Patrón de Diseño Beacon Action Manager para comunicar Aplicaciones Móviles (IoT)

Boccardo Yanina, Germán Montejano, Daniel Riesco

Departamento de Informática – Universidad Nacional de San Luis
San Luis, Argentina.

yani.boccardo@gmail.com, {gmonte,driesco}@unsl.edu.ar

Resumen. *Internet de las cosas permite que todos, personas y cosas, estén continuamente conectados y puedan recibir y procesar información en tiempo real. Esta nueva ola de tecnología, produjo una acelerada aceptación de los teléfonos inteligentes, donde las aplicaciones móviles se convierten en una herramienta muy útil para comprender el entorno de los usuarios. Con la llegada de IoT aparecen nuevas tecnologías y dispositivos como el caso de los beacons, dispositivos que permiten obtener información del contexto de los usuarios a través de las aplicaciones móviles. Ésto trae consigo nuevos desafíos de diseño para los desarrolladores que necesitan integrar ésta nueva tecnología en sus aplicaciones. En éste artículo se presenta un patrón de diseño para aplicaciones móviles que interactúan con dispositivos beacons.*

Abstract. *Internet of Things (IoT) allows all, people and things, are continuously connected and can receive and process information in real time. This new wave of technology, produced a rapid acceptance of smartphones, where mobile applications become a very useful tool to understand the user's environment. With the advent of IoT appears new technologies and devices such as beacon, devices that allow obtaining user's context information through mobile applications. This brings new design challenges for developers who need to integrate this new technology into their applications. This article describes a design pattern for mobile applications that interact with beacon devices.*

1. Introducción

Quince años atrás, cuando se hablaba del futuro hacia el que nos conducía el avance de la tecnología, se mencionaba la domótica: casas completamente robotizadas donde las tareas se realizan de forma automática o por control remoto. Sin embargo, la realidad ha ido aún más allá, e internet ha logrado algo más grande, un mundo de objetos interconectados y conectados a la red capaces de cumplir cada vez más funciones y más complejas, aplicable prácticamente a cualquier ámbito. Es el llamado Internet de las cosas (IoT por sus siglas en inglés Internet of Things).

IoT permite que todos, personas y cosas, estén permanentemente conectados y puedan recibir y procesar información en tiempo real. IoT ya es una realidad presente en

tendencias como wearables, electrodomésticos que ofrecen todo tipo de información al usuario. Siguiendo la filosofıa de los objetos interconectados, todo ello se puede controlar desde el celular. Esta gran ola de tecnologıa, produjo una acelerada aceptación de los teléfonos inteligentes y tablets, donde las aplicaciones móviles se convierten en una herramienta cada vez más útil para comparaciones contextuales en el entorno de los usuario.

La rápida evolución de la electrónica durante la última década hizo que la idea de IoT tome una gran relevancia práctica. El tamaño, el costo y el consumo de energía del hardware se han reducido drásticamente. La aparición de estos pequeños dispositivos junto con la expansión de las redes de comunicación, permiten incorporar inteligencia y conexión a los objetos del mundo real, transformando lo que era una red global de personas en una red global de todas las cosas.

En este contexto es donde surgen los dispositivos beacons, dispositivos que transmiten información para poder conocer el contexto de los usuarios a través de aplicaciones móviles. Los beacon son dispositivos que transmiten pequeñas señales (piezas de información) anunciando su presencia a los dispositivos que se encuentren dentro de su radio de alcance.

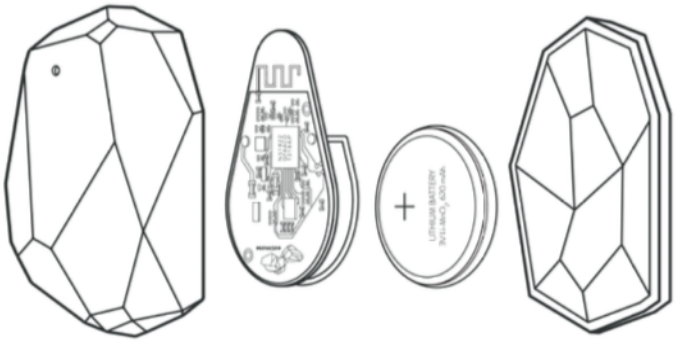

Figura 1. Un beacon en su interior.

Los beacon distribuyen pequeños paquetes de datos que se transmiten a intervalos regulares a través de ondas de radio utilizando la tecnología Bluetooth Low Energy (BLE) [Bluetooth 2016]. Es un método de comunicación de un solo sentido; simplemente anuncian sus paquetes de datos. Estos paquetes pueden entonces ser recogidos por dispositivos inteligentes compatibles dentro del radio de alcance y luego ser utilizados por distintas aplicaciones para disparar eventos tales como mensajes, notificaciones push, u otras.

Uno de los principales usos de los beacon es establecer una región alrededor de un objeto, lo que permite a un dispositivo móvil determinar cuando ha entrado o salido de esa región, junto con una estimación de la proximidad a un beacon. Esto brinda a las aplicaciones nuevas posibilidades de reconocimiento de ubicación. Cuando un teléfono inteligente entra en el rango de alcance de la señal de un beacon, la aplicación instalada escucha la señal transmitida y puede responder en consecuencia.

La aparición de nuevos dispositivos y tecnologías traen consigo nuevos desafíos y problemas para aquellos desarrolladores que desean incorporar estas nuevas tecnologías y dispositivos en sus aplicaciones. Es por eso que en este artículo se definirá

un patrón de diseño que permite prevenir aquellos problemas típicos y recurrentes que puedan surgir a la hora de desarrollar una aplicación móvil que interactúa con dispositivos beacon para obtener información del contexto de los usuarios.

