

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UN SISTEMA PARA LA INTERACCIÓN DE LOS MIEMBROS DE LA COMUNIDAD DE LA IGLESIA IFGF

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

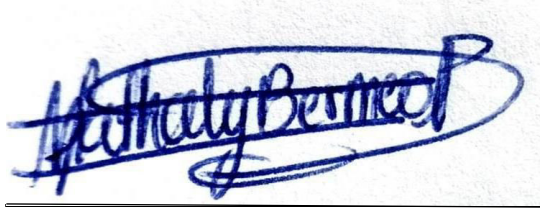
NATHALY LISSETTE BERMEO PANCHI

DIRECTOR: ING. BYRON GUSTAVO LOARTE CAJAMARCA, MSc.

DMQ, febrero 2022

CERTIFICACIONES

Yo, Nathaly Lissette Bermeo Panchi declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Nathaly Lissette Bermeo Panchi

nathaly.bermeo@epn.edu.ec

nathalybermeo97@outlook.es

Certifico que el presente trabajo de integración curricular fue desarrollado por Nathaly Lissette Bermeo Panchi, bajo mi supervisión.



Ing. Byron Loarte, MSc.

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

NATHALY LISSETTE BERMEO PANCHI

DEDICATORIA

Dedico este trabajo con todo mi corazón a Dios, por ayudarme con la fortaleza necesaria para poder culminarlo, a mi familia por brindarme un ejemplo de superación, honestidad, humildad y responsabilidad, por su perseverancia y dedicación al incentivar me cada día de mi carrera universitaria.

NATHALY LISSETTE BERMEO PANCHI

AGRADECIMIENTO

Agradezco a mis padres, hermanos y tías, quienes siempre han estado presentes en mi vida universitaria con su confianza y fortaleza en todo momento, por estar en mi vida dándome su apoyo incondicional.

Un agradecimiento a la Escuela Politécnica Nacional, por haberme permitido ser parte de ella y estudiar una carrera universitaria, a los profesores de la Escuela de Formación de Tecnólogos, quienes me han aportado conocimientos en toda mi etapa universitaria para culminar de manera exitosa este trabajo de titulación.

NATHALY LISSETTE BERMEO PANCHI

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
2 METODOLOGÍA	6
2.1 Metodología de Desarrollo	6
Roles.....	6
Artefactos.....	7
2.2 Diseño de interfaces	10
Herramienta utilizada para el diseño.....	10
2.3 Diseño de la arquitectura	11
Patrón arquitectónico	11
2.4 Herramientas de desarrollo.....	12
3 RESULTADOS.....	14
3.1 <i>Sprint</i> 0. Configuración del ambiente de desarrollo	14
Definición de requerimientos	14
Estructura del proyecto frontend	16
Roles de usuario	16
3.2 <i>Sprint</i> 1. Resultado de las interfaces para los usuarios con perfil administrador e invitado.....	17
Consumir un <i>endpoint</i> para visualizar la información general de la Iglesia.....	17
Consumir <i>endpoints</i> para gestionar la información general de la Iglesia	18
Consumir <i>endpoints</i> para el registro y autenticación de usuarios.....	19
Consumir <i>endpoints</i> para gestionar eventos	20

Consumir <i>endpoint</i> para gestionar donaciones	20
Consumir <i>endpoint</i> para visualizar y aprobar donaciones	21
Consumir <i>endpoints</i> para gestionar vídeos.....	22
Consumir <i>endpoints</i> para gestionar cuestionarios.....	22
3.3 <i>Sprint 2</i>. Resultado de las interfaces para el usuario miembro de la Iglesia ..	23
Consumir <i>endpoints</i> para interactuar con donaciones	23
Consumir <i>endpoints</i> para interactuar con eventos.....	25
Consumir un <i>endpoint</i> para interactuar con cuestionarios	25
Consumir un <i>endpoint</i> para visualizar vídeos	26
3.4 <i>Sprint 3</i>. Pruebas del <i>frontend</i>	27
Resultados de la ejecución de las pruebas unitarias.....	27
Resultados de la ejecución de las pruebas de compatibilidad.....	28
Resultado de la ejecución de las pruebas de aceptación.....	29
3.5 <i>Sprint 4</i>. Despliegue del <i>frontend</i>.....	30
Despliegue del sistema <i>web</i> en <i>Vercel</i>	30
4 Conclusiones	32
5 Recomendaciones	33
6 REFERENCIAS BIBLIOGRÁFICAS.....	34
7 ANEXOS	37
ANEXO I.....	38
ANEXO II.....	39
ANEXO III.....	81
ANEXO IV	82

RESUMEN

En la actualidad muchos gobiernos alrededor del mundo han implementado una serie de medidas para detener el avance del COVID-19 y entre ellas está el de prohibir las celebraciones religiosas de manera presencial, esto sin duda ha despertado un malestar entre los feligreses. Además, los pastores de varias Iglesias han optado por seguir cada una de las restricciones de no tener cultos presenciales ya que las restricciones impuestas por gobiernos locales impiden reunirse físicamente por lo que las personas han optado por usar la tecnología como un medio para estar en contacto con Dios.

Con el fin de apoyar a la comunidad religiosa, en el presente proyecto se ha desarrollado un *frontend* para la Iglesia Fraternidad Internacional del Evangelio Completo GISI (IFGF) de Quito – Ecuador, permitiendo la interacción de sus miembros con la Iglesia por medio de vídeos, eventos, donaciones, noticias, entre otros. Logrando de esta manera crear un espacio virtual para mejorar la comunicación e interacción con todos los miembros de la Iglesia por medio de la tecnología.

El presente trabajo de integración curricular se encuentra estructurado de esta manera: en la primera sección se detalla sobre los antecedentes, objetivos, alcance del proyecto y el correspondiente marco teórico. En la segunda sección se especifica como se ha realizado la implementación de la metodología *Scrum*, prototipos y herramientas en el desarrollo del *frontend*. En la tercera sección se especifica las actividades que se han desarrollado por cada iteración y el resultado que se ha obtenido por cada *Sprint*. Finalmente, en la cuarta sección se especifican las conclusiones y recomendaciones que se han obtenido.

PALABRAS CLAVE: *Scrum, frontend, React, Next*, Iglesia, Interacción.

ABSTRACT

Currently, many governments around the world have implemented a series of measures to stop the advance of COVID-19 and among them is to prohibit religious celebrations in person, this has undoubtedly aroused discomfort among parishioners. In addition, the pastors of various churches have chosen to follow each of the restrictions of not having face-to-face services since the restrictions imposed by local governments prevent physical meetings, so people have chosen to use technology as a means to be in contact. with God.

In order to support the religious community, in this project a frontend has been developed for the International Fraternity Church of the Full Gospel GSI (IFGF) of Quito - Ecuador, allowing the interaction of its members with the Church through videos, events, donations, news, among others. In this way, creating a virtual space to improve communication and interaction with all members of the Church through technology.

This curricular integration work is structured in this way: the first section details the background, objectives, scope of the project and the corresponding theoretical framework. In the second specific section, the implementation of the Scrum methodology, prototypes and tools in the development of the frontend has been carried out. In the third specific section are the activities that have been developed for each iteration and the result that has been obtained for each Sprint. Finally, the fourth section specifies the conclusions and recommendations that have been obtained.

KEYWORDS: *Scrum, frontend, React, Next, Church, Interaction.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El COVID-19 ha tenido un fuerte impacto sobre las condiciones de vida de los ciudadanos en el Ecuador [1]. Actualmente hay más de 457,489 casos confirmados con pruebas PCR y 21,545 personas fallecidas en el Ecuador a causa de esta pandemia [2]. No obstante, la emergencia sanitaria está poniendo a prueba la resistencia física, mental y social de muchas naciones. Debido a esto, muchas personas buscan consuelo y tranquilidad por lo que sienten la necesidad de buscar un camino hacia Dios por medio de la Iglesia [3].

El Comité de Operaciones de Emergencia (COE) informó que las Iglesias pueden realizar reuniones o misas manteniendo el 30% de aforo permitido [4]. Sin embargo, esta pandemia a sometido a que gran parte de las personas por temor a contagiarse por la aparición de nuevas variantes prefieran las redes sociales y el ciberespacio como estrategia para mantener una comunicación más directa con todos los miembros de la Iglesia [5].

En la actualidad la interacción entre personas, aplicaciones móviles y sistemas *web* es cada vez mayor, hoy en día estos sistemas y aplicaciones ofrecen herramientas que ayudan a resolver diferentes necesidades y problemas [6]. Sin embargo, el crear sistemas y programas facilitan la experiencia del usuario, la automatización de tareas, la comunicación e integración con *backends*, entre otras son actividades cruciales que desempeñan los desarrolladores. En ese sentido, el *frontend* es la parte visual de un sistema *web* en el que interactúan los usuarios finales, comúnmente llamada aplicaciones del lado del cliente [7], básicamente es todo lo que se observa en la pantalla de un navegador *web* cuando se accede al mismo, por ejemplo: tipo de letra, colores, adaptación a distintas pantallas, efectos del ratón, movimientos, efectos visuales y otros elementos que permiten mejorar la experiencia del usuario [7]. Es por esta razón, que en la actualidad el *frontend* y *backend* tienen cualidades únicas y distintivas ya que interactúan entre sí para asegurar la funcionalidad completa de los sistemas *web* [8].

Ante la problemática expuesta y con el fin de apoyar e informar a la comunidad religiosa en modalidad virtual, en el presente proyecto se ha desarrollado un *frontend* para la Iglesia Fraternidad Internacional del Evangelio Completo GISI (IFGF) de la ciudad de Quito - Ecuador con el propósito de que todos los feligreses interactúen con los miembros de la Iglesia por medio de vídeos, eventos, artículos, comentarios, noticias, entre otros. El trabajo de integración curricular apoya a la creación de un espacio virtual para mejorar la comunicación e interacción con todos los miembros de la Iglesia por medio de la tecnología.

1.1 Objetivo general

Desarrollar un sistema para la interacción de los miembros de la comunidad de la Iglesia (IFGF).

1.2 Objetivos específicos

1. Determinar los requerimientos funcionales y no funcionales para el *frontend*.
2. Diseñar los prototipos en base a los requerimientos.
3. Codificar el *frontend* en base al lenguaje de programación y requerimientos.
4. Verificar el funcionamiento del *frontend* y despliegue a producción.

1.3 Alcance

Hoy en día las herramientas tecnológicas han transformado totalmente la forma en como las personas se comunican entre sí, logrando tener una mayor importancia, ya que permite ahorrar tiempo al momento de realizar actividades cotidianas [9], es por ello que los sistemas *web* han ganado una mayor aceptación entre los ciudadanos, ya que son espacios digitales en *Internet* que contienen información expuesta para todo el mundo y en donde los usuarios pueden interactuar con todo el contenido de una manera mucho más fácil y entendible, todo esto a través de un *frontend* que junto con el *backend* dan funcionalidad y lógica completa a lo que es un sistema *web* [10].

Por lo antes citado, se ha desarrollado un *frontend* el cual permite por una parte mantener una comunicación directa con un *backend* a través de una serie de API's de tipo *RESTful* y, por otra parte, proporcionar un medio de comunicación virtual e informativo a toda la comunidad de la Iglesia Cristiana (IFGF) de la ciudad de Quito. Utilizando para ello, una serie de herramientas de desarrollo actuales y modernas que estén alineadas específicamente con desarrollo del lado del cliente.

Por último, el desarrollo del *frontend* muestra toda la funcionalidad y el consumo del *backend* por medio de los siguientes *endpoints*:

Información general

- Consumir varios *endpoints* para presentar toda la información de la Iglesia (noticias, mensajes bíblicos y álbum de fotos).
- Consumir varios *endpoints* para gestionar la información general (noticias, mensajes bíblicos y álbum de fotos).

Donaciones

- Consumir un *endpoint* para registrar donación.
- Consumir un *endpoint* para ver el registro de donaciones.

Eventos

- Consumir varios *endpoints* para gestionar eventos.
- Consumir un *endpoint* para inscribirse en un evento.
- Consumir un *endpoint* protegido para visualizar las personas inscritas en los eventos.

Vídeos

- Consumir varios *endpoints* para gestionar vídeos.
- Consumir un *endpoint* para visualizar los vídeos publicados.

Juegos (Cuestionarios)

- Consumir un *endpoint* para registrar cuestionarios.
- Consumir un *endpoint* para visualizar cuestionarios.
- Consumir un *endpoint* para registrar datos del jugador.
- Consumir un *endpoint* para visualizar los puntajes del juego.

1.4 Marco Teórico

En la actualidad, la propagación del COVID-19 en espacios públicos es un riesgo para la humanidad por lo que las autoridades han tomado medidas de seguridad como la designación de un determinado aforo para los diferentes sitios públicos. Por ello, las Iglesias han recurrido a una modalidad virtual debido a la suspensión de las reuniones tradicionales. Por tal motivo, es importante hoy en día la integración sistemas *web* y aplicaciones móviles en la vida cotidiana de la humanidad ya que es imprescindible su uso para comunicarse, automatizar tareas, compartir información, entre otras actividades que ayuden a reducir los contagios por la aparición de nuevas variantes ya que la pandemia aún no ha terminado [11].

Una metodología ágil es un conjunto de técnicas que se emplean al momento de organizar un equipo de trabajo para desarrollar un producto. Por otra parte, las metodologías han ido evolucionando con el paso del tiempo ya que es una base importante al momento de desarrollar cualquier tipo de producto *software* de una manera eficaz, óptima y con un buen producto de calidad para los usuarios finales [12]. Otra de las grandes ventajas que poseen

estas metodologías es que permiten responder rápidamente a las necesidades del cliente siendo capaces de cambiar la dirección del desarrollo del proyecto dependiendo de la situación o necesidades del cliente, con ello se puede decir que el objetivo principal es siempre maximizar el valor de cliente y minimizar el riesgo de crear productos no satisfactorios [13].

En la actualidad, el *frontend* comúnmente llamado “aplicación del lado del cliente” es la parte visual de un sistema *web*, es todo lo que visualizan e interactúan los usuarios por lo que se maneja estándares de usabilidad y estética para un mejor desempeño. Por lo general, el *frontend* comprende de una variedad de componentes como menús desplegables, iconos, colores, elementos gráficos entre otros. Además, en el desarrollo del código se utilizan lenguajes esenciales para el diseño y la interacción como CSS, HTML, JavaScript y una serie de Librerías o *Frameworks* como *Angular*, *React* o *Vue* [14].

React es una biblioteca de JavaScript para la elaboración de interfaces para el usuario, ayuda a crear interfaces interactivas de una forma sencilla, renderiza y actualiza sus componentes de manera eficiente cuando se maneja el proceso de cambio de datos. Además, *React* interactúa con otras bibliotecas y *Frameworks* para llevar a cabo el desarrollo completo de un sistema [15].

El *Framework Next.js*, que utiliza como base *React* permite una serie de opciones como: renderizado de páginas directamente en el servidor, crea páginas dinámicas, creación de una estructura de directorios y archivos, entre otras. Lo que en la actualidad permite que sea una buena opción al momento de desarrollar aplicaciones del lado del cliente de una forma rápida y elegante [16]. Al mismo tiempo, tiene múltiples funcionalidades que garantiza la creación de un proyecto en unos sencillos pasos [17].

Para el desarrollo completo de un sistema *web* es indispensable la implementación de un *backend* ya que es el encargado del manejo de solicitudes e interacciones del usuario, el *frontend* se comunica mediante solicitudes HTTP hacia el *backend* con el propósito de transferir datos ingresados o recuperar información de la base de datos, es decir, el *backend* se encarga del manejo de los datos y de la lógica de todo un sistema *web* [18].

Figma es una herramienta para el diseño de interfaces tanto para sistemas *web* como para aplicaciones móviles, esta herramienta preferida por los desarrolladores permite la creación de prototipos, obteniendo de esta manera resultados convincentes en el diseño de las diferentes interfaces, además, es una herramienta colaborativa que permite trabajar en tiempo real con todo el equipo de desarrollado [19].

Una etapa de pruebas es indispensable para cualquier producto de *software* que se desarrolle ya que esto permite la correcta funcionalidad de módulos, perfiles, navegabilidad, acceso a información, entre otros aspectos. Garantizando de esta manera la aprobación por parte del dueño del producto y un correcto despliegue a producción. Por otra parte, cada una de las pruebas deben cumplir con ciertas características fundamentales y se puede elegir la que más se acople al proyecto como, por ejemplo: Unitarias, Compatibilidad, Aceptación, etc. [20].

2 METODOLOGÍA

Un estudio de casos es una investigación que ha sido detallada de un tema en específico, el objetivo de un estudio es obtener un conocimiento concreto sobre un tema a profundidad a través de una recopilación de datos, ya sean por medio de cuestionarios, reuniones, entrevistas, entre otros. Logrando de esta manera obtener una serie de resultados favorables para un correcto análisis [21].

Debido a esto, el presente trabajo de integración curricular mantiene un estudio de casos ya que se basa en investigaciones sobre un tema en específico para desarrollar el *frontend*, por medio de entrevistas con el cliente e investigaciones a través de otros sitios *web* en donde se ha visualizado las necesidades que tienen las personas para acudir a espacios públicos a causa de la pandemia. Todo esto, permite llevar a cabo el desarrollo de un *frontend* para para que los miembros Iglesia IFGF puedan interactuar y comunicarse con toda la comunidad por medio de la tecnología.

2.1 Metodología de Desarrollo

Las metodologías ágiles buscan proporcionar una mejor calidad y garantizar un excelente funcionamiento en los proyectos de *software* mediante lapsos cortos de tiempo con el fin de cumplir las expectativas del cliente final. Es por esta razón, que estas metodologías desarrollan enfoques importantes por medio del trabajo en equipo para realizar constantes mejoras [22].

Un enfoque ágil permite al equipo de trabajo desarrollar y proporcionar respuestas en espacios de tiempo muy cortos, todo aquello para validar constantemente el desarrollo del proyecto, también es indispensable la supervisión y evaluación en cada iteración del desarrollo de un *software* [23]. En ese sentido, en las siguientes secciones se describe como se ha implementado cada uno de los aspectos de la metodología ágil *Scrum* en el desarrollo del *frontend*.

Roles

Los roles son capaces de manejar responsabilidades, actividades y cuentas que rendir dentro de un grupo de trabajo [24]. Todos estos roles permiten el cumplimiento de un producto final de calidad. En ese sentido, se presenta los roles que se han establecido para el desarrollo del *frontend*.

Product Owner

Este rol pertenece a la persona encargada de validar el producto final, es el altavoz de las peticiones y requerimientos. Además, es fundamental este rol ya que es capaz de tomar cualquier decisión antes y durante referente al proyecto de *software* [24]. En ese aspecto, este rol se ha establecido a un grupo de personas capaces de interactuar con el contenido del *frontend* como se evidencia en la **TABLA I**.

Scrum Master

Este rol pertenece al encargado de administrar el proceso *Scrum* y solventar cualquier tipo de impedimentos que afectan la entrega del producto. Es por esta razón, que la persona asignada en la **TABLA I** es el responsable de velar que el equipo *Scrum* salga adelante por medio de buenas prácticas y reuniones semanales [24].

Development Team

Este rol está compuesto por personas que se encargan de desarrollar e implementar el producto *software* en base a los requisitos del proyecto. Además, estas personas se organizan para entregar un producto de *software* bien elaborado [24]. Muestra de ello, en la **TABLA I** se presenta a la persona asignada para este rol.

TABLA I: Asignación de roles

ROLES	NOMBRES
<i>Product Owner</i>	Pastor Walter Hidalgo
<i>Scrum Master</i>	Ing. Byron Loarte, MSc.
<i>Development Team</i>	Nathaly Bermeo

Artefactos

Los artefactos en *Scrum* están orientados a garantizar el 100% del cumplimiento de un proyecto *software* mediante la recolección de información. No obstante, permiten determinar de forma detallada el propósito del producto final y cada uno de los *Sprints* [25].

Recopilación de Requerimientos

Los requerimientos para el *frontend* se han obtenido a través de una serie de entrevistas con el dueño del producto en donde se ha recopilado toda información necesaria sobre la

estructura, diseño y módulos que dispone el *frontend*. Muestra de ello, la **TABLA II** muestra el formato que se ha utilizado para la elaboración de la Recopilación de los requerimientos. Sin embargo, la tabla completa con toda la información se detalla en el **ANEXO II** del presente documento.

