

# Modelo de desarrollo de aplicación móvil multiplataforma de anuncios de oferta y demanda laboral en tiempo real con con la metodología Mobile- D

## Multiplatform mobile application development model for job supply and demand announcements in real time with the Mobile-D methodology

1st Enrique Edgardo Condor Tinoco  
Universidad Nacional José María Arguedas  
Facultad de Ingeniería  
Andahuaylas, Apurímac, Perú  
enricoti@unajma.edu.pe, enricoti@gmail.com  
<https://orcid.org/0000-0002-2711-9517>

2nd Ricardo Richard Huamantingo Navarro  
Universidad Nacional José María Arguedas  
Andahuaylas, Apurímac, Perú  
rrichardh95@gmail.com  
<https://orcid.org/0000-0002-0300-4184>

3rd Carlos Yinnel Castro Buleje  
Universidad Nacional José María Arguedas  
Facultad de Ingeniería  
Andahuaylas, Apurímac, Perú  
ccastro@unajma.edu.pe  
<https://orcid.org/0000-0002-9368-3634>

4th Julio César Machaca Mamani  
Universidad Nacional José María Arguedas  
Facultad de Ciencias de la Empresa  
Andahuaylas, Apurímac, Perú  
jcmachaca@unajma.edu.pe  
<https://orcid.org/0000-0003-4866-3885>

**Resumen**— Existen diversas aplicaciones móviles y su uso están cambiando la forma de interactuar de las personas, y para desarrollarlas existen una gran variedad de herramientas de desarrollo, y nos preguntamos ¿Cómo lo hago y con qué?, una respuesta a esto se encuentra en este artículo, 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 y producción, la solución que se presenta en el presente artículo es sobre una aplicación móvil que responda rápido a las peticiones, por esa razón se mejoró el modelo cliente servidor, también se optimizaron las funciones de la aplicación móvil en cuanto al consumo de datos, Se usaron las Notificaciones Push sin la dependencia de un servidor. Los resultados indicados responden a los objetivos: a) optimizar el tiempo de respuesta y petición de datos, con un modelo Cliente – Servidor más corto, b) desarrollar un módulo de calificación dinámico e interactivo y c) integrar Notificaciones Push sin la dependencia de un servidor. Para el desarrollo del Aplicativo, se usó la metodología Mobile-D y se aplicó en un tablón de anuncios de ofertas y demandas laboral en tiempo real con la finalidad de incrementar las posibilidades de encontrar anuncios actualizados.

**Palabras claves:** *Firestore, IONIC, Mobile-D, Notificaciones Push*

**Abstract**— There are various mobile applications and their use is changing the way people interact, and to develop them there are a wide variety of development tools, and we ask ourselves, how do I do it and with what?, an answer to this is found in this article, which shows the use of technologies such as IONIC and Firebase that together can result in a multiplatform mobile application without cost of implementation in the development and production phase, The solution that is presented in the This article is that the mobile application responds quickly to requests, for that reason the client server model was improved, the functions of

the mobile application were also optimized in terms of data consumption, Push Notifications were used without dependence on a server. The indicated results respond to the objectives: a) optimize the response time and data request, with a shorter Client - Server model, b) develop a dynamic and interactive qualification module and c) integrate Push Notifications without relying on a server. For the development of the Application, the Mobile-D methodology was used and it was applied to a notice board of job offers and demands in real time in order to increase the chances of finding updated announcements.

**Keywords:** *Firestore, IONIC, Mobile-D, Push Notifications*

### I. 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 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 este artículo, 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.

Rangel (2013) en su trabajo de investigación realizó una revisión bibliográfica sistemática que identifica aquellas particularidades del desarrollo de aplicaciones para dispositivos móviles. Tuvo como resultado de su validación la documentación de los ajustes necesarios del proceso de

desarrollo de aplicaciones para dispositivos móviles, el que se consideró por las siguientes razones: a) los grupos de trabajo usan metodologías ágiles como Mobile-D, para desarrollar aplicaciones para dispositivos móviles b) Mobile-D está optimizado para entregar una aplicación para dispositivos móviles completamente funcional en un corto periodo de tiempo [1].

