

**UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA
PUBLICACIÓN DE ANUNCIOS DE OFERTA Y
DEMANDA LABORAL EN TIEMPO REAL**

Presentado Por

RICARDO RICHARD HUAMANTINGO NAVARRO

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

ANDAHUAYLAS – APURÍMAC – PERÚ

2018

**UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



Presentado Por

Bach. RICARDO RICHARD HUAMANTINGO NAVARRO

**APLICACIÓN MÓVIL MULTIPLATAFORMA, PARA LA
PUBLICACIÓN DE ANUNCIOS DE OFERTA Y
DEMANDA LABORAL EN TIEMPO REAL**

Asesor:

M.Sc. Cecilia Edith García Rivas Plata

Co – Asesor:

M.Sc. Iván Soria Solís

ANDAHUAYLAS – APURÍMAC – PERÚ

2018

AGRADECIMIENTO

A dios por permitirme ser parte de este mundo y darme cuenta de que el conocimiento no es absoluto.

A mis padres Ricardo y Lisbeth, quienes fueron los que me dieron el mejor ejemplo de vida, me formaron para ser una persona fuerte y resistente a las adversidades, me incitaron al camino del conocimiento, por darme la vida y la oportunidad de hacer un mundo mejor.

A mis abuelitos Teófilo y Candelaria, con quienes crecí y me cuidaron como a más que un hijo.

A mis asesores del proyecto, que fueron también mis docentes y compartieron sus conocimientos, a la M.Sc. Cecilia Edith García Rivas Plata y al M.Sc. Iván Soria Solís, por brindar su apoyo en la elaboración de este proyecto.

Un sincero agradecimiento a mis jurados: Dr. Angel Fernando Navarro Raymundo, Ing. Enrique Edgardo Condor Tinoco y MSc. Edwin Roque Tito, con quienes compartí conocimientos y experiencias gratas.

A todos mis docentes y mentores de la Universidad, quienes fueron un ejemplo y formaron parte de mi crecimiento académico y profesional a quienes estaré siempre agradecido.

A las personas que de alguna u otra manera apoyaron de manera indirecta, para que este proyecto se elabore exitosamente.

A todas las personas y desarrolladores en el mundo que hacen que el conocimiento sea libre y para todos.

TABLA DE CONTENIDOS

	Pág.
RESUMEN.....	xii
ABSTRACT	xiii
INTRODUCCIÓN.....	1
1. DATOS GENERALES	2
1.1. TÍTULO DEL PROYECTO	2
1.2. AUTOR DEL PROYECTO	2
1.3. ASESOR(A) DEL PROYECTO	2
1.4. CO-ASESOR DEL PROYECTO	2
1.5. LÍNEA DE INVESTIGACIÓN	2
1.6. ÁREA PRIORIZADA DEL PROYECTO	2
2. PLANTEAMIENTO DEL PROBLEMA.....	3
2.1. REALIDAD PROBLEMÁTICA.....	3
2.2. FORMULACIÓN DEL PROBLEMA.....	4
2.3. OBJETIVOS	4
2.3.1. Objetivo General	4
2.3.2. Objetivos Específicos	4
2.4. JUSTIFICACIÓN	5
2.5. VIABILIDAD DE LA INVESTIGACIÓN	5
2.5.1. Viabilidad Técnica	5
2.5.1.1. Herramientas:.....	5
2.5.1.2. Equipos / Materiales:	6
2.5.1.3. Metodología:.....	6
2.5.2. Viabilidad Económica.....	6
2.5.3. Viabilidad Social.....	7
2.5.4. Viabilidad Legal.....	7
2.6. LIMITACIÓN DEL ESTUDIO	7
3. MARCO TEÓRICO	9
3.1. ANTECEDENTES	9
3.1.1. Tesis de Universidad Nacional Autónoma de México, Posgrado en Ciencia e Ingeniería de la Computación	9
3.1.2. Tesis de la Universidad de San Carlos de Guatemala, Facultad de Ingeniería – Escuela de Ingeniería en Ciencias y Sistemas.....	10
3.1.3. Tesis la Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática.....	10

3.1.4.	Tesis de la Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería.....	11
3.1.5.	Tesis de la Pontificia Universidad Católica del Perú, Facultad de Ciencias y Artes de la Comunicación.....	12
3.2.	BASES TEÓRICAS	13
3.2.1.	Empleo Informal.....	13
3.2.1.1.	Tipos de Empleos.....	13
3.2.2.	Tablón de Anuncios	14
3.2.3.	Tipos de Aplicaciones según su desarrollo.....	14
3.2.3.1.	Aplicaciones Nativas	14
3.2.3.2.	Aplicaciones Web (Web apps).....	14
3.2.3.3.	Aplicaciones Híbridas	15
3.2.4.	Tiempo Real (Real Time).....	15
3.2.5.	Dispositivo Móvil	16
3.2.5.1.	Definición.....	16
3.2.5.2.	Características.....	16
3.2.5.3.	Tipos de Dispositivos Móviles.....	17
3.2.6.	Base de Datos No Relacional.....	17
3.2.6.1.	NoSQL	17
3.2.7.	Modelo Cliente Servidor	18
3.2.8.	Test Driven Development (TDD)	19
3.2.9.	Progressive Web Apps.....	20
3.2.10.	IONIC.....	21
3.2.11.	Firebase.....	22
3.2.12.	Metodología Mobile-D	22
4.	INGENIERÍA DEL PROYECTO	24
4.1.	ADMINISTRACIÓN Y GESTIÓN DEL PROYECTO BASADO EN PMBOK	24
4.1.1.	Plan de Gestión de la Integración	24
4.1.2.	Plan de Gestión del Alcance	27
4.1.3.	Plan de Gestión del Tiempo	28
4.1.4.	Plan de Gestión de los Costos	29
4.1.5.	Plan de Gestión de la Calidad	30
4.1.6.	Plan de Gestión de los Recursos Humanos	31
4.1.7.	Plan de Gestión de las Comunicaciones	32
4.1.8.	Plan de Gestión de Riesgos.....	33

4.1.9.	Plan de Gestión de las Adquisiciones	34
4.1.10.	Plan de Gestión de los Interesados.....	35
4.2.	METODOLOGÍA DE DESARROLLO MOBILE-D.....	37
4.2.1.	Fase de Exploración	37
4.2.1.1.	Estableciendo actores	38
4.2.1.2.	Estableciendo Usuarios	38
4.2.1.3.	Información clave del Proyecto.....	38
4.2.1.4.	Principales Requerimientos Funcionales	39
4.2.1.5.	Principales Requerimientos No Funcionales.....	39
4.2.2.	Fase de Iniciación	40
4.2.2.1.	Stack de Desarrollo	40
4.2.2.2.	Hardware de Desarrollo.....	40
4.2.2.3.	Herramientas para pruebas (Hardware)	40
4.2.2.4.	Diagramas de casos de uso de la aplicación móvil.....	40
4.2.3.	Fase de Producción	58
4.2.3.1.	Diseño de la aplicación Móvil.....	59
4.2.3.2.	Dependencias de Paquetes de la aplicación móvil	60
4.2.3.3.	Cordova – Plugins necesarias para la aplicación móvil	60
4.2.3.4.	Configuración Inicial del Proyecto.....	62
4.2.3.5.	Estructura de directorios y archivos del proyecto.....	63
4.2.4.	Fase de Estabilización	73
4.2.4.1.	Integración del proyecto con Firebase	74
4.2.4.2.	Integración de los módulos de Acceso a la Aplicación.	76
4.2.4.3.	Integración de los módulos de la Página Principal.....	77
4.2.4.4.	Integración de los módulos para la Administración de Anuncios.....	79
4.2.4.5.	Integración de los módulos de Configuración de Perfil	80
4.2.4.6.	Integración de los módulos de la Sala de Chats	80
4.2.4.7.	Integración de los Providers (Servicios).....	81
4.2.4.8.	Estructura de la base de datos en Firebase.....	84
4.2.5.	Fase de Pruebas.....	91
4.2.5.1.	Errores “undefined” para las notificaciones.....	91
4.2.5.2.	Errores de incompatibilidad de plugins al compilar el .APK de Android.....	92
4.3.	NOTIFICACIONES PUSH	95
4.3.1.	Lo que se hacía (Usando un Servidor)	96

4.3.2.	Lo que se hizo (Sin Servidor)	96
4.3.3.	Resultados de las Notificaciones Push.....	99
4.4.	REDUCCIÓN DEL MODELO CLIENTE SERVIDOR	100
4.4.1.	Modelo Cliente Servidor	100
4.4.2.	Modelo Cliente Servidor Óptimo.....	101
4.5.	MEJORA Y OPTIMIZACIÓN DEL CÓDIGO	102
4.5.1.	Mejora para la Reducción del Consumo de Datos.....	102
4.5.2.	Mejora de la Gestión de Tokens.....	105
5.	EVALUACIÓN DE LA SOLUCIÓN.....	106
5.1.	TDD (TEST DRIVEN DEVELOPMENT).....	106
5.2.	PRUEBAS FUNCIONALES	108
5.2.1.	Pruebas de Caja Negra.....	108
5.3.	PRUEBAS DE COMPATIBILIDAD.....	110
6.	RESULTADOS	112
7.	CONCLUSIONES.....	116
8.	RECOMENDACIONES.....	117
9.	REFERENCIAS BIBLIOGRÁFICAS.....	118
10.	ANEXOS	121

LISTA DE IMÁGENES

	Pág.
Figura 1. Ciclo de vida del Software con Mobile-D	37
Imagen 2. Dependencias	60
Imagen 3. Cordova – Plugins	61
Imagen 4. Configuración app.module.ts	62
Imagen 5. Directorio del Proyecto “WorkApp”	63
Imagen 6. Carpeta “src”	65
Imagen 7. Carpeta “src/app”	65
Imagen 8. Carpeta “src/models”	66
Imagen 9. Carpeta “src/pages”	67
Imagen 10. Carpeta “src/providers”	72
Imagen 11. Install angularfire2 firebase	74
Imagen 12. Nuevo Proyecto en Firebase	74
Imagen 13. Script de configuración Firebase	75
Imagen 14. Archivo de configuración Firebase	75
Imagen 15. Importar AngularFireModule	75
Imagen 16. Inicializar con la configuración	75
Imagen 17. rootPage: Login	76
Imagen 18. Formulario de Login con etiquetas de Ionic	76
Imagen 19. Login.ts	77
Imagen 20. Página Principal – métodos	77
Imagen 21. Agregar Nuevo Anuncio import	78
Imagen 22. Método Postular Anuncio	78
Imagen 23. Menu Toggle Principal	79
Imagen 24. Tabs Mis Anuncios html	79
Imagen 25. Tabs Mis Anuncios ts	79
Imagen 26. Perfil – variables	80
Imagen 27. Sala de chats amigos	80
Imagen 28. Autenticación.ts	81
Imagen 29. Chat.ts	82
Imagen 30. Geolocation.ts	82
Imagen 31. Imghandler.ts	82
Imagen 32. Notificaciones.ts	82
Imagen 33. Solicitudes.ts	83
Imagen 34. Tareas.ts	83
Imagen 35. Usuarios.ts	83
Imagen 36. WorkApp Database	84
Imagen 37. Database – amigos	85
Imagen 38. Database – anunciochats	85
Imagen 39. Database – anunciosaceptados	86
Imagen 40. Database – apptokens	87
Imagen 41. Database – personachats	88
Imagen 42. Database – solicitudanuncios	89
Imagen 43. Database – solicitudes	89
Imagen 44. Database – tareas	89
Imagen 45. Database – usuarios	90
Imagen 46. Authentication	90

Imagen 47. Storage	90
Imagen 48. Import AngularFireDatabaseModule	92
Imagen 49. Imports AngularFireDatabaseModule	92
Imagen 50. Archivos para modificar	92
Imagen 51. Workapp-FCMPlugin.gradle – Anterior	93
Imagen 52. Workapp-FCMPlugin.gradle – Corregido	94
Imagen 53. build.gradle – Anterior 1.....	94
Imagen 54. build.gradle – Anterior 2.....	94
Imagen 55. build.gradle – Corregido 1	94
Imagen 56. build.gradle – Corregido 2	95
Imagen 57. project.properties – Anterior	95
Imagen 58. project.properties – Corregido	95
Imagen 59. Notificaciones Push – Servidor NodeJS	96
Imagen 60. plugin-fcm	97
Imagen 61. firebase-tools.....	97
Imagen 62. firebase login.....	97
Imagen 63. Firebase init.....	97
Imagen 64. Directorio functions.....	97
Imagen 65. Funciones para las Notificaciones Push	98
Imagen 66. Firebase deploy.....	98
Imagen 67. Firebase Functions.....	98
Imagen 68. Notificaciones Push en el Smartphone	99
Imagen 69. Modelo Cliente –Servidor	100
Imagen 70. Modelo Cliente - Base de Datos	102
Imagen 71. Método cámara	103
Imagen 72. Método cámara optimizado	104
Imagen 73. Imágenes Storage Firebase	104
Imagen 74. Salvar tokens.....	105
Imagen 75. TDD – this.getEstrellas() - error	106
Imagen 76. TDD – getEstrellas()	106
Imagen 77. TDD – this.getEstrellas() - ok.....	107
Imagen 78. TDD – this.getEstrellas() - refactor	107
Imagen 79. Prueba de Caja Negra – Alerts.....	109
Imagen 80. Prueba de Caja Negra – Resultados Esperados	109
Imagen 81. Información del Dispositivo	113
Imagen 82. Puntuaciones.....	114
Imagen 83. Acceso a la aplicación	121
Imagen 84. Página principal.....	122
Imagen 85. Información del anuncio.....	122
Imagen 86. Menú lateral – Menu Toggle	123
Imagen 87. Administración de mis anuncios	124
Imagen 88. Solicitudes recibidas por un anuncio.....	125
Imagen 89. Solicitudes aceptadas de un anuncio.....	126
Imagen 90. Usuario contratado – calificación	127
Imagen 91. Publicar Anuncio.....	128
Imagen 92. Perfil de Usuario.	129
Imagen 93. Sala de Chats.....	130
Imagen 94. Notificaciones en la aplicación móvil.....	131

LISTA DE TABLAS

	Pág.
Tabla 1. Plan de gestión de la integración.....	24
Tabla 2. Plan de Gestión del alcance	27
Tabla 3. Plan de Gestión del Tiempo.....	28
Tabla 4. Plan de Gestión de los Costos.....	29
Tabla 5. Plan de Gestión de la Calidad	30
Tabla 6. Plan de Gestión de los Recursos Humanos.....	31
Tabla 7. Plan de Gestión de las Comunicaciones.....	32
Tabla 8. Plan de Gestión de Riesgos	33
Tabla 9. Plan de Gestión de las Adquisiciones	34
Tabla 10. Plan de Gestión de los Interesados	35
Tabla 11. Información clave del proyecto	38
Tabla 12. Caso de uso: Iniciar Aplicación - Vista Login	41
Tabla 13. Caso de uso: Registrar Usuario.....	41
Tabla 14. Caso de uso: Configuración de Avatar de Perfil.....	41
Tabla 15. Caso de uso: Anuncios sin Login.....	42
Tabla 16. Caso de uso: Vista Principal	42
Tabla 17. Caso de uso: Seleccionar anuncio.....	43
Tabla 18. Caso de uso: Mapa del Anuncio	43
Tabla 19. Caso de uso: Detalle del Anuncio	43
Tabla 20. Caso de uso: Postular a la Oferta Laboral	44
Tabla 21. Caso de uso: Perfil del Usuario.....	44
Tabla 22. Caso de uso: Publicar Anuncio.....	44
Tabla 23. Caso de uso: Menú Toggle.....	45
Tabla 24. Caso de uso: Opción Administrar - tabs.....	45
Tabla 25. Caso de uso: Mis Anuncios	46
Tabla 26. Caso de uso: Cerrar mi Anuncio	46
Tabla 27. Caso de uso: Eliminar mi Anuncio	46
Tabla 28. Caso de uso: Mis Solicitudes.....	48
Tabla 29. Caso de uso: Mis Solicitudes – Opción Agregar	48
Tabla 30. Caso de uso: Mis Solicitudes – Opción Eliminar	49
Tabla 31. Caso de uso: Mis Solicitudes – Opción Llamar	49
Tabla 32. Caso de uso: Mis Solicitudes Aceptadas	49
Tabla 33. Caso de uso: Aceptados - Chat	50
Tabla 34. Caso de uso: Aceptados - Llamar.....	50
Tabla 35. Caso de uso: Aceptados - Contratar.....	50
Tabla 36. Caso de uso: Aceptados - Bloquear	50
Tabla 37. Caso de uso: Puntuación.....	51
Tabla 38. Caso de uso: Puntuación – Evaluar	51
Tabla 39. Caso de uso: Puntuación – Evaluar – Contratador	52
Tabla 40. Caso de uso: Puntuación – Evaluar – Contratado	52
Tabla 41. Caso de uso: Perfil	53
Tabla 42. Caso de uso: Perfil – Avatar	53
Tabla 43. Caso de uso: Perfil – Nombre.....	54
Tabla 44. Caso de uso: Perfil – Apellido.....	54
Tabla 45. Caso de uso: Perfil – Teléfono.....	54
Tabla 46. Caso de uso: Perfil – Ciudad	55

Tabla 47. Caso de uso: Perfil – Ubicación	55
Tabla 48. Caso de uso: Ampliar Curriculum	55
Tabla 49. Caso de uso: Sala de Chats	56
Tabla 50. Caso de uso: Solicitudes de amistad	56
Tabla 51. Caso de uso: Amigos.....	57
Tabla 52. Caso de uso: Búsqueda de amigos	57
Tabla 53. Caso de uso: Cerrar Sesión.....	57
Tabla 54. Caso de uso: Acerca De	58
Tabla 55. Estructura en términos generales	59
Tabla 56. Directorio del Proyecto “WorkApp”	63
Tabla 57. Tipos de archivos manejados en el proyecto	65
Tabla 58. Carpeta “src/app”	66
Tabla 59. Carpeta “src/assets”	66
Tabla 60. Carpeta “src/models”	67
Tabla 61. Carpeta “src/pages” (orden alfabético).....	67
Tabla 62. Carpeta “src/providers”	72

RESUMEN

El presente informe final del proyecto de tesis que tiene como objetivo principal desarrollar una aplicación móvil, para la publicación de anuncios de oferta y demanda laboral en tiempo real; esta aplicación móvil se logra haciendo uso de las tecnologías de IONIC y Firebase, las cuales proporcionan características únicas que hacen que sea una aplicación altamente funcional.

En la ejecución de este proyecto se logró optimizar el modelo cliente servidor de tres capas las cuales son “cliente – servidor web – servidor base de datos” a solo dos capas que son “cliente – servidor base de datos”, quitando la parte de Servidor web, ya que no es necesaria para esta aplicación, porque las funcionalidades y características que implementaría un servidor web están dentro de la aplicación en los providers (Servicios), esta optimización mejora el tiempo de respuesta de las peticiones, eventos y la interacción con la base de datos, gracias a que la aplicación móvil interactúa directamente con la base de datos y además Firebase proporciona la base de datos en tiempo real.

Los eventos manejados en IONIC permiten implementar una conexión tipo publish – subscribe y esto permite tener interacción en tiempo real, esto se usa para implementar una sala de chats que permita que los usuarios coordinen por medio de la aplicación, el cual genera una experiencia de usuario muy buena.

Los plugins de Cordova permiten obtener información que hacen que la aplicación sea confiable y segura.

La aplicación móvil desarrollada hace uso de Notificaciones Push, para ser más interactivo, esto se implementa sin la necesidad de crear un servidor web que administre y gestione dichas notificaciones, todo lo hace la aplicación móvil con ayuda de FCM (Firebase Cloud Messaging).

El proyecto puede crecer e integrar más funcionalidades, gracias a que el árbol JSON de la base de datos es muy escalable.

Es posible desarrollar este proyecto sin costo en la fase de producción de la aplicación móvil, debido a que no se necesita contratar un cloud o hosting.

Palabras Clave: IONIC, Firebase, Notificaciones Push, Tiempo Real, eventos.

ABSTRACT

The present final report of the thesis project whose main objective is to develop a mobile application, for the publication of real-time labor supply and demand announcements; this mobile application is achieved by using the IONIC and Firebase technologies, which provide unique features that make it a highly functional application.

In the execution of this project it was possible to optimize the server client model of three layers which are "client - web server - server database" to only two layers that are "client - server database", removing the server part web, since it is not necessary for this application, because the functionalities and features that a web server would implement are within the application in the providers (Services), this optimization improves the response time of requests, events and interaction with the database, thanks to the fact that the mobile application interacts directly with the database and also Firebase provides the database in real time.

The events managed in IONIC allow to implement a publish-subscribe connection and this allows interaction in real time, this is used to implement a chat room that allows users to coordinate through the application, which generates a user experience very good.

Cordova plugins allow you to obtain information that makes the application reliable and safe.

The developed mobile application makes use of Push Notifications, to be more interactive, this is implemented without the need to create a web server that manages and manages such notifications, and everything is done by the mobile application with the help of FCM (Firebase Cloud Messaging).

The project can grow and integrate more functionalities, thanks to the fact that the JSON tree of the database is very scalable.

It is possible to develop this project without cost in the production phase of the mobile application, because you do not need to hire a cloud or hosting.

Key words: IONIC, Firebase, Push Notifications, Real Time, events

INTRODUCCIÓN

Las aplicaciones móviles están cambiando la forma de interactuar de las personas, ya que existen diversas aplicaciones inimaginables para hacer algunas actividades y por ello las tecnologías para desarrollarlas también están cambiando, mejorando e innovando cada día, las herramientas de desarrollo que servían bien hace dos años, ya no son tan buenas ahora, es por eso que estar actualizados es una obligación, si quieres innovar. Debido a la necesidad de las aplicaciones, surge la pregunta ¿Cómo lo hago y con qué? Una respuesta a esto se encuentra en el informe de este proyecto, el cual muestra el uso de tecnologías como IONIC y Firebase que juntos pueden dar como resultado una aplicación móvil multiplataforma sin costo de implementación en la fase de desarrollo, ni en la fase de producción, con la característica de ser muy escalable, para proyectos de gran envergadura.

Hacer que la aplicación móvil responda rápido a las peticiones es una parte fundamental y para ello se mejoró el modelo cliente servidor, también importa el consumo de datos y para reducir ese consumo se optimizó las funciones de la aplicación móvil.

En las aplicaciones móviles tener disponible las notificaciones le dan un punto extra y el usuario se hace más fiel a ella, integrar en una aplicación móvil dichas notificaciones te dará más usuarios, entonces es vital implementar esta funcionalidad y para ello se usa Notificaciones Push, se puede integrar sin la necesidad de crear un servidor que administre dichas notificaciones.

En toda aplicación que haga interactuar a los usuarios entre ellos, existe y se hable del tema de seguridad y confianza, para ello se implementa unos métodos que tengan información que generen confianza en los usuarios y le proporcione la sensación de seguridad en su interacción.

Tener disponible un chat para comunicar a los usuarios ayuda a que la aplicación sea más confiable y así mismo sea más usado por los usuarios, proporcionándoles la sensación de seguir en sus redes sociales, con las tecnologías usadas esto es fácil de integrar.

La metodología usada es Mobile-D, porque está orientada al desarrollo de aplicaciones para móviles y te permite construir tu aplicación de manera rápida.

1. DATOS GENERALES

1.1. TÍTULO DEL PROYECTO

Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real.

1.2. AUTOR DEL PROYECTO

Nombres y apellidos : Ricardo Richard Huamantingo Navarro

Escuela Profesional : Ingeniería de Sistemas

E-mail : rrichardh95@gmail.com

1.3. ASESOR(A) DEL PROYECTO

Nombres y apellidos : Cecilia Edith García Rivas Plata

Departamento Académico: Ingeniería y Tecnología

Categoría Docente : Principal a Dedicación Exclusiva

1.4. CO-ASESOR DEL PROYECTO

Nombres y apellidos : Iván Soria Solís

Departamento Académico: Ingeniería y Tecnología

Categoría Docente : Contratado a Tiempo Completo

1.5. LÍNEA DE INVESTIGACIÓN

Ingeniería de Software (Cód. 0403 05)

1.6. ÁREA PRIORIZADA DEL PROYECTO

Investigación y desarrollo para mejorar la calidad de software (Cód. 0403 0501)

2. PLANTEAMIENTO DEL PROBLEMA

2.1. REALIDAD PROBLEMÁTICA

El tablón de anuncios o también conocida como cartelera, es un instrumento en el cuál se publican diferentes necesidades desde demanda y oferta de trabajo laboral o servicios requeridos hasta ventas de artículos y publicidad de toda índole, por ello muchas personas acuden a buscar una oportunidad laboral en estos anuncios día a día.

En la actualidad una persona que busca un empleo, lo hace buscando en los tabloneros o paneles de anuncio y para ello se aproxima físicamente, en ocasiones la información ya está desactualizada, porque una vez que se contrata el publicador deja su anuncio a la deriva y otras personas siguen leyendo y tratando de contactarse con el anunciador; o no se puede contactar oportunamente con el anunciante.

Conseguir un empleo decente es difícil para el 80% de jóvenes peruanos, tardando 12 meses en encontrar uno y 8 de cada 10 jóvenes peruanos ingresarán a su primer empleo de calidad en un sector informal. (Perú21, 2016)

De acuerdo con un estudio de Nielsen de este año, el rango de jóvenes entre 18 a 25 años que busca empleo por Internet se incrementó notablemente, en los dos últimos años, en un 14%, representando un 34% del total de personas que buscan empleos por Internet, mientras los jóvenes de 26 a 35, representan un 44% del total. (Agencia de Prensa Lima Norte, 2017)

Es muy común encontrar anuncios en el que se oferten servicios o habilidades que según, Diario El Cumbe (2016), en América Latina la acumulación de habilidades proviene de tres fuentes: el 57% del aprendizaje de los colegas de trabajo, el 31% del aprendizaje mediante la práctica y el 12% de la participación en actividades de capacitación, habilidades únicas que alguien no tiene pero podría necesitar en un determinado momento.

Por todo lo expuesto, se propone desarrollar e implementar un aplicativo móvil que reemplace al tablón de anuncios y optimice la disponibilidad de oferta y

demanda, en el cual se pueda publicar ofertas y demandas laborales en tiempo real, dar a conocer las habilidades de los usuarios para ayudar a otros a solucionar sus dificultades más eficientemente y rápido.

En la parte tecnológica, para los desarrolladores se pretende dar a conocer las herramientas para poder hacer aplicaciones móviles altamente funcionales con bajo costo de implementación en la fase de producción, gracias a la ausencia de un server y mostrarles métodos mejorados para el funcionamiento óptimo y bajo consumo de datos.

Esta solución será posible gracias al uso de tecnologías como firebase, que es un servicio proporcionado por Google, el cual permitirá la independencia de un Backend y la disponibilidad de la interacción con los datos de los usuarios en tiempo real, eso significa que los datos estarán actualizados en todo momento.

2.2. FORMULACIÓN DEL PROBLEMA

El tablón de anuncios solo es un panel de publicidad estático y el anuncio no siempre es actualizado, con información incompleta de la oferta laboral y del anunciante.

2.3. OBJETIVOS

2.3.1. Objetivo General

Desarrollar un aplicativo móvil multiplataforma mediante IONIC y Firebase, para incrementar las posibilidades de encontrar anuncios actualizados de oferta y demanda laboral en tiempo real.

2.3.2. Objetivos Específicos

- Optimizar el tiempo de respuesta y petición de datos, con un modelo Cliente – Servidor más corto.
- Desarrollar métodos y funciones para permitir la verificación de información, seguridad y confianza en las publicaciones de los anuncios.

- Desarrollar un módulo de calificación dinámico e interactivo.
- Integrar Notificaciones Push sin la dependencia de un servidor.
- Desarrollar una aplicación móvil sin costo en la fase de producción.

2.4. JUSTIFICACIÓN

Uno de los problemas más grandes de un país es el desempleo y el desperdicio de talento, habilidades y conocimiento de las personas, es un punto que se ataca de diferentes maneras y hoy en día se hace uso de la tecnología para incrementar las posibilidades de emplear a una persona.

El tablón de anuncios o cartelera donde se publican diferentes necesidades, es un medio que ayuda a las personas a encontrar una posible oportunidad de trabajo o dar a conocer las habilidades y destrezas de un individuo, pero tiene límites en cuanto a la ubicación, porque se tiene que estar presente físicamente para ponerse al tanto de las últimas publicaciones y a veces desconoces si la propuesta sigue vigente o no.

Es por eso que el aplicativo móvil pondría al tablón de anuncios que se encuentra en una esquina en la mano de una persona actualizada en tiempo real con las especificaciones necesarias y los requisitos que debe cumplir, estará informado con las últimas publicaciones sin la necesidad de desplazarse de un lugar a otro.

Otro punto importante es que facilitará a las personas a resolver sus problemas diarios teniendo una amplia comunidad de usuarios dispuestos a colaborar y apoyar inmediatamente en todo lo que necesite.

2.5. VIABILIDAD DE LA INVESTIGACIÓN

2.5.1. Viabilidad Técnica

2.5.1.1. Herramientas:

Las herramientas que se mencionan son las más aptas para el desarrollo del proyecto.

- Node.Js v8.0.0.
- Ionic.
- Cordova.
- Firebase.
- Sublime Text, Visual Studio Code.
- Navegadores Chrome, Firefox.
- Conexión a Internet.

2.5.1.2. Equipos / Materiales:

- Laptop Toshiba Satellite Intel CORE i7.
- Dispositivo Móvil Acer Liquid Z220.
- Dispositivo Móvil Samsung Galaxy J7 Neo.

2.5.1.3. Metodología:

- Mobile-D.

Todos los items mencionados están disponibles y son fáciles de conseguir, descargar e instalar, por ello el proyecto es viable técnicamente.

2.5.2. Viabilidad Económica

El presente proyecto, usa herramientas libres, como Node.Js que es un entorno en tiempo de ejecución de JavaScript y que usa NPM que es un ecosistema de librerías, ambos de código abierto, Ionic que es un framework para hacer aplicaciones móviles híbridas gratuito, Firebase es un servicio de Google que proporciona base de datos entre otros, proporciona un espacio gratuito, solo se tiene que registrarse y a programar, los navegadores son gratuitos y libres, el editor Visual Studio Code es gratuito y Cordova proporciona plugins nativos gratuitos para móviles. Los equipos hardware (laptop y dispositivo móvil) son propios.

En conclusión el proyecto es viable económicamente, porque no se recurrirá a gastos presupuestales grandes, en vista que casi todo es open source.

2.5.3. Viabilidad Social

El presente proyecto que tendrá como resultado, brindar un servicio móvil en tiempo real, que le dará al usuario una posible oportunidad de encontrar un empleo o incrementar sus ingresos, con solo brindar su apoyo a otras personas que necesiten sus habilidades y destrezas.

Lo pueden usar todas las personas que quieran buscar un trabajo o que quieran hacer algo en su tiempo libre. Hoy en día todos tienen un SmartPhone disponible a la mano con conexión a internet.

2.5.4. Viabilidad Legal

Las herramientas usadas para este proyecto son gratuitas, con las siguientes licencias:

- Node.js y Ionic son de código abierto publicado bajo una licencia MIT (Massachusetts Institute of Technology), permite reutilizar software dentro de Software propietario.
- Firebase y Cordova son libres y de código abierto bajo la licencia Apache, versión 2.0. requiere la conservación del aviso de derecho de autor y el descargo con responsabilidad.

En Conclusión, por lo descrito en los tres aspectos (técnico, económico y social) para la viabilidad del proyecto y en vista de que se cuenta con todos los recursos que se invertirán en el proyecto, los cuales son alcanzables y están completamente disponibles, por lo tanto el proyecto si es viable.

2.6. LIMITACIÓN DEL ESTUDIO

- El aplicativo móvil a desarrollar solo proporcionará la información de oferta y demanda laboral que se publiquen, dejando a los usuarios la libertad de contratar o no a los solicitantes.
- Es muy importante que el usuario se registre con información real, ya que ello será una carta de presentación para el usuario ofertante y demandante.

- Está dirigido a personas que tengan a la mano un Smartphone y de la misma manera que tengan experiencia usando servicios por internet, para no incurrir en malos usos de la aplicación.
- El acceso a internet, podría suponer una limitación para el usuario final, ya que todo el aplicativo gestiona datos de la nube en tiempo real.
- La aplicación funcionará perfectamente desde la versión de Android 4.0 en adelante, para versiones inferiores existe riesgo de incompatibilidad.
- La aplicación final solo se generará para la plataforma Android, debido a que para generar para iOS se requiere de una Mac, el cual no se tiene disponible para este proyecto.

3. MARCO TEÓRICO

3.1. ANTECEDENTES

Hoy en día el uso del internet es indispensable, las tecnologías que se usan ahora son robustas, gracias a las investigaciones y aportes de grandes desarrolladores a nivel mundial y ahora está a nuestro alcance.

Los servicios de todo tipo que hoy en día están disponibles, han llegado a cambiar la forma de pensar e interactuar con el entorno de las personas y seguirán cambiando, con la finalidad de mejorar la calidad de vida y facilitar cada vez más todas las cosas.

3.1.1. Tesis de Universidad Nacional Autónoma de México, Posgrado en Ciencia e Ingeniería de la Computación

“PROCESOS EN EL DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES” (Ian Moisés Rangel Villagrán, 2013)

En este trabajo de investigación se realizó una revisión bibliográfica sistemática que identifica aquellas particularidades del desarrollo de aplicaciones para dispositivos móviles, sus principales dificultades y soluciones actuales. A lo largo de la investigación se tuvieron contacto con grupos de trabajo en México dedicados al desarrollo de aplicaciones para dispositivos móviles, con la finalidad de recolectar información con base a sus experiencias. Las técnicas utilizadas fueron de observación, encuestas y entrevistas sobre su forma de trabajo y principales dificultades que enfrentan. Finalmente como resultado de su validación se documentaron los ajustes necesarios del proceso de desarrollo de aplicaciones para dispositivos móviles.

