



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Ciencias y Sistemas

**DESARROLLO DE APLICACIÓN MÓVIL Y PLATAFORMA ADMINISTRATIVA PARA
MEJORAR EL CONTROL DE ATENCIÓN Y FACILITAR LA CAPTURA DE DATOS EN LAS
ACTIVIDADES REALIZADAS POR LA FUNDACIÓN MISIÓN EL FARO**

Christian Leví González Rodríguez

Asesorado por el Ing. Raúl Antonio Mijangos Castro

Guatemala, mayo de 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DESARROLLO DE APLICACIÓN MÓVIL Y PLATAFORMA ADMINISTRATIVA PARA
MEJORAR EL CONTROL DE ATENCIÓN Y FACILITAR LA CAPTURA DE DATOS EN LAS
ACTIVIDADES REALIZADAS POR LA FUNDACIÓN MISIÓN EL FARO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

CHRISTIAN LEVÍ GONZÁLEZ RODRÍGUEZ
ASESORADO POR EL ING. RAÚL ANTONIO MIJANGOS CASTRO

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martinez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Marlon Francisco Orellana López
EXAMINADORA	Inga. Claudia Liceth Rojas Morales
EXAMINADORA	Inga. Virginia Victoria Tala Ayerdi
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**DESARROLLO DE APLICACIÓN MÓVIL Y PLATAFORMA ADMINISTRATIVA PARA
MEJORAR EL CONTROL DE ATENCIÓN Y FACILITAR LA CAPTURA DE DATOS EN LAS
ACTIVIDADES REALIZADAS POR LA FUNDACIÓN MISIÓN EL FARO**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería de Ciencias y Sistemas, con fecha 25 de agosto de 2021.

A handwritten signature in black ink, appearing to be 'Chrisitan Leví González Rodríguez', written in a cursive style with a large initial 'C' and 'L'.

Chrisitan Leví González Rodríguez

Guatemala, 12 de noviembre de 2022

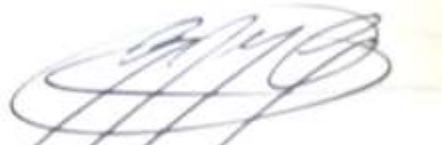
Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **CHRISTIAN LEVÍ GONZÁLEZ RODRÍGUEZ** con carné **200915168** y CUI **1642 36740 0101** titulado **"DESARROLLO DE APLICACIÓN MÓVIL Y PLATAFORMA ADMINISTRATIVA PARA MEJORAR EL CONTROL DE ATENCIÓN Y FACILITAR LA CAPTURA DE DATOS EN LAS ACTIVIDADES REALIZADAS POR LA FUNDACIÓN "MISIÓN EL FARO"**, luego de corroborar que el mismo se encuentra finalizado, lo he revisado y doy fé de que el mismo cumple con los objetivos propuestos en el respectivo protocolo, por consiguiente, procedo a la aprobación correspondiente.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Raúl Antonio Mijangos Castro
Colegiado No. 8578



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 8 de febrero de 2023

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **CHRISTIAN LEVI GONZÁLEZ RODRÍGUEZ** con carné **200915168** y CUI **1642 36740 0101** titulado "**DESARROLLO DE APLICACIÓN MÓVIL Y PLATAFORMA ADMINISTRATIVA PARA MEJORAR EL CONTROL DE ATENCIÓN Y FACILITAR LA CAPTURA DE DATOS EN LAS ACTIVIDADES REALIZADAS POR LA FUNDACIÓN - MISIÓN EL FARO-**", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA

LNG.DIRECTOR.108.EICCSS.2023

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **DESARROLLO DE APLICACIÓN MÓVIL Y PLATAFORMA ADMINISTRATIVA PARA MEJORAR EL CONTROL DE ATENCIÓN Y FACILITAR LA CAPTURA DE DATOS EN LAS ACTIVIDADES REALIZADAS POR LA FUNDACIÓN MISIÓN EL FARO**, presentado por: **Christian Leví González Rodríguez**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

"ID Y ENSEÑAD A TODOS"

Ing. Carlos Gustavo Alonzo
Director

Escuela de Ingeniería en Ciencias y Sistemas
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, mayo de 2023



La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DESARROLLO DE APLICACIÓN MÓVIL Y PLATAFORMA ADMINISTRATIVA PARA MEJORAR EL CONTROL DE ATENCIÓN Y FACILITAR LA CAPTURA DE DATOS EN LAS ACTIVIDADES REALIZADAS POR LA FUNDACIÓN MISIÓN EL FARO**, presentado por: **Christian Leví González Rodríguez**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Inga. Aurelia Anabela Cordova Estrada
Decana



Guatemala, mayo de 2023

AACE/gaoc

ACTO QUE DEDICO A:

- Dios** A quien doy gloria y honor por permitirme culminar esta etapa.
- Mis padres** Francisco González y Marlen Rodríguez, a quienes honro con este logro por su paciencia y esfuerzo todos estos años.
- Mi esposa** Sofia Fares, por su amor, apoyo incondicional y alentarme siempre a seguir adelante.
- Mis hermanos** Franklin y Jazmin González, por su cariño y por que creer en mi.
- Mis abuelos** Francisco González (q. e. p. d.), Augusto Rodríguez (q. e. p. d.).
- Mi abuela** Alba Vela, por alentarme a estudiar y animarme durante la carrera.
- Mis suegros** José Fares y Nora de Fares, por animarme a culminar esta etapa y por su apoyo.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ayudarme a alcanzar esta meta a través de todos estos años, dándome acceso a educación superior.
Facultad de Ingeniería	Por haber puesto a mi disposición recursos suficientes que aportaron a mi aprendizaje.
Escuela de Ciencias y Sistemas	Por todo el tiempo que me acogió y me brindo distintas oportunidades, y personas que influyeron mucho en mi carrera.
Ingenieros	Raúl Mijangos, Sergio Méndez, William Escobar, Byron González, Marco Castillo, Omar Vides, por ser de gran apoyo e influencia durante mi carrera.
Amigos	Erick Leonel, Abner Conde, Fernando García, Ricardo Barrios, Luis Marroquin, Jorge Rodriguez, Carlos Gómez, Esdras Orrego, Daniel Alvizures, Jonathan Herrarte, Hanns Guzmán, Luis Ramirez, Jherson Ortiz, Luis Barrios, Enrique González, Eddy Recinos, entre muchos otros que estuvieron siempre apoyándome y con quienes compartimos muy buenos momentos durante la carrera y en el ámbito laboral.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	XI
LISTA DE SÍMBOLOS	XXI
GLOSARIO	XXIII
RESUMEN.....	XXXI
OBJETIVOS.....	XXXIII
INTRODUCCIÓN	XXXV
1. MARCO TEÓRICO.....	1
1.1. Análisis del sistema actual.....	1
1.1.1. Descripción del procedimiento actual de registro y captura de datos	1
1.1.2. Deficiencias del sistema global.....	2
1.2. Aspectos fundamentales del sistema	3
1.2.1. Necesidades que están siendo atendidas	3
1.2.2. Necesidades que no han sido resueltas	3
1.3. Planteamiento de solución tecnológica	4
1.3.1. Propuesta global de solución.....	4
1.3.2. Arquitectura de solución	8
1.3.3. Casos de uso.....	13
1.3.3.1. Aplicación fuera de línea	13
1.3.3.2. Aplicación en línea.....	17
1.3.3.3. Sitio administrativo.....	18
1.4. Definición de las tecnologías a utilizar.....	37
1.4.1. Tecnologías que se utilizaran para el desarrollo del <i>frontend</i>	37

1.4.1.1.	React.....	37
1.4.1.2.	Material-UI.....	38
1.4.1.3.	Ionic.....	38
1.4.1.4.	JavaScript.....	39
1.4.1.5.	NodeJS y npm.....	39
1.4.1.6.	Capacitor.....	40
1.4.2.	Tecnologías que se utilizarán para el desarrollo del <i>backend</i>	41
1.4.2.1.	Flask.....	41
1.4.2.2.	Python.....	42
1.4.2.3.	MongoDB.....	43
1.4.2.4.	Autenticación con Firebase.....	43
1.4.2.5.	SQLite.....	44
1.4.3.	Otras tecnologías y servicios.....	45
1.4.3.1.	Git.....	45
1.4.3.2.	Google App Engine.....	46
1.4.4.	Proyección de costos.....	47
1.4.4.1.	Análisis de costos por uso de tecnologías en la nube.....	47
1.4.4.2.	Presupuesto a corto, mediano y largo plazo.....	50
1.4.4.3.	Plan de financiamiento de servicios para la solución.....	51
1.4.4.4.	Propuesta de mantenimiento y garantía del sistema.....	53
2.	VENTAJAS Y DESVENTAJAS DE LA SOLUCIÓN PROPUESTA.....	55
2.1.	Ventajas y desventajas del sistema actual.....	55

2.2.	Ventajas y desventajas de las herramientas para el <i>frontend</i>	56
2.2.1.	React vs otras tecnologías para <i>frontend</i>	56
2.2.1.1.	React	56
2.2.1.2.	Vue.js.....	57
2.2.1.3.	Angular	58
2.2.2.	Ionic vs otras tecnologías para construir aplicaciones móviles.....	59
2.2.2.1.	Ionic.....	59
2.2.2.2.	Xamarin	61
2.2.2.3.	Flutter	61
2.3.	Ventajas y desventajas de las herramientas para el <i>backend</i>	64
2.3.1.	Flask vs otras tecnologías para <i>backend</i>	64
2.3.1.1.	Flask.....	64
2.3.1.2.	Django	65
2.3.1.3.	Pyramid	65
2.3.2.	Modelos NoSql vs SQL.....	67
2.3.2.1.	NoSQL.....	67
2.3.2.2.	SQL	68
2.4.	Ventajas y desventajas de los servicios en la nube.....	70
2.4.1.	MongoDB Atlas vs Firestore	71
2.4.1.1.	MongoDB Atlas.....	71
2.4.1.2.	Firestore	72
2.4.2.	Google App Engine vs Google Compute Engine	74
2.4.2.1.	Google App Engine.....	74
2.4.2.2.	Google Compute Engine	75

3.	ARQUITECTURA Y DISEÑO.....	79
3.1.	Creación del modelo de base de datos.....	79
3.1.1.	Diseño de modelo NoSql para banco de datos y formularios.....	79
3.1.2.	Diseño de modelo SQL para captura de datos fuera de línea	81
3.2.	Explicación del modelo de base de datos NoSql	82
3.3.	Explicación del modelo de base de datos Sql.....	86
3.4.	Propuesta de arquitectura de servicios en la nube	87
3.4.1.	Función de Google App Engine.....	87
3.4.2.	Función de MongoDB.....	88
3.4.3.	Función de Firebase: autenticación con contraseña.....	89
3.5.	Repositorio de código.....	89
3.5.1.	Control de versiones con Git.....	90
3.6.	Diseño y propuesta para <i>backend</i>	91
3.6.1.	Lenguaje y framework a utilizar	91
3.7.	Diseño y propuesta para el <i>frontend</i>	92
3.7.1.	<i>Mockups</i> de aplicación móvil.....	92
3.7.2.	<i>Mockups</i> de sitio administrativo.....	96
3.7.2.1.	Ingreso al sistema	96
3.7.2.2.	Recuperación de contraseña.....	96
3.7.2.3.	Menú principal	98
3.7.2.4.	Módulo de lugares	99
3.7.2.5.	Módulo de formularios.....	100
3.7.2.6.	Módulo de eventos	102
3.7.2.7.	Módulo de beneficiarios.....	104
3.7.2.8.	Módulo de usuarios y roles.....	106
3.7.3.	Funcionalidades de aplicación móvil	109

3.7.4.	Funcionalidades de plataforma web administrativa.....	109
3.8.	Buenas prácticas y seguridad de transacciones y datos	111
3.8.1.	Utilizar autorización en los servicios del API REST	111
4.	ESPECIFICACIONES DE SERVICIOS DE LA NUBE	113
4.1.	Procedimiento para crear la cuenta de servicios de GCP	113
4.2.	Procedimiento para crear un proyecto y definir usuarios y accesos dentro del proyecto	114
4.3.	Configuraciones iniciales de servicios en la nube	118
4.3.1.	Creación de proyecto Firebase.....	118
4.3.2.	Habilitar método de autenticación en Firebase.....	119
4.3.3.	Creación de proyecto en MongoDB Atlas.....	120
5.	ESPECIFICACIONES DEL BACKEND	123
5.1.	Configuraciones iniciales del <i>backend</i>	123
5.1.1.	Instalación y configuraciones de Flask	123
5.1.2.	Instalación y configuraciones de Flask-Cors.....	124
5.1.3.	Instalación y configuraciones de Firebase	125
5.1.4.	Manejo de base de datos con MongoDB	127
5.2.	Estructura del framework.....	128
5.2.1.	Bloque de dependencias	129
5.2.2.	Bloque de api rest.....	130
5.2.3.	Bloque de inicialización de servidor.....	131
5.3.	Sincronización de aplicación móvil	131
5.3.1.	Sincronización de datos de aplicación móvil (en <i>backend</i>).....	132

5.3.2.	Sincronización de datos de aplicación móvil (en aplicación móvil).....	134
5.3.3.	Sincronización de datos de eventos (en aplicación móvil).....	136
5.3.4.	Sincronización de datos de eventos (<i>backend</i>).....	137
6.	DEFINICION DE SERVICIOS DE API	141
6.1.1.	Funciones de usuarios	141
6.1.1.1.	Creación de usuario	141
6.1.1.2.	Listar usuarios	142
6.1.1.3.	Editar de usuario	142
6.1.1.4.	Eliminar usuario.....	142
6.1.2.	Funciones de Roles.....	143
6.1.2.1.	Creación de rol	143
6.1.2.2.	Listar roles.....	143
6.1.2.3.	Editar de rol	143
6.1.2.4.	Eliminar rol	144
6.1.3.	Funciones de beneficiarios.....	144
6.1.3.1.	Creación de beneficiario.....	144
6.1.3.2.	Listar beneficiario	145
6.1.3.3.	Editar beneficiario.....	146
6.1.3.4.	Eliminar beneficiario	146
	6.1.3.4.1. Flujo de eliminación de beneficiario.....	147
6.1.4.	Funciones de formularios	148
6.1.4.1.	Creación de formulario	148
6.1.4.2.	Listar formularios.....	149
6.1.4.3.	Editar de formulario	149
6.1.4.4.	Eliminar formulario	150

6.1.5.	Funciones de eventos.....	150
6.1.5.1.	Creación de evento.....	150
6.1.5.2.	Listar eventos	151
6.1.5.3.	Editar de evento.....	152
	6.1.5.3.1. Flujo de actualización de evento.....	153
6.1.5.4.	Eliminar evento	154
	6.1.5.4.1. Flujo de eliminación de evento.....	155
6.1.5.5.	Asignación Formulario-Evento.....	156
	6.1.5.5.1. Flujo de asignación de formulario a evento.....	157
6.1.5.6.	Crear registro de evento	158
	6.1.5.6.1. Flujo de evento de registro libre.....	158
	6.1.5.6.2. Flujo de evento de registro único	161
6.1.5.7.	Listar registros de evento	163
6.1.6.	Funciones de lugares	163
6.1.6.1.	Creación de ciudad.....	163
6.1.6.2.	Listar ciudades.....	163
6.1.6.3.	Editar de ciudad.....	164
6.1.6.4.	Eliminar ciudad	164
6.1.6.5.	Creación de lugar	164
6.1.6.6.	Listar lugares	165
6.1.6.7.	Editar de lugar	165
6.1.6.8.	Eliminar lugar.....	165
6.1.7.	Funciones de sincronización.....	166
6.1.7.1.	Actualizar datos de aplicación	166

6.1.7.2.	Sincronizar beneficiarios y registros de evento.....	166
7.	ESPECIFICACIONES DEL FRONTEND	167
7.1.	Aplicación móvil	167
7.1.1.	Instalación de Ionic.....	167
7.1.2.	Inicialización de proyecto de Ionic.....	167
7.1.3.	Componentes de pantallas de aplicación móvil.....	169
7.1.3.1.	Componentes de navegación.....	169
7.1.3.1.1.	Menú	169
7.1.3.1.2.	Barra de herramientas	170
7.1.3.1.3.	Lista	171
7.1.3.1.4.	Tarjetas	171
7.1.3.2.	Componentes de interacción y formularios.....	172
7.1.3.3.	Iconos, alertas y notificaciones.....	173
7.2.	Plataforma web administrativa	174
7.2.1.	Instalación y configuraciones iniciales de React ...	174
7.2.2.	Componentes de pantallas de sitio administrativo.	177
7.2.2.1.	Componentes de navegación.....	177
7.2.2.1.1.	Cajón de navegación ..	177
7.2.2.1.2.	Barra de aplicación	178
7.2.2.1.3.	Menús simples	178
7.2.2.1.4.	Pestañas	179
7.2.2.1.5.	Tarjetas	179
7.2.2.2.	Componentes de interacción y formularios.....	180
7.2.2.3.	Iconos, alertas y notificaciones.....	181

8.	PUBLICACIÓN DE APLICACIONES.....	183
8.1.	Procedimiento para la publicación del <i>backend</i>	183
8.2.	Procedimiento para la publicación del <i>frontend</i>	188
8.3.	Publicación de aplicación Android en google drive.....	193
9.	PRUEBAS DE SITIO ADMINISTRATIVO Y APLICACIÓN MÓVIL	197
9.1.	Pruebas de sitio administrativo.....	197
9.2.	Pruebas de aplicación móvil.....	205
10.	RESULTADOS DE IMPLEMENTACIÓN.....	209
10.1.	Necesidades resueltas del sistema	209
10.2.	Características destacables de las soluciones	211
10.3.	Beneficios de la arquitectura implementada	214
10.4.	Resumen de las tecnologías utilizadas.....	214
10.5.	Costos de implementación	215
	CONCLUSIONES	217
	RECOMENDACIONES.....	219
	REFERENCIAS	221
	APÉNDICES	223

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Registro fuera de línea	9
2.	Vista general de arquitectura en línea.....	10
3.	Sincronización de datos de aplicación móvil	11
4.	Gestiones de plataforma administrativa	12
5.	Aplicación fuera de línea	16
6.	Aplicación en línea	18
7.	Mantenimiento de usuarios y roles.....	22
8.	Mantenimiento de eventos	25
9.	Mantenimiento de formularios	27
10.	Mantenimiento de beneficiario.....	29
11.	Mantenimiento de lugares	33
12.	Sitio administrativo	36
13.	Componentes de <i>frontend</i>	41
14.	Componentes de <i>backend</i>	45
15.	Flujo de Git-Flow	46
16.	Precios de autenticación con Firebase.....	48
17.	Precios de MongoDB Atlas.....	49
18.	Precios de App Engine (capa gratuita)	49
19.	Diagrama de tecnologías a utilizar.	77
20.	Modelo para base de datos NoSQL	80
21.	Modelo para base de datos SQL.....	81
22.	Arquitectura GAE <i>Frontend</i> y <i>Backend</i>	87
23.	Función de firebase para autenticación de usuarios	89

24.	Inicio de aplicación.....	92
25.	Menú principal.....	93
26.	Módulo de Eventos	93
27.	Búsqueda de perfil de beneficiario.....	94
28.	Registro de formularios.....	94
29.	Creación de perfil de beneficiario.....	95
30.	Ajustes	95
31.	Pantalla de inicio de sesión.....	96
32.	Solicitud para reestablecer contraseña.....	97
33.	Reestablecer contraseña	97
34.	Menú principal del sitio administrativo.....	98
35.	Vista y descarga de reportes de actividad	98
36.	Listado de ciudades y lugares.....	99
37.	Creación/Edición de ciudades y lugares	99
38.	Listado de formularios	100
39.	Detalle de formulario	100
40.	Editar formulario.....	101
41.	Creación de formulario.....	101
42.	Listado de eventos del sistema.....	102
43.	Detalle de evento	102
44.	Ver registros de evento	103
45.	Editar evento.....	103
46.	Crear evento	104
47.	Lista de beneficiarios	104
48.	Crear beneficiario.....	105
49.	Editar beneficiario	105
50.	Lista de usuarios.....	106
51.	Crear usuario	106
52.	Editar usuario.....	107

53.	Lista de roles	107
54.	Editar permisos de rol	108
55.	Crear rol con permisos	108
56.	Creación de cuenta en GCP.....	113
57.	Selección de proyecto GCP	114
58.	Creación de nuevo proyecto GCP.....	115
59.	Menu IAM y administración GCP	115
60.	Agregar miembro a proyecto GCP	116
61.	Agregar permisos a miembro del proyecto GCP	117
62.	Guardar nuevo miembro de proyecto GCP	117
63.	Creación de proyecto en firebase.....	118
64.	Detalles de creación de proyecto en firebase.....	119
65.	Habilitar autenticación por correo electrónico y contraseña	120
66.	Creación de proyecto de MongoDB	120
67.	Creación de usuario para MongoDB	121
68.	Creación de base de datos en MongoDB.....	122
69.	Instalación de librería Flask en Python.....	124
70.	Instalación de librería Flask-Cors en python	125
71.	Instalación de librería Firebase en Python	126
72.	Instalación libreria flask_pymongo	127
73.	Uso de libreria flask_pymongo	128
74.	Estructura de proyecto Flask.....	129
75.	Ejemplo de petición GET, usando Flask.....	130
76.	Estructura de ejemplo: Flask framework	131
77.	Flujo de sincronización de datos de aplicación móvil (en <i>backend</i>)	133
78.	Flujo de sincronización de datos en aplicación móvil (en aplicación móvil).....	135
79.	Flujo de sincronización de datos de eventos (aplicación móvil)	137
80.	Flujo de sincronización de datos de eventos (<i>backend</i>).....	139

81.	Flujo de eliminación de beneficiario	147
82.	Flujo de actualización de evento	154
83.	Flujo de eliminación de evento.....	156
84.	Flujo de asignación de formulario a evento.....	157
85.	Flujo de evento de registro libre	160
86.	Flujo de evento de registro único	162
87.	Instalación de Ionic	167
88.	Inicialización de proyecto de Ionic	168
89.	Ejecución local de aplicación móvil.....	168
90.	Vista de aplicación móvil en ejecución local	169
91.	Componente ion-menu.....	170
92.	Componente ion-toolbar.....	170
93.	Componente ion-list	171
94.	Componente ion-card.	171
95.	Componentes de botones y acciones de Ionic.....	172
96.	Componentes de formularios de Ionic.....	173
97.	Iconos, alertas y notificaciones de Ionic.....	174
98.	Creación de aplicación de React	175
99.	Aplicación de React creada con éxito.	175
100.	Vista de aplicación de React en ejecución.....	176
101.	Instalación de Material-UI	176
102.	Cajón de navegación, Material-UI.....	177
103.	Barra de aplicación, Material-UI.....	178
104.	Menús simples, Material-UI.....	178
105.	Pestañas, Material-UI.	179
106.	Tarjetas, Material-UI... ..	179
107.	Componentes de formularios, Material-UI.....	180
108.	Botones, Material-UI	181
109.	Iconos, Material-UI..... ..	181

110.	Notificaciones, Material-UI.....	182
111.	Diálogos, Material-UI.. ..	182
112.	Cambio de ruta para llave de firebase.....	183
113.	Creación de archivo “app.yaml”.....	184
114.	Revisión de requerimientos de Flask.	184
115.	Comprimir carpeta del código del proyecto	185
116.	Consola de GCP dentro del proyecto.....	185
117.	Opción “Subir” de consola de GCP	186
118.	Subir carpeta de código a consola de GCP.....	186
119.	Despliegue de aplicación <i>backend</i>	187
120.	Ruta de <i>backend</i> en db.js.....	188
121.	Comprimir carpeta del código del proyecto	188
122.	Consola de GCP dentro del proyecto	189
123.	Opción “Subir” de consola de GCP	189
124.	Subir carpeta de código a consola de GCP.....	190
125.	Comprimir carpeta del código del proyecto	191
126.	Creando paquete de instalación.....	191
127.	Creación de carpeta de despliegue.....	192
128.	Despliegue de aplicación <i>frontend</i>	193
129.	Abrir el proyecto en Android Studio.....	194
130.	Generación de APK.....	194
131.	Localizar APK generada.....	195
132.	Sitio administrativo.....	212
133.	Aplicación móvil.....	213

TABLAS

I.	CU1. Ingreso al sistema (o inicio de sesión).....	13
II.	CU2. Listar beneficiarios.....	13
III.	CU3. Crear beneficiario	14
IV.	CU4. Eliminar beneficiario	14
V.	CU5. Listar eventos	14
VI.	CU6. Seleccionar evento	15
VII.	CU7. Registro de evento: asignar beneficiario.....	15
VIII.	CU8. Registro de evento: llenar formulario	15
IX.	CU9. Cerrar sesión	16
X.	CU1. Sincronizar datos de aplicación	17
XI.	CU2. Sincronizar datos registrados	17
XII.	CU1. Recuperación de contraseña.....	18
XIII.	CU2. Ingreso al sistema (<i>Login</i>)	19
XIV.	CU3. Creación de rol	19
XV.	CU4. Edición de rol	19
XVI.	CU5. Listar roles	20
XVII.	CU6. Eliminar roles	20
XVIII.	CU7. Creación de usuario.....	20
XIX.	CU8. Edición de usuario	21
XX.	CU9. Listar usuario	21
XXI.	CU10. Eliminar usuarios	21
XXII.	CU11. Creación de evento.....	22
XXIII.	CU12. Edición de evento	23
XXIV.	CU13. Listar eventos	23
XXV.	CU14. Eliminar eventos	23
XXVI.	CU15. Ver registros de evento.....	24
XXVII.	CU16. Ver detalle de registro.....	24

XXVIII.	CU17. Descarga de registros de evento	24
XXIX.	CU18. Creación de formulario.....	25
XXX.	CU19. Edición de formulario	26
XXXI.	CU20. Listar formularios.....	26
XXXII.	CU21. Eliminar formularios	27
XXXIII.	CU22. Creación de beneficiario	28
XXXIV.	CU23. Edición de beneficiario	28
XXXV.	CU24. Listar beneficiarios	28
XXXVI.	CU25. Eliminar beneficiario.....	29
XXXVII.	CU26. Creación de ciudad	30
XXXVIII.	CU27. Edición de ciudad.....	30
XXXIX.	CU28. Listar ciudades	30
XL.	CU29. Eliminar ciudad	31
XLI.	CU30. Creación de lugar.....	31
XLII.	CU31. Edición de lugar	32
XLIII.	CU32. Listar lugares	32
XLIV.	CU33. Eliminar lugar	32
XLV.	CU34. Ver resumen de actividad	33
XLVI.	CU35. Reporte “Beneficiarios atendidos en el mes actual”	34
XLVII.	CU36. Reporte “Lugares poblados atendidos en el mes actual”	34
XLVIII.	CU37. Reporte “Eventos terminados del mes actual”	34
XLIX.	CU38. Reporte “Eventos pendientes del mes actual”	35
L.	CU39. Reporte “Personas atendidas por tipo de evento por año”	35
LI.	CU40. Reporte “Porcentaje de personas atendidas por tipo de evento por mes”	35
LII.	CU41. Cerrar sesión... ..	36
LIII.	Resumen de tecnologías a utilizar para <i>frontend</i>	40
LIV.	Resumen de tecnologías a utilizar para <i>backend</i>	44
LV.	Precios de GAE para plataforma administrativa y <i>backend</i>	50

LVI.	Precios de Firebase y MongoDB	51
LVII.	Cuota por servicios GCP y Firebase	51
LVIII.	Proyección de costos de servicios GCP y otros	52
LIX.	Ventajas y desventajas de React.....	59
LX.	Ventajas y desventajas de Ionic	63
LXI.	Ventajas y desventajas de Flask	66
LXII.	Ventajas y desventajas de NoSQL y SQL	70
LXIII.	Ventajas y desventajas de NoSQL y SQL	73
LXIV.	Ventajas y desventajas de Google App Engine y Google Compute Engine	76
LXV.	Archivo de despliegue <i>frontend</i>	192
LXVI.	Pruebas de token de autorización	198
LXVII.	Pruebas de Inicio y fin de sesión	198
LXVIII.	Pruebas de ciudad.....	199
LXIX.	Pruebas de lugar	199
LXX.	Pruebas de roles.....	200
LXXI.	Pruebas de usuarios..	200
LXXII.	Pruebas de formularios.....	201
LXXIII.	Pruebas de eventos...	202
LXXIV.	Pruebas de asignación de formulario.....	203
LXXV.	Pruebas de beneficiarios	203
LXXVI.	Pruebas de reportes y descargas	204
LXXVII.	Pruebas de token de autorización	205
LXXVIII.	Pruebas de inicio y fin de sesión	206
LXXIX.	Pruebas de eventos y registros	206
LXXX.	Pruebas de actualización y sincronización	207
LXXXI.	Necesidades que se realizaban de forma manual	209
LXXXII.	Necesidades identificadas	210

LISTA DE SÍMBOLOS

Símbolo	Significado
\$	Dólar
Gb	Giga Bite
Mb	Mega Bite

GLOSARIO

Android	Es un sistema basado en una modificación del Kernel de Linux orientada al uso de dispositivos con pantallas táctiles como teléfonos móviles y otros dispositivos.
API Rest	Es una interfaz de programación de aplicaciones (API o API web), la cual se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful.
Backend	Es un término que se usa para referirse a un ámbito de la programación de aplicaciones, la parte que se encarga del acceso a los datos y la seguridad de la información.
Backup	Es una copia de seguridad o el proceso de copia de seguridad.
Beneficiario	Que obtiene beneficio o provecho de determinada cosa.
Blaze plan	Es el plan de pago por uso de Firebase.
Cordova	Es un <i>framework</i> de licencia libre que cuenta con muchas Apis de diversos dispositivos móviles para

desarrollar aplicaciones nativas dentro de un smartphone.

Css CSS es un lenguaje para la definición del aspecto en los documentos Web.

Delete Método para eliminar un recurso del servidor.

Development Rama de *gitflow* que sirve como rama de integración para funciones.

ECMA Ecma International es una organización internacional basada en membresías de estándares para la comunicación y la información.

ECMAScript Es el estándar que a partir del año 2015 a la actualidad se encarga de regir como debe ser interpretado y funcionar el lenguaje JavaScript.

Features Rama de *gitflow* que sirve para el desarrollo de nuevas funciones. Utilizan la rama de desarrollo como rama primaria.

Firebase Firebase es un conjunto de herramientas orientadas a la creación de aplicaciones de alta calidad, al crecimiento de los usuarios y a ganar más dinero.

Firestore Es la base de datos más reciente de Firebase para el desarrollo de aplicaciones para dispositivos móviles.

Flask	Es un “micro” <i>Framework</i> escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC.
Flexible Environment	Ofrece acceso por SSH y despliegues sobre Docker y te permite usar Python, Go, Node.js, .Net, Ruby y PHP. Posee mayor flexibilidad, pero se deben realizar algunas configuraciones adicionales e instalar las librerías de App Engine que requiera la aplicación.
<i>Framework</i>	Es una plataforma de <i>software</i> universal y reutilizable para desarrollar aplicaciones de <i>software</i> , productos y soluciones.
<i>Frontend</i>	Es un concepto usado para referirse a la capa de presentación de las aplicaciones, que se encarga de definir cómo se verán las cosas y cómo se responderá a la interacción con el usuario.
<i>Full stack</i>	Se le llama así al <i>framework</i> que contiene las herramientas necesarias para ejecutar no solo el <i>frontend</i> , si no también el <i>backend</i> de una aplicación, además de muchos otros complementos.
GAE	Google App Engine es un servicio de alojamiento web que presta Google de forma gratuita hasta determinadas cuotas. Este servicio permite ejecutar aplicaciones sobre la infraestructura de Google.

GCP	Google cloud platform es el conjunto de herramientas, infraestructuras y servicios que Google utiliza a nivel interno, que ahora están disponible en una plataforma web.
Get	Método para obtener un recurso del servidor.
Git	Es un <i>software</i> de control de versiones.
Gitflow	Es un flujo de trabajo y conjunto de extensiones para Git que permiten gestionar de manera eficiente las ramas en nuestros repositorios Git.
Hotfix	Rama de <i>gitflow</i> para realizar correcciones en el código ya puesto en producción.
Html	Siglas en inglés de HyperText Markup Language ('lenguaje de marcado de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web.
IaaS	Siglas en inglés que significan infraestructura como servicio (Infrastructure as a service), y es un método para ofrecer funcionalidades de computación, almacenamiento, redes y de otros tipos a través de Internet.

Ionic	Es un SDK de <i>frontend</i> de código abierto para desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS).
Ip	Es un conjunto de números que identifica, de manera lógica y jerárquica, a una interfaz en la red de un dispositivo que utilice el protocolo o, que corresponde al nivel de red del modelo TCP/IP.
Javascript	JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.
Kernel	Es un <i>software</i> que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado.
Login	Es el proceso que controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.
Master	Rama de <i>gitflow</i> almacena el historial de publicación oficial.
Material Design	Es una normativa de diseño desarrollada por Google y enfocado en la visualización del sistema operativo Android, web y en cualquier plataforma.

Mockups	Es un modelo o un prototipo que se utiliza para exhibir o probar un diseño.
MongoDB	Es una base de datos NoSQL, orientada a documentos con alta escalabilidad y flexibilidad.
MongoDB Atlas	Es un servicio de base de datos en la nube que permite administrar una base de datos MongoDB alojada en la nube.
MVC	Modelo vista controlador.
NodeJS	Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.
NoSql	Es una base de datos no relacional que no cuenta con un identificador que relacione un conjunto de datos con otro.
Npm	es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript, bajo Artistic License 2.0.
Nube	Es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es internet.

Offline	Que está disponible o se realiza sin conexión a internet o a otra red de datos.
Online	Que está disponible o se realiza a través de internet o de otra red de datos.
PaaS	Siglas en inglés que significan plataforma como servicio (Platform as a service), y es un entorno de desarrollo e implementación completo en la nube.
PlayStore	Es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google.
Plugin	Es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al <i>software</i> .
Post	Método para crear un recurso del servidor.
Put	Método para actualizar un recurso del servidor.
Python	Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.
React	Es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el

objetivo de facilitar el desarrollo de aplicaciones en una sola página.

Release Rama de *gitflow* que nace a partir de una de desarrollo una vez que el desarrollo ha adquirido suficientes funciones para una publicación.

Rol Es una colección de permisos definida para todo el sistema que se puede asignar a usuarios específicos en contextos específicos.

Software Programa informático que hace posible la ejecución de tareas específicas dentro de un computador.

Spark plan Es el plan gratis de Firebase.

SQL Es una base de datos relacional, escrita en el lenguaje de consulta estructurado SQL.

Standard Environment Se caracteriza por la velocidad de auto escalamiento, lo que le permite responder a altos y repentinos picos de tráfico en segundos. Este ambiente tiene un precio bajo e incluye un nivel gratuito.

Virtual DOM Es una representación de JavaScript liviana del DOM que se usa en marcos web declarativos como React, Vue.js y Elm.

RESUMEN

En este trabajo de graduación se presenta la solución para el registro de beneficiarios en eventos realizados por la fundación Misión El Faro, a través de una aplicación móvil y una plataforma administrativa.

La aplicación móvil se utilizará para agilizar el proceso de registro en lugares donde muchas veces no se cuenta con acceso a internet. Por lo que también podrá funcionar fuera de línea para sincronizar a la nube los datos capturados luego de recuperar el acceso a internet.

La captación de datos con la aplicación móvil obtendrá el perfil de cada beneficiario y el registro del formulario del evento. Esto permitirá tener un mejor control del registro de beneficiarios en eventos y enriquecer la base de datos con datos demográficos.

El sitio administrativo facilitará a los usuarios del sistema la creación de formulario y eventos. Los formularios previamente creados podrán asignarse a los eventos. Esto con el fin de sustituir los formularios físicos que actualmente se utilizan y poder recuperar los datos registrados de una manera más rápida y segura durante el registro de eventos.

Así mismo, el sitio administrativo contará con un resumen de la actividad realizada como página principal, para dar un vistazo general del trabajo que la fundación está haciendo.

OBJETIVOS

General

Mejorar el control de atención y facilitar la captura de datos durante las diferentes actividades que realiza la fundación Misión El Faro, para beneficio de las comunidades aledañas a punta de palma, Izabal a través de una aplicación móvil, la cual debe diseñarse tomando en cuenta las limitaciones que existen en algunos de los lugares y una plataforma administrativa haciendo uso de la tecnología en la nube para que esta información este siempre disponible y visible para los administradores.

Específicos

1. Proporcionar una plataforma que facilite la captura de datos y control de personas atendidas durante las actividades realizadas por la fundación Misión El Faro, a beneficio de las comunidades de punta de palma, Izabal.
2. Hacer uso de tecnologías en la nube para crear un banco de datos y proveer la información obtenida a través de reportes en la plataforma web, que puedan tener una alta disponibilidad.
3. Crear un espacio con información que pueda dejar constancia del trabajo realizado por la fundación, para la alta gerencia.

4. Brindar una solución que funcione “fuera de línea” para que esta pueda trabajar sin necesidad de conexión y posteriormente sincronizar los datos recopilados.
5. Aplicar protocolos de seguridad en las transferencias de información y en las peticiones al servidor para proteger los datos obtenidos.

INTRODUCCIÓN

Durante más de 15 años la fundación Misión El Faro, ha realiza diferentes actividades para beneficio de las comunidades cercanas a Punta de Palma, Izabal, Guatemala. Entre estas actividades se pueden mencionar jornadas médicas, jornadas dentales, actividades educativas para niños y mamás, entre otras. Aunque la fundación cuenta también con algunas clínicas dentro de la institución, su mayor campo de trabajo es en las comunidades.

Por varios años, una de las necesidades que surgen durante estas actividades es tener un mejor control de los datos de las personas que se registran. Actualmente, el sistema de registro que ellos utilizan para estas actividades consta de formularios prediseñados que deben imprimir y llevar a cada lugar poblado. Al momento de regresar a la base, estos datos se deben ingresan manualmente a una hoja de cálculo para realizar los reportes correspondientes.

Debido a las condiciones climáticas del lugar y los caminos que deben recorrer para llegar, muchas veces se han dañado estos formularios y se ha perdido la información que se recolecta a lo largo del proceso de las actividades. Hasta ahora se ha tratado de proteger lo mejor posible estos documentos, pero no se ha desarrollado ninguna herramienta de tecnología que pueda apoyar o sustituir su sistema actual.

Por lo tanto, se busca realizar una solución tecnológica con una aplicación móvil y un sitio administrativo que pueda solventar las deficiencias planteadas del

sistema actual, utilizando la información recolectada para presentar métricas sobre el trabajo realizado.

1. MARCO TEÓRICO

Este capítulo trata de hacer un análisis del sistema actual utilizado por la fundación Misión El Faro, para la toma de datos durante el proceso de registro, en las diferentes actividades que realizan en beneficio de la comunidad de Punta de Palma, Izabal.

Se busca explicar más a detalle cual es el sistema actual, las ventajas y desventajas que tiene, las necesidades que debe cubrir la fundación para estos eventos, con el objetivo de plantear una solución tecnológica que permita hacer esta labor más fácil, eficiente y segura.

1.1. Análisis del sistema actual

A continuación, se realiza un análisis del sistema actual para la captura de datos de registro y beneficiarios.

1.1.1. Descripción del procedimiento actual de registro y captura de datos

Actualmente, durante el proceso de registro en las actividades se realiza forma manual. Se diseñan formularios de registro para las actividades que se realizan y a medida que se necesitan datos adicionales, se modifica o diseña un nuevo formulario. Estos formularios se deben imprimir y llevar hasta el lugar donde serán empleados.

La mayoría de las veces, las comunidades son bastante lejanas así que deben asegurarse de llevar suficientes formularios y cuidar que estos lleguen bien a su destino, y así mismo vuelvan a la sede para registrar los datos obtenidos.

Al llegar al lugar, dentro del equipo hay algunas personas designadas para el área de registro, quienes se encargan de llenar los formularios con los datos de los beneficiarios que se presentan a la actividad, los cuales pueden llegar solos o acompañados de su familia, siendo uno o muchos los beneficiarios por cada turno.

Al terminar la jornada, se guardan los formularios que se llenaron para que posteriormente sean ingresados de manera manual a una hoja de cálculo para su posterior análisis. Luego de hacer el ingreso, esta información permite saber a cuantas personas se atendió, promedio de edad, sexo, razón de visita (en caso de jornadas médicas), cantidad de hijos por familia, entre otros datos de interés de la administración.

1.1.2. Deficiencias del sistema global

Sobre la forma en la que se lleva a cabo el registro, se puede detectar que la pérdida de información es algo que seguirá sucediendo si no se resuelven los diferentes factores de riesgo que rodean al sistema.

Esta pérdida de información puede ser causada por el mal manejo de la papelería, la pérdida o daño de boletas, errores en el ingreso de la información a la hoja de cálculo, entre otras cosas.

Regularmente la información tarda mucho en ser procesada, porque debe ingresarse de manera manual al documento donde se lleva el control del registro de las actividades. Por lo mismo, no se pueden generar reportes en poco tiempo, ni tener una perspectiva rápida de los objetivos alcanzados hasta no tener toda la información guardada.

1.2. Aspectos fundamentales del sistema

En el sistema general, existen necesidades que están siendo atendidas de manera manual, y algunas que aun no han sido resueltas.

1.2.1. Necesidades que están siendo atendidas

Las necesidades que hoy en día están siendo cubiertas, respecto al proceso que se está haciendo para el registro de personas en las actividades son:

- Realizar el registro de datos personales de cada beneficiario que llega al lugar de la actividad.
- Hacer el registro de las necesidades atendidas por beneficiario o familia.
- Registro manual de la información obtenida en hojas de cálculo.

1.2.2. Necesidades que no han sido resueltas

Las necesidades que aún no se resuelven con el sistema de registro actual son:

- Proteger la integridad de la información recopilada en cada actividad

- Optimizar el tiempo en que se recupera la información de las boletas.
- Generar un resumen de la actividad realizada respecto a la información recopilada.
- Evitar que los beneficiarios se registren más de una vez en el transcurso de la actividad de registro único a través de la validación del DPI.
- Aprovechar los datos demográficos y de eventos que se obtengan para presentar un informe sobre el trabajo realizado.

1.3. Planteamiento de solución tecnológica

Respecto a las necesidades vistas anteriormente, se plantea una propuesta global de solución para automatizar el registro en eventos.

1.3.1. Propuesta global de solución

Para solventar las necesidades actuales del registro de actividades, se propone el desarrollo e implementación de una aplicación móvil y una plataforma administrativa, los cuales contarán con las siguientes características:

- Aplicación móvil de registro: para el registro de datos en las actividades y creación de formularios para toma de datos. Esta aplicación deberá contar con las siguientes características:
 - Ingreso de usuario: usuario y contraseña creados por Administrador.

