

Development of a Mobile Platform for Appointment Management of a Hospital Using Bluetooth Low Energy Technology with External Devices – Beacons

Harold Ernesto Caceres Zea, Bch¹, Eveling Castro Gutierrez, Msc¹

¹ Universidad Nacional de San Agustín, Perú, hcaceresze@unsa.edu.pe, ecastro@unsa.edu.pe

Abstract– *Currently booking an appointment at the hospital is a very complicated task for patients. Over the years the demand for an appointment will continue to grow and the discomfort and dissatisfaction of patients will also grow, for this reason this work proposes the development of a mobile platform for booking appointments in a hospital using Bluetooth Low Energy for the connection with external devices - Beacons that can improve the satisfaction of the patients when booking an appointment and that allows to meet the high demand that will exist in a few years.*

Mobile technology is revolutionizing the way you shop, work, and stay informed. Given the expansion and the large number of devices with Android made us inclined in their choice as a development platform.

Developing mobile applications takes a different approach than developing desktop applications, mobile application developers must consider some restrictions. In this work the application will make use of the Bluetooth connection that has all the smart phones incorporated to detect the data packets sent by the beacons, once the application detects the ads of the beacons, it will launch the functionality associated with said announcement. The power consumption of the mobile device will be conditioned to the handling of application states and the activity period of the Bluetooth. A distributed database has been implemented to allow manipulating the data from another application in the future. Keywords-beacons, mobile applications, android, hospital appointment, Bluetooth low energy

Digital Object Identifier (DOI):<http://dx.doi.org/10.18687/LACCEI2018.1.1.131>
ISBN: 978-0-9993443-1-6
ISSN: 2414-6390

Desarrollo de una Plataforma Móvil para la Gestión de Citas de un Hospital Usando la Tecnología de Bluetooth Low Energy con Dispositivos Externos – Beacons

Harold Ernesto Caceres Zea, Bch¹, Eveling Castro Gutierrez, Msc¹

¹Universidad Nacional de San Agustín, Perú, hcaceresze@unsa.edu.pe, ecastro@unsa.edu.pe

Abstract– Currently booking an appointment at the hospital is a very complicated task for patients. Over the years the demand for an appointment will continue to grow and the discomfort and dissatisfaction of patients will also grow, for this reason this work proposes the development of a mobile platform for booking appointments in a hospital using Bluetooth Low Energy for the connection with external devices - Beacons that can improve the satisfaction of the patients when booking an appointment and that allows to meet the high demand that will exist in a few years.

Mobile technology is revolutionizing the way you shop, work, and stay informed. Given the expansion and the large number of devices with Android made us inclined in their choice as a development platform.

Developing mobile applications takes a different approach than developing desktop applications, mobile application developers must consider some restrictions. In this work the application will make use of the bluetooth connection that has all the smart phones incorporated to detect the data packets sent by the beacons, once the application detects the ads of the beacons, it will launch the functionality associated with said announcement. The power consumption of the mobile device will be conditioned to the handling of application states and the activity period of the bluetooth. A distributed database has been implemented to allow manipulating the data from another application in the future.

Keywords-- beacons, mobile applications, android, hospital appointment, bluetooth low energy.

I. INTRODUCCIÓN

En el año 2015, 4 462 226 personas de todas las edades fueron atendidas en los diferentes hospitales de un total de 1 301,298 pobladores de la ciudad de Arequipa. Actualmente reservar una cita en el hospital, es una tarea muy complicada para los pacientes, los obliga a esperar desde muy temprano en la ventanilla, realizar largas colas o llamar reiteradas veces en el transcurso del día y demoran por lo menos tres días para conseguirla, o en el peor de los casos, este proceso puede demorar hasta un mes. Existen muchas personas que no logran adquirir la cita a pesar de realizar grandes esfuerzos. Con el transcurso de los años la demanda por conseguir una cita seguirá creciendo y la incomodidad e insatisfacción de los

pacientes también crecerá, por ese motivo este trabajo propone el desarrollo de una plataforma móvil para la reserva de citas en un hospital usando Bluetooth Low Energy para la conexión con dispositivos externos - Beacons que pueda mejorar la satisfacción de los pacientes al reservar una cita y que permita atender la alta demanda que existirá en unos años.

La tecnología móvil está revolucionando la manera de comprar, trabajar, y mantenerse informado. En EEUU la gran mayoría de los adultos tiene ahora un dispositivo móvil, y casi uno de tres tiene un smartphone. El mercado de los dispositivos móviles está dominado a nivel de plataforma por dos grandes compañías, Google con su Android y Apple con su IOS. Queda fuera de este trabajo la comparación de ambos sistemas, con sus ventajas e inconvenientes.

Dada la expansión y la gran cantidad de dispositivos con Android, al ser una plataforma de código abierto, la facilidad y disposición de herramientas para desarrollar aplicaciones y la dominación del mercado móvil hizo que nos inclináramos en su elección como plataforma de desarrollo. Este explosivo crecimiento está siendo repetido en muchos otros países y se espera que un futuro cercano exista más dispositivos móviles que habitantes. En el Perú el 82% de la población tiene un smartphone, es por lo que este trabajo propone la creación de una aplicación móvil para gestionar la reserva de citas en un hospital ya que considera que existe una gran cantidad de usuarios potenciales para adoptar esta nueva forma de gestionar una cita en un hospital.