Después del análisis de esta tesis y los resultados que obtuvieron, se ha tomado como guía para nuestro trabajo por las siguientes razones:

- Los grupos de trabajo usan metodologías ágiles como Mobile-D, para desarrollar aplicaciones para dispositivos móviles.

- Mobile-D está optimizado para entregar una aplicación para dispositivos móviles completamente funcional en un corto periodo de tiempo.

3.1.2. Tesis de la Universidad de San Carlos de Guatemala, Facultad de Ingeniería – Escuela de Ingeniería en Ciencias y Sistemas.

“DISEÑO DE LA INVESTIGACIÓN DEL PROYECTO DANTE SISTEMA MÓVIL PARA LA NOTIFICACIÓN, UBICACIÓN Y ALERTA DE INCIDENTES Y ÁREAS DE PELIGRO EN LA CIUDAD DE GUATEMALA” (Bacilio Alexander Bolaños Lima, 2013)

En este proyecto de tesis se busca mejorar la seguridad de los usuarios de dispositivos móviles, al brindarles información sobre la ocurrencia de incidentes de riesgo: como hechos delictivos, accidentes de tránsito y desastres naturales en la Ciudad de Guatemala. La aplicación que se desarrolló pretende mejorar la comunicación entre los usuarios del sistema y las autoridades correspondientes, de forma que la respuesta coordinada de los implicados, sea eficaz y eficiente en cada situación. El Proyecto busca que el sistema alcance un alto impacto en la comunidad, por beneficiar no sólo a los usuarios de la aplicación y autoridades implicadas, sino también a toda la sociedad en general.

Luego de revisar cuidadosamente la tesis mencionada, el cual ha sido desarrollado usando una metodología ágil, se tomará como aporte la elección por una metodología ágil para el desarrollo de la aplicación móvil del proyecto. Cabe mencionar que en esta tesis se tenía la necesidad de desarrollar en poco tiempo teniendo en cuenta que el equipo para el desarrollo de la aplicación era pequeño, esta situación nos ayuda, ya que se tiene la misma necesidad y para ello se opta por la metodología Mobile-D.

3.1.3. Tesis la Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática.

“PROTOTIPO DE APLICACIÓN MOVIL UTILIZANDO LA METODOLOGIA MOBILE-D PARA LA VERIFICACION DE LA FORMALIDAD EN EL

SERVICIO DE TAXI METROPOLITANO EN LA CIUDAD DE LIMA”

(Meneses Sanchez, Jesús D y Laveriano Meca, Elva C., 2016)

En esta tesina se propone el desarrollo de un prototipo de aplicación móvil cuyo objetivo es informar acerca del servicio de taxi al ciudadano, mediante el uso del Smartphone con disponibilidad de internet para que dicha aplicación pueda hacer la lectura de la placa del taxi. La información sobre un vehículo que brinda el servicio de taxi puede apoyar al ciudadano a decidir antes de tomar dicho servicio, siendo esto una ayuda para garantizar la seguridad y confiabilidad en el servicio.

Al igual que la anterior en esta tesina, opta claramente por una metodología ágil que es Mobile-D, debido a que son más convenientes para la implementación y desarrollo de las aplicaciones móviles. Siendo una aplicación móvil hecha con Mobile-D ayudará mucho al presente proyecto por las siguientes razones:

- Usar Mobile-D es más conveniente, debido a la rapidez con la que cambia el entorno de las tendencias tecnológicas y del mercado.
- Se trata de alcanzar iteraciones de desarrollo muy rápidos con equipos muy pequeños y reducido de programadores.
- Hace buen uso de las buenas prácticas y recomendaciones ampliamente reconocidas de otras metodologías de desarrollo ágil y se adapta óptimamente a las características de las aplicaciones móviles.

3.1.4. Tesis de la Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería.

“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN PARA ADMINISTRAR Y CONSULTAR AVISOS CLASIFICADOS PARA TABLETAS ANDROID” (Cornejo Aramayo, Jorge F. 2013)

En esta tesis se aborda el problema que radica en la necesidad de las personas por publicar o consultar sobre diferentes cosas, mientras se encuentran desarrollando otra actividad y las herramientas actuales como periódicos, páginas web o aplicaciones móviles, cuentan con pasos tediosos

e inclusive no pueden utilizarse debido a que las personas no siempre tienen a la mano una PC o estar cerca de una agencia de avisos clasificados. Actualmente, el consumismo que existe ha hecho que el comercio sea parte inherente de las personas, los avisos y anuncios clasificados son un medio de comunicación y se tornan fundamentales cuando se desea ofertar o tomar un servicio y/o producto.

Este trabajo que tiene como resultado final, el lanzamiento de una aplicación para tablets, será de guía, ya que toma varias características que suelen presentarse al momento de hacer un aviso clasificado por diferentes medios físicos (periódicos, revistas) y electrónicos (páginas web) y los integra de forma adecuada, estructurada y además con un buen diseño móvil.

3.1.5. Tesis de la Pontificia Universidad Católica del Perú, Facultad de Ciencias y Artes de la Comunicación.

“El Panel: Potencial publicitario en constante cambio frente al desarrollo comercial de la Ciudad de Lima” (Jara Rojas, Claudia M., 2008)

La tesis en mención muestra como las más grandes y diversas empresas, grandes o pequeñas entidades del Estado, ONGs o hasta grupos sociales; emplean este medio para anunciar sus productos y servicios, comunicar sus ideas o hacer llamados de toma de conciencia para que la actitud del que lo ve, cambie. Y nos deja muy en claro que los paneles publicitarios o llamados por algunos carteles tablón de anuncios, se han convertido en una herramienta emergente en detrimento de otros medios y en una alternativa viable para la exposición no sólo de la imagen de marca de una empresa, sino también de la de un candidato, de una causa social o de las continuas promociones de precios que ofrecen las grandes cadenas de tiendas por departamento o de especialización instaladas en las zonas de mayor crecimiento comercial y económico de la Capital.

Esta tesis hará las siguientes aportaciones, que ayudará a entender mucho mejor, el uso que le dan al panel publicitario:

- Ayudar a comprender los actos o actitudes de los anunciantes.

- Los anunciantes publican sin responsabilidad alguna.
- El panel publicitario muestra información en desorden y sin clasificar.

3.2. BASES TEÓRICAS

3.2.1. Empleo Informal

Empleo informal, es la actividad laboral de quienes trabajan y perciben ingresos, manteniéndose al margen del control tributario del Estado y de las disposiciones legales en materia laboral. El cuál se refiere a la actividad laboral de los trabajadores independientes, los vendedores ambulantes, las trabajadoras de servicio doméstico, entre otros.

El concepto de empleo informal fue planteado contemplando la posibilidad de que los países lo ajustaran a su propia situación y sus propias necesidades.

Sin embargo de acuerdo con KILM 8 de la International Labour Organization (2013), “La definición de la 15.a CIET se refiere al sector informal y al empleo en el mismo”. (p. 91)

3.2.1.1. Tipos de Empleos

- Los trabajadores que por cuenta propia (sin empleados) dueños de sus propias empresas (negocios) del sector informal.
- Los empleadores (trabajadores independientes con empleados) dueños de sus propias empresas (negocios) del sector informal.
- Trabajadores familiares auxiliares, independientemente de si trabajan en empresas del sector formal o informal.
- Miembros de cooperativas informales de productores (no constituidas como entidades jurídicas).
- Asalariados que poseen empleos informales.
- Trabajadores por cuenta propia que producen bienes exclusivamente para el propio uso final de su hogar.

3.2.2. Tablón de Anuncios

También conocido como Cartelera, es un tablero en que se fijan avisos, noticias, etc. o panel en el que se colocan anuncios públicos e informaciones. (Real Academia Española, 2017)

Es un lugar donde se pueden dejar mensajes públicos, publicar anuncios clasificados, donde vender o comprar algo de forma directa con el publicador, últimamente se hace referencia a un sitio web donde puedes pedir, vender o anunciar cualquier tipo de cosas. (Wikipedia, 2017)

3.2.3. Tipos de Aplicaciones según su desarrollo

3.2.3.1. Aplicaciones Nativas

Este tipo de aplicaciones que han sido creadas con el software que ofrece cada sistema operativo a los desarrolladores, denominado Software Development Kit o SDK. Así como Android, iOS y Windows Phone, los cuales son los más populares, cada uno tiene su propio Kit de Desarrollo y las aplicaciones nativas se programan y diseñan específicamente para cada plataforma. (Cuello y Vittone, 2013)

Entonces, una ventaja notable de una app nativa es que no requiere Internet para funcionar, por lo que ofrece una experiencia de uso más amigable y quedan integradas al móvil, permitiendo usar las características del hardware del dispositivo móvil, como la cámara, el acceso a internet y la ubicación.

Una aplicación o app nativa es un software diseñado para sacar el máximo provecho de las características de un dispositivo móvil; Estas aplicaciones nativas están desarrolladas en el lenguaje nativo de la propia terminal móvil: Objective-C par iOS, Java para Android o C# par Windows Phone. (Santiago, Trbaldo, Kamijo, y Fernández, 2015)

Por ello la necesidad de un desarrollo diferente para cada una de las plataformas hace necesario externalizar los proyectos a equipos especializados en cada sistema operativo.

3.2.3.2. Aplicaciones Web (Web apps)

Las aplicaciones que se basan en la web son aquellos recursos y herramientas que los usuarios pueden usar ingresando a un servidor web por internet o de una intranet por medio de un navegador. Estas Web Apps proporcionan un desarrollo multiplataforma basado en tecnologías Web como HTML, CSS, JavaScript, etc. Están desarrolladas en muchas ocasiones en servidores Web y son visibles en todo dispositivo que tenga un navegador y acceso a internet, es decir prácticamente cualquier celular o dispositivo móvil. Se desarrolla una sola vez y se ve en muchos sistemas distintos. (Santiago et al., 2015)

En este decir, las web apps no son más que una versión, que usa la tecnología responsive de la página web, optimizada para su correcta visualización en dispositivos móviles.

3.2.3.3. Aplicaciones Híbridas

Este tipo de aplicaciones mantienen la característica multiplataforma en gran parte, siguen haciendo uso de tecnologías Web, aunque funcionan localmente en el dispositivo, permitiendo ejecutarse también sin conexión a Internet. Al estar embebidas en un navegador web de una aplicación móvil nativa que tienen el mismo tipo de acceso a las APIs nativas de cada sistema operativo así como también a los recursos propios del sistema como el procesador, GPS, cámara, etc. El desarrollo de las aplicaciones móviles híbridas es un enfoque de programación, para dispositivos móviles que mezcla los puntos más resaltantes y fuertes de la programación nativa y el desarrollo de web apps. La forma de desarrollarlas es similar a la de una aplicación web y una vez que la aplicación está culminada, se compila y empaqueta de tal forma que resulta como si se tratase de una aplicación nativa, esta misma se puede compilar para diversas plataformas, convirtiéndose de esa manera en una aplicación multiplataforma que se puede distribuir en las AppStore. (Cuello y Vittone, 2013)

Una ventaja es que se puede acceder a las capacidades del teléfono, usando librerías, tal como lo haría una aplicación móvil nativa.

3.2.4. Tiempo Real (Real Time)

Se define de manera sintética la Real Time Web (Web en Tiempo Real) como un conjunto de tecnologías y prácticas que permiten a los usuarios la recepción de información tan rápido como esta es publicada por sus creadores, en vez de que estos o un programa (software) tenga que comprobar continuamente la fuente de la información en busca de actualizaciones. (Fundación Telefónica, 2011)

La novedad de la tecnología Real Time Web (RTW) o de la web en tiempo real es específicamente la posibilidad de acceder y recibir los datos o la información de todo tipo, de forma instantánea, datos e información que es proporcionada en la mayor parte de los casos por los propios usuarios y también la posibilidad de construir multitud de servicios sobre ella.

Entre las diferentes definiciones que le dan a este punto, rescato que real time web, es cuando la información se actualiza en menos de un segundo, se dice que “el tiempo real es el sueño de google”. Estamos comunicándonos constantemente en tiempo real, porque las herramientas de hoy nos permiten a todos emitir en tiempo real.

La parte web del “real time web” es hacia donde vamos, las cosas no las queremos en diferido (visualizar con retraso), las queremos en tiempo real.

3.2.5. Dispositivo Móvil

3.2.5.1. Definición

Es una máquina de tamaño pequeño que generalmente pueden ser llevados en el bolsillo, estos aparatos tienen capacidades de procesamiento para cumplir una función, acceso a internet y con memoria limitada, diseñadas específicamente para una función, pero que también pueden realizar otras más generales, por lo general se asocian al uso individual de una persona o usuario, tanto en posesión como en operación, el cual puede adaptarlos a su gusto. (Fernandez, 2006)

3.2.5.2. Características

Fernández (2006), indica que son características de los dispositivos móviles:

- Portabilidad, debido al reducido tamaño de los dispositivos móviles.
- Conectividad e inmediatez mediante las redes inalámbricas.
- Ubicuidad, debido a que se libera el aprendizaje de barreras espaciales o temporales.
- Adaptabilidad de aplicaciones y servicios a la necesidad del usuario.

3.2.5.3. Tipos de Dispositivos Móviles

Los tipos de dispositivos móviles son: teléfonos, Asistente digital personal (PDAs) y las consolas.

- **Teléfonos.** Su función principal es la de realizar llamadas, aunque ahora tienen funcionalidades de un ordenador.
- **PDAs.** Asistente personal digital que sirven como organizadores de agenda, calendario, gestión de contactos, reproductor de archivos multimedia y acceso a internet, actualmente la línea de diferencia que los separa de los teléfonos se está perdiendo.
- **Consolas.** Llamados comúnmente como “dispositivos orientados a jugar” por qué tienen gran capacidad de procesamiento de imágenes, también al igual que los PDAs reproducen archivos multimedia, cuentan con un navegador de internet y otros.

3.2.6. Base de Datos No Relacional

3.2.6.1. NoSQL

Las bases de datos NoSQL son bases de datos no relacionales, optimizadas para los modelos de datos sin esquema y sobre todo de desempeño escalable. También son muy conocidas por su baja latencia y facilidad de desarrollo. Hacen uso de una múltiple variedad de modelos de datos, como los almacenes de valor clave en memoria, de gráficos, de documentos y en columnas. (Amazon, 2018)

Los sistemas de bases de datos NoSQL crecieron con las principales redes sociales, como Google, Amazon, Twitter y Facebook. Estas tenían que

enfrentarse a desafíos con el tratamiento de datos que las tradicionales SGBDR (Sistema de Gestión de Bases de Datos Relacionales) no solucionaban. Con el crecimiento de la web en tiempo real existía una necesidad de proporcionar información procesada a partir de grandes volúmenes de datos que tenían unas estructuras horizontales más o menos similares. Estas compañías se dieron cuenta de que el rendimiento y sus propiedades de tiempo real eran más importantes que la coherencia, en la que las bases de datos relacionales tradicionales dedicaban una gran cantidad de tiempo de proceso.

Los sistemas de bases de datos NoSQL usan una variedad de modelos de administración de datos, como los almacenes de valor clave en memoria, los modelos de datos de gráficos y los almacenes de documentos. Estos tipos de bases de datos están optimizados para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles, lo que se logra mediante la flexibilización de algunas de las restricciones de coherencia de datos en las bases de datos relacionales tradicionales.

3.2.7. Modelo Cliente Servidor

El método habitual de organizar el software que se desea ejecutar en sistemas distribuidos consiste en separar las funciones en dos partes: clientes y servidores. Un cliente es un programa que utiliza los servicios que proporcionan otros programas. Los programas que proporcionan los servicios se llaman servidores. El cliente realiza una petición de un servicio, y un servidor lleva a cabo ese servicio. Un servidor sincroniza y gestiona el acceso al recurso, y responde a las peticiones de cliente con datos o información de estado. Los programas del cliente normalmente manejan las interacciones del usuario y con frecuencia solicitan datos o inician alguna modificación de datos en nombre de un usuario. (IBM, 2018)

Una generación más moderna de la arquitectura Cliente/Servidor añade una capa intermedia. En la arquitectura de tres capas (en general, en la arquitectura multicapa) el cliente implementa la lógica de presentación (cliente

fino), el servidor de la aplicación implementa la lógica de negocio y los datos residen en uno (o varios) servidor(es) de bases de datos.

Una arquitectura multicapa se define por tanto por las siguientes tres capas de componentes:

- Un componente front-end que es el responsable de proporcionar la lógica de presentación.
- Un componente back-end que proporciona acceso a servicios dedicados, tales como un servidor de bases de datos.
- Un componente que hace las funciones de capa intermediaria que permite a los usuarios compartir y controlar la lógica de negocio mediante su aislamiento de la aplicación real.

3.2.8. Test Driven Development (TDD)

TDD o Test-Driven Development (desarrollo guiado por pruebas) es una práctica de programación que consiste en escribir primero las pruebas (generalmente unitarias), después escribir el código fuente que pase la prueba satisfactoriamente y, por último, refactorizar el código escrito. Con esta práctica se consigue entre otras cosas: un código más robusto, más seguro, más mantenible y una mayor rapidez en el desarrollo. (Paradigma Digital, 2017)

TDD, es un proceso de desarrollo que consiste en codificar pruebas, desarrollar y refactorizar de forma continua el código construido. La idea principal de esta metodología es realizar de forma inicial las pruebas unitarias para el código que tenemos que implementar, es decir, primero codificamos la prueba y posteriormente se desarrolla la lógica de negocio.

El ciclo de vida de TDD se basa en una continua codificación y refactorización:

- 1) Elegir un requisito; se elige de una lista de requisitos.
- 2) Codificar la prueba; se comienza escribiendo una prueba para el requisito.

- 3) Verificar que la prueba falla; si la prueba no falla es porque el requerimiento ya estaba implementado o porque la prueba es errónea.
- 4) Codificar la implementación; escribir el código más sencillo posible que haga que la prueba funcione.
- 5) Ejecutar las pruebas automatizadas; verificar si todo el conjunto de pruebas funciona correctamente.
- 6) Refactorización; se utilizará principalmente para eliminar código duplicado y eliminar dependencias innecesarias.
- 7) Actualización de la lista de requisitos; se actualiza la lista de requisitos tachando el requisito implementado.

La programación TDD es una técnica de diseño e implementación de software, esta técnica nos permite obtener una cobertura de pruebas muy alta.

Para el programador es un cambio grande en su mentalidad, en su forma de procesar y gestionar la información. Cuesta acostumbrarse al principio, pero llega un momento en que su productividad y eficiencia a la hora de codificar los test y de desarrollar de forma simplificada el código se incrementa y resulta muy productivo.

3.2.9. Progressive Web Apps

Traduciendo en Español, significa, **Aplicaciones Web Progresivas**, es un término que se da a una nueva generación de aplicaciones que incrementan su funcionalidad, conforme las capacidades del dispositivo en el que se ejecutan, incrementan, de ahí la palabra progresiva. La siguiente parte del nombre web, hace referencia a que se construyen utilizando estándares de desarrollo web, algunos ya conocidos como HTML, CSS y JavaScript; y una nueva generación de APIs de JavaScript. La parte final app es porque las Progressive Web Apps se comportan como aplicaciones web nativas, pero usan tecnologías web.

Google (2017), las aplicaciones web progresivas es una nueva tendencia promovida por desarrolladores de Google, que básicamente son experiencias de usuario que tienen el alcance de la web y son:

- **Confiable:** Carga instantáneamente y nunca muestra el dinosaurio de Chrome, incluso en condiciones de red inciertas.
- **Rápido:** Responde rápidamente a las interacciones de los usuarios con animaciones suaves.
- **Atractivo:** Se siente como una aplicación natural en el dispositivo, con una experiencia de usuario de calidad.

Una aplicación web progresiva permite a los desarrolladores crear un sitio web móvil que funcione súper rápido y se comporte como una aplicación nativa, es una mejor manera de permitir que un sitio web funcione más como una aplicación nativa instalada en el dispositivo. (Mejia, Muñoz, Rocha, San Feliu, y Peña, 2016)

3.2.10. IONIC

Ionic es un framework gratuito y open source para desarrollar aplicaciones híbridas multiplataforma que utiliza HTML5, JavaScript, CSS (generado por SASS) y Cordova como base. Es uno de los framework del momento por utilizar AngularJS y TypeScript para gestionar las aplicaciones, lo que asegura aplicaciones rápidas y escalables. Es la plataforma de tecnología de desarrollo móvil multiplataforma más popular del mundo, impulsando el rápido crecimiento de las startups de algunas de las mayores compañías del mundo. Fue fundada en el 2012, cuando el uso de tecnologías web para construir aplicaciones nativas recién empezaba a surgir. (Ionic Framework, 2017)

Ionic en su página oficial dice que su misión es proporcionar herramientas y servicios de código abierto que permitan a las personas de todo el mundo participar en la economía de las aplicaciones móviles de rápido crecimiento. Ionic es el framework, libre y SDK móvil de código abierto para el desarrollo nativo y aplicaciones web progresivas con facilidad.

3.2.11. Firebase

Es todo un entorno enfocado al desarrollo de aplicaciones móviles, desarrollado por Google. Cuenta con distintas funciones como, analíticas, monetización, base de datos (no relacional - NoSQL) en tiempo real y notificaciones tipo push. Es un servicio capaz de proveernos de un Backend en la nube con una fuente de datos en tiempo real y librerías para acceder a ella desde aplicaciones web, iOS y Android. (Google Developers, 2017)

Básicamente es una base de datos remota, alojada en la nube y capaz de ser accedida desde navegadores y apps para dispositivos, que tiene como principal característica que responde en tiempo real a los cambios realizados en los datos. En la práctica, mediante Firebase podemos escribir datos en una base de datos y que éstos datos se comuniquen a todos los clientes conectados a esa misma fuente de datos.

3.2.12. Metodología Mobile-D

Mobile-D es un modelo ágil para el desarrollo de aplicaciones móviles la cual ha sido desarrollada por tres compañías de software móvil ha sido exitosamente evaluado en el nivel 2 de certificación CMMI.

Mobile-D es una creación un tanto antigua, ya que se desarrolló como parte de un proyecto finlandés, ICAROS, en el 2004, fue creado mediante un proyecto de cooperación muy estrecha con la industria. El grueso del trabajo fue realizado por los investigadores del VTT. Aun así la metodología de diseño se elaboró con una participación importante de las empresas de TI finlandesas. Tal como se puede ver en los experimentos que se han documentado esto consiguió que la investigación llevada a cabo no se alejara demasiado de las reglas de desarrollo de las aplicaciones comerciales.

Mobile-D es una mezcla de muchas técnicas, pero al mismo tiempo, consiguieron crear una contribución original para el nuevo escenario del desarrollo de aplicaciones para sistemas móviles. (Blanco, Camarero, Fumero, y Adam Werteski, 2009)

Mobile-D fue el primer intento de incorporar prácticas ágiles para el desarrollo de aplicaciones móviles. Mobile-D fue introducido en 2004 por Abrahamsson como metodología de desarrollo inspirado en Extreme Programming, Metodologías Crystal y Rational Unified Process (RUP). (Mechato y Gastañadui, 2014)

Consta de cinco fases:

- **Exploración:** En esta fase se lleva a cabo el proceso inicial de planificación, así como también, se establece los conceptos básicos del proyecto e identificar el objetivo general, tipo de aplicación y las plataformas para el cual se desarrollará el proyecto.
- **Iniciación:** Se prepara el proyecto para evitar todos los posibles problemas que puedan surgir durante el desarrollo de la aplicación, preparando todos los recursos físicos y tecnológicos para las actividades de producción, definiendo las herramientas de desarrollo a usarse, así mismo las herramientas de pruebas.
- **Producción:** Se lleva a cabo toda la implementación de la aplicación móvil usando un ciclo de desarrollo iterativo e incremental, en el cual se usa el desarrollo dirigido por pruebas (TDD, Test-Driven Development) y para ello, se tiene como regla que, antes de iniciar el desarrollo de una funcionalidad deberá existir una prueba que verifique el correcto funcionamiento.
- **Estabilización:** Esta fase se lleva a cabo cada vez que se complete una nueva iteración en la fase anterior; Procediendo a integrar las funcionalidades que se implementaron en diferentes módulos o sub-sistemas y cuando presentan error se realizan las debidas correcciones.
- **Prueba del Sistema:** Cuando se concluye totalmente el desarrollo se procede a la fase de pruebas, en el cual se deberá iterar hasta llegar a una versión estable de acuerdo a los requerimientos establecidos.

4. INGENIERÍA DEL PROYECTO

La ingeniería del proyecto para el desarrollo de la aplicación multiplataforma, para la publicación de anuncios de oferta y demanda laboral en tiempo real, consta de la gestión del proyecto, la metodología de desarrollo, las herramientas usadas tanto software como hardware y la mejora del modelo cliente – servidor, que es tan usado, gracias a la optimización de las funcionalidades en el código, las cuales se detallarán en profundidad a continuación:

4.1. ADMINISTRACIÓN Y GESTIÓN DEL PROYECTO BASADO EN PMBOK

Se hará uso del PMBOK (5ta. Edición), ya que la sexta edición fue publicada el 6 de septiembre del 2017, el cual el área de conocimiento **Gestión del Tiempo del Proyecto** cambia su denominación a **Gestión del Cronograma del Proyecto** y **Gestión de los Recursos Humanos** cambia a **Gestión de los Recursos del Proyecto**; y aún no está disponible la versión en español. Entonces se contará con 10 áreas de conocimiento, con las cuales se lleva a cabo el presente proyecto:

4.1.1. Plan de Gestión de la Integración

Tabla 1. Plan de gestión de la integración

PROYECTO	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real.
PATROCINADO, PREPARADO Y REVISADO POR	Ricardo Richard Huamantingo Navarro
REVISIÓN (Correlativo)	DESCRIPCIÓN (REALIZADA POR) (Motivo de la revisión y entre paréntesis quien la realizó)
01	Elaboración de Acta de Constitución (Ricardo Richard Huamantingo Navarro)
BREVE DESCRIPCIÓN DEL PROYECTO O SERVICIO DEL PROYECTO (Características, funcionalidades, soporte entre otros)	

En este proyecto se tiene como objetivo desarrollar una aplicación móvil multiplataforma, con interacción en tiempo real, con el uso de las herramientas Ionic y Firebase los cuales permitirán conseguir una aplicación híbrida muy funcional con un modelo cliente servidor más corto, sin dejar de lado las funcionalidades y ventajas del estilo FrontEnd y BackEnd, así mismo optimizar algunos métodos para mejorar el rendimiento de la aplicación y reducir el consumo de datos.

ALINEAMIENTO DEL PROYECTO

1. OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN (A qué objetivo estratégico se alinea el proyecto)	2. PROPÓSITO DEL PROYECTO (Beneficios que tendrá la organización una vez que el producto del proyecto esté operativo o sea entregado)
1.1. Desarrollar aplicaciones híbridas multiplataforma de alto desempeño con bajo costo en desarrollo y producción. 1.2. Desarrollar métodos optimizados para mejorar el rendimiento de las aplicaciones.	2.1. Impulsar la creación de aplicaciones híbridas multiplataforma de alto desempeño. 2.2. Fomentar la investigación del desarrollo de aplicaciones híbridas multiplataforma y aplicarlos en la mejora de los frameworks.
3. OBJETIVOS DEL PROYECTO (Principalmente en términos de costo, tiempo, alcance, calidad)	
3.1. Desarrollar una aplicación móvil multiplataforma que reemplace el uso del tablón de anuncios. 3.2. Lograr que la aplicación tenga un uso intuitivo con altas prestaciones funcionales. 3.3. Culminar el proyecto en un plazo de 8 meses con inicio de la fecha del Acta de Constitución del Proyecto.	
4. FACTORES CRÍTICOS DE ÉXITO DEL PROYECTO (Componentes o características que deben cumplirse en el proyecto para considerarlo exitoso)	
4.1. Adquisición de equipos y software dentro de las fechas establecidas y con las características especificadas. 4.2. Diseño de la aplicación con mejoras en la obtención de información en comparación con el tablón de anuncios. 4.3. Notificaciones a los usuarios de los eventos más importantes que se dan en la aplicación en Tiempo Real.	
5. REQUERIMIENTOS DE ALTO NIVEL (Principales condiciones y/o capacidades que debe cumplir el producto o servicio y la Gestión del Proyecto)	

<p>5.1. Exploración de las herramientas para el proyecto.</p> <p>5.2. Diseñar e integrar los módulos y servicios de la aplicación.</p> <p>5.3. La aplicación móvil se construirá siguiendo la metodología Mobile-D.</p>	
EXTENSIÓN Y ALCANCE DEL PROYECTO	
<p>6. FASES DEL PROYECTO</p> <p>(Agrupamiento lógico de actividades relacionadas que usualmente culminan elaborando un entregable principal. Cada Fase se ejecutará como un proyecto. Al fin de fase se puede tomar la decisión de continuar o no con las siguientes fases)</p>	<p>7. PRINCIPALES ENTREGABLES</p> <p>(Un único y verificable producto, resultado o capacidad de realizar un servicio que debe ser elaborado para completar un proceso, una fase o un proyecto)</p>
Fase I – Gestión del Proyecto	Acta de Constitución del Proyecto.
Fase II - Diseño	Proceso de definición de las interfaces de la aplicación móvil.
Fase III – Requerimiento de Adquisiciones	<p>Adquisición de laptop y Smartphones.</p> <p>Proceso de adquisición del entorno del desarrollador (Ionic 3, Firebase Console, Visual Studio Code, entre otros).</p>
Fase IV – Implementación	Implementar los módulos de la aplicación (Pages, Providers).
Fase V – Arranque del Sistema	Pruebas a la aplicación móvil.
Fase VI – Término de Obra	Aplicación móvil multiplataforma.
8. INTERESADOS CLAVE	
<p>(Persona u organización que está activamente involucrado en el proyecto o cuyos intereses pueden ser afectados positiva o negativamente por la ejecución del proyecto o por el producto que elabora)</p>	
<p>8.1. Personas desempleadas en busca de trabajo.</p> <p>8.2. Desarrolladores de la comunidad de Ionic y Firebase.</p> <p>8.3. Programadores en General de aplicaciones híbridas o multiplataforma con Ionic.</p>	
9. RIESGOS	
<p>(Evento o condición incierta que, si ocurriese, tiene un efecto positivo o negativo sobre los objetivos del proyecto)</p>	
<p>9.1. Retraso y demora en adquirir las herramientas de desarrollo.</p>	

<p>9.2. Cambio de políticas y licencias de uso en los frameworks, que puedan anular el uso.</p> <p>9.3. Restricciones de uso de la plataforma Firebase.</p> <p>9.4. No contar con los recursos económicos suficientes.</p>
<p>10. HITOS PRINCIPALES DEL PROYECTO (Un evento significativo para el proyecto)</p>
<p>10.1. Buena Pro y aprobación del acta de constitución del proyecto.</p> <p>10.2. Lograr implementar e integrar en la aplicación móvil todas las funciones requeridas para el óptimo desempeño de la aplicación.</p> <p>10.3. Correcciones y pruebas en la aplicación móvil desarrollada.</p>
<p>11. PRESUPUESTO DEL PROYECTO (La estimación aprobada para el proyecto o cualquier otro componente de la estructura de desglose de trabajo, u otra actividad del cronograma)</p>
<p>El costo total del proyecto lo asumirá el Jefe del Proyecto.</p>

Fuente: Elaboración Propia

4.1.2. Plan de Gestión del Alcance

Tabla 2. Plan de Gestión del alcance

NOMBRE DEL PROYECTO:	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real
PREPARADO:	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto
1. Describir cómo será administrado el alcance del Proyecto:	
<p>Primero se estableció, a quienes estará dirigido el proyecto, de ahí, se determinó que el alcance del proyecto, es que la aplicación móvil desarrollada será usada por las personas desempleadas en busca de trabajo y que los módulos optimizados serán usados por los desarrolladores que usen los mismos frameworks que se usan en este proyecto.</p>	
2. Evaluar la estabilidad del alcance del proyecto (Cómo manejar los cambios, la frecuencia e impacto de los mismos):	
<p>El Jefe del Proyecto cuantifica el impacto y proporciona alternativas de solución. Los cambios del proyecto deben ser evaluados y aprobados.</p> <p>Los cambios solicitados en el proyecto serán aprobados y revisados en las reuniones de cada semana, se indica el estado en la reunión siguiente a la que se solicitó.</p>	
3. ¿Cómo los cambios al alcance, serán identificados y clasificados?:	

La revisión de la solicitud de cambios del alcance estará a cargo del Jefe del Proyecto, quien hará una evaluación del mismo, de la misma manera si se necesita información adicional se hará respectivamente una petición al solicitante.
4. Describir cómo los cambios del alcance serán integrados al proyecto:
El Jefe del Proyecto aprobará el impacto del cambio que no modifica la línea base del proyecto y se actualizarán las líneas base del proyecto.
5. Comentarios adicionales:
Se necesita integrar adecuadamente la conexión de la aplicación con Firebase.

Fuente: Elaboración propia

4.1.3. Plan de Gestión del Tiempo

Tabla 3. Plan de Gestión del Tiempo

NOMBRE DEL PROYECTO:	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real	
PREPARADO POR:	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto	
Persona(s) autorizada(s) a solicitar cambio en cronograma:		
Nombre	Cargo	Ubicación
Ricardo Richard Huamantingo Navarro	Jefe del Proyecto	Talavera
Persona(s) que aprueba(n) el requerimiento de cambio de cronograma:		
Nombre	Cargo	Ubicación
Ricardo Richard Huamantingo Navarro	Jefe del Proyecto	Talavera
Razones aceptables para cambios en cronograma del Proyecto: (por ejemplo, retrasos debido a entrega de materiales o disponibilidad de personal, clima, adelantar el cumplimiento debido a término de fase o proceso, etc.)		
<ul style="list-style-type: none"> - Accidentes de trabajo. - Mal establecimiento de la secuencia de actividades. - Fallos de las herramientas de trabajo (Laptop, Smartphone y otros accesorios). - Fallos o averías del servicio de internet. 		

