

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DESARROLLO DE UN SISTEMA WEB Y MÓVIL PARA LA
ASIGNACIÓN DE TAREAS, CONTROL Y ADMINISTRACIÓN DE
PROYECTOS, CON ENVÍO DE NOTIFICACIONES EN TIEMPO
REAL**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

JULIO ISRAEL GONZÁLEZ PALLO

ALEXANDRA BARBARITA GUALOTUÑA FERNÁNDEZ

DIRECTOR: Ing. Pablo Hidalgo, MSc.

Quito, marzo 2021

AVAL

Certifico que el presente trabajo fue desarrollado por Julio Israel González Pallo y Alexandra Barbarita Gualotuña Fernández, bajo mi supervisión.

MSc. PABLO WILIAN HIDALGO LASCANO
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Nosotros, Julio Israel González Pallo y Alexandra Barbarita Gualotuña Fernández, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejamos constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

JULIO ISRAEL GONZÁLEZ PALLO

ALEXANDRA BARBARITA GUALOTUÑA
FERNÁNDEZ

DEDICATORIA

Con amor a mis padres, mi hermano, mis abuelitas y a Alexandra

DEDICATORIA

Con amor a mis padres, abuelita paterna, hermanos y a Julio

Alexandra

AGRADECIMIENTO

A mis padres Julio y Gloria, por el amor, el apoyo incondicional y las fuerzas que supieron darme en los momentos más difíciles, sin ustedes no habría logrado nada. Gracias por la confianza que supieron brindarme.

A mi hermano Guillermo, por todo su apoyo y grandes momentos que hemos pasado, más que un hermano es el mejor de mis amigos.

A mis abuelitas María Mercedes y María Esther, por el amor y la motivación de cada día, gracias por todas sus enseñanzas y consejos que a lo largo de mi vida me forjaron como persona.

A los mejores amigos, que todo el mundo quisiera tener, Daniel, Jessenia, Santiago, María José, Samanta, María Eugenia, que han sido parte de esta etapa tan importante en mi vida, gracias por todos los momentos maravillosos que hemos vivido.

A Stalyn, Cristian A, Cristian C, Juan, gracias por tantos años de amistad y por todo su apoyo.

A Alexandra, por haber formado parte de este logro, y haber dado todo en nuestro proyecto de titulación. Gracias por todos estos años, en los que siempre hemos salido adelante juntos. Por todo tu cariño, comprensión y confianza, a lo largo de todo este tiempo y por ser esa parte maravillosa de mi vida. El día de hoy cosechamos un gran éxito en nuestras vidas, y espero que sean muchos más los cuales obtengamos juntos.

Al Ingeniero Pablo Hidalgo, por la paciencia, tiempo, esfuerzo y ayuda, en este proyecto de titulación. Por sus consejos y ser guía en la universidad y en la vida. Gracias por ser un excelente docente y persona.

JULIO ISRAEL GONZALEZ PALLO

AGRADECIMIENTO

Agradezco a Dios por permitirme vivir este momento. A mis padres Gloria y Efraín, que son el motor de mi vida y mi razón de seguir adelante. Gracias por su apoyo incondicional y el ejemplo de responsabilidad, humildad y lucha que ha demostrado siempre.

A mi abuelita Dolores Masabanda quien ha cuidado de mí siempre, y que, a pesar de los años, en su corazón, su amor hacia mi persona sigue intacto.

A mis hermanos, Milton y Homero que siempre de una u otra manera están pendientes de mí, de mis metas y logros, gracias por ser ese ejemplo de perseverancia y por darme el regalo más bonito de mi vida, mis sobrinas Annethe, Emilia, Sarahi, Antonella y mi pequeño Jose Francisco. Gracias a Hipatia y Margarita por brindarme sus consejos y apoyo.

Durante esta etapa he conocido personas maravillosas, que me han brindado su amistad sincera e incondicional. Gracias a mi grupo de amigos de siempre, Yeyé, Jona L., Majo, Santo, Dani, Sami, con los cuales he compartido momentos increíbles desde el inicio de mi vida universitaria. Gracias a las chicas de fútbol, Cris, Carmita, Gaby, con las cuales he compartido triunfos, derrotas y risas.

A mi gran amiga, Rous, gracias por todo tu apoyo y por estar conmigo en los buenos y malos momentos, por ser esa amiga incondicional en la que puedo confiar y que sé que estará siempre para mí.

A Julio, coautor de este trabajo de titulación, gracias por el apoyo y todo lo que hiciste por sacar adelante este proyecto. Gracias por ser mi fortaleza en esta etapa, por estar conmigo a pesar de las adversidades. Hoy termina una etapa, pero empieza otra y sé que también serás parte de ella brindándome tu apoyo y amor. Gracias por ser ese hombre lleno de valores y respeto hacia mi persona, gracias por ser parte mi vida.

Un agradecimiento especial al Ingeniero Pablo Hidalgo, quien fue un excelente guía en este Trabajo de Titulación, quien me brindó su tiempo, conocimientos y paciencia. Gracias por los consejos, por su amistad y la confianza depositada. Gracias por ser un docente excepcional y una excelente calidad de persona.

ALEXANDRA BARBARITA GUALOTUÑA FERNÁNDEZ

ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	V
ÍNDICE DE CONTENIDO.....	VII
ÍNDICE DE FIGURAS	X
ÍNDICE DE CÓDIGOS	XVIII
ÍNDICE DE TABLAS	XIX
RESUMEN	XX
ABSTRACT.....	XXI
1 INTRODUCCIÓN.....	1
1.1 OBJETIVOS.....	1
1.2 ALCANCE	1
1.3 MARCO TEÓRICO	4
1.3.1 ANGULAR	4
1.3.2 IONIC.....	9
1.3.3 APLICACIONES WEB PROGRESIVAS (PWA).....	11
1.3.4 BASE DE DATOS NO RELACIONALES.....	12
1.3.5 FIREBASE	12
1.3.6 GOOGLE CLOUD PLATFORM	14
1.3.7 HERRAMIENTAS DE DESARROLLO.....	15
1.3.8 METODOLOGÍA ÁGIL KANBAN	17
2 METODOLOGÍA.....	19
2.1 DISEÑO DEL SISTEMA.....	19
2.1.1 ESTRUCTURA DEL SISTEMA.....	19

2.1.2	REQUERIMIENTOS GENERALES	21
2.1.3	ROLES Y PERMISOS	25
2.1.4	DIAGRAMA DE CASOS DE USO.....	26
2.1.5	ESQUEMAS DE LA INTERFAZ GRÁFICA.....	28
2.1.6	LÓGICA DE PROCESOS	58
2.1.7	DIAGRAMA DE CLASES	79
2.1.8	ESQUEMA DE LA BASE DE DATOS.....	80
2.2	IMPLEMENTACIÓN DEL SISTEMA.....	84
2.2.1	INSTALACIÓN DE LAS HERRAMIENTAS DE DESARROLLO	84
2.2.2	CODIFICACIÓN DE FIREBASE	89
2.2.3	CODIFICACIÓN DE LA INTERFAZ GRÁFICA.....	103
2.2.4	DESPLIEGUE DEL SISTEMA.....	113
3	RESULTADOS Y DISCUSIÓN	120
3.1	PRUEBAS INDIVIDUALES	121
3.1.1	MÓDULO REGISTRO	122
3.1.2	MÓDULO AUTENTICACIÓN	125
3.1.3	MÓDULO EQUIPO.....	130
3.1.4	MÓDULO PROYECTOS	138
3.1.5	MÓDULO ACTIVIDADES.....	147
3.1.6	MÓDULO CRONOGRAMA	162
3.1.7	MÓDULO CHAT	164
3.1.8	MÓDULO PERFIL USUARIO.....	167
3.1.9	MÓDULO NOTIFICACIONES	168
3.2	PRUEBAS DE INTEGRACIÓN	174
3.2.1	VERIFICACIÓN DE PWA.....	178
3.3	CORRECCIÓN DE ERRORES	180
3.4	PRUEBAS DE VALIDACIÓN	182
4	CONCLUSIONES Y RECOMENDACIONES	187

4.1	CONCLUSIONES	187
4.2	RECOMENDACIONES	190
5	REFERENCIAS BIBLIOGRÁFICAS.....	191
	ANEXOS	193
	ORDEN DE EMPASTADO.....	194

ANEXO A: Formato de la encuesta para la empresa HOMATELCO

ANEXO B: Conjunto de las interfaces implementadas en la aplicación web

ANEXO C: Conjunto de las interfaces implementadas en la aplicación móvil

ANEXO D: Encuesta de validación para el rol Gestor

ANEXO E: Encuesta de validación para el rol Delegado

ANEXO F: Código fuente de las aplicaciones

ÍNDICE DE FIGURAS

Figura 1.1. Funcionamiento general del sistema	3
Figura 1.2. Ejemplo de un código HTML.....	4
Figura 1.3. Ejemplo de códigos CSS (a) Etiqueta (b) Clase e (c) Identificador	5
Figura 1.4. Ejemplo de código SCSS.....	5
Figura 1.5. Funcionamiento de una SPA vs Aplicación Tradicional.....	6
Figura 1.6. Plataforma Firebase.....	14
Figura 1.7. Dashboard de Google Cloud Plataform.....	15
Figura 1.8 Visual Studio Code.....	16
Figura 1.9 Consola de Firebase	17
Figura 1.10 Tablero Kanban.....	18
Figura 2.1 Tablero Kanban - Etapa de diseño del sistema	20
Figura 2.2. Diagrama de estructura del Sistema desarrollado	23
Figura 2.3. Caso de uso: Usuario registrado como Gestor de proyectos	27
Figura 2.4. Caso de uso: Usuario registrado como Delegado.....	27
Figura 2.5. Logo del sistema	30
Figura 2.6. Sketch de la pantalla Login	31
Figura 2.7. Modal Soporte al usuario	32
Figura 2.8. Modal Recuperar contraseña	32
Figura 2.9. Sketch de la pantalla Registro.....	33
Figura 2.10. Sketch de la pantalla Inicio – Gráficos estadísticos	33
Figura 2.11. Sketch de la pantalla Inicio – Guia de usuario	34
Figura 2.12. Sketch de los íconos de la cabecera.....	34
Figura 2.13. Sketch de la pantalla Perfil	35
Figura 2.14. Sketch de la pantalla Notificaciones	35
Figura 2.15. Sketch de la pantalla Mi Equipo para el rol Gestor	36
Figura 2.16. Sketch de la pantalla Mi Equipo para el rol Gestor - Pestaña Editar Equipo.....	37
Figura 2.17. Sketch de la pantalla Mi Equipo para el rol Gestor - Pestaña Ver Equipo por proyecto.....	37
Figura 2.18. Sketch de la pantalla Mi Equipo para el rol Delegado – Ver Equipo.....	38
Figura 2.19. Sketch de la pantalla Mi Equipo para el rol Delegado - Pestaña Ver equipo por proyecto.....	38
Figura 2.20. Sketch del modal Recursos (a) Pestaña Observaciones (b) Pestaña Archivos.....	39

Figura 2.21. Sketch de la pantalla Proyectos para el rol Gestor.....	39
Figura 2.22. Sketch del modal Proyecto.....	40
Figura 2.23. Sketch de la pantalla Proyectos para el rol Delegado	40
Figura 2.24. Sketch del modal Evidencia para el rol Delegado (a) Pestaña Archivos y (b) Pestaña Observaciones.....	41
Figura 2.25. Sketch de la pantalla Cronograma para el rol Gestor.....	41
Figura 2.26. Sketch de la pantalla Cronograma para el rol Delegado	42
Figura 2.27. Sketch de la pantalla Chat.....	42
Figura 2.28. Sketch de la pantalla Actividades.....	43
Figura 2.29. Sketch de la pantalla Actividades - Actividades desplegadas	43
Figura 2.30. Sketch de la pantalla Actividades – Opciones de actividades y tareas	44
Figura 2.31. Sketch del modal Actividad.....	45
Figura 2.32. Sketch del modal Tarea.....	45
Figura 2.33. Sketch del modal Evidencia para el rol Gestor (a) Pestaña Archivos (b) Pestaña Observaciones y (c) Pestaña Avance.....	46
Figura 2.34. Sketch de un recuadro de Alerta	46
Figura 2.35. Sketch de un recuadro de Aviso.....	47
Figura 2.36. (a) Mockup de la pantalla Login (b) Mockup del modal Contacto y (c) Modal Recuperar contraseña	48
Figura 2.37. Mockup de la pantalla Registro	48
Figura 2.38. Mockup de la pantalla Inicio (a) Gráficos estadísticos y (b) Guía de usuario	49
Figura 2.39. Mockups de la pantalla (a) Mi Perfil y (b) Notificaciones.....	49
Figura 2.40. Mockups de la pantalla Mi Equipo para el rol Gestor (a) Pestaña Seleccionar Equipo (b) Ver Equipo y (c) Ver Equipo por proyecto	50
Figura 2.41. Mockups de la pantalla Mi Equipo para el rol Delegado (a) Ver Equipo y (b) Ver equipo por proyecto	51
Figura 2.42. Mockups del modal Recursos (a) Pestaña Observaciones y (b) Pestaña Archivos.....	51
Figura 2.43. (a) Mockup de la pantalla Proyectos para el rol Gestor y (b) Mockup del modal Proyecto	52
Figura 2.44. Mockups de la pantalla Actividades (a) Opciones de cada Actividad (b) actividad desplegada y (c) Opciones de cada tarea.....	53
Figura 2.45. (a) Mockup del modal Actividad y (b) Mockup del modal Tarea	54

Figura 2.46. Mockups del modal Evidencia para el rol Gestor (a) Pestaña Archivos (b) Pestaña Observaciones y (c) Pestaña Avance.....	54
Figura 2.47. Mockup de la pantalla Proyectos para el rol Delegado.....	55
Figura 2.48. Mockups del modal Evidencia (a) Pestaña Archivos y (b) Pestaña Observaciones	56
Figura 2.49. (a) Mockup de la pantalla Cronograma para el rol Gestor y (b) Mockup del modal Calendario	56
Figura 2.50. Mockup de la pantalla Cronograma para el rol Delegado.....	57
Figura 2.51. (a) Mockup de la pantalla Chat y (b) Mockup de la pantalla Conversación ...	57
Figura 2.52. Diagrama de actividad del proceso Registrar/Autenticar Usuario	60
Figura 2.53. Diagrama de secuencia del proceso Registrar Usuario.....	61
Figura 2.54. Diagrama de secuencia del proceso Autenticar Usuario	62
Figura 2.55. Diagrama de actividad del Proceso Conformar equipo.....	64
Figura 2.56. Diagrama de secuencia del Proceso Conformar equipo.....	65
Figura 2.57. Diagrama de actividad del Proceso Administrar proyectos	68
Figura 2.58. Diagrama de secuencia del Proceso Administrar proyectos	69
Figura 2.59. Diagrama de actividad del Proceso Cumplir proyectos	70
Figura 2.60. Diagrama de secuencia del Proceso Cumplir proyectos	71
Figura 2.61. Diagrama de actividad del Proceso Conversar un usuario.....	72
Figura 2.62. Diagrama de secuencia del Proceso Conversar con un usuario.....	73
Figura 2.63. Diagrama de actividad para el Proceso Configurar perfil.....	74
Figura 2.64. Diagrama de secuencia para el Proceso Configurar perfil	75
Figura 2.65. Diagrama de actividad para el Proceso Revisar cronograma.....	76
Figura 2.66. Diagrama de secuencia para el Proceso Revisar cronograma.....	77
Figura 2.67. Diagrama de actividad para el Proceso Revisar notificaciones.....	78
Figura 2.68. Diagrama de secuencia para el Proceso Revisar notificaciones.....	79
Figura 2.69. Diagrama de clases del sistema.....	80
Figura 2.70. Esquema de la colección users.....	81
Figura 2.71. Esquema de la colección teams.....	81
Figura 2.72. Esquema de la colección companies	82
Figura 2.73. Esquema de la colección projects	83
Figura 2.74. Esquema de la colección evidences	83
Figura 2.75. Esquema de la colección chats.....	84
Figura 2.76. Tablero Kanban - Etapa de implementación	85
Figura 2.77. Paquete de instalación Node.js	85
Figura 2.78. Node.js instalado y su gestor de paquetes.....	86

Figura 2.79. Instalación de Angular CLI.....	86
Figura 2.80. Verificación de la instalación de Angular CLI.....	86
Figura 2.81. Instalación de Ionic CLI.....	87
Figura 2.82. Verificación de la instalación de Ionic CLI.....	87
Figura 2.83. Estructuración de carpetas del sistema.....	88
Figura 2.84. Base de datos del sistema.....	89
Figura 2.85. Colección users en Firestore.....	90
Figura 2.86. Colección teams en Firestore.....	91
Figura 2.87. Colección projects en Firestore – Parte 1.....	92
Figura 2.88. Colección projects en Firestore - Parte 2.....	93
Figura 2.89. Colección evidences en Firestore.....	94
Figura 2.90. Colección companies en Firestore.....	95
Figura 2.91. Colección chats en Firestore.....	95
Figura 2.92. Instalación de las dependencias de Firebase.....	96
Figura 2.93. Archivos de la carpeta environment.....	97
Figura 2.94. Servicio de autenticación en Firebase.....	98
Figura 2.95. Lista de usuarios Autenticados.....	99
Figura 2.96. Dominios autorizados.....	99
Figura 2.97. Instalación de Angular Material.....	104
Figura 2.98. Interfaz de la pantalla Proyectos para el rol Gestor.....	107
Figura 2.99. Interfaz de la pantalla Proyectos para el rol Delegado.....	107
Figura 2.100. Interfaz del modal Proyecto.....	108
Figura 2.101. Interfaz del modal Evidencia para el rol Delegado.....	108
Figura 2.102. (a) Interfaz de la pantalla Chat y (b) Conversación entre usuarios.....	111
Figura 2.103. Instalación de la dependencia PWA angular.....	112
Figura 2.104. Archivo manifest.webmanifest de la aplicación PWA.....	113
Figura 2.105. Icono de la aplicación en diferentes resoluciones.....	113
Figura 2.106. Construcción del aplicativo web en producción.....	114
Figura 2.107. Construcción del aplicativo móvil en producción.....	114
Figura 2.108. Configuración del archivo angular.json en el aplicativo web.....	115
Figura 2.109. Configuración del archivo angular.json en el aplicativo móvil.....	115
Figura 2.110. Instalación de Firebase-Tools.....	115
Figura 2.111. Inicio de sesión en Firebase a través de la consola.....	116
Figura 2.112. Selección del servicio de Firebase Hosting.....	116
Figura 2.113. Sitio Web epn-gugo.....	117
Figura 2.114. Sitio móvil mobile-epn-gugo.....	117

Figura 2.115. Archivo de configuración para el aplicativo web	117
Figura 2.116. Archivo de configuración para el aplicativo móvil	118
Figura 2.117. Comando para indicar donde se hará el despliegue.....	118
Figura 2.118. Despliegue de la aplicación web en el hosting de Firebase	118
Figura 2.119. Despliegue de la aplicación móvil en el hosting de Firebase	119
Figura 2.120. Instalación de la aplicación móvil en un Smartphone	119
Figura 3.1. Actualización tablero Kanban – Etapa de Resultados Parte 1	120
Figura 3.2. Actualización tablero Kanban – Etapa de Resultados Parte 2	121
Figura 3.3. Formulario de Registro (a) Aplicación web y (b) Aplicación móvil.....	122
Figura 3.4. Validación de datos (a) Aplicación web y (b) Aplicación móvil	123
Figura 3.5. Error del uso empresarial.....	124
Figura 3.6. Error de verificación de correo electrónico	124
Figura 3.7. Registro Exitoso (a) Aviso y (b) Correo electrónico recibido.....	125
Figura 3.8. Formulario de autenticación (a) Aplicación web y (b) Aplicación móvil.....	125
Figura 3.9. Error de credenciales (a) Contraseña errónea y (b) Correo electrónico erróneo.....	126
Figura 3.10. Autenticación exitosa (a) Aplicación web y (b) Aplicación móvil.....	126
Figura 3.11. Formulario de recuperación de contraseña.....	127
Figura 3.12. Formulario de generación de nueva contraseña.....	127
Figura 3.13. Formulario de Contactarme con soporte - Aplicación Web	128
Figura 3.14. Formulario de Contactarme con soporte - Aplicación Móvil	128
Figura 3.15. Guía de usuario - Aplicación web	129
Figura 3.16. Guía de usuario - Aplicación móvil	129
Figura 3.17. Selección de equipo de trabajo - Aplicación móvil.....	130
Figura 3.18. Selección de equipo de trabajo - Aplicación web.....	130
Figura 3.19. Visualización de equipo de trabajo rol Gestor - Aplicación móvil	131
Figura 3.20. Visualización de equipo de trabajo rol Gestor - Aplicación web	131
Figura 3.21. Confirmación para remover un Delegado del equipo de trabajo	132
Figura 3.22. Error Remover Delegado con tareas a su cargo.....	132
Figura 3.23. Actualización de lista del equipo de trabajo rol Gestor - Aplicación web ...	132
Figura 3.24. Actualización de lista del equipo de trabajo rol Gestor - Aplicación móvil..	133
Figura 3.25. Equipos de trabajo por proyecto rol Gestor - Aplicación web	133
Figura 3.26. Equipos de trabajo por proyecto rol Gestor - Aplicación móvil	134
Figura 3.27. Visualización de equipo de trabajo rol Delegado – Aplicación web.....	135
Figura 3.28. Visualización de equipo de trabajo rol Delegado - Aplicación móvil.....	135
Figura 3.29. Equipos de trabajo por proyecto rol Delegado – Aplicación web	136

Figura 3.30. Equipos de trabajo por proyecto rol Delegado - Aplicación móvil	136
Figura 3.31 Formulario de recursos - Aplicación web (a) Observaciones y (b) Archivos	137
Figura 3.32. Formulario de recursos - Aplicación móvil (a) Observaciones y (b) Archivos	138
Figura 3.33. Formulario para crear un nuevo proyecto (a) Aplicación web, (b) Aplicación móvil y (c) Mensaje de éxito	139
Figura 3.34. Lista de proyectos actualizada Rol Gestor – Actualización 1 (a) Aplicación web y (b) Aplicación móvil	140
Figura 3.35. Mensaje de confirmación para eliminar un proyecto.....	141
Figura 3.36. Formulario para editar un Proyecto - Aplicación web	141
Figura 3.37. Formulario para editar un Proyecto - Aplicación móvil.....	142
Figura 3.38. Lista de proyectos actualizada – Actualización 2 - Aplicación web	142
Figura 3.39. Lista de proyectos actualizada – Actualización 3 - Aplicación móvil	143
Figura 3.40. Visualización de tareas asignadas (a) Aplicación web y (b) Aplicación móvil	144
Figura 3.41. Formulario evidencia rol Delegado – Aplicación web (a) Selección de un archivo (b) Carga de archivo y (c) Observaciones.....	145
Figura 3.42. Formulario evidencia rol Delegado – Aplicación móvil.....	146
Figura 3.43. Formulario de nueva actividad - Aplicación web	147
Figura 3.44. Formulario de nueva actividad - Aplicación móvil	148
Figura 3.45. Actualización de la lista de actividades – Aplicación Web.....	148
Figura 3.46. Actualización de lista de actividades – Aplicación móvil.....	149
Figura 3.47. Formulario de editar una actividad (a) Aplicación web y (b) Aplicación móvil	150
Figura 3.48. Eliminación de una actividad (a) Aplicación web y (b) Aplicación móvil.....	151
Figura 3.49. Actualización de la lista de Actividades - Aplicación web	151
Figura 3.50. Actualización de la lista de Actividades - Aplicación móvil	152
Figura 3.51. Formulario de nueva tarea (a) Aplicación web y (b) Aplicación móvil	153
Figura 3.52. Formulario de editar tarea (a) Aplicación web y (b) Aplicación móvil	154
Figura 3.53. Eliminación de una tarea (a) Aplicación web y (b) Aplicación móvil.....	155
Figura 3.54. Actualización de la lista de tareas (a) Aplicación web y (b) Aplicación móvil	156
Figura 3.55. Descarga de un archivo – Aplicación web.....	157
Figura 3.56. Descarga de un archivo – Aplicación móvil.....	157

Figura 3.57. Observaciones del formulario evidencia rol Gestor (a) Aplicación web y (b) Aplicación móvil.....	158
Figura 3.58. Confirmación de porcentaje de avance (a) Aplicación web y (b) Aplicación móvil	159
Figura 3.59. Corrección del porcentaje de avance (a) Aplicación web y (b) Aplicación móvil	160
Figura 3.60. Gráficos estadístico rol Gestor (a) Aplicación web y (b) Aplicación móvil	161
Figura 3.61. Visualización del módulo cronograma rol Gestor - Aplicación web	162
Figura 3.62. Visualización del módulo cronograma rol Gestor - Aplicación móvil	162
Figura 3.63. Lista de tareas pertenecientes al módulo Cronograma rol Gestor	163
Figura 3.64. Visualización del módulo Cronograma rol Delegado - Aplicación web	163
Figura 3.65 Visualización del módulo Cronograma rol Delegado - Aplicación móvil.....	164
Figura 3.66. Conversación entre un (a) Gestor y (b) Delegado - Aplicación web.....	165
Figura 3.67. Conversación entre un (a) Gestor, (b) Delegado y (c) Información intercambiada - Aplicación móvil	166
Figura 3.68. Agregar una foto al perfil de usuario - Aplicación móvil	167
Figura 3.69. Permisos para recibir notificaciones (a) Aplicación web y (b) Aplicación móvil	168
Figura 3.70. Visualización del módulo Notificaciones – Aplicación móvil	169
Figura 3.71. Visualización del módulo Notificaciones – Aplicación web	170
Figura 3.72. Ejemplos de notificaciones que recibe el Gestor - Aplicación web.....	170
Figura 3.73. Notificaciones que recibe el Gestor - Aplicación móvil	171
Figura 3.74. Ejemplos de notificaciones que recibe el Delegado – Aplicación web	172
Figura 3.75. Ejemplos de notificaciones que recibe el Delegado – Aplicación móvil	173
Figura 3.76. Número de notificaciones que recibe el usuario	174
Figura 3.77. Usuarios registrados en Firebase Authentication para las pruebas individuales.....	175
Figura 3.78. Pantalla Chat para el Gestor.....	176
Figura 3.79. Pantalla Chat para el Delegado.....	176
Figura 3.80. Acceso a los archivos del dispositivo móvil.....	177
Figura 3.81. Almacenamiento de la imagen en Cloud Storage.....	177
Figura 3.82. Visualización de la nueva foto de perfil	178
Figura 3.83. Service worker instalado.....	178
Figura 3.84. Parámetros de Lighthouse para la aplicación móvil.....	179
Figura 3.85. Compatibilidad del <i>prompt</i> de instalación en los navegadores.....	180

Figura 3.86. Resultados de la encuesta para ambos roles	183
Figura 3.87. Resultados de la encuesta para el rol Gestor	185
Figura 3.88. Resultados de la encuesta para el rol Delegado.....	185

ÍNDICE DE CÓDIGOS

Código 2.1. Código del método register	100
Código 2.2. Código del Método login	100
Código 2.3. Código del método createUser.....	101
Código 2.4. Código del método uploadResourcesFirebase.....	102
Código 2.5. Configuración del Service Worker.....	102
Código 2.6. Código del método requestPermission	103
Código 2.7. Código del método receiveMessage.....	103
Código 2.8. Código del archivo projects.component.html	104
Código 2.9. Código del archivo projects.component.css.....	105
Código 2.10. Código del archivo projects.component.ts	105
Código 2.11. Código del método openNewProject.....	106
Código 2.12. Código del método onUpdateFiles	106
Código 2.13. Código del archivo chat.pages.html – Segmento ion-header.....	109
Código 2.14. Código del archivo chat.pages.html – Segmento ion-content	109
Código 2.15. Código del archivo chat.pages.scss.....	110
Código 2.16. Código del archivo chat.pages.ts	110
Código 2.17. Código del método getChats.....	111

ÍNDICE DE TABLAS

Tabla 2.1. Resumen de los resultados de las encuestas	22
Tabla 3.1. Eventos generadores de notificaciones para el rol Gestor	169
Tabla 3.2. Eventos generadores de notificaciones para el rol Delegado	171
Tabla 3.3. Corrección de errores – Parte 1	181
Tabla 3.4. Corrección de errores – Parte 2	182

RESUMEN

En el manejo de proyectos, algunas empresas llevan un control manual de los mismos mediante anotaciones; esto provoca un trabajo laborioso y la posibilidad de errores humanos. Otras empresas utilizan aplicaciones de escritorio como Excel y MS Project, lo que hace que su uso esté limitado al computador donde estén instalados, que dependan del sistema operativo y requieran instalación personalizada. Además, estas alternativas para la administración de proyectos no permiten el uso de notificaciones, carga de archivos ni mensajería instantánea, para ello se requiere del uso de otros programas.

Por ello se ha desarrollado un sistema web y móvil para la asignación de tareas, control y administración de proyectos, con envío de notificaciones en tiempo real. Este sistema contiene una aplicación web y una aplicación móvil que se comunican entre sí, y utilizan los servicios de Firebase para cumplir con funcionalidades como: autenticación/registro del usuario, lectura y escritura de datos, almacenamiento de archivos y manejo de notificaciones.

En el primer capítulo se explican los *frameworks* Angular y Ionic que se emplean en el desarrollo del sistema, utilizando como entorno de desarrollo Visual Studio Code. En el segundo capítulo se presenta la metodología de desarrollo del sistema, la misma que está basada en Kanban; se indican los requerimientos funcionales y no funcionales, así como el diseño e implementación de las pantallas del sistema.

En el tercer capítulo se presentan las pruebas de funcionamiento generales del sistema, encuestas realizadas al usuario para la verificación de las funcionalidades y la corrección de posibles errores en base al análisis de resultados de las pruebas realizadas. Finalmente, en el cuarto capítulo se presentan conclusiones y recomendaciones obtenidas al finalizar este proyecto.

Todas las pantallas de la aplicación web y de la aplicación móvil se encuentran en el ANEXO B y en el ANEXO C respectivamente.

PALABRAS CLAVE: Sistema web y móvil, Angular, Ionic, Firebase, Servicios, Administración de Proyectos.

ABSTRACT

In the management of projects, some companies keep a manual control of them by means of notes; this causes a laborious work and the possibility of human errors. Other companies use desktop applications such as Excel and MS Project, which makes their use limited to the computer where they are installed, depending on the operating system and require custom installation. In addition, these alternatives for project management do not allow the use of notifications, file uploading or instant messaging, which requires the use of other programs.

For this reason, a web and mobile system has been developed for task assignment, control and project management, with the sending of notifications in real time. This system contains a web application and a mobile application that communicate with each other, and uses Firebase services to fulfill functionalities such as: user authentication/registration, data reading and writing, file storage and notification handling.

The first chapter explains the Angular and Ionic frameworks used in the development of the system, using Visual Studio Code as the development environment. The second chapter presents the system development methodology, which is based on Kanban; it indicates the functional and non-functional requirements, as well as the design and implementation of the system screens.

The third chapter presents the general functioning tests of the system, surveys conducted to the user to verify the functionalities and the correction of possible errors based on the analysis of test results. Finally, the fourth chapter presents conclusions and recommendations obtained at the end of this project.

All the screens of the web application and the mobile application can be found in APPENDIX B and APPENDIX C respectively.

KEYWORDS: Web and mobile system, Angular, Ionic, Firebase, Services, Project Management.

1 INTRODUCCIÓN

En este capítulo se explican temas relevantes sobre el desarrollo del trabajo, se definen los objetivos y el alcance de éste. Además, en la sección 1.3 se especifican los aspectos teóricos en los que se basa el trabajo y las herramientas que fueron parte del desarrollo del software, tanto para la aplicación web como para la aplicación móvil. También forma parte de este capítulo los entornos de desarrollo que se han utilizado para el desarrollo del sistema.

1.1 OBJETIVOS

El objetivo general de este Trabajo de Titulación es:

Desarrollar una aplicación web y una aplicación móvil para la asignación de tareas, control y administración de proyectos, con envío de notificaciones en tiempo real.

Los objetivos específicos propuestos en este Proyecto Integrador son:

- Analizar el funcionamiento de las herramientas necesarias para el desarrollo de las aplicaciones.
- Diseñar los diferentes módulos que conforman el sistema.
- Implementar los módulos diseñados.
- Analizar los resultados de las pruebas de funcionamiento.

1.2 ALCANCE

Existen varios métodos para el control y administración de proyectos; algunas empresas llevan un control manual mediante anotaciones, provocando un trabajo laborioso y con posibilidad de errores humanos. Otras empresas utilizan aplicaciones de escritorio propias de Microsoft como Excel y Project; estos programas son buenos para el control de proyectos, pero no tienen manera de comprobar si las tareas fueron o no cumplidas.

Además, no son útiles para labores específicas como el uso de notificaciones, carga de archivos y mensajería instantánea, para ello se requiere del uso de otros programas. [1]

Microsoft Excel y MS Project limitan su uso al computador donde estén instalados, dependen del sistema operativo y requieren instalación personalizada. Para realizar algún cambio o revisión del proyecto se necesita un computador que disponga la aplicación. Para trasladar información se debe utilizar cualquier sistema de almacenamiento masivo u otra aplicación, lo cual está sujeto a pérdidas de información. Microsoft Excel viene dentro del paquete de Microsoft Office de forma predeterminada, mientras que MS Project necesita otro paquete de instalación. Existen soluciones que permiten el control y administración de proyectos con herramientas adicionales dentro de una sola plataforma, pero éstas representan costos recursivos mensuales muy altos. [2]

Por ello se desarrolla un sistema web y móvil para brindar una solución fácil y rápida al usuario en el manejo de proyectos, dentro de una sola plataforma. Este sistema permite adicionalmente el envío de notificaciones, carga de archivos (evidencias), comunicación entre usuarios y almacenamiento en la nube entre otras tareas. La aplicación web provee portabilidad y su funcionalidad es independiente del sistema operativo, permitiendo el acceso desde cualquier navegador web. La aplicación móvil permite el acceso desde cualquier lugar sin la necesidad de un dispositivo de escritorio o portátil, ya que las personas interactúan todo el tiempo con su dispositivo móvil.

El desarrollo del sistema está basado en la metodología ágil Kanban. Se estudian los lenguajes HTML (*HyperText Markup Language*), CSS/SCSS (*Cascading Style Sheets / Sassy Cascading Style Sheets*) y el lenguaje de programación *TypeScript* que son parte de los *frameworks* Angular [3] y Ionic [4], y temas adicionales como el manejo de Angular Material [5], Servicios Web y el funcionamiento de Firebase[6] a través de *Google Cloud Platform*.

El sistema web y móvil permite realizar tareas propias del control y administración de proyectos tales como: seleccionar los miembros iniciales del equipo, la duración estimada del proyecto, la duración de las actividades, asignación de responsabilidades, listado de tareas y actividades, establecimiento de canales de comunicación, y visualización de resultados.

Adicionalmente, se tienen herramientas que ayudan a cumplir las tareas antes mencionadas, así como el envío de notificaciones automáticas cuando suceda un evento significativo, la carga de archivos para el manejo de evidencias y mensajería instantánea para la comunicación entre usuarios, todo esto dentro de una sola plataforma y sin la necesidad de la utilización de otros programas externos.

Las aplicaciones cuentan con el manejo de roles para un mejor uso de éstas. A continuación, se indican los roles a utilizar:

Gestor o Administrador: es la persona que dirige uno o varios proyectos, y es el encargado de que los proyectos se cumplan a tiempo.

Delegado: es la persona a la que el Gestor le asigna una actividad específica y será el encargado de cumplir cada actividad asignada.

En la Figura 1.1 se indica el funcionamiento general del sistema.

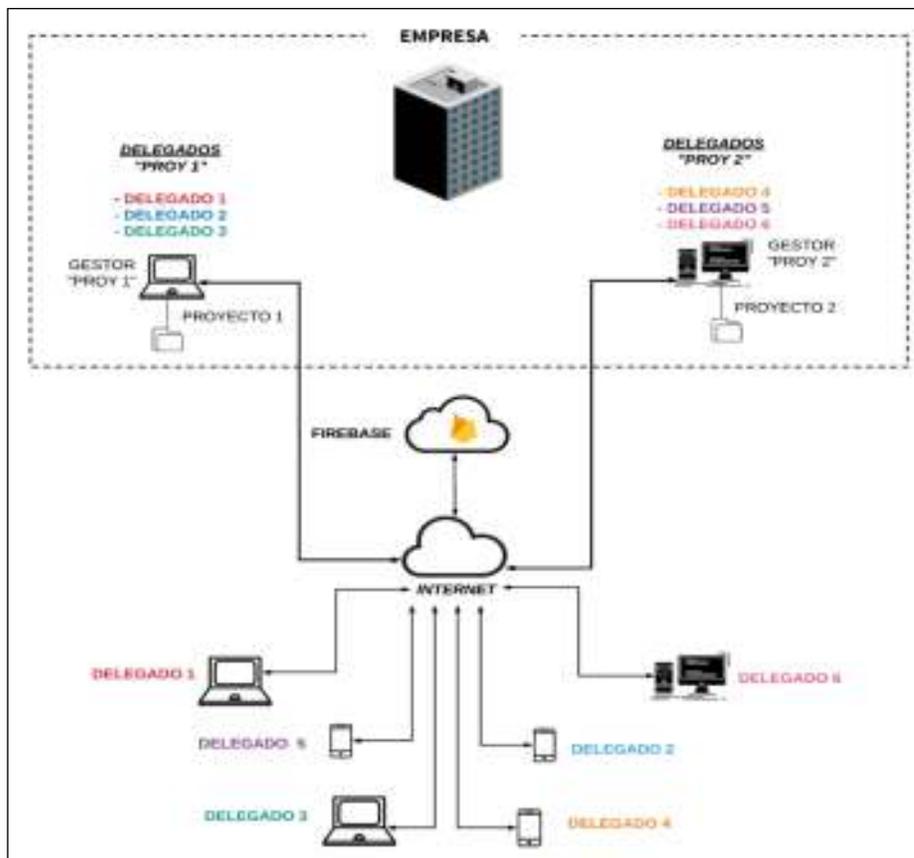


Figura 1.1. Funcionamiento general del sistema

1.3 MARCO TEÓRICO

1.3.1 ANGULAR

Angular es un *framework*¹ utilizado en el desarrollo de aplicaciones web SPA (*Single Page Application*), el cual hace uso del lenguaje de programación *TypeScript* y los lenguajes HTML y CSS, los cuales se explicarán más adelante.

Angular tiene como propósito la separación completa entre el *frontend*² y el *backend*³. Este *framework* construye la parte del *frontend*, permitiendo que un cliente consuma un servicio alojado en el *backend* y éste devuelve la información requerida, por lo que son totalmente independientes. [7]

Como ya se mencionó, Angular trabaja con HTML, CSS/SCSS y *TypeScript* por lo que es conveniente dar una breve descripción de cada uno de estos lenguajes.

HTML: Es un lenguaje de marcado de hipertexto utilizado en el desarrollo de páginas web. Utiliza etiquetas, las cuales permiten definir las partes de un documento, encabezados, cuerpo, párrafos y demás elementos que conforman una página web como se indica en la Figura 1.2.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Figura 1.2. Ejemplo de un código HTML

¹ *Framework*: es un conjunto de bibliotecas que permiten el desarrollo y organización de aplicaciones.

² *Frontend*: es la parte de la aplicación con la que el usuario interactúa, por lo que se encuentra en el lado del cliente.

³ *Backend*: es la parte que se conecta con la base de datos y el servidor de la aplicación, y se ejecuta en el lado del servidor. El usuario no tiene los permisos para su acceso.

CSS: Es un lenguaje que define el estilo de un documento HTML; posicionamiento, colores, tamaño, animación entre otros. En la Figura 1.3 se indican algunos ejemplos de código CSS; en la parte (a) se indica el estilo aplicado a una etiqueta HTML, esto se aplica a todos los elementos de esa misma etiqueta. En la parte (b) se indica el estilo aplicado a una clase CSS, el cual selecciona todos los elementos que se encuentren dentro de esta clase; y, en la parte (c) se indica el estilo aplicado a un identificador, mismo que permite seleccionar etiquetas individuales mediante el *id* de la etiqueta.

```
h1 {  
  font-size: 26px;  
}
```

```
.destacado {  
  font-weight: bold;  
  color: orange;  
}
```

```
#formularioAlta {  
  border: 1px solid #99c;  
}
```

(a)

(b)

(c)

Figura 1.3. Ejemplo de códigos CSS (a) Etiqueta (b) Clase e (c) Identificador

SCSS: Es un preprocesador de CSS que permite generar una hoja de estilos, proporcionando funcionalidades que no son propias de CSS, como es el uso de variables, funciones, selectores anidados, entre otras, y permite tener una organización de estilos más robusta y avanzada. En la Figura 1.4 se presenta un ejemplo de código SCSS.

```
$white: #ffffff;  
$ubuntu-font: $ubuntu-font: 'Ubuntu', 'Arial', 'Helvetica', sans-serif;  
  
body{  
  color: $white;  
  font: $ubuntu-font;  
  font-size: xx-large;  
  padding: 2rem;  
}
```

Figura 1.4. Ejemplo de código SCSS

TypeScript: Es un lenguaje de programación de código abierto orientado a objetos. Está basado y compilado en JavaScript, por lo tanto, es muy similar a éste, pero presenta mayor robustez y menos probabilidad de errores debido a la declaración de variables y funciones tipadas, es decir, a cada valor le corresponde un tipo de dato.

1.3.1.1 Aplicaciones SPA

Una aplicación web SPA es un tipo de aplicación donde todo el contenido está dentro de un solo archivo, por lo tanto, se tienen varias vistas, más no varias páginas, por lo que solo se sustituye el contenido. Por este motivo estas páginas son más rápidas al momento de cargar y no necesitan que el navegador recargue su página durante su uso [8]. Estas aplicaciones de una sola página usan JavaScript o lenguajes que están compilados en éste.

En la Figura 1.5 se puede ver las diferencias entre las aplicaciones tradicionales y aplicaciones SPA. En las aplicaciones tradicionales, se hace una llamada al servidor cada vez que requiere cargar una página, el servidor retorna el HTML necesario, lo entrega al navegador y recarga la página para mostrar los datos. Por el contrario, en una aplicación SPA, ésta solo interactúa con el servidor una vez y las demás interacciones las hace mediante llamadas AJAX⁴. El servidor devuelve los datos necesarios en formato JSON⁵ y sin la necesidad de recargar la página [9].

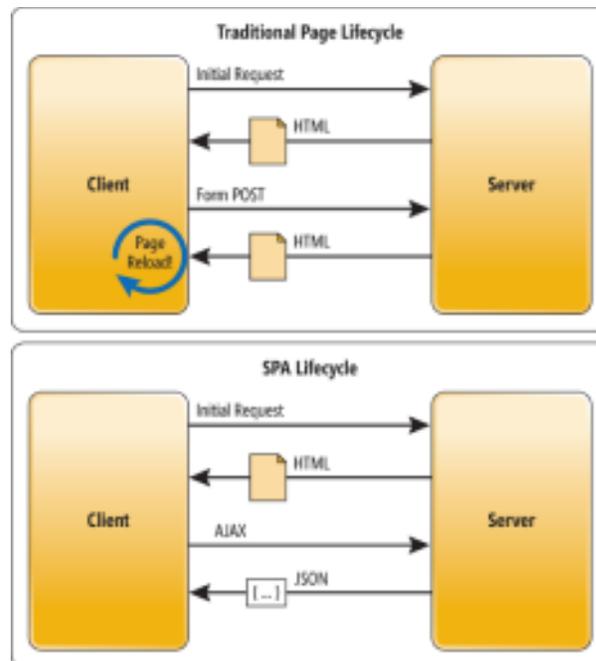


Figura 1.5. Funcionamiento de una SPA vs Aplicación Tradicional [9]

⁴ *AJAX (Asynchronous Javascript and XML)*: es una técnica usada en el desarrollo de sitios web, permite que un sitio web que ya ha sido cargado pueda solicitar nueva información al servidor.

⁵ *JSON (JavaScript Object Notation)*: es un formato utilizado para el intercambio de datos.

1.3.1.2 Estructura de una aplicación en angular

Angular posee algunos elementos que son parte de la estructura [10]; a continuación, se detallan los más relevantes y que han sido parte del desarrollo de este trabajo.

1.3.1.2.1 Módulos

Los módulos son mecanismos que agrupan servicios, componentes y directivas que están relacionadas; tienen la capacidad de utilizar librerías externas, así como combinar su funcionalidad con otros módulos para crear una aplicación. Además, permiten organizar las partes de la aplicación mediante ordenamiento en bloques.

Un módulo consta de un decorador llamado *ngModule*, en el cual se permite declarar, importar y exportar componentes para toda la aplicación, lo cual brinda flexibilidad ya que evita el hecho de importar los componentes y directivas cada vez que sean utilizadas. Un decorador es una función que agrega metadatos a una clase, sus miembros o sus argumentos de método determinan cómo se debe procesar, instanciar y usar el componente en tiempo de ejecución. Normalmente un decorador es prefijado con una '@'.

Cada aplicación hecha en Angular posee un módulo raíz llamado convencionalmente *Appmodule*, éste proporciona el mecanismo de arranque que permite que la aplicación sea iniciada.

1.3.1.2.2 Componentes

Un componente es un bloque de código estructurado que puede ser reutilizado, y está conformado generalmente por tres elementos: un archivo *TypeScript* el cual contiene los datos y la lógica del componente, un archivo HTML conformado por una plantilla de la página, y finalmente, un archivo CSS/SCSS encargado de darle estilos. Un componente puede ser o no una página; el objetivo de éstos es poder tener elementos reutilizables para toda la aplicación.

1.3.1.2.3 Plantillas

Una plantilla es la combinación entre HTML y recursos propios de Angular, que permite modificar elementos HTML antes de que éstos se muestren. Angular se encarga de evaluar las directivas y resolver la sintaxis en la plantilla, para modificar los elementos HTML y el

DOM⁶ (*Document Object Model*), según los datos y la lógica de la aplicación. Angular admite el enlace bidireccional de datos, esto quiere decir que los cambios en el DOM también serán reflejados en los datos de la aplicación [10].

1.3.1.2.4 *Servicios y enlaces de datos*

El objetivo principal de un servicio es organizar y compartir la lógica de negocios, modelos de datos y funciones con diferentes componentes de una aplicación en Angular. Los componentes no deberían buscar ni guardar datos directamente, sino que deben centrarse en presentar y delegar el acceso a datos a un servicio, por lo que los servicios son una excelente manera de compartir información entre componentes que no se conocen entre sí.

Cada servicio creado consta de un `@Injectable`, que es un decorador que marca una clase como disponible para ser proporcionada e inyectada como una dependencia. Por lo tanto un servicio podrá ser utilizado en varios componentes.

1.3.1.2.5 *Enrutamiento*

Aquí se asignan rutas que son similares a los *URL (Uniform Resource Locator)*; estas rutas son asignadas a distintos componentes que conforman la aplicación, y que serán manejadas como vistas de acuerdo con la definición de SPA. Una aplicación en Angular consta de un enrutador y es el encargado de detectar las acciones que hace el navegador para mostrar u ocultar las vistas.

1.3.1.3 **Herramientas de diseño**

Dentro del mundo de desarrollo web, existen algunas librerías que facilitan el diseño y mejoran la visualización ante el usuario. Entre éstas se tiene Bootstrap, Angular Material e incluso HTML puro, siendo Angular Material la elegida para este trabajo, debido a que es propia de Angular.

⁶ *DOM*: es una interfaz para páginas web. Es esencialmente una API para la página, que permite a los programas leer y manipular el contenido, la estructura y los estilos de la página.

Angular Material es una librería utilizada para el diseño de la interfaz de aplicaciones web, se encuentra basado en *Material Design* y presenta múltiples elementos, entre ellos se tienen botones, iconos, cajas de texto y muchos más. Además, presenta animaciones, tablas, gráficos, entre otros. Para hacer uso de esta librería, se requiere de una simple instalación mediante un comando a través del CLI de Angular [5].

1.3.2 IONIC

El desarrollo de aplicaciones móviles ha ido creciendo a través de los años, es así como surgen nuevas herramientas para el desarrollo de éstas. Bajo este contexto existen tres alternativas para desarrollarlas:

- **Aplicaciones nativas:** este tipo de aplicaciones son desarrolladas específicamente para cada sistema operativo y su respectivo lenguaje. *Objective-C*⁷ para iOS, *Java*⁸ y *Kotlin*⁹ para *Android*, y *.Net*¹⁰ para *Windows Phone*. Además, estas aplicaciones presentan algunos beneficios; tienen acceso a los recursos del dispositivo móvil (cámara, GPS), no requieren de una conexión a Internet para ser usadas, se carga en la pantalla inicial del dispositivo, envío de notificaciones automáticas, entre otras más.
- **Aplicaciones móviles web:** este tipo de aplicaciones son desarrolladas con tecnologías web JavaScript, HTML, CSS. Las aplicaciones móviles web se adaptan a cualquier sistema operativo y al navegador móvil del dispositivo. Estas aplicaciones requieren de una conexión a Internet para su funcionamiento y no tienen acceso a los recursos del dispositivo móvil.
- **Aplicaciones híbridas:** este tipo de aplicaciones son una combinación entre las aplicaciones nativas y web. Consisten en una *WebView*¹¹ que se ejecuta dentro de

⁷ Objective-C: es un lenguaje de programación orientado a objetos, utilizado en el desarrollo de aplicaciones para sistemas operativos como Mac OS X y iOS.

⁸ Java: es un lenguaje de programación orientado a objetos, utilizada para el desarrollo de aplicaciones web, aplicaciones Android.

⁹ Kotlin: es un lenguaje de programación, utilizado en el desarrollo de aplicaciones Android.

¹⁰ .Net: es una plataforma utilizada en el desarrollo de aplicaciones de Windows Phone.

¹¹ WebView: es un navegador que se encuentra integrado en una aplicación.

un contenedor nativo. Además, se empaquetan como aplicaciones para distribución y tienen acceso a las *APIs*¹² nativas del dispositivo móvil.

Bajo este contexto se define Ionic como un *framework* utilizado para el desarrollo de aplicaciones móviles híbridas con una alta calidad y rendimiento, haciendo uso del lenguaje de programación JavaScript o TypeScript y los lenguajes HTML y CSS/SCSS. Además, Ionic se integra fácilmente con otros *frameworks* como Angular, React o Vue, siendo Angular el elegido para este trabajo. El uso de Angular junto con Ionic permite el desarrollo de aplicaciones altamente interactivas y proporciona grandes herramientas y servicios que permiten que la ejecución de Ionic sea simple [4].

Ionic se encuentra preparado para soportar aplicaciones PWA (*Progressive Web App*) ya que existen herramientas que permiten el desarrollo de los componentes necesarios para este tipo de aplicaciones, estas herramientas son *services workers* y *web manifest* [11].

- **Services workers:** son archivos Javascript registrados por el navegador, y permiten que una PWA funcione, es decir permite el almacenamiento de los datos en cache, conexión sin Internet, actualizaciones, notificaciones *push*.
- **Web manifest:** se trata de un archivo JSON que define los parámetros de una PWA, los colores, el icono, el nombre de la aplicación, entre otras.

1.3.2.1 Estructura de una aplicación Ionic

Una aplicación en Ionic tiene la misma estructura que una aplicación en Angular debido a que se encuentra integrada con ésta. Angular es el encargado de procesar las peticiones, procesar datos, conectarse a servicios y Ionic es el encargado de que la aplicación se vea de manera adecuada en un dispositivo móvil. Por lo tanto, Ionic trabaja con módulos, componentes, enrutamiento y los demás elementos que ya han sido explicados en el *framework* Angular.

¹² APIs (*Application Programming Interface*): es un conjunto de protocolos que se utilizan para que las aplicaciones se puedan comunicar entre ellas.

1.3.3 APLICACIONES WEB PROGRESIVAS (PWA)

Las Aplicaciones Web Progresivas son una integración entre las aplicaciones web y las aplicaciones nativas. Este tipo de aplicaciones se cargan como páginas web, pero permiten al usuario tener funcionalidades propias de las aplicaciones móviles nativas, como por ejemplo notificaciones *push*, trabajo sin conexión a Internet, acceso al hardware del dispositivo y otros beneficios que proporcionan las aplicaciones nativas y que ya han sido mencionados.

Las PWA presentan algunos beneficios [11], que las hacen atractivas a los desarrolladores, a continuación, se presenta algunos de ellos:

- **Velocidad:** la velocidad de carga de una PWA es mayor que una aplicación web móvil, lo que brinda una mejor experiencia al usuario.
- **Sin instalación:** las PWA no requieren de una publicación en Google Play o en App Store, es decir no requieren necesariamente de una instalación. Estas solo requieren de un *click* en un enlace o código QR.
- **Multiplataforma:** estas aplicaciones funcionan sobre cualquier sistema operativo y sobre cualquier dispositivo.

1.3.3.1 Aplicaciones Web Progresivas vs. aplicaciones híbridas

Como se ha explicado antes, tanto una aplicación híbrida, como una Aplicación Web Progresiva, son una combinación entre una aplicación nativa y una aplicación móvil web, pero cada aplicación trabaja de una manera diferente.

Las aplicaciones híbridas son colocadas en los distintos sistemas operativos (Android o iOS) mediante una instalación. Esto conlleva mayor trabajo de programación debido a las configuraciones que deben realizarse para cada sistema operativo. Mientras que, en las Aplicaciones Web Progresivas, el usuario accede al sitio web y este le entrega la opción de proporcionar un icono de acceso directo en la pantalla principal del dispositivo, simulando una instalación. De esta manera el usuario puede volver a la aplicación cuando lo requiera y estará disponible en su lista de aplicaciones. Estas aplicaciones no requieren

de configuraciones personalizadas para cada sistema operativo, sino que dependen de la compatibilidad del dispositivo y el navegador en el cual se están ejecutando.

1.3.4 BASE DE DATOS NO RELACIONALES

Una base de datos es un conjunto de información que pertenece a un mismo contexto y que son almacenados sistemáticamente. Existen múltiples tipos de base de datos, pero para este trabajo nos centraremos en las bases de datos no relacionales y se hará una comparación entre las bases de datos relacionales y las no relacionales.

Las bases de datos relacionales se manejan mediante datos organizados en tablas. Además, se encuentran relacionadas unas con otras mediante un identificador y cumplen con el modelo entidad-relación. Las bases de datos no relacionales son aquellas que no cumplen con el esquema entidad-relación, por lo tanto, no hacen uso de tablas ni se tiene un identificador que sirva de relación entre una tabla y otra. Este tipo de bases organizan su información sin un esquema exacto [12]. Existen varios tipos de base de datos no relacionales; clave-valor, documentos y gráficos [13]. Las bases de datos no relacionales del tipo documento son las que se usan para este trabajo.

Primero, ¿Por qué una base de datos no relacional? Las bases de datos no relacionales son recomendadas cuando el volumen de los datos crece rápidamente [14]. Este trabajo está orientado a la creación y control de proyectos, por lo que a diario se crearán proyectos y se realizarán actualizaciones de éstos, por lo que se espera un crecimiento rápido de la información. Ahora, ¿Por qué del tipo documento? En las bases de datos no relacionales del tipo documento, los datos se muestran en un objeto o documento del tipo JSON, y es éste el utilizado por los *frameworks* de desarrollo. Por lo tanto, estas bases son ideales para el presente trabajo.

1.3.5 FIREBASE

Firebase es una plataforma creada por Google que ofrece un servicio *BaaS*¹³ (*Backend-as-a-Service*), cuya función principal es brindar soluciones sencillas y de alta calidad para el desarrollo de aplicaciones web y móviles. Esta plataforma permite mejorar el rendimiento de las aplicaciones, mediante la ejecución de funciones implementadas en la nube.

¹³ BaaS: es un conjunto de herramientas que se encuentran ya implementadas en la nube.

Firestore ofrece algunos servicios [6]; a continuación, se explican algunos de los servicios que oferta y que han sido utilizados para este trabajo:

- *Firestore RealTime Database y Cloud Firestore*: Firestore ofrece estas bases de datos NoSQL¹⁴ en tiempo real, por lo que los datos serán enviados apenas sean actualizados y todos los clientes conectados reciben los datos casi al instante. Cuando una aplicación requiere conectarse con Firestore, lo hace a través de un *WebSocket*¹⁵ y no a través de llamadas HTTP por lo que aumenta la rapidez de envío de los datos.
- *Firestore Authentication*: Firestore ofrece un servicio de autenticación incorporado, basado en correo electrónico y contraseña o número de teléfono, aunque también facilita la integración con otras aplicaciones como Facebook, Twitter, entre otras. Esta autenticación permite el almacenamiento de datos del usuario en la nube de forma segura.
- *Firestore Cloud Storage*: Firestore ofrece un servicio de almacenamiento de objetos, en donde el usuario puede guardar archivos como imágenes, video, entre otros.
- *Firestore Hosting*: Firestore ofrece un servicio de alojamiento de una o varias páginas o aplicaciones web. Permite la publicación y actualización del aplicativo tan solo con un comando de consola. Así también ofrece subdominios con los cuales se puede acceder al aplicativo.
- *Firestore Cloud Messaging*: Firestore ofrece un servicio de mensajería, el cual permite el envío de notificaciones a los usuarios.
- *Firestore Cloud Functions*: Firestore ofrece un servicio para implementar y ejecutar el código de *backend* sin la necesidad de servidores, administración ni escalamiento de éstos. Permite la ejecución de código ante solicitudes HTTPS realizadas por los aplicativos, y es almacenado en la nube de Google bajo un entorno administrativo.

¹⁴ NoSQL: Base de datos no relacionales.

¹⁵ *WebSocket*: es una API que permite la comunicación abierta entre dos dispositivos, esta comunicación es bidireccional.

En la Figura 1.6 se indica la plataforma Firebase con la que se trabajó, los servicios que oferta y que fueron utilizados.



Figura 1.6. Plataforma Firebase

1.3.6 GOOGLE CLOUD PLATFORM

Es un conjunto de servicios computacionales en la nube, y permite que la infraestructura manejada por Google se encuentre disponible para todo el público. Esta plataforma va de la mano de Firebase ya que aquí se pueden encontrar los recursos y servicios que está utilizando en la nube [15].

