

Desarrollo de una app (chat) con Phonegap

Daniela Alvarado M.
danielva06@gmail.com

Marian Solís C.
malusoce96@gmail.com

María José Madrigal G.
majomadrig5@gmail.com

Joseph Vega V.
jsphvega01@gmail.com

*Administración de Tecnología de Información
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica*

ABSTRACT

This document shows a description of chat application created with phonegap, including the architecture of phonegap, firebase platform, justification of use of different information repositories, etc.

KEY WORDS

Phonegap, firebase platform, aplicacion, Cloudinary.

I. INTRODUCCIÓN

Con la aparición de los dispositivos móviles incluyendo smartphone, tablets, etc., Acceder solo por medio de la web no era suficiente y debía acoplarse a este tipo de dispositivo, lo que dio lugar a las aplicaciones móviles.

Tomando en cuenta la forma en que la sociedad se conecta a la red, ha cambiado radicalmente en los últimos años.

Para la creación de este chat, se realizó una aplicación web (HTML-CSS-JAVASCRIPT) que correspondía al chat, suscripción de usuarios y subida de fotografías por cada usuario.

Utilización de repositorios no SQL como Firebase Real time Database y Cloudinary.

Push notification usando Phonegap push notification y también para la construcción de la aplicación móvil (de web a híbrida)

II. JUSTIFICACIÓN DE REPOSITORIOS DE INFORMACIÓN.

Para esta aplicación se utilizaron dos tipos de bases de datos no SQL, a continuación, se explica las razones de su escogencia.

Para el almacenamiento de los mensajes del chat y la información de los usuarios, se utiliza una base de datos real time de la plataforma Firebase Google, que es una base de datos no SQL que se encuentra alojada en la nube donde los datos se almacenan en formato JSON y se sincronizan en tiempo real.

Las razones por las que se escogió este repositorio son:

- Es apta para acoplarla con aplicaciones multiplataforma iOS, Android, lo que permite que los usuarios puedan accederlo desde su móvil y
- Todos los clientes comparten una misma instancia de la base de datos y reciben actualizaciones de forma automática.
- Es capaz de almacenar mucha cantidad de datos, lo que conviene debido al uso masivo de la aplicación.
- La API real time database está diseñada para permitir operaciones que se pueden ejecutar de forma rápida. Permite crear una gran experiencia en tiempo real que le puede servir a millones de usuarios sin comprometerse con la capacidad de respuesta.

- Por la facilidad para el desarrollador a la hora de usar la base de datos.

Las estructuras de JSON son ideales para la creación de aplicaciones de chat y el almacenamiento de los datos de los usuarios como se muestra a continuación, ya que son datos anidados y no tienen una estructura tan marcada como las bases de datos relacionales:

```
{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      "contacts": { "ghopper": true },
    },
    "ghopper": { ... },
    "eclarke": { ... }
  }
}
```

Fig. 1. Estructura de árbol en un JSON.

Donde el nodo es usuario y desde ahí se desprende las ramificaciones



Fig. 2. Estructura del chat

La base de datos se llama chat-68c9c y de ahí se desprende un nodo llamado chat conformado por cada mensaje enviado por los usuarios.

Para el almacenamiento de las fotografías de los usuarios se utiliza Cloudinary.

Ya que es un servicio web que ofrece a los usuarios gestionar imágenes, mediante el uso de la API.

Permite a los usuarios subir imágenes fuera de línea o en línea, y luego añadirlos a la biblioteca de imágenes que se almacenan en el mismo.

Es claro, que para el almacenamiento de este tipo de datos se necesitaba una base que fuera no SQL, Cloudinary es una buena opción porque se enfoca en la gestión de imágenes, audio y video, se acopla con las aplicaciones web mediante el uso de API, como se mencionó anteriormente. Además, todas las imágenes se pueden subir desde los dispositivos a la nube.