El artículo se encuentra estructurado en secciones. Luego de la introducción en la sección 1, la sección 2 describe el patrón propuesto, detallando su estructura, componentes, colaboraciones y aplicación, entre otras características. En la sección 3 se describen diferentes casos de estudio de aplicaciones móviles que se realizaron aplicando el patrón. Finalmente se presentan las conclusiones y trabajos futuros.

2. Trabajos Relacionados

En la actualidad no se ha encontrado un patrón de diseño para comunicar aplicaciones móviles con dispositivos beacons. Sin embargo se desarrollaron diferentes trabajos de investigación, aplicaciones y demos que se centran en las propiedades de estos dispositivos y en cómo funcionan. A continuación se nombran algunos de ellos.

En el año 2015 en CASCON [CASCON 2015], los autores del proyecto "Context-aware mobile apps using iBeacons: towards smarter interactions" [Sykes E., Pentland S., Nardi S. 2015] describen cuatro aplicaciones móviles para dispositivos iOS que utilizan el protocolo iBeacon e interactúan con los dispositivos beacons para proporcionar relevancia contextual y experiencias personalizadas al usuario. Presentan los antecedentes en el área, la arquitectura que diseñaron y desarrollaron para las aplicaciones, las propias aplicaciones, e informes sobre los resultados de los casos de prueba en escenarios reales.

El trabajo "Experiences with using iBeacons for Indoor Positioning" [Deepesh P. C., Rath R., Tiwary A., Rao V., Kanakalata N. 2016] publicado en ISEC '16 [ISEC 2016] habla de las experiencias de los investigadores con iBeacon, sobre su eficacia en localización en interiores (indoor location) .

El proyecto "An iBeacon primer for indoor localization: demo abstract" [Martin P., Ho B., Grupen N., Muñoz S., Srivastava M. 2014] publicado en BuildSys '14 [ACM 2014] presenta un conjunto de herramientas de localización desarrolladas utilizando iBeacon, proporcionando una mirada en profundidad a la viabilidad de BLE como una tecnología de posicionamiento en interiores. Su sistema muestra un error promedio de estimación de la posición de 53 centímetros.

En el articulo "uBeacon: Configuration based Beacon tracking" [Jain P., Khanwalkar S., Malhotra R., Dheenrajappa A., Gupta G., Kobsa A. 2016], publicado en IEEE '16 [IEEE 2016], los autores describen uBeacon, un servicio cloud con una interfaz web que permite a los usuarios definir las configuraciones que incluyen restricciones entre los receptores de señal de beacons y los beacons vinculados a objetos o personas. uBeacon almacena estas configuraciones en una base de datos, monitorea continuamente los datos de los beacons, y notifica a los usuarios a través de notificaciones push cada vez que se ha violado una regla de una configuración.

En el artículo "S-Beacon: Next generation BLE beacon solution for enhanced personalization" [Byun D., Cho J., Moon S., Hong D. 2016], publicado en IEEE '16 [IEEE 2016], los autores introducen S-Beacon, la próxima generación de beacon BLE, una tecnología BLE basada en smartphones y wearables para indicar la presencia del

usuario junto con las preferencias personales y los datos demográficos que sirve como un factor clave para la mejora de servicios personalizados. En este trabajo se discute la proximidad mejorada basada BLE, el consumo optimizado de los recursos, y capacidad de gestión de dispositivos beacons.

3. Especificación del Patrón

El término patrón fue utilizado por primera vez por el arquitecto Christopher Alexander. Según Alexander "Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y, a continuación, se describe la esencia de la solución a ese problema, de tal manera que se puede utilizar esta solución un millón de veces, sin tener que hacer dos veces lo mismo" [Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I, Angel S. 1997].

A pesar de que Alexander estaba hablando de patrones en edificios y ciudades, lo que decía era cierto aplicado a patrones de diseño orientados a objetos. En el software, las soluciones se expresan en términos de objetos e interfaces en lugar de paredes y puertas, pero la esencia de los dos tipos de patrones es una solución a un problema en un contexto [Gamma E., Helm R., Johnson R., Vlissides J. 1994].

Los patrones contienen el conocimiento de experiencias anteriores y pueden utilizarse para crear nuevas soluciones en contextos similares. A continuación se describen las características del patrón presentado en este artículo.

3.1. Nombre

Beacon Action Manager.

Si bien este proyecto está realizado en un contexto de habla hispana, toda la información investigada sobre dispositivos beacons, lenguajes de programación móvil, Bluetooth Low Energy, sistemas operativos móviles, se encuentra todo en el idioma inglés, al igual que el mayor ámbito de aplicación de la tecnología beacon, es por eso que se decidió utilizar un nombre en inglés para el patrón.

3.2. Intención

Determinar el comportamiento de las clases básicas que conforman aplicaciones que interactúan con dispositivos beacon para obtener información del contexto de los usuarios, ejecutando diferentes acciones cuando se ingresa o sale de la región de un beacon o cuando cambia su proximidad al mismo.

3.3. Motivación

A la hora de plantear la arquitectura de una aplicación que se relaciona con beacons para lograr un mejor conocimiento del contexto del usuario, es necesario conocer cómo funcionan estos dispositivos y qué es necesario hacer del lado de la aplicación para que el funcionamiento de la misma y la interacción con los dispositivos sea la correcta.