Meneses (2016) en su investigación 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; opta por una metodología ágil que es Mobile-D, debido a que son más convenientes para la implementación y desarrollo de las aplicaciones móviles por las siguientes razones: a) usar Mobile-D es más conveniente, debido a la rapidez con la que cambia el entorno de las tendencias tecnológicas y del mercado y b) se trata de alcanzar iteraciones de desarrollo muy rápidos con equipos muy pequeños y reducidos de programadores [2].

Por otro lado, el tablón de anuncios o también conocida como cartelera, es un instrumento en el cual 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 [3]. De acuerdo con un estudio de Nielsen, 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. [4]

[5] Es una investigación que 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. Este trabajo 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.

La solución que se presenta en el presente artículo es que, la aplicación móvil responda rápido a las peticiones considerando que 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 [2].

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.

Esta solución fue 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

Por lo tanto, para el desarrollo del aplicativo móvil se presenta los objetivos siguientes.

- Optimizar el tiempo de respuesta y petición de datos, con un modelo Cliente – Servidor más corto.
- Desarrollar un módulo de calificación dinámico e interactivo.
- Integrar Notificaciones Push sin la dependencia de un servidor.

El desarrollo del aplicativo móvil multiplataforma mediante IONIC y Firebase haciendo uso de la metodología Mobile-D se aplicó en un tablón de anuncios de ofertas y demandas laboral en tiempo real con la finalidad de incrementar las posibilidades de encontrar anuncios actualizados.

## II. MARCO TEÓRICO

### A. *Tablón de Anuncios*

También conocido como Cartelera, es un tablero en el que se fijan avisos, noticias, etc. o panel en el que se colocan anuncios públicos e informaciones. [6]

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. [7]

### B. *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. [8]

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.

### C. *Dispositivo Móvil*

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. [9]

Las características de un dispositivo móvil según Fernández [9] son:

- 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.

### D. *Base de datos no relacional, 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. [10]

### E. *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)

### F. *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. [11]

### G. *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 Back end en la nube con una fuente de datos en tiempo real y librerías para acceder a ella desde aplicaciones web, iOS y Android. [12]

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.

### H. *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 Centro Técnico de Investigación de Finlandia 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 subsistemas 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.

### III. METODOLOGÍA DE DESARROLLO CON 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

#### A. Fase de Exploración

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

#### • Estableciendo actores

Se identificó a los actores y al equipo de desarrollo para este proyecto:

- **Jefe del Proyecto:** Enrique E. Condor T. Responsable de determinar la buena disponibilidad y viabilidad de los recursos necesarios en el proyecto.
- **Analista de Sistemas:** Ricardo R. Huamantigo N. Responsable del diseño de la infraestructura de la aplicación móvil, considerando los requerimientos funcionales, el alcance y las limitaciones que la aplicación tiene.
- **Programador:** Carlos Y. Castro B. Responsable de implementar la aplicación con la infraestructura diseñada por el analista de sistemas.

#### • Estableciendo Usuarios

Se identificó 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.

#### • 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 I. INFORMACIÓN CLAVE DEL PROYECTO

<b>NOMBRE DE LA APLICACIÓN</b>	WorkApp (El cual nos indica que es una aplicación que está relacionada al trabajo)
<b>OBJETIVO GENERAL</b>	Desarrollar una aplicación, que permita hacer anuncios en Tiempo Real.
<b>PLATAFORMA</b>	El proyecto estará disponible para Android, IOS y Windows. Pero la aplicación final con la que se harán pruebas será Android.