Google Cloud Platform ofrece la capacidad aumentar el tamaño de la infraestructura sin preocuparse del mantenimiento y administración de servidores. Esta plataforma ha permitido alojar las funciones necesarias para el correcto funcionamiento del sistema desarrollado. En la Figura 1.7 se indica el *dashboard* con las funcionalidades más relevantes que ofrece *Google Cloud Platform*.

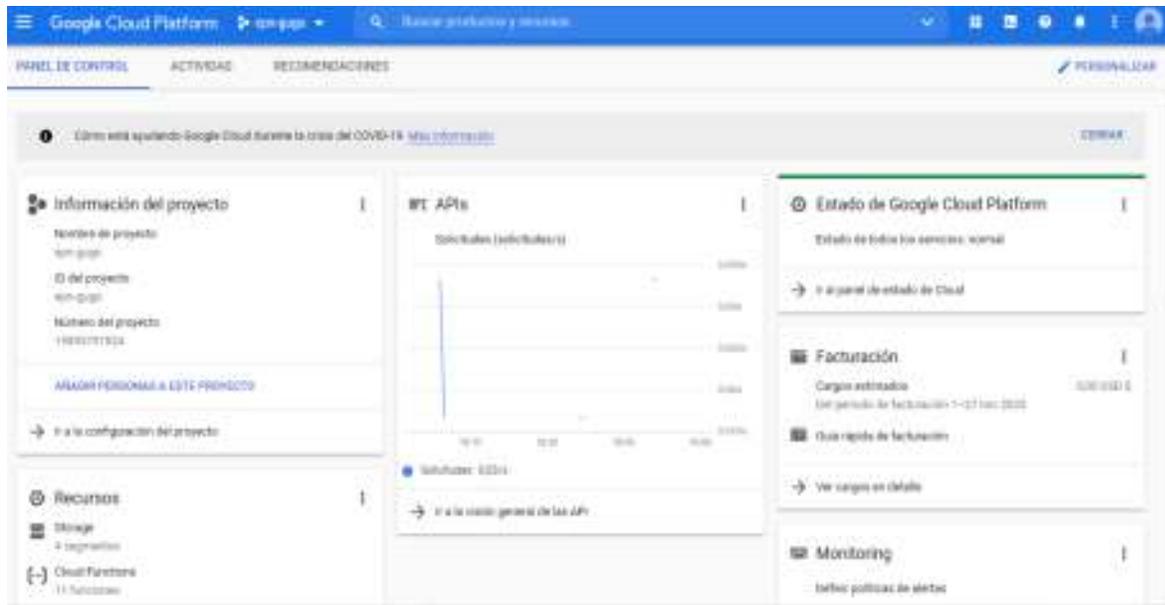


Figura 1.7. Dashboard de Google Cloud Plataform [16]

1.3.7 HERRAMIENTAS DE DESARROLLO

1.3.7.1 Node.js

Node.js es un entorno de ejecución para JavaScript multiplataforma y de código abierto. Éste maneja eventos asíncronos, sin bloqueo y ejecuta varias tareas con poco consumo de recursos lo que causa que sea liviano y eficiente.

Node.js maneja su propio sistema Gestor de paquetes llamado NPM (*Node Package Manager*), el cual permite instalar distintas dependencias para el desarrollo de aplicaciones. Estas herramientas permiten inicializar y desarrollar el sistema mediante la instalación de interfaces de línea de comandos y librerías.

1.3.7.1.1 Command line interface (CLI)

CLI es una herramienta de interfaz de línea de comandos utilizada para interactuar en el desarrollo de aplicaciones de manera simple.

Angular y Ionic poseen un CLI que les permite crear, correr, depurar, emular, probar y publicar aplicaciones. Además, permite la esquematización de carpetas y archivos requeridos en el desarrollo de aplicaciones.

1.3.7.2 Visual Studio Code

Es un editor de código desarrollado por Microsoft y que se encuentra disponible para Windows, Linux, MacOS. Brinda soporte para depuración, integración de Git, incorporación con JavaScript, TypeScript y Node.js y se puede extender para otros lenguajes tales como; C++¹⁶, C #¹⁷, Java, Python¹⁸, PHP¹⁹. En la Figura 1.8 se indica una captura de pantalla del editor de código Visual Studio Code [17].

1.3.7.3 Firebase console

Firebase permite la creación de proyectos mediante el manejo de su consola. Una vez creado un determinado proyecto éste puede utilizar los servicios que ofrece Firebase; *Authentication, Firestore, Storage, Hosting, Functions*. Cada servicio posee su propia interfaz, con la cual el usuario puede trabajar. En la Figura 1.9 se indica la consola que proporciona Firebase para crear proyectos, y los servicios que proporciona cada proyecto.



Figura 1.8 Visual Studio Code [17]

¹⁶ C++: es una extensión del lenguaje de programación C, pero presenta mecanismos de manipulación de objetos.

¹⁷ C#: es un lenguaje de programación orientado a objetos independiente que permite crear programas sobre la plataforma .Net.

¹⁸ Python: es un lenguaje de scripting orientado a objetos y que es independiente de la plataforma.

¹⁹ PHP (*Hypertext Preprocessor*): es un lenguaje de código abierto usado en el desarrollo de aplicaciones web.

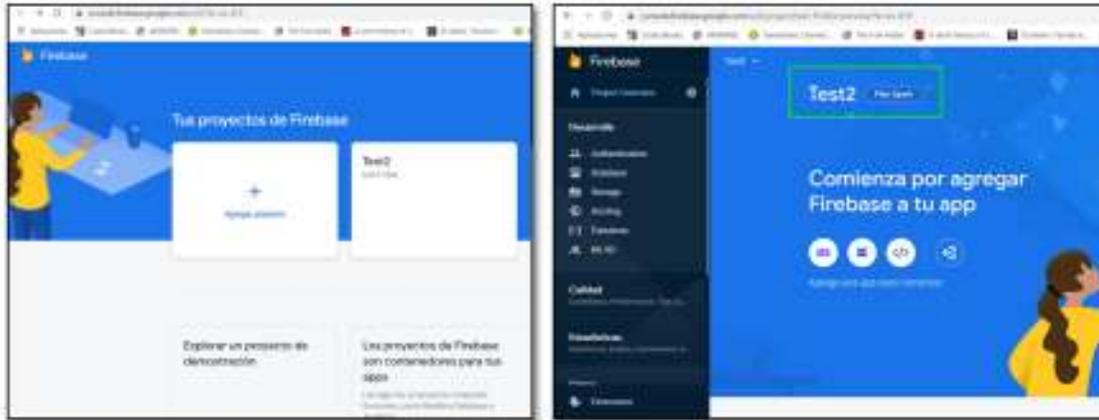


Figura 1.9 Consola de Firebase [18]

1.3.7.4 Git y Github

En el desarrollo de software es fundamental administrar un proyecto a través de versiones de código, con el fin de tener acceso a versiones anteriores en el caso que los nuevos cambios no han resultado satisfactorios o han causado algún tipo de problema [19].

Git es una herramienta de código abierto para el control de versiones, mantiene un registro acerca de lo que va sucediendo en un proyecto de manera local. Cuando se crea un proyecto lo que se tiene es una versión básica del mismo y luego se van añadiendo nuevas características. Git permite volver a una versión anterior; además, permite trabajar en equipo, cuyos miembros pueden realizar distintas partes del proyecto y fusionarlas de manera fácil [20].

GitHub es una plataforma para el control de versiones usando Git. Esta plataforma está basada en la nube por lo que es un servicio de alojamiento y administración de repositorios de Git [20].

1.3.8 METODOLOGÍA ÁGIL KANBAN

Una metodología de software es un entorno de trabajo cuyo objetivo es estructurar, planificar y controlar el desarrollo de un sistema con el fin de hacerlo más eficiente. Para este trabajo se ha seleccionado una metodología ágil basada en Kanban.

Kanban permite visualizar el flujo de trabajo mediante el uso de tareas, esta metodología es sencilla de utilizar y muy útil para determinar la situación en la que se encuentra el

trabajo. Para poder visualizar el flujo de trabajo se hace uso del tablero Kanban, este tablero se encuentra dividido en columnas, las cuales representan un estado dentro del flujo de trabajo.

Básicamente, el tablero Kanban está dividido en tres columnas: **por hacer, haciendo y hecho**. En la Figura 1.10 se indica un ejemplo del tablero Kanban, en cada columna se puede colocar la tarea según su estado. Además, cada tarea puede cambiar de estado según vaya avanzando.



Figura 1.10 Tablero Kanban

2 METODOLOGÍA

En el primer capítulo se explicaron los objetivos, el alcance del proyecto y el marco teórico. Dentro del marco teórico se detallaron los conceptos fundamentales que fueron necesarios para el desarrollo de este proyecto. Este proyecto desarrolla un sistema conformado por una aplicación web y una aplicación móvil. Bajo este contexto el presente capítulo está compuesto por dos ítems principales, el primero trata del diseño del sistema y el segundo trata de la implementación del mismo.

2.1 DISEÑO DEL SISTEMA

El diseño del sistema está conformado por el modelado de los diagramas de casos de uso, los esquemas de la interfaz gráfica, tanto del aplicativo web como del aplicativo móvil, la lógica de procesos y el diseño de la base de datos no relacional.

De acuerdo con la metodología en la que se encuentra basada este proyecto, se implementa el tablero Kanban con las tareas a realizar para el diseño e implementación del sistema, así como las pruebas de funcionalidad y resultados del proyecto. En la Figura 2.1 se indica el tablero Kanban correspondiente a la etapa de diseño del sistema.

Las tareas se encuentran especificadas por color, color naranja corresponden a las tareas del diseño del sistema, color verde corresponden a las tareas de implementación del sistema y color azul corresponden a las tareas de pruebas generales del funcionamiento del sistema.

2.1.1 ESTRUCTURA DEL SISTEMA

En la Figura 2.2 se indica la interacción de los usuarios con los módulos a implementar, así como la unión entre el *backend* y el *frontend*.

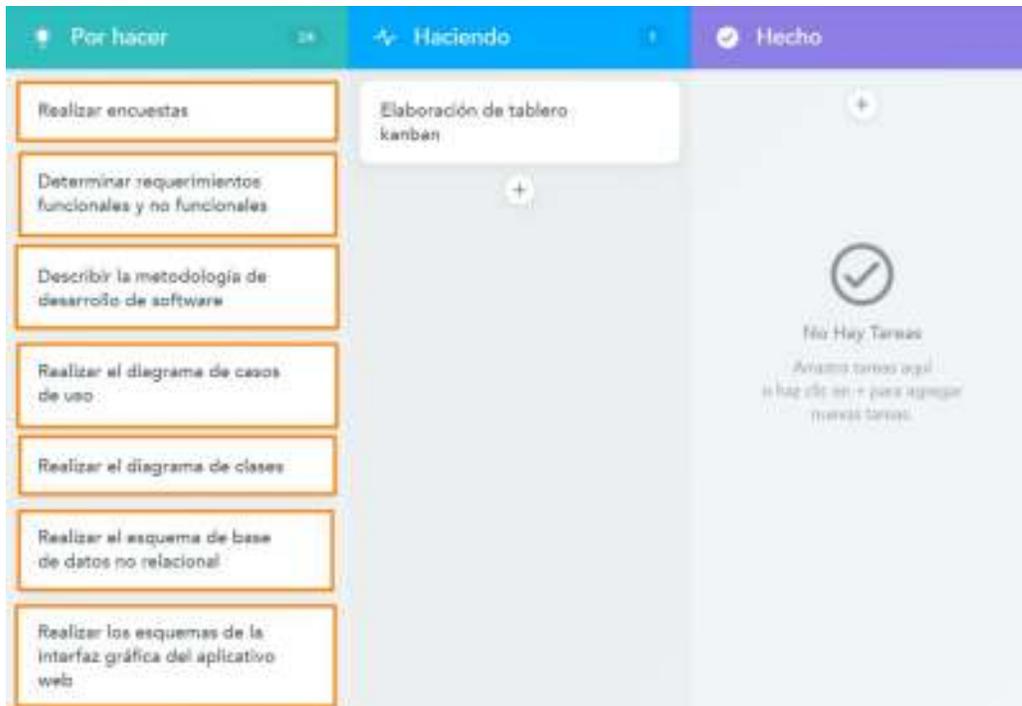


Figura 2.1 Tablero Kanban - Etapa de diseño del sistema - Parte 1

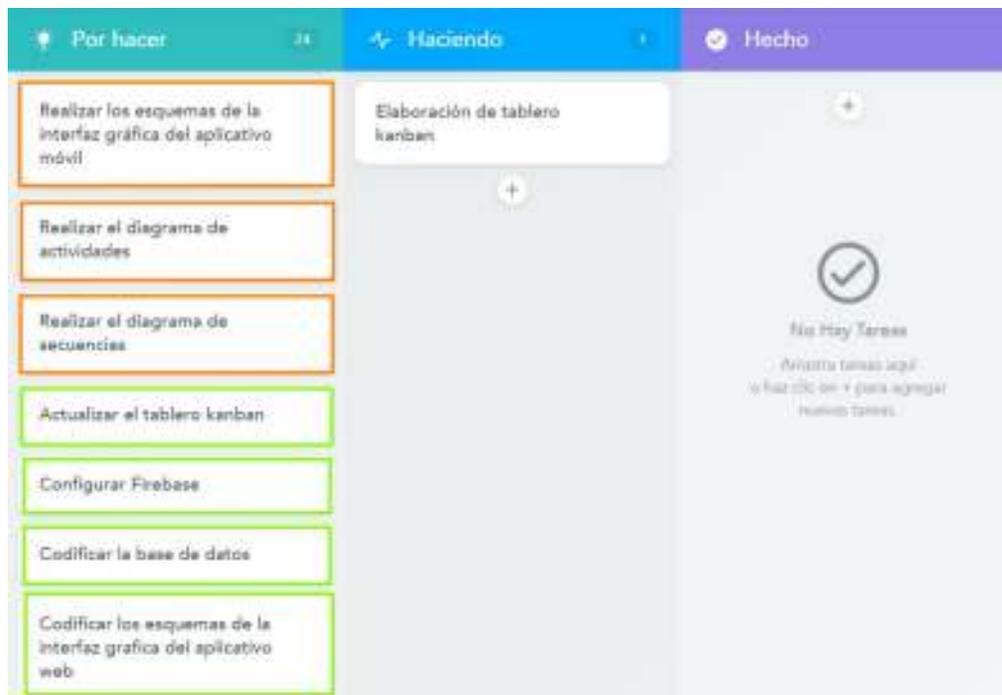


Figura 2.1 Tablero Kanban - Etapa de diseño del sistema – Parte 2

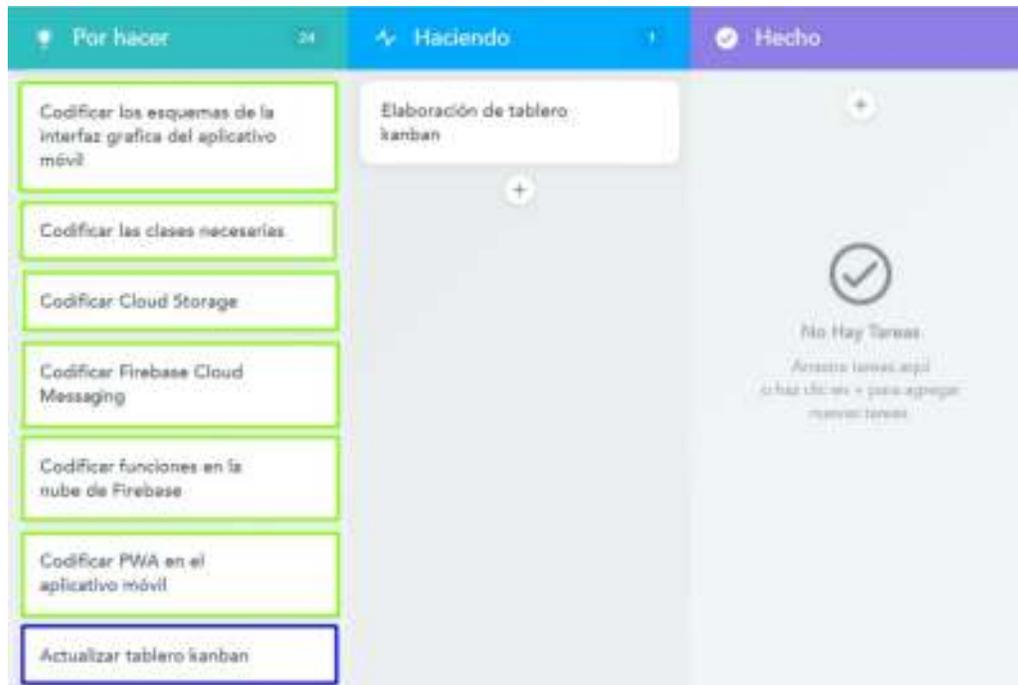


Figura 2.1. Tablero Kanban - Etapa de diseño del sistema – Parte 3

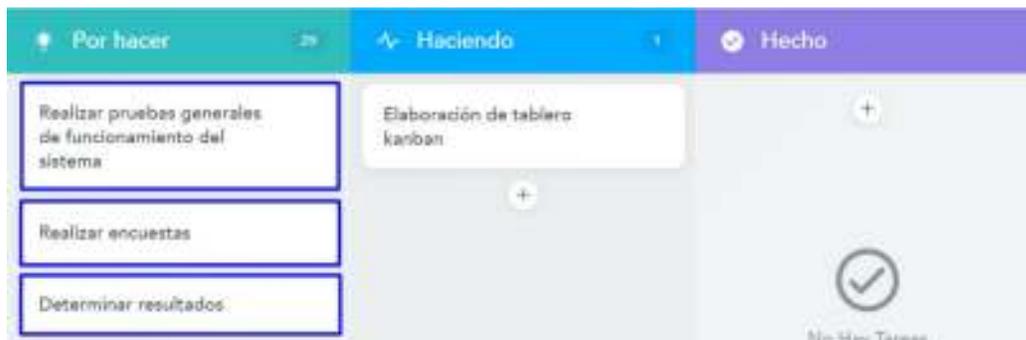


Figura 2.1. Tablero Kanban - Etapa de diseño del sistema – Parte 4

2.1.2 REQUERIMIENTOS GENERALES

Los requerimientos del sistema fueron determinados por medio de los datos obtenidos mediante encuestas realizadas a 10 personas dentro de la empresa HOMATELCO. El formato de la encuesta realizada se encuentra en el ANEXO A.

En la Tabla 2.1 se indica la tabulación de resultados de las encuestas realizadas. Se observa como datos relevantes, que el 100% de usuarios encuestados consideran que es útil tener un software para el control y administración de proyectos y un 80% de usuarios concuerdan que este software no debe estar ligado a una computadora específica y a su sistema operativo. Además, se determina que este software debe tener funcionalidades

específicas. El 100% de usuarios encuestados concuerdan que la formación de un equipo de trabajo y verificación de tareas es importante.

Se establece que el 90% de usuarios consideran que la comunicación entre usuarios es útil. El 100% de usuarios encuestados concuerdan que una versión móvil del software es importante y que ésta cuente con las mismas funcionalidades de la aplicación web.

Tabla 2.1. Resumen de los resultados de las encuestas

N.	Preguntas	Opciones	%
1.	¿En el manejo de proyectos, es útil tener un software que permita controlar y administrar los proyectos?	SI	100
		NO	0
2.	¿El software debe estar ligado a una computadora específica y a su sistema operativo?	SI	20
		NO	80
3.	¿Los usuarios deben tener una cuenta para acceder al software?	SI	100
		NO	0
4.	¿Considera usted que realizar un registro con su correo electrónico es bueno?	SI	70
		NO	30
5.	¿Dentro de un software es importante el manejo de roles y permisos para cada usuario?	SI	80
		NO	20
6.	¿Considera usted que es importante conformar un equipo de trabajo para la realización de un proyecto?	SI	100
		NO	0
7.	¿Para el control de proyectos es necesario verificar si las tareas fueron o no cumplidas?	SI	100
		NO	0
8.	¿Considera usted que la comunicación entre usuarios es útil dentro de un software de control y administración de proyectos?	SI	90
		NO	10
9.	¿Considera usted que el manejo de notificaciones es útil dentro de un software de control y administración de proyectos?	SI	90
		NO	10
10.	¿Además de una aplicación web, es útil tener una versión móvil?	SI	100
		NO	0
11.	¿Considera usted que la aplicación móvil debe tener las mismas funcionalidades que una aplicación web?	SI	100
		NO	0

Luego de analizar los resultados obtenidos en las encuestas y entrevistas, se han determinado los requerimientos funcionales y no funcionales que requiere el sistema. Adicionalmente para la determinación de los requerimientos funcionales se ha empleado igualmente la información de otros software utilizados en el manejo de proyectos, así como la teoría y bibliografía recibida en la asignatura de Formulación de Proyectos. A continuación, se indican dichos requerimientos.

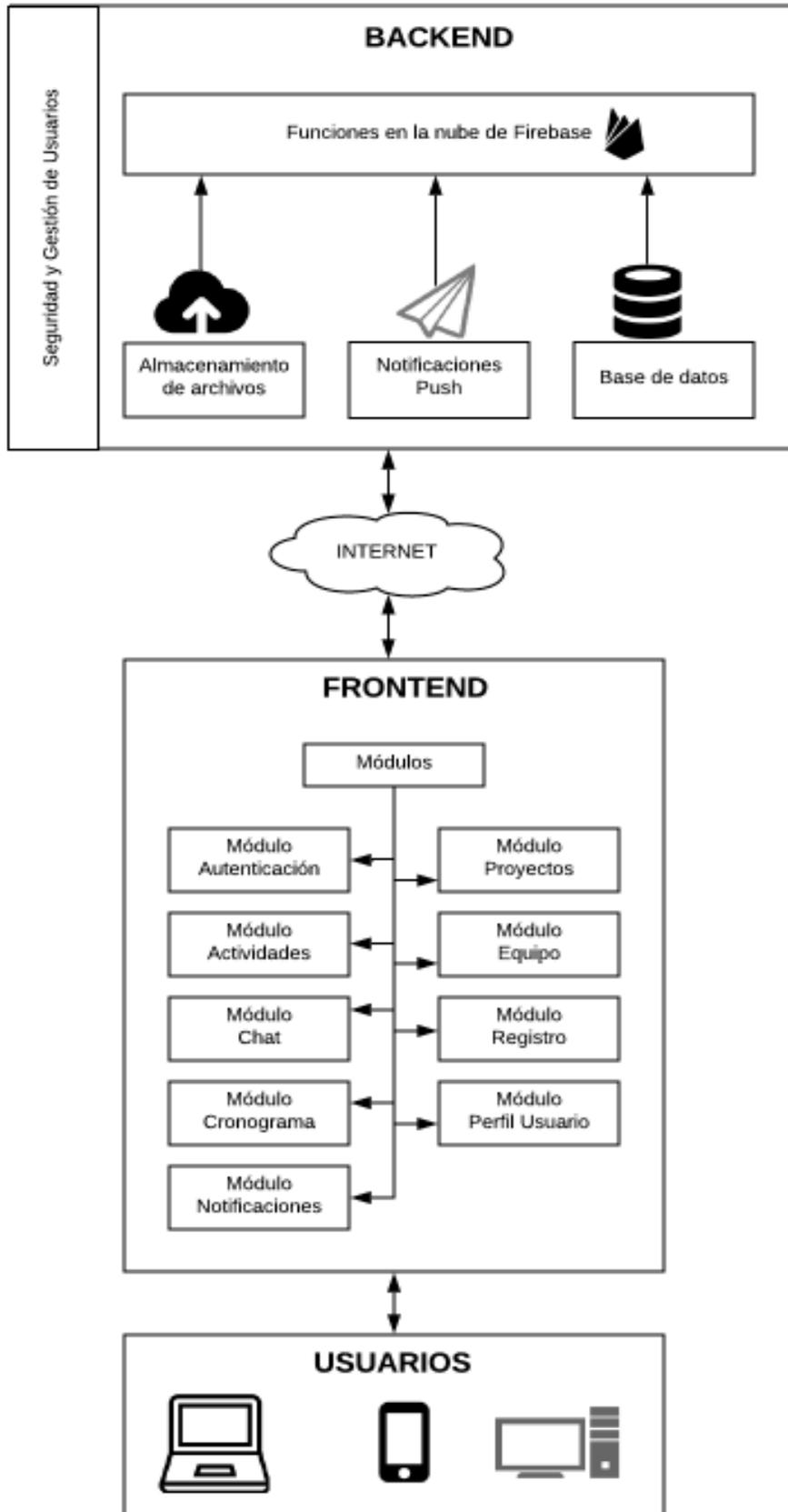


Figura 2.2. Diagrama de estructura del Sistema desarrollado

2.1.2.1 Requerimientos funcionales

Se han determinado los siguientes requerimientos funcionales:

- El usuario deberá crear una cuenta para acceder al sistema y a las funciones que éste proporciona.
- El sistema debe contar con pantallas y botones que permitan el control y administración de proyectos.
- El sistema debe autenticar a un usuario por medio de su correo electrónico.
- El sistema permitirá que el usuario se registre con un rol específico.
- El sistema proporcionará una guía de usuario.
- El sistema permitirá que el usuario pueda modificar su información de registro y agregar información adicional.
- El sistema permitirá crear proyectos.
- El sistema permitirá editar actividades.
- El sistema permitirá eliminar tareas.
- El sistema permitirá que un usuario le asigne tareas a otro usuario.
- El sistema permitirá la carga y visualización de evidencia de cada tarea.
- El sistema permitirá la comunicación entre usuarios, con la implementación de un *chat*.
- El sistema permitirá que el usuario reciba notificaciones en tiempo real cuando ocurran eventos significativos.

2.1.2.2 Requerimientos no funcionales

Se han determinado los siguientes requerimientos no funcionales:

- Crear una aplicación web con el *framework* Angular.
- Crear una aplicación móvil Android con el *framework* Ionic.
- Utilizar los servicios de Firebase.

- El sistema debe contar con una interfaz sencilla e intuitiva.

2.1.3 ROLES Y PERMISOS

Luego de analizar los requerimientos funcionales, se determinan los roles y permisos para los actores del sistema.

2.1.3.1 Actores

Se tienen dos tipos de actores:

- Usuario no registrado
- Usuario registrado como Gestor de proyectos
- Usuario registrado como Delegado

2.1.3.2 Roles

Se definen los siguientes roles para la interacción con el sistema:

- **Gestor:** es el encargado de gestionar los proyectos, generar acciones CRUD de proyectos, actividades y tareas, asignar tareas a cada Delegado que es parte de su equipo de trabajo.
- **Delegado:** es la persona que se encarga de cumplir las tareas que fueron asignadas por parte de su Gestor; proporciona la evidencia necesaria para verificar si las tareas fueron o no cumplidas.

2.1.3.3 Permisos

De acuerdo con los roles establecidos, se tienen dos tipos de permisos:

- **Permiso-Gestor:** tiene el control de los proyectos, es capaz de seleccionar su equipo de trabajo, realizar acciones CRUD de proyectos, actividades y tareas, y finalmente de verificar si las tareas fueron o no cumplidas.
- **Permiso-Delegado:** los permisos solo se limitan a subir evidencias y colocar un porcentaje de avance de cada tarea asignada.

2.1.4 DIAGRAMA DE CASOS DE USO

Luego de establecer roles y permisos se determinan los actores que son parte de la interacción con el sistema y se realiza un diagrama de casos de uso para cada actor, los cuales están basados en los requerimientos funcionales determinados mediante los distintos métodos anteriormente mencionados.

A continuación, se indican los diagramas de casos de uso para cada usuario registrado.

El registro le permite al usuario formar parte del sistema, como un Gestor o como un Delegado, cada uno tendrá sus correspondientes acciones. Ambos usuarios tienen algunas acciones en común, tales como: iniciar sesión, acceder a una guía de usuario, visualizar gráficos estadísticos, personalizar su perfil, comunicación entre usuarios y recibir notificaciones.

En la Figura 2.3 se indica el diagrama de caso de uso de un usuario que se encuentra registrado como Gestor de proyectos. Este usuario, además de realizar las acciones antes mencionadas, realiza acciones que difieren de un Delegado, tales como: seleccionar un equipo de trabajo, CRUD (*Create, Read, Update and Delete*) de proyecto, de actividades y de tareas, asignar tareas a un Delegado, visualizar evidencias y visualizar proyectos de próxima entrega.

En la Figura 2.4 se indica el diagrama de caso de uso de un usuario que se encuentra registrado como Delegado. Este usuario, además, de realizar las acciones comunes antes mencionadas, realiza acciones que difieren de un Gestor, tales como: visualizar tareas asignadas, visualizar a sus Gestores y compañeros de equipo, cargar evidencias y cargar porcentaje de avance por tarea.



Figura 2.3. Caso de uso: Usuario registrado como Gestor de proyectos

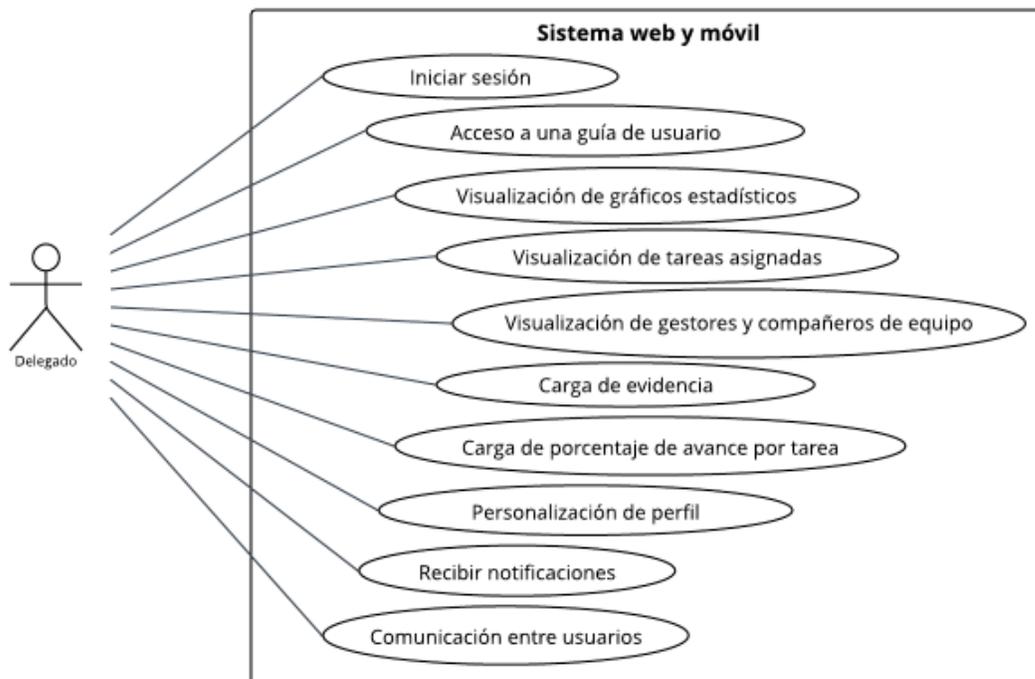


Figura 2.4. Caso de uso: Usuario registrado como Delegado

2.1.5 ESQUEMAS DE LA INTERFAZ GRÁFICA

Dentro del diseño de la interfaz gráfica, el sistema cuenta con distintos módulos que permiten la interacción con el usuario, y su funcionalidad depende del rol que tenga. A continuación, se indican los módulos que presenta el sistema.

Módulo Registro: Permite a los usuarios crear una cuenta con los parámetros: nombre completo, correo electrónico, contraseña, número celular, función que desempeña (rol Gestor o rol Delegado), uso de la aplicación. Con esta cuenta se podrá acceder a las aplicaciones.

Módulo Autenticación: Permite iniciar sesión con las credenciales antes creadas (Correo electrónico y contraseña).

Módulo Proyectos: Si el usuario se ha registrado como Gestor, este módulo permite la interacción con los proyectos, y la realización de un CRUD de los mismos. Cada proyecto contará con sus respectivas actividades y cada actividad con sus respectivas tareas, que son las que se les asignará un encargado para su cumplimiento. Este encargado no es nada más, que un usuario con rol Delegado que pertenece al equipo de trabajo que el Gestor seleccionó en el módulo equipo.

Si el usuario se ha registrado como Delegado, este módulo permite visualizar las tareas que su Gestor le ha asignado. La función de un Delegado dentro de este módulo es dar cumplimiento a estas tareas, proporcionando un porcentaje de avance y una evidencia para que el Gestor pueda verificar si la tarea fue cumplida o no.

Módulo Actividades: Si el usuario se ha registrado con un rol Gestor, este módulo permite la creación de actividades propias del proyecto con los atributos nombre, fecha de inicio, fecha de fin, y descripción, así también permite la creación de tareas con los atributos nombre, fecha de inicio, fecha de fin, descripción y encargado. Si el usuario se ha registrado con un rol Delegado, la funcionalidad de este módulo no será necesario ya que en el módulo Proyectos tendrá todas las tareas a las que ha sido asignado.

Módulo Cronograma: Si el usuario se ha registrado como Gestor, este módulo permitirá visualizar las fechas de entrega de todos los proyectos existentes. Además, cada fecha indicará los días restantes para la culminación del proyecto mediante el uso de colores. Si

el usuario se ha registrado como Delegado, este módulo permitirá visualizar las fechas de entrega de todas las tareas a las que ha sido asignado.

Módulo Equipo: Si el usuario se ha registrado como Gestor, este módulo permite crear su grupo de trabajo, con el cual se va a poder comunicar, enviarles recursos necesarios para el trabajo y asignarle las tareas correspondientes. El Gestor selecciona su grupo de trabajo de los usuarios que se ha registrado con un rol Delegado.

Si el usuario se ha registrado como Delegado, este módulo permitirá al usuario visualizar la lista de sus Gestores, es decir los usuarios con rol Gestor que lo han seleccionado para formar parte de su equipo de trabajo. Además, podrá visualizar la lista de sus compañeros de equipo, es decir aquellos usuarios que son parte del equipo de trabajo al que pertenece. Finalmente podrá recibir, enviar y descargar recursos, así también podrá enviar comentarios al equipo al que pertenece.

Módulo Chat: Si el usuario se ha registrado como Gestor, este módulo permitirá la comunicación entre los usuarios que estén dentro de su equipo de trabajo. Si el usuario se ha registrado como Delegado, este módulo permitirá comunicarse con sus Gestores y compañeros de equipo.

Módulo Perfil Usuario: Permite editar la información de cada usuario (Gestor o Delegado).

Módulo Notificaciones: Permite visualizar notificaciones que se generarán de manera automática acerca de los tiempos restantes de un proyecto, actividades asignadas y culminación del proyecto dependiendo del rol del usuario.

El diseño de las aplicaciones se genera mediante la herramienta Balsamiq; se elaboran plantillas (*sketches*) para la aplicación web y plantillas (*mockups*) para la aplicación móvil Android.

Es necesario mencionar que, para el diseño del sistema, se ha hecho uso de cuadros de diálogo que se sobrepone en las diferentes pantallas ya sea web y móvil, y que de aquí en adelante serán mencionados como modales.

Por otro lado, como parte del diseño del sistema se ha determinado un nombre y un logo. El nombre que se ha otorgado es GuGo y en la Figura 2.5 se muestra el logo que se ha adoptado para este trabajo.



Figura 2.5. Logo del sistema

Este proyecto se encuentra principalmente orientado hacia las empresas, por ello, la empresa que requiera el uso del sistema deberá contactarse con el personal autorizado mediante la opción **Contactarse con soporte**, con el fin de proporcionar la lista de usuarios que serán parte del sistema y especificar el rol de cada uno.

La persona autorizada, registrará a los usuarios dentro de la base de datos, con ello cuando un usuario haga su registro en el sistema, se verificará si este usuario existe dentro de esta base de datos y si el rol indicado es el correcto.

Si bien el proyecto se encuentra orientado al uso empresarial, se ha agregado un uso personal, el cual permitirá que un usuario se registre en el sistema sin necesidad de que el personal autorizado lo registre en la base de datos.

Por lo tanto, se definen dos usos del sistema: uso empresarial y uso personal. Dentro del sistema si el uso es empresarial, el usuario podrá relacionarse solo con usuarios pertenecientes a la misma empresa, mientras que, si el uso es personal, el usuario podrá relacionarse con todos los usuarios que tengan su cuenta GuGo de uso personal y no empresarial. Es importante mencionar que, para ambos usos, la aplicación brinda las mismas funcionalidades de acuerdo con el rol registrado.

2.1.5.1 Aplicación web

A continuación, se presentan los *sketches* de la aplicación y se hace una explicación de la interacción entre el usuario y cada uno de éstos, así como la interacción entre algunos *sketches*. El manejo y permisos de la aplicación dependerán del rol de usuario, por lo que para cada *sketch* se detalla la funcionalidad de la aplicación dependiendo su rol o si la funcionalidad es la misma para ambos roles.

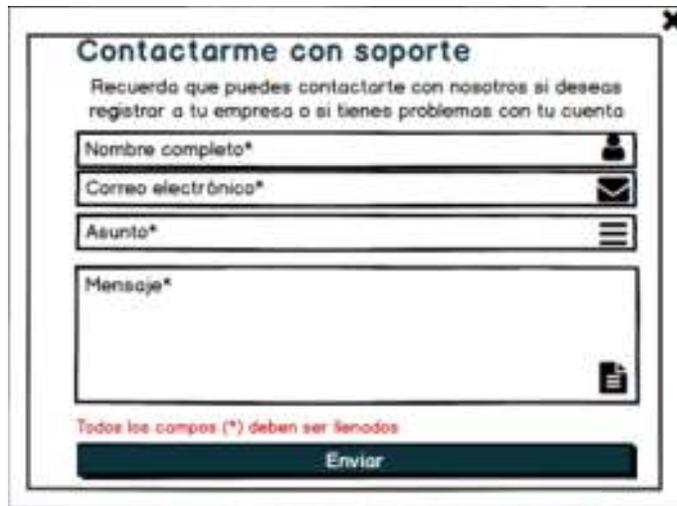
Cuando un usuario accede a la aplicación GuGo Web, ésta en primera instancia mostrará la pantalla Login como se indica en la Figura 2.6; el usuario puede ingresar a la aplicación mediante sus credenciales: correo electrónico y contraseña. También cuenta con las opciones: Obtener cuenta GuGo, Contactarse con soporte y ¿Haz olvidado tu contraseña?.

En la Figura 2.7 se indica el modal que aparece mediante la opción **Contactarse con soporte**, aquí la empresa que desee hacer uso del sistema podrá enviar un mensaje hacia el personal autorizado para registrar su empresa y lista de usuarios.

En la Figura 2.8 se indica el modal Recuperar mi contraseña, éste aparece mediante la opción **¿Has olvidado tu contraseña?** y permite al usuario colocar su correo electrónico con el fin de enviarle un enlace, el cual permitirá que el usuario coloque una nueva contraseña.



Figura 2.6. Sketch de la pantalla Login



Modal titled "Contactarme con soporte". It includes a header with the title and a sub-header: "Recuerda que puedes contactarte con nosotros si deseas registrar a tu empresa o si tienes problemas con tu cuenta". Below this are four input fields: "Nombre completo*" with a person icon, "Correo electrónico*" with an envelope icon, "Asunto*" with a list icon, and "Mensaje*" with a document icon. A red error message at the bottom states "Todos los campos (*) deben ser llenados". A dark blue "Enviar" button is at the bottom.

Figura 2.7. Modal Soporte al usuario



Modal titled "Recuperar mi contraseña". It features a single input field labeled "Ingrese el correo electrónico" and a dark blue "Listo" button below it.

Figura 2.8. Modal Recuperar contraseña

Si el usuario no posee una cuenta, puede acceder a la pantalla Registro mediante la opción **Obtener cuenta GuGo**. Esta pantalla pide al usuario ingresar los datos: nombre completo, correo electrónico, contraseña, número celular, y finalmente el usuario deberá seleccionar su rol dentro del sistema y el uso del mismo. Todo lo mencionado se indica en la Figura 2.9.

Cuando el usuario ingresa a la aplicación mediante el uso de sus credenciales, se le mostrará la pantalla Inicio. Esta pantalla, cuenta con un menú lateral izquierdo desplegable y una cabecera que contiene dos iconos: el icono Perfil y el icono Notificaciones. Si el usuario Gestor o Delegado ya cuenta con proyectos o tareas asignadas respectivamente, la aplicación proporcionará gráficos estadísticos acerca del estado de sus proyectos o tareas, como se indica en la Figura 2.10.

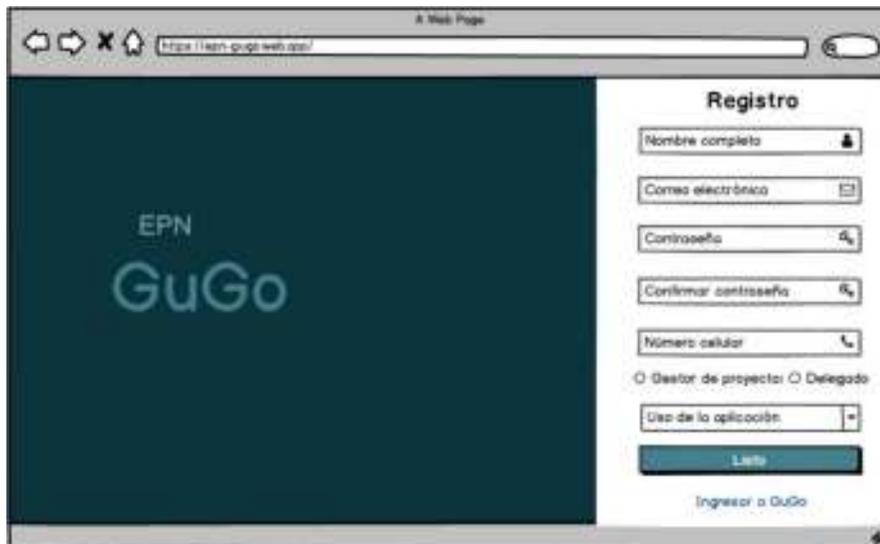


Figura 2.9. Sketch de la pantalla Registro



Figura 2.10. Sketch de la pantalla Inicio – Gráficos estadísticos

Si el usuario Gestor o Delegado ingresa a la aplicación y no cuenta con proyectos o tareas respectivamente, se le proporcionará una guía de usuario con una explicación de cómo utilizar la aplicación según su rol, tal como se indica en la Figura 2.11.

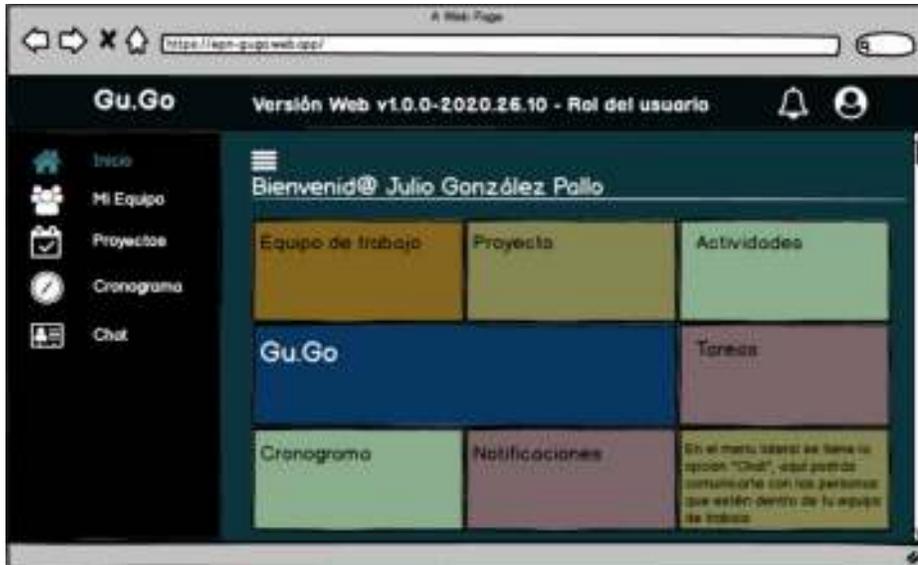


Figura 2.11. Sketch de la pantalla Inicio – Guía de usuario

En la Figura 2.12 se indica el *sketch* de los íconos de la cabecera, los cuales estarán presentes siempre dentro de la aplicación y son atajos que facilitan la funcionalidad de ésta. El ícono Notificaciones indica las notificaciones que recibirá el usuario según los eventos generados, con la opción **Ver todas**. El ícono Perfil indica información básica acerca del usuario, con dos opciones: **Editar perfil** y **Cerrar sesión**.



Figura 2.12. Sketch de los íconos de la cabecera

La opción **Editar perfil** redirige al usuario a la pantalla Perfil. En la Figura 2.13 se indica el *sketch* de la pantalla Perfil, aquí el usuario podrá modificar sus datos de registro, agregar

una imagen de perfil e incluso agregar nuevos datos a su perfil, como Género y Fecha de nacimiento.

La opción **Ver todas**, dentro del ícono Notificaciones, redirige al usuario a la pantalla Notificaciones. En la Figura 2.14 se indica el *sketch* de la pantalla Notificaciones; aquí el usuario podrá visualizar los detalles de cada notificación que reciba y mediante la opción **Ver notificaciones** podrá ir directamente a la pantalla que generó dicha notificación.



Figura 2.13. Sketch de la pantalla Perfil



Figura 2.14. Sketch de la pantalla Notificaciones

En la Figura 2.15 se indica el *sketch* de la pantalla Mi Equipo para el rol Gestor, aquí se tienen tres pestañas: Seleccionar Equipo, Editar Equipo y Ver Equipo por proyecto. En la pestaña **Seleccionar Equipo** podrá escoger a los usuarios con rol Delegado que formarán parte de su equipo de trabajo.

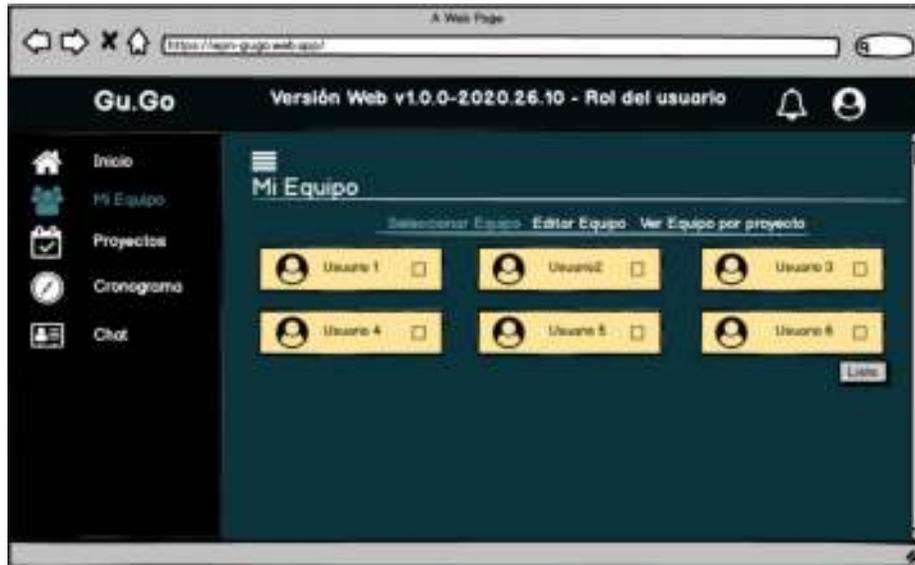


Figura 2.15. Sketch de la pantalla Mi Equipo para el rol Gestor

Una vez seleccionados los usuarios, en la pestaña **Editar Equipo** se indicará la lista de usuarios pertenecientes al equipo de trabajo. El Gestor podrá eliminar algún usuario de su equipo de trabajo mediante la opción **Eliminar usuario**, siempre y cuando el usuario seleccionado no tenga tareas a su cargo; esto se indica en la Figura 2.16. Además, en la pestaña **Ver Equipo por proyectos**, se tiene una visualización de todos los proyectos con su respectiva lista de equipo de trabajo; cada proyecto tiene la opción **Ver recursos** como se indica en la Figura 2.17.

En la Figura 2.18 se indica el *sketch* de la pantalla Mi Equipo para el rol Delegado; aquí se tienen dos pestañas: Ver Equipo y Ver Equipo por proyecto. En la pestaña **Ver Equipo** podrá visualizar los Gestores que lo han seleccionado para ser parte de su equipo de trabajo y los compañeros que son parte del equipo de cada proyecto al que el Delegado pertenece. En la pestaña **Ver equipo por proyecto**, podrá visualizar los proyectos en los que forma parte para cumplir una tarea asignada por su Gestor. Cada proyecto indica su Gestor y equipo de trabajo, y cuenta con la opción **Ver Recursos** como se indica en la Figura 2.19.

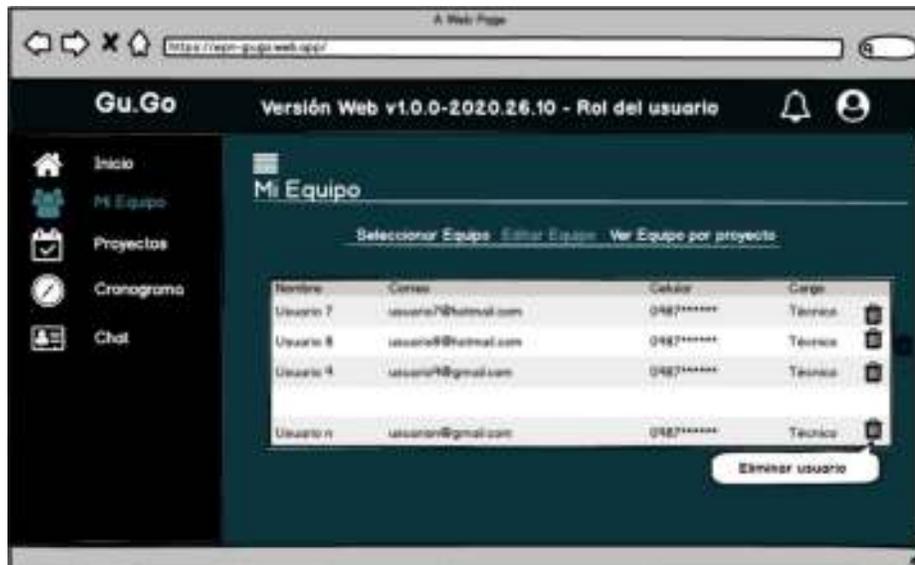


Figura 2.16. Sketch de la pantalla Mi Equipo para el rol Gestor - Pestaña Editar Equipo



Figura 2.17. Sketch de la pantalla Mi Equipo para el rol Gestor - Pestaña Ver Equipo por proyecto

La opción **Ver recursos** tiene la misma funcionalidad para ambos roles, pues la comunicación es bidireccional, entre Gestor y su equipo de trabajo. En la Figura 2.20 se indica el *Sketch* del modal Recursos; en la parte (a) se indica la pestaña Observaciones y en la parte (b) se indica la pestaña Archivos. Cada pestaña permite al usuario enviar-recibir mensajes y cargar-descargar archivos respectivamente.

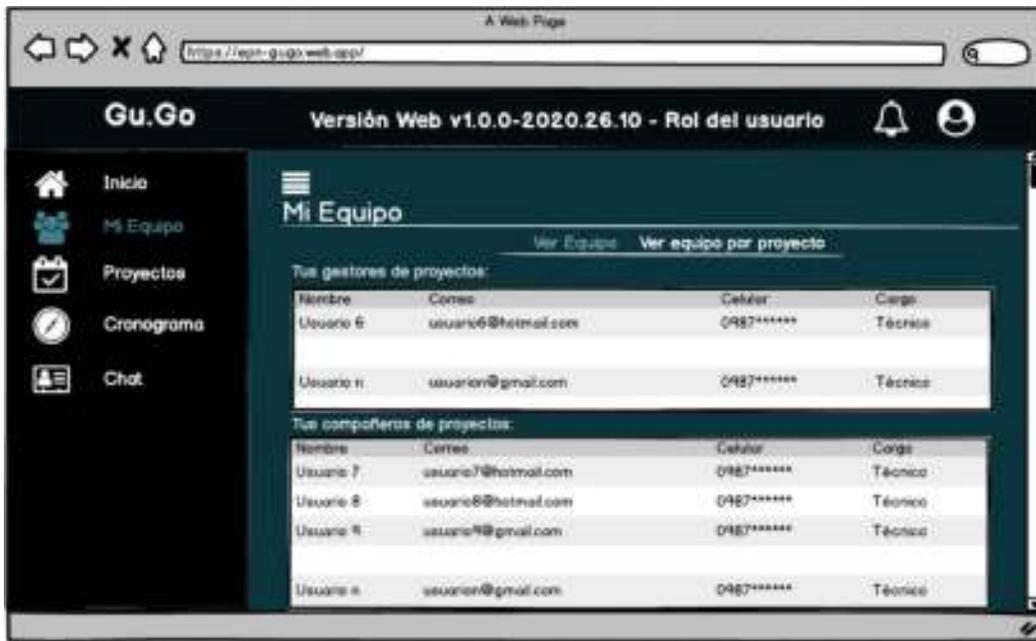
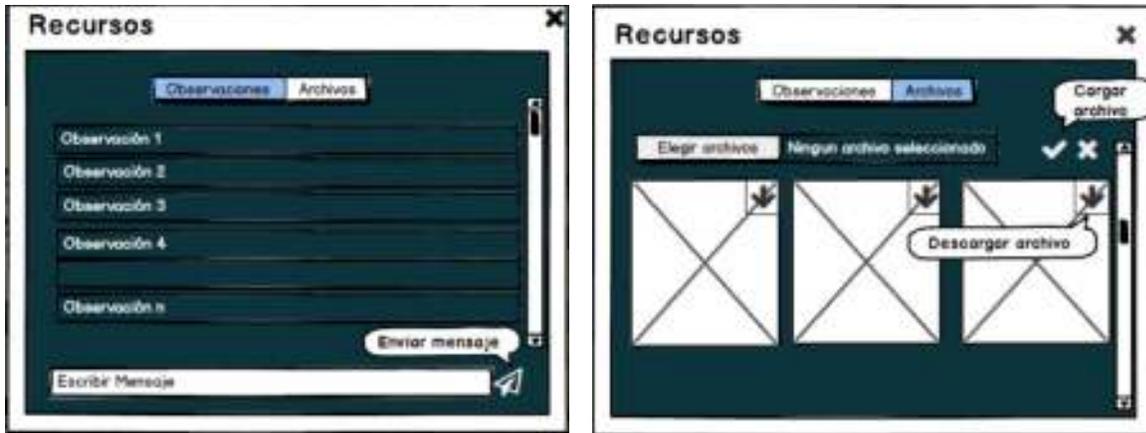


Figura 2.18. Sketch de la pantalla Mi Equipo para el rol Delegado – Ver Equipo



Figura 2.19. Sketch de la pantalla Mi Equipo para el rol Delegado - Pestaña Ver equipo por proyecto



(a)

(b)

Figura 2.20. Sketch del modal Recursos (a) Pestaña Observaciones (b) Pestaña Archivos

En la Figura 2.21 se indica el *sketch* de la pantalla Proyectos para el rol Gestor, aquí se indica la lista de proyectos existentes. Cada proyecto cuenta con las opciones: Editar proyecto, Ver actividades y Eliminar proyecto, las cuales serán explicadas más adelante con sus respectivos *sketches*. Además, el Gestor podrá crear nuevos Proyectos mediante el botón Nuevo proyecto. El formulario de Nuevo proyecto aparecerá en forma de un modal, el cual exige los atributos obligatorios: Nombre, Cliente, Tipo, Fecha de inicio, Fecha de fin y el atributo opcional Descripción, como se indica en la Figura 2.22.

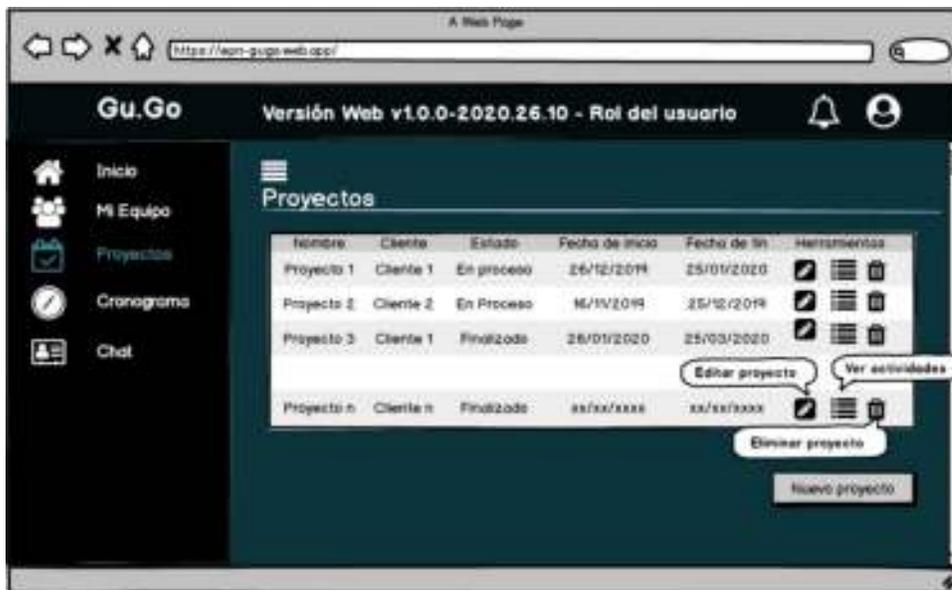


Figura 2.21. Sketch de la pantalla Proyectos para el rol Gestor



Figura 2.22. Sketch del modal Proyecto

En la Figura 2.23 se indica el *sketch* de la pantalla Proyectos para el rol Delegado, aquí podrá visualizar la lista de todas las tareas que sus Gestores le han asignado. El Delegado tiene la opción **Subir evidencia**, ésta indica el modal Evidencia, el mismo que cuenta con dos pestañas: la pestaña Archivos y la pestaña Observaciones.

