Sprint, qué características del Backlog se van a implementar, y qué estándares que debe cumplir el producto. Al final de un Sprint, un producto entregable estará disponible. Una vez que el preliminar se especifican las pautas y requisitos para el Sprint, el comienza el desarrollo durante un Sprint.

El siguiente gráfico de SCRUM proceso de desarrollo, enfatizando la fase Sprint [46].

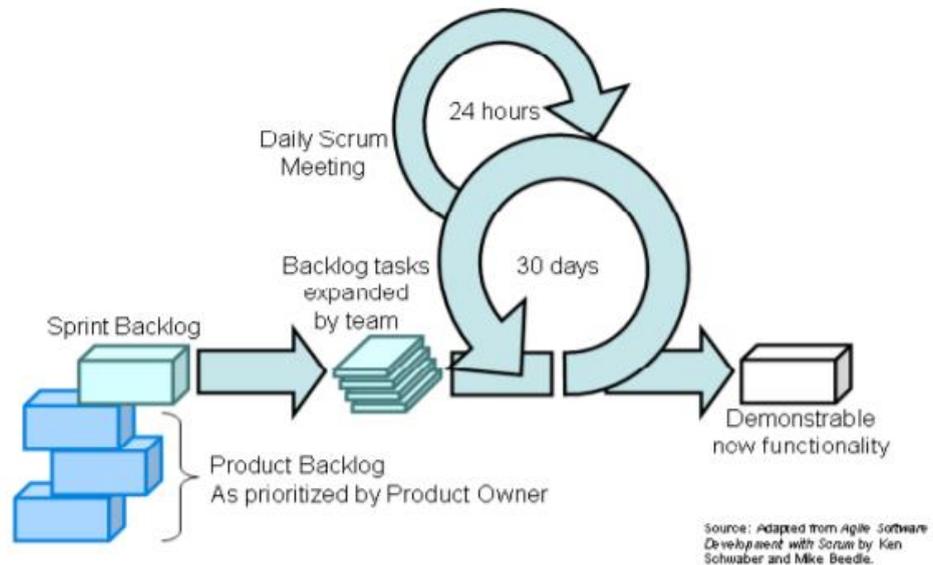


Figura 5. Esquema de la metodología Scrum [46].

Varias empresas y organizaciones de desarrollo de software han adoptado SCRUM como el método preferido para producir productos que funcionen y se puedan entregar en un entorno en el que los requisitos del sistema no están completamente definidos al comienzo del proceso de desarrollo. Proveedores de software independientes.

Visto desde el modelo de procesos, la gestión es crítica, ya que los equipos requieren una supervisión constante y minuciosa para asegurar que los problemas pueden detectarse y corregirse rápidamente. Al mismo tiempo, los gerentes externos a los Sprints deben permitir a los equipos un grado de libertad para tomar decisiones para ayudar al Sprint actual desarrollo [47].

Extreme programming XP.

Una metodología ágil particularmente popular es XP, fue desarrollado por Beck, Cunningham y Jeffries y es una disciplina ligera de desarrollo de software basado en principios de simplicidad, comunicación, retroalimentación y coraje [48].

Los valores se describen con más detalle a continuación:

Simplicidad. -En lugar de intentar predecir los requisitos futuros del sistema de software desde el principio, "los programadores extremos hacen ello más simple que podría funcionar", y "dejar el sistema en la condición más simple posible," [49]. Esta mejora la velocidad general de desarrollo sin dejar de manteniendo un énfasis en el software de trabajo.

Comunicación. -La mala comunicación en los equipos de software es una de las principales causas de fallas dentro de los proyectos. En XP, bien se enfatiza la comunicación entre todos los proyectos miembros: clientes, miembros del equipo y gerentes de proyecto.

Un representante del cliente debe estar presente en el sitio en todo momento para resolver dudas y aclarar proyecto requisitos Se utiliza la programación por parejas, por lo que cada programador puede revisar constantemente el trabajo del otro.

Comentarios. -Siempre debe haber alguna forma de obtener información sobre el sistema, para determinar con precisión el estado del proceso de desarrollo. Tal retroalimentación sirve como un indicador del progreso del proyecto e informa a los líderes del proyecto cuando hay que hacer cambios.

Coraje. -Se alienta a los programadores de XP a experimentar y reescribir código si no están satisfechos con el código o diseño existente.

Esto ayuda a mantener la moral sobre el proyecto y apoya mayor comunicación con otros miembros del proyecto [48].

Sobre la base de estos cuatro valores fundamentales, describir 12 prácticas básicas de XP [50].

Juego de planificación. - No existe una estrategia a largo plazo en un proyecto XP, sino más bien una fase de planificación del desarrollo, donde se recopilan "historias de usuario" del cliente.

Lanzamientos pequeños. - Similar al software evolutivo modelo de desarrollo, XP se propone ofrecer una serie de versiones totalmente funcionales y completamente probadas que conducen hasta la entrega del producto final.

Metáfora. - Una forma simple pero efectiva de obtener todos miembros del equipo del proyecto para visualizar el proyecto es por aportando una metáfora, que debería servir de inspiración, sugerir un vocabulario y una arquitectura básica.

Diseño simple. - Dado el énfasis en XP que se debe privilegiar la simplicidad, el diseño debe reflejar el deseo por el mínimo de código necesario para cumplir con el requerimiento en cada paso del desarrollo.

Pruebas. - XP adopta una postura muy diferente a la práctica tradicional de desarrollo de software en ese código de prueba se escribe antes de un código de implementación.

Integración continua. - Para garantizar que se mantenga el código consistente y disponible para todos los miembros del equipo de desarrollo, se realizan actualizaciones diarias en un repositorio de código compartido.

Programación en pares. - Este método ha demostrado ser muy exitoso, donde dos programadores comparten una máquina.

Propiedad colectiva. - Ningún individuo o grupo “posee” una pieza de software. Una vez depositado el código en el repositorio compartido, está disponible para cualquier equipo miembro a modificar, siempre y cuando no se rompan las pruebas como un resultado.

Refactorización. – Es una técnica disciplinada para reestructurar un cuerpo de código que ya existe, para alterar su estructura interna sin cambiar su comportamiento externo [51]. Como uno de los valores fundamentales que guían ágil métodos en general, los cambios de diseño son posibles incluso tarde en el ciclo de desarrollo.

Semana de 40 horas. - No definir estrictamente una semana específica semana laboral óptima, más bien se refiere a trabajar en un ritmo sostenible

Cliente en el sitio. - Tener al cliente disponible en todo momento durante el desarrollo del proyecto es crucial para XP prácticas, ya que en cualquier momento

decisiones que afecten al proyecto pueden surgir, y a menudo sólo el cliente puede precisar responder a tales preguntas.

Estándares de codificación. - Los estándares de codificación son prácticamente inevitable en XP, debido a la continua integración y propiedades de propiedad colectiva [48].

Esquema

El esquema de la metodología XP es el que se muestra en la figura 6.

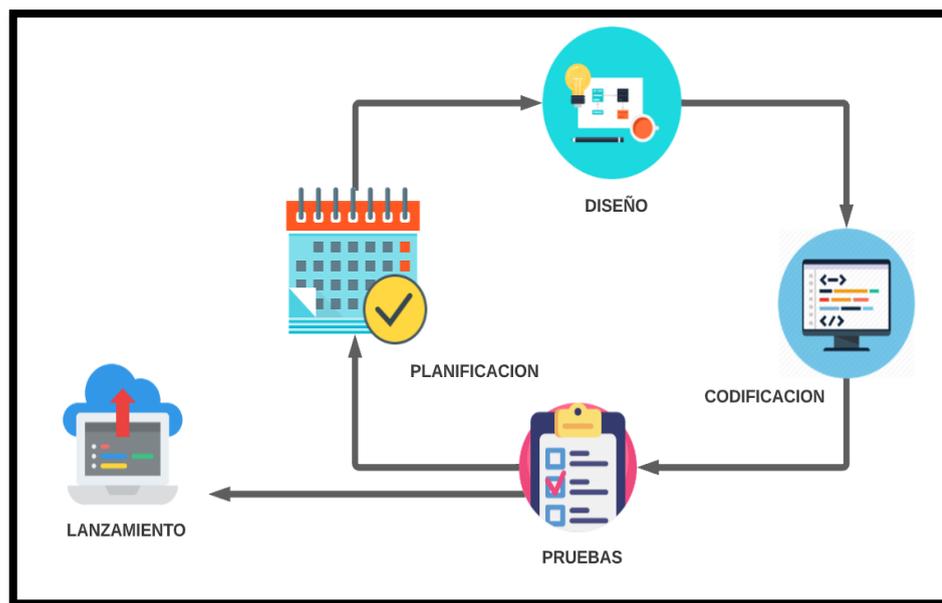


Figura 6. Esquema de la metodología Extreme programming.

Fuente: Elaborado por el autor.

1.9.4. Estudio comparativo entre las metodologías ágiles

Hemos analizado dos metodologías ágiles, en este caso scrum y xp ya que nos permiten gestionar aspectos y procesos distintos en nuestros equipos, con ello se pretenden mejorar el proceso de desarrollo de software, XP establece una lista de principios básicos que apoyan y promueven los valores de la sencillez, la comunicación, la retroalimentación y coraje y el método SCRUM, se pueden entregar lanzamientos confiables, a pesar de requisitos del sistema de cambios.

	SCRUM	XP
Alcance	<ul style="list-style-type: none"> • Elaboración la lista de backlog del proyecto • Elaboración del sprint de backlog del proyecto • Versionamiento • Revisión del progreso 	<ul style="list-style-type: none"> • Historias de usuarios • Ofrecer una serie de versiones totalmente funcionales y completamente probadas que conducen hasta la entrega del producto final
Calidad	<ul style="list-style-type: none"> • Revisión del diseño • Scrum diario • Desarrollo bien definido diseñado para reconocer y responder a los cambios en el entorno 	<ul style="list-style-type: none"> • Se especializa en el testing • Basado en principios de simplicidad, comunicación, retroalimentación. • Se enfatiza la comunicación entre todos los proyectos miembros
Plazos	<ul style="list-style-type: none"> • Iterar la planificación • Ciclos de ejecución muy cortos -entre una y cuatro semanas 	<ul style="list-style-type: none"> • Versionamiento • Iteración mensual

Tabla 5. Estudio comparativo entre la metodología ágiles.

Fuente: Elaborado por autor.

La tabla 5 muestra las metodologías que se adaptan de mejor manera al desarrollo del proyecto “APLICACIÓN WEB PARA LA GESTIÓN ADMINISTRATIVA DE LA COMUNIDAD ANGAHUANA ALTO DE LA PARROQUIA SANTA ROSA. “. Para mayor facilidad de desarrollo se seleccionó la metodología ágil XP, debido a las historias de usuario se puede ir lanzando pequeñas totalmente funcionales y completamente probadas que conducen hasta la entrega del producto final, la

metodología se especializa en el testing, debido a ello garantiza que las entregas del producto sea de óptima calidad.

1.10. Objetivos

1.10.1. Objetivo general

Implementar una aplicación web para la gestión administrativa en la comunidad Angahuana Alto de la parroquia Santa Rosa.

1.10.2 Objetivos específicos

1. Analizar las herramientas adecuadas para el desarrollo web con el fin de conocer las ventajas y desventajas.
2. Identificar los procesos administrativos y los requerimientos funcionales para el desarrollo web.
3. Desarrollar la aplicación web para la gestión administrativa de la comunidad Angahuana Alto de la parroquia Santa Rosa.

CAPITULO II. METODOLOGÍA

2.1. Materiales

En el presente proyecto de investigación se procedió a utilizar fuentes de información como: revistas científicas, sitios web, trabajos de titulación del área de informática y softwares, también se aplicó una entrevista a los directivos principales de la comunidad con el fin de determinar los requerimientos funcionales que fueron necesaria para el desarrollo del proyecto.

Material	Descripción
Computador	Se utiliza para la visualización de la página web administrativa.
Teléfono móvil	Se utiliza para la visualización de la adaptabilidad responsiva de la página web.
Mendeley Desktop	Software para el manejo de documentos y referencias bibliográficas.
Visual studio Code	Software para el desarrollo del sistema web
MongoDB Compass	Framework para la conexión con la base de datos.
Nodejs	Software para la creación de la aplicación.
Express	Framework para el alojamiento de ejecución de nodejs
Jade	Se utiliza para simplificar la sintaxis de html y agilizar el proceso de desarrollo.
Librería Npm	Sirve para instalar y gestionar versiones de paquetes y librería js.
Microsoft Word	Software para la redacción del proyecto.
Microsoft Excel	Software para el cálculo aritmético y creación de gráficos estadísticos.

Tabla 6. Materiales.

Fuente: Elaborado por el autor.

ENTREVISTA

DETERMINACIÓN DE REQUERIMIENTOS

EMPRESA: Comunidad Angahuana Alto

APLICACIÓN: Aplicación web para la gestión administrativa de la comunidad Angahuana Alto de la parroquia Santa Rosa.

OBJETIVO: Determinar los requerimientos y la funcionalidad de los procesos de la comunidad para el desarrollo de la aplicación web.

N.º	PREGUNTA
1	Cuáles son las actividades administrativas que realizan en la comunidad.?
2	Cuáles son los problemas que surgen habitualmente al momento de realizar las actividades.?
3	Que información maneja la comunidad.?
4	Cuales son los documentos que maneja la comunidad.?
5	Que sistemas informáticos maneja actualmente la comunidad.?
6	Le gustaría tener un sistema que administre la información de su comunidad.?
7	Cuáles serían los módulos que el sistema web contenga.?
8	Cuáles son sus expectativas a la facilidad de uso de la aplicación .?
9	Cuáles serían los usuarios que accederían a la aplicación

Tabla 7. Entrevista.

Fuente: Elaborado por el autor.

2.2 Métodos

2.2.1 Modalidad de Investigación

La siguiente investigación se basa en las modalidades: Campo, Bibliográfica-Documental y Aplicada. Las metodologías ágiles permitirán que el proyecto se adapte a cambios no previstos.

Modalidad de campo. - La investigación se de campo debido a que se realiza los estudios y los análisis necesario del problema, en el lugar donde se está desarrollando el proyecto, y también, se relaciona directamente con las personas interesadas, y de esta manera poder formular la mejor sistematización de los servicios que ofrece la comunidad Angahuana Alto a sus socios.

Modalidad bibliográfica o documental. - La investigación bibliográfica es porque consisten en la revisión de material bibliográfico, ya que incluye un conjunto de fases que abarcan la observación, la indagación, la conceptualización y criterios diversos de diferentes autores sobre el problema, en base a los diferentes documentos en diversos repositorios.

Modalidad aplicada. - Esta modalidad se utilizada porque se utiliza el conocimiento de los diferentes niveles adquiridos en la carrera universitaria.

2.2.2 Población y Muestra

Población: Para la presente investigación, la población esta conformada por los directivos principales de la comunidad Angahuana Alto.

Muestra: Por ser un número de población menor, se tomó como referencia poblacional a todos los directivos, ya que ellos son los usuarios que utilizaran el sistema.

Nº	Población	Número	Porcentaje(%)
1	Presidente del cabildo	1	20%
2	Vicepresidente	1	20%

3	Tesorero	1	20%
4	Secretario	1	20%
5	Vocal	1	20%
Total		5	100%

Tabla 8. Directivos de la comunidad Angahuana Alto.

Fuente: Elaborado por el Autor.

2.2.3 Recolección de Información

La información será recolectada a través de la observación y entrevista a los dirigentes principales que tiene la comunidad Angahuana Alto con la finalidad de sustentar el marco teórico.

Preguntas Básicas	Explicación
¿Para Qué?	Para alcanzar los objetivos de la investigación
¿De qué personas u objetos?	Servicios de la comunidad Angahuana Alto
¿Sobre qué aspectos?	Gestionar los servicios de los socios de la comunidad
¿Quién, Quienes?	Investigador: Segundo Carlos Iza Poaquiza
¿Cuándo?	Segundo semestre del 2021
¿Donde?	Comunidad Angahuana Alto de la Parroquia Santa Rosa
¿Cuántas veces?	Una
¿Qué técnicas de recolección?	Entrevista Observación
¿Con que?	Cuaderno de notas Guía de entrevista

Tabla 9. Recolección de información.

Fuente: Elaborado por autor.