- Módulo de eventos: servirá para listar eventos pendientes y finalizados y para realizar registro de datos durante eventos. Al realizar el registro, si no existe el beneficiario en el sistema, existirá la opción para hacer un nuevo registro de beneficiario. Esto permitirá realizar el registro con el nuevo perfil de beneficiario creado.
- Módulo de ajustes: sección donde podrá realizarse la sincronización de datos o cerrar la sesión. La sincronización se podrá realizar para 2 motivos:
 - Sincronizar datos de aplicación: descarga los datos necesarios para el registro (datos de beneficiarios, formularios, eventos, entre otros).
 - Sincronizar los datos registrados: sincronizará los datos de los nuevos perfiles de beneficiarios y los registros de evento que se hayan realizado.
- Sitio administrativo: este panel será una herramienta adicional, una plataforma web donde el administrador podrá llevar el control de usuarios, crear formularios para eventos, visualizar por evento los datos recolectados y acceder a reportes por evento.
 - Ingreso de usuario: usuario y contraseña creados por Administrador.
 - Resumen de actividad: esta será la sección principal, que podrá ser visualizada por cualquier usuario que tenga acceso al sistema. Este

reporte de actividad contendrá información sobre la información obtenida sobre los beneficiarios y eventos. Para este trabajo se contempla que el reporte tenga la siguiente información:

- Beneficiarios atendidos en el mes actual
 - Lugares poblados atendidos en el mes actual (reporte por tipo de evento).
 - Eventos terminados del mes actual (reporte con filtros por tipo de evento).
 - Eventos pendientes del mes actual (reporte con filtros por tipo de evento).
 - Personas atendidas por tipo de evento por año (reporte con filtro de año).
 - Porcentaje de personas atendidas por tipo de evento por mes (reporte con filtro de mes-año).
-
- Módulo de roles y usuarios: este módulo lo tendrá disponible el usuario administrador y los usuarios a quien es él designe estos permisos. Se utilizará para crear, listar, eliminar o modificar usuarios y roles dentro del sistema.
 - Módulo de gestión de eventos: dentro de este módulo se podrán crear, listar, eliminar o modificar eventos. Durante la creación o edición se podrá asignar un formulario previamente creado al

evento. El usuario podrá ingresar a cada evento y tener una vista previa de los datos que se obtuvieron. Adicionalmente podrá exportar los datos registrados en formato “.csv”. Los tipos de evento que existen en el sistema son:

- Salud integral
 - Construcción
 - Beneficio comunitario
 - Otros
- Módulo de gestión de formularios: dentro de este módulo se podrán crear formularios de registro para los distintos eventos del sistema. Cada formulario podrá asignarse posteriormente a un evento para que se puedan tomar los datos de registro del evento en el formulario asignado. Se tendrá la posibilidad de editar formularios y agregar nuevos campos. Los tipos de datos que se proponen para los formularios son:
- Caja de texto simple
 - Caja de texto multilínea
 - Grupo de opciones
 - Grupo de selección múltiple
- Módulo de beneficiarios: herramienta para llevar el control (crear, listar, modificar y eliminar) de los datos que se han registrado sobre los beneficiarios.
- Módulo de lugares: este módulo permitirá realizar un catálogo de ciudades y lugares donde cada lugar pertenecerá a una ciudad.

Esto será de mucha utilidad, porque los lugares serán asociadas a los eventos y los beneficiarios que pertenezcan a ellas.

Con la aplicación móvil se pretende cubrir las necesidades de 2 tipos de flujos de eventos que existen:

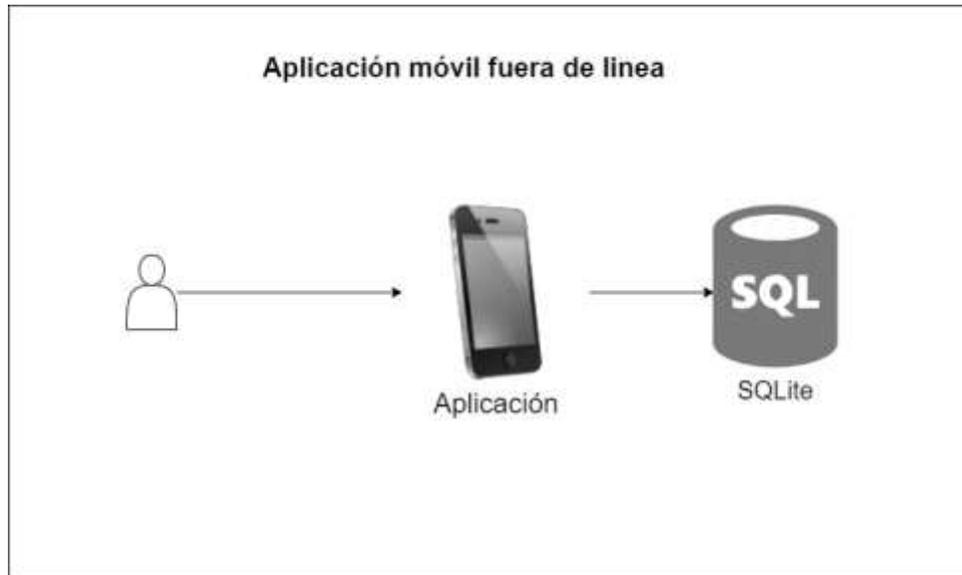
- Flujo de registro en evento de registro único: este evento es de registro único, lo que significa que los beneficiarios solo pueden tener un registro en el evento (ver apéndice 1).
- Flujo de registro en evento de registro libre: este evento es de libre registro, lo que significa que los beneficiarios pueden ingresar más de una vez al evento. Si el beneficiario no contara con su DPI por alguna razón de fuerza mayor o es menor de edad, se podrá crear un perfil de beneficiario con la llave: nombre, apellidos y fecha de nacimiento (ver apéndice 2).

Tanto el evento de registro único como el de registro libre necesitan un beneficiario a quien asignarle el registro, así que debe existir un perfil del beneficiario. En caso contrario, el usuario creará el perfil previo a realizar el registro.

1.3.2. Arquitectura de solución

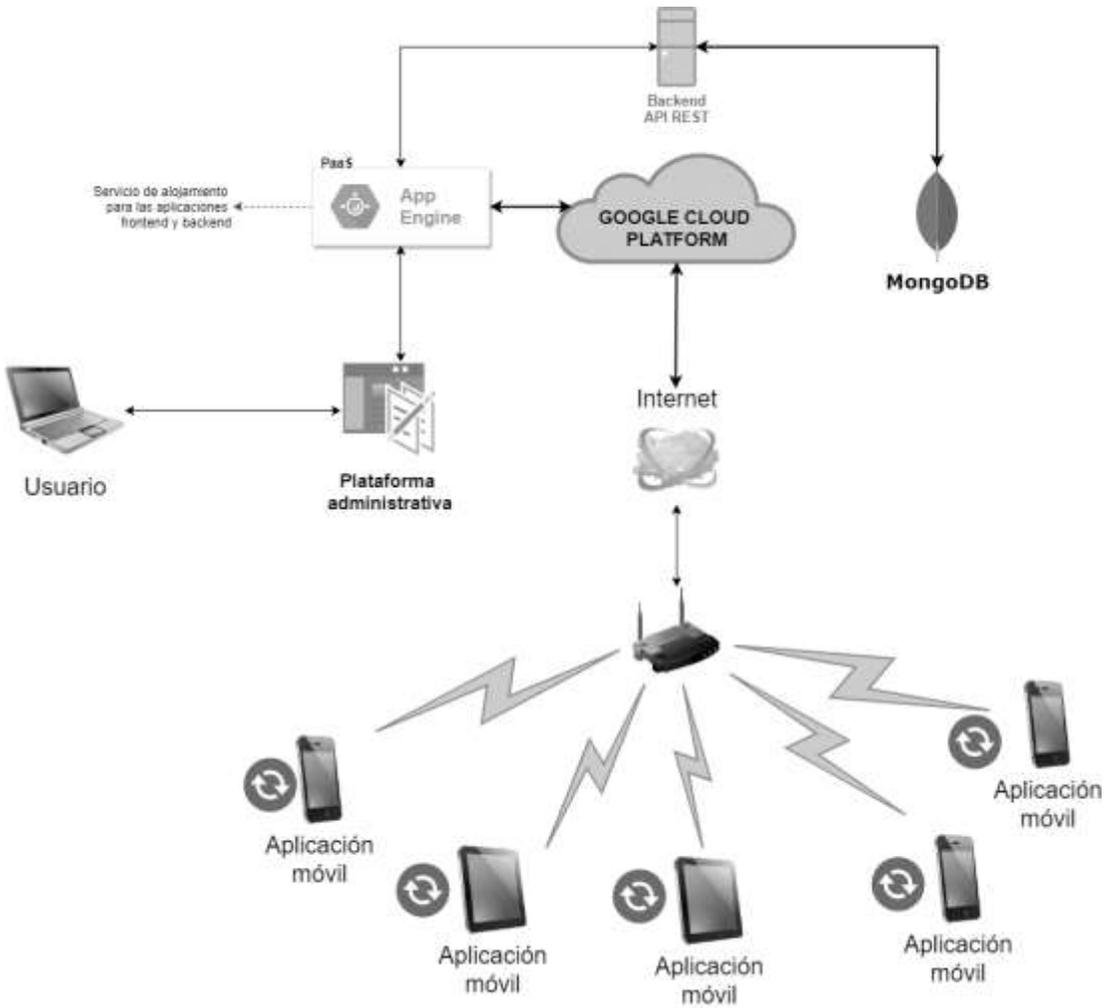
Se presentan los modelos de la arquitectura para el uso de la aplicación *offline*, *online*, y el sitio administrativo.

Figura 1. Registro fuera de línea



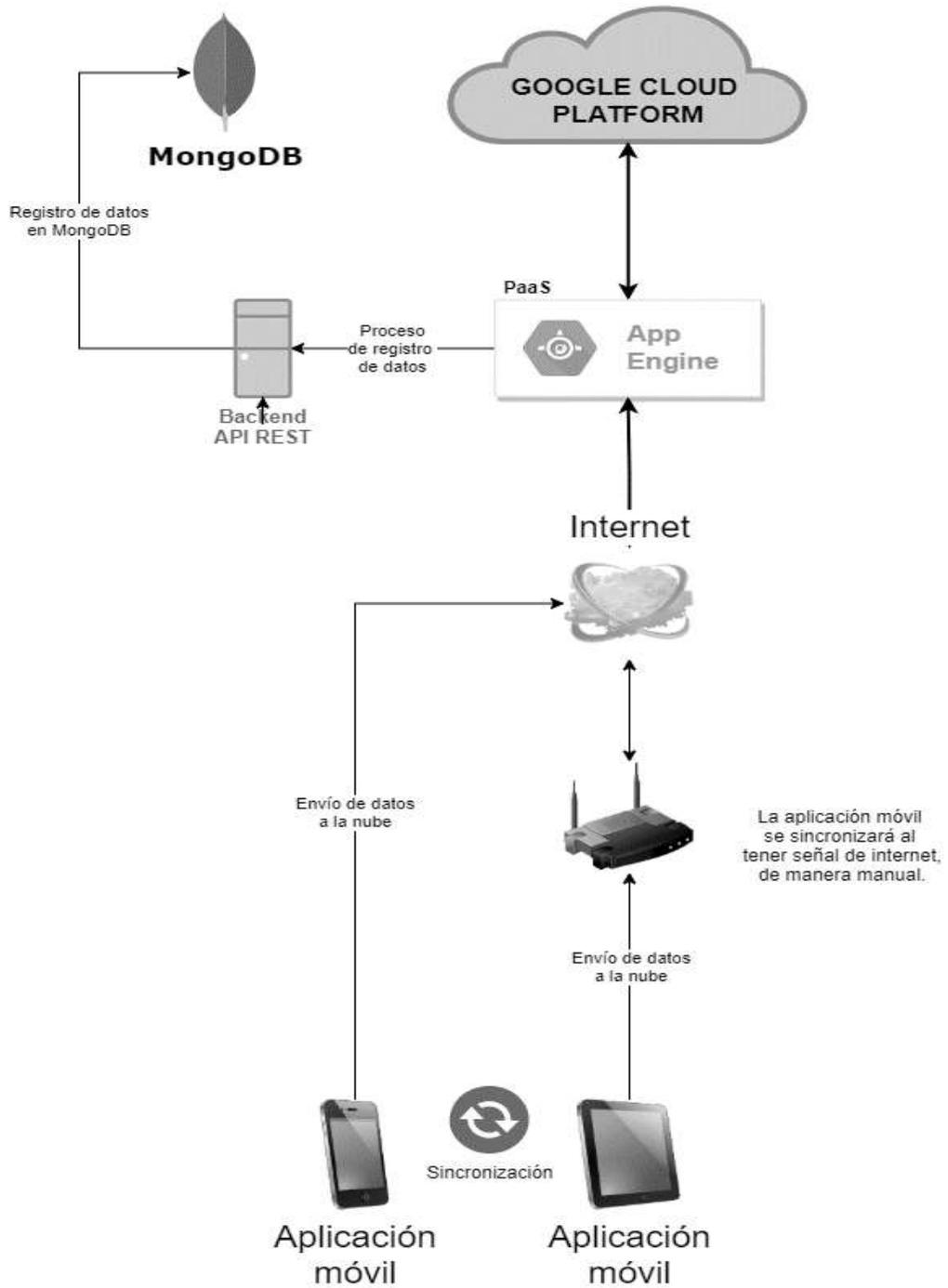
Fuente: elaboración propia, realizado con Draw.io.

Figura 2. Vista general de arquitectura en línea



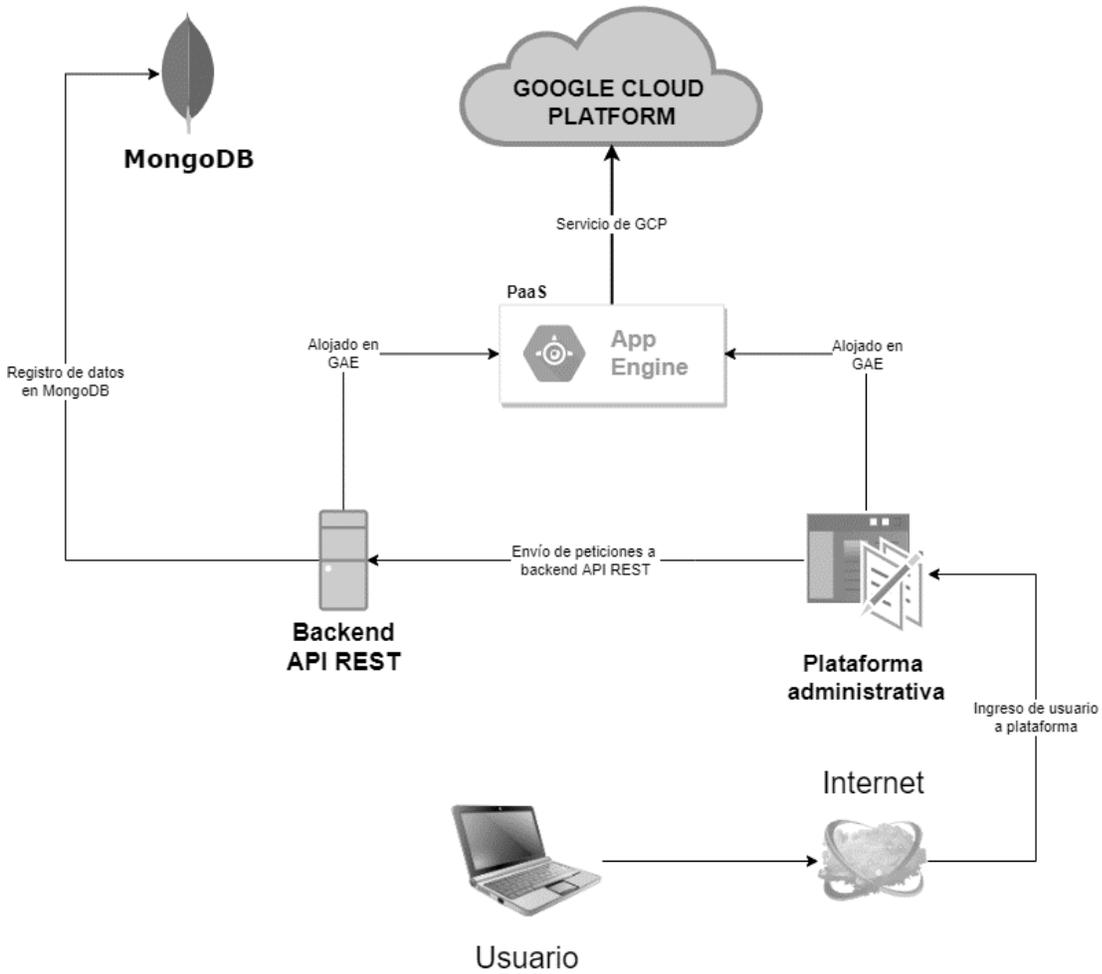
Fuente: elaboración propia, realizado con Draw.io.

Figura 3. Sincronización de datos de aplicación móvil



Fuente: elaboración propia, realizado con Draw.io.

Figura 4. **Gestiones de plataforma administrativa**



Fuente: elaboración propia, realizado con Draw.io.

1.3.3. Casos de uso

Se plantean los distintos casos de uso que se presentarán en la aplicación móvil y en el sitio administrativo.

1.3.3.1. Aplicación fuera de línea

Para el modo fuera de línea de la aplicación móvil, se crearán los módulos para ingreso al sistema, registro de perfil de beneficiario y registro de datos en eventos.

Tabla I. **CU1. Ingreso al sistema (o inicio de sesión)**

Nombre	Ingreso al sistema
Actores	Usuario de aplicación móvil
Contexto	Cuando el usuario necesita hacer el registro de un nuevo beneficiario del evento. Ingresará al sistema.
Descripción	Antes de iniciar cualquier acción en la aplicación, el usuario deberá hacer inicio de sesión, con su usuario y contraseña dados por el administrador.

Fuente: elaboración propia, realizado con Word.

Tabla II. **CU2. Listar beneficiarios**

Nombre	Listar beneficiarios
Actores	Usuario de aplicación móvil
Contexto	Cuando el usuario entre a la aplicación, y deba verificar la existencia de un beneficiario que llevo a registrarse al evento.
Descripción	La aplicación mostrará en el módulo de eventos, el listado de beneficiarios existentes disponibles para hacer el registro del evento.

Fuente: elaboración propia, realizado con Word.

Tabla III. **CU3. Crear beneficiario**

Nombre	Crear beneficiario
Actores	Usuario de aplicación móvil
Contexto	Cuando el usuario valide si existe el beneficiario, y no exista en el sistema, deberá crear un perfil.
Descripción	El usuario se dirige al módulo de eventos, y luego de seleccionar el evento, presiona el botón de crear nuevo beneficiario. Esta función permitirá ingresar los datos de un nuevo perfil de beneficiario.

Fuente: elaboración propia, realizado con Word.

Tabla IV. **CU4. Eliminar beneficiario**

Nombre	Eliminar beneficiario
Actores	Usuario de aplicación móvil
Contexto	El usuario de la aplicación necesita eliminar un perfil que ingreso con datos erroneos.
Descripción	El usuario de la aplicación se dirige al módulo de eventos, y busca al beneficiario que quiere eliminar en el listado de beneficiarios. El usuario confirma la eliminación del beneficiario. Solo se podrá eliminar el beneficiario si no se le han asociado registros.

Fuente: elaboración propia, realizado con Word.

Tabla V. **CU5. Listar eventos**

Nombre	Listar eventos
Actores	Usuario de aplicación móvil
Contexto	Cuando el usuario entre a la aplicación, previo a ingresar al beneficiario debe seleccionar el evento actual. Esto lo podrá hacer cuando liste los eventos activos.
Descripción	La aplicación mostrará al usuario el listado de eventos disponibles según la fecha actual, donde podrá seleccionar el evento correspondiente.

Fuente: elaboración propia, realizado con Word.

Tabla VI. **CU6. Seleccionar evento**

Nombre	Seleccionar evento
Actores	Usuario de aplicación móvil
Contexto	El usuario necesita seleccionar el evento activo de los que ha listado, al que asociaría un beneficiario para el nuevo registro.
Descripción	El usuario selecciona alguno de los eventos disponibles para poder registrar los datos del beneficiario en el formulario.

Fuente: elaboración propia, realizado con Word.

Tabla VII. **CU7. Registro de evento: asignar beneficiario**

Nombre	Realizar registro
Actores	Usuario de aplicación móvil
Contexto	Cuando el usuario ya ha seleccionado el evento en el que ingresará datos de beneficiarios, necesita buscar el perfil del beneficiario que llenará el formulario.
Descripción	En la primera sección para registro, habrá un espacio para buscar el perfil del beneficiario, realizando la búsqueda por su DPI o nombre, apellidos y fecha de nacimiento. Si existe el beneficiario, se procede a realizar el registro. Si no existe, se debe crear el perfil y luego de esto continuar con el registro.

Fuente: elaboración propia, realizado con Word.

Tabla VIII. **CU8. Registro de evento: llenar formulario**

Nombre	Realizar registro
Actores	Usuario de aplicación móvil
Contexto	Cuando el usuario ya ha seleccionado el evento en el que ingresará datos de beneficiarios, y ya encontró el perfil de beneficiario que asociará al registro.
Descripción	El usuario pasará a la siguiente sección donde podrá llenar los datos del formulario asociado al evento en curso.

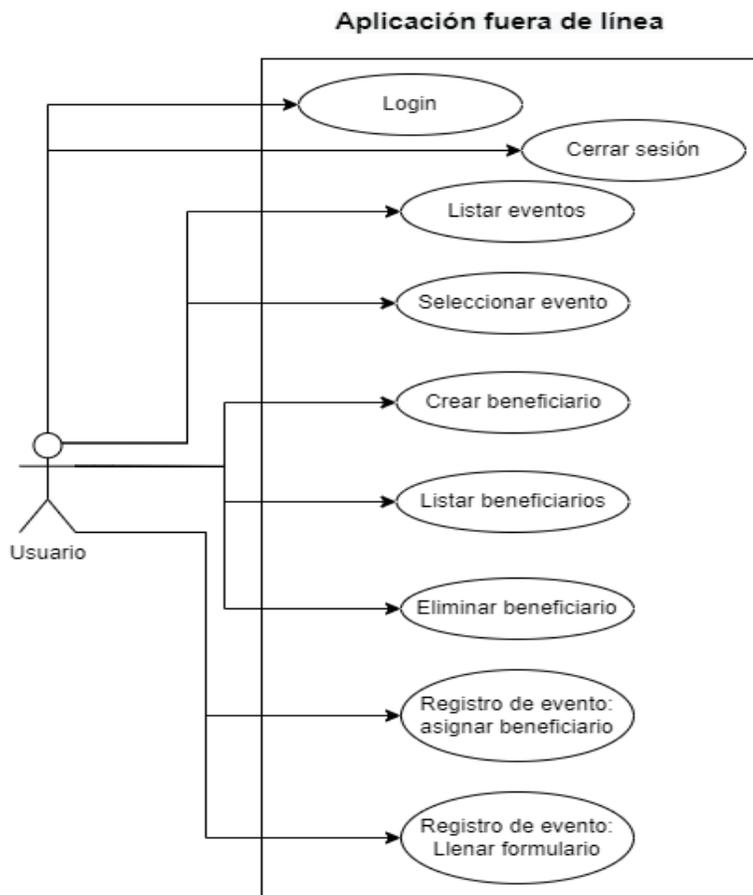
Fuente: elaboración propia, realizado con Word.

Tabla IX. **CU9. Cerrar sesión**

Nombre	Ingreso al sistema
Actores	Usuario de aplicación móvil
Contexto	Cuando el usuario desea finalizar la sesión actual.
Descripción	El usuario se dirige al módulo de ajustes donde estará el botón para cerrar sesión, lo presiona y confirma el cierre de sesión.

Fuente: elaboración propia, realizado con Word.

Figura 5. **Aplicación fuera de línea**



Fuente: elaboración propia, realizado con Draw.io.

1.3.3.2. Aplicación en línea

Para el modo en línea de la aplicación móvil, se aplican los mismos casos de uso que en el modo fuera de línea, además de los casos que se muestran a continuación.

Tabla X. **CU1. Sincronizar datos de aplicación**

Nombre	Sincronizar registros de eventos
Actores	Usuario de aplicación móvil
Contexto	Si la aplicación se utilizara para realizar registro de datos, se debe sincronizar con los datos actualizados en la nube. Esto debe realizarse previo a cualquier registro de evento para actualizar la base de datos de la aplicación fuera de línea.
Descripción	El usuario se dirige al módulo de ajustes y realiza la sincronización haciendo clic en el botón de sincronización de datos de aplicación. Esto actualizará la base de datos local para poder cargar los eventos y perfiles de beneficiario existentes.

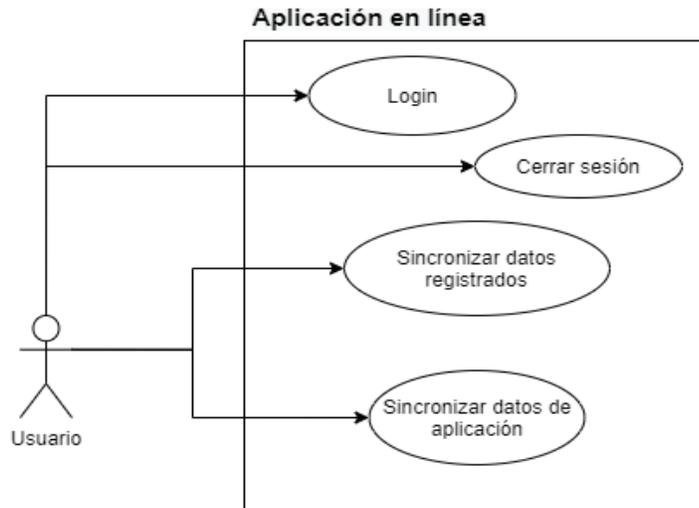
Fuente: elaboración propia, realizado con Word.

Tabla XI. **CU2. Sincronizar datos registrados**

Nombre	Sincronizar registros de eventos
Actores	Usuario de aplicación móvil
Contexto	El usuario ingresa al sistema y necesita realizar la carga de los registros de eventos y perfiles de beneficiarios
Descripción	El usuario se dirige al módulo de ajustes y realiza la sincronización haciendo clic en el botón sincronizar datos registrados.

Fuente: elaboración propia, realizado con Word.

Figura 6. **Aplicación en línea**



Fuente: elaboración propia, realizado con Draw.io.

1.3.3.3. **Sitio administrativo**

En el sitio administrativo, se plantean los casos de uso más complejos, ya que es donde se administrará toda la información de los usuarios, eventos, formularios, reportes, entre otros.

Tabla XII. **CU1. Recuperación de contraseña**

Nombre	Recuperación de contraseña
Actores	Usuario, usuario administrador y donadores (tipo de usuario)
Contexto	Si el usuario olvidó su contraseña, deberá tener la opción de recuperarla en la pantalla de ingreso al sistema, a través de su correo electrónico.
Descripción	El usuario tendrá la opción de recuperar su contraseña, en caso de no recordarla. Esta opción deberá aparecer en la pantalla inicial de ingreso.

Fuente: elaboración propia, realizado con Word.

Tabla XIII. **CU2. Ingreso al sistema (Login)**

Nombre	Ingreso al sistema (<i>Login</i>)
Actores	Cualquier usuario con acceso al sistema
Contexto	Cualquier usuario que deba hacer una gestión y tenga acceso al sitio administrativo podrá ingresar con su usuario y contraseña.
Descripción	El usuario tendrá la opción de ingresar al sistema administrativo utilizando su correo electrónico y contraseña. Esta opción deberá aparecer en la pantalla inicial de ingreso.

Fuente: elaboración propia, realizado con Word.

Tabla XIV. **CU3. Creación de rol**

Nombre	Creación de rol
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador será quien pueda agregar un nuevo rol de usuario en caso de necesitar uno nuevo con diferentes permisos a los que ya existen. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario tendrá la opción de crear un nuevo rol, y dar los permisos de vista, creación y edición en los diferentes módulos del sistema.

Fuente: elaboración propia, realizado con Word.

Tabla XV. **CU4. Edición de rol**

Nombre	Edición de rol
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador será quien pueda editar un rol de usuario en caso de querer cambiar sus permisos actuales. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario tendrá la opción de editar los permisos de un rol previamente creado.

Fuente: elaboración propia, realizado con Word.

Tabla XVI. **CU5. Listar roles**

Nombre	Listar roles
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador podrá listar los roles creados en el módulo de usuarios y roles.
Descripción	El usuario podrá listar los roles que han sido creados en el sistema. Se mostrará una lista con paginación y filtros de búsqueda para facilitar la búsqueda del rol.

Fuente: elaboración propia, realizado con Word.

Tabla XVII. **CU6. Eliminar roles**

Nombre	Eliminar roles
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador será quien pueda eliminar un rol de usuario. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario podrá seleccionar los roles que desee eliminar. Posteriormente podrá hacer clic en la opción eliminar y aceptar la acción.

Fuente: elaboración propia, realizado con Word.

Tabla XVIII. **CU7. Creación de usuario**

Nombre	Creación de usuario
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador será quien pueda agregar un nuevo usuario al sistema. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario tendrá la opción de crear un nuevo usuario, asignándole un correo electrónico y contraseña temporal.

Fuente: elaboración propia, realizado con Word.

Tabla XIX. **CU8. Edición de usuario**

Nombre	Edición de usuario
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador será quien pueda editar un usuario del sistema. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario tendrá la opción de editar la información de un usuario

Fuente: elaboración propia, realizado con Word.

Tabla XX. **CU9. Listar usuario**

Nombre	Listar usuarios
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador podrá listar a todos los usuarios del sistema. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario podrá listar los usuarios que han sido creados en el sistema. Se mostrará una lista con paginación y filtros de búsqueda para facilitar la búsqueda del usuario.

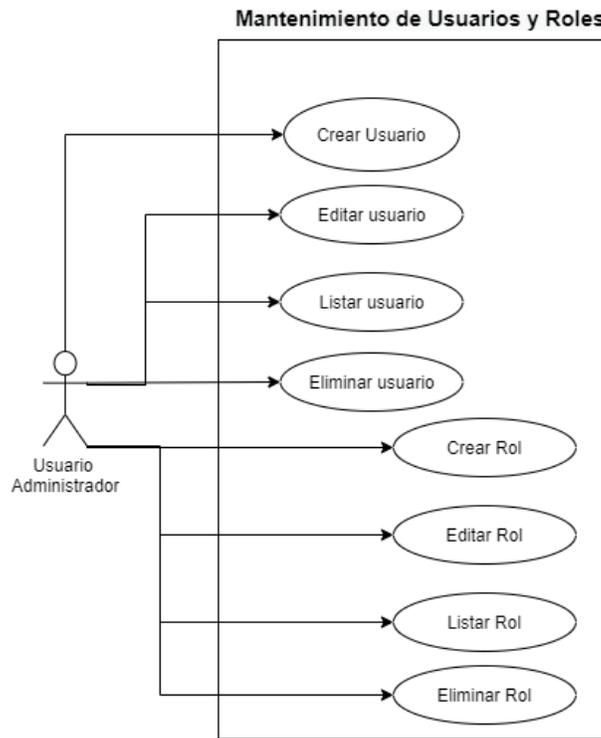
Fuente: elaboración propia, realizado con Word.

Tabla XXI. **CU10. Eliminar usuarios**

Nombre	Eliminar usuarios
Actores	Usuario administrador y usuario con estos permisos.
Contexto	El usuario administrador será quien pueda eliminar un usuario del sistema. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario podrá seleccionar los usuarios que desee eliminar. Posteriormente podrá hacer clic en la opción eliminar y aceptar la acción.

Fuente: elaboración propia, realizado con Word.

Figura 7. **Mantenimiento de usuarios y roles**



Fuente: elaboración propia, realizado con Draw.io.

Tabla XXII. **CU11. Creación de evento**

Nombre	Creación de evento
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando se necesite crear un evento nuevo, el usuario podrá ingresar al módulo de eventos para realizar esta acción.
Descripción	El usuario tendrá la opción de crear un nuevo evento, asignando la fecha o fechas entre las que se llevará a cabo y asignando un formulario previamente creado.

Fuente: elaboración propia, realizado con Word.

Tabla XXIII. **CU12. Edición de evento**

Nombre	Edición de evento
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando se necesite editar un evento previamente creado, el usuario podrá ingresar al módulo de eventos para realizar esta acción.
Descripción	El usuario tendrá la opción de editar la información de un evento. Así mismo cambiar las fechas establecidas en las que se llevará a cabo o reasignar otro formulario de registro.

Fuente: elaboración propia, realizado con Word.

Tabla XXIV. **CU13. Listar eventos**

Nombre	Listar eventos
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando se necesite ver los eventos creados, el usuario podrá ingresar al módulo de eventos para realizar esta acción.
Descripción	El usuario podrá listar los eventos que han sido creados en el sistema. Se mostrará una lista con paginación y filtros de búsqueda para facilitar la búsqueda del evento.

Fuente: elaboración propia, realizado con Word.

Tabla XXV. **CU14. Eliminar eventos**

Nombre	Eliminar eventos
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando se necesite eliminar un evento nuevo, el usuario podrá ingresar al módulo de eventos para realizar esta acción.
Descripción	El usuario podrá seleccionar los eventos que desee eliminar. Posteriormente podrá hacer clic en la opción eliminar y aceptar la acción.

Fuente: elaboración propia, realizado con Word.

Tabla XXVI. **CU15. Ver registros de evento**

Nombre	Ver registros de evento
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando el usuario necesite ver los datos que se han registrado en un evento específico, podrá ingresar al módulo de eventos para ver este detalle.
Descripción	El usuario podrá listar los datos registrados por los usuarios. Cuando se realicen todas las sincronizaciones, se podrán ver todos los datos que se obtuvieron durante el evento, seleccionando el evento específico y la opción "Ver registros".

Fuente: elaboración propia, realizado con Word.

Tabla XXVII. **CU16. Ver detalle de registro**

Nombre	Listar detalle de registro
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando el usuario liste los registros del evento, y necesite ver el detalle de los datos registrados por cada beneficiario.
Descripción	El usuario podrá listar los datos del formulario que se llenó en el registro del evento con la opción "Ver detalles" del registro listado.

Fuente: elaboración propia, realizado con Word.

Tabla XXVIII. **CU17. Descarga de registros de evento**

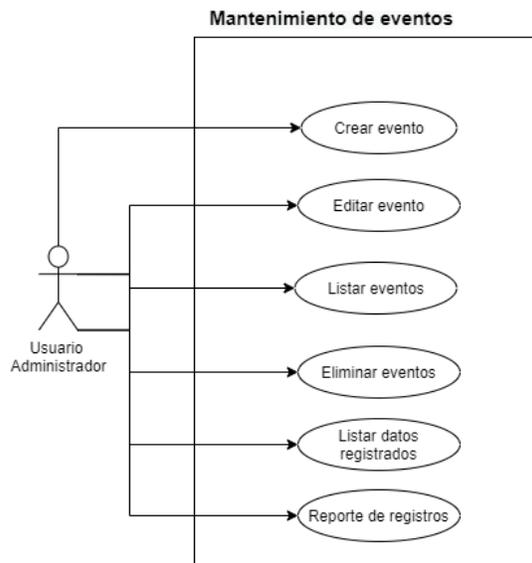
Nombre	Reporte de registros
Actores	Usuario administrador y usuario con estos permisos.
Contexto	En el módulo de eventos, luego de listar los registros del evento el usuario podrá descargar los datos registrados durante el evento.

Continuación de la tabla XXVIII.

Descripción	El usuario podrá descargar en un archivo los registros obtenidos durante el evento en el formato “.csv”, seleccionando el evento y la opción que permita la descarga del archivo.
-------------	---

Fuente: elaboración propia, realizado con Word.

Figura 8. **Mantenimiento de eventos**



Fuente: elaboración propia, realizado con Draw.io.

Tabla XXIX. **CU18. Creación de formulario**

Nombre	Creación de formulario
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando el usuario necesite crear un nuevo formulario de registro de datos para un evento.

Continuación de la tabla XXIX.

Descripción	El usuario tendrá la opción de crear un nuevo formulario. En el editor de formularios, se podrá dividir el formulario en secciones. Los tipos de datos que el formulario manejará son campos de texto, multilínea, numérico, campos de opción y selección múltiple.
-------------	---

Fuente: elaboración propia, realizado con Word.

Tabla XXX. **CU19. Edición de formulario**

Nombre	Edición de formulario
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando el usuario necesite editar un formulario de registro de datos previamente creado.
Descripción	El usuario tendrá la opción de editar la información de un formulario. En el editor de formularios, actualizar la información de las secciones creadas.

Fuente: elaboración propia, realizado con Word.

Tabla XXXI. **CU20. Listar formularios**

Nombre	Listar formularios
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando el usuario necesite listar los formularios de registro de datos previamente creados.
Descripción	El usuario podrá listar los formularios que han sido creados en el sistema. Se mostrará una lista con paginación y filtros de búsqueda para facilitar la búsqueda del formulario.

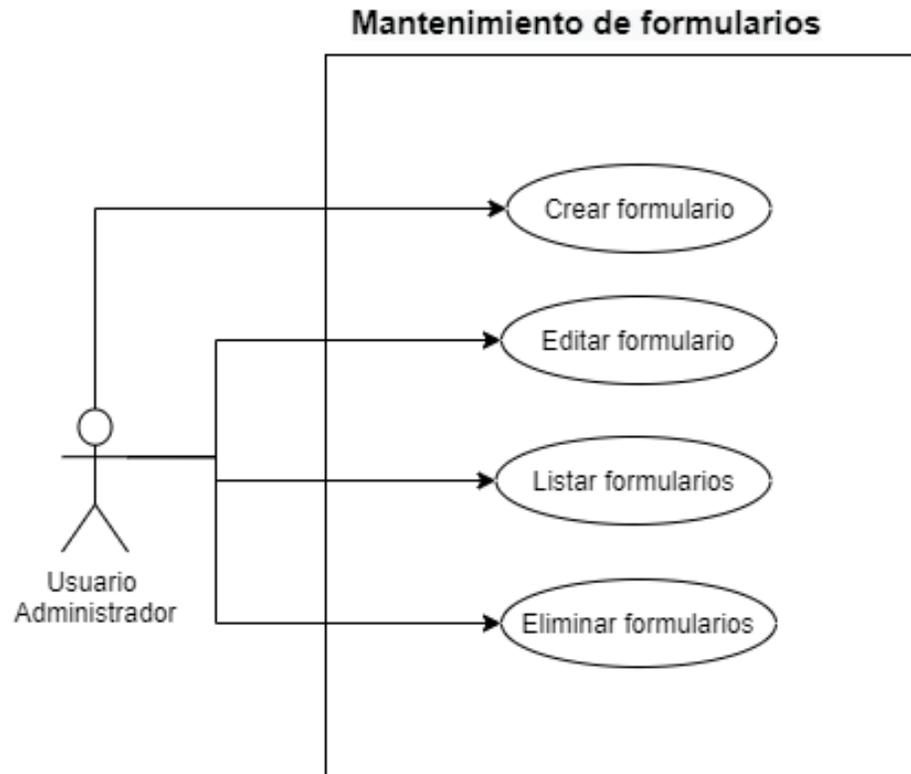
Fuente: elaboración propia, realizado con Word.

Tabla XXXII. **CU21. Eliminar formularios**

Nombre	Eliminar formularios
Actores	Usuario administrador y usuario con estos permisos.
Contexto	Cuando el usuario necesite eliminar un formulario de registro de datos previamente creado.
Descripción	El usuario podrá seleccionar los formularios que desee eliminar. Posteriormente podrá hacer clic en la opción eliminar y aceptar la acción.

Fuente: elaboración propia, realizado con Word.

Figura 9. **Mantenimiento de formularios**



Fuente: elaboración propia, realizado con Draw.io.

Tabla XXXIII. **CU22. Creación de beneficiario**

Nombre	Creación de beneficiario
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita registrar un nuevo perfil de beneficiario al sistema administrativo. Inicia sesión y se dirige al modulo de beneficiarios.
Descripción	El usuario selecciona la opción para crear un nuevo beneficiario. En esta ventana llenará todos los datos del nuevo perfil. En caso de que el beneficiario no cuente con DPI, el usuario proporcionará como llave el nombre completo, apellidos completos, y fecha de nacimiento del beneficiario. Luego de verificar estos datos, si el beneficiario no existe, el usuario procederá a registrar un nuevo perfil de beneficiario.

Fuente: elaboración propia, realizado con Word.

Tabla XXXIV. **CU23. Edición de beneficiario**

Nombre	Edición de beneficiario
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario del sistema administrativo ingresa al módulo de beneficiarios para realizar una actualización de los datos de un beneficiario en específico.
Descripción	El usuario del sistema administrativo selecciona el beneficiario que desea actualizar, y procede a realizar la acción, tomando en cuenta que el único campo que no podrá actualizar es el DPI (si posee).

Fuente: elaboración propia, realizado con Word.

Tabla XXXV. **CU24. Listar beneficiarios**

Nombre	Listar beneficiarios
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.

Continuación de la tabla XXXV.

Contexto	El usuario podrá listar a todos los beneficiarios del sistema. Esto lo podrá hacer en el módulo de beneficiarios.
Descripción	El usuario podrá listar los beneficiarios que han sido creados en el sistema. Se mostrará una lista con paginación y filtros de búsqueda para facilitar la búsqueda del beneficiario.

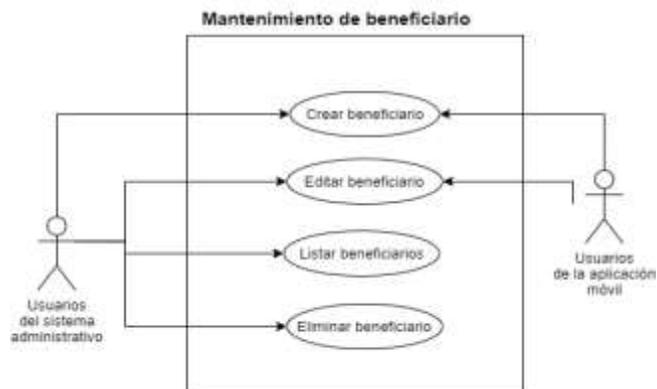
Fuente: elaboración propia, realizado con Word.

Tabla XXXVI. **CU25. Eliminar beneficiario**

Nombre	Eliminar beneficiario
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario será quien pueda eliminar un usuario del sistema. Esto lo podrá hacer en el módulo de usuarios y roles.
Descripción	El usuario podrá seleccionar los beneficiarios que desee eliminar. Posteriormente podrá hacer clic en la opción eliminar y aceptar la acción.

Fuente: elaboración propia, realizado con Word.

Figura 10. **Mantenimiento de beneficiario**



Fuente: elaboración propia, realizado con Draw.io.

Tabla XXXVII. **CU26. Creación de ciudad**

Nombre	Creación de ciudad
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita registrar una nueva ciudad al que pertenece un lugar donde se realizara un evento.
Descripción	El usuario ingresa a la aplicación y se dirige al módulo de lugares en el apartado de ciudades. Para agregar una nueva ciudad, selecciona la opción de agregar ciudad e ingresa los datos correspondientes. Por último, confirma la creación de la ciudad.

Fuente: elaboración propia, realizado con Word.

Tabla XXXVIII. **CU27. Edición de ciudad**

Nombre	Edición de ciudad
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita actualizar los datos de una ciudad.
Descripción	El usuario ingresa al sistema y se dirige al módulo de lugares en el apartado de ciudades. El usuario selecciona el lugar que desea modificar y selecciona la opción de edición. Luego de editar los datos necesarios, confirma la edición de la ciudad.

Fuente: elaboración propia, realizado con Word.

Tabla XXXIX. **CU28. Listar ciudades**

Nombre	Listar ciudades
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita ver la lista de ciudades disponibles en el sistema.

Continuación de la tabla XXXIX.

Descripción	El usuario se dirige al módulo de lugares en el apartado de ciudades, donde automáticamente se listarán las ciudades que se han creado.
-------------	---

Fuente: elaboración propia, realizado con Word.

Tabla XL. **CU29. Eliminar ciudad**

Nombre	Eliminar ciudad
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita eliminar una ciudad que no será utilizada.
Descripción	El usuario se dirige al módulo de lugares en el apartado de ciudades, donde selecciona la ciudad que desea eliminar y confirma la eliminación de la ciudad.

