



Diseño e implementación de una plataforma web para la difusión de información y servicios de la Iglesia Camino Nuevo de Estados Unidos.

Chinlle Peralta, Edison Steven y Cobacango Lima, Michael Armando

Departamento de Ciencias de la Computación

Carrera de Software

Trabajo de integración curricular, previo a la obtención del título de Ingeniero en Software

Ing. Lascano, Jorge Edison, PhD.

15 de agosto del 2023



Plagiarism report

MIC_Report_Chinlle_Cobacango_2023...

Scan details

Scan time:
August 14th, 2023 at 19:36 UTC

Total Pages:
129

Total Words:
32195



Plagiarism Detection



Types of plagiarism		Words
● Identical	0.8%	257
● Minor Changes	0%	0
● Paraphrased	0%	0
● Omitted Words	17%	5460

AI Content Detection



Text coverage
● AI text
● Human text



**Departamento de Ciencias de la Computación
Carrera de Software**

Certificación

Certifico que el trabajo de integración curricular: "**Diseño e implementación de una plataforma web para la difusión de información y servicios de la Iglesia Camino Nuevo de Estados Unidos**" fue realizado por los señores **Chinle Peralta, Edison Steven y Cobacango Lima, Michael Armando**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 15 de agosto del 2023.



.....
Lascano, Jorge Edison

C. C.: 1710893114



**Departamento de Ciencias de la Computación
Carrera de Software**

Responsabilidad de Autoría

Nosotros, **Chinlle Peralta, Edison Steven y Cobacango Lima, Michael Armando**, con cédulas de ciudadanía N° **1725043887** y **1722066956**, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Diseño e implementación de una plataforma web para la difusión de información y servicios de la Iglesia Camino Nuevo de Estados Unidos** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 15 de agosto de 2023

.....
Chinlle Peralta, Edison Steven

C.C.: 1725043887

.....
Cobacango Lima, Michael Armando

C.C.: 1722066956



Departamento de Ciencias de la Computación
Carrera de Software

Autorización de Publicación

Nosotros **Chinlle Peralta, Edison Steven y Cobacango Lima, Michael Armando**, con cédulas de ciudadanía N° **1725043887** y **1722066956**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Diseño e implementación de una plataforma web para la difusión de información y servicios de la Iglesia Camino Nuevo de Estados Unidos** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 15 de agosto de 2023

.....
Chinlle Peralta, Edison Steven

C.C.: 1725043887

.....
Cobacango Lima, Michael Armando

C.C.: 1722066956

Dedicatoria

Dedico este trabajo a Dios por darme las fuerzas para seguir adelante y no abandonarme en ningún momento, por ayudarme a cumplir una meta más y permitirme cumplir mis sueños, fuente de sabiduría y guía constante en mi camino.

A mis padres por el amor y apoyo incondicional que me han brindado en todas las etapas de mi vida, por su invaluable sacrificio y por ser mis ejemplos que seguir.

A mis hermanos por ser mis compañeros de risas y de llantos, confidentes en las adversidades y amigos de vida, por ser mis soportes en todo aspecto y por no dejarme decaer nunca.

A mi sobrino que es la luz que ilumina mi camino, por ser la alegría, ternura y por llegar en el momento justo cuando más lo necesitaba.

A toda mi familia y seres queridos por enseñarme la importancia de la unión, por darme las palabras de ánimo y gestos de aliento para ir siempre hacia adelante.

A todos mis compañeros que han formado parte de mi vida universitaria que, con sus consejos, locuras y conocimientos me han ayudado a culminar con esta etapa en mi vida.

Finalmente, a aquellas personas que ya no están entre nosotros, su legado vive en cada logro alcanzado y su memoria sigue siendo fuente de inspiración. Este trabajo es un tributo a su influencia perdurable en mi vida.

Edison Steven Chinlle Peralta

Dedico este trabajo a mis padres Alfonso y Norma quienes, con su paciencia, esfuerzo y sobre todo amor han sido mi fuente de inspiración para cumplir un sueño más. A mis hermanas Marjorie, Shirley y Daphne por su cariño y apoyo incondicional. A mis adorables sobrinos Gabriel y Zaid, quienes llenan mi vida de alegría y felicidad. A Cristian por sus sabios consejos, y aliento constante para alcanzar este logro.

Michael Armando Cobacango Lima

Agradecimiento

Quiero expresar mi más profundo y sincero agradecimiento a la Universidad de las Fuerzas Armadas "ESPE" por brindarme la invaluable oportunidad académica. Mi gratitud se extiende a mis dedicados docentes, quienes han guiado mi camino universitario con su conocimiento.

A mi padre Marcelo, por ser el guía de mis pasos y por enseñarme que "si se hace algo, hay que hacerlo bien", por enseñarme que con fe, constancia y trabajo duro todo es posible, además que la educación es la única manera de salir adelante. A mi madre Gloria, por enseñarme que, a pesar de los problemas y contratiempos, siempre hay un motivo para ser feliz y que después de la tormenta siempre viene la calma, por mostrarme el valor del amor.

A mis hermanos por ser mi compañía, mis cómplices y mis mejores amigos, en especial a Estefany por ser mi luz entre la oscuridad, por enseñarme que el amor de hermanos va más allá de lo imaginable y que una hermana mayor en ocasiones cumple el papel de segunda madre siempre velando por tu felicidad.

A mi sobrino Benjamín, que con su energía inagotable y sonrisa han iluminado mis días recordándome siempre la importancia de la alegría en cada paso de la vida. Tu amor y cariño han sido apoyo invaluable y aunque quizás no entiendas la magnitud de este logro, tu presencia ha sido inspiración y motivación para alcanzarlo.

A mi familia y seres queridos, que con su muestra de sacrificio y dedicación han sembrado en mí el valor de la perseverancia y el sentimiento de cercanía a pesar de la distancia, y a mis amigos que me han brindado momentos de alegría y consuelo.

Finalmente, a mi compañero y amigo de tesis Michael, cuya colaboración y amistad han hecho posible este logro compartido, y a mi tutor Edison, quien con su orientación y amor por la enseñanza han contribuido a esta invaluable experiencia.

Edison Steven Chinlle Peralta

Quiero expresar mi más profundo agradecimiento a todas las personas que me apoyaron en el transcurso de esta etapa importante en mi vida, y en el desarrollo de este proyecto.

A mis padres, Alfonso y Norma, les agradezco por su apoyo inquebrantable, por haberme transmitido sabiduría y valores que me han guiado en cada paso, y por enseñarme a siempre perseverar y seguir adelante.

A mis hermanas Marjorie, Shirley y Daphne, les estoy agradecido por su cariño constante y su apoyo incondicional.

A mis adorables sobrinos, Gabriel y Zaid, les agradezco por llenar mi vida de alegría y felicidad con su presencia. Su inocencia y entusiasmo son una fuente constante de inspiración.

A mi amigo Cristian, mi gratitud es profunda por tus consejos sabios y aliento constante. Tu presencia ha sido un faro de luz en los momentos de duda.

A mi compañero y amigo de tesis Steven por toda su dedicación y arduo trabajo en cada fase de este proyecto que han enriquecido enormemente el resultado final, y a nuestro director Ing. Jorge Edison Lascano, PhD. Por todo el tiempo brindado, por desempeñar un papel fundamental siendo la guía en el proyecto y sobre todo por ser un gran profesional.

A mi amigo Junior Jurado, cuyo constante apoyo y colaboración han sido un pilar fundamental durante los momentos más desafiantes de mi trayectoria académica y laboral.

A mis compañeros y amigos de la Universidad, quienes han compartido conmigo risas, estudios y desafíos.

Michael Armando Cobacango Lima

Índice de contenido

Dedicatoria.....	6
Agradecimiento.....	8
Resumen	22
Abstract.....	23
Capítulo I	24
Introducción	24
Antecedentes	24
Planteamiento del problema.....	26
Justificación.....	28
Objetivos	28
Objetivo General.....	28
Objetivos Específicos.....	29
Alcance	29
Metodología	30
Capitulo II.....	34
Marco Teórico	34
Religión	34
Tipos de Religión	36
Tipos de Religión Cristiana	38
Religión Cristiana Pentecostal	39
Software para la gestión de iglesias	43

Aplicación Web.....	44
Tecnologías para desarrollo web	45
Cliente-Servidor	46
API.....	47
Representational State Transfer (REST).....	47
API REST	47
JavaScript Object Notation (JSON).....	48
HTML.....	48
CSS	49
JavaScript.....	49
Backend.....	50
Frontend	51
Framework.....	53
Node.js	54
Express.js	54
Vue.js.....	55
Variables de entorno dotenv.....	56
Base de Datos No SQL.....	56
Git.....	58
IaaS (Infraestructura As A Service).....	59
Herramientas para desarrollo web.....	59

	12
MongoDB Atlas	59
Entorno de Desarrollo Integrado (IDE)	60
Visual Studio Code	61
Postman	61
Github	62
Jira.....	62
AWS.....	63
AWS EC2.....	63
Ubuntu Server.....	64
Apache2	64
GoDaddy.....	65
Documentación y proceso de desarrollo de software.....	66
Diagramas de Software UML	66
Pruebas de Aceptación de Usuario	67
Despliegue de software.....	67
Capitulo III.....	68
Análisis y Planificación del Desarrollo	68
Obtención de Requerimientos de Software	68
Análisis de los Requerimientos y Creación del Product Backlog	69
Creación de los Sprint Backlog	73
Plan de Pruebas de Aceptación del Usuario	82

Capitulo IV	82
Desarrollo Del Sistema	82
Arquitectura De la Aplicación	83
Preparación Del Ambiente De Desarrollo	85
Desarrollo de los Sprints	86
Sprint 1: Gestión De La Información.....	96
Daily Meetings	101
Aceptación del usuario.....	102
Resultados.....	103
Sprint Review.....	113
Sprint 2: Cambio del Idioma del Sitio.....	113
Daily Meetings	115
Aceptación del usuario.....	117
Resultados.....	117
Sprint Review.....	121
Sprint 3: Gestión del Contenido Multimedia.....	122
Daily Meetings	125
Aceptación del usuario.....	126
Resultados.....	127
Sprint Review.....	131
Sprint 4: Gestión de Redes Sociales	132

Daily Meetings	133
Aceptación del usuario.....	134
Resultados.....	135
Sprint Review.....	140
Sprint 5: Gestión de Eventos	140
Daily Meetings	142
Aceptación del usuario.....	143
Resultados.....	143
Sprint Review.....	147
Sprint 6: Gestión de Servicios	148
Daily Meetings	149
Aceptación del usuario.....	150
Resultados.....	150
Sprint Review.....	153
Sprint 7: Registro y Supervisión de Peticiones de Oración	153
Daily Meetings	155
Aceptación del usuario.....	157
Resultados.....	157
Sprint Review.....	160
Sprint 8: Gestión de Grupos de Vida	161
Daily Meetings	162

Aceptación del usuario.....	163
Resultados.....	163
Sprint Review.....	167
Sprint 9: Recepción de Donaciones.....	167
Daily Meetings	170
Aceptación del usuario.....	172
Resultados.....	172
Sprint Review.....	179
Capítulo V	181
Conclusiones.....	181
Recomendaciones.....	182
Trabajos Futuros	183
Referencias.....	184
Apéndices	191

Índice de tablas

Tabla 1	Tamaño y proyección de crecimiento de los principales grupos religiosos	37
Tabla 2	Historias de usuario	70
Tabla 3	Product Backlog del Proyecto	73
Tabla 4	Sprint Backlog Del Sprint 1 “Gestión De Información De La Iglesia Camino Nuevo”	74
Tabla 5	Sprint Backlog Del Sprint 2 “Cambio del Idioma del sitio”	74
Tabla 6	Sprint Backlog Del Sprint 3 “Gestión Del Contenido Multimedia”	75
Tabla 7	Sprint Backlog Del Sprint 4 “Gestión De Redes Sociales”	76
Tabla 8	Sprint Backlog Del Sprint 5 “Gestión De Eventos”	76
Tabla 9	Sprint Backlog Del Sprint 6 “Gestión De Servicios”	77
Tabla 10	Sprint Backlog Del Sprint 7 “Registro y Supervisión De Peticiones De Oración”	78
Tabla 11	Sprint Backlog Del Sprint 8 “Gestión De Grupos De Vida”	79
Tabla 12	Sprint Backlog Del Sprint 9 “Recepción De Donaciones”	79
Tabla 13	Resumen de actividades planificadas en las daily meetings del sprint 1	101
Tabla 14	Listado de endpoints generados en el sprint 1	103
Tabla 15	Componentes y vistas generadas en el sprint 1	106
Tabla 16	Resumen de actividades planificadas en las daily meetings del sprint 2	116
Tabla 17	Resumen de actividades planificadas en las daily meetings del sprint 3	125
Tabla 18	Sugerencias para el sprint 3.....	127
Tabla 19	Listado de endpoints generados en el sprint 3	128
Tabla 20	Resumen de actividades planificadas en las daily meetings del sprint 4	133
Tabla 21	Sugerencias para el sprint 4.....	134
Tabla 22	Listado de endpoints generados en el sprint 4	135
Tabla 23	Resumen de actividades planificadas en las daily meetings del sprint 5	142
Tabla 24	Listado de endpoints generados en el sprint 5	143

Tabla 25 Resumen de actividades planificadas en las daily meetings del sprint 6	149
Tabla 26 Listado de endpoints generados en el sprint 6	151
Tabla 27 Resumen de actividades planificadas en las daily meetings del sprint 7	156
Tabla 28 Listado de endpoints generados en el sprint 7	158
Tabla 29 Resumen de actividades planificadas en las daily meetings del sprint 8	162
Tabla 30 Listado de endpoints generados en el sprint 8	164
Tabla 31 Resumen de actividades planificadas en las daily meetings del sprint 9	171
Tabla 32 Listado de endpoints generados en el sprint 9	172
Tabla 33 Preguntas de la Encuesta de Usabilidad	180

Índice de figuras

Figura 1 Visión General de la Metodología Scrum.....	33
Figura 2 Servicio de adoración en la Iglesia Cristiana Pentecostal.....	42
Figura 3 Aplicación web en diferentes dispositivos.....	45
Figura 4 Consultas referentes a frameworks para backend en porcentaje por mes.....	51
Figura 5 Ranking de porcentaje de uso de frameworks para frontend.....	53
Figura 6 Ranking de sistemas de gestión de bases de datos.....	58
Figura 7 Acta de reunión: Primera Reunión.....	69
Figura 8 Backlog Jira Real Al Inicio Del Proyecto.....	81
Figura 9 Diagrama de Gantt del Proyecto.....	81
Figura 10 Arquitectura del sistema.....	84
Figura 11 Arquitectura del código.....	84
Figura 12 Mockup Sección Inicio.....	88
Figura 13 Mockup Sección Nosotros.....	89
Figura 14 Mockup Sección Servicios.....	90
Figura 15 Mockup Sección Grupos de Vida.....	91
Figura 16 Mockup Sección Eventos.....	92
Figura 17 Mockup Sección Donaciones.....	93
Figura 18 Mockup Sección Inicio de Sesión.....	94
Figura 19 Mockup Sección Administración.....	95
Figura 20 Diagrama de Navegabilidad de la Aplicación.....	96
Figura 21 Tablero del sprint Gestión de Información inicial.....	97
Figura 22 Codificación Backend Operación Create.....	98
Figura 23 Codificación Backend Operación Read.....	99
Figura 24 Codificación Backend Operación Update.....	99

Figura 25 Codificación Backend Operación Delete	100
Figura 26 Reunión de aceptación.....	103
Figura 27 Pantalla de administración de la Iglesia Camino Nuevo	107
Figura 28 Formulario para inicio de sesión a la consola de administración	107
Figura 29 Visualización de los valores centrales de la iglesia.....	108
Figura 30 Pantalla de gestión de valores centrales	109
Figura 31 Visualización de los datos de ubicación	109
Figura 32 Visualización de la información del pastor principal y el equipo pastoral	110
Figura 33 Pantalla de gestión de información del pastor principal	110
Figura 34 Pantalla de gestión de información del equipo pastoral.....	111
Figura 35 Visualización de las misiones de la iglesia	111
Figura 36 Pantalla de gestión de las misiones de la iglesia	112
Figura 37 Estado del tablero de Jira al finalizar el sprint 1.....	112
Figura 38 Tablero del sprint Cambio de Idioma inicial.....	113
Figura 39 Ejemplo de manejo de idiomas de la aplicación en la base de datos.....	114
Figura 40 Codificación de función para obtener información según idioma seleccionado.....	115
Figura 41 Selector de idiomas.....	118
Figura 42 Barra de títulos con cambio de idioma.....	118
Figura 43 Sección de “Inicio” con cambio de idioma.....	119
Figura 44 Sección de “Nosotros” con cambio de idioma.....	120
Figura 45 Estado del tablero de Jira al finalizar el sprint 2.....	121
Figura 46 Tablero del sprint Gestión de Contenido Multimedia inicial	122
Figura 47 Importación de los módulos necesarios para el manejo de contenido multimedia ..	123
Figura 48 Métodos para guardar y eliminar contenido multimedia.....	124
Figura 49 Visualización de imágenes y predicaciones.....	129
Figura 50 Gestión de imágenes	130

Figura 51 Gestión de predicaciones	130
Figura 52 Estado del tablero de Jira al finalizar el sprint 3.....	131
Figura 53 Tablero del sprint Gestión de Redes Sociales inicial.....	132
Figura 54 Visualización de redes sociales	137
Figura 55 Gestión de redes sociales	138
Figura 56 Sitio Web desplegado al Internet.....	139
Figura 57 Estado del tablero de Jira al finalizar el sprint 4.....	140
Figura 58 Tablero del sprint Gestión de Eventos inicial	141
Figura 59 Sección eventos	145
Figura 60 Gestión de eventos en vivo	145
Figura 61 Gestión de eventos importantes	146
Figura 62 Gestión de eventos en calendario	146
Figura 63 Estado del tablero de Jira al finalizar el sprint 5.....	147
Figura 64 Tablero del sprint Gestión de Servicios inicial.....	148
Figura 65 Sección servicios.....	152
Figura 66 Gestión de servicios	152
Figura 67 Estado del tablero de Jira al finalizar el sprint 6.....	153
Figura 68 Tablero del sprint Registro y Supervisión de Peticiones de Oración inicial	154
Figura 69 Función para recibir la información de las tarjetas de registro	155
Figura 70 Supervisión de peticiones de oración	159
Figura 71 Estado del tablero de Jira al finalizar el sprint 7.....	160
Figura 72 Tablero del sprint Gestión de Grupos de Vida inicial.....	161
Figura 73 Componente Tarjetas de Grupos de Vida	165
Figura 74 Sección Grupos de Vida.....	165
Figura 75 Gestión de Grupos de Vida	166
Figura 76 Estado del tablero de Jira al finalizar el sprint 8.....	167

Figura 77 Estado del tablero de Jira al iniciar el sprint 9.....	168
Figura 78 Configuración de la solicitud de donación	169
Figura 79 Creación y envío de la solicitud de donación.....	170
Figura 80 Configuración de conexión al servicio de PayPal	170
Figura 81 Tarjeta de donación por PayPal.....	174
Figura 82 Tarjeta de donación por código QR	174
Figura 83 Tarjeta de donación por otras formas	175
Figura 84 Sección de donaciones en el menú.....	175
Figura 85 Tarjetas para métodos de donación.....	176
Figura 86 Administración de donaciones por PayPal.....	176
Figura 87 Administración de donaciones por código QR	177
Figura 88 Administración de otros tipos de donaciones.....	177
Figura 89 Estado del tablero de Jira al finalizar el sprint 9.....	178
Figura 90 Product Backlog final.....	178
Figura 91 Resultados de la Encuesta de Usabilidad	180

Resumen

La pandemia de COVID-19 desencadenó un cambio de paradigma en las estrategias de comunicación de las instituciones religiosas, adoptando rápidamente herramientas tecnológicas como Internet para llegar a los feligreses. Este proyecto tiene como objetivo desarrollar el sitio web de la Iglesia Camino Nuevo, eliminando barreras idiomáticas y geográficas al proporcionar información y servicios de la iglesia en español e inglés. La arquitectura seleccionada emplea una estructura de n-niveles que se implementa utilizando el stack MEVN (MongoDB, ExpressJS, VueJS, NodeJS), que facilita la creación de una solución robusta y escalable. La integración con servicios de terceros, como PayPal, ayuda a implementar donaciones en línea seguras. La computación en la nube garantiza alta disponibilidad, lo que permite a los feligreses acceder en cualquier momento y lugar. Guiado por la metodología ágil Scrum, el proyecto evolucionó a través de la participación de los stakeholders y la entrega continua de funcionalidades. Las historias de usuarios y el product backlog guían el proceso de desarrollo, basado en los requisitos de la Iglesia Camino Nuevo. La fase de desarrollo cubre las implementaciones de front-end y back-end, las reuniones periódicas con los stakeholders y las implementaciones continuas. Las pruebas de usuario y los cuestionarios de usabilidad validan y verifican el sitio web y los resultados confirman adhesión a las expectativas iniciales de los feligreses. Como consecuencia, producimos un sitio web que rompe las barreras lingüísticas y fortalece a Iglesia Camino Nuevo para adaptarse a los desafíos tecnológicos contemporáneos.

Palabras clave: desarrollo de sitios web, sitio web para iglesias, metodología Scrum, sitio web multi idioma, stack MEVN

Abstract

The COVID-19 pandemic triggered a communication strategies paradigm change for religious institutions, prompting a quick adoption of technological tools such as Internet, to broadcast information and connect with parishioners. This project aims to develop the website for Iglesia Camino Nuevo, overcoming geographical and language barriers by providing relevant information and church services in both Spanish and English languages through the Web. The chosen architectural framework employs an n-tier structure that is implemented using the MEVN stack (MongoDB, ExpressJS, VueJS, NodeJS), that eases the creation of a robust and scalable technological solution. Integration with third-party services, like PayPal helps implement secure online donations, that are currently processed manually. Cloud computing ensures high availability, enabling parishioners to access it at anytime, anywhere. Guided by the agile Scrum methodology, the project evolved through active stakeholder engagement and continuous functionality delivery. User stories and the product backlog drive the development process, over Iglesia Camino Nuevo's requirements. The developmental phase covers frontend and backend implementations, regular stakeholder meetings, and continuous deployments. User (parishioners) testing validates and verifies the site through usability questionnaires. The questionnaire results confirm the website's adherence to initial expectations of parishioners. Consequently, we produced a website that breaks linguistic barriers, and strengthen Iglesia Camino Nuevo to adapt through contemporary technological challenges.

Keywords: website development, church website, Scrum methodology, multilingual website, MEVN stack

Capítulo I

Introducción

Antecedentes

Históricamente, la difusión de información de las iglesias se limitaba a la celebración de los servicios y la comunicación directa con los miembros de la congregación, mediante trato personal. Las iglesias utilizaban métodos tradicionales de comunicación como anuncios impresos, boletines y volantes para poder informar. Sumado a esto, un número reducido de congregaciones religiosas transmitían sus misas y servicios mediante la televisión y radio. Sin embargo, estos métodos presentan un alcance limitado y no ofrecen interactividad entre las partes interesadas.

En la época contemporánea, la tecnología se encuentra presente en prácticamente todos los ámbitos de la vida, tales como la educación, el entretenimiento, el comercio, la comunicación y el transporte. Gracias a ella, se ha hecho posible la automatización y simplificación de múltiples tareas cotidianas. Este fenómeno se debe en gran medida al incremento exponencial del uso de computadores desde el año 2000 con respecto a años anteriores. Por lo cual para el año 2008, en varios países de Europa, la cantidad de computadores por cada cien hogares llegó a ser de 86,4 (Ark, 2011), y una gran proporción de dichos equipos se encuentran conectados a Internet.

Se calcula que desde 2022, el número de usuarios de Internet a nivel global ha aumentado a 5.3 mil millones, lo que equivale a un 66 por ciento de la población mundial. El estado actual de desarrollo de las redes de comunicación y la computación en la nube han posibilitado la existencia de alrededor de 307 millones de usuarios de Internet, solo en los Estados Unidos. Por lo tanto, el Internet se ha convertido en un medio para llegar a la gente

con información, propagandas, noticias, negocios. (Petrosyan, 2023), el cual es uno de los países con mayor población cristiana en el mundo.

Un hecho relevante que ha acelerado el progreso de las tecnologías de la comunicación y la transmisión de información es la pandemia mundial COVID-19, que ha afectado a la población global desde noviembre de 2019 hasta la actualidad. Varios informes señalaron que la búsqueda de orientación espiritual y religiosa creció dramáticamente durante la pandemia. En consecuencia, “las distintas religiones se han visto en la necesidad de reinventarse con servicios, misas y meditaciones retransmitidas por medio del internet y distribuidas a través de redes sociales, facilitando el acceso de las personas que tenían prohibido salir de sus hogares y/o asistir a eventos masivos” (Wunsch et al., 2022).

Tras el retorno a la presencialidad, las organizaciones religiosas han optado por descartar medios convencionales de comunicación, como la radio o la televisión, debido a los elevados costos logísticos y la limitación en cuanto a público objetivo. Como solución, las instituciones religiosas han adoptado nuevas herramientas tecnológicas con el fin de transmitir sus servicios mediante internet. En la actualidad, “muchas iglesias ofrecen una experiencia híbrida que combina lo virtual y lo presencial, como resultado, para el año 2022, el 15% de adultos estadounidenses miraban servicios religiosos en línea con más frecuencia que antes de la pandemia” (Nortey & Rotolo, 2023).

De acuerdo a (Matthews, 2023), el 89% de las iglesias actualmente transmiten servicios de adoración en vivo, y de ellas, la mitad planea en el plazo de un año alejarse de redes sociales para emplear sitios web que permitan mostrar información básica del centro religioso o aplicaciones webs que permitan realizar operaciones más complejas, como conexiones a bases de datos para almacenar y procesar datos o gestión de donaciones en línea que sustituye métodos tradicionales de solicitud de fondos a los feligreses mediante el envío de correos físicos o electrónicos.

Actualmente, existe un centro religioso cristiano pentecostal ubicado en el estado de California – Estados Unidos llamado Iglesia Camino Nuevo, que ofrece servicios religiosos tales como celebración de alabanzas, conformación de grupos de vida, festejo de eventos especiales, entre otros propios de la religión practicada en el centro. La iglesia necesita un software que ayude a difundir sus servicios y eventos, además de información importante sobre la estructura organizativa de la misma como horarios, equipo pastoral, misión, valores centrales, redes sociales, con la finalidad de ampliar su audiencia y llevar su mensaje a un mayor número de personas.

Al comenzar este proyecto, la iglesia Camino Nuevo no posee un sitio web que cumpla con las tareas de informar a las personas sobre las actividades, eventos, servicios ofrecidos y difusión de los servicios en vivo, por consiguiente, surge la necesidad de adquirir un sitio web para la iglesia Camino Nuevo. La construcción del software se llevará a cabo por estudiantes de la Universidad de las Fuerzas Armadas “ESPE” como parte del proyecto de integración curricular. El sitio ofrecerá múltiples funcionalidades desarrolladas para cubrir las necesidades de una organización religiosa.

Planteamiento del problema

La Iglesia Camino Nuevo fue establecida en el año 2022 con el objetivo de encaminar a la comunidad hacia los valores centrales de la entidad y cumplir su misión en el transcurso del proceso, asimismo, el pastor Eric Lara encargado del centro religioso, declaró que la iglesia ofrece servicios y actividades tales como adoración, grupos de vida, escuela dominical, programas juveniles, matrimonios, bautizos. La proporción de feligreses que están al tanto de información relevante sobre la iglesia y los servicios que se ofrecen es reducida, lo que hace imprescindible el contacto directo o la presencia física en la ubicación de la iglesia para que los miembros de la congregación puedan obtener dicha información.

La iglesia de igual manera acepta peticiones de oración, donaciones, testimonios, y registro de nuevos miembros, sin embargo, estas peticiones son realizadas de manera manual por parte de los feligreses o personas interesadas (vía personal o vía telefónica), lo que resulta en generación de una gran cantidad de documentos físicos, así como la dedicación de tiempo y personal que, por lo general, se dedican a otras actividades en la iglesia.

Según (Pew Research Center, 2015), Estados Unidos continúa siendo la nación con la mayor cantidad de personas cristianas en el mundo. En este país, se observa un incremento en el interés de los individuos por explorar los servicios y beneficios que ofrecen sus instituciones religiosas, tal como se evidencia en el aumento del uso de diversas plataformas informativas en línea, incluyendo sitios web y redes sociales como Facebook, Twitter e Instagram.

La iglesia Camino Nuevo no cuenta con un sitio web que permita la difusión eficaz de información acerca de servicios, actividades y eventos relevantes a su congregación y a la comunidad en general. La implementación de un sitio web sería sumamente beneficiosa en la difusión de dicha información, tanto para los miembros internos como para los interesados externos. Además, el sitio web proporcionaría la ventaja de tener a una persona responsable de su administración, asegurando una supervisión completa del contenido compartido y actualización de estos.

La solución propuesta para satisfacer las necesidades previamente descritas consiste en la creación de un sitio web que incluya la gestión de la información de la iglesia, gestión de los idiomas del sitio, manejo del contenido multimedia mostrada en el sitio, difusión de redes sociales, administración de eventos, divulgación de servicios ofrecidos, gestión de peticiones de oración, dirección de grupos de vida y manejo de donaciones.

Justificación

En la actualidad, son pocas las instituciones cristianas que disponen de un sitio web que ofrezca la asistencia necesaria para aquellas personas que buscan información relevante acerca de la organización. Además, los sitios existentes carecen de personalización adecuada y proporcionan información muy superficial, siendo además estáticos, poco interactivos y difíciles de mantener, adicionalmente, estos sitios se limitan a un solo idioma, lo cual restringe el acceso para aquellas personas que hablan otro idioma y que están interesadas en los servicios ofrecidos por la organización.

La sección previa, que incluye los antecedentes y el planteamiento del problema, deja en claro la necesidad de desarrollar un sitio web que atienda las necesidades específicas de la iglesia. Este software se enfocará tanto en los feligreses, quienes necesitan tener acceso fácil y rápido a información importante de la iglesia, como en el usuario administrador, quien requiere de una herramienta intuitiva que permita editar y mantener el sitio de forma sencilla y eficiente.

De manera general, la iglesia obtendrá beneficios a través de la comodidad y simplicidad que se lograrán al compartir información, servicios y eventos hacia el público en general mediante internet. Esto se traducirá en un mayor interés por parte de los usuarios que naveguen por el sitio y en la llegada del mensaje de la iglesia hacia un mayor número de personas.

Objetivos

Objetivo General

Desarrollar un sitio web para la Iglesia Camino Nuevo que permita a los feligreses de habla hispana e inglesa acceder a información relevante de la organización, actividades, eventos y servicios de la iglesia, además de realizar solicitudes, peticiones de oración y compartir testimonios, con el fin de alcanzar una audiencia más amplia sin considerar el idioma ni la región de donde se acceda al sitio, además de gestionar las donaciones realizadas al

centro religioso de manera rápida y segura, utilizando el stack de desarrollo MEVN (Mongo, Express, Vue, Node) para la implementación y el servicio de alojamiento en la nube.

Objetivos Específicos

- i. Realizar una revisión sistemática de literatura y estudio de la situación actual de las iglesias post pandemia en el mundo digital.
- ii. Realizar una revisión sistemática de literatura y estudio de tecnologías actuales para el desarrollo de sitios web.
- iii. Realizar una revisión sistemática de literatura y estudio de las tecnologías de computación en la nube.
- iv. Efectuar reuniones y entrevistas virtuales para obtener los requisitos y determinar las historias de usuario basado en la metodología Scrum.
- v. Elaborar el product backlog basado en los requerimientos de la Iglesia Camino Nuevo.
- vi. Desarrollar los módulos que componen el product backlog del sitio web de la Iglesia Camino Nuevo.
- vii. Realizar un despliegue continuo de cada módulo desarrollado.
- viii. Llevar a cabo pruebas de aceptación de usuario juntamente con las personas involucradas en el proyecto y usuarios en general.

Alcance

El proyecto de desarrollo de un sitio web para la Iglesia Camino Nuevo tiene como objetivo ofrecer un alcance completo y funcional para satisfacer las necesidades de la comunidad religiosa.

El sitio web permitirá una gestión eficiente de la información de la iglesia, brindando a los usuarios acceso a datos relevantes sobre eventos, servicios y grupos de vida. Además, contará con un sistema de gestión del idioma que facilitará la navegación y comprensión para personas de diferentes nacionalidades y lenguas.

La gestión del contenido multimedia será una parte integral del sitio, permitiendo la publicación de imágenes, videos y audio que enriquezcan la experiencia de los visitantes. La integración de redes sociales será fundamental para la difusión y participación en línea, permitiendo compartir contenido y seguir las actualizaciones de la iglesia a través de diferentes plataformas.

La gestión de eventos será una característica clave, ya que los usuarios podrán acceder a información detallada sobre fechas, horarios y ubicaciones de los eventos de la iglesia. De igual forma, el sitio web facilitará la gestión de servicios, donde los usuarios podrán encontrar información relevante sobre los horarios de los servicios y acceder a transmisiones en vivo. Se ofrecerá un espacio dedicado a las solicitudes y peticiones de oración, así como la posibilidad de compartir testimonios, fomentando la interacción y apoyo comunitario.

Por último, la gestión de donaciones permitirá a los feligreses contribuir de manera segura y sencilla a la iglesia. Este proyecto de desarrollo de un sitio web integral busca potenciar la comunicación, participación y crecimiento de la comunidad de la Iglesia Camino Nuevo.

Metodología

El actual trabajo de integración curricular será desarrollado siguiendo parcialmente el marco de trabajo Scrum, que se define como un conjunto de prácticas para desarrollar, entregar y mantener productos complejos. Scrum ha sido usado para gestionar el trabajo en productos complejos desde principios de los años 90. Scrum muestra la eficacia relativa de las técnicas de gestión de producto y las técnicas de trabajo, de modo que podamos mejorar continuamente el producto, el equipo y el entorno de trabajo (Schwaber & Sutherland, 2017).

El marco de trabajo Scrum consiste en Equipos Scrum, con sus respectivos eventos, artefactos y reglas asociadas. Cada uno de estos elementos cumple con un propósito

específico y es fundamental para el éxito de Scrum y su implementación, tal y como se detalla en la obra de (Schwaber & Sutherland, 2017).

El Equipo Scrum (Scrum Team) se compone de tres elementos: el Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y el Scrum Master. Cada uno de estos elementos cumple con una función específica dentro del equipo. Los Equipos Scrum son autoorganizados y multifuncionales (Schwaber & Sutherland, 2017).

El Dueño de Producto (Product Owner) tiene la responsabilidad de maximizar el valor del producto que resulta del trabajo del Equipo de Desarrollo. La manera en que se lleva a cabo esta tarea puede variar significativamente entre distintas organizaciones, Equipos Scrum e individuos (Schwaber & Sutherland, 2017).

El Equipo de Desarrollo (Development Team) está compuesto por los profesionales que trabajan en la entrega de un incremento de producto "Terminado" que potencialmente se pueda poner en producción al final de cada Sprint. Esta definición refleja la importancia de que el trabajo del Equipo de Desarrollo tenga como resultado un incremento completo y potencialmente utilizable del producto (Schwaber & Sutherland, 2017).

El Scrum Master es el responsable de promover y apoyar el marco de trabajo Scrum tal como se define en la Guía de Scrum. Para ello, los Scrum Masters ayudan a todos los involucrados en el proyecto a comprender la teoría, las prácticas, las reglas y los valores de Scrum, y brindan apoyo continuo en la implementación efectiva del marco de trabajo. En general, juegan un papel fundamental en la promoción de la cultura de Scrum dentro de la organización y en la creación de un ambiente propicio para el trabajo en equipo y la colaboración (Schwaber & Sutherland, 2017).

Scrum establece eventos predefinidos con el objetivo de generar regularidad y minimizar la necesidad de reuniones adicionales no contempladas en el marco de trabajo.

Todos los eventos tienen una duración máxima establecida, por lo que se consideran bloques de tiempo (time-boxes) (Schwaber & Sutherland, 2017).

Un Sprint representa el núcleo central, es un bloque de tiempo (time-box) de un mes o menos, donde el equipo Scrum trabaja para crear un incremento de producto "Terminado" y potencialmente desplegable (Schwaber & Sutherland, 2017).

Visto de esta forma, para empezar a trabajar con Scrum se debe empezar por la conformación del equipo Scrum en base a los roles: PO¹, SM², DT³, luego se procede a la toma de requerimientos por parte del PO para que a su vez este los analice y defina las historias de usuario con las que se va a trabajar. Una vez obtenidas las historias de usuario se procede a la creación del product backlog donde el PO trabaja en conjunto con el DT para definir y priorizar cada una de las historias.

Con el product backlog listo, se da paso a la planificación de las iteraciones o sprints, donde el equipo debe realizar una reunión denominada sprint planning meeting para aclarar temas del sprint como los objetivos, definición de tareas, priorización de tareas o features y asignación de miembros del equipo a dicha tarea (Schwaber & Sutherland, 2017).

Dentro de cada sprint, se lleva a cabo el desarrollo de la tarea asignada y la realización de reuniones diarias o daily scrum meeting para realizar una breve revisión del progreso del equipo, además de las dificultades o impedimentos que se están enfrentando para solucionarlos (Schwaber & Sutherland, 2017). Por último, si las actividades del día han sido terminadas con éxito, se continúa con la actualización del burndown chart que consiste en una

¹ Product Owner: Rol dentro de Scrum encargado de representar los intereses y necesidades de las partes interesadas.

² Scrum Master: Rol dentro de Scrum encargado de asegurar que se sigan los principios y prácticas del marco de trabajo.

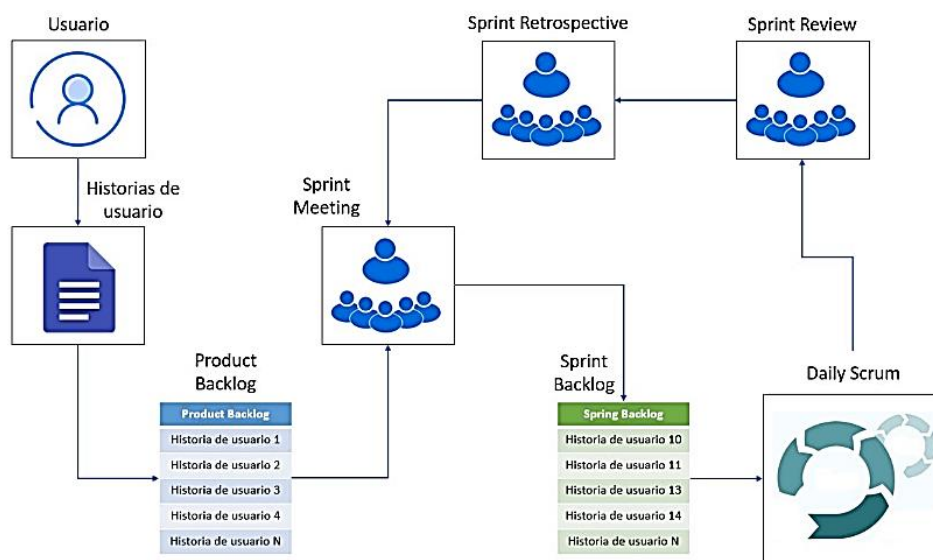
³ Development Team: Rol dentro de Scrum encargado de entregar los incrementos de producto potencialmente entregables.

herramienta que ayuda a visualizar el avance del proyecto y el manejo de fechas de desarrollo del product backlog.