2.2.4 Procesamiento y análisis de datos

Para el procesamiento y análisis de la información se procederá de la siguiente manera:

- Asistir a las reuniones con el fin de ver la fluidez del proceso
- Analizar los apuntes y la documentación que mantiene la comunidad
- Solicitar reuniones con los directivos con el fin de encontrar las necesidades.
- Realizar cuadros comparativos con la información adquirida.

CAPITULO III. RESULTADOS Y DISCUSION

3.1 Análisis y discusión de resultados

La observación de los procesos manuales y entrevista a los directivos de la comunidad Angahuana alto, se obtuvo los siguientes resultados las cuales permitió encontrar los requerimientos funcionales para el desarrollo del sistema web.

PREGUNTA	RESULTADO	CONCLUSIÓN
<p>Cuáles son las actividades administrativas que realizan en la comunidad.?</p>	<ul style="list-style-type: none"> • Gestión del socio • Reuniones • Mingas • Registro de asistencia • Cobro de rubros • Ingreso de gastos • Generación de actas • Seguimiento de actividades • Promocionar en mercados y plazas a la comunidad 	<p>Se concluye que la comunidad maneja sus procesos de forma manual, siendo este muy demoroso y dificulta al momento de gestionar las reuniones, asistencia, cobro de aportes y archivar los documentos, también la comunidad realiza gestión de promocionar las actividades que realizar los comuneros en las</p>
<p>Cuáles son los problemas que surgen habitualmente al momento de realizar las actividades.?</p>	<ul style="list-style-type: none"> • Perdida de información. • Demora en gestión de las reuniones y mingas. • Problemas en administrar los ingresos y gastos de la comunidad. • Deterioro y daño de 	<p>ferias libres en los cantones aledaños.</p>

	<p>las actas.</p> <ul style="list-style-type: none"> • Gastos en gestionar las promociones de la comunidad. 	<p>Estos problemas han generado inconvenientes y se busca solucionar con el desarrollo del sistema web, que permita la administración de los principales procesos y puedan administrar el contenido de su página principal con las actividades que realizan los comuneros, así como también, promocionar los lugares turísticos con el fin de reactivar la economía.</p>
<p>Que información maneja la comunidad.?</p>	<ul style="list-style-type: none"> • Informativa • Asistencia • Colaboraciones • Multas • Documentos • Entre otros 	
<p>Cuáles son los documentos que maneja la comunidad.?</p>	<ul style="list-style-type: none"> • Actas • Facturas de ingreso • Facturas de gastos • Solicitudes 	
<p>Que sistemas informáticos maneja actualmente la comunidad.?</p>	<p>Por el momento no manejan ningún sistema informático para la administración, si no cuadernos y archivadores.</p>	
<p>Le gustaría tener un sistema que administre la información de su comunidad.?</p>	<p>Si, la expectativa de la comunidad es mantener un sistema informático que administre los procesos principales y promocióne a la vez las principales actividades que realiza nuestros comuneros.</p>	

<p>Cuáles serían los módulos que el sistema web contenga.?</p>	<p>Administración de:</p> <ul style="list-style-type: none"> • Socios • Cobros • Gastos • Reuniones • Asistencia • Repositorio • Página principal • Usuarios • Seguimiento de Actividades 	<p>tecnológico van a ser los directivos que estén activos en ese periodo.</p>
<p>Cuáles son sus expectativas a la facilidad de uso de la aplicación .?</p>	<p>Que sea una aplicación 100% funcional, y sus módulos sean claras y ordenadas.</p>	
<p>Cuáles serían los usuarios que accederían a la aplicación</p>	<p>Los usuarios principales que accederían sería los directivos principales (presidente del cabildo, vicepresidente, secretario, tesorero y vocales principales) que estén activos en ese periodo.</p>	

Tabla 10. Resultado de la entrevista.

Fuente: Elaborado por el autor.

3.2 Desarrollo de la propuesta

Se desarrolló un sistema web para la automatización de los procesos en la comunidad Angahuana Alto. Para ello se utilizó una arquitectura MVC (Modelo Vista Controlador).

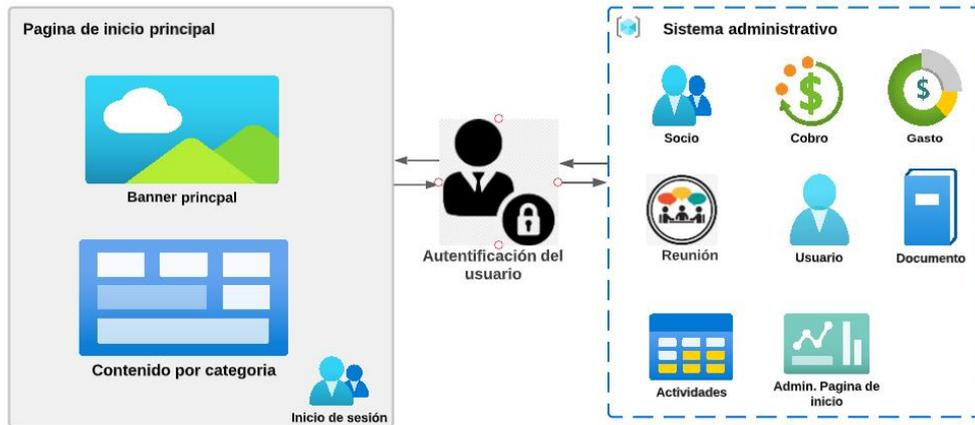


Figura 7. Diagrama de proceso del sistema.

Fuente: Elaborado por autor.

3.3 Aplicación de la metodología

Gracias a la elección de la metodología XP permitió presentar pruebas unitarias, con el fin de prevenir y corregir los errores, también obtener productos tangibles en periodos cortos.

3.3.1 Arquitectura del sistema web

La arquitectura del software permitió separar las distintas responsabilidades, así como también, reutilizar el código con el fin de facilitar futuros mantenimientos.

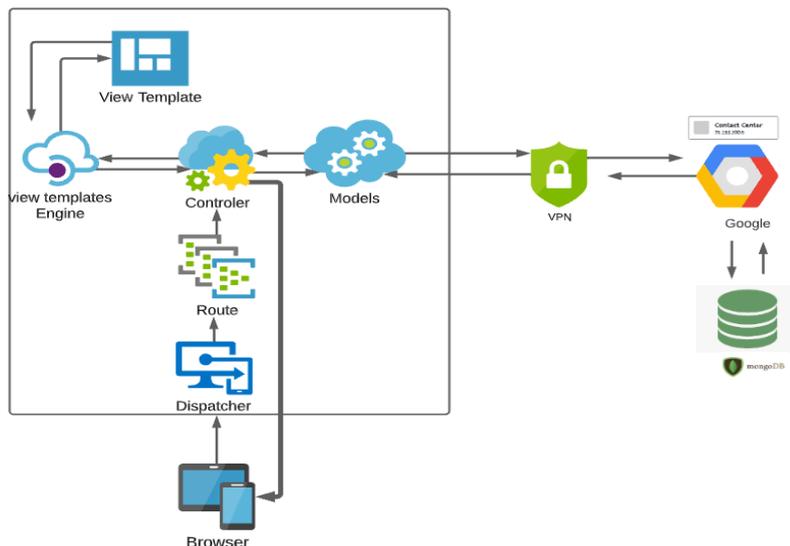


Figura 8. Arquitectura de sistema web para la automatización de procesos.
Fuerte: Elaborado por autor.

3.3.2 Fase 1: Exploración

Una vez definida la arquitectura y las funcionalidades generales se procede a generar las historias de usuario acorde a los requerimientos.

Historia de usuario	
Código: H01	Usuario: Desarrollador
Nombre Historia: Diseño de la entidad-relación de la base de datos.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 5	Iteración asignada: 1
Programador Responsable: Carlos Iza	
Descripción: El diseño de la entidad-relación de una base de datos es la pieza principal para un buen desarrollo, ya que es el responsable de guardar, procesar y devolver la información.	
Observación: Se declaró el modelo en nodeJS para posterior ingreso en la base de datos, así como, sus atributos y el tipo de columna.	

Tabla 11. Historia de usuario-Base de datos.

Fuerte: Elaborado por autor.

Historia de usuario

Código: H02	Usuario: Usuario final
Nombre Historia: Página de inicio de sistema administrativo.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador Responsable: Carlos Iza	
Descripción: La página de inicio dará la bienvenida con las novedades principales de la comunidad.	
Observación: La página será informativa, el cual no será navegable.	

Tabla 12. Historia de usuario-Página de inicio.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H03	Usuario: Administrador
Nombre Historia: Gestión de la página de inicio.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador Responsable: Carlos Iza	
Descripción: Formularios de administración del contenido, como el banner principal, contenido por categorías y actualización de los textos principales.	
Observación: Validar el tamaño de la imagen y su contenido.	

Tabla 13. Historia de usuario-Página de inicio dentro del sistema.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H04	Usuario: Usuario final
Nombre Historia: Formulario de ingreso al sistema.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 4	Iteración asignada: 1
Programador Responsable: Carlos Iza	
Descripción: El usuario podrá ingresar a la página web, con su cedula como	

usuario y su respectiva contraseña.
Observación: Los directivos principales serán los usuarios.

Tabla 14. Historia de usuario-Formulario de ingreso.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H05	Usuario: Usuario final
Nombre Historia: Página de inicio dentro del sistema web	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador Responsable: Carlos Iza	
Descripción: La página web contiene un menú de navegación, y un resumen de los movimientos de sistema.	
Observación: El sistema presenta un resumen del sistema como número de clientes, ingresos y egresos.	

Tabla 15. Historia de usuario-Página de inicio dentro del sistema.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H06	Usuario: Usuario final
Nombre Historia: Listado de socios.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador Responsable: Carlos Iza	
Descripción: Mostrar el listado de los socios activos, así como también, la implementación de la búsqueda y paginación.	
Observación: Los usuarios se pueden exportar en diferentes formatos como pdf, csv y excel.	

Tabla 16. Historia de usuario-Formulario para listar los socios.

Fuente: Elaborado por autor.

Historia de usuario

Código: H07	Usuario: Usuario final
Nombre Historia: Formulario para la creación del socio.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador Responsable: Carlos Iza	
Descripción: El formulario permite validar y registrar de los datos de los socios.	
Observación: Validar la información correspondiente.	

Tabla 17. Historia de usuario-Formulario de ingreso del socio.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H08	Usuario: Usuario final
Nombre Historia: Formulario para la edición del socio.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador Responsable: Carlos Iza	
Descripción: Formulario para la edición de los datos de un socio.	
Observación: Validar la nueva información correspondiente.	

Tabla 18. Historia de usuario-Formulario la edición del socio.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H09	Usuario: Administrador
Nombre Historia: Gestión de usuario.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 3
Programador Responsable: Carlos Iza	
Descripción: Se visualizará los usuarios registrados en el sistema, se podrá editar y dar de baja al usuario.	

Observación: Validar la nueva información correspondiente.

Tabla 19. Historia de usuario-Gestión de usuario.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H10	Usuario: Usuario final
Nombre Historia: Formulario de registro de una reunión.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 4
Programador Responsable: Carlos Iza	
Descripción: Formulario que permite validar y registrar los datos principales de una reunión, así como también el listado de los socios.	
Observación: El formulario cuenta con el listado de los socios.	

Tabla 20. Historia de usuario-Formulario de registro de reunión.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H11	Usuario: Usuario final
Nombre Historia: Formulario de registro de asistencia.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 4
Programador Responsable: Carlos Iza	
Descripción: Formulario que permite listar los socios participantes, así como también, la información general de la reunión.	
Observación: Permite la búsqueda y validación al momento del registro de asistencia.	

Tabla 21. Historia de usuario-Formulario de registro de asistencia.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H12	Usuario: Usuario final
Nombre Historia: Formulario de cobro de rubros.	

Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada:4
Programador Responsable: Carlos Iza	
Descripción: Formulario que permite listar los rubros pendientes para su posterior cobro.	
Observación: Se puede visualizar el listado total de rubros, pero se puede realizar cobros por partes.	

Tabla 22. Historia de usuario-Formulario de cobro.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H13	Usuario: Usuario final
Nombre Historia: Formulario para el registro de un nuevo rubro.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada:4
Programador Responsable: Carlos Iza	
Descripción: Formulario que permite validar y registrar del nuevo rubro, para su posterior asignación en cada uno de los socios.	
Observación: Los rubros se asignan automáticamente para todos los socios según sea el caso.	

Tabla 23. Historia de usuario-Formulario de cobro.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H14	Usuario: Usuario final
Nombre Historia: Formulario para el registro de los documentos.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada:5
Programador Responsable: Carlos Iza	
Descripción: El formulario de registro de documentos permite subir y	

almacenar la documentación, para su posterior descarga y visualización
Observación: Validar el ingreso de los documentos.

Tabla 24. Historia de usuario-Formulario de registro de documentos.

Fuente: Elaborado por autor.

Historia de usuario	
Código: H15	Usuario: Usuario final
Nombre Historia: Gestión de reportes.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 5
Programador Responsable: Carlos Iza	
Descripción: Se presenta principales reportes que necesita la comunidad.	
Observación: Permite la visualización de distintos reportes.	

Tabla 25. Historia de usuario-Formulario de registro de documentos.

Fuente: Elaborado por autor.

3.3.3 Fase 2: Planificación del proyecto

Se define la valoración del tiempo para cada historia de usuario, estimación de tiempo-empleado trabajando 25 horas a la semana de lunes a viernes. Esta fase se plantea la estimación de esfuerzo expuesta a continuación.

N.º	Historia de Usuario	Tiempo Estimado		
		Semana	Días	Horas
1	Diseño de la entidad-relación de la base de datos	1	5	25
2	Página de inicio de sistema administrativo	1,2	6	30
3	Gestión para la administración de la página de inicio	2	10	50
4	Formulario de ingreso al sistema	0,8	4	20
5	Página de inicio dentro del sistema web	0,8	4	20
6	Listado de socios	0,8	4	20
7	Formulario para la creación de un socio	0,8	4	20
8	Formulario para la edición del socio	0,8	4	20

9	Gestión de usuario	1,6	8	40
10	Formulario de registro de una reunión	1,6	8	40
11	Formulario de registro de asistencia	1,6	8	40
12	Formulario de cobro de rubros	1,6	8	40
13	Formulario para el registro de un nuevo rubro	1,2	6	30
14	Formulario para el registro de los documentos	1,8	9	45
15	Gestión de reportes	1,6	8	40
	Tiempo estimado total	19,2	96	480

Tabla 26. Historia de usuario-Formulario de registro de documentos.

Fuente: Elaborado por autor.

Una vez defina las estimaciones de esfuerzo de las historias de usuario, a continuación, se plantea las entregas mediante las iteraciones, la cual se detalla a continuación.

N.º Iteración	Historia de Usuario	Tiempo Estimado		
		Semana	Días	Horas
1	H01	1	5	25
	H02	1,2	6	30
	H03	2	10	50
	H04	0,8	4	20
2	H05	0,8	4	20
	H06	0,8	4	20
	H07	0,8	4	20
	H08	0,8	4	20
3	H09	1,6	8	40
4	H10	1,6	8	40
	H11	1,6	8	40
	H12	1,6	8	40
	H13	1,2	6	30
5	H14	1,8	9	45
	H15	1,6	8	40
Tiempo estimado total		19,2	96	480

Tabla 27. Plan de entregas.

Fuente: Elaborado por autor.

Se procede a definir roles del equipo, para una mejor planificación el cual se detalla a continuación.

Miembro	Rol
Carlos Iza	Desarrollador
Directivo de la comunidad Angahuana Alto	Cliente
Presidente de la comunidad Angahuana Alto	Pruebas
Ing. Felix Fernandez	Entrenador
Carlos Iza Ing. Felix Fernandez	Seguimiento

Tabla 28. Equipo y roles.

Fuente: Elaborado por autor.

Asignación de la fecha de inicio y fin para las tareas, así como al responsable que se encargará de realizar dicha tarea en el tiempo establecido.

Tareas: Diseño de base de datos – H01

Tarea	Responsable: Carlos Iza
N.º Tarea: T1	Código de Historia: H01
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Definición y modelado de la base de datos.	
Fecha inicio: 04/10/2021	Fecha fin: 08/10/2021
Descripción: Analizar los requerimientos y las funcionalidades del sistema, previo al modelado de la base de datos.	