Fuente: elaboración propia, realizado con Word.

Tabla XLI. **CU30. Creación de lugar**

Nombre	Creación de lugar
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita registrar un nuevo lugar donde se realizará un evento.
Descripción	El usuario ingresa a la aplicación y se dirige al módulo de lugares en el apartado de lugares. Para agregar un nuevo lugar, selecciona la opción de agregar lugar. Por último, ingresa los datos correspondientes al lugar asignándola a una ciudad existente y confirma la creación.

Fuente: elaboración propia, realizado con Word.

Tabla XLII. **CU31. Edición de lugar**

Nombre	Edición de lugar
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita actualizar los datos de un lugar.
Descripción	El usuario ingresa al sistema y se dirige al módulo de lugares en el apartado de lugares. El usuario selecciona el lugar y selecciona la opción de edición. Luego de editar los datos necesarios, confirma la edición del lugar.

Fuente: elaboración propia, realizado con Word.

Tabla XLIII. **CU32. Listar lugares**

Nombre	Listar lugares
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita ver la lista de lugares disponibles en la ciudad seleccionada.
Descripción	El usuario se dirige al módulo de lugares en el apartado de lugares, donde automáticamente se listarán los lugares que se han creado. El usuario podrá aplicar algunos filtros para mejorar la búsqueda.

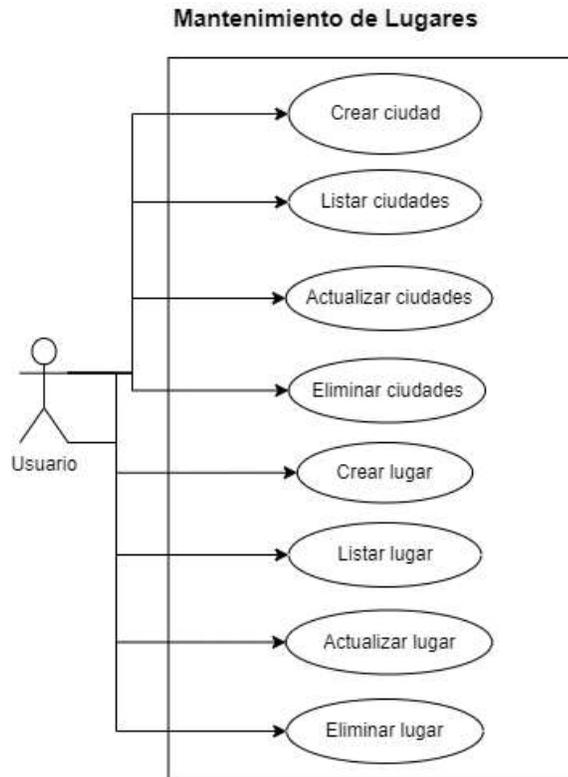
Fuente: elaboración propia, realizado con Word.

Tabla XLIV. **CU33. Eliminar lugar**

Nombre	Eliminar lugar
Actores	Usuario administrador, usuario de aplicación y usuario con estos permisos.
Contexto	El usuario necesita eliminar un lugar que no será utilizada.
Descripción	El usuario se dirige al módulo de lugares en el apartado de lugares, donde selecciona el lugar que desea eliminar y confirma la eliminación.

Fuente: elaboración propia, realizado con Word.

Figura 11. **Mantenimiento de lugares**



Fuente: elaboración propia, realizado con Draw.io.

Tabla XLV. **CU34. Ver resumen de actividad**

Nombre	Ver resumen de actividad
Actores	Cualquier usuario con acceso al sistema
Contexto	Cuando el usuario desee ver los datos más relevantes de la actividad reciente, relacionados con los beneficiarios y eventos.
Descripción	El usuario ingresará al sistema e inmediatamente se mostrará la sección de resumen de actividad. Aquí podrá encontrar los 6 resúmenes de los reportes relacionados con la actividad actual.

Fuente: elaboración propia, realizado con Word.

Tabla XLVI. **CU35. Reporte “Beneficiarios atendidos en el mes actual”**

Nombre	Reporte “Beneficiarios atendidos en el mes actual”
Actores	Cualquier usuario con permisos para ver reportes
Contexto	Cuando el usuario desee ver el detalle correspondiente al resumen de beneficiarios atendidos.
Descripción	El usuario ingresará al sistema y se dirigirá al apartado de reportes donde podrá ver el detalle de beneficiarios atendidos en el mes actual.

Fuente: elaboración propia, realizado con Word.

Tabla XLVII. **CU36. Reporte “Lugares poblados atendidos en el mes actual”**

Nombre	Reporte “Lugares poblados atendidos en el mes actual”
Actores	Cualquier usuario con permisos para ver reportes
Contexto	Cuando el usuario desee ver el detalle correspondiente al resumen de beneficiarios atendidos.
Descripción	El usuario ingresará al sistema y se dirigirá al apartado de reportes donde podrá ver el detalle de lugares poblados atendidos en el mes actual.

Fuente: elaboración propia, realizado con Word.

Tabla XLVIII. **CU37. Reporte “Eventos terminados del mes actual”**

Nombre	Reporte “Eventos terminados del mes actual”
Actores	Cualquier usuario con permisos para ver reportes
Contexto	Cuando el usuario desee ver el detalle correspondiente al resumen de eventos terminados del mes actual.
Descripción	El usuario ingresará al sistema y se dirigirá al apartado de reportes donde podrá ver el detalle de eventos terminados del mes actual.

Fuente: elaboración propia, realizado con Word.

Tabla XLIX. **CU38. Reporte “Eventos pendientes del mes actual”**

Nombre	Reporte “Eventos pendientes del mes actual”
Actores	Cualquier usuario con permisos para ver reportes
Contexto	Cuando el usuario desee ver el detalle correspondiente al resumen de eventos pendientes del mes actual.
Descripción	El usuario ingresará al sistema y se dirigirá al apartado de reportes donde podrá ver el detalle de eventos pendientes del mes actual.

Fuente: elaboración propia, realizado con Word.

Tabla L. **CU39. Reporte “Personas atendidas por tipo de evento por año”**

Nombre	Reporte “Personas atendidas por tipo de evento por año”
Actores	Cualquier usuario con permisos para ver reportes
Contexto	Cuando el usuario desee ver el detalle correspondiente al resumen de personas atendidas por tipo de evento por año.
Descripción	El usuario ingresará al sistema y se dirigirá al apartado de reportes donde podrá ver el detalle de personas atendidas por tipo de evento por año.

Fuente: elaboración propia, realizado con Word.

Tabla LI. **CU40. Reporte “Porcentaje de personas atendidas por tipo de evento por mes”**

Nombre	Reporte “Porcentaje de personas atendidas por tipo de evento por mes”
Actores	Cualquier usuario con permisos para ver reportes
Contexto	Cuando el usuario desee ver el detalle correspondiente al resumen del porcentaje de personas atendidas por tipo de evento por mes.
Descripción	El usuario ingresará al sistema y se dirigirá al apartado de reportes donde podrá ver el detalle del porcentaje de personas atendidas por tipo de evento por mes.

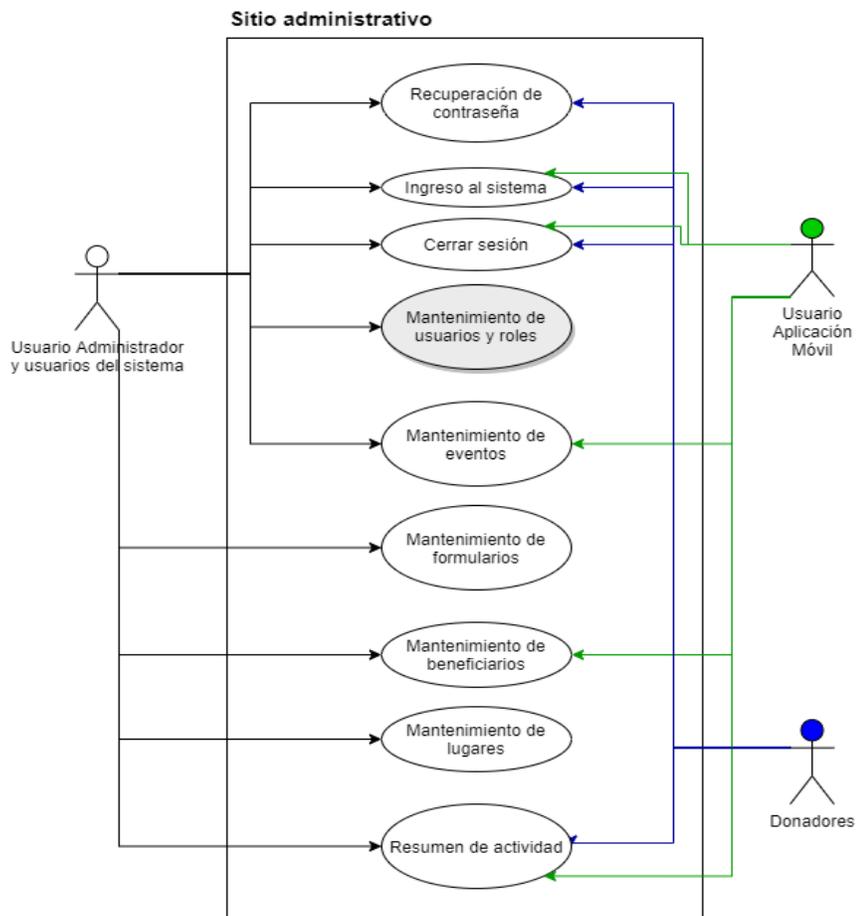
Fuente: elaboración propia, realizado con Word.

Tabla LII. **CU41. Cerrar sesión**

Nombre	Ingreso al sistema
Actores	Cualquier usuario con acceso al sistema
Contexto	Cuando el usuario desea finalizar la sesión actual.
Descripción	El usuario se dirige al módulo de ajustes donde estará el botón para cerrar sesión, lo presiona y confirma el cierre de sesión.

Fuente: elaboración propia, realizado con Word.

Figura 12. **Sitio administrativo**



Fuente: elaboración propia, realizado con Draw.io.

1.4. Definición de las tecnologías a utilizar

Se definen las diferentes tecnologías que se propone utilizar para crear la aplicación móvil y el sitio administrativo.

1.4.1. Tecnologías que se utilizaran para el desarrollo del *frontend*

Para la creación del *frontend*, existen muchas herramientas que pueden solventar la demanda del proyecto, pero se plantea utilizar las tecnologías descritas a continuación.

1.4.1.1. React

Es una biblioteca de JavaScript que permite construir de manera sencilla interfaces de usuario interactivas, dinámicas y ligeras. Es multiplataforma, basado en JavaScript y está rodeado de módulos y herramientas que pueden acoplarse al proyecto. Actualmente está disponible y estable la versión 16.

Desarrollada por Facebook, quienes son los que dan soporte de esta librería. React ha cobrado mucha importancia desde su lanzamiento en 2013, siendo una de las comunidades de desarrolladores más grandes y una de las herramientas de *frontend* más utilizadas en la actualidad.

Se escogió esta herramienta tomando en cuenta varios aspectos importantes, entre ellos, que su curva de aprendizaje es menor, es fácil de entender porque utiliza código de JavaScript y se basa en componentes, y la más importante para este proyecto, consume menos recursos en comparación con

otros *frameworks*. El uso de los recursos es muy importante ya que el lugar donde se implementará el sistema administrativo cuenta con poco ancho de banda.

1.4.1.2. Material-UI

Es un conjunto de componentes visuales de React. Esta librería implementa principios de *Material Design*. Está llena de herramientas que pueden utilizarse para desarrollar aplicaciones con una interfaz de usuario agradable.

1.4.1.3. Ionic

Es un *framework* de código abierto utilizado para la creación de aplicaciones móviles, tanto para Android como para iOS. Fue desarrollado sobre AngularJS y Cordova, en el año 2013. Para el desarrollo del proyecto se utilizará la versión 5 de Ionic, lanzada el 11 de febrero de 2020.

La evolución de Ionic hasta hoy en día ha sido tan grande, que actualmente es posible desarrollar la lógica en React, Vue, Angular o Javascript. Así que se creará la aplicación móvil por medio del lenguaje de preferencia, que en este caso será JavaScript por su amplia documentación y soporte.

También se decidió el uso de Ionic por su variedad de componentes gráficos predefinidos, por la facilidad de exportar la aplicación (para Android o iOS con la herramienta *Capacitor*), y porque permiten utilizar las herramientas nativas de los dispositivos, y por su fácil mantenimiento y amplia documentación.

1.4.1.4. JavaScript

Es un lenguaje de programación interpretado, es decir, que no necesita ser compilado para ejecutarse. Está basado en prototipos y funciona en los navegadores de forma nativa. El objetivo de utilizar este lenguaje es brindar a la interfaz de usuario mayor interactividad y dinamismo.

Adoptado como estándar en junio de 1997 por ECMA (*European Computer Manufacturers Association*), bajo el nombre de ECMAScript, ha ido evolucionando hasta la versión 7, lanzada el 1 de junio de 2016.

Se utilizará JavaScript en la aplicación móvil (Ionic). Esto permitirá tener un buen manejo de todas las funciones dinámicas que se necesiten hacer dentro de la aplicación. Adicionalmente es un lenguaje con mucho soporte técnico y muy conocido, lo que hará fácil darle mantenimiento.

1.4.1.5. NodeJS y npm

Tanto React como Ionic utilizarán módulos de NodeJS para su funcionamiento. NodeJS es un entorno de ejecución multiplataforma, de código abierto, que permite trabajar con JavaScript del lado del servidor. Está basado en el motor V8 de JavaScript de Google, para poder ejecutarse de manera mucho más rápida.

Por otro lado, npm (*node package manager*), es el sistema que administra los módulos de NodeJS. Esta herramienta permitirá instalar los módulos adicionales que se necesitarán en cada proyecto. Además, npm puede levantar un servidor local, una herramienta muy útil a la hora de realizar las pruebas.

1.4.1.6. Capacitor

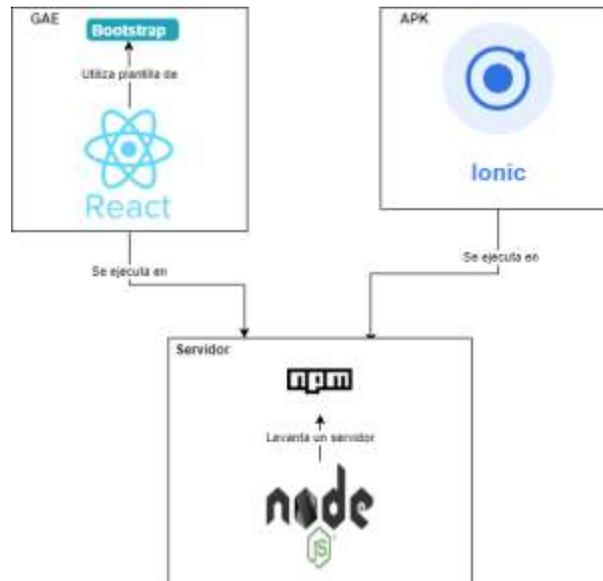
Es una herramienta de código abierto, que permite ejecutar aplicaciones web de manera nativa, en plataformas como iOS, Android, entre otras. En otras palabras, permite transformar una aplicación web, en aplicación móvil de una manera más sencilla.

Tabla LIII. **Resumen de tecnologías a utilizar para *frontend***

Tecnología	Función en la arquitectura
React	Plataforma web administrativa
Material-UI	Componentes visuales de React
Ionic	Aplicación móvil
Javascript	Lenguaje de programación utilizado por Ionic y React
NodeJS	Entorno de ejecución para javascript, que se utilizará para ejecutar la aplicación móvil y la plataforma administrativa.
npm	Gestión de paquetes para NodeJS
Capacitor	Herramienta para generar la aplicación móvil para iOS y Android.

Fuente: elaboración propia, realizado con Word.

Figura 13. Componentes de *frontend*



Fuente: elaboración propia, realizado con Draw.io.

1.4.2. Tecnologías que se utilizarán para el desarrollo del *backend*

Para la creación del *backend*, según los diferentes factores que demanda el proyecto se plantea utilizar las tecnologías descritas a continuación.

1.4.2.1. Flask

Es un *framework* escrito en Python, que facilita el desarrollo de las aplicaciones utilizando el patrón MVC. Es una herramienta de código abierto y posee mucha documentación, siendo la última versión la 1.1.2, que es la que se utilizará para este proyecto.

Flask es pequeño y ligero, pero posee las herramientas suficientes para desarrollar una aplicación potente. Para este proyecto, se utilizará Flask como *backend*, desarrollando un API Rest que pueda tener comunicación con el sitio administrativo y con la aplicación móvil.

Cabe mencionar que Flask no es *full stack*, como sí lo son otros *framework*, lo que es una ventaja debido a que no se necesita más que las funcionalidades básicas para desarrollar una API Rest.

1.4.2.2. Python

Es un lenguaje de programación interpretado, que nace a principios de los 90 y tiene como filosofía tener un código legible y elegante. Es multiparadigma, tomando en cuenta que soporta orientación a objetos, programación imperativa, programación funcional, entre otras. Python es multiplataforma, por lo que puede ejecutarse en donde se necesite.

Durante varios años, Python se ha posicionado como uno de los lenguajes que poco a poco ha ido cobrando más importancia, manteniéndose hasta ahora entre los primeros lugares como uno de los más utilizados, y con una gran comunidad en la que se puede encontrar soporte técnico.

Python será el lenguaje que se utilizará junto con el Flask, para realizar el API Rest. Su simple sintaxis y legibilidad permitirán ahorrarse muchas líneas de código y crear un *backend* funcional, de fácil mantenimiento y orientado a microservicios.

1.4.2.3. MongoDB

Es una base de datos NoSql orientada a documentos. Cada colección de datos posee varios documentos que pueden tener esquemas diferentes. Su entorno en la nube, llamado MongoDB Atlas posee un entorno gratuito ideal para proyectos en desarrollo cuya información no sobrepasa los 512Mb. MongoDB brinda una alta escalabilidad y flexibilidad, como también un modelo de consultas e indexación.

Se utilizará MongoDB para guardar toda la información recopilada por el sistema, tanto de la aplicación móvil para los registros de las personas, como en el sitio administrativo para las configuraciones del sistema, formularios, eventos, entre otras.

Se consideró el uso de MongoDB por sobre otras herramientas por que permitirá recuperar la información almacenada de una forma sencilla. Esto debido a que se necesita realizar distintos análisis de datos y consultas donde las herramientas de mongo serán de mucha utilidad y facilitarán esta tarea. Se espera que mientras se realiza la implementación del sistema, no se sobrepase el limite gratuito que MongoDB ofrece que es de 512 Mb. Los detalles de estos costos se amplían en la sección 1.4.4.

1.4.2.4. Autenticación con Firebase

Esta es una solución que permite almacenar los datos de los usuarios de manera segura. Proporciona diferentes servicios de *backend* que facilitan muchas de las tareas de autenticación, como la administración de usuarios, manejo de token de autenticación para verificar la identidad del usuario e incluso maneja el envío de correo electrónico para restablecer la contraseña.

Para este proyecto, se utilizará la herramienta de autenticación de Firebase basada en correo electrónico y contraseña, tanto para la aplicación móvil como del sitio administrativo. Se propone esta tecnología tomando en cuenta que también existe un plan sin cargos y límite de cuotas que se detalla en la sección 1.4.4.

1.4.2.5. SQLite

Esta herramienta es una base de datos muy ligera, de código libre, pero que al mismo tiempo es muy potente y rápida que puede ejecutarse en dispositivos con poca capacidad de procesamiento. Es multiplataforma, su última versión estable es la 3.33 lanzada en agosto de 2020.

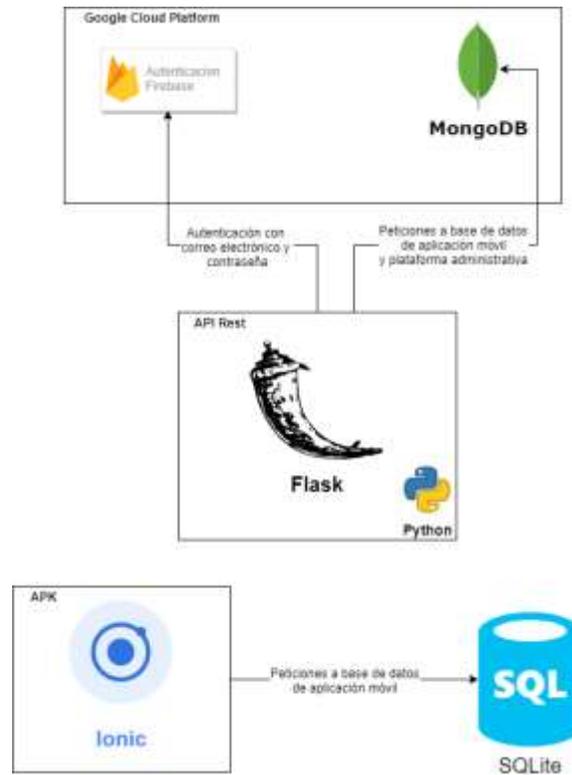
Ionic cuenta con un *plugin* nativo de SQLite, que puede implementarse de forma *offline*. Así que el uso de este será específicamente para la aplicación móvil, tomando en cuenta las necesidades que deben ser resueltas en el sistema actual. Esto permitirá almacenar la información de los registros de formularios que se completen durante los eventos, para luego realizar una transformación de la información obtenida y sincronizarla a la base de datos en la nube.

Tabla LIV. **Resumen de tecnologías a utilizar para *backend***

Tecnología	Función en la arquitectura
Flask	<i>Framework</i> para Python, con el que se desarrollará el API Rest
Python	Lenguaje de programación para el <i>backend</i> .
MongoDB	Base de datos NoSQL, alojada en Google Cloud Platform
Autenticación de Firebase	Herramienta para manejar la autenticación con contraseña.
SQLite	Base de datos SQL ligera, para almacenar datos temporales en aplicación móvil

Fuente: elaboración propia, realizado con Word.

Figura 14. Componentes de *backend*



Fuente: elaboración propia, realizado con Draw.io.

1.4.3. Otras tecnologías y servicios

Para llevar a cabo el desarrollo de las aplicaciones, se utilizarán algunas herramientas extras, y se describen en esta sección.

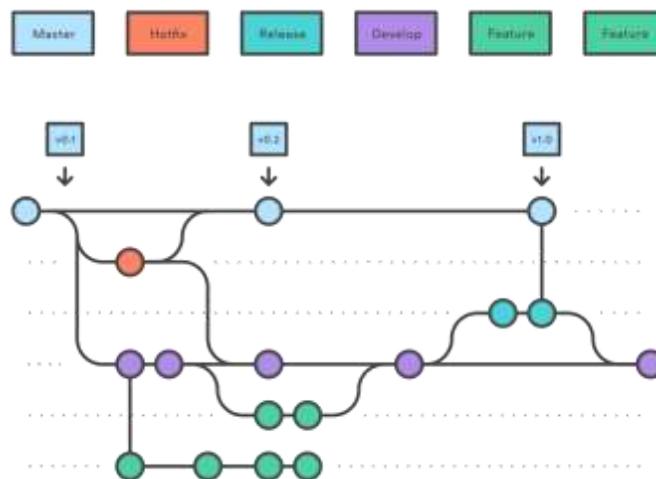
1.4.3.1. Git

Es un *software* de control de versiones de código de forma distribuida, que fue diseñado por Linus Torvalds, lanzado en abril de 2005. El propósito de este es llevar el control de cambios de los archivos compartidos dentro de un repositorio de código.

Git es multiplataforma, muy potente y rápido, y utilizarlo como repositorio permitirá tener el historial completo de los cambios que se realicen en el proyecto. No habrá problema para acceder al código desde cualquier lugar, al no depender de un repositorio central. Se utilizará la última versión estable, siendo actualmente la 2.30.2.

Como flujo de trabajo, se propone utilizar Git-Flow, ya que el proyecto tendrá entregables y ciclos de desarrollo bien definidos. Por lo que se seguirá esta metodología utilizando las siguientes ramas: maestra (*master*), desarrollo (*development*), características (*features*), revisión (*hotfix*), y versión (*release*).

Figura 15. Flujo de Git-Flow



Fuente: Atlassian (2021). *Flujo de trabajo de Gitflow*. Consultado el 17 de septiembre de 2021. Recuperado de <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>.

1.4.3.2. Google App Engine

Es un servicio de Google Cloud Platform que permite alojar aplicaciones sobre la infraestructura de Google. Posee dos entornos de ejecución: *Standard*

Environment para una cantidad limitada de lenguajes de programación y *Flexible Environment* para implementar cualquier otro lenguaje que no esté dentro de los que soporta el entorno standard.

Se elige esta opción ya que para el *frontend* (NodeJS) y *backend* (Python), se puede utilizar el entorno Standard. Utilizar este entorno brindará muchos beneficios. Por un lado, se aprovecharán las cuotas gratuitas que brinda el servicio de acuerdo con lo establecido por GCP. Por otro lado, GAE permite crear aplicaciones escalables, sin preocuparse por la infraestructura.

Otra de las cosas importantes es que no se necesitará un servidor dedicado al *frontend* o *backend*. Se tendrá un mejor control de los despliegues realizados y el proceso de despliegue será más sencillo.

Los detalles de estos costos se amplían en la sección 1.4.4.

1.4.4. Proyección de costos

Se muestra el análisis de los costos que representará el desarrollo y puesta en marcha del proyecto.

1.4.4.1. Análisis de costos por uso de tecnologías en la nube

Existe una capa gratuita que GCP ofrece a todos los usuarios, la cual cubre una variedad de servicios. Esto significa que cualquier usuario puede aprovechar estos servicios sin costo, mientras se mantenga dentro de las cuotas que GCP establece en la capa gratuita. Si el usuario sobrepasa esas cuotas, GCP cobrará el consumo extra de servicios.

Para autenticar los usuarios se utilizará el servicio de autenticación de Firebase, el cual tiene una capa sin cargo (Plan Spark), y capa de pago por uso (Blaze), que incluye el uso gratuito del plan Spark.

Se propone realizar la autenticación aprovechando la capa gratuita durante el desarrollo del proyecto.

Se muestra el detalle de precios de las soluciones de GCP que se implementarán en el proyecto:

Figura 16. Precios de autenticación con Firebase

Productos	Sin cargo Plan Spark <small>Generous limits to get started!</small>	Pago por uso Plan Blaze <small>Calcula los precios de las apps a gran escala. ✓ Se incluye el uso gratuito del plan Spark!</small>
Authentication		
Autenticación telefónica: Canadá, EE.UU. y la India 🇨🇦 🇺🇸 🇮🇳	10,000 por mes	USD 0.01 por verificación
Autenticación telefónica: Todos los demás países 🌐	10,000 por mes	USD 0.04 por verificación
Otros servicios de autenticación	✓	✓

Fuente: Google (2021). *Planes de precios*. Consultado el 17 de septiembre de 2021.

Recuperado de <https://firebase.google.com/pricing?hl=es-419>.

Con respecto a la base de datos, se propone utilizar la capa gratuita que brinda MongoDB para la plataforma administrativa y el *backend*, que estarán desplegados en un producto llamado *Google App Engine* de *Google Cloud Platform*.

Figura 17. Precios de MongoDB Atlas

Clúster compartido

¡Paga por uso! Se le facturará por hora con facturas mensuales.

 Servicios web de Amazon
  Microsoft Azure
  Plataforma de Google Cloud

Mongo branch: shared

Clúster	Almacenamiento	CARNERO	vCPU	Precio base
M0	512 MB	Compartido	Compartido	Gratis para siempre
M0	2 GB	Compartido	Compartido	\$5/mes
M6	5 GB	Compartido	Compartido	\$25/mes

Fuente: Mongoddb (2021). *Precios MongoDB*. Consultado el 17 de octubre de 2021. Recuperado de <https://www.mongodb.com/es/pricing>.

Figura 18. Precios de App Engine (capa gratuita)

 **App Engine**

28 horas de instancia por día

Plataforma para compilar aplicaciones web escalables y backends para dispositivos móviles

28 horas diarias de instancias "F"

9 horas diarias de instancias "B"

1 GB de salida por día

El nivel gratuito solo está disponible para el entorno estándar

Fuente: Google (2021). *Productos del nivel gratuito*. Consultado el 17 de septiembre de 2021. Recuperado de <https://cloud.google.com/free?hl=es-419>.

1.4.4.2. Presupuesto a corto, mediano y largo plazo

Se estima que, durante el tiempo de desarrollo e implementación (corto plazo), las cuotas se mantengan dentro de la capa gratuita para los servicios que se utilizarán.

Durante el primer año, se estima que las cuotas se mantengan debido a que la cantidad promedio de personas atendidas mensualmente no sobrepasa el límite de cuotas diarias de los servicios.

A futuro, se espera que el costo de almacenamiento crezca. Sin embargo, se estima que el espacio disponible será suficiente para cubrir primeros 2 o 3 años de información, pero se supondrá que el almacenamiento anual aumentará 256 Mb por año.

Tabla LV. Precios de GAE para plataforma administrativa y *backend*

Plazo	Google App Engine Plataforma administrativa	Google App Engine <i>Backend</i> API Rest	Total
Corto (6 meses)	Plataforma administrativa: • \$0.13 /mes	<i>Backend</i> API Rest: • \$0.07 /mes	\$1.2
Mediano (1 año)	Plataforma administrativa: • \$0.13 /mes	<i>Backend</i> API Rest: • \$0.07 /mes	\$2.4
Largo (5 años)	Plataforma administrativa: • \$0.15 /mes	<i>Backend</i> API Rest: • \$0.10 /mes	\$15

Fuente: elaboración propia, realizado con Word.

Tabla LVI. **Precios de Firebase y MongoDB**

Plazo	Cloud Firebase	MongoDB	Total
Corto (6 meses)	\$0.04 /mes	\$0.00 /mes	\$0.24
Mediano (1 año)	\$0.04 /mes	\$0.00 /mes	\$0.48
Largo (5 años)	\$0.08 /mes	Almacenamiento aproximado: 2Gb \$9 cada mes	4.8 Firebase + \$9 mongodb \$13.8

Fuente: elaboración propia, realizado con Word.

1.4.4.3. **Plan de financiamiento de servicios para la solución**

Se aprovecharán los beneficios gratuitos de GCP y de MongoDB Atlas, durante el desarrollo e implementación del proyecto se prevé que no habrá costos altos que puedan comprometer el proyecto.

Tabla LVII. **Cuota por servicios GCP y Firebase**

Cuota por servicio		
App engine	\$ -	1 instancia por proyecto
Firestore	\$ 0.48	Autenticacion(sin costo) + servicios adicionales de cloud
Otros servicios Cloud	\$ 2.40	Cada año (desde el 2do año de uso)
MongoAtlas	\$ 108.00	2GB (Desde el 3er año en adelante)

Fuente: elaboración propia, realizado con Excel.

Para el cálculo de estos costos se utilizó una calculadora de servicios que GCP tiene disponible en su sitio web. Esto dio un aproximado de los consumos

mensuales de los servicios y se entiende que no se necesitaran más recursos de los que las capas gratuitas ofrecen.

Se muestra una proyección a 5 años de los servicios que se utilizaran:

Tabla LVIII. **Proyección de costos de servicios GCP y otros**

Proyección a 5 años							
Servicios GCP y otros							
GCP	Tipo Servicio	2022	2023	AÑO			Sub total
				2024	2025	2026	
App Engine	Alojamiento Frontend	\$ 1.56	\$ 1.72	\$ 1.89	\$ 2.08	\$ 2.28	\$ 9.52
	Alojamiento Backend	\$ 0.84	\$ 0.92	\$ 1.02	\$ 1.12	\$ 1.23	\$ 5.13
Firebase	Servicio autenticacion	\$ 0.48	\$ 0.48	\$ 0.48	\$ 0.48	\$ 0.48	\$ 2.40
Despliegue de app	Playstore	\$ -	\$ -	\$ 25.00	\$ 25.00	\$ 25.00	\$ 75.00
	Appstore	\$ -	\$ -	\$ 100.00	\$ -	\$ -	\$ 100.00
MongoDB Atlas	Alojamiento datos	256 GB	512 GB	768 MB	1024 MB	2 GB	\$ -
		\$ -	\$ -	\$ 108.00	\$ 108.00	\$ 108.00	\$ 324.00
						Total	\$ 516.05

Fuente: elaboración propia, realizado con Excel.

Cualquier costo extra a la estimación realizada durante el periodo de desarrollo hasta la entrega oficial del proyecto planteado, será cubierto por el autor y desarrollador de la tesis.

La publicación de la aplicación móvil para Android se realizará en Google Drive de la institución. En la estimación se plantea realizar la publicación de la aplicación para Android y iOS desde el 3er año en adelante.

1.4.4.4. Propuesta de mantenimiento y garantía del sistema

Adicional a la entrega de la solución, se propone brindar soporte técnico durante 3 meses como mantenimiento del sistema, para solventar cualquier inconveniente que pueda surgir.

El mantenimiento de 3 meses, posterior a la entrega final, consistirá en:

- Realizar una revisión quincenal de código, para poder solventar errores reportados y actualizaciones que se puedan hacer sin cambiar el alcance del proyecto.
- Realizar los despliegues de las aplicaciones que necesiten ser actualizadas por algún cambio en el código.
- Actualizar el repositorio de código con la última versión desplegada.

Todos estos puntos están incluidos en la carta de compromiso del apéndice 3.

2. VENTAJAS Y DESVENTAJAS DE LA SOLUCIÓN PROPUESTA

El contenido de este capítulo ayudará a tener una idea de las ventajas y desventajas de la solución que se propone para el problema planteado. Se evalúan otras opciones que se podrían tomar y se justifica el uso de las herramientas elegidas para el desarrollo de la solución.

2.1. Ventajas y desventajas del sistema actual

Dentro de las ventajas y desventajas que se pueden mencionar del sistema actual, están:

- **Ventajas**
 - No se necesita luz eléctrica
 - No se necesitan aparatos electrónicos para el registro
 - Se pueden realizar anotaciones extras que no estén dentro del formato del formulario.

- **Desventajas**
 - Datos ilegibles o boletas mal impresas
 - Perdida, maltrato o deterioro de boletas
 - Procesar los resultados según la información obtenida pueden tomar mucho tiempo.
 - Algunas boletas no pueden reutilizarse.

- Las boletas pueden ocupar demasiado espacio físico y eso dificulta su traslado.
- Se debe llevar un número considerable de boletas para que no falten durante el registro.
- Aunque se puedan realizar anotaciones adicionales en las boletas, puede que el espacio no sea suficiente.

2.2. Ventajas y desventajas de las herramientas para el *frontend*

En esta sección se analizan las diferentes ventajas y desventajas del *frontend*.

2.2.1. React vs otras tecnologías para *frontend*

Se analizó React y todas sus características respecto a las de otras tecnologías parecidas.

2.2.1.1. React

- Ventajas
 - Es muy ligero, consume pocos recursos
 - Una gran comunidad de desarrolladores
 - Soporte dado por Facebook
 - Buena documentación, lo suficientemente clara para iniciar rápido un proyecto.
 - Muy buena flexibilidad y respuesta dinámica en la interfaz de usuario.
 - Bastante rápido por la implementación del *Virtual DOM*.

- Puede trabajar con alta carga sin problema.
 - Es de código abierto, con frecuentes actualizaciones y mejoras.
 - La migración entre versiones es muy sencilla.
 - Curva de aprendizaje menor, respecto a las demás.
- Desventajas
 - No es un *framework* completo, así que no posee todas las características de un *framework*. Debe adaptar las otras características conforme las necesite.
 - No posee un estándar de desarrollo.

2.2.1.2. Vue.js

- Ventajas
 - HTML empoderado, lo que lo hace muy similar a Angular.
 - Muy buena documentación, aunque tiene pocos años de haber sido lanzado.
 - Adaptabilidad, por sus similitudes con otros *frameworks*.
 - Alta escalabilidad y reutilización de código.
 - A diferencia de Angular, Vue también es ligero y rápido. Utiliza pocos recursos y es flexible.
 - Curva de aprendizaje baja.
- Desventajas
 - Es un lenguaje muy joven, aún está empezando a cobrar importancia, aunque se ha vuelto muy popular.

- Muchos de sus recursos están en fase inicial.
- Vue tiene mucho terreno por explorar, y cada día surgen problemas de implementación que aún no se resuelven.

2.2.1.3. Angular

- Ventajas
 - Angular es el más maduro de los *frameworks*. Tiene mucho respaldo y soporte por la cantidad de contribuyentes activos.
 - Typescript también posee mucho soporte y amplia documentación.
 - MVVM (Model-View-ViewModel), que permite que el desarrollador trabaje por separado en la misma sección de la aplicación con el mismo conjunto de datos.
 - Soporte dado por Google.
 - Su arquitectura brinda al proyecto una alta escalabilidad.
- Desventajas
 - Su curva de aprendizaje es muy empinada.
 - Sus diversas estructuras hacen que la comprensión sea difícil.
 - Su sintaxis puede ser compleja.
 - El rendimiento puede ir más lento puesto que consume más recursos.

Se elige utilizar React para la plataforma administrativa por ser más ligero, tener un buen soporte, documentación y una curva de aprendizaje baja. Además, React consumirá menos recursos y esto beneficiará a la fundación por algunas

limitaciones que tienen con respecto al consumo de internet, debido a su ubicación geográfica.

Tabla LIX. **Ventajas y desventajas de React**

Tecnología	Ventajas	Desventajas
React	Ligero y rápido. Buen soporte. Buena documentación Código abierto. Fácil de migrar. Curva de aprendizaje baja.	No es un <i>framework full stack</i> . Sin estándares.
Vue.js	Poca documentación. Escalable. Reutilización de código. Ligero y rápido.	Curva de aprendizaje baja. Es un lenguaje muy joven. No tiene mucho soporte.
Angular	<i>Framework</i> maduro. Uso de TypeScript Buena documentación. Soporte dado por Google. Alta escalabilidad.	Su curva de aprendizaje alta. Difícil comprensión y sintaxis. Consume más recursos.

Fuente: elaboración propia, realizado con Word.

2.2.2. Ionic vs otras tecnologías para construir aplicaciones móviles

Se analizó Ionic y todas sus características respecto a las de otras tecnologías parecidas.

2.2.2.1. Ionic

- Ventajas

- El desarrollo se hace con html, css y js, todos lenguajes muy conocidos.
 - Soportado por plataformas Android y iOS.
 - Permite crear una aplicación híbrida a partir de HTML5, por lo que el proceso de desarrollo se hace una sola vez, para Android, iOS y Web.
 - Se puede implementar en diferentes *frameworks*, incluyendo Angular, Vue, React o simplemente usar JavaScript.
 - Ofrece mucha simplicidad, omite muchas tareas innecesarias.
 - Ionic está hecho para ser rápido y fácil de entender. Su mantenimiento será más sencillo.
 - Posee de las comunidades de desarrolladores más grandes de la actualidad.
 - Con la ayuda de “Capacitor” ahora se pueden crear aplicaciones nativas para Android y iOS.
 - Alta escalabilidad, así que el número de usuarios activos no afectará su rendimiento.
 - Su facilidad y amplia documentación ayudará a crear una aplicación adecuada a las necesidades del proyecto, ya que la aplicación móvil es únicamente para el registro de los eventos.
- Desventajas
 - Comparado con aplicaciones nativas, su rendimiento puede ser inferior.
 - No poseerá todas las funcionalidades de una app nativa.
 - No es recomendable para proyectos muy grandes.
 - Al probarlo en navegadores puede dar algunos errores.
 - No tiene un estándar de desarrollo.

- Al ser una herramienta joven, puede dar problemas encontrar módulos de otros usuarios que funcionen con ella.

2.2.2.2. Xamarin

- Ventajas
 - Usa una base de código compartida
 - Es confiable, y se combina bien con los sistemas operativos del mercado (Android, iOS).
 - Fácil de instalar y de aprender.
 - Utiliza los lenguajes propios de Microsoft: C# y .NET.
 - Crea una interfaz de usuario nativa con rendimiento nativo.
 - Es posible utilizarlo con herramientas de depuración.
 - Alto porcentaje de reutilización de código.
- Desventajas
 - La aplicación final es muy pesada
 - No es muy bueno para la incorporación de elementos gráficos
 - Sus actualizaciones se retrasan mucho
 - Alto costo para uso profesional y empresarial
 - Tiene problemas de compatibilidad con librerías de terceros
 - Ha estado comprometido con problemas de estabilidad

2.2.2.3. Flutter

- Ventajas

- Tiene un ecosistema propio, y ofrece un rendimiento muy bueno
 - Su ejecución nativa es potente
 - Amplia librería de componentes para la interfaz gráfica
 - El uso de widgets permite un desarrollo rápido
 - El código es apto para cualquier plataforma
 - El renderizado de vistas es muy rápido y constante
 - Posee recarga en caliente, así que los cambios en código se verán inmediatamente.
- Desventajas
 - Es necesario aprender Dart, aunque es fácil de aprender
 - Es un *framework* muy reciente, con una comunidad reducida
 - La integración de *widgets* puede afectar la legibilidad del código
 - Actualmente las librerías a las que tienen acceso los desarrolladores son limitadas.

Se elige utilizar Ionic para la aplicación móvil por su simplicidad y diferentes vías de implementación. La aplicación será ligera, rápida y sencilla de entender, lo que facilitará el mantenimiento. Psee mucha documentación actualizada, una gran comunidad y con la nueva herramienta “Capacitor” se puede crear una aplicación nativa.

Tabla LX. **Ventajas y desventajas de Ionic**

Tecnología	Ventajas		Desventajas
Ionic	<p>Multiplataforma. Aplicación híbrida. Puede implementar diferentes frameworks. Lenguajes Simples. Rápido y fácil de entender. Mantenimiento sencillo. Comunidad grande. Capacitor para aplicación nativa. Alta escalabilidad. Ampla documentación.</p>		<p>Menor rendimiento que una aplicación nativa. Proyectos pequeños. Error en navegadores. Sin estándares. Herramienta joven.</p>
Xamarin	<p>Confiable y multiplataforma Fácil de instalar y de aprender. Lenguajes de Microsoft: C# y .NET. Interfaz nativa. Reutilización de código.</p>		<p>Aplicación muy pesada. Bajo rendimiento para elementos gráficos. Actualizaciones retrasadas. Alto costo. Problemas de compatibilidad con librerías. Poca estabilidad.</p>
Flutter	<p>Buen rendimiento. Ejecución nativa es potente. Ampla librería gráfica. Desarrollo rápido. Multiplataforma. Renderizado rápido y constante. Recarga en caliente.</p>		<p>Es necesario aprender Dart. Muy reciente. La integración de <i>widgets</i> puede afectar la legibilidad del código. Actualmente las librerías a las que tienen acceso los desarrolladores son limitadas.</p>

Fuente: elaboración propia, realizado con Word.

2.3. Ventajas y desventajas de las herramientas para el *backend*

En esta sección se analizan las diferentes ventajas y desventajas del *backend*.

2.3.1. Flask vs otras tecnologías para *backend*

Se analizó Flask y todas sus características respecto a las de otras tecnologías parecidas.

2.3.1.1. Flask

- Ventajas
 - Es un *framework* muy pequeño, no es *full stack*.
 - Consiste en un simple grupo de módulos.
 - Puede combinarse con otras herramientas para potenciar su funcionamiento, como Jinja2 o SQLAlchemy.
 - Incluye servidor web propio para pruebas.
 - Es rápido y brinda un gran desempeño debido a su diseño minimalista.
 - Su documentación es muy buena y extensa, con una comunidad grande y activa.
 - La implementación de Flask es muy sencilla y rápida.
 - Perfecto para aplicaciones y microservicios pequeños.