Describir cómo calcular y reportar el impacto en el proyecto por el cambio en cronograma: (Tiempo, costo, calidad, etc.)
<ul style="list-style-type: none"> - Se indica en el informe al responsable del mismo y la fecha de ocurrencia de dicho problema. - Se describe el problema indicando el grado de urgencia. - Impacto del mismo sobre el proyecto (Tiempo, Alcance, Calidad y Costo). - Se describe las alternativas de solución con detalle del impacto en las diversas áreas (Tiempo, Alcance, Calidad y Costo).
Describir cómo los cambios al cronograma serán administrados:
<p>El Alcance se llevará de la siguiente manera:</p> <ul style="list-style-type: none"> - Determinación de Responsabilidades: <ul style="list-style-type: none"> • Planificación (Jefe de Proyecto). • Seguimiento y Control (Jefe de Proyecto). • Ejecución (Jefe de Proyecto).

Fuente: Elaboración Propia

4.1.4. Plan de Gestión de los Costos

Tabla 4. Plan de Gestión de los Costos

PROYECTO	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real	
PREPARADO, REVISADO Y APROBADO POR	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto	
Persona(s) autorizada(s) a solicitar cambios en el costo:		
Nombre	Cargo	Ubicación
Ricardo Richard Huamantingo Navarro	Jefe del Proyecto	Talavera
Persona(s) que Aprueba(n) requerimientos de cambios en costo contractual:		
Nombre	Cargo	Ubicación
Ricardo Richard Huamantingo Navarro	Jefe del Proyecto	Talavera
Persona(s) que aprueba(n) el requerimiento de cambio de costo interno ofrecido:		
<ul style="list-style-type: none"> • Jefe del Proyecto 		

<p>Razones aceptables para cambios en el Costo del Proyecto: (por ejemplo: Aprobación de cambios en el alcance, incremento de costos en los materiales, etc.)</p> <ul style="list-style-type: none"> - Mejoramiento en el Alcance del Proyecto. - Ampliación de plazos en el Alcance del Proyecto.
<p>Describir cómo calcular e informar el impacto en el proyecto por el cambio en el costo: (Tiempo, calidad, etc.)</p> <ul style="list-style-type: none"> - Usuario que solicita un cambio. - Describir las características de la situación que requiere la solicitud de cambio de los costos. - Describir las alternativas de solución detallando el impacto en las diversas áreas (Tiempo, Alcance, Calidad y Costo). - Selección de las alternativas de solución (Propuesta).

Fuente: Elaboración Propia

4.1.5. Plan de Gestión de la Calidad

Tabla 5. Plan de Gestión de la Calidad

PROYECTO	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real
PREPARADO, REVISADO, APROBADO POR	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto
<p>GESTIÓN DE CALIDAD DEL PROYECTO</p> <p>(Descripción de cómo se van a aplicar los procesos de gestión de calidad del proyecto. Herramientas a emplear, normativas, reglamentos, responsables, áreas de aplicación, etc.)</p>	
<p>PLANIFICAR LA CALIDAD:</p> <p>(Explicar y sustentar cómo se ha elaborado el presente plan de gestión de calidad)</p>	
<p>En el plan de gestión de calidad se describe que el equipo de dirección, implementará las políticas de calidad de la organización que ejecuta. Es un plan subsidiario o un componente del plan para la dirección del proyecto. Dicho plan de calidad proporciona las entradas al plan general para la dirección del proyecto e integra el control de calidad, el aseguramiento de la calidad y los métodos que mejoran los procesos del proyecto.</p> <p>Para ello se aplicará, la calidad de software (SQA, Software Quality Assurance), el cual es un conjunto de actividades previamente planificadas y sistemáticas necesarias para la contribución a la confianza en que el</p>	

software satisfaga los requerimientos de calidad, el cual es aplicado a lo largo de todo el proceso del software.

REALIZAR ASEGURAMIENTO DE CALIDAD

(Explicar cómo se va a realizar el proceso de aseguramiento de calidad)

Aseguramiento de Calidad: Es responsabilidad del Jefe de Proyecto la ejecución del Aseguramiento de Calidad durante todo el Proyecto, supervisa el planeamiento de los procesos del proyecto contra lo que sea ejecutado, y plantea acciones correctivas o preventivas según sean necesarias. Y se informa semanalmente a todo el Equipo del Proyecto.

REALIZAR CONTROL DE CALIDAD

(Explicar cómo se va a realizar el proceso de control de calidad)

Control de Calidad: El Jefe de Proyecto se encargará de la ejecución del Control de Calidad, se revisa los entregables del proyecto, según se vayan presentando, se emiten las conformidades u observaciones en la reunión semanal. Se definen dos procedimientos para el Control de Calidad:

- Revisión de Forma y
- Revisión de Contenidos

Control de Calidad del Contenido: Se revisa la calidad de los entregables del proyecto con el equipo de expertos.

Control de Calidad de Redacción y Formato: Se revisa la redacción de los documentos y las observaciones que se puedan realizar.

Fuente: Elaboración Propia

4.1.6. Plan de Gestión de los Recursos Humanos

Tabla 6. Plan de Gestión de los Recursos Humanos

PROYECTO	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real
PREPARADO, REVISADO Y APROBADO POR	Ricardo Richard Huamantingo Navarro
ROL EN EL PROYECTO (Denominación del puesto)	Jefe del Proyecto
FUNCION / RESPONSABILIDAD PRINCIPAL (Descripción de los que se espera de este rol)	

<ul style="list-style-type: none"> - La ejecución y entrega del proyecto, está en la responsabilidad del Jefe del Proyecto. - Responsable de la designación del personal que integra el equipo del proyecto y de su organización. - Controlar la información de costos y gastos del proyecto. 				
COMPETENCIAS REQUERIDAS / RESPONSABILIDADES				
(Conocimientos, habilidades, actitudes)				
DISPONIBILIDAD				
(Posibilidad de contar con el recurso humano en el tiempo requerido)				
Descripción del puesto a requerir	Externo Interno	Fecha Ingreso	Fecha Retiro	Tiempo Requerido
Jefe de Proyecto	Interno	01/11/2017	01/05/2018	6 meses
INCORPORACIÓN AL PROYECTO				
(Cómo se incorpora el personal al proyecto)				
Evento disparador (Hecho que identifica la necesidad de hacer gestiones sobre el personal)		Sincronización prevista (Acciones requeridas para gestionar al persona)		
Renuncia		Al instante con la designación de reemplazo por el Jefe del Proyecto.		
Enfermedad		Al instante con la designación de reemplazo por el Jefe del Proyecto.		

Fuente: Elaboración Propia

4.1.7. Plan de Gestión de las Comunicaciones

Tabla 7. Plan de Gestión de las Comunicaciones

PROYECTO	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real
PREPARADO, REVISADO Y APROBADO POR	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto
GESTIÓN DE COMUNICACIONES DEL PROYECTO	
(Descripción de cómo se van a aplicar los procesos de gestión de comunicaciones del proyecto, herramientas a emplear, responsables, áreas de aplicación, etc.)	

El responsable es el Jefe del Proyecto, estará a cargo de la integración de los plugins, providers y de la base de datos en tiempo real.

PLANIFICACIÓN DE LA COMUNICACIÓN

(Explicar y sustentar cómo se ha elaborado el presente plan de gestión de comunicación)

Determinar la información, necesidad y comunicación de los interesados: qué información necesita, quién necesita, para cuando la necesita, cómo le será suministrada y por quién.

El Jefe del Proyecto debe prever la cantidad y calidad de canales requeridos desde el principio con el fin de que la información fluya y llegue a todo el personal involucrado.

Los requisitos para la comunicación incluyen:

- Relaciones de responsabilidad de la organización.
- Organigramas

Fuente: Elaboración Propia

4.1.8. Plan de Gestión de Riesgos

Tabla 8. Plan de Gestión de Riesgos

NOMBRE DEL PROYECTO	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real
PREPARADO POR	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto
Descripción de la metodología de gestión del riesgo a ser usada:	
Alcances	
<ul style="list-style-type: none"> - La priorización, seguimiento e identificación de los riesgos críticos será realizado por el Jefe de Proyecto. 	
Herramientas	
<ul style="list-style-type: none"> - Lluvia de Ideas. 	
Fuentes de datos	
<ul style="list-style-type: none"> - La determinación de los riesgos será por parte de los integrantes del proyecto según juicio y experiencia de especialistas. 	
Roles y Responsabilidades:	
<ul style="list-style-type: none"> - Jefe de Proyecto: Encargado de identificar, priorizar y hacer el seguimiento de los riesgos, quien propone acciones para resolver los riesgos identificados. 	

- | |
|--|
| <ul style="list-style-type: none">- Analista de Sistemas: Encargado de la determinación de la funcionalidad de la aplicación móvil.- Programador: Encargado de toda la integración y programación de la aplicación móvil. |
|--|

Fuente: Elaboración Propia

4.1.9. Plan de Gestión de las Adquisiciones

Recursos para la adquisición

(Todo personal involucrado en la gestión de las adquisiciones)

En este proyecto los encargados de realizar las labores de contratación y compra son:

- El Jefe del Proyecto en la parte de la adquisición de herramientas de desarrollo y equipos físicos.
- El Jefe del Proyecto en el lado de contrataciones (programador, analista).

En este proyecto el Jefe de Proyecto, el analista, el programador y el diseñador de las interfaces caerán bajo responsabilidad de una sola persona.

Productos y servicios contratados

(Describir los productos o servicios que el proyecto requerirá contratar indicando las restricciones, supuestos y límites que tendrán)

- Servicio del proveedor para la compra de los equipos físicos.
- Servicio de desarrollador para la aplicación móvil.

Procedimientos para la gestión de adquisiciones

(Describir los procedimientos que se usarán en el proyecto y que procedimientos usó el comprador).

Tabla 9. Plan de Gestión de las Adquisiciones

N°	PROCEDIMIENTO EN EL PROYECTO PARA SOLICITUD DE COMPRA	PROCEDIMIENTO EN EL ÁREA LOGÍSTICA PARA REALIZAR ORDEN DE COMPRA
1	En este documento se describe el producto que se pide, la cantidad en unidades a pedir. Una vez evaluado, el comprador de la empresa a los proveedores, se regresa una plantilla para la toma de la decisión económica y técnica del proveedor que mejor se ajuste a los requerimientos del proyecto.	Se elabora la orden de compra. En este documento se debe describir correctamente el nombre completo del proveedor, la fecha, la persona de contacto si es posible, el número de teléfono, la descripción del producto a pedir, la cantidad en unidades, el precio y el monto total.

Fuente: Elaboración Propia

4.1.10. Plan de Gestión de los Interesados

Tabla 10. Plan de Gestión de los Interesados

PROYECTO	Aplicación Móvil Multiplataforma, para la Publicación de Anuncios de Oferta y Demanda Laboral en Tiempo Real
PREPARADO POR	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto
REVISADO POR	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto
APROBADO POR	Ricardo Richard Huamantingo Navarro – Jefe del Proyecto

Nombres y Apellidos	Organización	Cargo	Información de contacto	Requerimientos sobre el Producto	Influencia sobre	Tipo de interés
Ricardo Richard Huamantlingo Navarro	UNAJM A	Jefe del Proyecto	rrichardh95@gmail.com	Responsable de brindar la buena disponibilidad y viabilidad de recursos necesarios en el proyecto.	Presente en todas las fases	Control del Proceso
Ricardo Richard Huamantlingo Navarro	UNAJM A	Analista de Sistemas	rrichardh95@gmail.com	Encargado del diseño de la infraestructura de la aplicación, considerando los requisitos funcionales, el alcance y las limitaciones que la aplicación tiene.	Fase de planificación, ejecución y cierre	Elaborar la infraestructura de la aplicación y dar solución a los problemas que se presentan.
Ricardo Richard Huamantlingo Navarro	UNAJM A	Programador de Sistemas	rrichardh95@gmail.com	Encargado de la implementación, integración de los módulos y pruebas de la aplicación, que es diseñada por el analista de sistemas.	Fase de planificación, ejecución y cierre	Desarrollar la aplicación móvil en el cronograma que se estableció.

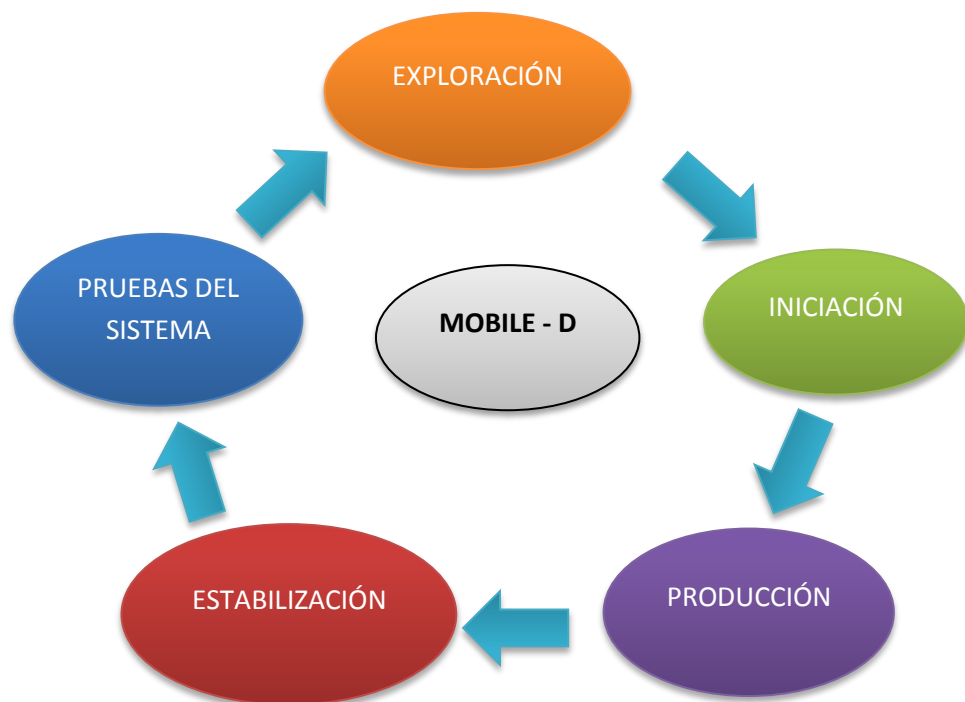
Fuente: Elaboración Propia

4.2. METODOLOGÍA DE DESARROLLO MOBILE-D

Mobile-D es la metodología que se usó para el desarrollo del proyecto en cuestión, el cual consiste de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema.

En la figura se observa el ciclo de vida del software desarrollado con la metodología Mobile-D:

Figura 1. Ciclo de vida del Software con Mobile-D



Fuente: Elaboración Propia

4.2.1. Fase de Exploración

Se busca determinar el plan del proyecto, la identificación de los usuarios e interesados del proyecto. También, se identifica con claridad cuál será el nombre de la aplicación, el objetivo general y la plataforma para el cual se desarrollará el proyecto.

4.2.1.1. Estableciendo actores

Se identifica a los actores y al equipo de desarrollo para este proyecto, es importante, se lista la relación de los actores que participarán de forma activa durante la ejecución del proyecto y el desarrollo de la aplicación móvil:

- **Jefe del Proyecto:** Responsable de determinar la buena disponibilidad y viabilidad de los recursos necesarios en el proyecto.
- **Analista de Sistemas:** Responsable del diseño de la infraestructura de la aplicación móvil, considerando los requerimientos funcionales, el alcance y las limitaciones que la aplicación tiene.
- **Programador:** Responsable de implementar la aplicación con la infraestructura diseñada por el analista de sistemas.

En este proyecto de tesis, fue desarrollado por solo un responsable (tesista), en el plazo establecido, cumpliendo con las responsabilidades mencionadas.

4.2.1.2. Estableciendo Usuarios

Se identifica a los usuarios, que son las personas que usarán la aplicación móvil, se define a continuación:

- **Usuarios Directos:** Personas desempleadas en busca de trabajo.
- **Usuarios Indirectos:** Desarrolladores que usen los frameworks, que son usados en este proyecto.

4.2.1.3. Información clave del Proyecto

Se definió el nombre comercial de la aplicación, el objetivo general y las plataformas para las cuales se desarrollará el proyecto.

Tabla 11. Información clave del proyecto

NOMBRE DE LA APLICACIÓN	WorkApp (El cual nos indica que es una aplicación que está relacionada al trabajo)
OBJETIVO GENERAL	Desarrollar una aplicación, que permita hacer anuncios en Tiempo Real.
PLATAFORMA	El proyecto estará disponible para Android, IOS y Windows.

	Pero la aplicación final con la que se harán pruebas será Android.
--	--

Fuente: Elaboración Propia

4.2.1.4. Principales Requerimientos Funcionales

- La publicación de anuncios en la aplicación solo será accesible para usuarios registrados.
- La aplicación enviará notificaciones cuando se cree un nuevo anuncio.
- La aplicación controlará el acceso y permitirá solamente a usuarios autenticados. Los usuarios deben ingresar a la aplicación con su correo y contraseña.
- La aplicación podrá ser utilizado en las plataformas Android, IOS, Windows.
- La estructura de la aplicación móvil debe ser escalable y desarrollarse para actualizaciones futuras.

4.2.1.5. Principales Requerimientos No Funcionales

- Los eventos conectados en tiempo real (sockets) deberán actualizar la información al instante.
- La aplicación deberá ser intuitiva, de fácil uso para el usuario.
- La aplicación debe mostrar los mensajes de error, de forma informativa y orientada a usuarios finales.
- La aplicación móvil debe tener un diseño “Responsive” para garantizar la visualización en diversos dispositivos móviles.
- El aplicativo debe poseer interfaces gráficas bien estructuradas.
- La aplicación no deberá ocupar más de 30 Mb de espacio en el dispositivo.
- El framework que se usará será IONIC 3 junto a firebase y plugins de Cordova.

4.2.2. Fase de Iniciación

Se define las herramientas (hardware para pruebas y software de desarrollo), los diagramas de los casos de uso. En seguida, se detalla las diferentes herramientas y recursos que se usó para el desarrollo de la aplicación móvil:

4.2.2.1. Stack de Desarrollo

- IONIC 3
- Apache Cordova
- AngularJS
- Firebase
- TypeScript v2.3.4
- Visual Studio Code
- Terminal Linux
- Seven Square

4.2.2.2. Hardware de Desarrollo

- 1 Laptop Toshiba Core i7, Sistema Operativo (Linux) Debian 8 (Jezzy), Procesador 2.60 GHz

4.2.2.3. Herramientas para pruebas (Hardware)

- 1 Smartphone con sistema operativo Android 5.0
- 1 Smartphone con sistema operativo Android 7.0
- 1 laptop Toshiba Core i7, sistema operativo (Linux) Debian 8

4.2.2.4. Diagramas de casos de uso de la aplicación móvil

- **Vista de Carga:** En esta vista se muestra un SplashScreen con el logo de Ionic, aquí no se realiza ninguna acción y de manera automática da paso a la vista de logeo.

- **Primera Vista – Login:** La vista de Login o Login Page es la primera de la aplicación, en el cual está presente un formulario de acceso a la aplicación, tiene las acciones de login, registrarse, recuperar contraseña y ver los anuncios publicados.

Tabla 12. Caso de uso: Iniciar Aplicación - Vista Login

CASO DE USO:	Iniciar Aplicación – Vista Login
ACTORES:	Usuario
PROPÓSITO:	Dar acceso a la vista principal de la aplicación móvil.
RESUMEN:	El usuario debe ingresar su correo y contraseña para autenticarse, de esa manera podrá ingresar a la vista de anuncios.
TIPO:	Primario
PRE-CONDICIONES:	Debe de haber finalizado la carga de la aplicación (SplashScreen).

Fuente: Elaboración Propia

Tabla 13. Caso de uso: Registrar Usuario

CASO DE USO:	Registrar Usuario
ACTORES:	Usuario
PROPÓSITO:	Registrar datos del usuario, para crear su cuenta y usar la aplicación.
RESUMEN:	La persona que desee hacer uso, tendrá que registrar sus datos verdaderos.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción “REGÍSTRATE” en la vista Login.

Fuente: Elaboración Propia

Tabla 14. Caso de uso: Configuración de Avatar de Perfil

CASO DE USO:	Configuración de Avatar de Perfil
ACTORES:	Usuario
PROPÓSITO:	Seleccionar una imagen para el avatar del perfil.

RESUMEN:	El usuario seleccionará una imagen de su galería o podrá tomarse una foto para ingresarla como avatar de su perfil.
TIPO:	Primario
PRE-CONDICIONES:	Caso de uso inmediato, después de registrar sus datos.

Fuente: Elaboración Propia

Tabla 15. Caso de uso: Anuncios sin Login

CASO DE USO:	Anuncios sin Login
ACTORES:	Usuario
PROPÓSITO:	Visualizar los anuncios publicados, pero sin acceder a ninguna opción.
RESUMEN:	La persona podrá visualizar los anuncios publicados, y si desea acceder a cualquiera de ellos, tendrá que registrarse.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción “¿QUIERES VER LOS ANUNCIOS?” en la vista Login.

Fuente: Elaboración Propia

- **Vista Principal:** Se visualiza los anuncios que fueron publicados en orden de fecha de publicación del más reciente al más antiguo, cada uno con sus respectivas acciones de ver en detalle, disponibilidad y ubicación; estará disponible un botón para publicar nuevos anuncios.

Tabla 16. Caso de uso: Vista Principal

CASO DE USO:	Vista Principal
ACTORES:	Usuario
PROPÓSITO:	Dar paso a la visualización de los anuncios disponibles.
RESUMEN:	El usuario podrá visualizar los anuncios y su disponibilidad, seleccionar cualquier anuncio y ver de forma detallada.
TIPO:	Primario
PRE-CONDICIONES:	Caso de uso inmediato después de haberse autenticado.

Fuente: Elaboración Propia

Tabla 17. Caso de uso: Seleccionar anuncio

CASO DE USO:	Seleccionar Anuncio
ACTORES:	Usuario
PROPÓSITO:	Ver un anuncio a detalle.
RESUMEN:	El usuario podrá ver el anuncio con toda la información disponible, de la misma manera tendrá la oportunidad de postularse a dicha oferta laboral, enviando una solicitud de postulación.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el anuncio deseado.

Fuente: Elaboración Propia

Tabla 18. Caso de uso: Mapa del Anuncio

CASO DE USO:	Mapa del Anuncio
ACTORES:	Usuario
PROPÓSITO:	Visualizar la ubicación del anuncio por medio del mapa de Google Maps
RESUMEN:	El usuario podrá visualizar un mapa que indica la posición del anuncio que seleccionó.
TIPO:	Primario
PRE-CONDICIONES:	El Anuncio seleccionado debe tener disponible las coordenadas de ubicación.

Fuente: Elaboración Propia

- **Detalle del Anuncio:** Cuando se seleccione un anuncio, se abrirá un ModalController, en el cual se verá toda la información concerniente a dicho anuncio, de la misma manera podrá ver el perfil del anunciante y enviar una solicitud de postulación.

Tabla 19. Caso de uso: Detalle del Anuncio

CASO DE USO:	Detalle del Anuncio
ACTORES:	Usuario

PROPÓSITO:	Ver el Anuncio con toda la información disponible.
RESUMEN:	Visualizar el anuncio en forma detallada (quién publicó, cuando, título y descripción) y tener acceso a botones para ver el perfil de la persona que publicó y para postular al anuncio.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el anuncio deseado.

Fuente: Elaboración Propia

Tabla 20. Caso de uso: Postular a la Oferta Laboral

CASO DE USO:	Postular a la Oferta Laboral
ACTORES:	Usuario
PROPÓSITO:	Postular al anuncio de trabajo deseado.
RESUMEN:	El usuario podrá postular al anuncio de trabajo que seleccionó y enviar una solicitud a la persona que publicó dicho anuncio.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el botón de postular.

Fuente: Elaboración Propia

Tabla 21. Caso de uso: Perfil del Usuario

CASO DE USO:	Perfil del Usuario
ACTORES:	Usuario
PROPÓSITO:	Ver el Perfil del Usuario.
RESUMEN:	Visualizar el Perfil del usuario con todos los datos informativos disponibles.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el botón de Perfil en cualquier vista.

Fuente: Elaboración Propia

Tabla 22. Caso de uso: Publicar Anuncio

CASO DE USO:	Publicar Anuncio
ACTORES:	Usuario

PROPÓSITO:	Publicar un anuncio nuevo de oferta laboral.
RESUMEN:	El usuario podrá publicar un anuncio, en el cual, pondrá un título, una descripción detallada, tomar una foto y poner la ubicación.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el botón de publicar.

Fuente: Elaboración Propia

- **Menú Principal de la aplicación:** Es un menú lateral, en el que se visualizará las diferentes opciones que tiene la aplicación. Con este menú se podrá acceder a cualquier función de la aplicación.

Tabla 23. Caso de uso: Menú Toggle

CASO DE USO:	Menú Toggle
ACTORES:	Usuario
PROPÓSITO:	Moverse por las distintas opciones de la aplicación.
RESUMEN:	El usuario seleccionará cualquier opción del menú.
TIPO:	Primario
PRE-CONDICIONES:	Caso de uso inmediato, después de haberse autenticado.

Fuente: Elaboración Propia

- **Opción Administrar:** Se visualiza antes que nada un menú al estilo Tabs de Ionic, en el cual se irá navegando por otra tres opciones que son: Mis Anuncios, Solicitudes y Puntuación.

Tabla 24. Caso de uso: Opción Administrar - tabs

CASO DE USO:	Opción Administrar - tabs
ACTORES:	Usuario
PROPÓSITO:	Navegar entre opciones.
RESUMEN:	Ir navegando entre las tres opciones de administración definidas.
TIPO:	Primario

PRE-CONDICIONES:	Caso de uso inmediato, después de haber seleccionado la opción del menú principal.
-------------------------	--

Fuente: Elaboración Propia

- **Administrar mis anuncios:** Es la primera vista de la opción administrar, en el cual se visualiza todos los anuncios que hizo el usuario autenticado, con múltiples opciones de configuración como: Crear nuevo anuncio, Ver el anuncio detalladamente, Ver la ubicación del anuncio, cerrar el anuncio y eliminar el anuncio; los tres primeros ya fueron definidos anteriormente.

Tabla 25. Caso de uso: Mis Anuncios

CASO DE USO:	Mis Anuncios
ACTORES:	Usuario
PROPÓSITO:	Visualizar todos los anuncios publicados por este usuario.
RESUMEN:	El usuario podrá cerrar el anuncio o eliminarlo.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción "Mis Anuncios" del tab.

Fuente: Elaboración Propia

Tabla 26. Caso de uso: Cerrar mi Anuncio

CASO DE USO:	Cerrar mi Anuncio
ACTORES:	Usuario
PROPÓSITO:	Indicar que se cerró el anuncio.
RESUMEN:	El usuario al cerrar el anuncio, cambiará la disponibilidad de dicho anuncio, esto quiere decir que el anuncio o convocatoria ya finalizó.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el botón de "Cerrar" del anuncio.

Fuente: Elaboración Propia

Tabla 27. Caso de uso: Eliminar mi Anuncio

CASO DE USO:	Eliminar mi Anuncio
ACTORES:	Usuario
PROPÓSITO:	Eliminar el anuncio seleccionado.
RESUMEN:	El usuario podrá eliminar el anuncio de la base de datos (cambiar estado a “ELIMINADO”).
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el botón “Eliminar” del anuncio.

Fuente: Elaboración Propia

- **Administrar mis solicitudes y aceptados:** Es la segunda vista en la opción administrar, en el cual se visualiza las solicitudes que se enviaron al usuario por su anuncio y de la misma forma a los usuarios que fueron aceptados.
 - **Solicitudes:** Cuando tengas algún anuncio publicado, podrás visualizar las solicitudes que te envíen con las siguientes opciones:
 - **Ver Perfil:** Te dará acceso al perfil del solicitante con su información personal y la reputación basada en cantidad de estrellas de cero a cinco, de mala a buena reputación respectivamente.
 - **Agregar:** Esta opción te permitirá aceptar la solicitud.
 - **Eliminar:** Eliminará la solicitud que le enviaron al usuario.
 - **Llamar:** Esta acción te permitirá llamar inmediatamente al número telefónico del solicitante.
 - **Solicitudes Aceptadas:** Inmediatamente tras aceptar una solicitud podrás visualizar en un apartado la solicitud que aceptaste con las siguientes opciones:
 - **Ver Perfil:** Te permitirá ver el perfil del usuario aceptado con su información personal y la reputación basada en

cantidad de estrellas de cero a cinco, de mala a buena reputación respectivamente; esta acción disponible al tocar el avatar (foto de perfil) o nombre del usuario.

- **Chat:** Opción que dará acceso a un chat con el otro usuario, para poder coordinar con respecto al anuncio.
- **Llamar:** Esta acción te permitirá llamar inmediatamente al número telefónico del solicitante, para coordinar con respecto al anuncio.
- **Contratar (Disponible solo para el Dueño del anuncio):** Opción que te permitirá contratar al usuario postulante ha dicho anuncio.
- **Bloquear (Disponible solo para el Dueño del anuncio):** Opción que te permitirá Bloquear al usuario postulante ha dicho anuncio, en caso de que no cumpla los requisitos.

Tabla 28. Caso de uso: Mis Solicitudes

CASO DE USO:	Mis Solicitudes
ACTORES:	Usuario
PROPÓSITO:	Visualizar las solicitudes recibidas.
RESUMEN:	Ver las solicitudes recibidas con respecto a un anuncio publicado.
TIPO:	Primario
PRE-CONDICIONES:	Haber ingresado en la opción “Recibidos” del tab.

Fuente: Elaboración Propia

Tabla 29. Caso de uso: Mis Solicitudes – Opción Agregar

CASO DE USO:	Mis Solicitudes – Opción Agregar
ACTORES:	Usuario
PROPÓSITO:	Aceptar la solicitud.
RESUMEN:	Permite aceptar la solicitud y pasar al apartado de aceptados, para la comunicación.

TIPO:	Primario
PRE-CONDICIONES:	Haber recibido una solicitud a un anuncio.

Fuente: Elaboración Propia

Tabla 30. Caso de uso: Mis Solicitudes – Opción Eliminar

CASO DE USO:	Mis Solicitudes – Opción Eliminar
ACTORES:	Usuario
PROPÓSITO:	Eliminar Solicitud.
RESUMEN:	Permite eliminar la solicitud recibida, por decisión del usuario.
TIPO:	Primario
PRE-CONDICIONES:	Haber recibido una solicitud a un anuncio.

Fuente: Elaboración Propia

Tabla 31. Caso de uso: Mis Solicitudes – Opción Llamar

CASO DE USO:	Mis Solicitudes – Opción Llamar
ACTORES:	Usuario
PROPÓSITO:	Llamar al usuario solicitante.
RESUMEN:	Permite llamar inmediatamente al usuario de la solicitud recibida.
TIPO:	Primario
PRE-CONDICIONES:	Haber recibido una solicitud a un anuncio.

Fuente: Elaboración Propia

Tabla 32. Caso de uso: Mis Solicitudes Aceptadas

CASO DE USO:	Mis Solicitudes Aceptadas
ACTORES:	Usuario
PROPÓSITO:	Visualizar las solicitudes aceptadas.
RESUMEN:	Ver las solicitudes aceptadas con respecto a un anuncio publicado, con opciones personalizadas.
TIPO:	Primario
PRE-CONDICIONES:	Haber ingresado en la opción “Recibidos” del tab.

Fuente: Elaboración Propia

Tabla 33. Caso de uso: Aceptados - Chat

CASO DE USO:	Aceptados – Chat
ACTORES:	Usuario
PROPÓSITO:	Brindar un chat con el postulante.
RESUMEN:	Permite tener una conversación en tiempo real mediante un chat, para coordinar con respecto al anuncio.
TIPO:	Primario
PRE-CONDICIONES:	Haber aceptado una solicitud.

Fuente: Elaboración Propia

Tabla 34. Caso de uso: Aceptados - Llamar

CASO DE USO:	Aceptados – Llamar
ACTORES:	Usuario
PROPÓSITO:	Llamar al usuario aceptado.
RESUMEN:	Permite hacer una llamada inmediata al usuario aceptado.
TIPO:	Primario
PRE-CONDICIONES:	Haber aceptado una solicitud.

Fuente: Elaboración Propia

Tabla 35. Caso de uso: Aceptados - Contratar

CASO DE USO:	Aceptados – Contratar
ACTORES:	Usuario Dueño del Anuncio
PROPÓSITO:	Contratar al usuario postulante.
RESUMEN:	Cambia el estado del anuncio y se cierra la convocatoria.
TIPO:	Primario
PRE-CONDICIONES:	Haber aceptado una solicitud y previamente haber coordinado.

Fuente: Elaboración Propia

Tabla 36. Caso de uso: Aceptados - Bloquear

CASO DE USO:	Aceptados – Bloquear
ACTORES:	Usuario Dueño del Anuncio

PROPÓSITO:	Bloquear las coordinaciones.
RESUMEN:	Permite Bloquear el contacto con el usuario postulante, acción que indica que no cumplió con los requisitos.
TIPO:	Primario
PRE-CONDICIONES:	Haber aceptado una solicitud.

Fuente: Elaboración Propia

- **Administrar las puntuaciones y calificación:** Es la tercera vista en la opción Administrar, en el cual se visualiza los usuarios contratados con respecto a un anuncio y ver las calificaciones que se realizaron por ambas partes, tanto contratador como contratado; tiene las siguientes acciones:
 - **Chat:** Disponible para seguir en comunicación y coordinación.
 - **Llamar:** Disponible para llamar al usuario contratado.
 - **Evaluar:** Acción que permite la evaluación por ambas partes, con respecto a cómo se va desarrollando las actividades que implica el trabajo, dicha evaluación en base a estrellas pasa a formar parte del perfil de cada usuario.