Desarrollar aplicaciones móviles tiene un enfoque diferente que el desarrollar aplicaciones para escritorio, los desarrolladores de aplicaciones móviles deben considerar las restricciones que tiene el dispositivo móvil, como la amplia variedad del tamaño de las pantallas y las especificaciones del hardware. En el desarrollo móvil existen dos factores claves; una es la interfaz de usuario y la otra es el uso eficiente de las capacidades del dispositivo como los sensores, cámaras, bluetooth, conexión con internet, etc.

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2018.1.1.131>

ISBN: 978-0-9993443-1-6

ISSN: 2414-6390

En este trabajo incluye la implementación de dos aplicaciones. Un backend server, que tiene la información de la planificación de citas en su base de datos y una aplicación móvil que hace uso de servicios para solicitar información al servidor. La aplicación también hará uso de la conexión bluetooth que tiene incorporado todos los smartphones para la detección de los paquetes de datos enviados por los beacons, una vez que la aplicación detecte los anuncios de los beacons, lanzará la funcionalidad asociada a dicho anuncio. El consumo de energía del dispositivo móvil estará condicionado al manejo de estados de la aplicación y al periodo de actividad del bluetooth.

El resto del trabajo está organizado como sigue: En la sección 2 se describen los trabajos relacionados. En la sección 3 mostramos la metodología de la aplicación. El desarrollo de la propuesta es mostrado en la sección 4. Y finalmente en la sección 5 se muestra las conclusiones de este trabajo.

II. TRABAJOS RELACIONADOS

En [1] explica las nuevas oportunidades que ofrece esta innovadora tecnología (beacon) en el marketing basado en la ubicación del usuario la cual también presenta un gran número de retos. El contenido creado basado en la ubicación de los consumidores será una gran demanda entre las agencias. Estas compañías también podrán aprovechar el conocimiento de que los anuncios móviles integrados en aplicaciones tienen un mejor rendimiento que los anuncios en la web. En [3] hace una comparación de las tecnologías WIFI y BLE, mostrando un comportamiento satisfactorio de BLE con sus respectivas características.

En [7] nos presenta una arquitectura para aplicaciones móviles, nos muestra que una solución integral tiene que abordar diferentes aspectos, tales como problemas para manejar dinámicamente la conexión a internet, la potencia, el almacenamiento y otras limitaciones de los dispositivos móviles. La arquitectura intenta ser más general con respecto a la adaptabilidad y flexibilidad. El autor comenta que una aplicación es exitosa cuando esta satisface los requerimientos de un cliente. Sin embargo, un framework puede ser exitoso cuando este puede ser personalizado para diferentes requerimientos. Por lo tanto, debemos notar que diseñar una arquitectura es un poco diferente que desarrollar una aplicación.

En [15] explica que hoy en día, los procesadores móviles y sistemas operativos están siendo desarrollados para optimizar el rendimiento y reducir el consumo de energía. Pero existen algunos procesos que no han sido optimizados para gestionar la energía en los dispositivos móviles. Por lo tanto nos presenta un sistema de gestión de energía considerando las características usadas en un smartphone. Los resultados experimentales mostraron que el modelo de administración de energía redujo el consumo de energía y el

tiempo de respuesta de la aplicación en ejecución en un entorno móvil multitarea.

En [9] explora el potencial de usar iBeacon como un sistema de posicionamiento interno para proporcionar al estudiante servicios e investigar las limitaciones y dificultades de usar iBeacon. Se implementó una aplicación para que los estudiantes puedan marcar su asistencia, también para mostrar la información relacionada al cuarto en el que el usuario se encuentra.

En [10] explica la creación de una aplicación para el registro de asistencia de los estudiantes y hace una comparación con las tecnologías existentes para el registro de asistencia (NFC, QR Codes) mostrando que para clases con más de 25 000 estudiantes, el tiempo que demora el registro mejora considerablemente usando beacons.

A. *Análisis de Aplicaciones*

En [1] nos explica algunas de las aplicaciones existentes en el mercado. Por ejemplo, Starwood Hotels Resorts ha implementado iBeacon en 30 hoteles para ayudar a los conserjes a saludar las llegadas VIP por nombre que genera una experiencia especializada. Un museo en Antwerp, Bélgica, ha aplicado la tecnología beacon para proporcionar a los visitantes experiencias personalizadas mientras exploran en su propio ocio con un guía turístico en mano. Finalmente en el 2014, Miami International Airport se convirtió en el primer aeropuerto en el mundo en instalar la tecnología beacon, proporcionando información que es relevante al pasajero.

II. METODOLOGÍA

A. *Bluetooth Low Energy*

Bluetooth Low Energy (BLE) es una tecnología emergente desarrollada por Bluetooth Special Interest Group (SIG) para un rango pequeño de comunicación. A diferencia de la versión previa de Bluetooth Classic, BLE ha sido diseñado como una solución de bajo consumo de energía para aplicaciones de control y monitoreo.

El uso generalizado de la tecnología Bluetooth (en smartphones, laptops, automóviles, etc.) puede impulsar la adopción de BLE, ya que la implementación de este último puede aprovechar las similitudes con el clásico Bluetooth [5].

Algunas características que posee BLE para el correcto funcionamiento de la aplicación propuesta, son las siguientes [3].