En la parte (a) de la Figura 2.24 se indica la pestaña Archivos, donde el Delegado tendrá la opción **Cargar archivos** y podrá subir una lista de archivos que comprueben la veracidad del cumplimiento de la tarea asignada; con la opción **Indicar porcentaje** podrá ingresar el porcentaje de avance en el que se encuentra la tarea. En la parte (b) de la Figura 2.24 se indica la pestaña Observaciones, donde se podrán visualizar los mensajes emitidos por su Gestor y que servirán como observaciones acerca de la evidencia proporcionada.

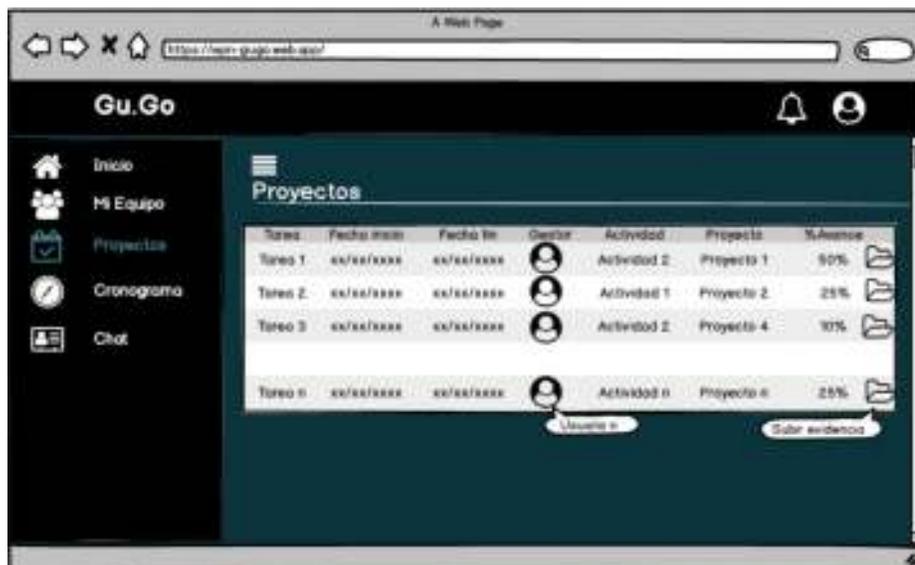


Figura 2.23. Sketch de la pantalla Proyectos para el rol Delegado



(a)

(b)

Figura 2.24. Sketch del modal Evidencia para el rol Delegado (a) Pestaña Archivos y (b) Pestaña Observaciones

En la Figura 2.25 se indica el *sketch* de la pantalla Cronograma para el rol Gestor, aquí podrá visualizar la lista de proyectos ordenados según su fecha de entrega (Fecha de fin). Además, cada fecha indicará los días restantes para la culminación del proyecto. La opción **Ver calendario** proporcionará la lista de tareas correspondientes a cada proyecto, estas tareas estarán organizadas por fechas de inicio y fin e indicará el porcentaje de avance de cada una.

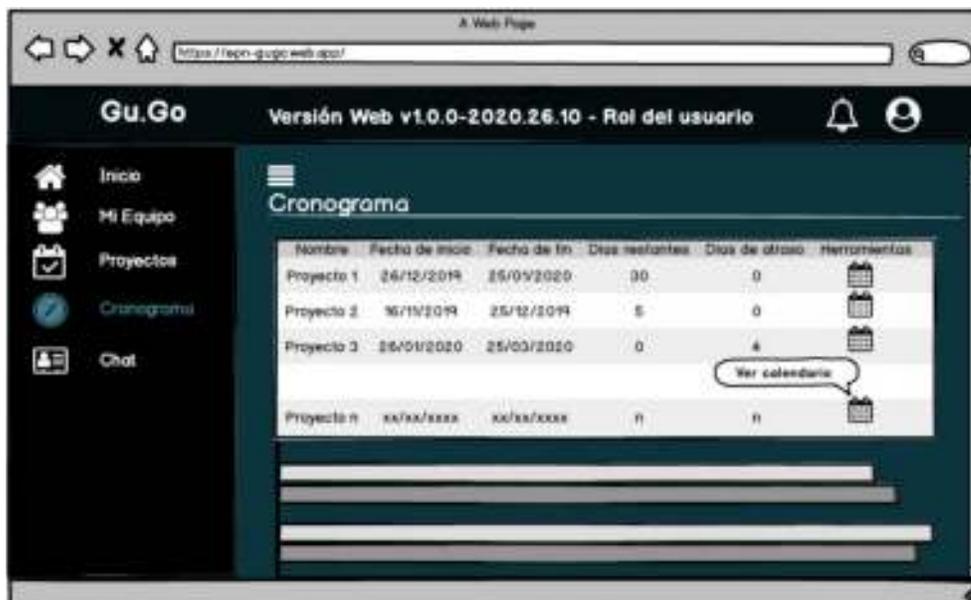


Figura 2.25. Sketch de la pantalla Cronograma para el rol Gestor

En la Figura 2.26 se indica el *sketch* de la pantalla Cronograma para el rol Delegado, aquí podrá visualizar la lista de tareas a las que ha sido asignado, las mismas que están ordenadas según su fecha de entrega (Fecha de fin). Además, cada fecha indicará los días restantes para la culminación de la tarea.

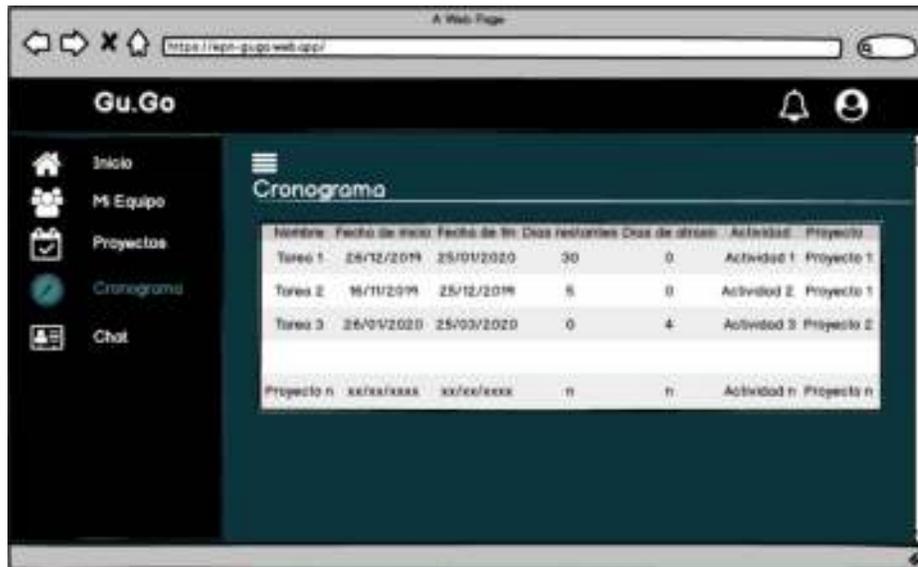


Figura 2.26. Sketch de la pantalla Cronograma para el rol Delegado

En la Figura 2.27 se indica el *sketch* de la pantalla Chat, aquí el usuario sea un Gestor o un Delegado, podrá comunicarse con su equipo de trabajo o con sus compañeros de equipo y Gestores respectivamente.



Figura 2.27. Sketch de la pantalla Chat

En la Figura 2.21 se mostró la opción **Ver actividades**, ésta redirige a la pantalla Actividades la cual se indica en la Figura 2.28. Aquí el Gestor podrá visualizar los detalles de un proyecto específico, como son: su lista de actividades y lista de tareas correspondientes a cada actividad y agregar actividades mediante la opción **Agregar actividad**. Cada actividad posee una lista de tareas, por lo que, para una mejor experiencia visual, cada actividad se presenta de forma desplegable. Cada vez que se despliega una actividad ésta presenta su lista de tareas correspondiente, como se indica en la Figura 2.29.

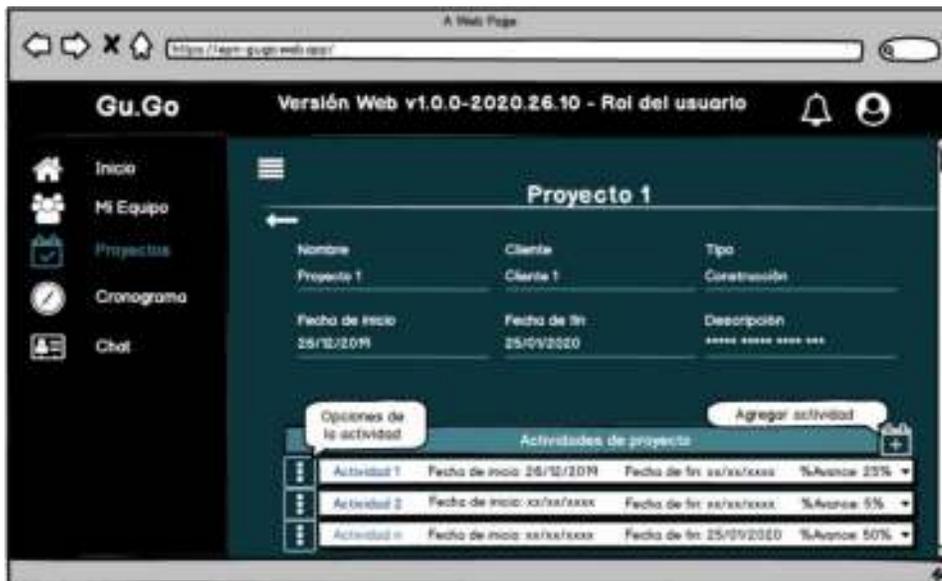


Figura 2.28. Sketch de la pantalla Actividades

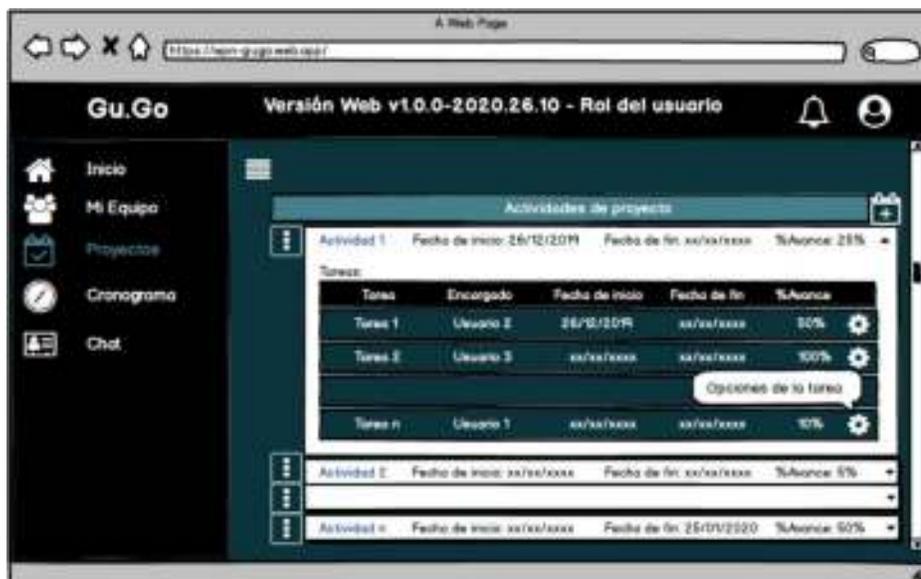


Figura 2.29. Sketch de la pantalla Actividades - Actividades desplegables

Cada actividad y cada tarea presenta sus respectivas opciones; para la actividad se tiene las opciones Editar actividad, Eliminar actividad y Agregar tarea, y cada tarea tiene las opciones: Editar tarea, Eliminar tarea, Ver Evidencia. Todo lo descrito se indica en la Figura 2.30.



Figura 2.30. Sketch de la pantalla Actividades – Opciones de actividades y tareas

Para las opciones **Agregar actividad** y **Agregar tarea** se hace uso de modales. En la Figura 2.31 se indica el *sketch* del modal Actividad, aquí el Gestor podrá crear una nueva actividad con los atributos Nombre, Fecha de inicio, Fecha de fin y Descripción. Las fechas de la actividad se encuentran limitadas dentro del rango de las fechas del proyecto al que pertenecen.

En la Figura 2.32 se indica el *sketch* del modal Tarea, aquí el Gestor podrá crear una nueva tarea, con los atributos Nombre, Fecha de inicio, Fecha de fin, Delegado y Descripción. Las fechas de las tareas se encuentran limitadas al rango de las fechas de la actividad a la que pertenecen. En el campo Delegado se debe seleccionar un usuario que pertenezca al equipo de trabajo que el Gestor previamente seleccionó.

Para las opciones **Editar proyecto**, **Editar actividad** y **Editar tarea**, se hace uso de los modales Proyecto, Actividad y Tarea respectivamente. El *sketch* del modal Proyecto se indicó en la Figura 2.22, en tanto que los correspondientes al de los modales Actividad y Tarea se presentan en las Figura 2.31 y Figura 2.32 respectivamente.

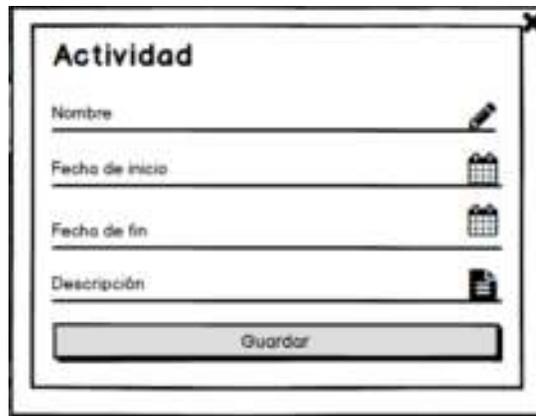


Figura 2.31. Sketch del modal Actividad

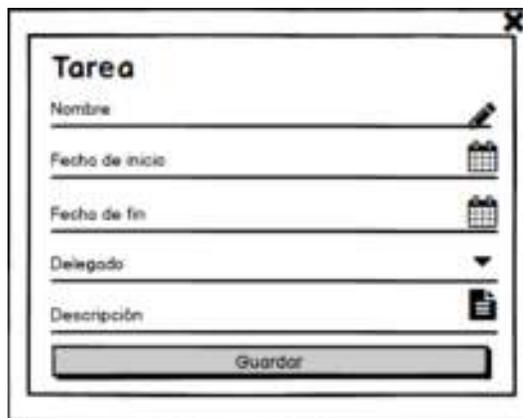
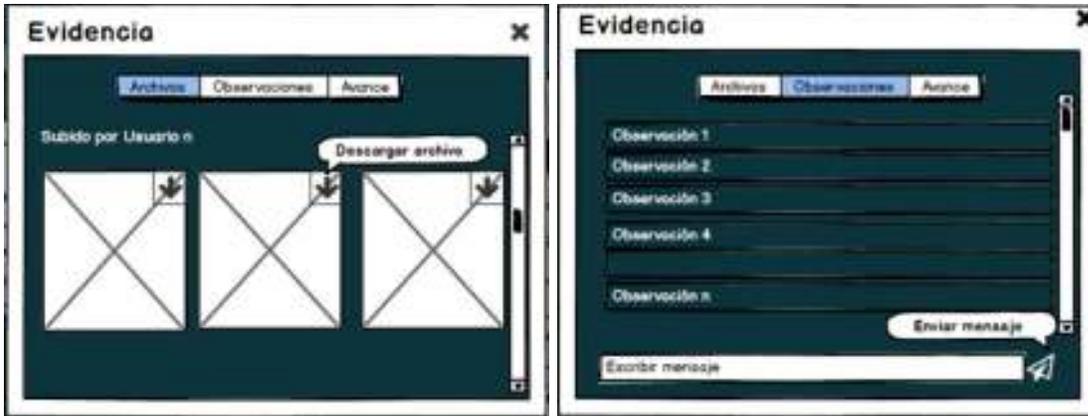


Figura 2.32. Sketch del modal Tarea

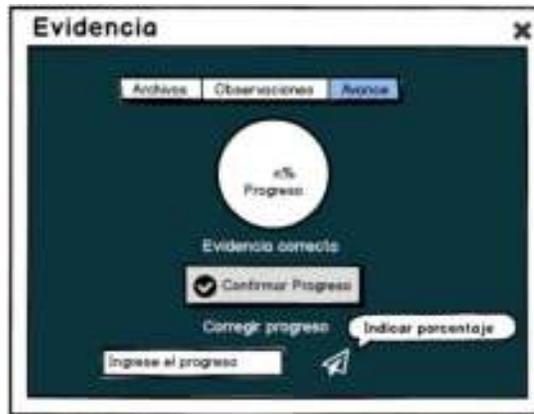
En la Figura 2.33 se indica el *sketch* del modal Evidencia para el rol Gestor; este modal aparece cuando se selecciona la opción **Ver Evidencia**, la cual cuenta con tres pestañas: Archivos, Observaciones y Avance. En la parte (a) se tiene la pestaña Archivos, aquí podrá visualizar la evidencia subida por su respectivo Delegado mediante el modal indicado en la Figura 2.24. Si el Gestor requiere visualizar cada evidencia, ésta ofrece un enlace mediante la opción **Descargar archivo**, la cual permitirá al Gestor descargar los archivos que desee.

En la parte (b) se tiene la pestaña Observaciones, el Gestor tendrá la opción **Enviar mensaje**, esto sirve como observaciones hacia el Delegado acerca de la evidencia proporcionada. En la parte (c) se tiene la pestaña Avance, mediante la opción **Confirmar progreso** el Gestor podrá aprobar el porcentaje de avance que proporcionará el Delegado, si considera que este valor coincide con la evidencia proporcionada. Si el Gestor considera que la evidencia no está acorde con el porcentaje de avance, tendrá la opción **Corregir progreso** para rectificar este valor y enviarlo hacia el Delegado.



(a)

(b)



(c)

Figura 2.33. Sketch del modal Evidencia para el rol Gestor (a) Pestaña Archivos (b) Pestaña Observaciones y (c) Pestaña Avance

Existen múltiples acciones que generan cuadros de alerta o de aviso. La acción Eliminar Proyecto, Actividad o Tarea generan un cuadro de alerta, permitiendo al usuario asegurarse de la acción que va a realizar. Cuando un usuario se registra o su autenticación es exitosa, la aplicación muestra un cuadro de aviso exitoso, al igual que cuando se crea, edita o elimina proyectos, actividades o tareas. En la Figura 2.34 se indica el *sketch* de un cuadro de Alerta y en Figura 2.35 se indica el *sketch* de un cuadro de Aviso.

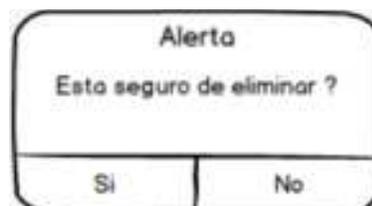


Figura 2.34. Sketch de un recuadro de Alerta

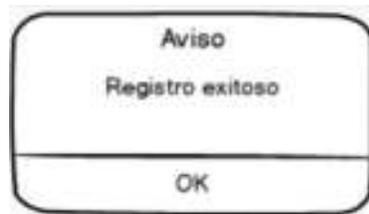


Figura 2.35. Sketch de un recuadro de Aviso

2.1.5.2 Aplicación móvil

La aplicación móvil Android desarrollada en este trabajo tiene la misma funcionalidad que la aplicación web, por lo que aquí se hará una descripción breve del diseño realizado. Para la aplicación web se maneja un menú lateral, mientras que para la aplicación móvil se manejan *tabs*, las mismas que proporcionan un componente de navegación de nivel inferior que permite una navegación basada en pestañas.

En la Figura 2.36 se indican los *mockups* para la autenticación del usuario. En la parte (a) se indica el *mockup* para el ingreso del usuario con sus respectivos atributos; en la parte (b) se indica el *mockup* del modal Contacto, el cual aparece mediante la opción **Contactarme con soporte**. En la parte (c) se indica el modal Recuperar contraseña el cual aparece mediante la opción **¿Has olvidado tu contraseña?**.

En la Figura 2.37 se indica el *mockup* de la pantalla Registro, el cual aparece mediante la opción **Obtener cuenta Gugo** y permite al usuario registrarse dentro del sistema y acceder al aplicativo móvil. Una vez que el usuario ingresa al aplicativo móvil se indica la pantalla Inicio; si el usuario tiene proyectos o tareas, aquí se indican gráficos estadísticos sobre el estado de éstos como se muestra en la parte (a) de la Figura 2.38. Si el usuario no tiene tareas ni proyectos, la aplicación proporciona una Guía de usuario como se muestra en la parte (b) de la Figura 2.38.

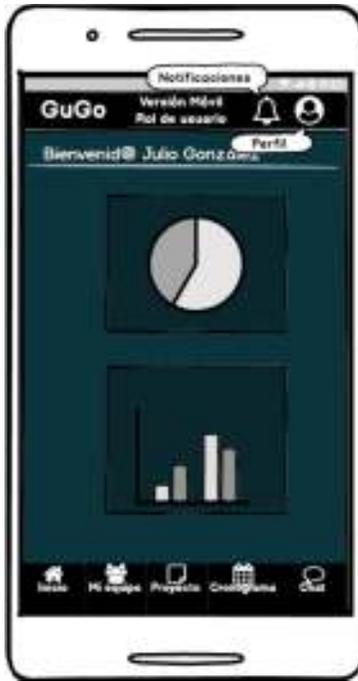
En la cabecera de la aplicación se tienen los iconos correspondientes a las Notificaciones y al Perfil de usuario como se indica en la parte (a) de la Figura 2.38. Cada uno de estos iconos redirige al usuario a su respectiva pantalla. En la parte (a) de la Figura 2.39 se indica el *mockup* de la pantalla Mi perfil, mientras que en la parte (b) se indica el *mockup* de la pantalla Notificaciones.



Figura 2.36. (a) Mockup de la pantalla Login (b) Mockup del modal Contacto y (c) Modal Recuperar contraseña



Figura 2.37. Mockup de la pantalla Registro



(a)



(b)

Figura 2.38. Mockup de la pantalla Inicio (a) Gráficos estadísticos y (b) Guía de usuario



(a)



(b)

Figura 2.39. Mockups de la pantalla (a) Mi Perfil y (b) Notificaciones

En la Figura 2.40 se presenta el *mockup* de la pantalla Mi Equipo para el rol Gestor; la parte (a) indica la pestaña Seleccionar Equipo, aquí el Gestor podrá seleccionar los Delegados que formarán parte de su Equipo. La parte (b) corresponde a la pestaña Editar Equipo, aquí visualizarán los integrantes que ya forman parte de su equipo de trabajo, al deslizar hacia la izquierda cada Delegado, se tiene la opción **Eliminar usuario**. Por último, en la parte (c) se indica la pestaña Ver Equipo por proyecto, aquí se podrá visualizar cada proyecto con su respectivo equipo de trabajo y se tiene la opción Ver recursos.

En la Figura 2.41 se indica el *mockup* de la pantalla Mi Equipo para el rol Delegado; la parte (a) corresponde a la pestaña Ver Equipo, aquí se podrán visualizar los Gestores y compañeros de trabajo. En la parte (b) se indica la pestaña Ver equipo por proyecto, aquí se podrán visualizar los proyectos a los que el Delegado pertenece y sus compañeros de equipo de cada proyecto y se tiene la opción Ver recursos.



(a)

(b)

(c)

Figura 2.40. Mockups de la pantalla Mi Equipo para el rol Gestor (a) Pestaña Seleccionar Equipo (b) Ver Equipo y (c) Ver Equipo por proyecto

La opción **Ver Recursos**, permite la aparición del modal Recursos el cual se utiliza para ambos roles; aquí el usuario podrá cargar y descargar archivos como se indica en la parte (b) de la Figura 2.42 y recibir y enviar mensajes hacia su Gestor y compañeros de equipo como se indica en la parte (a) de la Figura 2.42.



(a)



(b)

Figura 2.41. Mockups de la pantalla Mi Equipo para el rol Delegado (a) Ver Equipo y (b) Ver equipo por proyecto



(a)



(b)

Figura 2.42. Mockups del modal Recursos (a) Pestaña Observaciones y (b) Pestaña Archivos

En la parte (a) de la Figura 2.43 se indica el *mockup* de la pantalla Proyectos para el rol Gestor, aquí se presenta la lista de todos los proyectos existentes; se podrá crear nuevos proyectos mediante el botón Nuevo proyecto. Si se desliza cualquier proyecto hacia la izquierda, se proporcionan dos opciones: Editar proyecto y Eliminar proyecto. En la parte (b) de la Figura 2.43 se indica la interfaz del modal Proyecto, el cual se utiliza para crear y editar proyectos. Además, al dar *click* sobre cualquier proyecto, la aplicación redirige a la pantalla Actividades.



Figura 2.43. (a) Mockup de la pantalla Proyectos para el rol Gestor y (b) Mockup del modal Proyecto

En la Figura 2.44 se indican los *mockups* de la pantalla Actividades. En la parte (a) se presenta la pantalla que contiene los atributos del proyecto y su lista de actividades; para crear una nueva actividad se tiene la opción Agregar actividad. Al deslizar hacia la izquierda cada actividad, se proporcionan dos opciones: Editar actividad y Eliminar actividad, al deslizar hacia la derecha cada actividad, se proporciona la opción Agregar tarea.

Al dar *click* sobre cada actividad, se despliega su lista de tareas, como se indica en la parte (b) de la Figura 2.44. Al deslizar hacia la izquierda cada tarea, se proporcionan dos opciones: Editar Tarea y Eliminar Tarea, y al deslizar hacia la derecha cada tarea, se

proporciona la opción Ver evidencia. Todo lo descrito se presenta en la parte (c) de la Figura 2.44.

Las opciones **Agregar actividad**, **Agregar tarea** y **Ver evidencia** muestran los modales Actividad, Tarea y Evidencia respectivamente. En la parte (a) de la Figura 2.45 se indica el modal Actividad, el cual también es utilizado para la opción **Editar Actividad**; y, en la parte (b) de la Figura 2.45 se muestra el modal Tarea, el cual también es utilizado para la opción **Editar Tarea**.

En la Figura 2.46 se presentan los *mockups* del modal Evidencia para el rol Gestor, el cual aparece mediante la opción **Ver Evidencia**. La parte (a) corresponde a la pestaña Archivos, aquí el Gestor podrá descargar los archivos que fueron subidos por su respectivo Delegado. La parte (b) corresponde a la pestaña Observaciones donde el Gestor podrá enviar mensajes al Delegado que servirá como observaciones acerca de la evidencia proporcionada. La parte (c) corresponde a la pestaña Avance, aquí el Gestor podrá Confirmar el porcentaje de avance o corregir este valor si es necesario.

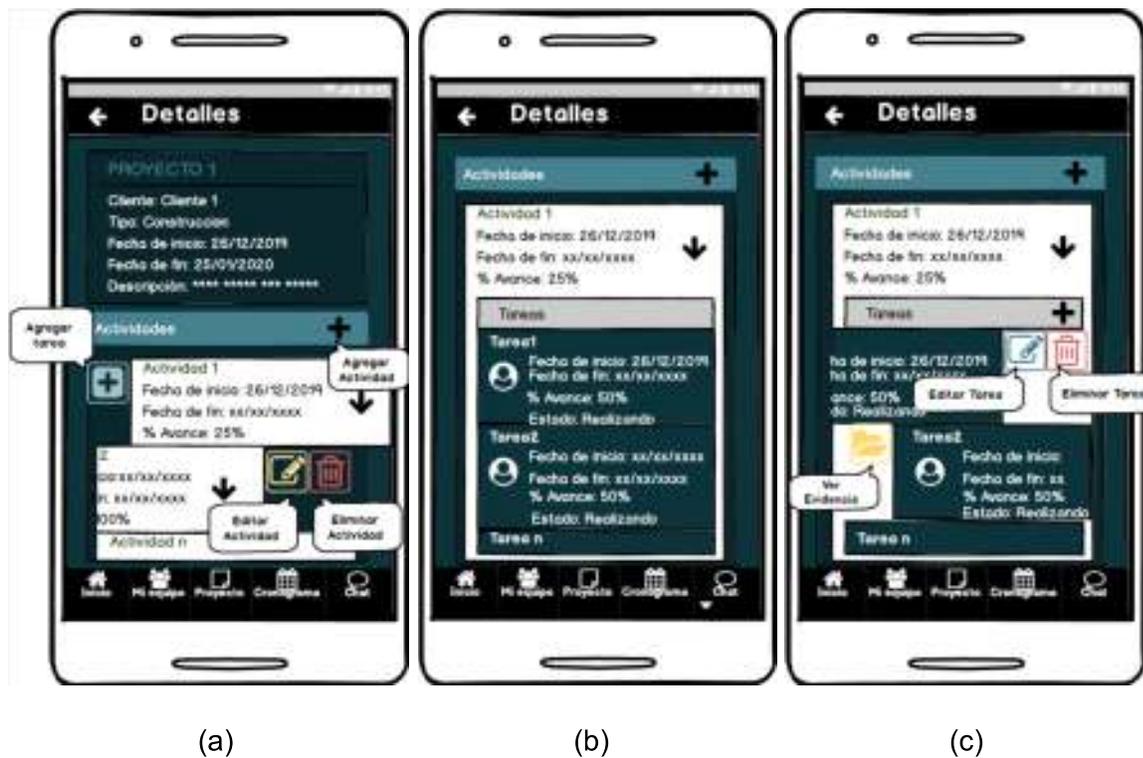


Figura 2.44. Mockups de la pantalla Actividades (a) Opciones de cada Actividad (b) actividad desplegada y (c) Opciones de cada tarea



(a)



(b)

Figura 2.45. (a) Mockup del modal Actividad y (b) Mockup del modal Tarea



Figura 2.46. Mockups del modal Evidencia para el rol Gestor (a) Pestaña Archivos (b) Pestaña Observaciones y (c) Pestaña Avance

En la Figura 2.47 se indica el *mockup* de la pantalla Proyectos para el rol Delegado, aquí se indica la lista de tareas que debe cumplir el Delegado; al deslizar cada tarea hacia la izquierda se tiene la opción **Subir evidencia**.

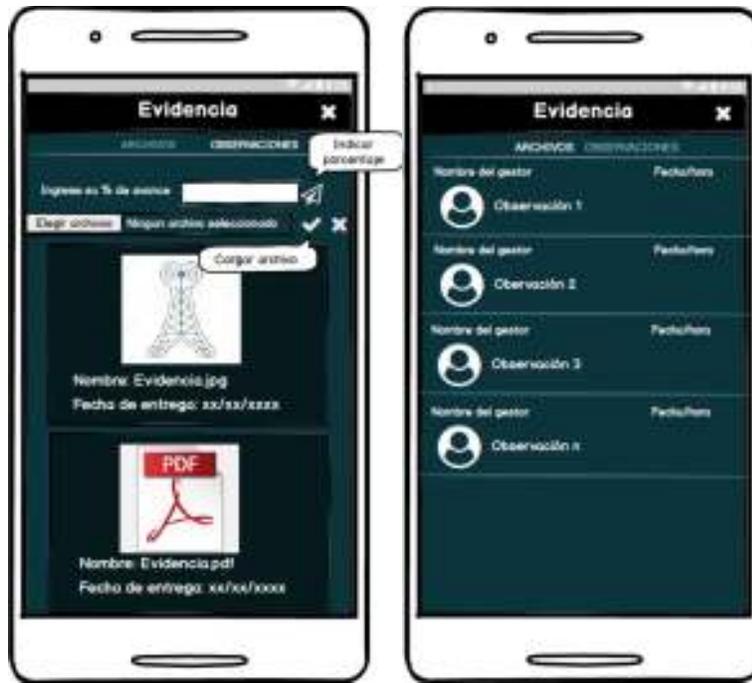
Al seleccionar esta opción aparece el modal Evidencia. En la parte (a) de la Figura 2.48 se indica el *mockup* de la pestaña Archivos donde el Delegado podrá cargar múltiples archivos e indicar su porcentaje de avance. En la parte (b) de la Figura 2.48 se indica el *mockup* de la pestaña Observaciones, aquí el Delegado podrá leer las observaciones emitidas por su Gestor.



Figura 2.47. Mockup de la pantalla Proyectos para el rol Delegado

En la parte (a) de la Figura 2.49 se muestra el *mockup* de la pantalla Cronograma para el rol Gestor donde podrá visualizar los proyectos ordenados según su fecha de entrega; al dar *click* sobre cada proyecto, se abre una nueva página que permite visualizar un calendario sobre las tareas del proyecto, como se indica en la parte (b) de la Figura 2.49.

En la Figura 2.50 se indica el *mockup* de la pantalla Cronograma para el rol Delegado, aquí se indica la lista de tareas a su cargo y ordenadas por fecha de entrega (Fecha de fin).



(a)

(b)

Figura 2.48. Mockups del modal Evidencia (a) Pestaña Archivos y (b) Pestaña Observaciones



(a)

(b)

Figura 2.49. (a) Mockup de la pantalla Cronograma para el rol Gestor y (b) Mockup del modal Calendario

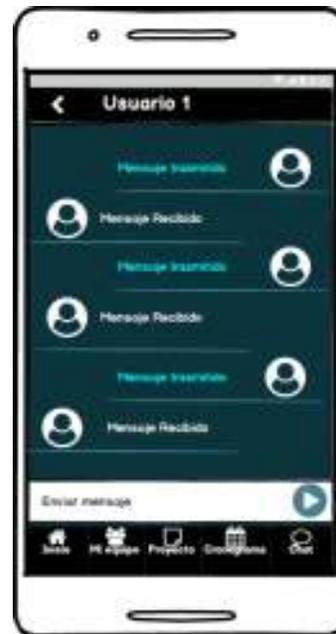


Figura 2.50. Mockup de la pantalla Cronograma para el rol Delegado

En la parte (a) de la Figura 2.51 se indica el *mockup* de la pantalla Chat, aquí se muestra la lista de usuarios disponibles según su rol. Si se trata del rol Gestor se tendrá la lista de Delegados que conforman su equipo de trabajo; si se trata del rol Delegado se tendrá la lista de Gestores y compañeros de equipo.



(a)



(b)

Figura 2.51. (a) Mockup de la pantalla Chat y (b) Mockup de la pantalla Conversación

Al seleccionar un usuario, aparece una nueva pantalla donde se indica la conversación con este usuario; aquí se puede enviar y recibir mensajes como se muestra en la parte (b) de la Figura 2.51.

2.1.6 LÓGICA DE PROCESOS

Para implementar las funcionalidades requeridas y describir el comportamiento dinámico del sistema se hace uso de la lógica de procesos, para ello se utilizan diagramas UML (*Unified Modeling Language*) de actividad y diagramas UML de secuencia.

En estos diagramas es útil dividir los estados en grupos y cada uno de estos se le asignará un nombre. A continuación, se describen los diagramas de actividades y secuencia correspondientes a cada módulo.

2.1.6.1 Módulos Registro y Autenticación

Se tiene el Módulo Registro y el Módulo Autenticación, pero ambos módulos pertenecen a un mismo proceso, ya que, si el usuario no tiene credenciales para su autenticación, deberá registrarse para obtenerlas y acceder al sistema. Por lo tanto, el proceso es: Registrar/Autenticar Usuario.

Proceso Registrar/Autenticar Usuario

En la Figura 2.52 se indica el diagrama de actividad del proceso Registrar/Autenticar Usuario, el cual ha sido dividido en tres grupos: Usuario, Sistema web y móvil, y Firebase, los cuales intervienen en el flujo de trabajo.

Cuando el usuario accede al Sistema, ya sea mediante su aplicativo web o móvil, el Sistema muestra su pantalla Login. Si el usuario cuenta con credenciales para el acceso al Sistema deberá ingresarlas en esta pantalla.

El Sistema verifica si el usuario ha dejado algún campo inválido; si existen campos inválidos, el Sistema presentará un mensaje de Error, por lo cual el usuario deberá volver a ingresar sus credenciales. Si no existen campos inválidos, el Sistema envía los datos hacia *Firebase Authentication* para que verifique si las credenciales ingresadas existen.

Si éstas existen, la aplicación muestra el mensaje Ingreso exitoso y muestra la pantalla Inicio. Si las credenciales no existen, el sistema muestra un mensaje de Error, por consecuencia el usuario deberá ingresar sus credenciales nuevamente.

Si el usuario no cuenta con credenciales para el acceso al Sistema, éste mostrará la pantalla Registro, donde deberá ingresar los datos correspondientes. El Sistema verifica si el usuario ha dejado algún campo inválido, si existen campos inválidos, la aplicación indicará un mensaje de Error y el usuario deberá ingresar sus datos de registro nuevamente.

El Sistema permite que el usuario al momento de registrarse pueda hacer uso del sistema de dos maneras: para uso empresarial o individual. Si se trata de un uso empresarial, se deberá verificar que el usuario se encuentre dentro de una lista de la empresa en la base de datos alojada en *Cloud Firestore*; si el usuario existe, *Firebase Authentication* deberá verificar si este correo electrónico se encuentra registrado en su servicio. Si el correo electrónico ya se encuentra registrado, el sistema mostrará un mensaje de Error y el usuario deberá volver a ingresar sus datos de registro.

Si se trata de uso personal no se realizará la verificación en la base de datos, sino solamente en el servicio de autenticación.

Si el correo electrónico no se encuentra registrado, el servicio de autenticación realiza el respectivo registro, retorna el usuario hacia el Sistema, éste selecciona los atributos necesarios para formar el esquema del nuevo usuario a partir del usuario retornado y se envían estos datos hacia *Cloud Firestore* para ser almacenados. El Sistema muestra un mensaje de Registro Exitoso y muestra la pantalla Login.

El diagrama de secuencia correspondiente al proceso Registro/Autenticación, ha sido dividido en dos gráficos, uno para Registro y otro para Autenticación.

En la Figura 2.53 se tiene el Diagrama de secuencia del proceso Registrar usuario. El usuario debe ingresar al Sistema, en la pantalla Registro deberá ingresar los datos de registro. El Sistema valida estos datos, si el uso es empresarial, enviará los datos hacia *Firebase* para que *Cloud Firestore* verifique si el usuario se encuentra registrado dentro la lista y envíe una respuesta exitosa; si el uso es personal, se omiten estos pasos.

Independientemente del uso del sistema, éste enviará los datos hacia *Firebase Authentication* para verificación del correo electrónico, éste retornará los datos y el Sistema estructurará el usuario y los enviará hacia *Firebase* para ser almacenado en *Cloud Firestore*. Este enviará una respuesta exitosa al Sistema para que muestre al usuario la pantalla Login.

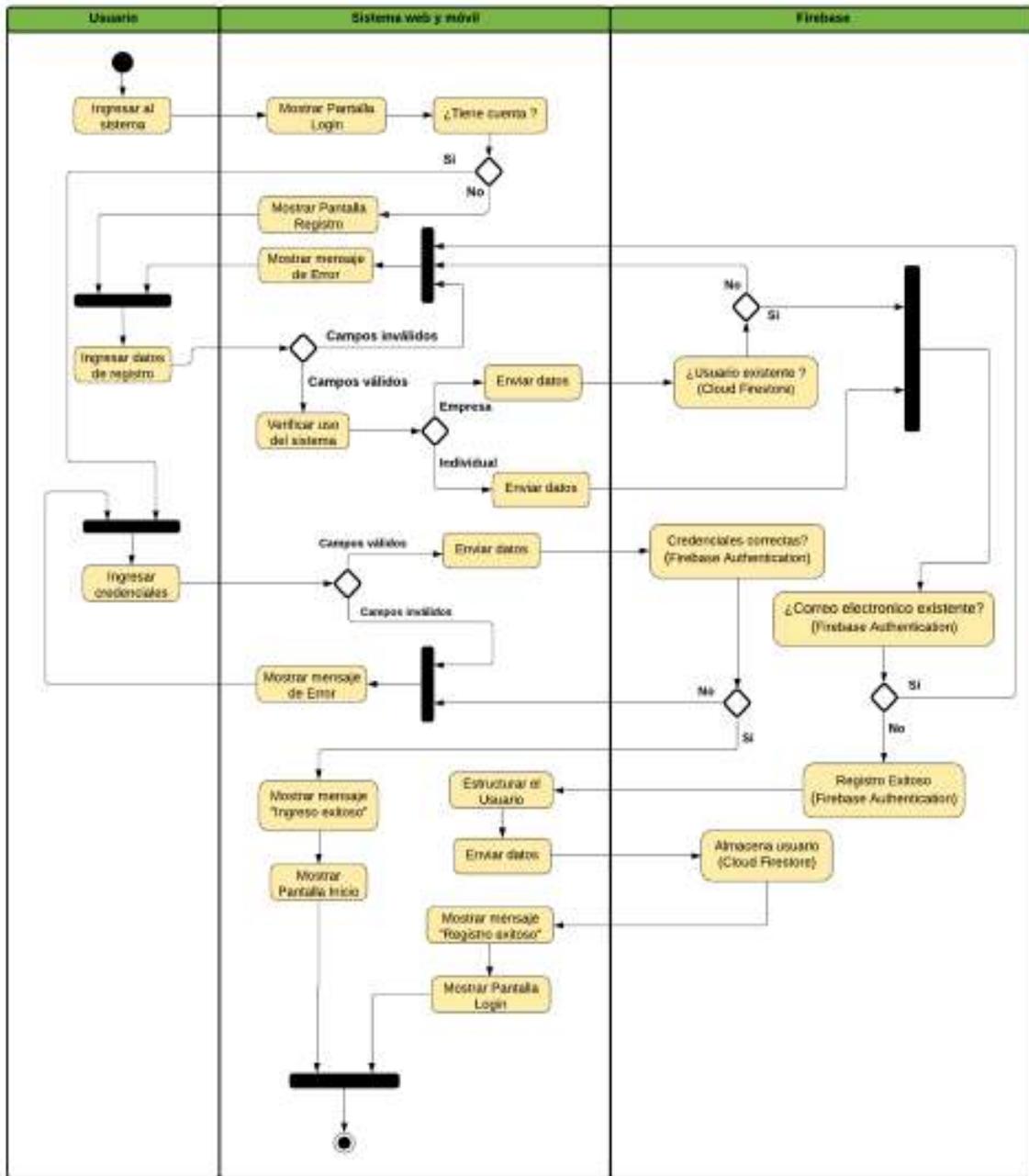


Figura 2.52. Diagrama de actividad del proceso Registrar/Autenticar Usuario

En la Figura 2.54 se tiene el Diagrama de secuencia del proceso Autenticar usuario; el usuario debe ingresar al Sistema, en la pantalla Login ingresar sus credenciales. El sistema

valida estos datos, envía los datos hacia Firebase, *Firestore* valida las credenciales y finalmente, si la respuesta de Firebase es correcta, el Sistema muestra la pantalla Inicio, caso contrario el Sistema indica un mensaje de error al usuario para que vuelva a intentar su Ingreso.

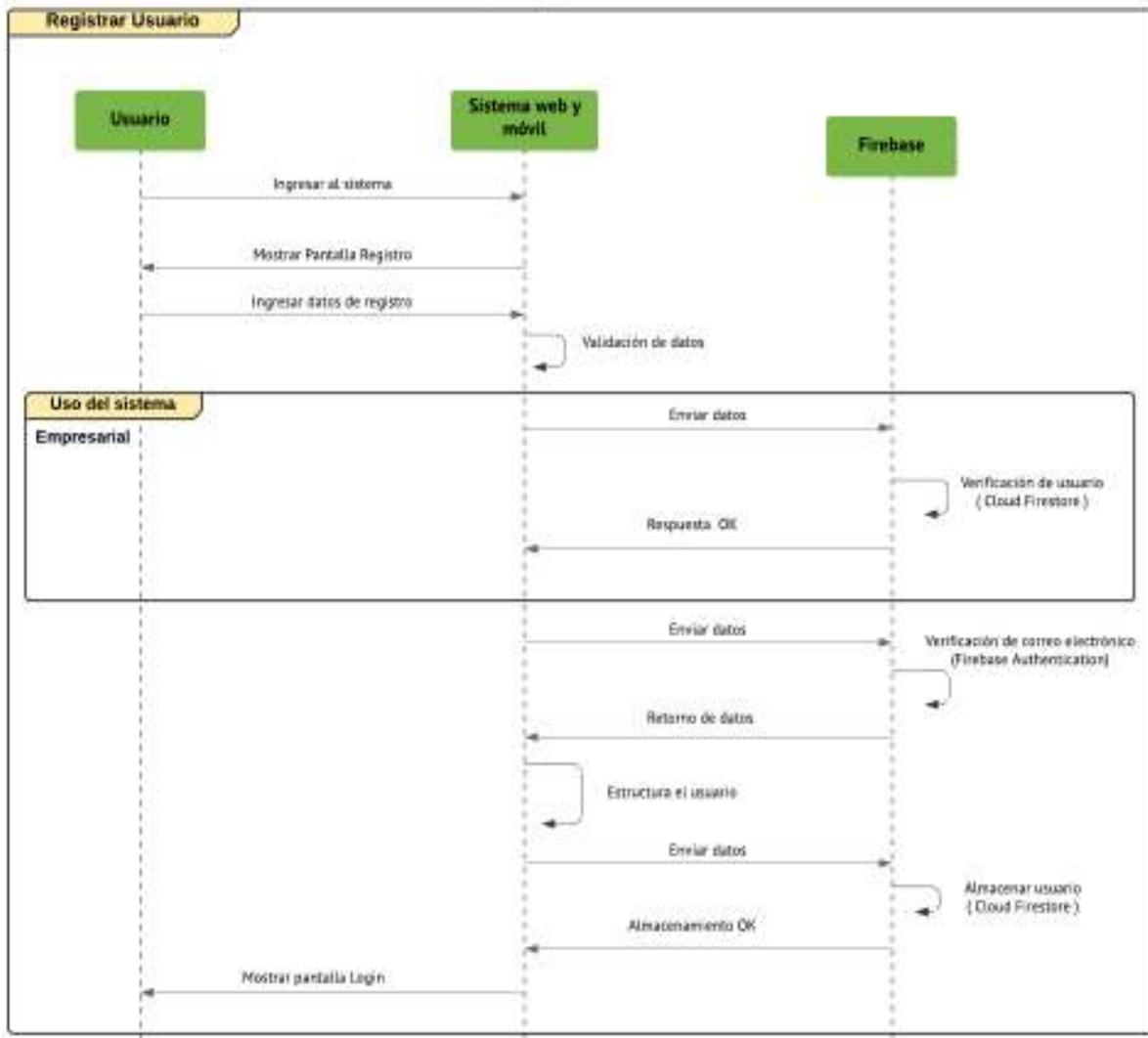


Figura 2.53. Diagrama de secuencia del proceso Registrar Usuario

Como ya se ha explicado, el proceso de registro y autenticación del usuario, a partir de este diagrama y los que vengan a continuación, se va a partir de que el usuario ya ingresó sus credenciales y ha tenido una validación exitosa y por lo tanto, el usuario se encuentra ya dentro del Sistema.

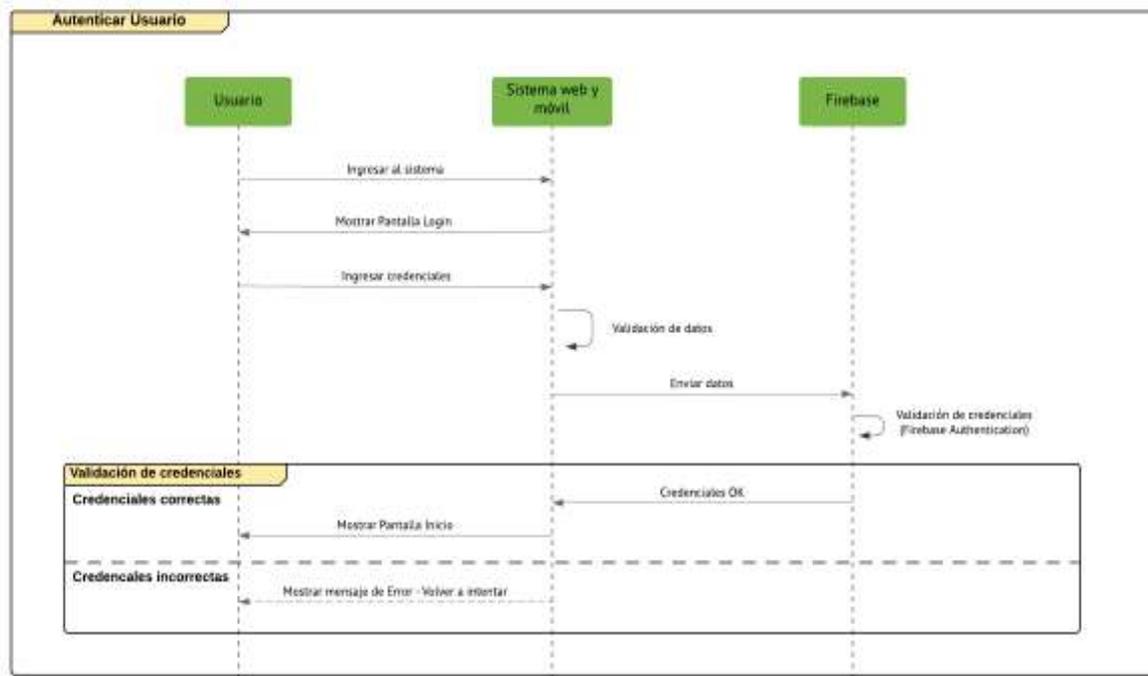


Figura 2.54. Diagrama de secuencia del proceso Autenticar Usuario

2.1.6.2 Módulo Equipo

Para este módulo se ha trabajado el rol Gestor y el rol Delegado dentro de un mismo proceso, debido a la poca carga de datos que maneja cada uno. Por esta razón para este módulo se ha definido el proceso: Conformar equipo.

Proceso Conformar equipo

En la Figura 2.55 se indica el Diagrama de actividades del proceso Conformar equipo; este ha sido dividido en tres grupos: Usuario, Sistema web y móvil, y Firebase, los cuales intervienen en el flujo de trabajo de este proceso.

El usuario ingresa al Sistema, éste muestra la pantalla Inicio y el usuario deberá seleccionar la opción **Mi Equipo**; el sistema indicará la pantalla Mi equipo. Si el usuario tiene un rol de Delegado, el Sistema muestra su lista de Gestores y compañeros de equipo. Si el usuario tiene un rol de Gestor, podrá realizar algunas acciones. Si decide seleccionar Delegados para conformar su equipo, el Sistema enviará los datos de los delegados seleccionados hacia *Cloud Firestore* para su almacenamiento y el Sistema actualizará la pantalla Mi Equipo.

Si el usuario selecciona la opción **Eliminar usuario**, el Sistema envía los datos del Delegado seleccionado hacia *Cloud Firestore* para que elimine el usuario y el Sistema actualizará la pantalla Mi Equipo.

La opción **Ver recursos**, es la misma para ambos roles; si el usuario selecciona esta opción, el Sistema muestra el modal Recursos, el cual presenta tres opciones: Escribir mensaje, Cargar archivo y Descargar archivos. Estas opciones se encuentran presentes dentro de las respectivas pestañas que se indicaron en la Figura 2.20.

Si el usuario selecciona la opción **Escribir mensaje**, el Sistema envía los datos hacia *Cloud Firestore* para que los almacene, y se actualiza la pantalla Mi Equipo. Si decide **Cargar archivos**, el usuario cargará archivos desde el ordenador o dispositivo móvil, el Sistema enviará los datos a *Cloud Storage* que es el encargado de almacenarlos y generar una *URL* que será almacenada en *Cloud Firestore*, y el Sistema actualizará la pantalla Mi Equipo. Si el usuario decide **Descargar archivos**, el Sistema redirecciona a la URL correspondiente al archivo seleccionado.

En la Figura 2.56 se indica el Diagrama de secuencia del proceso Conformar Equipo, aquí se tiene el condicional Rol de usuario; si se trata de un usuario Gestor, éste realizará todas las acciones antes ya explicadas en el diagrama de actividades correspondiente. Si las respuestas emitidas por cada respectivo servicio de Firebase son exitosas, la pantalla Mi Equipo se actualizará. Si se trata de un usuario Delegado, el sistema indicará la lista de Gestores y compañeros de equipo.

Una vez finalizado el condicional Rol de usuario, ambos roles podrán seleccionar opción Ver recursos, aquí se realizarán las acciones correspondientes a esta opción, las cuales se explicaron en el diagrama de actividades correspondientes. Si las respuestas emitidas por cada respectivo servicio de Firebase son exitosas, la pantalla Mi Equipo se actualizará.

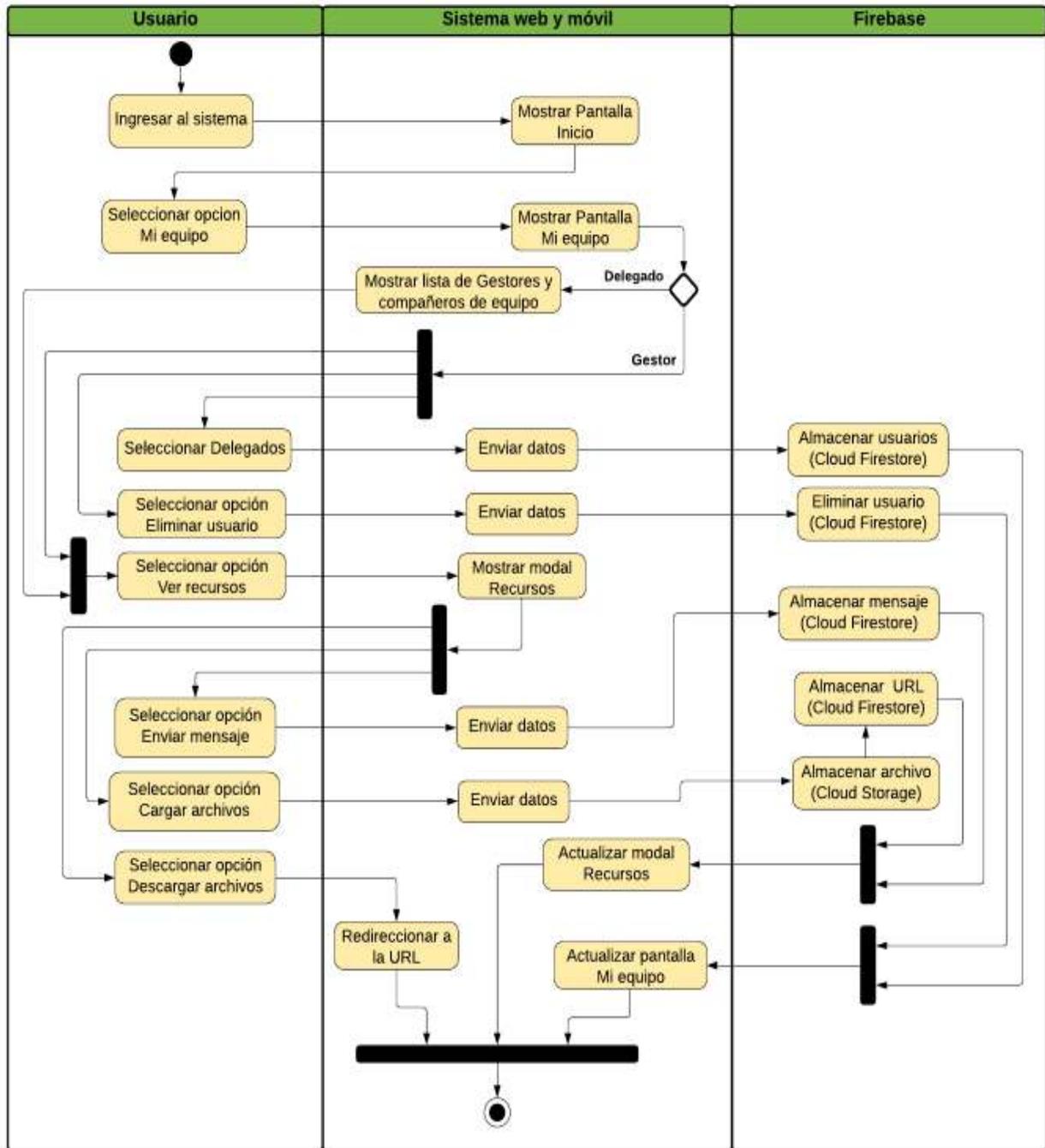


Figura 2.55. Diagrama de actividad del Proceso Conformar equipo

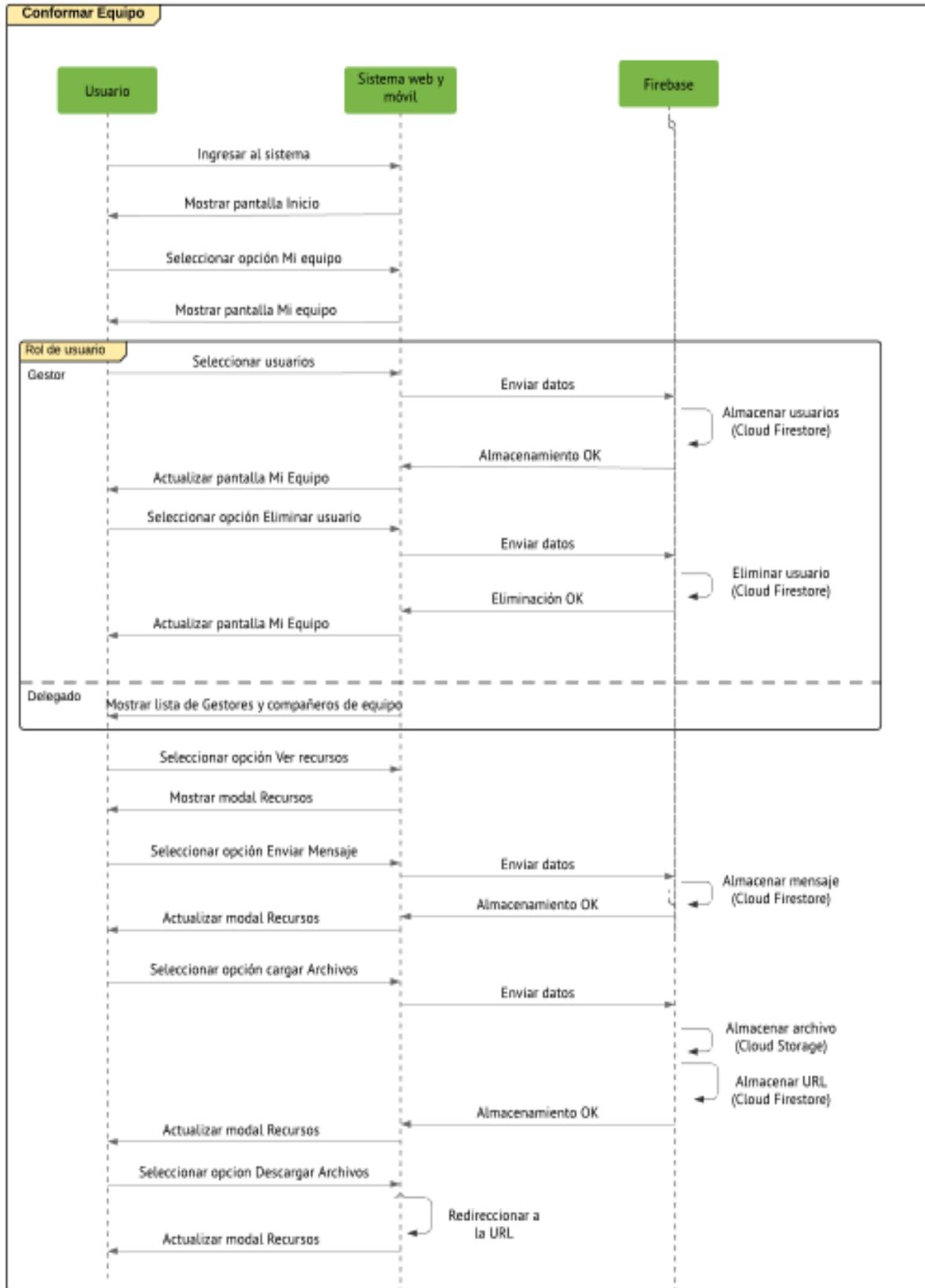


Figura 2.56. Diagrama de secuencia del Proceso Conformar equipo

2.1.6.3 Módulo Proyecto

Para este módulo se ha asignado un proceso para cada rol debido a la gran cantidad de flujo de trabajo que maneja cada uno. Por lo tanto, para este módulo se han definido los procesos Administrar proyectos para el Gestor y Cumplir proyectos para el Delegado.