Por último, se lleva a cabo la revisión del sprint en la que se exhibe el trabajo ejecutado y se entabla una conversación sobre los conocimientos adquiridos, además de lo requerido para llevar a cabo el próximo sprint (Schwaber & Sutherland, 2017). En caso de que existan nuevas necesidades, se ejecuta un análisis y se adapta el product backlog de ser necesario. Todo esto con la finalidad de discutir los problemas y las mejoras que se pueden implementar en el siguiente sprint. El proceso antes mencionado referente al marco de trabajo Scrum se describe en la Figura 1.

Figura 1

Visión General de la Metodología Scrum



Capítulo II

Marco Teórico

Para el presente proyecto es necesario realizar una revisión de conceptos y definiciones que serán útiles para el análisis y diseño del software, además de sustentar el desarrollo del proyecto. En este capítulo se presenta una base teórica sobre la religión, religión cristiana pentecostal, software de gestión de iglesias, tecnologías y herramientas para desarrollo de aplicaciones web y sobre el proceso de desarrollo de aplicaciones web.

Religión

La religión es un fenómeno complejo y multifacético que ha sido analizado desde múltiples perspectivas en el transcurso de diferentes disciplinas a lo largo de su historia. Una definición comúnmente aceptada es que la religión es “un fenómeno desconcertante para muchas personas en el mundo moderno. Se puede asociar favorablemente con cualidades humanas agradables como la ética, la moral y la espiritualidad, o negativamente con la superstición, sofocación, el rechazo del conocimiento científico y el progreso humano en general” (Jensen, 2019).

Según el sociólogo John Milton Yinger, la religión es "un sistema de creencias y prácticas por medio del cual un grupo de personas lucha con los problemas fundamentales de la vida humana" (Crawford, 2003), mientras que Durkheim menciona que “la religión es sólo el sentimiento inspirado por sus miembros, pero proyectado fuera de la conciencia que los experimenta, y objetivado” (Durkheim, 1982).

Otra definición interesante propuesta por el sociólogo Émile Durkheim sostiene que la religión es "un sistema unificado de creencias y prácticas relativas a las cosas sagradas, es decir, cosas apartadas y prohibidas, creencias y prácticas que unen en una sola comunidad moral llamada Iglesia a todos los que se adhieren a ellas" (Durkheim, 1982).

A nivel global, se encuentran diversas comunidades religiosas que siguen a una religión caracterizada por ser un sistema de creencias y prácticas que siguen a algo divino, sin embargo, para establecer una comunicación entre la comunidad y lo divino, resulta fundamental la presencia de líderes religiosos que desempeñan un papel crucial en la orientación y formación de sus respectivas comunidades religiosas. Estos líderes, a menudo reconocidos como autoridades espirituales, actúan como intermediarios entre lo divino y sus seguidores. Como se observó en (Smith, 1991), la religión se moldea a partir de las inquietudes de las personas por encontrar un propósito y sentido en sus vidas. Cada religión cuenta con líderes religiosos que afirman tener acceso a los reinos invisibles, inaccesibles a la percepción y el tacto humano, y es a través de ellos que se establece esta conexión.

Estas comunidades religiosas se fundamentan en textos sagrados que son fundamentales para las tradiciones religiosas y sirven como fuentes autorizadas de conocimiento, enseñanzas morales y guía espiritual. Por ejemplo, la Biblia tiene una importancia significativa en el cristianismo, el Corán en el islam y los Vedas en el hinduismo. Estos textos a menudo se consideran revelaciones divinas y son estudiados, interpretados y reverenciados por eruditos y líderes religiosos. Según (Eliade, 1987), el propósito de los libros sagrados de una religión es hacer comprensible el comportamiento y el universo mental del hombre religioso. Los líderes religiosos recurren con frecuencia a estos textos para brindar pautas éticas, dar forma a las prácticas religiosas y reforzar las creencias de sus comunidades.

Las comunidades religiosas subsisten a través de varias fuentes, incluidas las donaciones individuales, los diezmos, las subvenciones y otras formas de apoyo financiero. Estas contribuciones son utilizadas para financiar el mantenimiento de las instituciones religiosas, los salarios de los líderes religiosos y la organización de ceremonias y actividades religiosas. Asimismo, las comunidades religiosas pueden generar ingresos a través de programas educativos, iniciativas caritativas o la venta de materiales religiosos. Es importante

destacas que las formas en que las comunidades religiosas logran su subsistencia pueden variar ampliamente según las tradiciones específicas, los contextos culturales y la escala de la organización religiosa.

En base a los conceptos anteriores, la religión se define como un conjunto de creencias, prácticas y valores compartidos por una comunidad de personas que se relacionan con lo trascendente, lo divino y lo sagrado. La religión incluye rituales, ceremonias y enseñanzas que buscan proporcionar un significado y propósito a sus seguidores. Aunque el significado de la religión puede variar en diferentes culturas y sociedades, su esencia radica en otorgar sentido a la existencia humana y establecer una conexión espiritual con lo mas allá de lo material. Cada religión cuenta con líderes religiosos que desempeñan un papel central en la orientación y dirección de sus respectivas comunidades, considerados como autoridades. Estos líderes se basan en libros sagrados propios de cada comunidad religiosa que son considerados revelaciones divinas. En la estructura interna de las comunidades religiosas, se observa una variedad de modelos organizativos, desde jerarquías con una autoridad centralizada, como el papado en el catolicismo, hasta estructuras más descentralizadas, como las congregaciones protestantes con liderazgo pastoral local.

Tipos de Religión

La religión ha desempeñado un papel trascendental en la experiencia humana a lo largo de la historia, manifestándose en diversas expresiones a lo largo y ancho del planeta. Algunos de los tipos de religión más conocidos incluyen el cristianismo, el islam, el hinduismo, el budismo, el judaísmo y el sikhismo (Noss & Grangaard, 2017).

En el mundo existen numerosos tipos de religiones, que varían en su origen, creencias, prácticas y estructuras organizativas. Basado en el informe de (Pew Research Center, 2015)⁴

⁴ Pew Research Center: Centro de investigación fundado en 2004 con sede en Washington dedicado al estudio de actitudes públicas, tendencias sociales y datos demográficos a nivel mundial.

sobre las religiones del mundo, el cristianismo es la religión más grande del mundo, seguida del islam y el hinduismo. Sin embargo, también existen muchas religiones más pequeñas y menos conocidas, como las religiones africanas tradicionales o las religiones indígenas de América del Norte y del Sur (Wormald, 2015), por lo tanto, como se puede observar en la Tabla 1, el cristianismo junto a todas sus ramas tienden a seguir creciendo y manteniéndose como la religión más practicada en el mundo.

Tabla 1

Tamaño y proyección de crecimiento de los principales grupos religiosos

Religión	Población en 2010	% de la población mundial en 2010	Población proyectada para 2050	% de la población mundial en 2050	Crecimiento de la población 2010-2050
Cristianos	2,168,330,000	31.4%	2,918,070,000	31.4%	749,740,000
Musulmanes	1,599,700,000	23.2%	2,761,480,000	29.7%	1,161,780,000
No afiliados	1,131,150,000	16.4%	1,230,340,000	13.2%	99,190,000
Hindúes	1,032,210,000	15.0%	1,384,360,000	14.9%	352,140,000
Budistas	487,760,000	7.1%	486,270,000	5.2%	-1,490,000
Religiones folclóricas	404,690,000	5.9%	449,140,000	4.8%	44,450,000
Otras religiones	58,150,000	0.8%	61,450,000	0.7%	3,300,000
Judíos	13,860,000	0.2%	16,090,000	0.2%	2,230,000
Total mundial	6,895,850,000	100.0%	9,307,190,000	100.0%	2,411,340,000

Nota. Adaptado de The Future of World Religions: Population Growth Projections, 2010-2050, Pew Research Center, 2015, <https://www.pewresearch.org/religion/2015/04/02/religious-projections-2010-2050/> (Wormald, 2015)

Tipos de Religión Cristiana

En la actualidad, se pueden identificar diversas corrientes dentro de la religión cristiana, cada una de las cuales exhibe características particulares en términos de creencias, tradiciones y estructuras organizativas. Entre las corrientes más extendidas se encuentran el catolicismo, el protestantismo y la Iglesia ortodoxa.

La corriente más practicada en el mundo es el catolicismo romano, cuyos orígenes se remontan a la Iglesia Cristiana Primitiva⁵ y que reconoce la autoridad del Papa sobre toda la organización religiosa con sede en Roma (Walsh, 2016), contando con importantes templos sagrados conocidos como iglesias, que son utilizados como lugares de culto y encuentro para los fieles.

Otra rama significativa del cristianismo es la ortodoxa oriental caracterizada por su énfasis en la liturgia, los sacramentos y la vivencia mística de Dios (Smith & Sloan, 2017), está presente en varios países de Europa Oriental, Rusia y Oriente Medio, cuenta con el Patriarcado Ecuménico de Constantinopla, ubicado en Estambul, Turquía, es considerado como el centro espiritual más importante de la Iglesia Ortodoxa y el Patriarca Ecuménico es el líder de esta iglesia..

El protestantismo, por otro lado, abarca una amplia gama de denominaciones surgidas a partir del siglo XVI como resultado de la reforma protestante, como por ejemplo la Iglesia Luterana, que se caracteriza por tener una mayor autonomía congregacional y una interpretación más individual de la fe (Smith & Sloan, 2017). Por otro lado, el pentecostalismo, otra corriente importante dentro del protestantismo, se subdivide en diversas denominaciones, como la Iglesia de Dios en Cristo, las Asambleas de Dios y la Iglesia del Evangelio Cuadrangular. Estas denominaciones pentecostales ponen énfasis en la experiencia del

⁵ Iglesia Cristiana Primitiva se refiere al periodo temprano de la historia del cristianismo, es decir, abarca desde los primeros seguidores de Jesús hasta el siglo IV aproximadamente.

Espíritu Santo y la manifestación de los dones espirituales, así como la adoración carismática y la oración ferviente. Cada una de ellas cuenta con su propia estructura organizativa, líderes religiosos y enfoques teológicos que las distinguen dentro del contexto del protestantismo.

Religión Cristiana Pentecostal

La religión cristiana pentecostal es una corriente del cristianismo evangélico que se caracteriza por la creencia en la experiencia del bautismo en el Espíritu Santo, que se manifiesta a través del hablar en lenguas y la práctica de dones espirituales. Esta corriente surgió en los Estados Unidos a principios del siglo XX y se ha expandido a nivel mundial (Melton, 2023).

De acuerdo con (Melton, 2023), los pentecostales representan una de las denominaciones cristianas de más rápido crecimiento en todo el mundo. Los pentecostales creen en la autoridad de la Biblia y en la importancia de la salvación personal a través de la fe en Jesucristo. También se destacan por su énfasis en la evangelización y en la búsqueda de la sanidad divina.

En términos de organización, el cristianismo pentecostal se compone de varias estructuras e instituciones que apoyan su misión y facilitan su crecimiento. En esencia, el pentecostalismo adopta un modelo descentralizado en el que las iglesias y congregaciones individuales o pertenecientes a otra organización religiosa, ministerio o apostolado, funcionan de manera autónoma bajo la dirección de sus respectivos líderes, dirigidos generalmente por un pastor o un grupo de ancianos (Burgess & Maas, 2010).

No obstante, existen corrientes del pentecostalismo como la Iglesia Cuadrangular que sigue una estructura organizativa jerárquica en la cual se establecen diferentes niveles de liderazgo y autoridad. En la cúspide de esta estructura se encuentra el presidente de la denominación, actualmente Randy Remington, quien asume la responsabilidad del liderazgo

general, coordina las operaciones y actividades de la iglesia, y representa a la misma a nivel internacional. El supervisor general, ocupado actualmente por Wendy Nolasco, desempeña el papel de asesor y apoyo, guiando o dirigiendo a la comunidad hacia el cumplimiento de la visión y propósito. Subordinados al supervisor general, se encuentran los líderes regionales o supervisores de distrito, quienes tienen la responsabilidad de supervisar distritos específicos dentro del ámbito de la iglesia. Estos líderes trabajan en estrecha colaboración con los pastores y congregaciones locales, brindándoles apoyo y capacitación para garantizar la correcta enseñanza de los principios y doctrinas de la iglesia (*The Foursquare Church, 2023*).

A nivel local, cada Iglesia Pentecostal Cuadrangular es dirigida por un pastor o ministro principal, quien asume el liderazgo espiritual de la congregación. Este líder es responsable de guiar los servicios de adoración, predicar sermones, administrar los sacramentos y brindar apoyo a los miembros de la comunidad religiosa. Además, el pastor generalmente cuenta con un grupo de apoyo compuesto por diáconos o ministros asociados, quienes colaboran en las tareas operativas y actividades de la iglesia

El papel de un pastor dentro de la religión cristiana pentecostal es importante, ya que desempeña el rol de líder espiritual y guía pastoral de la congregación. Para acceder a este cargo, se requiere atravesar un proceso que comienza con un llamado interno, en el cual el individuo experimenta la convicción de ser llamado por Dios para servir en el ministerio pastoral. Posteriormente, es crucial que el aspirante emprenda un proceso de educación teológica formal a fin de adquirir conocimientos bíblicos y principios de liderazgo necesarios para ejercer su labor pastoral (SBC CALLED, 2021). Finalmente, los miembros de la congregación evalúan y afirman el llamado y los dones del creyente que se siente llamado al ministerio donde se busca garantizar que la persona se encuentre debidamente preparada y capacitado para asumir el cargo.

La Iglesia Cuadrangular se fundamenta en una serie de creencias esenciales que dan forma a su identidad y teología. Estas creencias comprenden la afirmación de la Biblia como la Palabra inspirada y autoritativa de Dios, la creencia en la Trinidad divina compuesta por el Padre, el Hijo (Jesucristo) y el Espíritu Santo, y la convicción de que la salvación se obtiene exclusivamente a través de la fe en Jesús como Salvador y Señor (*The Foursquare Church*, 2023), además se cree principalmente en dos sacramentos principales los cuales son el bautismo en agua y la santa cena similar a la comunión en el catolicismo.

La Iglesia Cuadrangular también destaca la relevancia y la importancia del bautismo en el Espíritu Santo como una experiencia posterior a la salvación, la cual capacita y llena a los creyentes con poder espiritual para vivir una vida cristiana victoriosa. De igual manera, la Iglesia Cuadrangular valora profundamente la sanidad divina y enfatiza la importancia del ministerio de los dones espirituales, reconociendo y creyendo en la pertinencia de los dones del Espíritu Santo para la edificación y el crecimiento de la iglesia, con el propósito de guiar a las personas hacia el cielo en lugar del infierno. La práctica del diezmo y las ofrendas también forman parte de la enseñanza y práctica en la Iglesia Cuadrangular (*The Foursquare Church*, 2023).

La Iglesia Cuadrangular, al igual que las diferentes corrientes pentecostales existentes se caracterizan por celebrar servicios de adoración definidos por alabanzas y adoración enérgica, predicación apasionada y resalto de experiencias espirituales personales. Estos servicios a menudo incluyen música animada, con un enfoque en canciones de adoración contemporáneas y el uso de instrumentos como guitarras, teclados y tambores. Los miembros de la congregación participan activamente en la adoración, expresando su fe cantando, aplaudiendo, levantando las manos y ocasionalmente hablando en lenguas como se observa en la Figura 2. Estos servicios, tradicionalmente, se han ofrecido de manera presencial, sin embargo, algunas iglesias iniciaron con la difusión de sus servicios de manera virtual,

modalidad que experimentó un crecimiento significativo a raíz de la pandemia mundial COVID-19, donde la modalidad virtual pasa de simples transmisiones en vivo a la creación de plataformas que permiten interacción bilateral entre la comunidad religiosa. A través de estas plataformas, los feligreses pueden participar en los servicios religiosos, realizar peticiones de oración, compartir testimonios y conectarse con otros miembros de la comunidad.

Figura 2

Servicio de adoración en la Iglesia Cristiana Pentecostal



Nota. Imagen tomada en la celebración de un servicio de adoración en la iglesia Camino Nuevo (Autoría: Eric Lara)

Las iglesias pentecostales han adoptado varias plataformas y tecnologías digitales con el fin de brindar experiencias de adoración en línea a sus congregaciones. A través de servicios de transmisión en vivo, videoconferencias y plataformas de redes sociales, las iglesias pentecostales han podido llegar a una audiencia más amplia y mantener un sentido de

comunidad y conexión espiritual, incluso en momentos en los que las reuniones físicas no son posibles (Kgatle, 2020).

Estas herramientas digitales han permitido facilitar la interacción entre los miembros de la comunidad religiosa, por lo que las iglesias han optado por publicar su información en Internet mediante páginas web. Con el tiempo, esto ha evolucionado hacia la creación de aplicaciones web que permiten realizar servicios religiosos en vivo, compartir testimonios y oraciones, difundir enseñanzas, anunciar eventos importantes e incluso realizar donaciones en línea, demostrando de esta forma, la adaptabilidad de las iglesias cristianas pentecostales y el compromiso de continuar sirviendo a su comunidad.

La aplicación web a diferencia de la página web proporciona mayores funcionalidades interactivas y dinámicas para los feligreses, quienes pueden participar activamente en las actividades religiosas y acceder a recursos espirituales desde cualquier lugar y en cualquier momento.

Software para la gestión de iglesias

El software de gestión de iglesias ha experimentado avances significativos en los últimos años, adaptándose a las necesidades cambiantes de las comunidades religiosas. Con el aumento de la digitalización, estos programas ofrecen soluciones integrales que abarcan diversos aspectos de la administración eclesiástica. Entre estas funcionalidades, se encuentran la gestión de información, programación de eventos, administración de donaciones, registro de miembros, coordinación de grupos, control de servicios y facilitación de la comunicación interna mediante el uso de aplicaciones web.

Según el informe de investigación de (Saha, 2022), se ha observado un crecimiento notable en la adopción de software de gestión de iglesias en congregaciones de diversos tamaños, ya que reconocen los beneficios de la automatización y la centralización de datos.

Además, la investigación destaca que para el año 2022 el mercado de software para gestión de iglesias fue avaluado en \$232.4 millones y para el año 2032 tiene proyección a crecer a \$405.0 millones.

Este crecimiento de la tasa compuesta anual se respalda en gran medida gracias a distintos factores mencionados en el informe de (Saha, 2022) como:

- La creciente demanda de aplicaciones empresariales basadas en la nube.
- La creciente adopción de teléfonos inteligentes y del uso de Internet.
- La necesidad de optimizar el rendimiento ahorrando tiempo.
- El aumento de la necesidad de reducir la administración y el creciente uso de análisis de datos.
- Integración del software de gestión de iglesias con aplicaciones de marketing.

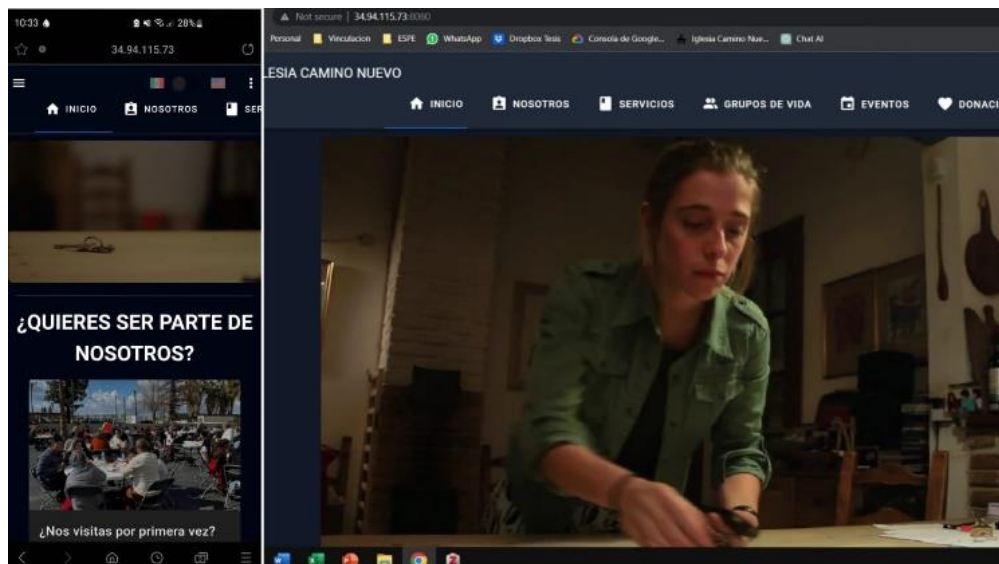
Aplicación Web

Una aplicación web es un programa informático que opera en un servidor web y se puede acceder mediante un navegador web utilizando conexión a Internet. Estas aplicaciones son utilizadas comúnmente para realizar tareas en línea, como la gestión de bases de datos, la edición de documentos o el acceso a servicios en la nube. Las aplicaciones web también pueden ser utilizadas para el comercio electrónico y la publicación de contenido en línea (AWS, 2023d).

Asimismo, cabe destacar que las aplicaciones web prescinden del uso de un SDK, lo que brinda la posibilidad de programar de forma independiente al sistema operativo en el cual se ejecutará la aplicación. De esta manera, las aplicaciones web se caracterizan por su capacidad de adaptarse y ser utilizadas en distintas plataformas y desde distintos navegadores web sin mayores obstáculos (Cuello & Vittone, 2013), ver Figura 3.

Figura 3

Aplicación web en diferentes dispositivos



Este tipo de aplicaciones, suelen constar de dos componentes principales llamados frontend (scripts del lado del cliente) y backend (scripts del lado del servidor), los cuales son los responsables de brindar una experiencia integral y satisfactoria al usuario. Los desarrolladores emplean diversas tecnologías, herramientas y siguen un proceso adecuado para llevar a cabo una implementación efectiva.

Tecnologías para desarrollo web

En la era digital actual, las aplicaciones web desempeñan un papel fundamental y, para cumplir con sus funciones de manera efectiva, han implementado una arquitectura dividida en dos componentes esenciales basada en la arquitectura cliente-servidor. El primer componente, conocido como frontend, se encarga de la interfaz de la aplicación, brindando una experiencia de usuario satisfactoria y permitiendo la interactividad con el sitio web. El segundo componente, conocido como backend, se ocupa de las operaciones y la lógica empresarial que impulsan el funcionamiento de la aplicación.

Estos componentes se comunican a través de distintos protocolos como REST mediante el uso de API's y pueden ser desarrollados utilizando herramientas y tecnologías básicas como HTML, CSS y lenguajes de programación como JavaScript, PHP. Sin embargo, para superar las limitaciones inherentes a estas opciones básicas, se han creado herramientas más avanzadas, lenguajes, frameworks y bibliotecas que facilitan un desarrollo más eficiente y efectivo tanto para el backend como para el frontend de las aplicaciones web. Estas soluciones permiten una mayor productividad, ofrecen características adicionales y simplifican tareas complejas, mejorando así el proceso de desarrollo en general.

Cliente-Servidor

Las aplicaciones que emplean el modelo cliente-servidor son una parte fundamental de la informática moderna y proporcionan una arquitectura distribuida que permite una comunicación eficiente y el uso compartido de recursos entre clientes y servidores. De acuerdo con (Yadav, 2009), el término "cliente-servidor" se define como "un modelo informático que involucra la distribución de funciones entre dos procesos independientes y autónomos: el servidor y el cliente". El cliente se refiere a cualquier proceso o entidad que solicita servicios específicos al servidor, mientras que el servidor es responsable de proveer los servicios solicitados por el cliente (Yadav, 2009).

El cliente y el servidor interactúan mediante protocolos y métodos de comunicación estandarizados, como el Protocolo de transferencia de hipertexto (HTTP) para aplicaciones basadas en web. Esto permite la interoperabilidad y que los programas cliente y servidor se desarrollen de forma independiente, lo que facilita la escalabilidad y la flexibilidad en el diseño del sistema (Rathore et al., 2017). El modelo cliente-servidor se clasifica en 4 tipos: arquitectura monolítica, arquitectura de dos capas, arquitectura de tres capas y arquitectura de múltiples capas. Entre los tipos más ampliamente utilizados, se encuentran las aplicaciones de dos niveles, que constan de un cliente y una base de datos, y las aplicaciones de tres niveles, que

incluyen una capa de presentación, una capa de aplicación y una capa de base de datos (Kumar, 2019).

API

Una API es un componente fundamental en la comunicación entre sistemas y aplicaciones en el ámbito de la tecnología de la información. Según (Lamothe et al., 2021), las APIs son interfaces que se utilizan como puntos de entrada a entidades de software reutilizables utilizadas por múltiples programas cliente fuera de la organización en desarrollo.

De esta manera, las APIs simplifican la creación de aplicaciones, al permitir que los desarrolladores utilicen servicios ya existentes para construir nuevas soluciones, entonces, las APIs son un elemento clave en la arquitectura de sistemas distribuidos y en el desarrollo de aplicaciones modernas.

Representational State Transfer (REST)

Representational State Transfer (REST) es un enfoque arquitectónico empleado en la construcción de aplicaciones web. Según (Fielding, 2000), quien acuñó el término, REST es "un conjunto de restricciones arquitectónicas que, cuando se aplican a los componentes de un sistema distribuido, dan como resultado un sistema con propiedades como: rendimiento, escalabilidad y simplicidad arquitectónica".

REST es un enfoque arquitectónico empleado en la creación de aplicaciones web escalables y de alto rendimiento. Según (Mozilla, 2022a) sobre REST, "en una arquitectura RESTful, los recursos son identificados por una URI única, y la manipulación de recursos se realiza mediante métodos HTTP estándar, tales como: GET, POST, PUT, DELETE".

API REST

La arquitectura de API REST (Representational State Transfer) es ampliamente utilizada en el desarrollo de aplicaciones web modernas. REST es un estilo arquitectónico que se basa

en los principios fundamentales de la web. Una API REST consiste en un conjunto de reglas y convenciones que permiten la comunicación entre diferentes sistemas a través de HTTP (Mozilla, 2022a).

La transferencia de información se efectúa principalmente utilizando el formato JSON, si bien es viable emplear otros tipos de formato como el XML. JSON es un formato ligero y fácil de entender para los desarrolladores y para otros programas software, lo que lo hace ampliamente aceptado en la comunidad de desarrollo de APIs REST (Richardson et al., 2007).

JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) es un formato liviano para el intercambio de datos que tiene una amplia aplicación en la programación web. Según la especificación de JSON, "JSON es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript, aunque hoy se considera un lenguaje independiente de la plataforma" (ECMA International, 2017, p. 1).

JSON es un formato de datos empleado en aplicaciones web para la transmisión eficaz y efectiva de información. Este formato permite la serialización y deserialización de objetos y estructuras de datos complejas para su intercambio a través de la red. Como se indica en la documentación de Mozilla sobre JSON, "el formato JSON es fácilmente legible tanto por humanos como por máquinas, y es compatible con la mayoría de los lenguajes de programación modernos" (Mozilla, 2022b).

HTML

HTML (Lenguaje de Marcado de Hipertexto) es el lenguaje fundamental utilizado para crear páginas web. Originalmente, HTML se diseñó como un lenguaje para describir semánticamente documentos científicos. Su diseño general, sin embargo, ha permitido su

adaptación, en los años siguientes, para describir una serie de otros tipos de documentos e incluso aplicaciones (WHATWG, s. f.).

Mediante el lenguaje HTML, los programadores pueden elaborar y organizar páginas web que incorporan elementos multimedia, como texto, imágenes y enlaces. La habilidad de construir estructuras de contenido y diseño en HTML es fundamental para realizar la vivencia del usuario, facilitando una visualización y acceso óptimos a la información en la web.

CSS

CSS (Cascading Style Sheets) es “un lenguaje utilizado para definir el estilo y la presentación de un documento HTML. Este lenguaje permite aplicar formatos como colores, fuentes, márgenes y diseños a los elementos de una página web” (W3Schools, s. f.). En palabras de (Meyer, 2006), CSS se ha convertido en una herramienta esencial y poderosa en el ámbito del diseño web, ya que permite transformar la presentación de los documentos y separar claramente el contenido de la presentación. Esta separación proporciona una ventaja significativa al facilitar la administración y modificación de estilos, lo que permite a los desarrolladores realizar cambios rápidos y eficientes en el aspecto visual de una página web.

La utilización de CSS ha generado un impacto revolucionario en el proceso de diseño y desarrollo de la apariencia visual de los sitios web. Como menciona (Duckett, 2011), CSS proporciona la capacidad de establecer reglas y estilos de manera individual para cada elemento, así como mediante la aplicación de selectores que agrupan elementos similares. Esta característica otorga un mayor nivel de control y flexibilidad en la presentación de los contenidos, lo que resulta en una mejora significativa de la experiencia del usuario.

JavaScript

JavaScript es un lenguaje de programación extensamente utilizado en el dominio del desarrollo web, conocido por su habilidad para infundir interactividad y dinamismo en las

páginas. Según (Flanagan, 2011), JavaScript se ha vuelto indispensable en la experiencia actual de navegación en la web, al permitir la creación de funciones y scripts que se ejecutan directamente en el navegador del usuario. Esta capacidad de interactuar en el lado del cliente ha generado un cambio significativo en la forma en que se diseñan y presentan los contenidos en línea.

En adición, JavaScript despliega un papel destacado en el ámbito del desarrollo de aplicaciones web y móviles, llegando a consolidarse como el lenguaje de programación más ampliamente utilizado en la historia y en la actualidad (W3Techs, 2023). Este lenguaje ofrece la posibilidad de crear aplicaciones interactivas y responsivas, lo cual ha impulsado la tendencia hacia el desarrollo de aplicaciones web de mayor envergadura y complejidad (MDN, 2023).

Backend

El backend en aplicaciones web se define como la parte del desarrollo ubicada en el servidor que se encarga de proporcionar la funcionalidad y el procesamiento de datos en una aplicación web. Esta área abarca la infraestructura subyacente, así como las bases de datos y los lenguajes de programación utilizados para procesar las solicitudes de los usuarios (Rea Chamorro, 2022).

No es accesible al usuario final y se centra en la lógica y las operaciones necesarias para procesar y almacenar datos, así como gestionar la comunicación entre el frontend y las bases de datos (Pérez Ibarra et al., 2021). El backend sirve como columna vertebral para las aplicaciones web, manejando tareas como la validación de datos, la seguridad y la integración con servicios externos.

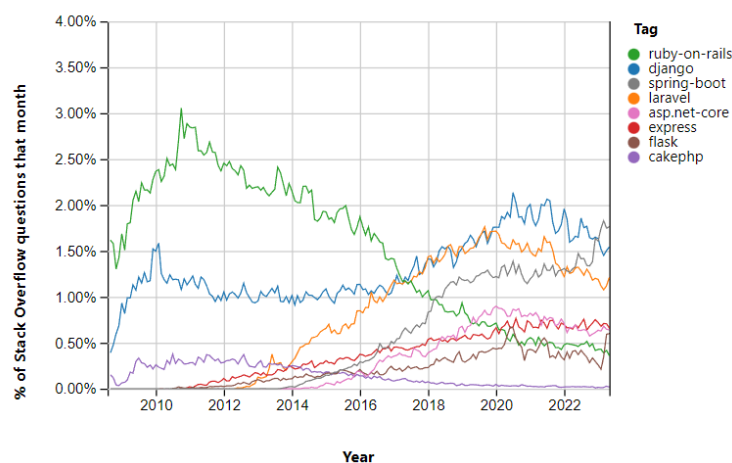
En el desarrollo del backend, se requiere considerar el uso de lenguajes de programación como Java, C#, PHP (Pérez Ibarra et al., 2021). Además, es importante destacar la utilización de frameworks, que son herramientas que agilizan el desarrollo del

backend en aplicaciones web. Entre los frameworks más reconocidos para backend se encuentran Django, Express JS, Spring Boot y Laravel (Stack Overflow, 2023), tal como se aprecia en la Figura 4.

Estos frameworks “proporcionan una estructura y funcionalidades predefinidas que simplifican el proceso de desarrollo, permitiendo a los desarrolladores enfocarse en la implementación de la lógica específica de la aplicación sin tener que preocuparse por aspectos técnicos más complejos” (Kaluža et al., 2019).

Figura 4

Consultas referentes a frameworks para backend en porcentaje por mes



Nota: El gráfico muestra el porcentaje de consultas de los principales frameworks para backend generado por StackOverflow (2023). (Fuente: <https://insights.stackoverflow.com/trends>)

Frontend

Frontend en aplicaciones web, se refiere al desarrollo del lado del cliente responsable de la presentación visual y la interacción del usuario de un sitio web o aplicación web. “Esta tecnologías y lenguajes de programación vienen implementados en los distintos navegadores web que existen, que son interpretadores de estos códigos” (Aranda Balaguera, 2018).

En el desarrollo de frontend, “es necesario tener en cuenta el uso de lenguajes de programación como JavaScript, que se combina con HTML y CSS para crear la interfaz de usuario de una aplicación web. Además, se utilizan bibliotecas como JQuery y AnimateCSS, que son herramientas básicas en el desarrollo de frontend” (Pérez Ibarra et al., 2021).

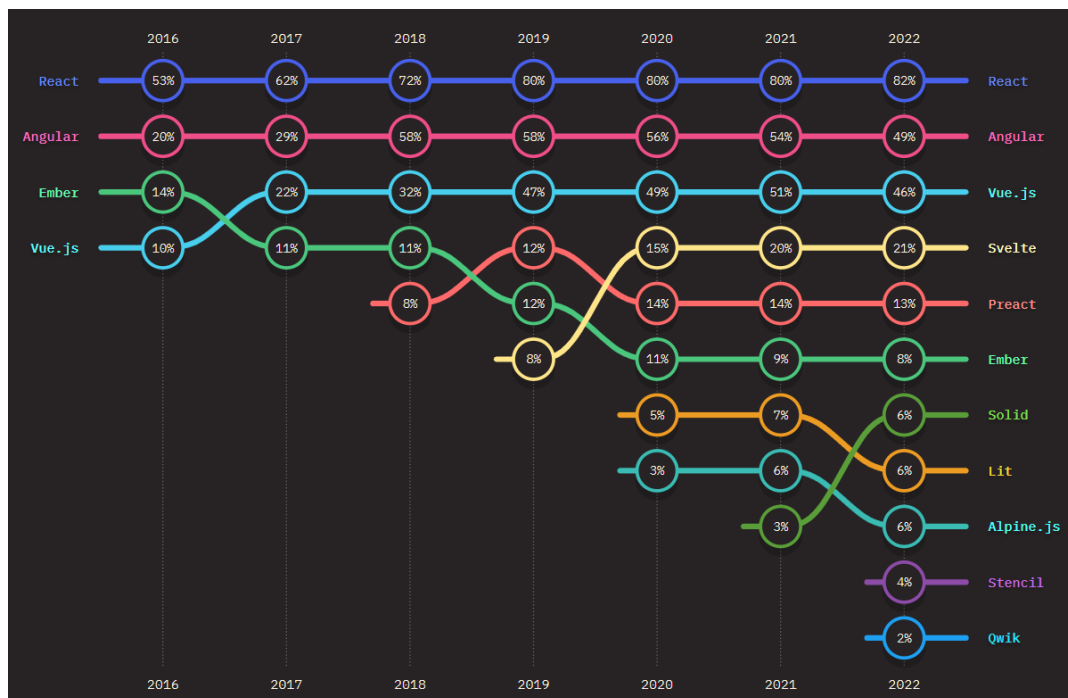
Igual que en el caso de desarrollo de aplicaciones backend, existen frameworks que facilitan el desarrollo y la implementación del frontend. Se pueden identificar varios frameworks de amplia adopción por parte de la comunidad de desarrolladores. Entre ellos, destacan Vue.js, React.js y Angular, los cuales han logrado consolidarse como las opciones más utilizadas en la actualidad (State of JavaScript, 2023).

Estos frameworks han demostrado mantener una prominente posición en el mercado por más de seis años, evidenciando su popularidad y el respaldo que han adquirido en la comunidad de desarrolladores (State of JavaScript, 2023), como se demuestra en la Figura 5.

Los frameworks mencionados previamente proveen estructuras y herramientas prediseñadas que agilizan la labor de desarrollar el frontend, habilitando a los programadores para elaborar interfaces de usuario interactivas y eficientes de manera más efectiva y productiva.

Figura 5

Ranking de porcentaje de uso de frameworks para frontend



Nota: El gráfico muestra el porcentaje de uso de los frameworks más populares para desarrollo frontend generado por State of JS. (Fuente: <https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/>)

Framework

Un framework o marco de trabajo conforme a (Gamma et al., 1994) es un “conjunto de clases cooperantes que conforman un diseño reutilizable para una clase específica de software, además de ser una estructura base que se puede personalizar a medida que el desarrollador lo adapta a las necesidades específicas de su proyecto”.

En el desarrollo web, existen frameworks tanto para el frontend como para el backend. Un framework para el frontend se centra en la capa de presentación y la interacción del usuario con la interfaz. De acuerdo con (Rathinam, 2023), un framework frontend, como React, Vue o

Angular, ofrece componentes reutilizables, estructura estandarizada, enrutamiento de páginas y manejo de estado, lo que agiliza el proceso de desarrollo y mejora la experiencia del usuario en aplicaciones de una sola página.

Por otro lado, un framework para el backend se enfoca en el manejo de la lógica de negocio y la comunicación con bases de datos y servicios externos. Según (Kaluža et al., 2019), un framework backend, como Django o Ruby on Rails, proporciona una estructura organizada para la implementación de la lógica del servidor, el enrutamiento de solicitudes y la gestión de bases de datos, lo que facilita la construcción de aplicaciones escalables y seguras.

Node.js

Node.js es un entorno de tiempo de ejecución de JavaScript que permite a los desarrolladores construir aplicaciones de servidor en JavaScript. Según la documentación oficial de (Node.js, 2023), el entorno proporciona una plataforma para el desarrollo de aplicaciones de red escalables, así como también una arquitectura de E/S orientada a eventos y no bloqueante, lo que permite una programación eficiente y sin bloqueos.