- Poder de consumo de energía: BLE puede durar en el rango de 2 a 6 años dependiendo del uso.

- Fácil de usar: Debe ser fácil asociar el dispositivo con BLE.
- Ecosistema disponible: La posibilidad de conectar con smartphones, PC, tablets, es importante.
- Rango de transmisión: Es necesario cubrir el rango de alcance suficiente sin tener un gran impacto en el costo del sistema.

Bluetooth Low Energy opera en la banda 2.4GHz con solo 40 canales y dos 2 MHz canales de espacio [3] y también comparte las mismas características de propagación en interiores que los transceptores WiFi de 2,4 GHz [2]. Tiene dos tipos de canales: canales de anuncios y canales de datos. Los canales de anuncios son usados para descubrir dispositivos, establecer conexión y emitir la transmisión por radiodifusión, mientras que los canales de datos son usados para una comunicación bidireccional entre dispositivos conectados. Cualquier dispositivo que transmite paquetes de anuncios es llamado “anunciante”. La transmisión de los paquetes a través de los canales toma lugar en intervalos de tiempos llamados eventos publicitarios. Dentro de un evento publicitario, el anunciante usa secuencialmente cada canal de anuncios para la transmisión de paquetes.

B. Beacons

Son dispositivos pequeños que permiten una localización más precisa que el GPS dentro de un rango estrecho, la triangulación de la torre celular y la proximidad Wi-Fi. Los beacons transmiten pequeñas cantidades de datos a través de BLE hasta 50 metros, y como resultado se usan a menudo para la tecnología de localización en interiores, aunque también se pueden usar beacons en el exterior.

La introducción de BLE beacons para detección de la proximidad proporciona una nueva señal de oportunidad para representar el posicionamiento de una forma más precisa. Así también esta nueva tecnología presenta una nueva oportunidad al marketing basado en la posición, enviando mensajes en donde el usuario está ubicado. Mientras esta tecnología innovadora también presenta una serie de desafíos: el mayor es la de los usuarios para adoptar los servicios de inserción en la tecnología basada en la ubicación.

La razón de que los Beacons sean considerados como parte de un importante desarrollo es que su función de micro posicionamiento es más precisa que el GPS. En el pasado, GPS solo podía dar una idea aproximada de la ubicación del usuario, pero con Beacon, el rango de posicionamiento se reduce precisamente a 2-100 metros, y es capaz de identificar cualquier dispositivo móvil que entra en el rango de transmisión de la señal [6].

La ventaja de la tecnología Beacon está en el uso de BLE. Con la promoción de Android 4.3 y Apple con la

compatibilidad con iBeacon, el protocolo que es será usado en muchos de los dispositivos móviles en el futuro. A diferencia de Wifi, no requiere conexión a Internet, solo requiere dos puntos para funcionar, como mostramos en la figura 1.

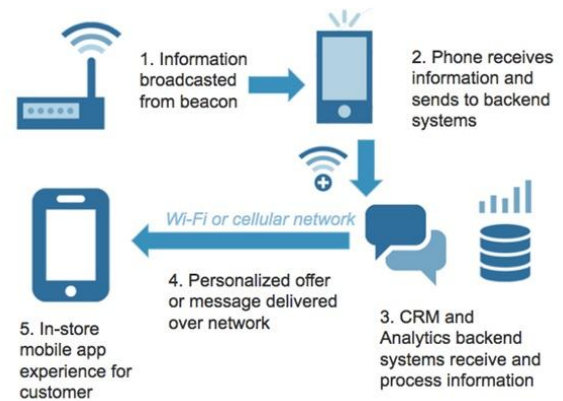


Fig. 1 Tecnología Beacon – BLE.

Podemos decir que el Beacon es como un faro que continuamente está emitiendo señales. Cuando un dispositivo móvil ingresa al rango de alcance del faro, el Beacon enviará una cadena de códigos hacia el dispositivo móvil. Después de que la aplicación en el móvil detecte el código, una serie de acciones serán lanzadas, como mostramos en la figura 1.

La señal de iBeacon contiene una dirección, la cual identifica a su emisor. Los componentes y uso común de cada componente son descritos en la tabla 1.

TABLE I
IBEACON COMPONENTS

Componente	Descripción	Uso común
UUID	128 bits	Identificar la compañía
Minor	16 bits, valores enteros, rango desde 0 hasta 65536	Identificar un edificio en la compañía
Major	16 bits, valores enteros, rango desde 0 hasta 65536	Identificar una región en el edificio. Ej.: una oficina en un edificio

C. Arquitectura de Android

Android es un conjunto de software para los dispositivos móviles que incluye un sistema operativo, middleware y una capa de aplicaciones. Google y otros miembros de Open Handset Alliance colaboran con el desarrollo y lanzamiento de Android. La arquitectura de Android y sus principales componentes son mostrados en la figura 2 [13].