Tabla 29. Tarea - Modelado de la base de datos.

Fuente: Elaborado por autor.

Tareas: Página de inicio de sistema administrativo – H02.

Tarea	Responsable: Carlos Iza
N.º Tarea: T2	Código de Historia: H02
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Nombre de la Tarea: Definir la plantilla para la página de inicio.	
Fecha inicio: 11/10/2021	Fecha fin: 12/10/2021
Descripción: Se utiliza la plantilla prediseñada con materialice, ya que se pretende un diseño responsive para diferentes dispositivos.	

Tabla 30. Tarea – Plantilla para la página de inicio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T3	Código de Historia: H02
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Nombre de la Tarea: Diseño de las secciones de la página de inicio	
Fecha inicio: 13/10/2021	Fecha fin: 14/10/2021
Descripción: Diseño y creación de las secciones informativas de la página de inicio.	

Tabla 31. Tarea – Diseño de las secciones de la página de inicio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T4	Código de Historia: H02
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad del menú de navegación de la página de inicio.	
Fecha inicio: 15/10/2021	Fecha fin: 18/10/2021
Descripción: Se añade las funcionalidades del menú de navegación de la página de inicio.	

Tabla 32. Tarea – Funcionalidad del menú de navegación en la página de inicio.

Fuente: Elaborado por autor.

Tareas: Gestión para la administración de la página de inicio – H03.

Tarea	Responsable: Carlos Iza
N.º Tarea: T5	Código de Historia: H03
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Diseño del formulario para la administración del banner principal de la página de inicio.	
Fecha inicio: 19/10/2021	Fecha fin: 19/10/2021
Descripción: Diseño del formulario para la administración del banner principal de la página de inicio, su contenido y logo.	

Tabla 33. Tarea – Diseño del formulario para la administración del banner principal de la página de inicio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T6	Código de Historia: H03
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad para el formulario para la administración del banner principal de la página de inicio.	
Fecha inicio: 20/10/2021	Fecha fin: 21/10/2021
Descripción: Se añade la funcionalidad para el formulario de administración del banner principal de la página de inicio.	

Tabla 34. Tarea – Funcionalidad para el formulario de la administración del banner de la página de inicio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T7	Código de Historia: H03
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Diseño del formulario para la administración del contenido principal de la página de inicio.	
Fecha inicio: 22/10/2021	Fecha fin: 22/10/2021
Descripción: Diseño del formulario para la administración del contenido principal, logo y descripción.	

Tabla 35. Tarea – Diseño del formulario para la administración del banner principal de la página de inicio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T8	Código de Historia: H03
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad para el formulario para la administración del contenido principal de la página de inicio	
Fecha inicio: 25/10/2021	Fecha fin: 26/10/2021
Descripción: Se añade la funcionalidad para el formulario para la administración del contenido principal de la página de inicio	

Tabla 36. Tarea – Funcionalidad para el formulario de la administración del banner de la página de inicio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T9	Código de Historia: H03
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Diseño del formulario para la administración de las categorías del contenido principal para la página de inicio.	
Fecha inicio: 27/10/2021	Fecha fin: 28/10/2021
Descripción: Diseño del formulario para la administración de las categorías del contenido principal para la página de inicio.	

Tabla 37. Tarea – Diseño del formulario para la administración del banner principal de la página de inicio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T10	Código de Historia: H03
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad para el formulario para la administración de las categorías del contenido principal para la página de inicio.	
Fecha inicio: 29/10/2021	Fecha fin: 01/11/2021
Descripción: Se añade la funcionalidad para el formulario para la administración de las categorías del contenido principal para la página de inicio.	

Tabla 38. Tarea – Funcionalidad para el formulario de la administración del banner de la página de inicio.

Fuente: Elaborado por autor.

Tareas: Formulario de ingreso al sistema– H04.

Tarea	Responsable: Carlos Iza
N.º Tarea: T11	Código de Historia: H04
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Diseño del formulario para el ingreso al sistema.	
Fecha inicio: 04/11/2021	Fecha fin: 05/11/2021
Descripción: Diseño del formulario en la página de inicio para el ingreso al sistema.	

Tabla 39. Tarea - Modelado de la base de datos.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T12	Código de Historia: H04

Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad en el formulario de ingreso al sistema.	
Fecha inicio: 08/11/2021	Fecha fin: 09/12/2021
Descripción: Funcionalidad en el formulario de ingreso al sistema, validación de usuario y datos.	

Tabla 40. Tarea – Funcionalidad en el formulario de ingreso al sistema.

Fuente: Elaborado por autor.

Tareas: Página de inicio dentro del sistema web – H05

Tarea	Responsable: Carlos Iza
N.º Tarea: T13	Código de Historia: H05
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Definición de la plantilla para la página de inicio dentro del sistema	
Fecha inicio: 10/11/2021	Fecha fin: 10/11/2021
Descripción: Se define la plantilla para la página de inicio dentro del sistema, esta debe ser responsive.	

Tabla 41. Tarea – Definición de plantilla para la página de inicio dentro del sistema.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T14	Código de Historia: H05
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Menú de navegación acorde a los perfiles.	
Fecha inicio: 11/11/2021	Fecha fin: 11/11/2021
Descripción: Analizar y desarrollo el menú de navegación acorde a los perfiles de usuario.	

Tabla 42. Tarea – Menú de navegación acorde a los perfiles.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T15	Código de Historia: H05
Tipo de Tarea: Desarrollo	Puntos Estimados: 4

Nombre de la Tarea: Funcionalidad del menú de navegación dentro del sistema.	
Fecha inicio: 12/11/2021	Fecha fin: 15/11/2021
Descripción: Añadir las funcionalidades de navegación entre diferentes páginas.	

Tabla 43. Tarea – Funcionalidad del menú de navegación.

Fuente: Elaborado por autor.

Tareas: Pagina para listar al socio – H06

Tarea	Responsable: Carlos Iza
N.º Tarea: T16	Código de Historia: H06
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario de carga de los socios.	
Fecha inicio: 16/11/2021	Fecha fin: 19/11/2021
Descripción: Formulario de carga de los socios registrados en el sistema.	

Tabla 44. Tarea - Modelado de la base de datos.

Fuente: Elaborado por autor.

Tareas: Formulario para la creación de un socio – H07

Tarea	Responsable: Carlos Iza
N.º Tarea: T17	Código de Historia: H07
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario para añadir un nuevo socio.	
Fecha inicio: 22/11/2021	Fecha fin: 23/11/2021
Descripción: Formulario para añadir un nuevo socio con las validaciones correspondientes.	

Tabla 45. Tarea – Formulario para añadir un nuevo socio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T18	Código de Historia: H07
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad del formulario para añadir un nuevo socio.	
Fecha inicio: 24/11/2021	Fecha fin: 25/11/2021
Descripción: Funcionalidad del formulario para añadir un nuevo socio con las validaciones correspondientes.	

Tabla 46. Tarea – Funcionalidad del formulario para añadir un nuevo socio.

Fuente: Elaborado por autor.

Tareas: Formulario para la edición del socio – H08.

Tarea	Responsable: Carlos Iza
N.º Tarea: T19	Código de Historia: H08
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario para la edición de un socio.	
Fecha inicio: 26/11/2021	Fecha fin: 29/11/2021
Descripción: Formulario para la edición de un socio, con sus respectivas validaciones.	

Tabla 47. Tarea – Formulario para la edición del socio.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T20	Código de Historia: H08
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad del formulario para la edición de un socio.	
Fecha inicio: 30/11/2021	Fecha fin: 01/12/2021
Descripción: Funcionalidad del formulario para la edición de un socio, con sus respectivas validaciones.	

Tabla 48. Tarea – Funcionalidad del formulario para la edición del socio.

Fuente: Elaborado por autor.

Tareas: Formulario para la edición del socio – H09.

Tarea	Responsable: Carlos Iza
N.º Tarea: T21	Código de Historia: H09
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario de creación de usuario.	
Fecha inicio: 02/12/2021	Fecha fin: 03/12/2021

Descripción: Formulario de creación de usuario con las validades correspondientes.

Tabla 49. Tarea – Formulario de creación de usuario.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T22	Código de Historia: H09
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad del formulario de creación de usuario.	
Fecha inicio: 06/12/2021	Fecha fin: 08/12/2021
Descripción: validación de datos en el formulario de creación de usuario.	

Tabla 50. Tarea – Funcionalidad del formulario de creación de usuario

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T23	Código de Historia: H09
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario para la edición del usuario.	
Fecha inicio: 09/12/2021	Fecha fin: 13/12/2021
Descripción: Formulario para la edición del usuario.	

Tabla 51. Tarea – Formulario para la edición del usuario.

Fuente: Elaborado por autor.

Tareas: Formulario de registro de reunión – H10.

Tarea	Responsable: Carlos Iza
N.º Tarea: T24	Código de Historia: H10
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario de registro de la reunión.	
Fecha inicio: 14/12/2021	Fecha fin: 17/12/2021
Descripción: Diseño del formulario para el registro de la reunión.	

Tabla 52. Tarea – Formulario de registro de la reunión.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T25	Código de Historia: H10
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad de registro de la reunión.	
Fecha inicio: 20/12/2021	Fecha fin: 23/12/2021
Descripción: Funcionalidad y validación de los datos de una reunión.	

Tabla 53. Tarea – Funcionalidad de registro de la reunión.

Fuente: Elaborado por autor.

Tareas: Formulario de registro de asistencia – H11.

Tarea	Responsable: Carlos Iza
N.º Tarea: T26	Código de Historia: H11
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario de registro de asistencia.	
Fecha inicio: 27/12/2021	Fecha fin: 30/12/2021
Descripción: Formulario de registro de asistencia.	

Tabla 54. Tarea – Formulario de registro de asistencia.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T27	Código de Historia: H11
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad para el registro de asistencia.	
Fecha inicio: 03/01/2022	Fecha fin: 06/01/2022
Descripción: Añadir la funcionalidad para el registro de asistencia.	

Tabla 55. Tarea – Funcionalidad para el registro de asistencia.

Fuente: Elaborado por autor.

Tareas: Formulario de cobro de rubros– H12.

Tarea	Responsable: Carlos Iza
N.º Tarea: T28	Código de Historia: H12
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario de cobro de rubro.	
Fecha inicio: 07/01/2022	Fecha fin: 12/01/2022
Descripción: Diseño del formulario de cobro de rubro.	

Tabla 56. Tarea – Formulario de cobro de rubro.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T29	Código de Historia: H12
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad del formulario para cobro del rubro.	
Fecha inicio: 13/01/2022	Fecha fin: 18/01/2022
Descripción: Añadir la funcionalidad al formulario para el cobro de rubro.	

Tabla 57. Tarea – Funcionalidad del formulario para cobro del rubro.

Fuente: Elaborado por autor.

Tareas: Formulario de registro de nuevo rubros– H13.

Tarea	Responsable: Carlos Iza
N.º Tarea: T30	Código de Historia: H13
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario para el ingreso de un rubro.	
Fecha inicio: 19/01/2022	Fecha fin: 21/01/2022
Descripción: Diseño del formulario para el ingreso del rubro.	

Tabla 58. Tarea – Formulario para el ingreso de un rubro.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T31	Código de Historia: H13
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad para el ingreso de un nuevo rubro.	
Fecha inicio: 24/01/2022	Fecha fin: 26/01/2022
Descripción: Añadir la funcionalidad para el ingreso de un nuevo rubro.	

Tabla 59. Tarea – Funcionalidad para el ingreso de un nuevo rubro.

Fuente: Elaborado por autor.

Tareas: Formulario para el registro de los documentos– H14.

Tarea	Responsable: Carlos Iza
N.º Tarea: T32	Código de Historia: H14
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Formulario para el ingreso de documento.	
Fecha inicio: 27/01/2022	Fecha fin: 01/02/2022
Descripción: Diseño del formulario de ingreso del documento.	

Tabla 60. Tarea – Formulario para el ingreso de documento.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T33	Código de Historia: H14
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Funcionalidad para el registro del documento.	
Fecha inicio: 02/02/2022	Fecha fin: 08/02/2022
Descripción: Añadir la funcionalidad para el registro del documento.	

Tabla 61. Tarea – Funcionalidad para el registro del documento.

Fuente: Elaborado por autor.

Tareas: Gestión de reportes– H15.

Tarea	Responsable: Carlos Iza
N.º Tarea: T34	Código de Historia: H15
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Diseño de informes.	
Fecha inicio: 09/02/2022	Fecha fin: 14/02/2022
Descripción: Presentar al usuario un diseño fácil de identificar la información requerida.	

Tabla 62. Tarea – Funcionalidad para el registro de asistencia.

Fuente: Elaborado por autor.

Tarea	Responsable: Carlos Iza
N.º Tarea: T35	Código de Historia: H15
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Nombre de la Tarea: Presentación de informes.	
Fecha inicio: 15/02/2021	Fecha fin: 18/02/2021
Descripción: Los informes será de gran ayuda a los directivos para la toma de decisiones por el bien del socio.	

Tabla 63. Tarea – Presentación de informes.

Fuente: Elaborado por autor.

3.3.4 Fase 3: Iteraciones

Luego de la elaboración y valoración de las historias de usuario se procede a ejecutar cada una de las iteraciones, así como lo dicta la metodología XP

3.3.4.1 Iteración 1

Las siguientes historias de usuario perteneciente a la iteración

		Tiempo Estimado
--	--	------------------------

N.º Iteración	Historia de Usuario	Días	Horas
1	H01	5	20
	H02	6	24
	H03	8	32
	H04	4	16

Tabla 64. Tarea – Presentación de informes.

Fuente: Elaborado por el autor.

H1: Diseño de la base de datos

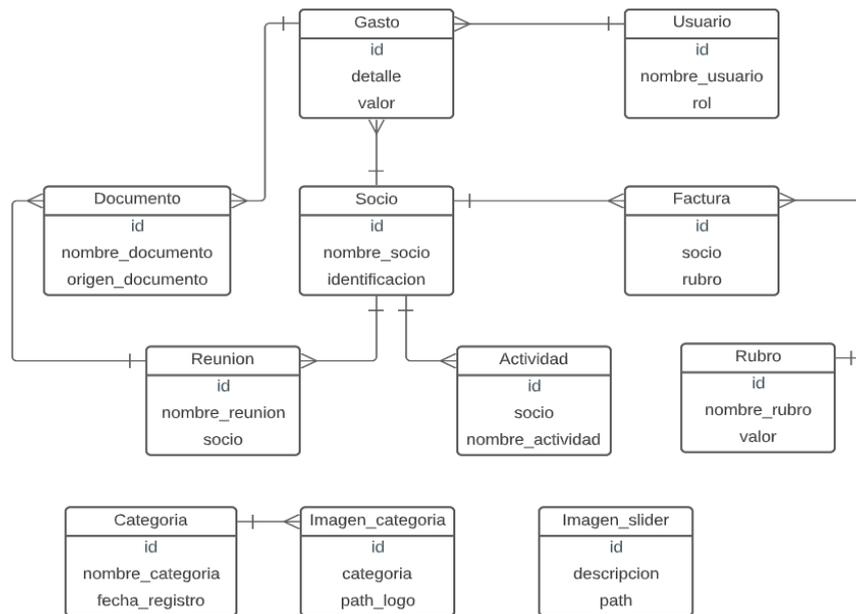


Figura 9. Modelo relacional de la base de datos.

Fuente: Elaboración por el autor.

Codificación

Se instala los complementos y las herramientas necesarias para el desarrollo, iniciado el proyecto en visual studio code se procede a conectar con la base de datos mongoDB.