Fig. 3. Subida de archivos a Cloudinary

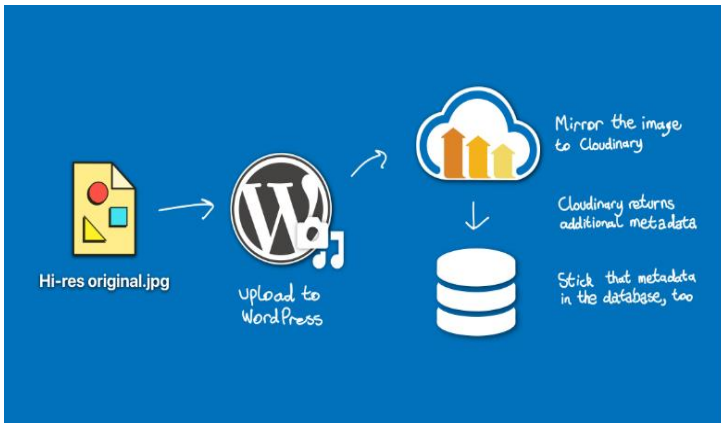


Fig. 4. Arquitectura de Cloudinary

Como se puede observar en la imagen anterior la arquitectura que sigue Cloudinary para subir algún recurso audiovisual, donde la nube se presenta como el punto medio entre la subida y la base de datos, en la cual se guardan los recursos audiovisuales.

III. CREACION DE LA APLICACIÓN CON PHONEGAP.

- **Backend de la aplicación:**

Para el almacenamiento de datos se utiliza los dos repositorios que se explicaron anteriormente, la plataforma Firebase provee un servicio muy completo incluyendo la base de datos en real time.

La plataforma en la nube, de Cloudinary para la gestión y procesamiento de las fotografías.

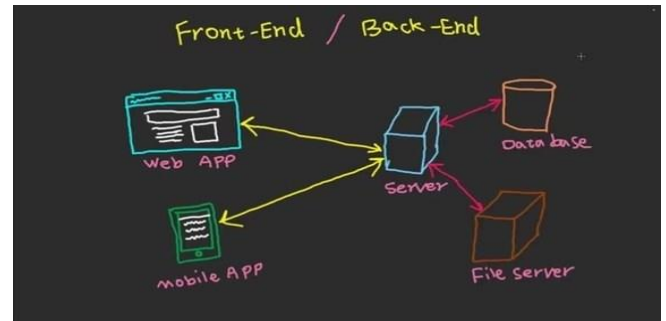


Fig. 5. Ejemplo de estructura de backend

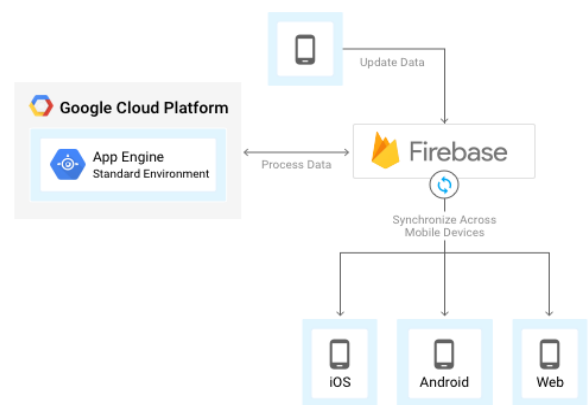


Fig. 6. Arquitectura de Firebase.

Se puede observar como Firebase permite la sincronización de la información para todos los dispositivos de las diferentes plataformas y utiliza la plataforma de Google Cloud para procesar la información.

- **Creación de la aplicación web.**

Para el desarrollo de la aplicación se decidió hacerla siguiendo el formato de HTML-CSS-JAVASCRIPT.

A continuación, se muestra en que consiste la aplicación del chat y las pantallas que lo componen.

Esta aplicación permite iniciar sesión, registrarse (ingresar texto y subir imágenes) y la participación en el chat con otro usuario.



Fig. 7. Estructura de la aplicación web

Para la aplicación se crearon las siguientes pantallas:

Login: El usuario deberá ingresar un usuario y su contraseña

En esta pantalla aparece un botón de olvido su contraseña y de registrarse.



Fig. 8. Pantalla de inicio de sesión.

Registrarse: El usuario debe ingresar sus datos personales como nombre, apellido, fecha de nacimiento, correo electrónico, usuario.

Así como la opción de cargar una foto.