Tabla 37. Caso de uso: Puntuación

CASO DE USO:	Puntuación
ACTORES:	Usuario
PROPÓSITO:	Visualizar a los usuarios contratados.
RESUMEN:	Permite visualizar la lista de usuarios contratados, con sus respectivas calificaciones en estrellas
TIPO:	Primario
PRE-CONDICIONES:	Haber Contratado a un usuario.

Fuente: Elaboración Propia

Tabla 38. Caso de uso: Puntuación – Evaluar

CASO DE USO:	Puntuación - Evaluar
---------------------	----------------------

ACTORES:	Usuario
PROPÓSITO:	Evaluar el desempeño del usuario.
RESUMEN:	Permite asignar estrellas de acuerdo al desempeño del usuario.
TIPO:	Primario
PRE-CONDICIONES:	Haber Contratado un usuario.

Fuente: Elaboración Propia

Tabla 39. Caso de uso: Puntuación – Evaluar – Contratador

CASO DE USO:	Puntuación – Evaluar – Contratador
ACTORES:	Usuario Contratador
PROPÓSITO:	Evaluar desempeño.
RESUMEN:	Permite evaluar el desempeño del usuario contratado cada vez que sea necesario y hacer un comentario al respecto
TIPO:	Primario
PRE-CONDICIONES:	Haber Contratado un usuario.

Fuente: Elaboración Propia

Tabla 40. Caso de uso: Puntuación – Evaluar – Contratado

CASO DE USO:	Puntuación – Evaluar – Contratado
ACTORES:	Usuario Contratado
PROPÓSITO:	Evaluar desempeño.
RESUMEN:	Permite evaluar el desempeño del usuario que lo contrato cada vez que sea necesario y hacer un comentario al respecto
TIPO:	Primario
PRE-CONDICIONES:	Haber sido Contratado.

Fuente: Elaboración Propia

- **Perfil del Usuario:** Opción del menú principal en el cual se visualiza toda la información al usuario autenticado. Con las siguientes acciones:

- **Modificar Avatar:** Acción que permite la modificación del avatar (foto de perfil), mediante selección de galería o tomar foto con la cámara del dispositivo.
- **Editar Nombre:** Acción que permite modificar el nombre del usuario.
- **Editar Apellido:** Acción que permite modificar el apellido del usuario.
- **Editar Teléfono:** Acción que permite modificar el teléfono del usuario.
- **Editar Ciudad:** Acción que permite modificar la ciudad del usuario.
- **Cambiar Ubicación:** Acción que permite modificar las coordenadas de ubicación del usuario.
- **Ver Perfil:** Acción que permite ver el perfil del usuario de manera detallada.
- **Ampliar Curriculum:** Acción que permite al usuario ampliar con mayor información su perfil de presentación.
- **Cerrar Sesión:** Acción que cierra la sesión iniciada y vuelve a la vista de login.

Tabla 41. Caso de uso: Perfil

CASO DE USO:	Perfil
ACTORES:	Usuario
PROPÓSITO:	Editar datos del usuario.
RESUMEN:	Permite actualizar o modificar los datos del usuario.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción “Perfil” del menú principal.

Fuente: Elaboración Propia

Tabla 42. Caso de uso: Perfil – Avatar

CASO DE USO:	Perfil – Avatar
ACTORES:	Usuario
PROPÓSITO:	Editar avatar del usuario.
RESUMEN:	Acción que permite actualizar o modificar el avatar (foto de perfil) del usuario, mediante acceso y carga de imagen de la galería del teléfono o con una foto capturada por la cámara del dispositivo.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado las opciones de imagen del avatar.

Fuente: Elaboración Propia

Tabla 43. Caso de uso: Perfil – Nombre

CASO DE USO:	Perfil – Nombre
ACTORES:	Usuario
PROPÓSITO:	Editar Nombre(s) del usuario.
RESUMEN:	Acción que permite modificar el nombre del usuario, mediante un AlertController.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el nombre del usuario.

Fuente: Elaboración Propia

Tabla 44. Caso de uso: Perfil – Apellido

CASO DE USO:	Perfil – Apellido
ACTORES:	Usuario
PROPÓSITO:	Editar Apellido(s) del usuario.
RESUMEN:	Acción que permite modificar el apellido del usuario, mediante un AlertController.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el apellido del usuario.

Fuente: Elaboración Propia

Tabla 45. Caso de uso: Perfil – Teléfono

CASO DE USO:	Perfil – Teléfono
ACTORES:	Usuario
PROPÓSITO:	Editar teléfono del usuario.
RESUMEN:	Acción que permite modificar el número de teléfono del usuario, mediante un alertController.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el N° de teléfono del usuario.

Fuente: Elaboración Propia

Tabla 46. Caso de uso: Perfil – Ciudad

CASO DE USO:	Perfil – Ciudad
ACTORES:	Usuario
PROPÓSITO:	Editar Ciudad del usuario.
RESUMEN:	Acción que permite modificar la ciudad del usuario, mediante un alertController.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la ciudad del usuario.

Fuente: Elaboración Propia

Tabla 47. Caso de uso: Perfil – Ubicación

CASO DE USO:	Perfil – Ubicación
ACTORES:	Usuario
PROPÓSITO:	Modificar la ubicación del usuario.
RESUMEN:	Acción que permite modificar o actualizar la ubicación del usuario, mediante un Toggle y el plugin de ubicación.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado toggle de ubicación.

Fuente: Elaboración Propia

Tabla 48. Caso de uso: Ampliar Curriculum

CASO DE USO:	Ampliar Curriculum
ACTORES:	Usuario

PROPÓSITO:	Actualizar datos sobre el curriculum del usuario.
RESUMEN:	Acción que permite actualizar los datos referentes al perfil o curriculum del usuario, llenando campos como: Ocupación, Experiencia y Habilidades.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el botón “AMPLIAR CURRICULUM”.

Fuente: Elaboración Propia

- **Sala de Chats:** La aplicación brinda una sala de chat, independiente de los anuncios publicados, donde el usuario podrá buscar a sus amigos y conversar con ellos.
 - **Solicitud de amistad:** solicitud que se recepciona del usuario que quiere conversar mediante el chat.
 - **Amigos:** Las solicitudes aceptadas pasan a formar parte del apartado de amigos, disponibles para conversar.
 - **Búsqueda de amigos:** El usuario podrá buscar a sus amigos dentro de todos los usuarios registrados en la aplicación y seleccionarlos para enviar la solicitud correspondiente.

Tabla 49. Caso de uso: Sala de Chats

CASO DE USO:	Sala de Chats
ACTORES:	Usuario
PROPÓSITO:	Visualizar solicitudes y amigos.
RESUMEN:	Poder ver las solicitudes recibidas y los amigos disponibles para conversar.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción “Chats” del menú principal.

Fuente: Elaboración Propia

Tabla 50. Caso de uso: Solicitudes de amistad

CASO DE USO:	Solicitudes de amistad.
---------------------	-------------------------

ACTORES:	Usuario
PROPÓSITO:	Visualizar las solicitudes recibidas.
RESUMEN:	Poder ver las solicitudes y aceptarlas o eliminarlas.
TIPO:	Primario
PRE-CONDICIONES:	Haber recibido una solicitud.

Fuente: Elaboración Propia

Tabla 51. Caso de uso: Amigos

CASO DE USO:	Amigos
ACTORES:	Usuario
PROPÓSITO:	Visualizar a los amigos del usuario.
RESUMEN:	Poder ver a todos los amigos de un usuario e inicializar una conversación por medio del chat de la aplicación.
TIPO:	Primario
PRE-CONDICIONES:	Haber aceptado una solicitud.

Fuente: Elaboración Propia

Tabla 52. Caso de uso: Búsqueda de amigos

CASO DE USO:	Búsqueda de amigos.
ACTORES:	Usuario
PROPÓSITO:	Buscar amigos y enviar solicitud de amistad.
RESUMEN:	Poder buscar a personas y enviarles una solicitud de amistad para poder conversar por el chat de la aplicación.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el botón de búsqueda.

Fuente: Elaboración Propia

- **Cerrar Sesión:** Permite cerrar la sesión del usuario autenticado. Toma las credenciales de autenticación de usuario y le da fin a la sesión iniciada con dichas credenciales.

Tabla 53. Caso de uso: Cerrar Sesión

CASO DE USO:	Cerrar Sesión
ACTORES:	Usuario
PROPÓSITO:	Dar fin a la sesión.
RESUMEN:	Acción que permite cerrar la sesión y salir al formulario de logeo de usuario.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción “Cerrar Sesión” del menú principal.

Fuente: Elaboración Propia

- **Acerca De:** En esta opción, se podrá visualizar información referente al desarrollador de la aplicación móvil.

Tabla 54. Caso de uso: Acerca De

CASO DE USO:	Acerca De
ACTORES:	Usuario
PROPÓSITO:	Créditos al desarrollador.
RESUMEN:	Información del desarrollador y de la aplicación móvil.
TIPO:	Secundario
PRE-CONDICIONES:	Haber seleccionado la opción “Acerca De” del menú principal.

Fuente: Elaboración Propia

4.2.3. Fase de Producción

En la fase de producción es donde se diseña la aplicación móvil de acuerdo a los casos de uso que se desarrolló en la fase de inicialización, en el que se determinó los requerimientos del proyecto, logrando determinar la funcionalidad que tiene la aplicación móvil.

La implementación de la aplicación móvil se llevó a cabo usando un ciclo de desarrollo iterativo e incremental, también se hizo uso del desarrollo dirigido por pruebas (TDD, Test-Driven Development), el cual significa que cada funcionalidad desarrollada fue probado.

Para cada nueva iteración que se hizo, se realizó las siguientes actividades:

- ✓ Análisis de requerimientos de la iteración.
- ✓ Desarrollo de la iteración con base a los requerimientos.
- ✓ Integración de la nueva funcionalidad desarrollada con la rama principal de la aplicación.
- ✓ Retroalimentación en base a la iteración.

4.2.3.1. Diseño de la aplicación Móvil

Teniendo en cuenta como base los análisis realizados en la fase de iniciación, en donde se definió los requerimientos (casos de uso) de la aplicación. En seguida se define la estructura de la aplicación móvil en términos generales.

Tabla 55. Estructura en términos generales

Acceso a la aplicación	En esta primera parte, están todas las interfaces y funcionalidades anteriores al inicio de sesión. El cual incluye la vista de login, recuperar contraseña, registrarse y ver los anuncios sin autenticarse.
Página Principal	Donde se muestra todos los anuncios con sus respectivos estados y opciones de acceso a otras interfaces necesarias, para la interacción del usuario con el anuncio.
Administrar anuncios	Sección que permite administrar los anuncios, solicitudes y calificaciones de contratados, con sus respectivas acciones de configuración y administración de las mismas. Da acceso al chat con los postulantes a la oferta laboral.
Configuración de Perfil	Sección donde se configura, edita, actualiza y modifica la información referente al usuario autenticado.
Sala de Chats	En esta sección se muestra la sala de chats de amigos con quienes puedes conversar independientemente de los anuncios, te permite buscar personas, enviar, recibir y aceptar solicitudes de amistad, para interactuar con otros usuarios de la aplicación.
Providers (Servicios)	Los Providers, son los servicios que se implementa para la manipulación de los datos, de conexiones a la base de datos, administrar LocalStorage y peticionar funciones desde las interfaces.

Flirebase	Integración del proyecto y de la aplicación móvil con la plataforma Firebase de Base de Datos en Tiempo Real.
------------------	---

Fuente: Elaboración Propia

4.2.3.2. Dependencias de Paquetes de la aplicación móvil

Para usar las funcionalidades de Ionic es necesario una serie de paquetes. Los paquetes son de uso general en el proyecto. De acuerdo a la necesidad de este proyecto se hará uso de los paquetes que se muestran en la imagen.

Imagen 2. Dependencias

```

dependencies: {
  "@angular/common": "4.1.3",
  "@angular/compiler": "4.1.3",
  "@angular/compiler-cli": "4.1.3",
  "@angular/core": "4.1.3",
  "@angular/forms": "4.1.3",
  "@angular/http": "4.1.3",
  "@angular/platform-browser": "4.1.3",
  "@angular/platform-browser-dynamic": "4.1.3",
  "@ionic-native/call-number": "^4.7.0",
  "@ionic-native/contacts": "^4.7.0",
  "@ionic-native/core": "^3.12.1",
  "@ionic-native/fcm": "^4.7.0",
  "@ionic-native/file": "^4.1.0",
  "@ionic-native/file-chooser": "^4.1.0",
  "@ionic-native/file-path": "^4.1.0",
  "@ionic-native/google-maps": "^4.7.0",
  "@ionic-native/network": "^4.7.0",
  "@ionic-native/sim": "^4.7.0",
  "@ionic-native/splash-screen": "3.12.1",
  "@ionic-native/status-bar": "3.12.1",
  "@ionic/storage": "2.0.1",
  "angularfire2": "^4.0.0-rc.2",
  "call-number": "^1.0.1",
  "cordova-android": "^6.2.3",
  "cordova-ios": "^4.4.0",
  "firebase": "^4.3.0",
  "ionic-angular": "3.6.0",
  "ionic-native": "^2.9.0",
  "ionic-plugin-keyboard": "^2.2.1",
  "ionicons": "3.0.0",
  "mx.ferreyra.callnumber": "~0.0.2",
  "promise-polyfill": "6.0.2",
  "rxjs": "5.4.0",
  "sw-toolbox": "3.6.0",
  "zone.js": "0.8.12"
},
devDependencies: {
  "@ionic/app-scripts": "2.1.3",
  "typescript": "2.3.4"
},

```

Fuente: Elaboración Propia

4.2.3.3. Cordova – Plugins necesarias para la aplicación móvil

Los plugins se instalan en la aplicación de acuerdo a la necesidad, estos plugins son de uso netamente para los dispositivos móviles de acuerdo a la plataforma para la cual se compilará, permiten acceder a las características nativas del sistema operativo del dispositivo móvil, así como acceso a características de hardware del dispositivo móvil como cámara, gps, entre otros.

En la siguiente imagen se visualiza los plugins que se usan en el proyecto de la aplicación móvil:

Imagen 3. Cordova – Plugins

```
"cordova": {
  "plugins": {
    "cordova-plugin-console": {},
    "cordova-plugin-device": {},
    "cordova-plugin-splashscreen": {},
    "cordova-plugin-statusbar": {},
    "cordova-plugin-whitelist": {},
    "ionic-plugin-keyboard": {},
    "cordova-plugin-filechooser": {},
    "cordova-plugin-filepath": {},
    "cordova-plugin-file": {},
    "cordova-plugin-geolocation": {},
    "cordova-plugin-camera": {},
    "cordova-plugin-googlemaps": {
      "API_KEY_FOR_ANDROID": "AIzaSyDxedK0rE4Z1YvqPxmot9GQZgXZkvJRL5A",
      "API_KEY_FOR_IOS": "AIzaSyDxedK0rE4Z1YvqPxmot9GQZgXZkvJRL5A",
      "NSLOCATIONWHENINUSEUSAGEDESCRIPTION": "Show your location on the map",
      "NSLOCATIONALWAYSUSAGEDESCRIPTION": "Trace your location on the map"
    },
    "cordova-plugin-sim": {},
    "cordova-plugin-network-information": {},
    "mx.ferreyra.callnumber": {},
    "cordova-plugin-contacts": {
      "CONTACTS_USAGE_DESCRIPTION": " "
    },
    "cordova-plugin-fcm": {}
  },
  "platforms": [
    "android",
    "ios"
  ]
}
```

Fuente: Elaboración Propia

4.2.3.4. Configuración Inicial del Proyecto

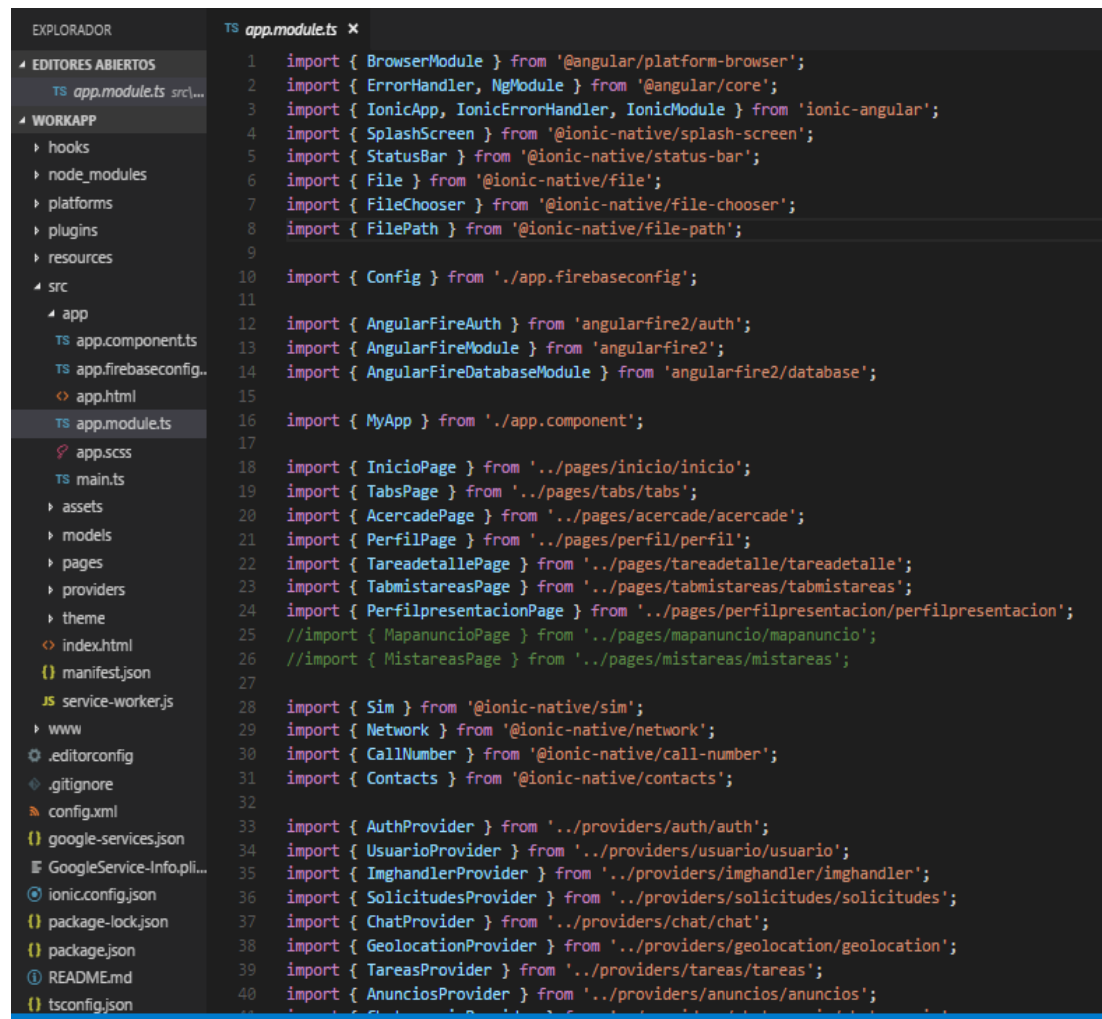
Primeramente se debe crear el proyecto por medio de la terminal, haciendo uso del comando:

\$ ionic start miApp blank

Al ejecutarse el comando se creará toda la estructura del proyecto, junto al Package.json, donde se encuentra todas las dependencias que se usarán.

En el archivo “app.module.ts” que se encuentra dentro del directorio “MyApp/src/app” se debe de importar todas las dependencias y páginas que se harán uso en el proyecto. Así como se puede ver en la imagen:

Imagen 4. Configuración app.module.ts



```
EXPLORADOR TS app.module.ts x
├ EDITORES ABIERTOS
  TS app.module.ts src\...
├ WORKAPP
  ├── hooks
  ├── node_modules
  ├── platforms
  ├── plugins
  ├── resources
  └ src
    ├── app
    │   ├── app.component.ts
    │   ├── app.firebaseconfig...
    │   ├── app.html
    │   └ TS app.module.ts
    ├── app.scss
    ├── main.ts
    ├── assets
    ├── models
    ├── pages
    ├── providers
    └ theme
      ├── index.html
      ├── manifest.json
      ├── service-worker.js
      └ www
        ├── .editorconfig
        ├── .gitignore
        ├── config.xml
        ├── google-services.json
        ├── GoogleService-Info.pli...
        ├── ionic.config.json
        ├── package-lock.json
        ├── package.json
        ├── README.md
        └ tsconfig.json
1  import { BrowserModule } from '@angular/platform-browser';
2  import { ErrorHandler, NgModule } from '@angular/core';
3  import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
4  import { SplashScreen } from '@ionic-native/splash-screen';
5  import { StatusBar } from '@ionic-native/status-bar';
6  import { File } from '@ionic-native/file';
7  import { FileChooser } from '@ionic-native/file-chooser';
8  import { FilePath } from '@ionic-native/file-path';
9
10 import { Config } from './app.firebaseconfig';
11
12 import { AngularFireAuth } from 'angularfire2/auth';
13 import { AngularFireModule } from 'angularfire2';
14 import { AngularFireDatabaseModule } from 'angularfire2/database';
15
16 import { MyApp } from './app.component';
17
18 import { InicioPage } from '../pages/inicio/inicio';
19 import { TabsPage } from '../pages/tabs/tabs';
20 import { AcercadePage } from '../pages/acercade/acercade';
21 import { PerfilPage } from '../pages/perfil/perfil';
22 import { TareadetallePage } from '../pages/tareadetalle/tareadetalle';
23 import { TabmistareasPage } from '../pages/tabmistareas/tabmistareas';
24 import { PerfilpresentacionPage } from '../pages/perfilpresentacion/perfilpresentacion';
25 //import { MapanuncioPage } from '../pages/mapanuncio/mapanuncio';
26 //import { MistareasPage } from '../pages/mistareas/mistareas';
27
28 import { Sim } from '@ionic-native/sim';
29 import { Network } from '@ionic-native/network';
30 import { CallNumber } from '@ionic-native/call-number';
31 import { Contacts } from '@ionic-native/contacts';
32
33 import { AuthProvider } from '../providers/auth/auth';
34 import { UsuarioProvider } from '../providers/usuario/usuario';
35 import { ImghandlerProvider } from '../providers/imghandler/imghandler';
36 import { SolicitudesProvider } from '../providers/solicitudes/solicitudes';
37 import { ChatProvider } from '../providers/chat/chat';
38 import { GeolocationProvider } from '../providers/geolocation/geolocation';
39 import { TareasProvider } from '../providers/tareas/tareas';
40 import { AnunciosProvider } from '../providers/anuncios/anuncios';
```

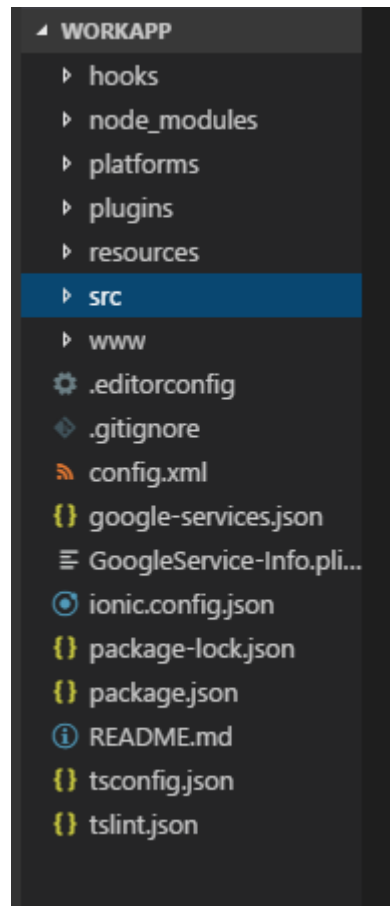
Fuente: Elaboración Propia

4.2.3.5. Estructura de directorios y archivos del proyecto

La estructura de directorios o carpetas inicia con la creación del proyecto desde la terminal, el cual genera la carpeta raíz del Proyecto y dentro las subcarpetas y archivos necesarios para empezar la codificación de la aplicación.

En seguida se observa la estructura de los directorios y archivos que se encuentran en el proyecto de la aplicación móvil:

Imagen 5. Directorio del Proyecto “WorkApp”



Fuente: Elaboración Propia

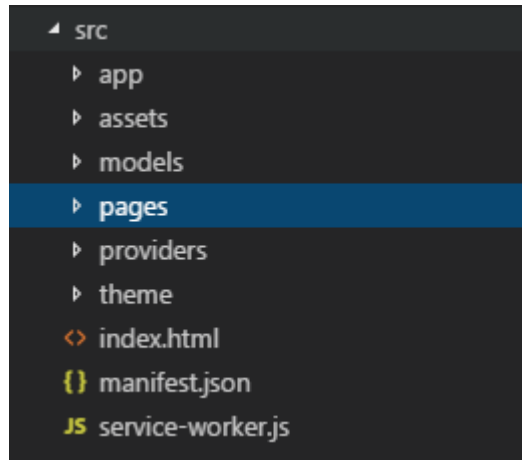
Tabla 56. Directorio del Proyecto “WorkApp”

Hooks	Contiene el archivo README.md creado por IONIC
Node_modules	Contiene los paquetes y dependencias del proyecto como: ionic-angular, @ionic-

	native/..., @angular/..., angularfire2, firebase, entre otros.
Platforms	Contiene carpetas como Android, ios y Windows dependiendo que plataformas agregaste previamente, cada sub-carpeta contiene los archivos, paquetes y plugins necesarios para la compilación de cada plataforma.
Plugins	Contiene subcarpetas de cada plugin de cordova u otros que se instalaron en el proyecto, con sus respectivos archivos.
resoruces	Contiene los iconos por defecto de la aplicación que se compilará y la vista SplashScreen de carga de la aplicación.
src	Esta es la carpeta más importante, porque en ella estarán alojadas todos los archivos de configuración y de programación que forman el proyecto y la aplicación.
www	Contiene archivos javascript en una carpeta build e imágenes estáticas para el icono de la aplicación y el SplashScreen, por default.
GoogleService-Info.plist y google-services.json	Estos dos archivos tienen que ser descargados desde la consola web de firebase y copiados a la carpeta raíz del proyecto. El archivo GoogleService-Info.plist es el archivo necesario para correr en la plataforma ios y google-services.json es el archivo necesario para correr en la plataforma Android.
Package.json	Es un archive json que contiene información general del proyecto, las dependencias y paquetes y los plugins instalados. Es un archivo muy importante que sive para migrar el proyecto e instalar nuevamente de ser necesario.
.editrconfig, .gitignore, config.xml, ionic.config.json, package-lock.json, README.md, tsconfig.json, tslint.json	Todo estos archivos son creados y usados por el framework IONIC 3.

Fuente: Elaboración Propia

Imagen 6. Carpeta “src”



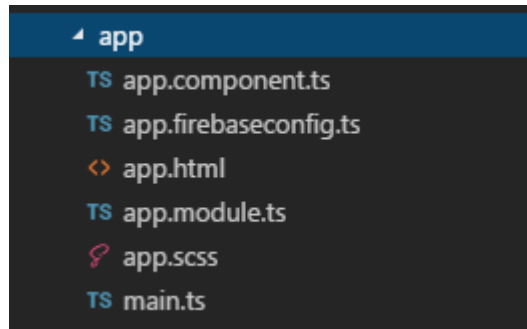
Fuente: Elaboración Propia

Tabla 57. Tipos de archivos manejados en el proyecto

Archivos TypeScript “.ts”	TypeScript es un lenguaje de programación libre y de código abierto desarrollado por Microsoft, es un super conjunto de JavaScript con herramientas de programación orientada a objetos y añade tipado estático.
Archivos .html	En este proyecto es el típico archivo html pero con etiqueta ion de ionic y uso de directivas y funcionalidades de angular.
Archivos Sass “.scss”	SASS es una extensión de CSS escrita en RUBY que añade características muy potentes y elegantes a este lenguaje de estilos (CSS). Permite el uso de variables, reglas CSS anidadas, mixins, importación de hojas de estilos y muchos más.

Fuente: Elaboración Propia

Imagen 7. Carpeta “src/app”



Fuente: Elaboración Propia

Tabla 58. Carpeta "src/app"

App.components.ts	Redirige la página para que cargue la página de login como página raíz o root page.
App.firebaseconfig.ts	Exporta la configuración de Firebase, proporcionada por la misma.
App.module.ts	Importa todos los paquetes o dependencias, los Page y los Providers (Servicios), que son usados en la aplicación.
App.html, app.scss y main.ts	No sufren modificación, se dejan por default.

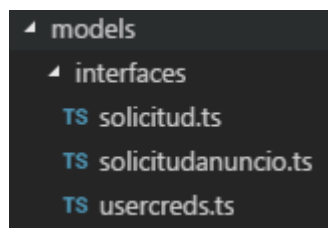
Fuente: Elaboración Propia

Tabla 59. Carpeta "src/assets"

Assets	Contiene sub-carpeta destinada para imágenes estáticas de la aplicación.
---------------	--

Fuente: Elaboración Propia

Imagen 8. Carpeta "src/models"



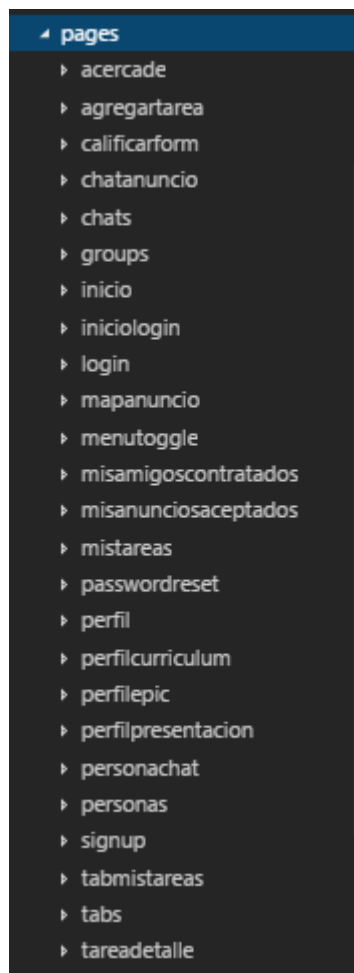
Fuente: Elaboración Propia

Tabla 60. Carpeta “src/models”

Interfaces	solicitud.ts	Exporta la interface requerida para las solicitudes de amistad.
	solicitudanuncio.ts	Exporta la interface requerida para las solicitudes de postulación de los anuncios.
	usercreds.ts	Exporta la interface requerida para las credenciales de los usuarios.

Fuente: Elaboración Propia

Imagen 9. Carpeta “src/pages”



Fuente: Elaboración Propia

Tabla 61. Carpeta “src/pages” (orden alfabético)

Acerca De	acercade.html	Etiquetas de ionic para la interfaz. Muestra la información del desarrollador y de la aplicación.
	acerdade.module.ts y acercade.scss	Archivos sin editar, valor por default.
	acercade.ts	Variables definidas para su uso en el archivo .html
Agregartarea	agregartarea.html	Diseño del formulario para agregar datos del nuevo anuncio a publicar.
	agregartarea.scss	Estilos para mostrar la imagen tomada por la cámara.
	agregartarea.ts	Importación de los servicios, captura de datos del formulario, obtención de la ubicación, obtención de la fecha, manipulación de la cámara del dispositivo y obtención de la imagen.
Calificarform	calificarform.html	Formulario de calificación de usuarios con ion-select dinámico para cantidad de estrellas.
	calificarform.ts	Importa servicios, captura datos del formulario e implementa el algoritmo de calificación de usuarios y asignación de estrellas al perfil.
chatanuncio	chatanuncio.html	Interfaz para el chat; sección de carga de imágenes, contenedor de mensajes escritos y formulario de envío de mensajes.
	chatanuncio.scss	Estilos para la interfaz del chat, para contenedor y muestra de mensajes e imágenes.
	chatanuncio.ts	Importación de servicios, captura de mensajes escritos y petición de mensajes enviados, identificación de los usuarios vinculados al anuncio, carga de avatar de usuarios, captura y envío de imagen seleccionado; conectados en tiempo real por un event.
chats	chats.html	Diseño de la sala de chats, visualización de solicitudes recibidas con opciones de agregar y eliminar; y lista de amigos aceptados.

	chats.ts	Importación de servicios, petición de solicitudes recibidas y suscripción a events para visualización en tiempo real.
inicio	inicio.html	Lista de todos los anuncios publicados con información resumida y estado de disponibilidad, con acceso a detalle y ubicación.
	inicio.ts	Importación de servicio, suscripción a event para actualización de lista en tiempo real y acciones de acceso a otras vistas.
iniciologin	iniciologin.html	Lista de todos los anuncios publicados con información resumida y estado de disponibilidad, sin acceso a detalle, solo a registro.
	Iniciologin.ts	Suscripción a event para actualización de lista en tiempo real.
login	login.html	Diseño del formulario de login.
	login.scss	Estilos para el formulario y la vista login.
	login.ts	Importación de servicios, acceso a otros formularios, revisión de estado de conexión a internet, autenticación de usuario por credenciales e inicio de sesión, registro de token que identifica de manera única a la aplicación en el dispositivo que se está usando.
mapanuncio	mapanuncio.html	Definición del contenedor del mapa.
	mapanuncio.scss	Estilos para el mapa.
	mapanuncio.ts	Importación de los servicios de Google Maps, carga del mapa con su respectivo marcador del anuncio.
menutoggle	menutoggle.html	Diseño del menú lateral desplegable.
	menutoggle.ts	Definición de variables de enrutamiento a las páginas definidas y método de cerrar sesión.
misamigoscontratados	misamigoscontratados.html	Diseño de la vista, visualización de la lista de usuarios contratados,

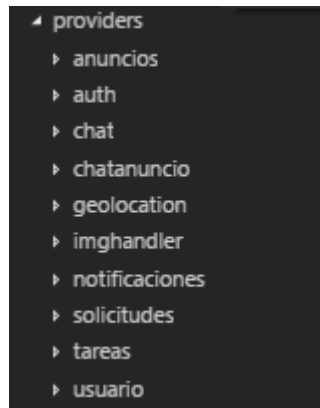
		visualización de las estrellas asignadas.
	misamigoscontratados.ts	Importación del servicio, suscripción a event para visualizar en tiempo real, carga de estrellas calificadas con respecto al anuncio.
misanunciosaceptados	misanunciosaceptados.html	Diseño de la vista, apartado de solicitudes: con lista de solicitudes recibidas y opciones de las mismas; apartado de solicitudes aceptadas: con lista de aceptados y opciones de las mismas; visualización de usuario y estrellas.
	misanunciosaceptados.ts	Importación de servicios, suscripción a eventos para visualizar en tiempo real las solicitudes y aceptados, programación de las funcionalidades de cada opción disponible.
mistareas	mistareas.html	Diseño de la vista administración de anuncios, lista de anuncios publicados con información resumida y botón opciones.
	mistareas.ts	Importación de servicio, suscripción a event de mis anuncios y programación de las funcionalidades de las opciones disponibles.
passwordreset	passwordreset.html	Diseño del formulario de recuperación de contraseña.
	passwordreset.scss	Estilos del formulario.
	passwordreset.ts	Validación de credencial email.
perfil	perfil.html	Diseño de la página, mostrar información del usuario para ser modificada.
	perfil.scss	Estilos de visualización de avatar del usuario.
	perfil.ts	Importación de servicios requeridos, cargar datos del usuario y programar cada funcionalidad de edición.
perfilcurriculum	perfilcurriculum.html	Formulario de llenado de información complementaria al perfil, con edición dinámica.

	perfilcurriculum.ts	Funcionalidad de carga de usuario y captura de datos del formulario para ser guardado.
perfilepic	perfilepic.html	Diseño de la opción de agregar perfil al usuario.
	perfilepic.scss	Estilos para visualización responsive del avatar.
	perfilepic.ts	Importación de servicios, programación de funcionalidades para el avatar y acceso a la galería del dispositivo.
perfilpresentacion	perfilpresentacion.html	Diseño del Modal para la presentación del perfil del usuario, con carga de comentarios de anuncios.
	perfilpresentacion.scss	Estilos para la visualización de la información.
	perfilpresentacion.ts	Importación de servicio, obtención de estrellas y comentarios.
personachat	personachat.html	Diseño de la sala de chats, envío y recepción de mensajes.
	personachat.scss	Estilos para la visualización de los mensajes y de las imágenes enviadas.
	personachat.ts	Importación de servicios, suscripción a evento para la conversación en tiempo real, obtención y carga de mensajes.
personas	personas.html	Diseño del formulario para la búsqueda de personas y visualización en lista de personas.
	personas.ts	Importación de servicios, implementación de carga y búsqueda de personas, opción de envío de solicitud de amistad.
signup	signup.html	Diseño del formulario de registro de usuario.
	signup.scss	Estilo para visualización del formulario.
	signup.ts	Importación de servicios, captura de datos del formulario, obtención de

		ubicación, validación y registro de datos, obtención de información del dispositivo.
tabmistareas	tabmistareas.html	Diseño y definición de los tabs.
	tabmistareas.ts	Variables de los tabs definidos.
tareadetalle	tareadetalle.html	Diseño de la vista detalle del anuncio, visualización de información del publicante y del anuncio, opciones adicionales.
	tareadetalle.scss	Estilo para la visualización de la información.
	tareadetalle.ts	Importación de servicios, carga de información de usuario y el respectivo anuncio, obtención de estrellas y programación de la función postular.