Aquellos desarrolladores que decidan incorporarla en sus aplicaciones van a enfrentarse a problemas comunes que surgen al utilizar una tecnología nueva. Lo que se pretende aquí es brindar una herramienta útil, confiable y probada para aquellos desarrolladores que decidan incorporar esta nueva tecnología en sus aplicaciones,

determinando los roles específicos de cada una de las clases que intervienen, evitando así problemas comunes que puedan surgir al utilizar esta nueva tecnología.

3.4. Arquitectura

Aquí se presenta el modelo estático del patrón de diseño describiendo las clases que intervienen.

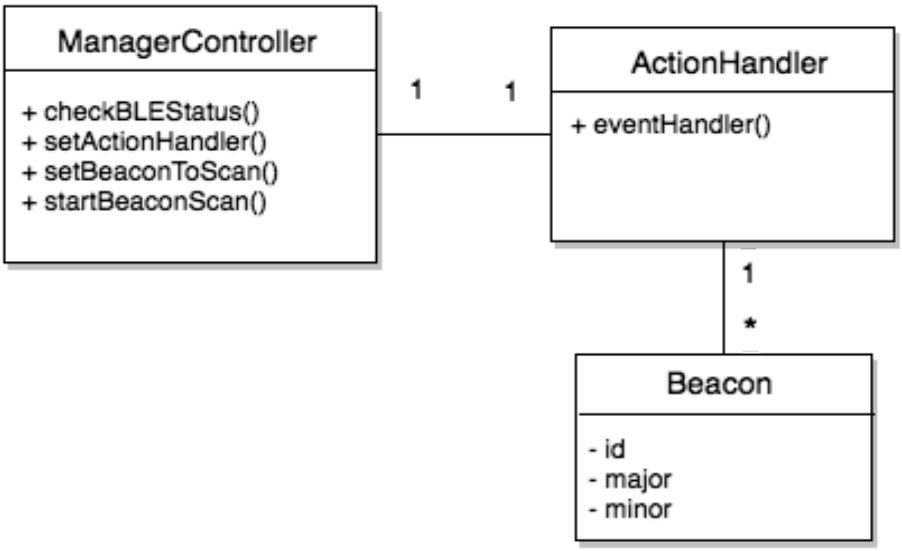

Figura 2. Diagrama de Clases del patrón Beacon Action Manager

3.5. Participantes

A continuación se listan y describen de las entidades (y sus roles) que participan en el patrón:

3.5.1. ManagerController

El rol principal de esta clase es iniciar la búsqueda de beacons próximos a la aplicación. Para poder lograrlo es necesario que realice ciertas tareas. A continuación se describen las tareas que la clase debe realizar:

- Verificar que el bluetooth del dispositivo móvil se encuentre encendido.
- Determinar el ID de aquellos beacons que a la aplicación le interese saber cuando el dispositivo móvil se encuentre dentro de su región.
- Determinar el objeto, de tipo ActionHandler, que va manejar los eventos que ocurran al interactuar con los dispositivos beacons, ya sea porque la aplicación entra o sale de la región de un beacon, porque la proximidad al mismo cambia, etc.
- Iniciar el escaneo de los beacons que se identifiquen con los ID especificados anteriormente.

3.5.2. ActionHandler

El rol de esta clase es manejar todos los eventos que se disparen cuando la aplicación se encuentra interactuando con los beacons. La clase ManagerController es la que determina cuál es el objeto que actúa como ActionHandler.

Los métodos de esta clase se determinan según las APIs o librerías que se utilicen. Son métodos que se disparan automáticamente cuando ocurren eventos al interactuar con beacons, ya sea cuando el dispositivo entra o sale de la región del beacon o su proximidad al mismo cambia.

3.5.3 Beacon

El rol de esta clase es representar los beacons detectados por los métodos de la clase ActionHandler. Representa aquellos beacon con los que está interactuando la aplicación. En la clase beacon se definen los métodos y atributos propios para este tipo de objetos que van a depender del tipo de aplicación en la que se lo utilice, por ejemplo: id, nombre, proximidad, etc.

3.6. Colaboraciones

Aquí se explican las interrelaciones que se dan entre los participantes:

La clase *ManagerController* es la que inicia el escaneo de los beacons y determina cual es el objeto, de tipo ActionHandler, que va a manejar los eventos que se disparen cuando la aplicación comience a interactuar con beacons. *ActionHandler* es la clase que maneja todos los eventos que se disparan al estar la aplicación en contacto con los beacons, ya sea porque la aplicación entra o sale de la región de un beacon, porque cambia la proximidad a un beacon, etc. *Beacon* es la clase que representa a todos los beacons detectados en los métodos del ActionHandler, o sea todos los beacon con los que está interactuando la aplicación.

En la Figura 3 se describe un diagrama de secuencia mostrando la colaboración entre los componentes.

3.7. Aplicabilidad

Utilizar el patrón Beacon Action Manager en las siguientes situaciones:

- La aplicación necesite ejecutar una acción cuando el dispositivo móvil se encuentre dentro de la región de beacons específicos. Se necesita identificar cuáles son los beacons con los que la aplicación va a interactuar.
- Se necesite ejecutar una misma acción o acciones diferentes cada vez que la aplicación interactúa con los beacons especificados.
- Se deban ejecutar acciones específicas cuando el dispositivo móvil ingresa en la región de un beacon.
- Se deban ejecutar acciones específicas cuando el dispositivo móvil sale de la región de un beacon.

- Estando el dispositivo móvil dentro de la región del beacon, se deban ejecutar acciones específicas según la proximidad a la que se encuentre el dispositivo móvil del beacon.
- Determinar qué acciones ejecutar al cambiar la proximidad del dispositivo móvil al beacon, encontrándose dentro de la región del beacon.