Fuente: Elaboración propia

#### • 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.

#### • 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.

#### B. Fase de Iniciación

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

#### • Stack de Desarrollo

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

#### • Hardware de Desarrollo

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

- **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
- **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.
  - **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.
  - **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.
  - **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.
  - **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.
  - **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.
  - **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.
    - **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.
    - **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.
- **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.
- **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

### C. Fase de producción

En la fase de producción se diseñó la aplicación móvil de acuerdo a los casos de usos desarrollado 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 probada.

Para cada nueva iteración que se hizo, se realizaron 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.

### • Diseño de la aplicación Móvil

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

TABLA II. ESTRUCTURA DEL DISEÑO DE LA APLICACIÓN MÓVIL EN TÉRMINOS GENERALES

<b>Acceso a la aplicación</b>	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.
<b>Página Principal</b>	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.
<b>Administrar anuncios</b>	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.
<b>Configuración de Perfil</b>	Sección donde se configura, edita, actualiza y modifica la información referente al usuario autenticado.
<b>Sala de Chats</b>	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.
<b>Providers (Servicios)</b>	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.
<b>Firebase</b>	Integración del proyecto y de la aplicación móvil con la plataforma Firebase de Base de Datos en Tiempo Real.

Fuente: Elaboración Propia

### • 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.

```

"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"
}

```

Fig. 1 Dependencias de paquetes de la aplicación móvil.

Fuente: Elaboración Propia

- **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 sub-carpetas 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.

TABLA III. ESTRUCTURA DE DIRECTORIO Y ARCHIVOS DEL PROYECTO

<b>Hooks</b>	Contiene el archivo README.md creado por IONIC
<b>Node_modules</b>	Contiene los paquetes y dependencias del proyecto como: ionic-angular, @ionic-native/..., @angular/..., angularfire2, firebase, entre otros.
<b>Platforms</b>	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.
<b>Plugins</b>	Contiene subcarpetas de cada plugin de cordova u otros que se instalaron en el proyecto, con sus respectivos archivos.
<b>resoruces</b>	Contiene los iconos por defecto de la aplicación que se compilará y la vista SplashScreen de carga de la aplicación.
<b>src</b>	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.
<b>www</b>	Contiene archivos javascript en una carpeta build e imágenes estáticas para el icono de la aplicación y el SplashScreen, por default.
<b>GoogleService-Info.plist y google-services.json</b>	Estos dos archivos tienen que ser descargados desde la consola web de firebase y copiados a la carpeta raíz del proyecto. El archivo <b>GoogleService-Info.plist</b> es el archivo necesario para correr en la plataforma ios y <b>google-services.json</b> es el archivo necesario para correr en la plataforma Android.
<b>Package.json</b>	Es un archíve 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.
<b>.editrconfig, .gitignore, config.xml, ionic.config.json, package-lock.json, README.md, tsconfig.json, tslint.json</b>	Todos estos archivos son creados y usados por el framework IONIC 3.

Fuente: Elaboración Propia

#### D. Fase de Estabilización

Se procedió a integrar las funcionalidades implementadas en diferentes módulos y de presentarse algún error se realizarán 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.

- **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 procedió 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 comando: npm install angularfire2 firebase-save
- ✓ **Tercero:** acceder a la consola de firebase y crear un nuevo proyecto.
- ✓ **Cuarto:** copiar el script de configuración proporcionado por firebase a nuestro archivo de configuración, solo lo que contiene la variable config.
- ✓ **Quinto:** importar e inicializar con la configuración en app.modules.ts (import {AngularFireModule} from 'angularfire2') y inicializar con la configuración (AngularFireModule. initilizeApp(Config))

- **Integración de los módulos de Acceso a la Aplicación**

- ✓ **Primero:** se establece como página raíz a LoginPage en app.component.ts

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

Se tiene un archivo login.ts, donde se da acceso a los demás formularios que forman parte de esta integración (Signup, Passwordreset, Inicilogin).

- **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.

Se tiene un 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 importaciones

- **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.