Fuente: Elaboración Propia

Imagen 10. Carpeta "src/providers"



Fuente: Elaboración Propia

Tabla 62. Carpeta "src/providers"

auth	auth.ts	Importa la interfaz de usuario, implementa el método de login para validar las credenciales del usuario.
chat	chat.ts	Implementa los métodos de obtención de mensajes y enviar nuevo mensaje tanto para el chat en amigos, así como para el chat entre anunciante y postulante con referencia a un anuncio.

geolocation	geolocation.ts	Retorna la posición actual del dispositivo.
Imghandler	imghandler.ts	Provider o servicio que se encarga de todas las funciones referentes a imágenes, como la obtención y subida de imágenes; implementa el algoritmo de optimización de manipulación de imágenes.
notificaciones	notificaciones.ts	Servicio que implementa la gestión de tokens (identificador único) de la aplicación móvil con referencia a un determinado dispositivo.
solicitudes	solicitudes.ts	Servicio que implementa todos los métodos de gestión de obtención de solicitudes y aceptación de las mismas
tareas	tareas.ts	Servicio que implementa todos los métodos de gestión de los anuncios, así como la obtención de la información de cada uno de ellos y el envío a la página principal.
usuario	usuario.ts	Servicio que implementa todos los métodos de gestión de usuarios de la aplicación móvil.

Fuente: Elaboración Propia

4.2.4. Fase de Estabilización

Se procede a integrar las funcionalidades implementadas en diferentes módulos y de presentarse algún error se realizaran las correcciones de codificación correspondientes. El propósito de esta fase es asegurar la calidad de la implementación del proyecto.

El punto más importante es esta fase es la integración de la aplicación móvil con la plataforma Firebase para la conexión con la Base de Datos en Tiempo Real.

Esta fase se lleva a cabo cada vez que se complete una nueva iteración de la fase anterior y para ello se realizó las siguientes actividades:

- ✓ Planear el proceso de integración.

- ✓ Integrar los módulos, procesos o sub-sistemas.
- ✓ Realizar pruebas de integración.
- ✓ Integra la iteración con la rama principal.

4.2.4.1. Integración del proyecto con Firebase

La aplicación móvil tiene como participante principal a Firebase ya que proporciona las funcionalidades necesarias para hacer una aplicación móvil híbrida altamente funcional.

Para la integración correctamente, se procederá de la siguiente manera:

- ✓ **Primero:** haber instalado npm en la terminal y tener disponible el comando en la terminal.
- ✓ **Segundo:** instalar angularfire2 y firebase con el siguiente comando:

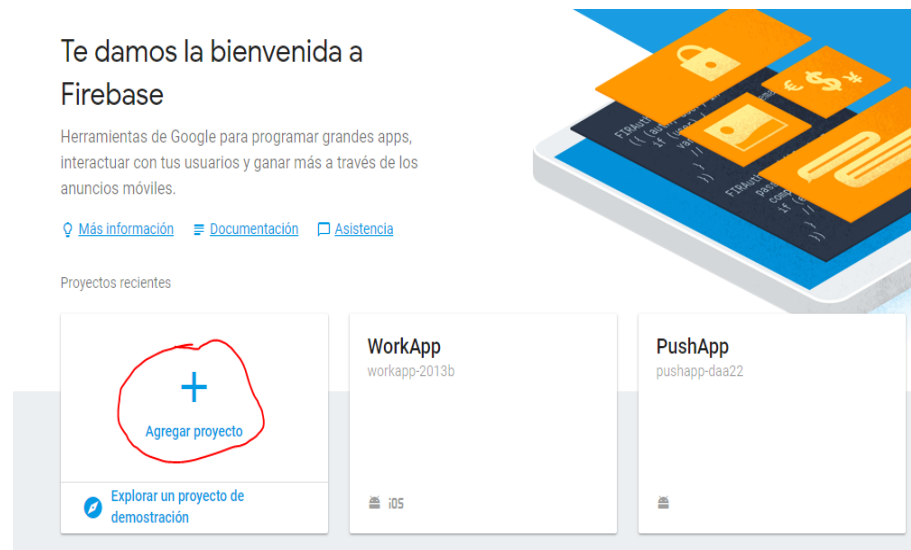
Imagen 11. Install angularfire2 firebase

```
npm install angularfire2 firebase --save
```

Fuente: Elaboración Propia

- ✓ **Tercero:** acceder a la consola de firebase y crear un nuevo proyecto.

Imagen 12. Nuevo Proyecto en Firebase



Fuente: Elaboración Propia

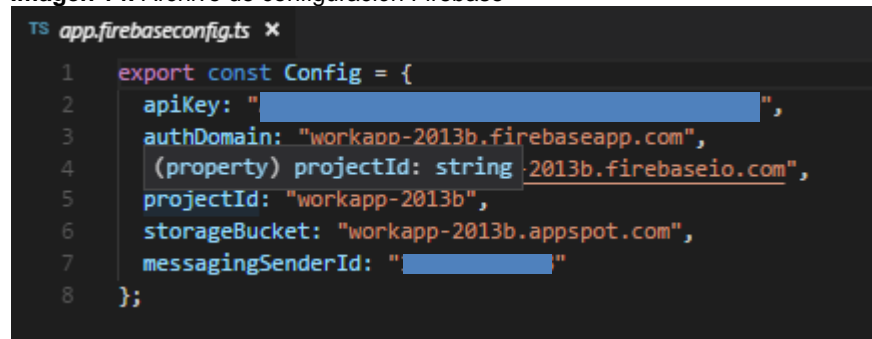
- ✓ **Cuarto:** copiar el script de configuración proporcionado por firebase a nuestro archivo de configuración, solo lo que contiene la variable config.

Imagen 13. Script de configuración Firebase



Fuente: Elaboración Propia

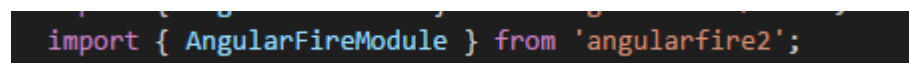
Imagen 14. Archivo de configuración Firebase



Fuente: Elaboración Propia

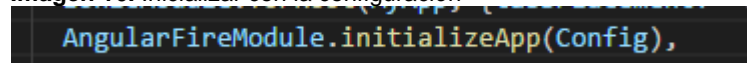
- ✓ **Quinto:** importar e inicializar con la configuración en app.modules.ts.

Imagen 15. Importar AngularFireModule



Fuente: Elaboración Propia

Imagen 16. Inicializar con la configuración



Fuente: Elaboración Propia

4.2.4.2. Integración de los módulos de Acceso a la Aplicación

Primero: se establece como página raíz a LoginPage en app.component.ts de la siguiente manera:

Imagen 17. rootPage: Login

```
@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage:any = 'LoginPage';
}
```

Fuente: Elaboración Propia

El formulario de LoginPage está diseñado con las etiquetas de ionic que están diseñadas para la adaptación responsive.

Imagen 18. Formulario de Login con etiquetas de Ionic

```
<ion-content padding >
  <ion-grid>
    <ion-row>
      <ion-col col-8>
        
      </ion-col>
    </ion-row>
  </ion-grid>

  <ion-list class="list-form" >
    <ion-item >
      <ion-icon color="light" name="person" item-left</ion-icon>
      <ion-label stacked>Email:</ion-label>
      <ion-input type="email" placeholder="Email" [(ngModel)]="credencial.email"></ion-input>
    </ion-item>
    <ion-item >
      <ion-icon color="light" name="key" item-left</ion-icon>
      <ion-label stacked>Contraseña:</ion-label>
      <ion-input type="password" placeholder="Password" [(ngModel)]="credencial.password"></ion-input>
    </ion-item>
  </ion-list>

  <button ion-button color="primary" block (click)="signIn()">Login</button>

  <div text-center (click)="passwordReset()">¿Olvidaste tu Contraseña?</div>
  <button ion-button block round (click)="signUp()">Regístrate</button>

  <button ion-button block round (click)="verAnuncios()">¿Quieres ver los anuncios?</button>
</ion-content>
```

Fuente: Elaboración Propia

Dentro del archivo login.ts se da acceso a los demás formularios que forman parte de esta integración (Signup, Passwordreset, Inicialogin).

Imagen 19. Login.ts

```
    signUp(){
      this.navCtrl.push('SignupPage');
    }

    passwordReset(){
      this.navCtrl.push('PasswordresetPage');
    }

    verAnuncios(){
      this.navCtrl.push('InicialoginPage');
    }
  }
}
```

Fuente: Elaboración Propia

4.2.4.3. Integración de los módulos de la Página Principal

Una vez autenticado el usuario con sus respectivas credenciales se inicia la Página Principal con un event para la visualización de anuncios publicados en Tiempo Real, esto incluye el acceso a crear un nuevo anuncio, visualizar el anuncio en detalle, ver mapa, ver perfil y la acción postular a un anuncio que está dentro del detalle del anuncio.

Imagen 20. Página Principal – métodos.

```
    agregarTarea(){
      this.navCtrl.push('AgregartareaPage');
    }

    tareaDetalle(tarea){
      let modal = this.modalControl.create(TareadetallePage, {
        tareaDetalle: tarea
      });
      modal.present();

      modal.onDidDismiss(data => console.log(data));
    }

    verMapa(anuncio){
      if(anuncio.lat && anuncio.lng){
        this.navCtrl.push('MapanuncioPage', anuncio);
      }
      else{
        alert('El Anuncio no tiene Ubicación!')
      }
    }
  }
}
```

Fuente: Elaboración Propia

El módulo de agregar un nuevo anuncio tiene incluido el algoritmo de optimización de gestión de imágenes tomadas con la cámara el mismo que hace uso de las siguientes importaciones:

Imagen 21. Agregar Nuevo Anuncio import

```
TS agregarTarea.ts x
1 import { Component } from '@angular/core';
2 import { IonicPage, NavController, NavParams, AlertController, LoadingController } from 'ionic-angular';
3 import { GeolocationProvider } from '../providers/geolocation/geolocation';
4 import { Camera, CameraOptions, Device } from 'ionic-native';
5 import { TareasProvider } from '../providers/tareas/tareas';
6 import { ImghandlerProvider } from '../providers/imghandler/imghandler';
7 import { NotificacionesProvider } from '../providers/notificaciones/notificaciones';
8
```

Fuente: Elaboración Propia

Una vez que se toca en un anuncio, esta acción abre un ModalController que contiene toda la información detallada del anuncio, permite ver todo el perfil del usuario que hizo la publicación y lo más importante implementa un método para postular a dicho anuncio, es decir, envía una solicitud de postulación al usuario que publico el anuncio,

Imagen 22. Método Postular Anuncio

```
postularAnuncio(anuncio){
  this.nuevaSolicitud.postulante = firebase.auth().currentUser.uid;
  this.nuevaSolicitud.usuarioAnuncio = this.Usuario.uid;
  this.nuevaSolicitud.anuncio = anuncio.id;
  if(this.nuevaSolicitud.postulante == this.nuevaSolicitud.usuarioAnuncio)
    alert('Este es tu Anuncio!');
  else{
    let successAlert = this.alertControl.create({
      title: 'Solicitud Enviada',
      subTitle: ' Tu solicitud fue enviada a ' + this.Usuario.displayName,
      buttons: ['Ok']
    })
  }

  this.solicitudService.enviarSolicitudAnuncio(this.nuevaSolicitud).then((res: any) => {
    if(res.success){
      successAlert.present();
    }
  }).catch((err) => {
    alert(err);
  })
}
```

Fuente: Elaboración Propia

Así mismo la parte que integra todos los módulos de la aplicación para el usuario autenticado es el Menu Toggle.

Imagen 23. Menu Toggle Principal

```
public rootPage:any;
public paginas: Array<{titulo: string, component: any, icon: string}>;

constructor(public navCtrl1: NavController, public navParams: NavParams) {

  this.rootPage = InicioPage;
  this.paginas = [
    {titulo: 'Inicio',    component: InicioPage,    icon: 'home'},
    {titulo: 'Administrar',component: TabmistareasPage, icon: 'bookmarks'},
    {titulo: 'Perfil',    component: PerfilPage,    icon: 'person'},
    {titulo: 'Chats',     component: TabsPage,     icon: 'contacts'},
    {titulo: 'Acerca De', component: AcercadePage, icon: 'information-circle'}
  ];
}
```

Fuente: Elaboración Propia

4.2.4.4. Integración de los módulos para la Administración de Anuncios

Empieza con la creación de Pestañas que integran los módulos de administrar mis anuncios, gestión de solicitudes y aceptados, contrato y evaluación, estos tabs o pestañas activan una interfaz de pestañas múltiples en una barra de pestañas que hacen llamado a los Pages, mediante variables estáticas definidas.

Imagen 24. Tabs Mis Anuncios html

```
<ion-tabs color="hcolor">
  <ion-tab [root]="tab1" tabTitle="Mis Anuncios" tabIcon="bookmarks" tabsHideOnSubPages="true"></ion-tab>
  <ion-tab [root]="tab2" tabTitle="Recibidos" tabIcon="book"></ion-tab>
  <ion-tab [root]="tab3" tabTitle="Puntuación" tabIcon="star"></ion-tab>
</ion-tabs>
```

Fuente: Elaboración Propia

Imagen 25. Tabs Mis Anuncios ts

```
export class TabmistareasPage {
  tab1: string = "MistareasPage";
  tab2: string = "MisanunciosaceptadosPage";
  tab3: string = "MisamigoscontratadosPage";
}
```

Fuente: Elaboración de Propia

4.2.4.5. Integración de los módulos de Configuración de Perfil

Los módulos del Perfil de usuario se encargan de la gestión de los datos del usuario permitiendo actualización, modificación, edición y ampliación de la información, los cuales se cargan en las variables definidas en el archivo .ts

Imagen 26. Perfil – variables

```
TS perfil.ts x
1 import { Component, NgZone } from '@angular/core';
2 import { IonicPage, NavController, NavParams, AlertController, ModalController } from 'ionic-angular';
3 import { ImghandlerProvider } from '../../providers/imghandler/imghandler';
4 import { UsuarioProvider } from '../../providers/usuario/usuario';
5 import { GeolocationProvider } from '../../providers/geolocation/geolocation';
6 import { PerfilpresentacionPage } from '../../pages/perfilpresentacion/perfilpresentacion';
7 import { Camera, CameraOptions, Device } from 'ionic-native';
8 import firebase from 'firebase';
9
10 @IonicPage()
11 @Component({
12   selector: 'page-perfil',
13   templateUrl: 'perfil.html',
14 })
15 export class PerfilPage {
16   Persona = [];
17   avatar: string;
18   displayName: string;
19   apellido: string;
20   telefono: string;
21   ciudad: string;
22   geolocalizacion: boolean = false;
23   conEstrella: boolean = false;
24   Estrellas;
25   semiEstrellas;
26
```

Fuente: Elaboración Propia

4.2.4.6. Integración de los módulos de la Sala de Chats

La sala de chats gestiona las solicitudes de amistad y a los aceptados; administra las conversaciones de cada usuario y toda la visualización en la página está suscrita a eventos conectados en tiempo real.

Imagen 27. Sala de chats amigos

```

agregarPersona(){
  this.navCtrl.push('PersonasPage');
}

aceptar(solic){
  this.solicitudService.aceptarSolicitud(solic).then(() => {
    let alert = this.alertControl.create({
      title: 'Amigo agregado',
      subTitle: 'Toque para chatear con él',
      buttons: ['Ok']
    });
    alert.present();
  })
}

ignorar(solic){
  this.solicitudService.eliminarSolicitud(solic).then(() => {
  }).catch((err) => {
    alert(err);
  });
}

amigoChat(person){
  this.chatService.inicializarChat(person);
  this.navCtrl.push('PersonachatPage');
}

```

Fuente: Elaboración Propia

4.2.4.7. Integración de los Providers (Servicios)

Todo Provider o Service (Servicio) que se implemente debe de importar “Injectable”, el cual permite que el servicio implementado y sus métodos y variables públicas si es que las tuviera de acuerdo a las necesidades de la aplicación o del programador para resolver algunas complicaciones, estén disponibles para ser llamados.

Imagen 28. Autenticación.ts

```

TS auth.ts x
1 import { Injectable } from '@angular/core';
2 import { AngularFireAuth } from 'angularfire2/auth';
3 import { usercreds } from '../../models/interfaces/usercreds';
4
5 @Injectable()
6 export class AuthProvider {
7

```

Fuente: Elaboración Propia

Imagen 29. Chat.ts

```
TS chat.ts ●
1  import { Injectable } from '@angular/core';
2  import firebase from 'firebase';
3  import { Events } from 'ionic-angular';
4
5
6  @Injectable()
7  export class ChatProvider {
8      firePersonaChats = firebase.database().ref('/personachats');
9      personaChat: any;
10
11     fireAnuncioChats = firebase.database().ref('/anunciochats');
12     anuncioChat: any;
13
```

Fuente: Elaboración Propia

Imagen 30. Geolocation.ts

```
TS geolocation.ts ✕
1  import { Injectable } from '@angular/core';
2  import { Geolocation } from 'ionic-native';
3
4  @Injectable()
5  export class GeolocationProvider {
6
```

Fuente: Elaboración Propia

Imagen 31. Imghandler.ts

```
TS imghandler.ts ✕
1  import { Injectable } from '@angular/core';
2  import { File } from '@ionic-native/file';
3  import { FileChooser } from '@ionic-native/file-chooser';
4  import { FilePath } from '@ionic-native/file-path';
5  import { Camera, Device } from 'ionic-native';
6  import firebase from 'firebase';
7
8
9  @Injectable()
10 export class ImghandlerProvider {
11     nativepath: any;
12     firestore = firebase.storage();
13
```

Fuente: Elaboración Propia

Imagen 32. Notificaciones.ts

```

TS notificaciones.ts ×
1  import { Injectable } from '@angular/core';
2  import { AngularFireDatabase } from 'angularfire2/database';
3
4  import firebase from 'firebase';
5
6
7  @Injectable()
8  export class NotificacionesProvider {
9      fireAppTokens = firebase.database().ref('/apptokens');
10

```

Fuente: Elaboración Propia

Imagen 33. Solicitudes.ts

```

TS solicitudes.ts ×
1  import { Injectable } from '@angular/core';
2  import { Events } from 'ionic-angular';
3  import { solicitudreq } from '../../models/interfaces/solicitud';
4  import { solicitudanuncio } from '../../models/interfaces/solicitudanuncio';
5  import { UsuarioProvider } from '../usuario/usuario';
6  import { TareasProvider } from '../tareas/tareas';
7  import firebase from 'firebase';
8
9
10 @Injectable()
11 export class SolicitudesProvider {
12     fireSolicitudAnuncio = firebase.database().ref('/solicitudanuncios');
13     fireAnunciosAceptados = firebase.database().ref('/anunciosaceptados');
14     fireSolicitud = firebase.database().ref('/solicitudes');
15     fireAmigos = firebase.database().ref('/amigos');

```

Fuente: Elaboración Propia

Imagen 34. Tareas.ts

```

TS tareas.ts ×
1  import { Injectable } from '@angular/core';
2  import { Events } from 'ionic-angular';
3  import { ImghandlerProvider } from '../imghandler/imghandler';
4  import firebase from 'firebase';
5
6  @Injectable()
7  export class TareasProvider {
8      fireTareas = firebase.database().ref('/tareas');
9      userTarea: any;

```

Fuente: Elaboración Propia

Imagen 35. Usuarios.ts


```
TS usuario.ts x
1 import { Injectable } from '@angular/core';
2 import { AngularFireAuth } from 'angularfire2/auth';
3 import firebase from 'firebase';
4
5 @Injectable()
6 export class UsuarioProvider {
7
8     userDB = firebase.database().ref('/usuarios');
9 }
```

Fuente: Elaboración Propia

4.2.4.8. Estructura de la base de datos en Firebase

La base de datos proporcionada por Firebase no es relacional, es NoSQL. Los datos en la plataforma Firebase Realtime Database se guardan y almacenan como objetos JSON.

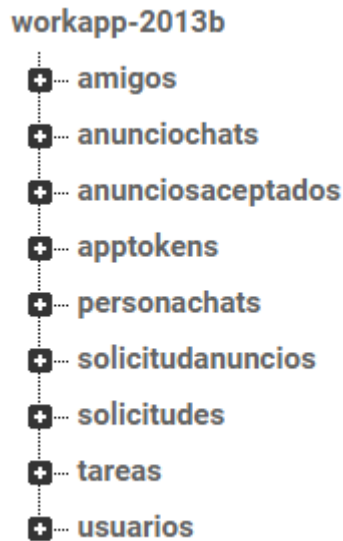
Una base de datos no relacional se define como un árbol JSON alojado en la nube, en diferencia de una base de datos relacional SQL, no hay registros ni tablas. Cuando se le agrega datos al árbol JSON, estos se transforman en una rama de la estructura JSON, con una clave (ID) asociada.

La creación y estructuración de esta base de datos de forma adecuada requirió bastante previsión, la parte más importante fue planificar cómo se guardarán y recuperarán los datos y que el proceso sea lo más simple posible.

Esta base de datos es altamente escalable ya que esta es una característica resaltante de este tipo de bases de datos.

La estructura que se usa se muestra en la imagen:

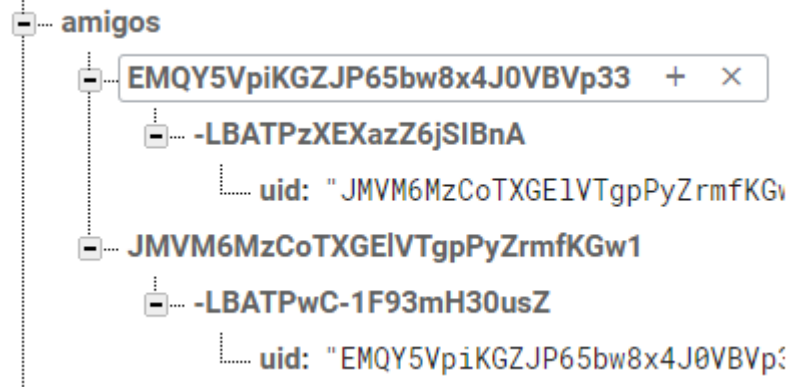
Imagen 36. WorkApp Database



Fuente: Elaboración Propia

La rama JSON “amigos” almacena los ids de los usuarios que son amigos, es decir, que se envió previamente una solicitud de amistad que fue aceptada.

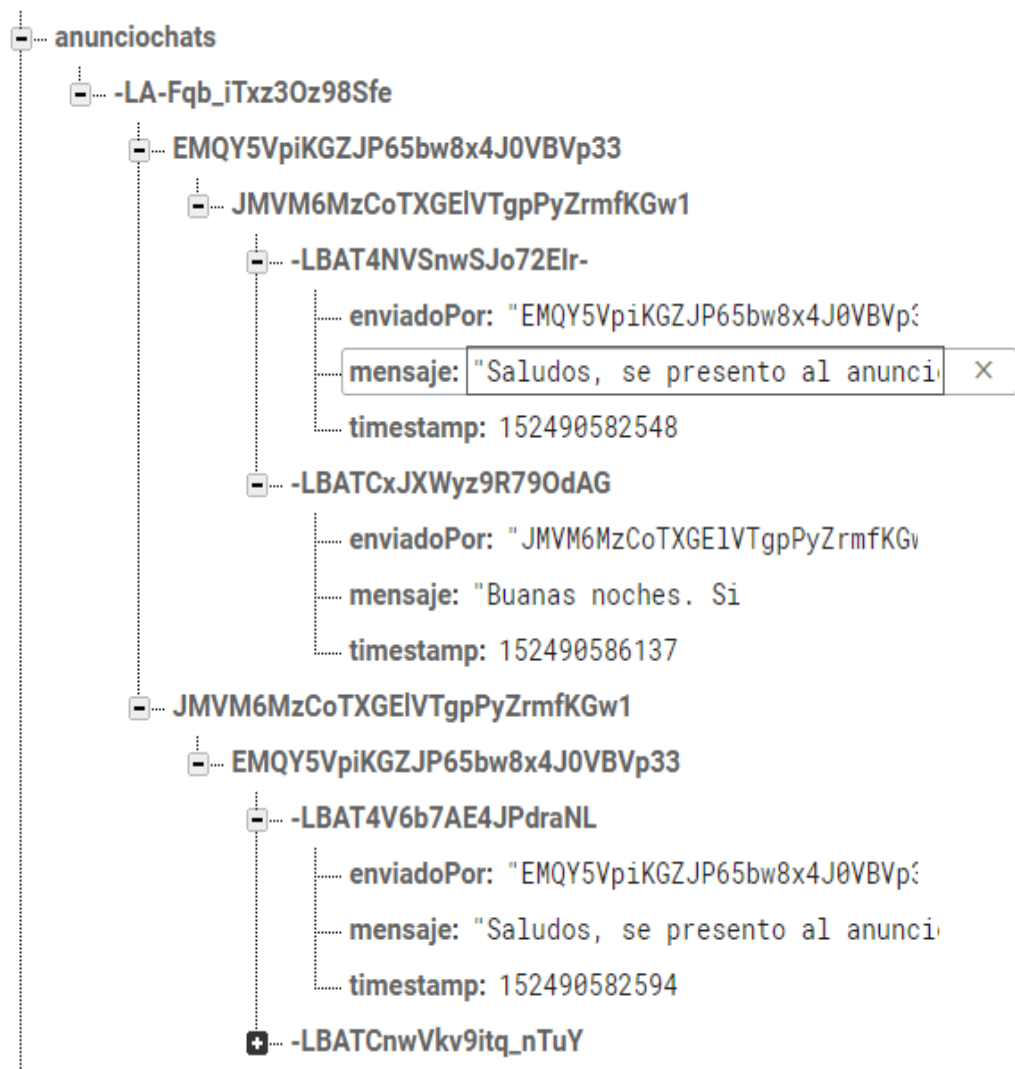
Imagen 37. Database – amigos



Fuente: Elaboración Propia

La rama JSON “anunciochats” almacena los mensajes enviados de las conversaciones entre anunciante y postulante con respecto al anuncio en cuestión.

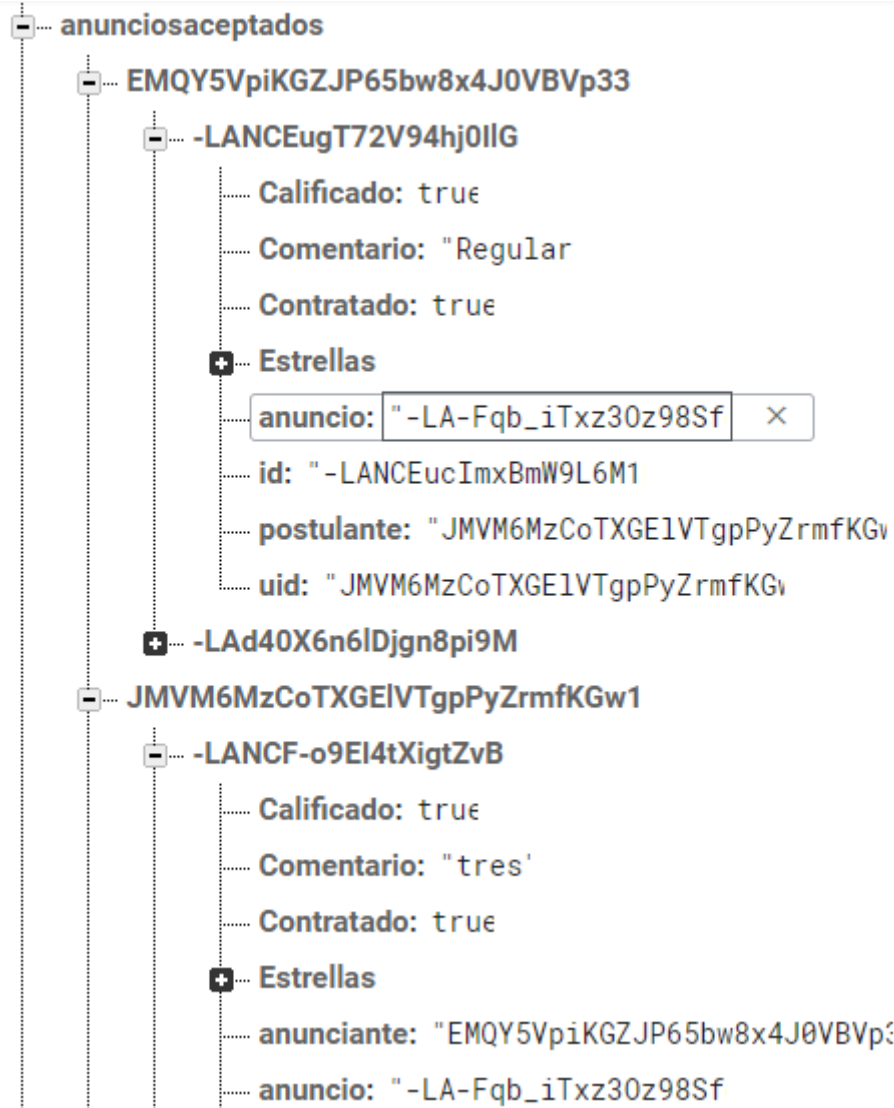
Imagen 38. Database – anunciochats



Fuente: Elaboración Propia

La rama JSON “anunciosaceptados” almacena los datos de los anuncios que fueron aceptados para contactar, previa solicitud de postulación, almacena los estados del anuncio en cuestión.

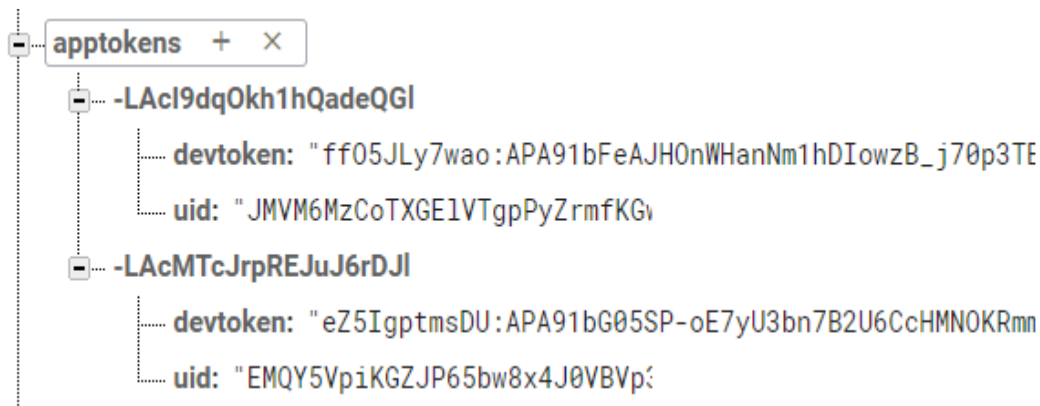
Imagen 39. Database – anunciosaceptados



Fuente: Elaboración Propia

La rama JSON “apptokens” almacena el token (identificador único de la app que se usa en un determinado dispositivo) y el id del usuario que usa esa aplicación instalada en ese Smartphone.