Proceso Administrar proyectos

Este proceso se complementa con el proceso del módulo Actividades, debido a que el módulo Proyectos contiene las actividades pertenecientes a un proyecto, las tareas pertenecientes a cada actividad, así como las evidencias para la verificación de tareas. Siendo las actividades, tareas y evidencias parte del módulo Actividades. Por lo tanto, se tiene un solo diagrama de actividades y un solo diagrama de secuencias que abarcan ambos módulos para el proceso Administrar proyectos.

En la Figura 2.57 se indica el Diagrama de actividades del proceso Administrar Proyectos para el Gestor; éste ha sido dividido en tres grupos: Usuario, Sistema web y móvil, y Firebase. El usuario ingresa al Sistema, éste presenta la pantalla Inicio y el usuario deberá seleccionar la opción Proyectos y el Sistema mostrará la pantalla Proyectos. Si el Gestor aún no ha conformado su equipo de trabajo no podrá manejar proyectos, por lo tanto, deberá seleccionar su equipo de trabajo en la opción Mi Equipo y seguirá el proceso Conformar equipo antes ya explicado.

Si el Gestor ya tiene su equipo de trabajo conformado, podrá realizar algunas acciones, como: Crear proyecto, Editar proyecto, Eliminar proyecto y Ver actividades, estas opciones se indicaron en la Figura 2.21. Si decide la opción **Crear proyecto**, el Sistema envía los datos hacia *Cloud Firestore* para almacenar este nuevo proyecto y por último actualiza la pantalla Proyectos. Si decide **Editar** o **Eliminar un proyecto**, el Sistema envía los datos a *Cloud Firestore* para editar o eliminar el proyecto y actualiza la pantalla Proyectos.

Si decide **Ver actividades**, el Sistema redirigirá a la pantalla Actividades. Aquí podrá realizar algunas acciones, como: Agregar actividad, Editar actividad, Eliminar actividad, Desplegar detalles de cada actividad y Agregar tarea a cada actividad.

Si decide alguna de las opciones **Agregar tarea** a una actividad, **Editar actividad**, **Eliminar actividad** o **Agregar Actividad**, el Sistema mostrará los respectivos formularios y enviará los datos hacia *Cloud Firestore* que es el encargado de almacenar las tareas,

editar actividad, eliminar actividad o almacenar una actividad respectivamente y, por último, el sistema actualizará la pantalla Actividades.

Si decide **Desplegar detalles de actividad**, el Sistema mostrará la lista de tareas; el Gestor podrá realizar algunas acciones sobre cada tarea, como: Eliminar tarea, Editar tarea o Ver evidencia.

Si decide alguna de las opciones **Eliminar tarea** o **Editar tarea**, el Sistema mostrará los respectivos formularios y enviará los datos hacia *Cloud Firestore* para que sea el encargado de eliminar la tarea o editar la tarea respectivamente; y, por último, el Sistema actualizará la pantalla Actividades.

Si decide la opción **Ver Evidencia**, la aplicación mostrará el modal Evidencia, El Gestor podrá realizar algunas acciones, como: Descargar archivo, Enviar mensaje, Corregir progreso y Confirmar progreso; estas opciones se indicaron en las pestañas correspondientes de la Figura 2.33.

Si decide algunas de las opciones **Confirmar progreso**, **Corregir progreso**, **Enviar mensaje**, envía los datos hacia *Cloud Firestore* para que sea el encargado de almacenar los datos de cada acción y, por último, el Sistema actualizará el modal Evidencia.

Si decide la opción **Descargar archivo**, el sistema redirige a la URL correspondiente para que el Gestor pueda obtener el archivo que desee.

En la Figura 2.58 se indica el Diagrama de secuencia del proceso Administrar proyectos, aquí se tiene el condicional Equipo de trabajo, si el usuario ya tiene su equipo de trabajo formado podrá realizar todas las acciones correspondientes, para que los servicios de Firebase haga los procesos de acuerdo a estas acciones y la pantalla Proyecto y Actividades se actualicen según se requiera.

Si el usuario no tiene su equipo de trabajo formado deberá seleccionar la opción **Mi Equipo** para realizar el proceso correspondiente que se indica en la Figura 2.55.

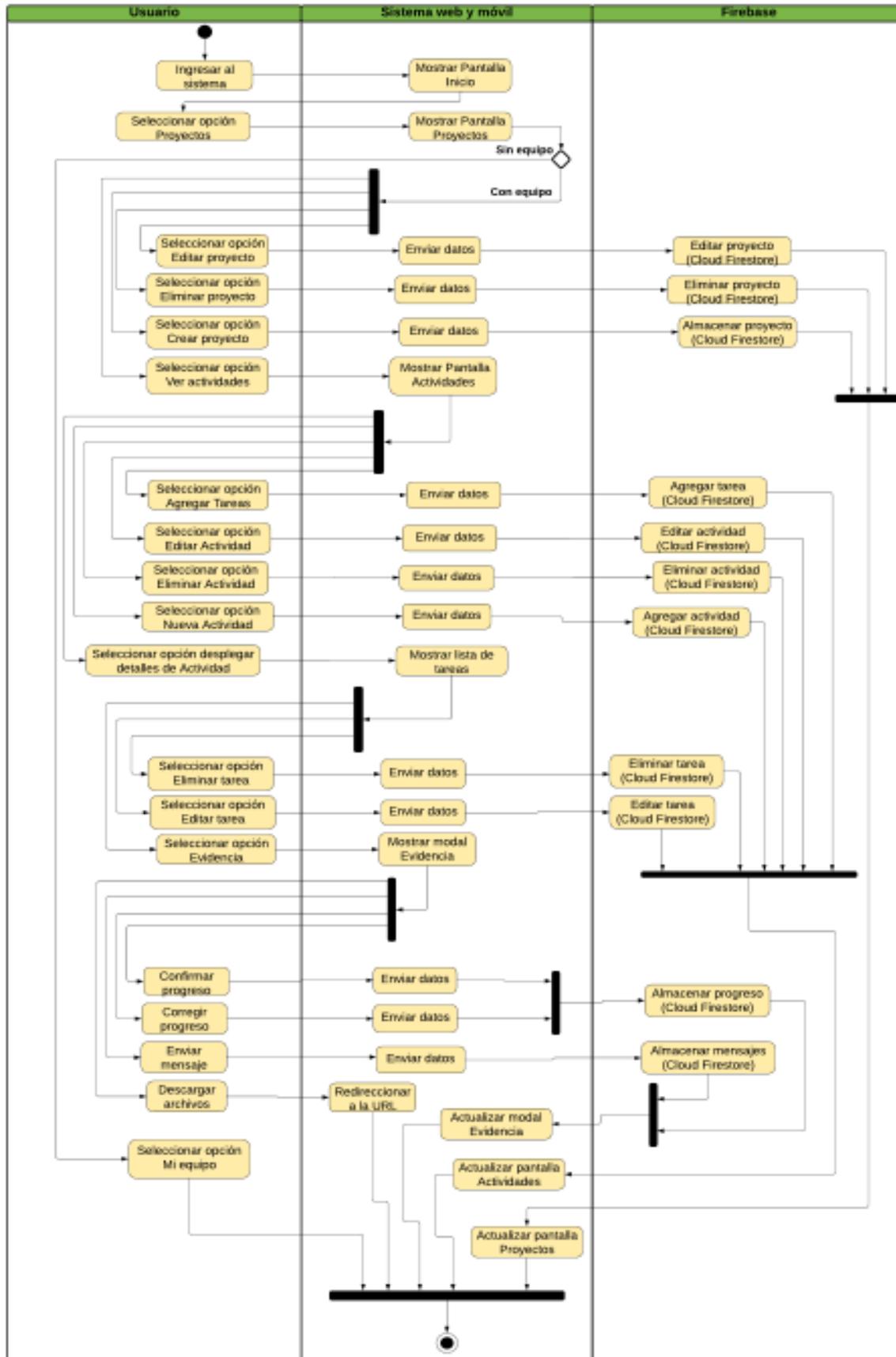


Figura 2.57. Diagrama de actividad del Proceso Administrar proyectos

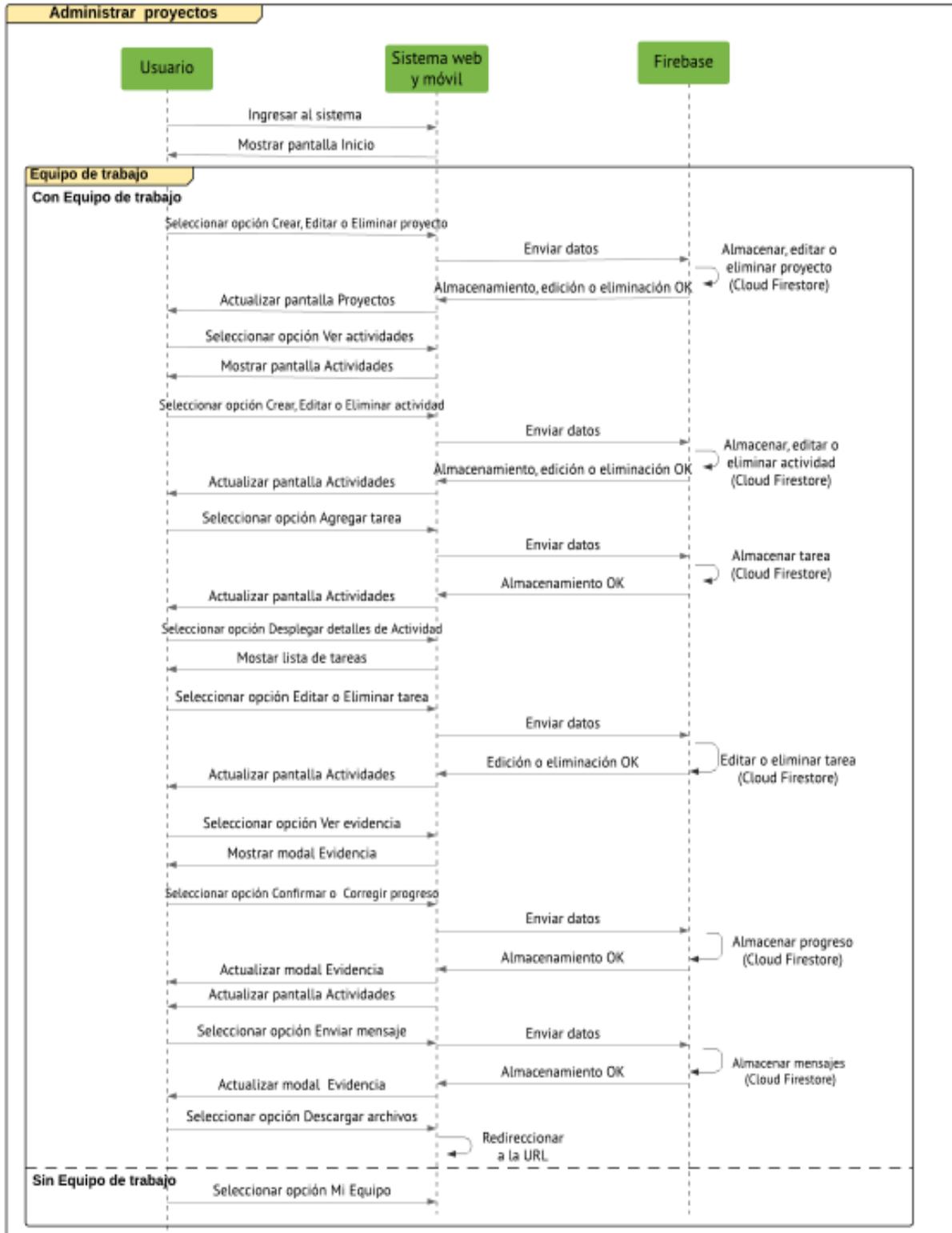


Figura 2.58. Diagrama de secuencia del Proceso Administrar proyectos

Proceso Cumplir proyectos

Este proceso abarca solo el módulo Proyectos debido a que el módulo Actividades no presenta funcionalidades para el rol Delegado.

En la Figura 2.59 se indica el Diagrama de actividades del Proceso Cumplir proyectos, éste ha sido dividido en tres grupos: Usuario, Sistema web y móvil, y Firebase. El Delegado ingresa al Sistema, éste muestra la pantalla Inicio y el usuario deberá seleccionar la opción Proyectos, con lo que el Sistema mostrará la pantalla Proyectos.

El Delegado podrá seleccionar la opción **Subir Evidencia** y el Sistema mostrará el modal Evidencia; aquí se tienen dos opciones: Indicar porcentaje de avance de cada tarea o Cargar archivos.

Si decide **Cargar archivos**, el Delegado seleccionará los archivos desde el ordenador o dispositivo móvil, el Sistema enviará los datos a *Cloud Storage* que será el encargado de almacenar los archivos y generar una *URL* la cual será almacenada en *Cloud Firestore*. Si decide **Indicar el porcentaje de avance** de cada tarea, el Sistema enviará los datos hacia *Cloud Firestore* para almacenar este valor y actualizar el modal Evidencia.

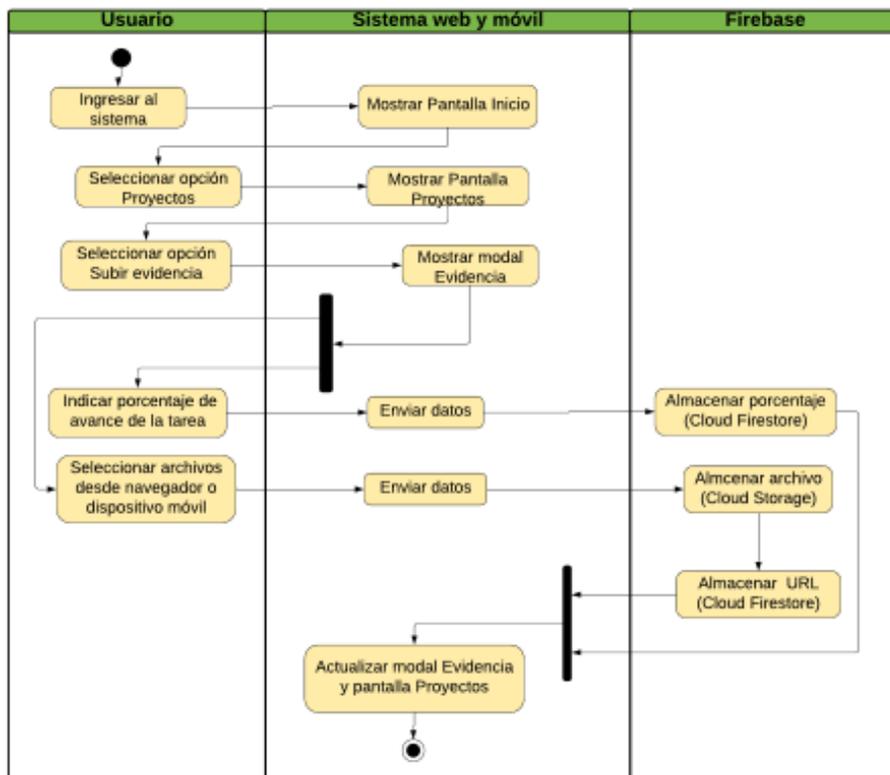


Figura 2.59. Diagrama de actividad del Proceso Cumplir proyectos

En la Figura 2.60 se tiene el Diagrama de secuencia del Proceso Cumplir proyecto, aquí el usuario podrá realizar las opciones correspondientes, los datos serán enviados hacia Firebase, para que sus servicios hagan los respectivos procesos, y si las respuestas emitidas por estos servicios son exitosas, el Sistema actualizará la pantalla Proyectos o el modal Evidencia según sea el procedimiento.

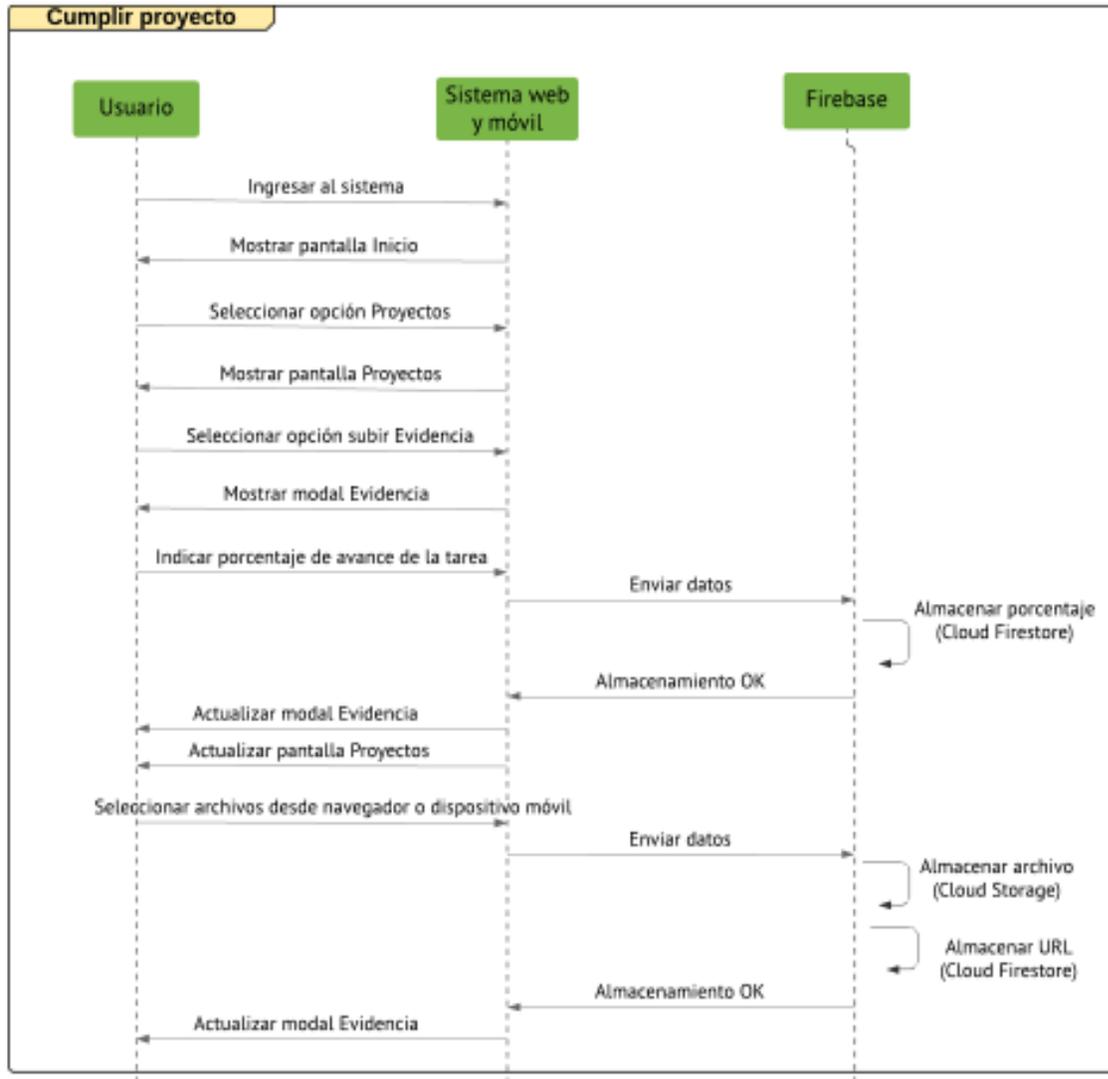


Figura 2.60. Diagrama de secuencia del Proceso Cumplir proyectos

2.1.6.4 Módulo Chat

Para este módulo se ha definido un solo proceso, debido a que la funcionalidad de éste es independiente del rol del usuario. Por la razón anterior, para este módulo se ha definido el proceso: Conversar con un usuario.

Proceso Conversar con un usuario

En la Figura 2.61 se indica el diagrama de actividades del proceso Conversar con un usuario, el cual ha sido dividido en tres grupos: Usuario, Sistema web y móvil, y Firebase. El usuario ingresa al sistema, éste muestra la pantalla Inicio y el usuario deberá seleccionar la opción Chat, ante lo cual el Sistema mostrará la pantalla Chat. El usuario deberá seleccionar el usuario al cual enviará el mensaje, el Sistema mostrará la conversación, el usuario escribirá el mensaje para que el Sistema envíe los datos hacia *Cloud Firestore* para que sean almacenados y, por último, el Sistema actualizará la conversación.

El usuario deberá seleccionar el usuario al cual enviará el mensaje, el Sistema mostrará la conversación, el usuario escribirá el mensaje para que el Sistema envíe los datos hacia *Cloud Firestore* para que sean almacenados y, por último, el Sistema actualizará la conversación.

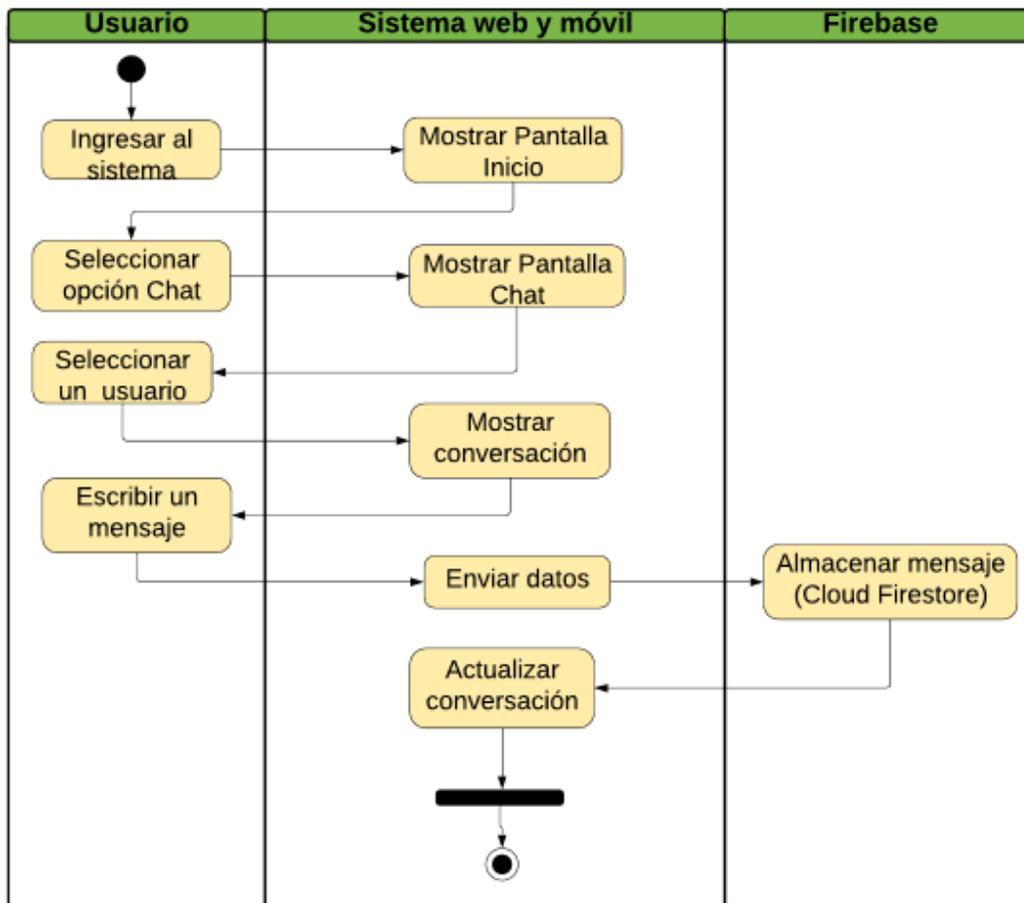


Figura 2.61. Diagrama de actividad del Proceso Conversar un usuario

En la Figura 2.62 se tiene el Diagrama de secuencia del Proceso Conversar con un usuario; el usuario selecciona al usuario remoto y envía el mensaje hacia Firebase para que el servicio de *Cloud Firestore* almacene este mensaje y retorne una respuesta exitosa hacia el Sistema para que actualice la conversación.

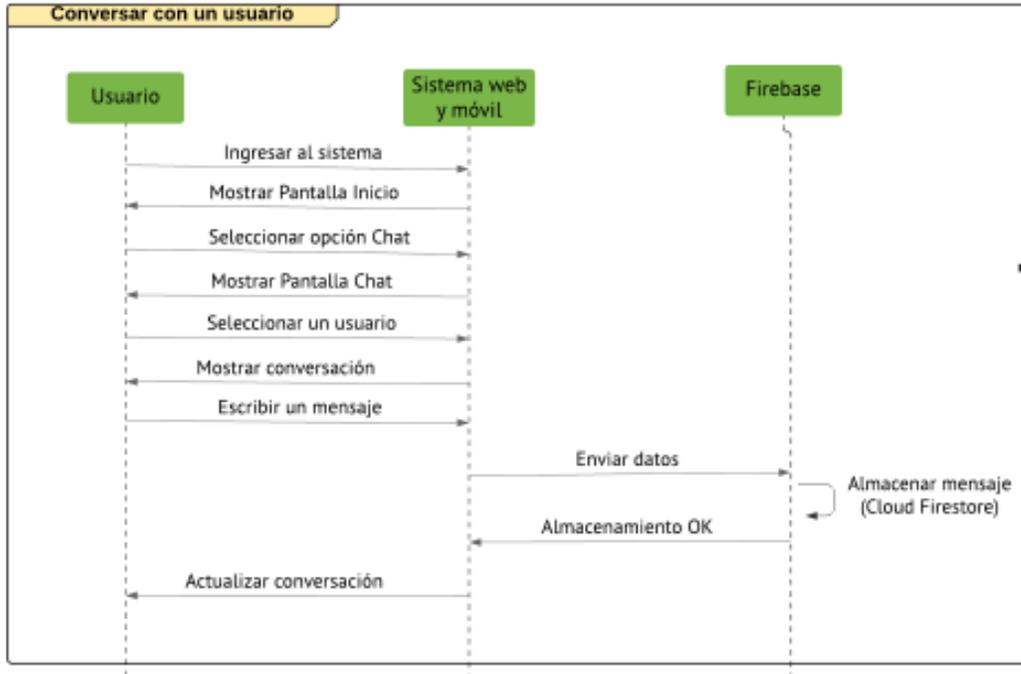


Figura 2.62. Diagrama de secuencia del Proceso Conversar con un usuario

2.1.6.5 Módulo Perfil de usuario

Para este módulo se ha definido un solo proceso, debido a que la funcionalidad de este es independiente del rol del usuario. Por la razón anterior, para este módulo se ha definido el proceso: Configurar perfil.

Proceso Configurar perfil

En la Figura 2.63 se indica el diagrama de actividades del Proceso Configurar perfil, el cual ha sido dividido en tres grupos: Usuario, Sistema web y móvil, y Firebase. El usuario ingresa al Sistema, éste muestra la pantalla Inicio, el usuario deberá seleccionar el icono Perfil ubicado en la esquina superior derecha, la aplicación despliega un pequeño menú acerca del perfil del usuario como se indicó en la Figura 2.12. El usuario tiene dos opciones: Cerrar sesión y Editar perfil.

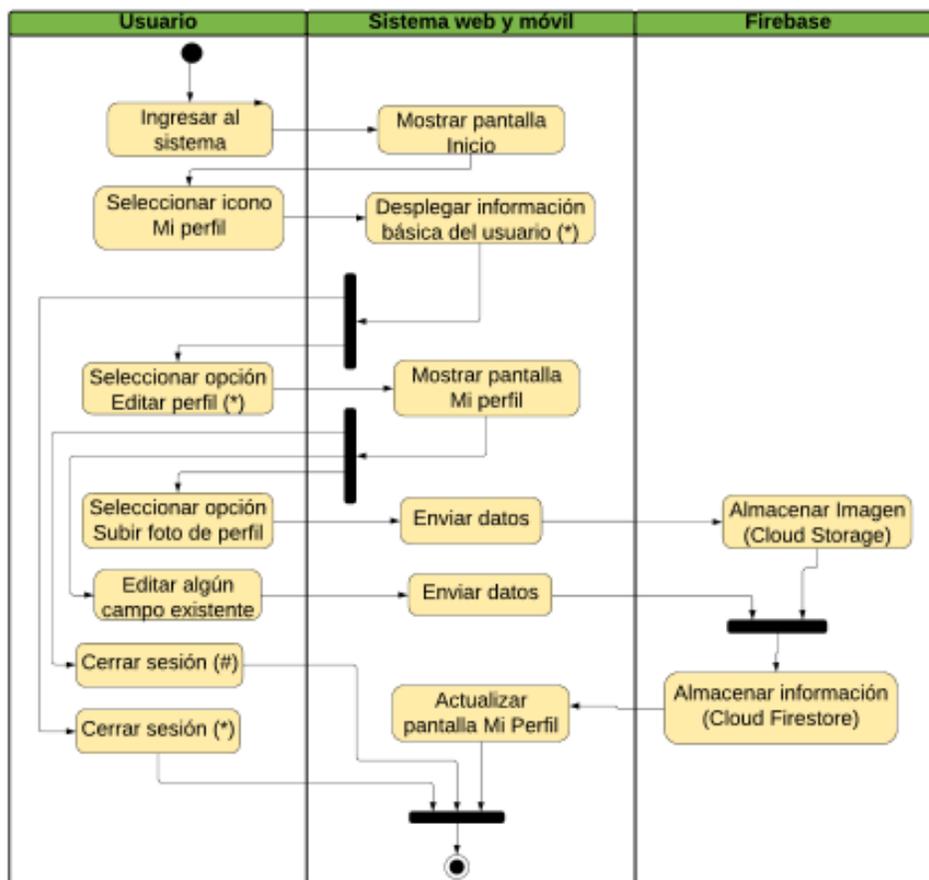
Si el usuario decide **Cerrar sesión**, el Sistema cerrará la sesión actual del usuario. Si el usuario decide **Editar perfil**, el Sistema mostrará la pantalla Mi perfil; aquí podrá editar los campos existentes o subir una foto para su perfil. Si el usuario decide **Subir foto de perfil**, deberá seleccionar una foto desde su ordenador o dispositivo móvil. El Sistema enviará la imagen seleccionada a *Cloud Storage* para almacenar este archivo y generar una *URL*, la

cual será almacenada en *Cloud Firestore*. Por último, la aplicación actualizará la visualización de la pantalla Mi Perfil.

Si el usuario decide **Editar algún campo existente**, la aplicación enviará los datos hacia *Cloud Firestore* para almacenar los nuevos datos y el Sistema actualizará la pantalla Mi Perfil.

El diagrama de actividades y el diagrama de secuencia de este proceso presentan el símbolo (*), este símbolo son acciones que el Sistema web y móvil ejecutará solamente en el aplicativo web y sigue el proceso explicado anteriormente. Para el aplicativo móvil estas acciones no son ejecutadas, por lo tanto, el usuario selecciona el ícono Mi perfil y el aplicativo redirecciona directamente a la pantalla Mi perfil y continúa normalmente con el proceso indicado.

El símbolo (#) indica la acción que realizará solo el aplicativo móvil, una vez que se visualice la pantalla Mi perfil, el usuario tendrá la opción de Cerrar sesión desde esta pantalla.



(*) Solo en Aplicación web (#) Solo en Aplicación móvil

Figura 2.63. Diagrama de actividad para el Proceso Configurar perfil

En la Figura 2.64 se tiene el Diagrama de secuencia del Proceso Configurar perfil; el usuario selecciona la acción a realizar para que los servicios de Firebase sean los encargados de hacer el correspondiente proceso y enviar la respuesta exitosa hacia el Sistema y éste actualice la pantalla Mi perfil.

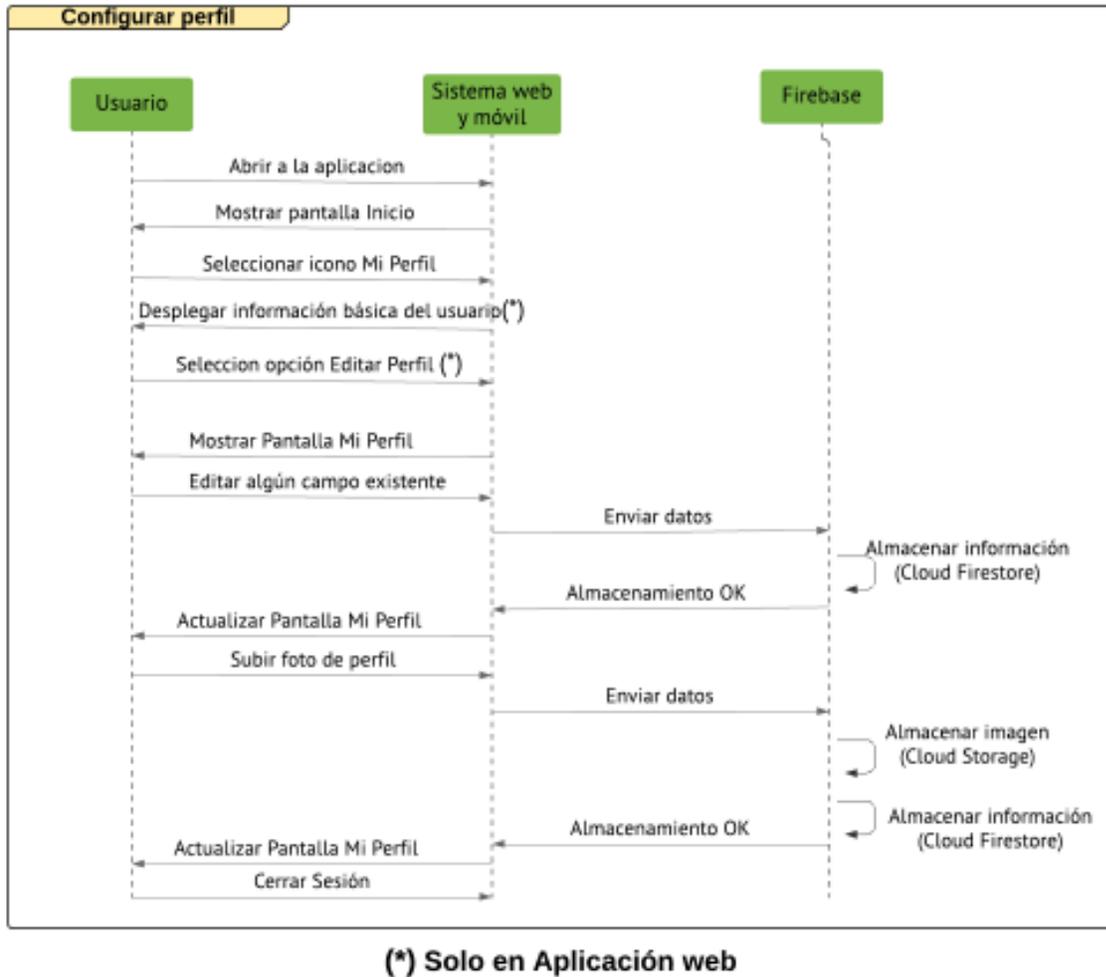


Figura 2.64. Diagrama de secuencia para el Proceso Configurar perfil

2.1.6.6 Módulo Cronograma

Para este módulo se ha trabajado el rol Gestor y el rol Delegado dentro de un mismo proceso. Por lo tanto, para este módulo se ha definido el proceso: Revisar Cronograma.

Proceso Revisar cronograma

En la Figura 2.65 se indica el diagrama de actividades del proceso Revisar Cronograma; este ha sido dividido en dos grupos: Usuario y Sistema web y móvil. El usuario ingresa al

aplicativo web o al aplicativo móvil, el Sistema muestra la pantalla Inicio y el usuario deberá seleccionar la opción Cronograma y el sistema presentará la pantalla Cronograma.

Si el usuario tiene el rol de Delegado, la aplicación mostrará la lista de tareas ordenadas por su fecha de entrega. Si el usuario tiene un rol de Gestor, la aplicación mostrará la lista de proyectos ordenados según su fecha de entrega. Este usuario podrá visualizar la secuencia de las tareas con sus fechas de inicio y fin, así como el porcentaje de avance, mediante la opción **Ver calendario**.

En la Figura 2.66 se tiene el Diagrama de secuencia del proceso Revisar Cronograma. Aquí el usuario tiene el condicional Rol de usuario; si el usuario es un Gestor, podrá visualizar la lista de sus proyectos y tendrá la opción Ver calendario, mientras que si es un Delegado podrá visualizar la lista de sus tareas.

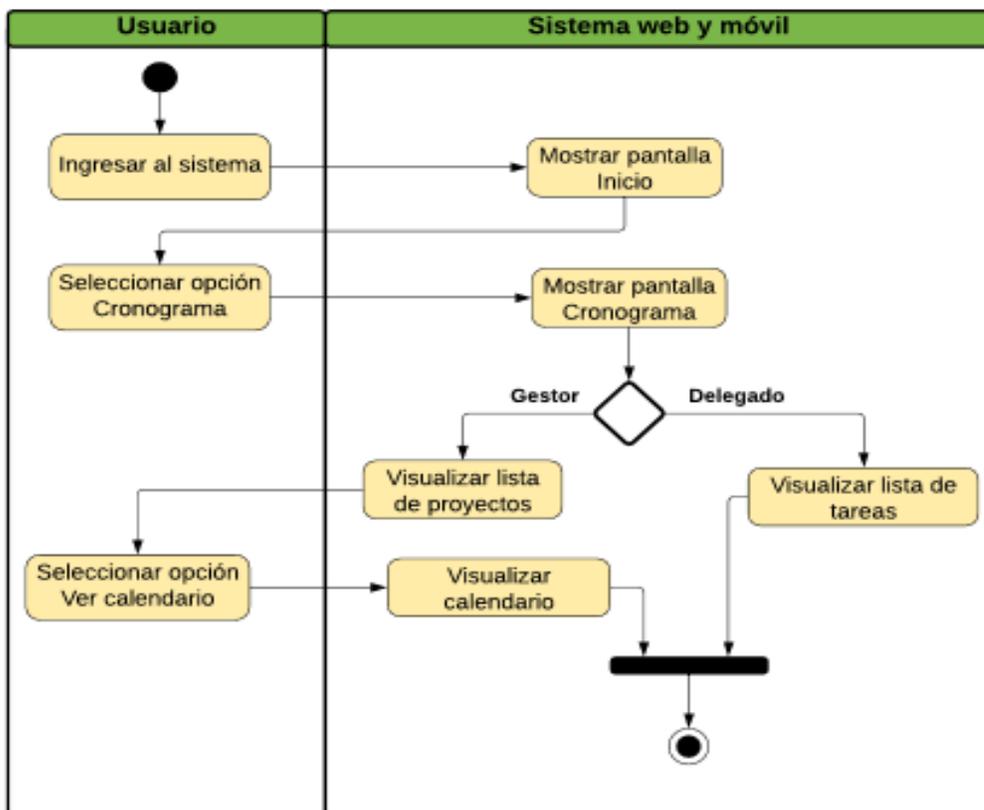


Figura 2.65. Diagrama de actividad para el Proceso Revisar cronograma

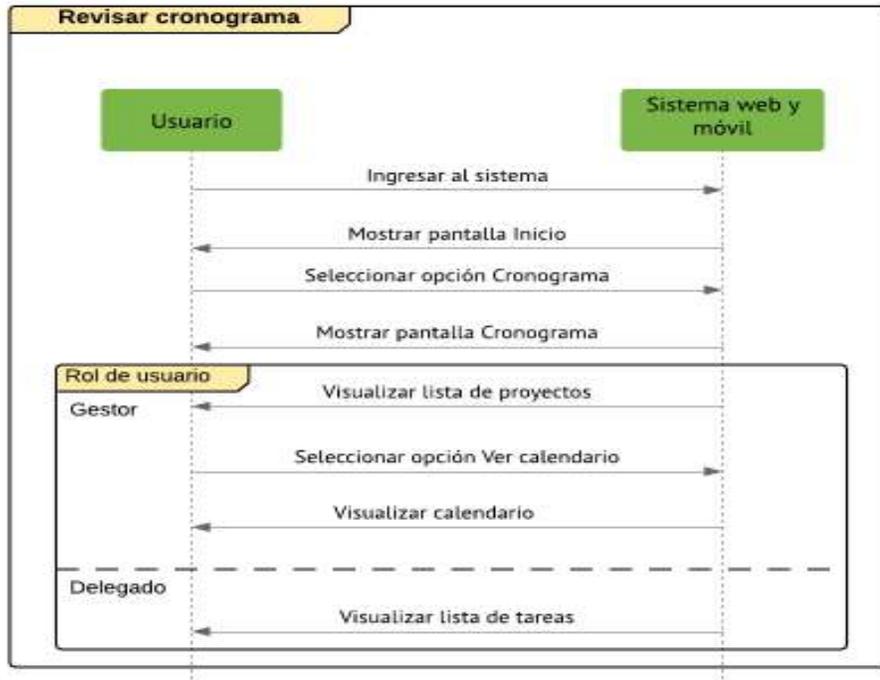


Figura 2.66. Diagrama de secuencia para el Proceso Revisar cronograma

2.1.6.7 Módulo Notificaciones

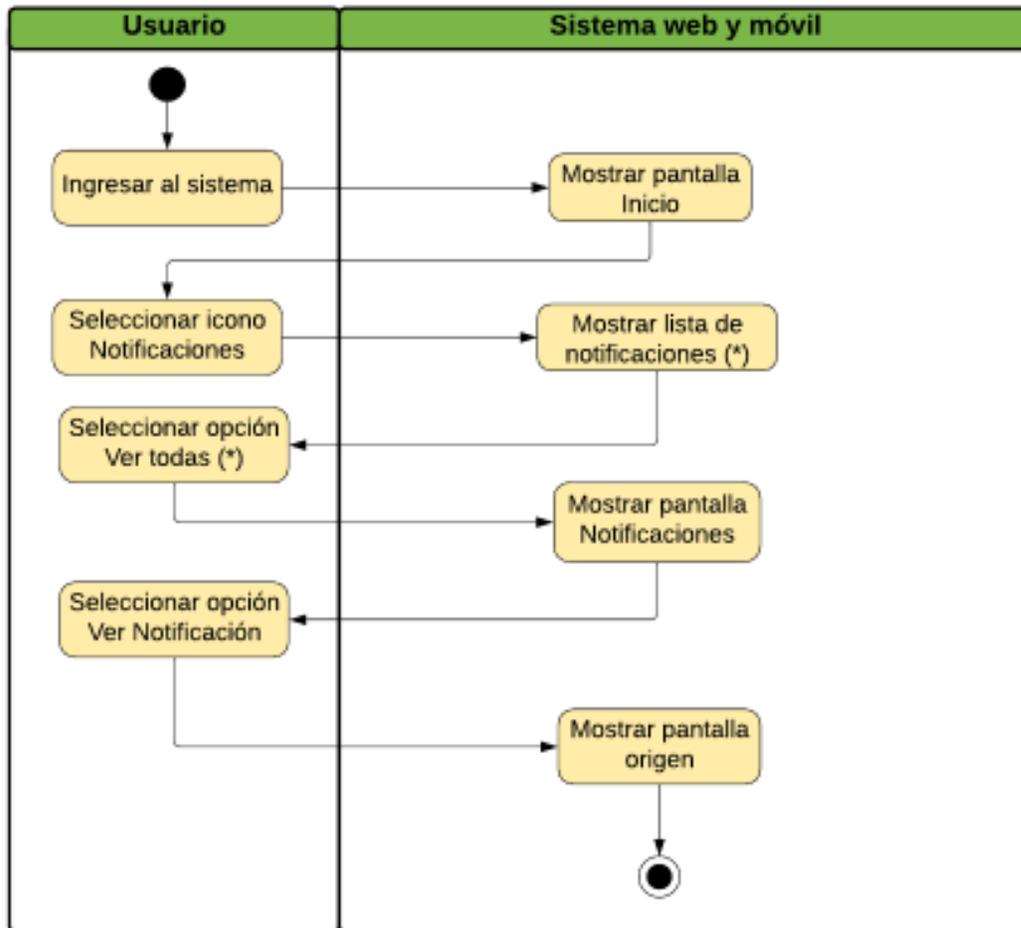
Para este módulo se ha trabajado el rol Gestor y el rol Delegado dentro de un mismo proceso, por lo tanto, para este módulo se ha definido el Proceso: Revisar notificaciones.

Proceso Revisar notificaciones

En la Figura 2.67 se indica el Diagrama de actividades del Proceso Revisar Notificaciones; éste ha sido dividido en dos grupos: Usuario y Sistema web y móvil. Este usuario ingresa al aplicativo web o al aplicativo móvil, el Sistema presenta la pantalla Inicio.

El usuario deberá seleccionar el ícono Notificaciones ubicado en la esquina superior derecha; la aplicación despliega un pequeño menú acerca de las notificaciones recibidas como se indicó en la Figura 2.12. El usuario tiene la opción **Ver todas**. Al seleccionar esta opción el Sistema muestra la pantalla Notificaciones. En esta pantalla si el usuario selecciona la opción **Ver notificación**, el Sistema muestra la pantalla origen, es decir la pantalla que generó la notificación.

Para el diagrama de actividad y el diagrama de secuencia de este proceso, el símbolo (*) son acciones que el Sistema web y móvil ejecutarán solamente en el aplicativo web. Para el aplicativo móvil estas acciones no son ejecutadas; el usuario selecciona el ícono Notificaciones y el aplicativo redirecciona directamente a la pantalla Notificaciones y continúa con el proceso indicado.



(*) Solo en Aplicación web

Figura 2.67. Diagrama de actividad para el Proceso Revisar notificaciones

En la Figura 2.68 se tiene el Diagrama de secuencia del Proceso Revisar Notificaciones, aquí el usuario independiente de su rol tendrá una visualización de la lista de notificaciones, mediante la opción **Ver todas** podrá acceder a la pantalla Notificaciones, en esta pantalla se tiene la opción **Ver notificación** la cual redirige a la pantalla que generó esa Notificación.

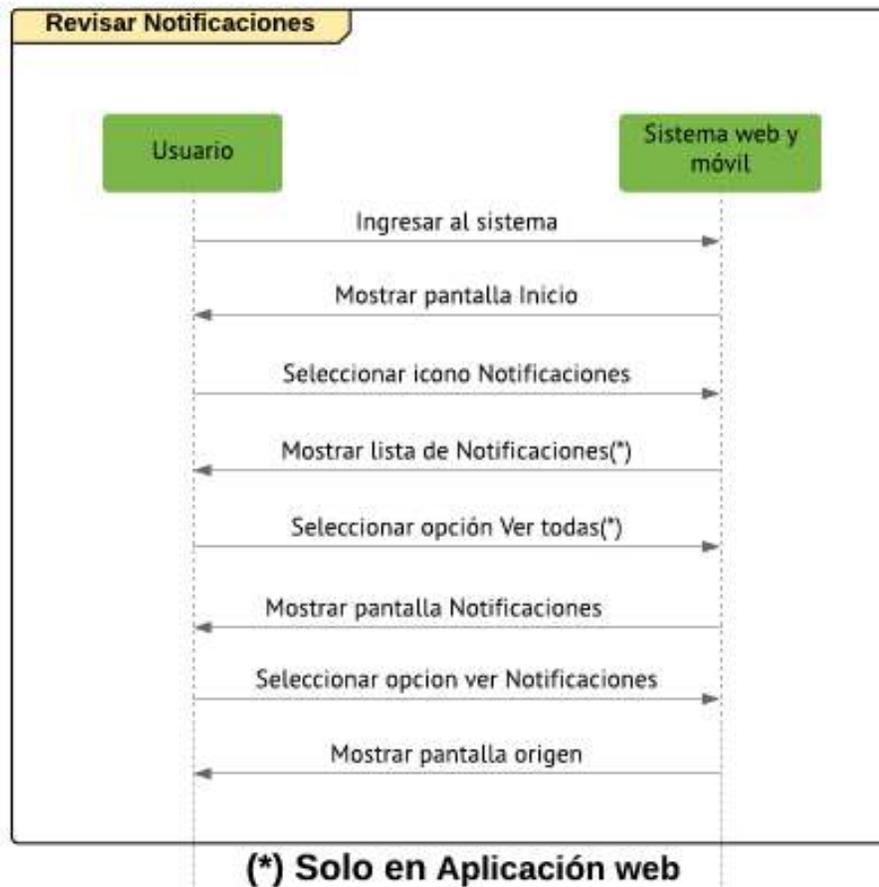


Figura 2.68. Diagrama de secuencia para el Proceso Revisar notificaciones

2.1.7 DIAGRAMA DE CLASES

Para concluir el diseño del Sistema, en la Figura 2.69 se presenta el Diagrama de clases, el cual está conformado por siete clases: la clase Project, la clase Activity, la clase Task, la clase Evidence, la clase User, la clase Team y la clase Chat.

En el diagrama de clases no se incluyen métodos debido a que en la codificación del sistema web y móvil, la cual se explicará más adelante en la sección 2.2.1.1 , se manejan modelos mas no clases, sin embargo, estos modelos se relacionan uno con otros como se indica en la Figura 2.70.

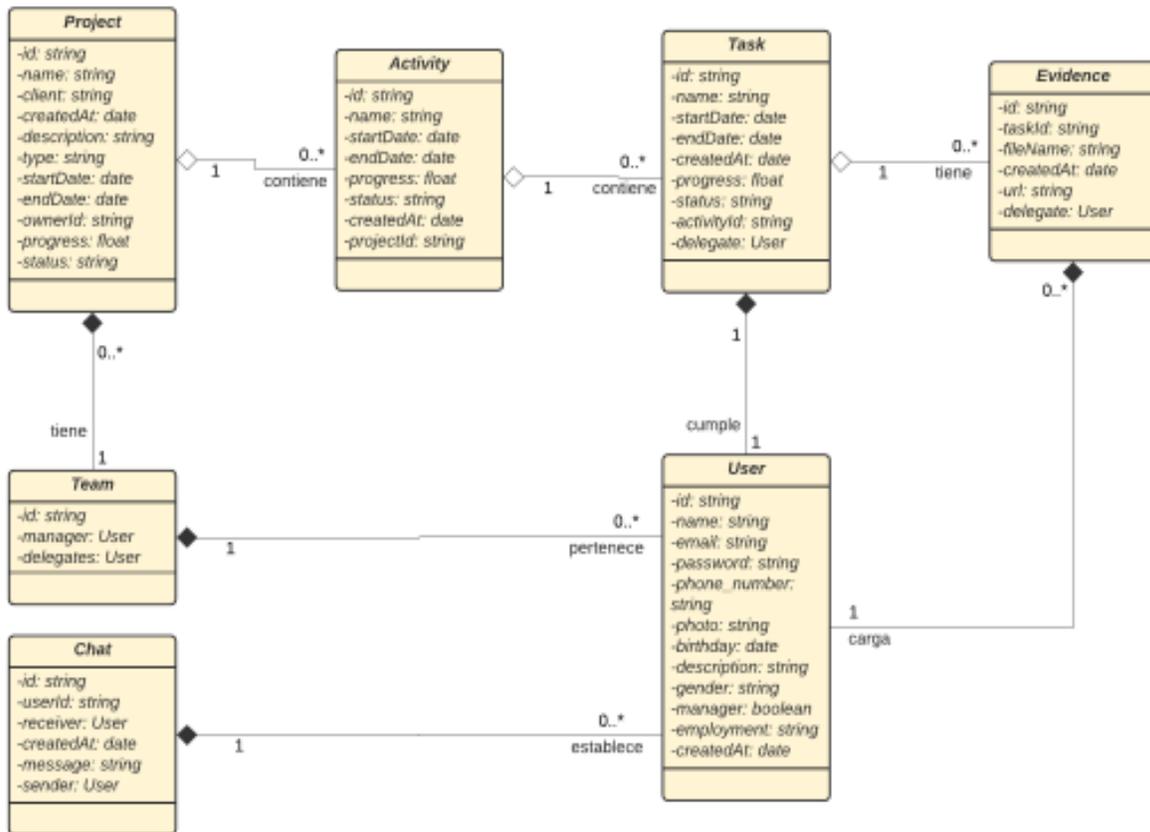


Figura 2.69. Diagrama de clases del sistema

2.1.8 ESQUEMA DE LA BASE DE DATOS

Cloud Firestore, al ser una base de datos NoSQL almacena sus datos en documentos, los mismos que son organizados en colecciones y no en tablas como en el caso de una base de datos SQL. Los documentos contienen un conjunto de atributos, además, pueden contener subcolecciones, las cuales pueden contener más documentos. A continuación, se detalla el esquema de la base de datos no relacional.

En la Figura 2.70 se presenta la esquematización de la colección users, aquí será almacenada la información personal de cada usuario y su rol.

En la Figura 2.71 se presenta la esquematización de la colección teams, aquí se almacena información de cada equipo de trabajo; cada documento contiene atributos con información del Gestor y una subcolección delegates que constituye documentos con información de los Delegados que conforman dicho equipo.



Figura 2.70. Esquema de la colección users

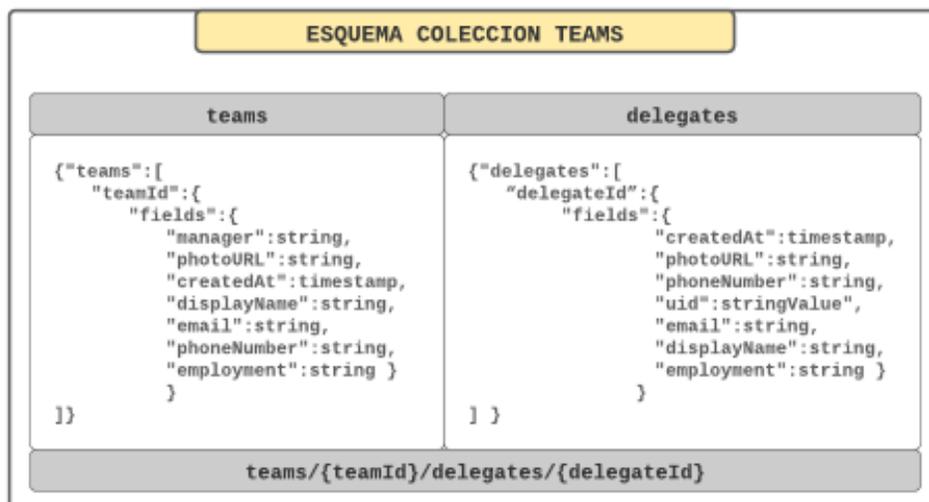


Figura 2.71. Esquema de la colección teams

En la Figura 2.72 se presenta la esquematización de la colección companies, en donde se almacena información de cada Empresa; aquí se obtiene una lista de los correos electrónicos pertenecientes a Delegados y Gestores. Si el usuario no consta en la lista de su empresa no podrá registrarse. Cabe recalcar que este proceso se lo realiza cuando el uso de la aplicación es empresarial.



Figura 2.72. Esquema de la colección companies

En la Figura 2.73 se presenta la esquematización de la colección projects, esta colección almacena información esencial del sistema.

Debido a que almacena datos que permiten llevar el control sobre los proyectos, consta de una subcolección activities, la cual contiene información acerca de las actividades de cada proyecto. Cada actividad contiene una subcolección tasks en la que se almacena información de las tareas que tiene cada actividad, el Delegado responsable y su respectivo progreso; este último es clave para el cálculo del avance de cada actividad y por consiguiente de cada proyecto.

En la Figura 2.74 se presenta la esquematización de la colección evidences, aquí se almacena información de la tarea a la que corresponde la evidencia, consta de una subcolección files, la cual está conformada por los atributos correspondientes y contiene un vínculo de los archivos almacenados en *Cloud Storage*.

En la Figura 2.75 se presenta la esquematización de la colección chats, esta colección almacena las conversaciones entre usuarios, estas conversaciones se encuentran dentro de la subcolección messages.