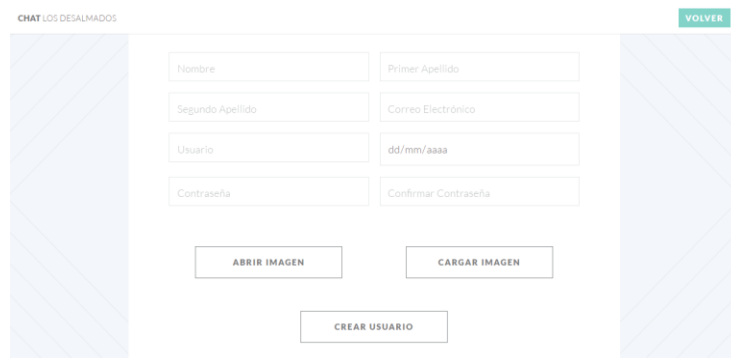


Fig. 9. Pantalla de inicio de registrarse

Chat: El usuario podrá enviar mensajes a otro usuario registrado y recibir mensajes también, en la aplicación

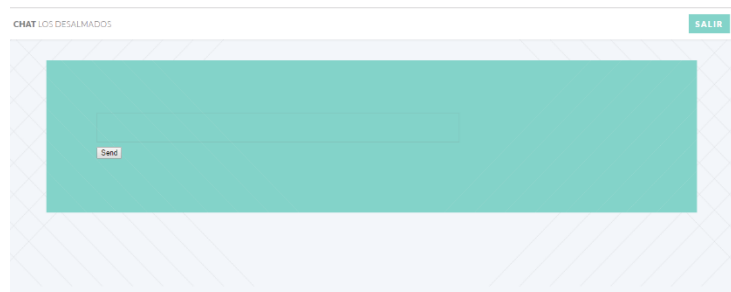


Fig. 10. Pantalla del chat.

- **Construcción de la aplicación híbrida con Phonegap.**

PhoneGap ofrece la posibilidad desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3.

Produciendo aplicaciones híbridas, es decir que no son realmente aplicaciones nativas al dispositivo (ya que el renderizado se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema).

A continuación, se muestra la arquitectura que sigue Phonegap, para la creación de aplicaciones híbridas a partir de una aplicación web.