Desde su lanzamiento en 2009, Node.js ha ganado popularidad y ha ido ganando popularidad en la comunidad de desarrollo de software. En la actualidad es utilizado por empresas como Netflix, LinkedIn, Uber y PayPal para desarrollar aplicaciones web de alto rendimiento (Trio, 2023). Node.js se ha convertido en una herramienta de desarrollo asíncrona potente y cada vez más popular, que se utiliza para crear aplicaciones JavaScript altamente escalables(Node.js, 2023).

Express.js

Express.js es un framework de backend para aplicaciones web, rápido, flexible y minimalista para Node.js. Según la documentación oficial de Express.js, "Es una infraestructura

de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para aplicaciones web y móviles" (Express.js, 2023).

Además, como señala (Vivah, 2019), Express "es un framework web que permite estructurar una aplicación web para manejar múltiples solicitudes http diferentes en una URI específica, de código abierto y flexible diseñado para facilitar mucho el desarrollo de sitios web, aplicaciones web y APIs".

Vue.js

Vue.js es un framework de JavaScript de código abierto para la construcción de interfaces de usuario (UI) y aplicaciones de una sola página (SPA). Según la documentación oficial de (Vue.js, 2023), "el framework se centra en la capa de vista de una aplicación y se integra fácilmente con otras bibliotecas o proyectos existentes. Además, Vue.js utiliza una sintaxis simple y clara para definir componentes y manejar el estado de la aplicación, lo que lo hace fácil de aprender y utilizar".

Vue.js tuvo 7600 estrellas en GitHub en el año 2016, 36700 estrellas en el 2017, más de 100,000 estrellas en el 2018 y más de 203,299 en el año 2023 lo que muestra un claro indicador del crecimiento del framework en la industria. Vue.js cubre la mayoría de las características comunes necesarias para el desarrollo de frontend. Está diseñado con flexibilidad en mente, lo que permite una integración incremental y mejoras en la manipulación de HTML sin requerir un paso de compilación adicional. Además, Vue.js permite la incrustación de componentes web en cualquier página web, ya sea para aplicaciones de escritorio, web, móviles o incluso terminales. Esta capacidad de adaptación lo convierte en una opción versátil para diversos escenarios de desarrollo (Vue.js, 2023).

Variables de entorno dotenv

El archivo `.env` es un archivo de configuración que se utiliza en muchos proyectos de desarrollo de software para almacenar información sensible como claves de acceso, contraseñas y otros datos de configuración. Según la documentación de Laravel, un popular framework de PHP, el archivo `.env` "contiene las variables de entorno del sistema utilizadas por la aplicación" y "se deben mantener fuera de control de versiones" (Laravel, 2023).

Este archivo se utiliza para separar la configuración de la aplicación del código fuente, lo que facilita la gestión de las configuraciones específicas de cada entorno (por ejemplo, desarrollo, prueba o producción). Además, el archivo `.env` se utiliza para cargar estas variables de entorno en la aplicación, lo que facilita el acceso a ellas desde cualquier parte del código.

Base de Datos No SQL

Las bases de datos NoSQL, también conocidas como bases de datos no relacionales, "constituyen un tipo de sistemas de gestión de bases de datos que se aparta del modelo relacional clásico. Estas bases de datos han sido concebidas para manejar volúmenes significativos de información desestructurada o parcialmente estructurada, tales como documentos, videos, imágenes e interacciones en plataformas sociales" (Gyorödi et al., 2015).

Contrario a las bases de datos convencionales, "las bases de datos NoSQL no están atadas a un esquema rígido, lo que les confiere una mayor flexibilidad y capacidad de adaptación a las modificaciones en las estructuras de datos. Utilizan diferentes modelos de datos, como los modelos clave-valor, orientados a documentos, de gráficos y de familia de columnas, para almacenar y gestionar datos" (Gyorödi et al., 2015).

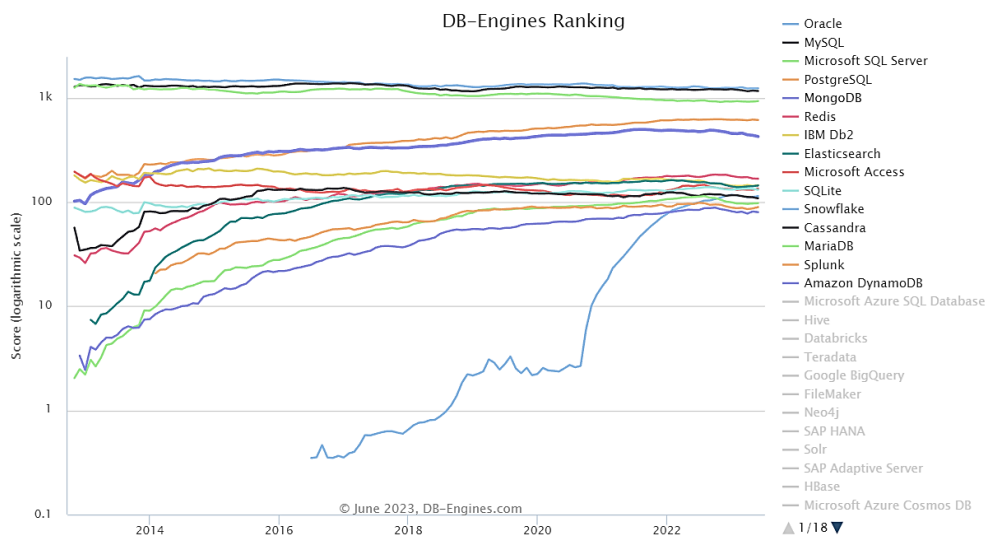
Las bases de datos NoSQL podría clasificarse en 4 tipos (Sadalage, 2014):

- Bases de datos clave-valor: cuáles son los almacenes de datos NoSQL más simples de usar, desde una perspectiva de API. Las más populares son Redis, Riak, etc.

- Base de datos de documentos: que almacenan y recuperan documentos como XML, JSON, BSON, etc. La base de datos de documentos más popular es MongoDB, que proporciona un lenguaje de consulta enriquecido.
- Almacenes de familias de columnas: estas bases de datos almacenan datos en familias de columnas como filas que tienen muchas columnas asociadas con una clave de fila. Uno de los más populares es Cassandra.
- Bases de datos gráficas: le permite almacenar entidades y relaciones entre estas entidades. Hay muchas bases de datos de gráficos, pero entre este tipo de base de datos podemos mencionar OrientDB, FlockDB, etc.

Las bases de datos NoSQL “se han vuelto cada vez más populares en los últimos años debido a su capacidad para manejar grandes volúmenes de datos y escalar horizontalmente, así como a su compatibilidad con la informática en la nube, asimismo, brindan un rendimiento superior” (AWS, 2023b).

MongoDB es reconocida como una de las bases de datos NoSQL más ampliamente utilizadas en la actualidad, ofreciendo versiones tanto en la nube como en entornos locales. Conforme al informe actualizado de (DB-Engines Ranking, 2023), MongoDB ocupa el quinto puesto en la clasificación de los sistemas de gestión de bases de datos más destacados por debajo de Oracle, MySQL, SQL Server y PostgreSQL que forman parte del grupo de base de datos SQL. Este posicionamiento evidencia su popularidad y la convierte en la opción más favorecida dentro del ámbito de las bases de datos NoSQL, ver Figura 6.

Figura 6*Ranking de sistemas de gestión de bases de datos*

Nota: El gráfico muestra la clasificación de los sistemas de gestión de bases de datos por popularidad generado por DB-Engines (2023). (Fuente: https://db-engines.com/en/ranking_trend)

Git

Git es "un sistema de control de versiones distribuido, gratuito y de código abierto diseñado para manejar todo, desde proyectos pequeños hasta muy grandes, con rapidez y eficacia" (Chacon & Straub, 2014).

A través del empleo de Git, los programadores pueden trabajar en paralelo en múltiples variantes del código, lo que facilita una colaboración eficaz en un proyecto. Es relevante mencionar que Git proporciona una variedad de herramientas y características que agilizan y mejoran el proceso de gestión de versiones, lo cual aporta ventajas al desarrollo de software.

IaaS (Infraestructura As A Service)

La infraestructura como servicio (IaaS) es un modelo de entrega de servicios en la nube que proporciona a los usuarios acceso a recursos informáticos virtualizados como servidores, redes y almacenamiento. Según una definición de NIST (National Institute of Standards and Technology), IaaS es "el suministro de capacidades de procesamiento, almacenamiento, redes y otros recursos fundamentales de cómputo, donde el consumidor es capaz de desplegar y ejecutar software arbitrario, incluyendo sistemas operativos y aplicaciones" (Mell & Grance, 2011).

Según lo indicado por (Dawoud et al., 2010), "la informática en la nube depende principalmente de la capa IaaS para proporcionar potencia de procesamiento, almacenamiento de datos y otros recursos compartidos económicos y de pago por uso". Además, "IaaS permite a los usuarios acceder a la última tecnología y recursos de infraestructura sin la necesidad de invertir en hardware costoso" (Goscinski & Brock, 2010). De manera general, la oferta de Infraestructura como Servicio (IaaS) resulta atractiva para las compañías que desean disminuir gastos y elevar la eficacia en la administración de su infraestructura de tecnologías de la información.

Herramientas para desarrollo web

En el ámbito del desarrollo web, resulta imprescindible disponer de herramientas adecuadas que permitan agilizar y optimizar el proceso de creación de aplicaciones y sitios web. En este sentido, se pueden encontrar herramientas para el manejo de bases de datos, entornos de desarrollo integrados, gestión de proyectos, control de versiones, alojamiento y pruebas de la aplicación.

MongoDB Atlas

MongoDB Atlas es una plataforma de base de datos en la nube que ofrece una solución para manejar grandes volúmenes de datos de forma eficiente y escalable. De conformidad con

lo descrito por la compañía, "MongoDB Atlas es la forma más sencilla de implementar, operar y escalar una base de datos MongoDB en la nube" (MongoDB, s. f.).

Además, esta plataforma también ofrece una variedad de características y herramientas que facilitan la administración y monitoreo de la base de datos, tales como "automatización, integración continua, resiliencia ante fallas y cumplimiento normativo" (MongoDB, s. f.).

La plataforma MongoDB Atlas proporciona una serie de funcionalidades y características que contribuyen a asegurar la disponibilidad, escalabilidad y cumplimiento de las normativas de seguridad. A través de su enfoque en la automatización, MongoDB Atlas facilita el mantenimiento del rendimiento de las aplicaciones a medida que estas evolucionan y crecen en escala (AWS, 2023c). Esta solución tecnológica permite a los usuarios gestionar eficientemente la infraestructura subyacente y los recursos necesarios para soportar las demandas cambiantes de las aplicaciones, sin comprometer la seguridad ni la calidad del servicio. Al ofrecer un entorno confiable y flexible, MongoDB Atlas se posiciona como una solución confiable y altamente eficiente para aquellos que buscan garantizar la continuidad operativa, el crecimiento escalable y el cumplimiento normativo en sus entornos de bases de datos (Google Cloud, 2023).

Entorno de Desarrollo Integrado (IDE)

Un Entorno Integrado de Desarrollo (IDE, por sus siglas en inglés, Integrated Development Environment) es una aplicación de software que capacita a los programadores para crear, depurar y ejecutar aplicaciones informáticas dentro de un contexto integrado. Según (AWS, s. f.), "un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente".

Otra definición muy aceptada sobre lo que es un IDE es que "un entorno de desarrollo integrado (IDE) es un software para crear aplicaciones que combina herramientas de desarrollo

comunes en una sola interfaz gráfica de usuario (GUI), herramientas como un editor de código fuente, automatización de compilación local y/o depurador" (RedHat, 2019).

Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft que se ha convertido en una herramienta popular para el desarrollo de aplicaciones web y móviles. Según la documentación de Microsoft, "Visual Studio Code es un editor de código fuente ligero y potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux" (Microsoft, 2021).

Visual Studio Code proporciona una amplia gama de funciones beneficiosas para los programadores, que abarcan desde la integración con Git y otros sistemas de control de versiones, hasta depuración incorporada, autocompletado de código y compatibilidad con diversos lenguajes de programación. Como señala la documentación de Visual Studio Code, "la gran cantidad de extensiones disponibles para Visual Studio Code lo hacen una herramienta muy flexible y adaptable para los desarrolladores" (*Visual Studio Code Introductory Videos*, 2021).

Postman

Postman es una plataforma de colaboración para el desarrollo de API (Interfaz de Programación de Aplicaciones) que permite a los desarrolladores crear, probar, documentar y compartir fácilmente API. Según (Postman, s. f.), "Postman es una herramienta de desarrollo de API que permite a los usuarios crear, compartir, probar y documentar API".

Postman ofrece una variedad de características útiles para los desarrolladores, incluyendo la capacidad de enviar y recibir solicitudes HTTP y HTTPS, la creación de entornos de prueba y la automatización de pruebas de API. Además, Postman permite a los

desarrolladores compartir fácilmente las API con sus compañeros de equipo y clientes a través de su plataforma de colaboración.

Github

GitHub es una plataforma web de control de versionamiento y gestión de proyectos de desarrollo de software, que se ha convertido en un componente fundamental para la colaboración y el control de versiones en la comunidad de desarrollo. Según, (Chacon & Straub, 2014), GitHub, también proporciona alojamiento de proyectos de desarrollo de software, y se ha convertido en un punto de encuentro para millones de desarrolladores quienes almacenan sus repositorios Git en esta plataforma, además de numerosos proyectos de código abierto.

De acuerdo con (Dabbish et al., 2012), el uso de GitHub ha cambiado la forma en que los desarrolladores trabajan y colaboran en proyectos, y ha permitido un mayor nivel de transparencia y colaboración en la comunidad de desarrolladores de software.

Jira

Jira es “una herramienta de gestión de proyectos utilizada por equipos de desarrollo de software para planificar, rastrear y administrar el progreso de los proyectos. El software se basa en una metodología ágil y es una de las soluciones de gestión de proyectos ágiles más buscadas” (*What is JIRA?*, 2022).

Por otro lado, siguiendo a (Jiménez, 2019) Jira es útil en la gestión de proyectos ágiles, ya que proporciona funcionalidades de gestión de incidencias y administración de tareas, permite la asignación de tareas y el seguimiento de la misma, y también facilita la comunicación entre los miembros del equipo.

AWS

En el ámbito de la tecnología de la información, Amazon Web Services (AWS) se posiciona como una destacada plataforma de computación en la nube, brindando una amplia gama de servicios que engloban desde recursos de almacenamiento y bases de datos hasta soluciones de escalabilidad, para respaldar el crecimiento y desarrollo de las organizaciones en el entorno digital actual (*¿Qué es AWS?*, 2023). La confiable y segura infraestructura global de AWS ha ganado reconocimiento, convirtiéndose en una elección popular tanto para pequeñas como para grandes empresas que buscan aprovechar los beneficios de la computación en la nube (*¿Qué es AWS?*, 2023).

La plataforma de AWS permite a las organizaciones capitalizar la elasticidad inherente a la nube, lo cual implica la capacidad de escalar rápidamente los recursos en respuesta a las fluctuaciones de la demanda, generando así mejoras en el rendimiento y en la eficiencia operativa (Sosinsky, 2010). En adición, AWS ofrece una extensa variedad de servicios, incluyendo Amazon EC2 para la computación, Amazon S3 para el almacenamiento y Amazon RDS para la gestión de bases de datos, por mencionar solo algunos ejemplos (*¿Qué es AWS?*, 2023).

AWS EC2

Amazon Elastic Compute Cloud (EC2) es un servicio central de la plataforma de computación en la nube proporcionada por Amazon Web Services (AWS), el cual ofrece una capacidad informática escalable y flexible (AWS, 2023a). Mediante EC2, los usuarios pueden lanzar y administrar instancias virtuales, lo que les permite ejecutar aplicaciones y servicios de software en un entorno virtualizado (Buyya et al., 2011).

El funcionamiento de EC2 se basa en el concepto de máquinas virtuales (VM) y hace uso de tecnologías de virtualización para crear y gestionar dichas instancias (AWS, 2023a). Los usuarios tienen la posibilidad de seleccionar el tipo de instancia que mejor se ajuste a sus

necesidades en términos de capacidad de procesamiento, memoria, almacenamiento y rendimiento de red (Buyya et al., 2011). Asimismo, EC2 ofrece opciones de escalado automático que permiten ajustar de manera automática la capacidad de acuerdo con la demanda de carga de trabajo (AWS, 2023a).

Ubuntu Server

Ubuntu Server es una distribución de Linux basada en el sistema operativo Ubuntu y diseñada específicamente para entornos de servidores. Es una opción popular para empresas y organizaciones que buscan una plataforma confiable y segura para ejecutar sus servicios y aplicaciones en la nube. Ubuntu Server ofrece una amplia variedad de características y herramientas orientadas a la administración de servidores, como una instalación y configuración sencillas, un sólido sistema de gestión de paquetes y una amplia compatibilidad con software de servidor (Ubuntu, s. f.).

Esta distribución de Linux se destaca por su enfoque en la seguridad y la estabilidad, con actualizaciones regulares y un sistema de soporte a largo plazo (LTS, por sus siglas en inglés) que garantiza actualizaciones de seguridad y correcciones de errores durante un período extendido (Ubuntu, s. f.). Además, Ubuntu Server cuenta con una amplia comunidad de usuarios y desarrolladores que brindan soporte y contribuyen a la mejora continua del sistema operativo (Rankin & Hill, 2013).

Apache2

Apache HTTP Server, comúnmente conocido como Apache2, es un servidor web ampliamente utilizado en la industria de la tecnología de la información. Desarrollado por la Apache Software Foundation, Apache2 ofrece una plataforma sólida y confiable para alojar sitios web y aplicaciones en línea. Apache2 se destaca por su estabilidad, seguridad y capacidad para manejar altos volúmenes de tráfico web (Apache Software Foundation, 2023).

Una de las características más destacadas de Apache2 es su flexibilidad y extensibilidad. Los usuarios “pueden configurar y personalizar el servidor según sus necesidades específicas, utilizando una amplia variedad de módulos y complementos disponibles” (Apache Software Foundation, 2023). Además, “Apache2 es compatible con diversos sistemas operativos, incluyendo Linux, Windows y macOS, esto lo convierte en una elección adaptable para diversos entornos” (Laurie & Laurie, 2002).

La seguridad es una prioridad en Apache2, y se han implementado numerosas medidas para proteger el servidor y los datos transmitidos. Esto incluye características como el cifrado SSL/TLS, autenticación de usuarios y acceso controlado a directorios y archivos (Laurie & Laurie, 2002). Además, Apache2 cuenta con una comunidad activa que proporciona actualizaciones de seguridad y parches para garantizar la protección continua del servidor (Apache Software Foundation, 2023).

GoDaddy

GoDaddy es una corporación estadounidense especializada en alojamiento web y registro de nombres de dominio. Con una amplia gama de servicios, GoDaddy se posiciona como la plataforma más grande del mundo para emprendedores de todas partes del planeta. Según la compañía, su objetivo principal es proveer a sus clientes de una plataforma integral en la que puedan nombrar sus ideas, crear marcas atractivas y sitios web atractivos, atraer clientes mediante marketing digital y social, y administrar sus trabajos de manera efectiva (GoDaddy, s. f.).

Según una publicación de (Williams, 2023), "GoDaddy es conocido por sus precios bajos y su amplia variedad de productos". La empresa ha sido reconocida por su fácil uso y sus opciones de personalización, lo que la hace atractiva para pequeñas empresas y emprendedores.

Documentación y proceso de desarrollo de software

En el contexto del desarrollo de software, la metodología Scrum ha surgido como un enfoque altamente efectivo para gestionar proyectos de manera ágil y colaborativa. Esta metodología se basa en la utilización de varios artefactos clave que desempeñan un papel fundamental en la organización y el seguimiento del proceso de desarrollo del equipo.

Entre estos artefactos se encuentran el Product Backlog, que representa una lista priorizada de los requisitos y funcionalidades del proyecto, y el Sprint Backlog, que define las tareas específicas a abordar durante cada sprint. Además de estos artefactos, se emplea documentación extra como el uso de diagramas para facilitar la organización y maximizar el rendimiento del equipo de desarrollo.

El enfoque Scrum y sus artefactos permiten una mayor adaptabilidad, transparencia y colaboración, lo que impulsa la eficiencia y la calidad del producto final.

Diagramas de Software UML

Los diagramas de software son herramientas valiosas en el proceso de desarrollo de software para comunicar y visualizar la arquitectura y el diseño de un sistema. También forman parte de la documentación del software y están presentes durante todo el ciclo de vida del software (Montalvo, 2021).

Los diagramas UML (Unified Modeling Language) son una herramienta gráfica utilizada en el modelado de sistemas de software, a la vez que propone estándares para describir la estructura y el comportamiento de los sistemas.

UML propone varios modelos de diagramas, entre ellos se pueden mencionar los diagramas de clases que muestra un conjunto de clases, interfaces y colaboraciones y sus relaciones, diagramas de interacción que muestra la interacción que existe entre un conjunto de objetos y sus relaciones, y diagramas de casos de uso que muestra un conjunto de casos de

uso y sus respectivos actores organizando los comportamientos del sistema (Booch et al., 2005).

Pruebas de Aceptación de Usuario

Las pruebas de aceptación del usuario de software son un conjunto de pruebas realizadas para verificar si el software cumple con los requisitos y expectativas del usuario final. Según (Pressman & Maxim, 2014), “estas pruebas se llevan a cabo para determinar si el software cumple con las necesidades y expectativas del cliente y para garantizar que el software se entregue con la calidad deseada”. Estas pruebas se realizan después de que se hayan completado las pruebas de integración y se han resuelto los problemas identificados en las pruebas de sistema.

En general, “las pruebas de aceptación del usuario están diseñadas para probar la funcionalidad del software, su usabilidad, su rendimiento y su capacidad para satisfacer las necesidades del usuario final. De esta manera, se garantiza que el software entregado sea de calidad y cumpla con los requisitos del usuario” (Sommerville, 2015).

Despliegue de software

El despliegue de software es el proceso de liberar y poner en funcionamiento una aplicación en un entorno de producción. Este es el proceso de hacer que el sistema esté disponible para sus usuarios, transferir datos de los sistemas existentes y establecer comunicaciones con otros sistemas en el entorno. El proceso culmina con una "salida en vivo" después de la cual los usuarios comienzan a usar el sistema para respaldar su trabajo (Sommerville, 2015).

Capítulo III

Análisis y Planificación del Desarrollo

Para lograr un desarrollo de software que satisfaga las expectativas de los involucrados, es necesario llevar a cabo un análisis previo para determinar los requisitos y funcionalidades a implementar. El capítulo actual se encarga de establecer los requisitos necesarios para el software y de planificar el desarrollo correspondiente.

Obtención de Requerimientos de Software

Con el propósito de obtener los requerimientos del software, se procedió a recopilar información mediante el uso de una serie de entrevistas y la observación del funcionamiento de diversas aplicaciones con un enfoque similar. Este proceso tiene como objetivo principal comprender las necesidades de los stakeholders involucrados, entre los que se encuentran el Scrum Master (Edison Lascano), el Product Owner (Mauro Narváez), el cliente (Eric Lara), y el Development Team (Steven Chinlle, Michael Cobacango). Los requerimientos resultantes se desarrollaron de manera flexible, con la intención de permitir cierto grado de variación durante el proceso de desarrollo del sistema. Pero al mismo tiempo, se evitó ambigüedad en sus descripciones con el propósito de que los requerimientos no se salgan del alcance del proyecto.

La primera reunión tuvo lugar el día 21 de octubre de 2022 utilizando la plataforma Google Meet. Durante dicha reunión, el propietario del centro religioso, desempeñando el rol de cliente, presentó una descripción general de la situación actual de la organización. En el transcurso del encuentro, se llevó a cabo un análisis exhaustivo de los requisitos que debían ser abordados. Como resultado de la reunión, se generó un documento oficial que registra los puntos tratados, el cual se adjunta como referencia en la Figura 7. Para acceder a un listado completo de las reuniones realizadas, ver el APÉNDICE A.1.

Figura 7

Acta de reunión: Primera Reunión

ACTA DE REUNIÓN 00: PRIMERA REUNIÓN

Fecha: 21/11/2022
Hora de inicio: 21:00

Asistentes:

Nombre	Cargo
Eric Lara	Stakeholder - Client
Mauro Narváez	Product Owner
Edison Lascano	Scrum Master
Edison Chinlle	Developer
Michael Cobacango	Developer

Puntos a tratar:

Bienvenida al proyecto y presentación.
Visión general de la organización.
Contexto del proyecto.
Lista de requerimientos.
Exposición de ideas.

Observaciones/Sugerencias/Descripciones relevantes:

Se recomienda elaborar las historias de usuario de manera flexible, para que se puedan analizar posibles cambios o modificaciones a las mismas, además, conversar temas que puedan quedar pendientes.

Capturas de pantalla:

Se resuelve:

Realizar un levantamiento de requerimientos más detallado, después de dicho levantamiento iniciar con el proyecto inmediatamente una vez generadas las historias de usuario y creado el product backlog.

Las reuniones subsiguientes se dedicaron a recopilar documentos proporcionados por los interesados, los cuales contenían datos generales relacionados con el modelo de negocio y desarrollo del proyecto. Durante estas reuniones, se consideraron varias páginas web que presentaban funcionalidades similares a las requeridas por los participantes. A partir de estas reuniones y recopilación de datos, se crearon las historias de usuario que sirven como base para la elaboración del Product Backlog y los Sprint Backlogs.

Análisis de los Requerimientos y Creación del Product Backlog

Los requerimientos expresados por el Product Owner y el cliente fueron convertidos en historias de usuario que contienen un identificador, nombre, prioridad, riesgo en desarrollo,

personal encargado, descripción y validación. Estas historias proporcionan una descripción concisa y precisa de las necesidades del usuario y su comportamiento general.

Se recopilaron en total cincuenta y tres historias de usuario, las cuales fueron clasificadas en categorías de prioridad alta, media y baja. Además, se tuvo en consideración el nivel de riesgo asociado al desarrollo de cada historia de usuario. La Tabla 2 contiene el listado de historias de usuario.

Tabla 2

Historias de usuario

Número de historia	Descripción
UH01	Visualizar misión
UH02	Agregar misión
UH03	Modificar misión
UH04	Eliminar misión
UH05	Visualizar valores centrales
UH06	Agregar valores centrales
UH07	Modificar valores centrales
UH08	Eliminar valores centrales
UH09	Visualizar ubicación
UH10	Contactos
UH11	Cambiar idioma
UH12	Visualizar imágenes
UH13	Agregar imágenes
UH14	Modificar imágenes
UH15	Eliminar imágenes
UH16	Visualizar video de presentación
UH17	Modificar video de presentación

Número de historia	Descripción
UH18	Visualizar video en vivo de YouTube
UH19	Modificar enlace del video de YouTube
UH20	Escuchar predicaciones
UH21	Agregar predicaciones
UH22	Eliminar predicaciones
UH23	Visualizar eventos importantes
UH24	Registrar evento importante
UH25	Eliminar eventos
UH26	Visualizar próximos eventos en el calendario
UH27	Agregar eventos al calendario
UH28	Modificar eventos
UH29	Eliminar eventos
UH30	Visualización de eventos en vivo
UH31	Modificar evento en vivo
UH32	Visualizar redes sociales
UH33	Agregar redes sociales
UH34	Modificar redes sociales
UH35	Eliminar redes sociales
UH36	Visualizar grupos de vida
UH37	Agregar grupo de vida
UH38	Modificar grupo de vida
UH39	Eliminar grupo de vida
UH40	Visualizar servicios
UH41	Agregar servicios
UH42	Modificar servicios
UH43	Eliminar servicios

Número de historia	Descripción
UH44	Visualizar feligreses que requieran servicios de oración o testimonio
UH45	Solicitar petición de oración o testimonio
UH46	Visualizar nuevos feligreses
UH47	Registrarse como nuevo feligrés
UH48	Visualizar usuario feligreses que recibieron a Cristo por primera vez
UH49	Registrarse si recibiste a Cristo por primera vez
UH50	Visualizar usuarios feligreses interesados en los grupos de vida
UH51	Registrarse como interesado de los grupos de vida
UH52	Visualizar donaciones de los usuarios feligreses
UH53	Registrar donaciones

La información detallada de las cincuenta y tres historias de usuario se encuentra documentada en el APÉNDICE B.1. Estas historias fueron revisadas por el Product Owner, el Development Team y el cliente durante la reunión llevada a cabo el día 12 de enero de 2023, el resumen de los puntos tratados se encuentra adjunto en el APÉNDICE A.2.

Una vez finalizada la revisión de las historias de usuario, se inicia la narración de los elementos que componen el Product Backlog, siendo el Development Team quien asume los roles de Product Owner (Michael Cobacango) y Scrum Master (Edison Chinlle). Durante este proceso, se realiza un análisis con el objetivo de categorizar adecuadamente las historias en función del esfuerzo requerido, la cohesión y el riesgo de desarrollo asociado. Como resultado de este análisis, se obtiene un total de 9 elementos, ver Tabla 3.

Tabla 3*Product Backlog del Proyecto*

ID	Descripción
ICN-01	Gestión de Información Iglesia Camino Nuevo
ICN-02	Cambio del Idioma del Sitio
ICN-03	Gestión de Contenido Multimedia
ICN-04	Gestión de Redes Sociales
ICN-05	Gestión de Eventos
ICN-06	Gestión de Servicios
ICN-07	Registro y Supervisión de Peticiones de Oración
ICN-08	Gestión de Grupos de Vida
ICN-09	Recepción de Donaciones

Creación de los Sprint Backlog

Se ha establecido un plan para llevar a cabo el total de nueve Sprints descritos en el Product Backlog, con una duración fija de 14 días para la realización de cada uno. Se ha realizado un ajuste cuidadoso en la asignación de puntos de esfuerzo para cada Sprint, con el objetivo de distribuirlos de manera equilibrada y moderada a lo largo del proyecto.

El primer sprint, denominado "Gestión de Información de la Iglesia Camino Nuevo", consta de un conjunto de once historias de usuario que abordan el manejo de información relevante para la organización, como se detalla en la Tabla 4.

Tabla 4

Sprint Backlog Del Sprint 1 "Gestión De Información De La Iglesia Camino Nuevo"

Sprint 1	
Gestión de Información	Descripción
13-mar/26-mar	
UH01	Visualizar misión
UH02	Agregar misión
UH03	Modificar misión
UH04	Eliminar misión
UH05	Visualizar valores centrales
UH06	Agregar valores centrales
UH07	Modificar valores centrales
UH08	Eliminar valores centrales
UH09	Visualizar ubicación
UH10	Contactos
UH16	Visualizar video de presentación

El segundo sprint, titulado "Cambio del Idioma del sitio", se centra en una única historia de usuario relacionada con la modificación del idioma utilizado en la aplicación web, como se muestra en la Tabla 5.

Tabla 5

Sprint Backlog Del Sprint 2 "Cambio del Idioma del sitio"

Sprint 2	
Cambio de Idioma	Descripción
27-mar/09-abr	
UH11	Cambiar idioma

El tercer sprint, denominado "Gestión de Contenido Multimedia", comprende ocho historias de usuario que abordan las características del manejo de contenido audiovisual, según se expone en la Tabla 6.

Tabla 6

Sprint Backlog Del Sprint 3 "Gestión Del Contenido Multimedia"

Sprint 3 Gestión de Contenido Multimedia 10-abr/23-abr	Descripción
UH12	Visualizar imágenes
UH13	Agregar imágenes
UH14	Modificar imágenes
UH15	Eliminar imágenes
UH17	Modificar video de presentación
UH20	Escuchar predicaciones
UH21	Agregar predicaciones
UH22	Eliminar predicaciones

El cuarto sprint, denominado "Gestión de Redes Sociales", incluye cuatro historias de usuario relacionadas con la integración de redes sociales en la aplicación, como se especifica en la Tabla 7.

Tabla 7

Sprint Backlog Del Sprint 4 "Gestión De Redes Sociales"

Sprint 4	
Gestión de Redes Sociales	Descripción
24-abr/07-may	
UH32	Visualizar redes sociales
UH33	Agregar redes sociales
UH34	Modificar redes sociales
UH35	Eliminar redes sociales

El quinto sprint, titulado "Gestión de Eventos", consta de once historias de usuario que se enfocan en la gestión de eventos a realizarse, como se precisa en la Tabla 8.

Tabla 8

Sprint Backlog Del Sprint 5 "Gestión De Eventos"

Sprint 5	
Gestión de Eventos	Descripción
8-may/21-may	
UH18	Visualizar video en vivo de YouTube
UH19	Modificar enlace del video de YouTube
UH23	Visualizar eventos importantes
UH24	Registrar evento importante
UH25	Eliminar eventos
UH26	Visualizar próximos eventos en el calendario
UH27	Agregar eventos al calendario

Sprint 5	
Gestión de Eventos	Descripción
8-may/21-may	
UH28	Modificar eventos
UH29	Eliminar eventos
UH30	Visualización de eventos en vivo
UH31	Modificar evento en vivo

El sexto sprint, denominado "Gestión de Servicios", comprende cuatro historias de usuario que se refieren a la organización de los servicios que se ofrecen dentro de la entidad, como se establece en la Tabla 9.

Tabla 9

Sprint Backlog Del Sprint 6 "Gestión De Servicios"

Sprint 6	
Gestión de Servicios	Descripción
22-may/4-jun	
UH40	Visualizar servicios
UH41	Agregar servicios
UH42	Modificar servicios
UH43	Eliminar servicios

El séptimo sprint, titulado "Registro y Supervisión de Peticiones de Oración", consta de seis historias de usuario que tratan sobre las peticiones realizadas por los usuarios hacia la iglesia con el propósito de ser escuchados, según se indica en la Tabla 10.

Tabla 10

Sprint Backlog Del Sprint 7 "Registro y Supervisión De Peticiones De Oración"

Sprint 7	
Registro y Supervisión	
de Peticiones de	Descripción
Oración	
05-jun/18-jun	
UH44	Visualizar feligreses que requieran servicios de oración o testimonio
UH45	Solicitar petición de oración o testimonio
UH46	Visualizar nuevos feligreses
UH47	Registrarse como nuevo feligrés
UH48	Visualizar usuario feligreses que recibieron a Cristo por primera vez
UH49	Registrarse si recibiste a Cristo por primera vez

El octavo sprint, denominado "Gestión de Grupos de Vida", incluye seis historias de usuario que se centran en la conformación y administración de grupos que permiten a los usuarios compartir ideas con personas que tienen intereses similares o que se encuentran en cercanía geográfica, como se señala en la Tabla 11.

Tabla 11

Sprint Backlog Del Sprint 8 "Gestión De Grupos De Vida"

Sprint 8	
Gestión de Grupos de Vida	Descripción
19-jun/02-jul	
UH36	Visualizar grupos de vida
UH37	Agregar grupo de vida
UH38	Modificar grupo de vida
UH39	Eliminar grupo de vida
UH50	Visualizar usuarios feligreses interesados en los grupos de vida
UH51	Registrarse como interesado de los grupos de vida

El noveno sprint, titulado "Recepción de donaciones", consta de dos historias de usuario que abordan la administración de las donaciones realizadas por los usuarios hacia la iglesia, como se describe en la Tabla 12.

Tabla 12

Sprint Backlog Del Sprint 9 "Recepción De Donaciones"

Sprint 9	
Recepción de Donaciones	Descripción
03-jul/16-jul	
UH52	Visualizar donaciones de los usuarios feligreses
UH53	Registrar donaciones

Para la gestión del Product Backlog, su desarrollo y control durante los Sprints, se utilizó la plataforma Jira. Esta plataforma proporciona una extensa variedad de funcionalidades que facilitan la organización y seguimiento de las tareas en cada etapa del proyecto fundamentado en Scrum.

En Jira, se creó un tablero personalizado con múltiples columnas interactivas que representan las diferentes etapas de cada tarea, permitiendo una visualización clara y ordenada del flujo de trabajo. Este tablero fue adaptado a las necesidades específicas de cada Sprint, ajustando las etapas y flujos de trabajo de acuerdo con los requisitos del proyecto.

Las tareas se encuentran inicialmente en el product backlog, donde se priorizan y detallan como se observa en la Figura 8. A medida que se inicia cada Sprint, se seleccionan las tareas más relevantes y se mueven a la columna correspondiente en el tablero de Jira. Durante el transcurso del Sprint, el equipo trabaja en las tareas asignadas y actualiza su progreso directamente.

A medida que se completan las etapas del trabajo en cada tarea, se van moviendo a las columnas correspondientes en el tablero de Jira, reflejando visualmente el avance del proyecto. Esto facilita la colaboración entre los miembros del equipo, ya que todos pueden ver el estado actual de cada tarea y el progreso general del proyecto.

Para visualizar el tablero real del proyecto en Jira, se puede acceder a través de la siguiente página web: <https://eschinlle.atlassian.net/jira/software/projects/ICN/boards/1/backlog>. Con Jira, se logrará una gestión eficiente del Product Backlog y un control efectivo del desarrollo en cada Sprint, esto brindará una plataforma integral para la planificación, seguimiento y colaboración en el proyecto.

Figura 8

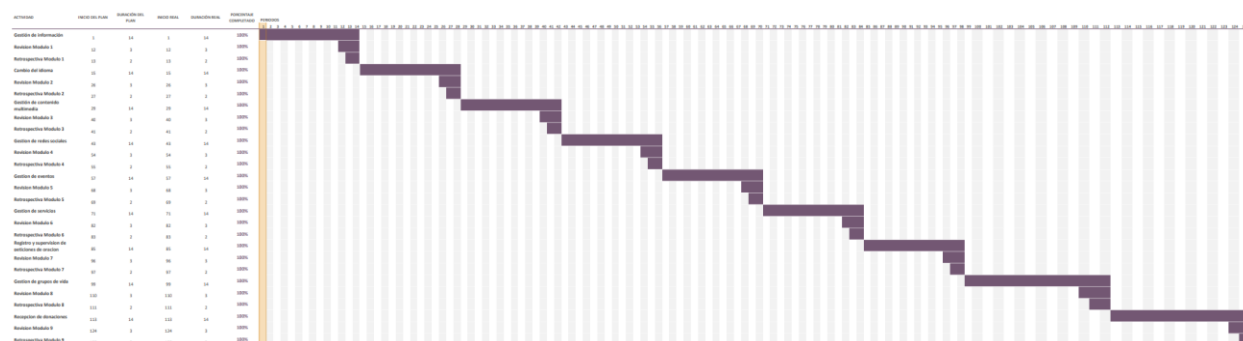
Backlog Jira Real Al Inicio Del Proyecto

▼ Backlog (9 incidencias)		0 0 0	Crear sprint
<input checked="" type="checkbox"/>	ICN-10 Gestión de información	- TAREAS POR HACER ▼	EP
<input checked="" type="checkbox"/>	ICN-11 Cambio de idioma	- TAREAS POR HACER ▼	MC
<input checked="" type="checkbox"/>	ICN-12 Gestión de contenido multimedia	- TAREAS POR HACER ▼	EP
<input checked="" type="checkbox"/>	ICN-13 Gestión de redes sociales	- TAREAS POR HACER ▼	MC
<input checked="" type="checkbox"/>	ICN-14 Gestión de eventos	- TAREAS POR HACER ▼	EP
<input checked="" type="checkbox"/>	ICN-15 Gestión de servicios	- TAREAS POR HACER ▼	MC
<input checked="" type="checkbox"/>	ICN-16 Registro y supervisión de peticiones de oración	- TAREAS POR HACER ▼	EP
<input checked="" type="checkbox"/>	ICN-17 Gestión de grupos de vida	- TAREAS POR HACER ▼	MC
<input checked="" type="checkbox"/>	ICN-18 Recepción de donaciones	- TAREAS POR HACER ▼	EP ...