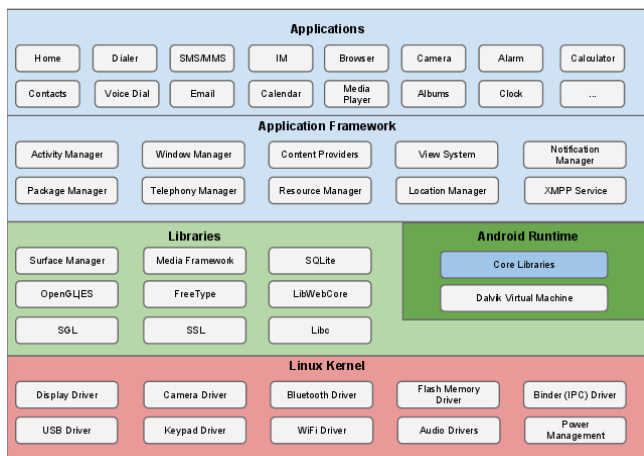


Fig. 2 Arquitectura de Android

Como se puede apreciar en la figura 2, Android consiste de un Linux Kernel, y algunas aplicaciones (ej. navegador, mapas) ejecutándose en java, un marco de aplicación orientado a objetos sobre las bibliotecas centrales que se ejecutan en una máquina virtual Dalvik. La máquina virtual Dalvik es una de las características más destacadas de Android. Especialmente, está optimizado para CPUs lentos, relativamente poca memoria y para ejecutar en el sistema operativo sin ningún espacio de intercambio.

Cada capa proporciona características diferentes como describimos a continuación [8]:

1) *Applications*: Posee un conjunto de principales aplicaciones como SMS, calendario, navegador, contactos y otros. Todas estas aplicaciones son desarrolladas en Java.

2) *Application Framework*: Proporciona todas las API's a los desarrolladores para reusar sus componentes.

3) *Libraries - Android Runtime*: La capa de librerías está dividido en dos componentes: Android Runtime que consiste de Java Core Library y Dalvik Virtual Machine y Android Library que pueden ser utilizadas por los diferentes componentes en el sistema.

4) *Linux Kernel*: El Linux kernel proporciona operaciones como almacenamiento interno, administración de los procesos, protocolos de internet.

D. Ciclo de Vida de una Aplicación Móvil

Android es una plataforma diseñada para dispositivos móviles. La pantalla, la batería y los recursos disponibles, entre otros aspectos, en los dispositivos nos condicionarán a la hora de desarrollar las aplicaciones. La mayoría de los dispositivos Android son equipos de comunicación, teléfonos en los que es posible realizar y recibir llamadas. Un usuario no espera cerrar una aplicación para contestar una llamada, espera que de forma automática se presente el interfaz del teléfono que le permita atenderla. Es por esto que el ciclo de vida de las

actividades estará condicionado por la interacción del usuario, así como otros posibles eventos que puedan ocurrir.

A diferencia de lo sucede en los ordenadores personales, el usuario únicamente lanza aplicaciones, no tiene la opción de finalizarlas. Es el propio sistema el que se encargará de finalizarlas cuando necesite recursos, pero cuando el usuario vuelve a la aplicación espera que esta se encuentre en el mismo estado en que la dejó. Así que las aplicaciones tienen un ciclo de vida que está controlado por el usuario y el sistema.

Las interfaces de usuario donde las ventanas se solapan han tenido una gran aceptación en los ordenadores personales, pero no son adecuadas para los dispositivos móviles donde la pantalla es reducida y debe primar la sencillez en la gestión gráfica. Por lo general Android utiliza una interfaz que ocupa toda la pantalla, aunque es posible mostrar notificaciones o diálogos que permiten ver parcialmente la pantalla sobre la que se han ejecutado.

En todo momento el usuario puede pulsar el botón del menú hacia atrás que le permite volver a la pantalla previa. Desde el punto de vista del usuario, una aplicación está formada por una pila de pantallas abiertas.

Android puede en cualquier momento pausar, parar, destruir nuestra aplicación según las necesidades de recursos del momento y el desarrollador debe controlar todos estos eventos para hacer una aplicación estable, eficiente y transparente al usuario.

De este modo, una actividad puede pasar por los siguientes estados mostrados en la figura 3:

- `onCreate()`: Este método se llama cuando la actividad ha sido creada.
- `onRestart()`: Llamada cuando tu actividad ha sido parada, antes de volver a ser reanudada.
- `onStart()`: Llamada justo antes de que la actividad vaya a ser visible por el usuario.
- `onResume()`: Se ejecuta en el momento en que la actividad se encuentra en la parte superior de la pila y la actividad interactúa con el usuario.
- `onPause()`: Se llama cuando el sistema va a empezar una nueva actividad, es decir, cuando se ha llamado al `onRestart()` de otra.
- `onStop()`: Se ejecuta cuando la actividad deja de ser visible al usuario.
- `onDestroy()`: Es la última llamada antes de destruir la actividad.

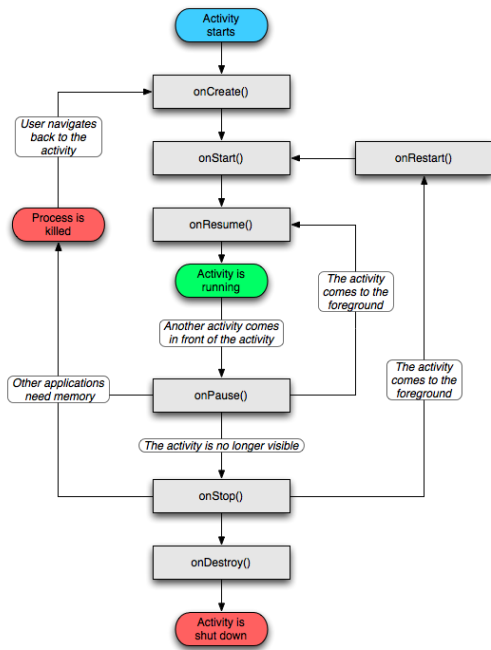


Fig. 3 Ciclo de vida de una aplicación móvil - Android.