Imagen 40. Database – apptokens



Fuente: Elaboración Propia

La rama JSON “personachats” almacena los mensajes de las conversaciones entre dos amigos en la sala de chats.

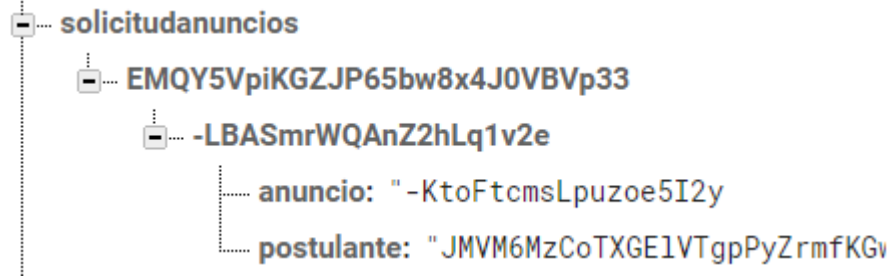
Imagen 41. Database – personachats



Fuente: Elaboración Propia

La rama JSON “solicitudanuncios” almacena los ids del anuncio y del usuario postulante a dicho anuncio.

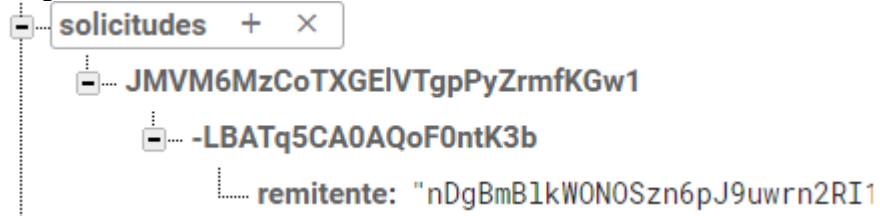
Imagen 42. Database – solicitudanuncios



Fuente: Elaboración Propia

La rama JSON “solicitudes” almacena las solicitudes de amistad enviadas desde la sala de chats, registra los ids de ambos usuarios.

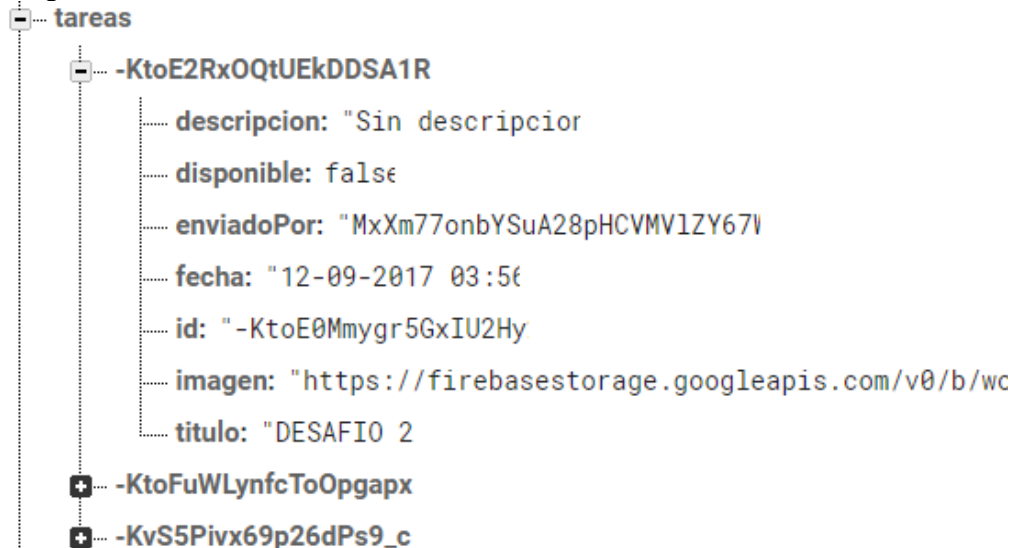
Imagen 43. Database – solicitudes



Fuente: Elaboración Propia

La rama JSON “tareas” almacena toda la información de todos los anuncios publicados.

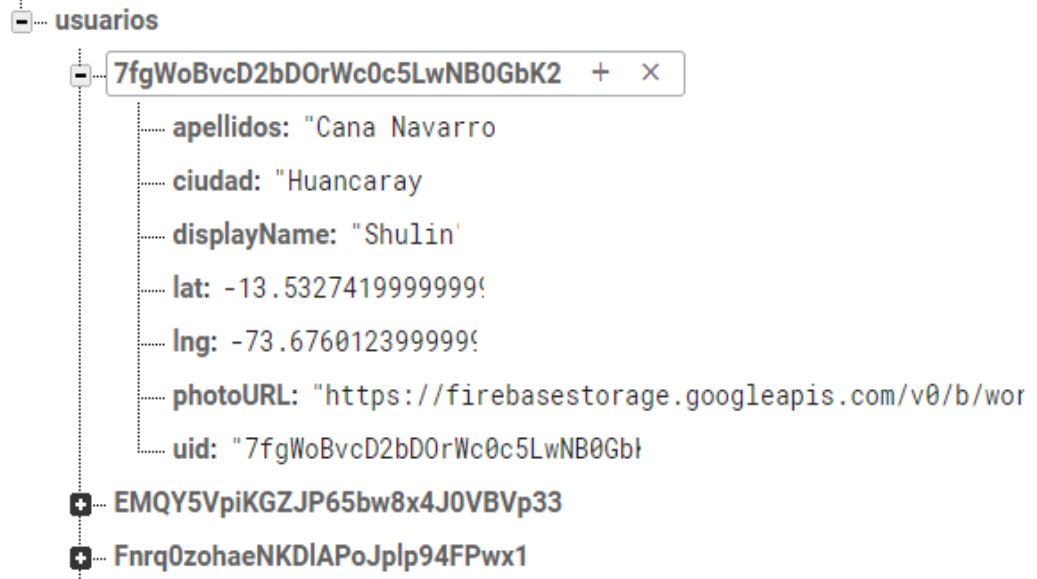
Imagen 44. Database – tareas



Fuente: Elaboración Propia

La rama JSON “usuarios” almacena toda la información del usuario que se registró en la aplicación móvil.

Imagen 45. Database – usuarios



Fuente: Elaboración Propia

Existe una sección aparte llamada “Authentication” en el cual están las credenciales de los usuarios que se registraron en la aplicación móvil, su email y contraseña, de esta manera se hace más seguro el inicio de sesión de cada usuario.

Imagen 46. Authentication

The image shows a screenshot of the Authentication user list interface. At the top, there is a search bar with the text 'Buscar por dirección de correo electrónico, número de teléfono o UID de usuario' and a blue button labeled 'AGREGAR USUARIO'. Below the search bar is a table with the following columns: 'Identificador', 'Proveedores', 'Creado', 'Accediste a tu cuenta', and 'UID de usuario ↑'. The table contains four rows of user data:

Identificador	Proveedores	Creado	Accediste a tu cuenta	UID de usuario ↑
shulin@test.com	✉	5 sep. 2017	19 sep. 2017	7fgWoBvcD2bD0rWc0c5LwNB0Gb...
acer@test.com	✉	22 ago. 2017	28 abr. 2018	EMQY5VpiKGZJP65bw8x4J0VBVp...
i@r.com	✉	31 ago. 2017	31 ago. 2017	Fnrq0zohaeNKDIAPoJlp94FPwx1
acer3@test.com	✉	14 abr. 2018	28 abr. 2018	JMVM6MzCoTXGEIVTgpPyZrmfK...

Fuente: Elaboración Propia

También se tiene un lugar para guardar las imágenes llamado “Storage” en el cual se guardan todas las imágenes como: imágenes del chat, del perfil y de los anuncios, en sus respectivas carpetas designadas.

Imagen 47. Storage

<input type="checkbox"/>	Nombre	Tamaño	Tipo	Última modificación
★	Según las reglas de seguridad predeterminadas, los usuarios deben autenticarse			MÁS INFORMACIÓN IGNORAR
<input type="checkbox"/>	imagenmsgs/	–	Carpeta	–
<input type="checkbox"/>	imagenmsgsanuncio/	–	Carpeta	–
<input type="checkbox"/>	perfilimagen/	–	Carpeta	–
<input type="checkbox"/>	sin_perfil.png	23.23 KB	image/png	21 ago. 2017
<input type="checkbox"/>	tareaimagen/	–	Carpeta	–

Fuente: Elaboración Propia

4.2.5. Fase de Pruebas

En esta última fase de la metodología Mobile-D, se realiza un conjunto de pruebas de la aplicación móvil, se identifica y se soluciona algunos errores, esta fase busca entregar una versión completamente estable y funcional de la aplicación móvil. Una vez que se solucionaron todos los errores identificados, recién se puede obtener el entregable final estable.

Al terminar el desarrollo de los módulos de la aplicación móvil se pasó a esta fase de pruebas, donde se iteró, para obtener un módulo funcional y al final de todo se llegó a la versión estable.

Para ello, se llevaron a cabo las siguientes actividades:

- ✓ Detección de defectos de los módulos implementados.
- ✓ Análisis de los defectos y corrección o modificación de los procesos.
- ✓ Solucionar los defectos y pruebas, para el correcto funcionamiento.
- ✓ Integración de la iteración corregida.

4.2.5.1. Errores “undefined” para las notificaciones

En las nuevas versiones del paquete angularfire2 es necesario importar “AngularFireDatabaseModule” para hacer una petición “list” a la base de datos de firebase, esto se importa en app.module.ts, como se muestra en la figura:

Imagen 48. Import AngularFireDatabaseModule

```
12 import { AngularFireAuth } from 'angularfire2/auth';
13 import { AngularFireModule } from 'angularfire2';
14 import { AngularFireDatabaseModule } from 'angularfire2/database';
15
```

Fuente: Elaboración Propia

Imagen 49. Imports AngularFireDatabaseModule

```
57 imports: [
58   BrowserModule,
59   IonicModule.forRoot(MyApp, {tabsPlacement: 'top'}),
60   AngularFireModule.initializeApp(Config),
61   AngularFireDatabaseModule
62 ],
```

Fuente: Elaboración Propia

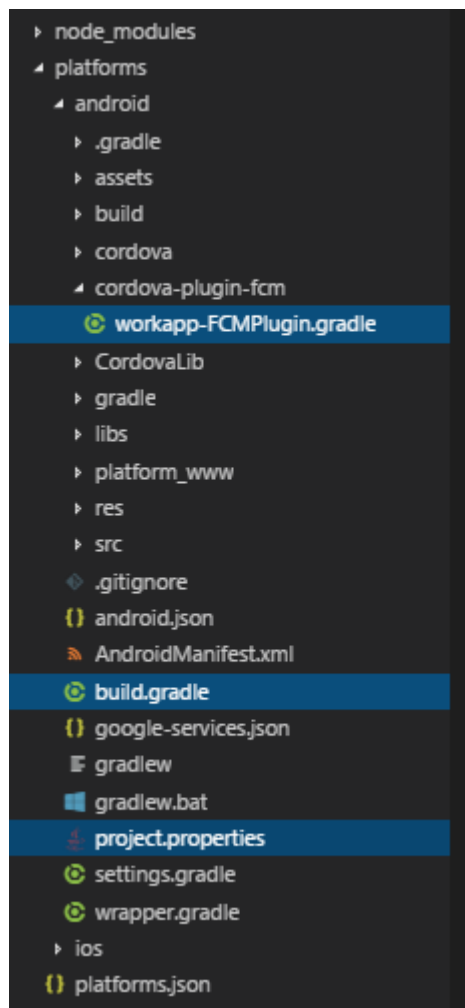
4.2.5.2. Errores de incompatibilidad de plugins al compilar el .APK de Android

Al intentar compilar y generar el .APK del proyecto para la plataforma Android, salió un error inesperado debido a la API de Android, esto sucede porque el proyecto en IONIC al generar los archivos necesarios para la compilación del proyecto, configura los plugins para usar los más actualizados y ahí es donde sucede el error ya que los plugins usados, por ahora están limitados a una versión.

La solución que se dio a este error consiste en editar ciertos archivos de la misma plataforma Android, los cuales son:

- **workapp-FCMPlugin.gradle:** “workapp” es el nombre de la app, cambiará solo esa letra en cada proyecto.
- **build.gradle**
- **project.properties**

Imagen 50. Archivos para modificar



Fuente: Elaboración Propia

Imagen 51. Workapp-FCMPlugin.gradle – Anterior

```

workapp-FCMPlugin.gradle
1  buildscript {
2      repositories {
3          jcenter()
4          mavenLocal()
5      }
6      dependencies {
7          classpath 'com.android.tools.build:gradle:+'
8          classpath 'com.google.gms:google-services:3.0.0'
9      }
10 }
11 // apply plugin: 'com.google.gms.google-services'
12 // class must be used instead of id(string) to be able to apply plugin
13 apply plugin: com.google.gms.google.services.GoogleServicesPlugin

```

Fuente: Elaboración Propia

Imagen 52. Workapp-FCMPlugin.gradle – Corregido

```
workapp-FCMPlugin.gradle x
1  buildscript {
2      repositories {
3          jcenter()
4          mavenLocal()
5      }
6      dependencies {
7          classpath 'com.android.tools.build:gradle:+'
8          // classpath 'com.google.gms:google-services:3.0.0'
9      }
10 }
11 // apply plugin: 'com.google.gms.google-services'
12 // class must be used instead of id(string) to be able to apply plugin
13 //apply plugin: com.google.gms.google-services.GoogleServicesPlugin
```

Fuente: Elaboración Propia

Imagen 53. build.gradle – Anterior 1

```
32      dependencies {
33          classpath 'com.android.tools.build:gradle:2.2.3'
34      }
```

Fuente: Elaboración Propia

Imagen 54. build.gradle – Anterior 2

```
249  dependencies {
250      compile fileTree(dir: 'libs', include: '*.jar')
251      // SUB-PROJECT DEPENDENCIES START
252      debugCompile(project(path: "CordovaLib", configuration: "debug"))
253      releaseCompile(project(path: "CordovaLib", configuration: "release"))
254      compile "com.android.support:support-v4:24.1.1+"
255      compile "com.google.firebase:firebase-core:+"
256      compile "com.google.firebase:firebase-messaging:+"
257      compile "com.google.android.gms:play-services-maps:+"
258      compile "com.google.android.gms:play-services-location:+"
259      // SUB-PROJECT DEPENDENCIES END
260  }
261
262  def promptForReleaseKeyPassword() {
263      if (!cdvReleaseSigningPropertiesFile) {
264          return;
265      }
266  }
```

Fuente: Elaboración Propia

Imagen 55. build.gradle – Corregido 1

```
32      dependencies {
33          classpath 'com.android.tools.build:gradle:2.2.3'
34          classpath 'com.google.gms:google-services:3.1.0'
35      }
```

Fuente: Elaboración Propia

Imagen 56. build.gradle – Corregido 2

```
249 dependencies {
250     compile fileTree(dir: 'libs', include: '*.jar')
251     // SUB-PROJECT DEPENDENCIES START
252     debugCompile(project(path: "CordovaLib", configuration: "debug"))
253     releaseCompile(project(path: "CordovaLib", configuration: "release"))
254     compile "com.android.support:support-v4:24.1.1+"
255     compile "com.google.firebase:firebase-core:9.8.0"
256     compile "com.google.firebase:firebase-messaging:9.8.0"
257     compile "com.google.android.gms:play-services-maps:9.8.0"
258     compile "com.google.android.gms:play-services-location:9.8.0"
259     // SUB-PROJECT DEPENDENCIES END
260 }
261
262 apply plugin: 'com.google.gms.google-services'
263
264 def promptForReleaseKeyPassword() {
265     if (!cdvReleaseSigningPropertiesFile) {
266         return;
267     }
268 }
```

Fuente: Elaboración Propia

Imagen 57. project.properties – Anterior

```
13 target=android-25
14 android.library.reference.1=CordovaLib
15 cordova.system.library.1=com.android.support:support-v4:24.1.1+
16 cordova.system.library.2=com.google.firebase:firebase-core:+
17 cordova.system.library.3=com.google.firebase:firebase-messaging:+
18 cordova.gradle.include.1=cordova-plugin-fcm/workapp-FCMPlugin.gradle
19 cordova.system.library.4=com.google.android.gms:play-services-maps:+
20 cordova.system.library.5=com.google.android.gms:play-services-location:+
```

Fuente: Elaboración Propia

Imagen 58. project.properties – Corregido

```
13 target=android-25
14 android.library.reference.1=CordovaLib
15 cordova.system.library.1=com.android.support:support-v4:24.1.1+
16 cordova.system.library.2=com.google.firebase:firebase-core:9.8.0
17 cordova.system.library.3=com.google.firebase:firebase-messaging:9.8.0
18 cordova.gradle.include.1=cordova-plugin-fcm/workapp-FCMPlugin.gradle
19 cordova.system.library.4=com.google.android.gms:play-services-maps:9.8.0
20 cordova.system.library.5=com.google.android.gms:play-services-location:9.8.0
```

Fuente: Elaboración Propia

4.3. NOTIFICACIONES PUSH

Las Notificaciones Push (Push Notifications) son mensajes que se envían de manera directa a dispositivos móviles (Smartphones y/o tablets) con sistemas

operativos iOS, Android, Windows Phone y otros. Las notificaciones push permiten a los desarrolladores y usuarios de aplicaciones a mantenerse informados, permitiendo una comunicación instantánea con los usuarios.

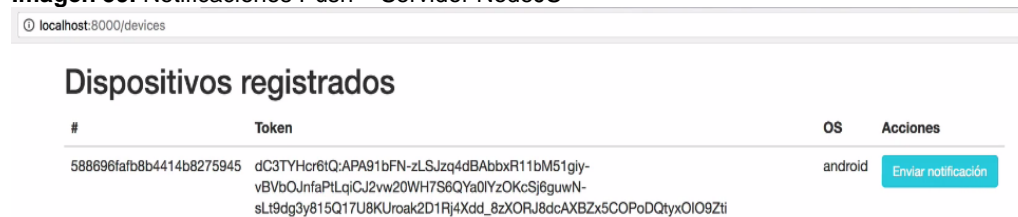
Firebase tiene un servicio denominado FCM (Firebase Cloud Messaging) el cual es una solución de mensajería instantánea multiplataforma que te ayuda a enviar mensajes de manera rápida, segura y gratuita, se puede notificar a una app cliente en particular. (Google Developers, 2017)

4.3.1. Lo que se hacía (Usando un Servidor)

La gran mayoría de aplicaciones móviles por no decir todas, para poder usar las Notificaciones Push hacen uso de un server que se encargue de administrar y gestionar dichas notificaciones, el cual conlleva a un problema, si se piensa en hacer una aplicación móvil (básica) que cuente con esta gran ventaja de las notificaciones, se estaría obligado a crear un servidor solo para implementar dicha función, el cual sería un gasto en recursos no tan aprovechables, al tener que levantar un servidor adicional, y la aplicación no sería tan independiente como debería ser.

En la figura se muestra un servidor que se implementó con NodeJS para las notificaciones push, el cual sería el camino más común por ahora, pero el más complicado debido a que el servidor se debe alojar en la nube.

Imagen 59. Notificaciones Push – Servidor NodeJS



Fuente: Elaboración Propia

4.3.2. Lo que se hizo (Sin Servidor)

El presente proyecto consiste en implementar una aplicación móvil independiente de servidores y la solución para usar notificaciones push, no está en implementar un servidor, para solucionar esto se usó FCM de Firebase.

- **Primero:** instalar el plugin en el proyecto de la aplicación móvil.

Imagen 60. plugin-fcm

```
"cordova": {
  "plugins": {
    "ionic-plugin-keyboard": {},
    "cordova-plugin-filechooser": {},
    "cordova-plugin-filepath": {},
    "cordova-plugin-file": {},
    "cordova-plugin-geolocation": {},
    "cordova-plugin-camera": {},
    "cordova-plugin-sim": {},
    "cordova-plugin-network-information": {},
    "mx.ferreyra.callnumber": {},
    "cordova-plugin-fcm": {}
  }
}
```

Fuente: Elaboración Propia

- **Segundo:** instalar mediante npm Firebase CLI, logearse e iniciar un directorio para el proyecto.

Imagen 61. firebase-tools

```
npm install -g firebase-tools
```

Fuente: Elaboración Propia

Imagen 62. firebase login

```
firebase login
```

Fuente: Elaboración Propia

Imagen 63. Firebase init

```
firebase init
```

Fuente: Elaboración Propia

- **Tercero:** del paso anterior se crea un directorio en el cual hay una subcarpeta denominada functions, el cual tiene un archivo "index.js" en el cual se implementa todas las funciones.

Imagen 64. Directorio functions

```
├─ PUSHWORKAPP
├─ functions
│  └─ node_modules
│     └─ JS index.js
├─ package-lock.json
├─ package.json
├─ .firebaserc
└─ firebase.json
```

Fuente: Elaboración Propia

Imagen 65. Funciones para las Notificaciones Push

```
JS index.js
1  var functions = require('firebase-functions');
2  var admin = require('firebase-admin');
3
4  admin.initializeApp();
5  var datosIngresados;
6  exports.PushtriggerWork = functions.database.ref('/tareas/{tareaId}').onWrite((change, context) => {
7
8      datosIngresados = change.after.val();
9
10     admin.database().ref('/apptokens').orderByChild('uid').once('value').then((alltokens) => {
11         var tokensExistentes = alltokens.val();
12         var tokens = [];
13         processtokens(tokensExistentes).then((tokensProcesados) => {
14
15             for (var token of tokensProcesados) {
16                 tokens.push(token.devtoken);
17             }
18
19             var payload = {
20
21                 "notification":{
22                     "title": 'Hay un Nuevo Anuncio',
23                     "body": 'Míralo ahora...',
24                     "sound": "default",
25                 },
26                 "data":{
27                     "titulo": 'Hay un Nuevo Anuncio',
28                     "mensaje": 'Míralo ahora...',
29                 }
30             }
31             return admin.messaging().sendToDevice(tokens, payload).then((response) => {
32                 console.log('Notificación push nuevo anuncio');
33             }).catch((err) => {
34                 console.log(err);
35             })
36         })
37     })
38 })
39
```

Fuente: Elaboración Propia

- **Cuarto:** subir las funciones a firebase y probar su correcto funcionamiento.

Imagen 66. Firebase deploy

```
firebase deploy --only functions:
```

Fuente: Elaboración Propia

Imagen 67. Firebase Functions

Función	Evento	Ejecuciones	Mediana de tiempo de ejecución
PushtriggerAceptado	ref.write /anunciosaceptados/(acceptadold)/(variableId)	33	68.54 ms
PushtriggerContratado	ref.write /anunciosaceptados/(acceptadold)/(variableId)	24	270.96 ms
PushtriggerSolicitud	ref.create /solicitudanuncios/(solicitudid)	22	274.96 ms
PushtriggerWork	ref.write /tareas/(tareaId)	7	304.96 ms

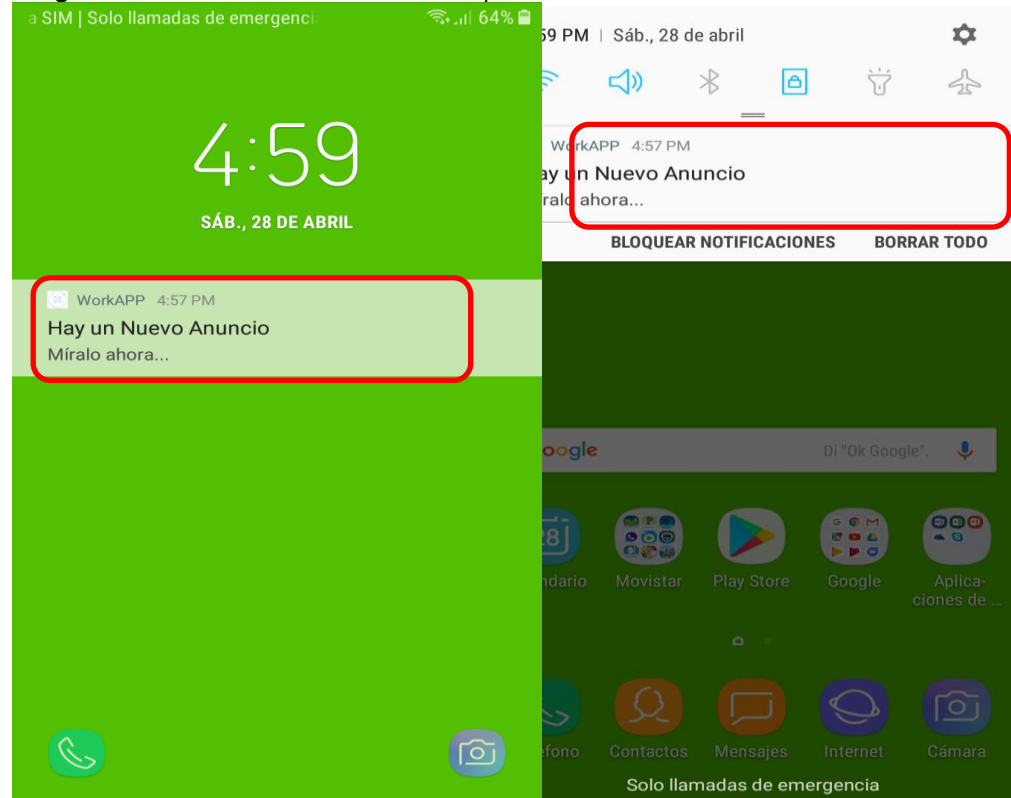
Fuente: Elaboración Propia

4.3.3. Resultados de las Notificaciones Push

Las Notificaciones Push funcionan correctamente, se logró conseguir que la aplicación móvil sea independiente de servidores y haga notificaciones automáticas a los demás usuarios con aplicaciones instaladas en su dispositivo por medio de los tokens.

En la imagen se observa el resultado de la implementación de las notificaciones.

Imagen 68. Notificaciones Push en el Smartphone



Fuente: Elaboración Propia

4.4. REDUCCIÓN DEL MODELO CLIENTE SERVIDOR

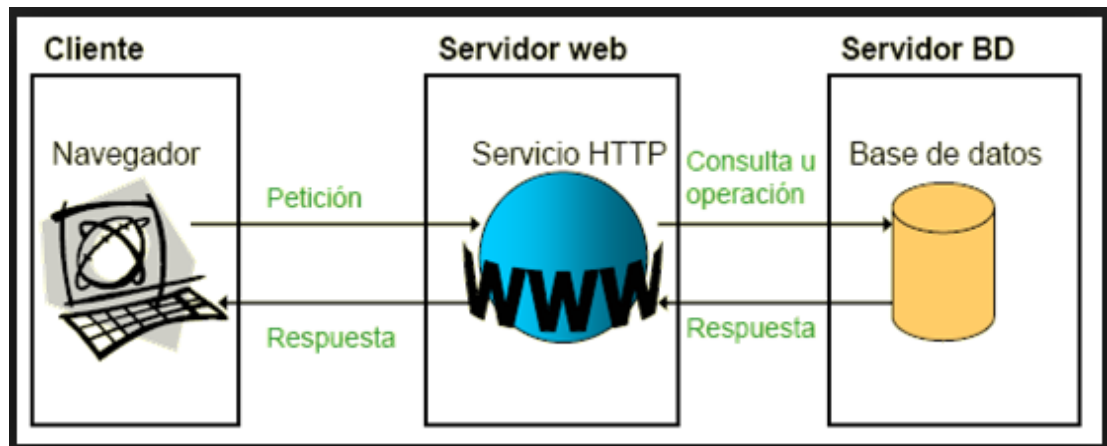
4.4.1. Modelo Cliente Servidor

Las páginas web o las aplicaciones tienen mucha interacción con la nube, esta interacción es posible gracias a un servidor alojado en la nube que cumple con las funciones de recibir, manipular, administrar, gestionar, direccionar, redireccionar y enviar peticiones y respuestas, así mismo se encargan de gestionar los datos de la base de datos de acuerdo a las peticiones que le hagan.

Cuando se crea una página web o una aplicación, para que sea más funcional y se pueda tener todos los datos disponibles e interactuar con los usuarios del mundo exterior, porque en la actualidad las personas buscan interactuar entre ellas por medio de las aplicaciones; para ello los datos se van actualizando constantemente, las aplicaciones pueden almacenar datos en los dispositivos pero esto solo es para esa aplicación y para ese usuario, lo que se necesita es tener una base de datos para todos los usuarios, la solución a ello es tener una base de datos disponible en la nube (internet), para que todas las aplicaciones peticionen a la base de datos, esto se implementa con un Servidor de Base de Datos, que cumple la función de almacenar datos, la aplicación no puede interactuar directamente con la base de datos, se necesita un programa encargado de gestionar y administrar la base de datos, el mismo que implementa todas las funcionalidades del servicio como un todo, a esto se le conoce como un Servidor Web, y es así como se implementó este modelo.

En la imagen se muestra el procesamiento que se hace en el modelo cliente – servidor, el cual consta de **dos pasos (Cliente – Servidor web – servidor BD)**:

Imagen 69. Modelo Cliente –Servidor



Fuente: <https://sites.google.com/site/redesordenadoresgrupoc/home/arquitectura-cliente-servidor>

4.4.2. Modelo Cliente Servidor Óptimo

Para este tipo de aplicaciones como el que se hizo, el modelo cliente servidor no es tan bueno como se quisiera, aprovechando las funcionalidades de los frameworks que se usaron al máximo, se ha obtenido un modelo más práctico y rápido, que es lo que se buscaba.

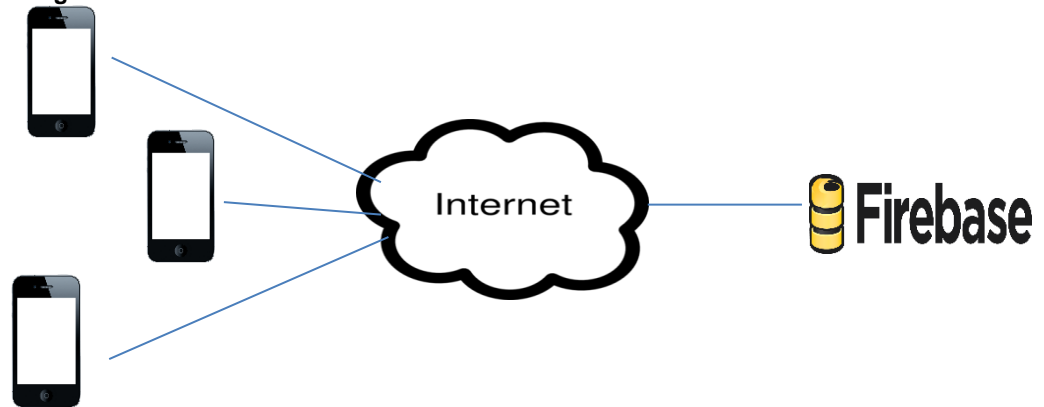
En el modelo optimizado ya no existe un servidor web, solo está el cliente y la base de datos, ¿Cómo es esto posible? Es posible gracias a Firebase Realtime Database que es una base de datos en tiempo real, es decir que informa las actualizaciones constantemente a la aplicación y gracias al manejo de eventos de IONIC se puede tener múltiples conexiones del tipo publish – subscribe, ello permite la creación de funcionalidades como el chat y otras que necesiten ser actualizadas constantemente en la interfaz; también es gracias al manejo de Providers de IONIC.

Al quitar el servidor web, la pregunta sería, ¿Dónde se encuentra la lógica de negocio? Ionic implementa por defecto el Modelo Vista Controlador (MVC) de manera estructurada, en el cual se separa la vista o interfaz del cliente de la lógica y el modelo de negocio.

En el modelo Cliente – Servidor se tiene tres capas, en el desarrollo de este proyecto, para cubrir las necesidades y requerimientos de la aplicación, se consiguió un modelo de dos capas (Cliente -ServidorBD).

En la imagen se muestra el modelo óptimo conseguido que funciona muy bien para aplicaciones como la del proyecto en cuestión y muchas otras, el cual costa de un **paso (Aplicación – Firebase BD)**.

Imagen 70. Modelo Cliente - Base de Datos



Fuente: Elaboración Propia

4.5. MEJORA Y OPTIMIZACIÓN DEL CÓDIGO

La mejora y optimización de los módulos implementados se llevaron a cabo mediante pruebas de funcionamiento y mejoras constantes, gracias a la investigación en los “docs” proporcionados por los frameworks en sus respectivas webs.

4.5.1. Mejora para la Reducción del Consumo de Datos

Las imágenes son una parte importante en todas las aplicaciones, ya que estas dan una mejor percepción de la aplicación y mejor experiencia de usuario, pero debido a que son imágenes ya sean editadas o capturas de la realidad (foto), estas son pesadas, y la mejor forma de gestionar las imágenes de manera rápida es reducir el peso de la imagen, pero esto conlleva una desventaja la calidad de la imagen se reduce drásticamente, lo más óptimo es encontrar el punto en que la imagen sea reducida sin pérdida de calidad.

El proceso de guarda de imagen es:

- Tomar la foto con la cámara.
- Obtener la imagen de la cámara.

- Guardar en la base de datos Firebase.

Lo que normalmente se hace al trabajar con la cámara del dispositivo móvil es, una vez capturada la imagen, el método Camera de ionic-native devuelve la imagen en una codificación base64, esta codificación hace uso del código ASCII, lo cual permite tener la imagen en un conjunto de caracteres raros, no en algún formato de imagen, el mismo que resulta ser muy pesado, es útil usar esa codificación en algunos casos que se necesite la imagen en el código, pero para este proyecto no sirve, solo hará que la aplicación consuma mucho más datos, por tener que enviar el dato en crudo a la base de datos. Con la codificación de la imagen en base64, el envío de la imagen a la base de datos consume más datos y se debe guardar en el mismo árbol JSON de la base de datos.