Phonegap - Arquitectura

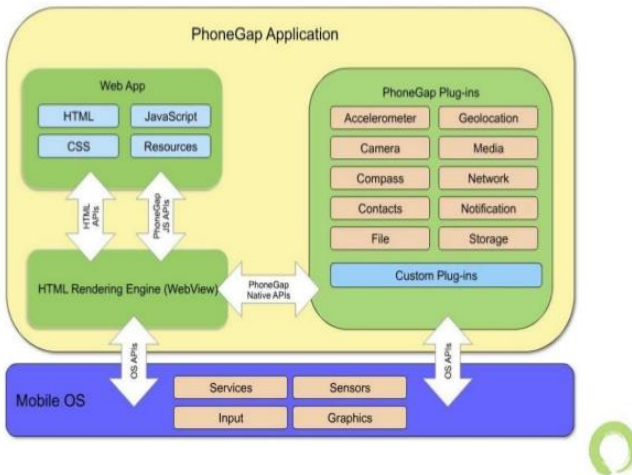


Fig. 11. Arquitectura de Phonegap

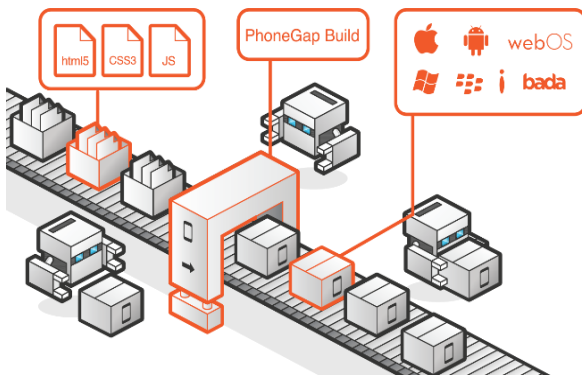


Fig. 12. Funcionamiento de Phonegap

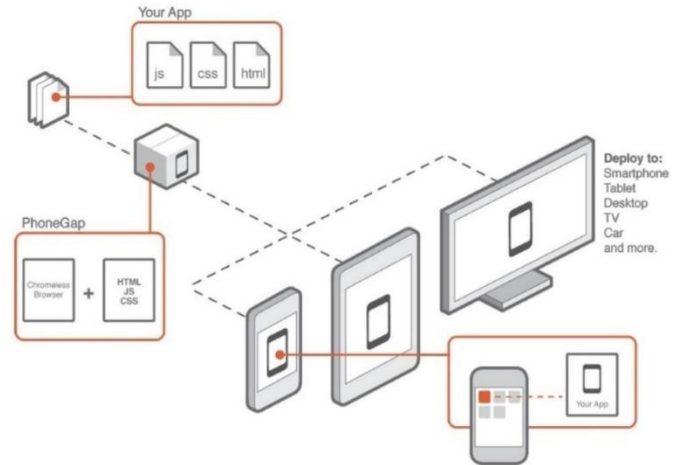


Fig. 13. Funcionamiento de Phonegap

Como se muestra en las imágenes anteriores, phonegap se encarga de encapsular una simple aplicación web y convertirla en una aplicación móvil híbrida que pueda ser utilizada desde cualquier dispositivo sea IOS, Android.

Donde el programador solamente desarrolla una aplicación web genérica y phonegap se encarga de lo demás.

El usuario podrá ver la aplicación como si hubiera sido diseñada para ese dispositivo en específico, pero en buena teoría es un código base que se acopla a cualquier plataforma móvil.

En vez de aprender las librerías propias de cada sistema con sus lenguajes, se hace uso de aquellas que proporciona el framework, usando un único lenguaje de programación, JavaScript

Por ejemplo, en vez de comunicarse directamente con la cámara, con las librerías de PhoneGap se comportan como especie de puente o interfaz y permiten interactuar con la cámara, hablando en el idioma que el sistema operativo del dispositivo requiere.

¿Cómo se utiliza Phonegap?

Primero debe accederse a la página de <https://build.phonegap.com/apps>, debe tenerse creado un Adobe ID, para poder tener una aplicación con Phonegap.

Se adjunta un link del repositorio de GitHub donde está la aplicación web implementada y se le asocia un nombre y descripción.

Cada vez que haga actualizaciones en el repositorio de GitHub, debe darle update code en la página de phonegap.

Para utilizar la aplicación desde el teléfono, solamente seleccione la plataforma y descargue el apk e instale en su dispositivo móvil o leyendo el código QR desde el dispositivo.

Así de simple, crea una aplicación híbrida con Phonegap.

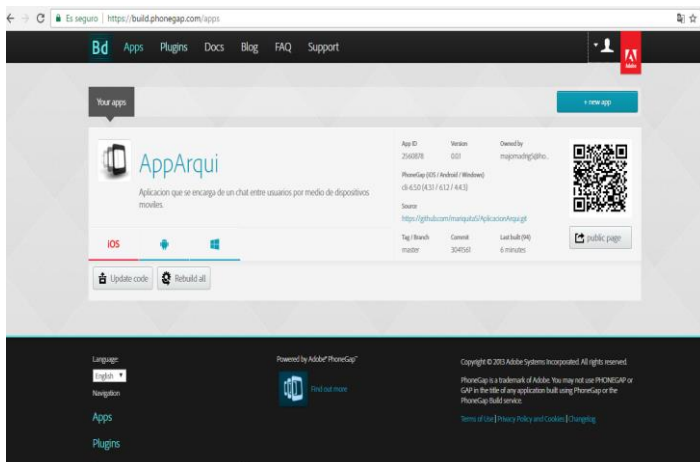


Fig. 14. Aplicación híbrida con Phonegap.

• Modelo 4 vistas +1

A continuación, se muestran el modelo 4 vistas +1 de Phonegap.