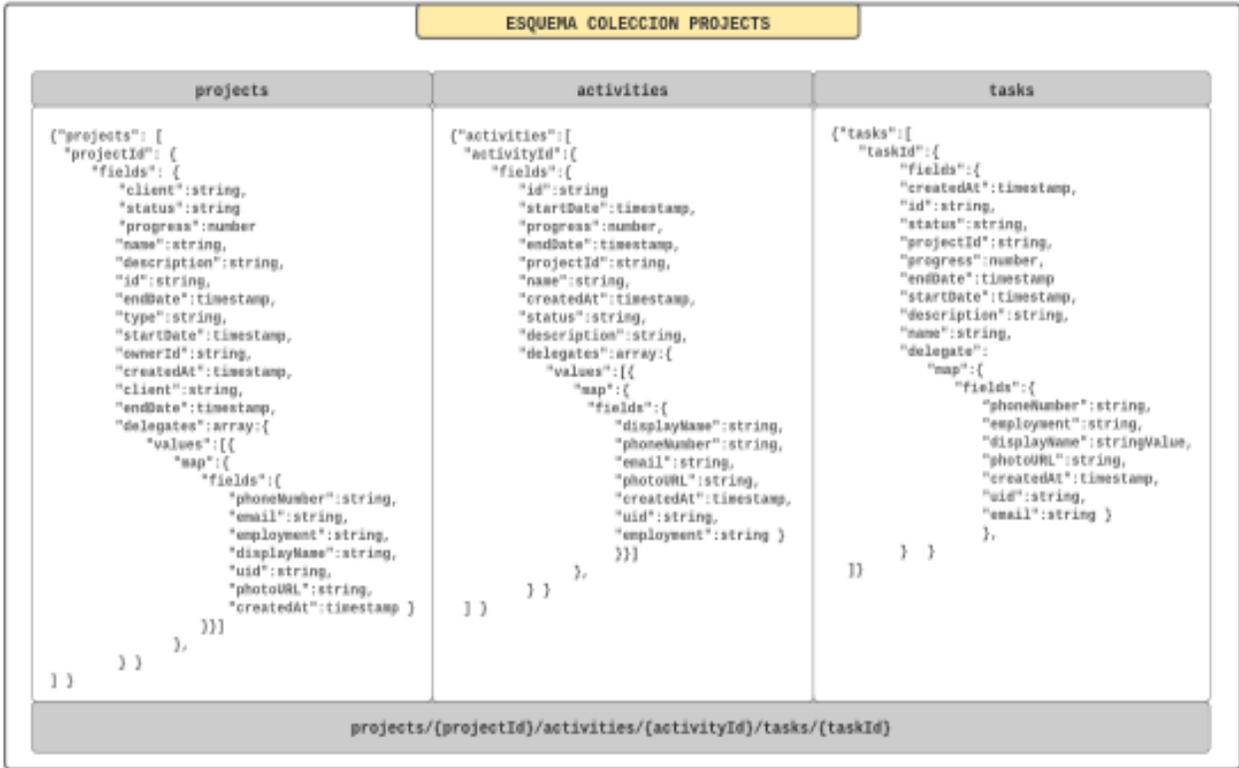


Figura 2.73. Esquema de la colección projects

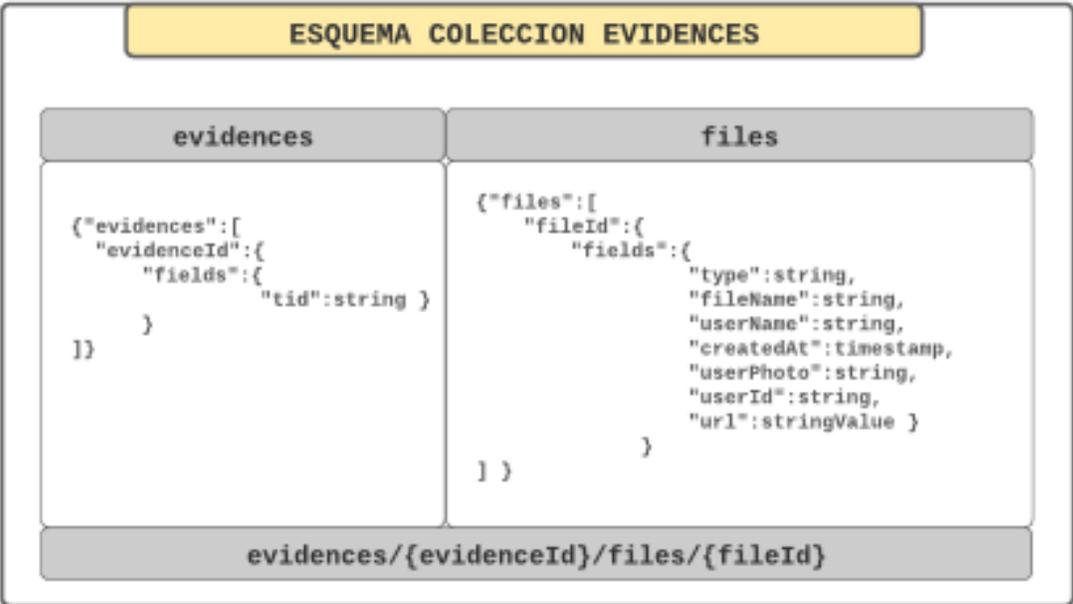


Figura 2.74. Esquema de la colección evidences

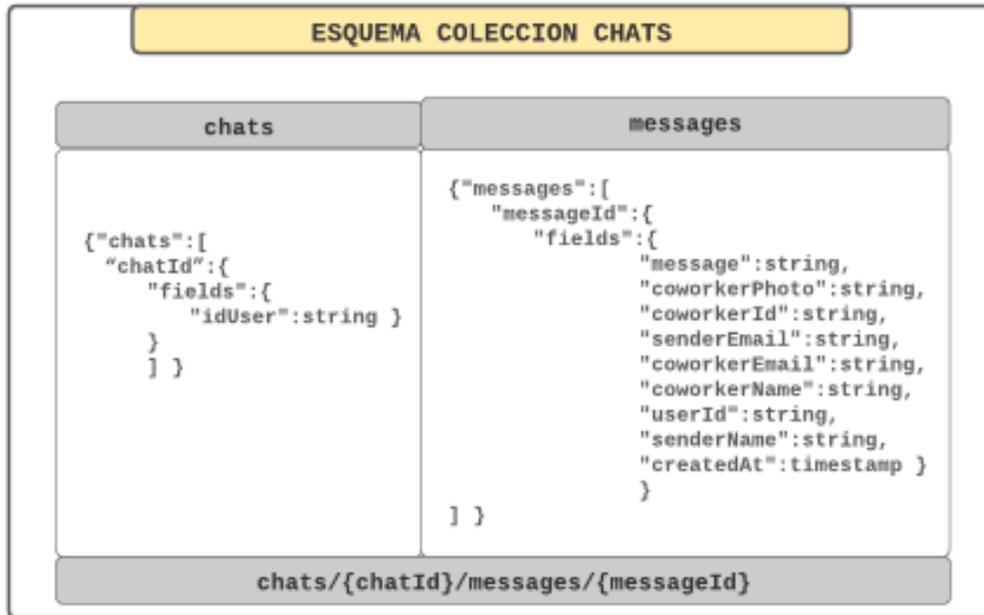


Figura 2.75. Esquema de la colección chats

2.2 IMPLEMENTACIÓN DEL SISTEMA

La implementación del Sistema consiste en convertir las especificaciones de éste en un sistema ejecutable, es decir el desarrollo del código fuente para el sistema. De acuerdo con la metodología basada en Kanban, las tareas correspondientes a la etapa de diseño se encuentran concluidas; las tareas correspondientes a la etapa de implementación **Se Están Haciendo** y las tareas correspondientes a la etapa de pruebas se encuentran **Por Hacer** como se indica en la Figura 2.76.

2.2.1 INSTALACIÓN DE LAS HERRAMIENTAS DE DESARROLLO

Para la implementación del sistema, se requiere de la instalación de algunas herramientas que son parte del desarrollo del sistema. En primera instancia, se requiere de la instalación del entorno Node.js, sobre el cual se instalarán las demás librerías mediante comandos NPM.

Para el desarrollo de este proyecto se ha utilizado el sistema operativo Windows, por lo que la implementación de Node.js se la hace mediante la descarga del paquete de instalación que proporciona su página oficial, como se indica en el recuadro azul de la Figura 2.77.



Figura 2.76. Tablero Kanban - Etapa de implementación

Una vez descargado este paquete, la instalación se la hace como cualquier instalación básica en Windows. Para verificar que ya se obtuvo el entorno Node.js y su Gestor de paquetes NPM se ejecutan los comandos que se indican en la Figura 2.78.



Figura 2.77. Paquete de instalación Node.js

```
FX XTREME@DESKTOP-KP6MQ35 MINGW64
$ node --version
v12.16.3

FX XTREME@DESKTOP-KP6MQ35 MINGW64
$ npm --version
6.14.5
```

Figura 2.78. Node.js instalado y su gestor de paquetes

A continuación, se debe instalar el respectivo CLI para Ionic y Angular. En la Figura 2.79 se indica el comando para la instalación del CLI correspondiente a Angular; una vez culminada dicha instalación se puede verificar que ésta ha sido exitosa mediante la información que se muestra en la Figura 2.80.

```
FX XTREME@DESKTOP-KP6MQ35 MINGW64
$ npm install -g @angular/cli
```

Figura 2.79. Instalación de Angular CLI

```
Angular CLI
Angular CLI: 8.1.2
Node: 12.16.3
OS: win32 x64
Angular: 8.1.2
... animations, cli, common, compiler, compiler-cli, core, forms
... language-service, platform-browser, platform-browser-dynamic
... router

Package                                  Version
-----
@angular-devkit/architect                0.803.29
@angular-devkit/build-angular            0.803.29
@angular-devkit/build-optimizer          0.803.29
@angular-devkit/build-webpack            0.803.29
@angular-devkit/core                     8.1.2
@angular-devkit/schematics               8.1.2
@angular/cdk                             8.1.1
@angular/fire                           5.4.2
@angular/flex-layout                     8.0.0-beta.26
@angular/material                        8.1.1
@angular/material-moment-adapter         8.1.1
```

Figura 2.80. Verificación de la instalación de Angular CLI

En la Figura 2.81 se indica el comando para la instalación del CLI correspondiente a Ionic; una vez culminada dicha instalación se puede verificar que ésta ha sido exitosa mediante el comando que se indica en la Figura 2.82.

```
FX XTREME@DESKTOP-KP6MQ35 MINGW64  
$ ng install -g @ionic/cli
```

Figura 2.81. Instalación de Ionic CLI

```
FX XTREME@DESKTOP-KP6MQ35 MINGW64  
$ ionic --version  
6.2.2
```

Figura 2.82. Verificación de la instalación de Ionic CLI

2.2.1.1 Codificación del sistema

Para iniciar el desarrollo de una aplicación en Angular o Ionic, se debe crear un nuevo proyecto mediante la línea de comandos respectiva. En cada proyecto, se pueden agregar modelos, servicios, páginas y componentes según los requerimientos. Además, cada proyecto cuenta con una estructura compleja y extensa, la cual está conformada por carpetas. Debido a la complejidad y extensión se explicarán solo las carpetas más relevantes.

Una de la carpetas más esenciales es `environments`, la cual contiene dos archivos, `environment.ts` utilizado en el entorno de desarrollo, donde la ejecución de la aplicación será de manera local y `environment.prod.ts` utilizado cuando la aplicación se encuentra en producción, es decir cuando está disponible en la web. Es importante mencionar que estos archivos contienen variables de entorno fundamentales, que en el caso de este proyecto son los parámetros y `keys` necesarias para establecer una conexión con Firebase.

Se tiene también la carpeta `source(src)`, es aquí donde se estructura todo el código fuente, archivos y carpetas importantes para el desarrollo de la aplicación. Entre los principales

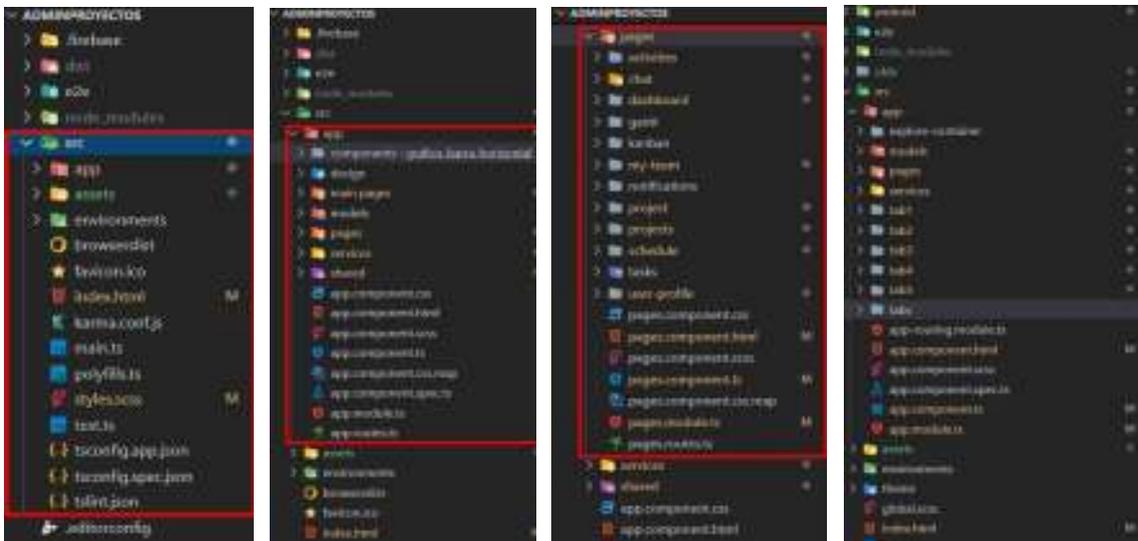
están: los archivos de estilo, el archivo principal HTML (`index.html`), las carpetas: `environments`, `assets` y `app` como se indica en la parte (a) de la Figura 2.83.

El archivo `index.html` contiene las importaciones de distintas herramientas para el proyecto, y es la página raíz donde se inicia toda la aplicación. La carpeta `assets` almacena recursos estáticos como imágenes, fuentes, iconos y estilos para importar dentro del proyecto.

En la Figura 2.83, en la parte (b), se indica la carpeta `app`, que se encuentra dentro de la carpeta `src`; `app` contiene las carpetas `pages`, `services`, `models`, `components`, entre otras, que son utilizadas para el desarrollo del código fuente.

En la parte (c) se indica el contenido de la carpeta `pages`, la cual contiene todos los módulos mencionados en la sección 2.1.6, los mismos que son estructurados en componentes de Angular para la aplicación web.

Para la aplicación móvil, se tienen carpetas extras denominadas `tabs`, como se indica en la parte (d) de la Figura 2.83; éstas proporcionan un componente de navegación de nivel inferior para permitir una navegación basada en pestañas, las cuales se detallan más adelante.



(a)

(b)

(c)

(d)

Figura 2.83. Estructuración de carpetas del sistema

2.2.2 CODIFICACIÓN DE FIREBASE

La codificación de Firebase requiere de algunos pasos, la configuración a nivel de código y la configuración a nivel de consola de Firebase.

2.2.2.1 Base de datos

La base de datos está conformada por documentos y colecciones, donde cada colección contiene subcolecciones y cada documento contiene los atributos que fueron indicados en la sección 2.1.9. En la Figura 2.84 se indica el esquema general de la base de datos del sistema. Se tienen colecciones principales que son: users, projects, teams, chats y companies.

En la Figura 2.85 se tiene la colección users, con sus respectivos documentos y cada documento con los atributos correspondientes. En la Figura 2.86 se indica la colección teams, y cada documento de esta colección tiene sus atributos y una subcolección delegates, donde se almacenan los delegados correspondientes a un documento team.

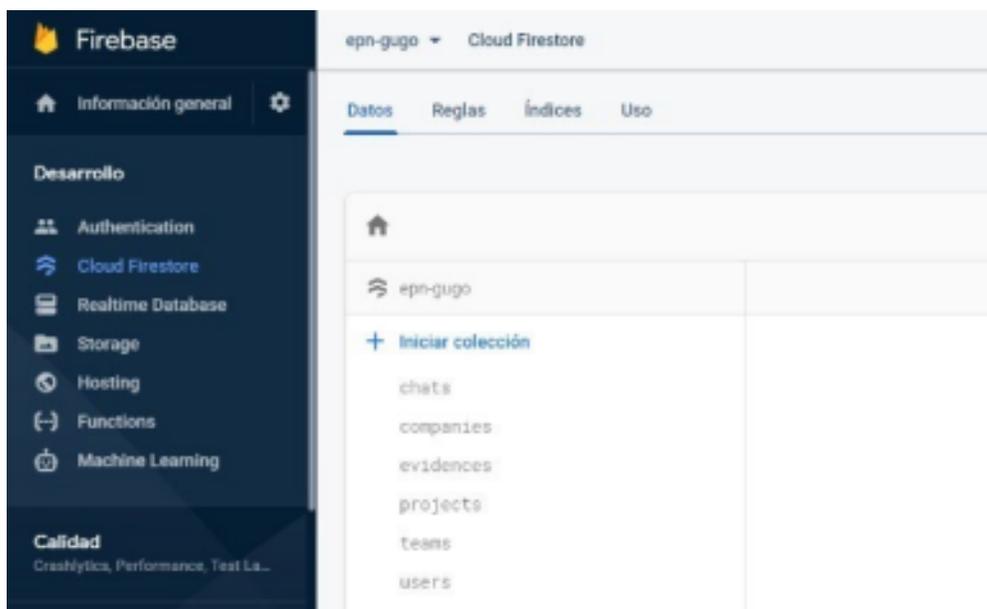


Figura 2.84. Base de datos del sistema

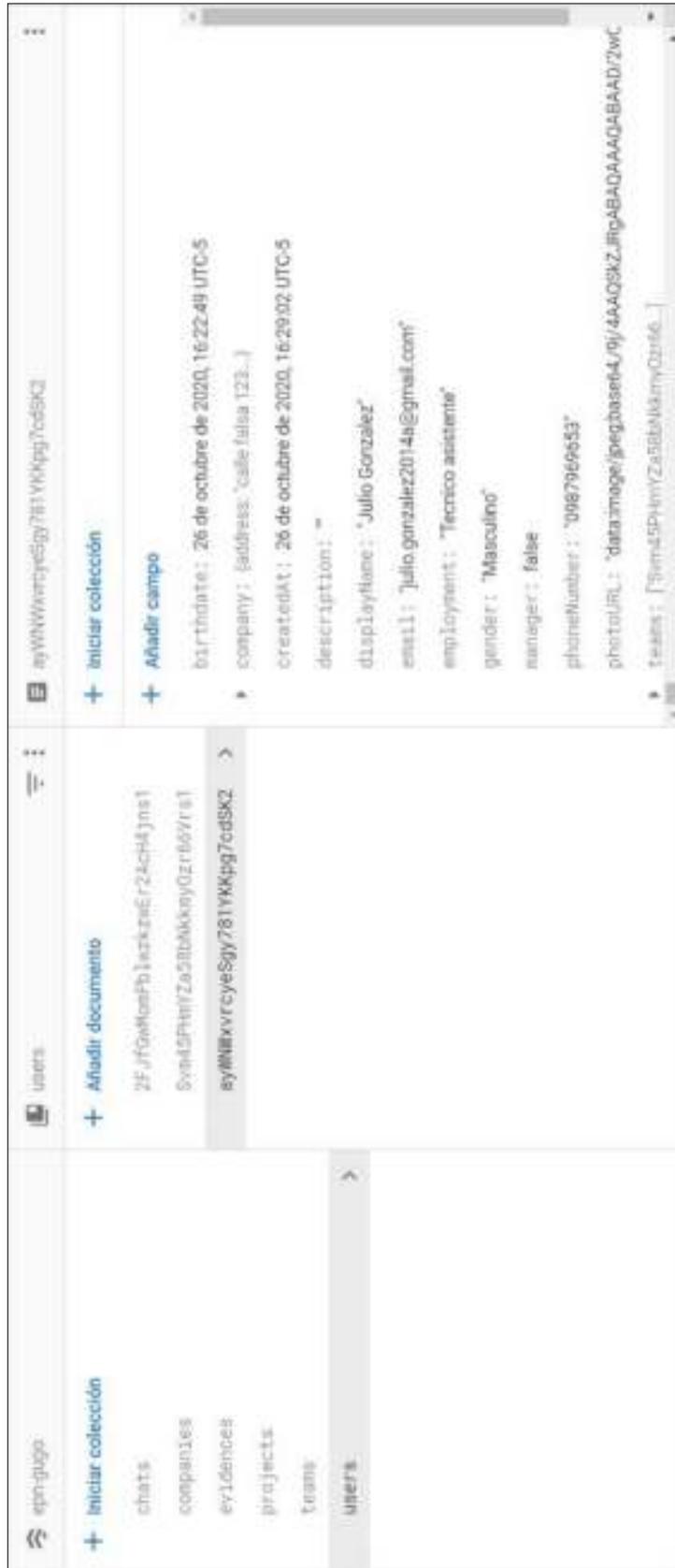


Figura 2.85. Colección users en Firestore

En la Figura 2.87 se tiene la colección projects; cada documento de ésta tiene sus atributos y una subcolección activities, en la cual se almacenan las actividades de cada proyecto y cada documento de esta subcolección cuenta con sus propiedades y una subcolección tasks, donde se alojan las tareas de dichas actividades como se indica en la Figura 2.88.

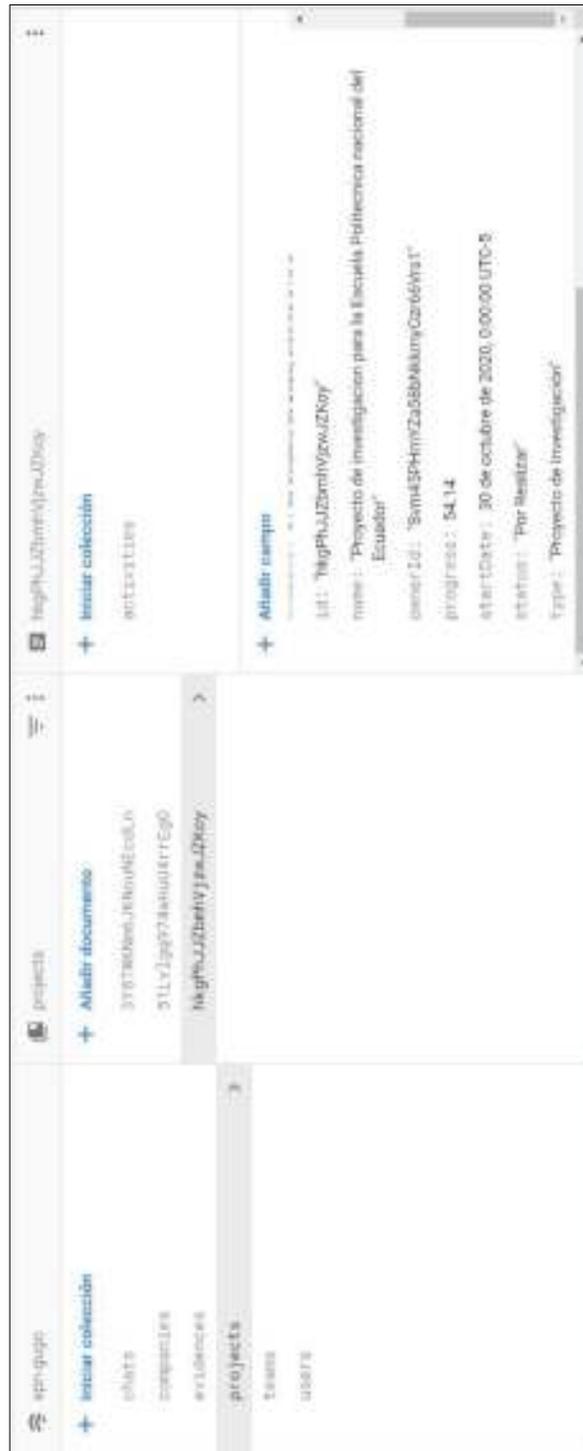


Figura 2.87. Colección projects en Firestore – Parte 1

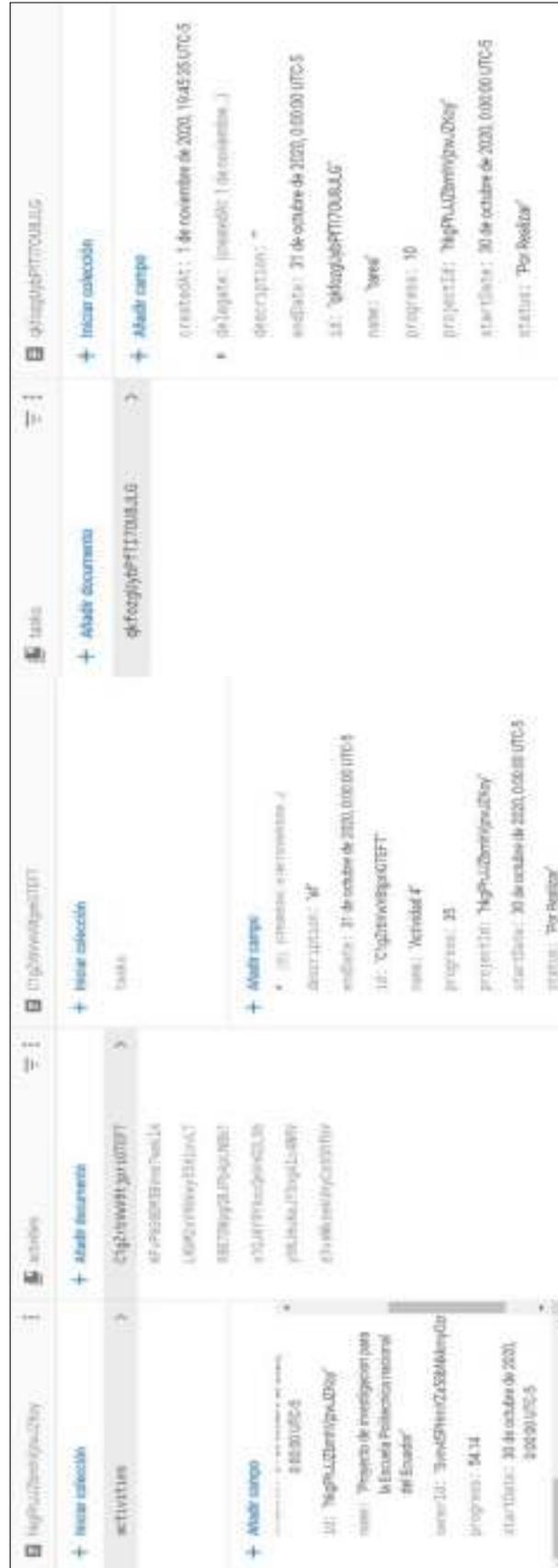


Figura 2.88. Colección projects en Firestore - Parte 2

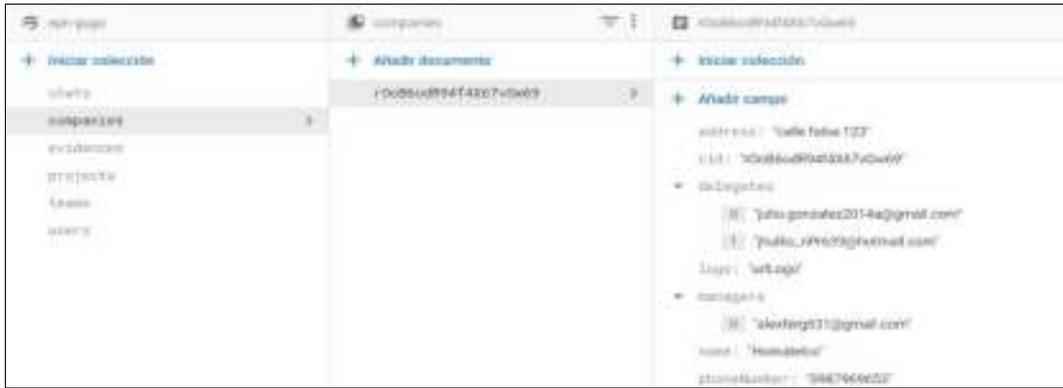


Figura 2.90. Colección companies en Firestore



Figura 2.91. Colección chats en Firestore

En la Figura 2.91 se tiene la colección chats, cada documento de esta colección tiene sus atributos y una subcolección messages; cada documento de esta subcolección cuenta con sus atributos correspondientes.

2.2.2.2 Implementación de Firebase sobre el Sistema

Para establecer la conexión con Firebase y el Sistema, se requiere de ciertas dependencias: `angularFire` y `Firebase`, las mismas que se instalan mediante línea de comandos como se indica en la Figura 2.92.

```
$ npm install --save firebase angularfire2
npm WARN deprecated angularfire2@5.4.2: AngularFire has moved, we're now @angular/fire

> core-js@3.6.5 postinstall D:\test\node_modules\core-js
> node -e "try(require('./postinstall'))catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> protobufjs@6.10.2 postinstall D:\test\node_modules\protobufjs
> node scripts/postinstall

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN @angular/fire@5.4.2 requires a peer of @angular/common@>=6.0.0 <9 || ^9.0.0-0 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/fire@5.4.2 requires a peer of @angular/core@>=6.0.0 <9 || ^9.0.0-0 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/fire@5.4.2 requires a peer of @angular/platform-browser@>=6.0.0 <9 || ^9.0.0-0 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/fire@5.4.2 requires a peer of @angular/platform-browser-dynamic@>=6.0.0 <9 || ^9.0.0-0 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/fire@5.4.2 requires a peer of firebase@>= 5.5.7 <8 but none is installed. You must install peer dependencies yourself.
npm WARN @angular/fire@5.4.2 requires a peer of rxjs@>=6.0.0 but none is installed. You must install peer dependencies yourself.
```

Figura 2.92. Instalación de las dependencias de Firebase

2.2.2.2.1 Configuración de parámetros

En la Figura 2.93 se indican los archivos de la carpeta `environment`; cada uno de los archivos contiene los parámetros necesarios para la conexión con Firebase. Dentro de cada archivo se tiene una constante `environment` del tipo `Json`, la cual contiene dos atributos; `production` y `firebase`; el primero cambia su valor dependiendo del tipo de entorno que se trabaje: desarrollo o producción, en tanto que `firebase` es un objeto del tipo `Json` que contiene la configuración del proyecto en Firebase.

```
environmentprod.ts
src > environments > environmentprod.ts > ...
...
1 export const environment = {
2   production: true,
3   firebase: {
4     apiKey: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
5     authDomain: 'epn-gugo.firebaseio.com',
6     databaseURL: 'https://epn-gugo.firebaseio.com',
7     projectId: 'epn-gugo',
8     storageBucket: 'epn-gugo.appspot.com',
9     messagingSenderId: 'XXXXXXXXXX',
10    appId: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
11    measurementId: 'XXXXXXXXXX'
12  }
13 };
14

environmentst.ts
src > environments > environmentst.ts > ...
...
1 export const environment = {
2   production: false,
3   firebase: {
4     apiKey: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
5     authDomain: 'epn-gugo.firebaseio.com',
6     databaseURL: 'https://epn-gugo.firebaseio.com',
7     projectId: 'epn-gugo',
8     storageBucket: 'epn-gugo.appspot.com',
9     messagingSenderId: 'XXXXXXXXXX',
10    appId: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
11    measurementId: 'XXXXXXXXXX'
12  }
13 };
14
15
```

Figura 2.93. Archivos de la carpeta environment

2.2.2.2.2 Establecimiento de la conexión

En el módulo raíz `app.module.ts`, se establece la conexión con Firebase y sus servicios; en este archivo se importan los siguientes módulos; `AngularFireModule`, `AngularFirestoreModule`, `AngularFireAuthModule`, `AngularFireStorage`, `AngularFireMessagingModule`, `AngularFireFunctionsModule`.

`AngularFireModule` es el módulo principal que permite inicializar la conexión.

2.2.2.3 Implementación de servicios

Como se ha mencionado antes, se hará uso de los servicios de Firebase para dar funcionalidad al sistema, esto se hace mediante la importación de los módulos mencionados en la sección 2.2.2.2.2.

Dentro de la codificación del sistema se explicó la estructuración de carpetas, aquí se tiene la carpeta `services`, ésta contiene múltiples archivos que son los que implementan los servicios de Firebase a través de métodos propios que ofrecen los módulos importados.

A continuación, se explican los servicios utilizados, su funcionalidad, su configuración y cómo fueron implementados dentro del proyecto.

2.2.2.3.1 Servicio de Autenticación

El servicio de autenticación *Firebase Authentication* permite identificar los usuarios que serán parte del sistema. Este servicio proporciona librerías y métodos necesarios para autenticar usuarios y guardarlos de manera sencilla y segura en la nube.

El primer paso para incorporar este servicio es a través de la consola de Firebase; existen algunos proveedores de inicio de sesión entre ellos Facebook, Google, Twitter o Correo electrónico/contraseña, este proyecto hace uso de autenticación basada en correo electrónico y contraseña, para esto es esencial habilitar el proveedor en la consola de Firebase. Todo esto se indica en la Figura 2.94; en la Figura 2.95 se indica la lista de usuarios que se encuentran autenticados.

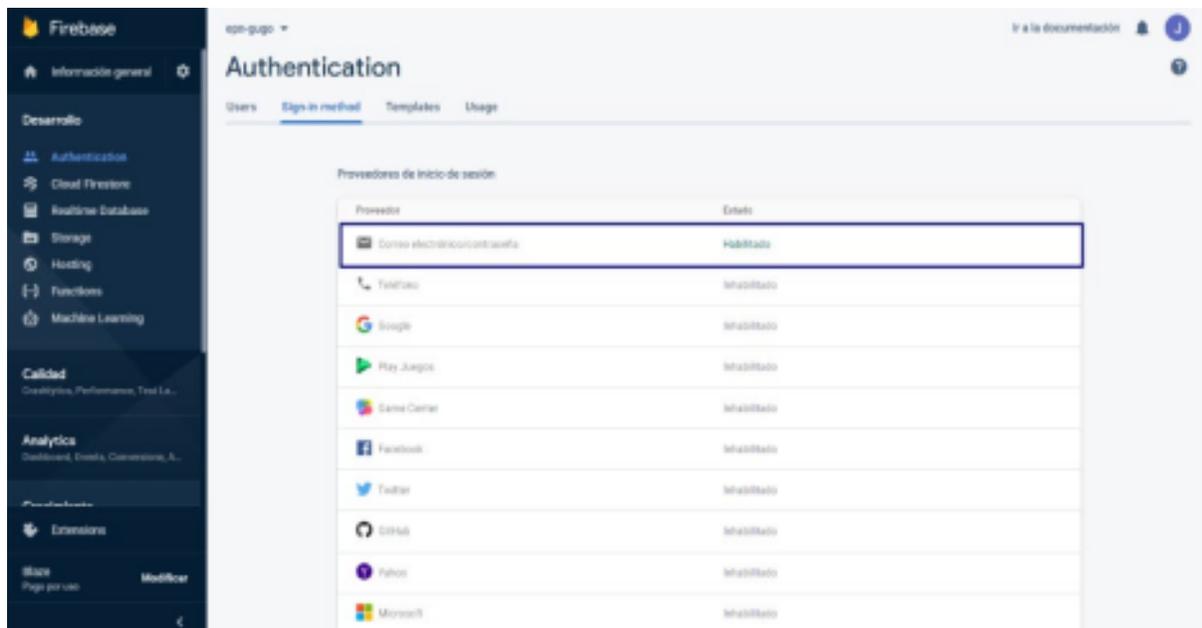


Figura 2.94. Servicio de autenticación en Firebase

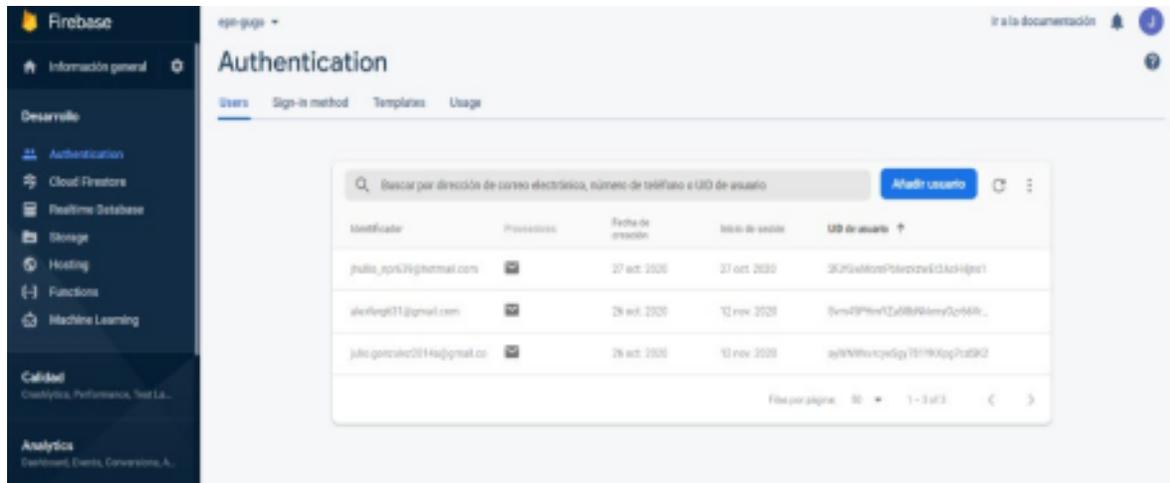


Figura 2.95. Lista de usuarios Autenticados

Como complemento, este servicio es el responsable de configurar los dominios autorizados en los que puede estar hospedado el Sistema, tal como se indica en la Figura 2.96. En este caso, el Sistema se encuentra hospedado dentro de Firebase, por ello la dirección URL indicada es la que éste proporciona.

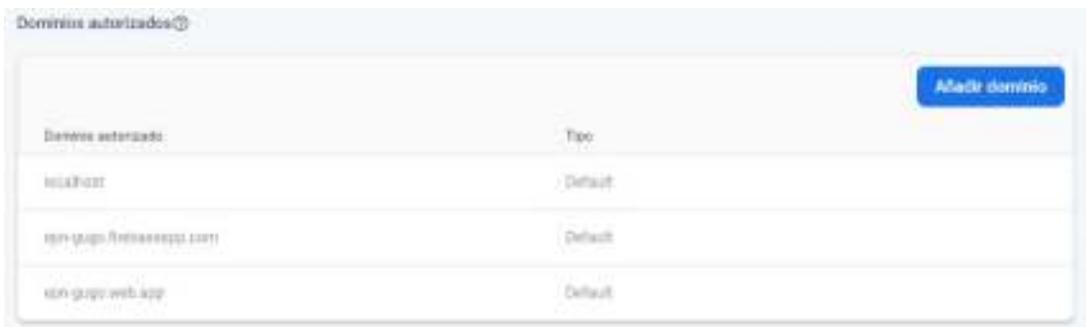


Figura 2.96. Dominios autorizados

Una vez concluida la configuración, es necesario integrar el servicio al Sistema, para ello se hace uso del módulo `AngularFireAuth` que proporciona los métodos necesarios para realizar el inicio de sesión y registro de usuarios en el sistema.

En el Código 2.1 se indica el método de registro de usuarios `register`, que permite crear nuevos usuarios a partir del correo electrónico y contraseña, los cuales son almacenados en *Firebase Authentication* a través de `createUserWithEmailAndPassword` después de haber sido verificado por todas las validaciones que impone el Sistema, esto es verificación de campos inválidos, de correo electrónico existente y de rol y uso del sistema.

```

async register(userAuth: User) {
  const { email, password } = userAuth;
  try {
    const result = await this.auth.auth.createUserWithEmailAndPassword(
      email,
      password
    );
    const { user } = result;
    return Promise.resolve(user);
  } catch (error) {
    return Promise.reject(error);
  }
}

```

Código 2.1. Código del método register

En el Código 2.2 se indica el método de inicio de sesión de usuarios `login`; el método `signInWithEmailAndPassword` envía las credenciales del usuario para que sean verificadas por *Firebase Authentication*.

```

async login(userAuth: User) {
  const { email, password } = userAuth;
  try {
    const result = await this.auth.auth.signInWithEmailAndPassword(
      email,
      password
    );
    const { user } = result;
    return Promise.resolve(user);
  } catch (error) {
    return Promise.reject(error);
  }
}

```

Código 2.2. Código del Método login

2.2.2.3.2 Servicio de Base de Datos

El servicio de base de datos permite almacenar la información de los distintos módulos que conforman el sistema. Para incorporar el servicio de base de datos se hace uso del módulo `AngularFirestoreModule` que proporciona métodos necesarios para realizar consultas de lectura, escritura, actualización y eliminación de documentos.

Es importante mencionar que los datos recuperados a través de este servicio son en tiempo real, por lo tanto, todos los módulos hacen uso de este servicio para el funcionamiento completo y óptimo del Sistema.

En el Código 2.3 se indica el método `createUser`, éste permite crear un nuevo documento en la colección `users`, siempre y cuando el usuario se haya registrado exitosamente en *Firebase Authentication*. Tiene por argumentos de entrada el usuario recuperado del formulario de registro y el identificador único que se recupera del servicio de autenticación, de esta manera se complementa con el servicio de base datos.

```
createUser(user: User, uid: string) {  
  this.afs  
    .collection('users')  
    .doc(uid)  
    .set({  
      ...user,  
    });  
}
```

Código 2.3. Código del método `createUser`

2.2.2.3.3 Servicio de Almacenamiento

El servicio de almacenamiento de archivos permite la carga y descarga de archivos de manera ágil a través de *Cloud Storage*. Para la integración con el Sistema se utiliza el módulo `AngularFireStorageModule` que proporciona los métodos necesarios para la subida de archivos en *Cloud Storage*. Aquí se tiene un contenedor de almacenamiento que permite guardar y leer las evidencias involucradas en el Módulo Proyectos, así también las imágenes de cada usuario en el Módulo perfil usuario.

El **Código 2.4** indica el Método `uploadResourcesFirebase`, éste permite subir los archivos a un proyecto específico; cada vez que se sube un archivo éste se guardará en una carpeta con el identificador del proyecto escogido.

2.2.2.3.4 Servicio de Notificaciones

El servicio de notificaciones *Cloud Messaging* es la solución más fácil de mensajería instantánea para informar a los usuarios de eventos relevantes ocurridos en el Sistema. La incorporación de notificaciones en el proyecto se hace a través del módulo `AngularFireMessagingModule`; éste permite realizar la petición de permisos al usuario para la recepción de notificaciones y extraer los *tokens* de dispositivos para luego utilizar éstos y enviar las respectivas notificaciones.

```

uploadResourcesFirebase(files: Evidence[], user: User, projectId: string) {
  const storageRef = firebase.storage().ref();
  for (const item of files) {
    item.isUploading = true;
    if (item.progress >= 100) {
      continue;
    }
    const uploadTask: firebase.storage.UploadTask = storageRef
      .child(`${this.RESOURCES_FOLDER}/${projectId}/ ${item.fileName}`)
      .put(item.file);

    uploadTask.on(
      firebase.storage.TaskEvent.STATE_CHANGED,
      (snapshot: firebase.storage.UploadTaskSnapshot) =>
        (item.progress =
          (snapshot.bytesTransferred / snapshot.totalBytes) * 100),
      (error) => console.error('Error al subir', error),
      () => {
        console.log('Imagen cargada correctamente');
        uploadTask.snapshot.ref
          .getDownloadURL()
          .then((downloadURL) => {
            console.log('File available at', downloadURL);
            item.url = this.getString(downloadURL);
          })
      })
  }
}

```

Código 2.4. Código del método uploadResourcesFirebase

Es requisito configurar un *services worker* que será el encargado de incorporar los eventos de escucha y envío de notificaciones en las aplicaciones, indicado en el Código 2.5. El Código 2.6 indica el método requestPermission, el cual realiza la petición de permisos al usuario para la recepción de notificaciones.

```

importScripts('https://www.gstatic.com/firebasejs/7.18.0/firebase-app.js');
importScripts(
  'https://www.gstatic.com/firebasejs/7.18.0/firebase-messaging.js'
);

firebase.initializeApp({
  apiKey:
  authDomain:
  databaseURL:
  projectId:
  storageBucket:
  messagingSenderId:
  appId:
  measurementId:
});

const messaging = firebase.messaging();

```

Código 2.5. Configuración del Service Worker

```

requestPermission() {
  this.afMessaging.requestPermission
    .pipe(mergeMapTo(this.afMessaging.tokenChanges))
    .subscribe(
      (token) => {
        localStorage.setItem('fcm', token);
      },
      (error) => {
        console.error(error);
      }
    );
}
}

```

Código 2.6. Código del método `requestPermission`

El Código 2.7 indica el método `receiveMessage` el cual permite recibir los mensajes de notificación, obteniendo el cuerpo de la notificación para mostrarla.

```

receiveMessage() {
  this.afMessaging.messages.subscribe((payload: any) => {
    console.log('new message received. ', payload);
    const { notification } = payload;
    this.notif = notification;
    this.currentMessage.next(payload);
  });
}

```

Código 2.7. Código del método `receiveMessage`

2.2.3 CODIFICACIÓN DE LA INTERFAZ GRÁFICA

Para la codificación de los esquemas de la interfaz gráfica, se ha hecho uso de los lenguajes HTML y SCSS/CSS. A continuación, se indica la codificación de dichos esquemas y más adelante se detalla de manera general la implementación a través de código.

2.2.3.1 Aplicación web

Para la implementación de la interfaz web se ha empleado *Material Design* para Angular; éste proporciona una librería de estilos llamada Angular Material, la cual permite integrarse perfectamente con dicho *framework*. Esta librería debe ser instalada mediante el comando que se indica en la Figura 2.97.


```

72 table {
73   width: 100%;
74   background: #rgb(12, 27, 34);
75   border: 3px solid #black;
76   margin: 0px auto;
77   color: #rgb(212, 212, 212);
78 }
79
80 thead {
81   background: #rgb(212, 212, 212);
82   color: #black;
83   border-bottom: solid 3px #black;
84 }
85
86 .title {
87   font-size: 18px;
88 }
89
90 tr:nth-child(even) {
91   background-color: #rgb(42, 48, 61);
92   color: #rgb(212, 212, 212);
93 }
94
95 td {
96   padding: 10px;
97   font-size: 15px;
98   text-align: center;
99   align-items: center;
100 }
101
102 mat-icon {
103   font-size: 20px;
104   text-align: center;
105 }

```

Código 2.9. Código del archivo projects.component.css

```

import { NewProjectModalComponent } from '../components/newProject/newProject-modal.component';
import { EditProjectModalComponent } from '../components/editProject/editProject-modal.component';
import { untilDestroyed } from '@orchestrator/ngx-until-destroyed';
import { OpenEvidenceDelegateModalComponent } from 'src/app/components/openEvidenceDelegate/openEvidenc
// src/app/projects/projects.component.ts
// last, a week ago | 3 authors (You and others)
@Component({
  selector: 'app-projects',
  templateUrl: './projects.component.html',
  styleUrls: ['./projects.component.scss']
})
export class ProjectsComponent implements OnInit, OnDestroy {

  constructor( public authService: AuthService,
               public projectService: ProjectService,
               public teamService: TeamService,
               public dialog: MatDialog
               ) { }

  team: string[] = [];
  userApp: User;
  delegates: User[] = [];
  isLoading = true;
  isLoadingTeam = true;
  projectsApp: Project[] = [];
  activitiesProjectsApp: Activity[] = [];
  tasksActivitiesApp: Task[] = [];
  tasksActivitiesApp1: Task[] = [];
  projectsOfDelegate: Project[] = [];

```

Código 2.10. Código del archivo projects.component.ts

En el Código 2.11 se indica el método `openNewProject`, el cual permite abrir el modal Proyecto, para crear un nuevo proyecto y almacenarlo en la base de datos.

```

openNewProject() {
  const dialogConfig = new MatDialogConfig();
  dialogConfig.disableClose = true;
  dialogConfig.autoFocus = false;
  dialogConfig.width = '700px';
  dialogConfig.panelClass = 'custom-dialog';

  const dialogRef = this.dialog.open(NewProjectModalComponent, dialogConfig);

  dialogRef.afterClosed().subscribe(
    data => {
      console.log('Dialog output:', data);
      const newProject: Project = {
        name: data.name as string,
        client: data.client as string,
        description: data.description as string,
        startDate: data.startDate as Date,
        endDate: data.endDate as Date,
        type: data.typeProj as string,
        ownerId: this.authService.userAuth.uid,
      };
      if (newProject.name) {
        this.projectService.addProject(newProject);
        Swal.fire({
          allowOutsideClick: false,
          icon: 'success',
          title: 'Proyecto agregado con éxito'
        });
      }
    });
}

```

Código 2.11. Código del método openNewProject

En el Código 2.12 se indica el método onUploadFiles que permite almacenar los archivos que el Delegado ha subido en el modal Evidencia. Esto se logra mediante el llamado al método uploadResourceFirebase implementado en el Código 2.4.

```

onUpdateFiles(files: FileList) {
  for (let index = 0; index < files.length; index++) {
    const evidence: Evidence = {
      file: files.item(index),
      fileName: files.item(index).name,
      size: files.item(index).size,
      type: files.item(index).type,
      tid: this.projectId
    };
    this.filesToUploadFirebase.push(evidence);
  }
  this.evidenceService.uploadResourcesFirebase(this.filesToUploadFirebase, this.delegate, this.projectId);
  this.onClearFiles();
}

```

Código 2.12. Código del método onUploadFiles

En la Figura 2.98 se encuentra la pantalla Proyectos para el rol Gestor que se ha implementado mientras que en la Figura 2.99 se encuentra la pantalla Proyectos para el rol Delegado; en la Figura 2.100 se indica el modal Proyecto que proporciona esta pantalla

y en la Figura 2.101 se indica el modal Evidencia para que el Delegado pueda subir sus archivos.



Figura 2.98. Interfaz de la pantalla Proyectos para el rol Gestor



Figura 2.99 Interfaz de la pantalla Proyectos para el rol Delegado



Figura 2.100. Interfaz del modal Proyecto



Figura 2.101. Interfaz del modal Evidencia para el rol Delegado

2.2.3.2 Aplicación móvil

Para la implementación de la interfaz gráfica de la aplicación móvil se han empleado los componentes propios de Ionic. Al igual que Angular, en Ionic se manejan componentes para la implementación de cada pantalla.

A continuación, se presenta un ejemplo de la implementación de la interfaz gráfica para la pantalla Chat y algunos métodos que permiten darle funcionalidad. Todas las interfaces implementadas en el aplicativo móvil se encuentran detalladas en el ANEXO C.

En la aplicación móvil, en lo que corresponde a la parte gráfica de cada componente se manejan dos secciones: *ion-header* y *ion-content*. En *ion-header* se coloca el nombre específico de cada pantalla, y los atajos hacia Perfil y Notificaciones mientras que en *ion-content* se coloca el cuerpo de la visualización de cada pantalla.

En el Código 2.13 se indica la sección *ion-header* y en el Código 2.14 se indica la sección *ion-content*, ambas secciones pertenecen al archivo `chat.page.html` y se proporciona estilo mediante el archivo `chat.page.scss` que se presenta en la Código 2.15. En el Código 2.16 se indica el archivo `chat.page.ts`, el cual hace llamados a los respectivos métodos para dar funcionalidad a esta pantalla.

```
1 <ion-header>
2   <ion-toolbar>
3     <div class="box align">
4       <ion-title>Cronogramas</ion-title>
5       <ion-buttons slot="end">
6         <ion-button id="notification-button">
7           <ion-badge id="notifications-badge" color="primary">5</ion-badge>
8           <ion-icon name="notifications"></ion-icon>
9         </ion-button>
10        <ion-button id="notification-button1">
11          <ion-avatar id="notification-button1">
12            <img [src]="userApp.photoURL">
13          </ion-avatar>
14        </ion-button>
15      </ion-buttons>
16    </div>
17  </ion-toolbar>
18 </ion-header>
```

Código 2.13. Código del archivo `chat.pages.html` – Segmento *ion-header*

```
19 <ion-content [fullscreen]="true" color="secondary">
20   <div class="container1">
21     <div *ngIf="userApp.manager == true; else delegate">
22       <ion-list>
23         <div *ngFor="let project of projectsAll; let i = index">
24           <ion-item color="dark" tappable (click)="projectRoute(project.id)"
25             [ngClass]="{'section-active': project.active, 'section': !project.active}"
26             <ion-icon slot="start" name="alarm-outline"></ion-icon>
27             <ion-label class="container">
28               <h2>[[project.name]]</h2>
29               <p style="color: red;">Días restantes: {{ diffDays[i] }}</p>
30               <div class="align">
31                 <ion-progress-bar color="primary" [value]="project.progress / 100"></ion-progress-bar>
32                 <p>[[project.progress]]%</p>
33               </div>
34               <p>Fecha de entrega: {{ allendates[i] | date: "dd/MM/y"}}</p>
35             </ion-label>
36           </ion-item>
37         </div>
38         <div *ngIf="post == true && project.active">
39           <ion-item>
40             <div class="scrolling-box">
41               <google-chart [data]="timelineChartData"></google-chart>
42             </div>
43           </ion-item>
44         </div>
45       </ion-list>
46     </div>
47   </div>
48 </ion-content>
```

Código 2.14. Código del archivo `chat.pages.html` – Segmento *ion-content*

```

6     .user_name {
7       color: #afafaf;
8     }
9     .message {
10      padding: 10px !important;
11      transition: all 250ms ease-in-out !important;
12      border-radius: 10px !important;
13      margin-bottom: 4px !important;
14    }
15    .my_message {
16      padding: 10px !important;
17      transition: all 250ms ease-in-out !important;
18      border-radius: 10px !important;
19      margin-bottom: 4px !important;
20      color: white !important;
21    }
22    ion-item {
23      right: 0;
24    }
25    .line-breaker {
26      white-space: pre-line;
27      font-size: 14px;
28    }

```

Código 2.15. Código del archivo chat.pages.scss

```

16 import { IonContent } from '@ionic/angular';
17
18 You, seconds ago | 2 authors (Julio and others)
19 @Component({
20   selector: 'app-chat',
21   templateUrl: './chat.page.html',
22   styleUrls: ['./chat.page.scss'],
23 })
24 export class ChatPage implements OnInit {
25   Julio, 6 months ago • Version 0.0.9
26 @ViewChild('content', { static: false }) content: IonContent;
27   userApp: User;
28   id: string;
29   coworker: any;
30   constructor( private route: ActivatedRoute,
31               private authService: AuthService,
32               public teamService: TeamService,
33               private chatService: ChatService ) {
34   }
35   ngOnInit() {
36     this.authService.getUser(this.authService.userAuth)
37       .subscribe(user => {
38         (this.userApp = user); });
39     this.id = this.route.snapshot.paramMap.get('id');
40     this.route.queryParams.subscribe(params => {
41       this.coworker = JSON.parse(params['coworker']);
42     });

```

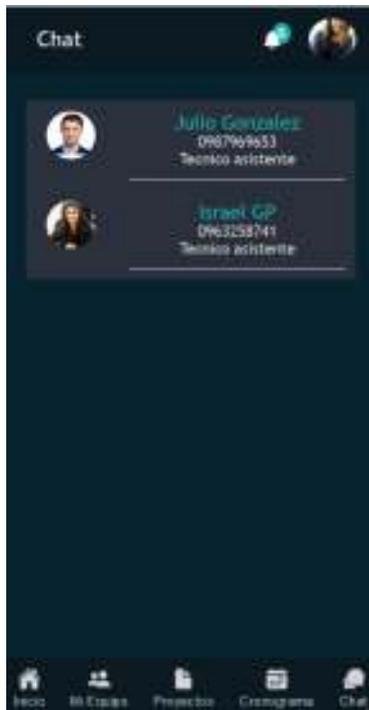
Código 2.16. Código del archivo chat.pages.ts

En el Código 2.17 se indica el método `getChats`, el cual permite obtener toda la conversación con un usuario específico, a su vez realizando un ordenamiento por fechas para una mejor visualización.

```
getChats() {  
  this.chatService.getChats(this.id).subscribe(chats => {  
    this.allChats = chats;  
    try {  
      this.allChats.sort((a, b) => a.createdAt['seconds'] - b.createdAt['seconds']);  
    } catch {}  
    setTimeout(() => {  
      this.content.scrollToBottom(300);  
    }, 20);  
  });  
}
```

Código 2.17. Código del método `getChats`

En la parte (a) de la Figura 2.102 se indica la interfaz de la pantalla Chat y en la parte (b) se indica la conversación con un usuario seleccionado de la parte (a).



(a)



(b)

Figura 2.102. (a) Interfaz de la pantalla Chat y (b) Conversación entre usuarios

2.2.3.2.1 Implementación de PWA

A continuación, se indica la configuración que se realizó en la aplicación móvil para convertirla en una aplicación PWA, para esto se ha codificado el *services worker* y *web manifest*.

Primero, se debe instalar la dependencia `@angular/pwa` como se indica en la Figura 2.103. Esta dependencia crea el archivo `ngsw-config.json`, el cual tiene el esquema del *service worker*; también se crea el archivo `manifest.webmanifest` e instala un icono de distintos tamaños que será tomado como icono de la aplicación si a ésta dentro del `webmanifest` no se le ha asignado un ícono personalizado.

En la Figura 2.104 se indica el archivo `manifest.webmanifest` que contiene el ícono del sistema que se indicó en la Figura 2.5; este icono se encuentra en distintos tamaños para adaptarse a la resolución del dispositivo móvil. En la Figura 2.105 se indican las diferentes resoluciones que se le asignan al ícono.