Con el propósito de avanzar en el proceso de planificación del proyecto, una vez creados el Product Backlog y los Sprint Backlogs, se procede a la elaboración del diagrama de Gantt. Este diagrama proporciona una representación visual de las tareas del proyecto dispuestas de manera cronológica, lo que permite obtener una visión clara y organizada de la totalidad del proyecto. Se establecen fechas específicas para cada tarea ver Figura 9.

Figura 9

Diagrama de Gantt del Proyecto



Plan de Pruebas de Aceptación del Usuario

Las entregas del producto finalizado al concluir cada Sprint serán sometidas a un proceso de aprobación tras la realización de las pruebas de aceptación de usuario. Con este fin, se ha elaborado un plan de pruebas de aceptación, detallado en el APÉNDICE C.1. El plan de pruebas incluye una guía para la ejecución de las pruebas, así como los resultados que se espera obtener durante su ejecución.

Una vez se ha llevado a cabo un análisis de las necesidades, requerimientos y objetivos del proyecto, estudiando a detalle cada aspecto clave para asegurar un enfoque estratégico y efectivo en el desarrollo, se plantea como siguiente etapa la implementación práctica del sistema, dando lugar a la materialización de la planificación previamente realizada.

Capítulo IV

Desarrollo Del Sistema

Tras un riguroso análisis de los requerimientos, se inicia con la fase de desarrollo del sistema. En este capítulo, se presenta el enfoque adoptado, la estructura arquitectónica y las decisiones estratégicas para garantizar eficiencia y escalabilidad. Asimismo, se aborda la configuración del entorno de desarrollo, se detalla las herramientas y directrices establecidas para facilitar un ambiente colaborativo y productivo.

El proceso consta de 9 Sprints, dentro de los cuales se llevan a cabo Daily Meetings como medio fundamental de comunicación y seguimiento del equipo. Se dedica especial atención al proceso de aceptación del usuario para finalizar cada Sprint (Sprint Review), en cada Sprint Review, el cliente desempeña un papel activo en la validación, verificación y aceptación de cada entrega de implementación de software (módulos del sistema).

Por último, se realiza un análisis de las Sprint Reviews para obtener retroalimentación y planificar el siguiente Sprint. Este capítulo brinda una visión integral del desarrollo del sistema, su estructura, implementación y evaluación a lo largo de su ciclo de vida.

Arquitectura De la Aplicación

La arquitectura del software en desarrollo se basa en el patrón cliente-servidor en 3 capas, ver Figura 10. La arquitectura establece una distinción clara entre los elementos del frontend, backend y base de datos, lo cual posibilita un enfoque modular y escalable durante la creación de la aplicación. Estos componentes se alojan en la nube a través del servicio de instancias virtuales de Amazon Web Services (AWS) conocido como EC2.

El frontend se encarga de la presentación y la interacción con el usuario, mientras que el backend se encarga del procesamiento de la lógica de negocio y la comunicación con la base de datos, ver Figura 11. Esta división de responsabilidades facilita la mantenibilidad del sistema y promueve la reutilización de componentes en futuros desarrollos. Además, al tener una capa de presentación independiente, es posible adaptar y mejorar la interfaz de usuario sin afectar el funcionamiento subyacente del backend y la base de datos.

Figura 10

Arquitectura del sistema

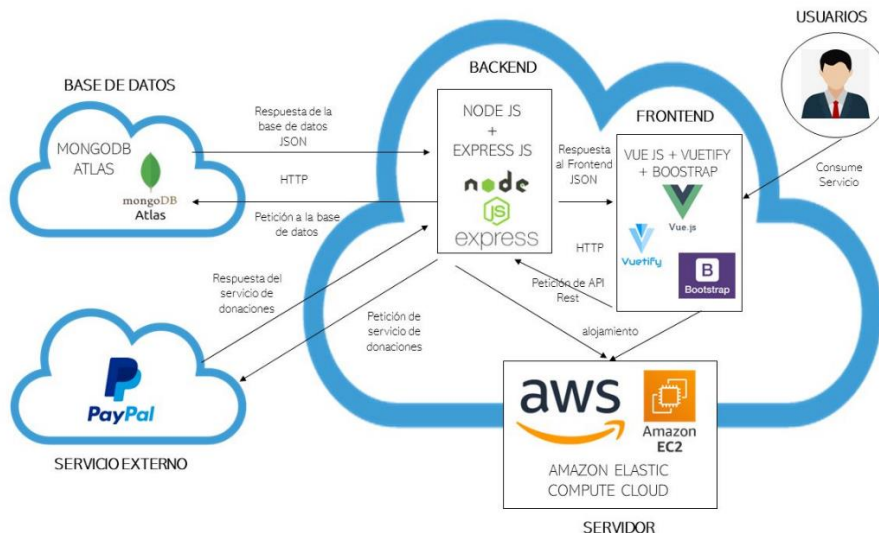
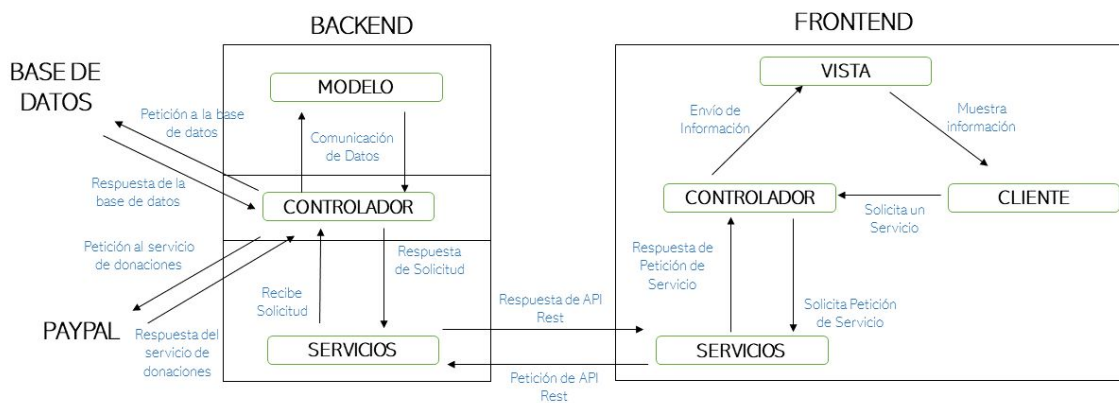


Figura 11

Arquitectura del código



Al efectuar una solicitud a la aplicación web mediante el navegador, la solicitud es procesada y dirigida al backend. El backend, a su vez, se encarga de enrutarla hacia la API local adecuada con el objetivo de gestionar la interacción necesaria con la base de datos. En consecuencia, el backend delega tareas a la capa de presentación y genera una respuesta en

formato JSON que contiene el resultado obtenido. Esta respuesta permite al frontend realizar la verificación de los componentes que experimentan cambios de estado, lo que habilita la renderización de la sección específica de la página necesaria para su adecuada visualización.

La capa de lógica de negocio está compuesta por APIs REST desarrolladas usando Node.js con su framework Express.js, las cuales consisten en conjuntos de nodos escritos en JavaScript. Estos nodos ejecutan la funcionalidad interna de cada API y se encargan de establecer la conexión con la base de datos para realizar consultas y modificaciones, además de permitir la comunicación con el servicio externo PayPal. Las APIs son responsables de gestionar la información de la aplicación, que incluye contenidos generales como títulos, subtítulos, contenido multimedia, pie de página, encabezados, inicio y cierre de sesión.

La capa de acceso a datos desempeña el papel de intermediario entre la lógica de negocio y la base de datos. En este contexto, se emplea un mapeo de objetos a documentos (ODM) específico de Node.js denominado Mongoose, para establecer la conexión con MongoDB Atlas.

La capa de presentación se encuentra conformada por componentes interactivos y dinámicos que contribuyen a proporcionar una experiencia de usuario fluida. Esta capa establece una interacción con la lógica de negocio, por lo que facilita una comunicación eficiente y segura entre ambas. El dinamismo evidente en la capa de presentación se logra mediante la implementación de Vue.js y bibliotecas especializadas en el diseño de interfaces, tales como Vuetify y Bootstrap.

Preparación Del Ambiente De Desarrollo

Una vez que se ha comprendido la arquitectura del sistema, se procede a la creación de dos proyectos. El Frontend se implementa utilizando el framework Vue.js, mientras que el

Backend se desarrolla con Node.js y su framework Express.js. Además, se emplea una base de datos alojada en la nube, MongoDB Atlas.

Estos componentes son desplegados en un entorno de nube mediante el servicio EC2 de AWS, con el propósito de brindar a los usuarios una plataforma permanente para utilizar el software. Asimismo, se configura un ambiente que replica las características del entorno de producción, lo cual posibilita a los stakeholders realizar pruebas y experimentar con el software en condiciones similares a las de su funcionamiento en ambiente de producción.

Se utilizan dos bases de datos distintas: una destinada a pruebas y otra para producción, lo que resulta en un desarrollo más eficiente gracias a que se trabaja de manera independiente en cada una de ellas y se evitan posibles interferencias en los datos.

Para el seguimiento de las actividades, se utiliza la plataforma Jira. En esta plataforma se registra la información relacionada con el Backlog del producto, sprints y tareas. Se selecciona la opción de iniciar un sprint y a medida que se lleva a cabo el desarrollo del módulo, se actualiza el estado de las tareas del sprint, donde a cada Sprint se lo clasifica en "por hacer", "en curso" o "finalizado".

Desarrollo de los Sprints

De acuerdo con la planificación propuesta en el capítulo anterior, cada sprint tiene una duración de 14 días calendario. Durante este periodo se llevan a cabo diversas tareas, entre las que se incluyen el desarrollo de las API's, la implementación de las vistas, la realización de pruebas de las API's y vistas, la ejecución de pruebas de integración y pruebas funcionales, la depuración, y la corrección de errores identificados por los desarrolladores.

Adicionalmente, se incluye la realización de revisiones con los stakeholders, pruebas de usuario y la aceptación por parte del usuario o cliente. En caso de requerir modificaciones,

antes de iniciar el siguiente sprint, se comunicarán los cambios propuestos y se procede a mantener el código para su aprobación final por parte del cliente.

Antes de la ejecución de cada sprint, se llevó a cabo un conjunto de análisis a posibles interfaces de usuario. Estas propuestas fueron compartidas con los interesados (stakeholders) y sometidas a un proceso de selección. Como resultado, se diseñaron ocho mockups representativos de las principales secciones de la aplicación.

Entre las secciones mencionadas se incluyen: la página de inicio (Figura 12), la sección "Acerca de nosotros" (Figura 13), la sección de servicios (Figura 14), la sección de grupos de vida (Figura 15), la sección de eventos (Figura 16), la sección de donaciones (Figura 17), la página de inicio de sesión para el administrador (Figura 18) y la sección de administración (Figura 19).

Figura 12

Mockup Sección Inicio



Figura 13

Mockup Sección Nosotros

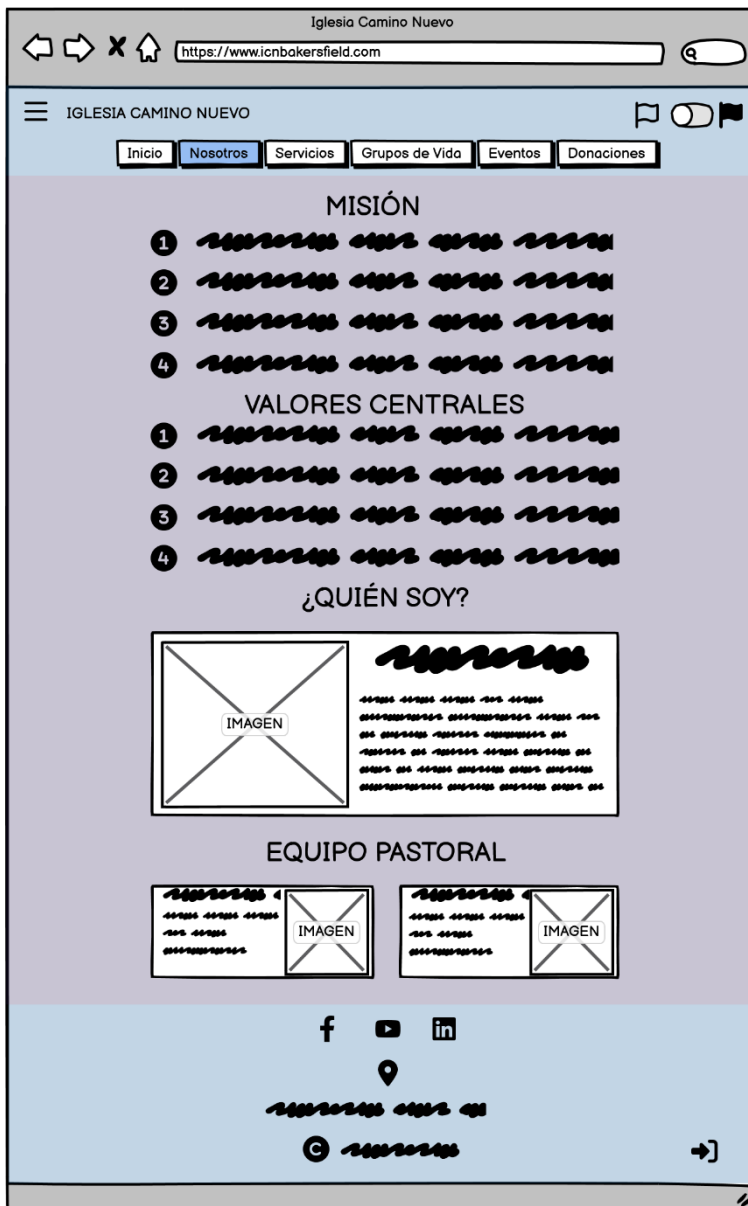


Figura 14

Mockup Sección Servicios

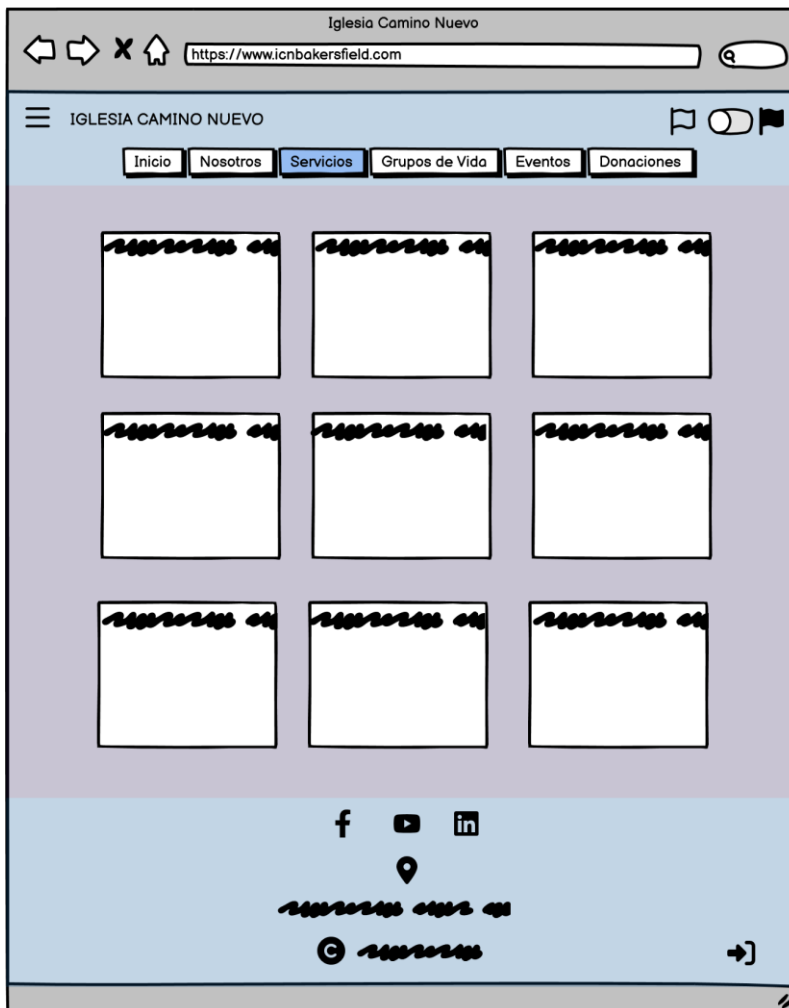


Figura 15

Mockup Sección Grupos de Vida

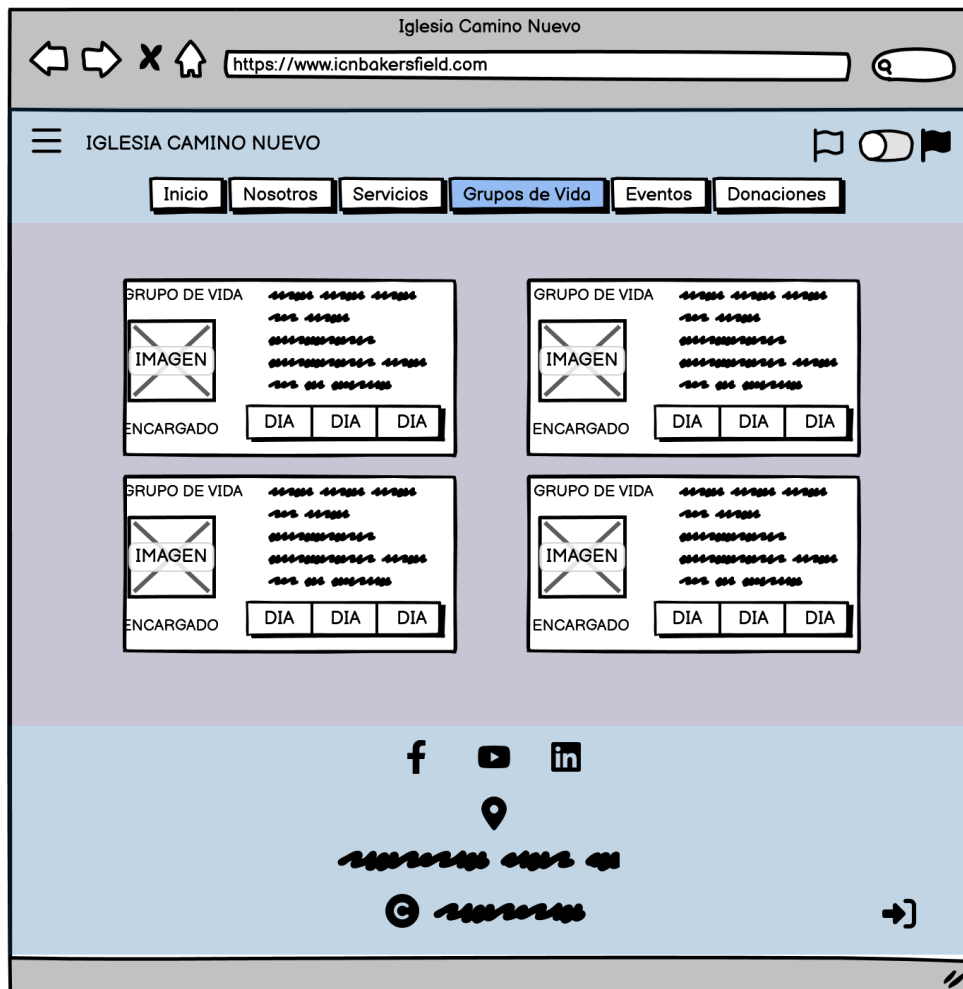


Figura 16

Mockup Sección Eventos



Figura 17

Mockup Sección Donaciones

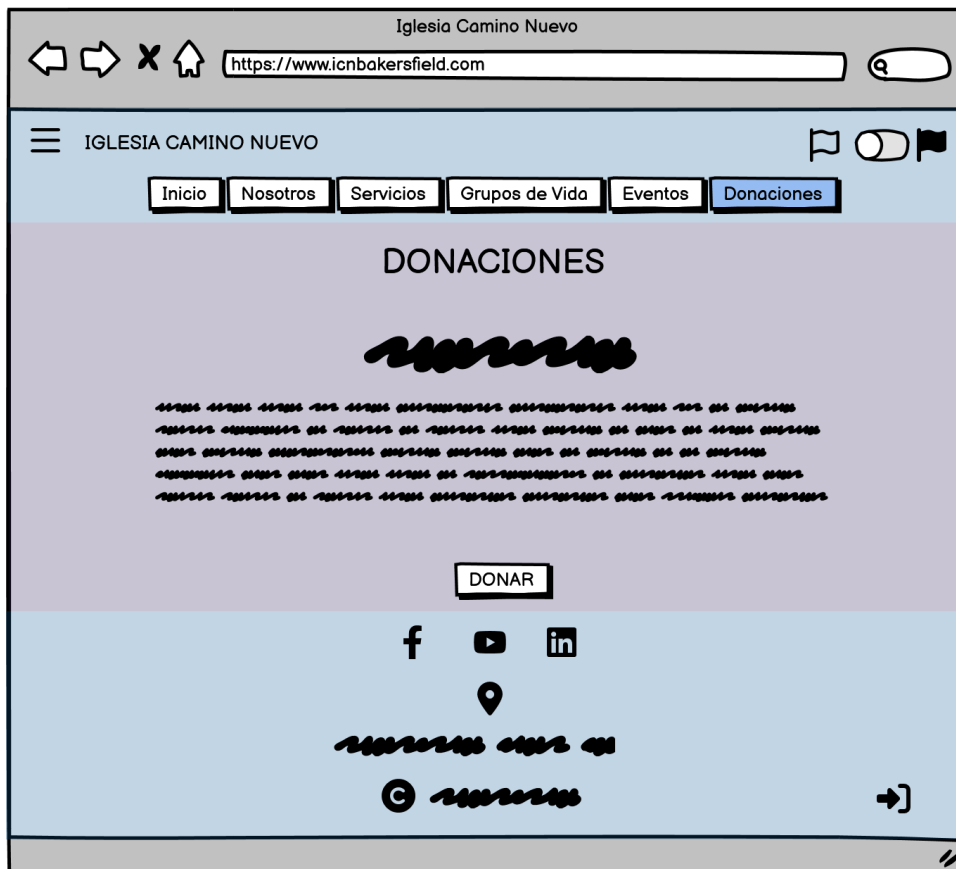


Figura 18

Mockup Sección Inicio de Sesión

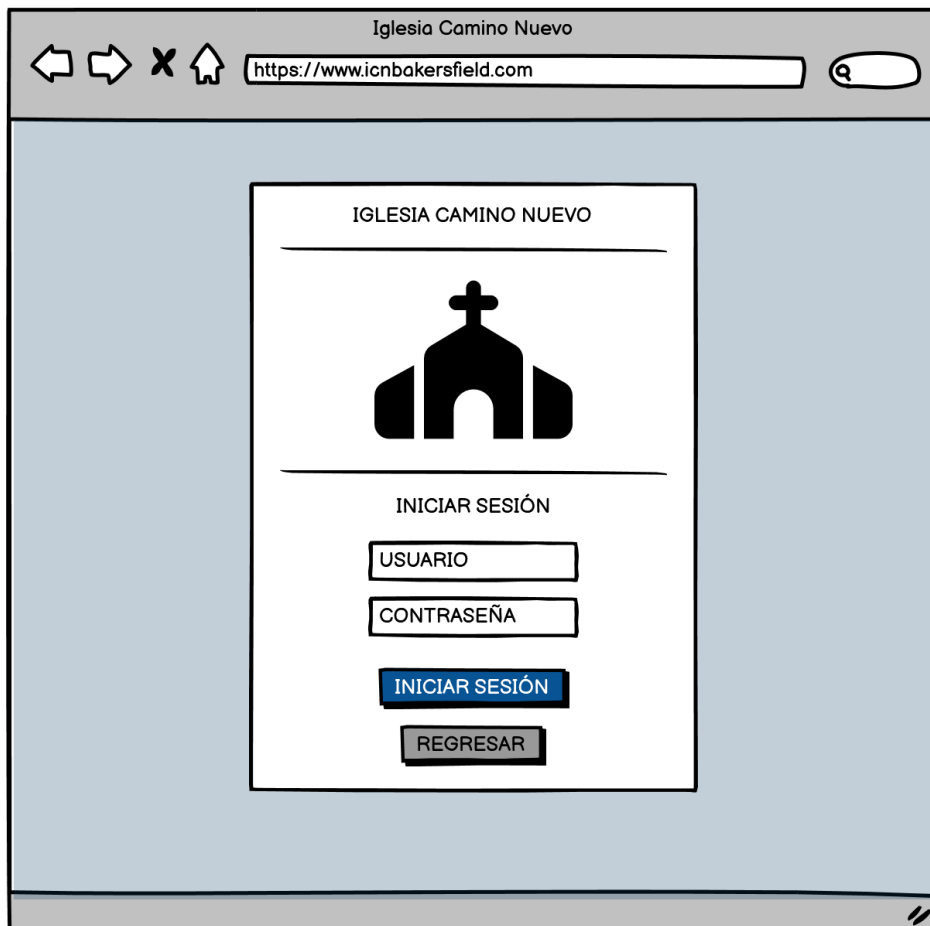
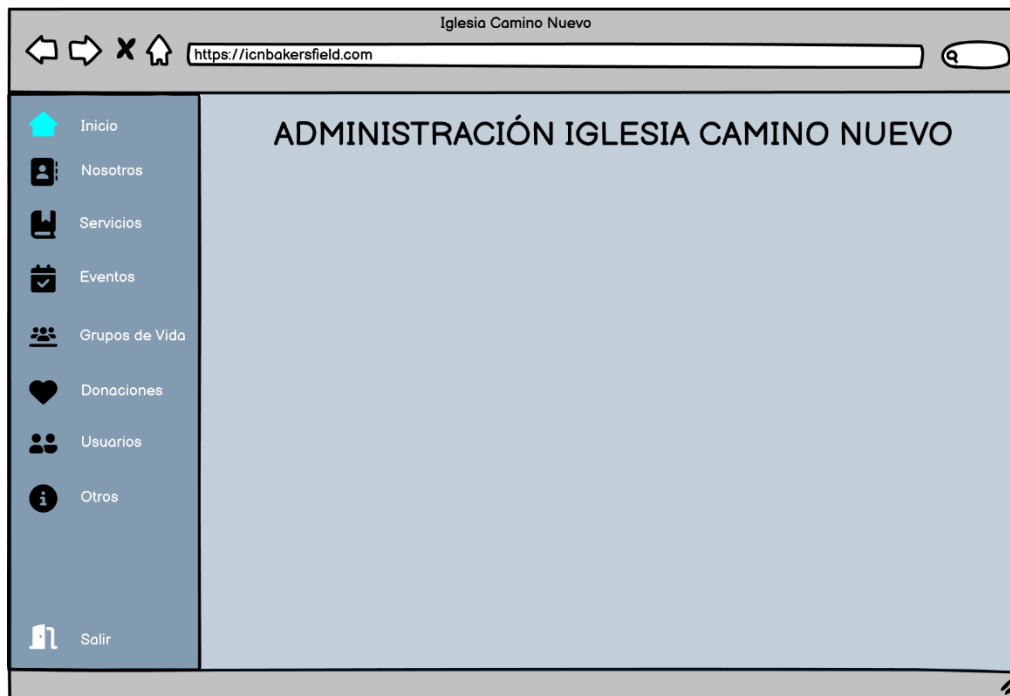
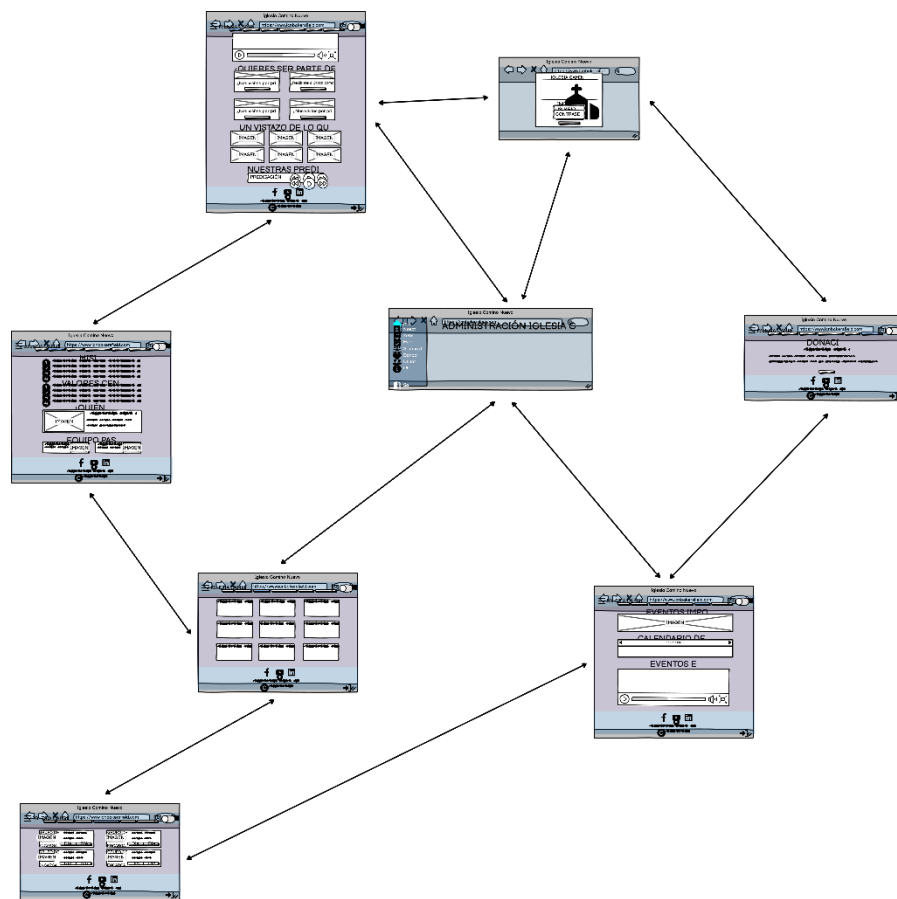


Figura 19*Mockup Sección Administración*

Además de los mockups, se creó un diagrama de navegabilidad que representa un esquema tipo red. A partir de la sección inicial, los usuarios pueden acceder a distintas secciones sin seguir un orden preestablecido (Figura 20).

Figura 20

Diagrama de Navegabilidad de la Aplicación



A continuación, se presentan descripciones detalladas de los diferentes sprints realizados, incluyendo los procesos que se llevaron a cabo para completarlos. Estos sprints se encuentran mencionados en la Tabla 3.

Sprint 1: Gestión De La Información

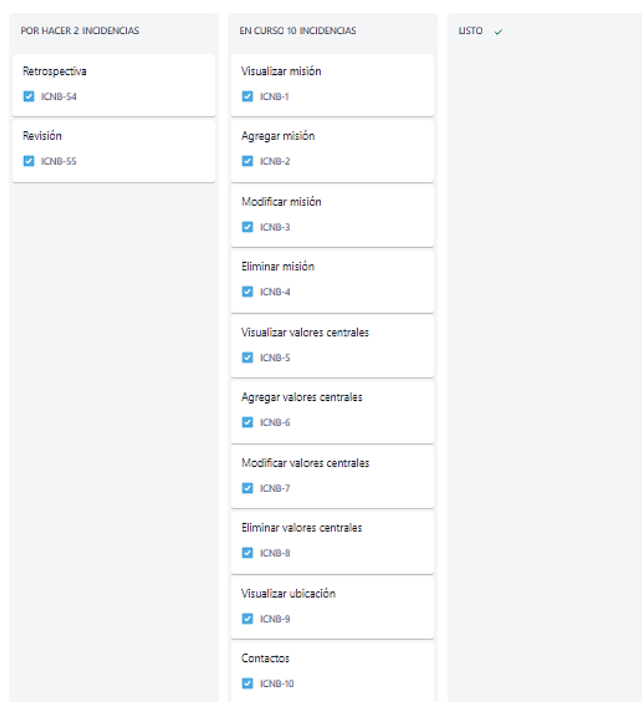
El sprint 1 (Gestión de la Información) inició el 13 de marzo y culminó el 26 de marzo de 2023. Durante el transcurso de este sprint, se llevaron a cabo tareas que incluyen la visualización de la misión, la adición de nuevas misiones, la modificación de misiones existentes, la eliminación de misiones, la visualización de los valores centrales, la incorporación

de nuevos valores centrales, la modificación de valores centrales existentes, la eliminación de valores centrales, la visualización de ubicaciones y la exhibición de contactos.

Estas tareas están debidamente descritas en el Product Backlog y asignadas en el Sprint Backlog, donde han pasado del estado "por hacer" al estado "en curso" como se observa en la Figura 21.

Figura 21

Tablero del sprint Gestión de Información inicial



El adecuado progreso del sprint se fundamenta en un proceso que inicia con un diseño ágil de la base de datos. Posteriormente, se procede a la implementación de la parte del backend con el propósito de establecer la conexión con la base de datos mediante Mongoose y de desarrollar las operaciones CRUD (Create, Read, Update, Delete) destinadas a gestionar la información de la Iglesia Camino Nuevo (como se puede apreciar en la Figura 22, Figura 23, Figura 24 y Figura 25).

El desarrollo del backend se lleva a cabo utilizando las tecnologías Node.js y Express.js, las cuales han sido seleccionadas después de un análisis previo. Para facilitar y optimizar el proceso de desarrollo, se emplea el entorno de desarrollo integrado (IDE) Visual Studio Code, junto con las extensiones Console Ninja y Vue Language Features (Volar) que proporcionan soporte y herramientas adicionales.

Figura 22

Codificación Backend Operación Create

```
/*Add new Core Value*/
Routers.post(`${pathAdmin}`, util.validateToken, (req, res) => {
  try {
    const data = new CoreValue({
      _id: new mongoose.Types.ObjectId(),
      value: {
        spanish: req.body.value.spanish,
        english: req.body.value.english,
      },
    });
    data.save((error, req) => {
      if (error) {
        res.json({
          result: false,
          message: 'Request not possible',
          error,
        });
      } else {
        res.json({
          result: true,
          message: 'Request executed successfully',
          req,
        });
      }
    });
  } catch (error) {
    console.log('Error save core value ' + error);
  }
});
```

Figura 23

Codificación Backend Operación Read

```

/*Get information by language*/
Routers.get(`${pathUser}:language`, (req, res) => {
  const { language } = req.params;
  CoreValue.find()
    .then((data) => {
      const values = data.map((value) => {
        return {
          _id: value._id,
          value: value.value[language],
        };
      });
      res.json(values);
    })
    .catch((error) => res.json({ message: error }));
});

```

Figura 24

Codificación Backend Operación Update

```

/*Update Core Value */
Routers.put(`${pathAdmin}:id`, util.validateToken, (req, res) => {
  try {
    const { id } = req.params;
    const value = {
      spanish: req.body.spanish,
      english: req.body.english,
    };
    CoreValue.updateOne({ _id: id }, { $set: { value } })
      .then((data) => {
        res.json({ result: true });
      })
      .catch((error) => res.json({ result: false }));
  } catch (error) {
    console.log('Error update core value ' + error);
  }
});

```

Figura 25*Codificación Backend Operación Delete*

```
/*Delete Core value */
Routers.delete(`${pathAdmin}:id`, util.validateToken, (req, res) => {
  try {
    const { id } = req.params;
    CoreValue.findOneAndDelete({ _id: id })
      .then((deletedCoreValue) => {
        if (deletedCoreValue) {
          res.status(200).json({
            result: true,
          });
        } else {
          res.status(404).json({ result: false });
        }
      })
      .catch((error) => {
        res.status(500).json({ result: false });
      });
  } catch (error) {
    console.log('Error delete core value ' + error);
  }
});
```

Adicionalmente, se usa la herramienta Postman para realizar pruebas de funcionamiento sobre las API's creadas. Esto permite verificar su correcto desempeño y garantizar su funcionalidad. En cuanto a la supervisión y administración de la base de datos, se recurre a las herramientas DataGrip y MongoDB Compass, que proporcionan una interfaz eficiente para llevar a cabo estas tareas.

Este proceso de desarrollo se replica en varios sprints, los cuales comparten funcionalidades de gestión, con la finalidad de alcanzar los objetivos establecidos en cada iteración del proyecto.

Daily Meetings

Las reuniones diarias se llevaron a cabo utilizando plataformas de comunicación remota, específicamente Google Meet, y contaron con la participación del equipo de desarrollo, quienes asumieron el rol de Scrum Master y Scrum Team (Desarrolladores). Durante estas reuniones, se llevó a cabo la planificación de las actividades diarias, las cuales abarcaron el desarrollo de nuevas características, la continuación de tareas pendientes y la resolución de problemas. Estos elementos se muestran en la Tabla 13, junto con las observaciones pertinentes.