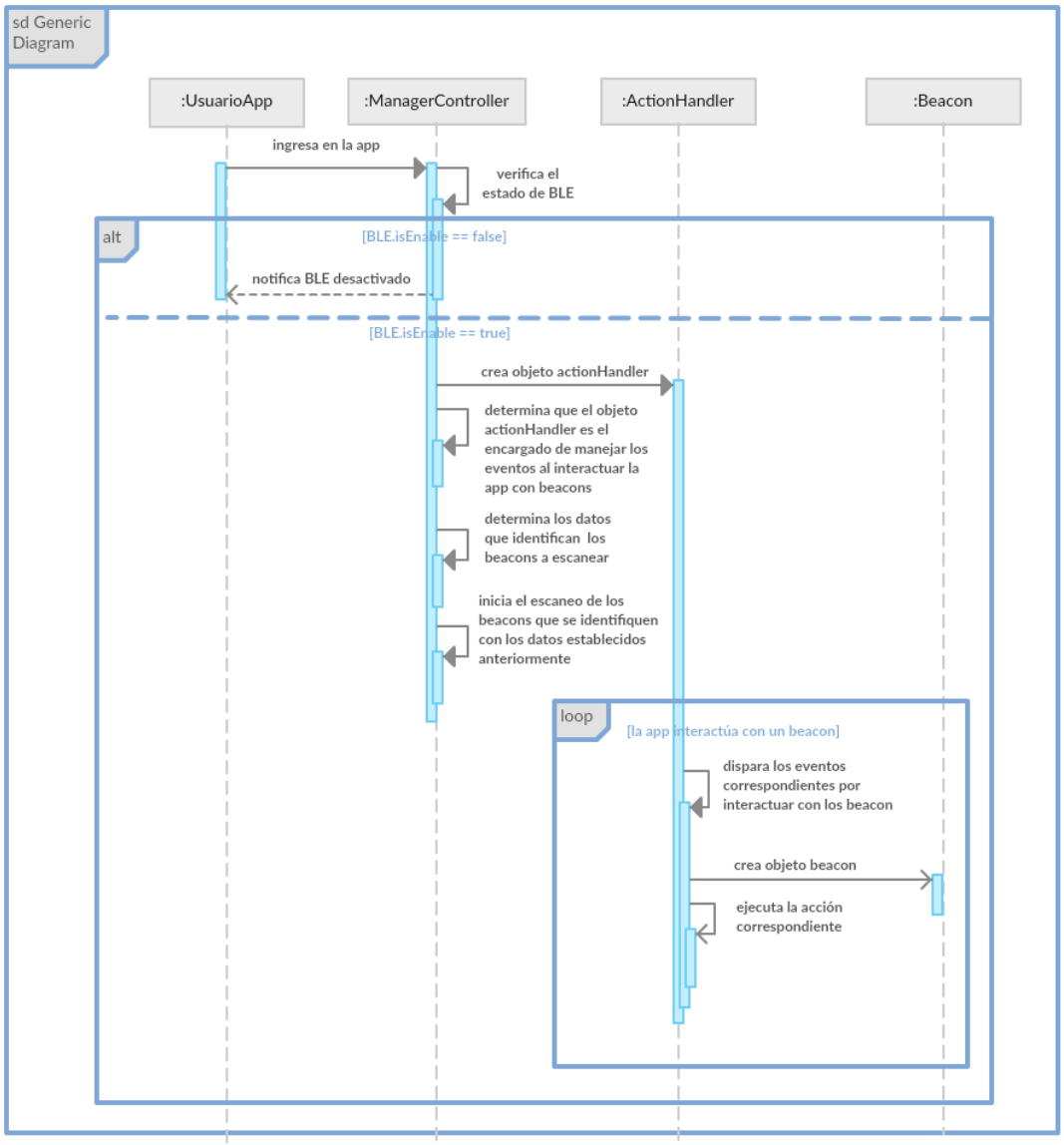

Figura 3. Diagrama de secuencia genérico del patrón Beacon Action Manager

4. Aplicación del Patrón

A continuación se presenta una aplicación móvil donde se implementa el patrón Beacon Action Manager presentado anteriormente.

4.1. Aplicación Luggage Finder

Luggage Finder es una aplicación iOS que permite saber cuándo recoger el equipaje en el aeropuerto. Todo lo que se necesita es colocar un beacon dentro de la valija, especificar los datos del beacon en la aplicación, y ésta se encargará de detectar la valija, indicando a qué distancia se encuentra. La distancia máxima de alcance es de aproximadamente 50 metros. En caso que la aplicación se encuentre en background el usuario también recibirá notificaciones sobre su equipaje.

Este caso de estudio se implementó utilizando el framework CoreLocation [Apple 2016] provisto por Apple.

4.1.1. Estructura de la aplicación

A continuación, se describen las principales clases que componen la aplicación y su correspondencia con las clases del patrón, seguido del diagrama de clases en la Figura 4.

- *UIViewController:* Al iniciar la aplicación el usuario observa una vista de la clase UIViewController que contiene un botón "+" (Add). Al presionarlo lo lleva a la siguiente vista de la clase AddBeaconTableViewController.
- *AddBeaconTableViewController:* En Esta clase el usuario ingresa los valores que permiten identificar el beacon que se colocará dentro de la valija. Estos valore son: UUID, Major y Minor.