TABLA II: Formato para la Recopilación de Requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DE SISTEMA	ID RR	DESCRIPCIÓN
FRONTEND	RR001	Como usuario invitado, administrador y/o miembro de la Iglesia, necesitan consumir varios <i>endpoints</i> para presentar la información general de la Iglesia.

Historias de Usuario

Son tarjetas con información concreta y detallada de cada requerimiento por parte del dueño del producto hacia una funcionalidad del producto *software* [26]. En ese sentido, las Historias de usuario para el *frontend* se han desarrollado juntamente con la Recopilación de requerimientos, la revisión y aprobación por parte del dueño del producto. Muestra de ello, la **TABLA III** presenta una de las primeras Historia de usuario que se han elaborado, mientras que las demás Historias de usuario se encuentran detalladas en el **ANEXO II** del presente documento.

TABLA III: Historia de usuario 01 - Consumir varios *endpoints* para presentar la información general de la Iglesia

HISTORIA DE USUARIO	
Identificador (ID): HU001	Usuario: Invitado, administrador y miembro de la Iglesia
Nombre Historia: Consumir varios <i>endpoints</i> para presentar la información general de la Iglesia.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Nathaly Bermeo	

<p>Descripción: El usuario invitado, administrador y miembro de la Iglesia necesitan consumir varios <i>endpoints</i> para visualizar la información general de la Iglesia e interactuar con los siguientes módulos:</p> <ul style="list-style-type: none"> • Noticias • Eventos • Donaciones • Juegos (Cuestionarios) • Vídeos
<p>Observación: El usuario invitado, administrador y miembro de la Iglesia pueden consumir varios <i>endpoints</i> del <i>backend</i> dependiendo del inicio de sesión.</p>

Product Backlog

Es una lista que se encuentra totalmente ordenada con toda la información que se requiere para desarrollar un producto *software*. Por otra parte, esta lista permite conocer las necesidades del producto *software* para lograr un mejor desempeño [27]. Es por esta razón, que la **TABLA IV** muestra el formato que se ha utilizado para listar el *Product Backlog*, mientras que la tabla completa se detalla en el **ANEXO II** del presente documento.

TABLA IV: Formato del *Product Backlog*

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID-HU	Historia de Usuario	No. Iteración	Prioridad	Estado
HU001	Consumir varios <i>endpoints</i> para presentar la información general de la Iglesia.	1	Media	Planificado
HU002	Consumir varios <i>endpoints</i> para iniciar y cerrar sesión	1	Alta	Planificado

Sprint Backlog

Es un cuadro de actividades, el cual facilita considerablemente el desarrollo de un producto *software*, así como sus módulos que lo integra. Por otra parte, permite que el desarrollador del proyecto identifique las funcionalidades y priorice las que se necesite entregar antes de tiempo. A continuación, la **TABLA V** se presenta el formato que se ha utilizado en el

desarrollo de este proyecto para listar los 5 *Sprints* que son: Configuración del entorno de desarrollo, diseño e implementación de la interfaz de los usuarios administrador e invitado, diseño e implementación de la interfaz del usuario miembro de la Iglesia, pruebas y despliegue del *frontend*, mientras que la tabla completa con todas las actividades se detalla en el **ANEXO II** del presente documento.

TABLA V: Formato del *Sprint Backlog*

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del ambiente de desarrollo	-----	-----	-----	<ul style="list-style-type: none"> Definición de requerimientos. Estructura del proyecto. 	20 H

2.2 Diseño de interfaces

El diseño de interfaces se elabora con la información de cada requerimiento que se ha obtenido anteriormente, para que el dueño del producto pueda tener una visión mucho más clara del producto *software* que se va a desarrollar en base a cada *Sprint* [28].

Herramienta utilizada para el diseño

Mediante el uso de la herramienta *online Figma*, se han realizado una serie de prototipos, así como sus correspondientes módulos, logrando de esta manera tener una base para la etapa de codificación del *frontend*. Muestra de ello la **Fig. 1** muestra el prototipo con la información general de la Iglesia la cual está dividida en diferentes secciones como: *header*, *navbar*, *cards*, *carousel* y un *footer*. Por último, el resto de los diseños se detallan en el **ANEXO II** del presente documento.



Fig. 1: Prototipo – Información general de la Iglesia

2.3 Diseño de la arquitectura

En la actualidad para cualquier desarrollo de *software* es vital la implementación de una arquitectura *web*, ya que proporciona una clara estructura y compatibilidad con las herramientas de desarrollo [29]. Es por esta razón, que en el desarrollo del *frontend* se ha optado por la integración de un patrón arquitectónico que se describe a continuación.

Patrón arquitectónico

El patrón de arquitectura Modelo Vista Controlador (MVC) es un patrón ampliamente utilizado en la industria del desarrollo, así como en proyectos de *software*, logrando de esta manera que toda la información se divida en tres componentes para una mejor organización de los datos, interfaz y la lógica del producto final [30].

- El componente Modelo gestiona los datos que provienen de la Base de Datos.
- El componente Vista presenta al usuario todo el contenido visual para su interacción.
- El componente Controlador gestiona todas las solicitudes que se realizan, siendo esto el punto de partida para cualquier aplicación.

Por lo antes mencionado, el desarrollo del *frontend* dispone del siguiente patrón arquitectónico como se presenta en la **Fig. 2**, permitiendo de esta manera que los cambios se los realice de una manera mucho más organizada y estructurada. Además, este patrón de arquitectura es altamente compatible con las herramientas que se han seleccionado para la etapa de codificación, pruebas y despliegue [31].

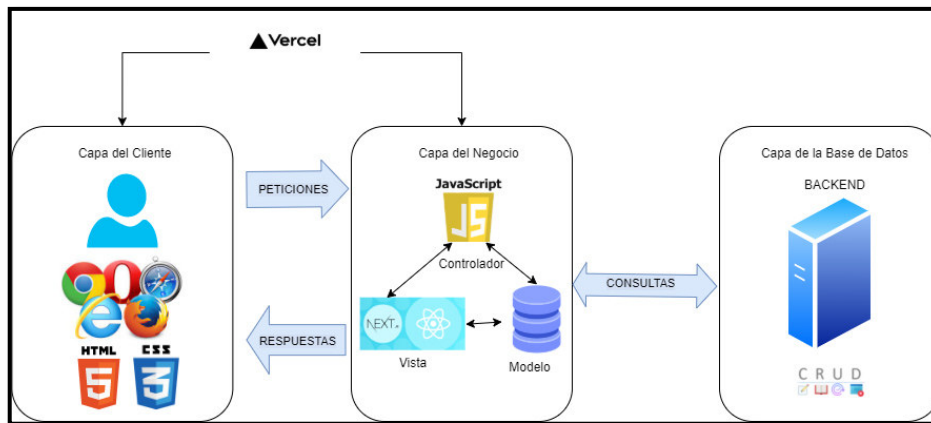


Fig. 2: Patrón arquitectónico – Frontend

2.4 Herramientas de desarrollo

Las herramientas que se han seleccionado para el desarrollo del *frontend* son en base al análisis de requerimientos y el conocimiento del equipo de desarrollo, permitiendo de esta manera tener un desarrollo mucho más ágil y que las mismas sean compatibles con las librerías que se describen en la siguiente sección [32]. Es por esta razón, que la **TABLA VI** presenta la justificación de cada una de las herramientas que se ha utilizado para el desarrollo del *frontend*.

TABLA VI: Herramientas para el desarrollo del *frontend*

HERRAMIENTA	JUSTIFICACIÓN
React	<i>React</i> es una biblioteca de JavaScript desarrollada por Facebook que facilita la creación de aplicaciones <i>web</i> del lado del cliente [33].
Next.js	<i>NextJS</i> es un <i>Framework</i> que permite funcionalidades de aplicaciones <i>web</i> basadas en <i>React</i> [34].
Bootstrap	<i>Bootstrap</i> es un <i>Framework</i> con una serie de componentes visuales que sean adaptables a diferentes tipos de pantalla [35].

Librerías

La **TABLA VII** presenta la descripción de cada una de las librerías que se ha utilizado para el desarrollo del *frontend*.

TABLA VII: Librerías utilizadas en el desarrollo del *frontend*

LIBRERÍA	DESCRIPCIÓN
<i>react-bootstrap</i>	Librería para el diseño de las interfaces del <i>frontend</i> [36].
react-bootstrap-icons	Librería para el manejo de íconos del <i>frontend</i> [37].
<i>react-hook-form</i>	Librería para realizar formularios “no-controlados” [38].
<i>styled-components</i>	Librería para escribir código <i>CSS</i> con estilo de <i>JavaScript</i> [39].
ESLint	Librería que ayuda a mejorar la calidad, legibilidad y mantenibilidad del código [40].

3 RESULTADOS

Desde esta sección en adelante, se detallan cada uno de los resultados que se han obtenido en la implementación de los diferentes módulos y componentes visuales para el *frontend*, así como el resultado de las pruebas y el proceso de despliegue a producción. Sin embargo, cada uno de los resultados se presenta por medio de *Sprints* que han sido previamente planificados y que los mismos se detallan en el **ANEXO II** del presente documento.

3.1 *Sprint* 0. Configuración del ambiente de desarrollo

Las actividades para el *Sprint* 0 tienen que ver con:

- Definición de requerimientos.
- Estructura del proyecto *frontend*.
- Roles de usuario.

Definición de requerimientos

Consumir *endpoints* de información general

El *frontend* a través de varios métodos tiene la posibilidad de consumir varios *endpoints* para ingresar, visualizar, modificar y eliminar la información de la Iglesia por medio del usuario con perfil administrador. No obstante, el usuario con perfil miembro de la Iglesia e invitado pueden visualizar toda la información general de la Iglesia.

Consumir *endpoints* para registro e inicio de sesión

El *frontend* a través de varios métodos tiene la posibilidad de consumir varios *endpoints* para el acceso de usuarios, el usuario con perfil administrador puede iniciar sesión mediante un correo electrónico y contraseña, las mismas que son creadas y otorgadas por el equipo de desarrollo. Por otra parte, el usuario con perfil miembro de la Iglesia tiene acceso a un formulario de registro y autenticación al igual que el usuario invitado.

Consumir *endpoints* para gestionar eventos

El *frontend* a través de varios métodos tiene la posibilidad de consumir varios *endpoints* para registrar, visualizar, modificar y eliminar eventos por medio del usuario con perfil administrador. Además, tiene la posibilidad de visualizar todos los miembros que están inscritos en cada evento. Por otra parte, para el usuario con perfil miembro de la Iglesia puede visualizar e inscribirse a un evento y a su vez tiene la opción de no participar del mismo en el caso de que se requiera.

Consumir *endpoints* para gestionar vídeos

El *frontend* a través de varios métodos tiene la posibilidad de consumir varios *endpoints* para registrar, visualizar, modificar y eliminar vídeos por medio del usuario con perfil administrador. Además, tiene la posibilidad de visualizar todos los miembros que están inscritos en cada evento. Por otra parte, para el usuario con perfil miembro de la Iglesia, puede visualizar todos los vídeos publicados.

Consumir *endpoints* para gestionar donaciones

El *frontend* a través de varios métodos tiene la posibilidad de consumir un *endpoint* para que el usuario con perfil administrador pueda visualizar las donaciones que se han realizado. Mientras que, el usuario con perfil miembro de la Iglesia puede realizar donaciones de dinero, ropa y alimentos haciendo uso de un formulario para registrar el tipo de donación.

Consumir *endpoints* para gestionar cuestionarios

El *frontend* a través de varios métodos tiene la posibilidad de consumir varios *endpoints* para registrar, visualizar, modificar y eliminar cuestionarios por medio del usuario con perfil administrador. Por otra parte, para el usuario con perfil miembro de la Iglesia puede visualizar, responder y obtener un puntaje de los cuestionarios. Por último, la **Fig. 3** presenta las acciones que pueden realizar cada uno de los usuarios dentro del *frontend*.

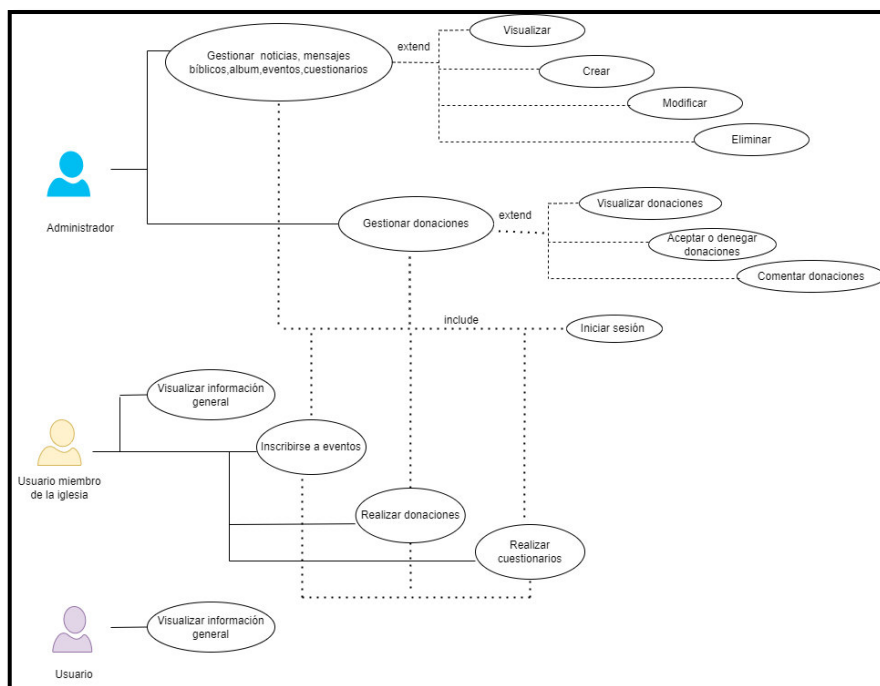


Fig. 3: Usuarios y funcionalidades dentro del *frontend*

Estructura del proyecto frontend

La herramienta *WebStorm* se ha utilizado como entorno de desarrollo para codificar todo el *frontend* y realizar la estructura de directorios y archivos en base al patrón de arquitectura que se ha seleccionado anteriormente, en ese sentido **Fig. 4** ilustra la estructura del proyecto.

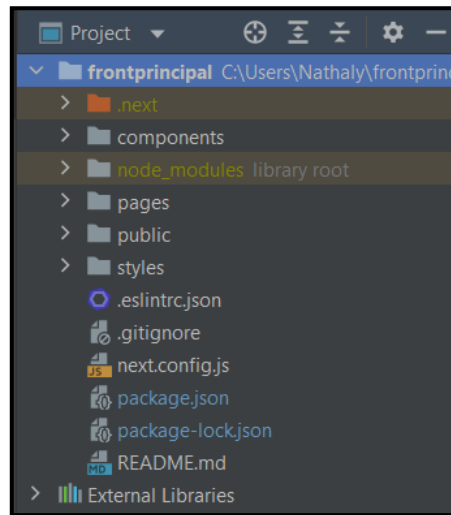


Fig. 4: Estructura del proyecto *frontend*

Roles de usuario

A continuación, la **Fig. 5** ilustra a los usuarios: administrador, miembro de la Iglesia e invitado, los cuales una vez que inicien sesión se presenta los módulos a los que tienen acceso.

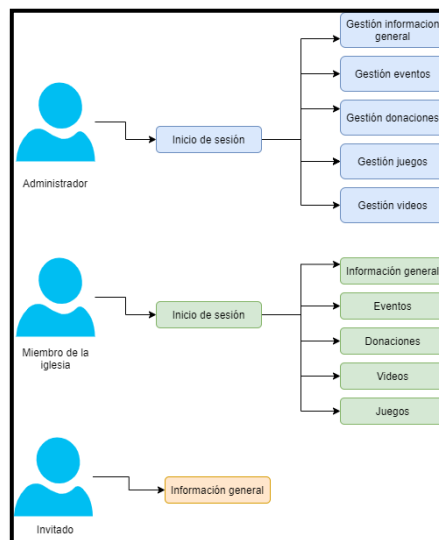


Fig. 5: Usuarios y módulos del *frontend*

3.2 *Sprint* 1. Resultado de las interfaces para los usuarios con perfil administrador e invitado

Las actividades para el *Sprint* 1 tienen que ver con:

- Consumir un *endpoint* para visualizar la información general de la Iglesia.
- Consumir *endpoints* para gestionar la información general de la Iglesia.
- Consumir *endpoints* para el registro y autenticación de usuarios.
- Consumir *endpoints* para gestionar eventos.
- Consumir *endpoint* para gestionar donaciones.
- Consumir *endpoint* para visualizar y aprobar donaciones.
- Consumir *endpoints* para gestionar vídeos.
- Consumir *endpoints* para gestionar cuestionarios.

Consumir un *endpoint* para visualizar la información general de la Iglesia

Toda la información referente a la Iglesia se puede visualizar en la página principal por medio de tarjetas que contienen información sobre noticias, eventos, mensajes bíblicos entre otros. Además, para poder visualizar esta información en el *frontend* se ha creado varios métodos y componentes visuales que permitan consumir una ruta pública de tipo *GET* por parte del *backend*, el cual permite mostrar toda la información como se observa en la **Fig. 6**. Asimismo, el proceso que presenta el consumo y la presentación de la información se detalla de mejor manera en el **ANEXO III** del presente documento.



Fig. 6: Página informativa

Consumir *endpoints* para gestionar la información general de la Iglesia

Para gestionar toda la información de la Iglesia se ha implementado un formulario el cual permite la interacción y comunicación con varios *endpoints* que han sido creados en el *backend*, los cuales permiten que el usuario con perfil administrador pueda ingresar la información referente a la Iglesia como: noticias, álbum de fotos, mensajes bíblicos, eventos, entre otros. Por otra parte, por medio del *frontend* se ha consumido una ruta pública de tipo *GET* el cual permite mostrar toda la información con respecto a noticias, mensajes bíblicos y fotos de la Iglesia como se ilustran en las **Fig. 7**, **Fig. 8** y **Fig. 9**, una ruta privada de tipo *POST* que permite el registro de la información del formulario, una ruta privada de tipo *PUT* para actualizar la información del formulario y una ruta privada de tipo *DELETE* para eliminar información que se ha registrado. Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.



Fig. 7: Página para la gestión de noticias

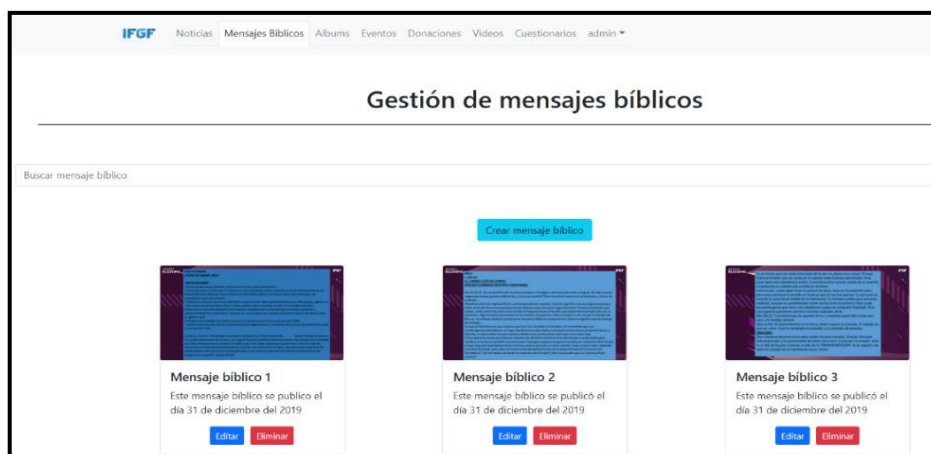


Fig. 8: Página para la gestión mensajes bíblicos

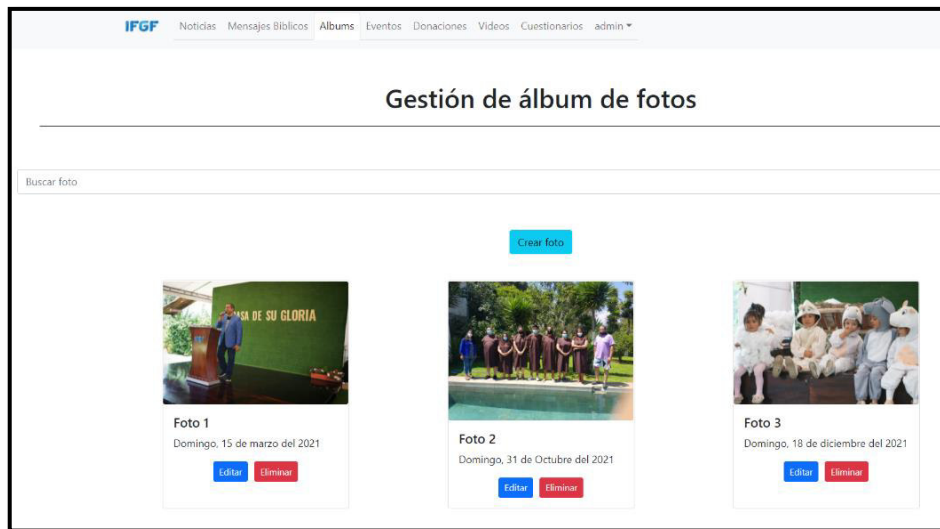


Fig. 9: Página para la gestión de álbum de fotos

Consumir *endpoints* para el registro y autenticación de usuarios

Para el inicio de sesión y registro de usuarios se han implementado formularios, los cuales permiten la interacción y comunicación con varios *endpoints* que se han creado en el *backend*, logrando de esta manera el ingreso de la información por parte de los usuarios para una respectiva validación y autenticación. Sin embargo, por medio del *frontend* se ha consumido una ruta privada de tipo *POST* el cual permite al usuario invitado por medio de un formulario el ingreso de datos como: nombre, apellido, email y contraseña para el registro, de la misma forma, por medio de otro formulario se ingresa el *email* y la contraseña permitiendo el inicio de sesión para el usuario con perfil miembro de la Iglesia como se observa en la **Fig. 10**. Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.