Tabla 13

Resumen de actividades planificadas en las daily meetings del sprint 1

Dia	Actividad Planificada	Observaciones
1	Creación de los proyectos de frontend y backend	Se generaron problemas en la versión de las herramientas
2	Corrección de las versiones y reestructuración de los proyectos	
3	Creación de la base de datos de desarrollo	
4	Creación de la vista principal y ubicación de componentes básicos iniciales	
5	Creación de endpoints y componentes para agregar y modificar misiones de la iglesia	
6	Creación de endpoints y componentes para eliminar y visualizar misiones de la iglesia	
7	Creación de endpoints y componentes para agregar y modificar los valores centrales de la iglesia	
8	Creación de endpoints y componentes para eliminar y visualizar los valores centrales de la iglesia	

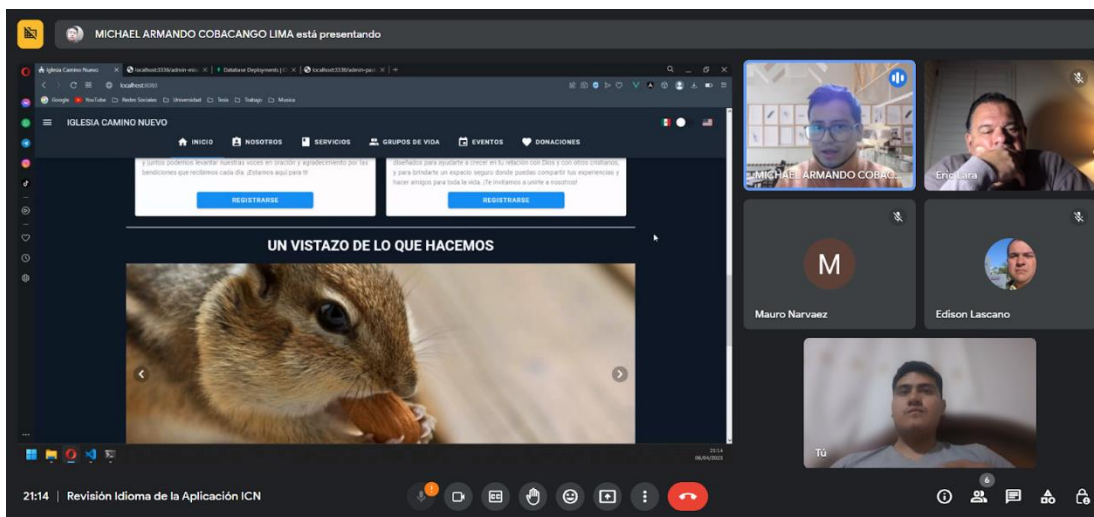
Día	Actividad Planificada	Observaciones
	Creación del pie de página con los datos de ubicación de la iglesia y adición de datos de contacto	
9		
10	Verificación de calidad en el código	
11	Pruebas de módulo	
12	Reunión para revisión de módulo	La reunión se llevó a cabo según lo previsto.
13	Retrospectiva	Se trabajó en mejorar los diseños como mejora continua.
14	Retrospectiva	Se trabajó en mejorar los diseños como mejora continua.

Aceptación del usuario

El día 24 de marzo de 2023 se llevó a cabo una reunión en la que participaron todos los actores involucrados en el proyecto, incluyendo el Development Team (Edison Chinlle, Michael Cobacango), el Scrum Master (Edison Lascano), el Product Owner (Mauro Narváez) y el cliente/usuario (Eric Lara), ver Figura 26. Durante esta reunión, se presentaron las funcionalidades desarrolladas a lo largo del sprint y se procedió a su verificación. El acta de reunión se encuentra adjunta en el APÉNDICE A.3.

Tras la reunión de aceptación, tanto el Cliente como el Product Owner aceptaron formalmente la entrega del avance del software funcional sin requerir correcciones o cambios adicionales.

Figura 26

Reunión de aceptación**Resultados**

En el proyecto de Backend, se desarrolló un total de ocho controladores que contienen 26 endpoints, tal como se muestra en la Tabla 14. Estos endpoints han sido diseñados con el propósito de ofrecer la funcionalidad de gestión de la información en la que se puede modificar o crear datos referentes a la iglesia Camino Nuevo.

Tabla 14

Listado de endpoints generados en el sprint 1

URI	Método	Función
menu/{language}	GET	Obtener el menú con el idioma seleccionado
video-presentation	GET	Obtener el video de presentación
cards-register/{language}	GET	Obtener información de las tarjetas de registro con el idioma seleccionado

URI	Método	Función
admin-cards-register	GET	Obtener información de las tarjetas de registro para el usuario administrador
admin-cards-register	PUT	Actualizar información de las tarjetas de registro para el usuario administrador
missions/{language}	GET	Obtener la misión en el lenguaje seleccionado
admin-missions	GET	Obtener la misión desde el usuario administrador
admin-missions	POST	Enviar información de la misión desde el usuario administrador
admin-missions/{id}	PUT	Actualizar misión desde el usuario administrador
admin-missions/{id}	DELETE	Eliminar la misión desde el usuario administrador
core-value/{language}	GET	Obtener los valores centrales con el idioma seleccionado
admin-core-values	GET	Obtener los valores centrales desde el usuario administrador
admin-core-values	POST	Enviar valores centrales desde el usuario administrador
admin-core-values/{id}	PUT	Actualizar los valores centrales desde el usuario administrador
admin-core-values/{id}	DELETE	Eliminar valores centrales desde el usuario administrador
user-pastoral{language}	GET	Obtener el equipo pastoral por lenguaje
senior-pastor/{language}	GET	Obtener el pastor principal por lenguaje
admin-pastoral-team	GET	Obtener el equipo pastoral desde el usuario administrador
admin-pastoral-team/senior	GET	Obtener el pastor principal desde el usuario administrador
admin-pastoral-team	POST	Agregar nuevo miembro del equipo pastoral

URI	Método	Función
admin-pastoral-team	PUT	Actualizar miembro del equipo pastoral
admin-pastoral-team/senior	PUT	Actualizar pastor principal desde el usuario administrador
admin-pastoral-team/{id}	DELETE	Eliminar miembro del equipo pastoral desde el usuario administrador
footer/{language}	GET	Obtener pie de página en el lenguaje seleccionado
administrator/login	POST	Enviar credenciales de acceso al sistema
administrator	GET	Obtener información del usuario administrador

En el proyecto de Frontend, se generaron un total de cinco componentes y diez vistas, según se muestra en la Tabla 15. Estos elementos fueron diseñados específicamente para satisfacer las necesidades visuales establecidas durante el sprint actual. El desarrollo de los componentes y vistas se llevó a cabo utilizando la tecnología Vue.js y se integraron con las bibliotecas Vuetify, Bootstrap y SweetAlert, con el objetivo de crear una experiencia de usuario atractiva, funcional y fluida.

La incorporación de Vuetify en el proyecto permitió la creación de componentes con una apariencia bien definida y estilizada. Por otro lado, la utilización de Bootstrap facilitó la creación de componentes específicos y garantizó la responsividad y adaptabilidad de la interfaz. Por último, SweetAlert se implementó para proporcionar notificaciones y alertas visuales más atractivas y amigables con el usuario. Gracias a esta biblioteca, fue posible personalizar las notificaciones de error, éxito y otras acciones importantes, mejorando así la interacción general del usuario con la aplicación.

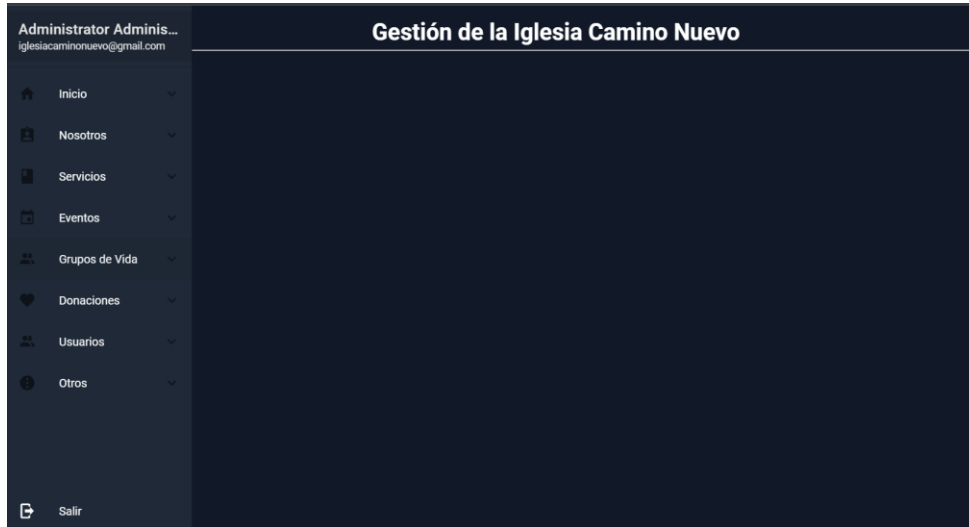
Tabla 15*Componentes y vistas generadas en el sprint 1*

Componentes	Vistas
Toolbar.vue	HomeUserView.vue
SectionRegister.vue	AboutUserView.vue
SectionAboutUs.vue	ApplicationAdminView.vue
Footer.vue	LoginView.vue
MenuBar.vue	VideoPresentationManagement.vue
	RegisterManagement.vue
	MissionsManagement.vue
	CoreValuesManagement.vue
	PastoralTeamManagement.vue
	SeniorPastoralManagement.vue

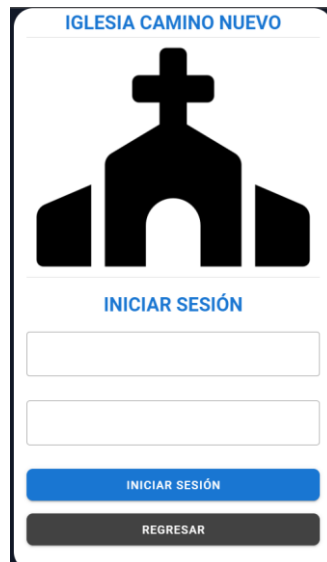
Las vistas abarcan diversas funcionalidades que contribuyen a la experiencia del usuario. Por ejemplo, la vista de la sección de administración (Figura 27) proporciona acceso a la consola de administración y las configuraciones disponibles. La vista de inicio de sesión (Figura 28) permite el ingreso de datos para acceder a la consola de administración. La vista de los valores centrales de la iglesia (Figura 29) muestra los principios fundamentales que guían a la organización.

Figura 27

Pantalla de administración de la Iglesia Camino Nuevo

**Figura 28**

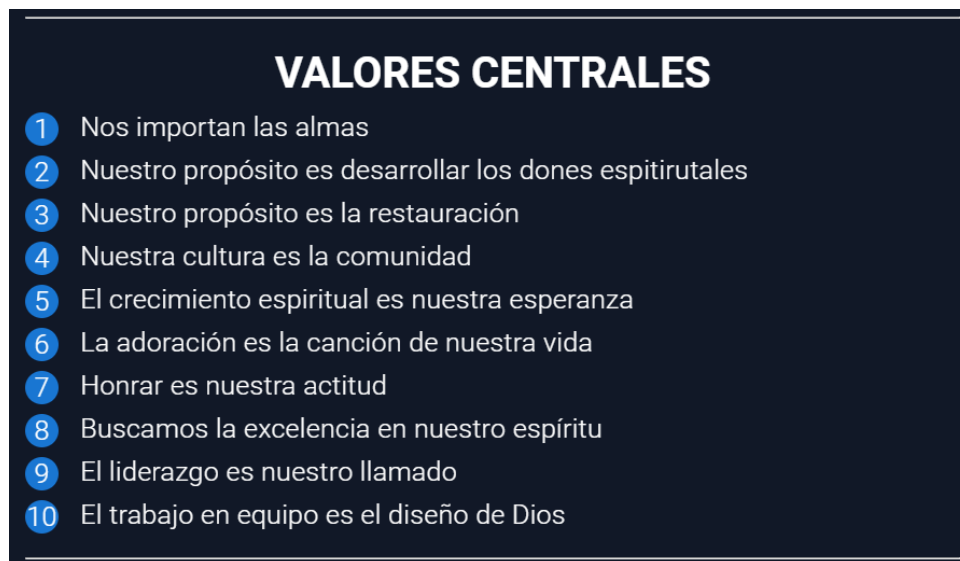
Formulario para inicio de sesión a la consola de administración



The login form is titled 'IGLESIA CAMINO NUEVO' and features a black silhouette of a church with a cross on top. Below the icon, the text 'INICIAR SESIÓN' is displayed. There are two input fields for username and password. At the bottom, there are two buttons: a blue 'INICIAR SESIÓN' button and a dark grey 'REGRESAR' button.

Figura 29

Visualización de los valores centrales de la iglesia



Además, se incluyen vistas para la gestión de los valores centrales (Figura 30), la visualización de la ubicación (Figura 31), los contactos (Figura 32), la información del pastor principal (Figura 33), el equipo pastoral (Figura 34), la misión de la iglesia (Figura 35) y su gestión (Figura 36). Estas vistas permiten la visualización, modificación y adaptación de la información según sea necesario.

Figura 30

Pantalla de gestión de valores centrales

Gestión Valores Centrales		
Inglés	Español	Acciones
We care about souls	Nos importan las almas	 
Our purpose is to develop spiritual gifts	Nuestro propósito es desarrollar los dones espirituales	 
Our purpose is restoration	Nuestro propósito es la restauración	 
Our culture is community	Nuestra cultura es la comunidad	 
Spiritual growth is our hope	El crecimiento espiritual es nuestra esperanza	 
Worship is the song of our life	La adoración es la canción de nuestra vida	 
Honor is our attitude	Honar es nuestra actitud	 
We strive for excellence in our spirit	Buscamos la excelencia en nuestro espíritu	 
Leadership is our calling	El liderazgo es nuestro llamado	 
Teamwork is God's design	El trabajo en equipo es el diseño de Dios	 

Rows per page: 10 1-10 of 10

Figura 31

Visualización de los datos de ubicación



Ubicación

5051 Stine Rd. Bakersfield, California, Estados Unidos 93313

Teléfono: +1 661-735-5667

Copyright @ 2023 - Iglesia Camino Nuevo

Figura 32

Visualización de la información del pastor principal y el equipo pastoral



Eric y Emma Lara

Los pastores Eric y Emma Lara han servido alrededor de 30 años en diferentes ministerios ayudando a jóvenes, matrimonios, niños, familias y levantar líderes en el reino de Dios, también se han preparado en el área de restauración espiritual, graduados con títulos ministeriales, el pastor Eric también es escritor y autor del libro *Salvando a Nuestras Familias*, y director de la escuela de ministerios Bakersfield Bible Institute. Como pastores llevan una trayectoria de alrededor de 12 años y hoy en día pastorean la iglesia Camino Nuevo en la ciudad de Bakersfield California.

EQUIPO PASTORAL

<p>Nombre Pastor Apellido Pastor</p> <p>Pastor</p> <p>Descripcion</p> <div style="display: flex; justify-content: space-between; align-items: center;"> pastor@gmail.com +593 963258741 </div>	<div style="display: flex; justify-content: space-between; align-items: center;"> <div> <p>Miguel Gomez</p> <p>lider</p> <p>pastor de matrimonios</p> <div style="display: flex; justify-content: space-between; align-items: center;"> elara@nlc.life +593 963258741 </div> </div>  </div>
--	--

Figura 33

Pantalla de gestión de información del pastor principal

Actualizar Información Pastor Principal

Eric y Emma Lara

Pastores Principales

Los pastores Eric y Emma Lara han servido alrededor de 30 años en diferentes ministerios ayudando a jóvenes, matrimonios, niños, familias y levantar líderes en el reino de Dios, también se han preparado en el área de restauración espiritual, graduados con títulos ministeriales, el pastor Eric también es escritor y autor del libro *Salvando a Nuestras Familias*, y director de la escuela de ministerios Bakersfield Bible Institute. Como pastores llevan una trayectoria de alrededor de 12 años y hoy en día pastorean la iglesia Camino Nuevo en la ciudad de Bakersfield California.

Eric y Emma Lara

Main Pastors

Pastors Eric and Emma Lara have served for more than 30 years in different ministries helping youth, marriages, children, families and raising up leaders, they have also been prepared in the area of spiritual restoration, graduated with ministerial degrees, Pastor Eric He is also a writer of the book *Saving Our Families*, and director of the Bakersfield Bible Institute school of ministries. As pastors they have a trajectory of around 12 years and today they pastor the Camino Nuevo church in the city of Bakersfield California.

eric@nlc.life
 +1 818 6401581

Figura 34

Pantalla de gestión de información del equipo pastoral

**Figura 35**

Visualización de las misiones de la iglesia

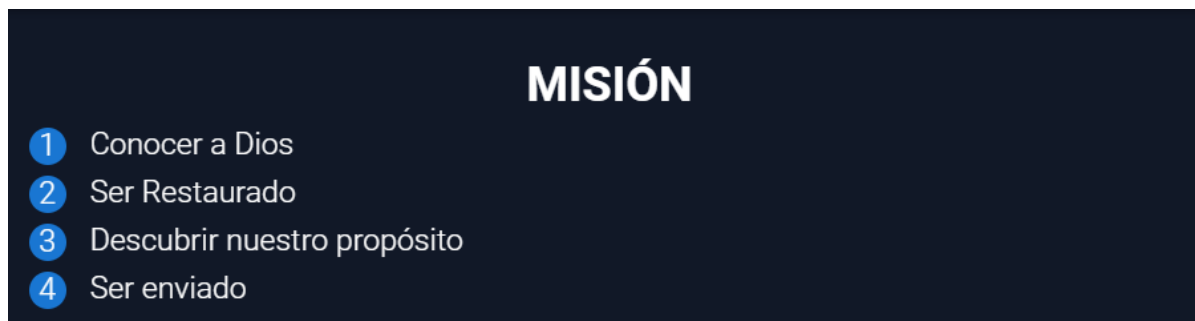


Figura 36

Pantalla de gestión de las misiones de la iglesia

Gestión Misiones			AGREGAR
Inglés	Español	Acciones	
Know God	Conocer a Dios		
To be Restored	Ser Restaurado		
Discover our purpose	Descubrir nuestro propósito		
Be send	Ser enviado		

Rows per page: 10 1-4 of 4

Una vez que se ha aprobado el incremento, se procede a trasladar las incidencias registradas en Jira de la columna "En curso" a la columna "Finalizado", ver Figura 37.

Figura 37

Estado del tablero de Jira al finalizar el sprint 1

POR HACER	EN CURSO	LISTO 12 INCIDENCIAS ✓	+
		Visualizar misión <input checked="" type="checkbox"/> IENB-1 ✓	
		Agregar misión <input checked="" type="checkbox"/> IENB-2 ✓	
		Modificar misión <input checked="" type="checkbox"/> IENB-3 ✓	
		Eliminar misión <input checked="" type="checkbox"/> IENB-4 ✓	
		Visualizar valores centrales <input checked="" type="checkbox"/> IENB-5 ✓	
		Agregar valores centrales <input checked="" type="checkbox"/> IENB-6 ✓	
		Modificar valores centrales <input checked="" type="checkbox"/> IENB-7 ✓	
		Eliminar valores centrales <input checked="" type="checkbox"/> IENB-8 ✓	

Sprint Review

Se llevó a cabo la revisión del sprint con la presencia del Product Owner, Edison Lascano y del Cliente, Eric Lara. Durante esta revisión, se evaluó el funcionamiento adecuado de las funcionalidades de gestión de información de la Iglesia Camino Nuevo. Como resultado, se constató que la ejecución del sprint fue satisfactoria, cumpliendo con las actividades planificadas en el sprint backlog dentro del tiempo establecido.

Se destaca, además, que no se recibieron sugerencias ni comentarios respecto al desarrollo del sprint por parte de los stakeholders, lo cual llevó a la decisión de concluir el sprint actual y dar inicio al siguiente, el cual se enfoca en el cambio de idiomas de inglés a español y viceversa.

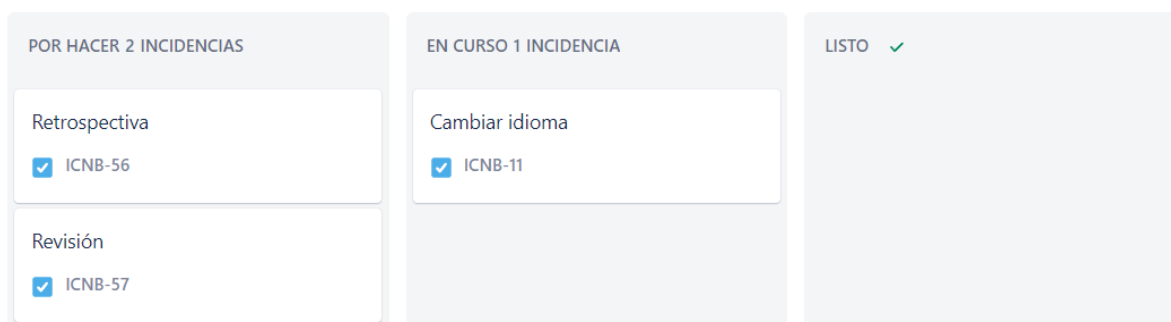
Sprint 2: Cambio del Idioma del Sitio

El sprint 2 (Cambio del Idioma del Sitio) inició el 27 de marzo y finalizó el 09 de abril de 2023. Durante el desarrollo de este sprint, se llevó a cabo la tarea de cambiar el idioma de la aplicación.

Para ello, se realizó el traslado de las incidencias correspondientes al sprint 2 de la columna "Por hacer" a la columna "En curso" en el tablero Jira, ver Figura 38.

Figura 38

Tablero del sprint Cambio de Idioma inicial



El sprint comienza con el diseño de la base de datos, y los documentos gestionados en MongoDB Atlas, destinados al manejo de idiomas. Posteriormente, se crean los objetos requeridos y se refactorizan los objetos ya existentes para el módulo previo. Con este fin, se generan nuevos objetos por cada colección, cada uno de los cuales contiene ahora la información necesaria en inglés y español. Por ejemplo, en la colección "missions", se almacena un documento por cada misión existente, en el cual se especifica su respectivo identificador (ID) y la información de la misión en ambos idiomas, ver Figura 39.

Figura 39

Ejemplo de manejo de idiomas de la aplicación en la base de datos

```
  _id: ObjectId('63f96a20c00cc4e7d6248896')
  ▼ mission: Object
    spanish: "Ser Restaurado"
    english: "To be Restored"
```

Cuando un usuario accede a la aplicación, se le brinda la opción de seleccionar el idioma de preferencia, inglés o español. Una vez que el usuario elige su idioma preferido, la aplicación utiliza esta selección para mostrar la información en el idioma correspondiente.

A través del backend, se implementa una función para el cambio de idioma, la cual recupera los datos del objeto específico mediante el parámetro enviado desde el frontend, ver Figura 40.

Vale destacar que el contenido debe ser ingresado por el administrador del sitio en ambos idiomas, por lo que no se usa un motor de traducción automática. De esta manera, se logra una solución eficiente y flexible para proporcionar una experiencia multilingüe a los usuarios de la aplicación, evitando problemas de precisión de traducción, contexto incorrecto y errores de formato que podrían surgir con motores de traducción automática.

Figura 40

Codificación de función para obtener información según idioma seleccionado

```
/*Get information by language*/
Routers.get(`${pathUser}:language`, (req, res) => {
  const { language } = req.params;
  Mission.find()
    .then((data) => {
      const values = data.map((value) => {
        return {
          _id: value._id,
          mission: value.mission[language],
        };
      });
      res.json(values);
    })
    .catch((error) => res.json({ message: error }));
});
```

Este proceso se mantiene en paralelo con el desarrollo de los siguientes sprints. Se inicia con el proceso del diseño ágil de la base de datos, junto con la creación de la información en español e inglés por defecto. Esto se lleva a cabo con el objetivo de asegurar un adecuado manejo de la información compartida en la aplicación.

Daily Meetings

Las reuniones diarias se llevaron a cabo mediante la utilización de la plataforma Google Meet, en las cuales participaron los miembros del equipo de desarrollo, quienes asumieron el rol adicional de Scrum Master además de sus propias responsabilidades. Durante las reuniones, se realizó la planificación de las actividades diarias, que incluyeron el desarrollo de nuevas características, la continuación de tareas pendientes y la resolución de problemas. Estos aspectos se encuentran detallados en la Tabla 16.

Tabla 16

Resumen de actividades planificadas en las daily meetings del sprint 2

Día	Actividad Planificada	Observaciones
1	Conversación con respecto a la forma en la que se manejara la conversión de idioma	Se analizan varias estrategias, pendiente búsqueda de tecnologías
2	Búsqueda de técnicas o tecnologías para la traducción de contenido confiable	
3	Selección de la técnica estudiada, traducción por llenado de información en los idiomas requeridos mediante JSON	
4	Implementación de la técnica con los títulos de la barra de inicio de la aplicación	
5	Pruebas de funcionamiento de la implementación de la técnica	
6	Implementación de la técnica con toda la información generada en el módulo anterior	
7	Solución de errores encontrados	
8	Modificación de la base de datos para optimización de información	
9	implementación de seguridad en el inicio de sesión a la consola de administración	
10	Verificación de calidad en el código	
11	Pruebas de funcionalidad	
12	Reunión para revisión de módulo	La reunión se llevó a cabo según lo previsto
13	Retrospectiva	Se trabajó en mejorar los diseños como mejora continua
14	Retrospectiva	Se trabajó en mejorar los diseños como mejora continua

Aceptación del usuario

El día 07 de abril de 2023 se efectuó una reunión en la que participaron todos los involucrados en el proyecto. Durante esta reunión, se presentó la funcionalidad desarrollada a lo largo del sprint y se procedió a su verificación. El acta de reunión correspondiente se encuentra en el APÉNDICE A.4.

Después de la reunión de aceptación, tanto el cliente como el Product Owner manifestaron formalmente su aceptación de la entrega del avance del software funcional, sin necesidad de solicitar correcciones o modificaciones adicionales.

Resultados

Como resultado de este sprint, no se ha producido ninguna generación de archivos en los proyectos de la aplicación. No obstante, se ha realizado la configuración en el proyecto frontend para habilitar el cambio de idioma en toda la aplicación a través de un parámetro, teniendo en cuenta que la implementación del cambio de idioma se ha llevado a cabo de forma simultánea en todo el proyecto. Además, se han abordado aspectos relacionados con validaciones y medidas de seguridad.

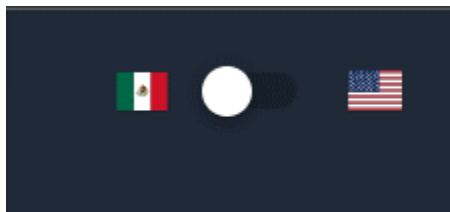
Aunque no se han generado nuevos archivos, se han efectuado modificaciones en los componentes del frontend creados hasta la fecha de culminación del sprint 2, además de la modificación a la base de datos, donde se trata la información en los dos idiomas y se llama según el parámetro indicado.

Dentro de las vistas que han experimentado modificaciones se incluyen la adición de un selector de idiomas en la sección de encabezado, como se aprecia en la Figura 41, la barra de títulos, como se presenta en la Figura 42, la sección de inicio, la cual se muestra en la Figura 43, y la sección de nosotros, como se puede observar en la Figura 44. Estas secciones han

sido actualizadas con sus respectivos componentes. Es importante destacar que el cliente y el Product Owner han solicitado que la sección de administración se desarrolle en idioma español.

Figura 41

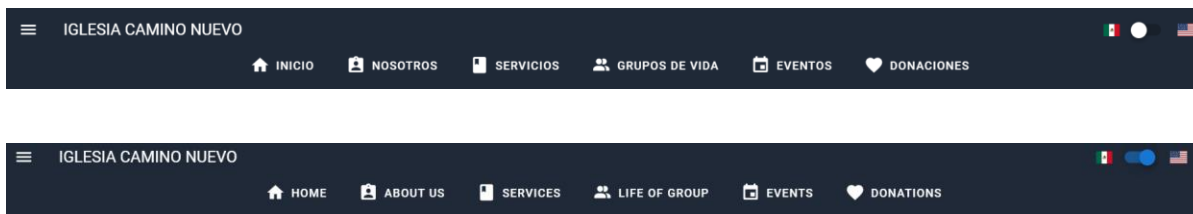
Selector de idiomas



Nota. En la figura se observa el selector de idioma diseñado, el cual representa cada idioma mediante la utilización de las respectivas banderas de los países en los cuales el español e inglés son predominantemente hablados y de donde son la mayoría de los feligreses de la Iglesia.

Figura 42

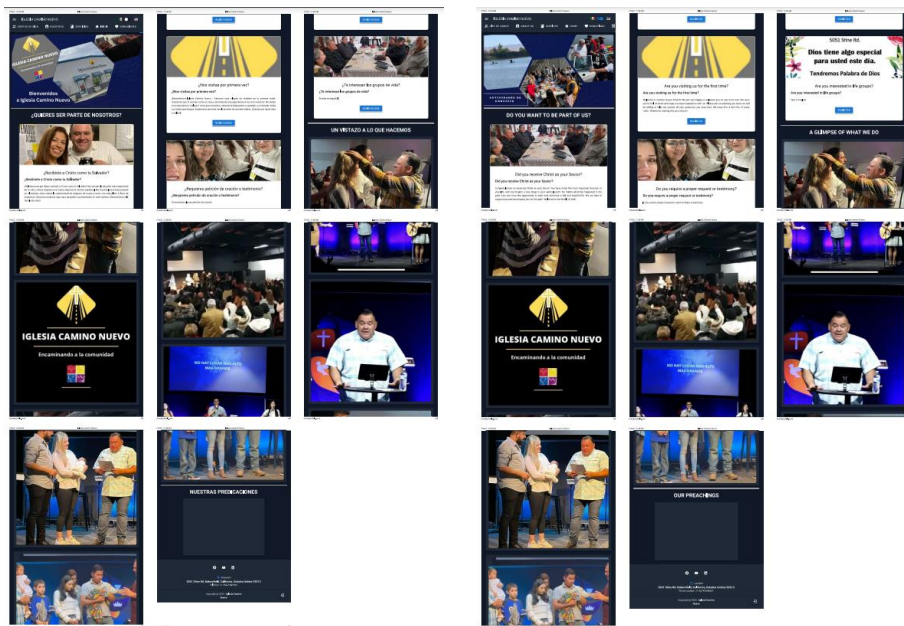
Barra de títulos con cambio de idioma



Nota. En la figura situada en la parte superior, se presenta la barra de títulos con la selección del idioma español. Por otro lado, en la imagen inferior se muestra el cambio de idioma efectuado utilizando el selector, y se pueden observar los títulos en inglés.

Figura 43

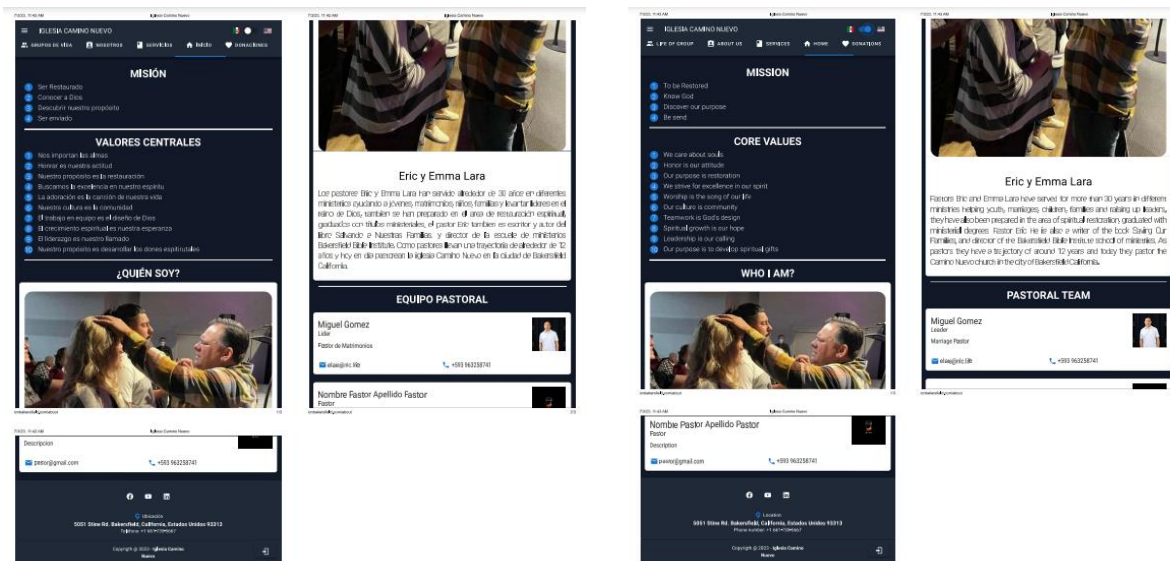
Sección de "Inicio" con cambio de idioma



Nota. En la figura situada en la parte izquierda, se presenta la sección de inicio con la selección del idioma español. Por otro lado, en la imagen ubicada en la parte derecha se muestra el cambio de idioma realizado utilizando el selector, y se puede apreciar la visualización de los contenidos en inglés.

Figura 44

Sección de "Nosotros" con cambio de idioma

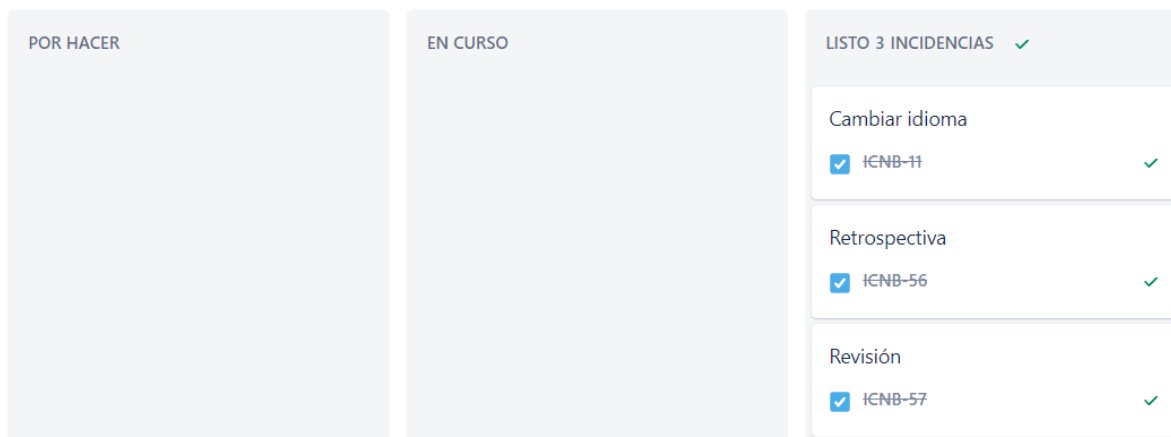


Nota. En la figura situada en la parte izquierda se exhibe la sección denominada "Nosotros" con la selección del idioma español. Por otro lado, en la imagen ubicada en la parte derecha se puede observar el cambio de idioma llevado a cabo a través del selector, lo cual conduce a la visualización de los contenidos en inglés.

Como consecuencia de la aceptación sin contratiempos del incremento, las incidencias que estaban en la columna "En curso" son trasladadas a la columna "Finalizado", tal como se muestra en la Figura 45.

Figura 45

Estado del tablero de Jira al finalizar el sprint 2



Sprint Review

Durante la revisión del sprint, se llevó a cabo una sesión con el Product Owner en la cual se realizó un recorrido por las funcionalidades implementadas, con el objetivo de revisar su funcionamiento y estado final. La ejecución del sprint se llevó a cabo de manera satisfactoria, habiéndose completado el desarrollo de todas las historias de usuario en el plazo establecido.

En este sprint, se ha comenzado a abordar los asuntos pertinentes a la migración del aplicativo a la nube, con el propósito de realizar una evaluación periódica del trabajo realizado para facilitar una retroalimentación constante. Además, se ha iniciado el análisis del tema del dominio del sitio web, con el objetivo de lograr un sólido posicionamiento entre las iglesias cristianas, siguiendo los principios fundamentales del marketing digital. Algunos de los posibles dominios considerados son los siguientes: icnbakersfield.com, icnbakersfield.org, iglesiacaminonuevo.org, icnbakersfield.church, iglesiacaminonuevo.church.

Tras revisar detenidamente los próximos elementos a desarrollar en el tercer sprint, se procedió a dar por finalizado el sprint actual para dar paso al siguiente, centrado en la "Gestión de Contenido Multimedia".

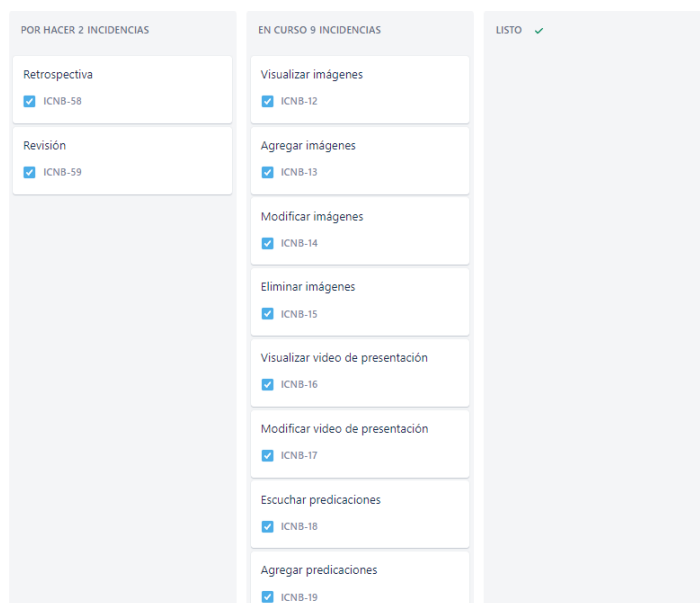
Sprint 3: Gestión del Contenido Multimedia

El sprint 3 (Gestión del Contenido Multimedia) inició el 10 de abril y finalizó el 23 de abril de 2023. A lo largo de este sprint, se han realizado diversas actividades que abarcan la visualización, adición, modificación y eliminación de imágenes, así como la modificación del video de presentación de la iglesia y las opciones para escuchar, agregar y eliminar predicaciones.

Con el propósito de dar inicio al nuevo sprint, se procedió a trasladar las incidencias correspondientes al sprint 3 desde la columna "Por hacer" hacia la columna "En curso" en el tablero de Jira, tal como se muestra en la Figura 46.

Figura 46

Tablero del sprint Gestión de Contenido Multimedia inicial



El sprint de gestión de contenido multimedia se aborda de manera similar a los sprints de gestión, aunque emplea enfoques y técnicas específicas para la administración de contenido multimedia. El proceso se inicia con la instalación de los paquetes requeridos de Node.js en el backend, como sharp para el procesamiento de imágenes y fs para el registro y eliminación del

contenido multimedia, como se puede apreciar en la Figura 47, es importante resaltar que mediante el sistema de archivos (FS), se facilita la capacidad de agregar y eliminar contenido multimedia dentro del almacenamiento interno del servidor. Posteriormente, se desarrollan métodos especializados encargados de gestionar el contenido multimedia en sus distintos formatos (imagen, video y audio), dado que dicho contenido se encuentra presente a lo largo del desarrollo de la aplicación, ver Figura 48.