✓ Vista lógica: Diagrama de comunicación:

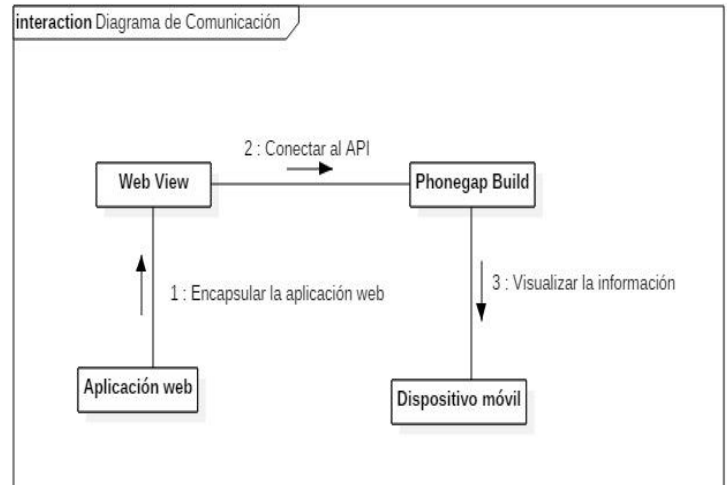


Fig. 15. Diagrama de comunicación Phonegap

✓ Vista proceso: Diagrama de actividad:

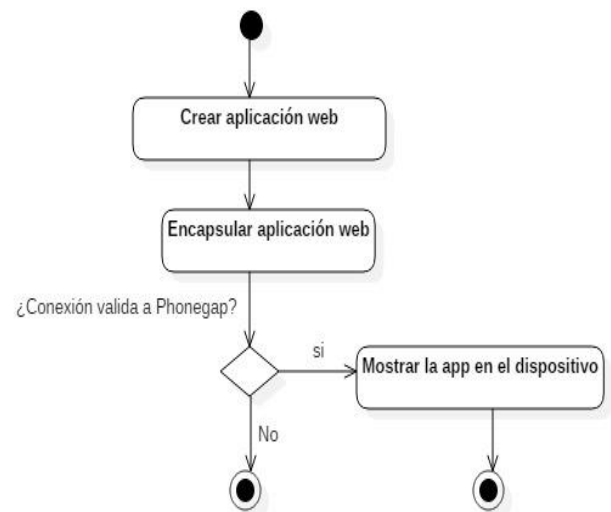


Fig. 16. Diagrama de actividad Phonegap

✓ **Vista desarrollo:**
Diagrama de componentes:

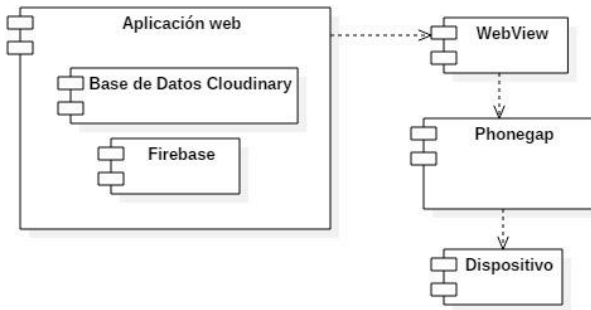


Fig. 17. Diagrama de componentes Phonegap

✓ **Vista física:**
Diagrama de despliegue

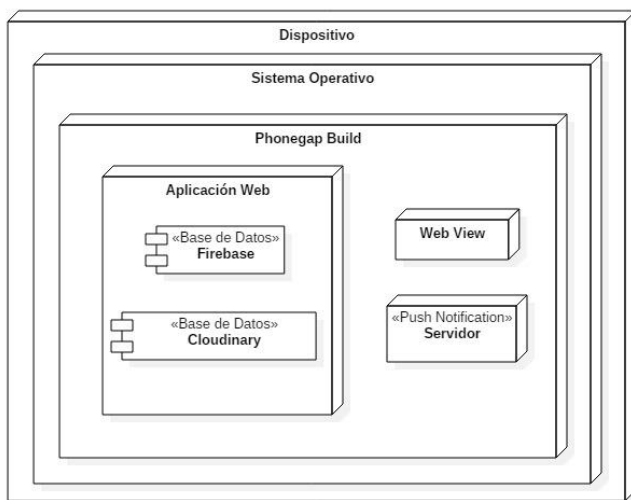


Fig. 18. Diagrama de despliegue Phonegap

✓ **Escenarios.**
Casos de uso:

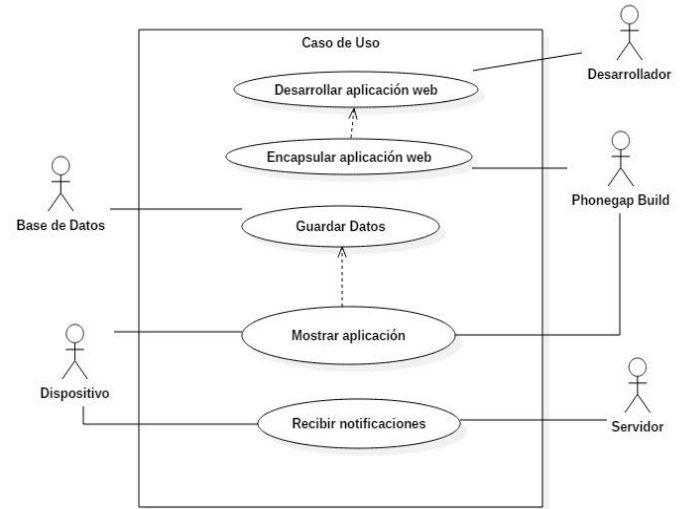


Fig. 19. Caso de uso Phonegap

• **Phonegap Push Notification:**

Las push notification son mensajes simples que se reciben de las aplicaciones instaladas en el dispositivo.

Phonegap dispone de un complemento de push notification para las aplicaciones con phonegap.

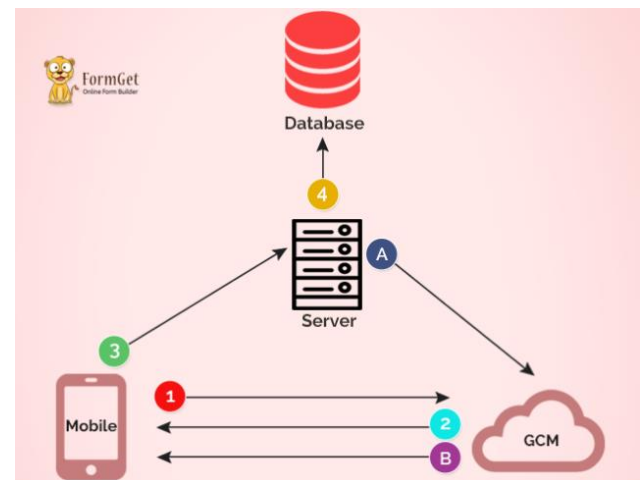


Fig. 20. Phonegap Push notification.

Los componentes que se podrían utilizar para implementar PhoneGap Push Notification se encuentran:

PhoneGap Plugin Push: Un complemento para implementar la notificación push en una aplicación PhoneGap.

Servicio Google Cloud Messaging (GCM): Google Cloud Messaging para enviar notificaciones push.

Phonegap provee un plugin para poder realizar push notification para los diferentes dispositivos.

Entre los proveedores de push notification se pueden encontrar:

- Android – Google Cloud Messaging Services (GCM)
<https://developers.google.com/cloud-messaging/>
- iOS Devices – Apple Push Notification Service (APNS)
<https://developer.apple.com/notifications/>
- Microsoft – Microsoft Push Notification Services (MPNS)
[https://msdn.microsoft.com/en-us/library/windows/apps/ff402558\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/ff402558(v=vs.105).aspx)

Fig. 21. Proveedores de push notification.

Se puede dar en dos pasos:

El registro y el envío de las notificaciones.

Se puede crear un proyecto de push notification con phonegap, como se muestra a continuación.