- **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 un archivo de tipo .ts

- **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.

- **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.

- **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.

La estructura que se usa se muestra en la siguiente figura:



Fig. 2 WorkApp Database.  
Fuente: Elaboración Propia

### E. Fase de pruebas

En esta última fase de la metodología Mobile-D, se realizó 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.

## IV. RESULTADOS

Los interfaces de inicio de la aplicación móvil después de abrirse 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.

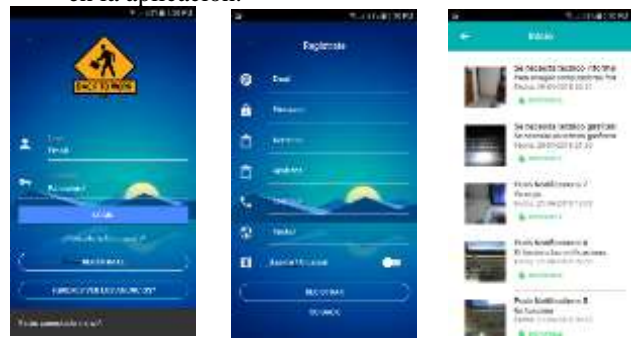


Fig. 3 Acceso a la aplicación  
Fuente: Elaboración Propia

Tras haber desarrollado y ejecutado el presente proyecto de investigación tecnológica se logró los objetivos siguientes:

### Objetivo 1, Optimizar el tiempo de respuesta y petición de datos, con un modelo Cliente – Servidor más óptimo:

En el modelo Cliente – Servidor tradicional se tiene tres capas (Cliente – Servidor Web – Servidor BD), en el desarrollo de la aplicación, para cubrir las necesidades y requerimientos de la aplicación, se consiguió un modelo de dos capas (Cliente - ServidorBD), como se observa en la figura.

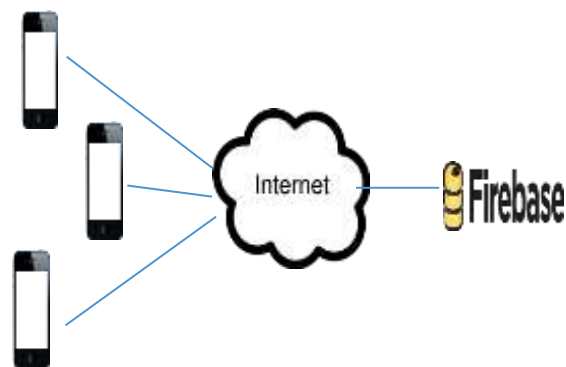


Fig. 4 Modelo Cliente - Base de Datos (dos pasos)  
Fuente: Elaboración Propia



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.

### **Objetivo 2, Desarrollar un módulo de calificación dinámico e interactivo:**

Se implementó en la aplicación módulos de calificación y gestión de puntuación por medio de estrellas, se tomó como referencia las puntuaciones de los grandes medios sociales y sobre ello se hizo un modelo de calificación más dinámico tanto para el usuario, así como para el anuncio, que garantice la confiabilidad entre los usuarios.

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.

### **Objetivo 3, Integrar Notificaciones Push sin la dependencia de un servidor:**

Las Notificaciones Push (Push Notifications) son mensajes que se envían de manera directa a dispositivos móviles y para ser implementados tradicionalmente es necesario tener un servidor web, pero implementar dicho servidor va en contra del objetivo “Optimizar el tiempo de respuesta y petición de datos, con un modelo Cliente – Servidor más óptimo” que se planteó.

Entonces para solucionar el inconveniente del servidor web se usó FCM (Firebase Cloud Messaging) de Firebase, el cual es una solución de mensajería instantánea que te ayuda a enviar mensajes de manera rápida, segura y gratuita, que está orientada para aplicaciones nativas, sin embargo, mediante el plugin de Cordova “cordova-plugin-fcm” se logró que una aplicación híbrida implemente esta funcionalidad sin la necesidad de que un servidor web gestione las notificaciones.