  Al presionar el botón 'Save' los datos se guardan en la aplicación y se accede a la siguiente vista de la clase SearchViewController.
- *SearchViewController:* Esta clase cumple el rol de ManagerController del patrón Beacon Action Manager. Entre sus tareas:
  - Verifica si el Bluetooth en el dispositivo móvil se encuentra activado, de lo contrario se muestra un mensaje de advertencia al usuario.
  - Específica de que beacon la aplicación está interesada en recibir información, que sería aquel que se identifique con los valores ingresados por el usuario.
  - Determina qué objeto va a ser el encargado de manejar todos los eventos que ocurran cuando el dispositivo móvil se encuentre dentro de la región del beacon especificado anteriormente.
  - Inicia el escaneo del beacon.
- *ActionHandler*: La clase ActionHandler actúa como ActionHandler en el patrón Beacon Action Manager. Esta clase implementa el protocolo CLLocationManagerDelegate que define métodos para recibir información sobre la localización del dispositivo. Éstos son invocados desde el thread donde se inició el servicio de localización (el inicio del escaneo de beacons). Se implementaron los métodos del protocolo que realizan las siguientes acciones:
  - Notificar cuando la aplicación ha entrado o salido de la región del beacon mostrando una notificación push en el dispositivo.

- Una vez dentro de la región de un beacon cuando cambie la proximidad del dispositivo móvil al beacon se notifica al usuario los cambios en la proximidad, se le informa al usuario la distancia a la cual se encuentra.

- *Beacon*: La clase Beacon actúa como Beacon en el patrón Beacon Action Manager. Contiene una variable de tipo CLBeacon que almacena información del beacon encontrado como ser: los valores UUID, major, minor que permiten identificar al beacon, la proximidad del dispositivo al beacon, la precisión de la proximidad y la intensidad de la señal emitida por el beacon.

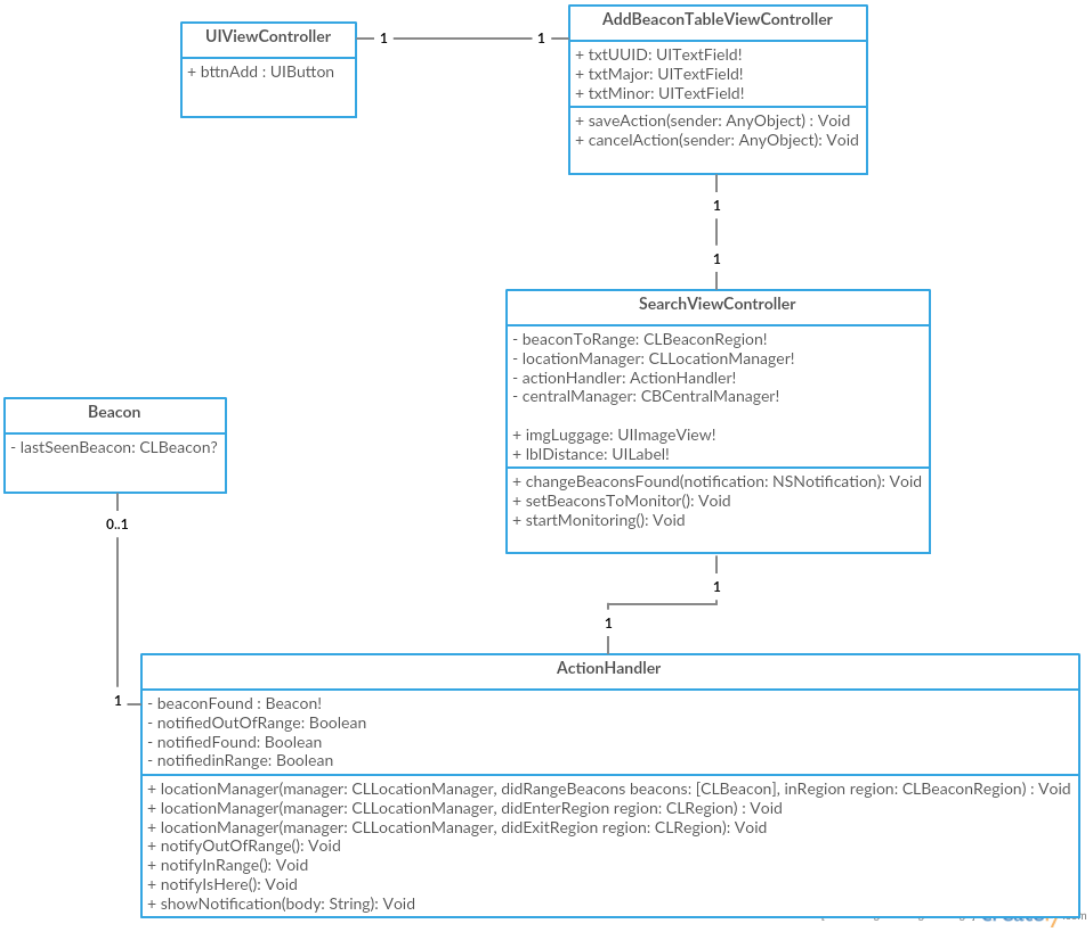

Figura 4. Diagrama de clases de la aplicación Luggage Finder

4.1.2 Diagramas

A continuación se muestran los diagramas correspondientes a algunos escenarios.

- *Escenario 1: inicio del escaneo*: El usuario inicia la aplicación, ingresa los datos que identifican al beacon, el bluetooth se encuentra activado y se inicia el escaneo y monitoreo del beacon (Figura 5).

- *Escenario 2: cambio de proximidad*: Se asume que el usuario inició la aplicación, ingresó los datos que identifican al beacon, el bluetooth del dispositivo se encuentra activado, el escaneo y monitoreo del beacon se iniciaron y la aplicación se encuentra dentro del rango del beacon y corriendo en

foreground. En el diagrama se modela el cambio de proximidad de la aplicación al beacon, donde pasa de ser 10 metros a 5 metros (Figura 6).