Fig. 10: Página para el registro de usuarios

Consumir *endpoints* para gestionar eventos

Para gestionar toda la información de la Iglesia se ha implementado un formulario el cual permite la interacción y comunicación con varios *endpoints* que han sido creados en el *backend*, los cuales permiten que el usuario con perfil administrador pueda gestionar toda la información relacionada a eventos. Así mismo, por medio del *frontend* se ha consumido una ruta pública de tipo *GET* el cual permite mostrar toda la información como se ilustra en la **Fig. 11**, una ruta privada de tipo *POST* que permite el registro de la información del formulario, una ruta privada de tipo *PUT* para actualizar la información del formulario y una ruta privada de tipo *DELETE* para eliminar información que se ha registrado. Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.

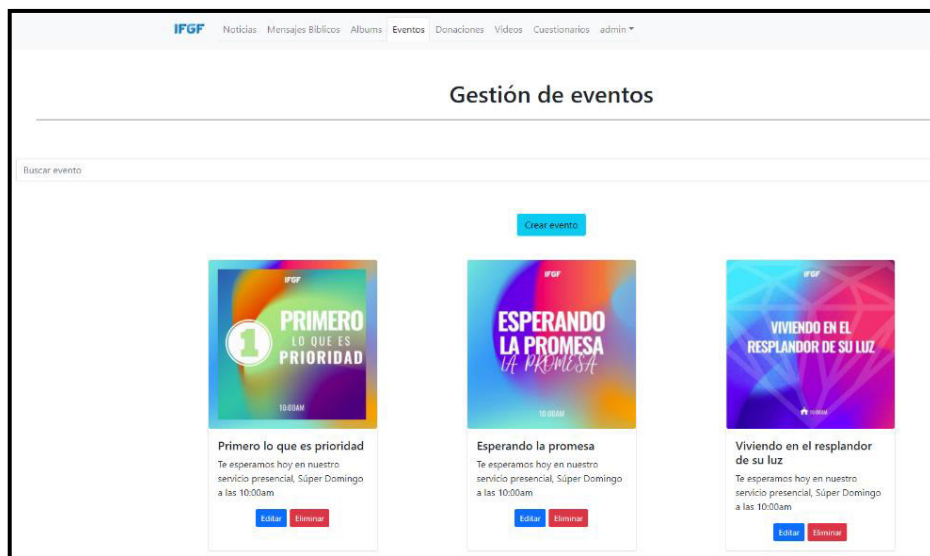


Fig. 11: Página para la gestión de eventos

Consumir *endpoint* para gestionar donaciones

Para gestionar las donaciones que se realizan hacia la Iglesia se ha creado un formulario el cual permite la interacción y comunicación con varios *endpoints* que han sido creados en el *backend*, los cuales permiten que el usuario con perfil administrador pueda visualizar las donaciones. Por otra parte, por medio del *frontend* se ha consumido una ruta pública de tipo *GET* el cual permite mostrar todas las donaciones que se han realizado por los miembros de la Iglesia como se ilustra en la **Fig. 12**.

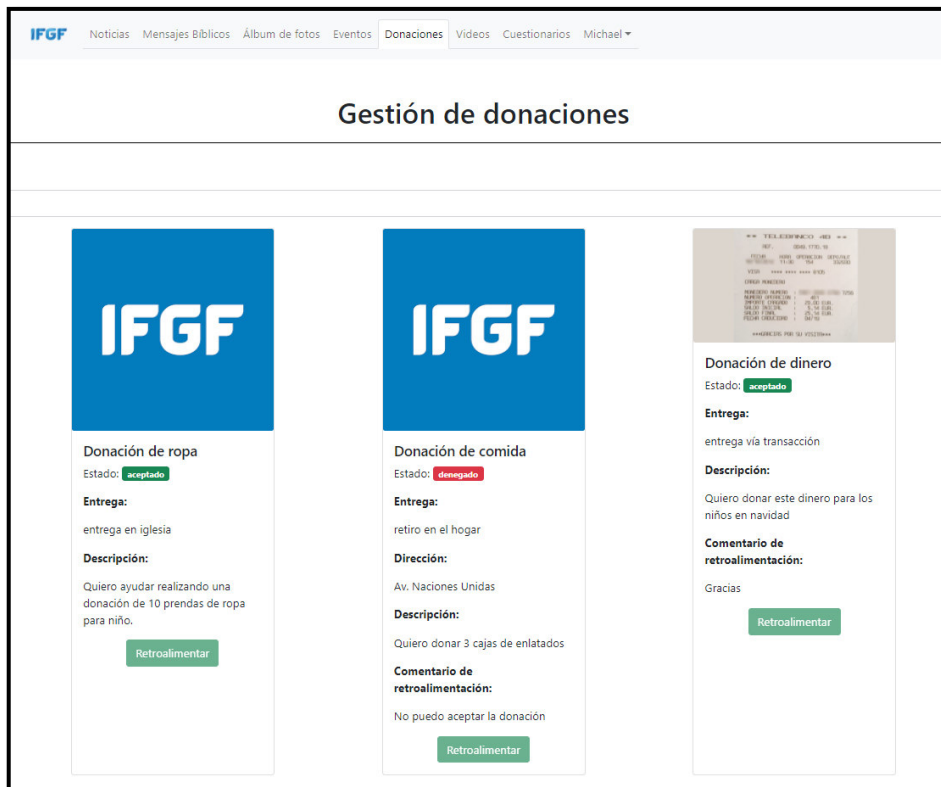


Fig. 12: Página para la gestión de donaciones

Consumir *endpoint* para visualizar y aprobar donaciones

Toda la información referente a las donaciones que han sido realizadas por los usuarios miembros de la Iglesia se puede visualizar en el módulo donaciones. También, para poder aprobar la donación en el *frontend* se ha creado varios métodos y componentes visuales que permitan consumir una ruta pública de tipo *POST* por parte del *backend*, el cual permite aceptar o denegar la donación como se observa en la Fig. 13. Además, el proceso que presenta el consumo y la validación respectiva se detalla de mejor manera en el ANEXO III del presente documento.

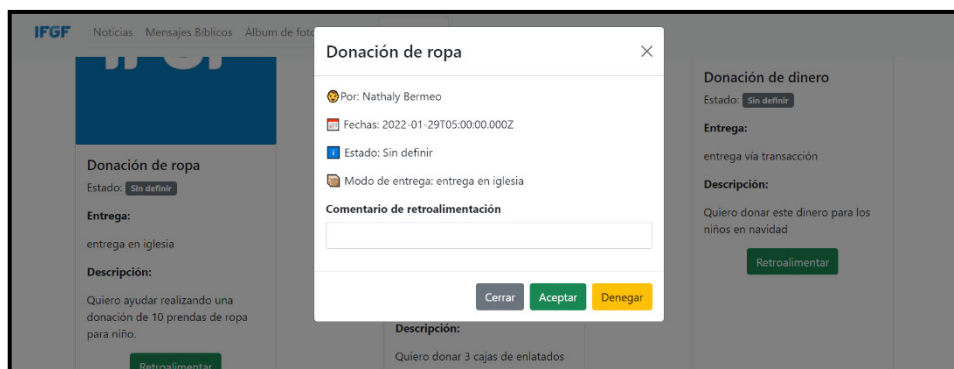


Fig. 13: Página para aprobar o denegar la donación

Consumir *endpoints* para gestionar vídeos

Para gestionar toda la información de la Iglesia se ha implementado un formulario el cual permite la interacción y comunicación con varios *endpoints* que han sido creados en el *backend*, los cuales permiten que el usuario con perfil administrador pueda gestionar toda la información relacionada a vídeos. Además, por medio del *frontend* se ha consumido una ruta pública de tipo *GET* el cual permite mostrar toda la información como se ilustra en la **Fig. 14**, una ruta privada de tipo *POST* que permite el registro de la información del formulario, una ruta privada de tipo *PUT* para actualizar la información del formulario y una ruta privada de tipo *DELETE* para eliminar información que se ha registrado. Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.

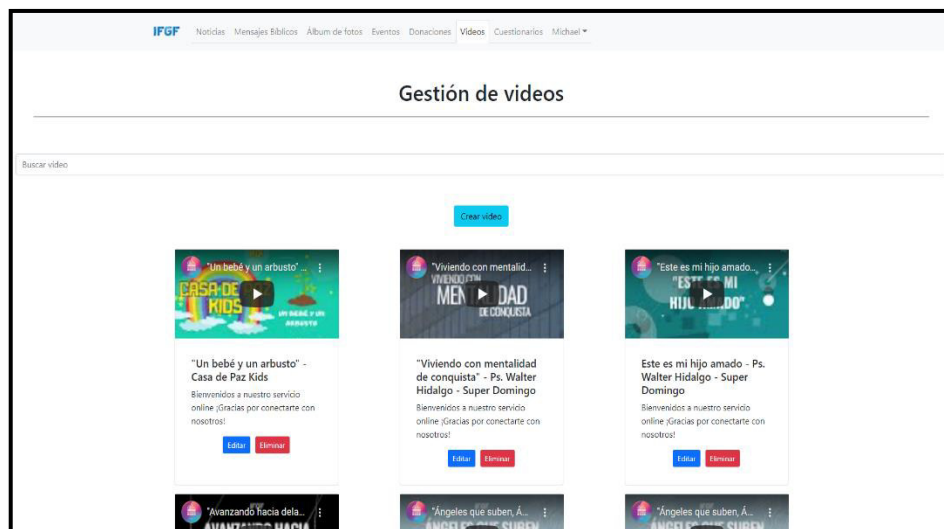


Fig. 14: Página para la gestión de vídeos

Consumir *endpoints* para gestionar cuestionarios

Para gestionar toda la información de la Iglesia se ha implementado un formulario el cual permite la interacción y comunicación con varios *endpoints* que han sido creados en el *backend*, los cuales permiten que el usuario con perfil administrador pueda gestionar toda la información relacionada a cuestionarios. Además, por medio del *frontend* se ha consumido una ruta pública de tipo *GET* el cual permite mostrar toda la información como se ilustra en la **Fig. 15**, una ruta privada de tipo *POST* que permite ingresar las respectivas preguntas referente a un solo tema, una ruta privada de tipo *PUT* para actualizar la información del formulario y una ruta privada de tipo *DELETE* para eliminar información que se ha registrado. Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.

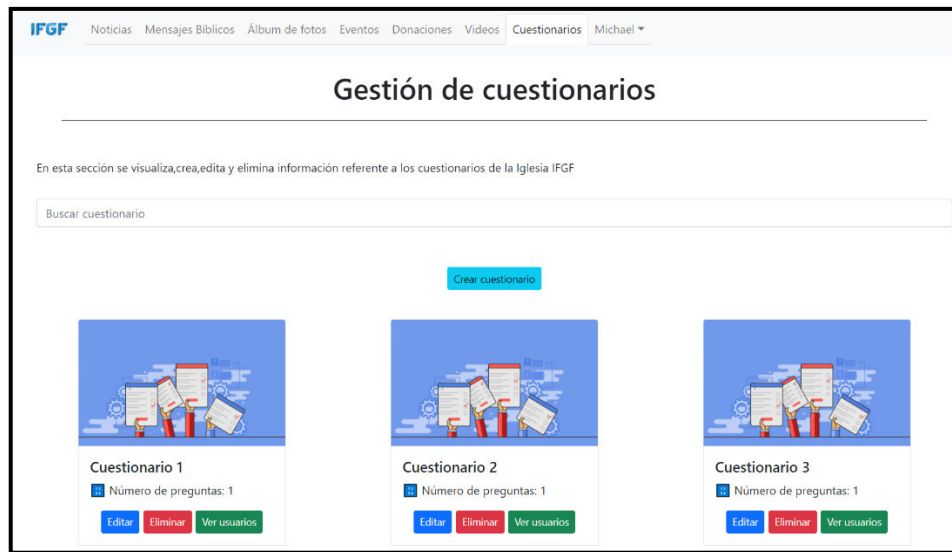


Fig. 15: Página para la gestión de cuestionarios

3.3 *Sprint 2*. Resultado de las interfaces para el usuario miembro de la Iglesia

Las actividades para el *Sprint 2* tienen que ver con:

- Consumir un *endpoint* para interactuar con donaciones.
- Consumir *endpoints* para interactuar con eventos.
- Consumir un *endpoint* para interactuar con cuestionarios.
- Consumir un *endpoint* para visualizar vídeos.

Consumir *endpoints* para interactuar con donaciones

En el *frontend* se han creado varios métodos y componentes visuales que permitan consumir rutas públicas y privadas que se han creado en el *backend*, logrando de esta manera que el usuario con perfil miembro de la Iglesia pueda realizar un tipo de donación para la Iglesia a través de un formulario. En ese sentido, por medio de una ruta pública de tipo *GET* permite mostrar todas las donaciones que se pueden realizar como se ilustran en las **Fig. 16**, **Fig. 17** y **Fig. 18**, una ruta privada de tipo *POST* para que el usuario puede realizar una donación a través de un formulario en el cual debe ingresar la siguiente información: nombre, apellido, dirección, tipo de donación y lugar. Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.



Fig. 16: Página para la donación de ropa

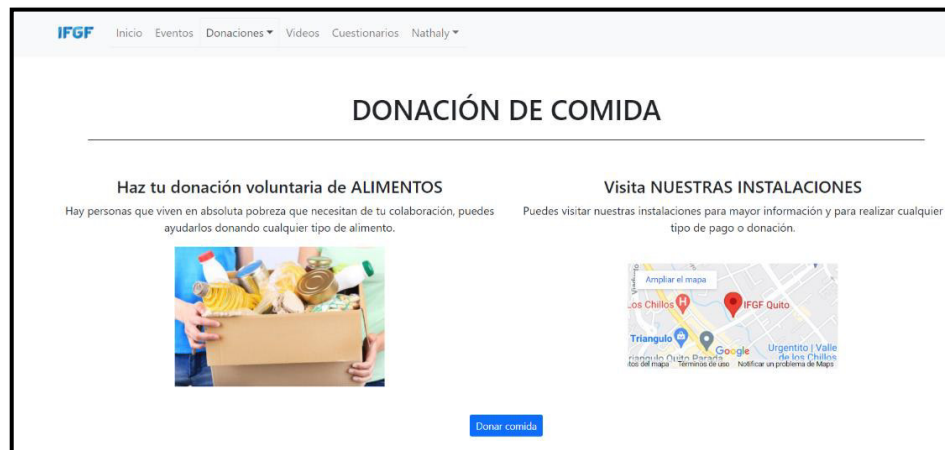


Fig. 17: Página para la donación de comida



Fig. 18: Página para la donación económica

Consumir *endpoints* para interactuar con eventos

En el *frontend* se han creado varios métodos y componentes visuales que permitan consumir rutas públicas y privadas que se han creado en el *backend*, logrando de esta manera que el usuario con perfil miembro de la Iglesia pueda inscribirse en un evento a través de un formulario. En ese sentido, por medio de una ruta pública de tipo *GET* permite mostrar todos los eventos como se ilustra en la **Fig. 19**, una ruta privada de tipo *POST* para que el usuario realice la inscripción a un evento a través de un formulario en el cual debe ingresar la siguiente información: nombre, apellido, dirección y confirmación del evento, por otra parte, se consume una ruta privada de tipo *DELETE* para cancelar la participación al evento. Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.

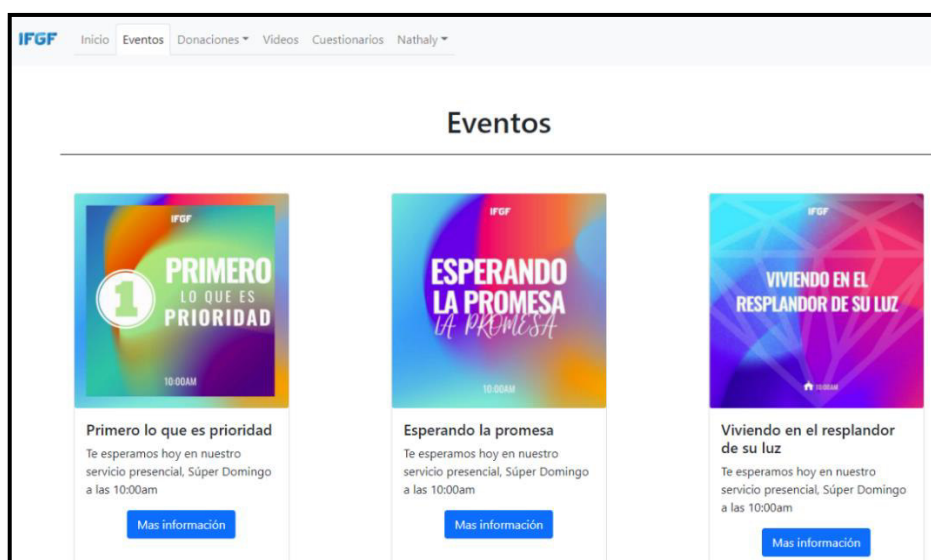


Fig. 19: Página para la interacción con eventos

Consumir un *endpoint* para interactuar con cuestionarios

En el *frontend* se han creado varios métodos y componentes visuales que permitan consumir rutas públicas y privadas que se han creado en el *backend*, logrando de esta manera que el usuario con perfil miembro de la Iglesia pueda contestar los cuestionarios de la Iglesia. En ese sentido, por medio de una ruta pública de tipo *GET* permite mostrar todos los cuestionarios como se ilustra en la **Fig. 20**, una ruta privada de tipo *POST* para que el usuario pueda contestar a cada una de las preguntas del cuestionario, una ruta privada de tipo *DELETE* para cancelar la realización del cuestionario, Por último, el proceso para mostrar el consumo de las demás rutas y validaciones respectivas se detalla en el **ANEXO III** del presente documento.