- Desventajas
 - No posee librerías integradas

- Genera algunas dificultades a la hora de realizar pruebas unitarias
- Su sistema de autenticación de usuarios es muy básico
- No posee un ORM y hay que recurrir a un mapeo de objetos externo, para conectar con bases de datos.

2.3.1.2. Django

- Ventajas
 - Es el mayor *framework* web basado en Python
 - Cuenta con una comunidad grande y activa
 - Posee un panel de administración para base de datos
 - Tiene su propio sistema de autenticación de usuarios
 - Utiliza el patrón MVC
 - Gran rendimiento y flexibilidad
 - Posee una amplia variedad de paquetes y librerías
 - Amplia documentación
- Desventajas
 - A pesar de tener una amplia documentación, puede llegar a ser confusa.
 - Resulta complejo el desarrollo para la realización de un APIRest, o al trabajar con sockets.

2.3.1.3. Pyramid

- Ventajas

- Es un *framework* minimalista, rápido y fiable
 - Es flexible, está diseñado para que, aunque el proyecto crezca, esto no afecte su funcionamiento y rapidez.
 - Posee buena documentación.
 - Es muy confiable, ya que cada lanzamiento de versión ha sido sometido a pruebas unitarias.
- Desventajas
 - Su rendimiento no es tan bueno como el de Flask
 - Pyramid está pensado más para aplicaciones grandes y flexibles
 - No posee una comunidad tan grande como las otras dos opciones

Se decidió utilizar Flask, porque se utilizarán microservicios, y con Flask es simple de implementar un API Rest, dando un buen rendimiento y desempeño. Posee una amplia comunidad de usuarios y no es tan pesado, debido a que solo cuenta con lo necesario para funcionar, y solo se añaden los elementos que se necesiten.

Tabla LXI. **Ventajas y desventajas de Flask**

Tecnología	Ventajas	Desventajas
Flask	Simple. Incluye servidor web. Es rápido y alto desempeño. Buena documentación. Amplia comunidad de usuarios. Sencillo de implementar. Bueno para microservicios. Fácil de realizar API Rest.	Difícil para pruebas unitarias. Autenticación básica.

Continuación de la tabla LXI.

Django	Es <i>full stack</i> . Gran comunidad activa. Administración de base de datos. Autenticación de usuarios. Utiliza el patrón MVC. Alto rendimiento. Flexibilidad. Amplia documentación.	Documentación confusa. Complejo realizar APIRest y sockets.
Pyramid	Minimalista y confiable. Flexible y escalable. Posee buena documentación.	Rendimiento medio. Enfocado en aplicaciones grandes y flexibles. Comunidad pequeña.

Fuente: elaboración propia, realizado con Word.

2.3.2. Modelos NoSql vs SQL

Se hizo un análisis de las características de un modelo de base de datos NoSQL respecto a un modelo SQL.

2.3.2.1. NoSQL

- Ventajas
 - Su escalabilidad y que la información no sea centralizada
 - Permite tener una base de datos mucho más flexible, adaptándose a las necesidades del proyecto de manera más fácil que un modelo entidad relación.
 - Escalabilidad horizontal, por lo que no necesita estar en un solo servidor de tan alto rendimiento. Pueden ejecutarse en máquinas

de bajos recursos, lo que ahorra también grandes cantidades de dinero.

- Permite cambiar esquemas sin detener la base de datos.
 - Permite optimizar consultas para grandes cantidades de datos.
 - Requieren de menor administración y ajustes.
 - Muchas de las soluciones NoSQL ya cuentan con herramientas gráficas y una gran comunidad de desarrolladores.
- Desventajas
 - Aunque las bases de datos NoSQL tienen mucho tiempo de estar en el mercado, muchas de las soluciones que actualmente se utilizan son relativamente nuevas, por lo que pueden no tener ciertas características.
 - Sin atomicidad de la información. Se debe pensar en una estrategia para guardar la integridad de los datos.
 - El lenguaje tiene a variar según la base de datos que se utilice. No hay un estándar definido.

2.3.2.2. SQL

- Ventajas
 - Existen distintas herramientas con mucho soporte y documentación.
 - Hay herramientas gráficas como los DBMS de uso libre que son fáciles de implementar.
 - Guarda la integridad de los datos.
 - Opciones de *software* muy maduros, con una amplia comunidad.

- Posee estándares bien definidos a los que se deben someter todos los procesos.
 - El lenguaje SQL es simple, fácil de entender.
 - Su portabilidad.
 - Posee mucho soporte, una gran comunidad de desarrolladores y herramientas que han sido depuradas durante muchos años
- Desventajas
 - Su escalabilidad es baja en comparación con las NoSQL. Si las bases de datos crecen mucho, crecerá el costo de mantenimiento y almacenamiento de datos.
 - Su instalación y configuración suele ser compleja.
 - No es tan flexible. Un cambio de estructura se puede volver muy complejo.
 - Necesitan muchos recursos para funcionar sin problemas. Eso implica que el costo es más alto.

Se utilizarán ambas soluciones. Por un lado, una herramienta NoSQL para guardar la información general de captura de datos, que es una información que puede tener muchos cambios así que debe ser flexible. Su costo de almacenamiento y mantenimiento serán bajos, debido a las cuotas gratuitas que maneja GCP.

Por otro lado, la solución SQL servirá para solventar el problema de conexión que la aplicación puede tener debido a que será utilizada en lugares sin acceso a internet. Así que se necesita que la aplicación funcione de manera fuera de línea, almacenando los datos que se obtengan a través de ella.

Tabla LXII. **Ventajas y desventajas de NoSQL y SQL**

Tecnología	Ventajas	Desventajas
NoSql	Alta escalabilidad. No centralizada Flexible. Fácil de implementar. Escalabilidad horizontal. Utiliza bajos recursos. Fácil cambio de esquema. Optimiza consultas de grandes cantidades de datos. Menor administración y ajustes.	Las soluciones actuales son muy jóvenes. Poca documentación. Comunidad mediana. No tiene atomicidad. El lenguaje de consultas varía.
SQL	Amplia documentación y muchas herramientas. Su implementación es fácil. Integridad de datos. Opciones de <i>software</i> maduros. Estándares definidos. Lenguaje simple. Portable Mucho soporte.	Baja escalabilidad. Alto costo de mantenimiento y almacenamiento. Su instalación y configuración compleja. No es tan flexible. Difícil cambio de estructura. Consume muchos recursos.

Fuente: elaboración propia, realizado con Word.

2.4. Ventajas y desventajas de los servicios en la nube

En esta sección, se analizan las ventajas y desventajas de los servicios en la nube que se utilizarán para el desarrollo de la solución.

2.4.1. MongoDB Atlas vs Firestore

Se muestran las ventajas y desventajas de utilizar MongoDB Atlas y respecto a Firestore para crear una base de datos NoSQL.

2.4.1.1. MongoDB Atlas

- Ventajas
 - Alta escalabilidad y flexibilidad
 - Existen herramientas gráficas de conexión que facilitan la visibilidad de los datos.
 - Posee una amplia variedad de herramientas para obtener los datos, filtrarlos, agruparlos, entre muchas otras funciones que son llamadas “funciones de agregación”.
 - La herramienta grafica permite realizar búsquedas más elaboradas, realizando uniones de datos de otras colecciones buscando referencias por sus identificadores únicos. Esto facilita las consultas para reportería.
 - Su integración con varios lenguajes, incluyendo python es sencilla y existe mucha información por ser una de las bases de datos NoSQL más utilizadas.
 - Posee una capa gratuita de 512Mb que se puede utilizar para el inicio del proyecto.
 - Acceso a la base de datos desde cualquier lugar del mundo.
 - Manejo de accesos y usuarios de la base de datos.

- Desventajas
 - Poco espacio de almacenamiento en su capa gratuita.
 - Sin atomicidad de la información. Sin embargo, esto se puede solucionar con las funciones de agregación.

2.4.1.2. Firestore

- Ventajas
 - Alta escalabilidad y flexibilidad.
 - Ofrece una capa gratuita con más capacidad de almacenamiento.
 - Su integración es sencilla.
 - Posee estándares bien definidos a los que se deben someter todos los procesos.
 - El lenguaje SQL es simple, fácil de entender.
 - Su portabilidad.
 - Posee mucho soporte, una gran comunidad de desarrolladores y herramientas que han sido depuradas durante muchos años.
 - Se puede administrar el acceso a los datos.
- Desventajas
 - Sin atomicidad de la información. Se debe pensar en una estrategia para guardar la integridad de los datos.
 - La estructura de sus funciones tiene a variar según el lenguaje. Existen funciones que aun están en desarrollo para algunos lenguajes, entre ellos Python, que es el lenguaje que se esta utilizando en el *backend*.

- Por ser una tecnología relativamente nueva, no hay mucha información sobre algunos temas y su comunidad es muy reducida.
- No posee una herramienta que apoye a realizar consultas más elaboradas y tampoco posee una herramienta gráfica para visualizar los datos almacenados que no sea en su plataforma en línea.

Tabla LXIII. **Ventajas y desventajas de NoSQL y SQL**

Tecnología	Ventajas	Desventajas
MongoDB	Alta escalabilidad. No centralizada Flexible. Fácil de implementar. Alojado en la nube. Administración de seguridad Funciones de agregación (amplia variedad) 512Mb de almacenamiento en capa gratis. Herramienta de consultas. Amplia comunidad de usuarios.	Poco espacio en su capa gratuita, pero suficiente para iniciar el proyecto.
Firestore	Alta escalabilidad. No centralizada Flexible. Fácil de implementar. Alojado en la nube. Administración de seguridad Funciones de agregación (en desarrollo) 1Gb de almacenamiento en capa gratis	Resolver el problema de atomicidad requiere de una estructura más compleja. Funciones de agregación en desarrollo. Sin herramienta de consultas. Comunidad reducida de usuarios.

Fuente: elaboración propia, realizado con Word.

2.4.2. Google App Engine vs Google Compute Engine

Se hace un análisis sobre el uso de Google App Engine y Google Compute Engine, tomando en cuenta sus ventajas y desventajas.

2.4.2.1. Google App Engine

- Ventajas
 - Es una Plataforma como servicio (*PaaS*).
 - Resulta una herramienta más económica para aplicaciones que no tienen mucho tráfico o negocios pequeños.
 - Las aplicaciones cuentan con escalado automático, al ejecutarse dentro de la infraestructura de Google.
 - Posee una administración de versiones, de los despliegues realizados.
 - Reduce la complejidad, al no tener una máquina virtual que administrar.
 - Es mucho más seguro en comparación con GCE, ya que posee opciones de seguridad integradas, como firewall y certificados SSL/TLS.
 - Soporta distintos lenguajes de programación en un entorno estándar, como Java, PHP, Node.js, Python, C#, .Net, Ruby y Go. Aunque posee un entorno flexible para ejecutar otros lenguajes no incluidos en la lista estándar.
 - Posee herramientas de diagnóstico y rendimiento.
 - Al diseñar una aplicación dentro de la capa estándar, se obtienen muchos beneficios gratuitos que bajan considerablemente el costo de mantenimiento de la aplicación.

- Desventajas
 - Tiene más restricciones a medida que las instancias son más pequeñas.
 - Puede haber casos donde una aplicación requiera de una máquina virtual más potente.
 - Las redes manejadas en Google Cloud Platform no están integradas en GAE.
 - No se pueden personalizar cuestiones de red, con tanto detalle como si lo permite GCE.

2.4.2.2. Google Compute Engine

- Ventajas
 - Es una Infraestructura como servicio (IaaS).
 - Brinda un control total de una máquina virtual, permitiendo configurar a detalle cada una de sus características, como sistema operativo y su versión, tamaño de memoria, ram, reglas de firewall, conexiones de red y configuración de VLAN, ip publica dedicada, configuración de balanceadores de carga, entre muchas otras cosas.
 - Es compatible con contenedores.
 - Permite la creación de imágenes y *backups* a partir de alguna de las máquinas virtuales que se han creado.

- Desventajas

- Aunque en ciertas ocasiones el control de la infraestructura es una ventaja, en este proyecto pueden quitarse esa preocupación con el uso de GAE. Con GCE se tendría que configurar firewalls, puertos de entrada y salida, permisos, entre otras cuestiones de seguridad.
- Se necesita un mayor conocimiento de infraestructura en la nube.
- El costo de mantenimiento es mayor, aunque dependerá de las características seleccionadas.
- El escalado automático es más lento que el de GAE.

Se escogió desplegar las aplicaciones en GAE ya que por los lenguajes que se escogieron, se puede acceder a cuotas gratuitas que este servicio brinda en cuanto a peticiones, almacenamiento de información, entre otras.

Esta solución ayudará a centrarse en las aplicaciones y preocuparse menos por la seguridad de la infraestructura y escalabilidad. Únicamente se harán las configuraciones necesarias para que las aplicaciones funcionen en este entorno.

Tabla LXIV. **Ventajas y desventajas de Google App Engine y Google Compute Engine**

Tecnología	Ventajas	Desventajas
GAE	Plataforma como servicio (<i>PaaS</i>). Económica para aplicaciones con poco tráfico. Escalado automático. Administración de versiones. Reduce administración. Seguridad integrada. Soporta distintos lenguajes. Herramientas de diagnóstico. Beneficios gratuitos.	Más restricciones. Costos dependiendo del uso de recursos. No se integran redes de GCP.

Continuación de la tabla LXIV.

GCE	<p>Infraestructura como servicio (<i>IaaS</i>). Control total de máquina virtual y todas sus reglas. Compatible con contenedores. Creación de imágenes de respaldo.</p>	<p>Configuraciones de seguridad manuales. Alto conocimiento de infraestructura en la nube. Mayor costo de mantenimiento. Escalado automático más lento.</p>
-----	--	--

Fuente: elaboración propia, realizado con Word.

Figura 19. Diagrama de tecnologías a utilizar



Fuente: elaboración propia, realizado con Draw.io.

3. ARQUITECTURA Y DISEÑO

3.1. Creación del modelo de base de datos

Antes de iniciar el proyecto, se analizaron las diferentes necesidades que se plantearon en el marco teórico para realizar un modelo de datos funcional.

3.1.1. Diseño de modelo NoSql para banco de datos y formularios

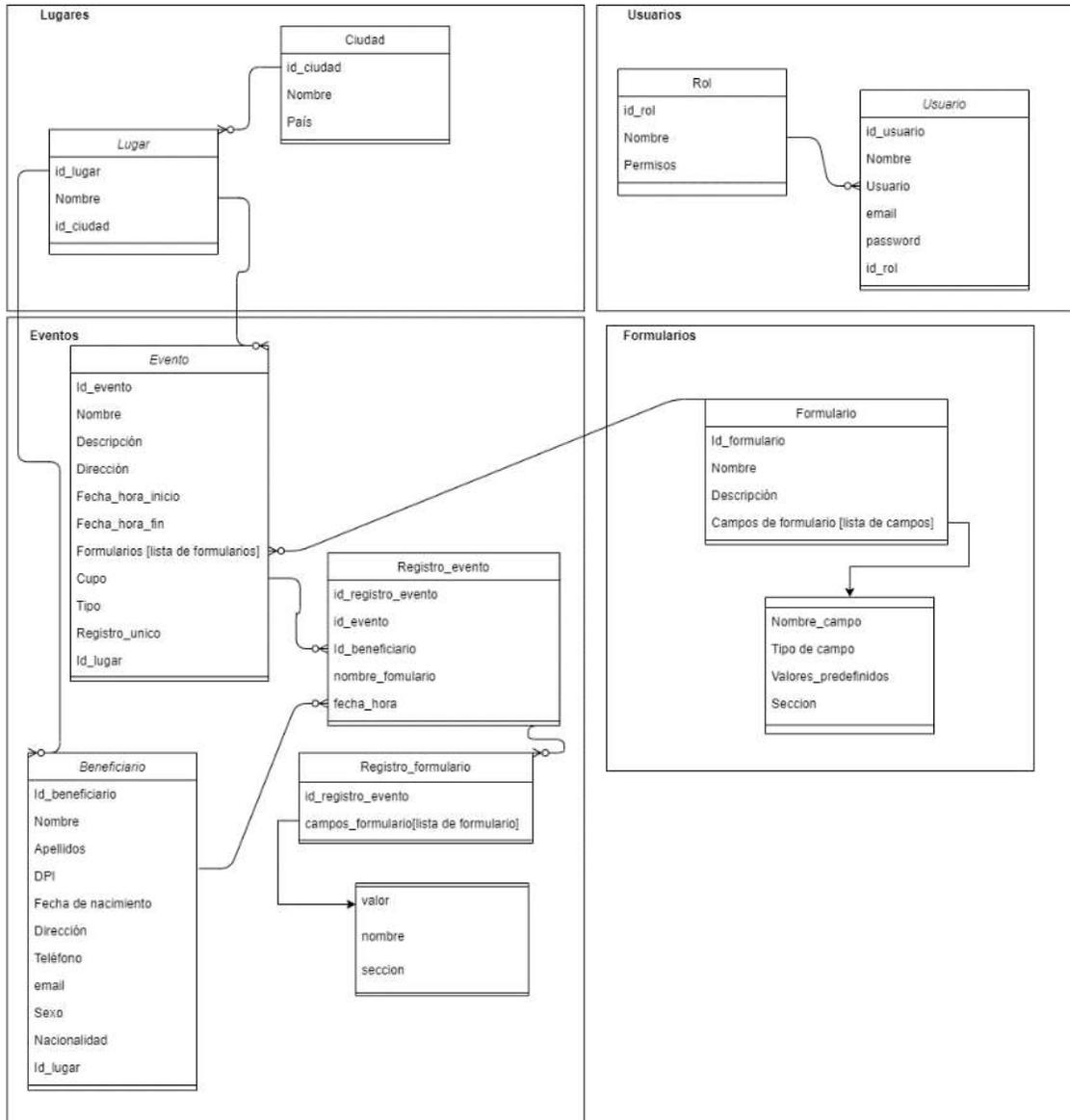
Se necesita hacer un modelo NoSQL que almacene toda la información del sistema, de tal manera que se pueda almacenar todos los datos que demandan las necesidades de la propuesta planteada.

El modelo de datos debería poder almacenar los datos de los usuarios del sistema, los lugares y ciudades donde se harán los eventos, los datos de los eventos, los formularios de los eventos y todos los registros de los beneficiarios en los eventos.

Al ser una base de datos NoSQL no se manejan relaciones entre tablas. La información se almacena en colecciones de datos. Así que, si se van a realizar reportes con esta información, se debe plantear una forma en que se pueda realacionar las colecciones entre sí de manera manual.

Según las necesidades planteadas, se propone el siguiente modelo:

Figura 20. Modelo para base de datos NoSQL

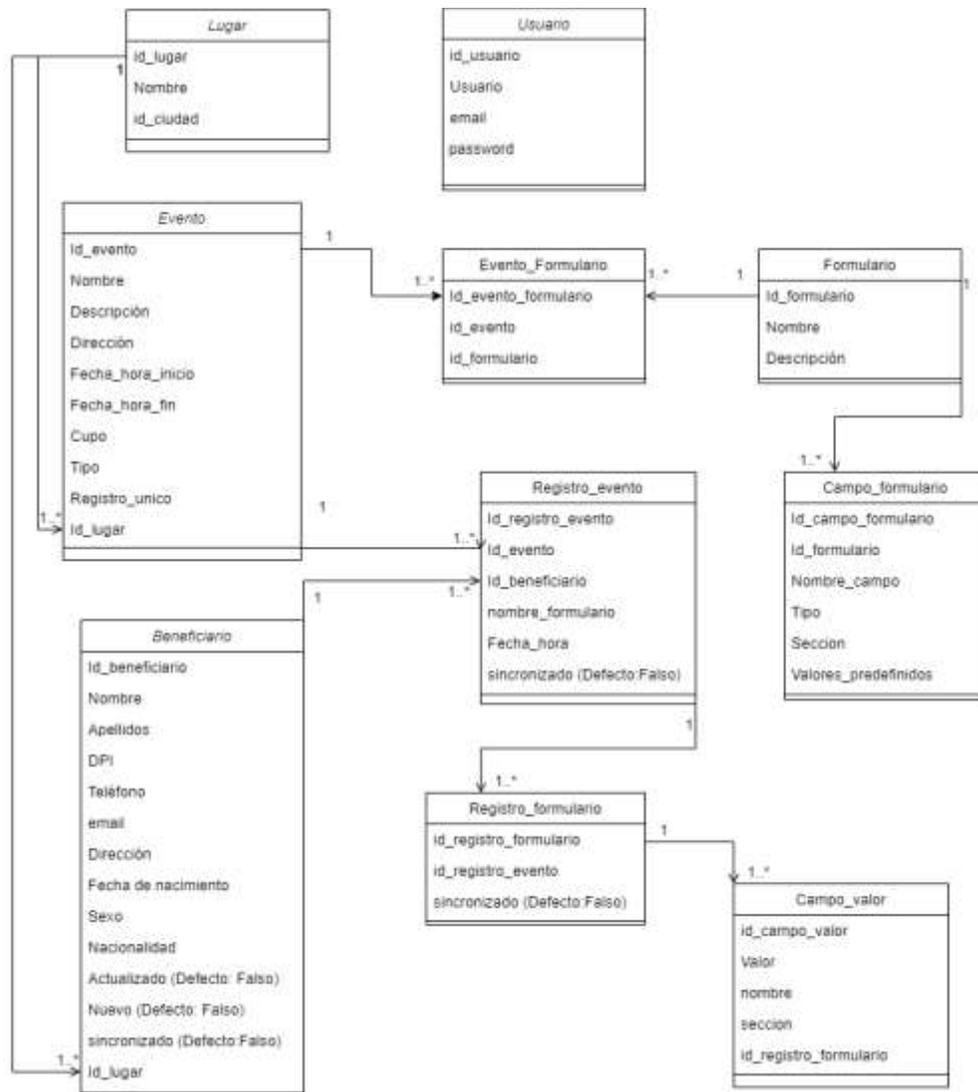


Fuente: elaboración propia, realizado con Draw.io.

3.1.2. Diseño de modelo SQL para captura de datos fuera de línea

Se necesita hacer un modelo que pueda funcionar *offline* para almacenar los datos de registro en eventos. El modelo propuesto es el siguiente:

Figura 21. Modelo para base de datos SQL



Fuente: elaboración propia, realizado con Draw.io.

3.2. Explicación del modelo de base de datos NoSql

Para el modelo NoSql propuesto en el capítulo 3, las colecciones de la base de datos se dividirán en 4 grupos:

- Lugares
 - Ciudad: para almacenar los datos de los donadores
 - Id_ciudad: identificador único
 - Nombre: nombre de ciudad
 - País: país al que pertenece la ciudad
 - Lugar: almacenará los lugares que pertenecen a las ciudades creadas.
 - Id_lugar: identificador único
 - Nombre: nombre del lugar
 - Id_ciudad: identificador de la ciudad a la que pertenece el lugar.
- Eventos
 - Beneficiario: persona que al evento para registrarse y ser parte de este.
 - Id_beneficiario: identificador único
 - Nombre: nombre de beneficiario
 - Apellidos: apellido del beneficiario
 - DPI: número de identificación personal
 - Teléfono: número de teléfono del beneficiario

- Dirección: dirección de domicilio
 - Correo electrónico: correo electrónico del beneficiario
 - Fecha de nacimiento: fecha de nacimiento del beneficiario
 - Sexo: sexo del beneficiario
 - Nacionalidad: nacionalidad del beneficiario
 - Id_lugar: identificador único del lugar a la que pertenece el beneficiario.
- Evento: actividad organizada por la fundación Misión El faro, para beneficio de la comunidad.
- Id_evento: identificador único
 - Nombre: nombre de evento
 - Descripción: descripción del evento
 - Dirección: dirección del evento
 - Fecha_hora_inicio: fecha y hora en que inicia el evento
 - Fecha_hora_fin: fecha y hora en que termina el evento
 - Cupo: cupo del evento
 - Tipo: existen 4 tipos de evento: salud integral, beneficio comunitario, construcción, otros.
 - Registro único: indica si el evento puede tener más de un registro de un mismo beneficiario. Si está activo, la validación se hace por DPI obligatoriamente.

Nota: se considera como evento “Pendiente” todo evento el cual su fecha_hora_inicio es mayor a la fecha actual. Si la fecha actual esta entre la fecha de inicio y fin el evento se encuentra “En curso”. Si la fecha_hora_fin ya ha pasado, el evento se considera “finalizado”.

- Registro_evento: para guardar el detalle general de cada registro realizado en el evento.
 - Id_beneficiario: identificador único de beneficiario
 - Id_evento: identificador único de evento
 - Fecha_registro: fecha en que el beneficiario se registró en evento.
 - Registro_formulario (conjunto de datos): los valores de cada conjunto de registro del formulario representan un campo del formulario. sus datos se componen de:
 - ✓ Id_campo_formulario: id del campo que pertenece a la plantilla de un formulario. Esto permitirá saber de qué tipo es el valor ingresado.
 - ✓ Valor: valor ingresado correspondiente al campo del formulario.
- Formularios
 - Formulario: para almacenar los datos generales del formulario, como su nombre y descripción.
 - Id_formulario: código identificador
 - Nombre: nombre del formulari
 - Descripción: descripción del formulario.
 - Formulario_evento: asignación de un formulario creado previamente a un evento.

- Id_formulario: código identificador de formulario
 - Id_evento: código identificador de evento
 - Campos_formulario (conjunto de datos): cada conjunto almacenará los campos que serán parte de un formulario, indicando su valor, tipo y formulario al que pertenece.
 - ✓ Id_campo_formulario: código identificador.
 - ✓ Nombre_campo: nombre del campo del formulario.
 - ✓ Tipo: tipo del campo del formulario.
 - ✓ Sección: sección al que pertenece el campo del formulario.

- Usuarios
 - Rol: control de roles que pueden asignarse a los usuarios del sistema. Esto permitirá manejar los permisos dentro de los diferentes módulos de la aplicación.
 - Id_rol: código identificador
 - Nombre: nombre del rol
 - Permisos (conjunto de datos): almacena los permisos del rol sobre cada módulo de la aplicación. Cada módulo será asignado con un identificador.

 - Usuarios: para almacenar los datos de los usuarios que el administrador del sistema registre. Estos serán los usuarios que podrán acceder al sistema a través de su correo electrónico.
 - Id_usuario: código identificador

- Nombre: nombre completo del usuario
- Usuario: nombre corto del usuario para la plataforma
- Email: correo electrónico
- Password: contraseña de usuario
- Id_rol: código de Rol asignado al usuario

3.3. Explicación del modelo de base de datos Sql

Para el modelo Sql propuesto en el capítulo 3, las clases para el modelo de datos poseen la misma función que las definidas para el modelo NoSql y son:

- Usuario: almacena los usuarios del sistema, que podrán ingresar a la aplicación.
- Lugar: lugares creados en el modulo de lugraes del sitio administrativo.
- Beneficiario: almacenará una copia de todos los beneficiarios del sistema.
- Formulario: guarda los formularios que se encuentren en el sistema.
- Campo_formulario: almacena los diferentes campos relacionados con los formularios creados. Un formulario puede tener varios registros en la tabla campo_formulario.
- Evento: guarda los eventos pendientes hasta la fecha actual.
- Evento_formulario: aquí se almacenará la relación entre el evento y los formularios, es decir, todas las asignaciones de formularios a evento.
- Registro_evento: almacena los registros de los beneficiarios en los eventos, con la fecha y hora en que hicieron el registro.
- Registro_formulario: detalle del registro del evento, que almacena la relación entre el valor de los campos y su registro.
- Campo_valor: valores que pertenecen a un registro de formulario. Su relación se encuentra en la tabla registro_formulario.

3.4. Propuesta de arquitectura de servicios en la nube

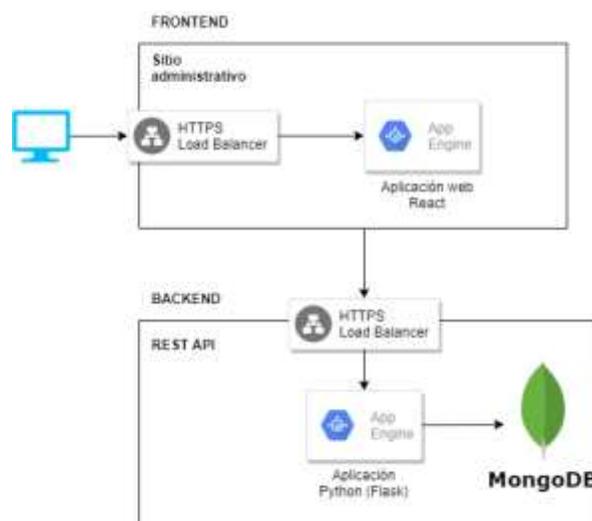
Se plantea la arquitectura de servicios en la nube para las necesidades de la propuesta planteada.

3.4.1. Función de Google App Engine

Existen varios beneficios de utilizar servicios en la nube. Para el desarrollo y puesta en marcha de este proyecto, se aprovechará la capa gratuita de GCP para los diferentes despliegues y servicios que se necesiten.

El servicio que se utilizará para hacer el despliegue del servicio administrativo (*frontend*) y el API REST (*backend*) es GAE. Esta plataforma como servicio permitirá montar las aplicaciones dentro de la infraestructura de Google, utilizar el auto escalado y el balanceador de carga.

Figura 22. Arquitectura GAE *Frontend* y *Backend*



Fuente: elaboración propia, realizado con Draw.io.

3.4.2. Función de MongoDB

Este servicio cuenta con una capa gratuita, la cual se utilizará para la base de datos NoSql. Su función es alojar todos los datos de registro de formularios, eventos, beneficiarios, lugares y usuarios en las colecciones definidas para este modelo.

El acceso a la base de datos MongoDB será desde la aplicación de Python (Flask) utilizando la librería “flask_pymongo”. Las colecciones de datos que MongoDB administrara son:

- Para datos de lugares
 - Ciudad
 - Lugar

- Para datos de eventos
 - Beneficiario
 - Evento
 - Registro_evento

- Para datos de los formularios
 - Formulario
 - Formulario_evento

- Para datos de los usuarios

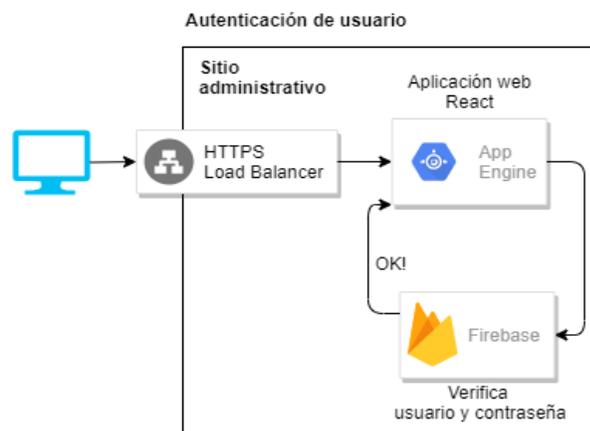
- Rol
- Usuario

3.4.3. Función de Firebase: autenticación con contraseña

Este servicio se utilizará como parte de la seguridad del sitio administrativo. Su función principal será realizar la autenticación por contraseña para que los usuarios puedan contar con un sistema seguro.

Otra de las funciones de Firebase será el manejo del restablecimiento de contraseña a través del correo electrónico del usuario.

Figura 23. Función de firebase para autenticación de usuarios



Fuente: elaboración propia, realizado con Draw.io.

3.5. Repositorio de código

Se utilizará una herramienta para llevar el control de versiones de código del proyecto.

3.5.1. Control de versiones con Git

Git es la herramienta que se utilizará para el control de versiones, y para tener un orden en el flujo de trabajo, se empleará *Gitflow*, cuyo funcionamiento es de la siguiente manera:

- Rama *Master* o maestra: rama principal donde se almacena el historial de la versión principal.
- Rama *Develop* o de desarrollo: rama que se utiliza para integrar las funciones.
- Rama *Feature* o de función: aquí se versionan las partes específicas de un desarrollo.
- Rama *Release* o de publicación: esta rama contiene la versión de un desarrollo terminado. A partir de esta rama se crea un nuevo punto de versión en la rama máster.
- Rama *Hotfix* o de corrección: rama de mantenimiento que se utiliza para realizar correcciones del código de producción, ubicado en la rama máster.

Se crearán 3 repositorios privados en github.com para separar los 3 proyectos a desarrollar. Las rutas a los repositorios son:

- *Backend*: <https://github.com/CRISLEV/200915168tesis-backend.git>
- Aplicación web: <https://github.com/CRISLEV/200915168tesis-frontend.git>
- Aplicación móvil: <https://github.com/CRISLEV/el-faro-app.git>

Para descargar los repositorios de código, se debe agregar el usuario de github en las configuraciones, en la sección de manejo de acceso de cada proyecto. Al tener los permisos, se puede descargar el código con el siguiente comando:

- `git clone ruta_del_repositorio.git`

3.6. Diseño y propuesta para *backend*

En esta sección se plantea la propuesta y diseño de la solución del *backend*.

3.6.1. Lenguaje y framework a utilizar

Para construir el *backend*, se empleará Flask, un *framework* para Python que permitirá generar un API REST de manera sencilla. La función de Flask será habilitar distintos microservicios, con ellos se podrá darle mantenimiento a toda la información que se manejará dentro del sistema administrativo y la aplicación móvil, como:

- Eventos
- Beneficiarios
- Formularios para eventos
- Registro de beneficiarios en eventos
- Lugares
- Usuarios y roles
- Sincronización de datos
- Reportes

3.7. Diseño y propuesta para el *frontend*

En esta sección se plantea la propuesta y diseño de la solución del *frontend*.

3.7.1. *Mockups* de aplicación móvil

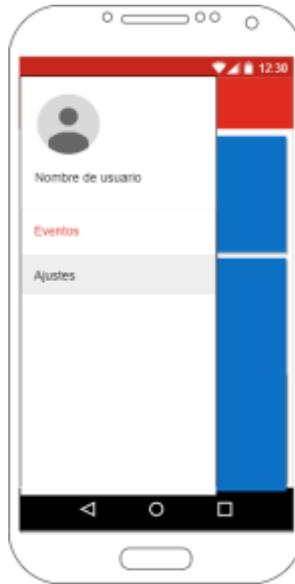
Se muestra la propuesta de las pantallas para la aplicación móvil. Los *mockups* se tomarán como referencia para construir la aplicación, aunque pueda contener más de las que se plantean en este documento.

Figura 24. Inicio de aplicación



Fuente: elaboración propia, realizado con Draw.io.

Figura 25. **Menú principal**



Fuente: elaboración propia, realizado con Draw.io.

Figura 26. **Módulo de Eventos**



Fuente: elaboración propia, realizado con Draw.io.

Figura 27. **Búsqueda de perfil de beneficiario**



Fuente: elaboración propia, realizado con Draw.io.

Figura 28. **Registro de formularios**



Fuente: elaboración propia, realizado con Draw.io.

Figura 29. Creación de perfil de beneficiario



Fuente: elaboración propia, realizado con Draw.io.

Figura 30. Ajustes



Fuente: elaboración propia, realizado con Draw.io.

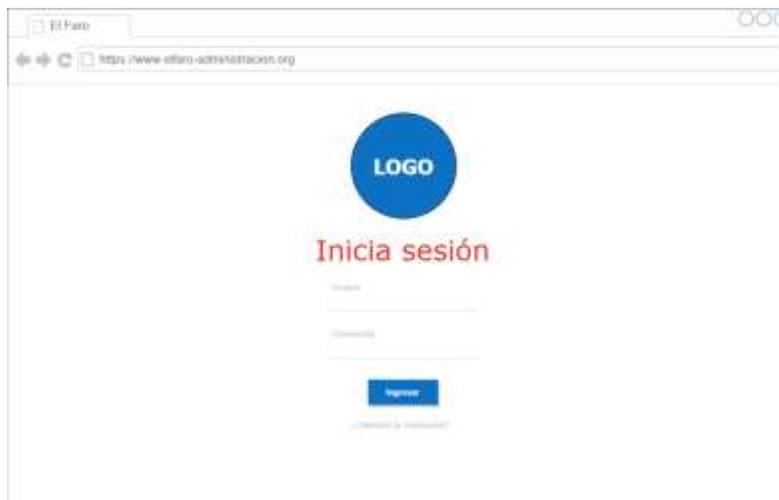
3.7.2. **Mockups de sitio administrativo**

Se muestra la propuesta de las pantallas para el sitio administrativo.

3.7.2.1. **Ingreso al sistema**

Las siguientes pantallas corresponden al inicio de sesión del sitio administrativo.

Figura 31. **Pantalla de inicio de sesión**

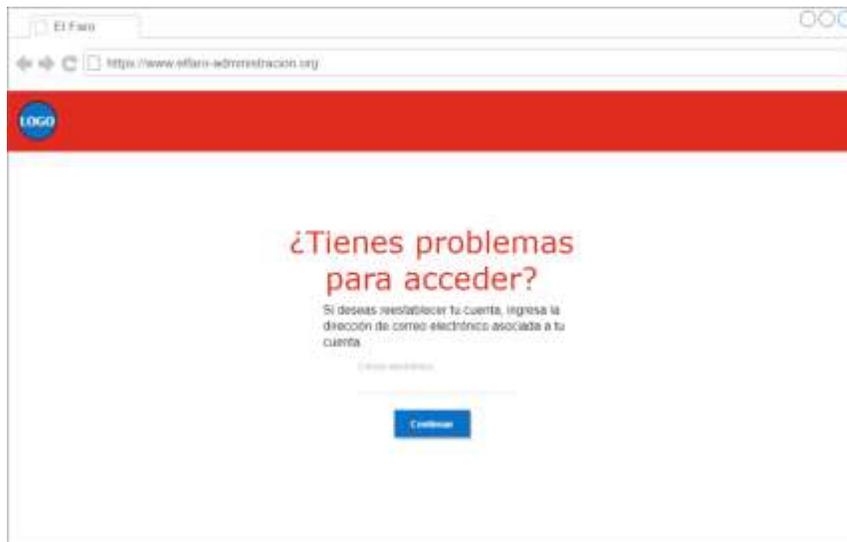


Fuente: elaboración propia, realizado con Draw.io.

3.7.2.2. **Recuperación de contraseña**

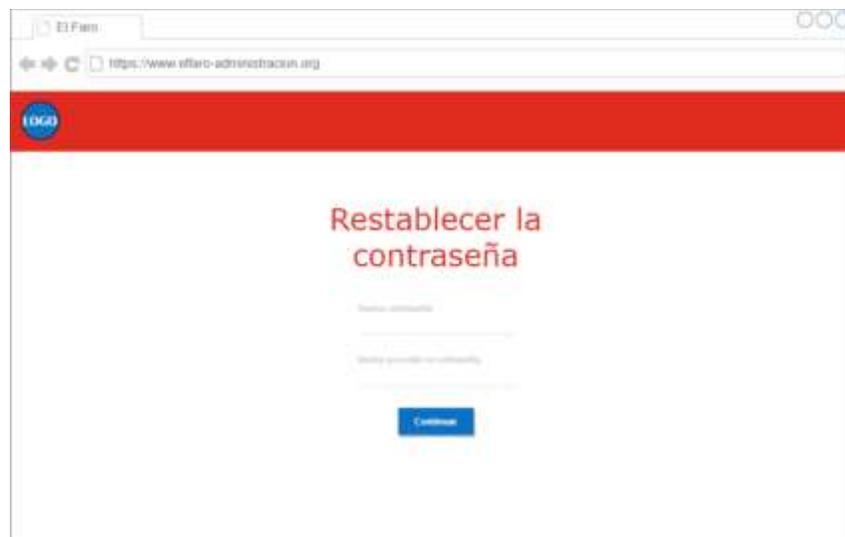
Las siguientes pantallas corresponden a la recuperación de contraseña de usuario.

Figura 32. Solicitud para reestablecer contraseña



Fuente: elaboración propia, realizado con Draw.io.

Figura 33. Reestablecer contraseña



Fuente: elaboración propia, realizado con Word.

3.7.2.3. Menú principal

Las siguientes pantallas corresponden al diseño del menú principal del sitio administrativo.

Figura 34. Menú principal del sitio administrativo



Fuente: elaboración propia, realizado con Draw.io.

Figura 35. Vista y descarga de reportes de actividad

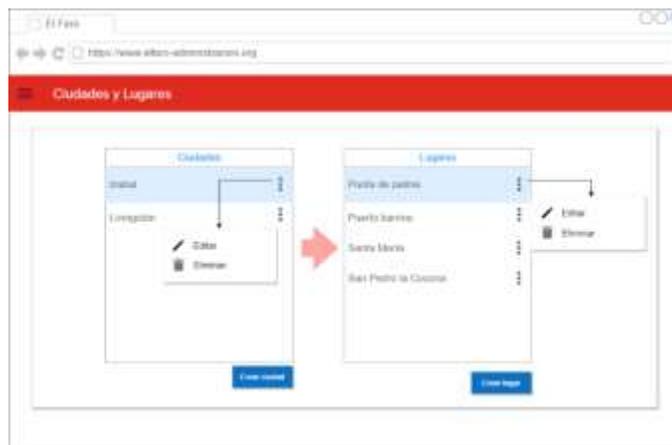


Fuente: elaboración propia, realizado con Draw.io.

3.7.2.4. Módulo de lugares

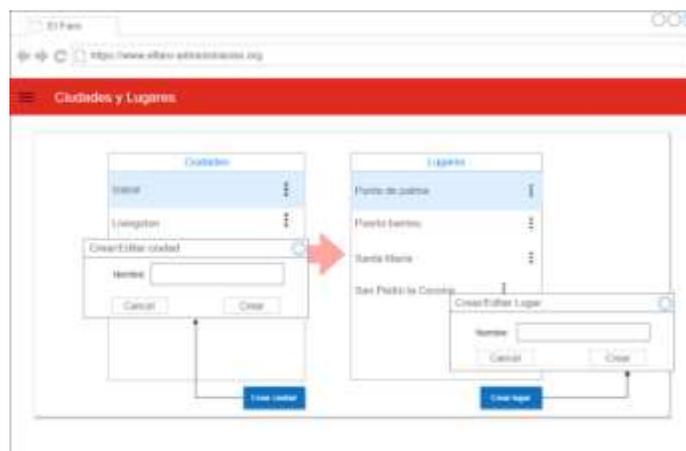
Las siguientes pantallas corresponden al mantenimiento de ciudades y lugares.

Figura 36. Listado de ciudades y lugares



Fuente: elaboración propia, realizado con Draw.io.

Figura 37. Creación/Edición de ciudades y lugares

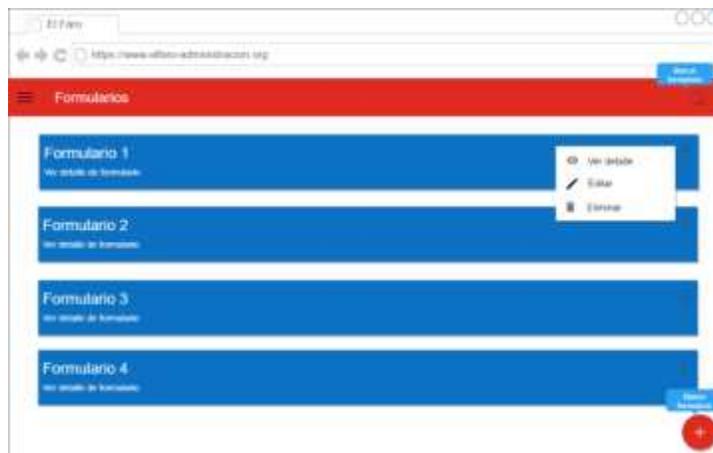


Fuente: elaboración propia, realizado con Draw.io.