Figura 47

Importación de los módulos necesarios para el manejo de contenido multimedia

```
const sharp = require('sharp');
const fs = require('fs');
const ffmpeg = require('fluent-ffmpeg');
require('dotenv').config();
const jwt = require('jsonwebtoken');
```

Figura 48

Métodos para guardar y eliminar contenido multimedia

```

saveImages: async function (image, name, width, height) {
  try {
    const processedImage = sharp(image.buffer);
    const resizedImage = processedImage.resize(width, height, {
      fit: 'cover',
      background: '#FFF',
    });
    const resizedImageBuffer = await resizedImage.toBuffer();
    const imageName = this.getDate().toString() + '_' + name.replace(/\s+/g, '') + '.png';
    fs.writeFileSync(`uploads/${imageName}`, resizedImageBuffer);
    return imageName;
  } catch (error) {
    console.log('Error save image ' + error);
  }
},

deleteImages: function (nameImage) {
  try {
    const pathImage = `uploads/${nameImage}`;
    fs.unlinkSync(pathImage);
    return true;
  } catch (error) {
    console.log('Error delete image ' + error);
    return false;
  }
},

```

Nota. La imagen muestra los métodos destinados a agregar y eliminar imágenes, similares al proceso efectuado con audio y video. Además, se evidencia la utilización del sistema de archivos (FS) para manipular el almacenamiento interno del servidor.

La verificación del correcto funcionamiento de los métodos desarrollados para el manejo del contenido multimedia se realizó sin emplear herramientas automatizadas. En su lugar, se realizaron pruebas de carga de contenido con el objetivo de verificar que los diversos tipos de archivos, tales como imágenes, videos y audio, se subieran al servidor sin errores y que fueran almacenados en la ubicación apropiada. Además, se efectuó una evaluación de los formatos compatibles con la aplicación.

Para asegurar el correcto enlace entre los archivos almacenados en el servidor y sus respectivas rutas en la base de datos, se empleó JetBrains DataGrip, una herramienta especializada en la gestión de bases de datos.

Por último, para garantizar la adecuada visualización del contenido multimedia, se verificó en la aplicación que los archivos se cargaran correctamente en los espacios designados para ello y se muestren sin novedades.

Daily Meetings

Las reuniones diarias se llevaron a cabo mediante la utilización de la plataforma Google Meet, en las cuales participaron los miembros del equipo de desarrollo, quienes asumieron el rol adicional de Scrum Master además de sus propias responsabilidades. Durante las reuniones, se realizó la planificación de las actividades diarias, que incluyeron el desarrollo de nuevas características, la continuación de tareas pendientes y la resolución de problemas. Estos aspectos se encuentran detallados en la Tabla 17, incluyendo las observaciones.

Tabla 17

Resumen de actividades planificadas en las daily meetings del sprint 3

Día	Actividad Planificada	Observaciones
1	Reunión de inicio del sprint, planificación y asignación de tareas	
2	Configurar la funcionalidad de visualización de imágenes	
3	Implementar la funcionalidad de agregar imágenes	
4	Desarrollar la capacidad de modificar imágenes	
5	Agregar la opción de eliminar imágenes	

Día	Actividad Planificada	Observaciones
6	Realizar mejoras en el video de presentación de la iglesia	
7	Implementar la funcionalidad de escuchar canciones	
8	Desarrollar la capacidad de agregar canciones	
9	Añadir la opción de eliminar canciones	
10	Realizar pruebas y correcciones necesarias	Errores o comportamientos inesperados durante las pruebas y correcciones
11	Realizar pruebas y correcciones necesarias	
12	Reunión para revisión de módulo	
13	Retrospectiva	
14	Retrospectiva	

Aceptación del usuario

El día 21 de abril de 2023 se desarrolló una reunión con la participación de todos los involucrados en el proyecto. Durante la reunión, se presentó la funcionalidad desarrollada a lo largo del sprint y se procedió a su verificación. El acta de reunión correspondiente se encuentra adjunta en el APÉNDICE A.5.

Tras la reunión de aceptación, tanto el cliente como el Product Owner expresaron que se tomen en cuenta algunas sugerencias, ver Tabla 18.

Tabla 18

Sugerencias para el sprint 3

N°	Sugerencia	Corrección
1	Tener en cuenta el asunto de los derechos de autor, para ello se piensa mejor no tomar canciones y en su defecto cambiarlas por predicaciones propias de la iglesia.	Se tomó la decisión de no incluir en la sección de predicaciones cualquier contenido protegido por derechos de autor, optando en su lugar por utilizar material propio de la iglesia en caso de ser necesario.
2	Cambiar el tiempo de expiración del Token a más de media hora hasta aprender a utilizar la administración.	Se ha decidido extender el período de expiración del token de sesión a 4 horas, con el propósito de permitir un tiempo suficiente hasta que se adquiriera el conocimiento y dominio adecuado de la sección de administración.
3	Subir la nueva versión de la aplicación al cloud.	Se subió la primera versión de la aplicación a la nube.

Después de haber aplicado las correcciones sugeridas, las partes interesadas manifestaron formalmente su aprobación de la entrega del avance del software funcional, sin requerir correcciones o modificaciones adicionales.

Resultados

Durante el transcurso de este sprint, se llevó a cabo el desarrollo de dos controladores en el marco del proyecto de Backend, los cuales albergan un total de seis endpoints, tal como se detalla en la Tabla 19. Estos endpoints fueron implementados con el objetivo de garantizar el correcto funcionamiento del módulo.

Tabla 19*Listado de endpoints generados en el sprint 3*

URI	Método	Función
images-home	GET	Obtener las imágenes de inicio
admin-images-home	POST	Guardar nuevas imágenes para mostrar en el inicio
admin-images-home/{id}	DELETE	Eliminar imágenes de inicio
songs	GET	Mostrar las canciones
admin-songs	POST	Guardar las nuevas canciones
admin-songs/{id}	DELETE	Eliminar las canciones

Dentro del contexto del proyecto de Frontend, se ha implementado un componente denominado "SectionMultimedia.vue" en la sección de inicio, así como dos vistas adicionales destinadas al administrador, "ImagesHomeManagement.vue" y "SongManagement.vue". El propósito del componente mencionado radica en facilitar la visualización de imágenes y predicaciones en la sección de inicio, como se ilustra en la Figura 49. Por otro lado, las vistas desarrolladas engloban funcionalidades de gestión de imágenes y predicaciones, como se observa en la Figura 50 y Figura 51 respectivamente. En el caso de la gestión de imágenes, se posibilita la creación y eliminación de imágenes a ser mostradas, mientras que en la gestión de predicaciones se facilita la creación y eliminación de predicaciones en el sitio.

Figura 49

Visualización de imágenes y predicaciones

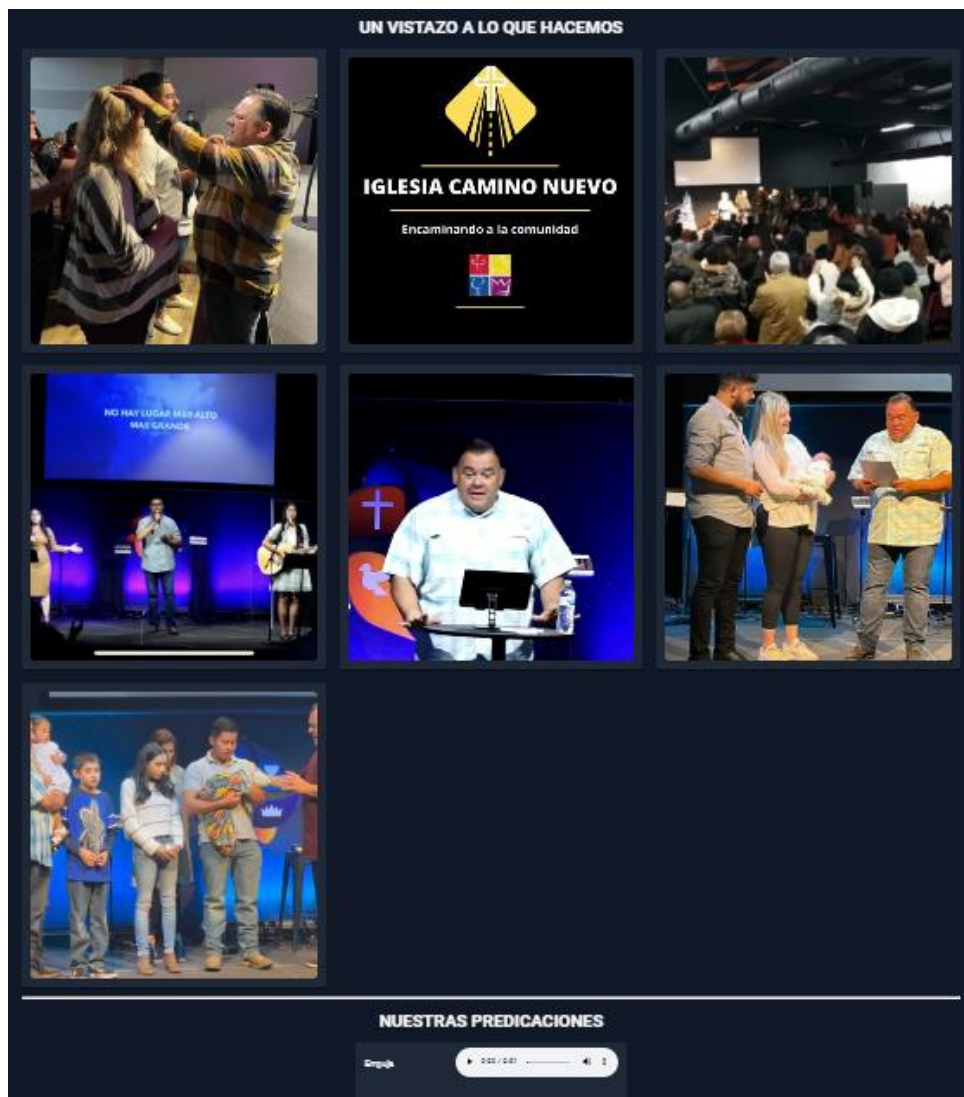


Figura 50

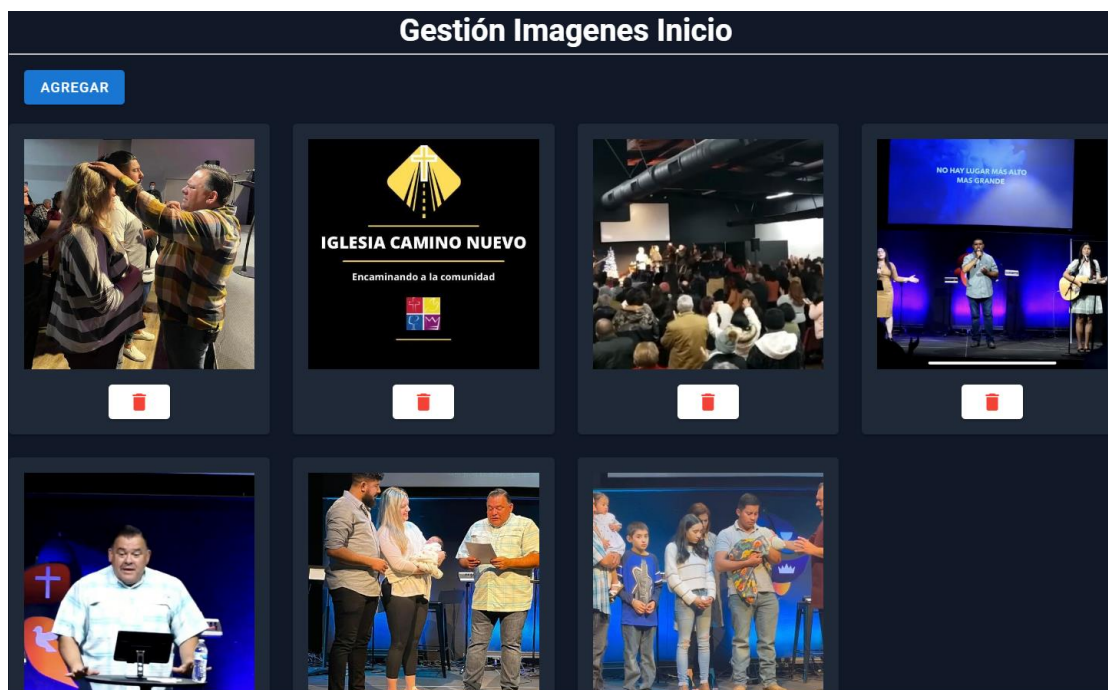
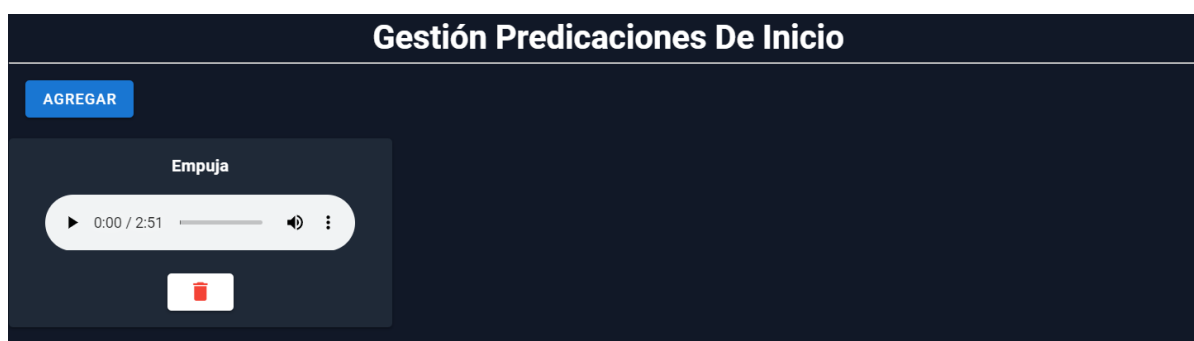
Gestión de imágenes

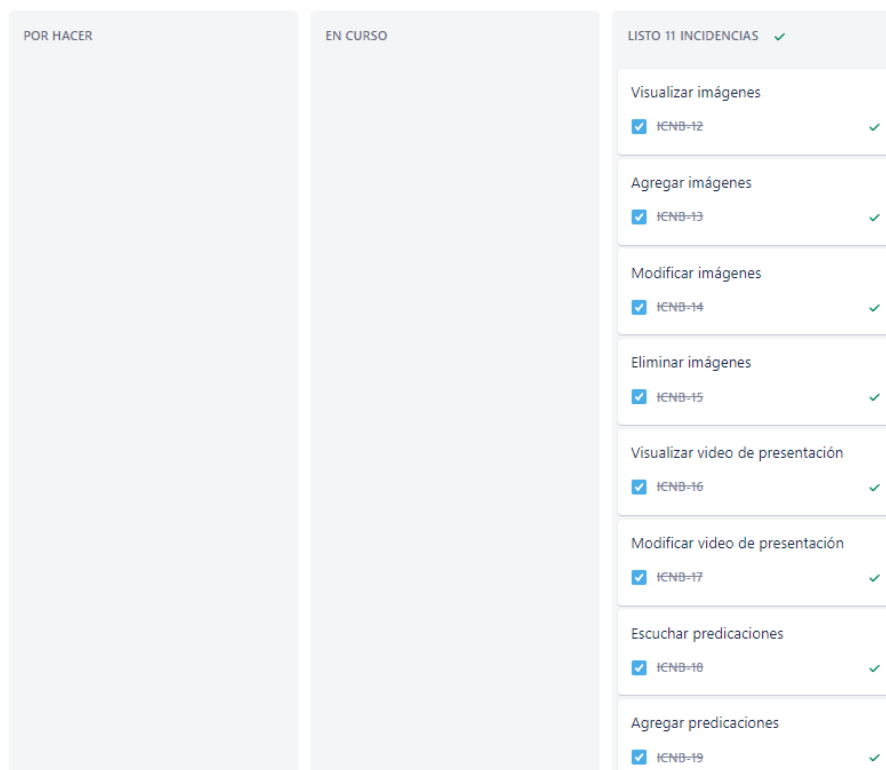
Figura 51

Gestión de predicaciones

Una vez que el incremento ha sido aprobado, se realiza la transferencia de las incidencias registradas en Jira desde la columna "En curso" hacia la columna "Finalizado", como se muestra en la Figura 52.

Figura 52

Estado del tablero de Jira al finalizar el sprint 3



Sprint Review

Se realizó una sesión de revisión del sprint en presencia del Product Owner, Edison Lascano. Durante esta reunión, se evaluó el correcto funcionamiento de las funcionalidades implementadas. Como resultado, se pudo constatar que la ejecución del sprint fue exitosa, logrando cumplir con todas las actividades planificadas en el sprint backlog dentro del plazo establecido.

Es importante destacar que se abordaron y resolvieron todas las sugerencias y comentarios planteados por los stakeholders con relación al desarrollo del sprint. Estas acciones permitieron tomar la decisión de concluir el sprint actual y dar inicio al siguiente, el cual se centrará en la gestión de redes sociales.

Sprint 4: Gestión de Redes Sociales

El sprint 4 (Gestión de Redes Sociales) inicio el 24 de abril y finalizó el 07 de mayo de 2023. En el transcurso de este sprint, se han ejecutado diversas actividades relacionadas con la visualización, adición, modificación y eliminación de redes sociales asociadas a la iglesia.

Estas tareas han sido asignadas en el tablero de Jira, donde se han movido del estado "Por hacer" al estado "En curso", tal como se puede apreciar en la Figura 53.

Figura 53

Tablero del sprint Gestión de Redes Sociales inicial



En este sprint, enfocado en la gestión de redes sociales, se puede observar una similitud con el proceso descrito en el sprint 1. En ambos casos, se realiza el diseño de la base de datos, se establece la conexión con la misma y se desarrollan las operaciones CRUD destinadas a gestionar las redes sociales.

Asimismo, para el desarrollo del backend se emplean las mismas tecnologías mencionadas previamente en el sprint 1, proporcionando soporte y herramientas adicionales.

Además, se utiliza la herramienta Postman para realizar pruebas de funcionamiento sobre las API creadas, lo que permite verificar su correcto desempeño y garantizar su funcionalidad. En cuanto a la supervisión y administración de la base de datos, se hacen uso de las herramientas DataGrip y MongoDB Compass, las cuales ofrecen una interfaz eficiente para llevar a cabo estas tareas.

Es relevante destacar que este proceso de desarrollo se replica en varios sprints, los cuales comparten funcionalidades de gestión con el propósito de alcanzar los objetivos establecidos en cada iteración del proyecto.

Daily Meetings

Las reuniones diarias se llevaron a cabo utilizando la plataforma de Google Meet, y se presenta a continuación el listado completo de las reuniones en la Tabla 20.

Tabla 20

Resumen de actividades planificadas en las daily meetings del sprint 4

Día	Actividad Planificada	Observaciones
1	Revisión y planificación del sprint	
2	Implementación de la funcionalidad de visualización de redes sociales	
3	Desarrollo de la funcionalidad de adición de redes sociales	
4	Pruebas y correcciones de errores	
5	Implementación de la funcionalidad de modificación de redes sociales	
6	Desarrollo de la funcionalidad de eliminación de redes sociales	
7	Integración y pruebas de la aplicación	Errores no detectados durante las pruebas

Día	Actividad Planificada	Observaciones
8	Preparación y configuración para la implementación en la nube	Configuración incorrecta de la plataforma de nube
9	Despliegue de la aplicación a la nube	Errores en la configuración de la nube o fallas en las pruebas
10	Solución de errores de despliegue en la nube	Errores en la configuración de la nube o fallas en las pruebas
11	Solución de errores de despliegue en la nube	
12	Reunión para revisión de módulo	
13	Retrospectiva	
14	Retrospectiva	

Aceptación del usuario

El día 07 de mayo de 2023 se llevó a cabo una reunión con la participación de todos los interesados en el proyecto. Durante esta reunión, se presentó la funcionalidad desarrollada a lo largo del sprint y se llevó a cabo su verificación. El acta de la reunión correspondiente se encuentra adjunta en el APÉNDICE A.6.

Después de la reunión de aceptación, tanto el cliente como el Product Owner manifestaron la importancia de tener en cuenta algunas sugerencias, tal como se detalla en la Tabla 21.

Tabla 21

Sugerencias para el sprint 4

N°	Sugerencia	Corrección
1	Se sugiere agregar redes sociales desde ordenador o en su defecto agregar páginas web sin enlaces de tipo móvil.	Se capacitó al usuario para que tome links que no redirijan a páginas en formato para móvil.

N°	Sugerencia	Corrección
2	Hacer script de pruebas para los usuarios.	Se realizó el script de pruebas para el módulo actual y los módulos anteriores, ver APÉNDICE C.1.
3	Realizar más pruebas a la aplicación.	Los usuarios realizan pruebas exhaustivas para encontrar errores.

Después de haber tomado en cuenta las sugerencias, las partes interesadas expresaron formalmente su aprobación de la entrega del avance del software funcional, sin requerir correcciones o modificaciones adicionales.

Resultados

Durante el transcurso de este sprint, se procedió al desarrollo de un controlador en el proyecto de Backend, el cual contiene un total de cuatro endpoints, según se detalla en la tabla. Estos endpoints se implementaron con el propósito de asegurar el correcto funcionamiento del módulo.

Tabla 22

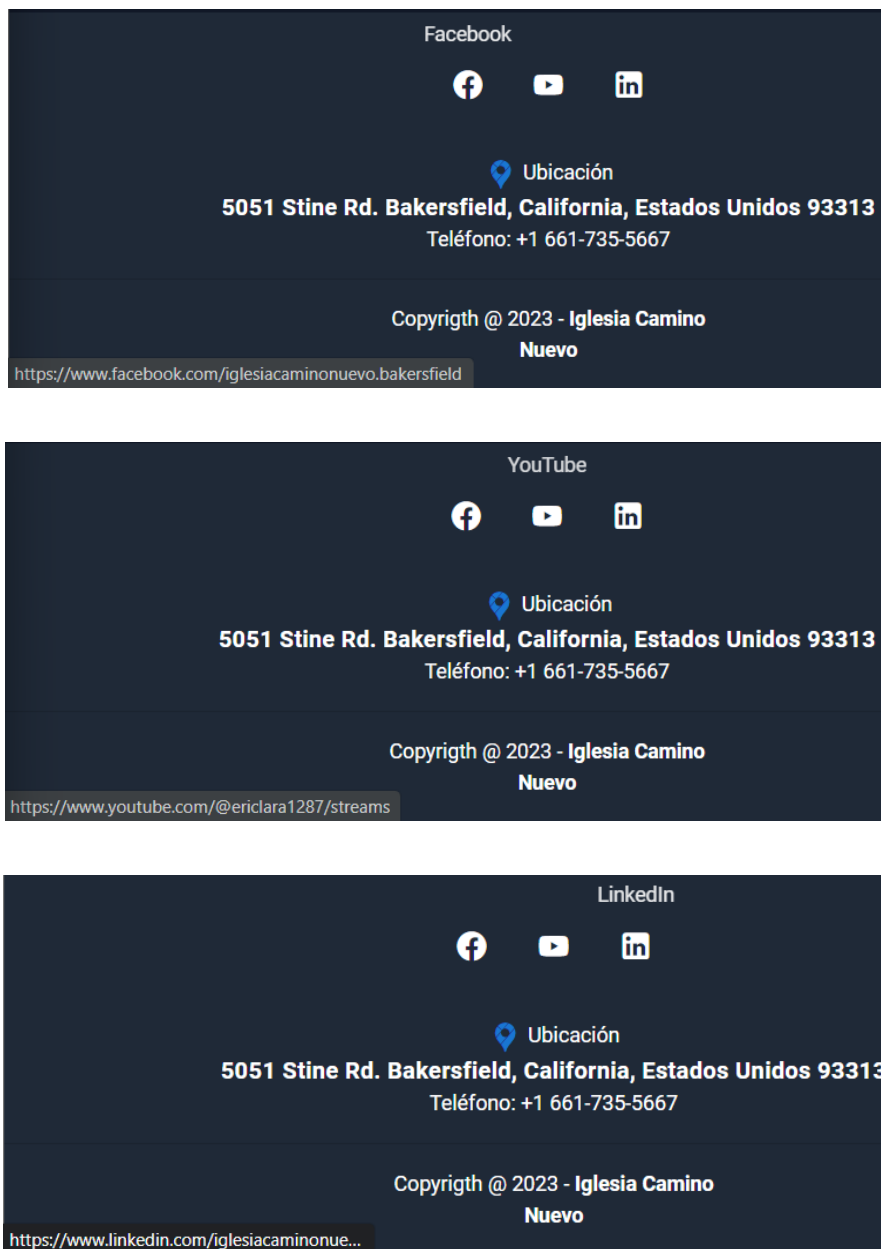
Listado de endpoints generados en el sprint 4

URI	Método	Función
social-network	GET	Obtener las redes sociales guardadas
admin-social-network	POST	Guardar nuevas redes sociales
admin-social-network/{id}	PUT	Actualizar redes sociales
admin-social-network/{id}	DELETE	Eliminar redes sociales


En el proyecto Frontend, se ha desarrollado la implementación de un componente llamado "Footer.vue", diseñado para ser mostrado como un pie de página en todas las secciones del sitio web accesibles para el usuario, esta integración se ilustra en la Figura 54.


Además, se ha creado una vista denominada "SocialNetworkManagement.vue", que se encarga de presentar al administrador las opciones disponibles para la gestión de las redes sociales. La representación visual de esta vista se muestra en la Figura 55.

Con el objetivo de preservar la fidelidad de los componentes y vistas al diseño inicial, se emplean herramientas como Vuetify, Bootstrap y SweetAlert, tal como se hizo en el primer sprint. No obstante, se destaca que Vuetify, además de proporcionar una apariencia predeterminada, ofrece la capacidad de recuperar los iconos de las redes sociales añadidas, lo que constituye una diferencia importante respecto al uso de otras tecnologías.

Figura 54*Visualización de redes sociales*

Nota: La figura presenta las tres redes sociales actualmente gestionadas por la Iglesia Camino Nuevo. En primer lugar, se muestra el icono de Facebook, el cual redirige al enlace indicado en la parte inferior izquierda. A continuación, se visualizan los iconos de Youtube y LinkedIn, los cuales siguen el mismo principio de funcionamiento.

Figura 55*Gestión de redes sociales*

Nombre	Enlace	Acciones
Facebook	https://www.facebook.com/iglesiacamionuevo.bakersfield	 
YouTube	https://www.youtube.com/@ericlara1287/streams	 
LinkedIn	https://www.linkedin.com/iglesiacamionuevo	 

Se ha generado una instancia en AWS del tipo t2.small con las siguientes especificaciones: 1 vCPU, 2 GiB de RAM y 8 GB de almacenamiento. Esta instancia ha sido configurada con el stack de programación MEVN (MongoDB, Express.js, Vue.js, Node.js) y se ha obtenido una dirección IP fija proporcionada por AWS.

Adicionalmente, la aplicación ha sido desplegada en la nube, lo que permite el acceso a cualquier usuario mediante la URL “www.icnbakersfield.com”, como se observa en la Figura 56. El nombre de dominio fue proporcionado por el Product Owner y se realizaron las configuraciones en el servidor.

Figura 56

Sitio Web desplegado al Internet

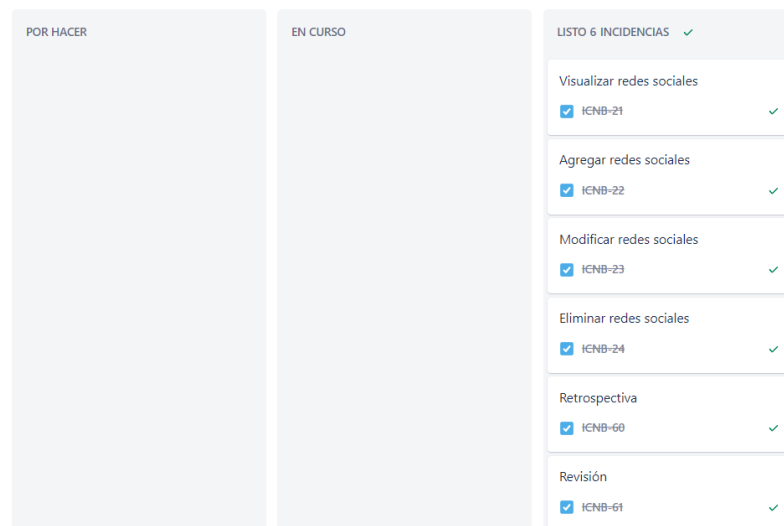


Nota: Página principal del sitio web, donde se refleja la URL con el nombre de dominio contratado.

Después de la aprobación del incremento, se procede a la migración de las incidencias registradas en Jira desde la columna "En curso" hacia la columna "Finalizado", tal y como se representa en la Figura 57.

Figura 57

Estado del tablero de Jira al finalizar el sprint 4



Sprint Review

Se llevó a cabo una sesión de revisión del sprint con la presencia del Product Owner, Edison Lascano. Durante esta reunión, se evaluó el correcto funcionamiento de las funcionalidades implementadas. Como resultado, se constató que la ejecución del sprint fue exitosa, logrando cumplir con todas las actividades planificadas en el sprint backlog dentro del plazo establecido.

Es relevante resaltar que todas las sugerencias y comentarios planteados por los stakeholders con relación al desarrollo del sprint fueron abordados y resueltos de manera satisfactoria. Estas acciones facilitaron la toma de decisión de concluir el sprint actual y dar inicio al siguiente, el cual se enfocará en la gestión de eventos.

Sprint 5: Gestión de Eventos

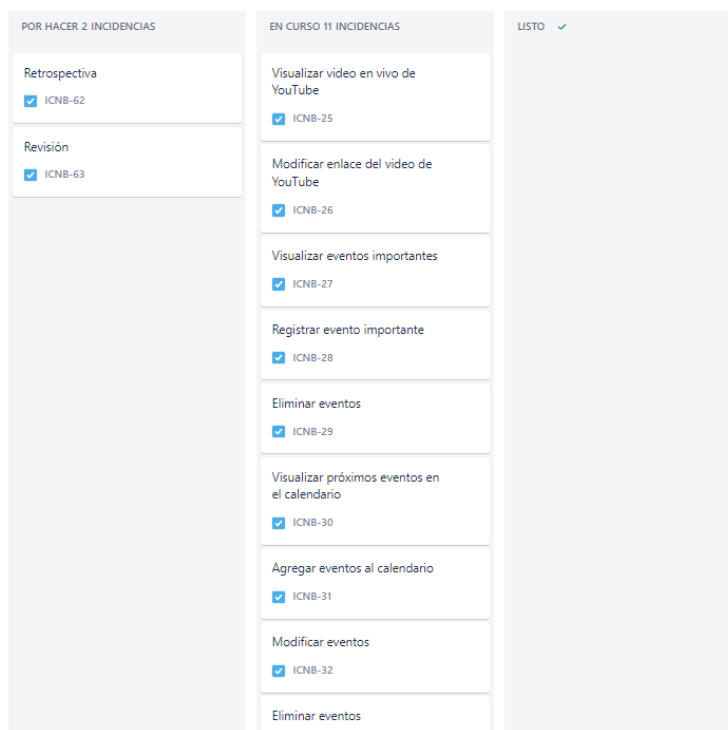
El sprint 5 (Gestión de Eventos) inicio el 08 de mayo y finalizó el 21 de mayo de 2023. Durante el transcurso de este sprint, se llevaron a cabo diversas actividades que engloban la

visualización de videos en vivo transmitidos en la plataforma YouTube, la modificación del enlace del video en vivo, la visualización, registro y eliminación de eventos importantes, así como la visualización, adición, modificación y eliminación de eventos en el calendario.

Con el fin de iniciar el nuevo sprint, se llevó a cabo la transferencia de las incidencias asignadas al sprint 5 desde la columna "Por hacer" a la columna "En curso" en el tablero de Jira, según se ilustra en la Figura 58.

Figura 58

Tablero del sprint Gestión de Eventos inicial



En el transcurso de este sprint, se ha seguido un proceso análogo al del primer sprint, dado que aborda la gestión de eventos mediante el concepto de CRUD (Crear, Leer, Actualizar, Eliminar). De manera similar, se han empleado las tecnologías y herramientas aplicadas en el primer sprint, este procedimiento se repite en otros sprints destinados a la gestión de datos.

Daily Meetings

Las reuniones diarias sirvieron para la adecuada planificación de actividades diarias, ver Tabla 23.

Tabla 23

Resumen de actividades planificadas en las daily meetings del sprint 5

Día	Actividad Planificada	Observaciones
1	Revisión y planificación del sprint	
2	Implementación de la visualización de videos en vivo de YouTube	Problemas de integración con la API de YouTube.
3	Solución y desarrollo de la funcionalidad de modificación del enlace del video en vivo	
4	Pruebas y correcciones de errores	
5	Implementación de la visualización de eventos importantes	
6	Registro de eventos importantes	Problemas de validación o persistencia de datos.
7	Solución y eliminación de eventos importantes	
8	Desarrollo de la funcionalidad de visualización de eventos en el calendario	
9	Desarrollo de la funcionalidad de adición de eventos en el calendario	
10	Preparación y configuración para la implementación en producción	Dificultades en la preparación y configuración para la implementación en producción.
11	Solución e integración y pruebas de la aplicación	
12	Reunión para revisión de módulo	
13	Retrospectiva	

Día	Actividad Planificada	Observaciones
14	Retrospectiva	

Aceptación del usuario

El día 21 de mayo de 2023 se llevó a cabo una reunión con la participación de todos los involucrados en el proyecto, en la cual se presentó la funcionalidad desarrollada durante el sprint y se procedió a verificarla. El acta de reunión se encuentra adjunta en el APÉNDICE A.7.

Posteriormente, tanto el cliente como el Product Owner expresaron formalmente su aprobación de la entrega del avance del software funcional, sin requerir correcciones o modificaciones adicionales.

Resultados

Como resultado en el backend se generaron tres controladores que abarcan 11 endpoints, tal como se muestra en la Tabla 24.

Tabla 24

Listado de endpoints generados en el sprint 5

URI	Método	Función
events/{language}	GET	Obtener los eventos según el parámetro de lenguaje ingresado
live-event	GET	Obtener los eventos en vivo
image-event	GET	Obtener las imágenes de los eventos
admin-image-event/{language}	GET	Obtener la administración de las imágenes de eventos
admin-image-event	POST	Guardar nuevas imágenes de eventos
admin-image-event/{id}	DELETE	Eliminar imágenes por id
admin-events	GET	Obtener eventos

URI	Método	Función
admin-events	POST	Guardar nuevos eventos
admin-events/{id}	PUT	Actualizar los eventos
admin-events/{id}	DELETE	Eliminar eventos por id
admin-live-event/{id}	PUT	Actualizar el evento en vivo

En el frontend se desarrolló el componente "SectionEvents.vue" con el propósito de mostrar la sección de eventos de la iglesia. Además, se implementaron cuatro vistas adicionales: "EventUserView.vue", "LiveEventManagement.vue", "ImportantEventManagement.vue" y "CalendarEventManagement.vue". Estas vistas tienen las siguientes funciones respectivamente: mostrar eventos en la pantalla del usuario, ver Figura 59; gestionar eventos en vivo, ver Figura 60; gestionar eventos importantes, ver Figura 61; y gestionar eventos en el calendario, ver Figura 62.

Con el propósito de mantener la integridad de los componentes y vistas conforme al diseño original, se utilizan herramientas tales como Vuetify, Bootstrap y SweetAlert, en línea con lo implementado en el primer sprint.

Figura 59

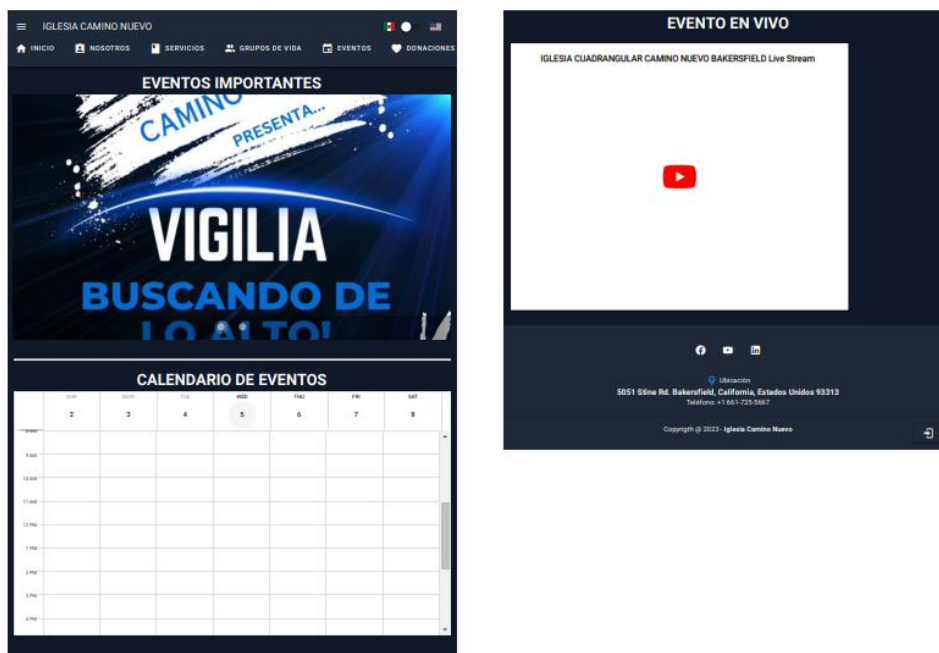
Sección eventos

Figura 60

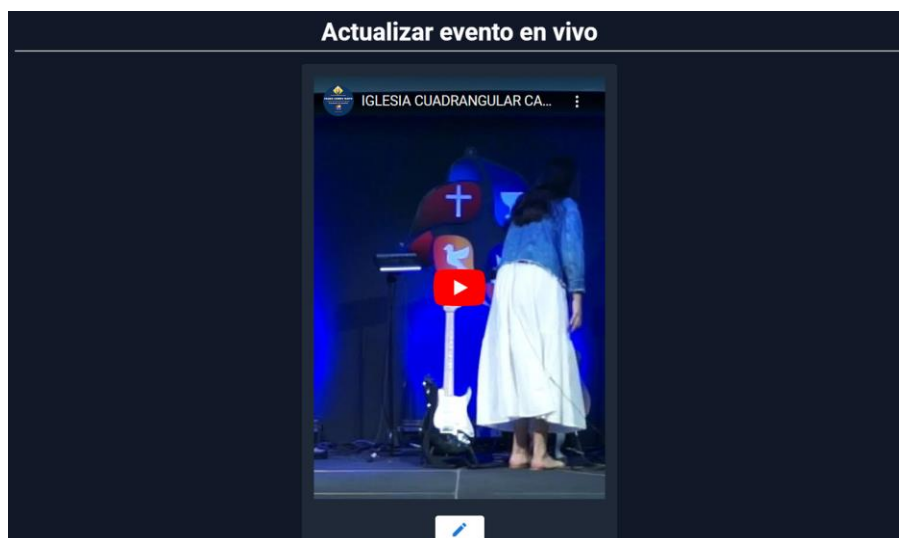
Gestión de eventos en vivo





Figura 61

Gestión de eventos importantes



Figura 62

Gestión de eventos en calendario

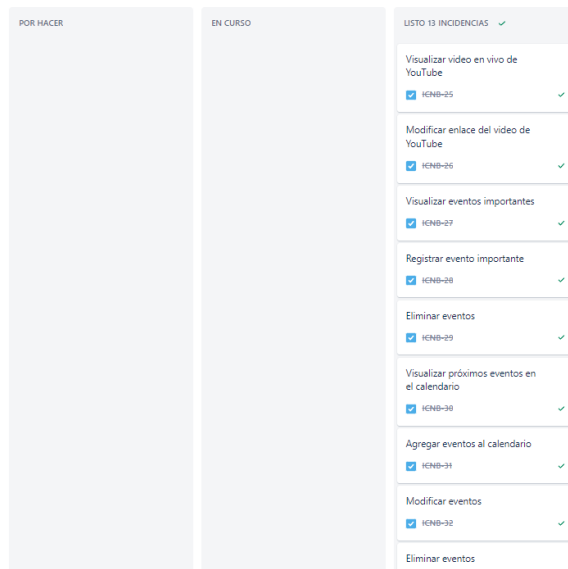
Titulo Ingles	Titulo Español	Fecha Inicio	Fecha Fin	Acciones
men event	evento de caballeros	2023-06-10 09:00	2023-06-15 10:00	 
Test Event ING	Evento ESP Prueba	2023-05-27 09:00	2023-05-27 09:05	 

Rows per page: 10 1-2 of 2

Una vez que se ha dado la aprobación al incremento, se lleva a cabo la transferencia de las incidencias previamente registradas en Jira, específicamente aquellas que se encuentran ubicadas en la columna denominada "En curso", hacia la columna designada como "Finalizado", tal como se puede observar en la Figura 63.