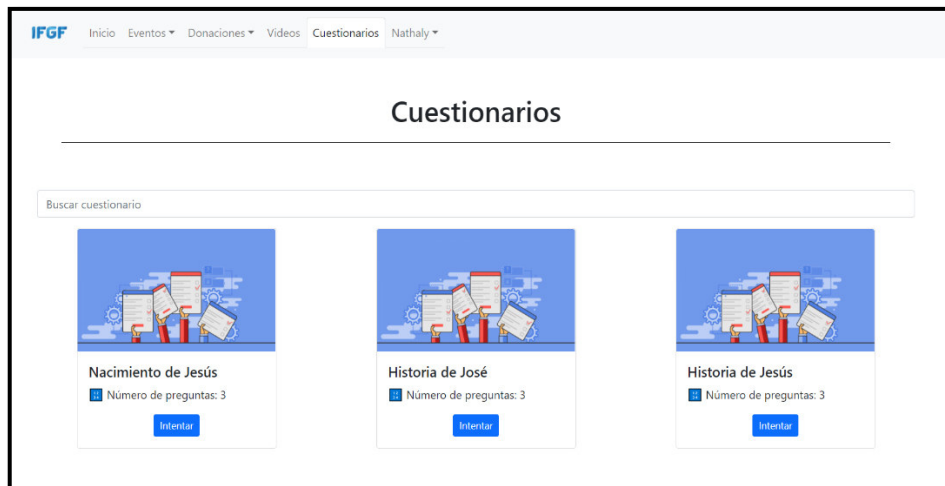


Fig. 20: Página para la interacción con cuestionarios

Consumir un *endpoint* para visualizar vídeos

Toda la información referente a los vídeos de la Iglesia se puede visualizar en la página por medio del módulo respectivo. No obstante, para poder visualizar esta información en el *frontend* se han creado varios métodos y componentes visuales que permitan consumir una ruta pública de tipo *GET* por parte del *backend*, el cual permite mostrar toda la información como se observa en la **Fig. 21**. Además, el proceso que presenta el consumo y la presentación de la información se detalla de mejor manera en el **ANEXO III** del presente documento.

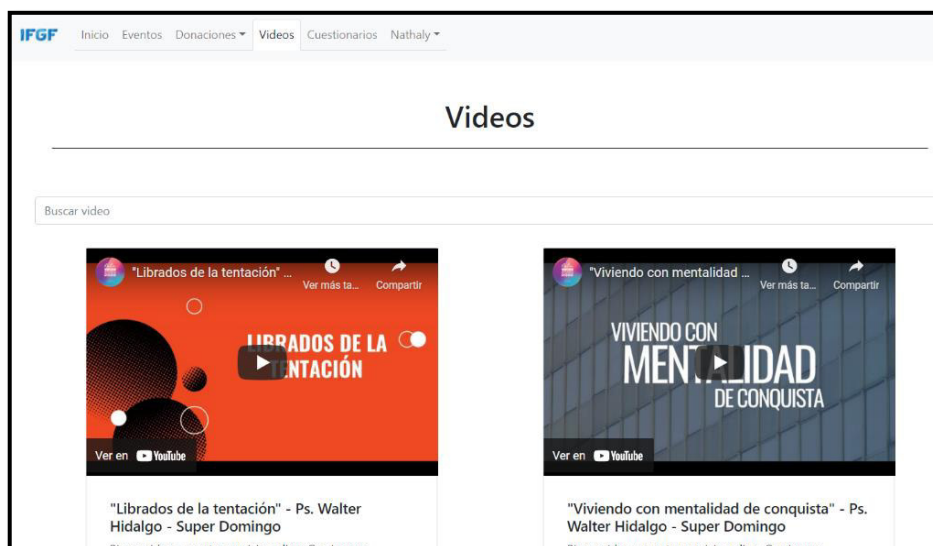


Fig. 21: Página para visualizar vídeos

3.4 *Sprint 3. Pruebas del frontend*

Una vez que se ha finalizado la etapa de codificación de cada uno de los componentes visuales por parte del *frontend* las actividades para el *Sprint 3* tienen que ver con:

- Resultados de la ejecución de las pruebas unitarias.
- Resultados de la ejecución de las pruebas de compatibilidad.
- Resultado de la ejecución de las pruebas de aceptación.

Resultados de la ejecución de las pruebas unitarias

Las pruebas unitarias ayudan a verificar la funcionalidad de cada parte de código del sistema *software*. En ese sentido, se ha utilizado la herramienta *Jest* que permite realizar casos de prueba similares al comportamiento del usuario en un sistema *software*. Por otra parte, *Jest* es muy útil para realizar este tipo de pruebas ya que garantiza la calidad del código y con ello evitar posibles *bugs* en un futuro [41]. Además, también se ha utilizado la herramienta *Testing Library* para probar los componentes de la interfaz de usuario [42].

Las **Fig. 22** muestra un fragmento del código para la ejecución de noticias que se ha implementado para realizar la prueba, por otra parte, en la **Fig. 23** se puede apreciar el resultado que se ha obtenido tras ejecutar la respectiva prueba. Mientras que el detalle completo de la ejecución y resultados de las demás pruebas se lo puede apreciar en el **ANEXO II** del presente documento.

```
import { renderHook, act } from '@testing-library/react-hooks'
import { useNews } from '../src/hooks/useNews'

test('should use useNews', () => {
  const { result } = renderHook( callback: () => useNews()
  const newNews = [{title: 'these are the News', description: ''}]
  result.current.setNews(newNews)
  expect(result.current.news.length).toBe(1)
  expect(result.current.news).toEqual(newNews)
})
```

Fig. 22: Fragmento de código para la actualización de estados de noticias

```
PASS __tests__/index.test.jsx
  ✓ should use useNews (181 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        3.676 s
Ran all test suites related to changed files.
```

Fig. 23: Resultado de la prueba

Una vez que se han ejecutado estas pruebas y en base a los resultados que se han obtenido, se determina que cada uno de los módulos del *frontend* no presenta ningún fallo a nivel de código, funcionalidad o validación respectivamente.

Resultados de la ejecución de las pruebas de compatibilidad

Este tipo de pruebas tienen el objetivo de verificar que la información de cada uno de los módulos se presente de forma correcta y ordenada en todos los navegadores o dispositivos donde se visualice [43]. A continuación, se procede a realizar esta prueba en diferentes navegadores como se observa en la **TABLA VIII** para verificar la adaptabilidad del contenido por parte del *frontend*. Por último, el detalle completo de la ejecución y resultados de las demás pruebas se lo puede apreciar en el **ANEXO II** del presente documento.

TABLA VIII: Navegadores que se han utilizado para las pruebas

NOMBRE	VERSIÓN	OBSERVACIÓN
Opera	82.0.4227.50	Completamente funcional
Google Chrome	97.0.4692.71	Completamente funcional
Microsoft Edge	96.0.1054.53	Completamente funcional

Una vez que se han ejecutado estas pruebas y en base a los resultados que se han obtenido en los diferentes navegadores se determina que todo el contenido se adapta de forma correcta sin presentar ninguna falla a nivel de la presentación de la información o funcionalidad.

Resultado de la ejecución de las pruebas de aceptación

Las pruebas de aceptación tienen el principal objetivo de que el dueño del producto verifique y certifique el producto *software* en base a los requerimientos que se han establecido, logrando de esta manera obtener resultados oportunos y con ello permitir realizar el despliegue a producción del *frontend* [44].

La **TABLA IX** presenta la primera prueba de aceptación que ha sido desarrollada de acuerdo a los requerimientos que han sido solicitados por parte del cliente al inicio del proyecto, mientras que en el **ANEXO II** del presente documento se detalla por completo las demás pruebas y los resultados que se han obtenido.

TABLA IX: Pruebas de aceptación 1 – Consumir varios *endpoints* para presentar la información general de la Iglesia

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA001	Identificador historia de Usuario: HU001
Nombre: Consumir varios <i>endpoints</i> para presentar la información general de la Iglesia.	
Descripción: El usuario invitado, administrador y miembro de la Iglesia necesitan consumir varios <i>endpoints</i> para visualizar la información general de la Iglesia e interactuar con los siguientes módulos: <ul style="list-style-type: none">• Noticias• Eventos• Donaciones• Cuestionarios• Vídeos	
Pasos de ejecución: <ul style="list-style-type: none">• Ir o digitar la URL del sistema <i>web</i> en el navegador.• Dar clic en “Inicio”, en el menú superior.• Dar clic en “Eventos”, en el menú superior.• Dar clic en “Donaciones”, en el menú superior.• Dar clic en “Vídeos”, en el menú superior.• Dar clic en “Cuestionarios”, en el menú superior.	
Resultado deseado: El <i>frontend</i> permite visualizar información general de la Iglesia.	

Evaluación de la prueba:

Se comprueba el resultado esperado.

Aprobación del cliente 100%.

Una vez que se han ejecutado estas pruebas y en base a los resultados que se han obtenido se determina una aceptación total por parte de los usuarios en el aspecto de funcionalidad e interacción de los módulos que ofrece el *frontend*, de esta forma se ha demostrado que se ha cumplido con los requerimientos del cliente y se puede proceder a la siguiente etapa.

3.5 *Sprint 4. Despliegue del frontend*

Una vez que se ha finalizado la etapa de codificación y pruebas de cada uno de los componentes visuales por parte del *frontend* las actividades para el *Sprint 4* tienen que ver con el despliegue del sistema *web* en *Vercel*.

Despliegue del sistema *web* en *Vercel*

En esta sección se determina los pasos que se han realizado para el despliegue del *frontend* en un ambiente de producción a través de la plataforma *Vercel*. Cabe recalcar que este procedimiento se encuentra detallado de mejor manera en el en el **ANEXO IV** del presente documento. Sin embargo, La **Fig. 24** ilustra la fase inicial la cual corresponde a la creación del proyecto en *Vercel*, mientras que, la **Fig. 25** lustra el resultado final de haber realizado el despliegue del *frontend* funcionando correctamente bajo un dominio.

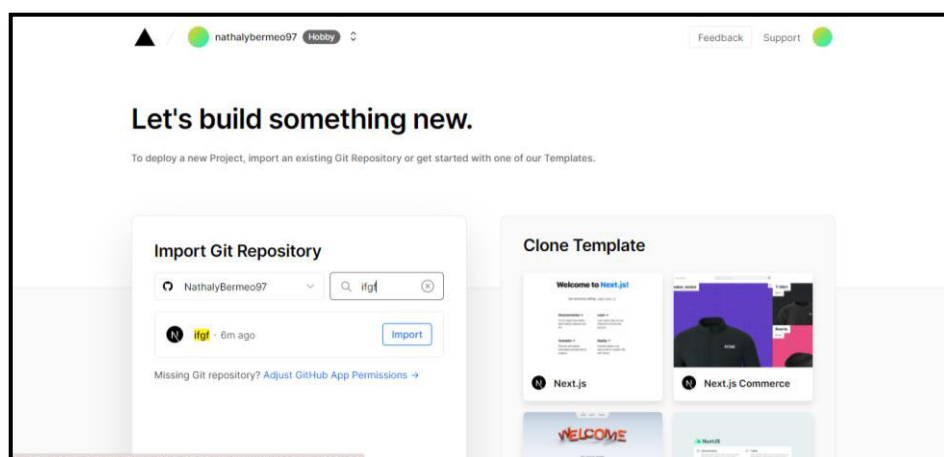


Fig. 24: Inicio de la creación del proyecto en *Vercel*

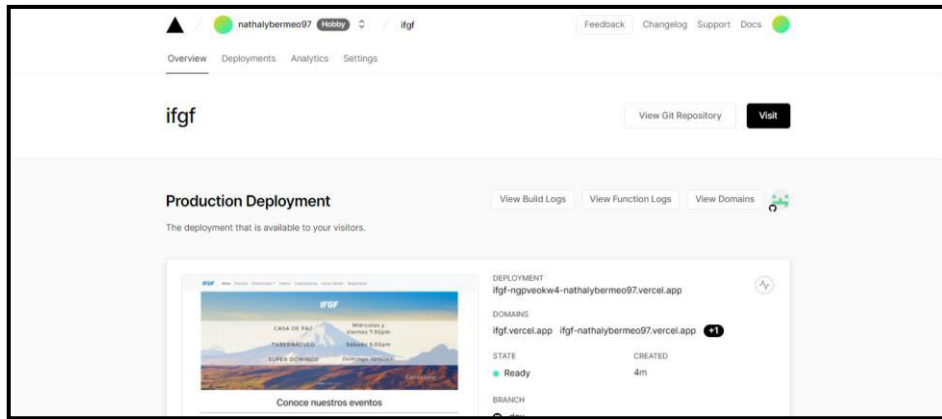


Fig. 25: Despliegue del *frontend* en Vercel

Finalmente, el pastor Walter Hidalgo de la Iglesia Fraternidad Internacional del Evangelio Completo GISI (IFGF) ha otorgado un certificado, el mismo que ratifica el cumplimiento de todos los requerimientos y funcionalidades del *frontend* que han sido solicitados al comienzo del proyecto. El certificado se adjunta en el **ANEXO II**.

4 CONCLUSIONES

En esta sección se muestran las conclusiones que se han obtenido durante el desarrollo de este trabajo de integración curricular.

- El *frontend* cumplen con el alcance y objetivos que se han establecido, permitiendo que la comunidad IFGF pueda interactuar con todos los miembros de la Iglesia, evitando de esta manera las aglomeraciones y la propagación de contagios por medio de la tecnología.
- La implementación de la metodología ágil *Scrum* ha permitido que el producto final sea desarrollado de manera adecuada logrando cumplir los periodos de tiempo que se han planificado por cada uno de los *Sprints*.
- La Recopilación de requerimientos al inicio del proyecto ha sido de gran importancia ya que debido a esto se ha desarrollado un *frontend* en base a las necesidades del dueño del producto.
- El uso correcto de herramientas y librerías para el desarrollo del *frontend* ha sido de gran ayuda en la etapa de codificación ya que han permitido el desarrollo de las diferentes funcionalidades de una forma óptima y sin tanta complejidad.
- Las pruebas que se han ejecutado en el *frontend* han servido para verificar el manejo de los diferentes módulos, componentes y funcionalidades de esta manera se ha obtenido resultados favorables.

5 RECOMENDACIONES

En esta sección se muestran las recomendaciones que se han obtenido durante el desarrollo de este trabajo de integración curricular.

- Es recomendable seguir trabajando con el *frontend*, para de esta forma seguir puliendo partes del diseño de cada una de las interfaces y nuevas funcionalidades.
- Una vez que se ha desplegado el *frontend* en un ambiente de producción es recomendable que el usuario con perfil administrador actualice la información con relación a la Iglesia.
- Es recomendable que el desarrollador del *frontend* se mantenga actualizado con las necesidades de la Iglesia, logrando de esta manera crear nuevos módulos conjuntamente con el *backend*.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Ministerio de Economía y Finanzas, «Impacto social del COVID-19 en Ecuador: desafíos y respuestas,» [En línea]. Available: http://www.latameconomy.org/Covid-19/Nota_1_Impacto_social_COVID19_Ecuador.pdf. [Último acceso: 2021 08 30].
- [2] CoronavirusEcuador, «Estadísticas COVID-19,» 2021 08 31. [En línea]. Available: <https://www.coronavirusecuador.com/estadisticas-covid-19/>.
- [3] M. Czerny, «La Iglesia frente a la emergencia del COVID-19,» 18 05 2020. [En línea]. Available: <https://www.humandevlopment.va/es/news/2020/la-iglesia-frente-a-la-emergencia-del-covid-19.html>. [Último acceso: 2021 08 30].
- [4] El telegrafo, «Iglesias podrán realizar misas con aforo del 30% de lunes a viernes,» 06 05 2021. [En línea]. Available: <https://www.eltelegrafo.com.ec/noticias/actualidad/44/iglesias-podran-realizar-misas-con-aforo-del-30-de-lunes-a-viernes>. [Último acceso: 30 08 2020].
- [5] A. J. Torrado, «Predicación virtual: ¿Una herramienta para la cuarentena o un recurso permanente para la iglesia?,» [En línea]. Available: <https://biteproject.com/predicacion-virtual-vs-presencial/>. [Último acceso: 30 08 2021].
- [6] «.w3,» 17 07 2020. [En línea]. Available: <https://www.paginas-web.site/la-importancia-del-desarrollo-web-en-nuestra-actualidad/>. [Último acceso: 06 10 2021].
- [7] P. Stefaniak, «descubre comunicacion,» 26 07 2019. [En línea]. Available: <https://descubrecomunicacion.com/que-es-backend-y-frontend/>. [Último acceso: 07 10 2021].
- [8] B. Garza, «Servnet,» 30 03 2021. [En línea]. Available: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programación-de-una-aplicación-web>. [Último acceso: 06 10 2021].
- [9] J. Torrencilla, «Importancia de las herramientas tecnológicas,» 5 07 2020. [En línea]. Available: <https://www.astraps.com/articulo/1185/importanciade-las-herramientas-tecnologicas/>. [Último acceso: 30 08 2021].
- [10] Neolo, «¿Qué es un sitio web?,» 02 08 2019. [En línea]. Available: <https://www.neolo.com/blog/que-es-un-sitio-web.php>. [Último acceso: 31 08 30].
- [11] «The New York Times,» [En línea]. Available: <https://www.nytimes.com/es/2020/07/12/espanol/mundo/iglesias-coronavirus.html>. [Último acceso: 26 12 2021].
- [12] S. Universidades, «Santander becas,» 21 12 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 01 01 2021].
- [13] digité, «digité,» 2020. [En línea]. Available: <https://www.digite.com/es/agile/metodologia-agil/>. [Último acceso: 01 01 2021].

- [14] I. d. Souza, «rockContent,» 05 03 2020. [En línea]. Available: <https://rockcontent.com/es/blog/front-end-y-back-end/>. [Último acceso: 01 01 2021].
- [15] «React,» [En línea]. Available: <https://es.reactjs.org>. [Último acceso: 04 01 2022].
- [16] D. D. LUCA, «DAMIAN DE LUCA,» [En línea]. Available: <https://damiandeluca.com.ar/next-js-un-framework-de-react-para-el-lado-servidor>. [Último acceso: 03 01 2021].
- [17] K. Sanchez, «freeCodeCamp,» 20 01 2021. [En línea]. Available: <https://www.freecodecamp.org/espanol/news/aprende-next-js-desde-cero/>. [Último acceso: 08 01 2022].
- [18] «Uniphyton,» [En línea]. Available: <https://unipython.com/frontend-y-backend/>. [Último acceso: 06 01 2022].
- [19] «visual engineering,» [En línea]. Available: <https://visual-engin.com/2019/09/04/figma-herramienta-de-diseno-napptilus/>. [Último acceso: 13 01 2022].
- [20] «loadview,» [En línea]. Available: <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>. [Último acceso: 26 01 2022].
- [21] QuestionPro, «QuestionPro,» [En línea]. Available: <https://www.questionpro.com/blog/es/que-es-un-estudio-de-caso/>. [Último acceso: 03 01 2022].
- [22] «RedHat,» [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology>.
- [23] L. Gonçalves, «Adapt Methology,» 30 10 2020. [En línea]. Available: <https://adaptmethodology.com/es/que-es-la-metodologia-agil/>.
- [24] J. Roche, «Deloitte.,» [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>.
- [25] C. Ramos, «cristinamosvega.com,» 20 02 2017. [En línea]. Available: <https://cristinamosvega.com/z-los-artefactos-scrum/>.
- [26] «SCRUM MEXICO,» 20 08 2018. [En línea]. Available: <https://scrum.mx/informate/historias-de-usuario>.
- [27] «EALDE,» [En línea]. Available: <https://www.ealde.es/product-backlog-sprint-backlog/>.
- [28] «SELECCIÓN DE TÉCNICAS DE INGENIERÍA DE SOFTWARE,» [En línea]. Available: https://virtual.itca.edu.sv/Mediadores/stis/35___diseo_de_la_interfaz_de_usuario.html.
- [29] «NeoAttack,» [En línea]. Available: <https://neoattack.com/neowiki/arquitectura-web/>.

