

## Capítulo 3

### 3.1 Modelado de la solución

#### 3.1.1 Solución Planteada

Se pretende diseñar e implementar un sistema gestor de eventos que permita la administración de los **eventos, participantes y beacons**, así como la experiencia de usuario para obtener información de los eventos con un enfoque en la proximidad usando dispositivos beacons. Este sistema será desarrollado como una aplicación sin servidor, es decir con un Backend en la nube para lo cual se ha elegido la plataforma **Firebase**, la cual provee servicios importantes que pueden ser usados en el sistema. Para esto se plantea dos módulos principales que componen el sistema los cuales serán:

- **App Admin:**

Esta aplicación será únicamente accesible de parte de un administrador y contara con diferentes funciones que permitan tener un control de los principales puntos del sistema, a continuación, se detallan las funcionalidades que permitirá el aplicativo para el administrador:

- Administrar Usuarios
- Administrar Beacons
- Administrar Eventos
- Monitoreo de usuarios
- Monitoreo de beacons

#### **Herramientas del módulo Admin:**

- **Angular.js:** Se utilizará este framework de trabajo de la empresa Google para poder desarrollar la capa de frontend que trabaja eficientemente con la plataforma de Backend en la nube que se explicará en el siguiente apartado.

- **Firestore:** Esta plataforma nos permitirá contar con un Backend en la nube que nos proporcione almacenamiento en una base de datos en tiempo real, hosting, y autenticación.
- **App Usuario:**

Esta aplicación tendrá un enfoque en la experiencia de usuario, es decir será desarrollada para el usuario la utilice. La aplicación del usuario deberá ofrecer las siguientes funcionalidades:

  - Administración de perfil
  - Detección de beacons por proximidad
  - Despliegue de información de eventos y participantes
  - Inscripción del usuario en evento

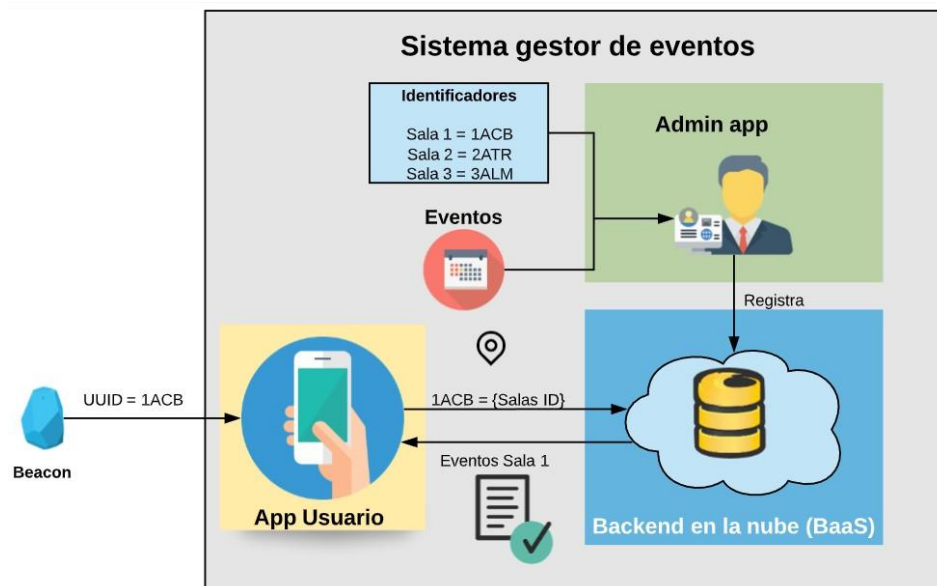
#### **Herramientas del módulo Usuario:**

- **Flutter:** Se utilizará este framework para el desarrollo de la aplicación de usuario la cual permitirá crear una aplicación Híbrida, es decir que funcionará tanto en Android como en iOS.
- **Firestore:** usará la misma base de datos en tiempo real para las peticiones de información sobre los beacons, eventos y usuarios.

### **3.1.2 Diseño del sistema**

Para el desarrollo del sistema se aplicará la metodología DevOps, por lo que este se desarrollará en un entorno de programación, de integración continua y entrega continua. Si bien es cierto la metodología con la que se va a trabajar está más orientada al pronto desarrollo de componentes y módulos que abarca un proyecto de software, pero para poder programar el sistema se requiere de un modelado previo para tener un panorama claro, tanto de los componentes que se van a programar, así como también de las diferentes funcionalidades que este va a contener. Es por eso que se realizó un análisis y un diseño previo para poder tener un esqueleto que servirá de apoyo al momento de programar el sistema. Los diagramas que se muestran a continuación corresponden a la estructura del sistema y sus diferentes funcionalidades.

### 3.1.2.1 Esquema Arquitectónico



**Fig 12.** Esquema Arquitectónico

*Elaborado por: Ayala A., Torres K.*

En la figura anterior podemos observar el esquema arquitectónico del sistema, en un flujo normal de trabajo, el que un beacon envía una señal de bajo consumo (BLE) hacia el aplicativo de usuario, esta señal enviara una información limitada que será la trama compuesta por el identificador del beacon (UUID), esta trama será recibida por la app usuario y está a la vez se conectará con el Backend en la nube Firebase, accederá directamente a la base de datos en tiempo real y consultará a que sala de conferencia pertenece el identificador del beacon detectado. Una vez que el beacon se haya detectado el aplicativo móvil mostrara la información de los eventos asignados a la sala que anteriormente se asignó a determinado beacon por un administrador. Dicho esto, queda claro que la app de usuario recibirá la información de los eventos, beacons y participantes siempre y cuando esta información haya sido registrada mediante el aplicativo de administrador en la base datos de Firebase.