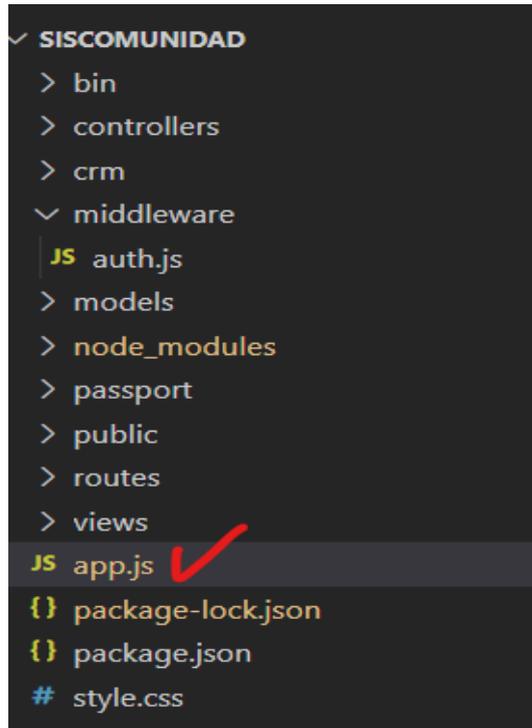


Figura 10. Clase principal del proyecto

Fuente: Elaboración por el autor.

```
var mongoose = require('mongoose');  
mongoose.connect('mongodb://35.193.200.5:27020/siscomunidad');
```

Figura 11. Conexión a la base de datos.

Fuente: Elaboración por el autor.

Se procede a exportar la librería 'mongoose', con el cual se utiliza la función 'connect' y como parámetro la cadena de conexión.

H02: Página de inicio del sistema administrativo.



Figura 12. Página de inicio del sistema administrativo.

Fuente: Elaboración por el autor.

El diseño se los realiza en motor HTML jade, donde se carga el contenido desde la base de datos, ya que la página es administrable.

H03: Gestión para la administración de la página de inicio.

Gestión Categorías

Nueva Categoría

Nombre de Categoría

CREAR CATEGORIA

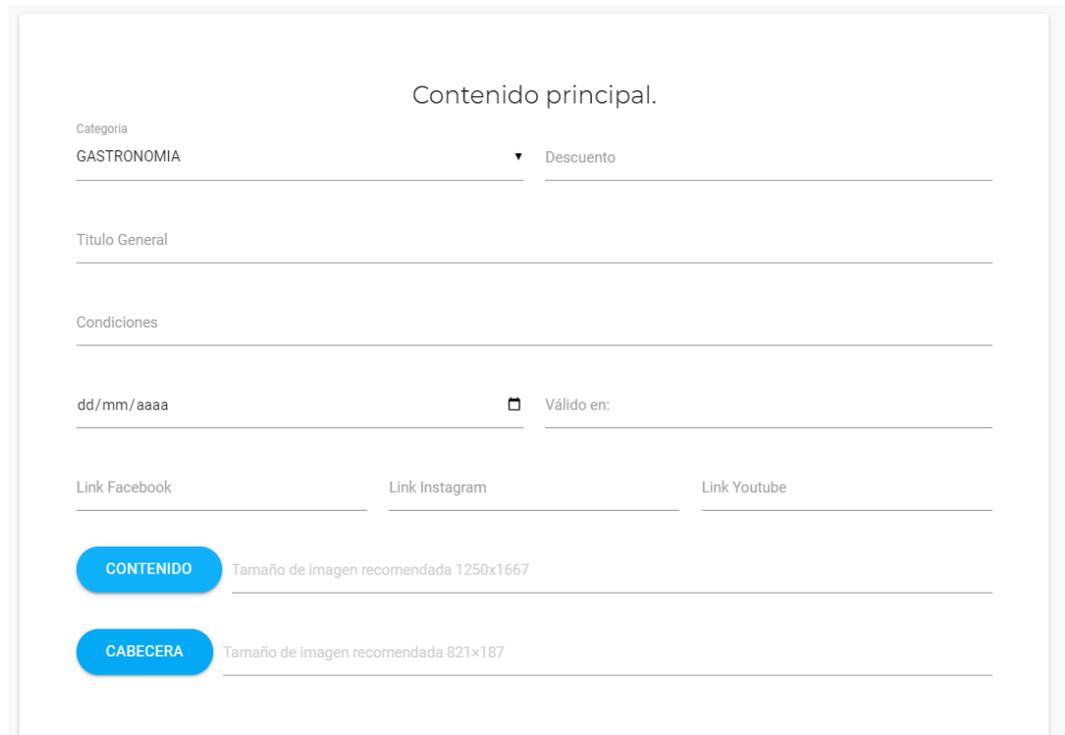
Categorías Existentes

N	Categoría
1	GASTRONOMIA
2	LUGARES TURISTICOS
3	UBICACIÓN
4	VESTIMENTA
5	INSTRUMENTOS MUSICALES

Figura 13. Formulario administración de las categorías de la página.

Fuente: Elaboración por el autor.

El diseño se los realiza en motor HTML jade, y la función principal para la administración del contenido principal con sus métodos get y post.



Contenido principal.

Categoría
GASTRONOMIA ▼ Descuento

Título General

Condiciones

dd/mm/aaaa Válido en:

Link Facebook Link Instagram Link Youtube

CONTENIDO Tamaño de imagen recomendada 1250x1667

CABECERA Tamaño de imagen recomendada 821x187

Figura 14. Formulario del contenido principal de la página de bienvenida.

Fuente: Elaboración por el autor.

El diseño se los realiza en motor HTML jade, donde se carga el contenido desde la base de datos, ya que la página es administrable.



Página principal

TAMAÑO RECOMENDADO
2376x807

Texto1 Texto2

Link(ver más) **IMAGEN** No se ha subido ninguna imagen

Figura 15. Formulario de administración del banner principal.

Fuente: Elaboración por el autor.

El diseño se los realiza en motor HTML jade, donde se carga el contenido desde la base de datos, ya que la página es administrable.

H04: Formulario de ingreso al sistema.

En el formulario de ingreso se tiene dos campos importantes que son: el usuario el número de cedula del socio y su contraseña correspondiente, de ser correcta las credenciales dirige a la pantalla principal dentro del sistema web, en caso de haber ingresado incorrectamente emite un mensaje de error.

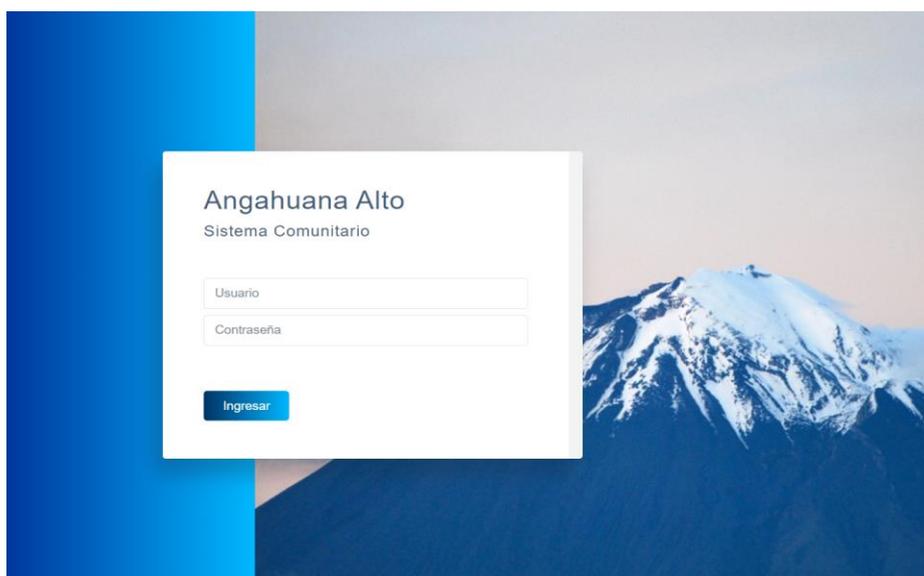


Figura 16. Formulario de ingreso al sistema.

Fuente: Elaboración por el autor.

La codificación se los realiza en la función llamado Passport.js la cual se dedica a la validación de los datos mediante su librería 'LocalStrategy'.

La librería de nodejs llamada 'bcryptjs' permite construir plataforma de seguridad de contraseñas.

```
var bcrypt = require('bcryptjs');
```

Figura 17. Librería de hashing

Fuente: Elaboración por el autor

3.3.4.2. Iteración 2

Se procede a desarrollar la segunda iteración conforme a la planificación.

N.º Iteración	Historia de Usuario	Tiempo Estimado	
		Días	Horas
2	H05	4	16
	H06	4	16
	H07	4	16
	H08	4	15

Tabla 65. Tarea – Historias de usuario iteración 2.

Fuente: Elaborado por el autor.

H05: Página de inicio dentro del sistema web.

En la figura 17 muestra la página de inicio dentro del sistema web, presentando un breve resumen del sistema administrativo, con su menú de navegación correspondiente al perfil establecido.

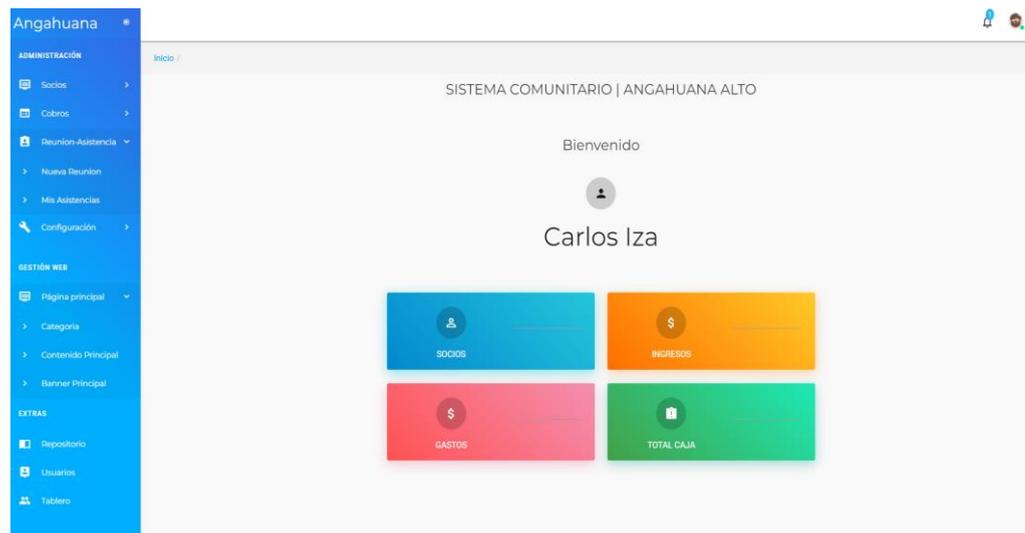


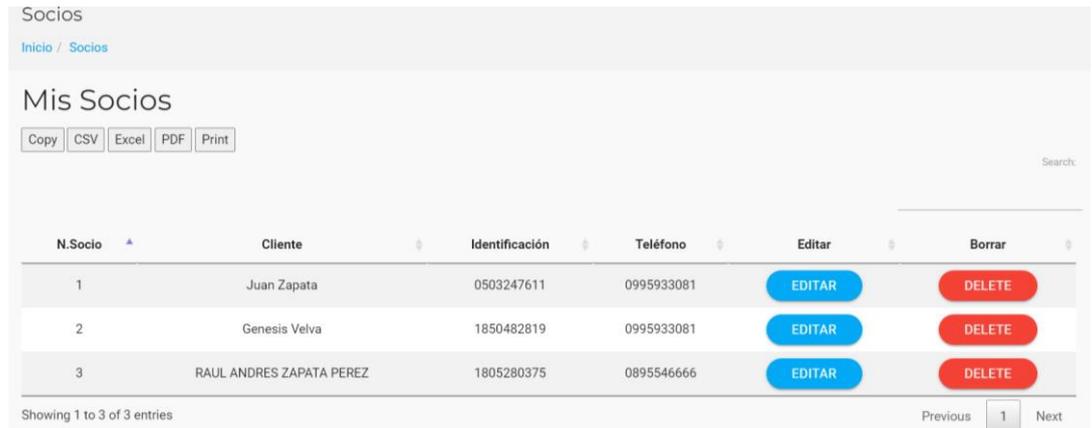
Figura 18. Página de inicio dentro del sistema web.

Fuente: Elaboración por el autor.

Validar la activación del menú conforme al perfil, recupera y muestra la información del usuario, así como el resumen de la información más relevante.

H06: Formulario para la listar al socio.

El formulario de listar socio permite llevar el control y seguimiento de los socios, ya que permite exportar en otros formatos gracias a las librerías que ofrece nodejs, así como también, la paginación y la búsqueda de maneja ágil.



Socios

Inicio / Socios

Mis Socios

Copy CSV Excel PDF Print

Search:

N.Socio	Cliente	Identificación	Teléfono	Editar	Borrar
1	Juan Zapata	0503247611	0995933081	EDITAR	DELETE
2	Genesis Velva	1850482819	0995933081	EDITAR	DELETE
3	RAUL ANDRES ZAPATA PEREZ	1805280375	0895546666	EDITAR	DELETE

Showing 1 to 3 of 3 entries

Previous 1 Next

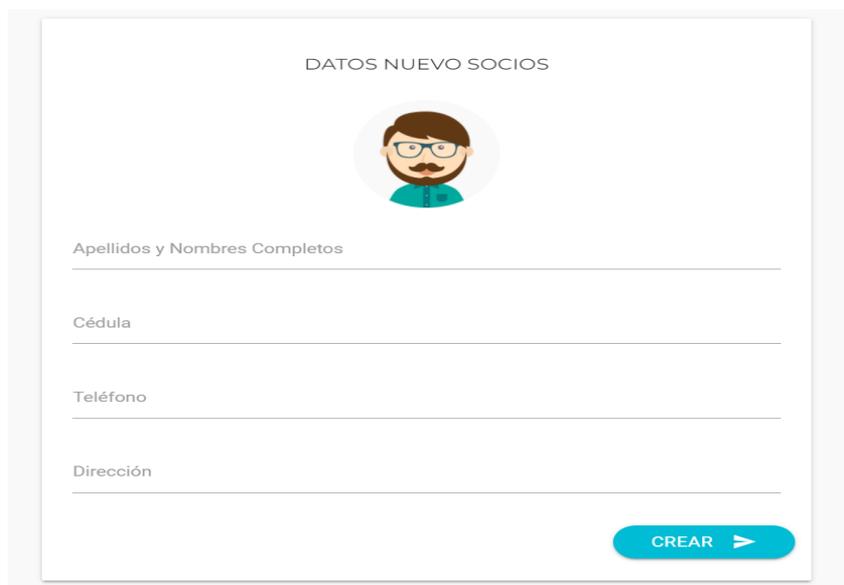
Figura 19. Formulario para listar socios.

Fuente: Elaboración por el autor.

El diseño del formulario de carga es uno de los más principales, ya permite llevar el control y seguimiento de los usuarios.

H07: Formulario para la creación de un socio.

La Figura 19 muestra el formulario de ingreso del socio el cual permite introducir los datos de forma estructurada en los campos destinadas para este propósito, para ser almacenados y procesados posteriormente.



DATOS NUEVO SOCIOS



Apellidos y Nombres Completos

Cédula

Teléfono

Dirección

CREAR >

Figura 20. Formulario para crear un nuevo socio.

Fuente: Elaboración por el autor.

Para realizar el registro de un nuevo socio se utiliza el modelo cuentas, para su respectiva validación de los datos identificados.

H08: Formulario para la edición del socio.

La Figura 20 muestra el formulario de modificación del socio el cual permite realizar cambio en los campos destinadas, para ser almacenados y procesados posteriormente.

The image shows a web interface for editing a user profile. At the top, it says 'Editar Socio: Juan Zapata'. Below that is a breadcrumb trail: 'Inicio / Editar Socio: Juan Zapata'. The main content area is a white box titled 'Editar Socio N° 1'. It contains four input fields with labels and values: 'Nombre de la Cuenta' with 'Juan Zapata', 'Identificación' with '0503247611', 'Teléfono' with '0995933081', and 'Dirección' with 'Ambato'. At the bottom right of the form is a blue button with the text 'ACTUALIZAR' and a right-pointing arrow.

Figura 21. Formulario para la actualización del socio.

Fuente: Elaboración por el autor.

Para la actualización de un socio se tiene varias funciones, con el fin de validar la integridad de los datos. Para obtener un socio específico se utiliza la función ‘getEditarCuenta’, y enviar los datos al formulario de editar socio.

3.3.4.2. Iteración 3

Se procede a desarrollar la tercera iteración conforme a la planificación.