- [30] U. d. Alicante, «Servicio de InformáticaASP.NET MVC 3 Framework,» [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.
- [31] J. S. Code, «jsourcecode,» 01 10 2019. [En línea]. Available: <https://jsourcecode.com/blog/patrones-arquitectonicos-que-son.html>.
- [32] slideshare. [En línea]. Available: <https://es.slideshare.net/maurelisvelez/presentacin-info-220-copia>.
- [33] «next_U,» [En línea]. Available: <https://www.nextu.com/blog/que-es-y-como-funciona-react-js/>.
- [34] M. Ruiz, «OpenWebinars,» 19 09 2017. [En línea]. Available: <https://openwebinars.net/blog/que-es-nextjs/>.
- [35] «rockcontent,» [En línea]. Available: <https://rockcontent.com/es/blog/bootstrap/>.
- [36] «React-Boostrap,» [En línea]. Available: <https://react-bootstrap.github.io/>. [Último acceso: 26 01 2022].
- [37] «react-icons,» [En línea]. Available: <https://react-icons.github.io/react-icons/icons?name=bs>. [Último acceso: 26 01 2022].
- [38] react-hook-form, «react-hook-form,» [En línea]. Available: <https://react-hook-form.com/>. [Último acceso: 25 Junio 2021].
- [39] S. Components, «Styled Components,» [En línea]. Available: <https://styled-components.com/>. [Último acceso: 25 Junio 2021].
- [40] E. Sarabia, «Código limpio y consistente. ESLint + Prettier en tus proyectos con React,» DEV Community, Diciembre 2019. [En línea]. Available: <https://dev.to/ericksarabia/codigo-limpio-y-consistente-eslint-prettier-en-tus-proyectos-con-react-462f>.
- [41] Geekflare, «Geekflare,» 2021. [En línea]. Available: <https://geekflare.com/es/javascript-unit-testing/>. [Último acceso: 12 12 2021].
- [42] «Testing Library,» [En línea]. Available: <https://testing-library.com/docs/>. [Último acceso: 26 01 2022].
- [43] «My server name,» [En línea]. Available: <https://es.myservername.com/what-is-software-compatibility-testing#:~:text=La%20prueba%20de%20compatibilidad%20consiste,manera%20para%20todas%20las%20plataformas.&text=Por%20lo%20general%2C%20el%20equipo,aplicación%20en%20una%20única%20plataforma..> [Último acceso: 12 12 2021].
- [44] Digité, «Digité,» [En línea]. Available: <https://www.digite.com/es/agile/pruebas-de-aceptacion/>. [Último acceso: 20 01 2022].

7 ANEXOS

A continuación, se presentan todos los Anexos que se han utilizado en el desarrollo del *frontend*, los cuales se encuentran divididos de la siguiente manera:

- **ANEXO I.** Resultado del programa antiplagio *Turnitin*.
- **ANEXO II.** Manual Técnico.
- **ANEXO III.** Manual de Usuario.
- **ANEXO IV.** Manual de Instalación.

ANEXO I

A continuación, se presenta el certificado que el Director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta antiplagio *Turnitin*.



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, de 14 febrero de 2022

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado Desarrollo de un frontend asociado al **DESARROLLO DE UN SISTEMA PARA LA INTERACCIÓN DE LOS MIEMBROS DE LA COMUNIDAD DE LA IGLESIA IFGF** elaborado por el estudiante **NATHALY LISSETTE BERMEO PANCHI** de la carrera en **TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE**, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito sesiones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 10%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente

documento para los trámites de titulación.

Atentamente,

Una firma manuscrita en tinta azul que parece decir 'BLoarte'.

Ing. Byron Loarte, MSc.

Profesor Ocasional

Escuela de Formación de Tecnólogos

ANEXO II

Recopilación de requerimientos

En la **TABLA X** se muestra los requerimientos que se han obtenido al inicio del proyecto de acuerdo a lo solicitado por el dueño del producto.

TABLA X: Recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
FRONTEND	RR002	Como usuario administrador y miembro de la Iglesia necesitan consumir a varios <i>endpoints</i> para poder iniciar y cerrar sesión.
	RR003	Como usuario administrador necesita consumir a varios <i>endpoints</i> para realizar lo siguiente: <ul style="list-style-type: none"> • Ingresar información de la Iglesia (noticias, mensajes bíblicos y álbum de fotos). • Visualizar información de la Iglesia (noticias, mensajes bíblicos y álbum de fotos). • Modificar información de la Iglesia (noticias, mensajes bíblicos y álbum de fotos). • Eliminar información de la Iglesia (noticias, mensajes bíblicos y álbum de fotos).
	RR004	Como usuario administrador necesita consumir a varios <i>endpoints</i> para realizar lo siguiente: <ul style="list-style-type: none"> • Registrar eventos. • Visualizar eventos. • Modificar eventos. • Eliminar eventos.
	RR005	Como usuario administrador necesita consumir un <i>endpoint</i> para visualizar los miembros inscritos a un evento.
	RR006	Como usuario administrador necesita consumir un <i>endpoint</i> para visualizar las donaciones realizadas por los usuarios miembros de la iglesia.

	RR007	<p>Como usuario administrador necesita consumir varios <i>endpoints</i> para realizar lo siguiente:</p> <ul style="list-style-type: none"> • Registrar vídeos. • Visualizar vídeos. • Modificar vídeos. • Eliminar vídeos.
	RR008	<p>Como usuario administrador necesita consumir varios <i>endpoints</i> para realizar lo siguiente:</p> <ul style="list-style-type: none"> • Registrar cuestionarios. • Visualizar cuestionarios. • Modificar cuestionarios. • Eliminar cuestionarios.
	RR009	<p>Como usuario invitado necesita consumir un <i>endpoint</i> para registrarse.</p>
	RR010	<p>Como usuario miembro necesita consumir varios <i>endpoints</i> para iniciar sesión y cerrar sesión.</p>
	RR011	<p>Como usuario miembro de la Iglesia necesita consumir varios <i>endpoints</i> para realizar lo siguiente:</p> <ul style="list-style-type: none"> • Realizar donación de dinero. • Realizar donación de ropa. • Realizar donación de alimentos.
	RR012	<p>Como usuario miembro de la Iglesia necesita consumir varios <i>endpoints</i> para realizar lo siguiente:</p> <ul style="list-style-type: none"> • Visualizar eventos creados. • Inscribirse a un evento. • Eliminar inscripción al evento.
	RR013	<p>Como usuario miembro de la Iglesia necesita consumir varios <i>endpoints</i> para realizar lo siguiente:</p> <ul style="list-style-type: none"> • Visualizar cuestionarios. • Realizar cuestionarios. • Visualizar puntaje.

	RR014	Como usuario miembro de la Iglesia necesita consumir un <i>endpoint</i> para realizar lo siguiente: <ul style="list-style-type: none"> • Visualizar vídeos.
--	--------------	--

Historias de Usuario

Culminada la etapa de recopilación de requerimientos, se procede a desarrollar las Historias de Usuario, para el *frontend*. A continuación, se presentan las 11 Historias de Usuario escritas en base a los requerimientos del proyecto que va desde la **TABLA XI** a la **TABLA XXI**.

TABLA XI: Historia de usuario 2 - Consumir varios *endpoints* para iniciar y cerrar sesión

HISTORIA DE USUARIO	
Identificador (ID): HU002	Usuario: Miembro de la Iglesia, administrador
Nombre Historia: Consumir varios <i>endpoints</i> para iniciar y cerrar sesión.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable (es): Nathaly Bermeo	
Descripción: El usuario miembro de la Iglesia puede consumir varios <i>endpoints</i> para iniciar sesión ya que es indispensable para interactuar con ciertos módulos e ingresar información. Por otra parte, el usuario administrador necesita consumir varios <i>endpoints</i> para iniciar sesión y gestionar los módulos asignados a su cargo. Además, el usuario administrador y miembro de la Iglesia pueden dejar de interactuar con el <i>frontend</i> mediante el cierre de sesión.	
Observación: Cuando el usuario administrador y miembro de la Iglesia ingresen usuario y contraseña de manera incorrecta se presenta un mensaje de error.	

TABLA XII: Historia de usuario 3 - Consumir varios *endpoints* para ingresar, visualizar, modificar y eliminar información de la Iglesia

HISTORIA DE USUARIO	
Identificador (ID): HU003	Usuario: Administrador

Nombre Historia: Consumir varios <i>endpoints</i> para ingresar, visualizar, modificar y eliminar información de la Iglesia.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Nathaly Bermeo	
<p>Descripción: El usuario administrador en el <i>frontend</i> puede consumir varios <i>endpoints</i> para ingresar, visualizar, modificar y eliminar información respecto a:</p> <ul style="list-style-type: none"> • Noticias • Mensajes bíblicos • Álbum de fotos <p>El administrador debe realizar el siguiente formulario:</p> <ul style="list-style-type: none"> • Título • Descripción • Imagen 	
<p>Observación:</p> <p>El usuario administrador necesita obligatoriamente iniciar sesión para poder gestionar toda la información anteriormente citada.</p>	

TABLA XIIIIV: Historia de usuario 4 - Consumir varios *endpoints* para registrar, visualizar, modificar y eliminar eventos

HISTORIA DE USUARIO	
Identificador (ID): HU004	Usuario: Administrador
Nombre Historia: Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar eventos.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Nathaly Bermeo	
<p>Descripción: El usuario administrador en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar eventos cuando el usuario lo requiera.</p> <p>El registro y modificación de un evento se realiza mediante un formulario donde se ingresa la siguiente información:</p> <ul style="list-style-type: none"> • Nombre • Descripción 	

<ul style="list-style-type: none"> • Ubicación • Fecha • Imagen • Horario • Costo • Número de personas para el aforo del evento <p>En caso de eliminar un evento existe la opción requerida.</p>
<p>Observación:</p> <p>El usuario administrador puede consumir los <i>endpoints</i> para registrar, visualizar, modificar y eliminar eventos a través de su autenticación en el <i>frontend</i> donde puede gestionar todas las actividades anteriormente mencionadas.</p>

TABLA XIV: Historia de usuario 5 - Consumir un *endpoint* para visualizar los miembros inscritos en un evento

HISTORIA DE USUARIO	
Identificador (ID): HU005	Usuario: Administrador
Nombre Historia: Consumir un <i>endpoint</i> para visualizar los miembros inscritos en un evento.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Nathaly Bermeo	
Descripción: El usuario administrador en el <i>frontend</i> puede consumir un <i>endpoint</i> para visualizar a los miembros inscritos a un evento donde puede observar el nombre, apellido, celular/teléfono del miembro de la Iglesia.	
Observación:	
El usuario administrador puede consumir el <i>endpoint</i> del <i>backend</i> para visualizar los miembros inscritos en un evento mediante el inicio de sesión.	

TABLA XVI: Historia de usuario - Consumir un *endpoint* para visualizar las donaciones realizadas

HISTORIA DE USUARIO	
Identificador (ID): HU006	Usuario: Administrador
Nombre Historia: Consumir un <i>endpoint</i> para visualizar las donaciones realizadas.	

Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable (es): Nathaly Bermeo	
<p>Descripción: El usuario administrador en el <i>frontend</i> puede consumir un <i>endpoint</i> para visualizar las donaciones realizadas donde se observa el nombre, apellido y tipo de donación realizada por parte del usuario miembro de la Iglesia.</p> <p>Según el tipo de donación por parte del usuario miembro de la Iglesia, el usuario administrador puede visualizar una fotografía verificando la donación realizada.</p>	
<p>Observación:</p> <p>El usuario administrador consume el <i>endpoint</i> del <i>backend</i> para visualizar las donaciones realizadas en el sistema <i>web</i> mediante el inicio de sesión.</p>	

TABLA XVII: Historia de usuario 7 - Consumir varios *endpoints* para registrar, visualizar, modificar y eliminar vídeos

HISTORIA DE USUARIO	
Identificador (ID): HU007	Usuario: Administrador
Nombre Historia: Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar vídeos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable (es): Nathaly Bermeo	
<p>Descripción: El usuario administrador puede consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar vídeos según lo requiera.</p> <p>El administrador debe realizar el siguiente formulario:</p> <ul style="list-style-type: none"> • Título • Descripción • Tipo de video (niños, jóvenes y familia) • Enlace url 	
<p>Observación:</p> <p>El usuario administrador puede consumir varios <i>endpoints</i> del <i>backend</i> para registrar, visualizar, modificar y eliminar vídeos en el <i>frontend</i> mediante el inicio de sesión.</p>	

TABLA XVIII: Historia de usuario - Consumir varios *endpoints* para registrar, visualizar, modificar y eliminar cuestionarios

HISTORIA DE USUARIO	
Identificador (ID): HU008	Usuario: Administrador
Nombre Historia: Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar cuestionarios.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable (es): Nathaly Bermeo	
<p>Descripción: El usuario administrador puede consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar cuestionarios.</p> <p>El administrador debe realizar el siguiente formulario:</p> <ul style="list-style-type: none"> • Tema • Pregunta • Opciones <p>Estos cuestionarios son realizados en base a diferentes temas religiosos para mejorar el aprendizaje de los miembros de la Iglesia.</p> <p>Además, el usuario administrador puede consumir un <i>endpoint</i> del <i>backend</i> para visualizar el puntaje obtenido por cada miembro de la Iglesia.</p>	
<p>Observación:</p> <p>El usuario administrador puede consumir varios <i>endpoints</i> del <i>backend</i> para registrar, visualizar, modificar y eliminar cuestionarios en el <i>frontend</i> mediante el inicio de sesión.</p>	

TABLA XVIIIIX: Historia de usuario 9 - Consumir un *endpoint* para registrarse en el *frontend*

HISTORIA DE USUARIO	
Identificador (ID): HU009	Usuario: Invitado
Nombre Historia: Consumir un <i>endpoint</i> para registrarse en el <i>frontend</i>	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable (es): Nathaly Bermeo	
<p>Descripción: El usuario invitado debe consumir un <i>endpoint</i> para registrarse en el <i>frontend</i> donde puede interactuar e ingresar información en los módulos que necesitan inicio de sesión como:</p>	

<ul style="list-style-type: none"> • Eventos • Donaciones • Juegos (Cuestionarios)
<p>Observación:</p> <p>El usuario invitado no registrado puede interactuar con el <i>frontend</i> únicamente para visualizar información general.</p>

TABLA XIX: Historia de usuario 10 - Consumir un *endpoint* para realizar donaciones

HISTORIA DE USUARIO	
Identificador (ID): HU0010	Usuario: Miembro de la Iglesia
Nombre Historia: Consumir un <i>endpoint</i> para realizar donaciones	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 2	
Responsable (es): Nathaly Bermeo	
<p>Descripción: El usuario miembro de la Iglesia puede consumir un <i>endpoint</i> para hacer donaciones de todo tipo mediante un formulario, donde se especifica:</p> <ul style="list-style-type: none"> • Nombre • Apellido • Tipo de donación • Descripción de donación • Tipo de entrega • Dirección • Fecha de entrega <p>El <i>frontend</i> muestra mensajes de error cuando se presente problemas en la validación de los diferentes campos, en caso de que el formulario no presente problemas de validación se puede confirmar la donación y enviar.</p> <p>El usuario miembro de la iglesia puede visualizar sus donaciones en el módulo "Mis donaciones".</p> <p>Además, el formulario tiene la opción para subir una fotografía que verifique la donación dependiendo del tipo de donación realizada.</p>	
<p>Observación:</p> <p>El usuario miembro de la Iglesia puede visualizar sus donaciones realizadas mediante el inicio de sesión.</p>	

El usuario miembro de la Iglesia puede verificar el envío de la donación mediante un comentario realizado por el usuario administrador.

TABLA XXI: Historia de usuario 11 - Consumir *endpoints* para visualizar, registrarse y eliminar eventos

HISTORIA DE USUARIO	
Identificador (ID): HU0011	Usuario: Miembro de la Iglesia
Nombre Historia: Consumir <i>endpoints</i> para visualizar, registrarse y eliminar eventos.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 2	
Responsable (es): Nathaly Bermeo	
Descripción: El usuario miembro de la Iglesia puede consumir varios <i>endpoints</i> para visualizar, registrarse y eliminar su inscripción de los eventos creados por la Iglesia. El usuario miembro de la Iglesia puede registrarse a un evento dando clic al botón de "Inscribirse" de cada evento, puede visualizar sus eventos inscritos en el módulo de "Mis eventos". En caso de eliminar su inscripción para el evento puede realizarlo mediante la opción eliminar que se encuentra en cada evento.	
Observación: El usuario miembro de la Iglesia puede visualizar, registrarse y eliminar su inscripción al evento creado por la Iglesia mediante el inicio de sesión.	

TABLA XXI: Historia de usuario 12 - Consumir varios *endpoints* para visualizar y realizar cuestionarios

HISTORIA DE USUARIO	
Identificador (ID): HU0012	Usuario: Miembro de la Iglesia
Nombre Historia: Consumir varios <i>endpoints</i> para visualizar y realizar cuestionarios	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 2	
Responsable (es): Nathaly Bermeo	
Descripción: El usuario miembro de la Iglesia puede consumir varios <i>endpoints</i> para visualizar y realizar cuestionarios de cualquier tema religioso.	

Además, el usuario miembro de la Iglesia puede visualizar una puntuación asignada cada vez que culmine un cuestionario.

Observación:

El usuario miembro de la Iglesia puede realizar el cuestionario creado por la Iglesia mediante el inicio de sesión.

Product Backlog

La **TABLA XXII** enumera la prioridad de cada requisito que se ha implementado en el *frontend*. Estos requisitos se clasifican de acuerdo con las necesidades del dueño del producto y la complejidad del desarrollo.

TABLA XXII: Product Backlog

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU003	Consumir varios <i>endpoints</i> para ingresar, visualizar, modificar y eliminar información de la Iglesia.	1	Terminado	Media
HU004	Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar eventos.	1	Terminado	Media
HU005	Consumir un <i>endpoint</i> para visualizar los miembros inscritos en un evento.	1	Terminado	Media
HU006	Consumir un <i>endpoint</i> para visualizar las donaciones realizadas.	1	Terminado	Media
HU007	Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar vídeos.	2	Terminado	Alta
HU008	Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar cuestionarios.	2	Terminado	Alta
HU009	Consumir un <i>endpoint</i> para registrarse en el <i>frontend</i> .	2	Terminado	Alta

HU010	Consumir un <i>endpoint</i> para realizar donaciones.	2	Terminado	Media
HU011	Consumir <i>endpoints</i> para visualizar, registrarse y eliminar eventos.	2	Terminado	Media
HU012	Consumir varios <i>endpoints</i> para visualizar y realizar cuestionarios	2	Terminado	Media

Sprint Backlog

La **TABLA XXIII** presenta los cinco *Sprints* en los que se ha desarrollado el *frontend*, listando las actividades y el tiempo determinado para cumplir con los entregables que se han establecido con el dueño del producto.

TABLA XXIII: *Sprint Backlog*

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
ID – SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB001	Diseño e implementación de la interfaz de los usuarios administrador e invitado y módulos a su cargo	Sección informativa	HU001	Consumir varios <i>endpoints</i> para presentar la información general de la Iglesia.	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para la página informativa. 	70H
		Módulo inicio de sesión	HU002	Consumir varios <i>endpoints</i> para iniciar y cerrar sesión	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para el acceso al <i>frontend</i>. • Diseño e implementación de interfaz de inicio y cierre de sesión del usuario. • Verificación de credenciales del miembro de la Iglesia y del administrador. • Carga de módulos asignados para el miembro de la Iglesia. 	

		Sección informativa	HU003	Consumir varios <i>endpoints</i> para ingresar, visualizar, modificar y eliminar información de la Iglesia	<ul style="list-style-type: none"> • Diseño e implementación de módulos. • Diseño e implementación de un formulario para ingresar, visualizar, modificar y eliminar información. • Guardar formulario. • Verificar los campos requeridos del formulario. • Prueba para ingresar, visualizar, modificar y eliminar información de la Iglesia. 	
		Módulo eventos	HU004	Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar eventos	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para eventos. • Diseño e implementación de un formulario para registrar, visualizar, modificar y eliminar un evento. • Verificar campos requeridos del formulario. • Guardar formulario 	

		Módulo eventos	HU005	Consumir un <i>endpoint</i> para visualizar los miembros inscritos en un evento	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para visualizar los miembros inscritos en un evento. • Listar miembros inscritos en un evento. • Prueba para visualizar los miembros inscritos en un evento. 	
		Módulo donaciones	HU006	Consumir un <i>endpoint</i> para visualizar las donaciones realizadas	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para visualizar las donaciones realizadas. • Listar donaciones. • Visualizar una donación en específico. 	
		Sección informativa	HU007	Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar vídeos	<ul style="list-style-type: none"> • Diseño e implementación de un formulario para registrar, visualizar, modificar y eliminar vídeos. • Verificar campos requeridos del formulario. • Guardar formulario 	

		Módulo cuestionarios	HU008	Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar cuestionarios	<ul style="list-style-type: none"> • Diseño e implementación de un formulario para registrar, visualizar, modificar y eliminar cuestionarios. • Verificar campos requeridos del formulario. • Guardar formulario 	
		Módulo registro	HU009	Consumir un <i>endpoint</i> para registrarse en el <i>frontend</i>	<ul style="list-style-type: none"> • Diseño e implementación de registro para el usuario invitado. • Validación de campos requeridos. • Verificación que el registro sea único. • Guardar registro. 	
SB002	Diseño e implementación de la interfaz del usuario miembro de la Iglesia y módulos a su cargo	Módulo Donaciones	HU0010	Consumir un <i>endpoint</i> para realizar donaciones	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para cada tipo de donación. • Visualizar formulario para cada tipo de donación. • Validación de campos requeridos. 	70H
		Módulo Eventos	HU0011	Consumir <i>endpoints</i> para visualizar, registrarse y eliminar eventos	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para eventos. • Visualizar eventos. 	