- *Escenario 3: BLE desactivado:* El usuario inicia la aplicación, ingresa los datos que identifican al beacon pero el bluetooth se encuentra desactivado en el dispositivo (Figura 7).

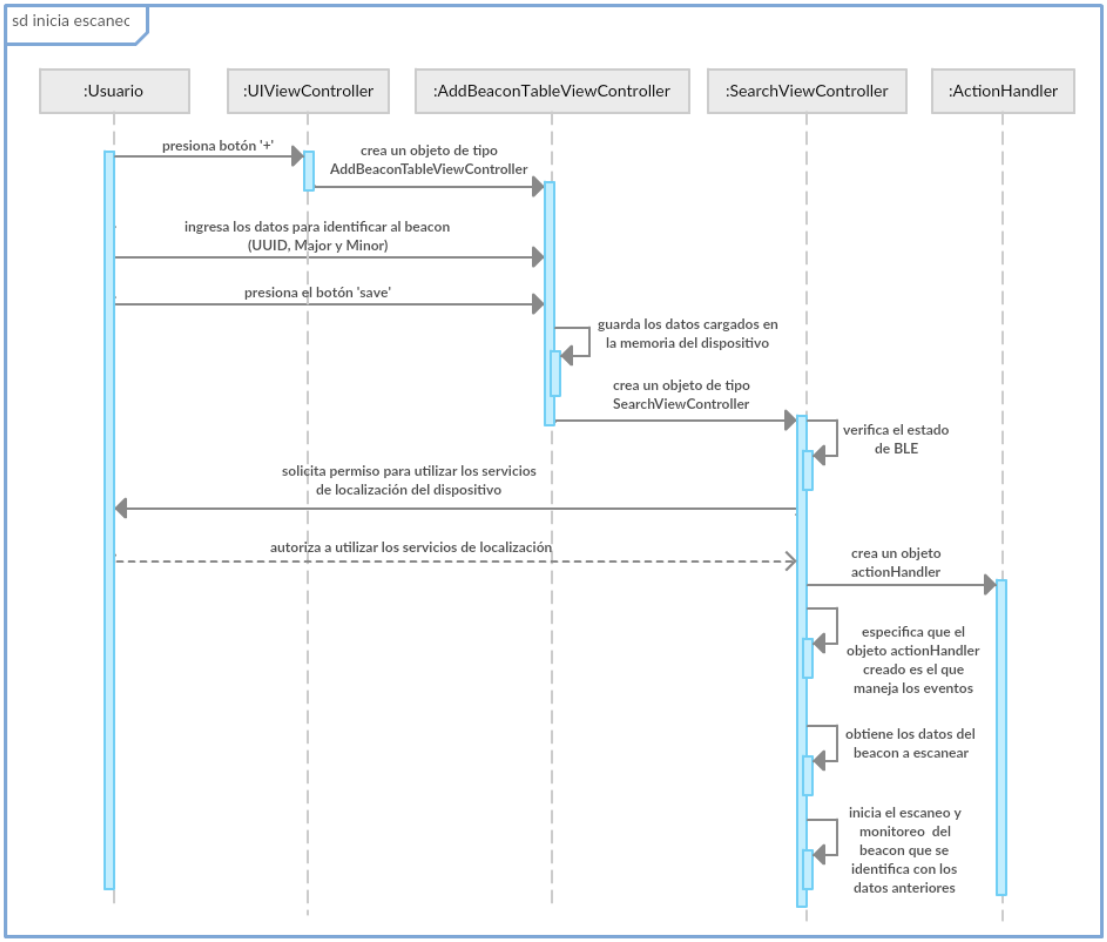

Figura 5: Diagrama de secuencia del Escenario 1: inicio del escaneo.