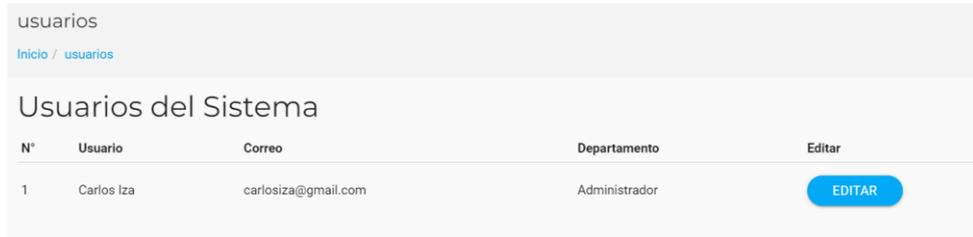
N.º Iteración	Historia de Usuario	Tiempo Estimado	
		Días	Horas
3	H09	8	32

Tabla 66. Tarea – Historias de usuario iteración 3.

Fuente: Elaborado por el autor.

H09: Gestión de usuario

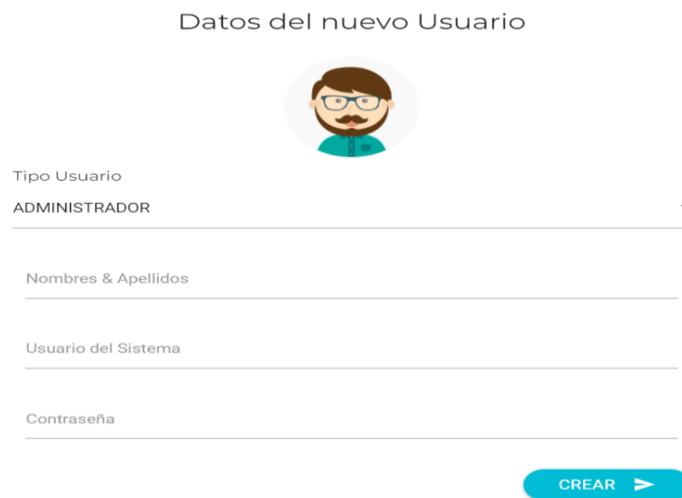
Como se muestra en Figura 21 se puede ver la lista de usuario que fueron creados en el sistema, la lista de usuario solo se puede ver si tiene permiso su rol correspondiente.



N°	Usuario	Correo	Departamento	Editar
1	Carlos Iza	carlosiza@gmail.com	Administrador	EDITAR

Figura 22. Formulario para la actualización del socio.

Fuente: Elaboración por el autor.



Datos del nuevo Usuario

Tipo Usuario
ADMINISTRADOR

Nombres & Apellidos

Usuario del Sistema

Contraseña

[CREAR >](#)

Figura 23. Formulario para la actualización del socio.

Fuente: Elaboración por el autor.

3.3.4.3. Iteración 4

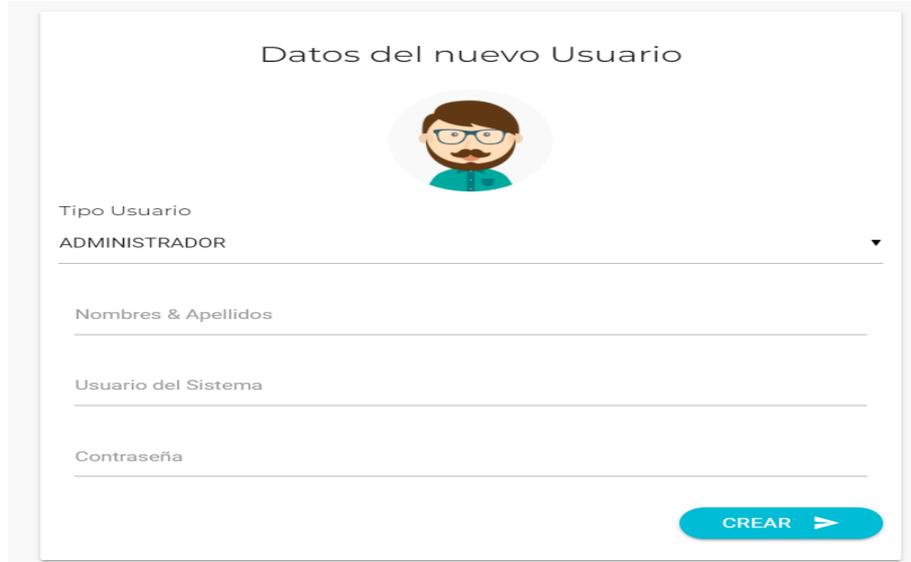
Se procede a desarrollar la cuarta iteración conforme a la planificación.

N.º Iteración	Historia de Usuario	Tiempo Estimado	
		Días	Horas
4	H10	8	32
	H11	8	32
	H12	8	32
	H13	6	24

Tabla 67. Tarea – Historias de usuario iteración 4.

Fuente: Elaborado por el autor.

H10: Formulario de registro de reunión.



Datos del nuevo Usuario



Tipo Usuario
ADMINISTRADOR

Nombres & Apellidos

Usuario del Sistema

Contraseña

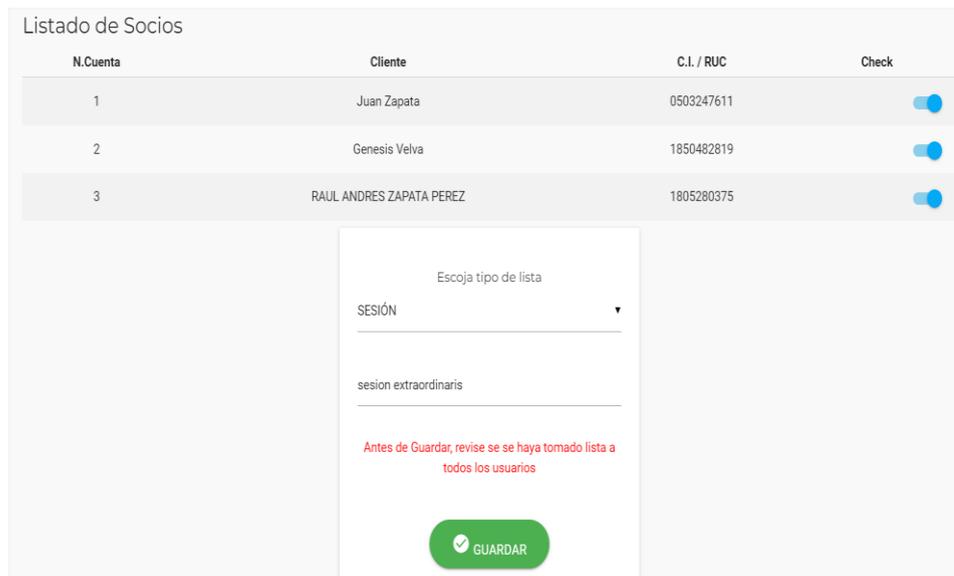
CREAR >

Figura 24. Formulario de registro de reunión.

Fuente: Elaboración por el autor.

H11: Formulario registrar la asistencia.

Para el registro de asistencia se procede a tomar lista a los usuarios correspondientes, así como motivo y la descripción.



Listado de Socios

N.Cuenta	Cliente	C.I. / RUC	Check
1	Juan Zapata	0503247611	<input checked="" type="checkbox"/>
2	Genesis Velva	1850482819	<input checked="" type="checkbox"/>
3	RAUL ANDRES ZAPATA PEREZ	1805280375	<input checked="" type="checkbox"/>

Escoja tipo de lista

SESIÓN

sesion extraordinaris

Antes de Guardar, revise se se haya tomado lista a todos los usuarios

GUARDAR

Figura 25. Formulario para el registro de asistencia.

Fuente: Elaborado por el autor

Mis Asistencias

Copy CSV Excel PDF Print

Search:

#	Tipo	Detalle	Fecha	Opción
1	SESION	Sesion 04 Enero 2022	Wed Feb 09 2022 21:51:16 GMT-0500 (Ecuador Time)	LISTA
2	SESION	Reunion ordinaria 10/02/2020/8:27	Thu Feb 10 2022 08:28:46 GMT-0500 (Ecuador Time)	LISTA
3	SESION		Tue Mar 08 2022 13:45:39 GMT-0500 (Ecuador Time)	LISTA
4	SESION	sesion 8 marzo 2022	Tue Mar 08 2022 15:23:12 GMT-0500 (Ecuador Time)	LISTA

Showing 1 to 4 of 4 entries

Previous 1 Next

Figura 26. Listado de reuniones.

Fuente: Elaboración por el autor.

Para la registrar la asistencia de los socios, procedemos a crear nuestro esquema la cual nos permite guardar en nuestra base de datos.

H12: Formulario de cobro de rubros.

Todos los clientes con rubros pendientes de pago aparecen como muestra la Figura 26.

Seleccionar Cliente

Inicio / Seleccionar Cliente

COBROS

Show 10

Search:

entries

N.Cuenta	Cliente	C.I. / RUC	Teléfono	Gestión
1	Juan Zapata	0503247611	0995933081	COBROS
2	Genesis Velva	1850482819	0995933081	COBROS
3	RAUL ANDRES ZAPATA PEREZ	1805280375	0895546666	COBROS

Showing 1 to 3 of 3 entries

Previous 1 Next

Figura 27. Formulario de socios por cobrar.

Fuente: Elaboración por el autor.

FACTURA

CLIENTE	IDENTIFICACIÓN	FECHA
RAUL ANDRES ZAPATA PEREZ	1805280375	03/24/2022
CUENTA	DIRECCIÓN	TELÉFONO
3	ANGAHUNA	0895546666

N	Producto	Valor	Excedente	Mora	Total	Borrar
1		5	0	0	5	

TOTAL

✓ REALIZAR COBRO

✓ IMPRIMIR

Figura 28. Formulario para el cobro del rubro por socio.

Fuente: Elaborado por el autor.

24/3/22, 12:18

FACTURA

RAUL ANDRES ZAPATA PE	CLIENTE
1805280375	IDENTIFICACIÓN
03/24/2022	FECHA
3	CUENTA
ANGAHUNA	DIRECCIÓN
0895546666	TELÉFONO

N	Producto	Valor	Excedente	Mora	Total	Borrar
1		5	0	0	5	<code>delete_forever</code>

TOTAL

Figura 29. Recibo de cobro.

Fuente: Elaborado por el autor.

Carga de los usuarios con rubros pendientes de cobro, mediante el método ‘GetSelCliente’.

H13: Formulario registro nuevo rubro.

El formulario de ingreso del nuevo rubro según la figura 29.

Crear Nuevo Rubro

Nombre de contribuciones, multas, aportes, etc.

Precio → Valor

CREAR RUBRO

Mis Rubros

#	Rubro	Precio	Opciones
1	Aporte 2022	\$ 5	BORRAR
2	COLABORACION	\$ 1	BORRAR

Figura 30. Formulario para ingresar un nuevo rubro.

Fuente: Elaboración por el autor.

3.3.4.5. Iteración 5

Se procede a desarrollar la quinta iteración conforme a la planificación.

N.º Iteración	Historia de Usuario	Tiempo Estimado	
		Días	Horas
5	H14	9	20
	H15	8	32

Tabla 68. Tarea – Historias de usuario iteración 5.

Fuente: Elaborado por el autor.

H14: Formulario para el registro de los documentos

Subir Documentos

Ingrese Nombre del Archivo

CARGAR ARCHIVO

Respositorio de Archivos

Figura 31. Formulario para guardar documentos.

Fuente: Elaborado por el autor

Mediante la función getRepo se procedió a obtener el documento y mostrar en la frond-end, con el objetivo de poder descargar y visualizar el documento.

H15: Gestión de reportes.

Sistema comunitario							
N.Socio	Identificación	Cliente	Tipo	Zona	Teléfono	Editar	Borrar
1	0503247611	JUAN ZAPATA PEREZ RAUL	SOCIO	ZONA1	0995933081	Editar	Delete
2	1852154683	frankin	DIRECTIVO	ZONA2	0985642315	Editar	Delete
3	1805280665	frankin	SOCIO	ZONA3	0865465123	Editar	Delete

Figura 32. Informe del listado de socios.

Fuente: Elaborado por el autor.

3.3.5 Fase 4: Producción

En la fase de producción se recalca los requerimientos, con el fin de probar todas las funcionalidades ya que en la metodología XP se considera muy importante, las historias de usuario serán de ayuda para tomar la decisión de poner o no en producción.

Prueba de aceptación	
Código: P01	Historia de usuario: H02
Nombre: Página de inicio de sistema administrativo	
Descripción: El usuario podrá visualizar las noticias más importantes de la comunidad, así también la cultura, gastronomía y los sitios turísticos.	
Condiciones de ejecución: ninguna	
Entrada/Proceso de Ejecución: El usuario seleccionara galería, gastronomía y contacto El usuario seleccionara ingresar al sistema administrativo	
Resultado esperado: El usuario visualizo el banner principal y las categorías con su respectivo imagen y descripción.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 69. Prueba de aceptación – Página de inicio administrativo.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P02	Historia de usuario: H03
Nombre: Administración del banner principal de la página de inicio	
Descripción: El usuario tendrá la facultad de administrar el banner de la página, así como su texto y la descripción.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario subirá nueva información El usuario podrá actualizar y eliminar según lo requiera Posterior el usuario presionara el botón guardar	
Resultado esperado: El usuario procedió a ingresar, modificar y eliminar una imagen, con su respectiva descripción para la modificación del banner principal.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 70. Prueba de aceptación – Administración del banner de la página de inicio.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P03	Historia de usuario: H03
Nombre: Administración del contenido de principal de la página de inicio	
Descripción: El usuario tendrá la facultad de administrar el contenido mediante imágenes y una breve descripción de la actividad al cual hace referencia.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario subirá nueva información El usuario podrá actualizar y eliminar según lo requiera Posterior el usuario presionara el botón guardar	
Resultado esperado: El usuario procedió añadir, modificar y eliminar la descripción con su respectivo logo e imagen principal para la modificación del contenido principal.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 71. Prueba de aceptación – Administración de la página de inicio.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P04	Historia de usuario: H03
Nombre: Administración de las categorías de la página de inicio	
Descripción: El usuario tendrá la facultad de administrar las categorías el cual se muestra en la página de inicio.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución:	

El usuario subirá nueva categoría El usuario podrá eliminar según lo requiera Posterior el usuario presionara el botón guardar
Resultado esperado: El usuario procedió a ingresar y eliminar una categoría.
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.

Tabla 72. Prueba de aceptación – Administración de la página de inicio.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P05	Historia de usuario: H03
Nombre: Administración de la página de inicio.	
Descripción: El usuario tendrá la facultad de administrar la información el cual se mostrará en la página de inicio.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario subirá nueva información El usuario podrá actualizar y eliminar según lo requiera Posterior el usuario presionara el botón actualizar	
Resultado esperado: El usuario pudo ingresar al formulario de administración del sistema de la página de inicio para la administración del contenido.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 73. Prueba de aceptación – Administración de la página de inicio.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P06	Historia de usuario: H04
Nombre: Ingreso al sistema web	
Descripción: El usuario podrá visualizar una página de bienvenida con la cual el usuario podrá ingresar al sistema	
Condiciones de ejecución: ninguna	
Entrada/Proceso de Ejecución: El usuario ingresa su usuario y la contraseña proporcionada El usuario presionara el botón ingresar	
Resultado esperado: El sistema valido el usuario y la contraseña, así como también espacio en blanco y el numero de caracteres en la contraseña, ingresando de forma satisfactorio al sistema administrativo.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 74. Prueba de aceptación – Ingreso al sistema web.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P07	Historia de usuario: H05
Nombre: Página de inicio dentro de sistema administrativo	
Descripción: El usuario podrá visualizar el resumen de las operaciones realizadas, así como el menú y el nombre de usuario.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario podrá seleccionar el ítem del menú de acuerdo con su perfil. El usuario podrá seleccionar la opción cerrar sesión.	
Resultado esperado: El usuario visualizo el resumen de las operaciones que se realizo en el sistema, así como el número de clientes, total de ingresos, total de gastos y las actividades.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 75. Prueba de aceptación – Página de inicio dentro del sistema.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P08	Historia de usuario: H06
Nombre: Lista de socios	
Descripción: El usuario podrá visualizar la lista de los socios activos	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario realiza las búsquedas en la tabla El usuario seleccionara nueva cuenta para crear un nuevo socio El usuario seleccionara editar para actualizar los datos del socio El usuario seleccionara Eliminar para dar de baja a un socio	
Resultado esperado: El usuario pudo listar a los socios y realizar búsquedas.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 76. Prueba de aceptación – Lista de socios.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P09	Historia de usuario: H07
Nombre: Creación del socio	
Descripción: Se visualizará un formulario donde el usuario podrá ingresar los datos requeridos	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario ingresara la información requerida para la creación del socio	

El usuario presionara el botón crear
Resultado esperado: El usuario valido los campos principales como la cedula, nombres y apellidos, dirección etc., validando los espacios en blanco y los campos requeridos para su posterior guardado.
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.