					<ul style="list-style-type: none"> • Visualizar botón para inscribirse a un evento. 	
		Módulo Cuestionarios	HU0012	Consumir varios <i>endpoints</i> para visualizar y realizar cuestionarios	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para cuestionarios. • Visualizar cuestionarios. • Validación de campos requeridos. • Guardar cuestionario. 	
SB003	Pruebas en el sistema <i>web</i>	<ul style="list-style-type: none"> • Pruebas unitarias. • Prueba de compatibilidad. • Prueba de aceptación. 				20H
SB004	Despliegue del sistema <i>web</i>	<ul style="list-style-type: none"> • Despliegue del sistema <i>web</i> en <i>Vercel</i>. 				10H
Documentación		<ul style="list-style-type: none"> • Informe Técnico. • Anexos. 				50H
TOTAL						240 H

Diseño de interfaces

A continuación, se muestran los prototipos de cada uno de los módulos del *frontend*, donde se puede observar el diseño de cada una de las interfaces. Por esta razón, desde la **Fig. 26** a la **Fig. 47** se muestran cada una de las interfaces del *frontend*.



Fig. 26: Diseño de interfaz – Página Información general

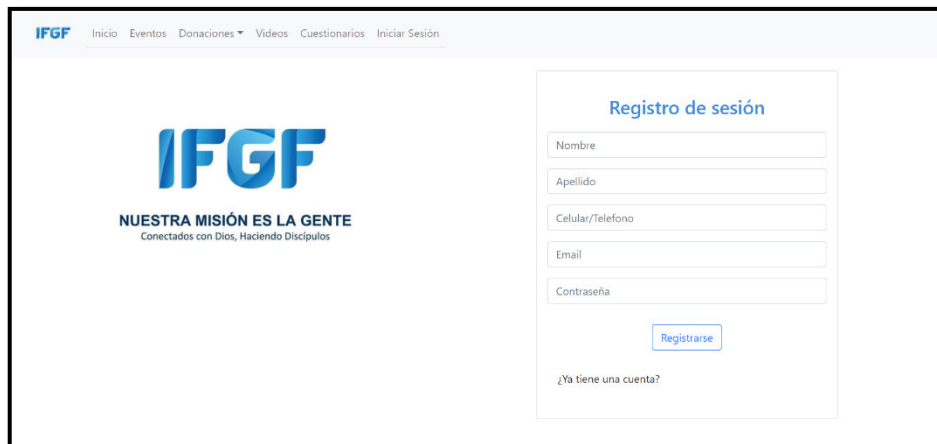


Fig. 27: Diseño de interfaz - Registro de usuarios



Fig. 28: Diseño de interfaz – Inicio de sesión

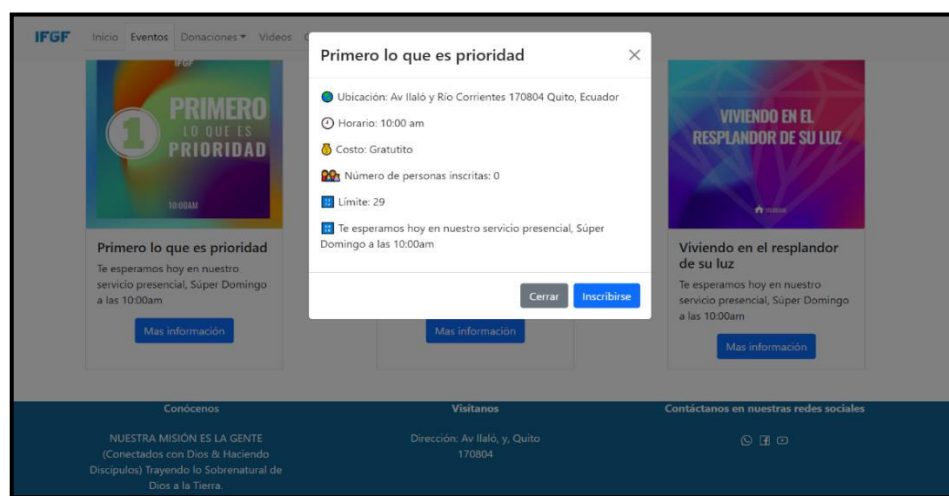


Fig. 29: Diseño de interfaz – Módulo para visualizar la información de cada evento

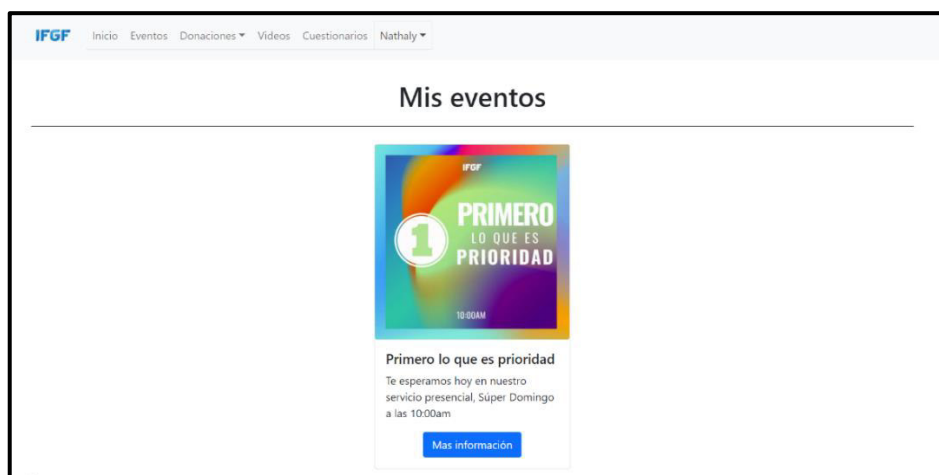


Fig. 30: Diseño de interfaz - Eventos inscritos del usuario miembro de la Iglesia