Esta pequeña situación se optimiza usando al dispositivo móvil, dejando que este haga el trabajo, una vez obtenido la imagen en base64 que devuelve la cámara, se debe pasar al dispositivo móvil para que usando su algoritmo de compresión de imagen lo pase a un formato estándar (.jpg) sin perder la calidad de la imagen y obtener la dirección (url) a dicha imagen, con la url de la imagen ya se puede enviar un archivo ultra ligero a Firebase, en vista de que ya es un archivo con formato de imagen no se puede guardar en el árbol JSON, entonces lo guardamos en el servicio “**Storage**” de Firebase, obtenemos la url a esa imagen y dicha url se guarda en el árbol JSON. Y Problema resuelto!

En la figura se observa cómo se optimizó la función:

Imagen 71. Método cámara

```

getPhotoNormal(){
  let cameraOptions: CameraOptions = {
    quality: 50,
    destinationType: Camera.DestinationType.DATA_URL,
    sourceType: Camera.PictureSourceType.CAMERA,
    allowEdit: false,
    encodingType: Camera.EncodingType.JPEG
    //targetWidth: 100,
    //targetHeight: 100
  };

  Camera.getPicture(cameraOptions).then((imageData)=>{
    //imagen codificada en base 64
    let base64Image = 'data:image/jpeg;base64,'+imageData;
    alert(imageData);
    this.imageData = base64Image;
    this.nuevaTarea.imagen = this.imageData;
  }).catch(err => console.log(err));
}

```

Fuente: Elaboración Propia

Imagen 72. Método cámara optimizado

```

getPhotoOptimizado(){
  let imageSource = (Device.isVirtual ? Camera.PictureSourceType.PHOTOLIBRARY : Camera.PictureSourceType.CAMERA);







  Camera.getPicture({
    quality: 50,
    destinationType: Camera.DestinationType.FILE_URI,
    sourceType: imageSource,
    targetHeight: 640,
    correctOrientation: true
  }).then((imagePath) => {
    alert(imagePath);
    this.imageData = imagePath;
    this.nuevaTarea.imagen = imagePath;
  }).catch(err => alert(err));
}

```

Fuente: Elaboración Propia

Cuando se guardó la imagen de la forma “Tradicional” o directa se observó que las imágenes pesaban entre 1.5 a 5 MB, con el método mejorado las imágenes en el Storage pesaban entre 10 a 35 KB, el cual es una diferencia notable y esto permite que la aplicación consuma menos datos al momento de gestionar las imágenes.

Imagen 73. Imágenes Storage Firebase

	 -LAbafPVb4zYA8CWqzeZ	14.61 KB	image/jpeg
	 -LAbY1FhJmXxXYgUDXM-	16.08 KB	image/jpeg
	 -LBFANNLGvBlv_njLNTL	13.56 KB	image/jpeg

Fuente: Elaboración Propia

4.5.2. Mejora de la Gestión de Tokens

Ya que las Notificaciones Push son una parte fundamental en la aplicación móvil, se debe gestionar los tokens (identificador único de la aplicación instalada en un determinado dispositivo) de manera óptima y sin redundancia para no generar conflictos y errores a la hora de hacer las notificaciones.

En vista de que un token es el identificador único de la aplicación instalada en un determinado dispositivo, este se guarda junto al id del usuario y se valida constantemente si un determinado usuario está en su determinada aplicación instalada en su determinado dispositivo o si se cambió el token, para que cuando se haga la notificación de un determinado usuario, este no llegue a más de uno por cambios de teléfonos o inicio de sesión en otros dispositivos.

Imagen 74. Salvar tokens

```

guardarToken(token){
  let noExisteToken = true;
  //alert(token);
  this.getAppTokens().then((appTokens) => {

    for(var r in appTokens){
      if(token == appTokens[r].devtoken){
        noExisteToken = false;
        continue
      }
    }
    //alert('No existe: ' + noExisteToken);
    if(noExisteToken){
      this.addToken(token);
    }
  });
}

```

Fuente: Elaboración Propia

5. EVALUACIÓN DE LA SOLUCIÓN

5.1. TDD (TEST DRIVEN DEVELOPMENT)

Todos los módulos y clases de la aplicación móvil se construyeron siguiendo el TDD como técnica de programación, para ello cada módulo, clase, método y función, se realizó con los siguientes pasos:

- **Elegir un requisito:** Para el ejemplo se tomará el requisito de “obtención de estrellas” el cual es parte del módulo de calificación.
- **Codificar la prueba:** Se implementará un método llamado `getEstrellas()` desde la promesa de un evento, en esta parte de acuerdo a lo que se necesita en el programa se va definiendo los métodos o funciones, se determina que acción se hará y el nombre de dicho método o función antes de implementar el código de esos métodos o funciones. Esto ayuda a determinar de forma clara y precisa lo que se necesita y evita la duplicidad de código.

En la imagen se observa la llamada al método `getEstrellas()`, el mismo que devuelve error debido a que dicho método no está implementado.

Imagen 75. TDD – `this.getEstrellas()` - error

```
this.events.subscribe('Anuncios Aceptados', () => {  
  this.getEstrellas();  
  console.log(this.AnunciosAceptados);  
})
```

Fuente: Elaboración Propia

- **Verificar que la prueba falla:** La línea de código “`this.getEstrellas();`” devuelve un error ya que aún no se ha implementado este método.
- **Codificar la implementación:** Se escribe el código lo sencillo, claro, preciso y simple posible para que el método implementado haga lo que tiene que hacer sin complicaciones.

En la imagen se aprecia la implementación del método `getEstrellas()`.

Imagen 76. TDD – `getEstrellas()`

```

getEstrellas(){
  this.Estrella = [];
  this.semiEstrella = [];
  var temp = 0;
  for(var r in this.AmigosAceptados){
    temp = 0;
    for(var h in this.AmigosAceptados[r].Estrella){
      temp = this.AmigosAceptados[r].Estrella[h];
    }
  }
}

```

Fuente: Elaboración Propia

- **Volver a ejecutar la prueba:** Una vez implementado el método `getEstrellas()`, ya no debería haber error en el programa.

En la imagen se observa que ya no se marca como error.

Imagen 77. TDD – `this.getEstrellas()` - ok

```

this.events.subscribe('Anuncios Aceptados', () => {
  this.getEstrellas();
  console.log(this.AnunciosAceptados);
})

```

Fuente: Elaboración Propia

- **Refactorización:** En este paso se elimina el código duplicado y otros, también se puede incluir mejores prestaciones en el código si es necesario, para no implementar otros métodos innecesarios.

En la imagen se observa la refactorización que implementa validaciones que son necesarias para que el método cumpla los requisitos de manera óptima.

Imagen 78. TDD – `this.getEstrellas()` - refactor


```

getEstrellas(){
  this.Estrella = [];
  this.semiEstrella = [];
  var temp = 0;
  for(var r in this.AmigosAceptados){
    temp = 0;
    for(var h in this.AmigosAceptados[r].Estrella){
      temp = this.AmigosAceptados[r].Estrella[h];
      if(temp == 6){
        this.semiEstrella.push(temp);
      }else{
        this.Estrella.push(temp);
      }
    }
  }
}

```

Fuente: Elaboración Propia

Una vez hecho este ciclo, al terminar con el requisito seleccionado, dicho requisito se da de alta o terminado y se toma otro siguiendo el mismo ciclo.

5.2. PRUEBAS FUNCIONALES

La prueba funcional se aplica para poder validar si el comportamiento de la aplicación móvil cumple con las especificaciones establecidas.

Esta prueba se realizó teniendo en cuenta los casos de uso definidos en la fase de iniciación de Mobile-D.

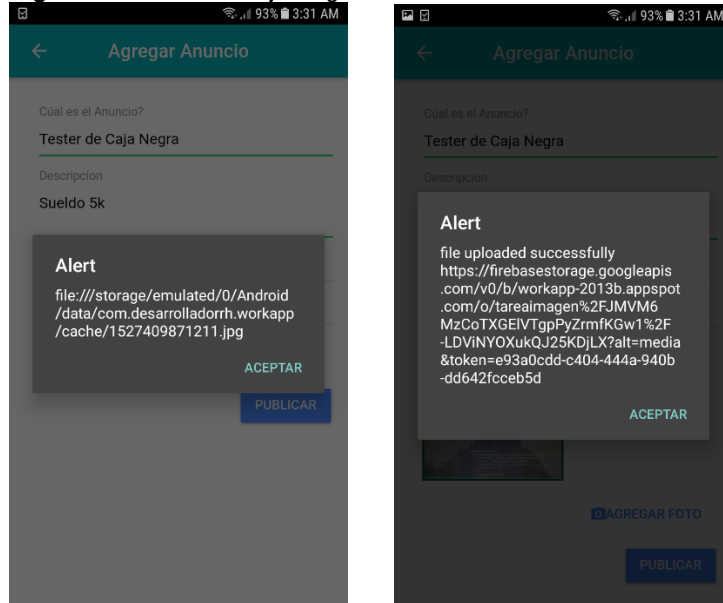
5.2.1. Pruebas de Caja Negra

Las pruebas se centraron en analizar los datos de entrada y salida de la aplicación móvil “WorkApp” y la interacción de los eventos con otros usuarios.

Las pruebas a las interfaces se realizaron de forma manual, se probó toda la aplicación desde el registro de un nuevo usuario, pasando por la publicación de anuncios, envíos de solicitudes de amistad y de postulación, calificación y asignación de estrellas, hasta la manipulación de la sala de chats y la creación de un Chat Group, en dichas pruebas se encontraron algunos alerts implementados durante el desarrollo para hacer seguimiento a los procesos de los módulos de la aplicación.

En la imagen se observa algunos alerts que se dejaron sueltos.

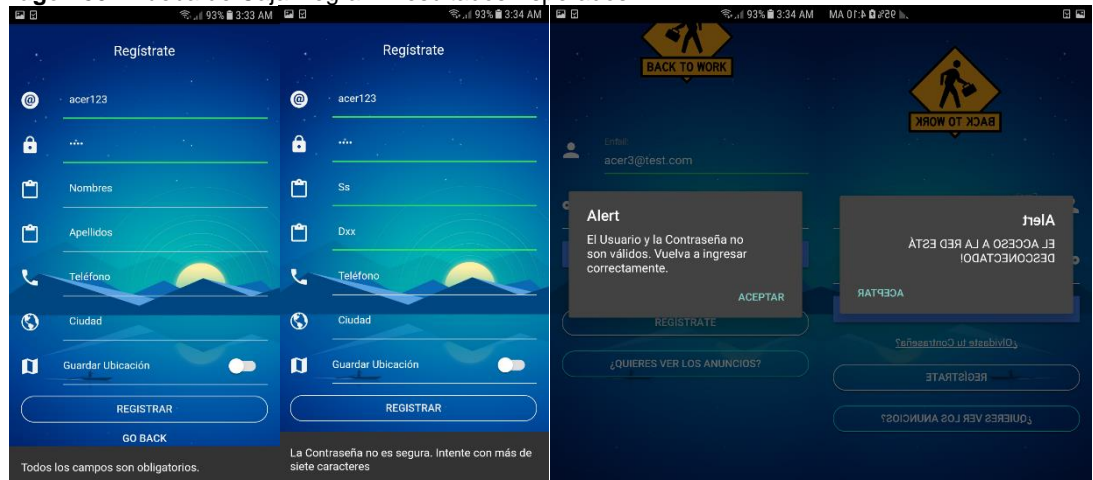
Imagen 79. Prueba de Caja Negra – Alerts

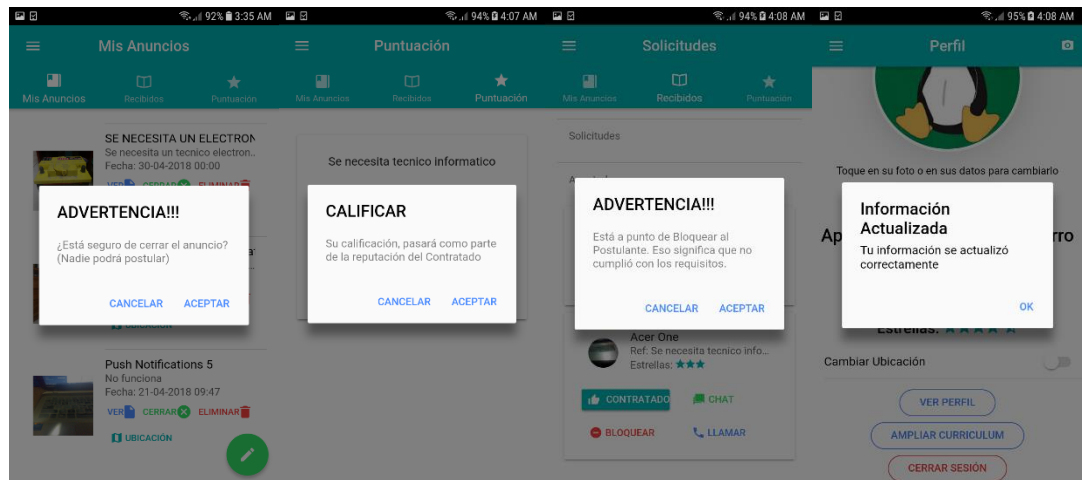


Fuente: Elaboración Propia

Una vez detectados dichos alerts, se procedió a quitar dichas líneas de código, para dejar la aplicación móvil limpia.

Imagen 80. Prueba de Caja Negra – Resultados Esperados





Fuente: Elaboración Propia

Luego de realizar las pruebas a toda la aplicación móvil, los resultados esperados fueron satisfactorios, ya que cada acción tiene su respectivo control en el cual es necesario aceptar para continuar.

Las pruebas a los eventos en tiempo real se realizaron de forma automatizada, subiendo datos en la rama anuncios de la base de datos, en la interfaz principal en el cual se visualiza los anuncios en tiempo real, las actualizaciones de los nuevos anuncios fueron instantáneos, ello cual cumple con lo esperado.

Finalmente las pruebas aplicadas resultaron satisfactorias, el cual da paso a la aplicación móvil, terminar con la fase de desarrollo y entrar a la fase producción.

5.3. PRUEBAS DE COMPATIBILIDAD

Se realizó pruebas de funcionamiento de la aplicación móvil desarrollado, en diferentes dispositivos móviles con diferentes versiones de Android.

- **Android 4.2:** interfaces adaptables al tamaño del dispositivo, colores de las interfaces regulares, buen funcionamiento de toda la aplicación.
- **Android 5.0:** interfaces adaptables al tamaño del dispositivo, colores nítidos, buen funcionamiento de toda la aplicación.
- **Android 7.0:** interfaces adaptables al tamaño del dispositivo, colores hd, buen funcionamiento de toda la aplicación, mejor rendimiento.

Pruebas en dispositivos móviles de gama baja, media y alta:

- **SmartPhone Gama Baja:** La aplicación tiene cierto retraso al momento de lanzar la aplicación (tolerable), al momento de conectarse a la red y descargar los datos necesarios, los colores en las interfaces son regulares, el rendimiento es regular, el consumo de datos de internet es bajo, el funcionamiento de toda la aplicación móvil es bueno.
- **SmartPhone Gama Media – Alta:** La aplicación carga rápido, las conexiones son rápidas al igual que la descarga de datos, los colores en las interfaces tienen muy buena resolución (hd), el rendimiento es muy bueno, el consumo de datos de internet es muy bajo, los service workers trabajan bien, el funcionamiento de toda la aplicación móvil es excelente.

6. RESULTADOS

Basados en el objetivo general **“Desarrollar un aplicativo móvil multiplataforma mediante IONIC y Firebase, para incrementar las posibilidades de encontrar anuncios actualizados de oferta y demanda laboral en tiempo real”**, el cual fue logrado como resultado de la implementación de la aplicación móvil, la investigación realizada con respecto a las tecnologías usadas y al cumplir con los objetivos específicos:

- **Objetivo específico 1: Optimizar el tiempo de respuesta y petición de datos, con un modelo Cliente – Servidor más corto.**

Este objetivo fue logrado gracias a las funcionalidades de las tecnologías usadas en este proyecto, a las cuales se les sacó su máximo provecho, la documentación que valida el logro de este objetivo se puede ver en la: **Sección 4.4 Optimización del Modelo Cliente – Servidor.**

- **Objetivo específico 2: Desarrollar métodos y funciones para permitir la verificación de información, seguridad y confianza en las publicaciones de los anuncios.**

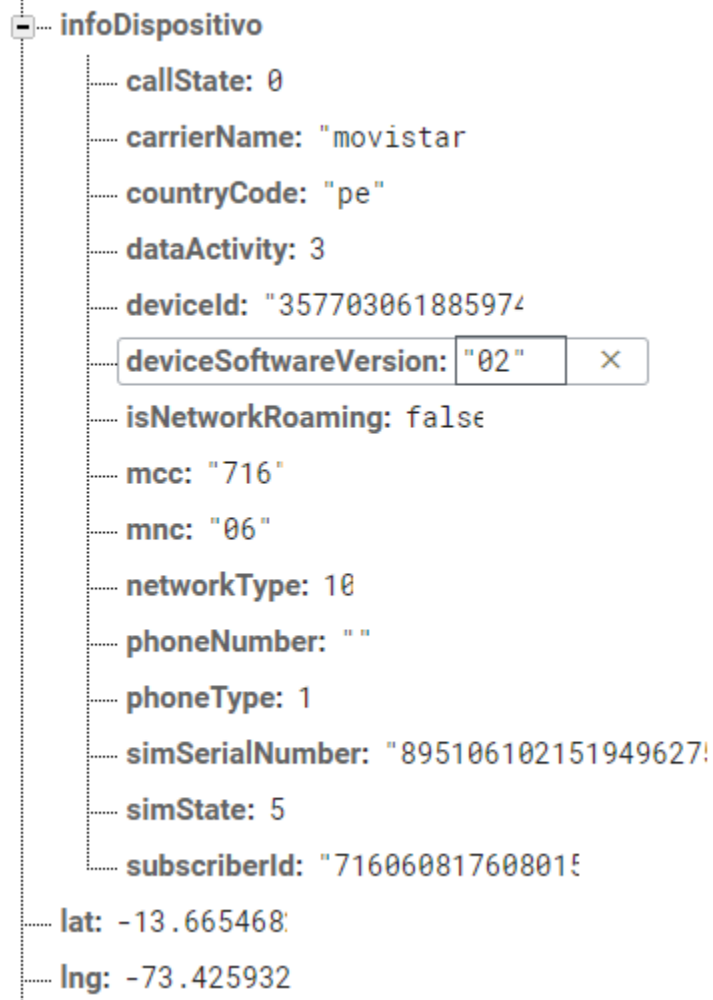
Este objetivo se logró al implementar dichos métodos y funciones que están en los Providers.

Uno de los problemas en esta aplicación que básicamente proporciona el contacto entre un anunciante y un postulante, es la seguridad, por ello dentro de las limitaciones se definió que la aplicación solo proporciona información y dejando la libertad de decisión a los usuarios, pero se proporciona el chat para que los usuarios puedan comunicarse y dar confianza antes de hacer la contratación tanto en la aplicación como en la realidad.

Pero no se dejó de lado el problema de la seguridad, la verificación de datos del usuario no se implementó ya que para ello se tendría que gestionar de la RENIEC, ello no es parte fundamental del proyecto, pero se enfocó en el dispositivo móvil que tiene instalada la aplicación, y se pensó en obtener las coordenadas de ubicación del usuario, cuando se registra y cada vez que hace

una publicación, el cual permite tener un grado de confianza al saber dónde se encuentra, también se obtiene la información del dispositivo que usa el usuario, se puede obtener el id del dispositivo que lo identifica de manera única a nivel mundial, la línea telefónica, el país y más datos, también en ocasiones dependiendo de cómo esté configurada la tarjeta sim se puede obtener el número telefónico del dispositivo móvil y validarlo con el número que ingresó, esto ayuda a ser más segura a la aplicación móvil.

Imagen 81. Información del Dispositivo



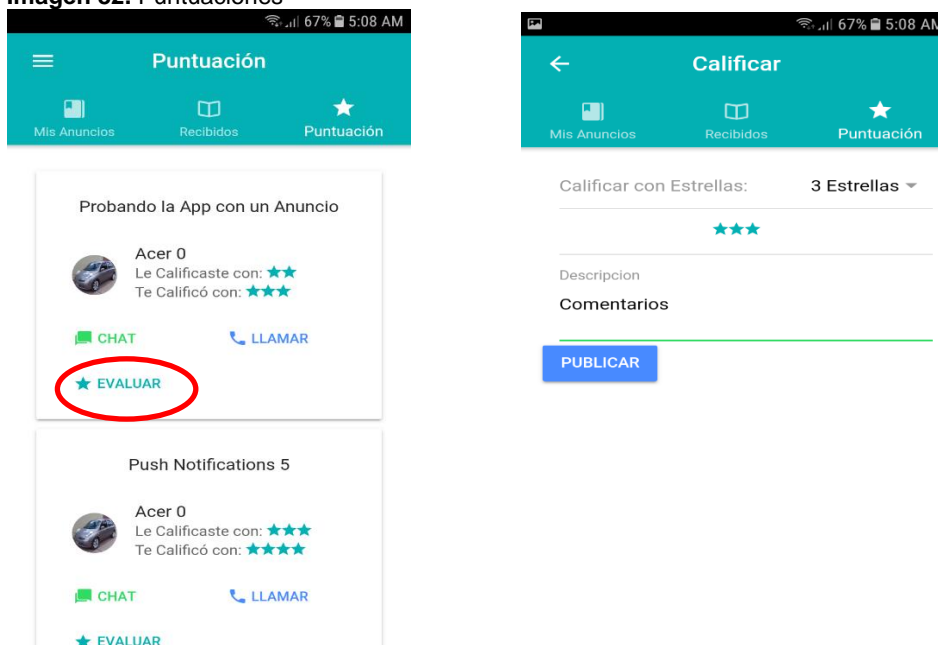
Fuente: Elaboración Propia

- **Objetivo específico 3: Desarrollar un módulo de calificación dinámico e interactivo.**

Este objetivo se logró cuando se implementó el módulo de calificación y gestión de estrellas, en la fase de producción de Mobile-D.

Las calificaciones que se hacen entre los usuarios, empiezan cuando un usuario postulante es contratado mediante la aplicación móvil, ambos usuarios pueden evaluar el desempeño constantemente tanto del contratado como del contratador, estas calificaciones que van haciendo con respecto a un anuncio o trabajo que se publicó, forman parte de la puntuación del usuario que se refleja en su perfil, el cual da más confianza a los otros usuarios al momento de contratarlos o postular a sus anuncios.

Imagen 82. Puntuaciones



Fuente: Elaboración Propia

- **Objetivo específico 4: Integrar Notificaciones Push sin la dependencia de un servidor.**

Este objetivo fue logrado cuando se obtuvo la primera versión estable completa de la aplicación móvil y se implementó las notificaciones push, la documentación que demuestra el logro de este objetivo se puede ver en la: **Sección 4.3 Notificaciones Push.**

- **Objetivo específico 5: Desarrollar una aplicación móvil sin costo en la fase de producción.**

La fase de producción empieza cuando termina la fase de desarrollo, con una versión estable de la aplicación, en esta fase de producción la aplicación móvil tiene que estar disponible para el uso de los usuarios y para ello se debe tener listo todo (la aplicación móvil, la integración a la web “server” y la base de datos), básicamente consiste en contratar los servicios de terceros para el alojamiento del servidor y la base de datos, entre otros servicios de pago que se usarán.

En el caso de este proyecto todo lo usado es gratuito, es más, no se necesita pensar en contratar un cloud o hosting, ya que no hace uso de un servidor, la aplicación interactúa con Firebase el cual es gratuito, pero no ilimitado, te da un límite que es más que suficiente para pasar una aplicación a la etapa de producción ya que recién se empieza a tener usuarios, una vez que se tenga una base de usuarios considerable, se puede optar pagar el servicio premium.

Por ello este objetivo es logrado indirectamente al usar estas tecnologías.

7. CONCLUSIONES

Tras haber ejecutado el presente proyecto de investigación tecnológica se concluye que:

- Es posible mejorar el modelo cliente servidor de cuatro pasos de ida y vuelta a tan solo dos pasos (ida y vuelta), con la ausencia de un servidor, para este tipo de aplicaciones móviles; para mejorar y reducir el tiempo de respuesta a las peticiones, eventos e interacción de la aplicación con la base de datos, el cual mejora mucho la experiencia del usuario, esto acompañado del bajo consumo de datos en la aplicación móvil.
- Con las tecnologías usadas en este proyecto es posible desarrollar una aplicación móvil, con bajos costos de implementación en la fase de producción de la aplicación.
- Gracias a los plugins de Cordova disponibles para ser usados con IONIC, se puede implementar una aplicación móvil híbrida, casi al mismo nivel que una aplicación nativa, con manipulación y acceso a las características del dispositivo móvil; estos plugins también ayudan a proporcionar mayor confianza y seguridad a los usuarios, gracias a que se puede obtener información que respalde dicha seguridad.
- Una aplicación móvil puede integrar las Notificaciones Push sin la necesidad de crear un server para ello, a cada aplicación móvil instalada en un determinado dispositivo móvil por un determinado usuario, puede recibir las notificaciones de acuerdo al evento que se realice, las mismas que pueden ser notificaciones individuales a un determinado usuario.
- La metodología Mobile-D permitió agilizar y organizar el trabajo, el cual conllevó a desarrollar la aplicación móvil con éxito gracias a la practicidad de ejecución de sus fases, la elasticidad y adaptabilidad de la metodología en los procesos, permitió llegar a la versión estable de la aplicación móvil.

8. RECOMENDACIONES

En el transcurso del desarrollo de este proyecto de investigación se logró identificar varios puntos clave que se deben tomar en cuenta para futuras investigaciones de esta índole.

- El árbol JSON de la base de datos, por si solo es altamente escalable, el cual permite que la aplicación móvil también tenga esta característica, pero se recomienda tomarse un tiempo para hacer la estructura del árbol lo más eficiente posible.
- Las versiones más recientes de las dependencias no siempre son las mejores, a veces se generan errores aparentemente graves, que en el fondo solo son incompatibilidades entre dichos paquetes.
- Para poder desarrollar este tipo de aplicaciones y cualquier otro, es muy importante leer siempre las documentaciones de los frameworks o tecnologías que se estén usando, para poder saber con precisión qué es lo que hacen.
- Si quieres implementar algo nuevo a tu aplicación móvil, sin saber bien como resultará, es mejor crear un nuevo proyecto, para no comprometer el proyecto principal y hacer los experimentos en él, una vez que esté bien desarrollado lo integras recién a tu aplicación principal.
- Para este tipo de aplicaciones, se debe de hacer la programación lo más generalista que se pueda, procurar no hacer métodos específicos para un solo caso, si es necesario un método específico, puedes recurrir a la base de datos y agregar atributos que te ayuden a que el método sea más general, recuerda que la base de datos es un árbol JSON (NoSQL) y te permite agregar atributos en cualquier lugar y momento sin que los datos agregados anteriormente sufran algún problema.
- Cada módulo y método que vas desarrollando, se debe ir probando y solucionando los errores que presenta.
- La terminal de Linux trabaja muy bien con todos los comandos que se usan, pero puede haber dificultades al inicio cuando se integra el ADB de Android, solo necesitas saber dónde instalar para que este accesible.

9. REFERENCIAS BIBLIOGRÁFICAS

- Agencia de Prensa Lima Norte. (20 de Agosto de 2017). *En 14% se incrementó el número de jóvenes que busca empleo por Internet*. Obtenido de <http://agenciadepresalimanorte.com/en-14-se-incremento-el-numero-de-jovenes-que-busca-empleo-por-internet/>
- Amazon. (2018). *¿Qué es NoSQL?* Obtenido de <https://aws.amazon.com/es/nosql/>
- Apache Cordova. (2017). *Apache Cordova*. Recuperado el 21 de Agosto de 2017, de <https://cordova.apache.org/>
- Blanco, P., Camarero, J., Fumero, A., & Adam Werteski, R. P. (2009). *Metodología de desarrollo ágil para sistemas móviles*. Madrid: dit UPM. Recuperado el 20 de Agosto de 2017, de http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf
- Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*. Gunroad. Recuperado el 21 de Agosto de 2017, de https://books.google.com.pe/books?id=ATiqsjH1rvwC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Diario El Cumbe. (26 de Octubre de 2016). *Empleo decente es difícil de conseguir para el 80% de jóvenes peruanos*. Obtenido de https://issuu.com/diarioelcumbecajamrca/docs/diario_26_de_octubre_892d5c2876d277/8
- Fernandez, J. (1 de Septiembre de 2006). *Tipos de dispositivos móviles*. Recuperado el 10 de Agosto de 2017, de http://leo.ugr.es/J2ME/INTRO/intro_4.htm
- Fernández, J. (Septiembre de 2006). *Tipos de Dispositivos Móviles*. Recuperado el 15 de Junio de 2016, de http://leo.ugr.es/J2ME/INTRO/intro_4.htm
- Figueroa, R. (2003). *Desempleo y precariedad en la sociedad de mercado* (2 ed.). Santiago de Chile: RIL Editores. Recuperado el 5 de Agosto de 2017, de https://books.google.com.pe/books?id=nXxtERFgPgIC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Fundación Telefónica. (2011). *Real time web: Una nueva conciencia global*. Madrid: Ariel, S.A. Recuperado el 19 de Agosto de 2017, de https://publiadmin.fundaciontelefonica.com/index.php/publicaciones/add_des_cargas?title=Real+Time+Web%3A+a+New+Lens+to+See+the+World&code=120&lang=en&file=Real_time_web_nw1.pdf
- Fundación Telefónica. (2011). *Real Time Web: una nueva conciencia global*. Madrid: Ariel, S.A. Obtenido de https://publiadmin.fundaciontelefonica.com/index.php/publicaciones/add_des

cargas?title=Real+Time+Web%3A+a+New+Lens+to+See+the+World&code=120&lang=en&file=Real_time_web_nw1.pdf

- Google. (2017). *Progressive Web Apps*. Recuperado el 22 de Agosto de 2017, de <https://developers.google.com/web/progressive-web-apps/?hl=en>
- Google Developers. (31 de Mayo de 2017). *Firestore*. Recuperado el 23 de Agosto de 2017, de <https://firebase.google.com>
- IBM. (2018). *Modelo de Cliente/Servidor*. Obtenido de https://www.ibm.com/support/knowledgecenter/es/SSAL2T_9.1.0/com.ibm.ci.cs.tx.doc/concepts/c_clnt_sevr_model.html
- International Labour Organization. (2013). *Key Indicators of the Labour Market*. America Latina y el Caribe: KILM. Obtenido de http://www.ilo.org/wcmsp5/groups/public/---dgreports/---stat/documents/publication/wcms_501585.pdf
- Ionic Framework. (2017). *The top open source framework for building amazing mobile apps*. Recuperado el 22 de Agosto de 2017, de <https://ionicframework.com/>
- Mechato, J., & Gastañadui, C. (2014). *Metodologías ágiles para el desarrollo de aplicaciones móviles*. Lima: San Marcos. Recuperado el 20 de Agosto de 2017
- Mejia, J., Muñoz, M., Rocha, Á., San Feliu, T., & Peña, A. (2016). *Trends and Applications in Software Engineering*. Mexico: Springer. Recuperado el 22 de Agosto de 2017, de https://books.google.com.pe/books?id=t8w6DQAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Paradigma Digital. (26 de Abril de 2017). *TDD, una metodología para gobernarlos a todos*. Obtenido de <https://www.paradigmadigital.com/techbiz/tdd-una-metodologia-gobernarlos-todos/>
- Perú21. (25 de Octubre de 2016). *Empleo decente es difícil de conseguir para el 80% de jóvenes peruanos*. Obtenido de <https://peru21.pe/economia/decente-dificil-80-jovenes-peruanos-231630>
- Perú21. (03 de Mayo de 2017). *Sueldo mínimo de peruanos podría subir solo a S/ 895*. Obtenido de <https://peru21.pe/economia/sueldo-minimo-peruanos-subir-s-895-74893>
- Planas, M. G. (2007). *Gestión Práctica de la Seguridad Social* (4 ed.). España: CISS. Recuperado el 30 de Julio de 2017, de https://books.google.com.pe/books?id=EQCmf-x_PAYC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Real Academia Española. (20 de Agosto de 2017). Recuperado el 18 de Agosto de 2017, de <http://dle.rae.es/srv/fetch?id=YtSuzm8>

Santiago, R., Trbaldo, S., Kamijo, M., & Fernández, Á. (2015). *Mobile learning: nuevas realidades en el aula*. España: Oceano. Recuperado el 21 de Agosto de 2017

Significados.com. (8 de Enero de 2014). *Qué es el Empleo Informal*. Recuperado el 15 de Agosto de 2017, de <https://www.significados.com/empleo-informal/>

Wikipedia. (5 de Febrero de 2017). *Tablón de Anuncios*. Recuperado el 19 de Agosto de 2017, de https://es.wikipedia.org/wiki/Tabl%C3%B3n_de_anuncios

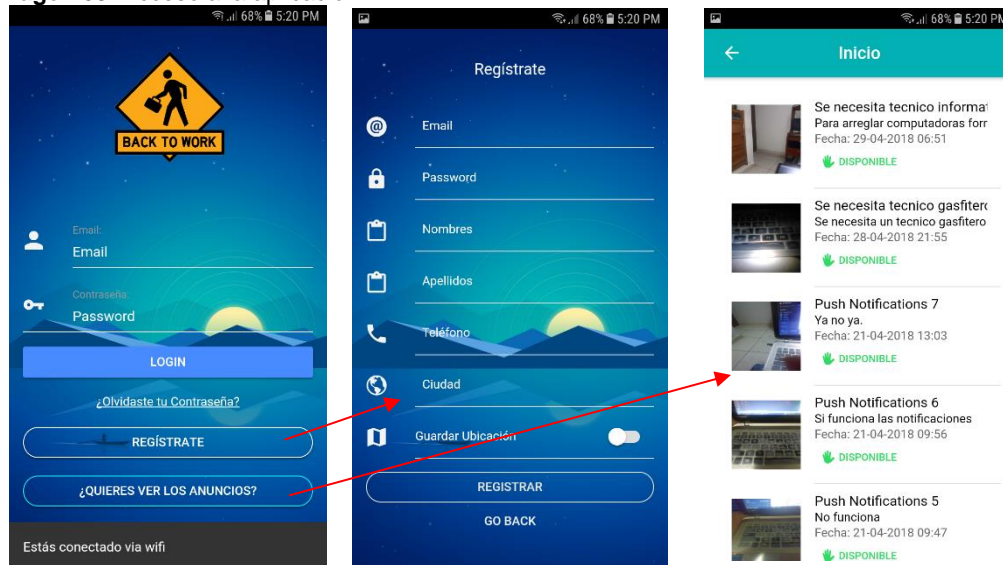
10. ANEXOS

MANUAL DEL USUARIO

A continuación se detalla el uso y funcionamiento de la aplicación móvil WorkApp.