E. Desarrollo de Aplicaciones Móviles

Existen muchas opciones que están disponibles para el desarrollo de aplicaciones móviles, lo podemos simplificar explicando tres maneras [4]:

- Aplicaciones Nativas: Construir la aplicación en código nativo para ejecutarla en un dispositivo con un sistema operativo específico.
- Aplicaciones Multiplataformas: Construir la aplicación usando un framework multiplataforma y ejecutarla en múltiples dispositivos con diferentes sistemas operativos.
- Aplicaciones Web Móvil: Construir la aplicación y ejecutarla sobre internet en un servidor central, puede ser accedida desde cualquier dispositivo con un navegador web.

La figura 4 muestra un resumen de las principales características de los 3 enfoques de desarrollo móvil.

Hay dos factores claves en el desarrollo de aplicaciones móviles, una es el diseño de la interfaz de usuario y la otra es la utilización eficiente de las capacidades del dispositivo móvil, tal como los sensores, cámaras y conexión a internet. El desarrollo de la aplicación móvil como método nativo es bueno para el rendimiento, ya que puede utilizar completamente la disponibilidad del kit de desarrollo y los recursos del sistema móvil con la ayuda del kit y la API [11].

Features	Native Apps	Mobile Web Apps	Hybrid Apps
Performance	High	Low	Medium
Offline Mode	Supported	Not Supported	May be Supported
Distribution	App Store	Mobile Browser	App Store
Cross Platform Support	No	Yes	Yes
Device Level Access	High	Low	Medium
User Interface	Good	Average	Good
Development language	Native only	Web only	Native / Web
Development Time	High	Low	Medium
Development Cost	High	Low	Medium
Code Portability	Low	High	High
Maintenance	High	Low	Medium

Fig. 4 Comparación de los enfoques de desarrollo de apps.

Para propósitos de nuestra aplicación muchas de las características mostradas solo las cumplen el enfoque de desarrollo nativo, por ese motivo para este estudio se desarrollará la aplicación solo para dispositivos Android.

IV. DESARROLLO DE LA PROPUESTA

Nuestro objetivo es desarrollar una aplicación de reserva de citas para Android, que sea fácil de usar y cumpla con todos los requerimientos.

A. Análisis de la aplicación

1) *Requerimientos no funcionales:* A continuación se detallarán los requisitos no funcionales.

- **Requerimiento no funcional 1:** El dispositivo debe tener Bluetooth, tenerlo activo y con la visibilidad activada, para que puede detectar la señal de los beacons.
- **Requerimiento no funcional 2:** El dispositivo debe tener conexión a internet para obtener los datos de la programación de especialidades del hospital.
- **Requerimiento no funcional 3:** El dispositivo debe tener una versión de Android 4.0 o superior, y 512 MB o más de memoria RAM.

2) *Requerimientos funcionales:* A continuación se detallarán los requisitos funcionales.

- **Autenticación:** La aplicación debe validar y aceptar peticiones del cliente.
- **Autorización:** La aplicación debe verificar que el usuario ha sido registrado. También debe prevenir que un usuario pueda ingresar a la aplicación con diferentes dispositivos.
- **Registro de usuario:** La aplicación debe tener un formulario de registro para nuevos usuarios.
- **Listar el historial de las citas registradas:** La aplicación debe mostrar al usuario el historial de las

reservas realizadas con la toda la información de la reserva.

- **Listar las especialidades disponibles:** La aplicación debe mostrar la planificación de las especialidades disponibles y el número de citas disponibles para la especialidad.
 - **Registrar cita:** La aplicación debe permitir al usuario realizar una reserva de cita que todavía no tenga el número máximo de pacientes disponibles para dicha especialidad. Para reservar una cita la aplicación debe retornar la información asociada de donde se encuentra el usuario. La aplicación detectará las señales que el beacon emite y las enviará al servidor. El servidor identificará los datos y retornará la información asociada al beacon.
- 3) *Casos de uso:* Mediante el diagrama de casos de uso de la figura 5 mostraremos de una forma gráfica la funcionalidad de todo el sistema y como el usuario de la aplicación (Actor) interactúa con el mismo.

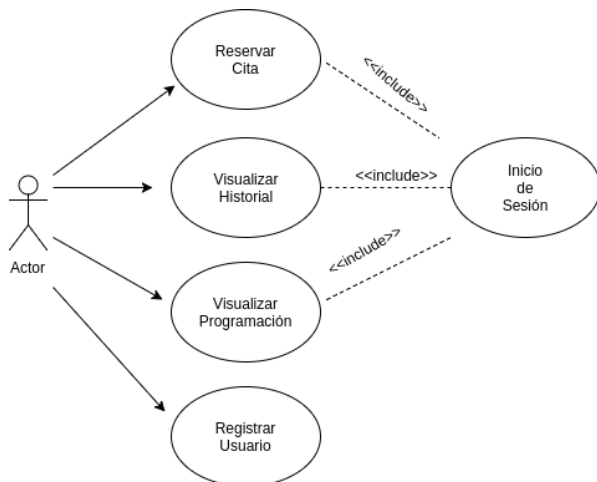


Fig. 5 Diagrama de casos de uso.