Fig. 31: Diseño de interfaz – Formulario para realizar donación de ropa

Fig. 32: Diseño de interfaz – Formulario para realizar donación de comida

Fig. 33: Diseño de interfaz – Formulario para realizar donación de dinero

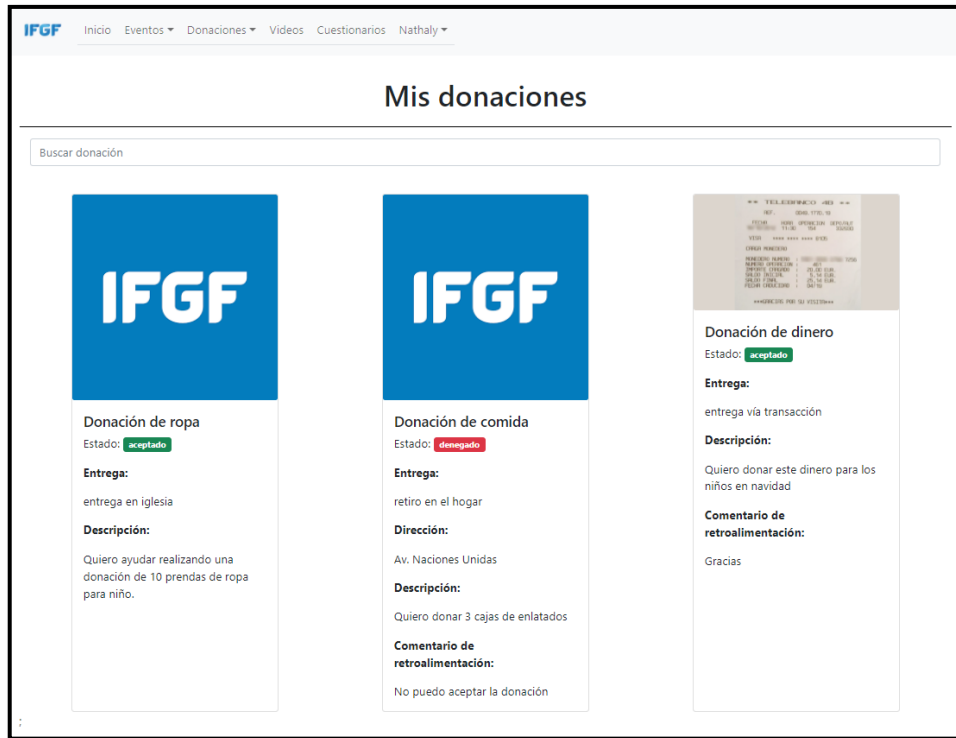


Fig. 34: Diseño de interfaz – Donaciones del usuario miembro de la Iglesia

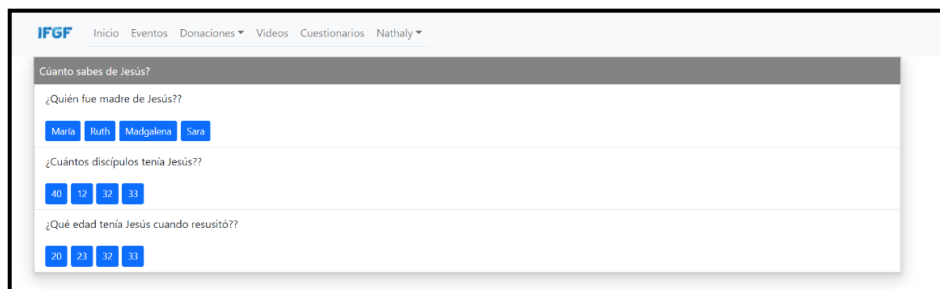


Fig. 35: Diseño de interfaz - Preguntas de un cuestionario

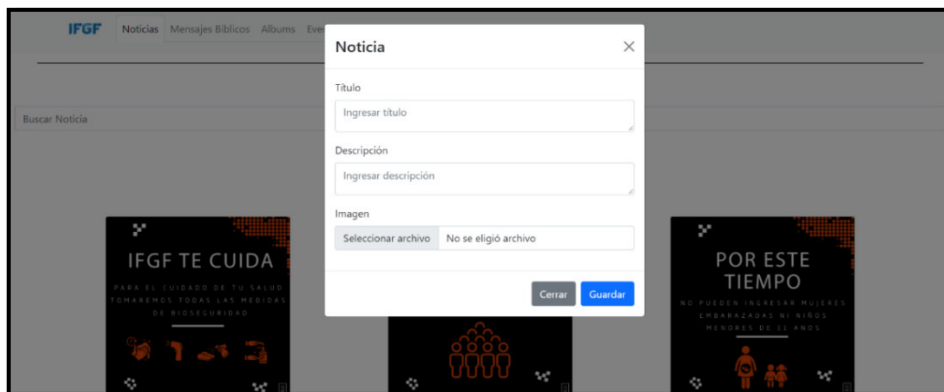


Fig. 36: Diseño de interfaz - Formulario para crear noticias

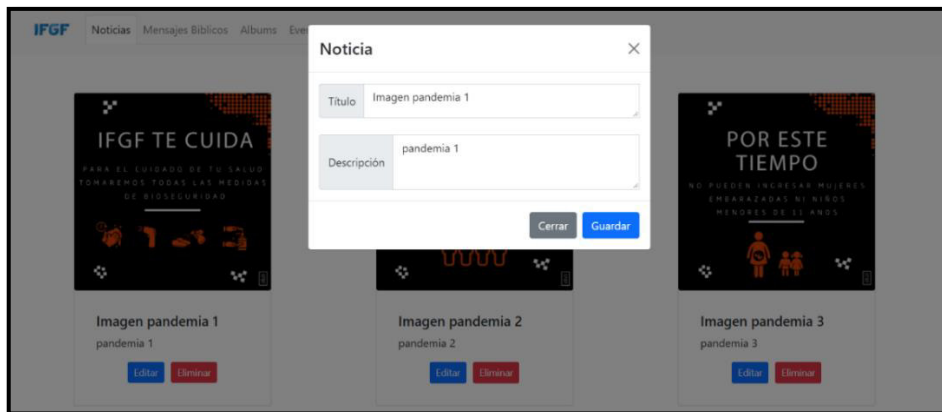


Fig. 37: Diseño de interfaz – Formulario para editar noticias

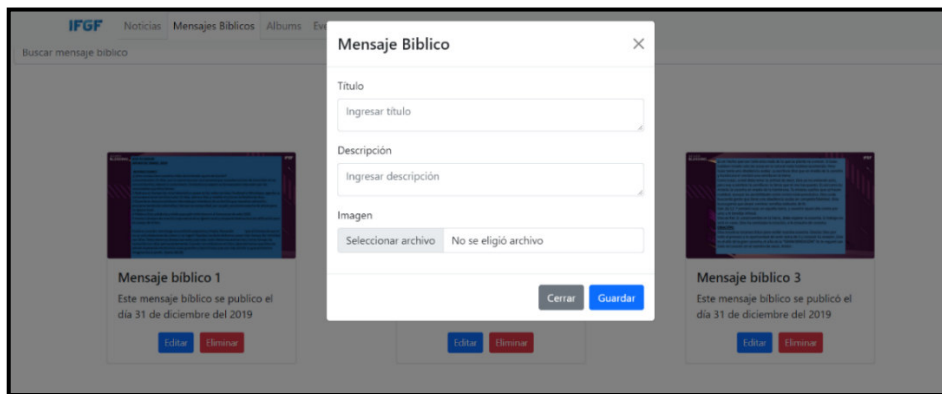


Fig. 38: Diseño de interfaz - Formulario para crear mensajes bíblicos

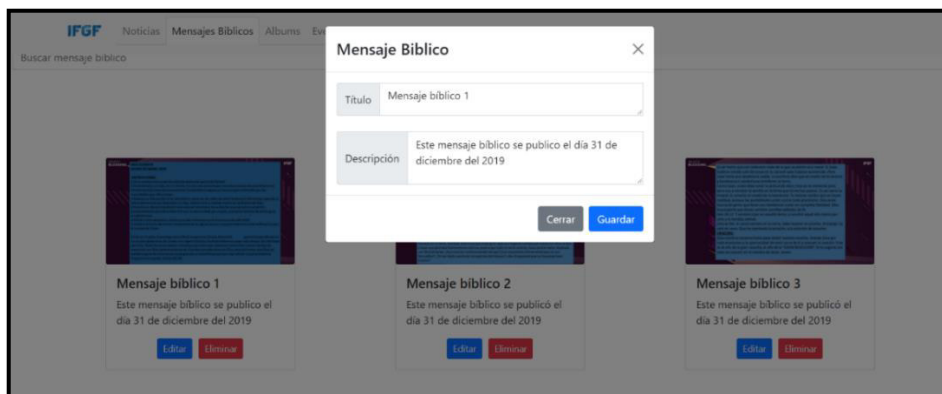


Fig. 39: Diseño de interfaz – Formulario para editar mensajes bíblicos

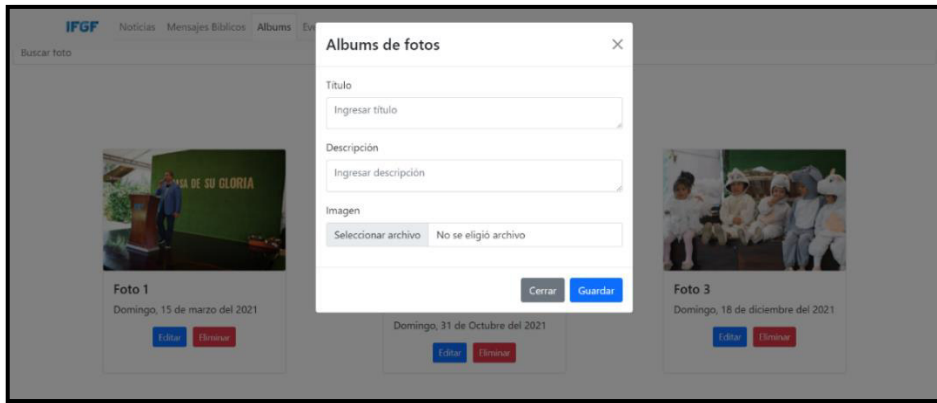


Fig. 40: Diseño de interfaz – Formulario para crear fotos

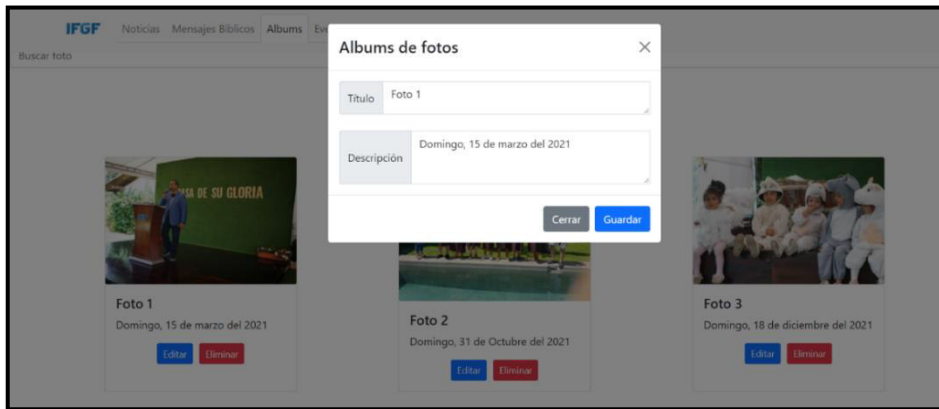


Fig. 41: Diseño de interfaz – Formulario para editar fotos

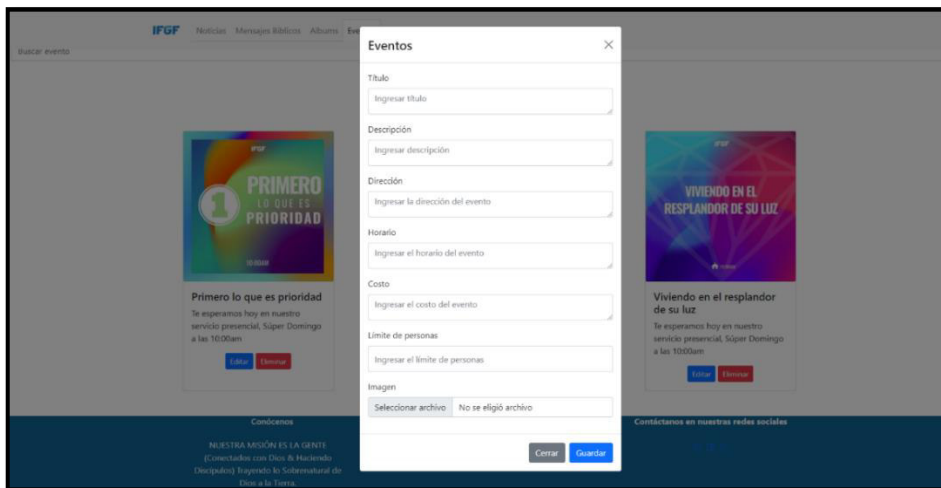


Fig. 42: Diseño de interfaz – Formulario para crear evento

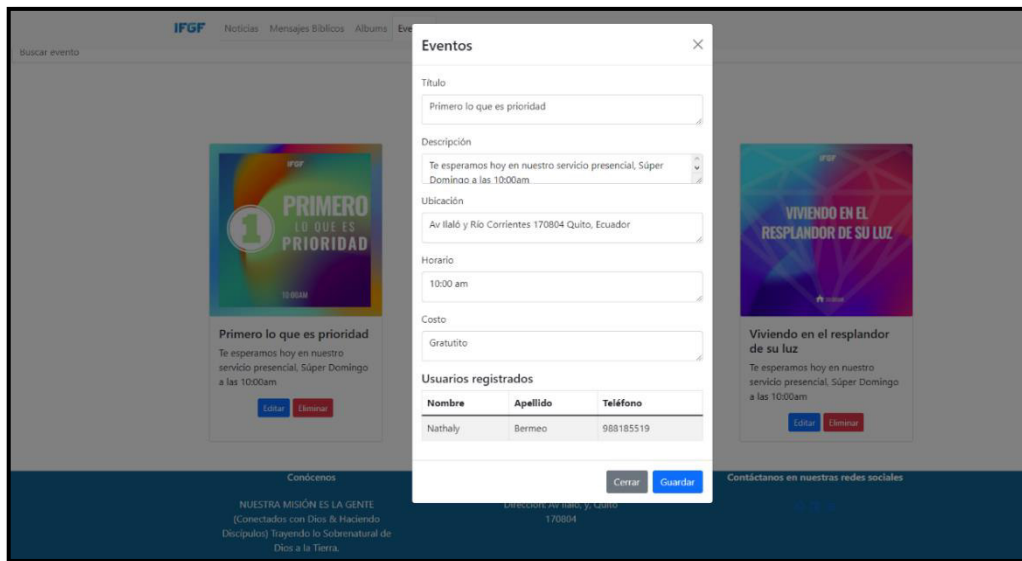


Fig. 43: Diseño de interfaz – Formulario para editar evento

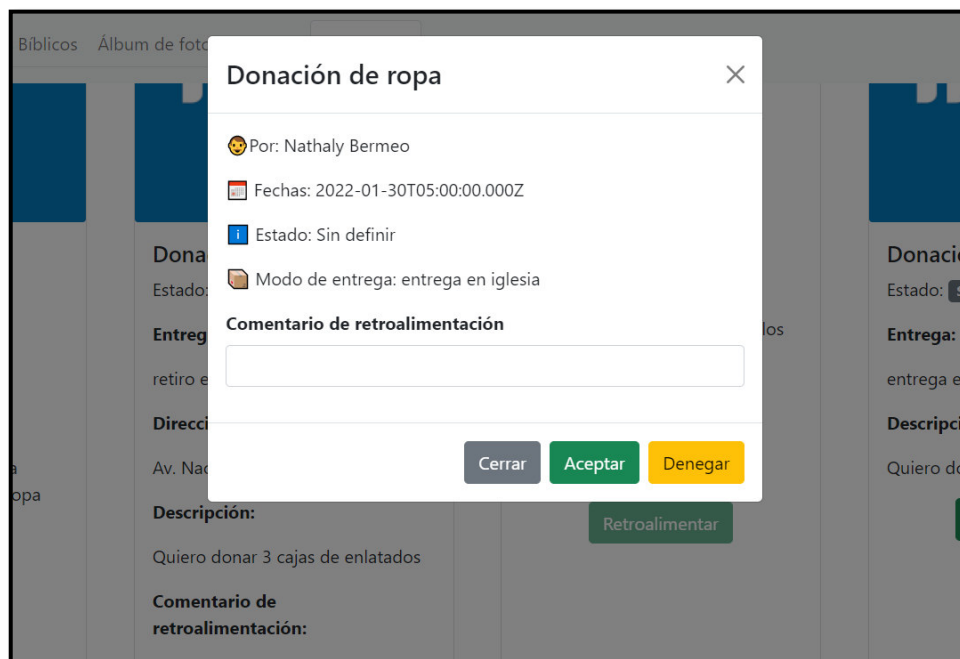


Fig. 44: Diseño de interfaz – Modal para realizar comentarios a las donaciones de los usuarios miembros de la Iglesia

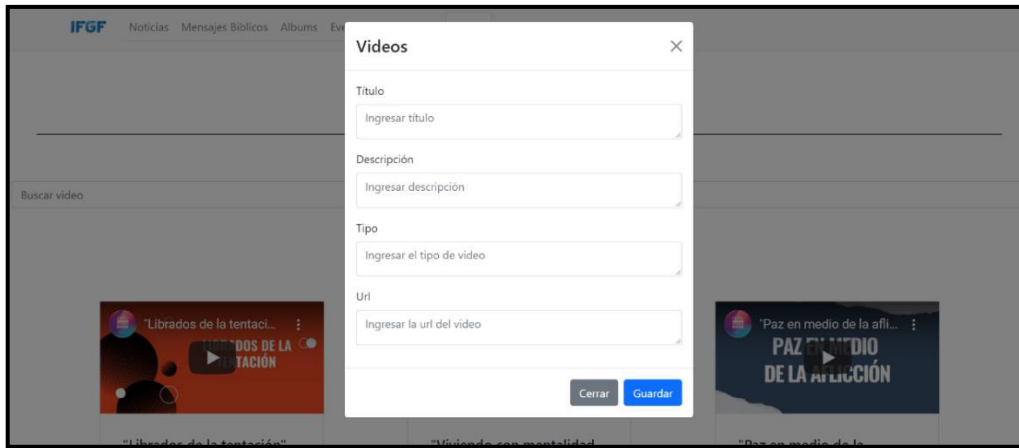


Fig. 45: Diseño de interfaz – Formulario para crear videos

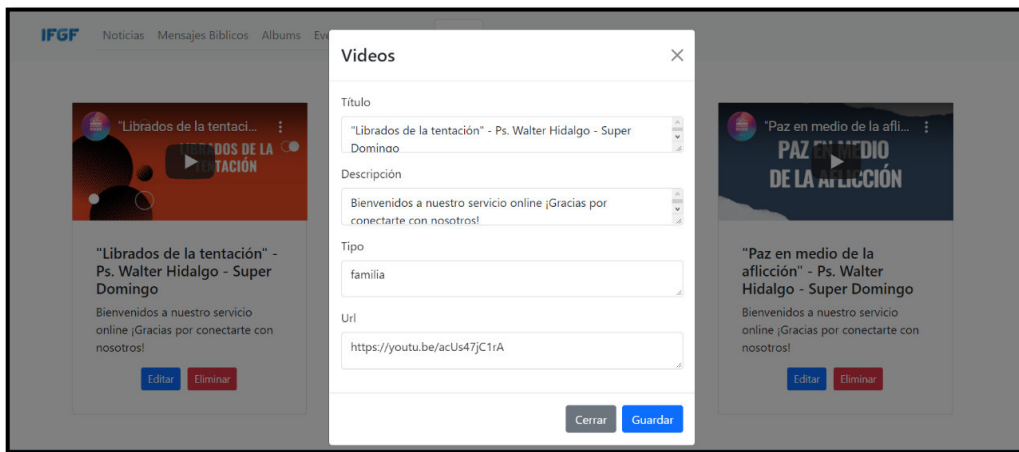


Fig. 46: Diseño de interfaz – Formulario para editar videos

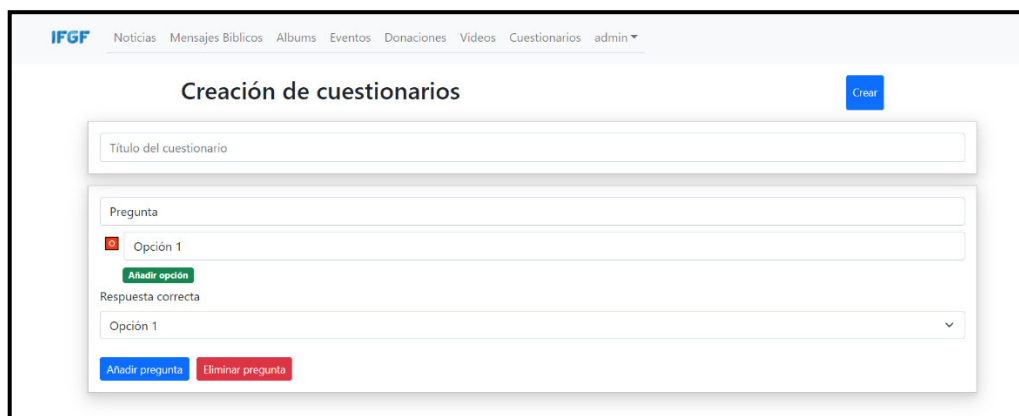


Fig. 47: Diseño de interfaz – Formulario para la creación de cuestionarios

Pruebas

Terminada la etapa de codificación se ha desarrollado la ejecución de pruebas unitarias, de compatibilidad y de aceptación para comprobar la calidad del código del *frontend* y cada uno de los módulos respectivamente.

Pruebas unitarias

La **Fig. 48** muestra el comando de *Jest* para la ejecución de las pruebas unitarias de algunas funciones del *frontend*, indicando si la ejecución de la respectiva prueba paso el *testing*, las mismas que se muestran desde la **Fig. 49** a la **Fig. 56**.

```
PS C:\Users\Nathaly\frontprincipal> npm run test  
  
> test  
> jest --watch
```

Fig. 48: Comando para realizar pruebas unitarias

```
import { renderHook, act } from '@testing-library/react-hooks'  
import { useMessages } from '../src/hooks/useMessages'  
  
test('should use useMessages', () => {  
  const { result } = renderHook( callback () => useMessages()  
  const newMessages = [{title: 'Este es un mensaje biblico', description: ''}]  
  result.current.setMessages(newMessages)  
  expect(result.current.messages.length).toBe(1)  
  expect(result.current.messages).toEqual(newMessages)  
})
```

Fig. 49: Prueba unitaria #2

```
PASS  __tests__/index.test.jsx  
  ✓ should use useMessages (119 ms)  
  
Test Suites: 1 passed, 1 total  
Tests:      1 passed, 1 total  
Snapshots: 0 total  
Time:       2.925 s  
Ran all test suites related to changed files.
```

Fig. 50: Resultado de la prueba #2


```

import { renderHook, act } from '@testing-library/react-hooks'
import { useAlbums } from '../src/hooks/useAlbums'

test('should use useAlbums', () => {
  const { result } = renderHook( callback: () => useAlbums())
  const newAlbums = [{title: 'Este es un album', description: ''}]
  result.current.setAlbums(newAlbums)
  expect(result.current.albums.length).toBe(1)
  expect(result.current.albums).toEqual(newAlbums)
})

```

Fig. 51: Prueba unitaria #3

```

PASS  __tests__/index.test.jsx
  ✓ should use useAlbums (111 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        3.064 s
Ran all test suites related to changed files.

```

Fig. 52: Resultado de la prueba #3

```

import { renderHook, act } from '@testing-library/react-hooks'
import { useEvents } from '../src/hooks/useEvents'

test('should use useEvents', () => {
  const { result } = renderHook( callback: () => useEvents())
  const newEvents = [{title: 'Esta es un evento', description: ''}]
  result.current.setEvents(newEvents)
  expect(result.current.events.length).toBe(1)
  expect(result.current.events).toEqual(newEvents)
})

```

Fig. 53: Prueba unitaria #4

```

PASS  __tests__/index.test.jsx
  ✓ should use useEvents (122 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        3.012 s
Ran all test suites related to changed files.

```

Fig. 54: Resultado de la prueba #4

```
import { renderHook, act } from '@testing-library/react-hooks'
import {useDonation} from '../src/hooks/useDonation'

test('should use useDonation', () => {
  const { result } = renderHook( callback () => useDonation())
  const newDonation = [{title: 'Esta es una donación', description: ''}]
  result.current.setDonations(newDonation)
  expect(result.current.donations.length).toBe(1)
  expect(result.current.donations).toEqual(newDonation)
})
```

Fig. 55: Prueba unitaria #5

```
PASS  __tests__/index.test.jsx
  ✓ should use useDonation (113 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.912 s
Ran all test suites related to changed files.
```

Fig. 56: Resultado de la prueba #5

Pruebas de compatibilidad

En este tipo de pruebas, se procede a verificar la funcionalidad del *frontend* en diferentes navegadores con el fin de observar posibles errores y la presentación de la información. En ese sentido, desde la **Fig. 57** hasta la **Fig. 65** se muestra los resultados que se han obtenido en cada uno de los navegadores *web*.

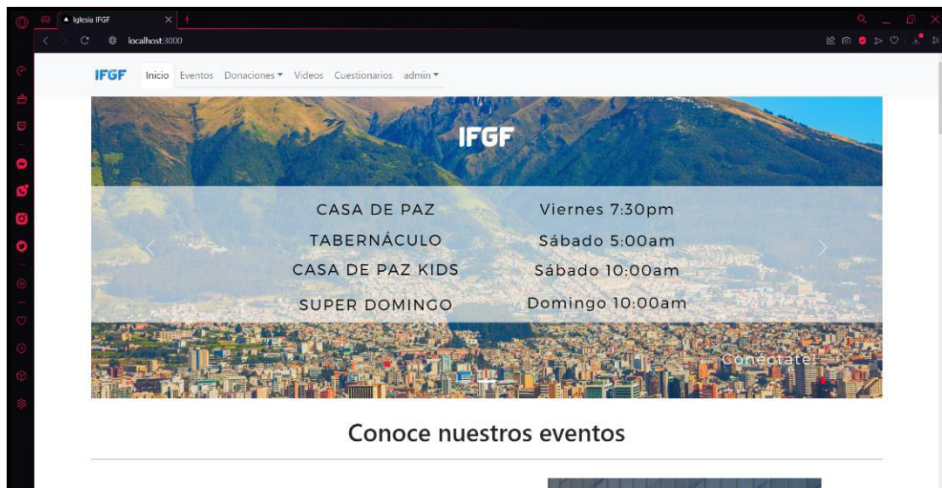


Fig. 57: Página con información general vista desde el navegador Opera

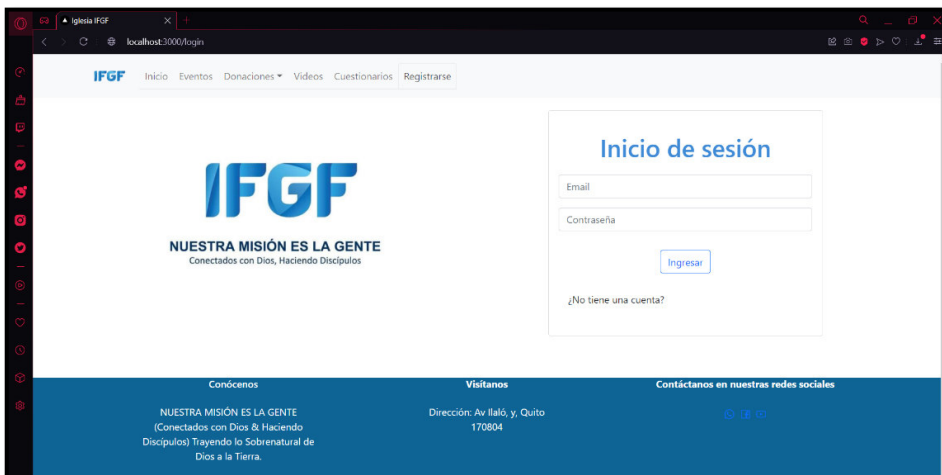


Fig. 58: Página de inicio de sesión vista desde el navegador Opera

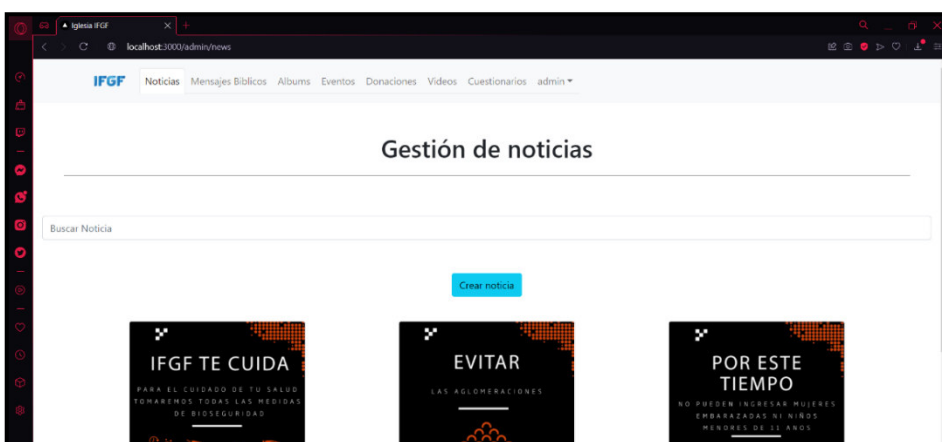


Fig. 59: Página de gestión de noticias vista desde el navegador Opera

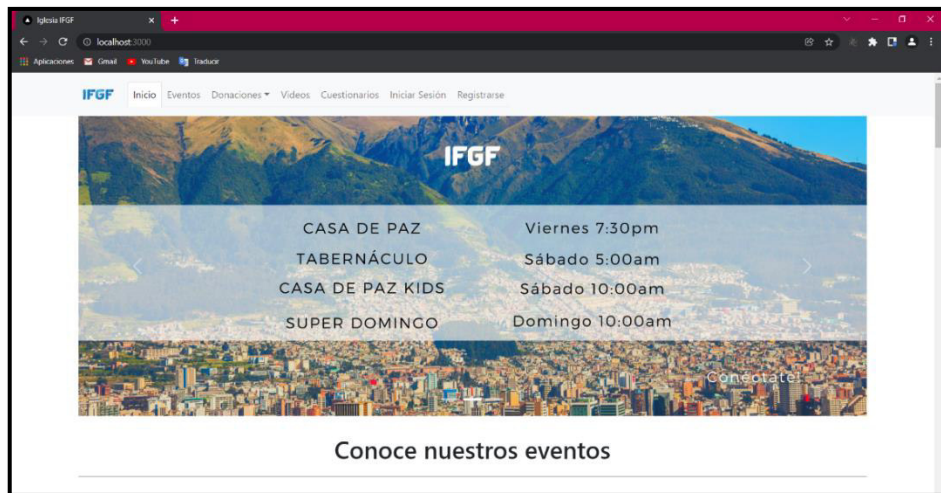


Fig. 60: Página con información general vista desde el navegador Google Chrome

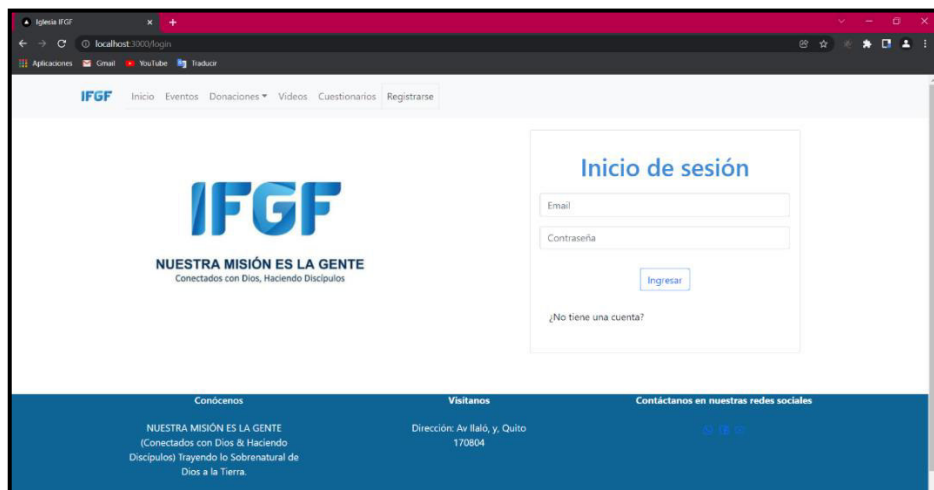


Fig. 61: Página de inicio de sesión vista desde el navegador Google Chrome

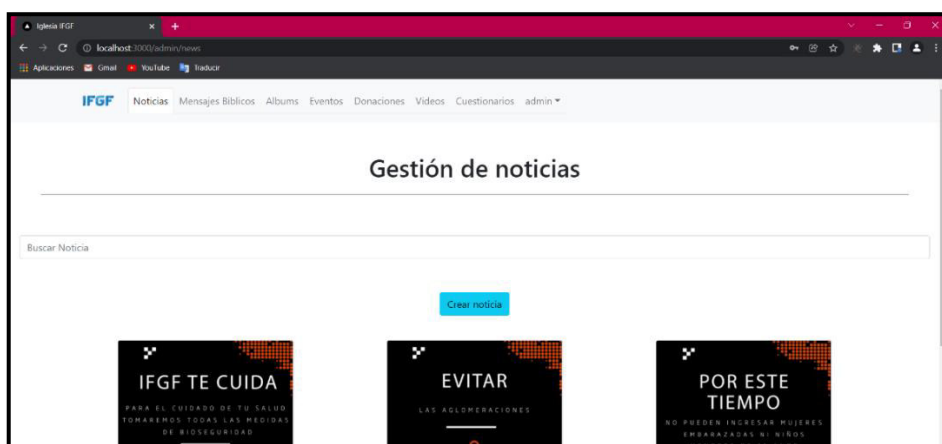


Fig. 62: Página de gestión de noticias vista desde el navegador Google Chrome

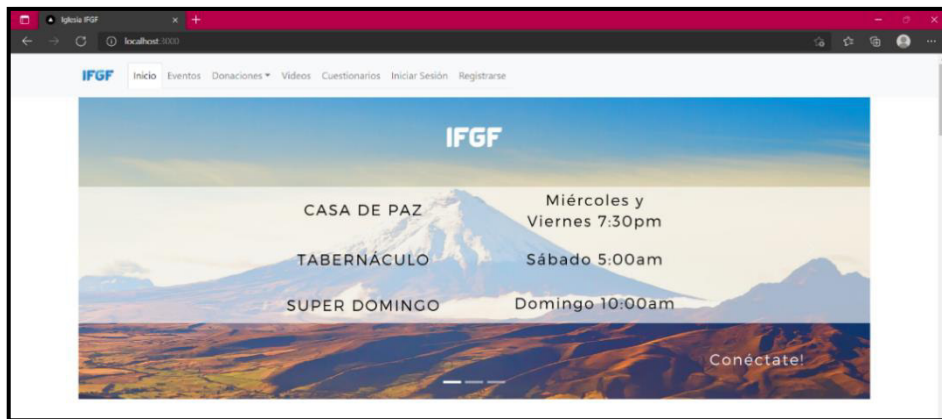


Fig. 63: Página con información general vista desde el navegador *Microsoft Edge*

