

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

Juan Fernández Sosa¹ [0000-0002-0482-3392], Santiago Medina¹ [0000-0001-6852-7165], Pablo Thomas¹ [0000-0001-9861-987X],
Leonardo Corbalán¹ [0000-0001-9026-8059], Marcelo Naiouf¹ [0000-0001-9127-3212]

¹ Instituto de Investigación en Informática LIDI (III-LIDI). Facultad de Informática – Universidad Nacional de La Plata, La Plata, Argentina.

Centro Asociado a la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

{jfernandez, smedina, pthomas, corbalan, mnaiof}@lidi.info.unlp.edu.ar

Resumen. Los BLE beacons son pequeños dispositivos que utilizan la tecnología Bluetooth de bajo consumo para establecer comunicación con otros dispositivos compatibles. Para este fin, existen diferentes protocolos de comunicación, entre los cuales se destacan el protocolo iBeacon y Eddystone. La mayoría de los smartphones actuales incorporan tecnología Bluetooth de bajo consumo y pueden aprovechar aplicaciones de software específicas para detectar beacons y, de esta manera, ofrecer servicios basados en la proximidad o posicionamiento indoor, entre otras posibilidades. Este trabajo tiene como objetivo estudiar y analizar diferentes alternativas de interacción entre dispositivos móviles y los BLE beacons mediante una aplicación móvil multiplataforma desarrollada con un enfoque híbrido.

Keywords: BLE beacons, análisis comparativo, aplicaciones multiplataforma, servicios basados en la proximidad, entornos inteligentes, aplicaciones híbridas, iBeacon, Eddystone.

1. Introducción

El desarrollo de las tecnologías digitales en los últimos tiempos ha permitido avanzar sobre la digitalización e interconexión de cosas, espacios y experiencias. Un ejemplo de esto es la implementación de los servicios basados en la proximidad (PBS, por sus siglas en inglés), cuyo objetivo es alertar sobre la proximidad de un objeto o persona a un punto de interés. Este tipo de servicio permite crear soluciones que transforman los espacios en entornos inteligentes e interactivos [1]. Por ejemplo en museos [2] o centros de actividades tecnológicas [3], los visitantes pueden llevar consigo una aplicación en sus dispositivos móviles, y conforme se acercan a estos puntos de interés, reciben notificaciones e información relevante. Estas aplicaciones mejoran significativamente la experiencia de los visitantes al enriquecer su interacción con el entorno.

La implementación de este tipo de soluciones puede llevarse a cabo mediante pequeños dispositivos con capacidad de transmitir una señal utilizando tecnología Bluetooth de bajo consumo (BLE). Estos dispositivos son conocidos comúnmente como BLE Beacons y presentan diferentes protocolos de comunicación, entre los cuales

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

se destacan iBeacon (desarrollado por Apple) y Eddystone (desarrollado por Google). El funcionamiento de los beacons es sencillo: emiten periódicamente una señal con una pequeña cantidad de información que puede ser captada por otros dispositivos compatibles [4]. Los smartphones poseen la capacidad de interactuar con los beacons a partir de una aplicación de software (app) desarrollada específicamente para ese fin.

En el desarrollo de aplicaciones de software para dispositivos móviles existen dos grandes estrategias: el desarrollo nativo, y el desarrollo multiplataforma [5]. El primero se centra en la utilización de los entornos y herramientas de desarrollo que provee cada uno de los sistemas operativos móviles especializados, siendo los más populares Android e iOS. Esto garantiza el acceso a todas las características del dispositivo, aunque se debe escribir un código fuente específico por cada una de estas plataformas.

El desarrollo multiplataforma se contrapone al nativo y se centra en la reutilización de código. Dentro del desarrollo multiplataforma existen diferentes enfoques, uno de ellos es el enfoque híbrido basado en tecnologías web (HTML, Javascript y CSS). Las aplicaciones desarrolladas con este enfoque son ejecutadas en un contenedor web especial que brinda un mayor acceso a las capacidades del dispositivo (acelerómetro, bluetooth, cámara, GPS, entre otras) a través de plugins externos [6]. Por lo tanto, surge el interrogante de determinar si es posible desarrollar una aplicación móvil con un enfoque híbrido que permita detectar dispositivos BLE beacons y ofrecer, por ejemplo, servicios basados en la proximidad y posicionamiento indoor.

El objetivo de este trabajo consiste en realizar un análisis de posibilidades de implementación de una aplicación móvil que dé soporte a la detección de dispositivos BLE beacons empleando un enfoque de desarrollo multiplataforma del tipo híbrido.