Cuando se instala la dependencia `@angular/pwa` se añaden las rutas del *webmanifest* y del *service worker* al archivo `angular.json`, y también se importa el módulo `ServiceWorkerModule` dentro del archivo `app.module.ts`. Esto permite que el *service worker* funcione y se registre en el navegador web

```
FX XTREME@DESKTOP-KPGMQ35 MINGW64 /d/Tesis-adminProy
$ ng add @angular/pwa
Installing packages for tooling via yarn.
Installed packages for tooling via yarn.
CREATE ngsw-config.json (624 bytes)
CREATE src/manifest.webmanifest (1330 bytes)
CREATE src/assets/icons/icon-128x128.png (1253 bytes)
CREATE src/assets/icons/icon-144x144.png (1394 bytes)
CREATE src/assets/icons/icon-152x152.png (1427 bytes)
CREATE src/assets/icons/icon-192x192.png (1790 bytes)
CREATE src/assets/icons/icon-384x384.png (3557 bytes)
CREATE src/assets/icons/icon-512x512.png (5008 bytes)
CREATE src/assets/icons/icon-72x72.png (792 bytes)
CREATE src/assets/icons/icon-96x96.png (958 bytes)
UPDATE angular.json (5620 bytes)
UPDATE package.json (3184 bytes)
UPDATE src/app/app.module.ts (1906 bytes)
UPDATE src/index.html (948 bytes)
```

Figura 2.103. Instalación de la dependencia PWA angular

```
manifest.webmanifest X
src > {} manifest.webmanifest > [ ] icons > {} 4 > src
You, an hour ago | 1 author (You)
1 {
2   "name": "Gufo-Mobile",
3   "short_name": "Gufo",
4   "theme_color": "#1976d2",
5   "background_color": "#fafafa",
6   "display": "standalone",
7   "gcm_sender_id": "242212842353",
8   "scope": "./",
9   "start_url": "./",
10  "icons": [{
11    "src": "assets/icons/icon-72x72.png",
12    "sizes": "72x72",
13    "type": "image/png",
14    "purpose": "maskable any"
15  },
16  {
17    "src": "assets/icons/icon-96x96.png",
18    "sizes": "96x96",
19    "type": "image/png",
20    "purpose": "maskable any"
21  },
22  {
23    "src": "assets/icons/icon-128x128.png",
24    "sizes": "128x128",
25    "type": "image/png",
26    "purpose": "maskable any"
27  },
```

Figura 2.104. Archivo manifest.webmanifest de la aplicación PWA



Figura 2.105. Icono de la aplicación en diferentes resoluciones

2.2.4 DESPLIEGUE DEL SISTEMA

Para publicar el sistema es necesario hacer un despliegue utilizando el servicio de *Firebase Hosting*, éste permitirá alojar los archivos que se crean cuando las aplicaciones cambian del modo de desarrollo al modo de producción mediante los comandos correspondientes.

Existen dos métodos para que la aplicación web pase al modo de producción; JIT (*Just-in-time*) y AOT (*Ahead-on-time*). JIT permite ahorrar el tiempo en compilar el código *TypeScript* ya que esto se lo hace al momento de ejecución, para lo cual se debe incorporar

el compilador de Angular que es bastante pesado. AOT requiere más tiempo de compilación, pero ahorra el espacio del compilador y en modo de producción ofrece un código precompilado, libre de errores y listo para ser usado de inmediato.

AOT es el método utilizado para la aplicación, mediante el comando `ng build --prod --aot` como se indica en la Figura 2.106.

Por otro lado, Ionic tiene un método predeterminado para pasar al modo de producción, esto se lo hace a través del comando `ionic build --prod` indicado en la Figura 2.107

```
$ ng build --prod --aot

chunk {0} runtime.b744c98b11dac124ec19.js (runtime) 3.11 kB [entry] [rendered]
chunk {1} main.722d7e8af2a6eab6a5df.js (main) 2.89 MB [initial] [rendered]
chunk {2} polyfills.c640b608146d16a0f572.js (polyfills) 68.2 kB [initial] [rendered]
chunk {3} polyfills-es5.2c449ae1e52f9932568c.js (polyfills-es5) 151 kB [initial] [rendered]
chunk {4} styles.987d9081f801c47ffa34.css (styles) 1.65 MB [initial] [rendered]
chunk {5} 5.987d9081f801c47ffa34.css, 5.559590280ce7ad47c49e.js () 1.65 MB [rendered]
chunk {6} 6.e9decf330e7fc86b0b66.js () 41.2 kB [rendered]
Date: 2020-11-27T12:29:51.374Z - Hash: ec529649ac9cc3dff007 - Time: 208884ms
```

Figura 2.106. Construcción del aplicativo web en producción

```
$ ionic build --prod
> ng.cmd run app:build:production
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.

chunk {2} runtime-es2015.0121385ef083402e1b1b.js (runtime) 4.13 kB [entry] [rendered]
chunk {2} runtime-es5.0121385ef083402e1b1b.js (runtime) 4.13 kB [entry] [rendered]
chunk {11} polyfills-es2015.6d9170a30a91a55a499a.js (polyfills) 36.3 kB [initial] [rendered]
chunk {0} common-es2015.6f93b26b0e207a8dbb93.js (common) 6.05 kB [rendered]
chunk {0} common-es5.6f93b26b0e207a8dbb93.js (common) 9.94 kB [rendered]
chunk {4} 4-es2015.f996a88874718207ff6d.js () 65.6 kB [rendered]
```

Figura 2.107. Construcción del aplicativo móvil en producción

Para construir los aplicativos finales es necesario especificar en el archivo `angular.json` la ruta en donde se guardarán los archivos que genera, mediante los comando mostrados en la Figura 2.106 y en la Figura 2.107. Esta ruta se indica en el recuadro de color rojo de la Figura 2.108 para la aplicación web y en la Figura 2.109 para la aplicación móvil.

en Firebase a través de la consola presentada en la Figura 2.111, esto permitirá acceder a la cuenta donde están ubicados los proyectos que se ha realizado en Firebase.

Posteriormente se configura el proyecto con el comando `firebase init`, éste presenta los servicios disponibles. En este caso se seleccionará el servicio de *hosting* como se indica en la Figura 2.112.

```
FX XTREME@DESKTOP-KP6MQ35 MINGW64 /d/Tesis-admin
$ firebase login
Already logged in as cesar14gonzalez@gmail.com
```

Figura 2.111. Inicio de sesión en Firebase a través de la consola

```
FX XTREME@DESKTOP-KP6MQ35 MINGW64 /d/Tesis-adminProyectos/Tesis/Admin_mobile/Admin
$ firebase init

#####      ##      #####      #####      #####      ##      #####      #####
##          ## ##      ## ##      ##      ## ##      ## ##      ##
#####      ##      #####      #####      #####      #####      #####      #####
##          ## ##      ## ##      ##      ## ##      ## ##      ## ##
##          #####      ##      #####      #####      ##      ##      #####      #####

You're about to initialize a Firebase project in this directory:

  D:\Tesis-adminProyectos\Tesis\Admin_mobile\AdminProyectos-mobile

Before we get started, keep in mind:

  * You are initializing in an existing Firebase project directory

? Are you ready to proceed? Yes
? Which Firebase CLI features do you want to set up for this folder? Press Space to
  ( ) Database: Deploy Firebase Realtime Database Rules
  ( ) Firestore: Deploy rules and create indexes for Firestore
  ( ) Functions: Configure and deploy Cloud Functions
  >(*) Hosting: Configure and deploy Firebase Hosting sites
  ( ) Storage: Deploy Cloud Storage security rules
```

Figura 2.112. Selección del servicio de Firebase Hosting

Para poder alojar la aplicación web y la aplicación móvil en el mismo proyecto de Firebase se requiere de la configuración de diferentes sitios web. Por defecto, *Firebase Hosting* presenta un espacio para alojar aplicativos llamado sitio web y es aquí donde se va a desplegar la aplicación desarrollada en Angular como se presenta en la Figura 2.113. Mientras que para la aplicación móvil se ha creado un nuevo sitio web llamado *mobile-epn-gugo* indicado en la Figura 2.114.



Figura 2.113. Sitio Web epn-gugo



Figura 2.114. Sitio móvil mobile-epn-gugo

Firebase hosting requiere de un nombre con el cual vincular su sitio web, este nombre es definido en el archivo `firebase.json` mediante la propiedad *target* que contendrá el conjunto de archivos obtenidos en producción, a los cuales se hace referencia mediante la ruta definida en la propiedad *public*. Esto se presenta en la Figura 2.115 para el aplicativo web y en la Figura 2.116 para el aplicativo móvil.

Hecha esta configuración es necesario vincular el sitio web con el *target* de los archivos, para esto se usa el comando indicado en la Figura 2.117.

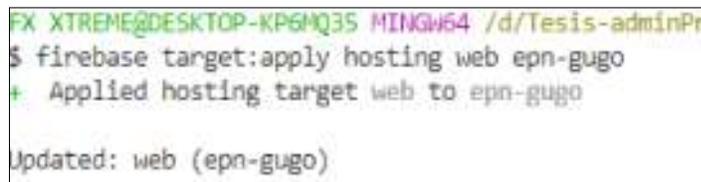
```
firebase.json
firebase.json > ...
1 {
2   "hosting": {
3     "target": "web",
4     "public": "dist/epn-gugo"
5   }
6 }
```

Figura 2.115. Archivo de configuración para el aplicativo web



```
firebase.json X
firebase.json > ...
1 {
2   "hosting": {
3     "target": "mobile",
4     "public": "www"
5   }
6 }
```

Figura 2.116. Archivo de configuración para el aplicativo móvil



```
FX XTREME@DESKTOP-KP6MQ35 MINGW64 /d/Tesis-adminPr
$ firebase target:apply hosting web epn-gugo
+ Applied hosting target web to epn-gugo

Updated: web (epn-gugo)
```

Figura 2.117. Comando para indicar donde se hará el despliegue

Luego de esta configuración solamente queda por realizar el despliegue en Firebase, a través del comando `firebase deploy --only hosting`. Si el proceso se ha realizado satisfactoriamente se obtiene como resultado la URL de cada aplicación. La URL obtenida para la aplicación web es <https://epn-gugo.web.app> y para la aplicación móvil es <https://mobile-epn-gugo.web.app> como se indica en la Figura 2.118 y Figura 2.119 respectivamente.



```
$ firebase deploy --only hosting

--- Deploying to 'epn-gugo'...

i deploying hosting
i hosting[epn-gugo]: beginning deploy...
i hosting[epn-gugo]: found 77 files in dist/epn-gugo
+ hosting[epn-gugo]: file upload complete
i hosting[epn-gugo]: finalizing version...
+ hosting[epn-gugo]: version finalized
i hosting[epn-gugo]: releasing new version...
+ hosting[epn-gugo]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/epn-gugo/overview
Hosting URL: https://epn-gugo.web.app
```

Figura 2.118. Despliegue de la aplicación web en el hosting de Firebase

```
FX XTREME@DESKTOP-KPSM315 MINGW64 /d/Tesis-adminProyectos/Tesis/Admin_mobile/Admin
$ firebase deploy --only hosting

--- Deploying to 'epn-gugo'...

i deploying hosting
i hosting[mobile-epn-gugo]: beginning deploy...
i hosting[mobile-epn-gugo]: found 1487 files in www
+ hosting[mobile-epn-gugo]: file upload complete
i hosting[mobile-epn-gugo]: finalizing version...
+ hosting[mobile-epn-gugo]: version finalized
i hosting[mobile-epn-gugo]: releasing new version...
+ hosting[mobile-epn-gugo]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/epn-gugo/overview
Hosting URL: https://mobile-epn-gugo.web.app
```

Figura 2.119. Despliegue de la aplicación móvil en el hosting de Firebase

2.2.4.1 Instalación de la aplicación móvil en un Smartphone

Cuando se accede a la URL de la aplicación móvil, ésta muestra la opción **Agregar GuGo a la pantalla principal** en la parte inferior de la pantalla del dispositivo, como se indica en la parte (a) de la Figura 2.120. Al seleccionar esta opción, la aplicación pregunta al usuario si se desea instalar en el dispositivo como se indica en la parte (b), esto permite agregar la aplicación a la pantalla principal del dispositivo y hacer uso de ella cuando se requiera.

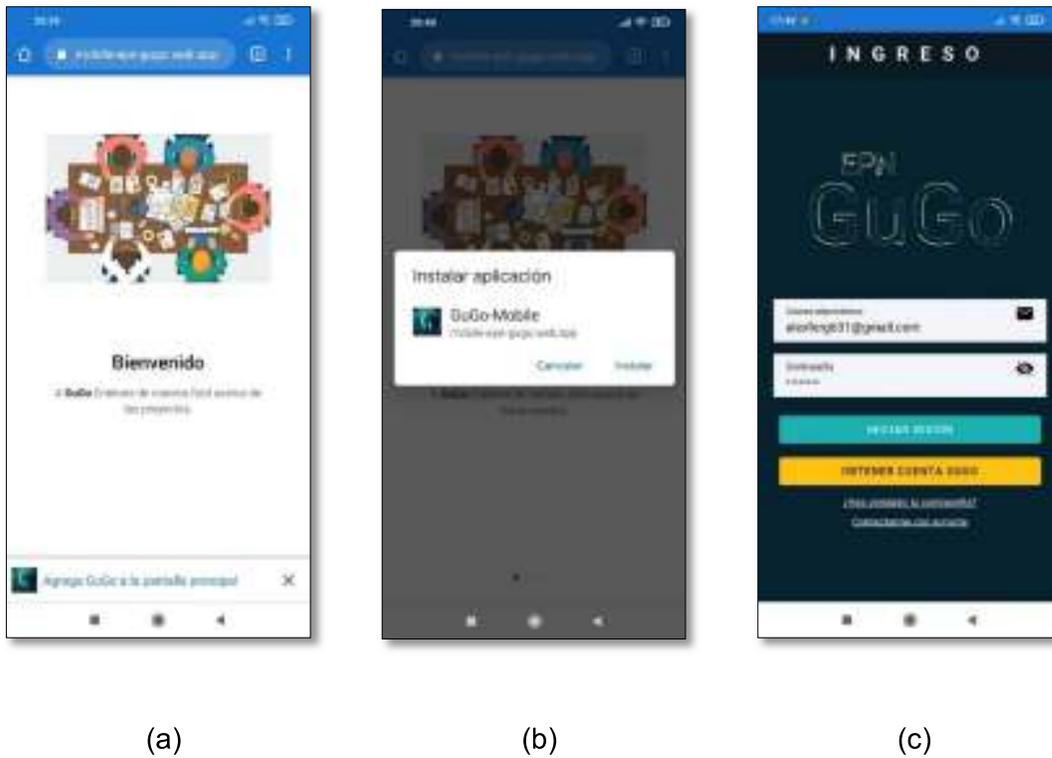


Figura 2.120. Instalación de la aplicación móvil en un Smartphone

3 RESULTADOS Y DISCUSIÓN

Con la finalidad de validar el sistema implementado, en esta sección se presentan los resultados de las pruebas de funcionalidad más relevante obtenidas a través de la evaluación del sistema final y la corrección de errores del sistema.

Previo a realizar las pruebas de funcionalidad del sistema, en la Figura 3.1 se puede visualizar el estado del tablero Kanban correspondiente a este capítulo. Las tareas correspondientes al diseño y a la implementación se encuentran concluidas, mientras que las tareas correspondientes a este capítulo se encuentran en el estado **haciendo** y finalmente ya no se tienen tareas **por hacer**.

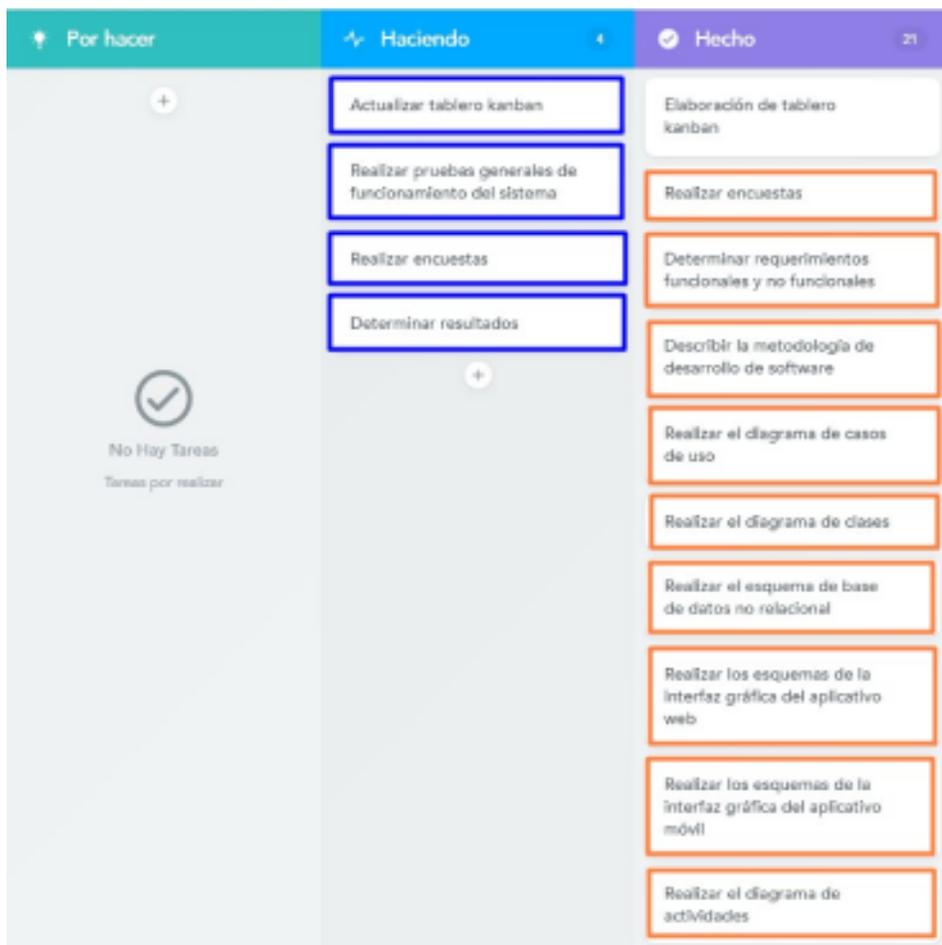


Figura 3.1. Actualización tablero Kanban – Etapa de Resultados Parte 1

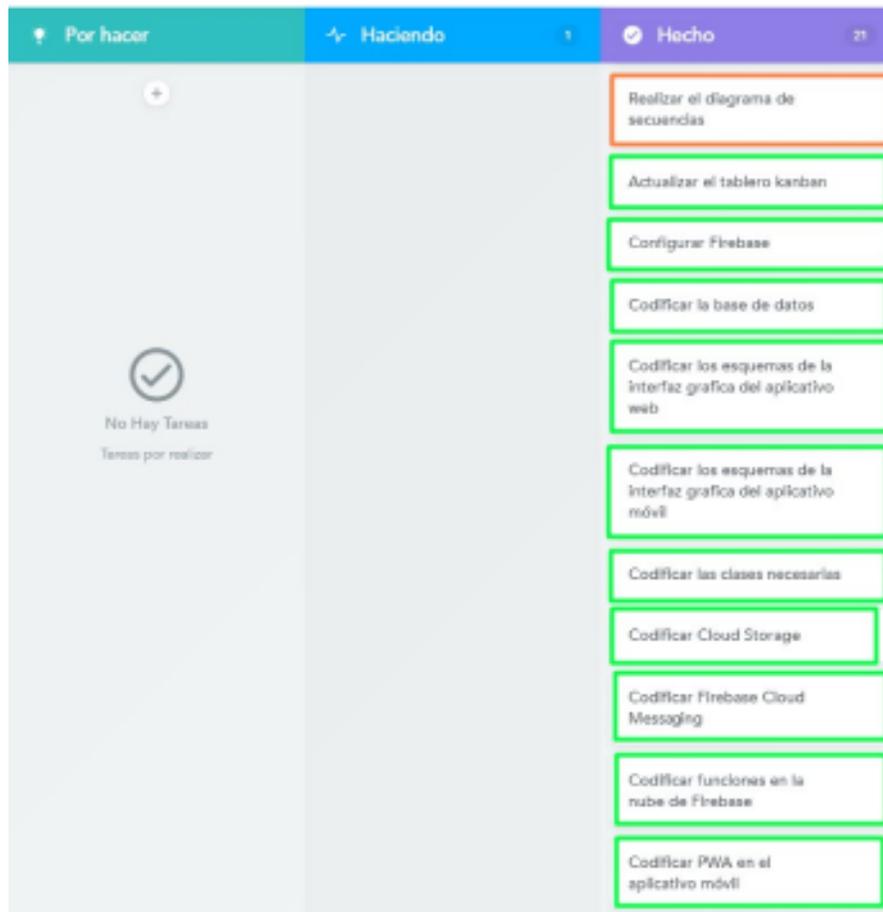


Figura 3.2. Actualización tablero Kanban – Etapa de Resultados Parte 2

Las pruebas de funcionamiento del sistema se las realizó en un navegador Chrome y en un dispositivo con sistema operativo Android; la versión mínima del dispositivo para que la aplicación móvil se ejecute es Android Lollipop 5.0 – 5.1.1. Las pruebas del sistema corresponden a: la funcionalidad individual de cada aplicación, la comunicación entre ambas aplicaciones, así como la comunicación con los servicios de Firebase.

3.1 PRUEBAS INDIVIDUALES

Las pruebas individuales se las ha realizado en cada aplicación perteneciente al sistema y consiste en verificar las funcionalidades de los módulos mencionados en la sección 2.1.6.

3.1.1 MÓDULO REGISTRO

El módulo registro permite al usuario obtener una cuenta GuGo para acceder al sistema web y móvil; mediante el rol especificado en el registro otorga al usuario las funcionalidades correspondientes. Además, el usuario debe especificar el uso de la aplicación.

En la Figura 3.3 se indica el formulario de registro para la aplicación web y para la aplicación móvil.

Figure 3.3 displays two versions of a registration form. (a) Web application: The form is titled 'Registro' and includes fields for 'Nombre Completo' (Julio González), 'Correo electrónico' (julio.gonzalez@epn.edu.ec), 'Contraseña', 'Confirmar Contraseña', and 'Número Celular' (0987163161). It also features radio buttons for 'Gestor de proyectos' (selected) and 'Delegado', and a dropdown menu for 'Uso de la aplicación' (Empresarial). A 'Listo' button is at the bottom. (b) Mobile application: The form is titled 'REGISTRO' and includes fields for 'Nombre Completo' (Homero Gualotuña), 'Correo electrónico' (homero.g@homatelco.com), 'Contraseña', 'Confirmar contraseña', and 'Número celular' (0963578743). It also features radio buttons for 'Gestor de proyectos' (selected) and 'Delegado', and a dropdown menu for 'Uso de la aplicación' (Empresarial). A 'LISTO' button is at the bottom.

Figura 3.3. Formulario de Registro (a) Aplicación web y (b) Aplicación móvil

Para el formulario de registro, el sistema realiza la validación de cada uno de los campos del formulario Registro, a medida que el usuario llena cada uno de ellos. Si en los datos de registro existe algún campo inválido, el sistema retorna un mensaje de advertencia como

se indica en la Figura 3.4, correspondiendo la parte (a) a la aplicación web y la parte (b) a la aplicación móvil.

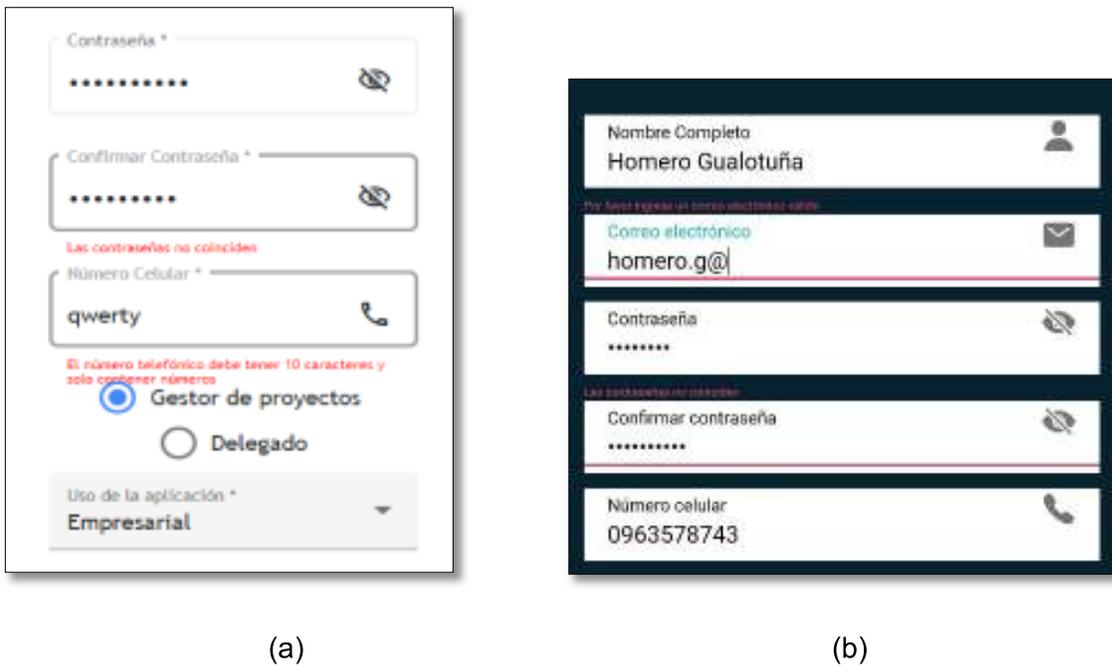


Figura 3.4. Validación de datos (a) Aplicación web y (b) Aplicación móvil

Si el uso de la aplicación es empresarial, el sistema verifica que el usuario se encuentre previamente registrado en la base de datos y que el rol especificado en el registro coincida con el rol almacenado en ésta, de lo contrario el sistema muestra el mensaje de error que se observa en la parte (c) de la Figura 3.5.

Luego, el sistema realiza la verificación de correo; si el correo registrado ya existe, el usuario no podrá registrarse y el sistema indicará el mensaje de error que se observa en la Figura 3.6.

Si el uso de la aplicación es personal, el sistema solo verifica que el correo electrónico no exista y valida los campos de registro; si no existen errores de ningún tipo, el sistema emite un mensaje exitoso como se observa en la Figura 3.7 y el usuario se lo registrará dentro del sistema con su respectivo uso y rol.

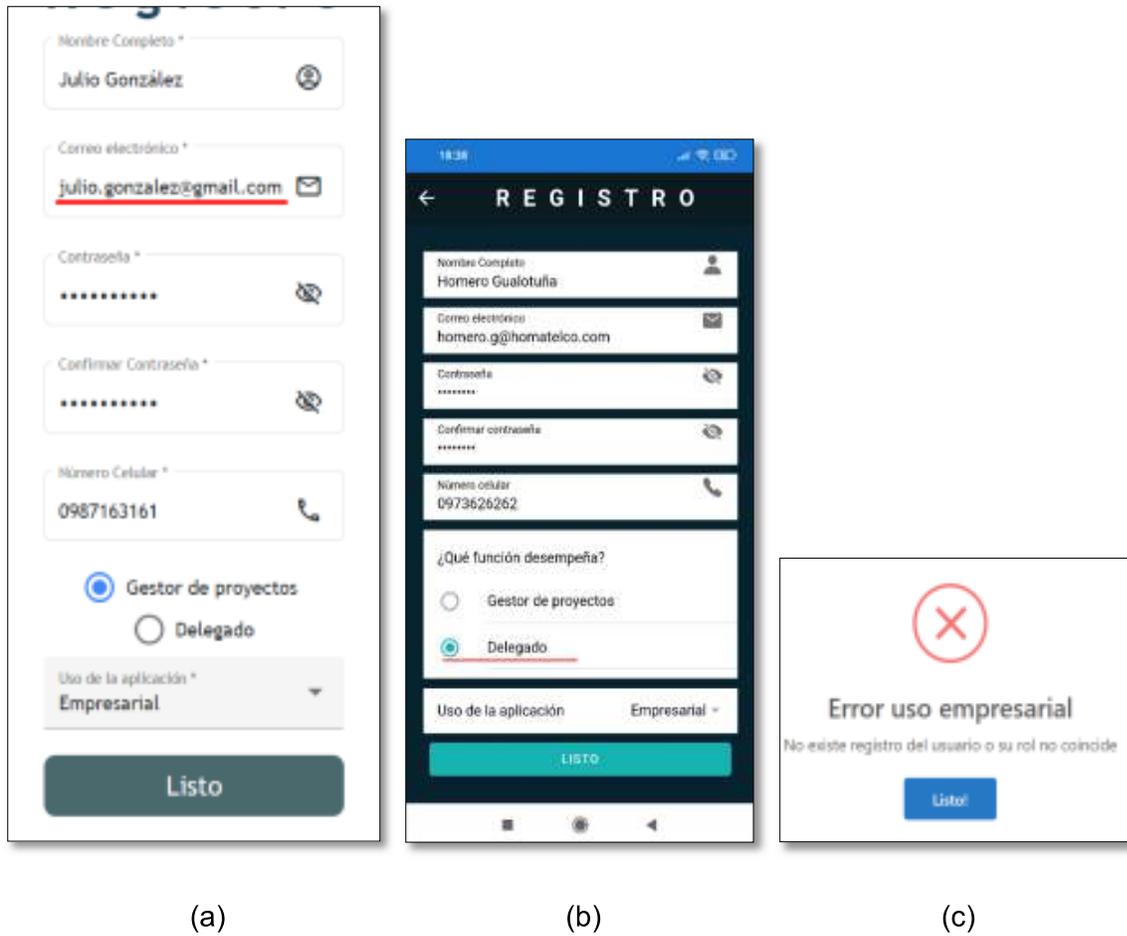


Figura 3.5. Error del uso empresarial

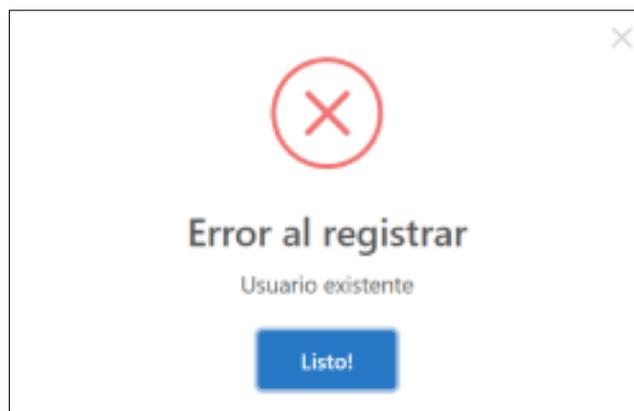
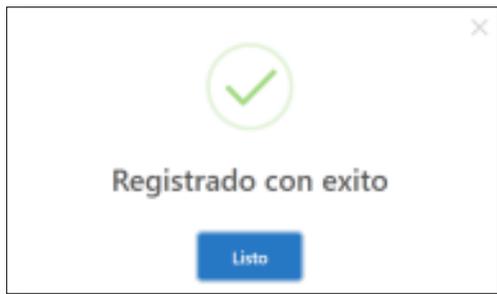


Figura 3.6. Error de verificación de correo electrónico



(a)



(b)

Figura 3.7. Registro Exitoso (a) Aviso y (b) Correo electrónico recibido

3.1.2 MÓDULO AUTENTICACIÓN

El módulo autenticación permite al usuario ingresar al sistema mediante sus credenciales. En la Figura 3.8 se indica el formulario de autenticación para la aplicación web y la aplicación móvil respectivamente.



(a)



(b)

Figura 3.8. Formulario de autenticación (a) Aplicación web y (b) Aplicación móvil

Si el correo electrónico o contraseña son erróneas, el sistema indica un mensaje de error como se presenta en la Figura 3.9.

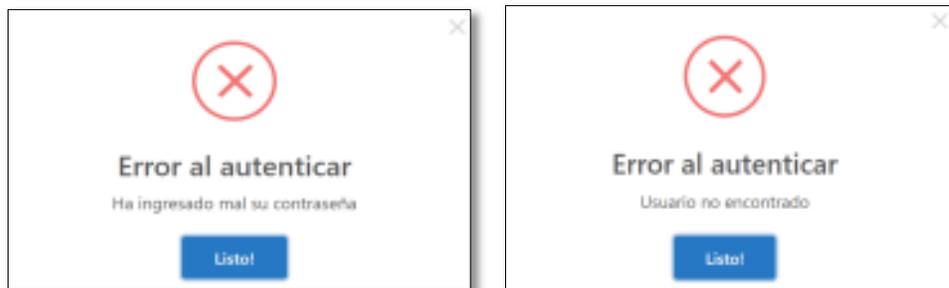
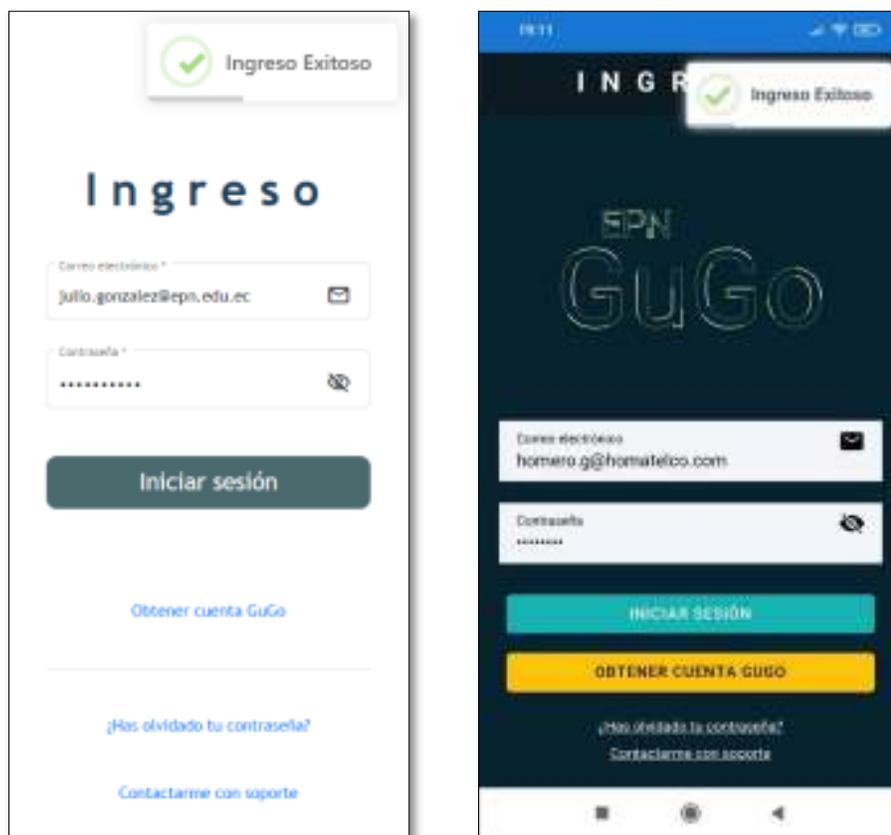


Figura 3.9. Error de credenciales (a) Contraseña errónea y (b) Correo electrónico erróneo

Si las credenciales son correctas se ingresa al sistema y se indica un mensaje de éxito como se presenta en la Figura 3.10.



(a)

(b)

Figura 3.10. Autenticación exitosa (a) Aplicación web y (b) Aplicación móvil

Además, el formulario de autenticación permite recuperar la contraseña en caso de haberla olvidado, como se indica en la Figura 3.11. Al ingresar el correo electrónico, el sistema remite un mensaje hacia este correo electrónico, el mismo que redirecciona a un pequeño formulario que permite la generación de una nueva contraseña; esto se presenta en la Figura 3.12.

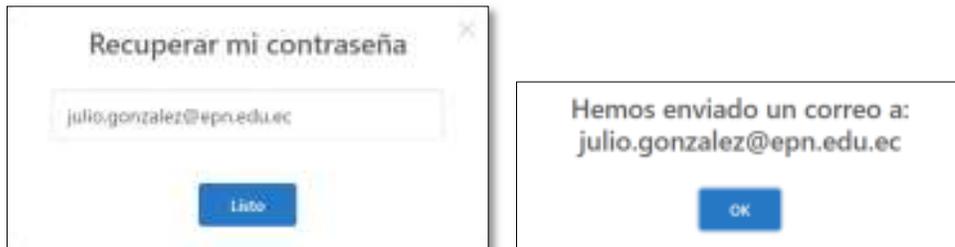


Figura 3.11. Formulario de recuperación de contraseña

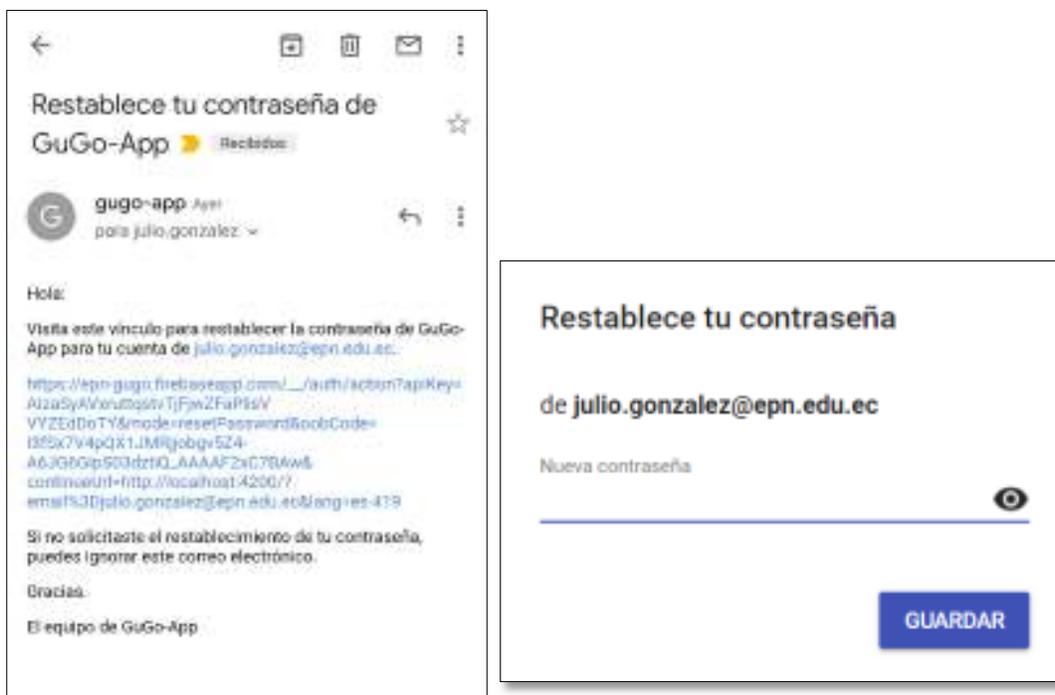


Figura 3.12. Formulario de generación de nueva contraseña

La opción Contactarme con soporte permite al usuario comunicarse con uno de los desarrolladores del sistema, con distintos propósitos: reporte de alguna anomalía en el sistema o registro de una lista de usuarios para el uso empresarial como se presenta en la Figura 3.13 para la aplicación web y en la Figura 3.14 para la aplicación móvil.

Contactarme con soporte

Recuerda que puedes contactarte con nosotros si deseas registrar a tu empresa o si tienes problemas con tu cuenta

Nombre completo *
Homero Gualotufa

Correo electrónico *
homero.g@homatalco.com

Asunto *
Registro de empresa

Mensaje *
Buenas tardes,
El motivo de este correo es para registrar mi empresa en su sistema, espero su respuesta con el debido proceso a seguir.
Muchas gracias.

Todos los campos (*) deben ser llenados

Enviar

Figura 3.13. Formulario de Contactarme con soporte - Aplicación Web

CONTACTO

Recuerda que puedes contactarte con nosotros si deseas registrar a tu empresa o si tienes problemas con tu cuenta

Nombre completo *
Homero Gualotufa

Correo electrónico *
homero.g@homatalco.com

Asunto *
Registro de empresa

Mensaje *
Buenas tardes, deseo registrar mi e

Enviar

Figura 3.14. Formulario de Contactarme con soporte - Aplicación Móvil

Cuando el usuario ingresa al sistema, en primera instancia se indica la pantalla Inicio; si el usuario no cuenta con proyectos o tareas asignadas, el sistema proporciona una guía de usuario como se presenta en la Figura 3.15 para la aplicación web y en la Figura 3.16 para

la aplicación móvil. Caso contrario se indican gráficos estadísticos acerca de proyectos o tareas existentes, según el rol del usuario, como se mostrará más adelante.



Figura 3.15. Guía de usuario - Aplicación web



Figura 3.16. Guía de usuario - Aplicación móvil

3.1.3 MÓDULO EQUIPO

- Rol Gestor

El Gestor selecciona a los delegados que desee formen parte de su equipo de trabajo como se observa en la Figura 3.17 para la aplicación móvil y en la Figura 3.18 para la aplicación web. Una vez seleccionados los usuarios, éstos se agregan a la lista de equipo de trabajo como se observa en la Figura 3.19 y en la Figura 3.20 para las aplicaciones móvil y web respectivamente.

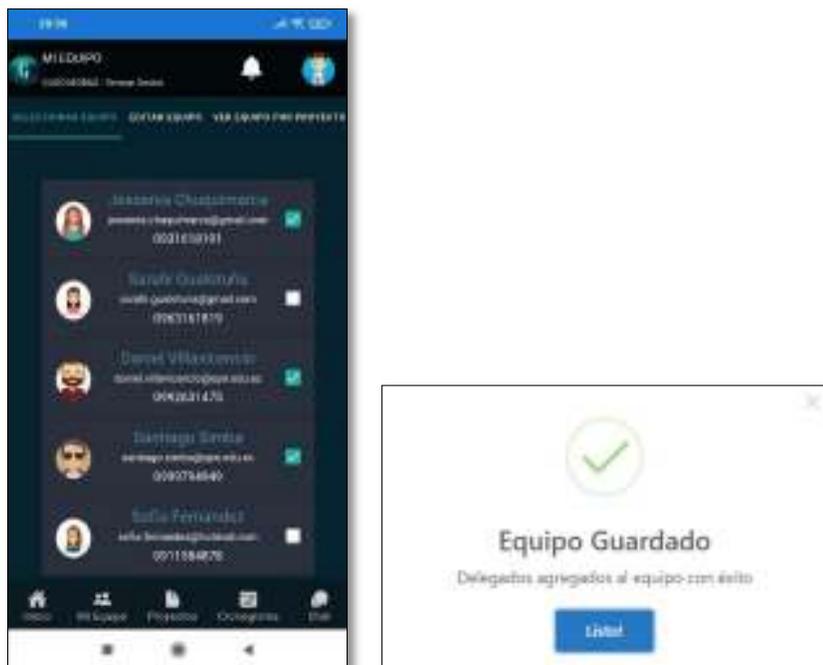


Figura 3.17. Selección de equipo de trabajo - Aplicación móvil

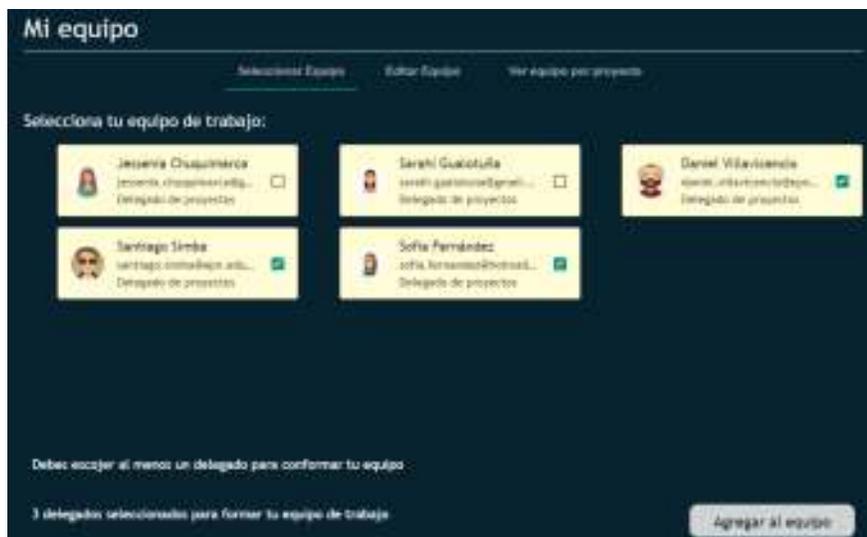


Figura 3.18. Selección de equipo de trabajo - Aplicación web



Figura 3.19. Visualización de equipo de trabajo rol Gestor - Aplicación móvil

Mi equipo

Seleccionar Equipo Editar Equipo Ver equipo por proyecto

Delegados de tu Equipo

	Nombre	Correo	Celular	Cargo	Herramientas
	Daniel Villavicencio	daniel.villavicencio@epn.edu.ec	0992631475	Delegado de proyectos	
	Santiago Simba	santiago.simba@epn.edu.ec	0999794949	Delegado de proyectos	
	Sofia Fernández	sofia.fernandez@hotmail.com	0911584878	Delegado de proyectos	
	Jessenia Chuquimarca	jessenia.chuquimarca@gmail.com	0931618191	Delegado de proyectos	

Figura 3.20. Visualización de equipo de trabajo rol Gestor - Aplicación web

Al remover un Delegado del equipo de trabajo, el sistema indica un mensaje de confirmación para que el Gestor pueda remover el usuario seleccionado como se observa en la Figura 3.21. Este proceso se realiza siempre y cuando el Delegado seleccionado no

tenga tareas a su cargo, caso contrario el sistema presenta un mensaje de error como se observa en la Figura 3.22.

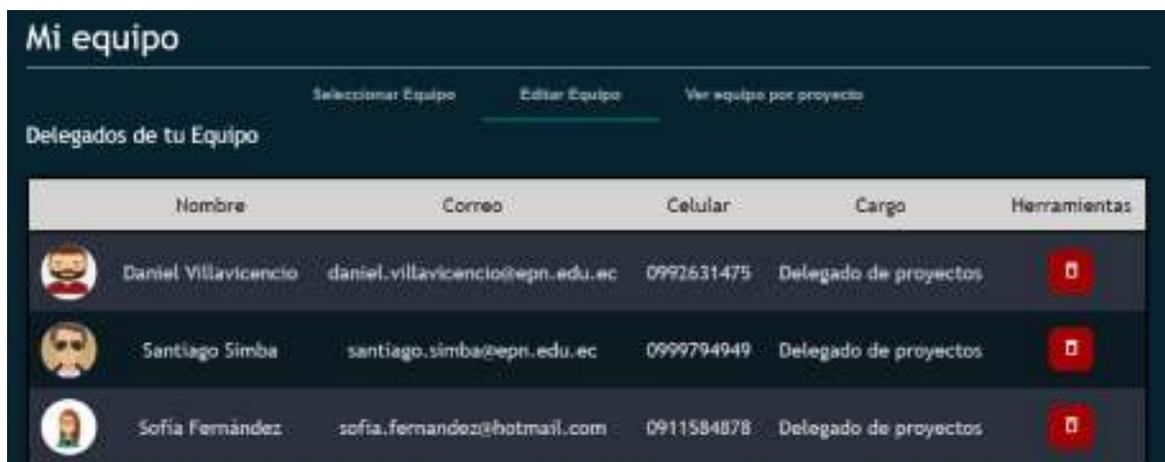


Figura 3.21. Confirmación para remover un Delegado del equipo de trabajo



Figura 3.22. Error Remover Delegado con tareas a su cargo

Finalmente, la lista de equipo se actualiza cómo se observa en la Figura 3.23 para la aplicación web y en la Figura 3.24 para la aplicación móvil; el usuario removido es Jessenia Chuquimarca, por la tanto en la lista actualizada este usuario ya no aparece.



Nombre	Correo	Celular	Cargo	Herramientas
 Daniel Villavicencio	daniel.villavicencio@epn.edu.ec	0992631475	Delegado de proyectos	
 Santiago Simba	santiago.simba@epn.edu.ec	0999794949	Delegado de proyectos	
 Sofia Fernández	sofia.fernandez@hotmail.com	0911584878	Delegado de proyectos	

Figura 3.23. Actualización de lista del equipo de trabajo rol Gestor - Aplicación web



Figura 3.24. Actualización de lista del equipo de trabajo rol Gestor - Aplicación móvil

El Gestor visualiza los equipos de trabajo de cada proyecto que dirija, los cuales se crearon en el módulo Proyectos, como se indica en la Figura 3.25 para la aplicación web y en la Figura 3.26 para la aplicación móvil, además, envía y recibe recursos al equipo de trabajo de un proyecto específico.

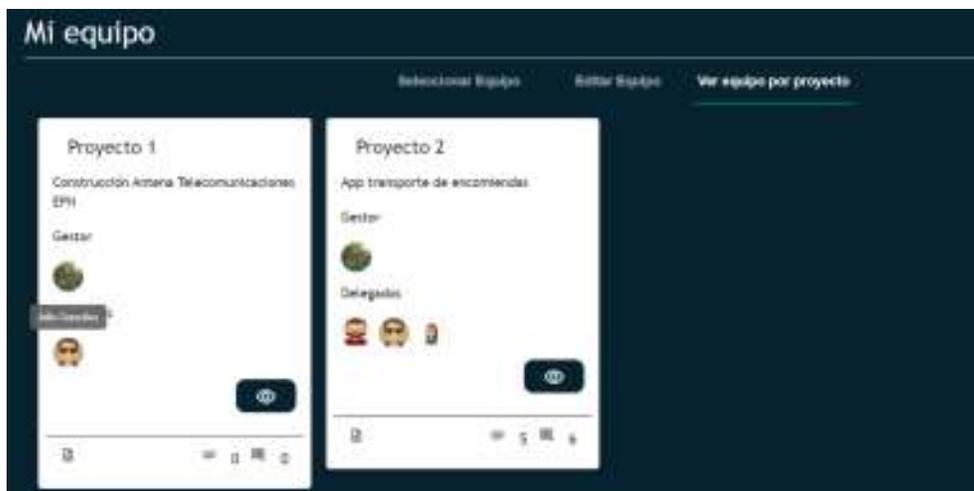


Figura 3.25. Equipos de trabajo por proyecto rol Gestor - Aplicación web



Figura 3.26. Equipos de trabajo por proyecto rol Gestor - Aplicación móvil

- **Rol Delegado**

El Delegado visualiza la lista de Gestores que lo han seleccionado en su equipo de trabajo y compañeros Delegados que son parte de este equipo, como se observa en la Figura 3.27 para la aplicación web y en la Figura 3.28 para la aplicación móvil. Si el Delegado desea ver a qué proyecto corresponde cada Gestor, lo podrá hacer en la pestaña Ver equipo por Proyecto.

El Delegado visualiza los proyectos de los cuales es parte, debido a que ha sido asignado en una tarea, como se indica en la Figura 3.29 para la aplicación web y en la Figura 3.30 para la aplicación móvil, además, envía y recibe recursos al Gestor y equipo de trabajo de un proyecto específico.

Mi equipo

Ver Equipo Ver equipo por proyecto

Tus gestores de proyectos:

NOMBRE	CORREO	CARGO	CELULAR
 Julio González	julio.gonzalez@epn.edu.ec	Gestor de proyectos	0987163161

Tus compañeros de proyectos:

NOMBRE	CORREO	CARGO	CELULAR
 Santiago Simba	santiago.simba@epn.edu.ec	Delegado de proyectos	0999794949
 Sofia Fernández	sofia.fernandez@hotmail.com	Delegado de proyectos	0911584878

Figura 3.27. Visualización de equipo de trabajo rol Delegado – Aplicación web



Figura 3.28. Visualización de equipo de trabajo rol Delegado - Aplicación móvil



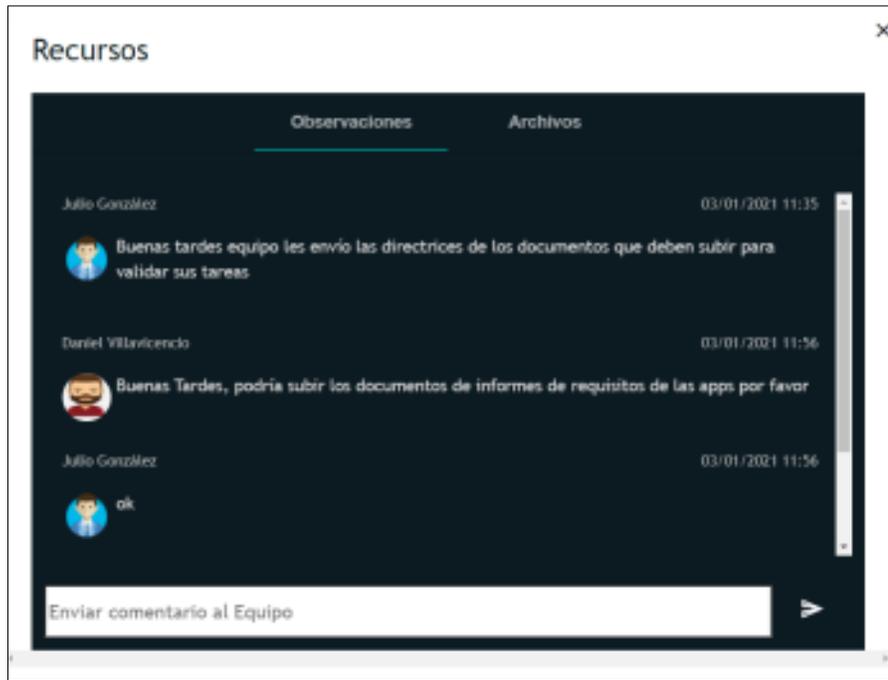
Figura 3.29. Equipos de trabajo por proyecto rol Delegado – Aplicación web



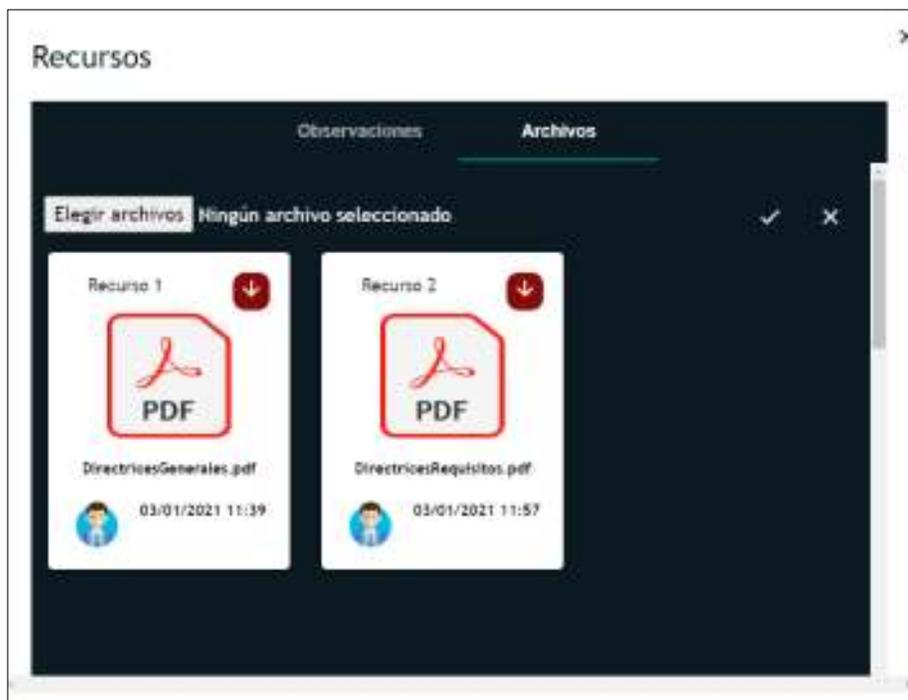
Figura 3.30. Equipos de trabajo por proyecto rol Delegado - Aplicación móvil

El formulario de Recursos lo manejan ambos roles, por lo tanto, el Gestor podrá enviar un comentario a todo el equipo de trabajo del proyecto y visualizar los comentarios recibidos de los integrantes del mismo, así como la carga archivos que desee enviar y visualizar

archivos que ha subido algún integrante del equipo, como se observa en la Figura 3.31 para la aplicación web y en la Figura 3.32 para la aplicación móvil.

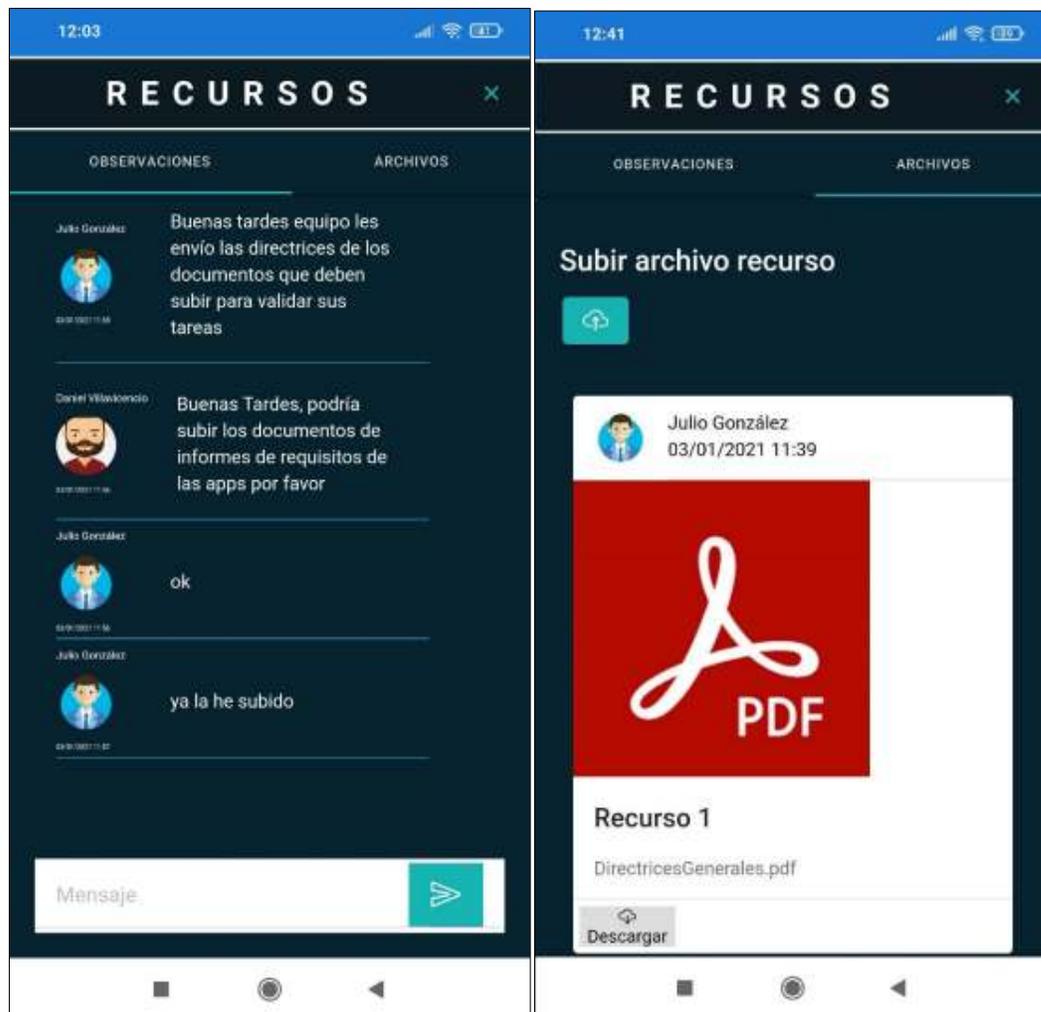


(a)



(b)

Figura 3.31 Formulario de recursos - Aplicación web (a) Observaciones y (b) Archivos



(a)

(b)

Figura 3.32. Formulario de recursos - Aplicación móvil (a) Observaciones y (b) Archivos

3.1.4 MÓDULO PROYECTOS

- **Rol Gestor**

El Gestor visualiza la lista de proyectos que dirige y la creación de un nuevo proyecto mediante el formulario respectivo como se presenta en las partes (a) y (b) de la Figura 3.33. Al igual que el formulario de registro, este formulario valida los datos a medida que son llenados; si existen errores, el sistema indica un mensaje de advertencia, caso contrario indica un mensaje de éxito como se observa en la parte (c) de la Figura 3.33.