Como podemos ver en la figura 5 los casos de uso de Reservar Cita, Visualizar Historial, Visualizar Programación tienen la precondition de que el usuario debe haber iniciado sesión en la aplicación. Después de que usuario ha sido autorizado para ingresar a la aplicación se mostrará el menú con las opciones disponibles.

B. Arquitectura de la aplicación

La arquitectura de la aplicación es una arquitectura cliente-servidor como se muestra en la figura 6. Los servicios son ofrecidos por el servidor y pueden ser accedidos usando una conexión a internet por el dispositivo móvil.

Como muestra la figura 6 la lógica de la aplicación será con los siguientes componentes:

- 1) *Beacon Emisor:* La selección del emisor de señales BLE no hace una diferencia con respecto a la señal iBeacon. Los emisores usualmente se diferencian en el rango de señal, tamaño del dispositivo y duración de la batería. El beacon Jaalee ha sido elegido para el desarrollo de la aplicación. Jaalee tiene su propia aplicación móvil para configurar los parámetros que posee el beacon descritos en la tabla 1. Para la conexión y detección de la aplicación del beacon, hemos utilizado la librería AltBeacon, que permite la configuración con diferentes marcas de beacons, entre ellas está Jaalee.
- 2) *Dispositivo Móvil:* Para el desarrollo de la aplicación se ha instalado la primera versión en un celular con una versión de Android 5.1.1 con 1GB de RAM y un CPU Quad Core 1.1 GHz.
- 3) *Servidor:* El diseño y la implementación de la infraestructura del servidor no es parte de este trabajo. Sin embargo, la aplicación que se desplegará en el servidor está dentro del alcance del proyecto. Por esta razón, una plataforma como servicio (PaaS) ha sido elegida. Heroku es un servicio de hosting para el servidor de la aplicación. Heroku permite a los desarrolladores enfocarse solamente en el diseño de la aplicación en vez de preocuparse por la infraestructura. Heroku se encarga de todo el diseño de la infraestructura y las consideraciones de seguridad. El servidor de la aplicación será el encargado de la parte lógica y la publicación de los servicios disponibles para que sean consumidos desde el dispositivo móvil. Los servicios han sido desarrollados en PHP con el framework Codeigniter. En el servidor también se encuentra desplegado el gestor de base de datos para almacenar los datos de la aplicación.

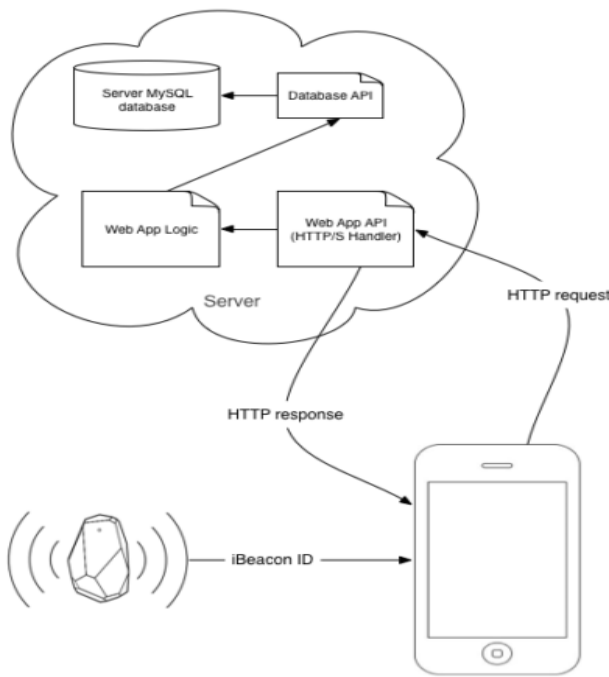


Fig. 6 Arquitectura de la aplicación reserva de citas.

C. Diseño de base de datos

El diagrama de Entidad Relación se muestra en la figura 7 con el objetivo de que sea una base de datos centralizada, para que sirva de datos a la aplicación móvil y a una aplicación web que pueda administrar esos datos con la creación de servicios web [14].