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.

## V. CONCLUSIONES

- Es posible mejorar el modelo cliente servidor de cuatro pasos (Cliente – Servidor Web – Servidor BD) de ida y vuelta a tan solo dos pasos (Cliente –Servidor BD) de ida y vuelta, con la ausencia de un servidor Web, 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.

- La implementación de módulos de calificación y gestión de puntuación de forma dinámica e interactiva contribuye a la confiabilidad de los usuarios de la aplicación, tanto al perfil de la persona como en el anuncio publicado, más aún si estas puntuaciones tienen como referencia a los medios sociales más usados en la actualidad.
- 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.

## VI. REFERENCIAS

- [1] I. M. Rangel Villagran, «Proceso en el desarrollo de aplicación para dispositivos móviles,» Mexico, 2013.
- [2] J. Meneses Sanchez y E. Laveriano Meca, «Prototipo de aplicación móvil utilizando la metodología Mobile-D para la verificación de la formalidad en el servicio de taxi metropolitano en la ciudad de Lima,» Lima, Perú, 2016.
- [3] Perú21, «Empleo decente es difícil de conseguir para el 80% de jóvenes peruanos,» 25 Octubre 2016. [En línea]. Available: <https://peru21.pe/economia/decente-difícil-80-jovenes-peruanos-231630>.
- [4] Agencia de Prensa Lima Norte, «En 14% se incrementó el número de jóvenes que busca empleo por Internet,» 20 Agosto 2017. [En línea]. Available: <http://agenciadepresalimanorte.com/en-14-se-incremento-el-numero-de-jovenes-que-busca-empleo-por-internet/>.
- [5] J. Cornejo Aramayo, «Análisis, diseño e implementación de una aplicación para administrar y consultar avisos clasificados para tabletas Android,» Lima, Perú, 2013.
- [6] Real Academia Española, 20 Agosto 2017. [En línea]. Available: <http://dle.rae.es/srv/fetch?id=YtSuzm8>. [Último acceso: 18 Agosto 2017].
- [7] Wikipedia, «Tablón de Anuncios,» 5 Febrero 2017. [En línea]. Available: [https://es.wikipedia.org/wiki/Tabl%C3%B3n\\_de\\_anuncios](https://es.wikipedia.org/wiki/Tabl%C3%B3n_de_anuncios). [Último acceso: 19 Agosto 2017].
- [8] Fundación Telefónica, Real time web: Una nueva conciencia global, Madrid: Ariel, S.A, 2011.
- [9] J. Fernandez, «Tipos de dispositivos móviles,» 1 Septiembre 2006. [En línea]. Available: [http://leo.ugr.es/J2ME/INTRO/intro\\_4.htm](http://leo.ugr.es/J2ME/INTRO/intro_4.htm). [Último acceso: 10 Agosto 2017].

- [10] Amazon, «¿Qué es NoSQL?», 2018. [En línea]. Available: <https://aws.amazon.com/es/nosql/>.
- [11] IBM, «Modelo de Cliente/Servidor», 2018. [En línea]. Available: [https://www.ibm.com/support/knowledgecenter/es/SSAL2T\\_9.1.0/com.ibm.cics.tx.doc/concepts/c\\_clnt\\_sevr\\_model.html](https://www.ibm.com/support/knowledgecenter/es/SSAL2T_9.1.0/com.ibm.cics.tx.doc/concepts/c_clnt_sevr_model.html).
- [12] Ionic Framework, «The top open source framework for building amazing mobile apps», 2017. [En línea]. Available: <https://ionicframework.com/>. [Último acceso: 22 Agosto 2017].
- [13] Google Developers, «Firebase», 31 Mayo 2017. [En línea]. Available: <https://firebase.google.com>. [Último acceso: 23 Agosto 2017].