Nuevo Proyecto

Nombre *
Construcción Antena EPN

Cliente *
Escuela Politécnica Nacional

Tipo de proyecto *
Proyecto de Construcción

Fecha de inicio *
02/01/2021

Fecha de fin *
01/03/2021

Descripción
Construcción de una base de telecomunicaciones en las instalaciones de la Escuela Politécnica Nacional.

Tiene los campos (*) obligatorios

Guardar

(a)

22:39

NUEVO PROYECTO

Nombre *
App transporte de encomiendas

Cliente *
GAD Pichincha

Tipo *
Proyecto de Desarrollo de Software

Fecha de inicio *
02-Jan-2021

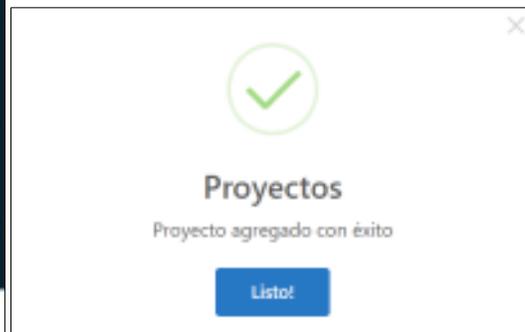
Fecha de fin *
04-May-2021

Descripción
Aplicación para el envío de encomiendas para el GAD de la provincia de Pichincha

Tiene los campos (*) obligatorios

Guardar

(b)



(c)

Figura 3.33. Formulario para crear un nuevo proyecto (a) Aplicación web, (b) Aplicación móvil y (c) Mensaje de éxito

Si no existen errores la lista de proyectos se actualiza como se observa en la Figura 3.34.



The screenshot shows a web application interface titled "Proyectos". It features a table with the following columns: Nombre, Cliente, Estado, Fecha de inicio, Fecha de Fin, Progreso, and Herramientas. There are three rows of project data, each with a progress indicator and a set of three action icons (edit, delete, and another function).

Nombre	Cliente	Estado	Fecha de inicio	Fecha de Fin	Progreso	Herramientas
Construcción Antena Telecomunicaciones EPH	Escuela Politécnica Nacional	Por Realizar	02/01/2021	01/03/2021	0%	[Edit] [Delete] [Action]
App transporte de encomiendas	GAD Pichincha	Por Verificar	02/01/2021	03/01/2021	27%	[Edit] [Delete] [Action]
Evaluación técnica del sistema de información de la empresa MIP	MIP	Por Realizar	01/02/2021	23/04/2021	0%	[Edit] [Delete] [Action]

A "Nuevo Proyecto" button is located at the bottom right of the table area.

(a)



(b)

Figura 3.34. Lista de proyectos actualizada Rol Gestor – Actualización 1 (a) Aplicación web y (b) Aplicación móvil

Al eliminar un proyecto específico, el sistema realiza un mensaje de confirmación para realizar esta acción, como se observa en la Figura 3.35.

Se puede editar un proyecto, a través del formulario que se presenta en la Figura 3.36 para la aplicación web y en la Figura 3.37 para la aplicación móvil; este formulario también realiza las respectivas validaciones y en caso de existir error muestra mensajes de advertencia caso contrario de indica un mensaje de éxito.

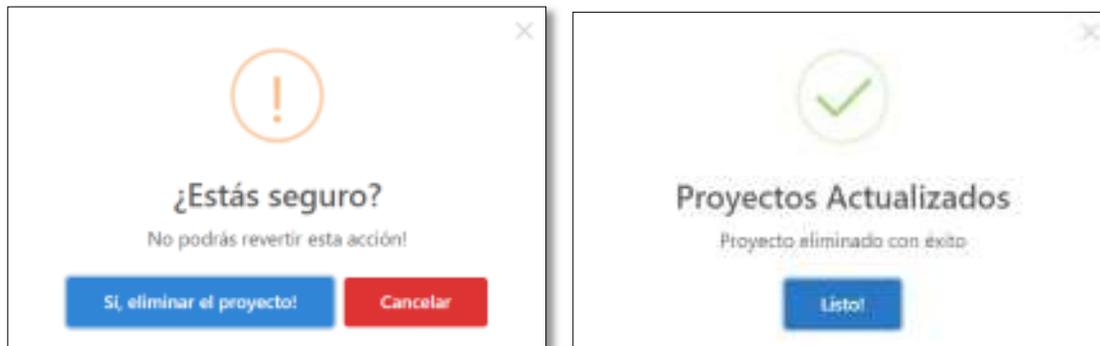


Figura 3.35. Mensaje de confirmación para eliminar un proyecto

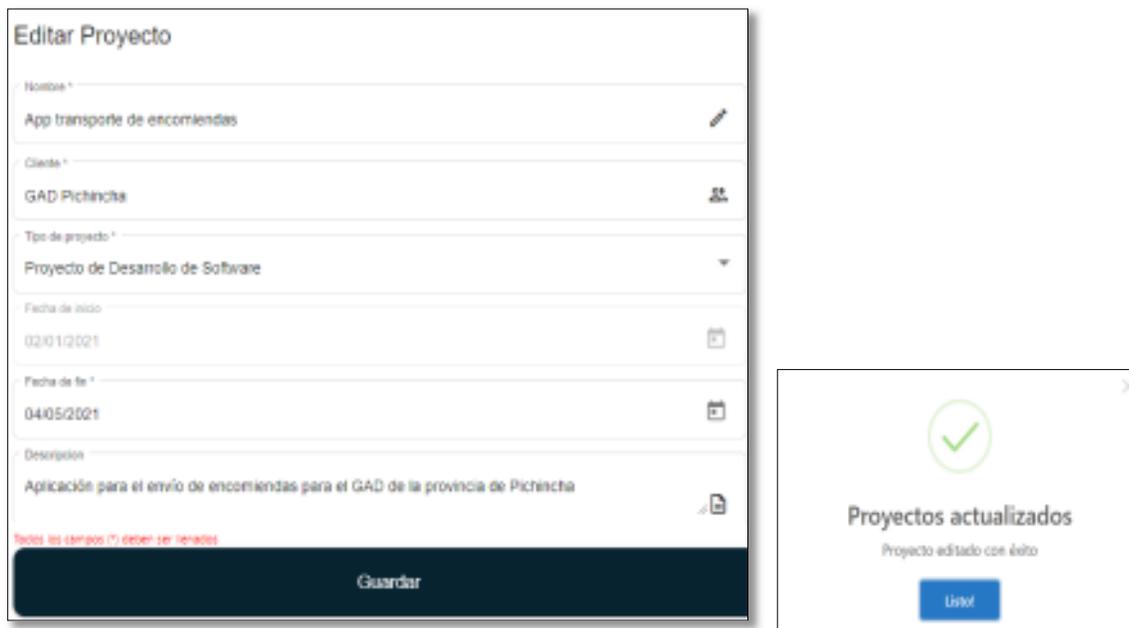


Figura 3.36. Formulario para editar un Proyecto - Aplicación web

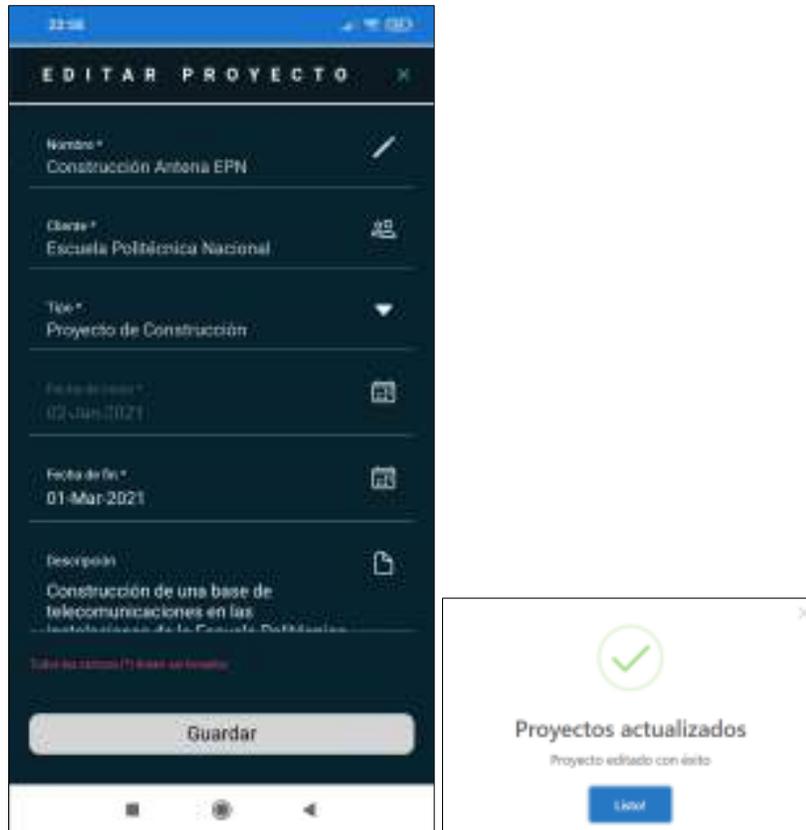


Figura 3.37. Formulario para editar un Proyecto - Aplicación móvil

Una vez realizadas las acciones correspondientes, la lista de proyectos se actualiza, tal como se presenta en la Figura 3.38 para la aplicación web y en la Figura 3.39 para la aplicación móvil.

Proyectos

Nombre	Cliente	Estado	Fecha de Inicio	Fecha de Fin	Progreso	Herramientas
Construcción Antena Telecomunicaciones EPN	Escuela Politécnica Nacional	Por Realizar	02/01/2021	01/03/2021	0%	[Editar] [Eliminar] [Imprimir]
App transporte de encomiendas	GAO Politecnica	Por Verificar	02/01/2021	03/01/2021	27%	[Editar] [Eliminar] [Imprimir]

Nuevo Proyecto

Figura 3.38. Lista de proyectos actualizada – Actualización 2 - Aplicación web

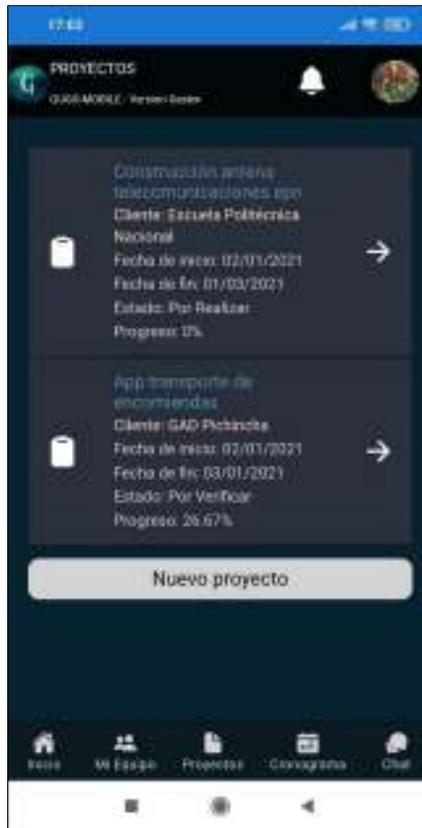


Figura 3.39. Lista de proyectos actualizada – Actualización 3 - Aplicación móvil

- **Rol Delegado**

El Delegado visualiza la lista de tareas a las que ha sido asignado mediante el Módulo Actividades y cuáles debe cumplir como se indica en la Figura 3.40. Permite cargar archivos desde el dispositivo móvil o computador e indica el porcentaje de avance de una tarea, tal como se presenta en la Figura 3.41 para la aplicación web. La parte (a) corresponde a seleccionar un archivo desde el computador, la parte (b) corresponde al archivo seleccionado y visualización del resto de archivos que ya han sido subidos y la parte (c) corresponde a las observaciones que el Gestor emite hacia el Delegado encargado de la tarea.

En la Figura 3.42 se presenta el mismo proceso del formulario de evidencia, pero para la aplicación móvil. Adicionalmente, cuando el Delegado cuenta con tareas a su cargo, la pantalla Inicio se actualiza e indica gráficos estadísticos del estado de las tareas, los cuales son similares a los de rol Gestor.

Proyectos

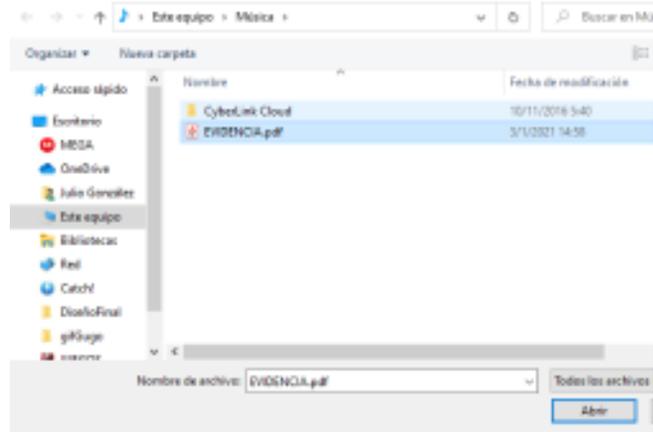
Tareas	Fecha Inicio	Fecha Fin	Gestor	Actividad	Proyecto	% Avance
Desarrollo de los popups/modales de CRUD	31/01/2021	03/02/2021		Desarrollo FrontEnd	App transporte de encomendas	0%
Desarrollo de las paginas principales	27/01/2021	30/01/2021		Desarrollo FrontEnd	App transporte de encomendas	0%
Informe de la reunion establecida	05/01/2021	06/01/2021		Requisitos previos de la aplicación	App transporte de encomendas	0%
Integración con el Backend	10/02/2021	13/02/2021		Desarrollo FrontEnd	App transporte de encomendas	0%

(a)



(b)

Figura 3.40. Visualización de tareas asignadas (a) Aplicación web y (b) Aplicación móvil



(a)

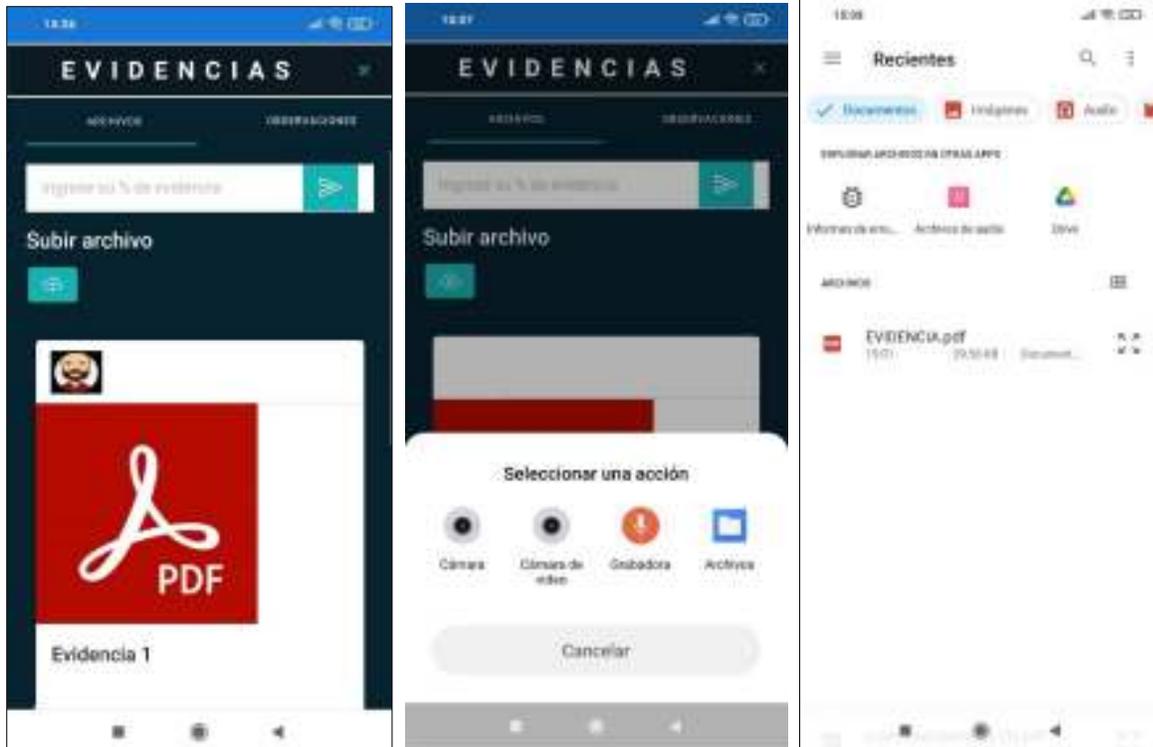


(b)



(c)

Figura 3.41. Formulario evidencia rol Delegado – Aplicación web (a) Selección de un archivo (b) Carga de archivo y (c) Observaciones



(a)

(b)

(c)



(d)

(e)

Figura 3.42. Formulario evidencia rol Delegado – Aplicación móvil

3.1.5 MÓDULO ACTIVIDADES

- **Rol Gestor**

El Gestor accede a este módulo mediante la opción Ver actividades del módulo Proyectos con rol Gestor, aquí visualiza las actividades correspondientes a un proyecto específico y las tareas correspondientes a cada actividad.

Se crea una actividad mediante el formulario que se observa en la Figura 3.43 para la aplicación web y la Figura 3.44 para la aplicación móvil. El sistema realiza la respectiva validación de datos y crea la actividad; si la validación es correcta, se actualiza la lista de actividades como se observa en la Figura 3.45 para la aplicación web y en la Figura 3.46 para la aplicación móvil.



El formulario 'Nueva Actividad' para la aplicación web contiene los siguientes campos:

- Nombre ***: Campo de texto vacío.
- Requisitos de la aplicación**: Campo de texto con un ícono de lápiz a la derecha.
- Fecha de inicio ***: Campo de fecha con el valor '04/01/2021' y un ícono de calendario a la derecha.
- Fecha de fin ***: Campo de fecha con el valor '07/01/2021' y un ícono de calendario a la derecha.
- Descripción**: Campo de texto con el valor 'Se necesita conocer los requisitos de la aplicación para el adecuado desarrollo' y un ícono de documento a la derecha.

Debajo de los campos, hay un mensaje de error en rojo: 'Todos los campos (*) deben ser llenados'. En la parte inferior del formulario hay un botón azul con el texto 'Guardar'.

Figura 3.43. Formulario de nueva actividad - Aplicación web

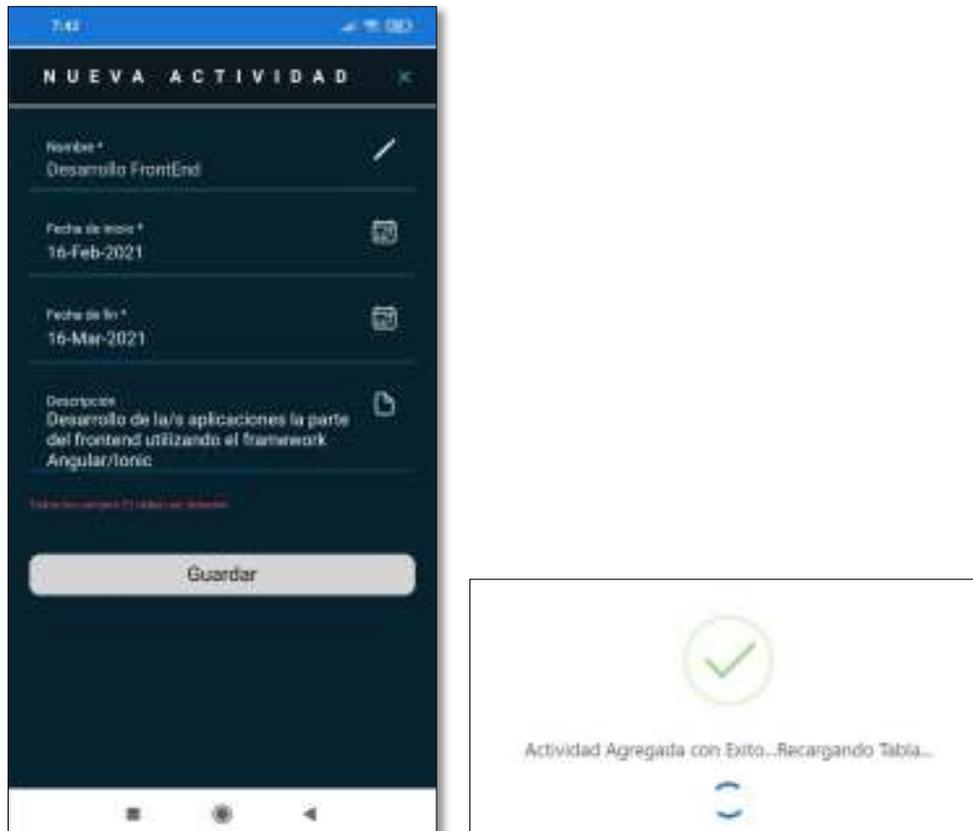


Figura 3.44. Formulario de nueva actividad - Aplicación móvil



Figura 3.45. Actualización de la lista de actividades – Aplicación Web

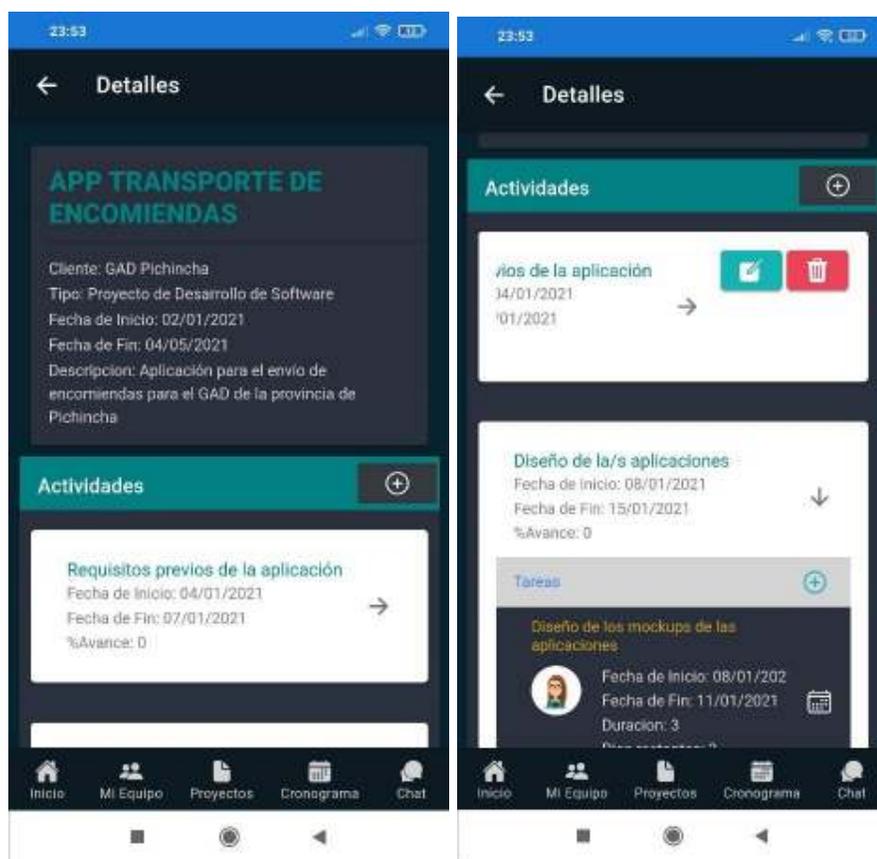


Figura 3.46. Actualización de lista de actividades – Aplicación móvil

La edición de una actividad por parte del Gestor se realiza mediante el formulario que se observa en la Figura 3.47; el sistema realiza la validación de los datos, si esta validación es correcta, emitirá un mensaje de confirmación. La eliminación de una actividad por parte del Gestor se efectúa mediante la confirmación que realiza al sistema, como se observa en la Figura 3.48; la actividad eliminada es Desarrollo *Frontend*.

Luego de las acciones realizada sobre las actividades, se actualiza su lista como se indica en la Figura 3.49 para la aplicación web y en la Figura 3.50 para la aplicación móvil.

Editar Actividad

Nombre *

Requisitos de la aplicación

Fecha de inicio *

04/01/2021

Fecha de fin *

07/01/2021

Descripción

Se necesita conocer los requisitos de la aplicación para el adecuado desarrollo

Todos los campos (*) deben ser requeridos

Guardar

(a)

12:42

EDITAR ACTIVIDAD

Nombre *

Diseño de las aplicaciones

Fecha de inicio *

08-Jan-2021

Fecha de fin *

15-Jan-2021

Descripción

De acuerdo a los requisitos realizar el diseño de las aplicaciones

Todos los campos (*) deben ser requeridos

Guardar

Actividades Actualizadas

Actividad editada con éxito

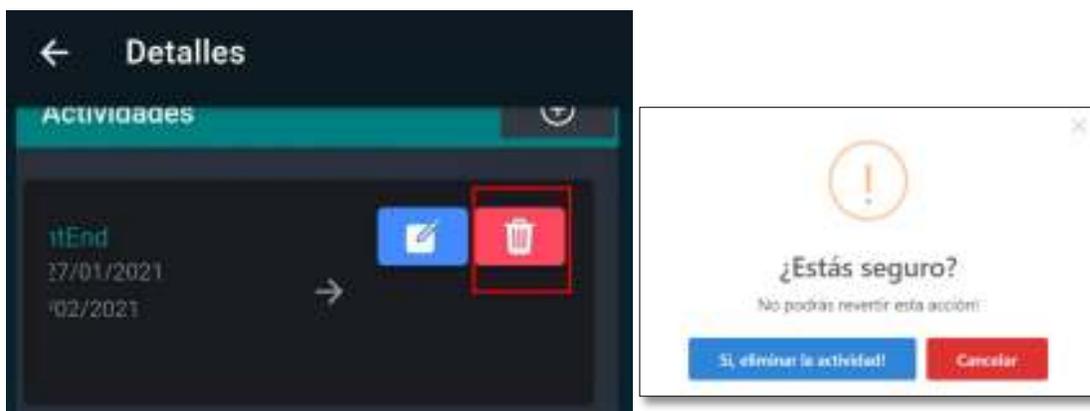
Listar

(b)

Figura 3.47. Formulario de editar una actividad (a) Aplicación web y (b) Aplicación móvil



(a)



(b)

Figura 3.48. Eliminación de una actividad (a) Aplicación web y (b) Aplicación móvil



Figura 3.49. Actualización de la lista de Actividades - Aplicación web

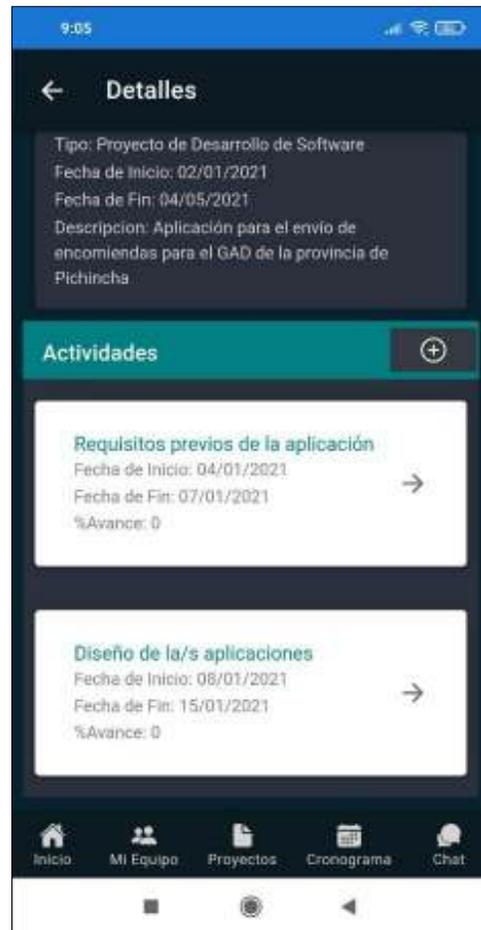


Figura 3.50. Actualización de la lista de Actividades - Aplicación móvil

Las acciones realizadas sobre cada tarea son crear, editar y eliminar, los mensajes de confirmación y éxito son los mismos que los anteriores por lo tanto se hará solo una breve descripción de los formularios correspondientes.

El Gestor crea una tarea mediante el formulario que se presenta en la Figura 3.51; el Gestor edita una tarea mediante el formulario que se observa en la Figura 3.52; el Gestor elimina una tarea tal como se observa en la Figura 3.53.

Nueva Tarea

Nombre *
Diseño de los mockups de las aplicaciones segunda parte

Fecha de inicio *
12/01/2021

Fecha de fin *
14/01/2021

Delegado *
Sofía Fernández

Descripción
Diseño de las pantallas secundarias

Todos los campos (*) deben ser llenados

Guardar

(a)

NUEVA TAREA

Nombre *
Realización de la cotización del proyecto

Fecha de inicio *
06-Jan-2021

Fecha de fin *
07-Jan-2021

Delegado * Santiago Simón

Descripción
Una vez realizado el informe de los requisitos se necesita la cotización del

Todos los campos (*) deben ser llenados

Guardar

Tarea Agregada con Exito...Recargando Tabla...

(b)

Figura 3.51. Formulario de nueva tarea (a) Aplicación web y (b) Aplicación móvil

Editar Tarea

Nombre *
Informe de la reunion establecida

Fecha de inicio *
05/01/2021

Fecha de fin *
06/01/2021

Delegado *
Daniel Milevicencio

Descripcion

Todos los campos (*) deben ser llenados

Guardar

(a)

10:38

EDITAR TAREA

Nombre *
Reunión con el cliente

Fecha de inicio *
04-Jan-2021

Fecha de fin *
05-Jan-2021

Delegado * Santiago Simba

Descripcion
Reunión con el cliente para registrar los requisitos de la aplicación

Todos los campos (*) deben ser llenados

Guardar

Tareas actualizadas
Tarea editada con éxito

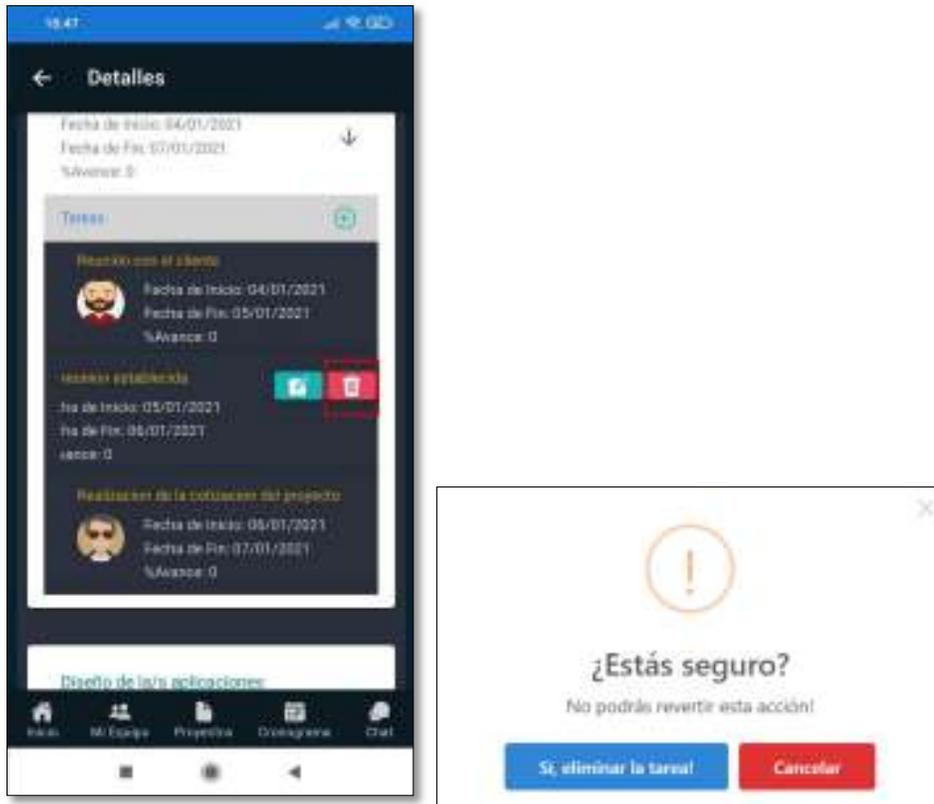
Listar

(b)

Figura 3.52. Formulario de editar tarea (a) Aplicación web y (b) Aplicación móvil



(a)



(b)

Figura 3.53. Eliminación de una tarea (a) Aplicación web y (b) Aplicación móvil

Luego de las acciones generadas sobre la lista de tareas, se actualiza ésta como se indica en la Figura 3.54.



(a)



(b)

Figura 3.54. Actualización de la lista de tareas (a) Aplicación web y (b) Aplicación móvil

El Gestor visualiza la evidencia proporcionada mediante el respectivo formulario; este formulario se actualiza a medida que el respectivo Delegado de esa tarea proporcione la información necesaria, es decir realice la carga de archivos y establezca el porcentaje de

avance de la tarea, esto lo hace el Delegado a través del formulario indicado en la Figura 3.41 para la aplicación web y en la Figura 3.42 para la aplicación móvil.

Los documentos se descargan directamente, mientras que las imágenes se visualizan en otra pestaña de donde son descargadas. Un ejemplo se presenta en la Figura 3.55 para la aplicación web y en la Figura 3.56 para la aplicación móvil.

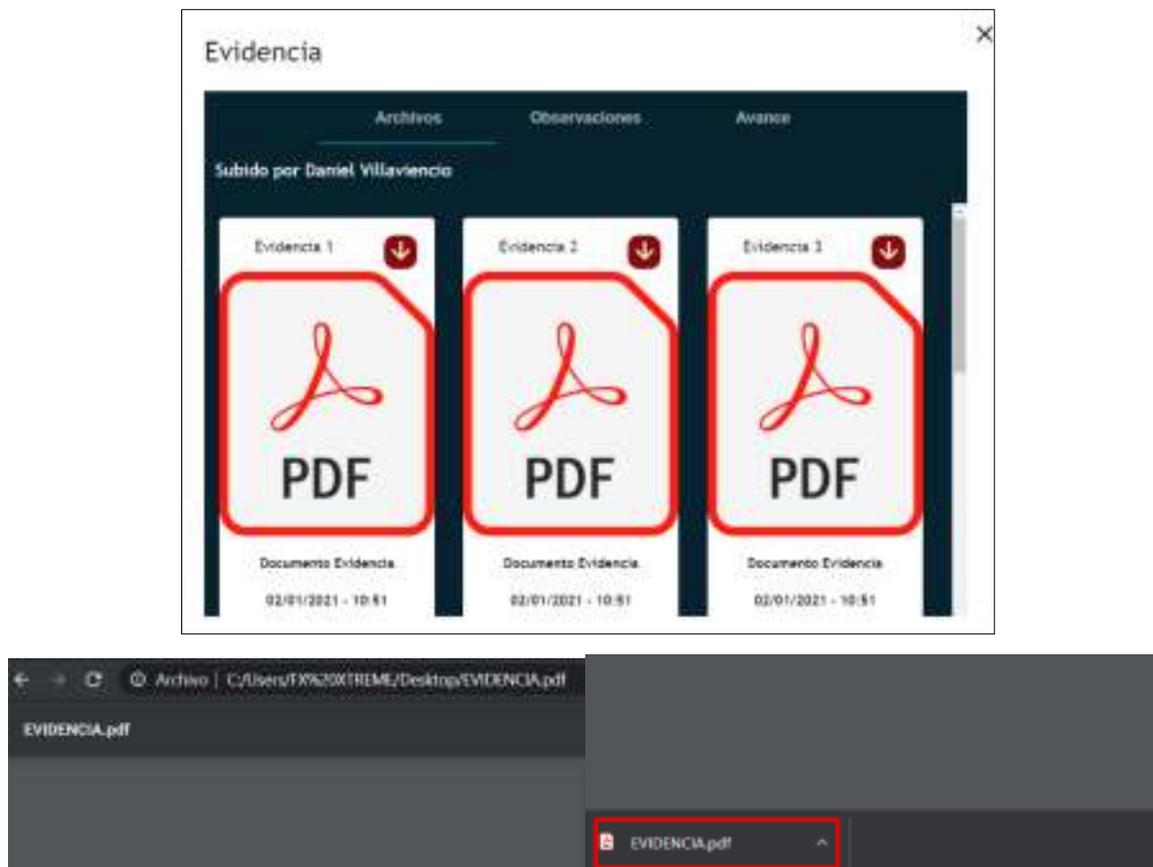


Figura 3.55. Descarga de un archivo – Aplicación web

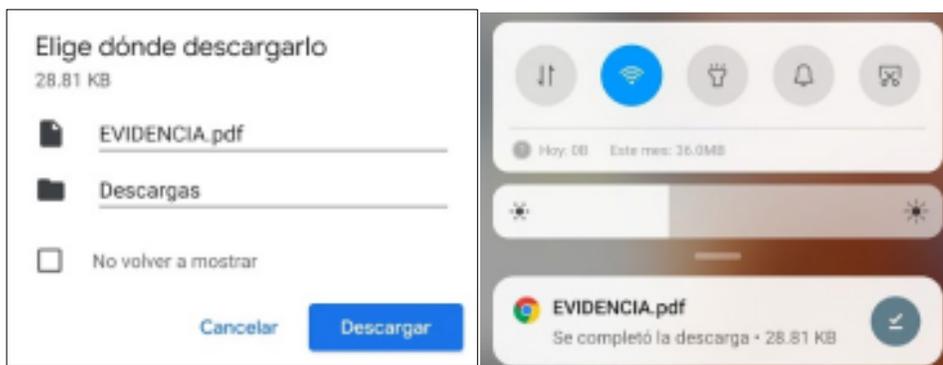
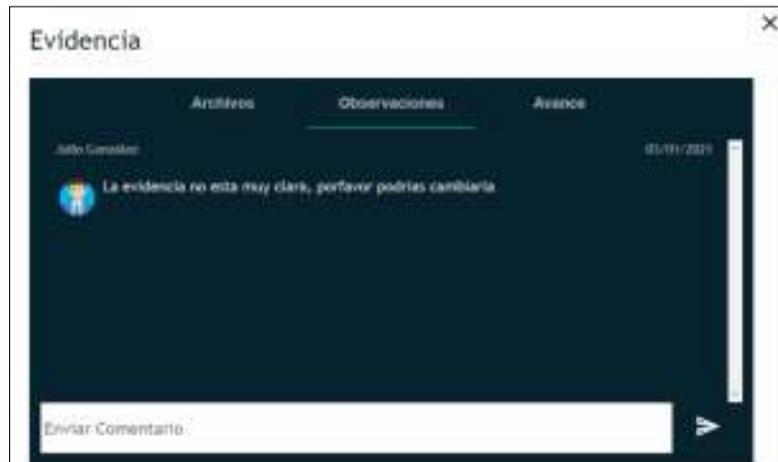


Figura 3.56. Descarga de un archivo – Aplicación móvil

El Gestor emite observaciones hacia el Delegado encargado de la tarea, tal como se indica en la Figura 3.57. Además, confirma el porcentaje de avance tal como se observa en la Figura 3.58; la corrección del porcentaje, en caso de ser necesario, se presenta en la Figura 3.59.

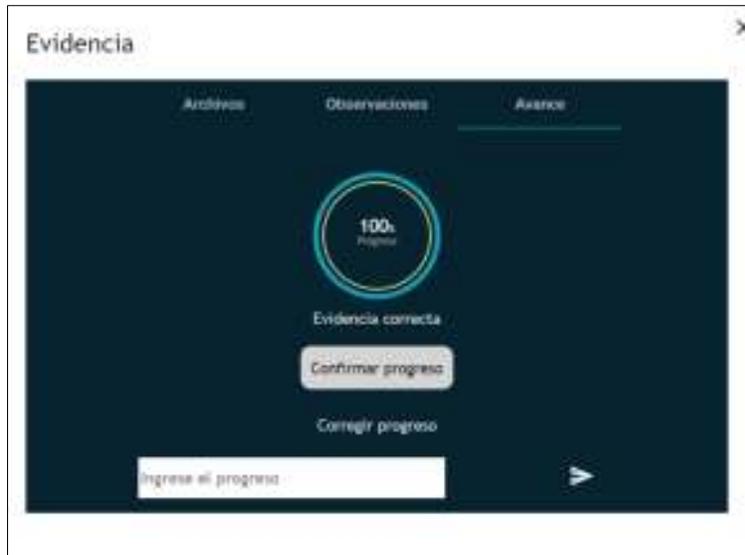


(a)

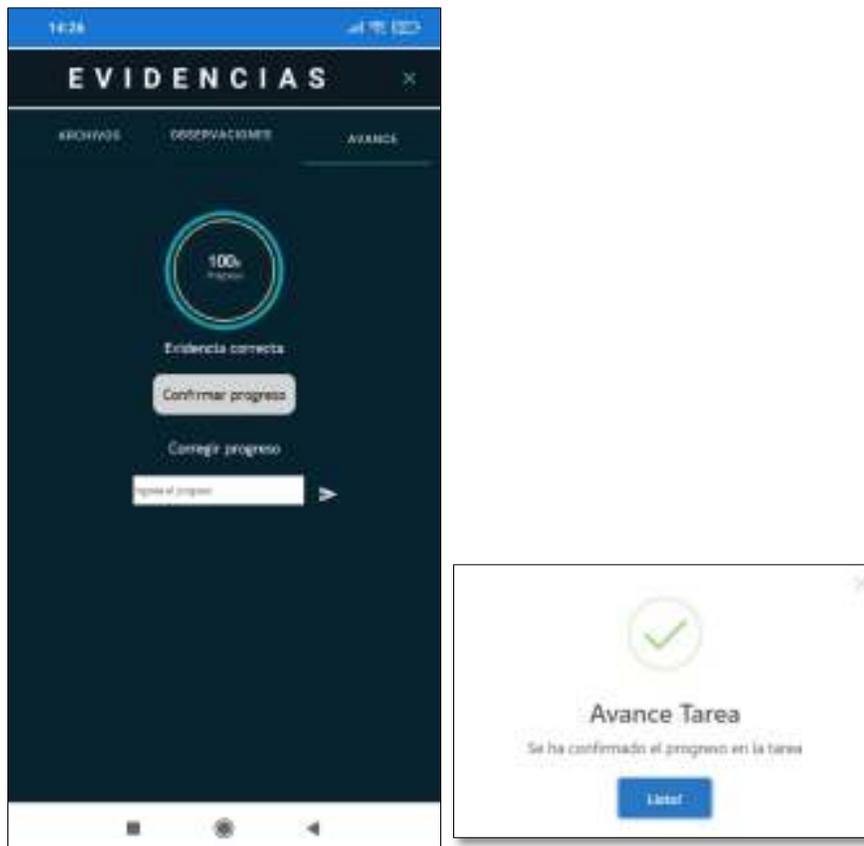


(b)

Figura 3.57. Observaciones del formulario evidencia rol Gestor (a) Aplicación web y (b) Aplicación móvil

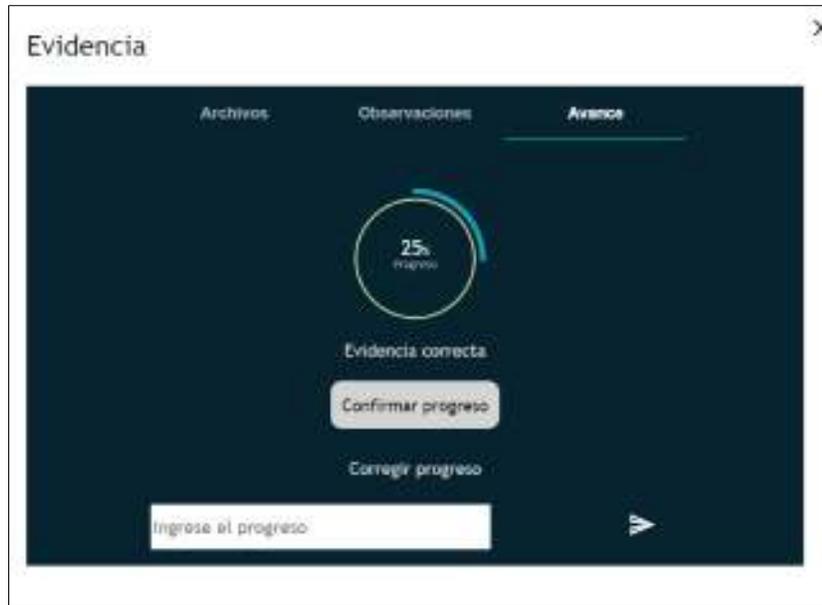


(a)

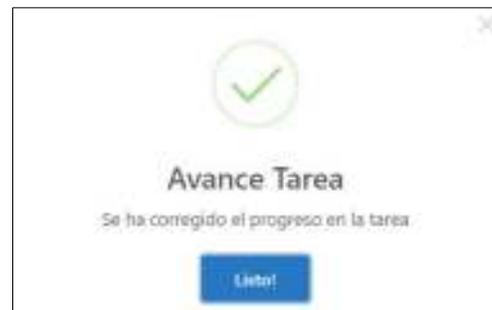
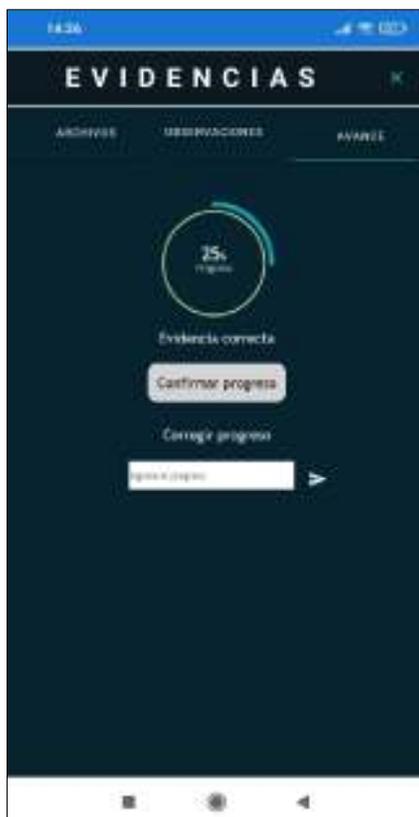


(b)

Figura 3.58. Confirmación de porcentaje de avance (a) Aplicación web y (b) Aplicación móvil



(a)



(b)

Figura 3.59. Corrección del porcentaje de avance (a) Aplicación web y (b) Aplicación móvil

Una vez confirmado el porcentaje de avance de la tarea, el sistema automáticamente actualizará este porcentaje. De esta manera el estado del proyecto cambia y en la pantalla Inicio aparecen los gráficos estadísticos correspondientes, tal como se presenta en la Figura 3.60.



(a)



(b)

Figura 3.60. Gráficos estadístico rol Gestor (a) Aplicación web y (b) Aplicación móvil

3.1.6 MÓDULO CRONOGRAMA

- Rol Gestor

El Gestor visualiza la lista de proyectos ordenados según su fecha de fin como se indica en la Figura 3.61 para la aplicación web y en la Figura 3.62 para aplicación móvil.



The screenshot shows a web application interface with a dark blue header titled "Cronograma". Below the header is a table with the following data:

Proyecto	Fecha de inicio	Fecha de fin	Duracion	Dias restantes	Dias de atraso	Herramientas
Construcción Antena Telecomunicaciones EPN	02/01/2021	01/03/2021	58	57	0	[Icon]
App transporte de encomiendas	02/01/2021	04/05/2021	122	121	0	[Icon]

Figura 3.61. Visualización del módulo cronograma rol Gestor - Aplicación web



Figura 3.62. Visualización del módulo cronograma rol Gestor - Aplicación móvil

Cada proyecto permite visualizar la lista de sus tareas, donde cada tarea indica el Delegado encargado y su porcentaje de avance, tal como se observa en la Figura 3.63.



Figura 3.63. Lista de tareas pertenecientes al módulo Cronograma rol Gestor

- **Rol Delegado**

En este módulo el Delegado solo visualiza la lista de tareas ordenadas por su fecha de fin y no realiza ningún acción; por lo tanto se presenta la visualización del módulo cronograma para la aplicación web, en la Figura 3.64 y en Figura 3.65 para la aplicación móvil.

Tarea	Fecha Inicio	Fecha Fin	Gestor	Duración	Días Restantes	Días de Atraso	Actividad	Proyecto
Informe de la reunión establecida	05/01/2021	06/01/2021		1	1	0	Requisitos previos de la aplicación	App transporte de encomiendas
Desarrollo de las paginas principales	27/01/2021	30/01/2021		3	3	0	Desarrollo FrontEnd	App transporte de encomiendas
Desarrollo de los popups/modales de CRUD	31/01/2021	03/02/2021		3	3	0	Desarrollo FrontEnd	App transporte de encomiendas
Integración con el Backend	10/02/2021	13/02/2021		3	3	0	Desarrollo FrontEnd	App transporte de encomiendas

Figura 3.64. Visualización del módulo Cronograma rol Delegado - Aplicación web

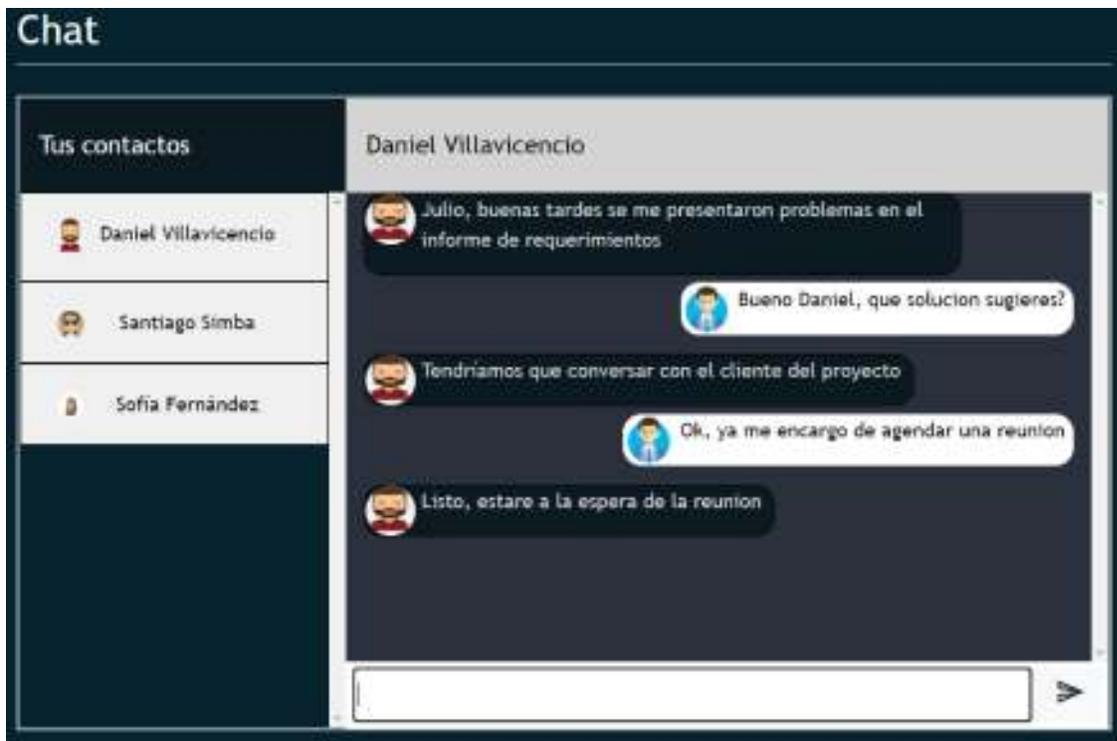


Figura 3.65 Visualización del módulo Cronograma rol Delegado - Aplicación móvil

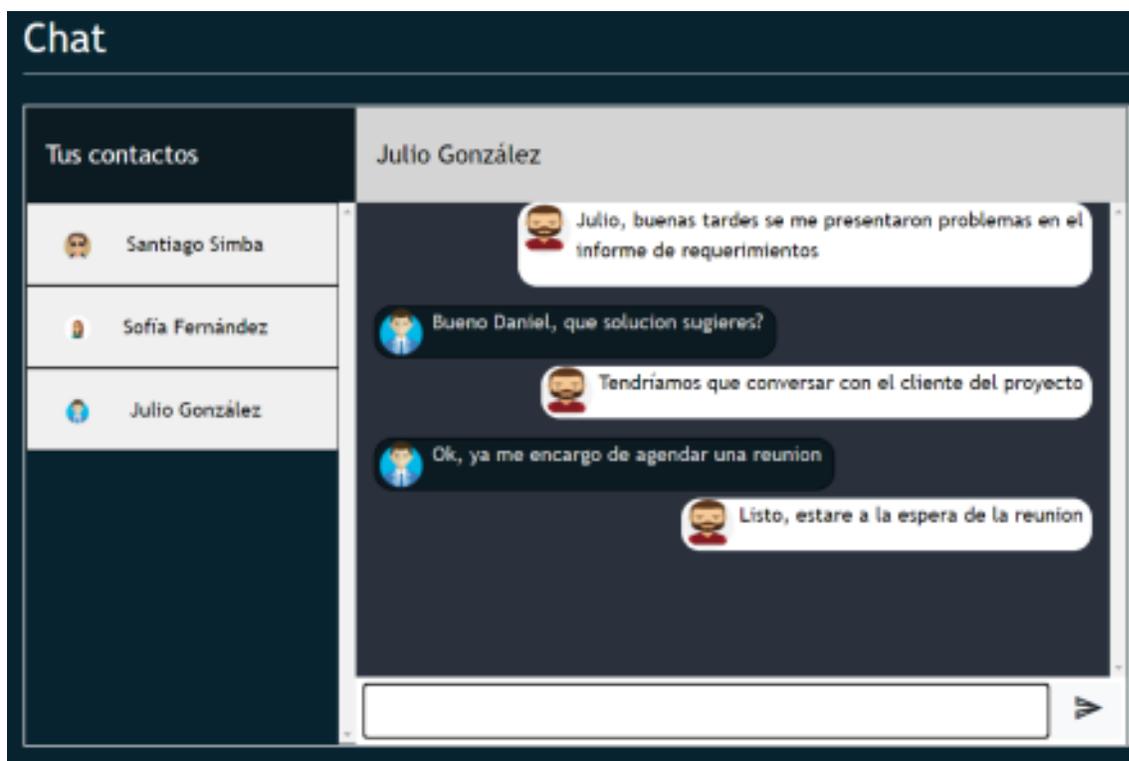
3.1.7 MÓDULO CHAT

El usuario con rol Gestor o rol Delegado envía mensajes a un usuario específico que pertenezca a su equipo de trabajo, el cual es conformado en el módulo Equipo.

El Gestor envía un mensaje hacia un Delegado y el Delegado emite su respuesta, como se observa en la Figura 3.66 para la aplicación web y en la Figura 3.67 para la aplicación móvil.

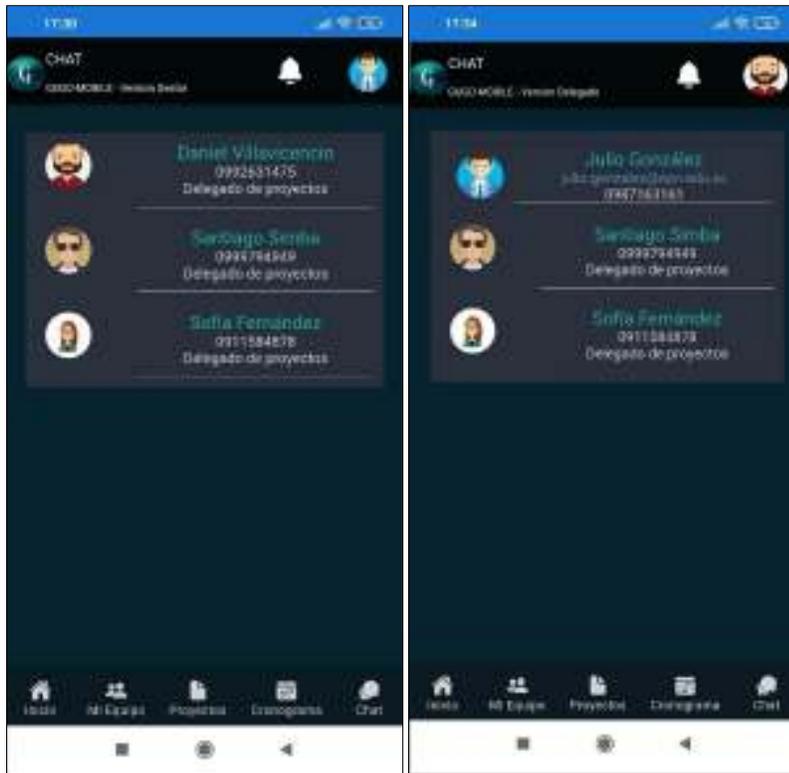


(a)



(b)

Figura 3.66. Conversación entre un (a) Gestor y (b) Delegado - Aplicación web



(a)

(b)



(c)

Figura 3.67. Conversación entre un (a) Gestor, (b) Delegado y (c) Información intercambiada - Aplicación móvil

3.1.8 MÓDULO PERFIL USUARIO

Cuando el usuario se registra, crea un perfil dentro del sistema, pudiendo agregar una imagen de perfil mediante el acceso a la cámara del dispositivo. La aplicación móvil permite subir imágenes desde la galería (Navegar) o desde la Cámara; para esta prueba se ha seleccionado la opción Cámara, todo esto se muestra en la Figura 3.68.