Tabla 77. Prueba de aceptación – Creación de usuario.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P10	Historia de usuario: H08
Nombre: Actualización de un socio	
Descripción: El usuario podrá actualizar los datos de un socio, cuando lo requiera	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario modificara la información del socio El usuario presionara el botón actualizar	
Resultado esperado: El usuario procedió a cargar los datos del socio para la verificación, validación y envío de los nuevos datos.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 78. Prueba de aceptación – Editar la información del socio.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P11	Historia de usuario: H09
Nombre: Registro de un nuevo usuario	
Descripción: La persona encargada registrara un nuevo usuario con su perfil correspondiente.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario ingresara la información requerida para la creación del usuario El usuario seleccionara el perfil de usuario El usuario seleccionara el botón crear	
Resultado esperado: El usuario valido los datos requeridos, así como los espacios en blanco para la creación de nuevo usuario del sistema.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 79. Prueba de aceptación – Registro de un nuevo usuario.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P12	Historia de usuario: H09
Nombre: Actualización del usuario	
Descripción: El usuario podrá actualizar la información de un usuario previamente	

ingresado
Condiciones de ejecución: Usuario iniciado sesión en el sistema
Entrada/Proceso de Ejecución: El usuario modifica la información requerida El usuario selecciona el botón modificar
Resultado esperado: El usuario procedió a cargar los datos en el formulario de actualización, para su posterior validación y guardado de nuevos datos.
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.

Tabla 80. Prueba de aceptación – Actualización del usuario.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P13	Historia de usuario: H09
Nombre: Inactivar un usuario	
Descripción: El usuario podrá dar de baja un usuario, con el fin de llevar una organización y control del sistema	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario presionara eliminar usuario El usuario podrá visualizar un mensaje de confirmación.	
Resultado esperado: El usuario procedió a inactivar al usuario, así como la observación del mensaje de confirmación.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 81. Prueba de aceptación – Inactivar un usuario.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P14	Historia de usuario: H10
Nombre: Registro de una nueva reunión	
Descripción: El usuario podrá registrar una nueva reunión con las validaciones respectivas	
Condiciones de ejecución: Usuario iniciado sesión en el sistema	
Entrada/Proceso de Ejecución: El usuario ingresa la información para la creación de una reunión El usuario selecciona el botón crear	
Resultado esperado: El usuario procedió a crear una nueva reunión validando los campos y los espacios en blancos, así como también el tipo de reunión para su respectiva creación.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 82. Prueba de aceptación – Registro de una nueva reunión.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P15	Historia de usuario: H11
Nombre: Registro de asistencia.	
Descripción: El usuario podrá seleccionar la reunión, así como a sus participantes de la reunión.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema.	
Entrada/Proceso de Ejecución: El usuario podrá seleccionar la reunión. El usuario podrá seleccionar a los socios participantes. El usuario selecciona el botón guardar.	
Resultado esperado: El usuario procedió a marcar y guardar a los socios que asistieron a la reunión previamente creada.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 83. Prueba de aceptación – Registro de asistencia.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P16	Historia de usuario: H12
Nombre: Cobro de rubros.	
Descripción: El usuario podrá realizar el cobro de los rubros que mantienen cada socio, ya sea por aporte o no registrar la asistencia.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema.	
Entrada/Proceso de Ejecución: El usuario seleccionara la opción cobros. El usuario seleccionara el botón realizar cobro. El usuario seleccionara el botón imprimir para desplegar el recibo.	
Resultado esperado: El usuario procedió a validar el listado de los ítems a pagar, así como la cantidad total o parcial.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 84. Prueba de aceptación – Cobro de rubros.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P17	Historia de usuario: H13
Nombre: Registro de un nuevo rubro.	
Descripción: El usuario podrá ingresar el nuevo rubro con las validaciones correspondientes.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema.	
Entrada/Proceso de Ejecución:	

El usuario ingresara la información en los campos establecidos. El usuario seleccionara el botón crear rubro.
Resultado esperado: El usuario procedió a ingresar un nuevo rubro con sus respectivas validaciones ya que estas se añaden para todos los socios.
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.

Tabla 85. Prueba de aceptación – Registro de un nuevo rubro.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P18	Historia de usuario: H14
Nombre: Registro de documentos.	
Descripción: El usuario podrá guardar y visualizar los documentos generados en la comunidad.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema.	
Entrada/Proceso de Ejecución: El usuario ingresa el nombre del archivo. El usuario selecciona el botón cargar archivo.	
Resultado esperado: El usuario procedió a cargar la documentación con su respectiva validación y para su respectivo guardado y visualización.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 86. Prueba de aceptación – Registro de documentos.

Fuente: Elaborado por el autor.

Prueba de aceptación	
Código: P19	Historia de usuario: H15
Nombre: Generación de informe.	
Descripción: La persona encargada del sistema administrativo podrá generar informes de los socios, reuniones, asistencia y cobro de los rubros.	
Condiciones de ejecución: Usuario iniciado sesión en el sistema.	
Entrada/Proceso de Ejecución: El usuario seleccionara en la opción generar informe de socios. El usuario seleccionara en la opción generar informe de reuniones. El usuario seleccionara en la opción generar informe de asistencia.	
Resultado esperado: El usuario visualizo la información del informe previamente seleccionado.	
Evaluación de la prueba: La prueba se realizó de forma satisfactoria.	

Tabla 87. Prueba de aceptación – Generación de informe.

Fuente: Elaborado por el autor.

3.3.6 Fase 5: Mantenimiento

El sistema cuenta con características acordes a los requerimientos, el desarrollador del sistema será la persona encargada de dar el mantenimiento, siguiendo la metodología XP se tendría que generar una nueva iteración para garantizar la integración y cumplimiento de los requerimientos establecidos.

CAPITULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Los cuadros comparativos de las tecnologías del front-end y el back-end, permitió elegir de una manera transparente las mejores herramientas para el desarrollo del sistema web para la gestión administrativa de la comunidad Angahuana Alto de la parroquia Santa Rosa. En base a los requerimientos del sistema se eligió la metodología XP la cual permitió un desarrollo ordenado del proyecto acorde a las historias de usuario, estas técnicas ayudaron a planificar y cumplir en forma ordenada las tareas conforme al cronograma establecido.
- Las reuniones con los directivos y la observación de los procesos permitieron conocer las necesidades que tiene la comunidad, convirtiendo la información recolectada en requerimientos funcionales para la construcción de la solución. Las herramientas aplicadas en el desarrollo permitieron crear una aplicación web responsiva y dinámica para el usuario, gracias a ello pueden utilizar en cualquier dispositivo llegando a satisfacer la experiencia del usuario.
- Una vez desarrollada la aplicación web se automatizó los procesos que mantiene actualmente la comunidad, brindando un mejor manejo de la

información y el seguimiento de los rubros generados. Dicho sistema permite promocionar la cultura, gastronomía y los lugares turísticos que la comunidad cuenta y es una fuente de ingreso de muchas familias. Con ello se indica que la página web alcanzó los objetivos propuestos, ya que la automatización de los procesos permite que los directivos y los socios lleven de una mejor manera las actividades que realiza la comunidad con las métricas generadas, tanto al inicio del mes y fin de año.

4.2 Recomendaciones

- Para el correcto funcionamiento se recomienda utilizar navegadores actualizados, ya que la compilación y la ejecución es mucho más veloz y asincrónica.
- Capacitar a los directivos en el manejo del sistema web, ya que ellos son los usuarios principales.
- Establecer políticas sobre el uso del sistema, ya que existe un administrador encargado de manejar y dar mantenimiento del sistema.
- Establecer proyectos futuros para solventar requerimientos que van a ir apareciendo con el uso de la tecnología.
- El cambio y la seguridad de las contraseñas debe ser administrada por el responsable del manejo del sistema.

BIBLIOGRAFIA

- [1] V. Calle y M. Hernández, Sistema web para el control de procesos administrativos en la escuela Particular de Educación Básica "Triunfadores del Futuro", Milagro: Universidad Agraria del Ecuador, 2020.
- [2] B. Zurita, Sistema web para la gestión académica y administrativa de la empresa de capacitación profesional DIENAV, Quito: Universidad Tecnológica Israel, 2020.
- [3] D. Hurtado, Aplicación web administrativa para reserva de servicios de transporte y envío de encomiendas para la empresa Romero y Asociados (AMBASEUR) de la ciudad de Ambato, Ambato: Universidad Técnica de Ambato, 2019.
- [4] O. Curichumbi, Sistema Automatizado para la gestión de las comunidades de la parroquia Salasaca de la ciudad de Pelileo, Ambato: Universidad Técnica de Ambato, 2020.
- [5] Luisa del Rosario Guachamín Chiluisa and DIRECTORA: Lcda. Mg. Mónica Narciza López Pazmiño, Artist, *La Gestión De Archivos Incide En La Imagen Institucionaldel Área Administrativa De La Unidad Metropolitana De Salud Centro De La Ciudad De Quito*. [Art]. Facultad de Ciencias Informáticas, 2013.
- [6] Y. P. Q. M. a. L. G. Z. García, Artist, *Facultad de Ciencias Informáticas, Software de Gestión Educativa*,. [Art]. 2013.
- [7] D. Martinez, Artist, *Actualizacion implementacion y capacitacion del sitio web del gobierno autonomo descentralizado parroquial rural de Atahualpa*. [Art]. Universidad Tecnica de Ambato, 2014.
- [8] M. Avella, «Importancia de los KPI en la Logística y su impacto en el servicio al cliente,» *Universidad Santiago de Cali. Especialización en Logística Integral*, pp. 1-19, 2019.
- [9] G. Juanamasta, N. Made, E. Hendrawati y W. W. Wahyuni, «The role of customer service through customer relationship management (CRM) to increase customer loyalty and good image,» *International Journal of Scientific and Technology Research* , vol. 8, nº 10, 2019.

- [10] M. Gómez, J. Cervantes y P. González, Fundamentos de ingeniería de software, México D.F: Universidad Autónoma Metropolitana, 2019.
- [11] B. Cuesta y J. Parra, «Modelo de desarrollo en proyectos de software libre y de código abierto [FOSS]: una mirada desde la teoría de la cooperación,» *Researchgate*, vol. 8, n° 20, pp. 1-11, 2014.
- [12] M. G. Díaz-Antón , M. A. Pérez, A. Grimmán y L. Mendoza, Artists, *PROPUESTA DE UNA METODOLOGÍA DE DESARROLLO DE*. [Art]. Universidad Simón Bolívar, 2010.
- [13] A. Sánchez, «La gestión de documentos como estrategia de innovación empresarial,» *Revista Venezolana de Información de Tecnología y Conocimiento*, vol. 11, n° 2, pp. 25-50, 2014.
- [14] J. Puebla, Artist, *Implementación de un Sistema de Gestión Documental, para la administración y gestión de documentos, a través de servicios y aplicaciones web, necesarias para la certificación de empresas en la Norma ISO 9001: 2008*. [Art]. Universidad San Francisco de Quito USFQ, 2015.
- [15] M. Arroyo, J. Rico, J. Barron y J. AGUIRRE, Artists, *Sistema de Administración Web de Evidencias Documentales para Organismos Acreditadores*. [Art]. Universidad Tecnológica del Suroeste de Guanajuato, 2015.
- [16] J. Fonseca y A. Martillo, Gestión administrativa y su efecto en la operatividad de un departamento de producción, Guayaquil: Grupo Editorial Compás, 2021.
- [17] X. Soledispa, J. Pionce y M. Sierra, «La gestión administrativa, factor clave para la productividad y competitividad de las microempresas,» *Dom. Cien*, vol. 8, n° 1, pp. 280-294, 2022.
- [18] G. Gorotiza y E. Romero, «El sistema de gestión de calidad con ISO 9001:2015 como estrategia para el mejoramiento de los procesos de la Comercializadora ITM,» *Pol. Con.*, vol. 6, n° 4, pp. 270-294, 2021.
- [19] F. Zabala, «La gestión de la calidad y el servicio al cliente como factor de competitividad en las empresas de servicios - Ecuador,» *Dom. Cien.*, vol. 6, n° 3, pp. 264-281, 2020.
- [20] G. Martínez, G. Camacho y D. Biancha, «Aplicaciones, Diseño de Framework

- Web Para el desarrollo dinámico de,» *Scientia et Technica*, nº 44, p. 6, 2010.
- [21] S. Gómez, Generación de formularios utilizando la tecnología Angular para interactuar con el core bancario mediante servicios REST, Quito, 2021.
- [22] L. Anchundia, Análisis comparativo de Tecnologías Front End Angular JS vs React Js, en el modelo de procesos para el desarrollo de aplicaciones Web., Los Rios .
- [23] G. Popoter, Rediseño de aplicaciones utilizando las tecnologías modernas para el desarrollo web en su parte Front-end., Valencia, 2015.
- [24] S. Lang, «Web Development with Jade,» de *Web Development with Jade*, Mumbai, Packt Publishing , 2014, pp. 15-18.
- [25] J. Ferrandis, Desarrollo back-end en .NET de una aplicación web para el reconocimiento de DNIs en exámenes manuscritos y su clasificación para su posterior distribución, valencia, 2021.
- [26] A. De la Torre, Introducción a NodeJs a través de Koans, Madrid, 2013.
- [27] K. Vallejo, Estudio comparativo de las tecnologías para el desarrollo del Back-end "NoseJs y Pho", Babahoyo, 2022.
- [28] B. Dayley, Node.js, MongoDB, and AngularJS Web Development, Addison-Wesley Professional, 5 jun 2014.
- [29] M. Pérez, Microsoft SQL Server 2008 R2. Motor de base de datos y administración, España: RC Libros, 2008.
- [30] J. S. y. J. Hernández, «SQL SERVER VS MySQL».
- [31] Á. Arias, «MySQL,» de *Bases de Datos con MySQL: 2ª Edición*, IT Campus Academy, 2015, pp. 39-40.
- [32] S. Combaudon, «MySQL 5.7: administración y optimización,» Barcelona, 2018, pp. 25-33.
- [33] P. Herrera, Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL, México, 2016.
- [34] M. Sharma, V. Sharma y M. Bundele, «Performance Analysis of RDBMS and No SQL Databases: PostgreSQL, MongoDB and Neo4j,» *3rd International Conference and Workshops on Recent Advances and Innovations in*

Engineering, pp. 22-25, 2018.

- [35] S. Bradshaw y K. Chodorow, *MongoDB: The Definitive Guide*, Beijing: O'Reilly, 2020.
- [36] M. Inc., «Mongo DB,» 2009. [En línea]. Available: <https://www.mongodb.com/es/what-is-mongodb>.
- [37] E. Maida y J. Pacienza, «Metodologías de desarrollo de software,» *Facultad de Química e Ingeniería "Fray Rogelio Bacon"*, vol. 1, pp. 19-20, 2015.
- [38] J. Zumba y C. León, «Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software,» *INNOVA Research Journal*, vol. 3, n° 10, pp. 20-33, 2018.
- [39] R. Figueroa, C. Solís y A. Cabrera, «Metdologías tradicionales vs metodología ágiles,» *Researchgate*, 2007.
- [40] B. Molina, H. Vite y J. Dávila, «Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,» *Espirales*, vol. 2, n° 17, pp. 114-121, 2018.
- [41] A. López, *Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software*, Universidad de Valladolid, 2018.
- [42] O. Tinoco Gómez, P. P. Rosales López y J. Salas Bacalla, *Artists, Criterios de selección de metodologías de desarrollo de software*. [Art]. Universidad Nacional Mayor de San Marcos, 2010.
- [43] M. Ortega y E. Camacho, *Uso de los modelos tradicionales y las metodologías ágiles aplicadas en la industria de software colombiano*, Colombia: Universidad Santiago de Cali, 2019.
- [44] L. Goncalves, «Scrum: The methodology to become more agile,» *Controlling & Management Review*, vol. 62, n° 4, 2018.
- [45] K. Schwaber, «SCRUM Development Process».
- [46] M. Cohn, «The Scrum Development Process».
- [47] W. i. Scrum?, «<http://dotnetjunkies.com/WebLog/darrell.norton/articles/3910.aspx> ,,» 23 Noviembre 2003. [En línea]. Available:

<http://dotnetjunkies.com/WebLog/darrell.norton/articles/3910.aspx> ,.