El resto de este trabajo se organiza de la siguiente manera: en la sección 2 se presenta la tecnología Bluetooth de bajo consumo y los BLE beacons junto a sus principales protocolos de comunicación. Luego, en la sección 3, se describen los experimentos realizados y en la sección 4 el análisis de resultados de cada uno de ellos. Finalmente se mencionan conclusiones y trabajo futuro.

2. BLE beacons

Bluetooth es una tecnología de comunicación inalámbrica que fue presentada en 2010 y que se incluye dentro del estándar IEEE 802.15 que define las *Wireless Personal Area Network* (WPAN), redes orientadas a cubrir distancias cortas. Particularmente en IEEE 802.15.1 se especifica por completo la arquitectura y el funcionamiento de Bluetooth y Bluetooth de bajo consumo (BLE). Este último define en su capa física el uso de 40 canales de radiofrecuencia que operan en la banda libre ISM de 2.4GHz dedicada a industria, ciencia y medicina [7].

Una gran variedad de tipos de dispositivos como celulares, pulseras inteligentes o notebooks son compatibles con la tecnología BLE para la transmisión y recepción de datos. Además, esta tecnología es soportada por los sistemas operativos más utilizados en estos dispositivos, como por ejemplo Windows, Linux, Android e iOS.

Los beacons son pequeños dispositivos inalámbricos que se basan en el uso de la tecnología BLE para comunicación. Su pequeño tamaño, su bajo costo y la posibilidad

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

de alimentarlos con baterías por un largo tiempo vuelven su uso popular en aplicaciones que requieren de un mínimo caudal de datos [1].

El beacon emite un paquete cada cierto intervalo de tiempo con su identificador y en algunos casos también agrega datos telemétricos recogidos por sensores integrados por el fabricante. En los dispositivos receptores se ejecuta una aplicación de software para detectar la señal, recibir el paquete con el identificador y en función de su procesamiento, realizar una acción determinada.

La posibilidad de generar nodos sensores de bajo consumo energético que se pueden comunicar con una gran cantidad de dispositivos heterogéneos llevó a BLE a convertirse en una de las tecnologías más establecidas en el mercado [8]. Estos nodos sensores se utilizan principalmente en el dominio de Internet de las Cosas (IoT) y Ciudades Inteligentes, ofreciendo numerosas posibilidades de aplicación, como el desarrollo de soluciones de marketing y publicidad, servicios basados en la proximidad, localización indoor, así como el posicionamiento y seguimiento de objetos, entre otras utilidades relevantes [7] [9] [10].

Para lograr una interacción fluida y segura entre los BLE beacons y otros dispositivos compatibles, existen diferentes protocolos de comunicación. Entre los cuales se destacan iBeacon y Eddystone [11]. Dichos protocolos se describen a continuación.

2.1 Protocolo iBeacon

iBeacon es un framework basado en BLE desarrollado por Apple y lanzado junto con el sistema operativo iOS 7 en 2013 [12].

El objetivo de iBeacon es ofrecer servicios basados en la proximidad para las aplicaciones móviles. Se utiliza la potencia de transmisión y el indicador de intensidad de señal recibida (RSSI), para determinar la distancia entre el iBeacon y el dispositivo móvil [13].

En la Figura 1 se muestra un detalle de la estructura del paquete de datos que utiliza el protocolo iBeacon.

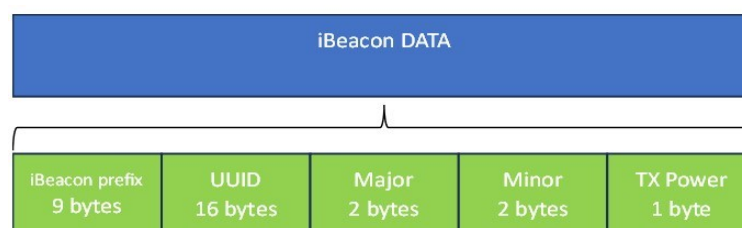


Fig 1. Paquete iBeacon

Los valores UUID, Major y Minor brindan información que permite a las aplicaciones identificar al iBeacon. Generalmente, esta información es de naturaleza jerárquica, los campos major y minor permiten subdividir la identidad establecida por el UUID de un beacon dentro de una aplicación específica. El TX Power es un parámetro que se utiliza para determinar la distancia del usuario al beacon. El prefijo

de iBeacon contiene flags, el header, el identificador del fabricante, el tipo de iBeacon y la longitud restante del paquete.