3.7.2.5. Módulo de formularios

Las siguientes pantallas corresponden al mantenimiento de formularios de eventos.

Figura 38. Listado de formularios



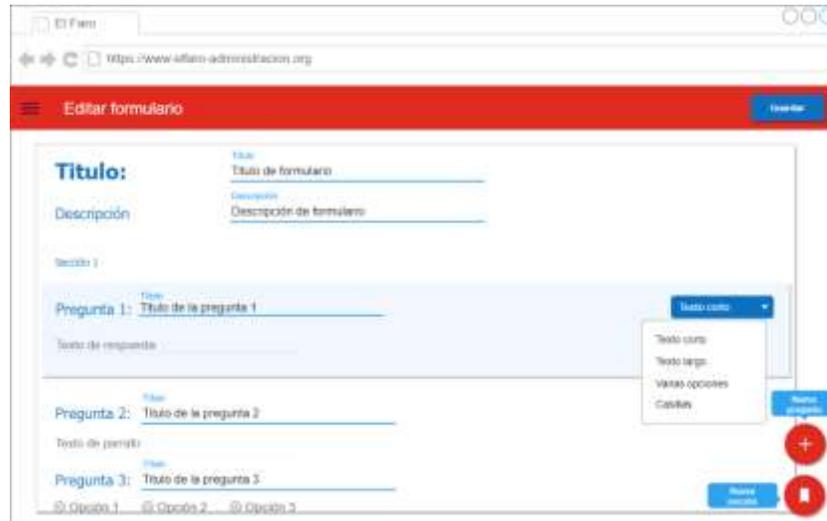
Fuente: elaboración propia, realizado con Draw.io.

Figura 39. Detalle de formulario



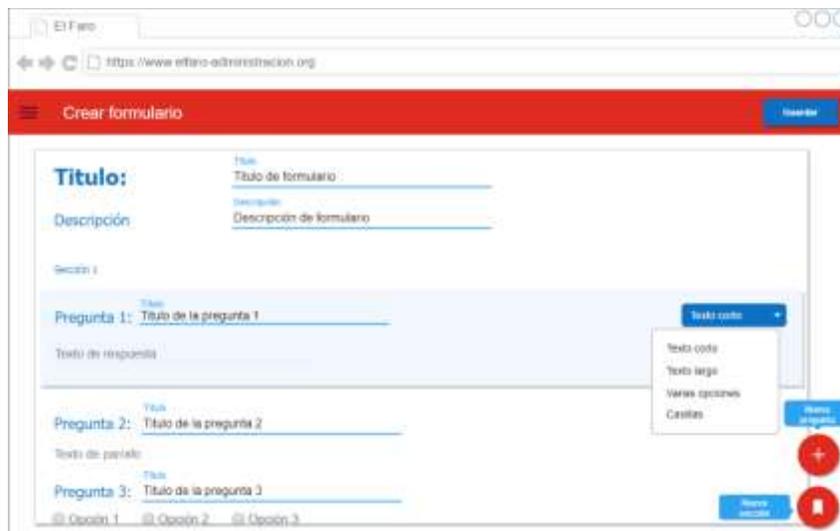
Fuente: elaboración propia, realizado con Draw.io.

Figura 40. Editar formulario



Fuente: elaboración propia, realizado con Draw.io.

Figura 41. Creación de formulario



Fuente: elaboración propia, realizado con Draw.io.

3.7.2.6. Módulo de eventos

Las siguientes pantallas corresponden al mantenimiento de eventos del sistema.

Figura 42. Listado de eventos del sistema



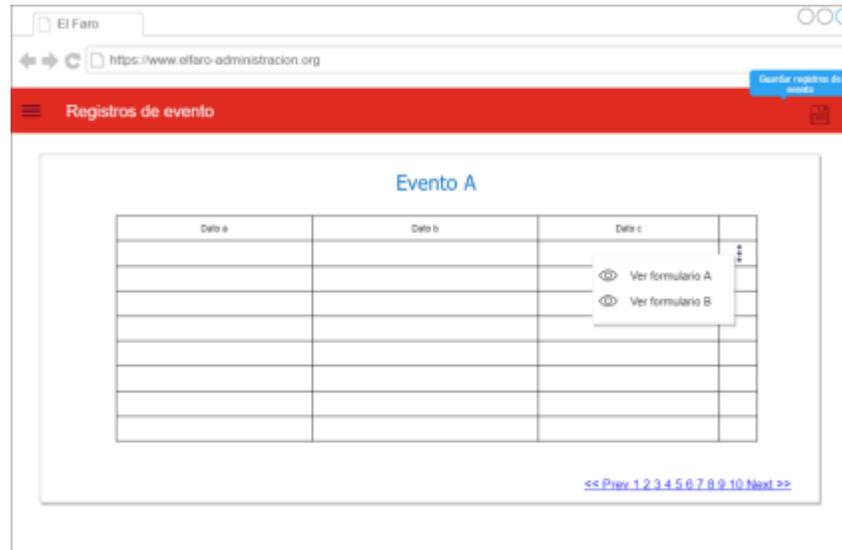
Fuente: elaboración propia, realizado con Draw.io.

Figura 43. Detalle de evento



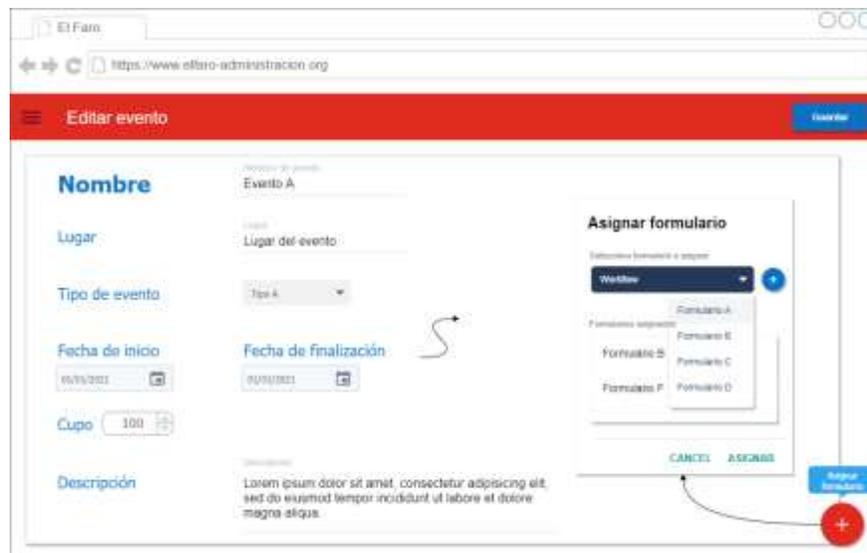
Fuente: elaboración propia, realizado con Draw.io.

Figura 44. Ver registros de evento



Fuente: elaboración propia, realizado con Draw.io.

Figura 45. Editar evento



Fuente: elaboración propia, realizado con Draw.io.

Figura 46. Crear evento



Fuente: elaboración propia, realizado con Draw.io.

3.7.2.7. Módulo de beneficiarios

Las siguientes pantallas corresponden al mantenimiento de beneficiarios del sistema.

Figura 47. Lista de beneficiarios



Fuente: elaboración propia, realizado con Draw.io.

3.7.2.8. Módulo de usuarios y roles

Las siguientes pantallas corresponden al mantenimiento de usuarios y roles.

Figura 50. Lista de usuarios



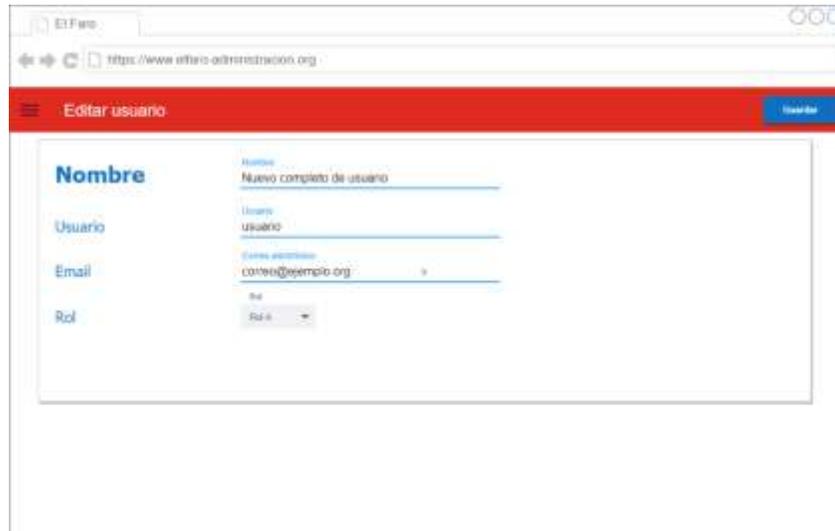
Fuente: elaboración propia, realizado con Draw.io.

Figura 51. Crear usuario



Fuente: elaboración propia, realizado con Draw.io.

Figura 52. Editar usuario



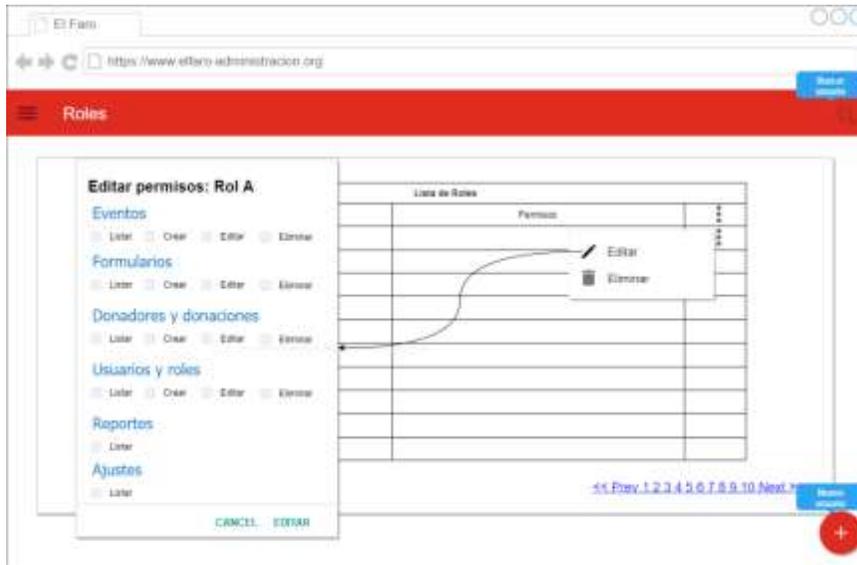
Fuente: elaboración propia, realizado con Draw.io.

Figura 53. Lista de roles



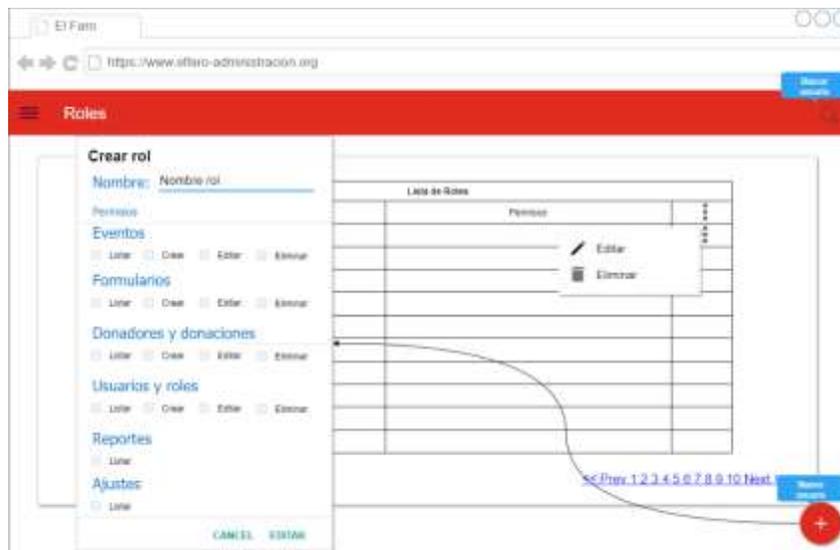
Fuente: elaboración propia, realizado con Draw.io.

Figura 54. Editar permisos de rol



Fuente: elaboración propia, realizado con Draw.io.

Figura 55. Crear rol con permisos



Fuente: elaboración propia, realizado con Draw.io.

3.7.3. Funcionalidades de aplicación móvil

Las funcionalidades que debe cubrir la aplicación móvil son:

- Módulo de eventos
 - Listar de eventos
 - Seleccionar evento
 - Crear nuevo registro en evento

- Módulo de beneficiarios
 - Crear perfil de beneficiario
 - Listar beneficiarios
 - Modificar beneficiario
 - Eliminar beneficiario

- Módulo de ajustes
 - Sincronizar datos registrados
 - Sincronizar datos de aplicación

3.7.4. Funcionalidades de plataforma web administrativa

Las funcionalidades que debe cubrir el sitio administrativo son:

- Ingreso al sistema y recuperación de contraseña
- Resumen de actividad

- Mostrar el resumen de actividad que consiste en los reportes de resumen.
- Módulo de beneficiarios
 - Crear, Listar, editar y eliminar beneficiarios
- Módulo de eventos
 - Crear, Listar, editar y eliminar eventos
 - Ver datos de registros de eventos
 - Descargar datos de registros de eventos
- Módulo de formularios
 - Crear, Listar, editar y eliminar formularios
- Módulo de lugares
 - Crear, Listar, editar y eliminar ciudades
 - Crear, Listar, editar y eliminar lugares
- Módulo de reportes
 - Listar, ver y descargar los reportes del resumen de actividad
- Módulo de usuarios y roles
 - Crear, Listar, editar y eliminar roles

- Crear, Listar, editar y eliminar usuarios

3.8. Buenas prácticas y seguridad de transacciones y datos

Dentro de las consideraciones del proyecto, esta aplicar buenas prácticas de desarrollo y seguridad de la información.

3.8.1. Utilizar autorización en los servicios del API REST

Habilitar un API REST es una tarea sencilla. Sin embargo, es importante que este tipo de desarrollos estén acompañados de algún tipo de seguridad.

En el caso del *backend* de este proyecto, será de mucha importancia utilizar un *token* de autorización para las peticiones que se realicen. Las opciones que se propone implementar son:

- Bearer token
- OAuth 2.0

4. ESPECIFICACIONES DE SERVICIOS DE LA NUBE

4.1. Procedimiento para crear la cuenta de servicios de GCP

Para obtener una cuenta de GCP se recomienda seguir los siguientes pasos:

- Crear una cuenta de Google (si no cuenta con una)
- Dirigirse a la dirección de Google Cloud Platform: <https://console.cloud.google.com/>.
- En el mensaje de bienvenida estarán las opciones para escoger el país y aceptar las condiciones del servicio de GCP.

Figura 56. Creación de cuenta en GCP



The screenshot shows the Google Cloud Platform welcome page. At the top, it says "Google Cloud Platform". Below that, it says "Te damos la bienvenida, [redacted]" and "Crea y administra tus instancias, discos, redes y otros recursos de Google Cloud Platform desde un solo lugar." There is a "País" dropdown menu with "Guatemala" selected. Under "Condiciones del Servicio", there is a checked checkbox and the text "Acepto las Condiciones del Servicio de Google Cloud Platform y de las API y los servicios aplicables." Under "Actualizaciones por correo electrónico", there is an unchecked checkbox and the text "Quiero recibir correos electrónicos periódicos sobre novedades, actualizaciones de productos y ofertas especiales de Google Cloud y Google Cloud Partners." At the bottom right, there is a button labeled "ACEPTAR Y CONTINUAR".

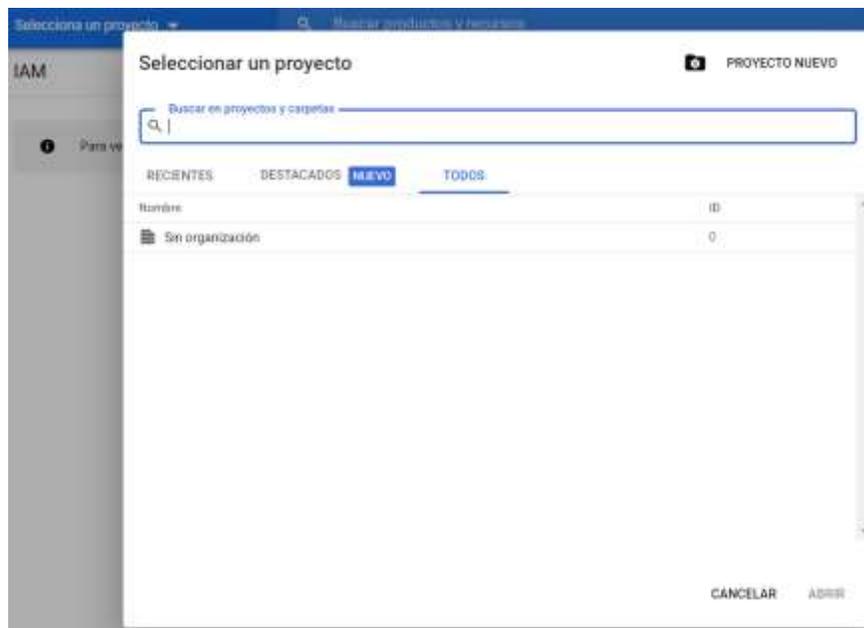
Fuente: elaboración propia, realizado con captura de pantalla.

4.2. Procedimiento para crear un proyecto y definir usuarios y accesos dentro del proyecto

Se listan los pasos para crear un proyecto en GCP:

- Abrir ventana de proyectos: en la parte superior, hacer clic en la opción que dice “Selecciona un proyecto”. Esto abrirá una ventana donde se podrá ver la lista de proyectos que se tienen y la opción para crear.

Figura 57. Selección de proyecto GCP



Fuente: elaboración propia, realizado con captura de pantalla.

- Al hacer clic en el botón PROYECTO NUEVO, se abre una nueva ventana donde se describirá el nombre del proyecto. También se puede escoger la ubicación de este.

Figura 58. Creación de nuevo proyecto GCP



Fuente: elaboración propia, realizado con captura de pantalla.

- Dirigirse al menú “IAM y Administración” para agregar nuevos miembros al proyecto

Figura 59. Menu IAM y administración GCP



Fuente: elaboración propia, realizado con captura de pantalla.

- Al hacer *clic* en botón agregar, se podrá visualizar la pantalla para agregar un nuevo miembro. Se utiliza una cuenta de Google para agregar un nuevo miembro al proyecto.

Figura 60. **Agregar miembro a proyecto GCP**

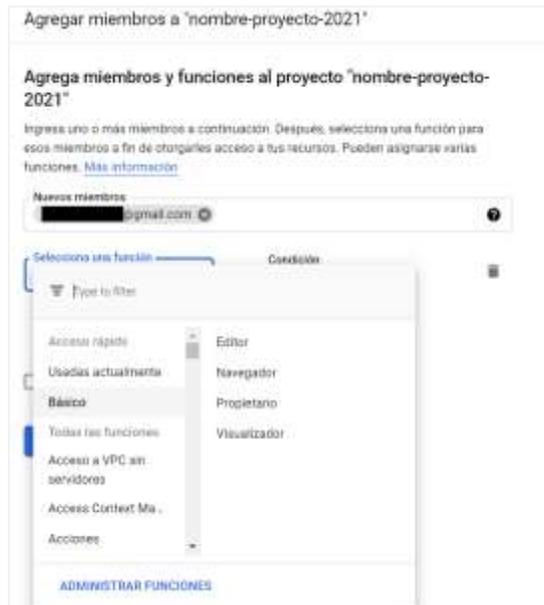


The screenshot shows the 'Agregar miembros a "nombre-proyecto-2021"' interface. It includes a title, a sub-header 'Agrega miembros y funciones al proyecto "nombre-proyecto-2021"', and a paragraph of instructions. Below this is a search bar for 'Nuevos miembros' containing an email address. There is a dropdown menu for 'Selecciona una función', a 'Condición' section with an 'Agregar condición' link, and a '+ AGREGAR OTRA FUNCIÓN' button. At the bottom, there is a checkbox for 'Enviar un correo electrónico de notificación' and two buttons: 'GUARDAR' and 'CANCELAR'.

Fuente: elaboración propia, realizado con captura de pantalla.

- Para los permisos, es posible seleccionar exactamente los servicios a los que el nuevo miembro tendrá acceso. Esto se puede explorar en el menú desplegable.

Figura 61. **Agregar permisos a miembro del proyecto GCP**



Fuente: elaboración propia, realizado con captura de pantalla.

- Luego de seleccionar los permisos seleccionar el botón guardar para crear el nuevo usuario con los permisos asignados.

Figura 62. **Guardar nuevo miembro de proyecto GCP**



Fuente: elaboración propia, realizado con captura de pantalla.

4.3. Configuraciones iniciales de servicios en la nube

Se detallan algunas de las configuraciones iniciales que se deben realizar para utilizar los servicios de la nube.

4.3.1. Creación de proyecto Firebase

Al iniciar sesión en firebase, se debe iniciar con la cuenta de Google que se creó el proyecto. Esto permitirá enlazar el proyecto en el proceso de creación. Cabe mencionar que el proyecto debe tener deshabilitada la facturación para poder utilizar el plan Spark de firebase. El proceso de creación de un proyecto en firebase es el siguiente:

- En el panel de inicio seleccionar la opción crear nuevo proyecto, donde se mostrará un panel donde se podrá seleccionar el proyecto de GCP previamente creado.

Figura 63. Creación de proyecto en firebase



Fuente: elaboración propia, realizado con captura de pantalla.

- Se mostrará un recordatorio de algunos detalles importantes. Presionar el botón continuar.

Figura 64. Detalles de creación de proyecto en firebase



Fuente: elaboración propia, realizado con captura de pantalla.

4.3.2. Habilitar método de autenticación en Firebase

Es necesario habilitar el método de autenticación en el proyecto de firebase para obtener las credenciales necesarias para usar este servicio.

Para habilitar este método, se realizarán los siguientes pasos:

- Seleccionar el método de autenticación por correo electrónico y contraseña.
- No habilitar la opción de acceso sin contraseña.
- Guardar cambios.

Figura 65. **Habilitar autenticación por correo electrónico y contraseña**



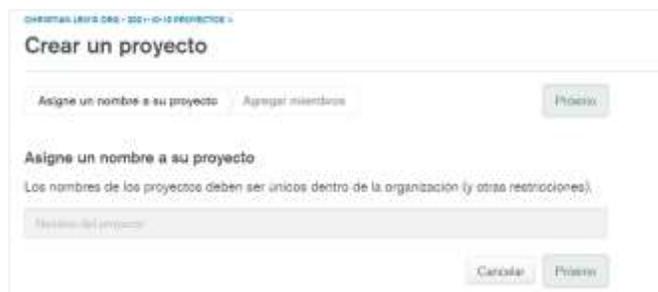
Fuente: elaboración propia, realizado con captura de pantalla.

4.3.3. **Creación de proyecto en MongoDB Atlas**

Para la base de datos NoSql se utilizará la herramienta MongoDB Atlas. Para su creación y configuración se seguirán los siguientes pasos:

- Crear una cuenta o iniciar sesión con un correo de Google en <https://cloud.mongodb.com>.
- Crear un nuevo proyecto donde se tendrá la base de datos.

Figura 66. **Creación de proyecto de MongoDB**



Fuente: elaboración propia, realizado con captura de pantalla.

- Crear el usuario administrador de la base de datos para acceder a la base de datos con contraseña.

Figura 67. Creación de usuario para MongoDB

Agregar nuevo usuario de base de datos

Cree un usuario de base de datos para conceder a una aplicación o usuario, acceso a bases de datos y colecciones en sus clústeres en este proyecto Atlas. El control de acceso granular se puede configurar con privilegios predeterminados o roles personalizados. Puede conceder acceso a un proyecto u organización de Atlas utilizando el [Administrador de acceso](#).

Método de autenticación

Contraseña Certificado AWS IAM (MongoDB 4.4 y posterior)

MongoDB utiliza SCRAM como su método de autenticación predeterminado.

Autenticación de contraseña

Introduzca un nombre de usuario

Introduzca la contraseña MOSTRAR

Fuente: elaboración propia, realizado con captura de pantalla.

- Añadir la IP pública a la lista de acceso, para poder realizar pruebas locales. Cuando se haga el despliegue se deberá agregar la ip del proyecto del *backend*.
- Luego ir a la opción Crear, para hacer la base de datos. Seleccionar la opción Sin servidor, y en la sección Proveedores y regiones de la nube, seleccionar *Google cloud*, y la región *us-central1*.

Figura 68. Creación de base de datos en MongoDB



Fuente: elaboración propia, realizado con captura de pantalla.

- Por último, nombrar la instancia de base de datos MongoDB libre de servidor.

5. ESPECIFICACIONES DEL BACKEND

5.1. Configuraciones iniciales del *backend*

Para comenzar a desarrollar el *backend*, primero se deben realizar las siguientes instalaciones:

- Instalar Python versión 3
- Instalar Pip
- Instalar Cloud SDK
- Instalar dependencias de flask
- Instalar dependencias de mongodb

5.1.1. Instalación y configuraciones de Flask

Para la instalación de Flask, que es el *framework* que se utilizará para el API Rest, bastará también con ejecutar los comandos de instalación y agregar la dependencia al archivo de requerimientos. Los pasos para la instalación de Flask son los siguientes:

- Ejecutar el comando de instalación
 - `python -m pip install Flask`

Figura 69. Instalación de librería Flask en Python

```
PS C:\Users\usuario> pip install flask
WARNING: Value for urllib3 does not match, please report this to: https://github.com/pypa/pip/issues/9617
Defaulted to using installation location C:\Users\usuario\AppData\Local\Programs\Python\Python39\Scripts
sysconfig: C:\Users\usuario\AppData\Local\Programs\Python\Python39\Scripts
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
Collecting flask
  Using cached flask-1.1.2-py2.py3-none-any.whl (94 kB)
Requirement already satisfied: click>=7.1.1 in c:\users\usuario\appdata\local\program\python\python39\lib\site-packages (from flask) (7.1.2)
Requirement already satisfied: werkzeug>=0.15 in c:\users\usuario\appdata\local\program\python\python39\lib\site-packages (from flask) (1.0.1)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\usuario\appdata\local\program\python\python39\lib\site-packages (from flask) (2.11.0)
Requirement already satisfied: MarkupSafe>=0.2.1 in c:\users\usuario\appdata\local\program\python\python39\lib\site-packages (from flask) (2.1.1)
Installing collected packages: flask
  WARNING: The script flask.exe is installed in 'C:\Users\usuario\AppData\Local\Programs\Python\Python39\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: Value for urllib3 does not match, please report this to: https://github.com/pypa/pip/issues/9617
Defaulted to using installation location C:\Users\usuario\AppData\Local\Programs\Python\Python39\Scripts
sysconfig: C:\Users\usuario\AppData\Local\Programs\Python\Python39\Scripts
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
Successfully installed flask-1.1.2
WARNING: You are using pip version 21.3, however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\usuario\AppData\Local\Programs\Python\Python39\python.exe -m pip install --upgrade pip' command.
PS C:\Users\usuario>
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Añadir al documento requirements.txt la dependencia de Flask con su versión:
 - Flask==1.1.2
- Importar a la aplicación la librería:
 - from flask import Flask

5.1.2. Instalación y configuraciones de Flask-Cors

Flask-Cors es necesario dentro del proyecto de Python para controlar el uso compartido de recursos entre orígenes (CORS). Esto permitirá recibir las peticiones desde el *frontend* y evitar problemas de este tipo.

- Ejecutar el comando de instalación

- o python -m pip install -U flask-cors

Figura 70. Instalación de librería Flask-Cors en python

```

PS C:\Users\Usuario> python -m pip install -U flask-cors
WARNING: Value for urllib3.headers does not match. Please report this to https://github.com/pypa/pip/issues/8629
distutils: c:\Users\Usuario\AppData\Local\Programs\Python\Python311\Scripts
sysconfig: c:\Users\Usuario\AppData\Local\Programs\Python\Python311\Scripts
WARNING: Running pip as the 'root' user can result in broken permissions, conflicting
dependencies, and is not recommended.
user = false
home = None
root = None
prefix = None
Collecting flask-cors
  using cached flask-cors-3.0.9-py2.py3-none-any.whl (18 kB)
Requirement already satisfied: flask<=2.1 in c:\Users\Usuario\AppData\Local\Programs\Python\Python311\site-packages (from flask-cors) (1.1.2)
Requirement already satisfied: flask in c:\Users\Usuario\AppData\Local\Programs\Python\Python311\site-packages (from flask-cors) (1.1.2)
Requirement already satisfied: Werkzeug>=0.15 in c:\Users\Usuario\AppData\Local\Programs\Python\Python311\site-packages (from flask-cors) (1.0.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\Users\Usuario\AppData\Local\Programs\Python\Python311\site-packages (from flask-cors) (2.1.1)
Requirement already satisfied: Werkzeug<=2.1 in c:\Users\Usuario\AppData\Local\Programs\Python\Python311\site-packages (from flask-cors) (2.1.1)
Installing collected packages: flask-cors
WARNING: Value for urllib3.headers does not match. Please report this to https://github.com/pypa/pip/issues/8629
distutils: c:\Users\Usuario\AppData\Local\Programs\Python\Python311\Scripts
sysconfig: c:\Users\Usuario\AppData\Local\Programs\Python\Python311\Scripts
WARNING: Running pip as the 'root' user can result in broken permissions, conflicting
dependencies, and is not recommended.
user = false
home = None
root = None
prefix = None
Successfully installed flask-cors-3.0.9
WARNING: You are using pip version 21.1; however, version 21.2.1 is available.
You should consider upgrading via the 'C:\Users\Usuario\AppData\Local\Programs\Python\Python311\Scripts.exe -m pip install --upgrade pip' command.
PS C:\Users\Usuario>

```

Fuente: elaboración propia, realizado con captura de pantalla.

- Añadir al documento requirements.txt la dependencia de flask-cors con su versión:
 - o Flask-Cors==3.0.9
- Importar a la aplicación la librería.
 - o from flask_cors import CORS, cross_origin

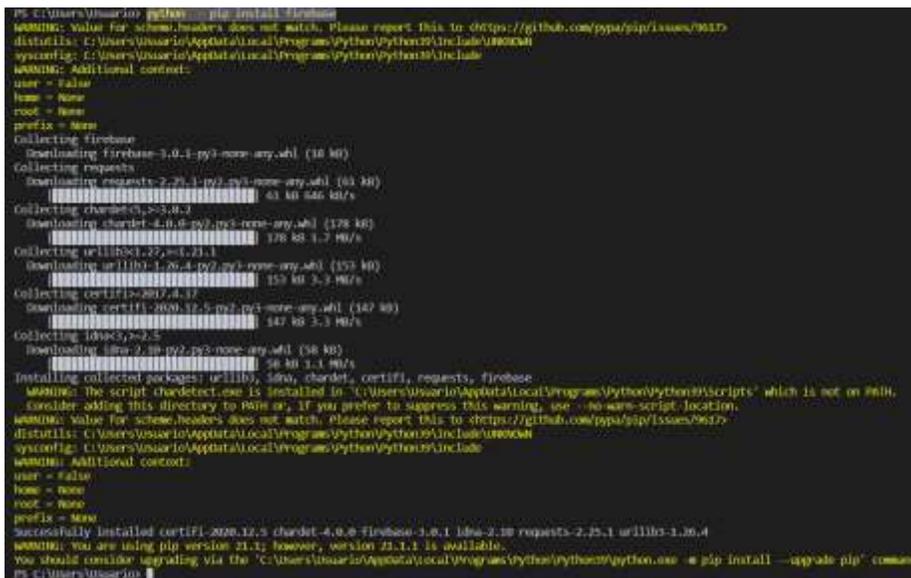
5.1.3. Instalación y configuraciones de Firebase

Para el manejo de autenticación de usuarios con correo electrónico y contraseña se utilizará Firebase. Para que esto quede funcionando del lado del

backend se empleará la librería de Python que maneja el API Rest de Firebase. Basta con instalarla de la manera siguiente:

- Ejecutar el comando de instalación.
 - `python -m pip install firebase`

Figura 71. Instalación de librería Firebase en Python



```
PS C:\Users\Usuario> python -m pip install firebase
WARNING: Value for scheme.headers does not match. Please report this to https://github.com/pypa/pip/issues/9617
distutils: C:\Users\Usuario\AppData\Local\Programs\Python\Python39\include\win32
sysconfig: C:\Users\Usuario\AppData\Local\Programs\Python\Python39\include
WARNING: Additional context:
user = false
home = None
root = None
prefix = None
Collecting firebase
  Downloading firebase-3.0.1-py3-none-any.whl (10 kB)
Collecting requests
  Downloading requests-2.25.1-py2.py3-none-any.whl (63 kB)
  |#####| 63 kB 546 kB/s
Collecting chardet<=3.0.2
  Downloading chardet-3.0.2-py3-none-any.whl (178 kB)
  |#####| 178 kB 1.7 MB/s
Collecting urllib3<1.27,=>1.21.1
  Downloading urllib3-1.26.4-py2.py3-none-any.whl (153 kB)
  |#####| 153 kB 3.1 MB/s
Collecting certifi<2020.12.5,=>2020.11.5
  Downloading certifi-2020.11.5-py2.py3-none-any.whl (147 kB)
  |#####| 147 kB 3.3 MB/s
Collecting idna<=2.10,=>2.10
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
  |#####| 58 kB 1.1 MB/s
Installing collected packages: urllib3, idna, chardet, certifi, requests, firebase
  WARNING: The script chardetect.exe is installed in 'C:\Users\Usuario\AppData\Local\Programs\Python\Python39\scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: Value for scheme.headers does not match. Please report this to https://github.com/pypa/pip/issues/9617
  distutils: C:\Users\Usuario\AppData\Local\Programs\Python\Python39\include\win32
  sysconfig: C:\Users\Usuario\AppData\Local\Programs\Python\Python39\include
  WARNING: Additional context:
  user = false
  home = None
  root = None
  prefix = None
Successfully installed certifi-2020.11.5 chardet-3.0.2 firebase-3.0.1 idna-2.10 requests-2.25.1 urllib3-1.26.4
WARNING: You are using pip version 21.1; however, version 21.1.1 is available.
You should consider upgrading via the 'C:\Users\Usuario\AppData\Local\Programs\Python\Python39\python.exe -m pip install --upgrade pip' command.
PS C:\Users\Usuario>
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Añadir al documento requirements.txt la dependencia de Firebase con su versión:
 - `firebase==3.0.1`
- Luego, se importará la librería al proyecto de la siguiente manera:

- from firebase import Firebase

5.1.4. Manejo de base de datos con MongoDB

Para manejar todos los datos del sistema se utilizará MongoDB. La comunicación con este servicio se debe hacer desde el *backend*, así que se instalará la librería flask_pymongo, y pymongo[srv], de la siguiente manera:

- Ejecutar el comando de instalación
 - python -m pip install Flask-PyMongo
 - python -m pip install pymongo[srv]

Figura 72. Instalación librería flask_pymongo

```
PS C:\Users\Usuario\Documents\Proyectos\SemInv\Proyecto_Final\200915168tesis-backend> python -m pip install flask-pymongo
Collecting Flask-PyMongo
  Using cached flask-pymongo-2.1.0-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: Flask<=0.11 in c:\users\usuario\appdata\local\programs\python\python39\lib\site-packages (fr
2)
Requirement already satisfied: PyMongo>=3.3 in c:\users\usuario\appdata\local\programs\python\python39\lib\site-packages (F
2.0)
Requirement already satisfied: werkzeug>=0.15 in c:\users\usuario\appdata\local\programs\python\python39\lib\site-packages
k-pymongo) (1.0.1)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\usuario\appdata\local\programs\python\python39\lib\site-packa
Flask-PyMongo) (1.1.0)
Requirement already satisfied: click>=5.1 in c:\users\usuario\appdata\local\programs\python\python39\lib\site-packages (fro
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\usuario\appdata\local\programs\python\python39\lib\site-packages
k-pymongo) (2.11.3)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\usuario\appdata\local\programs\python\python39\lib\site-package
>Flask<=0.11;Flask-PyMongo) (1.1.1)
Installing collected packages: Flask-PyMongo
Successfully installed Flask-PyMongo-2.3.0
WARNING: you are using pip version 21.2.4; however, version 21.3 is available.
You should consider upgrading via the 'C:\Users\Usuario\AppData\Local\Programs\Python\Python39\python.exe -m pip install --
PS C:\Users\Usuario\Documents\Proyectos\SemInv\Proyecto_Final\200915168tesis-backend> python -m pip install pymongo[srv]
Requirement already satisfied: pymongo[srv] in c:\users\usuario\appdata\local\programs\python\python39\lib\site-packages (3
Collecting dnspython<2.0.0, >=1.16.0
  Downloading dnspython-1.16.0-py2.py3-none-any.whl (188 kB)
    |#####| 188 kB 123 kB/s
Installing collected packages: dnspython
Successfully installed dnspython-1.16.0
WARNING: you are using pip version 21.2.4; however, version 21.3 is available.
You should consider upgrading via the 'C:\Users\Usuario\AppData\Local\Programs\Python\Python39\python.exe -m pip install --
PS C:\Users\Usuario\Documents\Proyectos\SemInv\Proyecto_Final\200915168tesis-backend>
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Añadir al documento requirements.txt la dependencia de Firebase con su versión:

- Flask-PyMongo==2.3.0
- Luego, se importará la librería al proyecto de la siguiente manera:
 - `from flask_pymongo import PyMongo`
- Por ultimo, configurar la URI de mongo dentro de las configuraciones de la aplicación flask, e inicializar la variable de conexión PyMongo.

Figura 73. Uso de libreria flask_pymongo

```
from flask_pymongo import PyMongo
app.config['MONGO_URI'] = 'mongodb+srv://administrator:<password>@elfaro-db.hirmz.mongodb.net/test'
mongo = PyMongo(app)
```

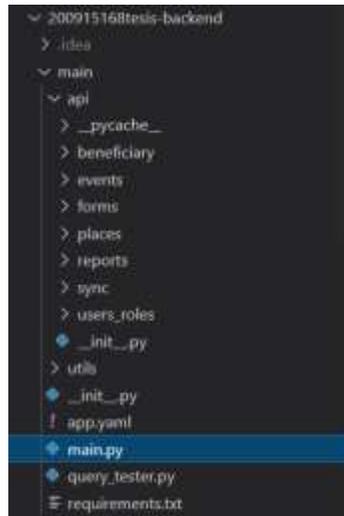
Fuente: elaboración propia, realizado con captura de pantalla.

5.2. Estructura del framework

Para poder realizar correctamente el despliegue del *backend*, se debe estructurar el proyecto de manera que se tenga por separado el código del api, el código de funciones generales (sección llamada “utils”), y los archivos principales que son:

- `main.py`
- `app.yaml`
- `requirements.txt`

Figura 74. Estructura de proyecto Flask



Fuente: elaboración propia, realizado con captura de pantalla.

El archivo `app.yaml`, es muy importante, porque contiene toda la información sobre el despliegue que se realizará en Google App Engine.

La estructura de Flask es bastante sencilla de implementar. El archivo principal llamado `main.py`, contiene el manejo de todos los servicios que se utilizarán para comunicarse con la base de datos. Este archivo se puede dividir en 3 partes importantes:

5.2.1. Bloque de dependencias

Este bloque consiste en importar las dependencias de Flask para que las características del *framework* estén disponibles dentro del proyecto.

Para hacer esto, en la parte superior del archivo, se debe indicar de la siguiente manera:

- `from flask import Flask`

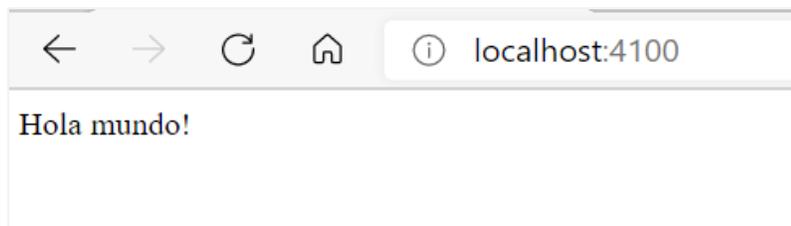
5.2.2. Bloque de api rest

El bloque de definiciones del API Rest se utilizará para especificar todas las rutas de las peticiones que el *backend* tendrá disponibles. Cada una de estas rutas debe especificar el nombre de la ruta y método de petición. Los métodos de petición se utilizarán de la siguiente manera:

- POST: para realizar cualquier petición de creación en la base de datos. También se utilizará para búsquedas avanzadas.
- PUT: para realizar cualquier petición de actualización en la base de datos.
- GET: para realizar cualquier petición de consulta de datos.
- DELETE: para realizar cualquier petición de eliminación en la base de datos.

Cada ruta definida tendrá anclado un método. Este método será el encargado de ejecutar cualquier acción previa a la acción en base de datos y posterior al recibir la respuesta de la petición, como en el siguiente ejemplo:

Figura 75. Ejemplo de petición GET, usando Flask



Fuente: elaboración propia, realizado con captura de pantalla.

5.2.3. Bloque de inicialización de servidor

Este bloque se encuentra al final del archivo. Su función es levantar el servidor en el puerto especificado, habilitando las rutas definidas en la sección del API Rest.

Figura 76. Estructura de ejemplo: Flask framework