Figura 63

Estado del tablero de Jira al finalizar el sprint 5



Sprint Review

Se llevó a cabo la revisión del sprint en presencia del Product Owner, Edison Lascano, con el propósito de evaluar el desempeño de las funcionalidades implementadas, detalladas en la tabla adjunta. Como resultado de la revisión, se constató que la ejecución del sprint fue exitosa, habiéndose completado todas las actividades planificadas en el sprint backlog dentro del plazo establecido.

Es relevante destacar que durante la revisión no se recibieron sugerencias ni comentarios por parte de los stakeholders con relación al desarrollo del sprint. Esta situación condujo a la decisión de concluir el sprint actual y dar inicio al siguiente, que se centrará en la gestión de servicios de la iglesia Camino Nuevo.

Sprint 6: Gestión de Servicios

El sprint 6 (Gestión de Servicios) inició el 22 de mayo y concluyó el 04 de junio de 2023. Durante el desarrollo de este sprint, se llevaron a cabo diversas actividades que abarcaron la visualización de servicios, así como la inclusión, modificación y eliminación de estos.

Estas tareas se encuentran registradas en el Product Backlog y se detallan en el Sprint Backlog, donde experimentan una transición del estado "por hacer" al estado "en curso", tal y como se muestra en la Figura 64.

Figura 64

Tablero del sprint Gestión de Servicios inicial



Durante el desarrollo de este sprint, se ha seguido una metodología comparable a la del primer sprint, puesto que se enfoca en la gestión de servicios mediante el concepto de CRUD (Crear, Leer, Actualizar, Eliminar). Asimismo, se han utilizado las tecnologías y herramientas que se aplicaron en el primer sprint, y este proceso se repite en otros sprints destinados a la gestión de datos.

Daily Meetings

Las reuniones diarias se llevaron a cabo a través de la plataforma Google Meet, en estas reuniones participó el equipo de desarrollo. Las reuniones diarias se realizaron mediante el uso de la plataforma Google Meet, y en ellas participó el equipo de desarrollo.

Durante estas reuniones, se llevó a cabo la planificación de las actividades diarias, que incluyeron el desarrollo de nuevas funcionalidades, la continuación de tareas pendientes y la solución de errores. Toda esta información se encuentra detallada en la Tabla 25, en la cual se registraron también observaciones, sugerencias y novedades surgidas a lo largo del sprint.

Tabla 25

Resumen de actividades planificadas en las daily meetings del sprint 6

Día	Actividad Planificada	Observaciones
1	Planificación de Sprint	
2	Planificación de vistas que se van a desarrollar y organización de componentes visuales	
3	Adición de servicios	
4	Adición de servicios	
5	Modificación de servicios	
6	Modificación de servicios	
7	Eliminación de servicios	
8	Visualización de servicios	Problemas de redimensionamiento de componentes visuales
9	Visualización de servicios, solución de problemas encontrados	
10	Implementación del módulo en la nube	
11	Pruebas de módulo en el ambiente de nube	

Día	Actividad Planificada	Observaciones
12	Reunión para revisión de módulo	La reunión se desarrolló conforme a lo planificado
13	Retrospectiva	Corrección utilizar número de la iglesia mas no personales.
14	Retrospectiva	

Aceptación del usuario

El día 02 de junio de 2023 tuvo lugar la reunión de aceptación, en la cual participaron todos los stakeholders del proyecto. Durante este encuentro, se presentó el módulo desarrollado, exponiendo todas sus funcionalidades y su manejo. El acta de la reunión se adjunta en el APÉNDICE A.8.

Posterior a la reunión de aceptación, tanto el cliente como el Product Owner sugirieron que en la sección "Nosotros", subsección equipo pastoral no se incluya el número de teléfono personal, ya que en la iglesia se manejan las áreas bajo el número telefónico propio de la institución.

Resultados

Como resultado de este sprint, se ha implementado un controlador en el proyecto Backend. Este controlador incluye cinco endpoints, los cuales se describen en la Tabla 26. Estos endpoints se han desarrollado con el objetivo de habilitar la funcionalidad del módulo.

Tabla 26

Listado de endpoints generados en el sprint 6

URI	Método	Función
services/language	GET	Obtener los servicios según el parámetro de lenguaje ingresado para la sección de usuario
admin/services	GET	Obtener los servicios para la sección de administrador
admin/services	POST	Registrar nuevo servicio para la sección de administrador
admin/services/{id}	PUT	Actualizar el servicio seleccionado para la sección de administrador
admin/services/{id}	DELETE	Eliminar el servicio seleccionado para la sección de administrador

En el proyecto de Frontend, se han desarrollado dos vistas denominadas "SectionService.vue" y "ServicesManagement.vue". Estas vistas tienen la funcionalidad de mostrar la sección de "Servicios" y gestionar los servicios en el área de administración, respectivamente. Se puede observar las representaciones visuales de estas vistas en la Figura 65 y Figura 66.

Figura 65

Sección servicios

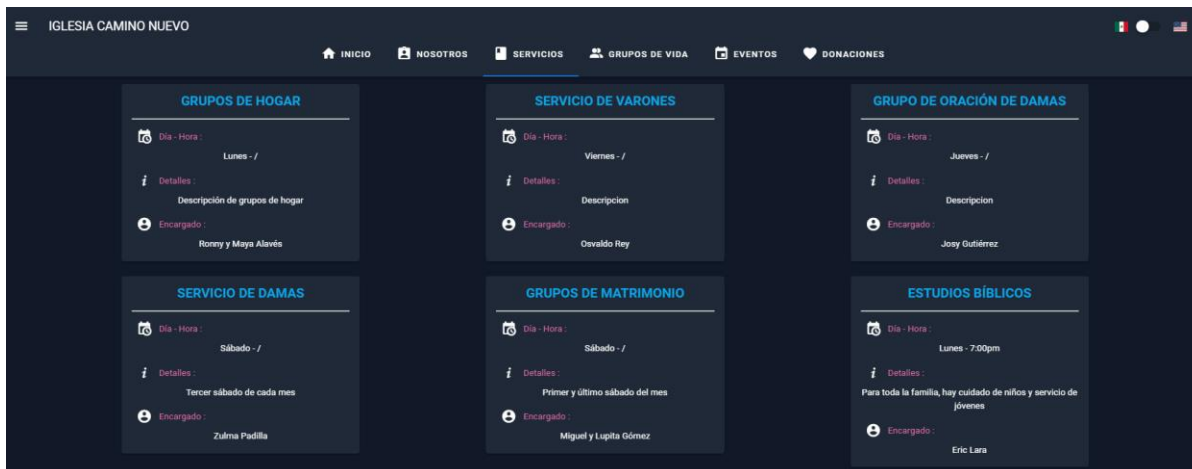
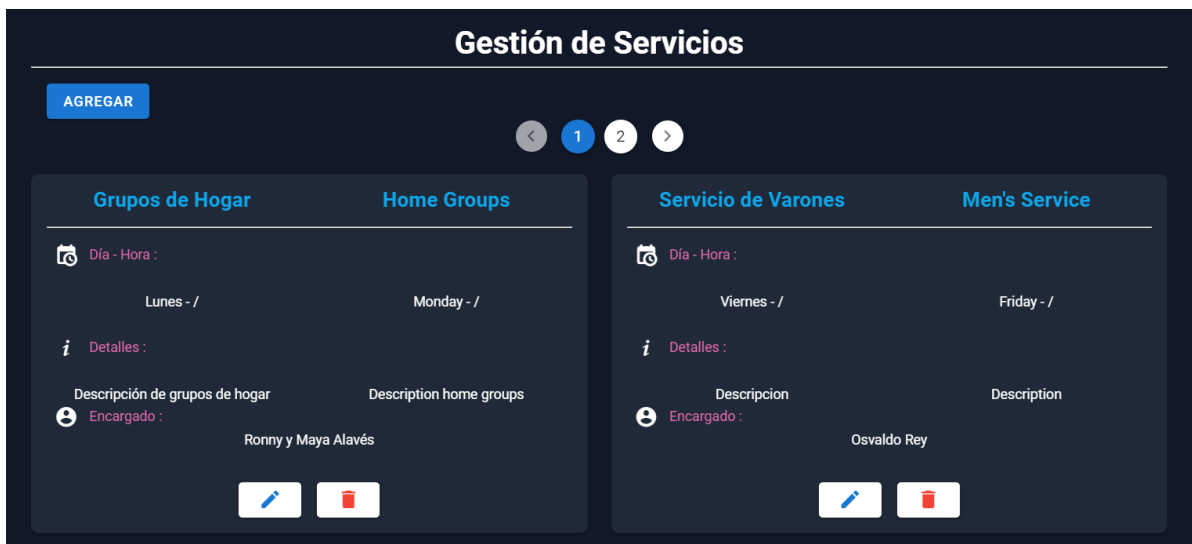


Figura 66

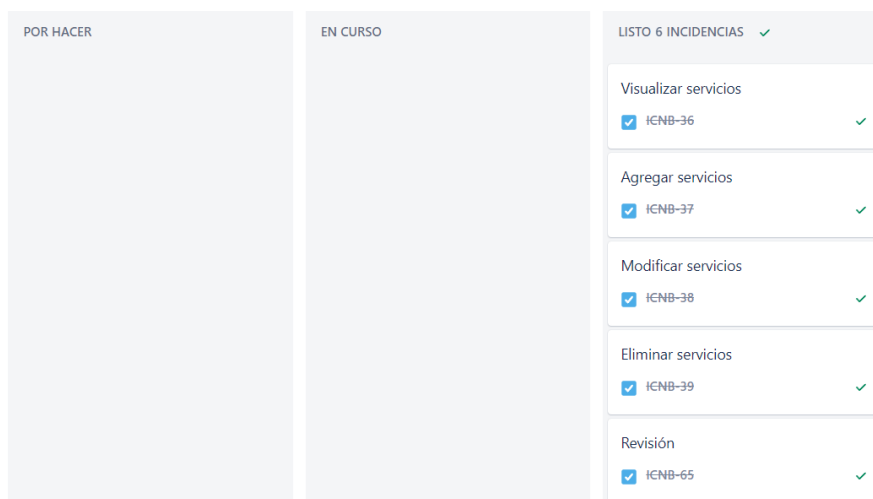
Gestión de servicios



Una vez que el incremento ha sido aprobado, se procede a transferir las incidencias registradas en Jira de la columna "En curso" a la columna "Finalizado", tal como se ilustra en la Figura 67.

Figura 67

Estado del tablero de Jira al finalizar el sprint 6



Sprint Review

Se llevó a cabo una sesión de revisión del sprint en presencia del Product Owner, Edison Lascano. Durante esta reunión, se realizó una evaluación del funcionamiento de las funcionalidades implementadas. Como resultado, se pudo constatar que la ejecución del sprint fue exitosa, logrando cumplir con todas las actividades planificadas en el sprint backlog dentro del plazo establecido.

Es destacable mencionar que se han abordado y resuelto todas las sugerencias y comentarios planteados por los stakeholders con relación al desarrollo del sprint. Estas acciones han permitido tomar la decisión de concluir exitosamente el sprint actual y dar inicio al siguiente, el cual se enfocará en la gestión de peticiones de oración.

Sprint 7: Registro y Supervisión de Peticiones de Oración

Durante el transcurso de este sprint, se han llevado a cabo diversas actividades en el marco de nuestro proyecto. Estas actividades incluyen la visualización de feligreses que requieren servicios de oración o testimonio, la solicitud de peticiones de oración o testimonio, la

visualización de feligreses que se registran como nuevos miembros de la comunidad, el registro de nuevos feligreses, la visualización de feligreses que declaran haber recibido a Cristo por primera vez, y el registro de feligreses que hacen tal declaración.

Para gestionar y dar seguimiento a estas tareas, utilizamos el tablero de Jira, donde se asignan como incidencias y se clasifican. A medida que avanzamos en el sprint, las tareas se trasladan del estado "Por hacer" al estado "En curso", como se puede apreciar en la Figura 68.

Figura 68

Tablero del sprint Registro y Supervisión de Peticiones de Oración inicial



En este sprint enfocado en el Registro y Supervisión de Peticiones de Oración, se puede apreciar que el proceso guarda similitudes con el primer sprint. Sin embargo, en esta ocasión, se maneja información proveniente de formularios que es procesada en el backend, como se observa en la Figura 69, donde la información no es totalmente requerida, es decir, existen campos que no son obligatorios, por ejemplo, el número de teléfono o correo electrónico, ya que los usuarios feligreses pueden elegir no proporcionar dicha información. Además, se ha excluido la opción de borrar registros, siguiendo una solicitud específica del

pastor principal de la Iglesia Camino Nuevo, quien expresó su necesidad de contar con una lista completa de personas interesadas en peticiones de oración para una futura auditoría. Para ello, se ha proporcionado una opción para marcar los registros ya tratados.

Figura 69

Función para recibir la información de las tarjetas de registro

```

Routers.post(`${path}new-user`, async (req, res) => {
  try {
    const user = new UserRegister({
      _id: new mongoose.Types.ObjectId(),
      name: req.body.name,
      lastname: req.body.lastname,
      gender: req.body.gender,
      petition: req.body.petition,
      description: req.body.description,
      birthday: req.body.date,
      phone: req.body.phone,
      email: req.body.email,
      agreement: req.body.agreement,
      typeRegister: req.body.typeRegister,
      newUser: true,
      registrationDate: util.getDate(),
    });
    const userSaved = await user.save();
    res.json({
      result: true,
      message: 'Request executed successfully',
      req: userSaved,
    });
  } catch (error) {
    console.log('Error save new user ' + error);
    res.json({
      result: false,
      message: 'Request not possible',
      error,
    });
  }
});

```

En relación con las herramientas empleadas, se utilizan las mismas que en el primer sprint, tanto para pruebas como para la supervisión y administración de la base de datos. Esto ha permitido, entre otros aspectos, alcanzar los objetivos planteados para el desarrollo de este sprint actual.

Daily Meetings

Las reuniones diarias se llevaron a cabo utilizando Google Meet. En estas reuniones, el equipo de desarrollo asumió el rol de Scrum Masters, además de desempeñar sus funciones

habituales. Durante las reuniones, se realizó la planificación de las actividades diarias, las cuales abarcaron el desarrollo de nuevas características, la continuación de tareas pendientes y la resolución de problemas. Estos elementos se presentan en la Tabla 27, junto con sus observaciones.

Tabla 27

Resumen de actividades planificadas en las daily meetings del sprint 7

Día	Actividad Planificada	Observaciones
1	Reunión de inicio del sprint. Establecer los objetivos y alcance del módulo	
2	Desarrollo del formulario de registro de nuevos feligreses	
3	Implementación de la visualización de feligreses que requieren oración	
4	Desarrollo del formulario de solicitud de peticiones de oración o testimonio	
5	Integración de la funcionalidad de registro de aceptación a Cristo	
6	Pruebas de funcionalidad y corrección de errores	Fallas en la interfaz de usuario, desbordamiento de componentes
7	Mejoras en la interfaz de usuario y experiencia de usuario	Se configura la sección para ser responsiva
8	Implementación de la visualización de nuevos feligreses	
9	Pruebas de integración y ajustes finales	
10	Despliegue del módulo en la nube	El navegador lanza alertas y advertencias mediante consola
11	Solución de errores presentados en la nube	
12	Reunión para revisión de módulo	Revisión de términos y condiciones

Día	Actividad Planificada	Observaciones
13	Retrospectiva	Género de los usuarios solo masculino y femenino, evitar otros.
14	Retrospectiva	

Aceptación del usuario

El día 16 de junio de 2023 se llevó a cabo una reunión en la cual participaron todos los stakeholders del proyecto. Durante la reunión, se presentó la funcionalidad desarrollada a lo largo del sprint y se procedió a su verificación. El acta de reunión se encuentra adjunta en el APÉNDICE A.9.

Tras la reunión de aceptación, tanto el cliente como el Product Owner expresaron formalmente su aceptación de la entrega del avance del software funcional, solicitando que se realice un ajuste en los formularios que requieren el ingreso del género del usuario, eliminando la opción "Otros", dado que la Iglesia y su religión reconocen únicamente dos géneros: Masculino y Femenino. Asimismo, se solicitó que se permita en los formularios la omisión del teléfono y correo electrónico, ya que el usuario puede optar por proporcionar o no esta información.

Resultados

Como resultado de este sprint, en el proyecto Backend se ha desarrollado un controlador que contiene cinco endpoints, los cuales se describen en la Tabla 28.

Tabla 28

Listado de endpoints generados en el sprint 7

URI	Método	Función
register/new-user	PUT	Modificar parámetros de usuarios registrados
register/visit	GET	Obtener los usuarios que visitan por primera vez el sitio
register/received	GET	Obtener los usuarios que recibieron a Cristo por primera vez
register/petition	GET	Obtener los usuarios que solicitan peticiones de oración
register/group-life	GET	Obtener los usuarios que solicitan información sobre grupos de vida

En el proyecto Frontend, se ha realizado la implementación de una vista llamada `UserVisitManagement.vue`, la cual tiene como función la administración de las peticiones de oración realizadas por los usuarios del sitio web. Esta vista está diseñada específicamente para el rol de administrador y permite visualizar toda la información de manera tabular, como se muestra en la Figura 70.

Esta etapa del proyecto se llevó a cabo empleando las mismas tecnologías y herramientas mencionadas en sprints anteriores. Aquí se hacen uso de los componentes de Vuetify y Bootstrap para crear tablas dinámicas que brindan al usuario una experiencia visual más agradable al visualizar la información.

Figura 70

Supervisión de peticiones de oración

Nuevos usuarios en la aplicación							
Buscar							
Apellidos/Nombres	Edad	Género	Teléfono	Email	Fecha registro	Nuevo Usuario	Contactado
Edison Steven Chinlle Peralta	22	Masculino	0960065951	chinllesteven8@gmail.com	Thu Jan 31 2611	No	
Estefany Gallegos	28	Femenino	0987654321	test@gmail.com	Thu Jan 31 2611	No	
Junior Jurado	28	Masculino	0987654321	test@gmail.com	Thu Jan 31 2611	Sí	✓

Rows per page: 10 1-3 of 3 < >

Usuarios que recibieron a Cristo como su Salvador							
Buscar							
Apellidos/Nombres	Edad	Género	Teléfono	Email	Fecha registro	Nuevo Usuario	Contactado
Michael Cobacango	24	Masculino	0999082067	michaelprm63@gmail.com	Thu Jan 31 2611	No	
Gloria Gallegos	48	Femenino	0987654321	test@gmail.com	Thu Jan 31 2611	Sí	✓

Rows per page: 10 1-2 of 2 < >

Usuarios que solicitaron Petición de Oración o Testimonios									
Buscar									
Apellidos/Nombres	Edad	Género	Teléfono	Email	Fecha registro	Tipo de Petición	Nuevo Usuario	Contactado	Descripción Petición
Andrew Murillo	13	Masculino	0987654321	test@gmail.com	Thu Jan 31 2611	Oración	Sí	✓	▼
Junior Jurado	28	Masculino	0987654321	test@gmail.com	Thu Jan 31 2611	Testimonio	Sí	✓	▼
Stalin Jurado	23	Masculino	0987654321	test@gmail.com	Thu Jan 31 2611	Oración	No		▼

Rows per page: 10 1-3 of 3 < >

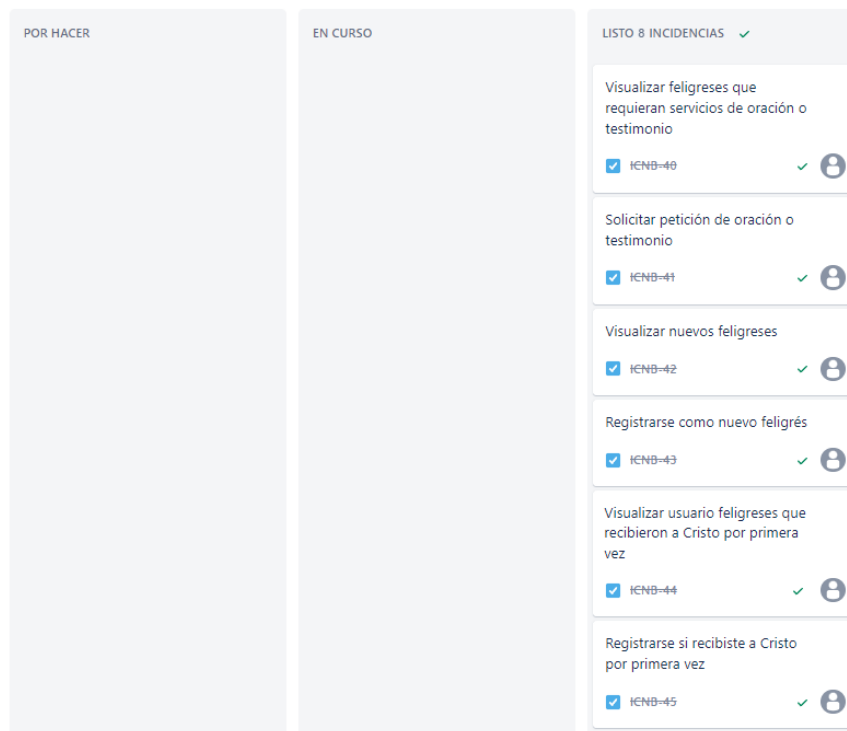
Usuarios que requieren Grupos de Vida							
Buscar							
Apellidos/Nombres	Edad	Género	Teléfono	Email	Fecha registro	Nuevo Usuario	Contactado
Valentina Peralta	9	Femenino	0987654321	test@gmail.com	Thu Jan 31 2611	Sí	✓

Rows per page: 10 1-1 of 1 < >

Una vez aprobado el incremento, se lleva a cabo la migración de las incidencias registradas en Jira desde la columna "En curso" hacia la columna "Finalizado", como se ilustra en la Figura 71.

Figura 71

Estado del tablero de Jira al finalizar el sprint 7



Sprint Review

Se llevó a cabo una sesión de revisión del sprint con la participación del Product Owner, Edison Lascano. Durante esta reunión, se realizó una evaluación del correcto funcionamiento de las funcionalidades implementadas. Como resultado, se constató que la ejecución del sprint fue exitosa, logrando cumplir satisfactoriamente con todas las actividades planificadas en el sprint backlog y dentro del plazo establecido.

Es importante destacar que todas las sugerencias y comentarios presentados por los stakeholders respecto al desarrollo del sprint fueron atendidos y resueltos de manera satisfactoria. Estas acciones contribuyeron significativamente a la toma de decisión de finalizar el sprint actual y dar paso al siguiente, el cual se centrará en la gestión de grupos de vida.

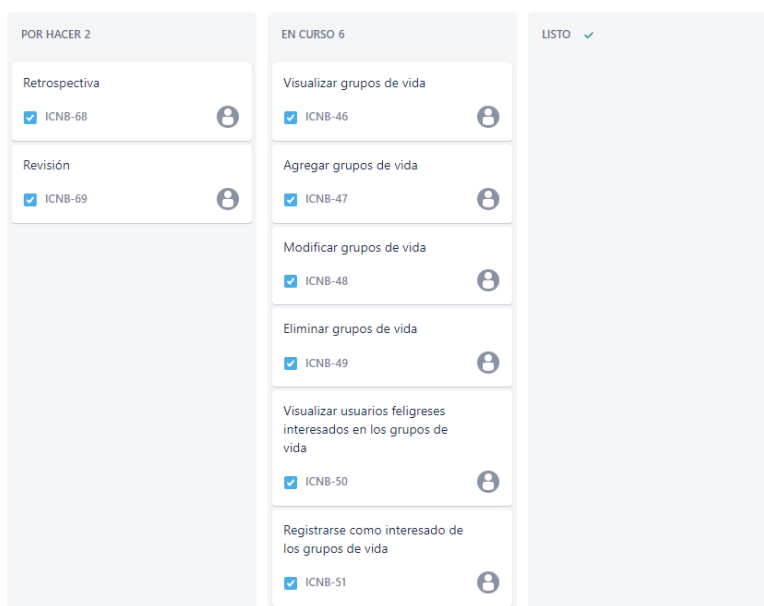
Sprint 8: Gestión de Grupos de Vida

El sprint 8 (Gestión de Grupos de Vida) inicio el 19 de junio de 2023 y finalizo el 02 de julio de 2023. A medida que el sprint avanza, se llevan a cabo diversas tareas, las cuales abarcaron la visualización, adición, modificación y eliminación de grupos de vida, así como la visualización de usuarios feligreses interesados en los grupos y la posibilidad de registrarse como interesado en los mismos.

Para llevar a cabo el desarrollo de estas actividades, se procedió a transferir las incidencias correspondientes desde la columna "Por hacer" hacia la columna "En curso" en el tablero de Jira, tal como se muestra en la Figura 72.

Figura 72

Tablero del sprint Gestión de Grupos de Vida inicial



En el presente sprint, se aborda la temática de la gestión de grupos de vida. Este sprint guarda una notable similitud con el proceso descrito en el primer sprint. Se inicia con la implementación del diseño ágil de la base de datos y posteriormente se establece la conexión

con la misma. A continuación, se llevan a cabo las operaciones CRUD (Crear, Leer, Actualizar y Borrar) que están destinadas a garantizar la adecuada funcionalidad del módulo.

Para el desarrollo del backend, se emplean las mismas tecnologías previamente mencionadas. Este enfoque metodológico, seguido en el presente sprint, se replica de manera consistente en varios módulos dedicados a la gestión de datos.

Daily Meetings

Las reuniones diarias se llevaron a cabo utilizando la plataforma Google Meet, con la participación de los miembros del equipo de desarrollo, quienes asumieron el rol adicional de Scrum Master, además de sus propias responsabilidades. Durante las reuniones, se llevó a cabo la planificación de las actividades diarias, las cuales abarcaron el desarrollo de nuevas características, la continuación de tareas pendientes y la resolución de problemas. Estos aspectos se encuentran detallados en la Tabla 29, junto con las observaciones.

Tabla 29

Resumen de actividades planificadas en las daily meetings del sprint 8

Día	Actividad Planificada	Observaciones
1	Reunión de inicio del sprint y definición de objetivos	
2	Implementación de visualización de grupos de vida	
3	Corrección de la interfaz de usuario para la visualización de grupos de vida	
4	Implementación de adición de grupos de vida	Envío de información no validada
5	Validación de campos y parámetros	
6	Implementación de modificación de grupos de vida	

Día	Actividad Planificada	Observaciones
7	Implementación de eliminación de grupos de vida	
8	Implementación de visualización de usuarios interesados en grupos de vida	
9	Implementación del módulo al servidor en la nube	Errores en los métodos para guardar la información
10	Corrección de errores con respecto al módulo en el servidor	
11	Pruebas al aplicativo subido al servidor en la nube	
12	Reunión para revisión de módulo	
13	Retrospectiva	
14	Retrospectiva	

Aceptación del usuario

El día 30 de junio de 2023 se llevó a cabo una reunión en la que participaron todos los actores involucrados en el proyecto, incluyendo el Development Team compuesto por Edison Chinlle y Michael Cobacango, el Scrum Master Edison Lascano, el Product Owner Mauro Narváez y el cliente/usuario Eric Lara. Durante esta reunión, se presentaron las funcionalidades desarrolladas a lo largo del sprint y se procedió a su validación y verificación. El acta de reunión se encuentra adjunta en el APÉNDICE A.10.

Tras la reunión de aceptación, tanto el Cliente como el Product Owner aceptaron formalmente la entrega del avance del software funcional sin requerir correcciones o cambios adicionales

Resultados

Dentro del proyecto de Backend, se ha llevado a cabo el desarrollo de un controlador que abarca un total de cinco endpoints, según se detalla en la Tabla 30. Estos endpoints han

sido concebidos con el propósito de proporcionar la funcionalidad de gestión de grupos de vida, permitiendo así la modificación o creación de datos referentes a los grupos de vida que son administrados en la iglesia Camino Nuevo.

Tabla 30

Listado de endpoints generados en el sprint 8

URI	Método	Función
group-life/{language}	GET	Obtener la información de los grupos de vida por lenguaje seleccionado
admin-group-life	GET	Obtener los grupos de vida por parte del administrador
admin-group-life	POST	Enviar datos para la creación de nuevos grupos de vida
admin-group-life/{id}	PUT	Actualizar grupo de vida seleccionado
admin-group-life/{id}	DELETE	Eliminar grupo de vida seleccionado

En el contexto del proyecto de Frontend, se ha creado un componente denominado "SectionGroupLife.vue" y dos vistas denominadas "GroupLifeView.vue" y "GroupLifeManagement.vue". El propósito del componente desarrollado es mostrar los grupos de vida que la iglesia ofrece en forma de tarjetas, según se muestra en la Figura 73. Por otro lado, las vistas implementadas engloban las funcionalidades de mostrar los componentes creados y presentar la sección de administración de los grupos de vida, tal como se ilustra en la Figura 74 y Figura 75, respectivamente.

De manera similar a sprints anteriores, con el propósito de preservar la coherencia y fidelidad de los componentes y vistas con respecto al diseño inicial, se recurrió nuevamente a las herramientas mencionadas previamente, tal como se especificó en el sprint inicial.

Figura 73

Componente Tarjetas de Grupos de Vida

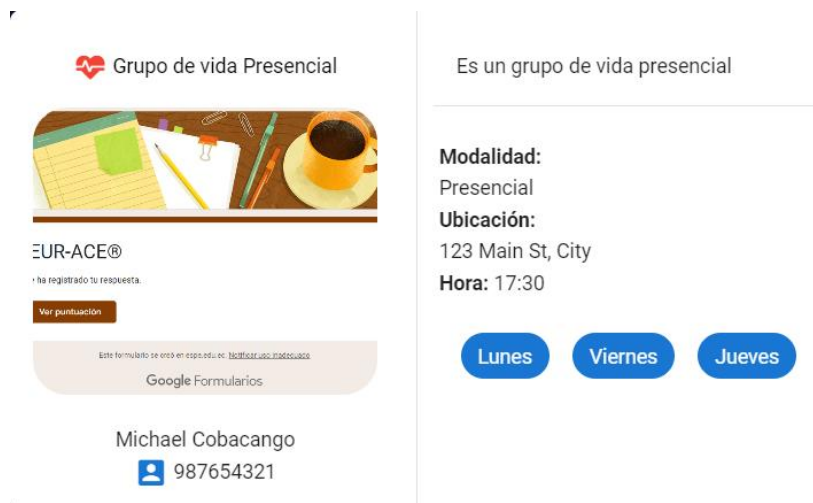


Figura 74

Sección Grupos de Vida

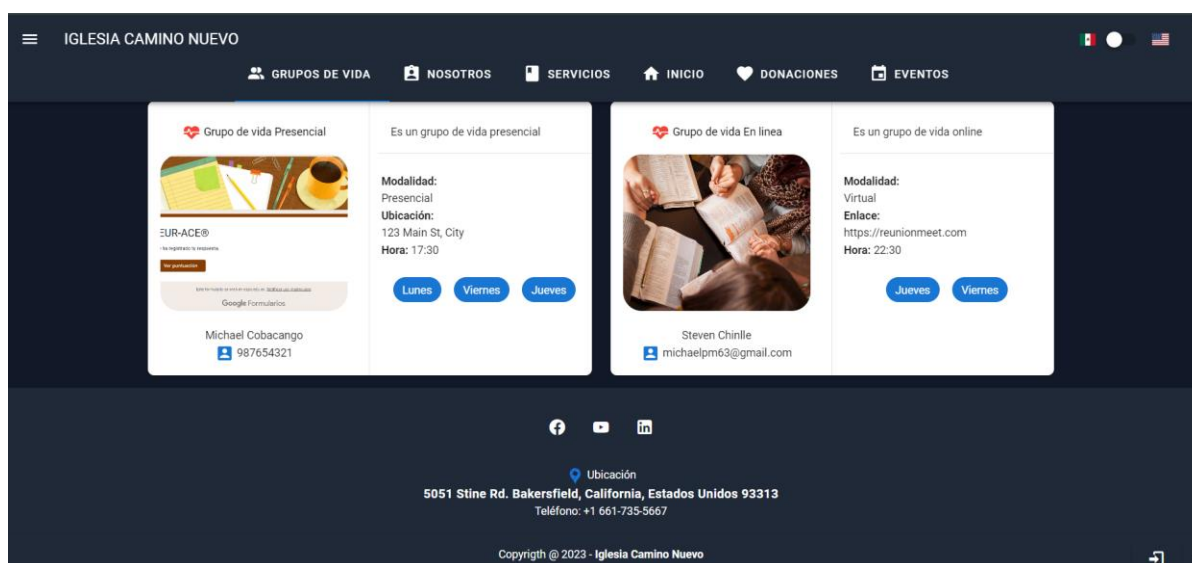


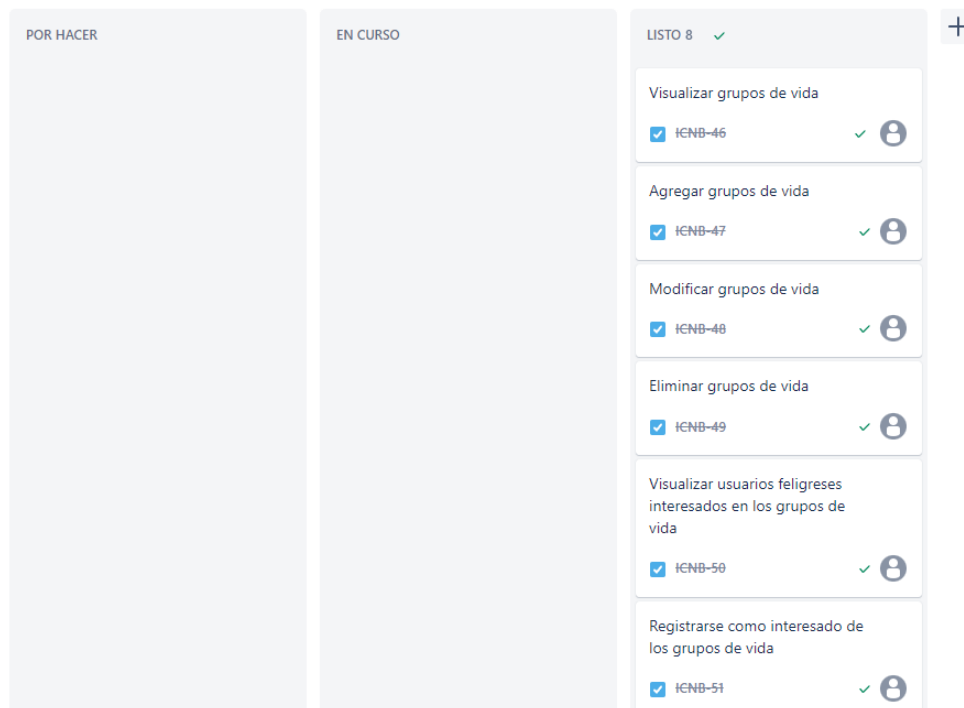
Figura 75

Gestión de Grupos de Vida

Después de que el incremento ha sido aprobado, se procede a transferir las incidencias previamente registradas en Jira desde la columna "En curso" hacia la columna "Finalizado", tal y como se representa en la Figura 76.

Figura 76

Estado del tablero de Jira al finalizar el sprint 8



Sprint Review

Se llevó a cabo una sesión de revisión del sprint en la que estuvo presente el Product Owner, Edison Lascano. Durante esta reunión, se llevó a cabo una evaluación del correcto funcionamiento de las funcionalidades implementadas. Como resultado, se pudo constatar que la ejecución del sprint fue exitosa, logrando cumplir con todas las actividades planificadas en el sprint backlog dentro del plazo establecido.

Sprint 9: Recepción de Donaciones

El sprint 9 (Recepción de Donaciones) inició el 03 de julio y finalizó el 19 de julio de 2023. En el transcurso de este sprint, se han ejecutado diversas actividades relacionadas con la recepción de donaciones y su visualización.

Las tareas han sido asignadas en el tablero de Jira, donde se han transferido del estado "Por hacer" al estado "En curso", como se puede observar en la Figura 77.

Figura 77

Estado del tablero de Jira al iniciar el sprint 9



Durante el transcurso de este sprint, se ha seguido un enfoque distinto respecto a los sprints de gestión, con el propósito de abordar la recepción de donaciones mediante la integración de API's externas propias del servicio de donaciones escogido (PayPal). El proceso inicia con la creación de una cuenta de desarrollador en PayPal, lo cual permite el acceso y consumo de las API's REST proporcionadas por el servicio.

A continuación, se procede a crear dos nuevas colecciones encargadas de proporcionar la información necesaria a las nuevas vistas y componentes. Para realizar la solicitud de donación a PayPal, se envía como parámetro la cantidad de donación, la cual es obtenido desde el método "getPayment" y se sigue el flujo establecido en la Figura 78, hacia la API REST correspondiente, finalmente se crea el pago y se envía, ver Figura 79. Cabe destacar que la API ha sido configurada previamente como se aprecia en la Figura 80. Es importante mencionar que PayPal se encarga de manejar la solicitud y, como resultado genera un token único que garantiza la autenticidad y seguimiento de la donación.