2.2 Protocolos Eddystone

Eddystone es un protocolo de BLE Beacon abierto lanzado por google en julio de 2015 compatible con Android e iOS [14].

Entre los objetivos de este protocolo se encuentran la implementación sencilla para una amplia variedad de dispositivos BLE, una arquitectura flexible que permite el desarrollo de nuevos tipos de mensajes y compatibilidad con la especificación de Bluetooth Core.

Define cuatro tipos de tramas distintas para diferentes aplicaciones: UID, URL, TLM y EID. Eddystone-TLM es la trama utilizada para transmitir información telemétrica sobre un beacon, como la temperatura, el voltaje y un conteo de los paquetes emitidos [15]. En la Figura 2 se puede ver un ejemplo de estructura de un frame TLM-Eddystone.

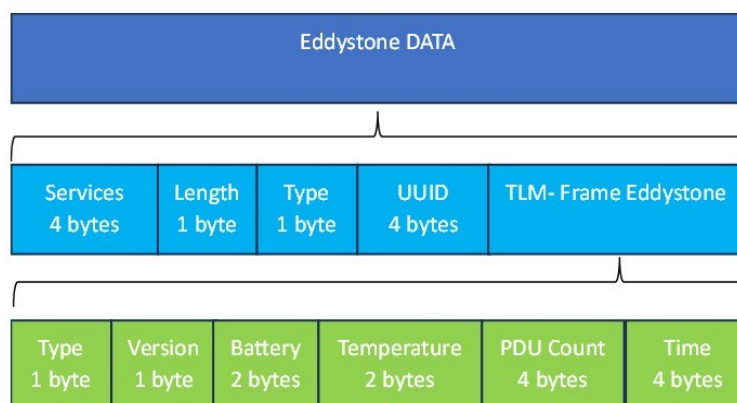


Fig. 2. TLM-Eddystone

3. Experimentación

3.1 Motivación

En el desarrollo de aplicaciones de software para dispositivos móviles, existen dos estrategias principales: la nativa y la multiplataforma [16] [17]. La estrategia nativa implica crear una aplicación específica para cada plataforma (sistema operativo) en la que se pretende distribuir el software. Por otro lado, la estrategia multiplataforma consiste en desarrollar un solo proyecto utilizando un mismo código fuente, el cual se ejecutará en diversas plataformas.

Mientras que en el enfoque nativo se emplean las herramientas de desarrollo (SDK) proporcionadas por cada plataforma, el enfoque multiplataforma ofrece mayor flexibilidad al permitir la utilización de diversas tecnologías. Específicamente, existe

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

una subcategoría dentro del desarrollo multiplataforma conocida como desarrollo de aplicaciones móviles del tipo híbridas que emplea tecnologías web estándar (HTML, JavaScript y CSS).

Las aplicaciones híbridas permiten la reutilización de código en las distintas plataformas, el acceso al hardware del dispositivo, y la distribución a través de las tiendas de aplicaciones. Sin embargo, para acceder al hardware del dispositivo en este tipo de desarrollo se deben utilizar plugins o librerías externas. A pesar de esta dependencia, el desarrollo híbrido ofrece la ventaja de requerir menos líneas de código para interactuar con el hardware del dispositivo en comparación con el enfoque nativo [18].

El objetivo de la experimentación en este trabajo consiste en identificar cuáles son las librerías o plugins que existen y permiten llevar a cabo la detección de dispositivos BLE beacons. Así mismo, se busca determinar cuáles de éstos podrían utilizarse para crear aplicaciones orientadas a dar soporte a servicios basados en la proximidad.

3.2 Tecnología seleccionada

Para el desarrollo de la aplicación se optó por utilizar el framework de código abierto Ionic [19]. Ionic es un framework que se emplea para desarrollar aplicaciones móviles híbridas mediante el uso de tecnologías web como HTML, CSS y

JavaScript. Esta herramienta proporciona un conjunto de componentes visuales que permiten lograr una apariencia similar a la de una aplicación nativa. Asimismo, ofrece la flexibilidad de integrar otros frameworks de desarrollo, entre los cuales se ha seleccionado Angular para este proyecto en particular.