```
1  ##BLOQUE DE DEPENDENCIAS -----
2  from flask import Flask
3  from flask_cors import CORS, cross_origin
4
5  app = Flask(__name__)
6  cors = CORS(app)
7  app.config['CORS_HEADERS'] = 'Content-Type'
8
9  ##BLOQUE API-REST -----
10 @app.route('/', methods=['GET'])
11 def HolaMundo():
12     return fnMensaje()
13
14 ##OTRAS FUNCIONES -----
15 def fnMensaje():
16     return "Hola mundo!"
17
18 # INICIALIZACION DEL SERVIDOR-----
19 if __name__ == '__main__':
20     app.run(debug=True, port=4100)
21
```

Fuente: elaboración propia, realizado con captura de pantalla.

5.3. Sincronización de aplicación móvil

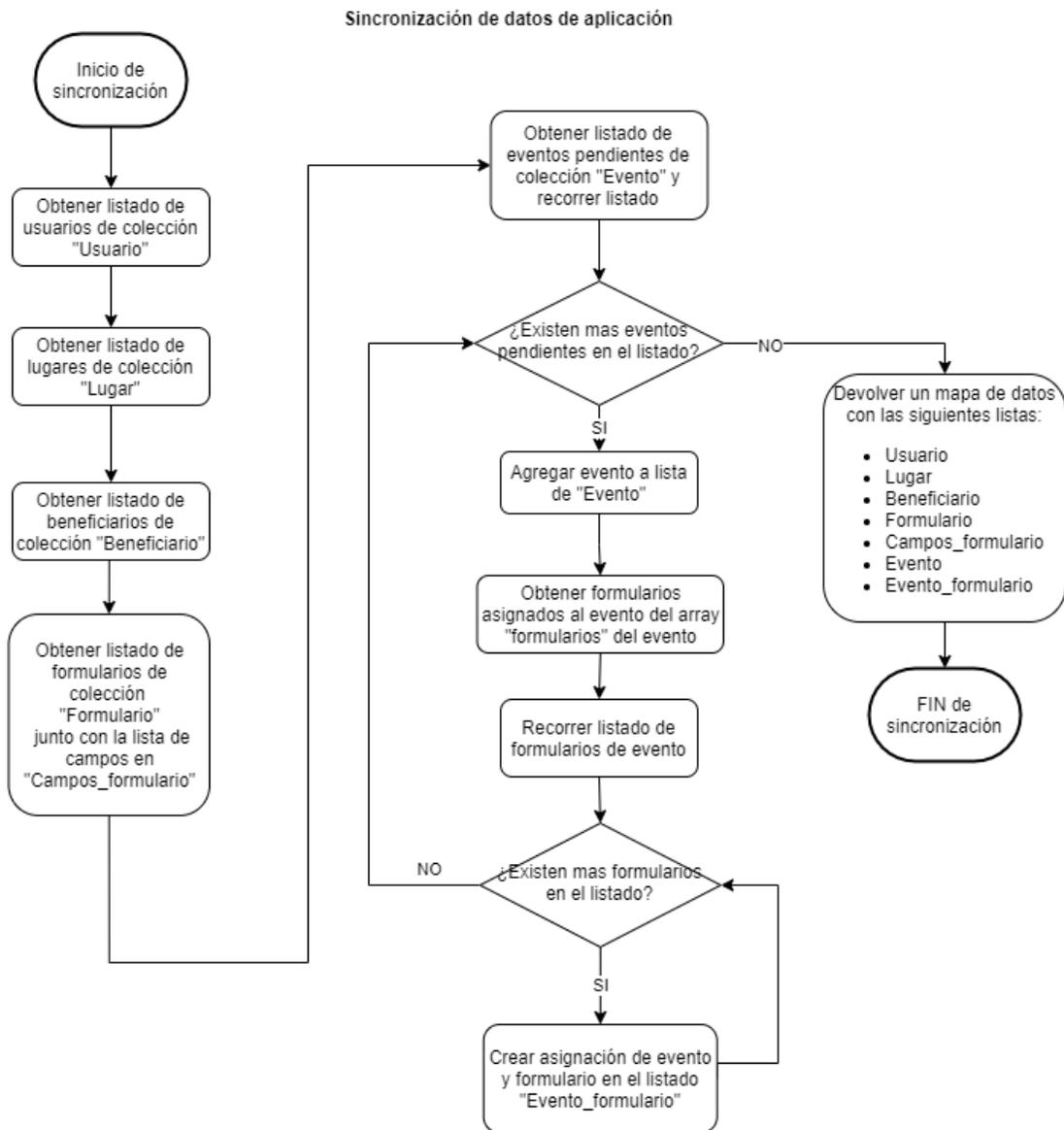
Los siguientes flujos representan los procesos de sincronización de datos para el correcto funcionamiento de la aplicación móvil, y la recuperación de los datos obtenidos con ella.

5.3.1. Sincronización de datos de aplicación móvil (en *backend*)

Este es el proceso que debe realizar el *backend* para obtener los datos que necesita la aplicación móvil:

- Obtener listado de usuarios
- Obtener listado de lugares
- Obtener listado de beneficiarios
- Obtener listado de formularios
- Obtener listado de campos de formulario
- Obtener listado de eventos pendientes
- Obtener listado de asignación de evento-formulario según los formularios obtenidos.
- Juntar las listas en un mapa de datos para devolver como respuesta del servicio `/sincronizar_datos_aplicación`.

Figura 77. Flujo de sincronización de datos de aplicación móvil (en *backend*)



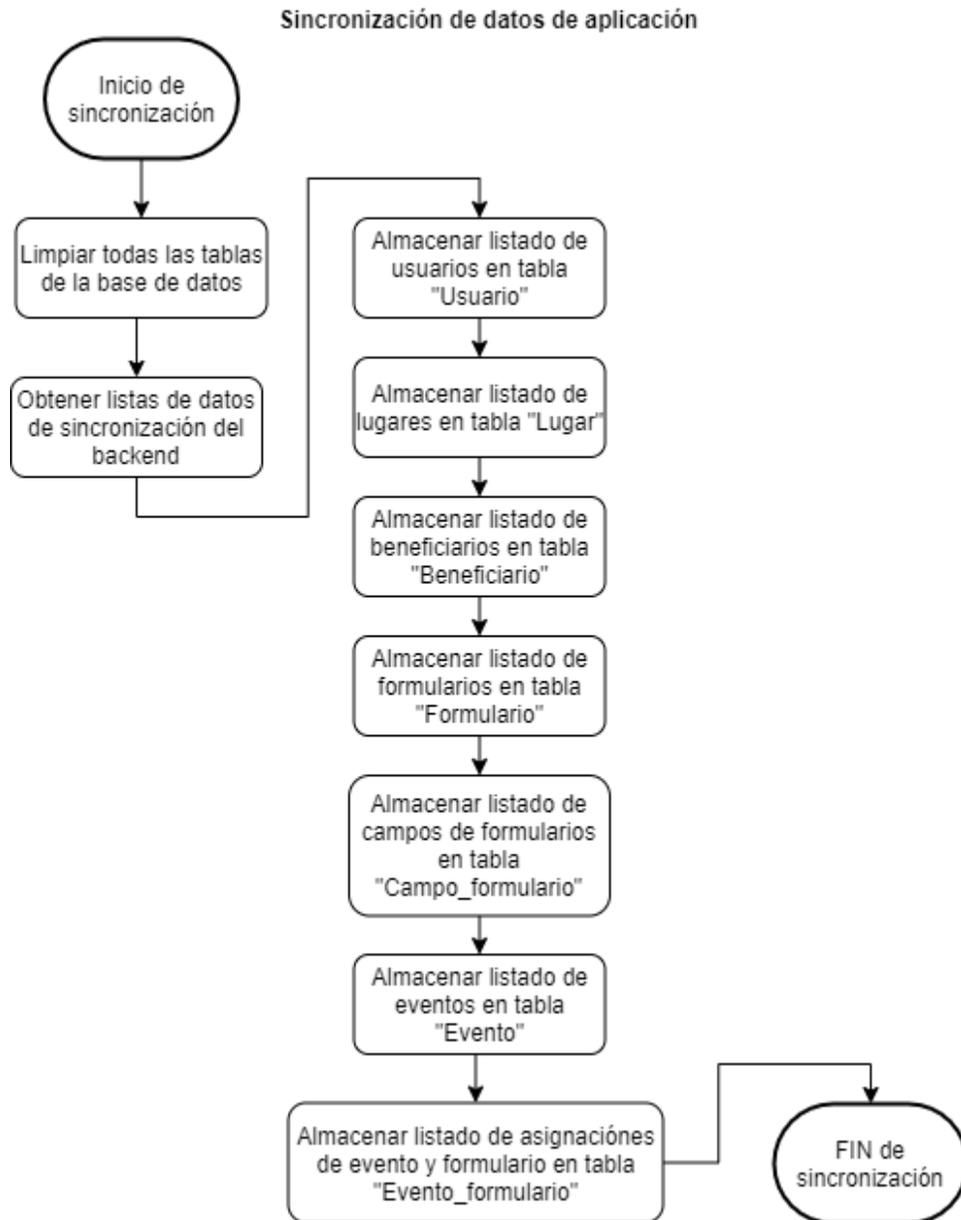
Fuente: elaboración propia, realizado con Draw.io.

5.3.2. Sincronización de datos de aplicación móvil (en aplicación móvil)

Este es el proceso que debe realizar la aplicación móvil para obtener los datos que necesita de la base de datos.

- Limpiar todas las tablas locales de la aplicación móvil (sqlite), validando que los datos que aun no han sido sincronizados se queden en sus respectivas tablas (datos de registros y nuevos beneficiarios). Además, eliminar los registros de la tabla Campo valor, correspondientes a los registros que fueron eliminados.
- Obtener listados de sincronización del *backend* del servicio /sincronizar_datos_aplicación.
- Recorrer la lista de cada grupo de datos obtenidos.
 - Almacenar usuarios en tabla Usuario
 - Almacenar lugares en tabla Lugar
 - Almacenar beneficiarios en tabla Beneficiario
 - Almacenar formularios en tabla Formulario
 - Almacenar campos de formulario en tabla Campo_formulario
 - Almacenar eventos en tabla Evento
 - Almacenar asignación de eventos y formularios en tabla Evento_formulario.

Figura 78. Flujo de sincronización de datos en aplicación móvil (en aplicación móvil)



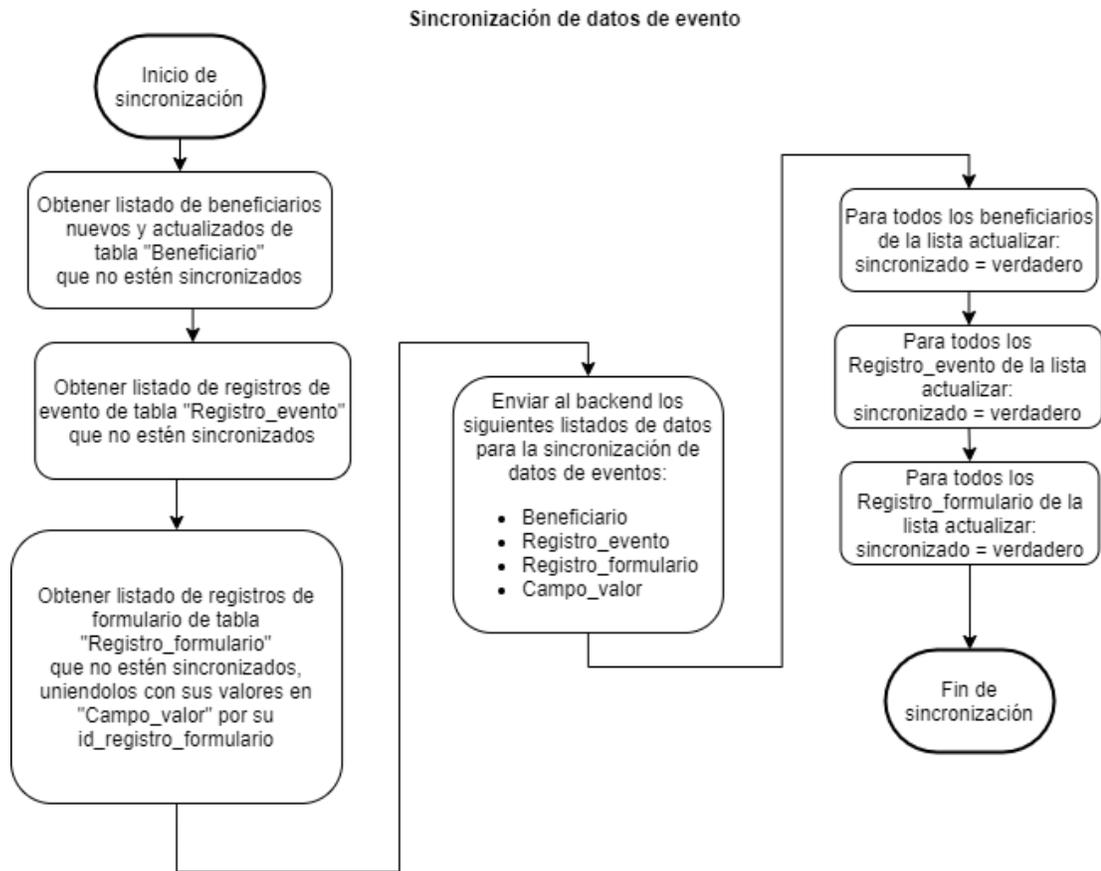
Fuente: elaboración propia, realizado con Draw.io.

5.3.3. Sincronización de datos de eventos (en aplicación móvil)

Este es el proceso que debe realizar la aplicación móvil para enviar los datos recopilados de nuevos beneficiarios, actualizaciones de beneficiarios y registros de eventos a la base de datos en la nube.

- Obtener listado de tabla Beneficiario, actualizados y nuevos, que no estén sincronizados.
- Obtener listado de tabla Registro_evento, que no estén sincronizados.
- Obtener listado tabla Registro_formulario, que no estén sincronizados.
- Obtener listado de tabla Campo_valor, que no estén sincronizados.
- Juntar las listas en un mapa de datos para enviar los datos a sincronizar al servicio del *backend*.
- Enviar lista de datos a través del servicio `/sincronizar_beneficiarios_y_registros_evento`.
- Actualizar listado de tabla Beneficiario, sincroinzado = verdadero.
- Actualizar listado de tabla Registro_evento, sincroinzado = verdadero.
- Actualizar listado tabla Registro_formulario, sincroinzado = verdadero.

Figura 79. Flujo de sincronización de datos de eventos (aplicación móvil)



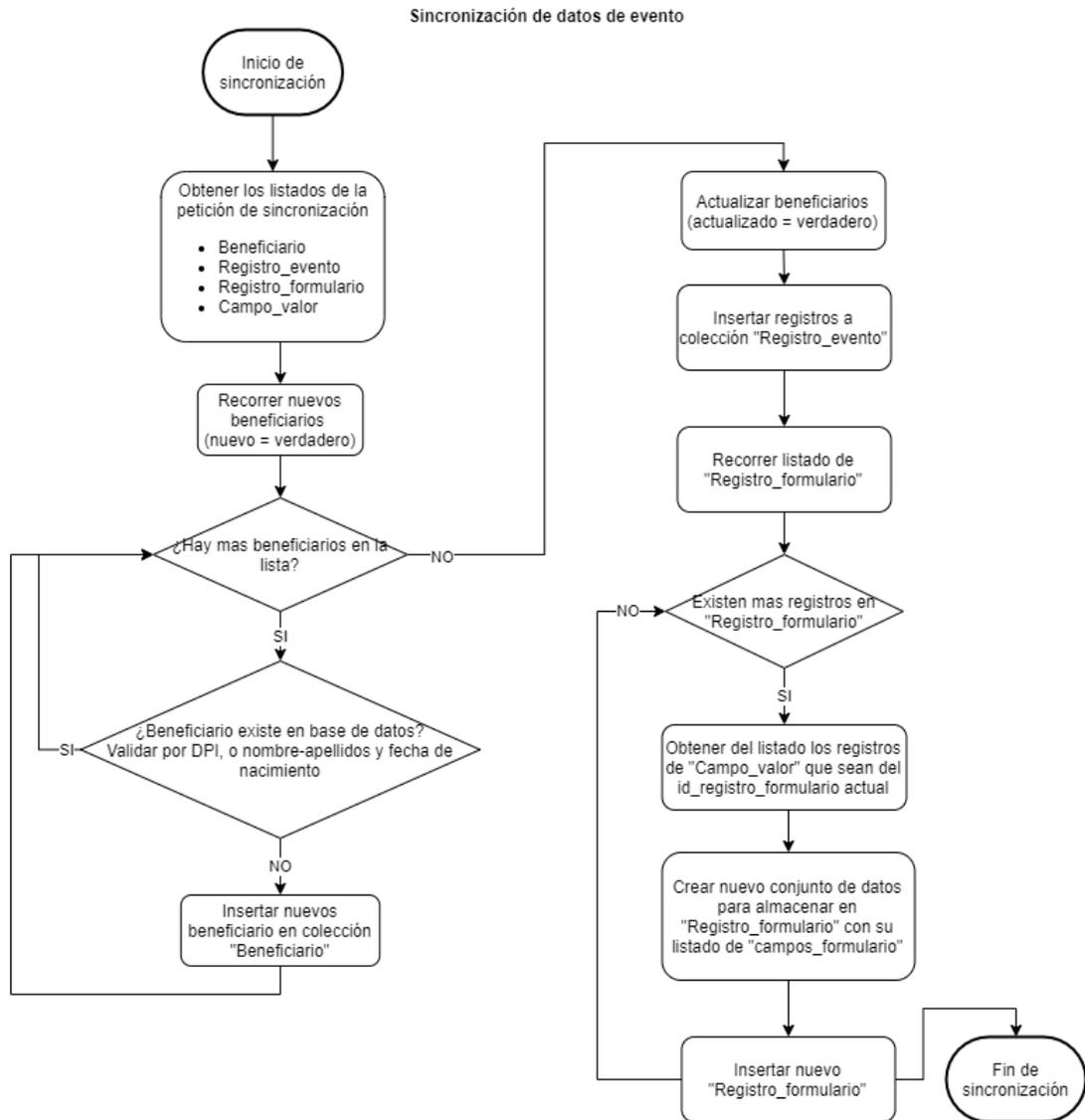
Fuente: elaboración propia, realizado con Draw.io.

5.3.4. Sincronización de datos de eventos (*backend*)

Este es el proceso que debe realizar el *backend* para enviar los datos recopilados de nuevos beneficiarios, actualizaciones de beneficiarios y registros de eventos a la base de datos en la nube.

- Recorrer la lista de cada grupo de datos obtenidos
 - Obtener listado de tabla Beneficiario
 - Crear beneficiarios donde su atributo Nuevo, sea verdadero
 - Actualizar beneficiarios donde su atributo Actualizado, sea verdadero.
 - Crear registros en colección RegistroEvento
 - Recorrer listado de registros de formulario
 - Obtener los registros de la lista campo_valor que tengan el id_registro_formulario actual.
 - Crear un nuevo registro de datos para la colección RegistroFormulario, con su listado de campos_formulario.

Figura 80. Flujo de sincronización de datos de eventos (*backend*)



Fuente: elaboración propia, realizado con Draw.io.

6. DEFINICION DE SERVICIOS DE API

En este apartado se encuentra la definición de cada servicio del Api Rest. Cada definición está acompañada de la ruta, método, parámetros y el cuerpo que debe de llevar la petición. Las peticiones para las que se especifican los valores del cuerpo poseen también el tipo de cada valor del cuerpo del mensaje.

6.1.1. Funciones de usuarios

A continuación, se detallan las funciones de la colección Usuario, de la base de datos.

6.1.1.1. Creación de usuario

- Ruta: /createUser
- Método: *post*
- Parámetros: n/a
- Cuerpo: json
 - Nombre: cadena
 - Usuario: cadena
 - Email: cadena
 - Password: cadena
 - Id_rol: código generado por MongoDB
 - Contraseña

Nota: la creación de usuario creará también el usuario de firebase para el inicio de sesión. La contraseña se almacena únicamente en firebase.

6.1.1.2. Listar usuarios

- Ruta: /readUsers
- Método: get
- Parámetros: n/a
- Cuerpo: n/a

6.1.1.3. Editar de usuario

- Ruta: /updateUser
- Método: put
- Parámetros: n/a
- Cuerpo: json
 - Id_usuario: código generado por MongoDB
 - Nombre: cadena
 - Usuario: cadena
 - Email: cadena
 - Password: cadena
 - Id_rol: código generado por MongoDB

6.1.1.4. Eliminar usuario

- Ruta: /deleteUser
- Método: delete
- Parámetros

- Id_usuario: código generado por MongoDB
- Cuerpo: n/a

6.1.2. Funciones de Roles

A continuación, se detallan las funciones de la colección “Rol” de la base de datos.

6.1.2.1. Creación de rol

- Ruta: /createRol
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Nombre: cadena
 - Permisos: lista enteros

6.1.2.2. Listar roles

- Ruta: /readRoles
- Método: get
- Parámetros: n/a
- Cuerpo: n/a

6.1.2.3. Editar de rol

- Ruta: /updateRol

- Método: put
- Parámetros: n/a
- Cuerpo: json
 - Id_rol: código generado por MongoDB
 - Nombre: cadena
 - Permisos: lista enteros

6.1.2.4. Eliminar rol

- Ruta: /deleteRol
- Método: delete
- Parámetros
 - Id_rol: código generado por MongoDB
- Cuerpo: n/a

6.1.3. Funciones de beneficiarios

A continuación, se detallan las funciones de la colección “Beneficiarios” de la base de datos.

6.1.3.1. Creación de beneficiario

- Ruta: /createBeneficiary
- Método: post
- Parámetros: n/a
- Cuerpo: json

- Nombre: cadena
- Apellidos: cadena
- DPI: cadena
- Fecha_nacimiento: fecha
- Dirección: cadena
- Telefono: cadena
- Email: cadena
- Sexo: entero
- Nacionalidad: cadena
- Id_lugar: código generado por MongoDB

6.1.3.2. Listar beneficiario

- Ruta: /readBeneficiaries
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Nombre: cadena
 - Apellidos: cadena
 - DPI: cadena
 - Dirección: cadena
 - Telefono: cadena
 - Email: cadena
 - Sexo: entero
 - Nacionalidad: cadena
 - Id_lugar: código generado por MongoDB
 - Fecha_nacimiento_ini: fecha
 - Fecha_nacimiento_fin: fecha

6.1.3.3. Editar beneficiario

- Ruta: /updateBeneficiary
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Id_beneficiario: código generado por MongoDB
 - Nombre: cadena
 - Apellidos: cadena
 - DPI: cadena
 - Fecha_nacimiento: fecha
 - Dirección: cadena
 - Telefono: cadena
 - Email: cadena
 - Sexo: entero
 - Nacionalidad: cadena
 - Id_lugar: código generado por MongoDB

Los parámetros del cuerpo son opcionales. Servirán como filtros de búsqueda para recuperar los registros de beneficiario.

6.1.3.4. Eliminar beneficiario

- Ruta: /deleteBeneficiary
- Método: delete
- Parámetros
 - Id_rol: código generado por MongoDB

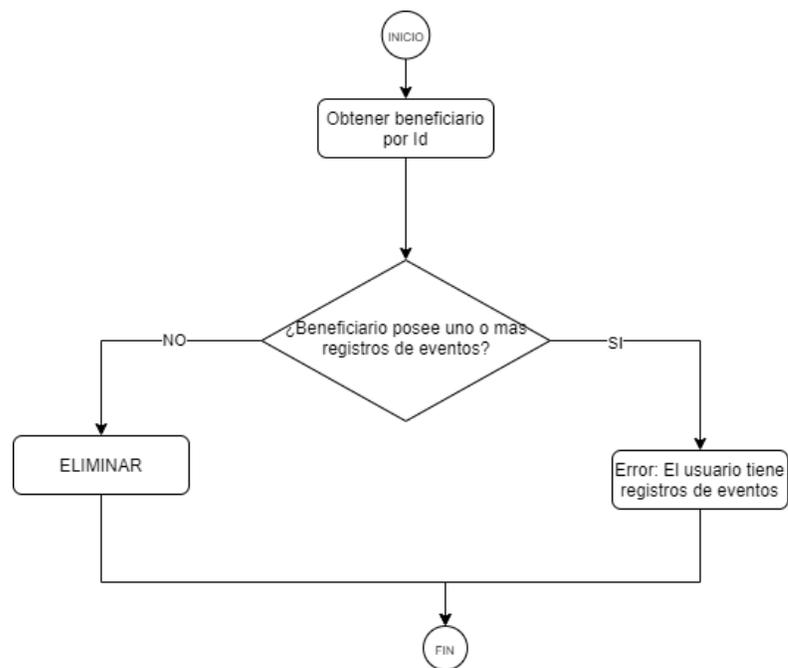
- Cuerpo: n/a

6.1.3.4.1. Flujo de eliminación de beneficiario

El flujo de eliminación de beneficiario tiene como objetivo verificar si el beneficiario ya cuenta con algún registro de evento, antes de realizar su eliminación de la base de datos.

- Si el beneficiario cuenta con algún registro de evento no se podrá eliminar para no perder esa relación con el registro de evento.
- En caso contrario, el usuario se podrá eliminar sin problema.

Figura 81. Flujo de eliminación de beneficiario



Fuente: elaboración propia, realizado con Draw.io.

6.1.4. Funciones de formularios

A continuación, se detallan las funciones de la colección “Formulario” de la base de datos.

6.1.4.1. Creación de formulario

- Ruta: /createForm
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Nombre: cadena
 - Descripción: cadena
 - Campos: lista de campo_formulario
 - Nombre_campo: cadena
 - Tipo: entero
 - ✓ 0: caja de texto simple
 - ✓ 1: caja de texto multilinea
 - ✓ 2: grupo de opciones
 - ✓ 3: grupo de selección multiple
 - Sección: entero
 - Valores predefinidos: lista de cadenas

6.1.4.2. Listar formularios

- Ruta: /readForms
- Método: get
- Parámetros: n/a
- Cuerpo: n/a

6.1.4.3. Editar de formulario

- Ruta: /editForm
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Id_formulario: entero
 - Nombre: cadena
 - Descripción: cadena
 - Campos: lista de campo_formulario
 - Id_campo_formulario: código generado por MongoDB
 - Nombre_campo: cadena
 - Tipo: entero
 - ✓ 0: caja de texto simple
 - ✓ 1: caja de texto multilinea
 - ✓ 2: grupo de opciones
 - ✓ 3: grupo de selección multiple
 - Sección: entero
 - Valores predefinidos: lista de cadenas

6.1.4.4. Eliminar formulario

- Ruta: /deleteForm
- Método: delete
- Parámetros
 - Id_formulario: código generado por MongoDB
- Cuerpo: n/a

6.1.5. Funciones de eventos

A continuación, se detallan las funciones de la colección “Evento” de la base de datos.

6.1.5.1. Creación de evento

- Ruta: /createEvent
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Nombre: cadena
 - Descripción: cadena
 - Direccion: cadena
 - Id_lugar: código generado por MongoDB
 - Fecha_hora_inicio: fecha y hora
 - Fecha_hora_fin: fecha y hora
 - Cupo: entero

- Tipo: entero
 - 0: salud integral
 - 1: construcción
 - 3: beneficio comunitario
 - 4: otros

- Registro único: booleano

6.1.5.2. Listar eventos

- Ruta: /readEvents
- Método: post
- Parámetros: n/a
- Cuerpo: JSON
 - Id_evento: código generado por MongoDB
 - Nombre: cadena
 - Descripción: cadena
 - Direccion: cadena
 - Id_lugar: código generado por MongoDB
 - Fecha_hora_inicio: fecha y hora
 - Fecha_hora_fin: fecha y hora
 - Cupo: entero
 - Tipo: entero
 - 0: salud integral
 - 1: construcción
 - 3: beneficio comunitario

- 4: otros
- Registro único: booleano

Los parámetros del cuerpo son opcionales. Servirán como filtros de búsqueda para recuperar los registros de eventos.

6.1.5.3. Editar de evento

- Ruta: /editEvent
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Id_evento: código generado por MongoDB
 - Nombre: cadena
 - Descripción: cadena
 - Direccion: cadena
 - Id_lugar: código generado por MongoDB
 - Fecha_hora_inicio: fecha y hora
 - Fecha_hora_fin: fecha y hora
 - Cupo: entero
 - Tipo: entero
 - 0: salud integral
 - 1: construcción
 - 3: beneficio comunitario
 - 4: otros
 - Registro único: booleano

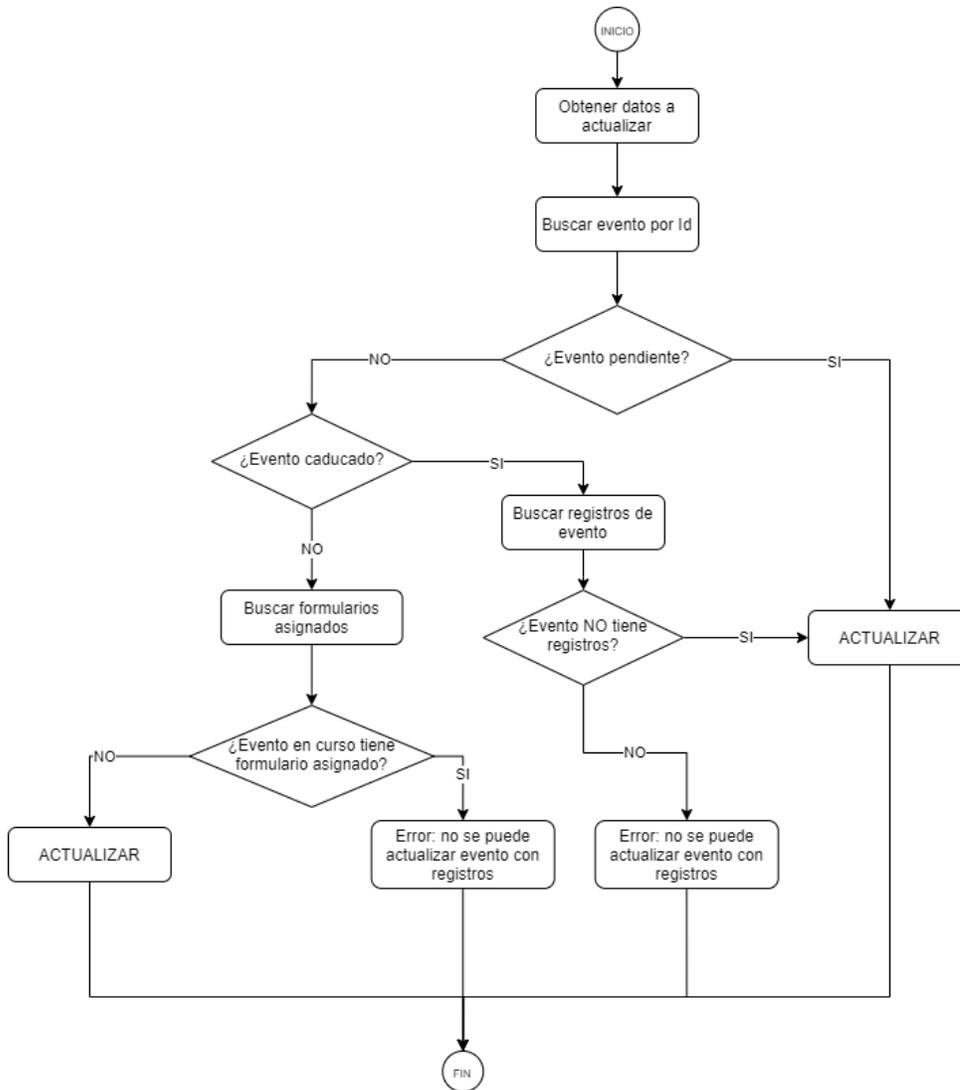
6.1.5.3.1. Flujo de actualización de evento

Este flujo verificará el estado del evento antes de hacer su actualización en la base de datos, la cuál se realizará cuando:

- El evento esta pendiente
- El evento esta caducado, y no tiene registros de evento
- El evento esta en curso y no tiene formulario asignado

En caso contrario, no se podrá realizar la actualización del evento para no perjudicar a los eventos en curso, o eventos caducados que ya cuenten con registros.

Figura 82. Flujo de actualización de evento



Fuente: elaboración propia, realizado con Draw.io.

6.1.5.4. Eliminar evento

- Ruta: /deleteEvent
- Método: delete

- Parámetros
 - Id_evento: código generado por MongoDB
- Cuerpo: n/a

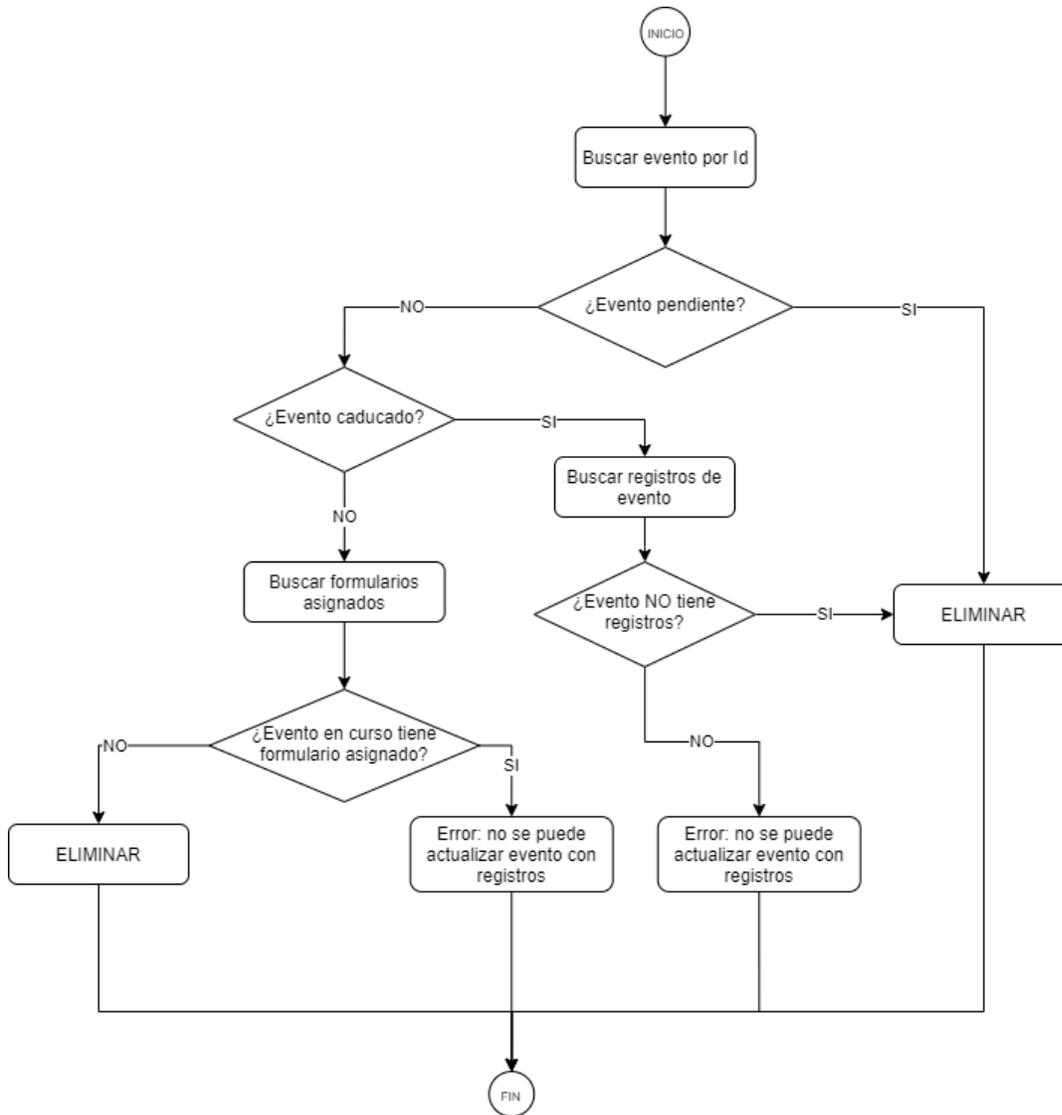
6.1.5.4.1. Flujo de eliminación de evento

Este flujo verificará el estado del evento antes de hacer su eliminación en la base de datos, la cuál se realizará cuando:

- El evento esta pendiente
- El evento esta caducado, y no tiene registros de evento
- El evento esta en curso y no tiene formulario asignado

En caso contrario, no se podrá realizar la actualización del evento para no perjudicar a los eventos en curso, o eventos caducados que ya cuenten con registros.

Figura 83. Flujo de eliminación de evento



Fuente: elaboración propia, realizado con Draw.io.

6.1.5.5. Asignación Formulario-Evento

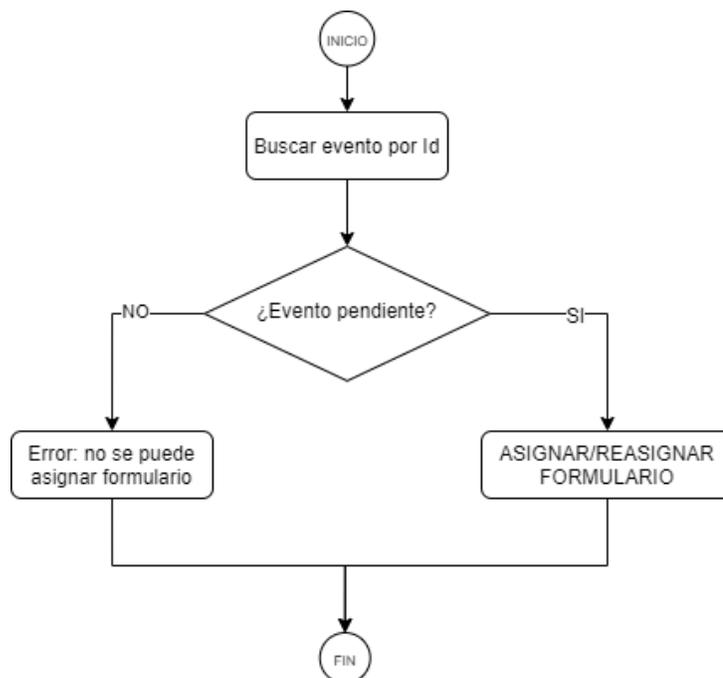
- Ruta: /eventFormAssignment
- Método: post

- Parámetros: n/a
- Cuerpo: json
 - Id_evento: código generado por MongoDB
 - Formularios: arreglo de Ids de formularios

6.1.5.5.1. Flujo de asignación de formulario a evento

Los formularios se podrán asignar únicamente a los eventos que estén pendientes. Los eventos pendientes son los que su fecha de inicio es mayor a la actual.

Figura 84. Flujo de asignación de formulario a evento



Fuente: elaboración propia, realizado con Draw.io.

6.1.5.6. Crear registro de evento

- Ruta: /eventFormAssignment
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Id_evento: código generado por MongoDB
 - Id_beneficiario: código generado por MongoDB
 - Nombre_formulario: cadena
 - Fecha_hora: fecha
 - Registros_formulario: lista de registros de formulario
 - Campos formulario: lista de campos del formulario
 - ✓ Valor: cadena
 - ✓ Nombre: cadena
 - ✓ Sección: entero

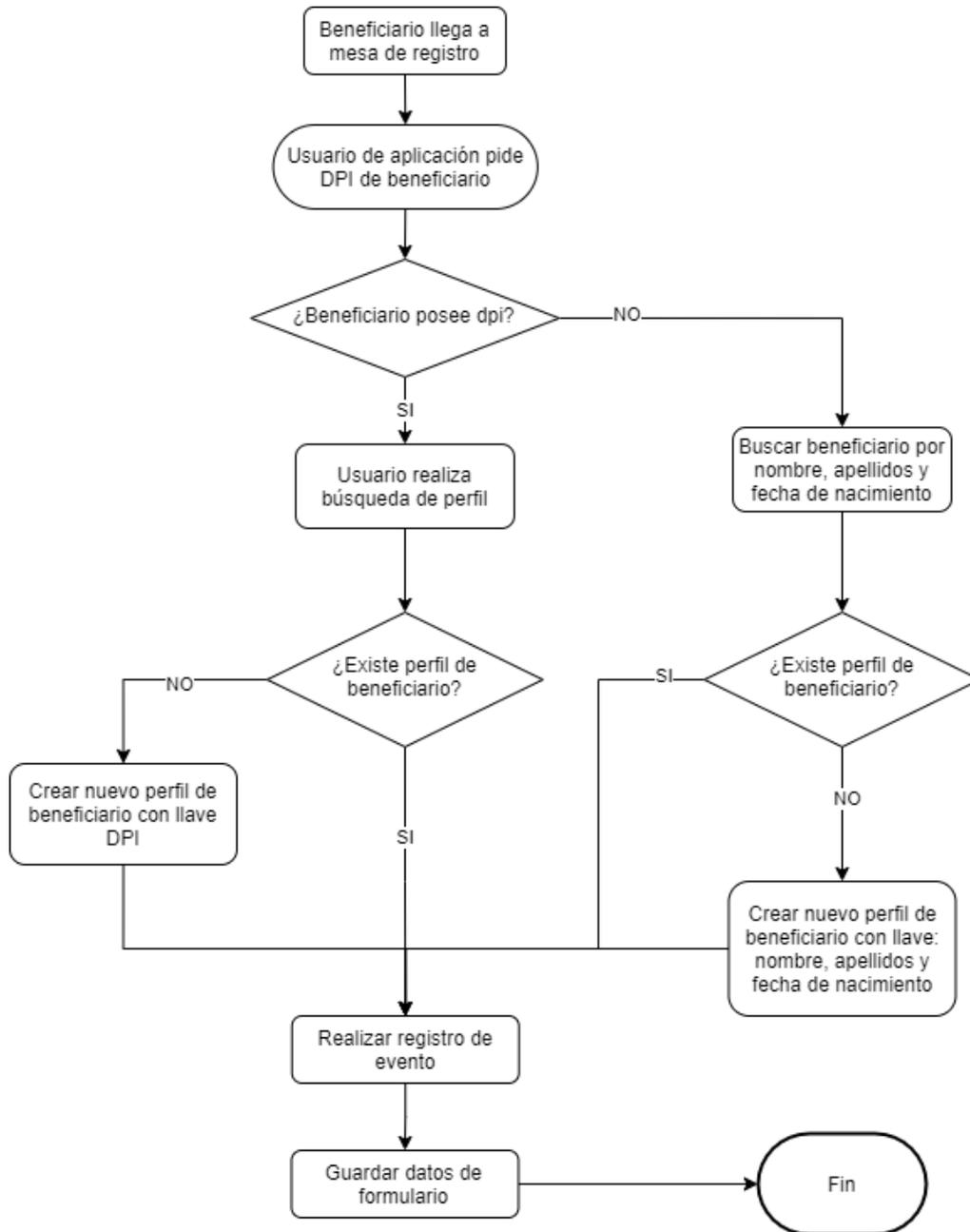
6.1.5.6.1. Flujo de evento de registro libre

Los beneficiarios pueden asistir a eventos de registro libre, donde pueden llegar una o más veces y ser atendidos.

- Antes de realizar cualquier registro de evento, la aplicación debe encontrar el perfil del beneficiario.
 - Encontrar perfil de beneficiario por DPI

- Encontrar perfil de beneficiario por nombre, apellidos y fecha de nacimiento.
- Si existe perfil.
 - Crear registro en evento y guardar en base de datos
- Si no existe perfil, crear un nuevo perfil de beneficiario
 - Crear perfil con DPI
 - Si no posee DPI, crear perfil con nombre, apellidos y fecha de nacimiento.
 - Crear registro en evento y guardar en base de datos.

Figura 85. Flujo de evento de registro libre



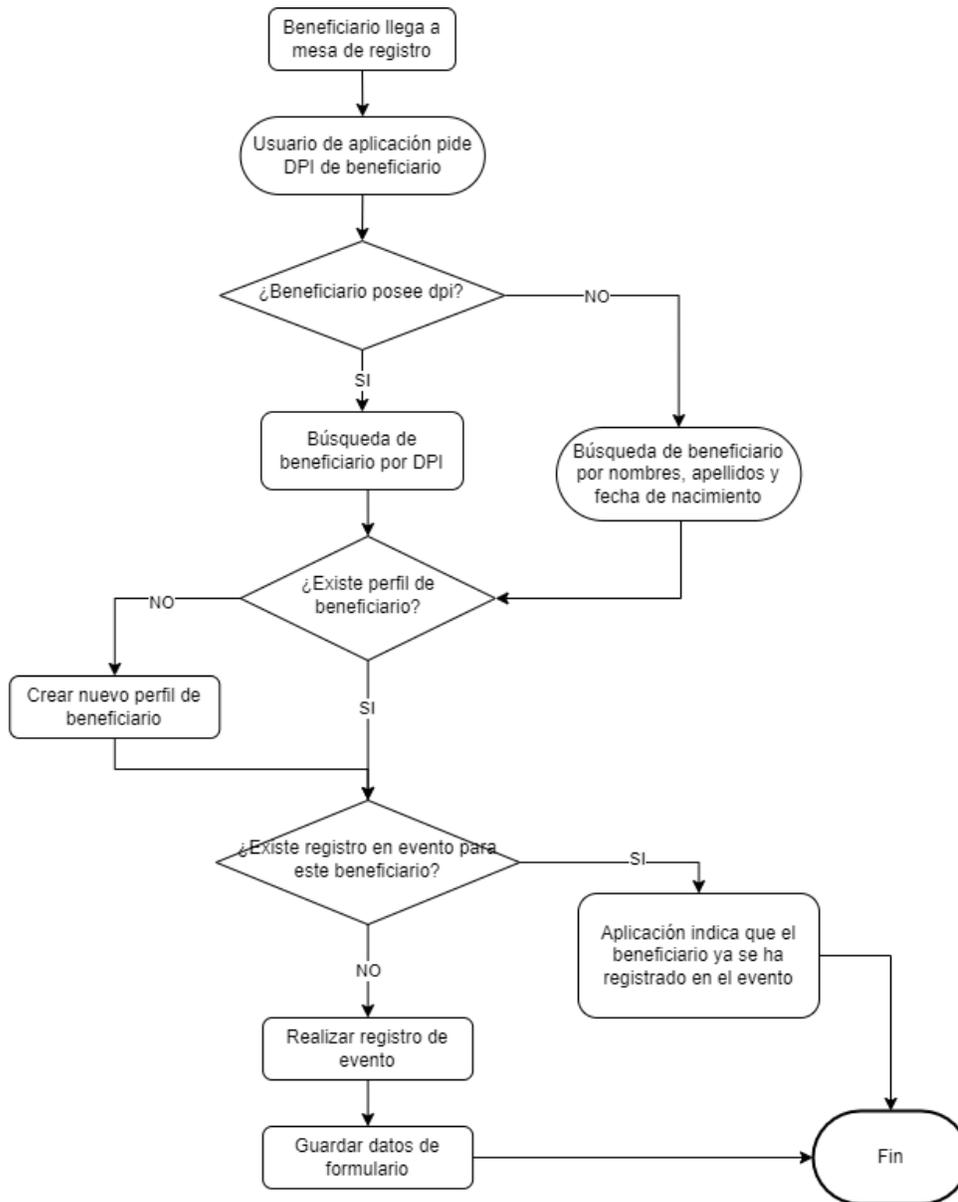
Fuente: elaboración propia, realizado con Draw.io.

6.1.5.6.2. Flujo de evento de registro único

Los beneficiarios pueden asistir a eventos de registro único, donde únicamente pueden registrarse una vez.

- Antes de realizar cualquier registro de evento, la aplicación debe encontrar el perfil del beneficiario.
 - Encontrar perfil de beneficiario por DPI
 - Encontrar perfil de beneficiario por nombre, apellidos y fecha de nacimiento.
 - Si existe perfil.
 - Buscar si existe registro previo del beneficiario en el evento
 - ✓ Si existe: no realizar registro
 - ✓ Si no existe: crear registro en evento y guardar en base de datos.
 - Si no existe perfil, crear un nuevo perfil de beneficiario
 - Crear perfil con DPI
 - Si no posee DPI, crear perfil con nombre, apellidos y fecha de nacimiento.
 - Crear registro en evento y guardar en base de datos.

Figura 86. Flujo de evento de registro único



Fuente: elaboración propia, realizado con Draw.io.

6.1.5.7. Listar registros de evento

- Ruta: /eventFormAssignment
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Id_evento: código generado por MongoDB

6.1.6. Funciones de lugares

A continuación, se detallan las funciones de la colección Ciudad, y Lugar de la base de datos.

6.1.6.1. Creación de ciudad

- Ruta: /createCity
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Nombre: cadena
 - País: cadena

6.1.6.2. Listar ciudades

- Ruta: /readCities
- Método: get
- Parámetros: n/a

- Cuerpo: n/a

6.1.6.3. Editar de ciudad

- Ruta: /createCity
- Método: post
- Parámetros: n/a
- Cuerpo: json
 - Id_ciudad: código generado por MongoDB
 - Nombre: cadena
 - País: cadena

6.1.6.4. Eliminar ciudad

- Ruta: /deleteCity
- Método: delete
- Parámetros
 - Id_ciudad
- Cuerpo: n/a

6.1.6.5. Creación de lugar

- Ruta: /createTown
- Método: post
- Parámetros: n/a
- Cuerpo: json

- Nombre: cadena
- Id_ciudad: código generado por MongoDB

6.1.6.6. Listar lugares

- Ruta: /readTowns
- Método: get
- Parámetros: n/a
- Cuerpo: n/a

6.1.6.7. Editar de lugar

- Ruta: /updateTown
 - Método: post
 - Parámetros: n/a
 - Cuerpo: json
- Id_lugar: código generado por MongoDB
 - Nombre: cadena
 - Id_ciudad: código generado por MongoDB

6.1.6.8. Eliminar lugar

- Ruta: /deleteTown
 - Método: delete
 - Parámetros
- Id_donacion: código generado por MongoDB

- Cuerpo: n/a

6.1.7. Funciones de sincronización

Se detallan las funciones de la actualización y sincronización de datos de aplicación.

6.1.7.1. Actualizar datos de aplicación

- Ruta: /actualizar_datos_aplicacion
- Método: get
- Parámetros: n/a
- Cuerpo: n/a

6.1.7.2. Sincronizar beneficiarios y registros de evento

- Ruta: /sincronizar_beneficiarios_y_registros_evento
- Método: post
- Parámetros
 - beneficiarios: lista de beneficiarios para crear y actualizar
- registro_evento: lista de registros de evento
- registro_formulario: lista de registros de formulario
- campo_valor: lista de campos que pertenecen al registro de formulario

7. ESPECIFICACIONES DEL FRONTEND

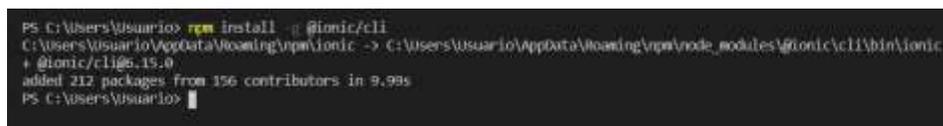
7.1. Aplicación móvil

En esta sección se detallan las especificaciones técnicas de la tecnología con que se desarrollará la aplicación móvil.

7.1.1. Instalación de Ionic

- Previo a iniciar la construcción de la aplicación móvil, se instalará ionic ejecutando lo siguiente en la línea de comandos:
 - `npm install -g @ionic/cli`

Figura 87. Instalación de Ionic