Adicionalmente, se han implementado otras dos API's encargadas de informar sobre el estado de la solicitud. La primera informa de que la donación se haya completado correctamente, mientras que la segunda muestra el estado de error si la donación no se ha efectuado con éxito. El resto del proceso es manejado internamente por el servicio.

Figura 78

Configuración de la solicitud de donación

```
Routers.get('/payment/:donationAmount', (req, res) => {
  const donationAmount = req.params.donationAmount;
  const paymentDetails = {
    intent: 'sale',
    payer: {
      payment_method: 'paypal',
    },
    redirect_urls: {
      return_url: 'http://localhost:3336/success',
      cancel_url: 'http://localhost:3336/cancel',
    },
    transactions: [
      {
        item_list: {
          items: [
            {
              name: 'DAR',
              sku: 'DONATION001',
              price: donationAmount,
              currency: 'USD',
              quantity: 1,
            },
          ],
        },
        amount: {
          currency: 'USD',
          total: donationAmount,
        },
        description: 'DONACIÓN PARA LA IGLESIA CAMINO NUEVO',
      },
    ],
  },
};
```

Figura 79*Creación y envío de la solicitud de donación*

```

paypal.payment.create(paymentDetails, async (error, payment) => {
  try {
    if (error) {
      console.error(error);
      res.sendStatus(500);
    } else {
      const redirectUrl = await payment.links.find((link) => link.rel === 'approval_url')
        .href;
      res.json(redirectUrl);
    }
  } catch (error) {
    console.log('Error get url paypal ' + error);
  }
});
});

```

Figura 80*Configuración de conexión al servicio de PayPal*

```

Routers.use((req, res, next) => {
  res.setHeader('Access-Control-Allow-Origin', '*');
  res.setHeader('Access-Control-Allow-Methods', 'GET, POST, OPTIONS');
  res.setHeader('Access-Control-Allow-Headers', 'Origin, Content-Type, Accept');
  next();
});

paypal.configure({
  mode: 'sandbox',
  client_id: process.env.CLIENT_PAYPAL,
  client_secret: process.env.PASSWORD_PAYPAL,
});

```

Daily Meetings

Las reuniones diarias se llevaron a cabo mediante la plataforma Google Meet, donde se trataron temas relacionados a la recepción de donaciones, ver Tabla 31.

Tabla 31*Resumen de actividades planificadas en las daily meetings del sprint 9*

Día	Actividad Planificada	Observaciones
1	Reunión de planificación del sprint. Definición de objetivos y alcance.	
2	Desarrollo del componente para mostrar las opciones de donación.	
3	Desarrollo del componente para mostrar las opciones de donación.	
4	Implementación de la funcionalidad para agregar nuevas donaciones.	
5	Implementación de la funcionalidad para agregar nuevas donaciones.	
6	Integración de la pasarela de pagos para recibir donaciones en línea.	
7	Integración de la pasarela de pagos para recibir donaciones en línea.	Error al momento de redireccionar al servicio de PayPal
8	Pruebas del componente de donaciones y corrección de errores.	Error al momento de finalizar la donación, redirección incorrecta
9	Pruebas del componente de donaciones y corrección de errores.	
10	Despliegue de la aplicación al servidor en la nube	Errores de conexión de la aplicación frontend y backend en el servidor
11	Corrección de errores del módulo en el servidor	
12	Reunión para revisión de módulo	
13	Retrospectiva	
14	Retrospectiva	

Aceptación del usuario

El 14 de julio de 2023 se llevó a cabo una reunión en la que participaron todos los involucrados en el proyecto. Durante esta reunión, se presentó la funcionalidad desarrollada a lo largo del sprint y se procedió a su verificación. El acta de reunión correspondiente se encuentra en el APÉNDICE A.11.

Después de la reunión de aceptación, tanto el cliente como el Product Owner manifestaron la posibilidad de agregar un código QR que redirecciona a una pasarela de pagos propia de la iglesia, además de presentar más opciones al usuario para realizar su donación, entre estas opciones se encuentran: donación mediante SMS y donación en efectivo. Además, se acordó mantener la opción de donación por PayPal, pero con la opción de permitir ocultarla y mostrarla según sea pertinente.

Resultados

Como resultado de este sprint, en el proyecto Backend se implementaron tres controladores que contienen un total de 13 endpoints, tal como se menciona en la Tabla 32. Estos endpoints fueron desarrollados con el objetivo de asegurar la satisfacción del requerimiento inicial, así como de incorporar los dos nuevos identificados durante la revisión.

Tabla 32

Listado de endpoints generados en el sprint 9

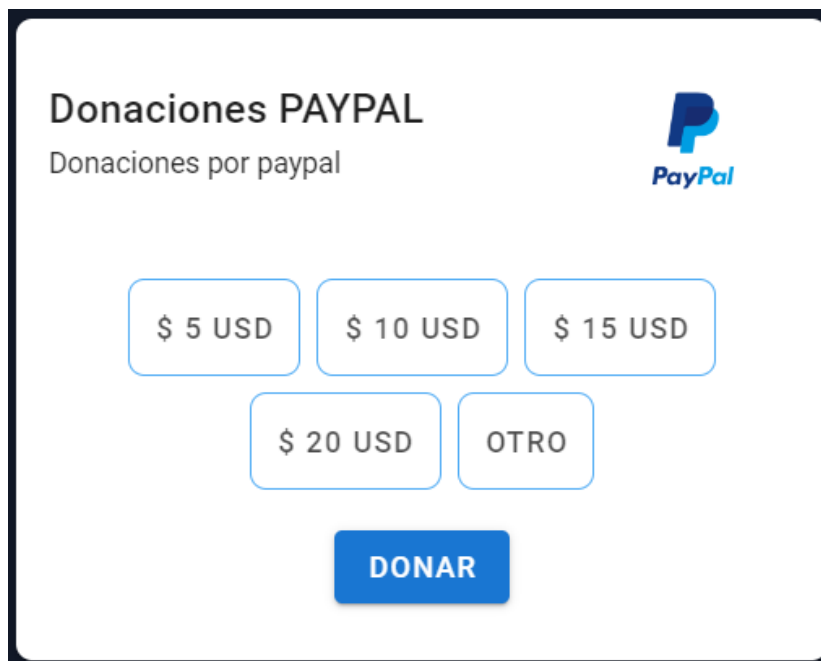
URI	Método	Función
donation/{language}	GET	Obtener la información para las tarjetas por lenguaje
donationshow/{id}	PUT	Cambiar el estado del componente para mostrarlo u ocultarlo
donation	PUT	Actualizar información de la tarjeta que muestra donación por PayPal

URI	Método	Función
donation	GET	Obtener información de las tarjetas de donación para administrador
payment/{donationAmount}	GET	Realizar una petición de donación al servicio externo PayPal
success	GET	Obtener el estado de la donación en caso de ser exitosa
cancel	GET	Obtener el estado de la donación en caso de ser errónea
donation-way/{language}	GET	Obtiene la información de otras formas de donar por lenguaje
donation-way	GET	Obtener la información de otras formas de donar para administrador
donation-way	PUT	Actualiza a información de otras formas de donar
donation-qr/{language}	GET	Obtener la información de donación por QR
donation-qr	GET	Obtener la información de donación por QR para usuario administrador
donation-qr/{id}	PUT	Actualizar la información de la tarjeta de donación por código QR

Dentro del contexto del proyecto de Frontend, se han implementado tres componentes con los nombres de "CardDonationPaypal.vue," "CardWayDonation.vue," y "DonationQR.vue." El primero de ellos muestra la información de la tarjeta de donaciones por PayPal, ver Figura 81, el segundo presenta la información de la tarjeta de donaciones mediante código QR, ver Figura 82, mientras que el tercero se encarga de exhibir la información de las tarjetas de donaciones por otras formas, ver Figura 83.

Figura 81

Tarjeta de donación por PayPal

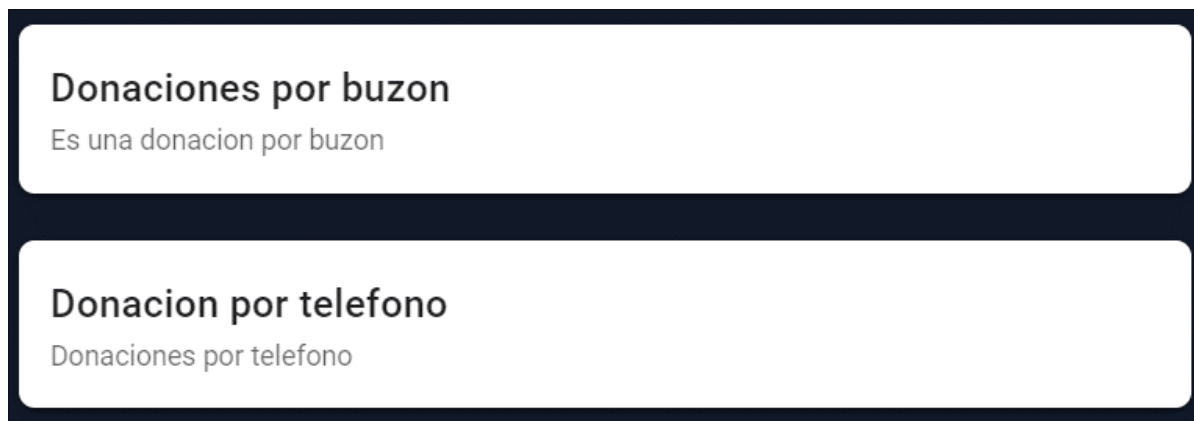
**Figura 82**

Tarjeta de donación por código QR



Figura 83

Tarjeta de donación por otras formas



Además, se han creado cinco vistas distintas para abordar diferentes aspectos de la aplicación. Estas vistas reciben los nombres de "DonationView.vue," que muestra la sección de donaciones, ver Figura 84; "SectionDonation.vue," que exhibe las tarjetas con los tipos de donaciones disponibles, ver Figura 85; "CardDonationPaypalManagement.vue", "CardDonationQRManagement.vue," y "CardDonationWayManagement.vue" destinadas al usuario administrador para mostrar las tarjetas de donación por PayPal, por código QR y por otras formas de pago, respectivamente, ver Figura 86, Figura 87 y Figura 88. Cada componente y vista desempeña un papel esencial en el proyecto, facilitando la recepción de donaciones en sus diversas modalidades.

Figura 84

Sección de donaciones en el menú



Figura 85

Tarjetas para métodos de donación

Figura 86

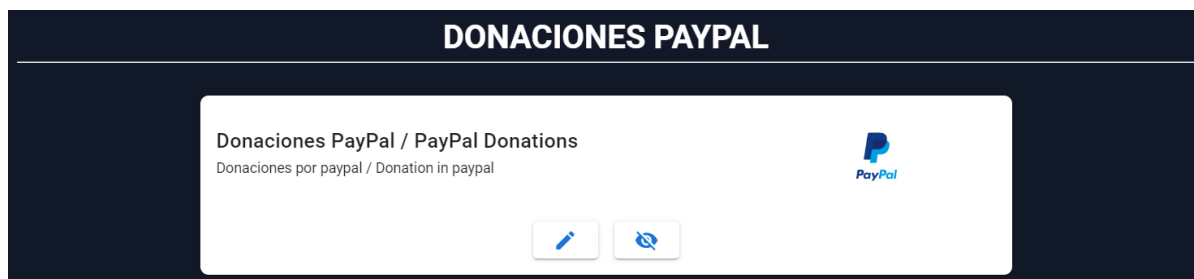
Administración de donaciones por PayPal

Figura 87

Administración de donaciones por código QR

**Figura 88**

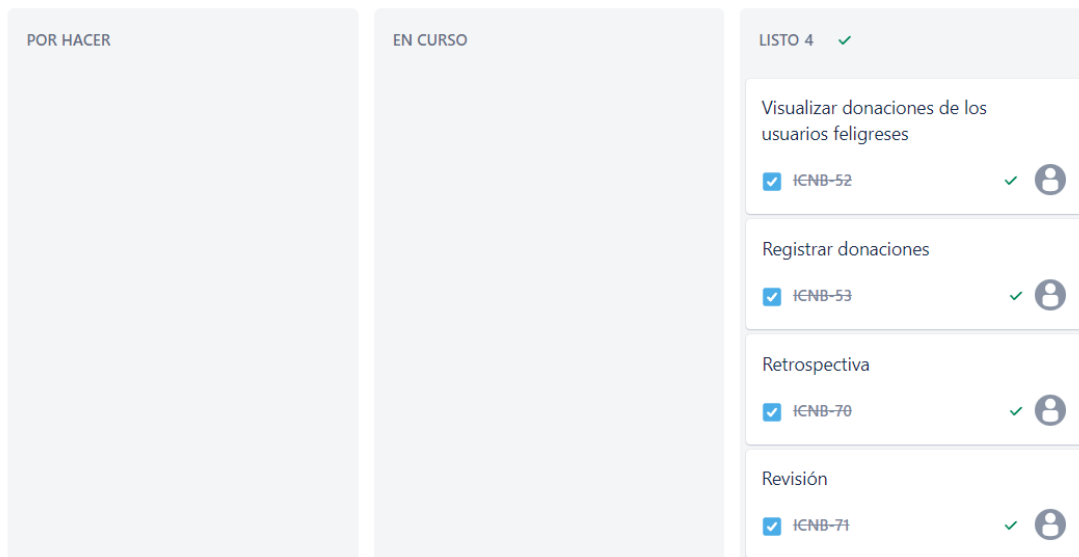
Administración de otros tipos de donaciones



Una vez que el incremento ha sido aprobado, se realiza la transferencia de las incidencias registradas en Jira desde la columna "En curso" hacia la columna "Finalizado", como se muestra en la Figura 89.

Figura 89

Estado del tablero de Jira al finalizar el sprint 9



Por último, se exhibe la sección correspondiente al Product Backlog del proyecto una vez finalizado el último sprint, tal y como se muestra en la Figura 90, donde se aprecia que todas las tareas tienen el estado de “Finalizado”.

Figura 90

Product Backlog final

▼ Backlog (9 incidencias)		0 0 0	Crear sprint
<input checked="" type="checkbox"/>	ICN-10 Gestión de información	FINALIZADA	EP
<input checked="" type="checkbox"/>	ICN-11 Cambio de idioma	FINALIZADA	MC
<input checked="" type="checkbox"/>	ICN-12 Gestión de contenido multimedia	FINALIZADA	EP
<input checked="" type="checkbox"/>	ICN-13 Gestión de redes sociales	FINALIZADA	MC
<input checked="" type="checkbox"/>	ICN-14 Gestión de eventos	FINALIZADA	EP
<input checked="" type="checkbox"/>	ICN-15 Gestión de servicios	FINALIZADA	MC
<input checked="" type="checkbox"/>	ICN-16 Registro y supervisión de peticiones de oración	FINALIZADA	EP
<input checked="" type="checkbox"/>	ICN-17 Gestión de grupos de vida	FINALIZADA	MC
<input checked="" type="checkbox"/>	ICN-18 Recepción de donaciones	FINALIZADA	EP

Sprint Review

Durante la revisión del sprint, se llevó a cabo una sesión con el Product Owner en la cual se realizó un recorrido por las funcionalidades implementadas, con el objetivo de revisar su funcionamiento y estado final. La ejecución del sprint se cumplió de manera satisfactoria, habiéndose completado el desarrollo de las historias de usuario dentro del plazo establecido, además de los dos nuevos requerimientos que surgieron durante la reunión de aceptación con el usuario.

En este sprint, se ha iniciado la consideración y abordaje de los asuntos relacionados con la migración del aplicativo a los servidores propios del usuario, con el propósito de transferir los permisos necesarios para que la aplicación sea entregada con éxito. Se ha iniciado el análisis de la creación de cuentas y la planificación de la última reunión de entrega formal, con el objetivo de entregar la aplicación completa con su respectiva documentación desde configuraciones de servidor hasta configuraciones de aplicación y bases de datos.

Con la finalización de este sprint, se da por concluido de manera satisfactoria el desarrollo integral del proyecto, en pleno cumplimiento de los requerimientos iniciales y aquellos surgidos de manera adicional durante las revisiones realizadas con el cliente.

Finalmente, se procedió con la elaboración de un formulario con el propósito de evaluar la usabilidad del sitio web y verificar el grado de satisfacción de los usuarios feligreses en relación con los requerimientos y las necesidades de estos, las preguntas realizadas se muestran en la Tabla 33. Tras la ejecución del cuestionario, se ha constatado que el sitio web efectivamente satisface las expectativas de los usuarios, tal como se evidencia en la Figura 91, donde se observa que la mayoría de los resultados son favorables.

Tabla 33

Preguntas de la Encuesta de Usabilidad

Nº	Pregunta
1	¿Encuentras fácilmente la información que buscas en el sitio web?
2	¿Consideras que la navegación del sitio web es intuitiva y clara?
3	¿Has utilizado la funcionalidad de cambio de idioma para ver el contenido en tu idioma preferido?
4	¿Cómo calificarías la calidad del contenido multimedia disponible en el sitio web (imágenes, videos, audio)?
5	¿Recomendarías el sitio web de la Iglesia Camino Nuevo a otros feligreses?
6	¿Te resulta fácil inscribirte en grupos de vida o realizar peticiones de oración a través del sitio web?
7	¿Consideras que la presentación visual y el diseño del sitio web son atractivos y coherentes con la identidad de la Iglesia?
8	¿Has experimentado algún problema de rendimiento o carga lenta al navegar por el sitio web?
9	¿Consideras que la información sobre los servicios religiosos (horarios, ubicaciones, temáticas) es clara y detallada en el sitio web?
10	¿Tienes alguna sugerencia o comentario adicional para mejorar la experiencia del sitio web de la Iglesia Camino Nuevo?

Figura 91

Resultados de la Encuesta de Usabilidad

Marca temporal	1. ¿Encuentras	2. ¿Consideras	3. ¿Has utiliza	4. ¿Cómo calif	5. ¿Recomend	6. ¿Te resulta	7. ¿Considera	8. ¿Has experi	9. ¿Considera	10. ¿Tienes alg
7/08/2023 7:36:14	A veces	De acuerdo	Sí, ocasionalme	Bueno	Sí, probableme	Sí, en general	De acuerdo	No, nunca he te	En general, aur	No
7/08/2023 7:39:24	Casi siempre	De acuerdo	Sí, ocasionalme	Bueno	Sí, probableme	Sí, en general	Neutral	Sí, frecuenteme	En general, aur	No
7/08/2023 8:54:02	A veces	De acuerdo	No, no lo sabía	Bueno	Sí, probableme	Sí, muy fácil	Neutral	No, nunca he te	Sí, la informac	No
7/08/2023 9:07:02	Casi siempre	Neutral	Sí, ocasionalme	Bueno	Sí, definitivame	Sí, muy fácil	De acuerdo	No, nunca he te	En general, aur	.
7/08/2023 9:49:14	Casi siempre	De acuerdo	No, no lo sabía	Bueno	Sí, probableme	Sí, en general	De acuerdo	Sí, frecuenteme	Sí, la informac	No, todo esta bi
7/08/2023 9:52:17	Casi siempre	De acuerdo	Sí, ocasionalme	Bueno	Sí, probableme	Sí, muy fácil	De acuerdo	No, nunca he te	Sí, la informac	No
7/08/2023 10:07:2	Siempre	Totalmente de e	Sí, ocasionalme	Excelente	Sí, definitivame	Sí, en general	De acuerdo	No, nunca he te	Sí, la informac	Ninguna
7/08/2023 10:20:5	A veces	Neutral	Sí, ocasionalme	Bueno	Sí, definitivame	No, encuentro	Totalmente de e	Sí, en algunas	(Sí, la informac	Más informac
7/08/2023 10:25:2	Siempre	De acuerdo	No, no es relev	Bueno	Sí, definitivame	Sí, muy fácil	Totalmente de e	No he notado n	Sí, la informac	No ninguna me
7/08/2023 10:31:0	A veces	En desacuerdo	Sí, siempre	Malo	No estoy segur	A veces es con	Totalmente de e	No he notado n	A veces, encue	nada

Asimismo, se proporciona la dirección del repositorio que aloja el proyecto <https://github.com/MichaelCobacango/IglesiaCaminoNuevo.git>. Es importante destacar que este repositorio es de carácter privado. Si surge la necesidad de acceder al código fuente, le solicitamos que se ponga en contacto directamente con los desarrolladores para gestionar dicha visualización.

Capítulo V

Conclusiones

En conclusión, el desarrollo del sitio web para la Iglesia Camino Nuevo ha sido ejecutado de manera satisfactoria, cumpliendo con los objetivos establecidos al inicio del proyecto. La revisión sistemática de literatura y el estudio de las iglesias en el contexto post pandemia COVID-19 en el mundo digital, permitieron identificar desafíos y oportunidades para la adaptación al nuevo entorno digital, lo cual influyó en la selección de las tecnologías y estrategias implementadas en el proyecto.

Mediante la revisión sistemática de literatura y estudio de tecnologías actuales, se procedió a seleccionar el stack MEVN (Mongo, Express, Vue, Node) para el desarrollo del sitio web. La elección de utilizar la computación en la nube demostró ser acertada, ya que proporcionó una infraestructura sólida y escalable para el funcionamiento óptimo del sitio web.

La comunicación continua con los stakeholders y la aplicación de la metodología Scrum aseguraron una comprensión adecuada de los requisitos y una planificación efectiva del desarrollo. Este enfoque altamente eficiente y colaborativo resultó ser un recurso valioso para fomentar una comunicación efectiva, independientemente de la ubicación geográfica entre el equipo de desarrollo y Scrum Master, con el cliente y el Product Owner, y culminar con éxito cada sprint y consecuentemente el proyecto final.

El product backlog se estructuró de manera completa, priorizada y organizada, reflejando los requerimientos de la Iglesia y guiando el desarrollo de los módulos. Como resultado, se ha alcanzado una base sólida que asegura la escalabilidad del proyecto. La implementación del despliegue continuo permitió una implementación ágil y eficiente de las funcionalidades, asegurando la estabilidad del sistema y la liberación sin errores de nuevos módulos.

Las pruebas de aceptación de usuario y la retroalimentación recibida han sido fundamentales en el proceso de mejora continua de la funcionalidad y usabilidad del sitio. La participación activa de los stakeholders (cliente y usuarios finales) en el proyecto ha facilitado una valiosa retroalimentación y perspectivas diversas. Gracias a ello se ha logrado identificar de forma temprana posibles problemas y áreas de mejora, para realizar ajustes antes del lanzamiento final.

En general, se destaca el enfoque ágil y colaborativo del equipo de desarrollo, el cual ha permitido adaptarse a los cambios y necesidades del proyecto. Con la combinación de las tecnologías y estrategias adoptadas, se anticipa que el sitio web de la Iglesia Camino Nuevo alcance su objetivo de brindar una experiencia enriquecedora y relevante a sus feligreses en los idiomas que predominan entre sus miembros, al mismo tiempo que facilita la recepción de donaciones de manera rápida y segura.

Recomendaciones

Es fundamental adecuar continuamente el sitio a las necesidades y expectativas de los usuarios, asegurando así una experiencia óptima y satisfactoria para los feligreses. Para mantenerse actualizado y competitivo en el entorno digital en constante evolución, se

recomienda continuar explorando tecnologías emergentes y tendencias del desarrollo web, incorporando aquellas que resulten relevantes y enriquecedoras para la comunidad.

Para una gestión efectiva del desarrollo, se aconseja seguir un enfoque ágil y colaborativo, utilizando la metodología Scrum para una planificación efectiva y una adaptación ágil a los cambios y requerimientos del proyecto. De esta manera, se puede mantener el product backlog actualizado y priorizar adecuadamente los nuevos requisitos.

Es fundamental seleccionar un proveedor de servicios en la nube confiable y con experiencia en alojamiento de sitios web para asegurar una infraestructura escalable, segura y confiable para el sitio de la iglesia. La implementación de medidas de seguridad sólidas es crucial para salvaguardar la integridad y privacidad de los datos y la información de los usuarios.

Se recomienda prestar una atención constante a las necesidades del usuario y llevar a cabo pruebas de aceptación de usuario. Estas pruebas son esenciales para garantizar la satisfacción y usabilidad del sitio web de la Iglesia Camino Nuevo.

Trabajos Futuros

Para futuros desarrollos del sistema, se sugiere enfocarse en mejorar la experiencia de usuario mediante la implementación de medidas de personalización en el sitio web. Esto se logrará al ofrecer notificaciones individualizadas sobre eventos y servicios relevantes, así como proporcionar una sección de recomendaciones basadas en los intereses específicos de los feligreses.

Se recomienda, además, fortalecer la presencia en redes sociales mediante una integración más sólida, permitiendo a los usuarios compartir contenido del sitio directamente en sus perfiles. Adicionalmente, es crucial ampliar el soporte de idiomas para alcanzar una

audiencia más diversa y garantizar la accesibilidad web, en especial para personas con discapacidades visuales.

Otras mejoras sugeridas incluyen el adoptar un enfoque basado en análisis de datos para tomar decisiones informadas. En términos de seguridad y privacidad, es imperativo fortalecer las medidas para proteger los datos personales y las donaciones de los usuarios.

Estas mejoras representan un paso significativo para la Iglesia Camino Nuevo al consolidar su presencia digital, satisfacer las necesidades de su audiencia y mantenerse a la vanguardia tecnológica. De este modo, podrán expandir su alcance y fortalecer su impacto en la comunidad de una manera más efectiva y relevante.

Referencias

Apache Software Foundation. (2023). *Welcome! - The Apache HTTP Server Project*.

<https://httpd.apache.org/>

Aranda Balaguera, J. S. (2018). *Fortalecimiento del frontend y backend del sitio web*

[www.vendetucarroya.com.co](http://repository.udistrital.edu.co/handle/11349/13876). <http://repository.udistrital.edu.co/handle/11349/13876>

Ark, B. van. (2011). *Un mundo conectado: Las TIC transforman sociedades, culturas y economías*. Fundación Telefónica.

AWS. (s. f.). *¿Qué es un IDE? - Explicación de los entornos de desarrollo integrado - AWS*.

Amazon Web Services, Inc. Recuperado 26 de abril de 2023, de

<https://aws.amazon.com/es/what-is/ide/>

AWS. (2023a). *AWS | Elastic compute cloud (EC2) de capacidad modificable en la nube*.

Amazon Web Services, Inc. <https://aws.amazon.com/es/ec2/>

AWS. (2023b). *Bases de datos no relacionales | Bases de datos de gráficos | AWS*. Amazon

Web Services, Inc. <https://aws.amazon.com/es/nosql/>

- AWS. (2023c). *MongoDB Atlas en AWS - Solución de socio*. Amazon Web Services, Inc.
<https://aws.amazon.com/es/solutions/partners/mongodb-atlas/>
- AWS. (2023d). *¿Qué es una aplicación web? - Explicación de las aplicaciones web - AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/web-application/>
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide*.
- Burgess, S. M., & Maas, E. M. van der. (2010). *The New International Dictionary of Pentecostal and Charismatic Movements: Revised and Expanded Edition*. Zondervan.
- Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2011). *Cloud Computing: Principles and Paradigms*. <https://www.amazon.com/Cloud-Computing-Principles-Rajkumar-Buyya/dp/0470887990>
- Chacon, S., & Straub, B. (2014). *Pro Git*.
- Crawford, R. (2003). *What is Religion?* Routledge.
- Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*. José Vittone — Javier Cuello.
- Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository. *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 1277-1286.
<https://doi.org/10.1145/2145204.2145396>
- Dawoud, W., Takouna, I., & Meinel, C. (2010). *Infrastructure as a service security: Challenges and solutions*. 1-8.
- DB-Engines Ranking. (2023). *DB-Engines Ranking*. DB-Engines. <https://db-engines.com/en/ranking>
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*.
- Durkheim, É. (1982). *Las formas elementales de la vida religiosa*. Akal.
- Eliade, M. (1987). *The Sacred and The Profane: The Nature of Religion* (W. R. Trask, Trad.).
- Express.js. (2023). *Express.js*. Express. <https://expressjs.com/>

- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* [Phd]. University of California, Irvine.
- Flanagan, D. (2011). *JavaScript: The Definitive Guide: Activate Your Web Pages*.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J., & Booch, G. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*.
- GoDaddy. (s. f.). *Acerca de nosotros*. GoDaddy Inc. - Acerca de nosotros. Recuperado 25 de abril de 2023, de <https://aboutus.godaddy.net/about-us/overview/default.aspx>
- Google Cloud. (2023). *MongoDB Atlas*. Google Cloud. <https://cloud.google.com/mongodb?hl=es>
- Goscinski, A., & Brock, M. (2010). Toward dynamic and attribute based publication, discovery and selection for cloud computing. *Future Generation Computer Systems*, 26(7), 947-970. <https://doi.org/10.1016/j.future.2010.03.009>
- Gyorödi, C., Gyorödi, R., & Sotoc, R. (2015). A Comparative Study of Relational and Non-Relational Database Models in a Web- Based Application. *International Journal of Advanced Computer Science and Applications(ijacsa)*, 6(11). <http://dx.doi.org/10.14569/IJACSA.2015.061111>
- Jensen, J. S. (2019). *What Is Religion?* Routledge.
- Jiménez, J. (2019). *Análisis de las principales herramientas de gestión de proyectos y su adecuación a los proyectos de Tecnologías de la Información y Comunicación* [BachelorThesis, Universitat Oberta de Catalunya]. <https://openaccess.uoc.edu/bitstream/10609/96349/6/jfjimenezTFG0619memoria.pdf>
- Kaluža, M., Kalanj, M., & Vukelić, B. (2019). A COMPARISON OF BACK-END FRAMEWORKS FOR WEB APPLICATION DEVELOPMENT. *Zbornik Veleučilišta u Rijeci*, 7(1), 317-332. <https://doi.org/10.31784/zvr.7.1.10>
- Kgatle, M. S. (2020). Religious live-streaming in response to coronavirus disease 2019 pandemic and the subsequent lockdown in South Africa. *Verbum et Ecclesia*, 41(1), Article 1. <https://doi.org/10.4102/ve.v41i1.2120>

- Kumar, S. (2019). A REVIEW ON CLIENT-SERVER BASED APPLICATIONS AND RESEARCH OPPORTUNITY. *International Journal of Scientific Research*, 10, 33857-33862.
<https://doi.org/10.24327/ijrsr.2019.1007.3768>
- Lamothe, M., Guéhéneuc, Y.-G., & Shang, W. (2021). A Systematic Review of API Evolution Literature. *ACM Computing Surveys*, 54(8), 171:1-171:36.
<https://doi.org/10.1145/3470133>
- Laravel. (2023). *Laravel—The PHP Framework For Web Artisans*. <https://laravel.com/>
- Laurie, B., & Laurie, P. (2002). *Apache: The Definitive Guide*.
- Matthews, M. (2023, febrero 20). The Defining Church Tech Trend of 2023 Will Be Refinement. *Pushpay*. <https://pushpay.com/blog/the-defining-church-tech-trend-of-2023-will-be-refinement/>
- MDN. (2023, marzo 29). *Introduction—JavaScript | MDN*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing* (NIST Special Publication (SP) 800-145). National Institute of Standards and Technology.
<https://doi.org/10.6028/NIST.SP.800-145>
- Melton, G. (2023). Pentecostalism. En *Encyclopedia Britannica*.
<https://www.britannica.com/topic/Pentecostalism>
- Meyer, E. A. (2006). *CSS: The Definitive Guide: The Definitive Guide*.
- Microsoft. (2021). *Visual Studio Code*. Documentation for Visual Studio Code.
<https://code.visualstudio.com/docs>
- MongoDB. (s. f.). *MongoDB Atlas | Multi-cloud Developer Data Platform*. MongoDB.
 Recuperado 25 de abril de 2023, de <https://www.mongodb.com/atlas>
- Montalvo, K. (2021). *Aplicación para la optimización del monitoreo y control del entrenamiento de deportistas. Caso de estudio: Centro de Especialización Deportiva GoSport*

- [BachelorThesis, Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Sistemas e Informática]. <http://repositorio.espe.edu.ec/jspui/handle/21000/25084>
- Mozilla. (2022a, noviembre 30). *MDN Web Docs*. REST - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web | MDN. <https://developer.mozilla.org/es/docs/Glossary/REST>
- Mozilla. (2022b, diciembre 5). *MDN Web Docs*. Trabajando con JSON - Aprende sobre desarrollo web | MDN. <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>
- Node.js. (2023). *Node.js*. Acerca de Node.js. <https://nodejs.org/es/about>
- Nortey, J., & Rotolo, M. (2023, marzo 28). How the Pandemic Has Affected Attendance at U.S. Religious Services. *Pew Research Center's Religion & Public Life Project*. <https://www.pewresearch.org/religion/2023/03/28/how-the-pandemic-has-affected-attendance-at-u-s-religious-services/>
- Noss, D. S., & Grangaard, B. R. (2017). *A History of the World's Religions* (14.^a ed.). Routledge. <https://doi.org/10.4324/9781315097886>
- Pérez Ibarra, S. G., Quispe, J. R., Mullicundo, F. F., & Lamas, D. A. (2021). *Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd*. XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja). <http://sedici.unlp.edu.ar/handle/10915/120476>
- Petrosyan, A. (2023, febrero 23). *Number of internet users worldwide 2022*. Statista. <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>
- Pew Research Center, N. (2015). *America's Changing Religious Landscape* (p. 4). <https://www.pewresearch.org/religion/2015/05/12/americas-changing-religious-landscape/>
- Postman. (s. f.). *Postman*. Postman API Platform. Recuperado 25 de abril de 2023, de <https://www.postman.com>

- Pressman, R. S., & Maxim, B. (2014). *Software Engineering: A Practitioner's Approach*.
- ¿Qué es AWS? (2023). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is-aws/>
- Rankin, K., & Hill, B. M. (2013). *The Official Ubuntu Server Book*.
- Rathinam, S. (2023). Analysis and Comparison of Different Frontend Frameworks. En S. Prabhu, S. R. Pokhrel, & G. Li (Eds.), *Applications and Techniques in Information Security* (pp. 243-257). Springer Nature. https://doi.org/10.1007/978-981-99-2264-2_19
- Rathore, B., Saxena, S., Gupta, M., Bhatnagar, S., & Kumar, R. (2017). *Client Server Technology* (2017.^a ed.). Laxmi Publications. <https://dde-ac.in/Books/C132.pdf>
- Rea Chamorro, J. A. (2022). *Desarrollo de sistema web de un ecommerce con facturación electrónica: Desarrollo del Back-End*. [BachelorThesis, Quito : EPN, 2022.]. <http://bibdigital.epn.edu.ec/handle/15000/23101>
- RedHat. (2019, enero 8). ¿Qué es un IDE? <https://www.redhat.com/en/topics/middleware/what-is-ide>
- Richardson, L., Ruby, S., & Hansson, D. H. (2007). *RESTful Web Services*.
- Sadalage, P. (2014, octubre 2). Thoughtworks [Data science and analytics]. *NoSQL Databases: An Overview*. <https://www.thoughtworks.com/insights/blog/nosql-databases-overview>
- Saha, S. (2022). *Church Management Software Market* (Technology 14861; p. 299). Future Market Insights. <https://www.futuremarketinsights.com/reports/church-management-software-market>
- SBC CALLED. (2021, junio 12). *Entendiendo el Llamado al Ministerio Pastoral—Estudiantes Ministeriales SBC*. <https://sbccalled.com/the-call-to-ministry/>
- Schwaber, K., & Sutherland, J. (2017). *La Guía Definitiva de Scrum: Las Reglas del Juego*. <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-SouthAmerican.pdf>
- Smith, H. (1991). *The World's Religions: Our Great Wisdom Traditions*. <https://www.amazon.com/Worlds-Religions-Great-Wisdom-Traditions/dp/0062508113>

- Smith, H., & Sloan, S. (2017). *The World's Religions*.
- Sommerville, I. (2015). *Software Engineering*.
- Sosinsky, B. (2010). *Cloud Computing Bible* (1.^a ed.). <https://www.amazon.com/Cloud-Computing-Bible-Barrie-Sosinsky/dp/0470903562>
- Stack Overflow. (2023). *Stack Overflow Trends*. <https://insights.stackoverflow.com/trends?tags=>
- State of JavaScript. (2023). *State of JavaScript 2022: Front-end Frameworks*.
<https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/>
- The Foursquare Church*. (2023, marzo 21). The Foursquare Church.
<https://www.foursquare.org/>
- Trio. (2023). *15 Companies That Use Node.js in 2023 Successfully | Trio Developers*.
<https://www.trio.dev/blog/companies-use-node-js>
- Ubuntu. (s. f.). *Ubuntu Server—For scale out workloads*. Ubuntu. Recuperado 17 de mayo de 2023, de <https://ubuntu.com/server>
- Visual Studio Code Introductory Videos*. (2021).
<https://code.visualstudio.com/docs/getstarted/introvideos>
- Vivah, L. (2019, septiembre 12). THE BEGINNER'S GUIDE: Understanding Node.js & Express.js fundamentals. *Medium*. <https://medium.com/@LindaVivah/the-beginners-guide-understanding-node-js-express-js-fundamentals-e15493462be1>
- Vue.js. (2023). *Vue.js*. Introduction - Vue.js. <https://vuejs.org/guide/introduction.html>
- W3Schools. (s. f.). *CSS Tutorial*. Recuperado 2 de junio de 2023, de
<https://www.w3schools.com/css/>
- W3Techs. (2023, junio). *Usage Statistics of JavaScript as Client-side Programming Language on Websites*. <https://w3techs.com/technologies/details/cp-javascript>
- Walsh, M. (2016). *Roman Catholicism: The Basics*. Routledge.

What is JIRA?: How to Use Jira Testing Software Tool | Simplilearn. (2022, mayo 2).

Simplilearn.com. <https://www.simplilearn.com/tutorials/jira/what-is-jira-and-how-to-use-jira-testing-software>

WHATWG. (s. f.). *HTML Living Standard* [HTML Standard]. HTML. Recuperado 25 de abril de 2023, de <https://html.spec.whatwg.org/#introduction>

Williams, M. (2023, enero 27). *GoDaddy review*. Techradar.

<https://www.techradar.com/reviews/godaddy>

Wormald, B. (2015, abril 2). The Future of World Religions: Population Growth Projections, 2010-2050. *Pew Research Center's Religion & Public Life Project*.

<https://www.pewresearch.org/religion/2015/04/02/religious-projections-2010-2050/>

Wunsch, L. P., Freitas, E. D. S. de, & Nikolay, J. R. (2022). Chatbot: Comunicación digital y religiosidad tras la pandemia en Latinoamérica. *Universitas*, 37, Article 37.

<https://doi.org/10.17163/uni.n37.2022.04>

Yadav, S. C. (2009). *Introduction to Client/Server Computing*.

Apéndices