Ionic se basa en Apache Cordova para acceder a las funcionalidades del hardware del dispositivo y permitir que las aplicaciones escritas en tecnologías web se ejecuten como aplicaciones móviles nativas. Cordova actúa como un puente entre el código web y las API nativas del dispositivo. Al compilar una aplicación con Cordova, se crea un contenedor nativo que embebe el código web y lo ejecuta como una aplicación móvil nativa en el dispositivo, esto permite que las aplicaciones desarrolladas con esta tecnología sean multiplataforma y se puedan ejecutar en diferentes sistemas operativos, utilizando de base el mismo código fuente.

3.3 Plugins evaluados

El proceso de búsqueda y selección de los plugins se llevó a cabo siguiendo un criterio específico. Se priorizó la elección de aquellos plugins que ofrecieran soporte para los protocolos de comunicación iBeacon y Eddystone, así como aquellos que permitieran la comunicación y detección de dispositivos utilizando la tecnología BLE. Se descartaron aquellos plugins que estaban diseñados exclusivamente para conectar con beacons de un fabricante específico, optando en su lugar por opciones más genéricas y versátiles.

Como resultado de este proceso se identificaron tres plugins que cumplían con los criterios planteados:

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

- **cordova-plugin-ibeacon**: plugin diseñado específicamente para la detección de dispositivos BLE beacons empleando el protocolo iBeacon [20].
- **cordova-plugin-eddystone**: plugin diseñado específicamente para la detección de dispositivos BLE beacons empleando el protocolo Eddystone [21]
- **cordova-plugin-ble-central**: este plugin permite la comunicación con dispositivos Bluetooth de bajo consumo, incluyendo Beacons. Proporciona funciones para escanear dispositivos BLE, conectarse a ellos y leer/escribir características [22].

Se realizó un análisis inicial de cada uno de estos plugins mediante la revisión de su documentación. Durante este proceso, se evaluaron diferentes datos, incluyendo la especificación de los protocolos de comunicación compatibles, así como su disponibilidad para las plataformas Android y/o iOS. Por otra parte, se recopiló el número de versión de cada plugin y la fecha de su última actualización. Estos detalles se sintetizan y presentan en la tabla 1.

Tabla 1. Listado de plugins seleccionados para la experimentación.

Plugin	Versión	Última actualización	Protocolos compatibles		Plataformas compatibles	
			iBeacon	EddyStone	Android	iOS
<i>cordova-plugin-ibeacon</i>	3.7.0	Julio 2018	Si	No	Si	Si
<i>cordova-plugin-eddystone</i>	1.3.0	Julio 2016	No	Si	Si	Si
<i>cordova-plugin-ble-central</i>	1.7.0	Mayo 2023	No indica	No indica	Si	Si

3.4 Diseño del experimento

La experimentación se centra en probar cada uno de los plugins seleccionados y verificar su capacidad para detectar un dispositivo BLE beacon, así como la información proporcionada por esta interacción. El objetivo es determinar la viabilidad para crear soluciones basadas en la proximidad utilizando dichos plugins.

Para alcanzar este objetivo, en primer lugar, fue necesario configurar y programar los dispositivos BLE beacons que se emplearían en los experimentos. Posteriormente, se procedió al desarrollo de la aplicación móvil que permitiera utilizar los plugins seleccionados. Ambos procesos son descritos en las subsecciones siguientes y el código fuente utilizado para construir tanto la aplicación móvil como los beacons se encuentra disponible en [23].

3.4.1 Configuración de BLE beacon

Para el despliegue de los BLE beacons se utilizó una placa de desarrollo basada en el microcontrolador ESP32 fabricado por Espressif.

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

El ESP32 combina la potencia de dos procesadores Xtensa de 32 bits con conectividad Wi-Fi y Bluetooth 4.0 integrada, lo que le permite comunicarse fácilmente con otros dispositivos y acceder a redes inalámbricas. Además, este microcontrolador se destaca por su amplia gama de interfaces y periféricos, que incluyen GPIOs, UART, I2C, SPI, ADC, DAC y más [24].

Para la experimentación se programaron dos beacons con el IDE Arduino. Cada uno de ellos utilizando un protocolo de comunicación específico: iBeacon y Eddystone-TLM. Como base para el desarrollo se utilizaron los ejemplos proporcionados por la librería BLE ESP32 para Arduino [25].

3.4.2 Desarrollo de la aplicación móvil

La aplicación móvil fue desarrollada utilizando la versión 6 del framework Ionic, y fue compilada para las plataformas Android e iOS. Se emplearon tres dispositivos móviles diferentes, cuyas características se mencionan en la tabla 2. En cada uno de estos dispositivos se evaluaron los tres plugins para la interacción con los beacons, resultando en un total de nueve pruebas realizadas.