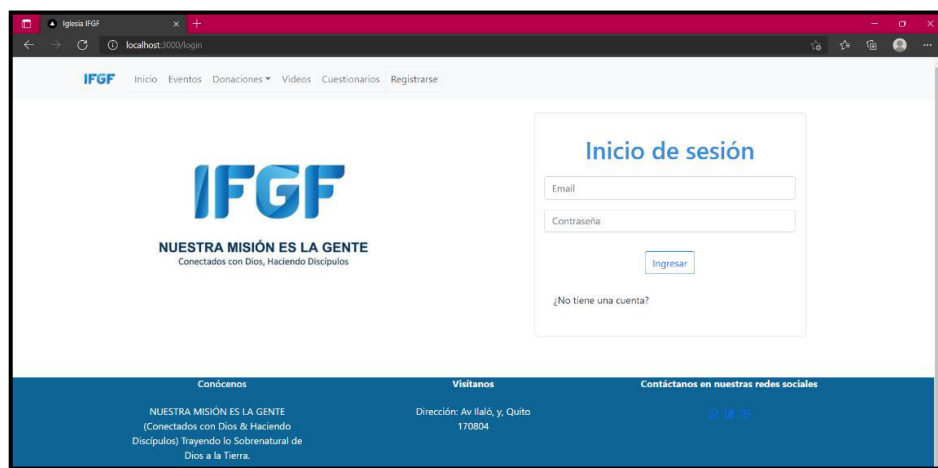


Fig. 64: Página de inicio de sesión vista desde el navegador *Microsoft Edge*

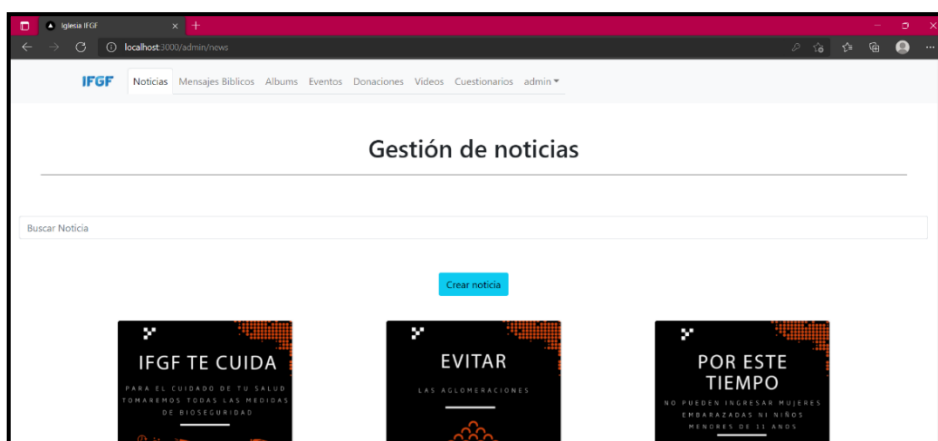


Fig. 65: Página de gestión de noticias vista desde el navegador *Microsoft Edge*

Pruebas de aceptación

A continuación, se muestran las 11 Pruebas de aceptación que van desde la **TABLA XXIV** hasta **TABLA XXXIV**. Además, cada prueba describe el procedimiento que se le ha asignado a los diferentes tipos de usuario para la correcta ejecución, comprobación y a su vez la aprobación de cada una.

TABLA XXIV: Prueba de aceptación 2 - Consumir varios *endpoints* para iniciar y cerrar sesión.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA002	Identificador historia de Usuario: HU002
Nombre: Consumir varios <i>endpoints</i> para iniciar y cerrar sesión.	
Descripción: El usuario miembro de la Iglesia puede consumir varios <i>endpoints</i> para iniciar sesión ya que es indispensable para interactuar con ciertos módulos e ingresar información. Por otra parte, el usuario administrador necesita consumir varios <i>endpoints</i> para iniciar sesión y gestionar los módulos asignados a su cargo. Además, el usuario administrador y miembro de la Iglesia pueden dejar de interactuar con el <i>frontend</i> mediante el cierre de sesión.	
Pasos de ejecución: Para iniciar sesión: <ul style="list-style-type: none">• Ingresar a la URL del sistema <i>web</i> en el navegador.• En el menú superior dar clic en el botón “Iniciar Sesión”.• Digitar su correo electrónico y contraseña.• Dar clic en “Ingresar”. Para cerrar sesión: <ul style="list-style-type: none">• Iniciar sesión en el sistema <i>web</i>.• Dar clic en el nombre de usuario ubicado en el menú superior, a continuación, se despliega la opción de “Cerrar Sesión”.	
Resultado deseado: El <i>frontend</i> permite iniciar y cerrar sesión del usuario con perfil miembro de la Iglesia y administrador.	
Evaluación de la prueba:	

Se comprueba el resultado esperado.

Aprobación del cliente 100%.

TABLA XXV: Prueba de aceptación 3 - Consumir varios *endpoints* para ingresar, visualizar, modificar y eliminar información de la Iglesia.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA003	Identificador historia de Usuario: HU003
Nombre: Consumir varios <i>endpoints</i> para ingresar, visualizar, modificar y eliminar información de la Iglesia.	
Descripción: El usuario administrador en el <i>frontend</i> puede consumir varios <i>endpoints</i> para crear, visualizar, modificar y eliminar información respecto a: <ul style="list-style-type: none">• Noticias• Mensajes bíblicos• Álbum de fotos El administrador debe realizar el siguiente formulario: <ul style="list-style-type: none">• Título• Descripción• Imagen	
Pasos de ejecución: <ul style="list-style-type: none">• Iniciar sesión con perfil administrador• En el menú superior dar clic en el botón “Noticias”.• Dar clic en crear noticia• Llenar el formulario• Dar clic en Guardar• Dar clic en Editar• Editar el formulario• Dar clic en Guardar• Dar clic en Eliminar• En el menú superior dar clic en el botón “Mensajes bíblicos”.• Dar clic en crear mensaje bíblico• Llenar el formulario	

<ul style="list-style-type: none"> • Dar clic en Guardar • Dar clic en Editar • Editar el formulario • Dar clic en Guardar • Dar clic en Eliminar • En el menú superior dar clic en el botón “Álbum de fotos”. • Dar clic en crear foto • Llenar el formulario • Dar clic en Guardar • Dar clic en Editar • Editar el formulario • Dar clic en Guardar • Dar clic en Eliminar
<p>Resultado deseado:</p> <p>El <i>frontend</i> permite gestionar la información general de la Iglesia a través del perfil administrador.</p>
<p>Evaluación de la prueba:</p> <p>Se comprueba el resultado esperado.</p> <p>Aprobación del cliente 100%.</p>

TABLA XXVI: Prueba de aceptación 4 - Consumir varios *endpoints* para registrar, visualizar, modificar y eliminar eventos.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA004	Identificador historia de Usuario: HU004
Nombre: Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar eventos.	
Descripción: El usuario administrador en el sistema <i>web</i> puede consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar eventos cuando el usuario lo requiera.	

El registro y modificación de un evento se realiza mediante un formulario donde se ingresa la siguiente información:

- Nombre
- Descripción
- Ubicación
- Imagen
- Horario
- Costo
- Aforo permitido
- En caso de eliminar un evento existe la opción requerida.

Pasos de ejecución:

- Iniciar sesión con perfil administrador
- En el menú superior dar clic en el botón “Eventos”
- Dar clic en Crear evento
- Llenar formulario de eventos
- Dar clic en Guardar
- Dar clic en Editar
- Editar el formulario de eventos
- Dar clic en Guardar
- Dar clic en Eliminar

Resultado deseado:

El *frontend* permite gestionar la información de eventos de la Iglesia a través del perfil administrador.

Evaluación de la prueba:

Se comprueba el resultado esperado.

Aprobación del cliente 100%.

TABLA XXVII: Prueba de aceptación 5 - Consumir un *endpoint* para visualizar los miembros inscritos en un evento.

PRUEBA DE ACEPTACIÓN

Identificador (ID): PA005	Identificador historia de Usuario: HU005
Nombre: Consumir un <i>endpoint</i> para visualizar los miembros inscritos en un evento.	
Descripción: El usuario administrador en el <i>frontend</i> puede consumir un <i>endpoint</i> para visualizar a los miembros inscritos a un evento donde puede observar el nombre, apellido y la confirmación de la inscripción por parte del miembro de la Iglesia.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Iniciar sesión con perfil administrador • En el menú superior dar clic en “Eventos” • Dar clic en Ver más 	
Resultado deseado:	
El <i>frontend</i> permite visualizar la información de usuarios inscritos en los eventos de la Iglesia a través del perfil administrador.	
Evaluación de la prueba:	
Se comprueba el resultado esperado.	
Aprobación del cliente 100%.	

TABLA XXVIII: Prueba de aceptación - Consumir un *endpoint* para visualizar las donaciones realizadas

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA006	Identificador historia de Usuario: HU006
Nombre: Consumir un <i>endpoint</i> para visualizar las donaciones realizadas.	
Descripción: El usuario administrador en el <i>frontend</i> puede consumir un <i>endpoint</i> para visualizar las donaciones realizadas donde se observa el nombre, apellido y tipo de donación realizada por parte del usuario miembro de la Iglesia. Según el tipo de donación por parte del usuario miembro de la Iglesia, el usuario administrador puede visualizar una fotografía verificando la donación realizada.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Iniciar sesión con perfil administrador • En el menú superior dar clic en “Donaciones” 	

<p>Resultado deseado:</p> <p>El <i>frontend</i> permite visualizar la información de donaciones realizadas por usuarios miembro de la Iglesia a través del perfil administrador.</p>
<p>Evaluación de la prueba:</p> <p>Se comprueba el resultado esperado.</p> <p>Aprobación del cliente 100%.</p>

TABLA XXIX : Prueba de aceptación 8 - Consumir varios *endpoints* para registrar, visualizar, modificar y eliminar videos

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA007	Identificador historia de Usuario: HU007
<p>Nombre: Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar videos.</p>	
<p>Descripción: El usuario administrador puede consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar videos según lo requiera.</p> <p>El administrador debe realizar el siguiente formulario:</p> <ul style="list-style-type: none"> • Titulo • Descripción • Tipo • Enlace url 	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Iniciar sesión con perfil administrador • En el menú superior dar clic en el botón “Videos” • Dar clic en Crear video • Llenar formulario • Dar clic en Guardar • Dar clic en Editar • Editar formulario • Dar clic en Guardar • Dar clic en Eliminar 	

<p>Resultado deseado:</p> <p>El <i>frontend</i> permite gestionar los videos publicados para la visualización de los miembros de la Iglesia a través del perfil administrador.</p>
<p>Evaluación de la prueba:</p> <p>Se comprueba el resultado esperado.</p> <p>Aprobación del cliente 100%.</p>

TABLA XXX : Prueba de aceptación 8 - Consumir varios *endpoints* para registrar, visualizar, modificar y eliminar cuestionarios

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA008	Identificador historia de Usuario: HU008
<p>Nombre: Consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar cuestionarios.</p>	
<p>Descripción: El usuario administrador puede consumir varios <i>endpoints</i> para registrar, visualizar, modificar y eliminar cuestionarios.</p> <p>El administrador debe realizar el siguiente formulario:</p> <ul style="list-style-type: none"> • Tema • Preguntas • Opciones <p>Estos cuestionarios son realizados en base a diferentes temas religiosos para mejorar el aprendizaje de los miembros de la Iglesia.</p> <p>Además, el usuario administrador puede consumir un <i>endpoint</i> del backend para visualizar el puntaje obtenido por cada miembro de la Iglesia.</p>	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Iniciar sesión con perfil administrador • En el menú superior dar clic en el botón “Cuestionarios” • Dar clic en Crear cuestionario • Llenar formulario • Dar clic en Guardar • Dar clic en Editar 	

<ul style="list-style-type: none"> • Editar formulario • Dar clic en Guardar • Dar clic en Eliminar • Dar clic en Ver usuarios
<p>Resultado deseado:</p> <p>El <i>frontend</i> permite gestionar los cuestionarios publicados para la interacción de los miembros de la Iglesia a través del perfil administrador.</p>
<p>Evaluación de la prueba:</p> <p>Se comprueba el resultado esperado.</p> <p>Aprobación del cliente 100%.</p>

TABLA XXXI: Prueba de aceptación 9 - Consumir un *endpoint* para registrarse en el *frontend*

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA009	Identificador historia de Usuario: HU009
Nombre: Consumir un <i>endpoint</i> para registrarse en el <i>frontend</i>	
<p>Descripción: El usuario invitado debe consumir un <i>endpoint</i> para registrarse en el <i>frontend</i> donde puede interactuar e ingresar información en los módulos que necesitan inicio de sesión como:</p> <ul style="list-style-type: none"> • Eventos • Donaciones • Cuestionarios 	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ingresar a la URL del sistema <i>web</i> en el navegador. • En el menú superior dar clic en el botón “Registrarse” • Llenar los campos de nombre, apellido, celular, email y contraseña • Dar clic en Registrarse 	
Resultado deseado:	

El *frontend* permite registrar usuarios para la interacción de cada una de las páginas con sus respectivas actividades.

Evaluación de la prueba:

Se comprueba el resultado esperado.

Aprobación del cliente 100%.

TABLA XXXII : Prueba de aceptación 10 - Consumir un *endpoint* para realizar donaciones

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0010	Identificador historia de Usuario: HU0010
Nombre: Consumir un <i>endpoint</i> para realizar donaciones	
Descripción: El usuario miembro de la Iglesia puede consumir un <i>endpoint</i> para hacer donaciones de todo tipo mediante un formulario, donde se especifica: <ul style="list-style-type: none">• Nombre• Apellido• Tipo de donación• Descripción de donación• Tipo de entrega• Dirección• Fecha de entrega <p>El <i>frontend</i> muestra mensajes de error cuando se presente problemas en la validación de los diferentes campos, en caso de que el formulario no presente problemas de validación se puede confirmar la donación y enviar.</p> <p>Además, el formulario tiene la opción para subir una fotografía que verifique la donación dependiendo del tipo de donación realizada.</p>	
Pasos de ejecución: <ul style="list-style-type: none">• Iniciar sesión como usuario miembro de la Iglesia• En el menú superior dar clic en el botón "Donaciones"• Seleccionar el tipo de donación• Dar clic en Donar• Llenar formulario	

<ul style="list-style-type: none"> • Dar clic en Guardar
<p>Resultado deseado:</p> <p>El <i>frontend</i> permite realizar 3 tipos de donaciones a los usuarios que son miembros de la Iglesia.</p>
<p>Evaluación de la prueba:</p> <p>Se comprueba el resultado esperado.</p> <p>Aprobación del cliente 100%.</p>

TABLA XXXIII: Prueba de aceptación 11 - Consumir *endpoints* para visualizar, registrarse y eliminar eventos

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0011	Identificador historia de Usuario: HU0011
Nombre: Consumir <i>endpoints</i> para visualizar y registrarse eventos.	
<p>Descripción: El usuario miembro de la Iglesia puede consumir varios <i>endpoints</i> para visualizar, registrarse y eliminar su inscripción de los eventos creados por la Iglesia.</p> <p>El usuario miembro de la Iglesia puede registrarse a un evento dando clic al botón de "Inscribirse" de cada evento, puede visualizar sus eventos inscritos en el módulo de "Mis eventos".</p> <p>En caso de eliminar su inscripción para el evento puede realizarlo mediante la opción eliminar que se encuentra en cada evento.</p>	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Iniciar sesión como usuario miembro de la Iglesia • En el menú superior dar clic en el botón "Eventos" • Dirigirse al evento de su interés y dar clic en Más información • Dar clic en Inscribirse 	
<p>Resultado deseado:</p> <p>El <i>frontend</i> permite realizar la inscripción de un evento del usuario miembro de la Iglesia.</p>	

Evaluación de la prueba:

Se comprueba el resultado esperado.

Aprobación del cliente 100%.

TABLA XXXIV: Prueba de aceptación 12 - Consumir varios *endpoints* para visualizar y realizar cuestionarios

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0012	Identificador historia de Usuario: HU0012
Nombre: Consumir varios <i>endpoints</i> para visualizar y realizar cuestionarios	
Descripción: El usuario miembro de la Iglesia puede consumir varios <i>endpoints</i> para visualizar y realizar cuestionarios de cualquier tema religioso. Además, el usuario miembro de la Iglesia puede visualizar una puntuación asignada cada vez que culmine un cuestionario.	
Pasos de ejecución: <ul style="list-style-type: none">• Iniciar sesión como usuario miembro de la Iglesia• En el menú superior dar clic en el botón “Cuestionarios”• Dar clic en Intentar• Dar clic en la opción de respuestas• Dar clic en finalizar• Visualizar puntuación• Visualizar retroalimentación	
Resultado deseado: El <i>frontend</i> permite realizar cuestionarios al usuario miembro de la Iglesia.	
Evaluación de la prueba: Se comprueba el resultado esperado. Aprobación del cliente 100%.	

Certificado de la Iglesia IFGF



Iglesia Fraternidad Internacional del Evangelio Completo GISI (IFGF)

Quito, 10 de febrero del 2022

CERTIFICADO

Yo, Walter Hidalgo, con CC. 1707795173, como pastor de Iglesia Fraternidad Internacional del Evangelio Completo IFGF GISI de Quito – Ecuador.

Por medio del presente certifico:

Que la Srta. **Nathaly Lissette Bermeo Panchi** con cédula de ciudadanía **175614988-4**, estudiante de la Carrera de Tecnología Superior en Desarrollo de Software, realizó para su trabajo de integración curricular el desarrollo de un sistema (*frontend*) para la interacción de los miembros de la comunidad de la Iglesia Fraternidad Internacional del Evangelio Completo GISI (IFGF), mismo que cumple con todos los requerimientos y funcionalidades definidas en las reuniones mantenidas.

Es todo en cuanto puedo certificar en honor a la verdad, el interesado puede hacer uso del mismo como bien tuviere.

Atentamente

Pastor Walter Hidalgo

0984575828

ANEXO III

A continuación, para acceder al enlace del Manual de Usuario se debe ingresar a la siguiente URL:

<https://www.youtube.com/watch?v=SQKP-GNjwq4>

en el que se menciona de forma clara y detallada cada una de las funcionalidades del *frontend*, así como los perfiles que intervienen en la misma.

ANEXO IV

A continuación, se procede a definir las credenciales de acceso para el *frontend*, así como el enlace al repositorio en *GitHub* en donde se encuentra el código fuente y en el apartado de *README* los pasos para realizar la instalación de forma local.

Acceso al *frontend*

Para acceder al *frontend* en producción, ingresar a la siguiente URL:

<https://ifgf.vercel.app>

Repositorio del código fuente del *frontend*

El código fuente de todo el proyecto, se encuentra alojado en el repositorio *GitHub*, el cual se puede acceder a través de la siguiente URL:

<https://github.com/NathalyBermeo97/ifgf>