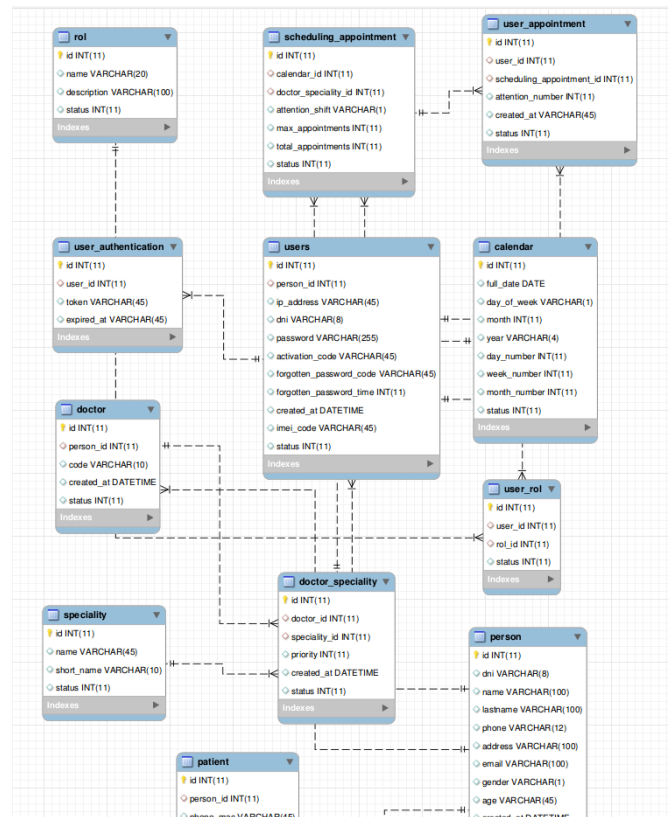


Fig. 7 Diagrama entidad relación de la aplicación.

Se ha declarado doce entidades para almacenar los datos y cumplir con todos los requerimientos funcionales descritos anteriormente. Para registrar los roles de la aplicación se usa la tabla rol, donde se registran los roles disponibles, actualmente la aplicación solo tiene el rol de paciente, pero nos permitirá registrar varios roles sin alterar el diagrama de entidad relación.

En la tabla person se guardan los datos de un usuario, cuando se registra un nuevo usuario en la aplicación, se almacenan los datos principales en la tabla person y creamos un registro en la tabla user, para asociar el nuevo usuario con su rol usamos la tabla user_rol, donde se guarda el rol del usuario, esta lógica nos permite asociar a un usuario con 1 hasta n roles en la aplicación.

Para la autenticación hace uso de las siguientes tablas: users, user_rol, user_authentication, donde se almacenará el token de autenticación después de que el servidor haya validado al usuario.

Tenemos una tabla speciality, donde se registran a todas las especialidades del hospital, una tabla doctor, donde se registran todos los doctores del hospital. Para relacionar un doctor con su especialidad tenemos la tabla doctor_speciality, donde asociamos al doctor con su especialidad, dándose el caso de que un doctor puede tener 0 hasta n especialidades.

La tabla calendar tiene el registro de todos los días del año que nos permitirá relacionarla con la tabla scheduling_appointment para registrar todas las especialidades que el usuario puede reservar en un determinado día. Las reservas del usuario se guardan en la tabla user_appointment guardando el número de cita del usuario.

D. Implementación

Todos los requerimientos de seguridad han sido completados en nuestra aplicación, para ingresar el servidor valida al usuario con su DNI y password, y asociamos el código IMEI para evitar que otro usuario pueda acceder desde otro dispositivo. Una vez que el usuario a ingresado a la aplicación se generará un token para el envío de la petición de datos al servidor, dicho token es generado después que el servidor haya verificado al usuario y haya iniciado sesión en la aplicación.

Para guardar los datos de sesión y evitar que el usuario vuelva a ingresar los datos de acceso, hacemos uso del almacenamiento interno que nos proporciona el API de android llamado "SharedPreferences". Existen otras maneras para guardar datos en el dispositivo y consultar esos datos sin necesidad de tener conexión a internet. Evaluamos los diferentes métodos y este es el que cumple con los requerimientos que necesitamos. La interfaz de inicio de sesión es mostrada en la figura 8.

Después que el usuario ha sido validado, se le muestra la interfaz para que manipule sus datos de la cita, se le muestra el historial de las citas realizadas, la planificación del día de las especialidades y finalmente se muestra el formulario para que pueda reservar su cita, cumpliendo el requerimiento de que todavía exista cupos disponibles para dicha especialidad como se muestra en la figura 9.

Toda la comunicación con el servidor se realiza mediante servicios, los cuales envían los datos en un formato JSON, se hace uso de un patron Singleton para manejar la peticiones al servidor.

Para la detección del beacon se a configurado la emisión de los anuncios cada dos segundos, dicha configuración influye en la duración de la batería del beacon. Una vez detectado el beacon, compara el UUID asociado, para determinar en qué espacio se encuentra.

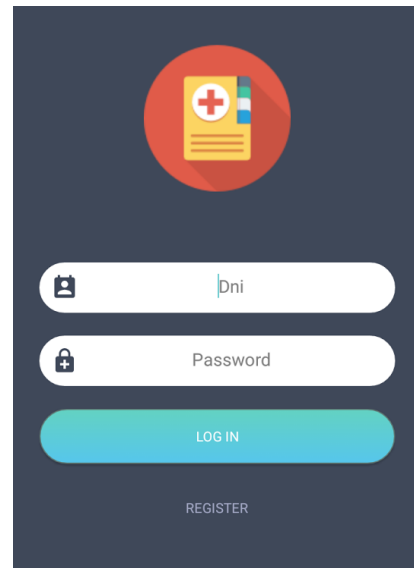


Fig. 8 Interfaz de inicio de sesión de la aplicación.

Para manejar las señales de los diferentes beacons, la aplicación solo va a responder cuando la distancia entre el beacon y el dispositivo sea menor que 5 metros. Esta distancia se ve afectada por el ambiente en el que se encuentre el usuario, por lo que no es una medida exacta, pero ayuda a solucionar el cruce de las señales que emiten los beacons.