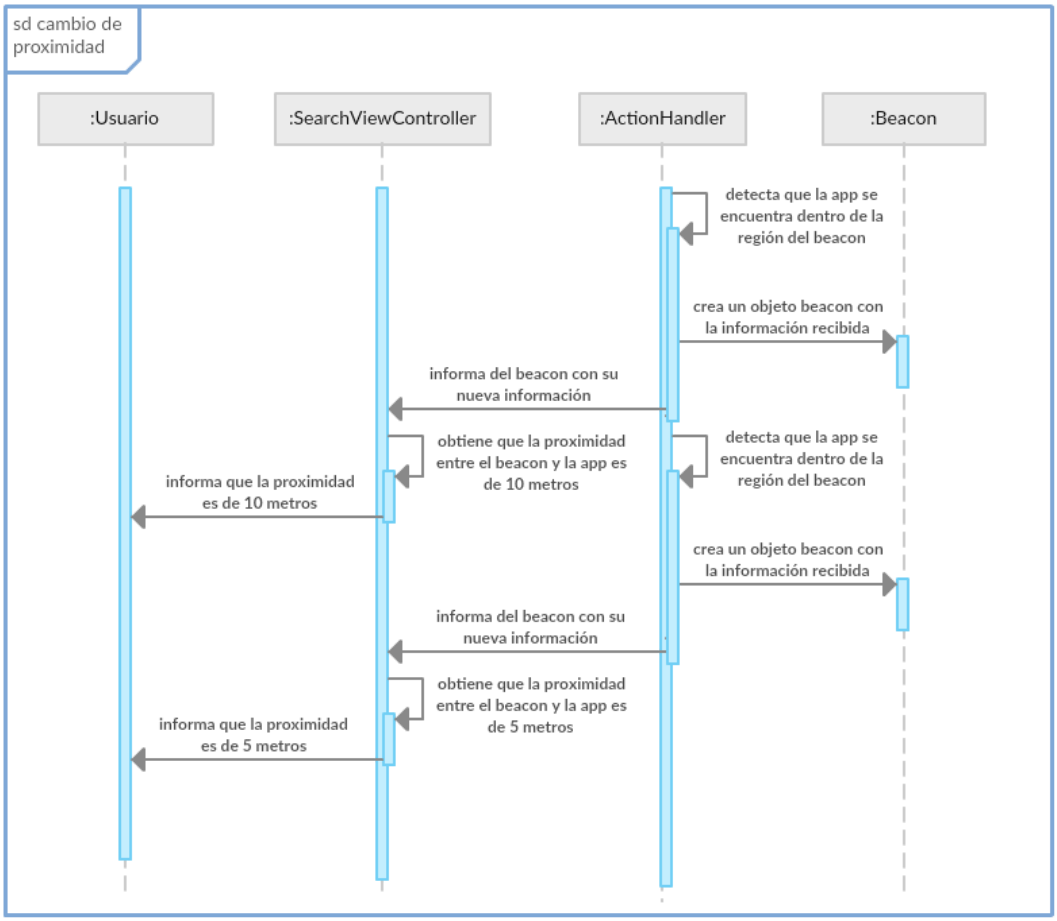

Figura 6: Diagrama de secuencia del Escenario 2: cambio de proximidad.

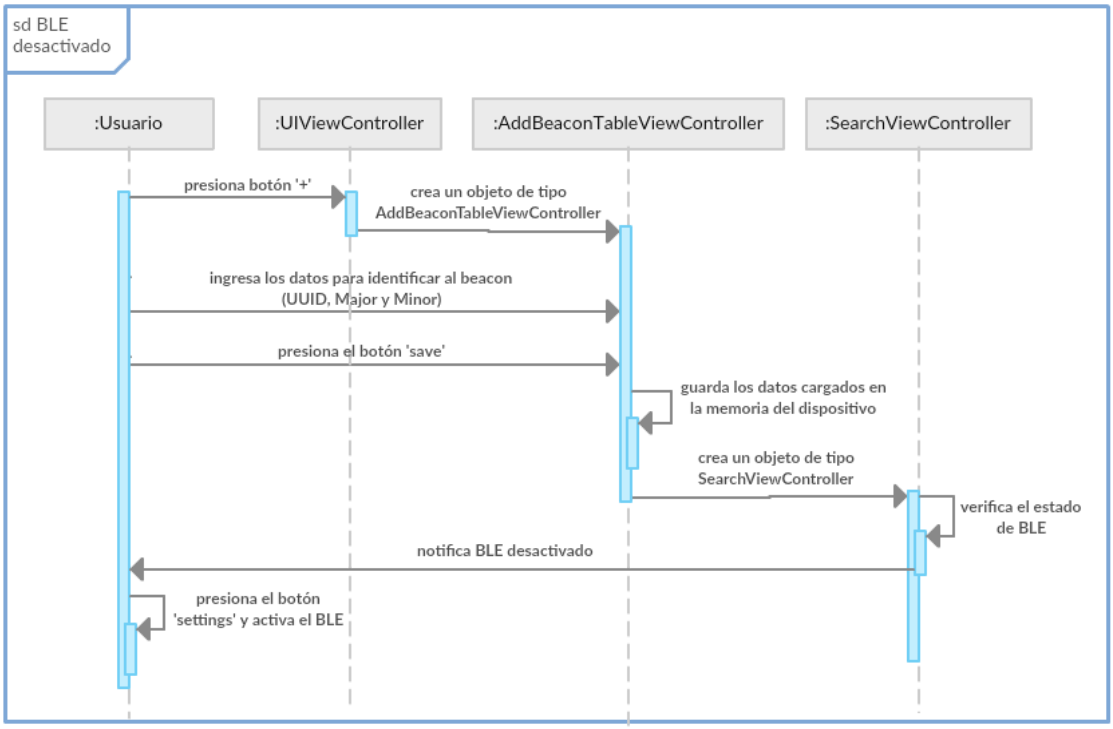

Figura 7: Diagrama de Secuencia del Escenario 3: BLE desactivado.

4.2. Otras aplicaciones

Si bien el patrón ha sido aplicado exitosamente en diversas aplicaciones móviles de diferentes ámbitos no es posible detallarlas a todas. A continuación se nombran algunas de éstas aplicaciones:

4.2.1. Airport App

Airport App es una aplicación Android que ayuda a los usuarios a ubicarse dentro del Aeropuerto. El usuario ingresa el número de gate de su próximo vuelo, y los beacons distribuidos en el interior del aeropuerto interactúan con la aplicación indicandole hacia dónde dirigirse y en qué otro lugar del aeropuerto podrá encontrar más información para llegar a destino. Al aproximarse a los beacons, la aplicación muestra un mensaje con la información y la reproduce en audio para que personas con capacidades visuales reducidas puedan utilizarla también.

4.2.2. Zoo App

Zoo App es una aplicación iOS de un zoológico que brinda información de los animales a medida que se va recorriendo el lugar. El usuario selecciona los animales de los cuales le interesa conocer más información durante el recorrido. Dentro del zoológico se encuentran distribuidos diferentes beacons que interactúan con la aplicación de tal forma que cuando el usuario se encuentre observando los animales que seleccionó previamente, reciba información detallada sobre ellos.