- [48] P. S. U. Nicholas R. Zuiderveld, «eXtreme Programming and SCRUM: A Comparative Analysis of Agile Methods».
- [49] C. Team, «Chrysler Goes to ‘Extremes,» *Distributed Computing*, pp. 23-28, 1998.
- [50] K. Beck, «Extreme Programming Explained: Embrace Change,» Addison-Wesley, Boston, 1999.
- [51] M. Fowler, «Refactoring,» [En línea]. Available: <https://www.refactoring.com/>.

ANEXOS

Codificación H02: Página de inicio del sistema administrativo

El diseño se los realiza en motor HTML jade, donde se carga el contenido desde la base de datos, ya que la página es administrable.

```
header.header-area.header-style-1.header-height-2
  .header-bottom.header-bottom-bg-color.sticky-bar
    .container
      .header-wrap.header-space-between.position-relative
        .logo.logo-width-1
          a(href='/')
            img(src='assets-front/imgs/theme/images.png' alt='logo')
        .header-nav.d-none.d-lg-flex
          .main-menu.main-menu-padding-1.main-menu-lh-2.d-none.d-lg-block.font-heading
            nav
              ul
                li
                  a(href='/') Inicio
                li
                  a(href='#')
                    | Servicios
                    i.fi-rs-angle-down
                    ul.sub-menu
                      each cat,i in categorias
                        li
                          a(href='/promocion-categoria?id='+cat._id)=cat.category
                li
                  a(href='/admin')
                    | Iniciar Sesión
```

Figura 33. Diseño de la cabecera de la página de inicio.

Fuente: Elaboración por el autor.

```
each pr,i in promo_category
  .col-xl-3.col-lg-4.col-md-6
    .product-cart-wrap.style-2.wow.animate__animated.animate__fadeInUp(data-wow-delay='0')
      .product-img-action-wrap
        .product-img
          a(href='/promocion?id='+pr._id)
            img(src=pr.path alt='contenido principal')
        .product-content-wrap
          .deals-content
            h2
              a(href='/promocion?id='+pr._id)=pr.titulo
            h2
              a(href='#'+pr._id)="Vigencia: "+moment(pr.fech_vigencia).format('MM/DD/YYYY')
          .product-rate-cover
            .product-rate.d-inline-block
              .product-rating(style='width: 100%')
        .center-promo
          .deals-countdown(data-countdown=moment(pr.fech_vigencia).format('MM/DD/YYYY'))
```

Figura 34. Carga del banner y el contenido principal.

Fuente: Elaboración por el autor.

```

getHome: function(req, res, next)
{
  console.log("Get home slider ..")
  var hoy=new Date().toISOString()
  var dte= hoy.split('T')
  console.log(dte)
  image_model.find(function(err, docs){
    promo_model.find({fech_vigencia: {$gte:dte[0]}}, function(err, promo){
      category_model.find(function(err, category){
        var _aux=[]
        promo.forEach(function(el){
          category.forEach(function(el2){
            if(el.id_category==el2._id){
              _aux.push({
                _id: el._id,
                id_category:el.id_category ,
                path: el.path,
                path_logo: el.path_logo,
                active: el.active,
                descuento: el.descuento,
                titulo: el.titulo,
                condiciones: el.condiciones,
                fech_vigencia: el.fech_vigencia,
                valido: el.valido,
                linkfb: el.linkfb,
                linkin: el.linkin,
                linkyt: el.linkyt,
                fecharegistro: el.fecharegistro,

```

Figura 35. Función principal para la carga de la página de inicio.

Fuente: Elaboración por el autor.

Codificación H03: Gestión para la administración de la página de inicio.

El diseño se los realiza en motor HTML jade, y la función principal para la administración del contenido principal con sus métodos get y post.

```

span(class="card-title")='Nueva Categoría'
form(method="post" action="/admin/category")
  div(class="input-field col s12 m12")
    input(type="text" name="category" class="validate")
    label="Nombre de Categoría"
  div(class="col s12")
    input(type="submit" value="Crear Categoría" class="waves-effect waves-light btn")
div(class="row col s12 m2")
div(class="row col s12 card")
  div(class="card-content")
    span(class="card-title")='Categorías Existentes'
    table
      thead
        th='N'
        th='Categoría'
        //th='Editar'
        th='Eliminar'
      tbody
        if(data)
          each c, i in data
            tr
              td=(i+1)
              td=c.category
              //td
              a(class="btn-floating blue" href="/editar-usuario?cod="+usuario._id)
                i(class="material-icons")="create"
              td
              a(class="btn-floating red" href="/borrar-usuario?cod="+usuario._id)
                i(class="material-icons")="delete"

```

Figura 36. Diseño para el formulario del contenido principal en HTML.

Fuente: Elaboración por el autor.

```

getCategoryProducts:function(req,res,next){
  category_model.find(function(err,docs){
    return res.render('dashboard/products/category',{user:req.user, data:docs})
  })
},
postCategoryProducts:function(req,res,next){
  console.log(req.body)
  var document= new category_model(req.body)
  document.save(function(err2){
    return res.redirect('/admin/category')
  })
},
postInactiveAdd:function(req,res,next){
  var document= new list_model({
    activo:0,
    featured:0,
    min:1,
    max:1,
    codigo:req.body.codigo,
    categoria:req.body.category,
    descripcion:req.body.descripcion
  })
  document.save(function(err2){
    return res.redirect('productos-inactivos')
  })
}

```

Figura 37. Función principal para la administración de la categoría

Fuente: Elaboración por el autor.

El diseño se los realiza en motor HTML jade, donde se carga el contenido desde la base de datos, ya que la página es administrable.

```

each pr,i in promo_category
  .col-xl-3.col-lg-4.col-md-6
  .product-cart-wrap.style-2.wow.animate__animated.animate__fadeInUp(data-wow-delay='0')
  .product-img-action-wrap
  .product-img
  a(href='/promocion?id='+pr._id)
  img(src=pr.path alt='lugares turisticos')
  .product-content-wrap
  .deals-content
  h2
  a(href='/promocion?id='+pr._id)=pr.titulo
  h2
  a(href='#'+pr._id)="Vigencia: "+moment(pr.fech_vigencia).format('MM/DD/YYYY')
  .product-rate-cover
  .product-rate.d-inline-block
  .product-rating(style='width: 100%')
  .center-promo
  .deals-countdown(data-countdown=moment(pr.fech_vigencia).format('MM/DD/YYYY'))

```

Figura 38. Diseño de para la administración del contenido principal.

Fuente: Elaboración por el autor.

```

postAddStock:function(req,res,next){
  product_model.find({codigo:req.body.codigo},function(err,docs){
    //console.log("Longitud: "+docs.length)
    list_model.find({_id:req.body._id},function(err,list){
      if(docs.length>0){
        disponibilidad={
          color:req.body.color,
          codigo_sys:req.body.codigo+"-"+req.body.talla,
          precio:req.body.precio,
          stock:req.body.stock,
          talla:req.body.talla,
          categoria:list[0].categoria,
          embalaje:req.body.embalaje
        }
      }
      product_model.findByIdAndUpdate({_id:docs[0]._id},{ $push:{"disponibilidad":disponibilidad}},function(err,docs1){
        //console.log(err1)
        //console.log(docs1)
        return res.redirect('/producto-add-stock?_id='+req.body._id+'&codigo='+req.body.codigo)
      })
    })
  }
}

```

Figura 39. Función para el guardado de contenido principal.

Fuente: Elaboración por el autor.

```

getInactiveProducts:function(req,res,next){
  console.log("Promo model..")
  promotion_model.find(function(err,docs){
    console.log(docs)
    category_model.find(function(err,category){
      var _aux=[]
      docs.forEach(function(el){
        category.forEach(function(el2){
          if(el.id_category==el2._id){
            _aux.push({
              _id: el._id,
              id_category:el.id_category ,
              path: el.path,
              path_logo: el.path_logo,
              active: el.active,
              descuento: el.descuento,
              titulo: el.titulo,
              condiciones: el.condiciones,
              fech_vigencia: el.fech_vigencia,
              valido: el.valido,
              linkfb: el.linkfb,
              linkin: el.linkin,
              linkyt: el.linkyt,
              fecharegistro: el.fecharegistro,
              category:el2.category
            })
          }
        })
      })
    })
  })
}

```

Figura 40. Función para dar de baja el contenido principal.

Fuente: Elaboración por el autor.

Codificación:

El diseño se los realiza en motor HTML jade, donde se carga el contenido desde la base de datos, ya que la página es administrable.

```

.row
  each pr,i in promo
    .col-x1-3.col-lg-4.col-md-6
      .product-cart-wrap.style-2.wow.animate__animated.animate__fadeInUp(data-wow-delay='0')
        .product-img-action-wrap
          .product-img
            a(href='/promocion?id='+pr._id)
              img(src=pr.path alt='Banner principal')
          .product-content-wrap
            .deals-content
              h2
                a(href='/promocion?id='+pr._id)=pr.titulo
              h2
                a(href='#'+pr._id)="Vigencia: "+moment(pr.fech_vigencia).format('MM/DD/YYYY')
            .product-rate.d-inline-block
              .product-rating(style='width: 100%')
          .center-date
            .deals-countdown(data-countdown=moment(pr.fech_vigencia).format('MM/DD/YYYY'))

```

Figura 41. Diseño del formulario del banner principal.

Fuente: Elaboración por el autor.

```

postSubir:function(req,res,next)
{
  console.log("***** Upload *****")
  console.log("Image logo");
  var subir = new formidable.IncomingForm();
  var ruta = "/images/logo/";
  var body;
  //console.log(req);
  subir.uploadDir = 'public\\images\\logo';
  subir.parse(req);
  subir.on('error', function (err) {
    console.log("HUbo un error");
    console.log(err);
  });
  subir.on('end', function (field, file) {
    console.log("Subido Correctamente");
    console.log('archivoimage:'+file);
  });
  subir.on('aborted', function(dat) {
    //console.log(dat);
  });
  subir.on('fileBegin', function (field, file) {
    console.log(file.mimetype)
    if (file.newFilename) {
      var ext;
      if (file.mimetype == 'image/png') {
        ext = ".png"
      }
    }
  });
}

```

Figura 42. Función para subir el banner principal.

Fuente: Elaboración por el autor.

```
postUpdateImage:function(req,res,next){
  console.log("***** SE EJECUTA EL API *****")
  id=req.body.id
  _index=req.body.index
  aux=[]
  console.log(req.body)
  list_model.find({_id:id},function(err,docs){
    console.log(err)
    console.log(docs)
    docs[0].image.forEach(function(el, index){
      if(index==_index)
      {
        aux.push(req.body.data)
      }
      else
      {
        aux.push(el)
      }
    })
    list_model.update({_id:id},{image:aux}, function(err1,docs1){
      console.log(err1)
      return res.json('success')
    })
  })
})
```

Figura 43. Función para dar de baja un banner.

Fuente: Elaboración por el autor

Codificación H04: Formulario de ingreso al sistema.

La codificación se los realiza en la función llamado Passport.js la cual se dedica a la validación de los datos mediante su librería ‘LocalStrategy’.

```
passport.use(new LocalStrategy({
  passReqToCallback : true
}, function(req, email, password, done){

  usuario_model.find({correo_usuario:email},function(err,docs){
    console.log(err)
    console.log(docs)
    if(!err)
    {
      console.log(err)
      return done(null, false);
    }
    else
    {
      if(docs.length>0){
        var user = docs[0];

        if(bcrypt.compareSync(password, user.password_usuario)){
          return done(null, {
            id: user._id,
            nombre : user.nombre_usuario,
            email : user.correo_usuario,
            tipo : user.tipo_usuario,
            id_grupo : user.agencia,
            grupo : user.grupo,
            token_google:user.token_google,
            foto:user.foto
          });
        }
        else
        {
          return done(null, false);
        }
      }
    }
  })
})
```

Figura 44. Método validación de datos

Fuente: Elaboración por el autor

Iteración 2

Codificación H05: Página de inicio dentro del sistema web.

Validar la activación del menú conforme al perfil, recupera y muestra la información del usuario, así como el resumen de la información más relevante.

```
// Card stats start
#card-stats
  h5.header.mt-0
  | SISTEMA COMUNITARIO | ANGAHUANA ALTO

  br
  br
  h5="Bienvenido"
  i.material-icons.background-round.mt-2 person
  h3=user.nombre
  br
  br
  .row
    .col.s12.m6.l3
    .col.s12.m6.l3
      .card.gradient-45deg-light-blue-cyan.gradient-shadow.min-height-100.white-text
      .padding-4
      .col.s7.m7
        i.material-icons.background-round.mt-5 perm_identity
        p SOCIOS
      .col.s5.m5.right-align
        input(style="font-size:20px;" ,id="cuenta_ap")

    .col.s12.m6.l3
      .card.gradient-45deg-amber-amber.gradient-shadow.min-height-100.white-text
      .padding-4
      .col.s7.m7
        i.material-icons.background-round.mt-5 attach_money
        p INGRESOS
      .col.s5.m5.right-align
        input(style="font-size:18px;" ,id="facturas")

  .row
```

Figura 45. Diseño del menú.

Fuente: Elaboración por el autor.

```
> JS usuario.js > [E] usuarioSchema
var mongoose = require ('mongoose');
  Schema = mongoose.Schema;

var usuarioSchema = new Schema({
  nombre_usuario: {type:String},
  correo_usuario: {type:String},
  tipo_usuario: {type:String},
  password_usuario: {type:String},
  celular_usuario: {type:String},
  token_google: {type:String},
  agencia: {type:String},
  grupo: {type:String},
  foto: {type:String},
  f_creacion: {type:Date, default: Date.now}
})

module.exports = mongoose.model('Usuario', usuarioSchema);
```

Figura 46. Modelo del usuario.

Fuente: Elaboración por el autor.

```

module.exports = {
  signIn : function(req, res, next){
    if(req.user){
      if(req.user.tipo == 1){
        return res.render('layout_material',{
          isAuthenticated : req.isAuthenticated(),
          user: req.user
        });
      }
      else{
        if(req.user.tipo == 2){
          return res.render('layout_material2',{
            isAuthenticated : req.isAuthenticated(),
            user: req.user
          });
        }
        else{
          console.log("Error auth")
          return res.render('mymessage', {titulo: "Se ha producido un error", mensaje: 'El usuario no existe'});
        }
      }
    }
    else{
      return res.render('public/home');
    }
  }
}

```

Figura 47. Función de control de roles.

Fuente: Elaboración por el autor.

```

getDash:function(req,res,next){
  var cuenta_model = require('../models/cuenta');
  var cuenta_model_ar = require('../models/cuentaar');
  cuenta_model.find(function(err,docs){
    cuenta_model_ar.find(function(err,cuenta_ar){
      factura_model.find(function(err, facturas){
        gasto_model.find(function(err,gasto){
          factura2_model.find(function(err, facturas2){
            gasto2_model.find(function(err,gasto2){
              if(err)
              {
                console.log(err)
                return res.redirect('/')
              }
              else
              {
                tfact=0
                tgasto=0
                if(facturas.length>0){
                  facturas.forEach(function(e1){
                    e1.detalle.forEach(function(e2){
                      tfact+=(parseFloat(e2.costo)+parseFloat(e2.exedente)+parseFloat(e2.mora))
                    })
                  })
                }
                if(gasto.length>0){
                  gasto.forEach(function(e1){
                    tgasto+=parseFloat(e1.valor)
                  })
                }
              }
            }
          }
        }
      }
    }
  )
}

```

Figura 48. Función para el resumen del dashboard.

Fuente: Elaboración por el autor.

Codificación H06: Formulario para la listar al socio.