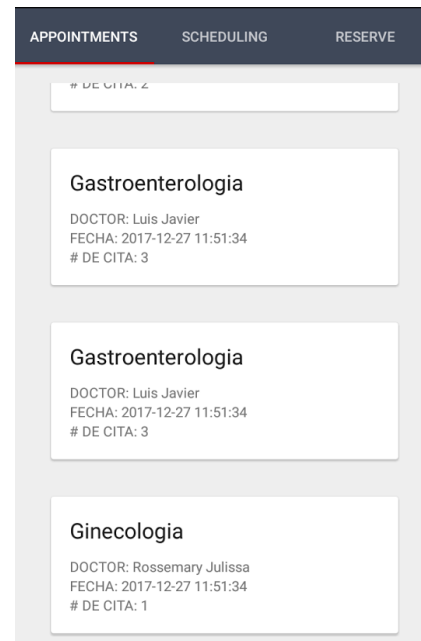


Fig. 9 Interfaz de reserva de citas de la aplicación.

II. CONCLUSIONES

Hay varios enfoques para el desarrollo de una aplicación móvil, pero se ha comprobado que, para lograr un mayor control sobre la capa del Framework y Librerías, es mejor utilizar el lenguaje nativo del dispositivo. Además, existen dos

factores claves en el desarrollo de aplicaciones móviles, el diseño de la interfaz, el uso de patrones de diseño, y el uso eficiente de las capacidades del dispositivo móvil.

En este trabajo, hemos desarrollado una aplicación móvil para la reserva de citas en un hospital, haciendo uso de la tecnología de BLE con los dispositivos externos beacons, donde se ha comprobado que para el objetivo de esta aplicación el mejor método para el desarrollo era desarrollar una aplicación nativa.

Para administrar el consumo de energía que demanda la activación del bluetooth y detección del beacon, el algoritmo en la aplicación maneja adecuadamente el ciclo de vida de todas las actividades dentro de la aplicación. Además hemos configurado el beacon, para que emita paquetes de datos en un periodo de tiempo menor a 2 segundos, para así controlar el consumo de energía, de esta manera se amplía el tiempo de vida del beacon.

Para manejar la detección de varios beacons en la aplicación y mostrar al usuario la información adecuada, se ha limitado el rango de detección de los beacons en la aplicación en dos metros. De esta manera la aplicación enviará la información del beacon más cercano a la ubicación del usuario.

La satisfacción del usuario aumentará debido a que el tiempo que demoran actualmente para obtener una cita será disminuido y ayudará a atender la demanda que irá creciendo en el transcurso de los años.

Hemos usado una tecnología emergente que cada vez es más usado en el área de Internet de las cosas y que en futuro será muy usado por las empresas para desarrollar aplicaciones basadas en la ubicación del usuario.

Creemos que desarrollar una arquitectura consistente para el desarrollo de aplicaciones móviles va a permitir adecuar y escalar fácilmente en el desarrollo de la aplicación.

REFERENCIAS

- [1] Marisa Moody, "Analysis of promising beacon technology for consumers", Strategic Communications, Elon University. M. King, B. Zhu, and S. Tang, "Optimal path planning," *Mobile Robots*, vol. 8, no. 2, pp. 520-531, March 2001.
- [2] Ramsey Faragher, Robert Harle, "An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications", University of Cambridge, UK.
- [3] Mario Collota, Giovanni Pau, "A Solution Based on Bluetooth Low Energy for Smart Home Energy Management", Faculty of Engineering and Architecture, Kore University of Enna.
- [4] Rapid Value - White paper, "How to Choose the Right Architecture For Your Mobile Application", Nov 2012.
- [5] Carles Gomez, Joaquim Oller and Josep Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology", Nov 2012.
- [6] Ding-Yu Lu, Chia-Sui Wang and Kuei-Shu-Hsu, "Beacon Applications in information services", Nov 2016.
- [7] Haeng-Kon Kim, "Architecture for Adaptive Mobile Applications", School of IT Engineering, Catholic University of Daegu, Korea.

- [8] Li Ma, Lei Gu and Jin Wang, "Research and Development of Mobile Application for Android Platform", University of Science and Technology, Nanjing.
- [9] Mohammed Binsabbar, "An iPhone Application for Providing iBeacon-based Services to Students", University of Manchester School of Computer Science, May 2014.
- [10] D. Deugo, "Using Beacons for Attendance Tracking", Carleton University, Ottawa, Ontario, Canada.
- [11] Seung-Ho Lim, "Experimental Comparison of Hybrid and Native Applications for Mobile", Department of Digital Information Engineering Hankuk University of Foreign Studies, 2015.
- [12] Shaoxuan Wang, Sujit Dey, "Adaptive Mobile Cloud Computing to Enable Rich Mobile Multimedia Applications", Department, University of California, San Diego, 2013.
- [13] Jae Kyu Lee, Jong Yeol Lee, "Android Programming Techniques for Improving Performance", Department of Electronic Engineering, Korea, 2013.
- [14] N. Mehta, S. Shinde, N. Tiku, "Centralized Database for Android and Web Application", VES Institute of Technology, Mumbai - India, Nov 2015.
- [15] S. Yoon, S. Cho, K. Kwon, J. Jeon "Application-based Power Management System for Smartphones", Sungkyunkwan University, Korea, 2014.