```
PS C:\Users\Usuario> npm install -g @ionic/cli
C:\Users\Usuario\AppData\Local\Programs\Node.js> C:\Users\Usuario\AppData\Local\Programs\Node.js\node_modules\npm\bin\npm-cli.js install -g @ionic/cli@6.15.0
added 212 packages from 156 contributors in 9.98s
PS C:\Users\Usuario>
```

Fuente: elaboración propia, realizado con captura de pantalla.

7.1.2. Inicialización de proyecto de Ionic

- Para inicializar un nuevo proyecto de Ionic, se ejecutará en la línea de comandos lo siguiente:
 - `ionic start <nombre_de_aplicación> <tipo_de_diseño>`

Donde nombre_de_aplicación, es el nombre que se le dará a la aplicación móvil y tipo_de_diseño, es el tipo de diseño predefinido. Los tipos de diseño que se pueden escoger en son: *tabs*, *sidemenu*, *blank*.

Figura 88. Inicialización de proyecto de Ionic

```
$ ionic start faroapp sidemenu
✓ Preparing directory .\faroapp in 799.60pp
✓ Downloading and extracting sidemenu starter in 818.61ms
--no-conda: Integrate your new app with capacitor to target native iOS and Android? No
Installing dependencies may take several minutes.

-----
Ionic Advisory, tailored solutions and expert services by Ionic
-----
Go to market faster
Real-time troubleshooting and guidance
Custom training, best practices, code and architecture reviews
Customized strategies for every phase of the development lifecycle
Learn more: https://ion.link/advisory
-----
> npm.cmd i
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDoS regression when used
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Al finalizar la instalación, se puede ver la plantilla de aplicación funcionando con el comando:
 - ionic serve

Figura 89. Ejecución local de aplicación móvil

```
$ ionic serve
> ng.cmd run app:serve --host=localhost --port=8100
[ng] Compiling @angular/core : es2015 as esm2015
[ng] Compiling @angular/common : es2015 as esm2015
[ng] Compiling @angular/platform-browser : es2015 as esm2015
[ng] Compiling @angular/forms : es2015 as esm2015
[ng] Compiling @angular/platform-browser-dynamic : es2015 as esm2015
[ng] Compiling @angular/router : es2015 as esm2015
[ng] Compiling @ionic/angular : es2015 as esm2015
[ng] - Generating browser application bundles...
[ng] ✓ Browser application bundle generation complete.
[ng] Initial chunk files: ...
```

Fuente: elaboración propia, realizado con captura de pantalla.

Figura 91. Componente ion-menu



Fuente: elaboración propia, realizado con captura de pantalla.

7.1.3.1.2. Barra de herramientas

Se utilizará para mostrar información y acciones relacionadas con la sección de la aplicación, como búsquedas, guardar, edición.

Figura 92. Componente ion-toolbar



Fuente: elaboración propia, realizado con captura de pantalla.

7.1.3.1.3. Lista

Las listas permiten mostrar de manera ordenada conjuntos de información. Ayudará a mostrar las listas de eventos en la aplicación.

Figura 93. Componente ion-list



Fuente: elaboración propia, realizado con captura de pantalla.

7.1.3.1.4. Tarjetas

Se utilizarán estos componentes como contenedores de información, tanto en formularios, como en las listas.

Figura 94. Componente ion-card

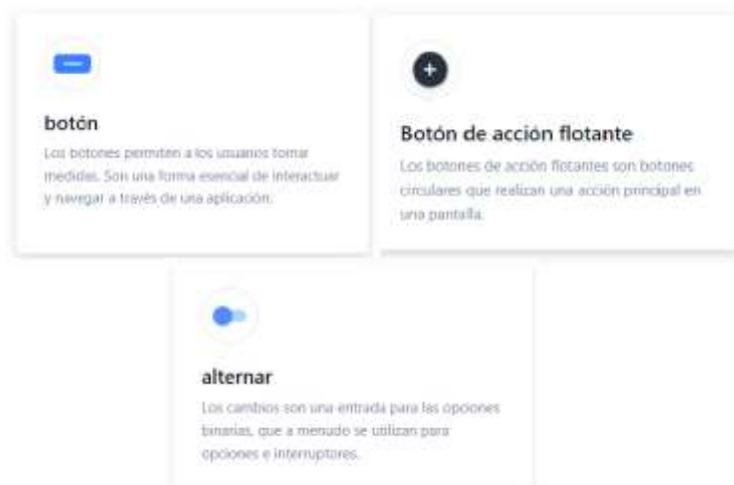


Fuente: elaboración propia, realizado con captura de pantalla.

7.1.3.2. Componentes de interacción y formularios

Los distintos componentes de botones ayudarán a realizar acciones. Cada acción que se dispare en la aplicación dependerá de la interacción con estos componentes.

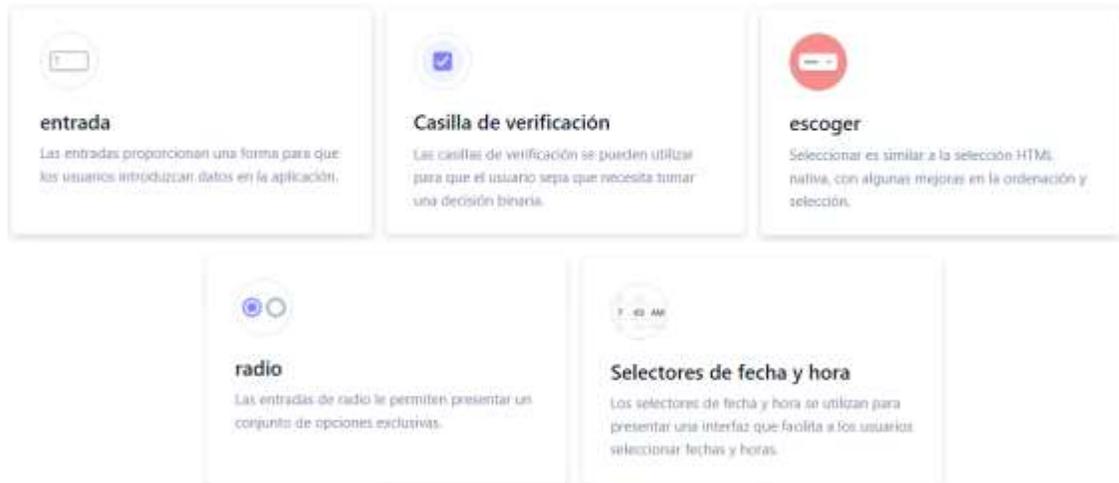
Figura 95. Componentes de botones y acciones de Ionic



Fuente: elaboración propia, realizado con captura de pantalla.

Los componentes de formularios se utilizan dentro de las pantallas para capturar datos e información ingresada por el usuario. Dentro de estos elementos se tiene entrada de texto, selector de fecha, casilla de verificación, elemento de opciones y elemento de selección.

Figura 96. Componentes de formularios de Ionic



Fuente: elaboración propia, realizado con captura de pantalla.

7.1.3.3. Iconos, alertas y notificaciones

Para que la aplicación sea amigable al usuario, se hará uso de diferentes iconos para indicar las diferentes secciones de la aplicación y las acciones que se pueden realizar.

Muchas de las acciones dentro de la aplicación generarán notificaciones que el usuario podrá leer para entender lo que ha sucedido. Se utilizarán los iconos de Ionic, las alertas y el componente ion-toast.

Figura 97. Iconos, alertas y notificaciones de Ionic



Fuente: elaboración propia, realizado con captura de pantalla.

7.2. Plataforma web administrativa

En esta sección se detallan las especificaciones técnicas de la tecnología con que se desarrollará el sitio administrativo.

7.2.1. Instalación y configuraciones iniciales de React

- Previo a realizar el proyecto de React, se instalarán las dependencias necesarias para su funcionamiento. El primer paso es crear el proyecto de React con la ejecución del siguiente comando:
 - `Npx create-react-app <nombre_aplicacion>`

Donde nombre_aplicacion, es el nombre que se le dará a la aplicación de React.

Figura 98. Creación de aplicación de React

```
PS C:\Users\Usuario\far-dev> npx create-react-app faro-admin
npm: instaló 67 en 10.966s

Creating a new React app in C:\Users\Usuario\far-dev\far-admin.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

> core-js@2.6.12 postinstall C:\Users\Usuario\far-dev\far-admin\node_modules\babel-runtime\node_modules\core-js
> node -e "try(require('./postinstall'))catch(e){}"

> core-js@3.12.1 postinstall C:\Users\Usuario\far-dev\far-admin\node_modules\core-js
> node -e "try(require('./postinstall'))catch(e){}"

> core-js-pure@3.12.1 postinstall C:\Users\Usuario\far-dev\far-admin\node_modules\core-js-pure
> node -e "try(require('./postinstall'))catch(e){}"
```

Fuente: elaboración propia, realizado con captura de pantalla.

Figura 99. Aplicación de React creada con éxito

```
Success! Created faro-admin at C:\Users\Usuario\far-dev\far-admin
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

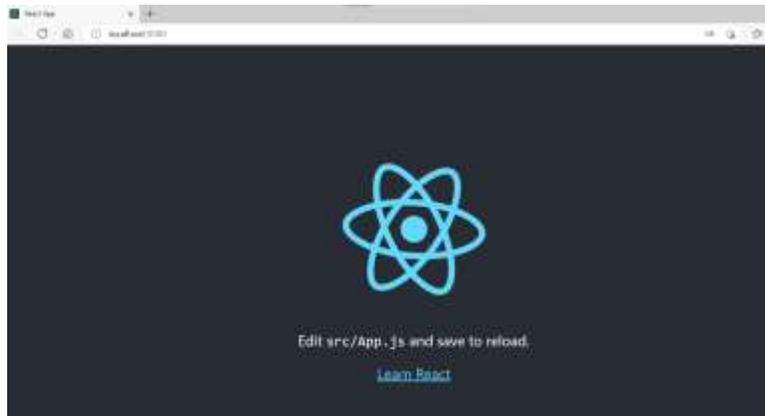
  cd faro-admin
  npm start

Happy hacking!
PS C:\Users\Usuario\far-dev> |
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Para ejecutar la aplicación creada, basta con ejecutar lo siguiente en la línea de comandos:
 - npm start

Figura 100. Vista de aplicación de React en ejecución



Fuente: elaboración propia, realizado con captura de pantalla.

- Por último, se instalará la librería de Material-UI ejecutando en la línea de comandos:
 - `npm install @material-ui/core`

Figura 101. Instalación de Material-UI

```
PS C:\Users\Usuario\far0-dev\far0-admin> npm install @material-ui/core
npm WARN @babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining@7.13.12 requires a peer of @babel/core@>=7.12.0 but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os": "darwin", "arch": "arm64"} but got {"os": "windows", "arch": "x64"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os": "darwin", "arch": "arm64"} but got {"os": "windows", "arch": "x64"}
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os": "darwin", "arch": "arm64"} but got {"os": "windows", "arch": "x64"}

+ @material-ui/core@4.11.4
added 33 packages from 49 contributors and audited 2004 packages in 23.866s

141 packages are looking for funding
  run `npm fund` for details
```

Fuente: elaboración propia, realizado con captura de pantalla.

7.2.2. Componentes de pantallas de sitio administrativo

Para el sitio administrativo se hará uso de los componentes de Material-UI. Material-UI es un set de componentes visuales de React que implementan principios de *Material Design*.

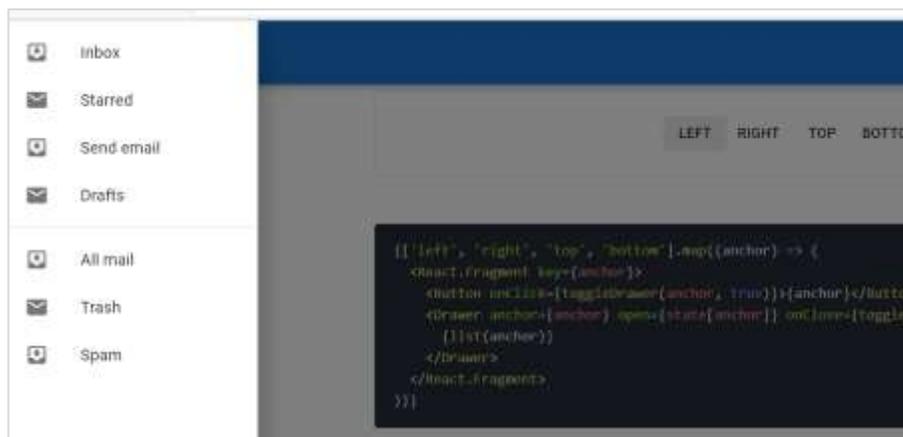
7.2.2.1. Componentes de navegación

A continuación, los componentes de navegación de React para el sitio administrativo.

7.2.2.1.1. Cajón de navegación

Este componente se utilizará para mostrar el menú del sitio administrativo. El cajón se accionará desde un botón en la barra de herramientas que mostrará todas las opciones del cajón.

Figura 102. Cajón de navegación, Material-UI



Fuente: elaboración propia, realizado con captura de pantalla.

7.2.2.1.2. Barra de aplicación

La barra de aplicación es un componente que puede contener distintas acciones que permiten navegar e interactuar con el sitio. También pueden brindar información y títulos que corresponden a la sección de la aplicación en la que se encuentran.

Figura 103. Barra de aplicación, Material-UI

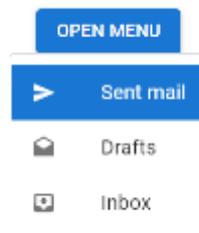


Fuente: elaboración propia, realizado con captura de pantalla.

7.2.2.1.3. Menús simples

Componente que se utilizará en algunas secciones del sitio para contener distintas acciones.

Figura 104. Menús simples, Material-UI

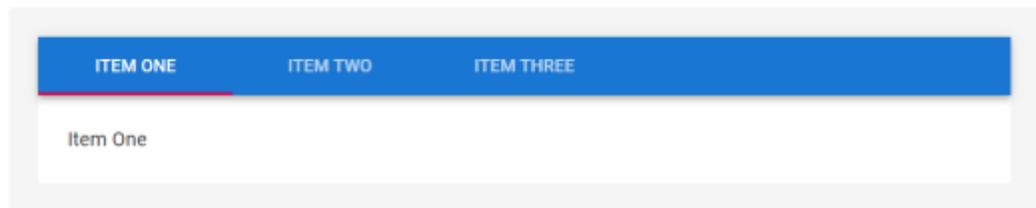


Fuente: elaboración propia, realizado con captura de pantalla.

7.2.2.1.4. Pestañas

Este componente ayudará a separar por grupos los resultados de algunas listas de datos que se utilizarán.

Figura 105. Pestañas, Material-UI



Fuente: elaboración propia, realizado con captura de pantalla.

7.2.2.1.5. Tarjetas

Son superficies que ayudarán a mostrar información y a realizar acciones sobre la información que muestra.

Figura 106. Tarjetas, Material-UI



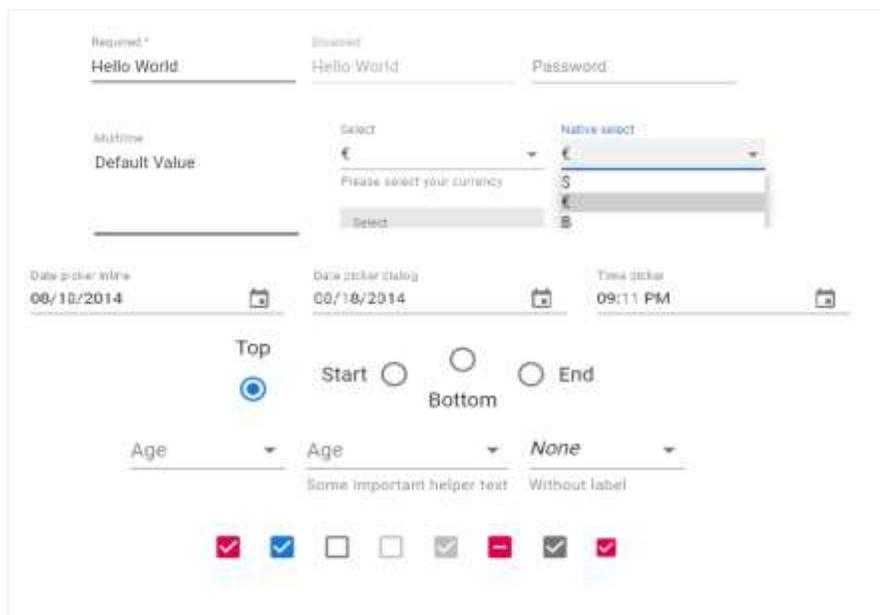
Fuente: elaboración propia, realizado con captura de pantalla.

7.2.2.2. Componentes de interacción y formularios

Dentro del sitio administrativo, se utilizarán varios componentes para capturar datos e información ingresada por el usuario.

Dentro de estos elementos se tiene entrada de texto, selector de fecha, casilla de verificación, elemento de opciones y elemento de selección.

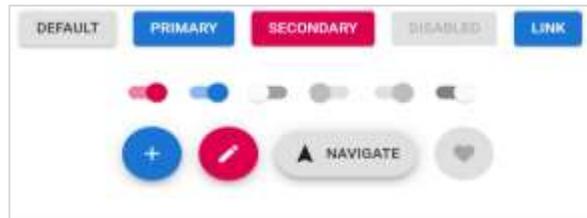
Figura 107. Componentes de formularios, Material-UI



Fuente: elaboración propia, realizado con captura de pantalla.

También se emplearán componentes que realizarán acciones dentro del sitio. De esta manera el usuario tendrá interacción con el sistema. Dentro de estos elementos se tienen botones, botones flotantes, e interruptores.

Figura 108. Botones, Material-UI

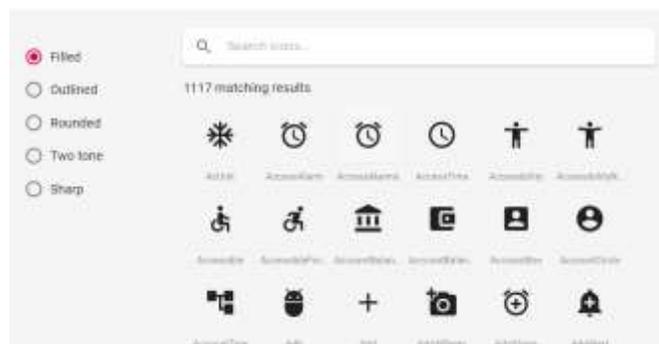


Fuente: elaboración propia, realizado con captura de pantalla.

7.2.2.3. Iconos, alertas y notificaciones

Para que el sitio administrativo sea amigable al usuario, se hará uso de diferentes iconos de Material-UI disponibles para identificar mejor las acciones que se pueden realizar dentro del sitio.

Figura 109. Iconos, Material-UI



Fuente: elaboración propia, realizado con captura de pantalla.

Las notificaciones se podrán visualizar con el componente llamado Snackbar. Este componente puede contener texto y en ocasiones una acción. Se utilizará para que el usuario sepa que acción acaba de ocurrir en el sistema.

Figura 110. **Notificaciones, Material-UI**

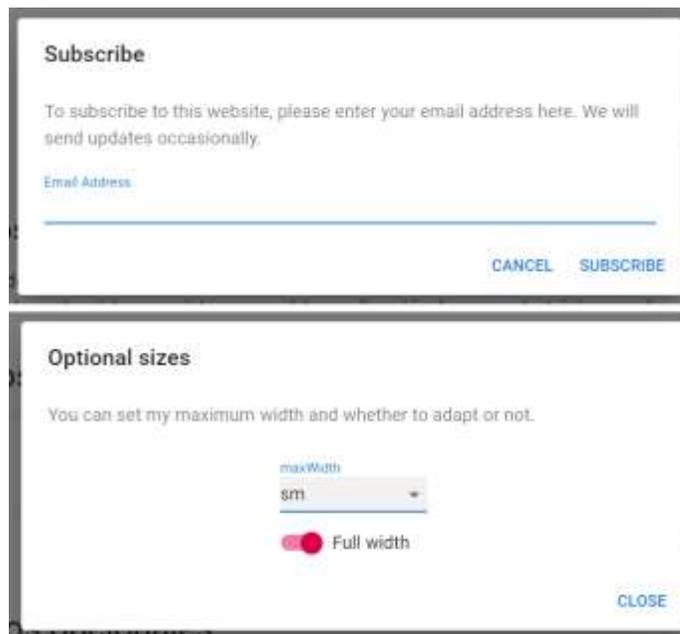


Fuente: elaboración propia, realizado con captura de pantalla.

Los diálogos y alertas son muy importantes dentro de la aplicación. Este componente ayudará a tener la validación del usuario ante una acción a realizar.

A diferencia de las notificaciones, los diálogos y alertas se utilizarán previo a una acción significativa y pueden contener otros componentes dentro de ellos.

Figura 111. **Diálogos, Material-UI**



Fuente: elaboración propia, realizado con captura de pantalla.

8. PUBLICACIÓN DE APLICACIONES

Antes de realizar el despliegue de la aplicación, se debe crear el proyecto en *Google App Engine*, de GCP. La creación del proyecto se detalla en el apartado 4.2 de este documento. A continuación, se detalla el proceso de despliegue para la aplicación del *backend* y *frontend* en Google App Engine. Además, se muestra como se realizó la generación del APK para Android.

8.1. Procedimiento para la publicación del *backend*

Los pasos para realizar la publicación son los siguientes:

- Si se están realizando pruebas en Windows, se debe cambiar la ruta en el archivo de la llave de firebase en `utils/firebase_auth.py`, para que quede de la siguiente manera: `/elfaro-backend-firebase-adminsdk-key.json`.

Figura 112. Cambio de ruta para llave de firebase

```
200915168tesa-backend > main > utils > firebase_auth.py > _
1 import firebase_admin
2 import pathlib
3 from firebase_admin import firestore, credentials
4
5 # firebase auth
6 certificate_path = str(pathlib.Path(__file__).parent.resolve())+"/elfaro-backend-firebase-adminsdk-key.json"
7 print (certificate_path)
8 cred = credentials.Certificate(certificate_path)
9 firebase_admin.initialize_app(cred)
10 fstoredb = firestore.client()
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Revisar que exista el archivo app.yaml. En caso contrario, crear el archivo con el siguiente contenido:

Figura 113. Creación de archivo “app.yaml”

```
200915168tesis-backend > main > ! app.yaml
1 runtime: python39|
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Revisar que se encuentren todas las dependencias utilizadas en el archivo requirements.txt.

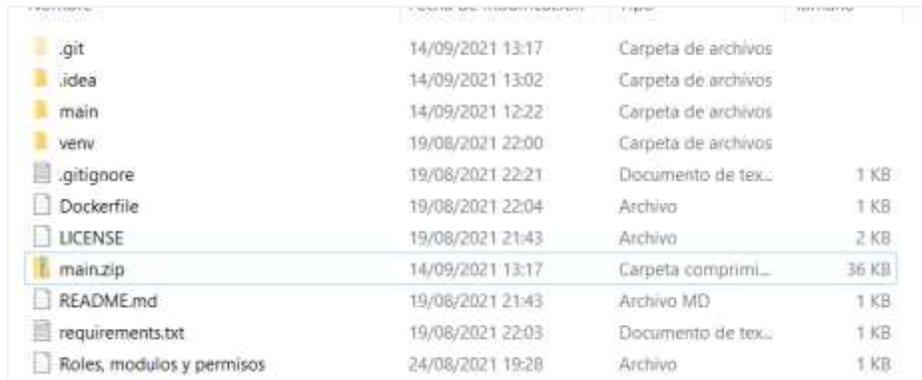
Figura 114. Revisión de requerimientos de Flask

```
200915168tesis-backend > main > ≡ requirements.txt
1 Flask==1.1.2
2 Flask-Cors==3.0.9
3 PyJWT~=2.1.0
4 firebase-admin==5.0.2
5 pymongo[srv]==3.12.0
6 Flask-PyMongo==2.3.0
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Comprimir la carpeta main, que contiene el código del proyecto para subir a la consola de GCP.

Figura 115. Comprimir carpeta del código del proyecto

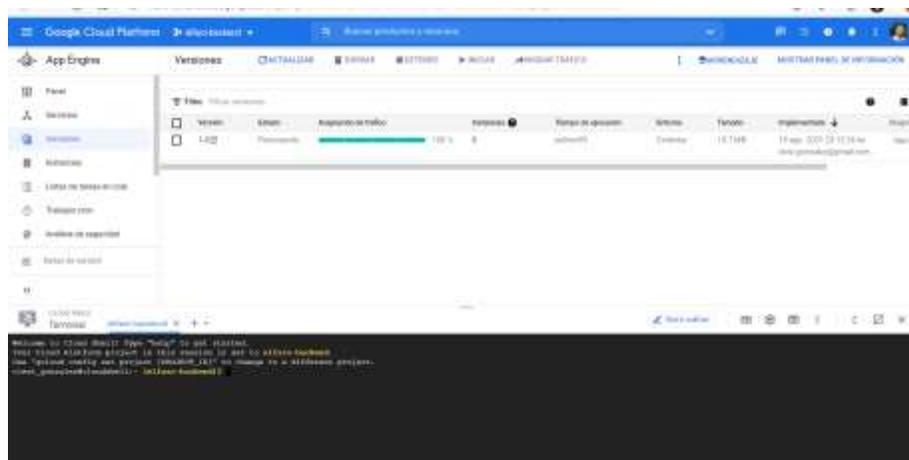


Nombre	Fecha de modificación	Tipo	Tamaño
.git	14/09/2021 13:17	Carpeta de archivos	
.idea	14/09/2021 13:02	Carpeta de archivos	
main	14/09/2021 12:22	Carpeta de archivos	
veriv	19/08/2021 22:00	Carpeta de archivos	
.gitignore	19/08/2021 22:21	Documento de texto	1 KB
Dockerfile	19/08/2021 22:04	Archivo	1 KB
LICENSE	19/08/2021 21:43	Archivo	2 KB
main.zip	14/09/2021 13:17	Carpeta comprimida	36 KB
README.md	19/08/2021 21:43	Archivo MD	1 KB
requirements.txt	19/08/2021 22:03	Documento de texto	1 KB
Roles, módulos y permisos	24/08/2021 19:28	Archivo	1 KB

Fuente: elaboración propia, realizado con captura de pantalla.

- Abrir la consola en línea de Google cloud platform dentro del proyecto del *backend*.

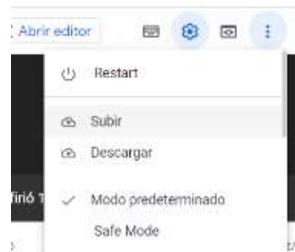
Figura 116. Consola de GCP dentro del proyecto



Fuente: elaboración propia, realizado con captura de pantalla.

- Dirigirse a las opciones de la consola, y presionar el botón Subir.

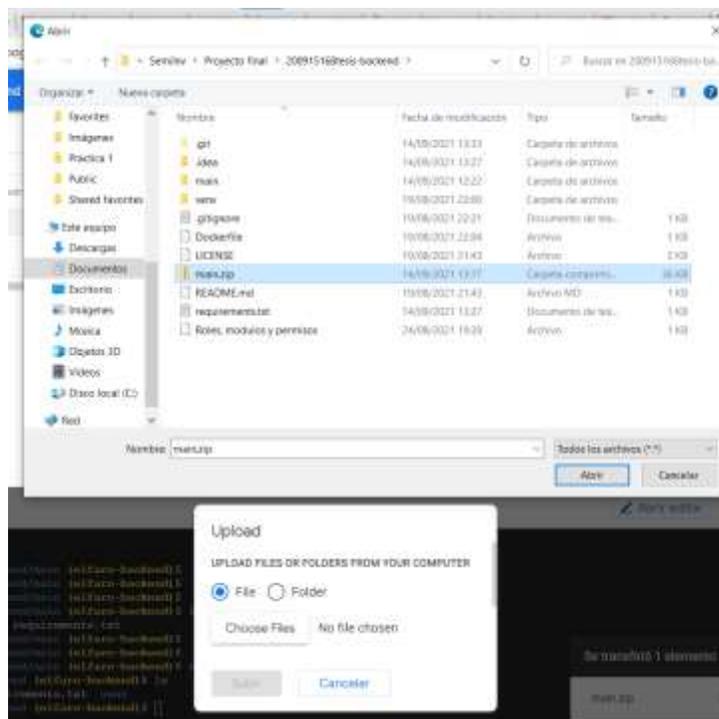
Figura 117. Opción “Subir” de consola de GCP



Fuente: elaboración propia, realizado con captura de pantalla.

- Seleccionar la carpeta comprimida del código que se llama main.zip.

Figura 118. Subir carpeta de código a consola de GCP



Fuente: elaboración propia, realizado con captura de pantalla.

- Dentro de la consola de GCP, descomprimir la carpeta que se sube (main.zip), e ingresar a ella.
 - `unzip main.zip`
 - `cd main/`
- Seleccionar el proyecto que se creó, para el despliegue con el comando. En este caso el nombre del proyecto es elfaro-backend:
 - `gcloud config set project elfaro-backend`
- Realizar despliegue de aplicación, indicando el número de versión con el siguiente comando:
 - `gcloud app deploy -v 1-0`

Figura 119. Despliegue de aplicación *backend*

Version	Status	Progress de subida	Instancias	Time to provision	Status	Size	Deployed
v1.0	Running	100%	2	2024-01-15	Running	10.1 MB	15 ago. 2021 15:08:04 by user@project.com
v1.1	Running	0%	1	2024-01-15	Running	10.1 MB	14 ago. 2021 20:29:36 by user@project.com

Fuente: elaboración propia, realizado con captura de pantalla.

- Ruta del *backend*: <https://elfaro-backend.uc.r.appspot.com>

8.2. Procedimiento para la publicación del *frontend*

Los pasos para realizar la publicación son los siguientes:

- Revisar que la ruta en el archivo `src/utills/db.js` apunte a la versión del *backend* desplegada en el primer punto:

Figura 120. Ruta de *backend* en `db.js`

```
20091516@tesis-frontend > src > utills > db.js > ...
1  const SERVER = "https://eifano-backend.uc.r.appspot.com";
2  |
3  export const SUCCESS = 200;
4  export const BAD_REQUEST = 400;
5  export const SERVER_ERROR = 500;
6
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Comprimir todos los archivos y carpetas del proyecto, exceptuando la carpeta `/node_modules`, y el archivo `git`.

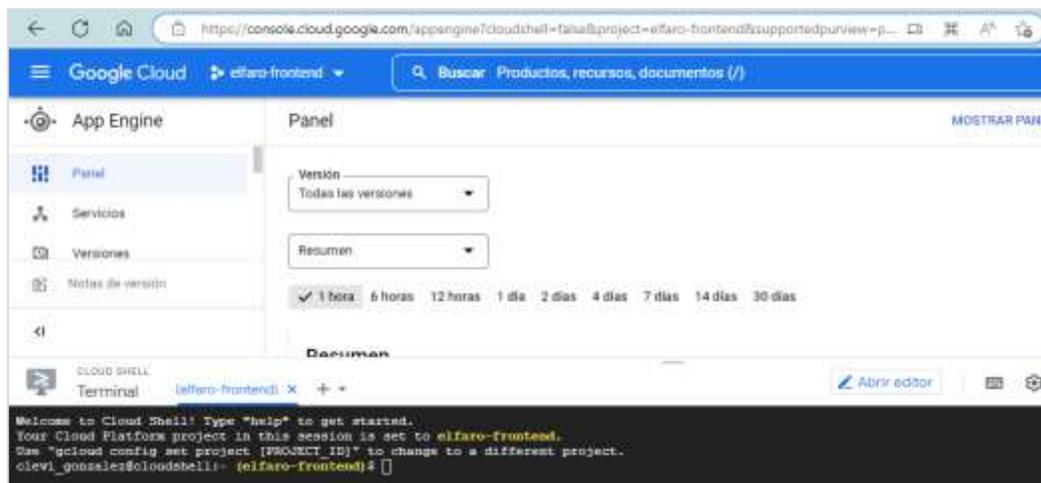
Figura 121. Comprimir carpeta del código del proyecto

Nombre	Fecha de modificación	Tipo	Tamaño
.git	2/11/2022 12:26	Carpeta de archivos	
node_modules	22/10/2022 16:43	Carpeta de archivos	
public	10/09/2022 19:33	Carpeta de archivos	
src	10/09/2022 19:33	Carpeta de archivos	
eslintrc	10/09/2022 19:33	Archivo ESLINTRC	1 KB
gitignore	10/09/2022 19:33	Archivo de origen	1 KB
prettierrc	10/09/2022 19:33	Archivo PRETTIER	1 KB
CHANGELOG.md	10/09/2022 19:33	Archivo de origen	1 KB
jsconfig.json	10/09/2022 19:33	Archivo de origen	1 KB
LICENSE.md	10/09/2022 19:33	Archivo de origen	2 KB
my-app.zip	2/11/2022 14:03	Carpeta comprimi...	1,024 KB
package.json	22/10/2022 16:43	Archivo de origen	3 KB
package-lock.json	22/10/2022 16:43	Archivo de origen	1,833 KB
README.md	10/09/2022 19:33	Archivo de origen	3 KB
yarn.lock	22/10/2022 16:43	Archivo LOCK	584 KB

Fuente: elaboración propia, realizado con captura de pantalla.

- Abrir la consola en línea de Google cloud platform dentro del proyecto del *backend*.

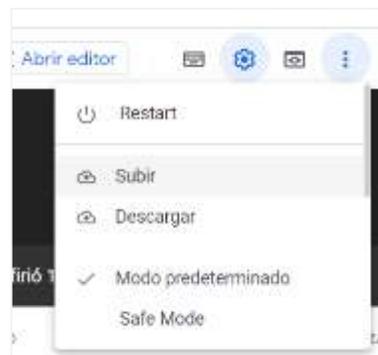
Figura 122. Consola de GCP dentro del proyecto



Fuente: elaboración propia, realizado con captura de pantalla.

- Dirigirse a las opciones de la consola, y presionar el botón Subir

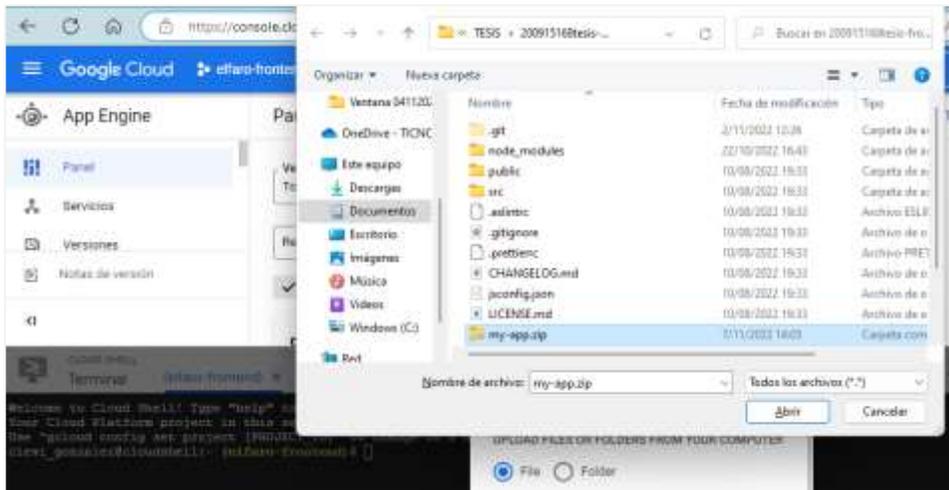
Figura 123. Opción “Subir” de consola de GCP



Fuente: elaboración propia, realizado con captura de pantalla.

- Seleccionar la carpeta comprimida del código que se llama my-app.zip.

Figura 124. Subir carpeta de código a consola de GCP



Fuente: elaboración propia, realizado con captura de pantalla.

- Dentro de la consola de GCP, descomprimir la carpeta que se sube (my-app.zip), e ingresar a ella.
 - `unzip my-app.zip`
 - `cd my-app/`
- Seleccionar el proyecto que se creó para el despliegue con el comando. En este caso el nombre del proyecto es elfaro-frontend:
 - `gcloud config set project elfaro-frontend`
- Instalar dependencias de nodeJS con el comando `npm install`.

Tabla LXV. Archivo de despliegue *frontend*

```
runtime: nodejs14
handlers:
# Serve all static files with url ending with a file extension
- url: /\.(.+)$
  static_files: build/1
  upload: build/(.+)$
# Catch all handler to index.html
- url: /*
  static_files: build/index.html
  upload: build/index.html
```

Fuente: elaboración propia, realizado con Word.

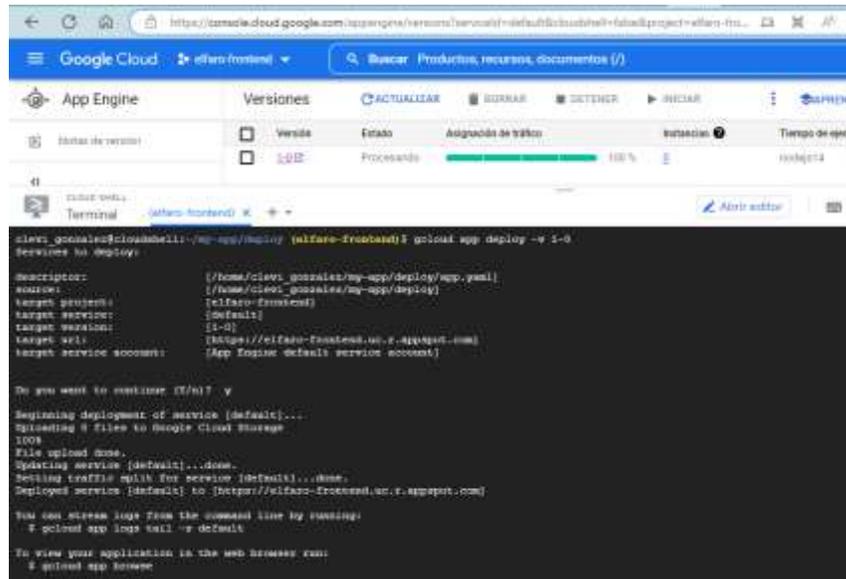
Figura 127. Creación de carpeta de despliegue