Las pruebas sobre cada uno de los plugins corresponden con un número de experimento, que permitirá abordar, en la siguiente sección, los resultados: el experimento #1 y #2 utilizan los plugins *cordova-plugin-ibeacon* y *cordova-plugin-eddystone* respectivamente. El experimento #3 emplea el plugin para comunicación con dispositivos BLE genéricos *cordova-plugin-ble-central*.

Tabla 2. Listado de dispositivos móviles donde se realizaron los experimentos

Dispositivo	Sistema Operativo	Versión
iPhone 13 Mini	iOS	16.5.1
Moto G 9 Plus	Android	11
Huawei P9 Lite	Android	7

Los plugins fueron instalados siguiendo los pasos y recomendaciones detallados en la documentación oficial de cada uno de ellos. Incluyendo la gestión de los permisos necesarios para que funcionen correctamente.

La interfaz de la aplicación es sencilla y consta de tres botones, cada uno de los cuales habilita la detección de los beacons utilizando uno de los plugins seleccionados.

4. Resultados

A continuación, se presentan los resultados obtenidos luego de realizar las experimentaciones con cada uno de los plugins y los beacons configurados con los protocolos iBeacon y Eddystone.

4.1 Detección de BLE beacons en el experimento #1

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

En los resultados del experimento #1, se observó que la detección e identificación del beacon configurado con el protocolo iBeacon requería la especificación del valor del UUID, el valor mayor (Major) y el valor menor (Minor) dentro de la aplicación. Estos datos deben coincidir con la configuración establecida en el propio beacon para lograr una detección e identificación exitosa. Además, se encontró que es factible detectar varios beacons con un mismo UUID si solo se establece este parámetro.

La detección se logró realizar de manera exitosa tanto para los dispositivos con sistema operativo Android como para el dispositivo con iOS. Es decir que, para ambas plataformas, el plugin logró reconocer el beacon.

De la interacción entre la aplicación móvil y el beacon se obtuvieron un conjunto de datos en formato JSON, los cuales se presentan en la tabla 3. Si bien la detección se logró para ambas plataformas, se puede apreciar una diferencia en las respuestas para cada una de ellas. En el caso de Android se logró obtener los datos necesarios para determinar la distancia y la potencia de la señal (RSSI) del beacon. No obstante, en iOS sólo se logró la detección del beacon, sin recibir ningún dato que permita determinar la proximidad del dispositivo móvil al beacon.

Tabla 3. Resultado del experimento #1

Plugin	Detección BLE Beacon Protocolo iBeacon		Resultado de la interacción	
	Android	iOS	Android	iOS
<i>cordova-plugin-ibeacon</i>	Si	Si	{ "uuid": "4d6fc88bbe75-5555-4444-3333a36ec78e", "major": "0", "minor": "53", "proximity": "ProximityFar", "rssi": -66, "tx": 0, "accuracy": 92233720368547760 }	{ "minor": 15, "rssi": 0, "major": 0, "proximity": "ProximityUnknown", "accuracy": -1, "uuid": "4D6FC88B-BE75-5555-4444-3333A36EC78E" }

4.2 Detección de BLE beacons en el experimento #2

A diferencia del experimento #1, para detectar los beacons mediante el protocolo Eddystone, no hace falta especificar el identificador del beacon, sino que se hace un escaneo genérico intentando identificar a los beacons presentes.

Sin embargo, durante las pruebas realizadas, se encontró que no fue posible detectar la presencia del beacon configurado con el protocolo Eddystone. El plugin utilizado reportó un error de compatibilidad con otras librerías, lo que impidió la realización de la detección. Es relevante mencionar que este plugin recibió su última actualización en

el año 2016, lo que sugiere una falta de mantenimiento y, por consiguiente, lo descarta como una opción viable para este propósito.

Tabla 4. Resultados del experimento #2

Plugin	Detección BLE Beacon Protocolo Eddystone		Resultado de la interacción	
	Android	iOS	Android	iOS
<i>cordova-plugin-eddystone</i>	No	No	Sin resultados	Sin resultados

4.3 Detección de BLE beacons en el experimento #3

En el tercer experimento, se utilizó un plugin que no fue específicamente diseñado para la detección de beacons, pero que ofrece la funcionalidad de detectar dispositivos que utilizan Bluetooth de bajo consumo. Este plugin fue probado para detectar tanto el beacon configurado con el protocolo iBeacon, como aquel configurado con el protocolo Eddystone. Los resultados de la detección, tanto para Android como para iOS, se muestran en la tabla 5, y en las tablas 6a y 6b se presentan los datos obtenidos durante la interacción.