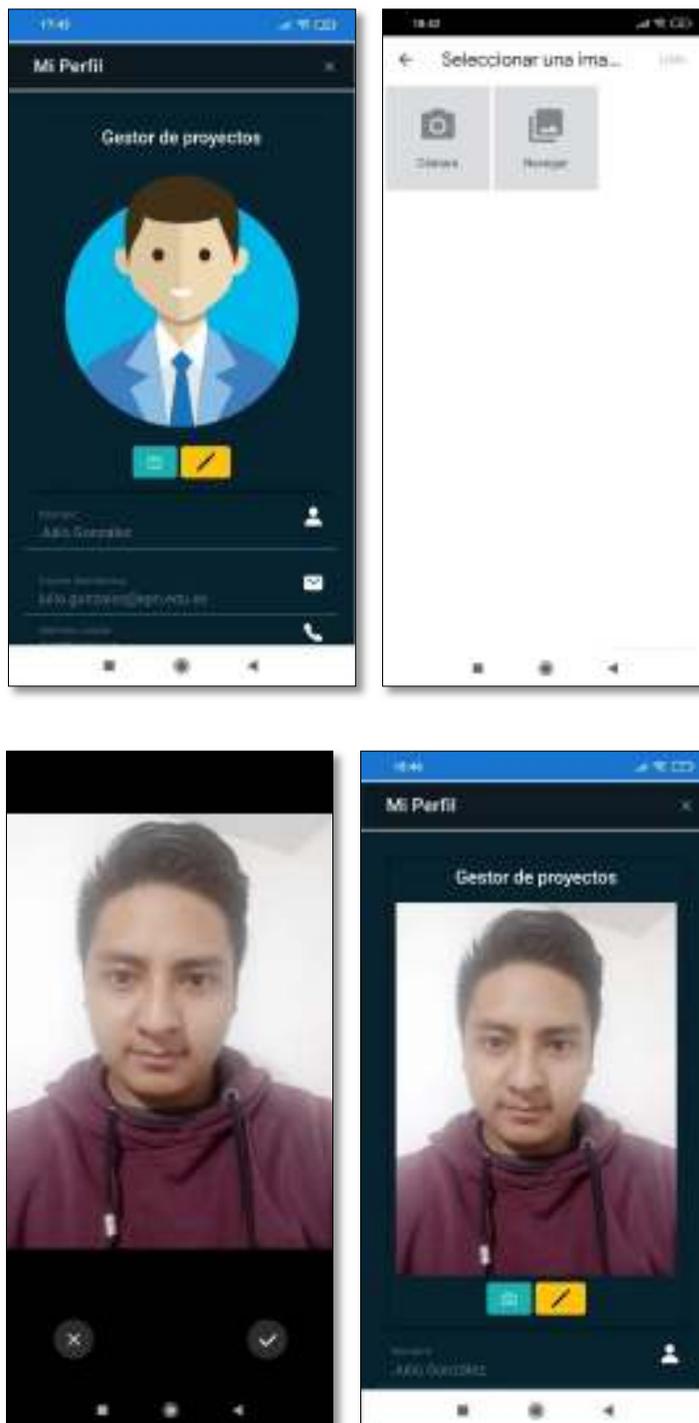
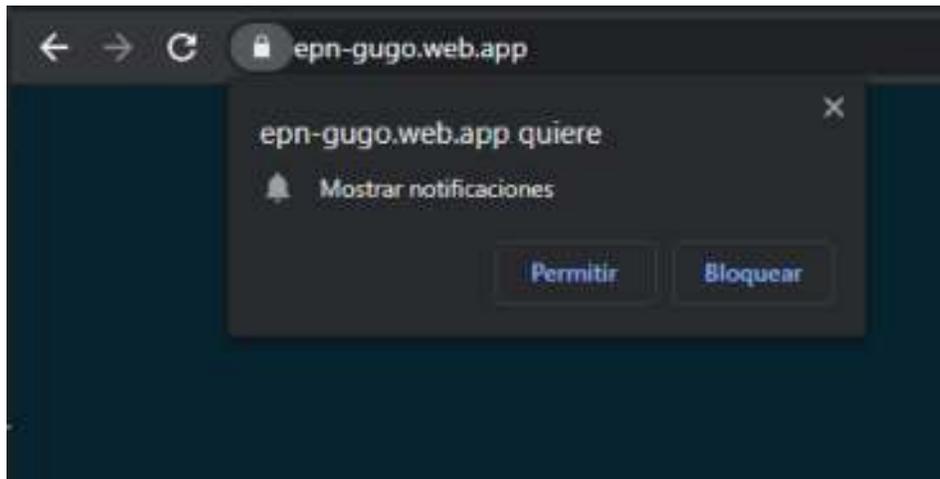


Figura 3.68. Agregar una foto al perfil de usuario - Aplicación móvil

3.1.9 MÓDULO NOTIFICACIONES

El usuario otorga permisos en ambas aplicaciones para que éste pueda recibir notificaciones; esto se indica en la parte (a) de la Figura 3.69 para la aplicación web y en la parte (b) de la Figura 3.69 para la aplicación móvil.



(a)



(b)

Figura 3.69. Permisos para recibir notificaciones (a) Aplicación web y (b) Aplicación móvil

El sistema genera notificaciones de manera automática acerca de eventos relevantes que realice el Gestor o Delegado. A continuación, se indican algunos de los eventos para cada rol.

- **Rol Gestor**

En la Tabla 3.1 se presentan los eventos que generan notificaciones hacia el Gestor.

Tabla 3.1. Eventos generadores de notificaciones para el rol Gestor

No.	Razón
1.	Cuando un Delegado en una tarea específica ha proporcionado un nuevo porcentaje de avance con su respectiva evidencia
2.	Cuando un usuario perteneciente a su equipo de trabajo ha emitido comentarios o ha cargado archivos dentro de los recursos de un proyecto

En la Figura 3.70 se tiene la visualización del módulo Notificaciones para el rol Gestor mientras que en la Figura 3.71 se presenta la correspondiente visualización para la aplicación web.

Adicionalmente, en la Figura 3.72 y Figura 3.73 se presentan ejemplos de cómo el Gestor recibe notificaciones, tanto para la aplicación web como para la aplicación móvil respectivamente.

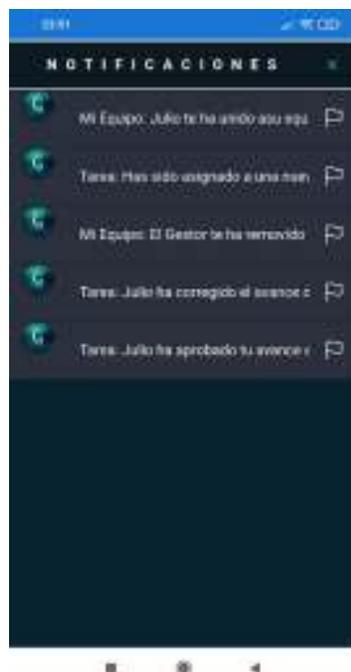


Figura 3.70. Visualización del módulo Notificaciones – Aplicación móvil

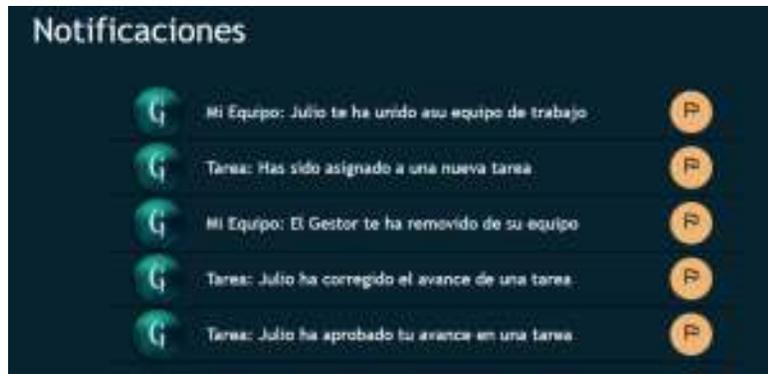


Figura 3.71. Visualización del módulo Notificaciones – Aplicación web



(a)

Nombre	Correo	Celular	Cargo	Herramientas
Daniel Villavicencio	daniel.villavicencio@epn.edu.ec	0991631475	Delegado de proyectos	[Icon]
Santiago Simba	santiago.simba@epn.edu.ec	0999794949	Delegado de proyectos	[Icon]
Sofía Fernández	sofia.fernandez@hotmail.com	0911584878	Delegado d	[Icon]

(b)

Figura 3.72. Ejemplos de notificaciones que recibe el Gestor - Aplicación web



Figura 3.73. Notificaciones que recibe el Gestor - Aplicación móvil

- **Rol Delegado**

En la Tabla 3.2 se presentan los eventos que generan notificaciones hacia el Delegado.

Tabla 3.2. Eventos generadores de notificaciones para el rol Delegado

No.	Razón
1.	Cuando un Gestor lo ha seleccionado para formar parte de su equipo de trabajo.
2.	Cuando un Gestor lo ha eliminado de su equipo de trabajo.
3.	Cuando ha sido asignado a cargo de una tarea.
4.	Cuando el Gestor ha corregido el porcentaje de avance de una tarea.
5.	Cuando el Gestor ha aprobado el porcentaje de avance de una tarea.
6.	Cuando un usuario perteneciente a su equipo de trabajo ha emitido comentarios o ha cargado archivos dentro de los recursos de un proyecto.

La visualización del módulo Notificaciones para el rol Delegado es el mismo que para el rol Gestor, difieren en el contenido de las notificaciones que reciben, por lo tanto solo se

presentan ejemplos de las notificaciones que recibe el Delegado; dichos ejemplos se presentan en la Figura 3.74 para la aplicación web y en la Figura 3.75 para la aplicación móvil.



(a)



(b)

Figura 3.74. Ejemplos de notificaciones que recibe el Delegado – Aplicación web

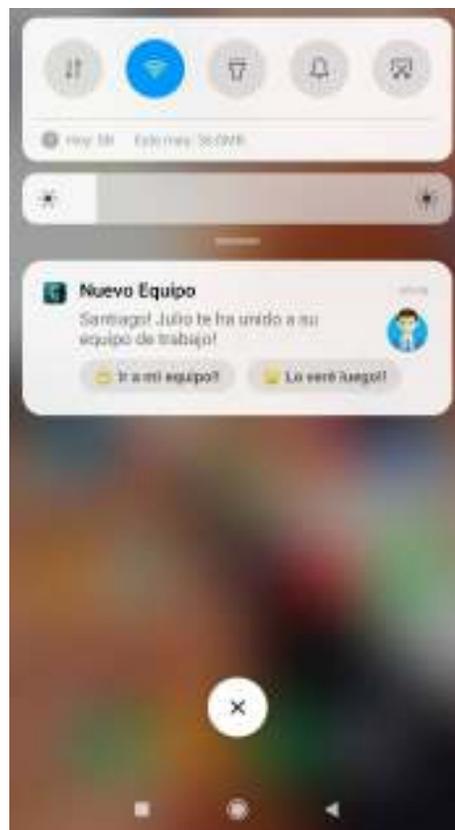
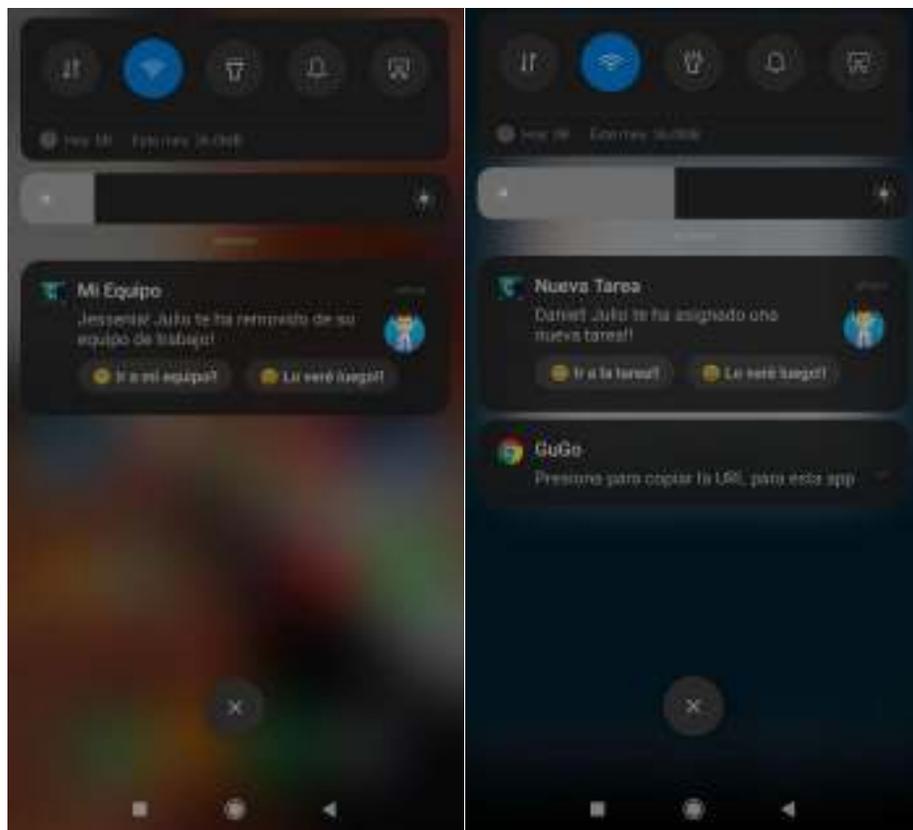


Figura 3.75. Ejemplos de notificaciones que recibe el Delegado – Aplicación móvil

Cada vez que el usuario reciba una nueva notificación, el icono Notificaciones indicará el número de notificaciones que ha recibido, como se presenta en la Figura 3.76.

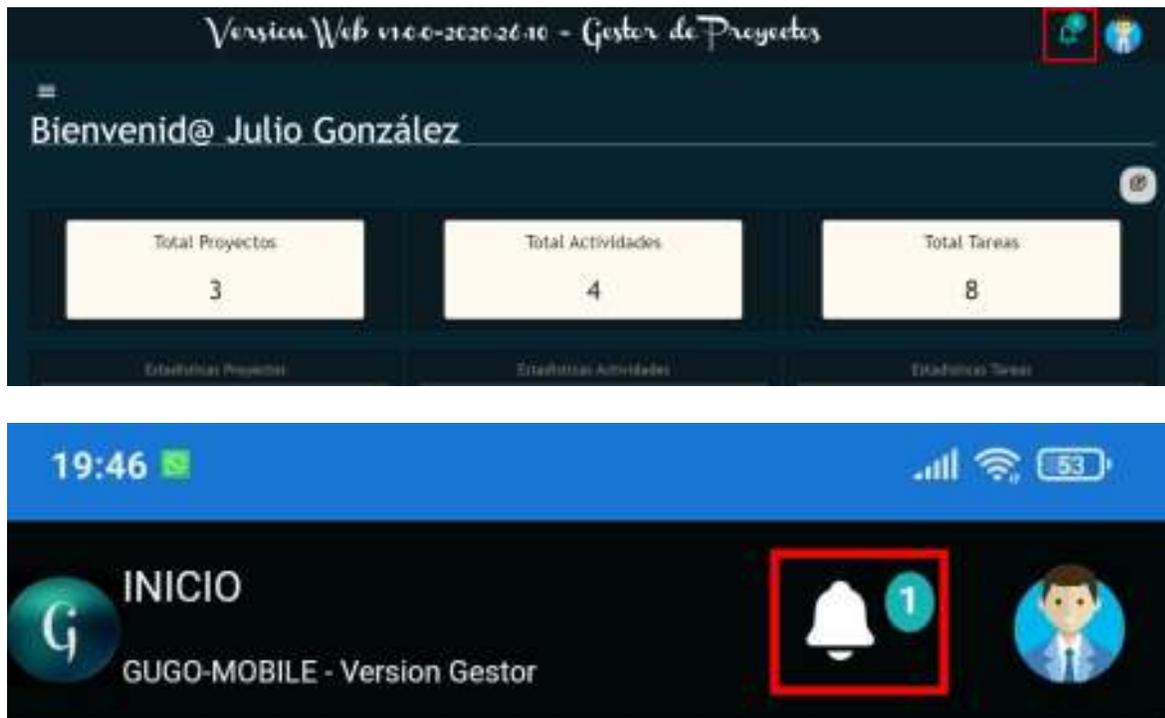


Figura 3.76. Número de notificaciones que recibe el usuario

3.2 PRUEBAS DE INTEGRACIÓN

Las pruebas de integración constan de la comunicación entre aplicaciones y los servicios de Firebase. Para realizar estas pruebas, se realizan acciones sobre una aplicación y los resultados se observan sobre la otra aplicación, así como en los servicios de Firebase.

En primera instancia se presenta la lista de usuarios que se encuentran autenticados en el servicio *Firebase Authentication*.

En la Figura 3.77 se muestra esta lista y en el recuadro rojo se presentan los usuarios que se registraron en el módulo registro de las pruebas individuales.

Identificador	Proveedor	Fecha de creación	Fecha de cambio	UID de usuario
josena.chaparrero@gmail.com	Google	2 ene. 2021	2 ene. 2021	8Eo2Bv34417c2p3kagp0Rv2
simo.gonzalez@hem.edu.ec	Google	2 ene. 2021	2 ene. 2021	0341Dq1E2TWA8P8eajp0tj1
romero.g@ferminico.com	Google	2 ene. 2021	2 ene. 2021	QsDre9j0q71vbr0hT03M20u02
carla.galtona@gmail.com	Google	2 ene. 2021	2 ene. 2021	d9K04p17Mgag0u0p70q2e7g1
carla.rivarcano@epn.edu.ec	Google	2 ene. 2021	2 ene. 2021	nn0206w030M6j0f0e4A03h28R...
centigrados@epn.edu.ec	Google	2 ene. 2021	2 ene. 2021	u7F07w9j7e0y0j0d020000w0...
sofia.fernandez@hotmail.com	Google	2 ene. 2021	2 ene. 2021	y57B04120W02H000000200x1

Figura 3.77. Usuarios registrados en Firebase Authentication para las pruebas individuales

Debido a que en las pruebas individuales se realizaron pruebas en cada aplicación para probar su funcionalidad, aquí se trabaja con dos ejemplos que permiten verificar la comunicación entre las aplicaciones.

Un ejemplo se presenta en la comunicación entre Gestor y Delegado mediante el Chat; otro ejemplo se presenta en la actualización de la foto de perfil de un Delegado a través del acceso a la galería del dispositivo y la visualización de la nueva imagen a través del aplicativo web y su almacenamiento el servicio *Cloud Storage*.

A continuación, se presenta una conversación entre el Gestor y el Delegado, el Gestor envía mensajes a través del aplicativo web como se observa en la Figura 3.78 y el Delegado envía mensajes desde el aplicativo móvil como se presenta en la Figura 3.79.

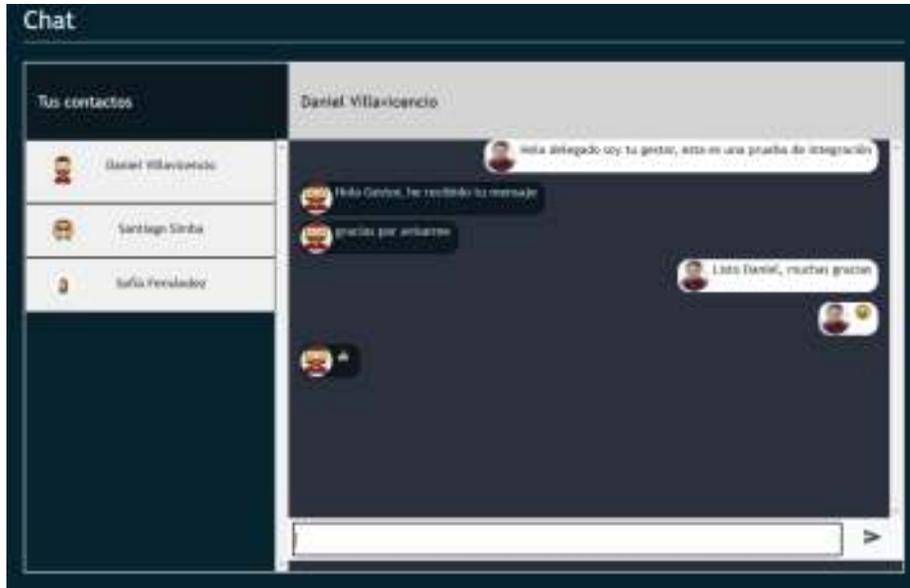


Figura 3.78. Pantalla Chat para el Gestor



Figura 3.79. Pantalla Chat para el Delegado

En la Figura 3.80 se observa la actualización de foto de perfil mediante la aplicación móvil. En la Figura 3.81 se presenta la imagen almacenada en el servicio *Cloud Storage*; y

finalmente, la visualización de la foto de perfil mediante la aplicación web se muestra en la Figura 3.82.

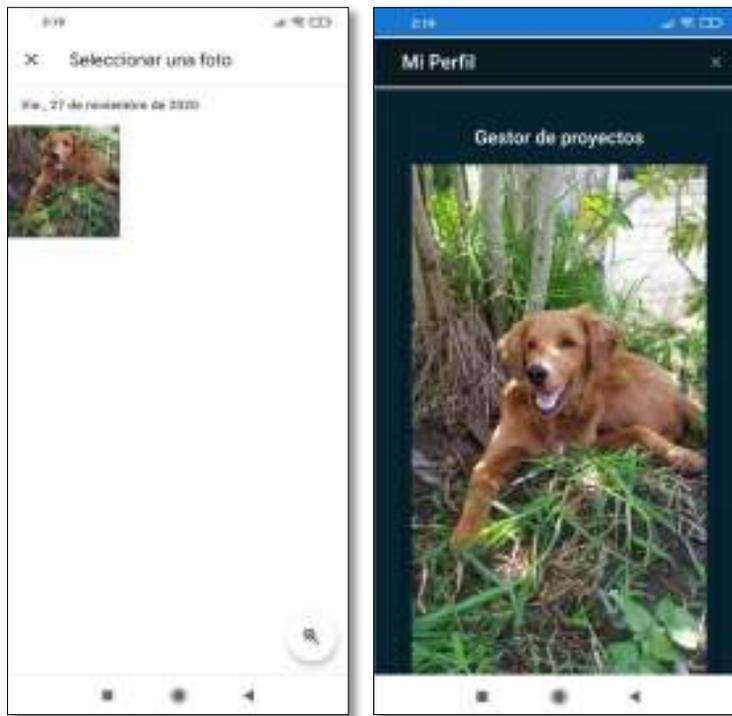


Figura 3.80. Acceso a los archivos del dispositivo móvil

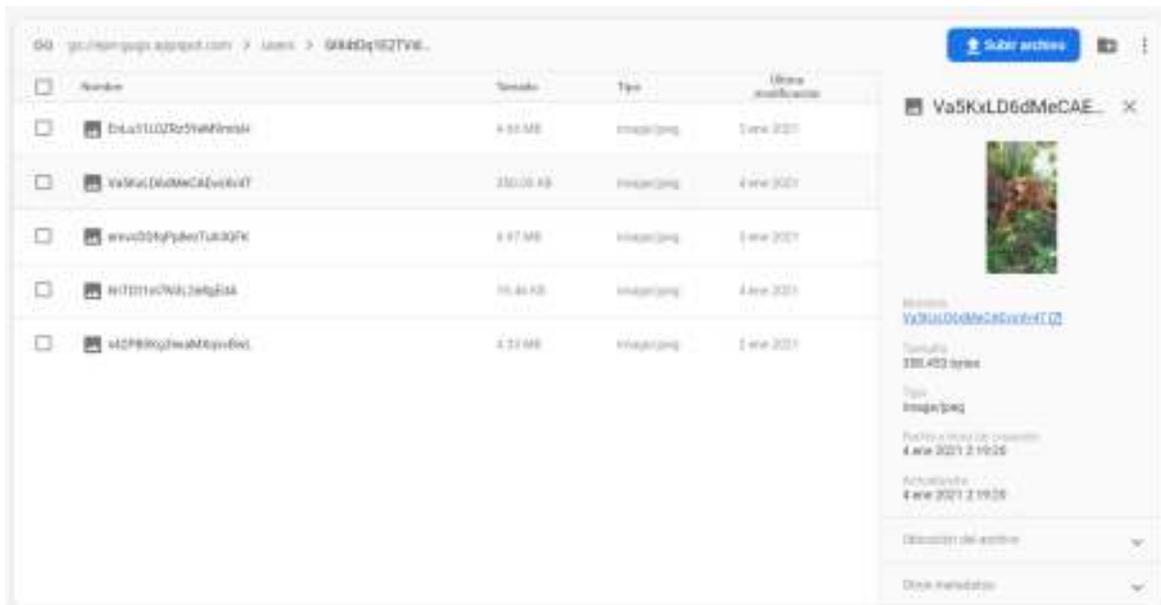


Figura 3.81. Almacenamiento de la imagen en Cloud Storage



Figura 3.82. Visualización de la nueva foto de perfil

3.2.1 VERIFICACIÓN DE PWA

Para verificar que la aplicación móvil es una PWA, se muestra que el *service worker* se ha registrado en el navegador Chrome, tal como se presenta en la Figura 3.83. Una vez registrado, este *service worker* permitirá que la aplicación se instale sobre el dispositivo móvil, lo cual se presentó en la sección 2.2.4.1.

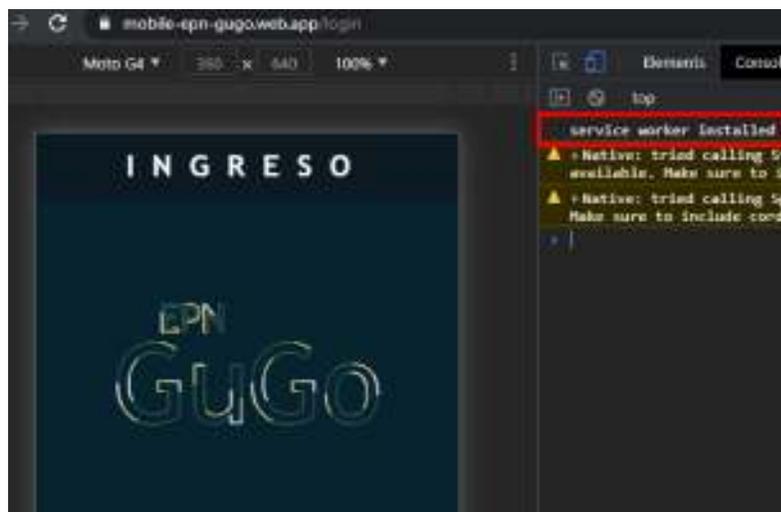


Figura 3.83. Service worker instalado

Además, se ha utilizado la herramienta Lighthouse²⁰ para observar de una manera más detallada como la PWA se está ejecutando. En la Figura 3.84 se presentan los parámetros con los que cumple para ser una PWA, entre ellos se tiene, rapidez y confiabilidad, así como que sea instalable y optimizada.

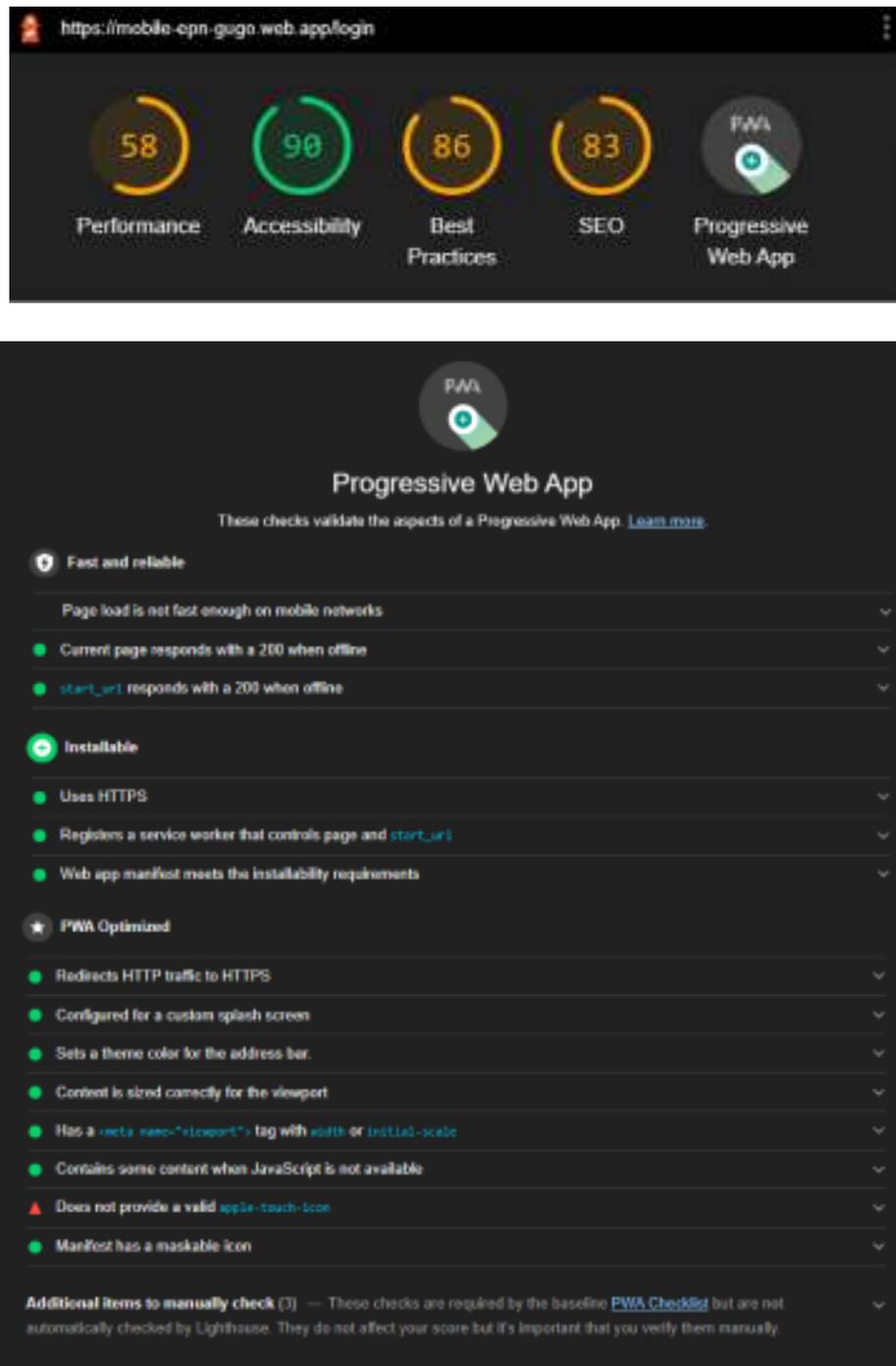


Figura 3.84. Parámetros de Lighthouse para la aplicación móvil

²⁰ Lighthouse: es una herramienta utilizada para la mejora del rendimiento, calidad y corrección de aplicaciones web.

Si bien la instalación de la aplicación móvil en un dispositivo se la efectuó en un navegador Chrome, también se realizaron pruebas en otros navegadores, un ejemplo de ello Firefox. Sin embargo, se evidenció que el comportamiento era distinto para este proceso, la causa de esto se detalla en la Figura 3.85.

En la Figura 3.85 se presenta la compatibilidad de los navegadores para presentar el cuadro de instalación (*prompt*) de una PWA. Como se observa, los navegadores Firefox Android, Opera Android y iOS Safari no son compatibles con el *prompt* mientras que el navegador Chrome Android si es compatible.

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	🖥️						📱					
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox Android	Opera Android	iOS Safari	Samsung Internet
<code>BeforeInstallPromptEvent</code> 🚩 🚩	45	≤79	No	No	No	No	45	45	No	No	No	5.0
<code>BeforeInstallPromptEvent() constructor</code> 🚩 🚩	45	≤79	No	No	No	No	45	45	No	No	No	5.0
<code>platforms</code> 🚩 🚩	45	≤79	No	No	No	No	45	45	No	No	No	5.0
<code>prompt</code> 🚩 🚩	45	≤79	No	No	No	No	45	45	No	No	No	5.0
<code>userChoice</code> 🚩 🚩	45	≤79	No	No	No	No	45	45	No	No	No	5.0

Figura 3.85. Compatibilidad del *prompt* de instalación en los navegadores

3.3 CORRECCIÓN DE ERRORES

En el desarrollo del sistema se encontraron múltiples errores, los cuales fueron resueltos en el transcurso de la realización de este trabajo. En la Tabla 3.3 y en la Tabla 3.4 se observan los errores que se presentaron y la solución de éstos.

Tabla 3.3. Corrección de errores – Parte 1

Error	Solución
Pérdida de rendimiento del sistema web y móvil, debido al exceso de lecturas en Firestore.	Optimización del código fuente del sistema para evitar el exceso de lecturas y escrituras en Firestore.
Duplicación de miembros de un mismo equipo.	Tomando en cuenta la id de usuario, se utilizó un filtro para evitar dicha duplicación.
Firebase no permite el despliegue de ciertas funciones en el plan gratuito.	Se cambió la cuenta de Firebase a pago por uso para la habilitación de dichas funciones.
El <i>service worker</i> de la PWA no se registraba en el navegador, causando problemas en la instalación de la aplicación móvil	Se agregó un <i>script</i> en el index.html para que realice el registro del <i>service worker</i> apenas el usuario inicie la aplicación.
La foto de perfil no se actualizaba en la aplicación móvil.	Existía un problema en los datos locales del dispositivo, por lo que en cada actualización de foto de perfil se realizó una lectura en Firestore de la foto recién subida y guardada de inmediato en los datos locales.
No se abre la cámara en la aplicación móvil	Para un adecuado funcionamiento de la cámara en los dispositivos se utilizó un evento para abrir la cámara a través del navegador y no con instalación de librerías externas.
En la visualización de tablas se perdían los títulos de proyectos o actividades.	Se realizó un código en CSS que mantenga los títulos sin importar el tamaño de la fuente, adaptándose de manera correcta al tamaño de la pantalla.
El usuario recibe los mensajes de chat de manera desordenada.	Se realizaron consultas integrando un atributo que contenía la fecha de creación y hora de cada mensaje, permitiendo así ordenar dichos mensajes de forma ascendente según su fecha de creación y hora.

Tabla 3.4. Corrección de errores – Parte 2

Error	Solución
El usuario recibe los mensajes de chat de manera desordenada.	Se realizaron consultas integrando un atributo que contenía la fecha de creación y hora de cada mensaje, permitiendo así ordenar dichos mensajes de forma ascendente según su fecha de creación y hora.
Por cada actualización de progresos las estadísticas mostraban el doble de número de proyectos, actividades y tareas.	Esto se debía a que en cada actualización se sumaban todos los proyectos, actividades y tareas nuevamente; se solucionó con un reseteo de todas las cantidades en cada actualización de progresos, lo cual evita la acumulación de cantidades.
Las fechas de inicio de proyectos, actividades y tareas causaban conflictos en los tiempos de entrega provocando así desorganización en los mismos.	Para evitar este problema se eliminó la posibilidad de editar la fecha de inicio de un proyecto, actividad o tarea.

3.4 PRUEBAS DE VALIDACIÓN

Como se ha mencionado, las funcionalidades del sistema no dependen del uso que se le da a la aplicación (empresarial o personal) sino del rol que tenga el usuario. Por lo tanto, las pruebas de validación se las realizó mediante una encuesta a 12 personas. Dentro de estas 12 personas se ha considerado la siguiente distribución: cuatro usuarios con rol Gestor y 8 usuarios con rol Delegado; es importante considerar que las personas encuestadas trabajaron bajo el contexto del uso empresarial.

Se implementaron dos tipos de encuestas: una para los usuarios con rol Gestor y otra para los usuarios con rol Delegado, las cuales se presentan en el ANEXO D y ANEXO E respectivamente.

Todas las preguntas de las encuestas tienen como opciones de respuesta SÍ o NO. La encuesta correspondiente al Gestor consta de 18 preguntas, mientras que la encuesta correspondiente al Delegado consta de 16 preguntas. Debido a que ciertas funcionalidades son para ambos roles, las primeras 13 preguntas de cada encuesta son las mismas y los resultados se presentan en la Figura 3.86; por tanto quedan cinco preguntas extras para el

Gestor, cuyos resultados se presentan en la Figura 3.87 y tres preguntas extras para el delegado, cuyos resultados se muestran en la Figura 3.88.

El análisis de las 13 primeras preguntas será para ambos roles, es decir para un total de 12 personas; mientras que las cinco preguntas restantes para el rol Gestor se harán para un total de 4 personas y las tres preguntas restantes para el rol Delegados serán para un total de 8 personas.

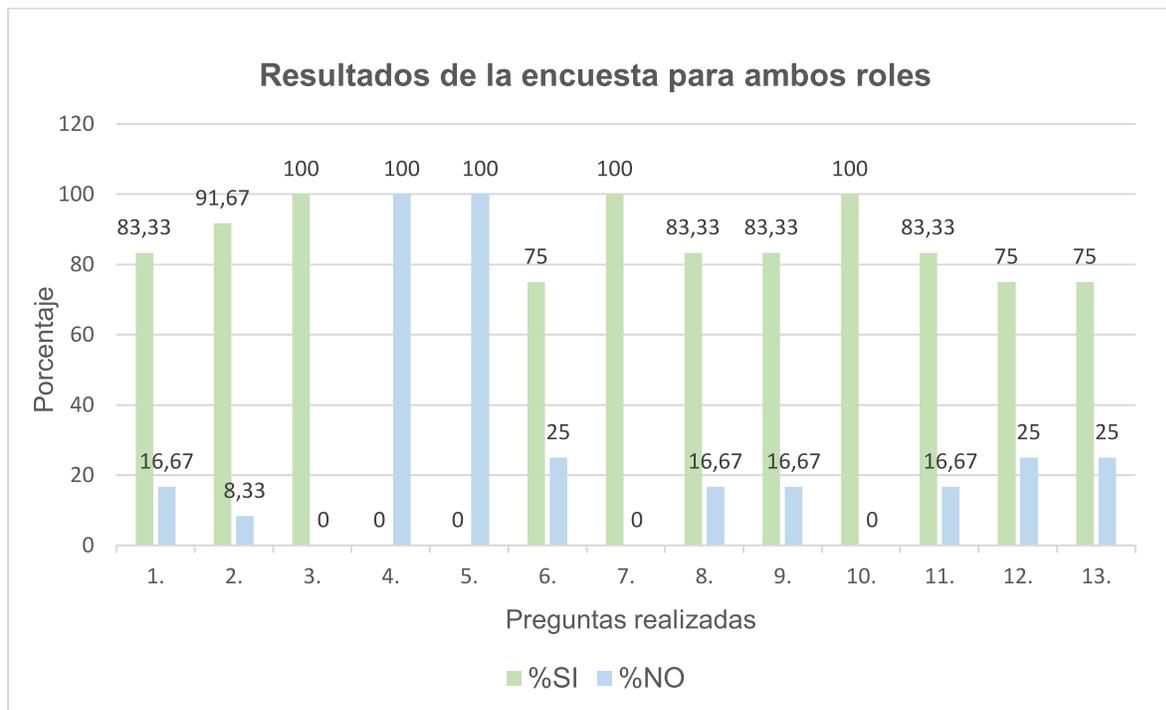


Figura 3.86. Resultados de la encuesta para ambos roles

A continuación, se realiza el análisis de los resultados más relevantes obtenidos en la encuesta para ambos roles.

La pregunta uno ¿La opción Has olvidado tu contraseña, permitió al usuario cambiar su contraseña de manera exitosa? permite que el usuario cambie su contraseña en el caso de haberla olvidado. El resultado indica que el 83.33% logró cambiar su contraseña de manera exitosa mientras que el 16.67% no lo pudo hacer.

La pregunta dos ¿El usuario tuvo respuesta al contactarse con soporte? permite que el usuario se contacte con un desarrollador del sistema para reportar anomalías o registrar a

su empresa. El resultado indica que el 91.76% obtuvo una respuesta de soporte mientras que un 8.33% no tuvo respuesta.

La pregunta tres ¿El usuario pudo instalar el aplicativo móvil en su dispositivo? obtuvo un buen resultado, ya que el 100% de usuarios encuestados logró instalar la aplicación móvil en su dispositivo.

La pregunta cuatro ¿El sistema le permitió al usuario registrarse dos veces con el mismo correo electrónico? y la pregunta cinco ¿Para uso empresarial, la aplicación le permitió al usuario registrarse con un rol distinto o un correo electrónico distinto?, ambas preguntas hacen referencia a las validaciones del formulario de registro. Las dos preguntas obtuvieron un buen resultado, ya que el 100% de usuarios encuestados no lograron registrarse dos veces con el mismo correo electrónico, y en el uso empresarial no lograron registrarse con un rol distinto.

La pregunta seis ¿La guía de usuario proporcionada por el sistema ayudó al usuario en el manejo de las aplicaciones? permite darle una breve explicación al usuario de cómo utilizar el sistema. El resultado indica que el 75% de usuarios encuestados consideran que la guía de usuario ayudó en el manejo del sistema y el 25% considera que no ayudó.

La pregunta siete ¿En el chat del sistema, el usuario pudo enviar y recibir mensajes? y la pregunta diez ¿El sistema presenta las correspondientes validaciones dentro de cada formulario? Obtuvo un buen resultado ya que en ambas preguntas el 100% de usuarios encuestados enviaron y recibieron mensajes sin ningún inconveniente y los formularios del sistema cumple con las validaciones correspondientes.

La pregunta ocho ¿El sistema permitió cargar y descargar archivos en ambas aplicaciones?, pregunta nueve ¿El usuario recibió notificaciones acerca de comentarios y carga de archivos en los recursos de un proyecto? y pregunta once ¿La aplicación móvil permitió usar la cámara del dispositivo? Obtuvieron un resultado igual, el 83.33% de los usuarios encuestados pudieron cargar y descargar archivos, recibir notificaciones y acceder a la cámara del dispositivo móvil, en tanto que el 16.67% presentó inconvenientes.

La pregunta doce ¿Considera usted que la interfaz de la aplicación web es intuitiva y fácil de utilizar? y la pregunta trece ¿Considera usted que la interfaz de la aplicación móvil es intuitiva y fácil de utilizar? Los resultados obtenidos indican que el 75% consideran que la interfaz es intuitiva y fácil de utilizar, mientras que un 25% consideran que no lo es.

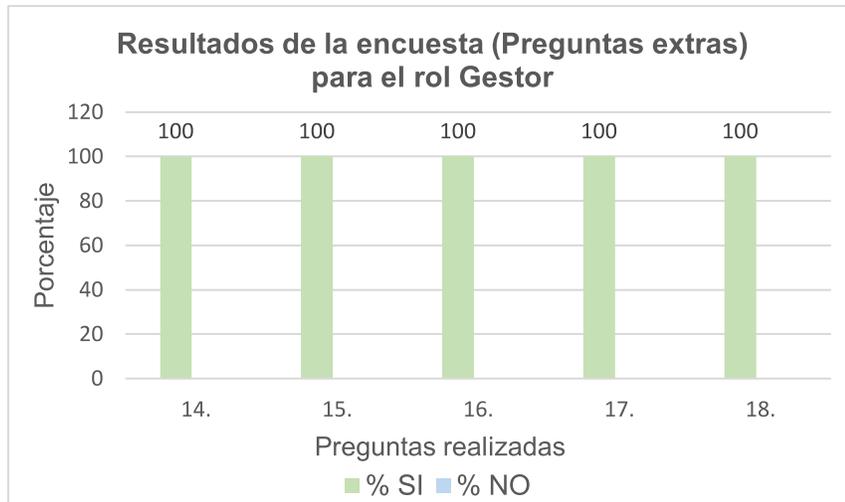


Figura 3.87. Resultados de la encuesta para el rol Gestor

A continuación, se realiza el análisis de los resultados más relevantes obtenidos en la encuesta para el rol Gestor.

Las preguntas extras realizadas al Gestor son las siguientes: pregunta catorce ¿El sistema permitió llevar un CRUD de cada proyecto que dirija?, pregunta quince ¿El sistema permitió llevar un CRUD de cada actividad?, pregunta dieciséis ¿El sistema permitió llevar un CRUD de cada tarea?, pregunta diecisiete ¿El sistema permitió asignar tareas a un usuario específico? y pregunta dieciocho ¿El usuario recibió notificaciones acerca de evidencia de tareas? Los resultados de estas preguntas fueron que el 100% de usuarios encuestados indican que las funcionalidades del sistema trabajan de manera correcta.

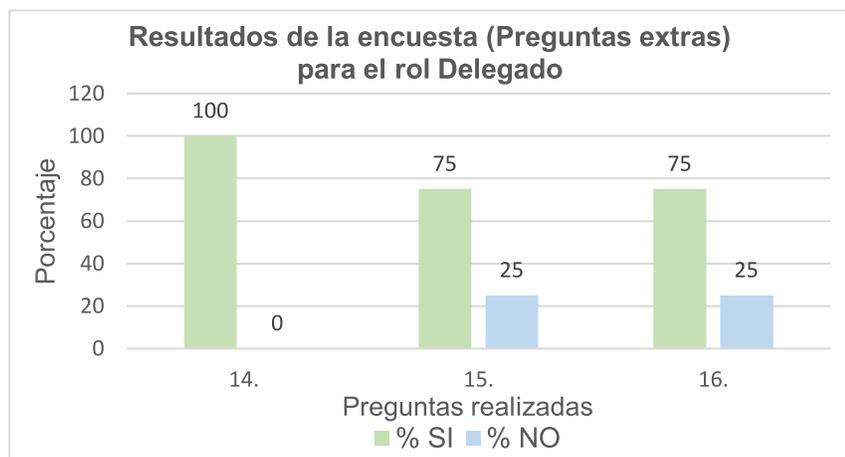


Figura 3.88. Resultados de la encuesta para el rol Delegado

A continuación, se realiza el análisis de los resultados más relevantes obtenidos en la encuesta para el rol Delegado.

Las preguntas extras realizadas al Delegado son las siguientes: pregunta catorce ¿El usuario pudo visualizar la lista de Gestores que lo han seleccionado como parte de su equipo de trabajo? pregunta quince ¿El usuario recibió notificaciones acerca de cuándo lo han seleccionado para formar parte de un equipo de trabajo? Y pregunta dieciséis ¿El usuario recibió notificaciones acerca de cuándo ha sido asignado a cargo de una tarea? Los resultados indican que en la pregunta catorce el 100% de usuarios encuestados indican que se visualizó de manera correcta su lista de gestores; en cuanto a las preguntas quince y dieciséis el 75% indican que recibieron las notificaciones correspondientes, mientras que el 25% tuvieron problemas con las notificaciones y no las recibieron.

Luego de indagar los resultados se determinó que:

El 75% de Delegados que no recibieron notificación, fue debido a que no aceptaron los permisos correspondientes, los cuales son necesarios para que el sistema recepte dichas notificaciones.

El 16.67% de los usuarios no pudieron utilizar la cámara del dispositivo debido a que el navegador presentó problemas al momento de ejecutar el evento correspondiente a la utilización de la cámara.

El 16.67% de los usuarios no pudieron cargar ni descargar archivos debido a que las imágenes no permiten descarga directa sino una previsualización de la imagen.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Con la realización de este trabajo de titulación, se ha desarrollado un Sistema web y móvil que permite la administración y control de proyectos. Este sistema brinda flexibilidad y eficiencia en el proceso de realización y culminación de un proyecto. Con esto se espera agilizar el trabajo dentro de una empresa y optimizar el tiempo de cada usuario.
- Firebase es una plataforma que ofrece servicios implementados en la nube, por lo tanto, el usuario se preocupa solo de desarrollar funciones que consuman estos servicios. Esto ha permitido añadir distintas funcionalidades al Sistema tales como; manejo de evidencia a través del almacenamiento de archivos en *Cloud Storage*, manejo de notificaciones de acuerdo al servicio *Cloud Messaging*, lectura y escritura de datos por medio del servicio de *Cloud Firestore* y alojamiento del Sistema sobre el servicio *Firebase Hosting*.
- El manejo de versiones en el desarrollo del Sistema fue primordial, ya que ayuda a organizar el código fuente de una mejor manera, evita la pérdida de información, permite dividir el desarrollo de las características del Sistema y trabajar en equipo.
- La plataforma Firebase proporcionó la facilidad de alojar y publicar ambas aplicaciones mediante su hosting *gratuito*, brindando un dominio sencillo, con el cual se puede identificar el Sistema en la web.
- A diferencia de otros sistemas, no se ha levantado ningún tipo de infraestructura, ya que con la implementación de Firebase, es Google quien se encarga de brindar un entorno administrado de servidores. Con esto se evita la configuración, mantenimiento y aprovisionamiento de servidores, lo que cual puede llegar a ser costoso en tiempo y dinero.
- Realizar las encuestas iniciales hacia usuarios de una empresa que lleven a diario informes acerca de la realización de sus proyectos, fue clave para la obtención de los requerimientos funcionales y no funcionales del Sistema web y móvil. Una vez

recopilada la información se tomó en cuenta que es necesario tener archivos que certifiquen el cumplimiento de las tareas.

- Debido a que Ionic se integra fácilmente con otros *frameworks* como Angular, el trabajo en conjunto de ambos ha permitido aligerar el desarrollo del Sistema web y móvil. De esta manera se logra la reutilización de componentes web en el desarrollo de la plataforma móvil, ya que permitiría la reducción de costos en desarrollo al no necesitar profesionales para cada plataforma.
- El rendimiento de la aplicación móvil es mejor con respecto a otras aplicaciones no nativas, debido a que Ionic aprovecha las nuevas tecnologías web (HTML5, CSS3), haciendo que los componentes que integran la aplicación trabajen directamente con Javascript, con lo que se tienen respuestas bastante rápidas en su ejecución. Sin embargo, este rendimiento siempre será menor al de una aplicación nativa.
- Para evitar el exceso de lecturas en el Sistema, se ha optimizado el código mediante el cierre de escucha de datos en tiempo real cada vez que se cambien de una pantalla a otra, y así prevenir la fuga de memoria en el Sistema y las lecturas innecesarias en Firestore. Esto a largo plazo es importante ya que ahorraría costos de lectura en el plan de pago de Firebase.
- El Sistema proporciona el manejo de evidencias y porcentaje de avance de cada tarea, lo que permitirá mejorar el control sobre cada proyecto y que cada Delegado tenga una realimentación por parte de un Gestor de proyectos, mejorando la calidad de trabajo e incluso la productividad de los delegados.
- Las aplicaciones web y móvil presentan un esquema de gráficos iniciales acerca de las estadísticas de proyectos, actividades y tareas, para el Gestor de proyectos. Esto permite que el Gestor tenga una visual global acerca del estado de sus proyectos y así pueda mejorar la estrategia de trabajo.
- La aplicación móvil al momento de ser instalada tiene un peso ligero en comparación a otras aplicaciones móviles, lo cual constituye una de las grandes ventajas de ser una PWA. Esto último podría ser muy útil para dispositivos que tengan poca memoria de almacenamiento.

- El servicio de *hosting* de Firebase ha permitido hospedar las dos aplicaciones dentro del mismo servicio, brindando así subdominios correspondientes para cada sitio. Esto posibilita personalizar el nombre de cada subdominio, proceso que se ha llevado a cabo para el alojamiento del Sistema web y móvil.
- Debido a la configuración PWA en la aplicación móvil, se pudo visualizar la información a pesar de que ésta se encuentre sin conexión a Internet. Esto permite que los usuarios puedan acceder a los datos de la aplicación desde cualquier lugar y en cualquier momento.
- El uso de Firebase permite que el Sistema web y móvil sea escalable y no tan costoso, en el caso de que el registro de empresas o usuarios crezca. Esto gracias a que Firebase ofrece un plan de pago económico en todos sus servicios, además de brindar un límite de uso gratuito por día.
- De acuerdo con los diferentes temas de estilos y colores proporcionados tanto por Angular Material para la aplicación web y los elementos gráficos propios de Ionic para la aplicación móvil, el sistema ha sido optimizado en los diferentes modos (modo oscuro o modo claro) de presentación en navegadores web y dispositivos móviles, para mayor comodidad para los usuarios.
- La utilización de Ionic permite empaquetar el mismo proyecto en diferentes versiones, ya sea como aplicación híbrida, web o PWA, logrando que este Sistema web y móvil, con los respectivos ajustes, pueda ser presentado como una aplicación ejecutable en versiones Android e iOS, lo que permite que sea subido a las diferentes tiendas virtuales.
- El manejo de equipos por proyectos ha permitido que los miembros de cada equipo, sea Gestor o Delegado, puedan tener una visualización más ordenada de los proyectos a los que pertenecen, esto con la finalidad de evitar el trabajo de buscar recursos o personas en algún proyecto específico.

4.2 RECOMENDACIONES

- A medida que el número de empresas o usuarios crezca va a ser necesario disponer de un Administrador del Sistema, por lo tanto, se recomienda la creación de este nuevo rol, enfocado sobre todo en el registro de empresas con sus Gestores y Delegados; actualmente toda esa información se añade mediante el panel de Firestore.
- La aplicación móvil se encuentra gráficamente disponible para los sistemas Android e iOS, pero el sistema operativo iOS no permite la recepción de notificaciones, por lo que para iOS se recomienda una configuración en Ionic de recepción de notificaciones bajo el contexto de aplicación híbrida y no de una PWA.
- En los formularios de carga de archivos como son recursos y evidencias, se recomienda realizar una compresión de los mismos antes de que sean subidos a Firebase, ya que éstos pueden rebasar los límites de almacenamiento en los contenedores de *Cloud Storage*.
- Para el tema de realización de tareas por parte de los Delegados a través de evidencias y subida de porcentajes de avance por cada tarea, se podría agregar una funcionalidad de seguimiento diario de acciones específicas que ejecuta un Delegado por cada tarea asignada.
- Se podría agregar la funcionalidad de reportes semanales o mensuales de cómo ha llevado cada Delegado las tareas a él asignadas.
- Sería de gran utilidad poder importar o exportar datos de proyectos, para mayor facilidad del Gestor, por lo que se recomienda agregar librerías que ayuden con dichas tareas.
- Se recomienda tener actualizados los CLI de cada *framework*, de otra manera será complicado hacer una modificación al sistema; también es importante que la instalación de librerías externas no se encuentre discontinuada ya que podría causar daño a todo el proyecto.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] Jose Gregorio Guevara Liendo, "Gestión de Proyectos con Project y Excel," *Udemy*, 2019. [Online]. Available: <https://www.udemy.com/course/gestion-de-proyectos-con-project-y-excel-bajo-enfoque-pmbok/>. [Accessed: 11-Feb-2020].
- [2] M.V. Pruneda, "Diferencias entre aplicaciones Web y Desktop," *Buyto*. [Online]. Available: <http://www.buyto.es/general-diseno-web/diferencias-entre-aplicaciones-web-y-aplicaciones-desktop.html>. [Accessed: 11-Feb-2020].
- [3] Angular, "One Framework Mobile & Desktop," 2010. [Online]. Available: <https://angular.io/>. [Accessed: 06-Feb-2020].
- [4] Ionic, "One codebase. Any platform.," 2013. [Online]. Available: <https://ionicframework.com/>. [Accessed: 06-Feb-2020].
- [5] Angular, "Angular Material- Material Design components for Angular," 2010. [Online]. Available: <https://material.angular.io/>. [Accessed: 06-Feb-2020].
- [6] Google group, "Documentación," 2010. [Online]. Available: <https://firebase.google.com/docs>.
- [7] Quality devs, "¿Qué es Angular y para qué sirve?," *Quality devs*, 2019. [Online]. Available: <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>. [Accessed: 22-Feb-2020].
- [8] Miguel Angel Alvarez, "¿Qué es una SPA?," *desarrolladoresweb.com*, 2010. [Online]. Available: <https://desarrolloweb.com/articulos/que-es-una-spa.html>. [Accessed: 22-Feb-2020].
- [9] GitBook, "SPA," 2015. [Online]. Available: https://juanda.gitbooks.io/webapps/content/spa/arquitectura_de_un_spa.html. [Accessed: 23-Feb-2020].
- [10] Angular, "Introducción a los conceptos angulares," *Google*, 2010. [Online]. Available: <https://angular.io/guide/architecture>.
- [11] Carlos Rojas, "Crea tu primer Progressive Web App con Ionic," *desarrolladoresweb.com*, 2017. [Online]. Available: <https://desarrolloweb.com/articulos/primer-progressive-web-app-ionic.html>.

[Accessed: 25-Feb-2020].

- [12] Yecid Alexis Rendón, "Base de datos relacionales vs. no relacionales," *Academia Pragma*, 2019. [Online]. Available: <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales>. [Accessed: 11-Feb-2020].
- [13] Amazon Web Services, "¿Qué es NoSQL?," *AWS*, 2012. [Online]. Available: <https://aws.amazon.com/es/nosql/>. [Accessed: 12-Feb-2020].
- [14] IONOS, "NoSQL: la tendencia hacia un soporte de datos estructurado," *Digital Guide Ionos*, 2020. [Online]. Available: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/nosql/>. [Accessed: 22-Feb-2020].
- [15] Google group, "Firebase y Google Cloud Platform," *Firebase*, 2020. [Online]. Available: <https://firebase.google.com/firebase-and-gcp?hl=es-419>. [Accessed: 25-May-2020].
- [16] Google group, "Google Cloud Plataform," *Google*, 2020. [Online]. Available: <https://console.cloud.google.com/home/dashboard?project=notificaciones-304b7&hl=es&pli=1>. [Accessed: 25-Apr-2020].
- [17] Microsoft, "Visual Studio Code," *Microsoft*, 2016. [Online]. Available: <https://code.visualstudio.com/docs>. [Accessed: 13-Feb-2020].
- [18] Google developers, "Firebase," *Firebase*, 2020. [Online]. Available: <https://console.firebase.google.com/project/test1-35487/overview>.
- [19] Israel Alcazar, "Introducción a sistemas de control de versiones, a Git y Github," *desarrolladoresweb.com*, 2020. .
- [20] Kinsta, "Git vs Github: ¿Cuál es la Diferencia y cómo Empezar?," *Kinsta*, 2020. .

ANEXOS

ORDEN DE EMPASTADO