```
clevi_gonzalez@cloudshell:~/my-app (elfaro-frontend)$
clevi_gonzalez@cloudshell:~/my-app (elfaro-frontend)$ cd deploy/
clevi_gonzalez@cloudshell:~/my-app/deploy (elfaro-frontend)$ ls
app.yaml  build
clevi_gonzalez@cloudshell:~/my-app/deploy (elfaro-frontend)$
```

Fuente: elaboración propia, realizado con captura de pantalla.

- Realizar despliegue de aplicación, indicando el número de versión con el siguiente comando:
 - `gcloud app deploy -v 1-0`

Figura 128. Despliegue de aplicación *frontend*



Fuente: elaboración propia, realizado con captura de pantalla.

- Ruta de *frontend*: <https://elfaro-frontend.uc.r.appspot.com>

8.3. Publicación de aplicación Android en google drive

Los pasos para generar el APK y subirlo a Google drive son los siguientes:

- Ejecutar el comando `npx cap open android` para abrir el proyecto en Android studio.

Figura 129. Abrir el proyecto en Android Studio

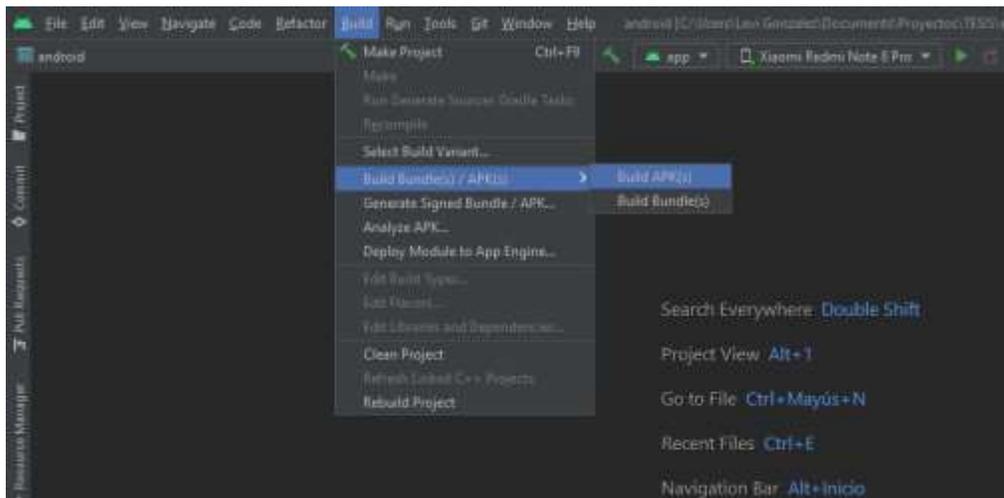
```
PS C:\Users\Levi_Gonzalez\Documents\Proyectos\TESIS> cd .\e1-faro-app\
PS C:\Users\Levi_Gonzalez\Documents\Proyectos\TESIS\e1-faro-app> npx cap open android
[info] Opening Android project at: android.
PS C:\Users\Levi_Gonzalez\Documents\Proyectos\TESIS\e1-faro-app> |
```

Ln 2, Col 1 Spaces

Fuente: elaboración propia, realizado con captura de pantalla.

- Seleccionar en el menú Build, luego Build Bundle(s) / APK(s).

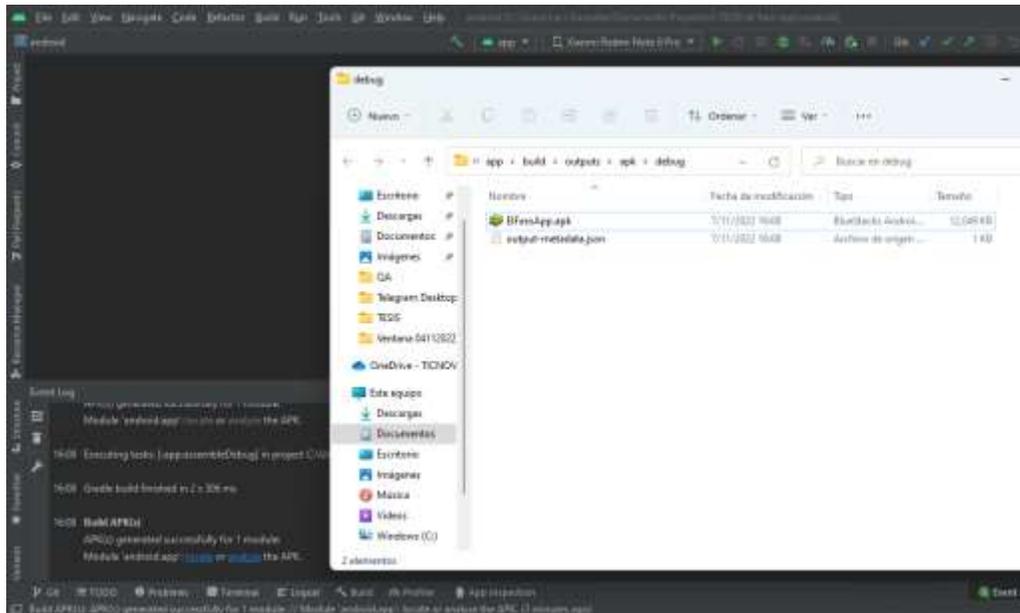
Figura 130. Generación de APK



Fuente: elaboración propia, realizado con captura de pantalla.

- Abrir ubicación de archivo haciendo clic en Locate, cuando el proyecto termine de generar el APK. Renombrar el APK con el nombre EIFaroApp.apk.

Figura 131. Localizar APK generada



Fuente: elaboración propia, realizado con captura de pantalla.

- Crear carpeta en Google Drive y subir el archivo EIFaroApp.apk. Tomar en cuenta que se deben cambiar los permisos para que cualquiera con el enlace pueda acceder a la aplicación.
- Ruta de APK:
 - https://drive.google.com/file/d/1GBhoSFPOOZy_RsqaqVEA-7i8e2ARokRr/view?usp=share_link

9. PRUEBAS DE SITIO ADMINISTRATIVO Y APLICACIÓN MÓVIL

Se realizaron pruebas de todas las funcionalidades que se contemplaron en los casos de uso. Todos los resultados esperados de las pruebas se obtuvieron con éxito. En cada una de las tablas se muestran los siguientes elementos:

- Tipo de prueba: nombre descriptivo de la prueba realizada. La prueba puede ser para cumplir la función o para obtener un mensaje de error, indicando el fallo a evaluar.
- Descripción: descripción de la prueba realizada. Se detallan las condiciones que deben cumplirse para realizar la prueba.
- Resultado: resultado de la prueba exitosa. Se recibe como respuesta un mensaje indicando el éxito de la operación, o el error detallado de la prueba de error.

9.1. Pruebas de sitio administrativo

Se muestran los resultados de cada una de las pruebas realizadas en el sitio administrativo.

Tabla LXVI. Pruebas de token de autorización

Tipo de prueba	Descripción	Resultado
Obtener token de autorización	Se obtiene el token de autorización al proporcionar el client_id y client_secret correcto.	Se retorna token de autorización.
Error en la verificación	Error por client_id o client_secret inválidos.	Se devuelve mensaje de error en la verificación
Error campos obligatorios	No se enviaron los campos obligatorios.	Se devuelve mensaje de error por campos obligatorios

Fuente: elaboración propia, realizado con Word.

Tabla LXVII. Pruebas de Inicio y fin de sesión

Tipo de prueba	Descripción	Resultado
Inicio de sesión	Se ingresa usuario y contraseña para iniciar sesión, luego se presiona el botón de inicio de sesión.	Inicio de sesión en aplicación móvil, mostrando lista de eventos.
Error en datos de inicio de sesión	Error en datos de inicio de sesión. Usuario o contraseña incorrectos.	Se muestra mensaje indicando que el usuario o contraseña son incorrectos.
Fin de sesión	Se presiona el botón para salir de sesión.	Se finaliza la sesión y se redirige a la pantalla inicial.
Recuperación de contraseña	Seleccionar recuperación de contraseña. Proporcionar correo electrónico para recuperar contraseña.	Se recibe correo electrónico indicando las instrucciones para recuperar contraseña.
Fallo en recuperacion de contraseña	Seleccionar recuperación de contraseña. Proporcionar correo electrónico que no este registrado para recuperar contraseña.	Se indica que hubo un fallo en la recuperación de contraseña para los datos proporcionados.

Fuente: elaboración propia, realizado con Word.

Tabla LXVIII. Pruebas de ciudad

Tipo de prueba	Descripción	Resultado
Creación de Ciudad	Creación de Ciudad con todos los campos necesarios para la creación	Creación de un nuevo Ciudad
Error en la creacion de Ciudad x campos obligatorios	Error en creación de Ciudad al no tener los campos obligatorios	Devolver mensaje de error "Faltan campos obligatorios".
Eliminar Ciudad	Eliminaciónde Ciudad con todos los campos necesarios para la eliminación	Eliminacion de Ciudad previamente creado.
Error al eliminar Ciudad x campos obligatorios	Error en eliminación de Ciudad al no tener los campos obligatorios	Devolver mensaje de error "Faltan campos obligatorios"
Listar Ciudades	Listar Ciudades creados en base de datos.	La respuesta muestra la lista de Ciudadess
Modificar Ciudad	Modificación de Ciudad con todos los campos necesarios para la modificación	Modificación de un Ciudad previamente creado en el sistema.
Error al modificar Ciudad x campos obligatorios	Error en modificación de Ciudad al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"

Fuente: elaboración propia, realizado con Word.

Tabla LXIX. Pruebas de lugar

Tipo de prueba	Descripción	Resultado
Creación de Lugar	Creación de Lugar con todos los campos necesarios para la creación.	Creación de un nuevo Lugar
Error en la creacion de Lugar x campos obligatorios	Error en creación de Lugar al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Eliminar Lugar	Eliminaciónde Lugar con todos los campos necesarios para la eliminación	Eliminacion de Lugar previamente creado.
Error al eliminar Lugar x campos obligatorios	Error en eliminación de Lugar al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios".
Listar Lugares	Listar Lugares creados en base de datos.	La respuesta muestra la lista de Lugares

Continuación de la tabla LXIX.

Modificar Lugar	Modificación de Lugar con todos los campos necesarios para la modificación	Modificación de un Lugar previamente creado en el sistema.
Error al Modificar Lugar x campos obligatorios	Error en modificación de Lugar al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"

Fuente: elaboración propia, realizado con Word.

Tabla LXX. Pruebas de roles

Tipo de prueba	Descripción	Resultado
Creación de Rol	Creación de Rol con todos los campos necesarios para la creación	Creación de un nuevo rol.
Error en la creación de rol x campos obligatorios	Error en creación de rol al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Eliminar Rol	Eliminación de rol con todos los campos necesarios para la eliminación	Eliminación de rol previamente creado.
Error al eliminar rol x campos obligatorios	Error en eliminación de rol al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Listar Roles	Listar roles creados en base de datos.	La respuesta muestra la lista de roles
Modificar Rol	Modificación de rol con todos los campos necesarios para la modificación	Modificación de un rol previamente creado en el sistema.
Error al modificar rol x campos obligatorios	Error en modificación de rol al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"

Fuente: elaboración propia, realizado con Word.

Tabla LXXI. Pruebas de usuarios

Tipo de prueba	Descripción	Resultado
Creación de Usuario	Creación de usuario con todos los campos necesarios para la creación.	Creación de un nuevo usuario

Continuación de la tabla LXXI.

Error en la creacion de usuario x campos obligatorios	Error en creación de usuario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Eliminar Usuario	Eliminación de usuario con todos los campos necesarios para la eliminación	Eliminación de usuario previamente creado.
Error al eliminar usuario x campos obligatorios	Error en eliminación de usuario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios".
Listar Usuarios	Listar usuarios creados en base de datos.	La respuesta muestra la lista de usuarios
Modificar Usuario	Modificación de usuario con todos los campos necesarios para la modificación	Modificación de un usuario previamente creado en el sistema.
Error al Modificar usuario x campos obligatorios	Error en modificación de usuario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"

Fuente: elaboración propia, realizado con Word.

Tabla LXXII. **Pruebas de formularios**

Tipo de prueba	Descripción	Resultado
Creación de formulario	Creación de formulario con todos los campos necesarios para la creación.	Creación de un nuevo formulario
Error en la creacion de formulario x campos obligatorios	Error en creación de formulario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Eliminar formulario	Eliminación de formulario con todos los campos necesarios para la eliminación	Eliminación de formulario previamente creado.
Error al eliminar formulario x campos obligatorios	Error en eliminación de formulario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Listar formularios	Listar formularios creados en base de datos.	La respuesta muestra la lista de formularios
Modificar formulario	Modificación de formulario con todos los campos necesarios para la modificación	Modificación de un formulario previamente creado en el sistema.

Continuación de la tabla LXXII.

Error al modificar formulario x campos obligatorios	Error en modificación de formulario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
---	---	---

Fuente: elaboración propia, realizado con Word.

Tabla LXXIII. **Pruebas de eventos**

Tipo de prueba	Descripción	Resultado
Creación de evento	Creación de evento con todos los campos necesarios para la creación.	Creación de un nuevo evento
Error en la creación de evento x campos obligatorios	Error en creación de evento al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Eliminar evento	Eliminación de evento con todos los campos necesarios para la eliminación	Eliminación de evento previamente creado.
Error al eliminar evento: no se puede eliminar evento con registros.	Error en eliminación de evento. El evento no se puede eliminar cuando tiene registros asociados.	Se devuelve mensaje de error "No se puede eliminar evento con registros."
Error al eliminar evento x campos obligatorios	Error en eliminación de evento al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Listar eventos	Listar eventos creados en base de datos.	La respuesta muestra la lista de eventos
Modificar evento	Modificación de evento con todos los campos necesarios para la modificación	Modificación de un evento previamente creado en el sistema.
Error al modificar evento: no se puede actualizar evento con registros.	Error en modificación de evento. El evento no se puede modificar cuando tiene registros asociados.	Se devuelve mensaje de error "No se puede actualizar evento con registros."
Error al modificar evento x campos obligatorios	Error en modificación de evento al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Listar registros de evento	Listar registros de evento creados en base de datos.	La respuesta muestra la lista de registro de eventos
Listar registros de evento (evento sin registros)	Listar registros de evento creados en base de datos.	No se muestran registros para el evento seleccionado.

Fuente: elaboración propia, realizado con Word.

Tabla LXXIV. Pruebas de asignación de formulario

Tipo de prueba	Descripción	Resultado
Asignación de formulario a evento	Se busca un formulario en la lista para asignar al evento pendiente seleccionado.	Crear asignación de formulario a evento. El formulario se asocia al evento.
Error en asignación de formulario a evento	Se busca un formulario en la lista para asignar al evento en curso o caducado seleccionado.	Se devuelve mensaje de error "No se puede asignar un formulario a un evento en curso o caducado."

Fuente: elaboración propia, realizado con Word.

Tabla LXXV. Pruebas de beneficiarios

Tipo de prueba	Descripción	Resultado
Creación de beneficiario	Creación de beneficiario con todos los campos necesarios para la creación.	Creación de un nuevo beneficiario
Error en la creación de beneficiario x campos obligatorios	Error en creación de beneficiario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Ingreso de homónimo	Se intenta crear un usuario con el mismo DPI en el sistema, y se genera un error en el servidor.	Se devuelve mensaje de error: [ERROR] El Beneficiario ya fue ingresado en la base de datos.
Ingreso de homónimo	Se intenta crear un usuario con el mismo nombre, apellido y fecha de nacimiento.	Se devuelve mensaje de error: [ERROR] El Beneficiario ya fue ingresado en la base de datos.
Eliminar beneficiario	Eliminación de beneficiario con todos los campos necesarios para la eliminación	Eliminación de beneficiario previamente creado. Revisar que el beneficiario exista antes y después de la modificación
Error al eliminar beneficiario: tiene registros de evento	Error en eliminación de beneficiario al tener registros de evento asociados.	Se devuelve mensaje de error " El beneficiario tiene registros de evento"
Error al eliminar beneficiario x campos obligatorios	Error en eliminación de beneficiario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
Listar beneficiarios	Listar beneficiarios creados en base de datos.	La respuesta muestra la lista de beneficiarios
Modificar beneficiario	Modificación de beneficiario con todos los campos necesarios para la modificación	Modificación de un beneficiario previamente creado en el sistema.

Continuación de la tabla LXXV.

Error al modificar beneficiario x campos obligatorios	Error en modificación de beneficiario al no tener los campos obligatorios	Se devuelve mensaje de error "Faltan campos obligatorios"
---	---	---

Fuente: elaboración propia, realizado con Word.

Tabla LXXVI. Pruebas de reportes y descargas

Tipo de prueba	Descripcion	Resultado
Filtro y descarga de beneficiarios	Se utilizan los filtros de beneficiario y luego de realizar la búsqueda se presiona el botón de descarga.	Descarga de reporte de beneficiarios en formato .csv
Filtro y descarga de eventos	Se utilizan los filtros de eventos y luego de realizar la búsqueda se presiona el botón de descarga.	Descarga de reporte de eventos en formato .csv
Vista de resumen de actividad	Dirigirse a la pagina principal o al modulo de "Resumen de Actividad"	Se muestra la vista del resumen de actividad con los 7 reportes disponibles
Descarga de beneficiarios atendidos en el mes	En el resumen de actividad, presionar el botón de descarga de beneficiarios atendidos en el mes, y luego el botón descargar, sin modificar los filtros.	Descarga de reporte de beneficiarios atendidos en el mes en formato .csv
Descarga de lugares atendidos en el mes	En el resumen de actividad, presionar el botón de descarga de lugares atendidos en el mes, y luego el botón descargar, sin modificar los filtros.	Descarga de reporte de lugares atendidos en el mes en formato .csv
Descarga de eventos terminados en el mes	En el resumen de actividad, presionar el botón de descarga de eventos terminados en el mes, y luego el botón descargar, sin modificar los filtros.	Descarga de reporte de eventos terminados en el mes en formato .csv
Descarga de eventos en curso en el mes	En el resumen de actividad, presionar el botón de descarga de eventos en curso en el mes.	Descarga de reporte de eventos en curso en el mes en formato .csv
Descarga de eventos pendientes en el mes	En el resumen de actividad, presionar el botón de descarga de eventos pendientes en el mes, y luego el botón descargar, sin modificar los filtros.	Descarga de reporte de eventos pendientes en el mes en formato .csv

Continuación de la tabla LXXVI.

Descarga de personas atendidas en el año por tipo de evento	En el resumen de actividad, presionar el botón de descarga de personas atendidas en el año por tipo de evento, y luego el botón descargar, sin modificar los filtros.	Descarga de reporte de personas atendidas en el año por tipo de evento, en formato .csv
Descarga de porcentaje de personas atendidas por tipo de evento.	En el resumen de actividad, presionar el botón de descarga de porcentaje de personas atendidas por tipo de evento en el mes, y luego el botón descargar, sin modificar los filtros.	Descarga de reporte de porcentaje de personas atendidas por tipo de evento, en formato .csv

Fuente: elaboración propia, realizado con Word.

9.2. Pruebas de aplicación móvil

A continuación, se muestran los resultados de cada una de las pruebas realizadas en la aplicación móvil.

Tabla LXXVII. **Pruebas de token de autorización**

Tipo de prueba	Descripción	Resultado
Obtener token de autorización	Se obtiene el token de autorización al proporcionar el client_id y client_secret correcto.	Se retorna token de autorización.
Error en la verificación	Error por client_id o client_secret inválidos.	Se devuelve mensaje de error en la verificación
Error campos obligatorios	No se enviaron los campos obligatorios.	Se devuelve mensaje de error por campos obligatorios

Fuente: elaboración propia, realizado con Word.

Tabla LXXVIII. Pruebas de inicio y fin de sesión

Tipo de prueba	Descripción	Resultado
Inicio de sesión	Se ingresa usuario y contraseña para iniciar sesión, luego se presiona el botón de inicio de sesión.	Inicio de sesión en aplicación móvil, mostrando lista de eventos.
Error en datos de inicio de sesión	Error en datos de inicio de sesión. Usuario o contraseña incorrectos.	Se muestra mensaje indicando que el usuario o contraseña son incorrectos.
Fin de sesión	Se presiona el botón para salir de sesión.	Se finaliza la sesión y se redirige a la pantalla inicial de la aplicación web.

Fuente: elaboración propia, realizado con Word.

Tabla LXXIX. Pruebas de eventos y registros

Tipo de prueba	Descripción	Resultado
Listar registro de eventos	Hacer login en la aplicación para ver la pantalla principal, o ingresar en el menú de eventos.	Se listaron todos los eventos que existen en curso o pendientes.
Creación de registro de evento	Se selecciona un evento, luego un beneficiario de la lista. Se llenan los datos del formulario y se presiona el botón registrar.	Creación de un nuevo registro de evento registro libre
Creación de registro de evento sin formulario	Se selecciona un evento, luego un beneficiario de la lista y se presiona el botón registrar.	Creación de un nuevo registro de evento registro único
2da creación de registro de evento para beneficiario (Registro único)	Se selecciona un evento, luego se busca a un beneficiario que ya haya sido registrado.	El beneficiario está bloqueado porque el evento es de registro único.
Registro de evento con cupo (cupos disponibles)	Se selecciona un evento. En la pantalla de conteo de registros dice cuántos cupos hay disponibles. Si hay cupos disponibles, se debe seleccionar un beneficiario de la lista y se presiona el botón registrar.	Se registra el beneficiario al evento, y se ve que el contador de la pantalla ha sumado 1.

Continuación de la tabla LXXIX.

Registro de evento con cupo (cupos llenos)	Se selecciona un evento. En la pantalla de conteo de registros dice cuantos cupos hay disponibles.	La pantalla muestra que ya no hay cupos disponibles y no se pueden registrar más beneficiarios.
--	--	---

Fuente: elaboración propia, realizado con Word.

Tabla LXXX. Pruebas de actualización y sincronización

Tipo de prueba	Descripción	Resultado
Actualización de datos de aplicación	Sincronización de datos para la aplicación. Se debe reunir el conjunto de datos que se enviará a la aplicación móvil para que esta funcione con datos actualizados.	Listado de datos que contiene: Usuarios, lugares, formularios, campos formularios, eventos, asignación evento-formulario, beneficiarios.
Sincronización de datos (beneficiarios y registros de eventos)	Sincronización de beneficiarios y registros de eventos. El <i>backend</i> recibe un conjunto de beneficiarios nuevos. También recibe el conjunto de registros que se tomaron por cada evento, los cuales se almacenarán en las colecciones de registro-evento y registro-formulario	Los datos se crean y actualizan correctamente en la base de datos. Se muestra un mensaje que se ha realizado la sincronización y un detalle de los datos sincronizados.
Sincronización de datos (datos ya sincronizados).	Cuando se presiona el botón de sincronización de datos, se verifica que haya datos pendientes de sincronizar.	Se muestra un mensaje diciendo que la sincronización está al día.

Fuente: elaboración propia, realizado con Word.

10. RESULTADOS DE IMPLEMENTACIÓN

Se muestran los puntos más importantes planteados en el marco teórico, respecto a las necesidades de la fundación Misión El Faro, y la propuesta de solución planteada.

El procedimiento de registro de beneficiarios en eventos y la captura de datos de beneficiarios fue automatizado con la implementación del sitio administrativo y la aplicación móvil.

Las deficiencias del sistema de registro de eventos que se ha estado utilizando, fueron solucionadas a través de la implementación de las dos aplicaciones desarrolladas en este proyecto, las cuales permiten reducir el riesgo de pérdida de información y reduce el tiempo de captura y carga de datos de los eventos y beneficiarios. Esto también acelera la reportería para los administradores.

10.1. Necesidades resueltas del sistema

Se solventaron las siguientes necesidades que se realizaban de forma manual:

Tabla LXXXI. **Necesidades que se realizaban de forma manual**

Necesidad atendida	Solución aplicada
Registro de datos personales de cada beneficiario	Se tiene la opción de realizar desde el sitio administrativo o desde la aplicación móvil a través del registro en eventos.

Continuación de la tabla LXXXI.

Registro de información de formularios	Se puede realizar desde la aplicación móvil a través del registro en eventos.
Registro manual de la información obtenida en hojas de cálculo	La aplicación móvil se encarga de sincronizar los datos a la nube, y el sitio administrativo permite descargar la información en formato .csv.

Fuente: elaboración propia, realizado con Word.

En la siguiente tabla se muestran las necesidades identificadas, para las cuales no se tenía un control aplicado:

Tabla LXXXII. **Necesidades identificadas**

Necesidad identificada	Solución aplicada
Proteger la integridad de la información de los datos que se obtienen durante los eventos	Los datos ahora se sincronizan a la base de datos en la nube desde la aplicación. Esto solo se puede realizar por el usuario de la aplicación
Optimizar el tiempo de recuperación de la información obtenida durante los eventos	Ahora la información de los formularios es sincronizada a la nube de forma sencilla desde la aplicación móvil. Esto se refleja automáticamente en el sitio administrativo.
Crear reportes que muestren el resumen de las actividades realizadas por la fundación.	En el sitio administrativo están disponibles los reportes sobre beneficiarios atendidos, lugares atendidos, tipos de eventos que se han llevado a cabo, eventos en el sistema (pendientes, en curso y caducados), entre otros.
Llevar un mejor control en eventos de registro único, para evitar que una persona se registre más de una vez a un evento.	Se implementó el manejo de eventos de registro único, que permite a los usuarios registrar una sola vez a un beneficiario en este tipo de eventos.

Continuación de la tabla LXXXII.

Capturar datos demográficos de las personas que llegan a los eventos realizados por la fundación.	Durante el registro en eventos, los datos que se obtienen sobre los beneficiarios alimentan la base de datos del sistema, permitiendo que se puedan realizar informes más detallados sobre esta información
---	---

Fuente: elaboración propia, realizado con Word.

10.2. Características destacables de las soluciones

La solución implementada brinda una mejor atención a las necesidades de la fundación Misión El Faro, para la captura de datos de eventos y beneficiarios, pero se pueden destacar los siguientes puntos de la propuesta global:

- Sitio administrativo
 - El módulo de usuarios y permisos brinda a la administración control sobre el acceso a la información del sistema.
 - El módulo de formularios permite la creación y reutilización de formularios con distintos tipos de datos, que podrán utilizarse en múltiples eventos.
 - El módulo de beneficiarios provee un espacio donde se centraliza toda la información de los beneficiarios, con filtros de búsqueda específicos y opción para descargar la información filtrada de la base de datos.

- El módulo de eventos provee herramientas para filtrar los eventos por sus diferentes características (tipo, cupo, fechas, entre otros), y la descarga de la información de los eventos y los registros en los eventos con la información de los formularios correspondiente a los beneficiarios.
- La página principal brinda un resumen de la actividad reciente donde se ve reflejado el trabajo realizado en el mes actual y durante el año. También brinda la descarga de la información de la actividad reciente y la modificación de algunos parámetros para la generación de estos reportes.

Figura 132. Sitio administrativo

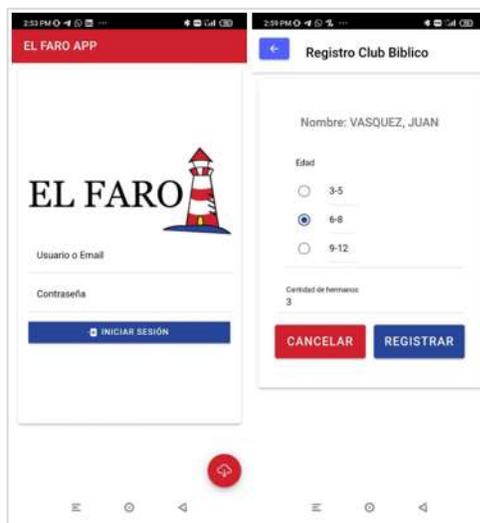


Fuente: elaboración propia, realizado con captura de pantalla.

- Aplicación móvil

- Se brinda una aplicación segura que previene el acceso a la información a través de un inicio de sesión con una cuenta creada en el sitio administrativo.
- Los datos de la aplicación se actualizan al entrar a la aplicación, o a petición del usuario en los botones de actualización de datos de la aplicación. Esto mantiene la información actualizada de los beneficiarios y eventos, agiliza y facilita el registro en eventos.
- La aplicación móvil permite la captura de información de los registros de eventos y beneficiarios de manera *offline*, así como la sincronización de todos los datos capturados. Esta sincronización se hace a petición del usuario, para que lo pueda realizar con una buena conexión a internet.

Figura 133. Aplicación móvil



Fuente: elaboración propia, realizado con captura de pantalla.

10.3. Beneficios de la arquitectura implementada

La arquitectura de la solución permite tener un sistema que cuenta con las siguientes bondades:

- Escalabilidad: el uso de Google App Engine provee un sistema autoescalable, que responde según las necesidades de los usuarios que lo utilicen.
- Flexibilidad: el sistema se adapta a las diferentes necesidades que tiene la fundación Misión El Faro, para el control de registros en eventos y la captura de información de los beneficiarios.
- Alta disponibilidad: se cuenta con la capacidad necesaria en infraestructura y en base de datos para que la solución sea útil a largo plazo.
- Sistema robusto: el sistema ha sido sometido a una serie de pruebas que validan el correcto funcionamiento de todas las funciones que provee, y el manejo de errores que pueden surgir durante su uso.
- Seguridad de la información: se ha resguardado la información del sistema a través del uso de autenticación de usuarios, encriptando contraseñas y utilizando tokens de autenticación.

10.4. Resumen de las tecnologías utilizadas

Para las aplicaciones de *frontend*, las soluciones que se implementaron fueron realizadas con 2 frameworks de Javascript: ReactJS y Ionic. El sitio

administrativo fue desarrollado con ReactJS, junto con Firebase para la autenticación de usuarios y desplegado en Google App Engine. La aplicación móvil fue desarrollada con Ionic junto con Capacitor que facilitó la creación de una aplicación disponible para Android y iOS.

Respecto a las aplicaciones de *backend*, se realizó una solución basada en Python 3. Esta solución también fue desplegada en Google App Engine y es la capa que tiene conexión directa con la base de datos en MongoDB. El *backend* utiliza una base de datos NoSQL en MongoDB, sin embargo, para el manejo *offline* de la información en la aplicación móvil, se utilizó una base de datos SQLite.

10.5. Costos de implementación

Se ha establecido un nivel transaccional para operar sin costo, así como un nivel de operación donde se incurrirá en un pago mensual; De la misma manera se ha indicado a los administradores el nivel transaccional a partir del cual se incurrirá en un costo adicional.

El presupuesto planteado en los capítulos anteriores está basado en la información de la cuenta de Google Cloud Platform que aloja el *frontend* y *backend*, así como la información brindada por MongoDB Atlas, que es donde se aloja la información del sistema.

CONCLUSIONES

1. Se desarrolló una plataforma para facilitar la captura de datos y control de personas atendidas durante las actividades realizadas por la fundación Misión El Faro, y se desarrolló el sitio administrativo para el manejo de todos los datos correspondientes a los eventos.
2. Se utilizaron tecnologías en la nube para almacenar toda la información obtenida de los eventos y beneficiarios, que ahora se encuentra disponible en el sitio administrativo con diferentes opciones para filtrar y descargar la información.
3. Se creó un resumen de la actividad reciente en el sitio administrativo donde se puede visualizar el trabajo realizado por la fundación Misión El Faro, y se incluyó la descarga de esta información para poder utilizarse como reportes
4. Se desarrollo una aplicación móvil que funciona *offline*, de modo que puede utilizarse sin conexión y posteriormente sincronizar los datos recopilados de los registros en eventos y beneficiarios.
5. Se utilizaron protocolos de seguridad en las aplicaciones para las transferencias de información y peticiones al servidor, además se encriptaron las contraseñas de los usuarios.

RECOMENDACIONES

1. Monitorear el funcionamiento del aplicativo en general. Esto puede ser realizado desde el log del despliegue en la nube (del *frontend* y *backend*), y también los logs que se almacenan en el sistema.
2. Mejorar los siguientes aspectos de la aplicación en general: agregar otros tipos de campos a la creación de formularios, agregar fotografía de beneficiario en la creación de beneficiarios, recuperación de contraseña desde aplicación móvil y búsqueda rápida de beneficiarios y eventos en sitio administrativo.
3. Complementar los desarrollos que quedaron pendientes por el alcance delimitado de la solución, por ejemplo: un módulo específico para reportes, un módulo de ajustes del sistema, el manejo de secciones en los formularios y asignación de múltiples formularios a un evento.
4. Crear un presupuesto para el control de gastos mensuales del proyecto, dentro de la plataforma de Google Cloud Platform, en el módulo de facturación. Se recomienda realizar esta acción para llevar el control del consumo de los gastos realizados por las aplicaciones desplegadas en la nube.
5. Plantear una estrategia para que se puedan realizar análisis y transformación de datos, utilizando la información recopilada con el sitio administrativo y la aplicación móvil.

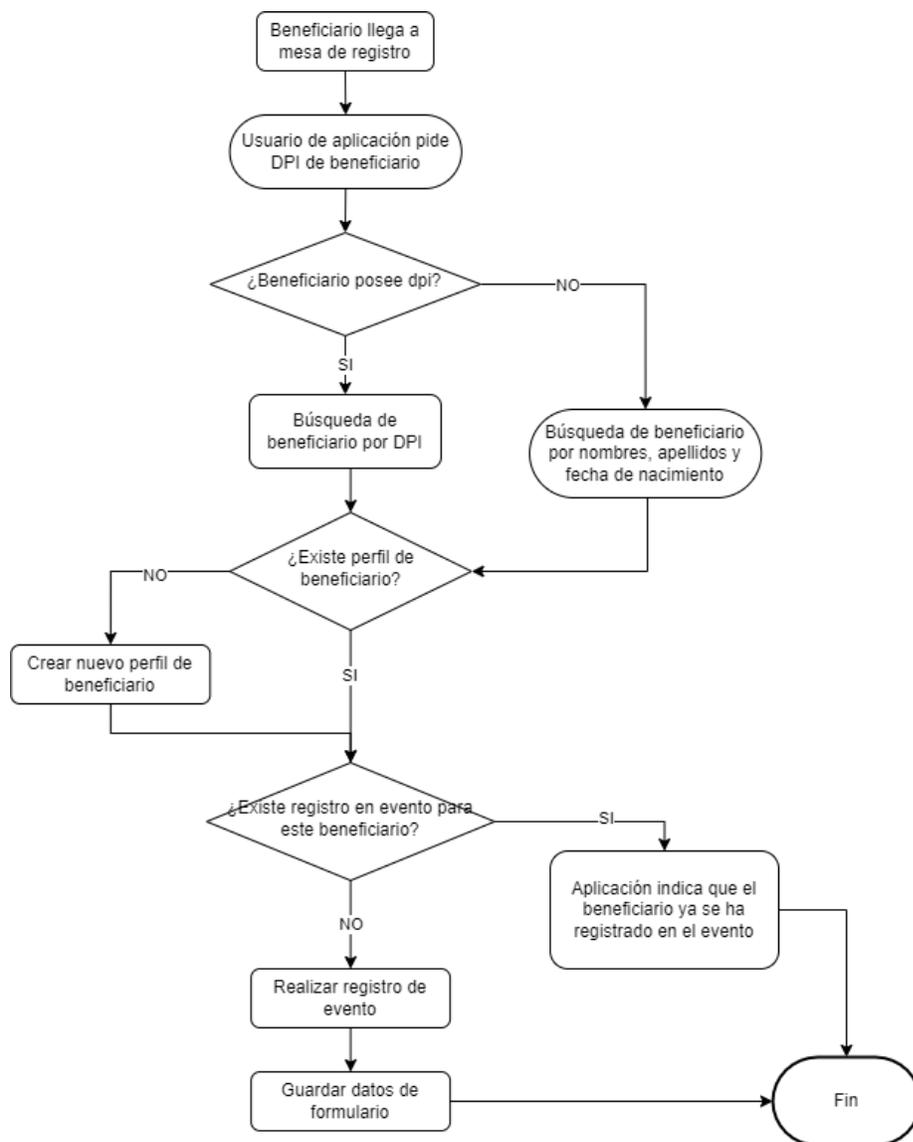
REFERENCIAS

1. Angular. (2021). *Introducción a la documentación de angular*. California: Autor.
2. Atlassian. (2021). Flujo de trabajo de *Gitflow*. Sydney: Autor.
3. Firebase. (2021). *Obtén información sobre los conceptos fundamentales de Firebase*. California: Autor.
4. Flask. (2021). *Guía de usuario*. Austria: The Pallets projects.
5. Google Cloud Platform. (2021). *Primeros pasos con Google Cloud*. California: Autor.
6. Grimm, S. (30 de octubre, 2018). Construcción de una aplicación Ionic 4 en modo Offline [Mensaje en un blog]. Recuperado de <https://devdactic.com/ionic-4-offline-mode>
7. Grimm, S. (14 de junio, 2022). De React web a una aplicación móvil nativa con Ionic y Capacitor [Mensaje en un blog]. Recuperado de <https://devdactic.com/react-web-to-native-capacitor>
8. Gupta, P. (6 de noviembre, 2019). Consulta de MongoDB como una base de datos SQL [Mensaje en un blog]. Recuperado de <https://betterprogramming.pub/querying-mongodb-like-an-sql-db-using-aggregation-pipeline-9b7e1dc780b7>

9. Herman, R. (5 de julio, 2020). Guía de 5 pasos para implementar tu aplicación React en el motor de aplicaciones de Google [Mensaje en un blog]. Recuperado de <https://javascript.plainenglish.io/quickly-deploy-your-react-app-on-googles-app-engine-6bb97480cc9c>
10. Ionic (2021). Introducción a Ionic. Wisconsin: Autor.
11. Mahugh, D. (17 de febrero, 2020). Despliegue de una aplicación Flask en GAE [Mensaje en un blog]. Recuperado de https://medium.com/@dmahugh_70618/deploying-a-flask-app-to-google-app-engine-faa883b5ffab
12. Material UI (2021). Guía rápida. Paris: Autor.
13. MongoDB (2021). Configuraciones iniciales de MongoDB en Flask. Nueva York: Autor.
14. SQLite (2021). Documentación oficial de SQLite. Carolina del Norte: Autor.
15. Xia, S. (9 de enero, 2020). Conectando Flask con una aplicación MongoDB [Mensaje en un blog]. Recuperado de <https://medium.com/@summerxialinqiao/connect-flask-app-to-mongodb-atlas-using-pymongo-328e119a7bd8>

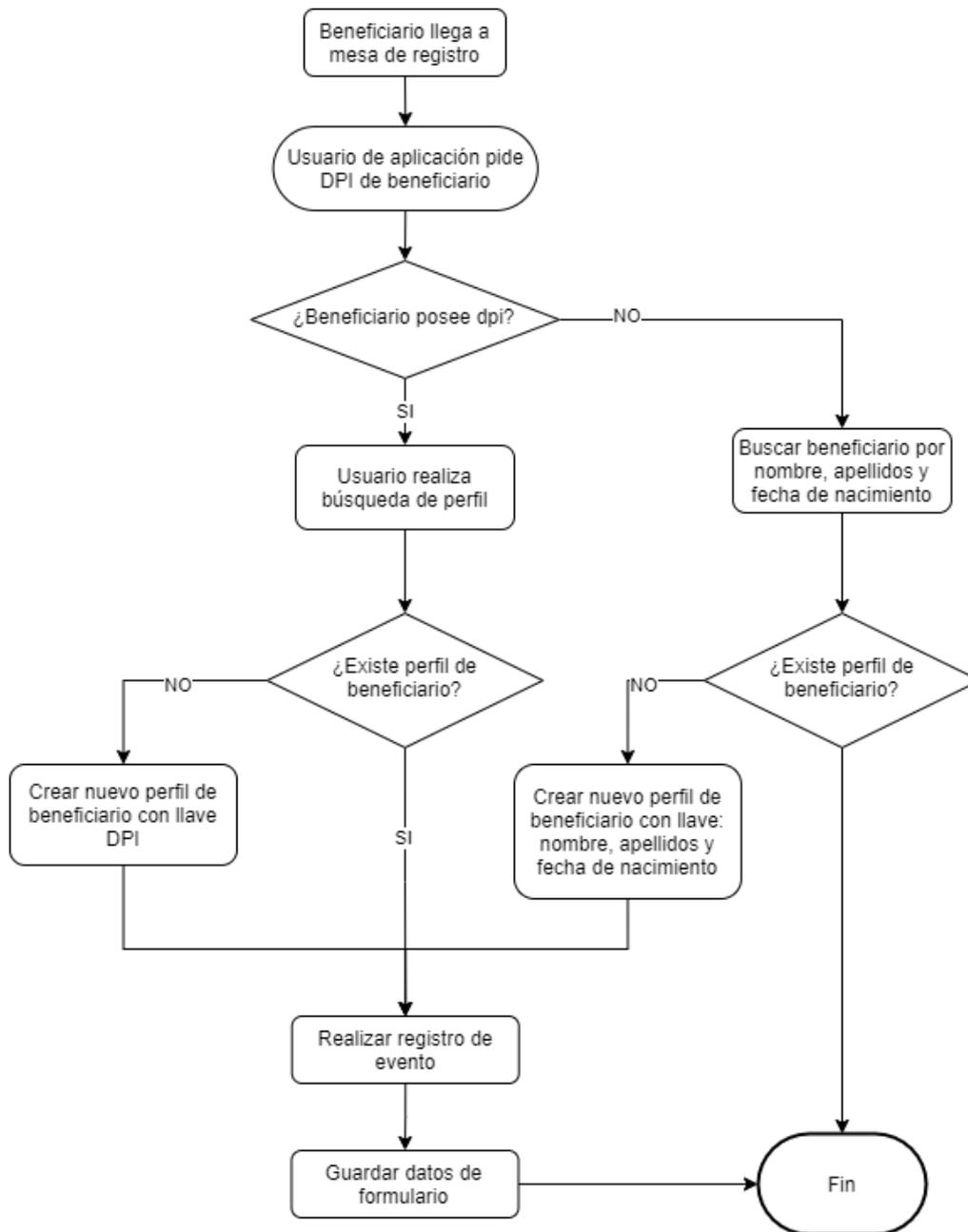
APÉNDICES

Apéndice 1. Flujo de registro en evento de registro único



Fuente: elaboración propia, realizado con Draw.io.

Apéndice 2. Flujo de registro en evento de registro libre



Fuente: elaboración propia, realizado con Draw.io.

Apéndice 3. Carta de compromiso de mantenimiento

Guatemala 10/11/2022

Lic. David Vásquez
Administrador de "Misión El Faro"
Presente

Es un gusto para mí poder saludarles por este medio, esperando que tengan muchos éxitos en sus actividades diarias.

Yo, Christian Levi Gonzalez Rodriguez, estudiante de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala, me comprometo a brindar soporte técnico durante 3 meses como mantenimiento del sistema, para solventar cualquier inconveniente que pueda surgir.

El mantenimiento de 3 meses, posterior a la entrega final, consistirá en lo siguiente:

- Realizar una revisión quincenal de código, para poder solventar errores reportados y actualizaciones que se puedan hacer sin cambiar el alcance del proyecto.
- Realizar los despliegues de las aplicaciones que necesiten ser actualizadas por algún cambio en el código.
- Actualizar el repositorio de código con la última versión desplegada.

Todos los costos generados por los servicios de la nube serán asociados a la cuenta de facturación de "Misión El Faro" al realizar la entrega oficial del proyecto (entrega del sitio administrativo y aplicación móvil).

Este mantenimiento forma parte de la propuesta planteada en el trabajo final de investigación, por lo que no se cobrará ningún costo adicional a "Misión El Faro" o a cualquier otra persona involucrada en el proyecto.

Sin más que agregar, agradezco de antemano su apoyo durante todo este proceso. Reciba un cordial saludo.

Atentamente,



Christian Levi Gonzalez Rodriguez

Fuente: elaboración propia, realizado con Word.