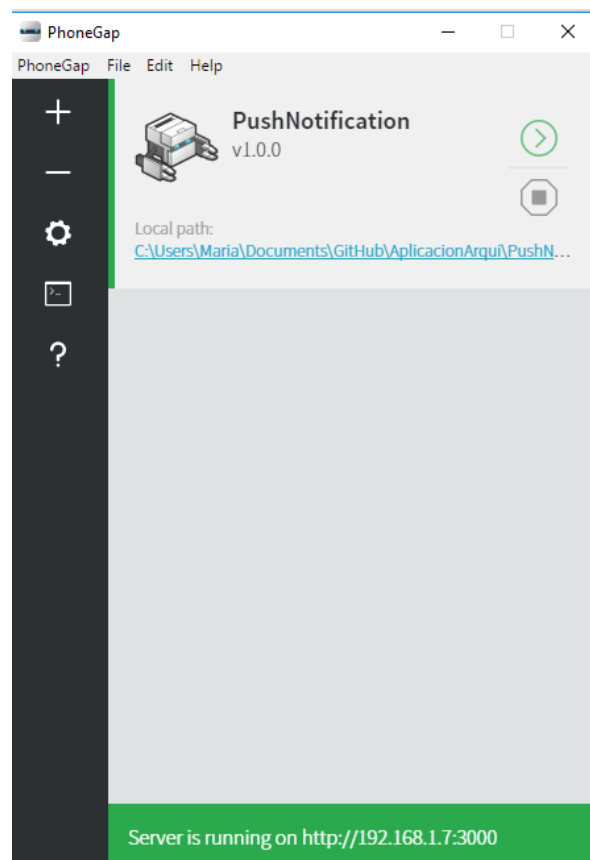


Fig. 22. Proyecto de Push notification con phonegap.

Lo que permite es que cualquier dispositivo pueda conectarse a ese servidor y registrarse por medio de un ID y Phonegap se conecte con alguno de los proveedores como GCM, APNs.

Se puede descargar una aplicación que permite que el dispositivo con solo ingresar el número del servidor, conectarse con el servidor de push notification.

IV. CONCLUSIONES

- Es recomendable usar phoneGap cuando se desarrolle una aplicación móvil para diferentes sistemas operativos, ya que esta plataforma resuelve

las necesidades de creación de aplicaciones de una manera única y compatible con todos los dispositivos.

- Para solventar este problema se desarrolló una aplicación híbrida ya que se requiere acceder a capacidades nativas del sistema operativo.
- Se concluye que Firebase es factible utilizarla cuando se ocupe una base No SQL en línea, que todos los clientes estén sincronizados en tiempo real y la información siga disponible cuando se pierda la conexión.

V. REFERENCIAS

- ✓ Adobe Systems Inc. (2017). *PhoneGap Documentation* | PhoneGap Docs Recuperado el 08 de Abril del 2017, de <http://webcache.googleusercontent.com/search?q=cache:http://docs.phonegap.com/>
- ✓ Baquero, J. (2015, 25 de Mayo). *¿Qué es PhoneGap y para qué podemos utilizarlo?* - *blog.arsys.es* Recuperado el 08 de Abril del 2017, de <https://blog.arsys.es/que-es-phonegap-y-para-que-puede-servirnos/>
- ✓ Cloudinary (2017). *Cloudinary - Cloud image service, upload, storage* & CDN Recuperado el 08 de Abril del 2017, de <http://cloudinary.com/>
- ✓ Cloudinary (2017). *Upload images* | Cloudinary Recuperado el 08 de Abril del 2017, de http://cloudinary.com/documentation/upload_images
- ✓ Creative Commons Attribution 3.0 License (2016, 06 de Julio). *Firebase Cloud Messaging* | Firebase Recuperado el 08 de Abril del 2017, de <https://firebase.google.com/docs/cloud-messaging/>
- ✓ Creative Commons Attribution 3.0 (2016, 05 de Octubre). *Estructura tu base de datos* | Firebase Recuperado el 08 de Abril del 2017, de <https://firebase.google.com/docs/database/web/structure-data?hl=es>
- ✓ Creative Commons Attribution 3.0 (2016, 05 de Octubre). *Firebase Real-*

time Database / *Firebase* Recuperado el 08 de Abril del 2017, de <https://firebase.google.com/docs/database/?hl=es>

- ✓ MagnetBrains LLC (2017). *PhoneGap : How Push Notifications Work?* / *FormGet* Recuperado el 08 de Abril del 2017, de <https://www.formget.com/phonegap-push-notifications/>
- ✓ Nerlandia (2014). *Cloudinary: Una forma inteligente de gestionar y manipular imágenes en línea - Nerdilandia* Recuperado el 08 de Abril del 2017, de <http://www.nerdilandia.com/cloudinary-una-forma-inteligente-de-gestionar-y-manipular-imagenes-en-linea>
- ✓ Scio Consulting (2014, 17 de Octubre). *Aplicaciones Móviles Híbridas* Recuperado el 08 de Abril del 2017, de <https://es.slideshare.net/ScioMx/aplicaciones-moviles-hbridas-v>