4.2.3. Bakery App

Bakery App es una aplicación Android de la pastelería Bakery Good que ofrece diferentes promociones a los clientes que se encuentran en su local. Según la ubicación de los usuarios dentro de la tienda reciben descuentos o notificaciones diferentes.

5. Conclusiones y Futuras Extensiones

A la hora de plantear la arquitectura de una aplicación que se relaciona con dispositivos beacon aparecen nuevos problemas y desafíos de diseño para los desarrolladores que necesitan integrar ésta nueva tecnología en sus aplicaciones móviles.

En este artículo se mostró que es posible definir un patrón de diseño para aplicaciones móviles que interactúan con dispositivos beacons para conocer el contexto de los usuarios. La utilización de este patrón permite prevenir aquellos problemas típicos y recurrentes que puedan surgir a la hora de desarrollar este tipo de aplicaciones móviles.

Para llevarlo a cabo se desarrollaron diferentes aplicaciones móviles que interactúan con dispositivos beacon situadas en diversos contextos. En este artículo solo se mencionaron algunas de ellas. En cada una de éstas aplicaciones se implementó exitosamente el patrón especificado en este proyecto para establecer la comunicación entre las aplicaciones móviles y los dispositivos beacon, obteniendo un diseño claro de fácil comprensión.

Se realizó una validación experimental donde se compartió el patrón con diferentes desarrolladores de aplicaciones móviles. Se los introdujo en el tema y se les

pidió que desarrollen una aplicación utilizando el patrón. Los resultados fueron los esperados:

- Pudieron realizar las aplicaciones en relativamente poco tiempo (comparado con el tiempo que se invirtió en investigación antes de desarrollar el patrón).
- Las aplicaciones desarrolladas pertenecen a diferentes ámbitos de aplicación.
- El mayor tiempo de investigación se invirtió en las librerías/SDKs a utilizar según el lenguaje de implementación.
- El patrón les brindó un marco de trabajo estándar dentro del cual manejarse que les facilitó el desarrollo.

Como contribución, se puede mencionar que este proyecto proporciona una herramienta útil, confiable, probada e independiente del lenguaje para aquellos desarrolladores que decidan incorporar esta nueva tecnología en sus aplicaciones.

Algunas de las posibles extensiones de este trabajo de investigación incluyen:

- Adaptar el patrón para que los dispositivos beacons puedan interactuar con dispositivos wearables proveídos de la tecnología BLE. Los dispositivos wearables hacen referencia a aquellos aparatos y dispositivos electrónicos que se incorporan en alguna parte del cuerpo interactuando de forma continua con el usuario y con otros dispositivos, por ejemplo relojes inteligentes.
- Extender el patrón para ser utilizado en aplicaciones móviles que interactúan con dispositivos beacons para establecer localización en interiores (indoor location). La información que una aplicación móvil recibe al interactuar con un beacon se puede utilizar para situar al usuario en un mapa dentro del lugar donde se encuentre. Llegados aquí se aprecian ya tres de las limitaciones actuales de la localización en interiores: 1) No se trata únicamente de posicionar en un mapa la información provista por los beacons, sino que cada lugar donde se implemente tiene que crear su propia "cartografía". 2) Hay que asociar los beacons a una posición permanente dentro de esa cartografía. 3) La exactitud debe ser mucho mayor que en el caso del GPS.
- Desarrollar una Herramienta que permita al usuario representar gráficamente la interacción entre una aplicación móvil y los dispositivos beacons y a partir de allí generar automáticamente el código fuente utilizando el patrón planteado en este documento.

Referencias

[ACM 2014], The 12th ACM Conference on Embedded Networked Sensor Systems. http://sensys.acm.org/2014/.

[Alexander C., Ishikawa S., Silverstein M., Jacobson M., Fiksdahl-King I, and Angel S. 1997] "A Pattern Language". Oxford University Press, NewYork, 1997.

[Apple 2016] Core Location Framework Reference, https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation_Framework/index.html#//apple_ref/doc/uid/TP40007123.

[Bluetooth 2016] “What is Bluetooth Technology?”, Low Energy, http://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy.

[Byun D., Cho J., Moon S., Hong D. 2016] “S-Beacon: Next generation BLE beacon solution for enhanced personalization”, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7430566&queryText=beacon&sortType=desc_p_Publication_Year&rowsPerPage=100.

[CASCON 2015] Centre for Advanced Studies Conference, https://www-01.ibm.com/ibm/cas/cascon/

[Deepesh P. C., Rath R., Tiwary A., Rao V., Kanakalata N. 2016] “Experiences with using iBeacons for Indoor Positioning”, http://dl.acm.org/citation.cfm?id=2856654.

[Gamma E., Helm R., Johnson R., Vlissides J. 1994] “Design Patterns. Elements of Reusable Object-Oriented Software”. Addison Wesley (GoF- Gang of Four).

[IEEE 2016], Institute of Electrical and Electronics Engineers. http://ieeexplore.ieee.org

[ISEC 2016], 9th India Software Engineering Conference. http://www.isec2016.org.

[Jain P., Khanwalkar S., Malhotra R., Dheenrajappa A., Gupta G., Kobsa A. 2016] “uBeacon: Configuration based Beacon tracking”, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7457066&queryText=beacon&sortType=desc_p_Publication_Year&rowsPerPage=100.

[Martin P., Ho B., Grupen N., Muñoz S., Srivastava M. 2014] “An iBeacon primer for indoor localization: demo abstract”, http://dl.acm.org/citation.cfm?id=2675028.

[Sykes E., Pentland S., Nardi S. 2015] “Context-aware mobile apps using iBeacons: towards smarter interactions”, http://dl.acm.org/citation.cfm?id=2886463.