El plugin permitió realizar un escaneo general de todos los dispositivos cercanos al dispositivo móvil donde se ejecutó la aplicación. Por esta razón, muchos de los resultados detectados pertenecían a dispositivos no buscados. No obstante, al aplicar filtros adecuados, fue posible reconocer y aislar los beacons de interés.

En el caso de los dispositivos con Android, el plugin logró detectar a los beacons configurados con ambos protocolos de comunicación. La identificación de los beacons se llevó a cabo mediante el ID del beacon que corresponde a la dirección MAC del dispositivo. En el caso de los beacons Eddystone, la respuesta incluye además un atributo con el nombre del dispositivo, que es configurable desde el lado del beacon, lo que facilitó su identificación.

Con el dispositivo iOS los resultados al utilizar el plugin presentaron algunas diferencias. El plugin permitió detectar correctamente al beacon configurado con el protocolo Eddystone. Al igual que en las pruebas realizadas para Android, la identificación y filtrado de los resultados se llevaron a cabo utilizando el nombre configurado para el beacon.

En cuanto a los iBeacons, durante los experimentos en iOS, inicialmente se enfrentó a un desafío para identificar cuál de los dispositivos detectados correspondía al beacon configurado. Esto se debió a que el valor del ID que se retornaba no era el UUID real del dispositivo, sino uno generado adrede por el sistema operativo. Dicho valor puede variar en diferentes dispositivos móviles que estén intentando detectar al beacon, por lo que el proceso de detección fue dificultoso. Para resolver este problema, se realizó una modificación en el beacon configurado con el protocolo iBeacon para que devuelva un adicional en la interacción, el nombre del beacon. Con esta modificación del lado del beacon fue posible identificarlo de manera certera en la aplicación.

Tanto para dispositivos Android como para iOS, una vez que se aislaron los resultados de los beacons, se pudo observar que ambos protocolos devuelven la

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

potencia de la señal detectada (RSSI). Este dato es valioso, ya que permite estimar la distancia entre el dispositivo móvil y el beacon, y resulta crucial para diversas aplicaciones que se basan en la proximidad del usuario a los beacons detectados.

Tabla 5a. Resultados del experimento #3

Plugin	Detección BLE Beacon Protocolo iBeacon		Detección BLE Beacon Protocolo Eddystone	
	Android	iOS	Android	iOS
<i>cordova-plugin-ble-central</i>	Si	Si*	Si	Si

Tabla 6a. Resultados del experimento #3 - iBeacon

Plugin	Resultados interacción iBeacon	
	Android	iOS
<i>cordova-plugin-ble-central</i>	<pre>{ "id": "30:C6:F7:21:23:F2", "advertising": {}, "rssi": -27 }</pre>	<pre>{ advertising = { kCBAAdvDataIsConnectable = 0; kCBAAdvDataLocalName = "ESP32-iBeacon"; }; id = "3FFB9E7F-31AD-F11A-85B1-E4039055AF28"; name = "ESP32-iBeacon"; rssi = "-69"; state = disconnected; }</pre>

Tabla 6b. Resultados del experimento #3 - Eddystone

Plugin	Resultados interacción Eddystone	
	Android	iOS
<i>cordova-plugin-ble-central</i>	<pre>{ "name": "ESP32", "id": "30:C6:F7:21:3C:46", "advertising": {}, "rssi": -74 }</pre>	<pre>{ advertising = { kCBAAdvDataIsConnectable = 1; kCBAAdvDataLocalName = ESP32; }; id = "3FFB9E7F-31AD-F11A-85B1-E4039055AF28"; name = ESP32; rssi = "-69"; state = disconnected; }</pre>

4.4 Análisis general de resultados

El análisis de los resultados obtenidos en cada uno de los experimentos permite formular las siguientes recomendaciones: en el caso de desarrollar una aplicación para la detección de dispositivos BLE beacons utilizando el protocolo Eddystone, se aconseja no emplear el plugin "*cordova-plugin-eddystone*". Sin embargo, como alternativa viable, se sugiere utilizar el plugin de detección general de beacons "*cordova-plugin-ble-central*", y aplicar un proceso de filtrado según el nombre del beacon. Esta recomendación se extiende tanto para plataformas Android como para iOS.