El diseño del formulario de carga es uno de los más principales, ya permite llevar el control y seguimiento de los usuarios.

```
extends ../layout_material
block content
  h4="Mis Socios"
  div(class="col s12")
    table#dt(class="striped centered")
      thead
        //th="#"
        th="N.Socio"
        th="Cliente"
        th="Identificación"
        th="Teléfono"
        th="Editar"
        th="Borrar"
      tbody
        each cuenta, i in cuentas
          tr
            //td=(i+1)
            td=cuenta.medidor
            td=cuenta.nombre_cuenta
            td=cuenta.ruc
            td=cuenta.telefono
            td
              a(href="/cuentas-editar/?cod="+cuenta._id class="waves-effect waves-light btn")="Editar"
            td
              a(href="/cuentas-borrar?cod="+cuenta._id class="waves-effect waves-light btn red")="Delete"
    div(class="col s12 center" style="padding-top:25px; padding-bottom:30px;")
      br
      p
        a(href="/cuentas-nuevo" class="waves-effect waves-light btn-large gradient-45deg-green-teal")="Nueva cuenta"
  script.
    $(document).ready(function() {$('#dt').DataTable({
      dom: 'Bfrtip',
      buttons: ['copy', 'csv', 'excel', 'pdf', 'print']
    })
  })
})
```

Figura 49. Diseño del formulario para listar socios.

Fuente: Elaboración por el autor.

```
var mongoose = require ('mongoose');
Schema = mongoose.Schema;

var cuentaSchema = new Schema({
  nombre_cuenta: {type:String},
  tipo: {type:String},
  medidor: {type:Number, default:1},
  telefono: {type:String},
  metraje: {type:String},
  sector: {type:String},
  ruc: {type:String},
  direccion: {type:String},
  ciudad: {type:String},
  coordenadas: {type:String},
  propietario: {type:String},
  visible: {type:String},
  f_creacion: {type:Date, default: Date.now}
});

module.exports = mongoose.model('Cuenta', cuentaSchema);
```

Figura 50. Modelo del esquema socio.

Fuente: Elaboración por el autor.

```

var mongoose = require ('mongoose');
var ObjectId = require('mongoose').Types.ObjectId;
module.exports={
  getCuentas:function(req,res,next){
    var cuenta_model = require('./models/cuentaar');
    cuenta_model.find(function(err,docs){
      if(err)
      {
        console.log(err)
        return res.redirect('/')
      }
      else
      {
        return res.render('cuentasar/cuentas',{
          isAuthenticated : req.isAuthenticated(),
          user: req.user,
          title:"Socios",
          cuentas:docs
        })
      }
    });
  },
};

```

Figura 51. Función para la obtención de los socios.

Fuente: Elaboración por el autor.

Codificación H07: Formulario para la creación de un socio.

Para realizar el registro de un nuevo socio se utiliza el modelo cuentas, para su respectiva validación de los datos identificados.

```

var mongoose = require ('mongoose');
Schema = mongoose.Schema;

var cuentaSchema = new Schema({
  nombre_cuenta: {type:String},
  tipo: {type:String},
  medidor:{type:Number,default:1},
  telefono: {type:String},
  metraje: {type:String},
  sector: {type:String},
  ruc: {type:String},
  direccion: {type:String},
  ciudad: {type:String},
  coordenadas: {type:String},
  propietario: {type:String},
  visible: {type:String},
  f_creacion: {type:Date, default: Date.now}
});

module.exports = mongoose.model('Cuenta', cuentaSchema);

```

Figura 52. Modelo del esquema socio.

Fuente: Elaboración por el autor.

```

postCrearCuenta : function(req,res,next){
  var cuenta_model = require('../models/cuentaar');
  cuenta_model.find(function(err,docs){
    if(docs.length>0){
      var cuenta = new cuenta_model({
        nombre_cuenta: req.body.nombre_cuenta,
        ruc: req.body.ruc,
        telefono: req.body.telefono,
        direccion: req.body.direccion,
        medidor:docs.length+1
      });
      cuenta.save(function(error, docs){
        if(error)
        {
          console.log(error)
          return res.redirect('/')
        }
        else
        {
          return res.redirect('/cuentasar')
        }
      })
    }
    else{
      var cuenta = new cuenta_model(req.body);
      cuenta.save(function(error, docs){
        if(error)
        {
          console.log(error)
          return res.redirect('/')
        }
        else
        {
          //console.log(docs)
          return res.redirect('/cuentasar')
        }
      })
    }
  })
}

```

Figura 53. Función para el guardado de socio

Fuente: Elaboración por el autor.

Codificación H08: Formulario para la edición del socio.

Para la actualización de un socio tenemos varias funciones, con el fin de validar la integridad de los datos.

```

var mongoose = require ('mongoose');
Schema = mongoose.Schema;

var cuentaSchema = new Schema({
  nombre_cuenta: {type:String},
  tipo: {type:String},
  medidor:{type:Number,default:1},
  telefono: {type:String},
  metraje: {type:String},
  sector: {type:String},
  ruc: {type:String},
  direccion: {type:String},
  ciudad: {type:String},
  coordenadas: {type:String},
  propietario: {type:String},
  visible: {type:String},
  f_creacion: {type:Date, default: Date.now}
})

module.exports = mongoose.model('Cuenta', cuentaSchema);

```

Figura 54. Modelo del esquema socio.

Fuente: Elaboración por el autor.

```

getEditarCuenta: function(req,res,next){
  var cuenta_model = require('../models/cuentaar');
  var ObjectId = require('mongoose').Types.ObjectId;
  cuenta_model.find({_id:ObjectId(req.query.cod)}),function(err,docs){
    console.log(docs)
    if(err)
    {
      console.log(err);
      return res.redirect('/')
    }
    else
    {
      return res.render('cuentasar/editar',{
        isAuthenticated : req.isAuthenticated(),
        user: req.user,
        title:"Editar Socio: "+docs[0].nombre_cuenta,
        cuenta: docs[0]
      })
    }
  })
},

```

Figura 55. Función para obtener un socio específico.

Fuente: Elaboración por el autor.

```

postEditarCuenta: function(req,res,next){
  var cuenta_model = require('../models/cuentaar');
  _id=req.body._id;
  delete req.body._id
  if(req.body.email)
  {
    b=req.body.email.split(",")
    var c=[]
    b.forEach(function(e1){
      c.push(e1)
    })
    req.body.email=c
  }
  cuenta_model.update({_id:_id}, { $set:req.body}, function(err, docs){
    if(err)
    {
      console.log(err)
      return res.redirect('/')
    }
    else
    {
      console.log(docs);
      return res.redirect('/cuentasar')
    }
  })
}

```

Figura 56. Función para actualizar el socio.

Fuente: Elaboración por el autor.

Codificación Iteración 3

Para el listado de usuario se utiliza el modelo usuario.

```
var mongoose = require ('mongoose');
    Schema = mongoose.Schema;

var usuarioSchema = new Schema({
  nombre_usuario: {type:String},
  correo_usuario: {type:String},
  tipo_usuario: {type:String},
  password_usuario: {type:String},
  celular_usuario: {type:String},
  token_google: {type:String},
  agencia: {type:String},
  grupo: {type:String},
  foto:{type:String},
  f_creacion: {type:Date, default: Date.now}
})

module.exports = mongoose.model('Usuario', usuarioSchema);
```

Figura 57. Modelo de esquema usuario

Fuente: Elaboración por el autor.

```
module.exports={
  getUsuario:function(req,res,next){
    usuario_model.find(function(err,docs){
      if(err)
      {
        console.log(err)
        return res.redirect('/')
      }
      else
      {
        return res.render('usuarios/usuario',{
          isAuthenticated : req.isAuthenticated(),
          user: req.user,
          title:"usuarios",
          usuarios:docs
        })
      }
    });
  },
},
```

Figura 58. Función para la obtención de todos los usuarios.

Fuente: Elaboración por el autor.

```

postCrearUsuario : function(req,res,next){
  _aux=req.body
  console.log("Save User")
  console.log(_aux)
  bcrypt.hash(_aux.password_usuario, 10, function(err, pswd) {
    var user={
      nombre_usuario: _aux.nombre_usuario,
      correo_usuario: _aux.correo_usuario,
      password_usuario: pswd,
      tipo_usuario: _aux.tipo_usuario,
      agencia: "Ambato"
    }

    var usuario = new usuario_model(user);
    usuario.save(function(error, docs){
      if(error)
      {
        console.log(error)
        return res.redirect('/')
      }
      else
      {
        return res.redirect('/usuario')
      }
    })
  });
}

```

Figura 59. Función para guardar un nuevo usuario.

Fuente: Elaboración por el autor

```

getEditarusuario: function(req,res,next){
  var ObjectId = require('mongoose').Types.ObjectId;
  usuario_model.find({_id:ObjectId(req.query.cod)},function(err,docs){
    console.log(docs)
    if(err)
    {
      console.log(err);
      return res.redirect('/')
    }
    else
    {
      return res.render('usuarios/editar',{
        isAuthenticated : req.isAuthenticated(),
        user: req.user,
        title:"Editar usuario: "+docs[0].descripcion,
        usuario: docs[0]
      })
    }
  })
},

```

Figura 60. Función para editar un usuario

Fuente: Elaboración por el autor

```

postEditarusuario: function(req,res,next){
  _id=req.body._id;
  usuario_model.update({ _id: _id }, { $set:req.body}, function(err, docs){
    if(err)
    {
      console.log(err)
      return res.redirect('/')
    }
    else
    {
      console.log(docs);
      return res.redirect('/usuario')
    }
  })
}

```

Figura 61. Función para editar el usuario.

Fuente: Elaboración por el autor

Iteración 4

Codificación H11: Formulario registrar la asistencia.

Para la registrar la asistencia de los socios, se procede a crear el esquema, el cual permite guardar en nuestra base de datos.

```
var mongoose = require ('mongoose');
    Schema = mongoose.Schema;

var listaapSchema = new Schema({
  lista: {type:Object},
  tipo: {type:String},
  detalle: {type:String},
  f_creacion: {type:Date, default: Date.now}
})

module.exports = mongoose.model('Listaap', listaapSchema);
```

Figura 62. Modelo para el registro de asistencia.

Fuente: Elaboración por el autor.

```
getAsistencia:function(req,res,next){
  listaap_model.find(function(err,docs){
    console.log("Mi Asistencia ...")
    console.log(docs)
    if(err)
    {
      console.log(err)
      return res.redirect('/')
    }
    else
    {
      return res.render('listado/mi-asistencia',{
        isAuthenticated : req.isAuthenticated(),
        user: req.user,
        title:"Asistencias",
        asistencias:docs
      })
    }
  });
},
```

Figura 63. Función para la obtención de las asistencias de los socios

Fuente: Elaboración por el autor

```

/*****GUARDA mingas y sesiones*****/
postListaAp: function(req,res,next){
  console.log("Lista Back")
  data=JSON.parse(req.body.data)
  var document = new listaap_model({
    lista:data,
    tipo:req.body.lista,
    detalle:req.body.detalle
  });
  document.save(function(err,docs){
    return res.json("ok")
  })
}

```

Figura 64. Función para guardar la asistencia

Fuente: Elaboración por el autor

Codificación H12: Formulario de cobro de rubros.

Carga de los usuarios con rubros pendientes de cobro, mediante el método 'GetSelCliente'.

```

getSelCliente: function(req,res,next)
{
  cuenta_model.find(function(err,docs){
    if(err)
    {
      console.log(err);
      return res.redirect('/')
    }
    else
    {
      return res.render('facturas/seleccionar_cliente',{
        isAuthenticated : req.isAuthenticated(),
        user: req.user,
        title:"Seleccionar Cliente",
        cuentas:docs
      })
    }
  })
}
}

```

Figura 65. Función para obtener el detalle de rubros del cliente.

Fuente: Elaborado por el autor.

```

postFacturar: function(req,res,next){
  //data=JSON.parse(req.body.data)
  console.log("Factura Back")
  console.log(req.body.detalle)
  contenido=JSON.parse(req.body.detalle)
  //console.log(contenido)
  var documento= new factura_model(
    {
      cliente:req.body.cliente,
      estado:"Facturado",
      detalle:contenido,
    }
  )
  documento.save(function(err, dtc){
    if(err)
    {
      console.log(err)
      return res.redirect('/')
    }
    else
    {
      return res.json("ok")
    }
  })
  //aux=rellenar_factura(data,res)
},

```

Figura 66. Función para guardar la factura.

Fuente: Elaborado por el autor.

```

data_product=[]
products.forEach(function(e1){
  data_product.push({
    _id:"ninguno",
    nombre_producto:e1.nombre_producto,
    tipo_precio:"precio",
    costo:e1.costo,
    exedente:0,
    mora:0
  })
})
listadoap.forEach(function(e1){
  data_product.push({
    _id:"ninguno",
    nombre_producto:e1.detalle,
    tipo_precio:"precio",
    costo:e1.multa.toString(),
    exedente:0,
    mora:0
  })
})
catastro_model.find({"cuenta._id":ObjectId(req.query.cod)},function(err,catastro){
  if(catastro.length==0){
    console.log("no existe catstros")
    console.log(req.query.cod)
  }else{
    console.log(catastro)
    var exedente=0
    var fecha_conteo = new Date(2019, 07, 01); //31 de diciembre de 2015
    var fecha_actual = new Date(); //30 de noviembre de 2014

    if(fecha_actual > fecha_conteo){mora=(catastro[0].terreno)*0.00002}
    else{mora=0}

    data_product.push({
      _id:"ninguno",

```

Figura 67. Función para detallar los rubros

Fuente: Elaborado por el autor.

```

getFactura: function(req,res,next){
  //console.log(req.query)
  var ObjectId = require('mongoose').Types.ObjectId;
  cuenta_model.find({_id:ObjectId(req.query.cod)},function(err,docs){
    if(err)
    {
      console.log(err);
      return res.redirect('/')
    }
    else
    {
      listaap_model.find(function(err,list){
        multa=0
        detalle=0
        listadoap=[]
        list.forEach(function(el){
          el.lista.forEach(function(el2){
            //console.log(el2)
            if(el2.cuenta==docs[0].medidor && el2.check=="1"){
              if(el.tipo=="SESION"){
                listadoap.push({
                  multa:5,
                  detalle:el.detalle
                })
              }
              else{
                listadoap.push({
                  multa:10,
                  detalle:el.detalle
                })
              }
            }
          })
          multa=1
        })
      })
    }
  })
}

```

Figura 68. Función para obtener la factura

Fuente: Elaborado por el autor.

Codificación H13: Formulario registro nuevo rubro.

Para la creación de un nuevo rubro se basa en el siguiente modelo.

```

var mongoose = require('mongoose');
Schema = mongoose.Schema;

var productoSchema = new Schema({
  nombre_producto: {type:String},
  precio_min: {type:String},
  precio_max: {type:String},
  costo: {type:String},
  tipo_precio:{type:String},
  f_creacion: {type:Date, default: Date.now},
  f_creacion: {type:Boolean, default:true}
})

module.exports = mongoose.model('Producto', productoSchema);

```

Figura 69. Modelo del nuevo rubro

Fuente: Elaboración por el autor

```

getProducto : function(req,res,next){
  productos_model.find(function(err,docs){
    if(err)
    {
      console.log(err);
      return res.redirect('/')
    }
    else
    {
      return res.render('productos/nuevo',{
        isAuthenticated : req.isAuthenticated(),
        user: req.user,
        title:"Productos",
        productos:docs
      })
    }
  })
}

```

Figura 70. Función para obtener los rubros.

Fuente: Elaboración por el autor.

```

postProducto : function(req,res,next){
  var producto = new productos_model(req.body);
  producto.save(function(err,docs){
    if(err)
    {
      console.log(err);
      return res.redirect('/')
    }
    else
    {
      return res.redirect('/productos')
    }
  })
}

```

Figura 71. Función para guardar el rubro.

Fuente: Elaboración por el autor

Codificación Iteración 5

Mediante la función getRepo se procedió a obtener el documento y mostrar en la frond-end, con el objetivo de poder descargar y visualizar el documento.

```

var mongoose = require ('mongoose');
var ObjectId = require('mongoose').Types.ObjectId;
var repositorio_model = require('../models/repositorio');
module.exports={
  getRepo:function(req,res,next){
    repositorio_model.find(function(err,docs){
      if(err)
      {
        console.log(err)
        return res.redirect('/')
      }
      else
      {
        return res.render('repositorio/subir',{
          isAuthenticated : req.isAuthenticated(),
          user: req.user,
          title:"Repositorio",
          repositorio:docs
        })
      }
    });
  }
}

```

Figura 72. Función para la obtención del documento.

Fuente: Elaboración por el autor.