- Las interfaces disponibles inmediatamente después de abrirse la aplicación son:
 - **Login:** es la página principal, el cual sirve para autenticarse en la aplicación e ingresar a ver los anuncios y administrarlos.
 - **Password Reset:** es donde se reestablece la contraseña del usuario.
 - **Registro de Usuario:** es donde se registra los datos de un nuevo usuario y se crea la cuenta de acceso.
 - **Ver anuncios:** es una página, donde se visualiza los anuncios publicados, pero no tiene acceso a más opciones, ayuda a que el usuario se interese y registre en la aplicación.

Imagen 83. Acceso a la aplicación

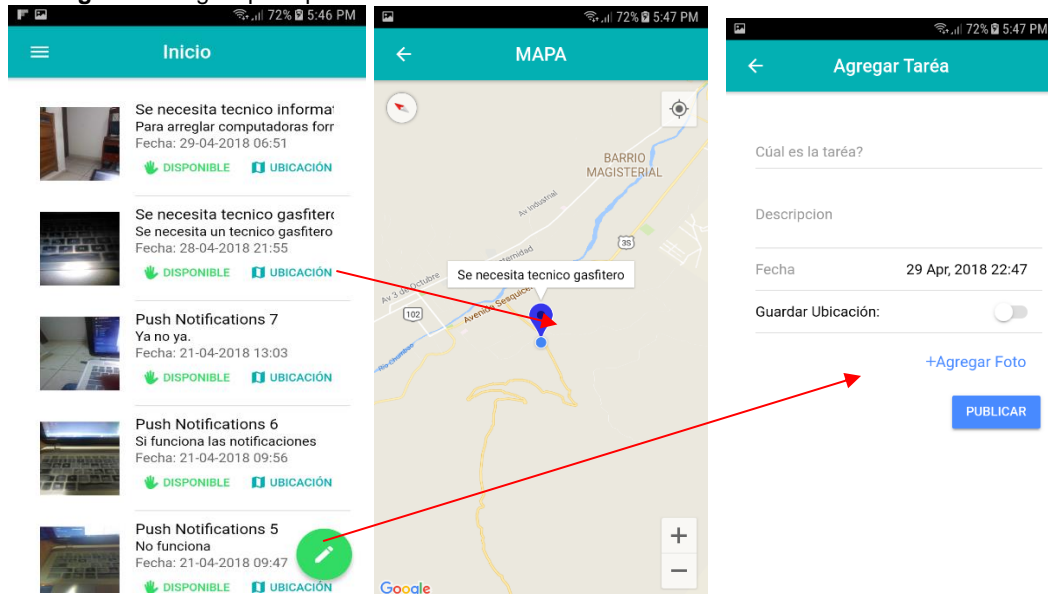


Fuente: Elaboración Propia

- Cuando el usuario se autentica (login) e inicia sesión, se accede a la página principal donde puede visualizar todos los anuncios publicados, con opciones a:

- **Ver el mapa:** se muestra la ubicación del anuncio, es decir donde se sitúa el lugar de trabajo.
- **Crear un Nuevo Anuncio:** formulario donde ingresas la información del anuncio a crear.
- **Ver Información detallada del anuncio:** al seleccionar el anuncio se carga toda su información.

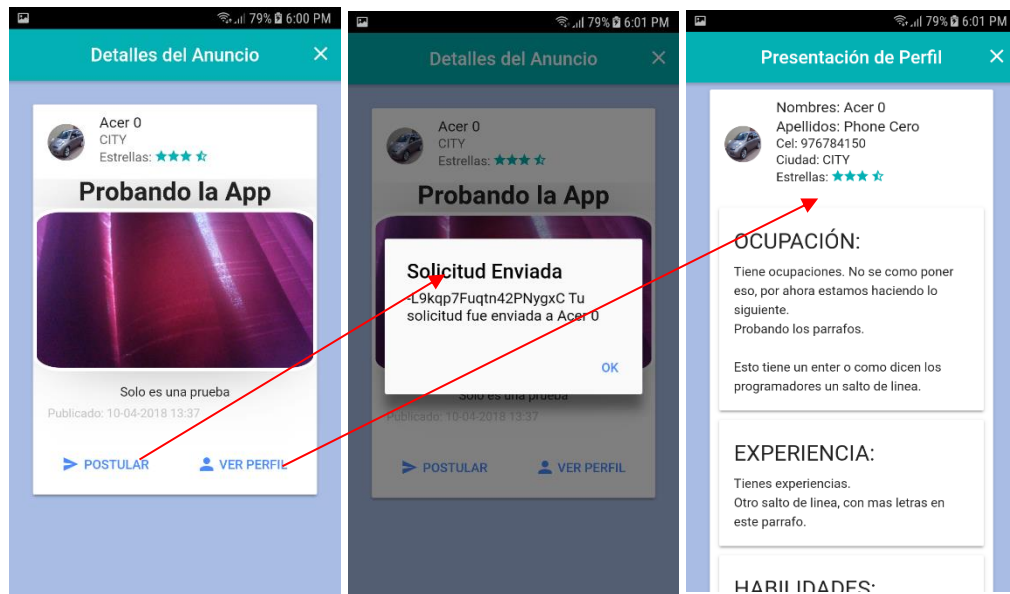
Imagen 84. Página principal



Fuente: Elaboración Propia

- Al tocar en un anuncio se carga en una página toda la información detallada del anuncio y del anunciante, con opciones a:
 - **Postular a la Oferta de Trabajo:** envía una solicitud de postulación al usuario de la publicación.
 - **Ver Perfil del Anunciante:** para mayor confiabilidad se muestra toda la información disponible del usuario anunciante, de cuantas estrellas es, los comentarios que le hicieron otros usuarios contratados por este, si están disponibles.

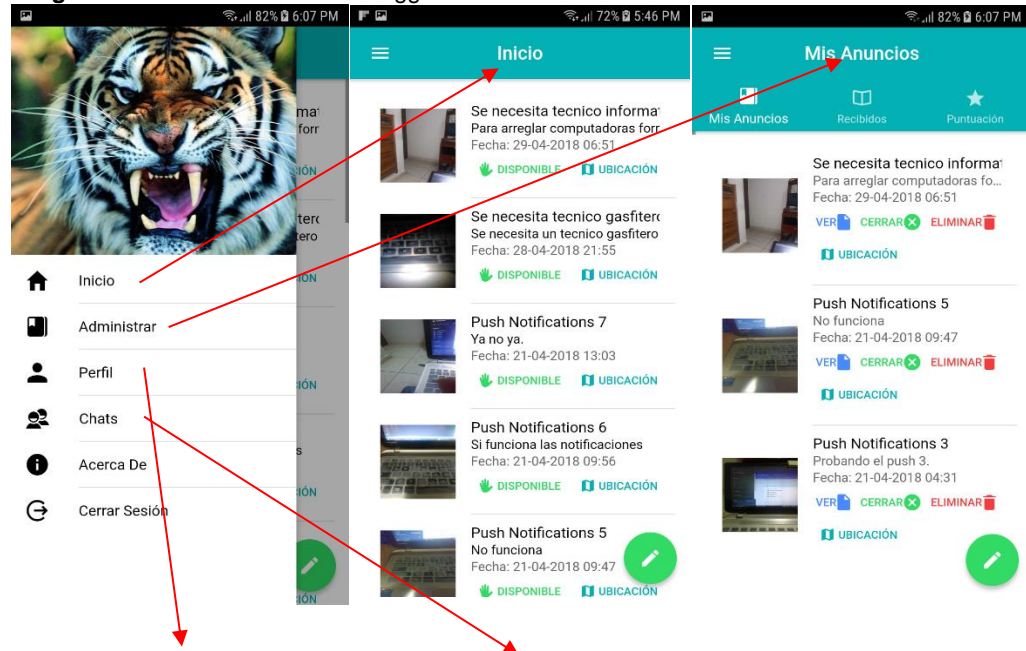
Imagen 85. Información del anuncio

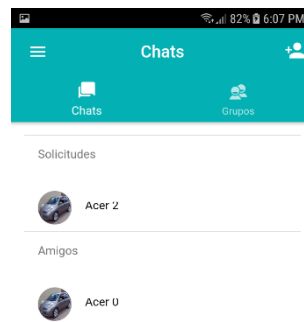


Fuente: Elaboración Propia

- El menú lateral da acceso a las siguientes páginas: Página Principal, Administración de Anuncios, Perfil del Usuario, Sala de Chats, Acerca de la Aplicación y Cerrar Sesión.

Imagen 86. Menú lateral – Menu Toggle

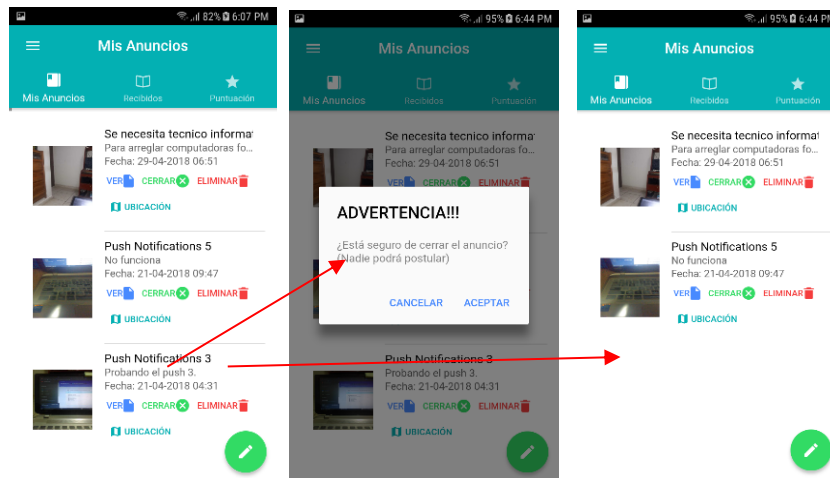




Fuente: Elaboración Propia

- La sección de administración de anuncios se divide en tres partes importantes:
 - ✓ **Administración de mis anuncios:** es donde se administra los anuncios creados por el usuario, en el cual se pueden hacer las siguientes acciones.
 - **Ver el Anuncio:** accede a ver el anuncio de manera detallada.
 - **La Ubicación del Anuncio:** se visualiza mediante el mapa.
 - **Cerrar el Anuncio:** que consiste en dar por terminado la convocatoria de dicha oferta laboral.
 - **Eliminar el Anuncio:** Elimina el anuncio publicado.

Imagen 87. Administración de mis anuncios

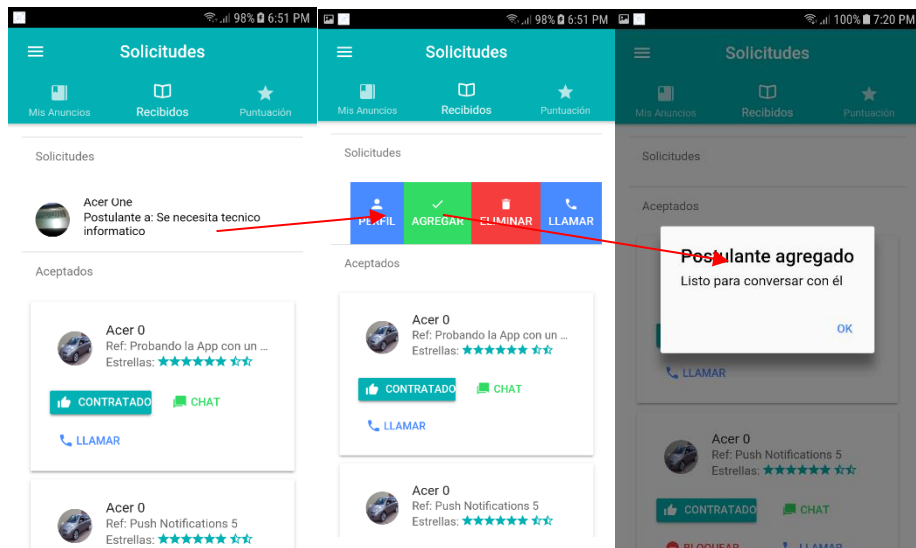


Fuente: Elaboración Propia

✓ **Gestión de solicitudes recibidas y aceptados:** es donde visualizas las solicitudes de los postulantes a tu oferta laboral que publicaste y a quienes aceptaste:

- **Solicitudes Recibidas:** sección donde se observa las solicitudes que te enviaron por tu anuncio, el cual permite las siguientes acciones:
 - **Ver perfil:** Puedes ver el perfil del postulante.
 - **Agregar o Aceptar:** acción que te permite aceptar la solicitud.
 - **Eliminar Solicitud:** acción que elimina la solicitud recibida.
 - **Llamar:** acción que permite llamar al postulante.

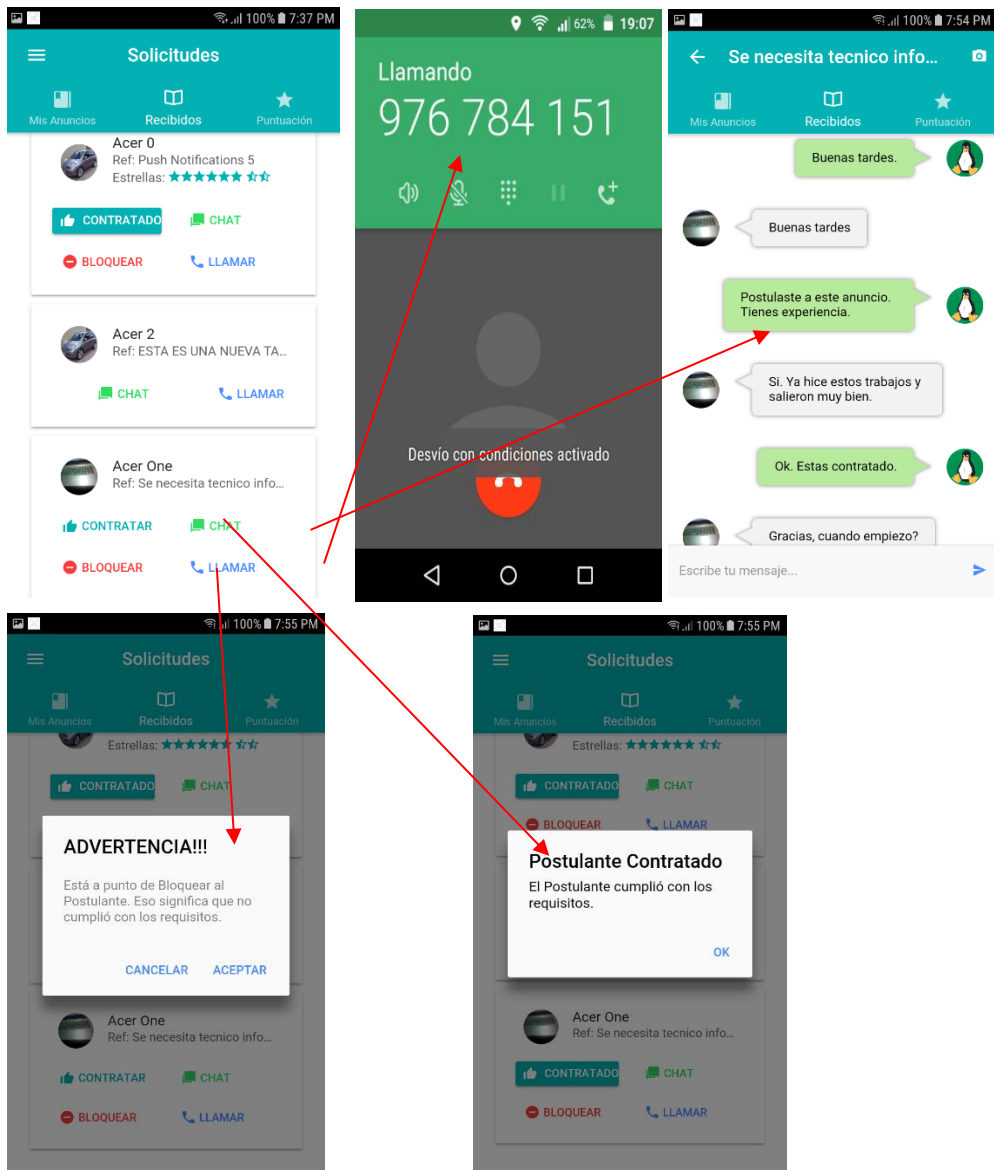
Imagen 88. Solicitudes recibidas por un anuncio



Fuente: Elaboración Propia

- **Solicitudes Aceptadas:** sección donde se observa las solicitudes recibidas que se aceptaron y las solicitudes que enviaste que te aceptaron, el cual tiene las siguientes opciones:
 - **Ver Perfil:** permite ver el perfil del usuario.
 - **Chat:** acción que permite conversar a los usuarios con respecto a un anuncio, previa solicitud y aceptación de la misma.
 - **Llamar:** acción que permite hacer una llamada directa al otro usuario.
 - **Contratar:** solo está disponible para el usuario dueño del anuncio, esta acción permite cambiar el estado a contratado de un postulante.
 - **Bloquear:** solo está disponible para el usuario dueño del anuncio, esta acción permite bloquear el usuario postulante e indica que no cumplió con los requisitos.

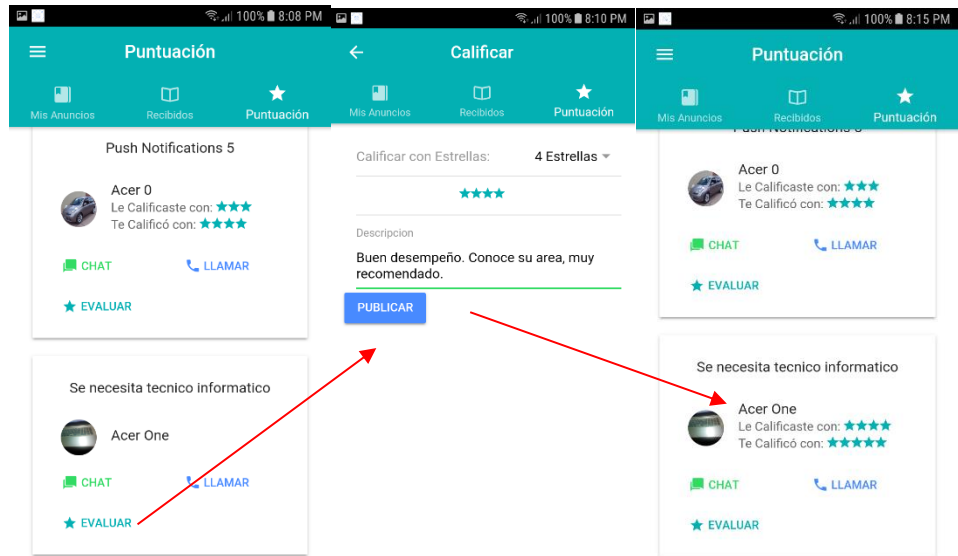
Imagen 89. Solicitudes aceptadas de un anuncio



Fuente: Elaboración Propia

- ✓ **Gestión de Usuarios Contratados:** sección donde se observa a los usuarios contratados y que permite la evaluación dinámica entre ambos usuarios, la función evaluar permite asignar estrellas de acuerdo al desempeño de los usuarios, acompañado de un comentario descriptivo.

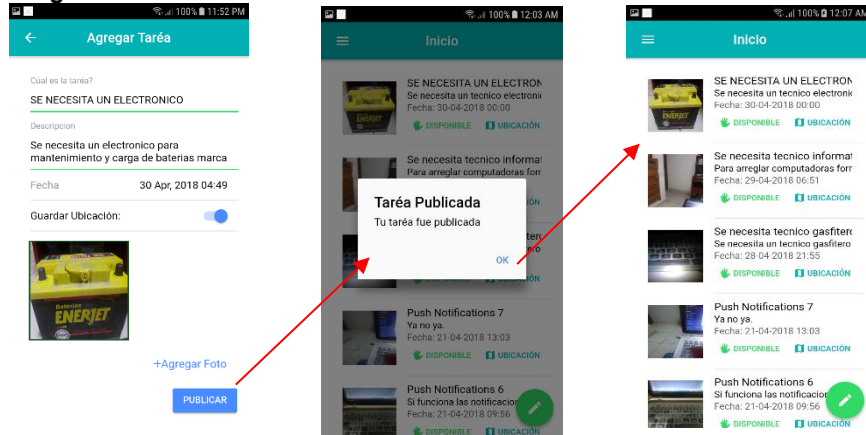
Imagen 90. Usuario contratado – calificación



Fuente: Elaboración Propia

- Para agregar un nuevo anuncio, se debe ir al botón de agregar, llenar los datos correspondientes y hacer la publicación, cuando se publica un nuevo anuncio, se actualiza automáticamente los anuncios en la página principal y además todos los usuarios que no estén usando la aplicación en ese instante son notificados de dicho evento.

Imagen 91. Publicar Anuncio

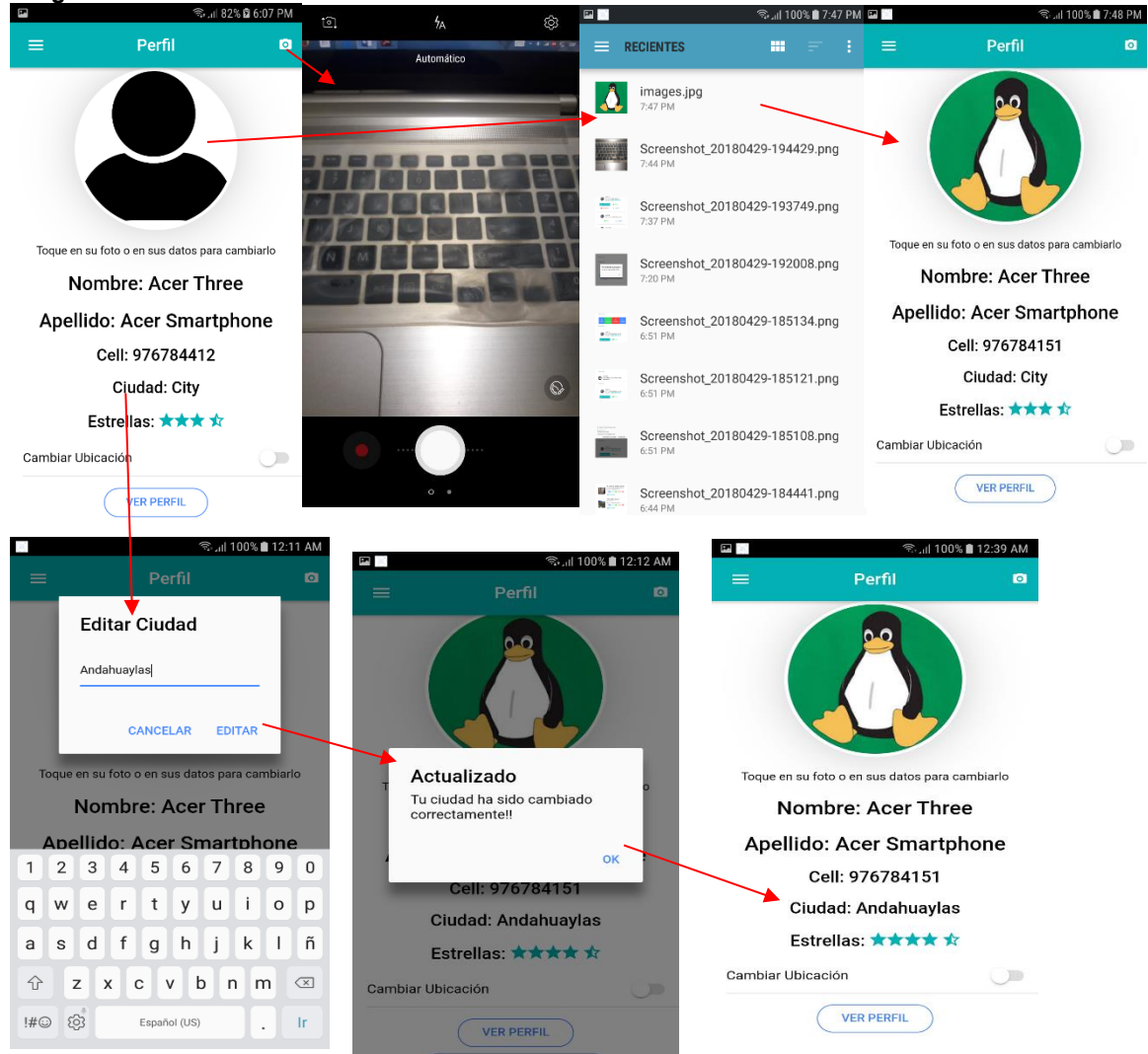


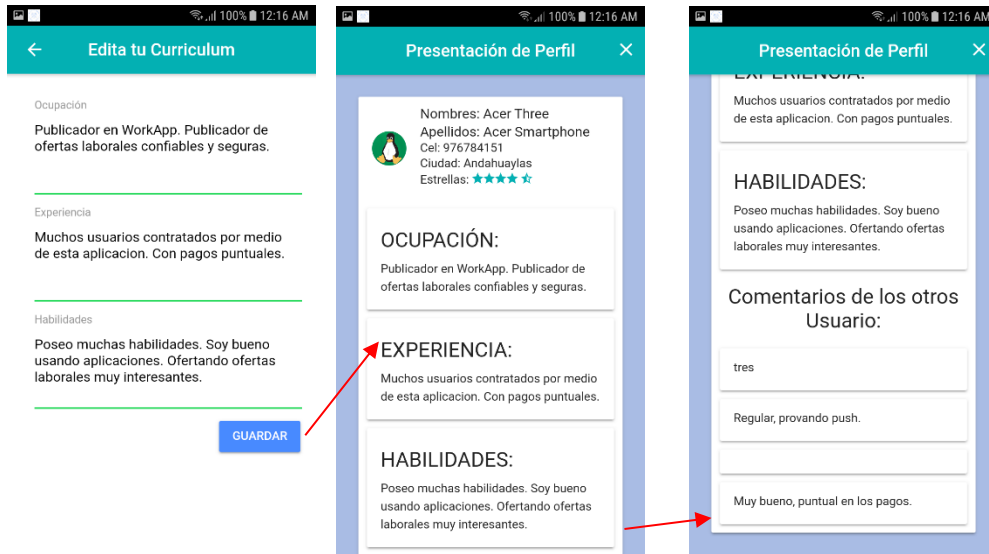
Fuente: Elaboración Propia

- En la sección de Perfil de Usuario se tiene acciones que actualizan y modifican los datos del usuario, estas acciones son:
 - **Actualización del Avatar del usuario.**
 - **Modificación del Nombre, Apellido, Teléfono, Ciudad y Ubicación.**

- **Ver Perfil Detallado:** permite visualizar la presentación del perfil.
- **Ampliar Curriculum:** permite agregar más información para tener un perfil más llamativo.
- **Cerrar Sesión.**

Imagen 92. Perfil de Usuario.

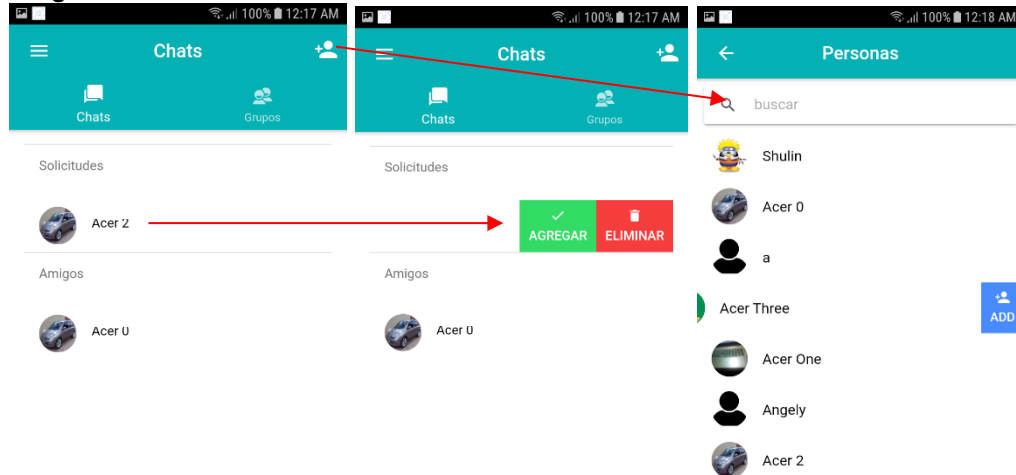


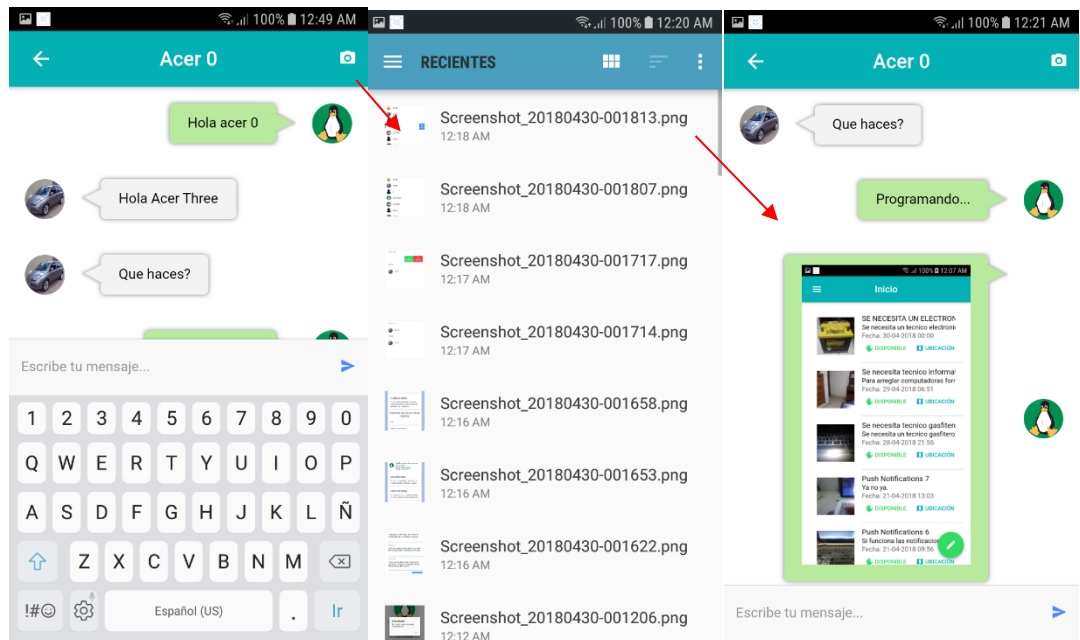


Fuente: Elaboración Propia

- Sala de Chats, en esta sección se gestiona las conversaciones y las solicitudes de amistad recibidas y aceptadas, tiene las siguientes acciones.
 - Visualización de las solicitudes de amistad, con opciones de aceptar o eliminar.
 - Visualización de los amigos y con acceso al chat.
 - Búsqueda de personas, con opción a enviar solicitud de amistad.

Imagen 93. Sala de Chats.

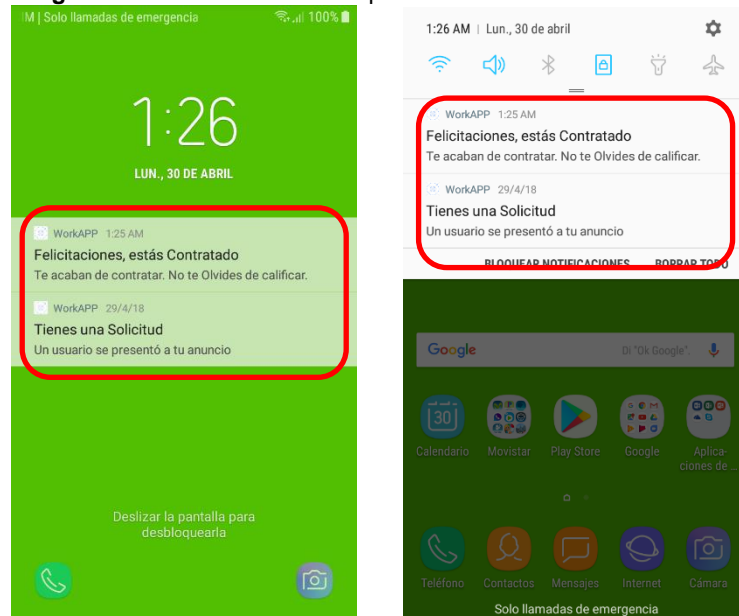




Fuente: Elaboración Propia

- Cerrar sesión es la acción que al ser seleccionada, destruye las credenciales del usuario y redirecciona a la página de login.
- La aplicación móvil hace una notificación al usuario de acuerdo al evento que se realizó.

Imagen 94. Notificaciones en la aplicación móvil



Fuente: Elaboración Propia