En el contexto de los iBeacons, para aplicaciones orientadas a la plataforma Android, se recomienda el uso del plugin específico para este tipo de beacons, "*cordova-plugin-ibeacon*". Este plugin permite la detección de iBeacons y, además, incluye datos en la respuesta que facilitan el cálculo de la distancia entre el dispositivo y el beacon. Por ende, es posible considerar este plugin para la creación de soluciones que aprovechen la proximidad para brindar servicios basados en esta característica. No obstante, para dispositivos iOS, este plugin solo posibilita la detección del beacon, sin proporcionar una forma precisa de determinar la distancia entre el dispositivo móvil y el mismo.

Para superar esta limitante en iOS, se recomienda emplear el plugin "*cordova-plugin-ble-central*", el cual, según los resultados de las pruebas realizadas, demostró la capacidad de identificar tanto beacons con protocolo iBeacon como Eddystone. En el caso específico de los iBeacons en iOS, es fundamental configurar el beacon para que muestre el dato del nombre, de modo que pueda ser identificado de manera certera por la aplicación.

A pesar de que el plugin utilizado no fue específicamente diseñado para la detección de beacons, su capacidad para detectar dispositivos Bluetooth de bajo consumo y reconocer beacons configurados con diferentes protocolos resultó ser una ventaja significativa en el contexto del estudio. No obstante, es fundamental tener en cuenta las limitaciones inherentes a un escaneo general, y la necesidad de aplicar filtros adecuados para obtener resultados precisos y relevantes en la detección e identificación de los beacons de interés.

5. Conclusiones y Trabajo Futuro

En este trabajo se presentó un análisis de diferentes opciones que existen para desarrollar una aplicación de software para dispositivos móviles que sea capaz de detectar dispositivos BLE beacons. Particularmente el análisis se focalizó en el desarrollo de una aplicación multiplataforma del tipo híbrido.

Los BLE beacons poseen diferentes protocolos que permiten a las aplicaciones poder detectarlos e intercambiar información con ellos. En particular, en este trabajo se estudiaron los protocolos iBeacon y Eddystone.

La experimentación consistió en el análisis y prueba de tres plugins diferentes: a) *cordova-plugin-ibeacon*, b) *cordova-plugin-eddystone* y c) *cordova-plugin-ble-central*. Estos plugins fueron probados en tres dispositivos móviles diferentes, dos de ellos con

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

sistema operativo Android y uno con sistema operativo iOS. La aplicación desarrollada, fue compilada y ejecutada en ambos sistemas operativos.

Es necesario considerar que los resultados presentados en este trabajo, están ligados a las versiones tanto del framework Ionic como de los plugins utilizados al momento de llevar a cabo la experimentación y, por lo tanto, podrían variar en el futuro, conforme éstos reciban actualizaciones.

A partir de los resultados de las experimentaciones se puede establecer una serie de recomendaciones, en lo relativo al desarrollo de una aplicación móvil para la detección de BLE beacons empleando el framework Ionic. Si se tiene que interactuar con un beacon con protocolo Eddystone es conveniente utilizar el plugin *cordova-plugin-ble-central*, ya que se logró una detección del beacon tanto en los dispositivos Android como en el dispositivo iOS. Además, este plugin permite obtener los datos de la intensidad de la señal que permite calcular distancias para ofrecer servicios basados en la proximidad. Este plugin se utiliza como alternativa al plugin específico para la detección de beacons con el protocolo Eddystone *cordova-plugin-eddystone*, el cual tuvo que ser descartado al no poder ejecutar el plugin correctamente.

Por otra parte, en el caso de los iBeacons hay que diferenciar el comportamiento en dispositivos Android de iOS. En Android, para la detección de iBeacons se recomienda utilizar el plugin *cordova-plugin-ibeacon*, ya que con éste se logró detectar los beacons y obtener información relevante para determinar la distancia del dispositivo móvil al beacon. En iOS si bien la detección de los iBeacons se logró con éxito, no así la obtención de los datos necesarios para calcular la proximidad al beacon. Para superar esta limitante en iOS, se recomienda emplear el plugin *cordova-plugin-ble-central* en caso de poder manipular la configuración del beacon agregando un nombre para poder identificar a este dispositivo de los otros que este plugin puede detectar.

Como trabajo futuro se plantea continuar con el análisis de librerías o plugins para otros Frameworks de desarrollo de aplicaciones móviles multiplataforma. También se pretende desarrollar diferentes tipos de beacons basados en otros microcontroladores que están disponibles en el mercado.

Referencias

- [1] Spachos, P., & Plataniotis, K. N. (2020). BLE beacons for indoor positioning at an interactive IoT-based smart museum. *IEEE Systems Journal*, 14(3), 3483-3493.
- [2] Steiniger, S., Neun, M., & Edwardes, A. (2006). Foundations of location based services. *Lecture Notes on LBS*, 1(272), 2.
- [3] Fernández Sosa, J. F., Medina, S., Segovia, M., Thomas, P. J., & De Giusti, A. E. (2023). Propuesta e implementación de un sistema basado en servicios de proximidad con BLE Beacons. In XXVIII Congreso Argentino de Ciencias de la Computación (CACiC)(La Rioja, 3 al 6 de octubre de 2022).
- [4] Lindh, J. (2015). Bluetooth low energy beacons. Texas Instruments, 2.
- [5] Thomas, P. J., Delía, L. N., Corbalán, L. C., Cáseres, G., Fernández Sosa, J., Tesone, F., ... & Pesado, P. M. (2018). Tendencias en el desarrollo de Aplicaciones para Dispositivos

Análisis comparativo de alternativas para la detección y comunicación con dispositivos BLE Beacons en Aplicaciones Móviles Multiplataforma

- Móviles. In XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste)..
- [6] Kaczmarczyk, A., Zając, P., & Zabierowski, W. (2022). Performance Comparison of Native and Hybrid Android Mobile Applications Based on Sensor Data-Driven Applications Based on Bluetooth Low Energy (BLE) and Wi-Fi Communication Architecture. *Energies*, 15(13), 4574.
- [7] Darroudi, S. M., & Gomez, C. (2017). Bluetooth low energy mesh networks: A survey. *Sensors*, 17(7), 1467.
- [8] Aguilar, S., Vidal, R., & Gomez, C. (2017). Opportunistic sensor data collection with bluetooth low energy. *Sensors*, 17(1), 159.
- [9] Chang, K. H. (2014). Bluetooth: a viable solution for IoT?[Industry Perspectives]. *IEEE Wireless Communications*, 21(6), 6-7.
- [10] Akinsiku, A., & Jadav, D. (2016, April). BeaSmart: A beacon enabled smarter workplace. In NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium (pp. 1269-1272). IEEE.
- [11] Jeon, K. E., She, J., Soonsawad, P., & Ng, P. C. (2018). Ble beacons for internet of things applications: Survey, challenges, and opportunities. *IEEE Internet of Things Journal*, 5(2), 811-828. <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>
- [12] Kwok, C. Y. T., Wong, M. S., Griffiths, S., Wong, F. Y. Y., Kam, R., Chin, D. C., ... & Mok, E. (2020). Performance evaluation of iBeacon deployment for location-based services in physical learning spaces. *Applied Sciences*, 10(20), 7126. [14] Hernández-Rojas, D. L., Fernández-Caramés, T. M., Fraga-Lamas, P., & Escudero, C. J. (2017). Design and practical evaluation of a family of lightweight protocols for heterogeneous sensing through BLE beacons in IoT telemetry applications. *Sensors*, 18(1), 57. <https://github.com/google/eddystone/blob/master/protocol-specification.md>
- [13] Delia, L., Galdamez, N., Thomas, P., Corbalan, L., & Pesado, P. (2015, Mayo).
- [14] Multi-platform mobile application development analysis. In 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS) (pp. 181-186). IEEE.
- [15] Delia, L., Thomas, P., Corbalan, L., Sosa, J. F., Cuitiño, A., Cáseres, G., & Pesado, P. (2018, Julio). Development approaches for mobile applications: comparative analysis of features. In *Science and Information Conference* (pp. 470-484). Springer, Cham.
- [16] Kaczmarczyk, A., Zając, P., & Zabierowski, W. (2022). Performance Comparison of Native and Hybrid Android Mobile Applications Based on Sensor Data-Driven Applications Based on Bluetooth Low Energy (BLE) and Wi-Fi Communication Architecture. *Energies*, 15(13), 4574.
- [17] <https://ionicframework.com/>, accedido en Julio 2023.
- [18] <https://github.com/petermetz/cordova-plugin-ibeacon>, accedido en Julio 2023.
- [19] <https://github.com/evotthings/cordova-eddystone>, accedido en Julio 2023.
- [20] <https://github.com/don/cordova-plugin-ble-central>, accedido en Julio 2023.
- [21] <https://gitlab.com/iii-lidi/papers/beacons-app-cacic2023>

[22] https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, accedido en Julio 2023.

[23] https://github.com/nkolban/ESP32_BLE_Arduino, accedido en Julio 2023.