

## SISTEMA DE NAVEGACIÓN EN INTERIORES PARA DISPOSITIVOS MÓVILES UTILIZANDO BEACONS CON BLUETOOTH DE BAJA ENERGÍA

*Andrés Morales Reyes, Isidro Robledo Vega, Rogelio E. Baray Arana*

Tecnológico Nacional de México  
Instituto Tecnológico de Chihuahua  
División de Estudios de Posgrado e Investigación  
Ave. Tecnológico #2909, Col. 10 de Mayo, Chihuahua, México  
Tel. +52(614)201-2014  
[amoralesre, irobledo, rbaray]@itchihuahua.edu.mx

### RESUMEN

En este artículo se describe el diseño y desarrollo de un sistema de navegación en interiores para dispositivos móviles utilizando beacons que difunden señales Bluetooth de baja energía. El sistema consta de tres etapas: En la primera etapa se desarrolla el mapa del edificio en donde se ubicarán los usuarios. En la segunda etapa se localiza el dispositivo móvil dentro del edificio para ubicarlo en el mapa. En la tercera etapa se genera la ruta de la ubicación actual a un destino deseado. Se utilizaron beacons de la marca Kontakt para la localización y se utilizó Unity para el desarrollo de la aplicación que despliega el mapa y la ruta de navegación. Se realizó una extensa experimentación para caracterizar la operación de los beacons y definir el algoritmo de posicionamiento en interiores. También se probaron diferentes tecnologías para crear y desplegar mapas en 2D y 3D para lograr los efectos de animación deseados en la aplicación desarrollada para los dispositivos móviles.

**Palabras clave:** posicionamiento en interiores, navegación en interiores, beacons con Bluetooth de baja energía

### ABSTRACT

In this paper, we describe the design and development of an indoor navigation system for mobile devices using Bluetooth Low Energy (BLE) Beacons. The system consists of three stages: In the first stage, we developed the map of the building where the users will be located. In the second stage, the physical position of the mobile device inside the building is obtained to locate it on the map. In the third stage, the route from the current location to a desired destination is generated. Kontakt beacons were used to locate the mobile devices and Unity was used to program the app that displays the map and navigation paths. Extensive experimentation was carried out to characterize the operation of the beacons and define the indoor positioning algorithm. Different technologies were also tested to create and display map in 2D and 3D to achieve the desired animation effects in the application developed for mobile devices.

**Keywords:** indoors positioning, indoors navigation, BLE Beacons.

### 1. INTRODUCCIÓN

La ubicación de lugares es un inconveniente cuando se desconoce la zona y se ha ido complicando cada vez más con el crecimiento de las ciudades. En 1960, diversas organizaciones del gobierno de E.U.A. incluidas entre ellas el

Departamento de Defensa, la NASA y el Departamento de Transporte iniciaron el desarrollo del Sistema de Posicionamiento Global, hoy conocido como GPS [1]. Gracias a esta tecnología es posible conocer nuestra ubicación de una manera precisa y poder navegar a través del mundo con mayor facilidad y rapidez. Aunque el GPS ha pasado a través de diversas mejoras, aún tiene un inconveniente, las señales de satélite que hacen posible la localización no son capaces de atravesar la estructura de los edificios limitando su uso solamente en exteriores.

Para el desarrollo de un sistema de navegación en interiores son necesarios tres componentes. El primer componente es la tecnología para localizar el objetivo, el segundo es el mapa del edificio en el cual se desplazará el objetivo y el tercero es el algoritmo que permite definir una ruta trazada sobre el mapa del edificio para ir de un lugar a otro. Existen diferentes opciones de tecnología para el desarrollo de cada componente. En esta investigación se integraron varias tecnologías para el desarrollo de un sistema de navegación en interiores.

Para el posicionamiento en interiores, existen muchas investigaciones que trabajan con distintas tecnologías como rayos infrarrojos (IR), ultrasonido, Bluetooth de baja energía (BLE) [2], identificación por radio frecuencia (RFID), redes de área local inalámbricas (WLAN), ultra banda ancha (UWB), señales magnéticas [3], redes de sensores, sistemas de visión por computadora, luz visible y sonido audible. Todas ellas con sus respectivas ventajas y desventajas [4–7]. Existen sistemas híbridos que han combinado distintas tecnologías para cubrir las desventajas que se presentan y obtener mejores resultados [8]. También existen investigaciones en las que se apoyan en distintos modelos matemáticos, técnicas y algoritmos para mejorar la precisión de estas tecnologías [9–12]. Aun así, no existe un desarrollo contundente que se defina como el estándar para los sistemas de posicionamiento en interiores.

Para saber dónde estamos ubicados es muy importante tener bien definido el mapa del espacio físico. El ser humano ha trazado mapas desde hace más de 2300 años [13], los cuales con el paso del tiempo y gracias a la tecnología se han ido detallando y evolucionando hasta llegar a lo que hoy conocemos como mapas digitales, estos muestran información a nivel mundial sobre la ubicación de todo tipo de lugares,

incluso algunos muestran su mapa interior. Existen distintos proveedores de mapas, como los son Google, Bing, Apple, OpenStreet, Here, MapQuest, Yandex, Wikimapia y otros. Estos cuentan con distintas características que los hacen únicos. Algunos son muy exclusivos, como es el caso de Apple Maps los cuales solo están disponibles para dispositivos con iOS y OS X, otros están abiertos al público y son compatibles con gran parte de los dispositivos móviles, este es el caso de Google, Bing, Open Street y Here. Sin embargo, existe una notable diferencia entre ellos y está relacionada con la forma en que se actualiza su información. Existen mapas, como Open Street, que son de código abierto y pueden ser editados en tiempo real sin restricción alguna [14]. Gran parte de los proveedores de mapas cuentan con una interfaz de programación de aplicaciones (API), que permite utilizar los mapas de un proveedor en el desarrollo aplicaciones [15]. Otra opción para la creación de mapas es por medio de la utilización de HTML, CSS y SVG, en conjunto con otras herramientas web como Javascript [16]. También existe un estándar establecido por la OCG (Consortio Geoespacial Abierto) llamado IndoorGML el cual está diseñado específicamente para la construcción de espacios interiores y funciona mejor que otras herramientas debido a su separación topográfica por capas [17].

Para poder navegar en interiores es necesario definir la ruta a seguir en un mapa. Definir una ruta es una tarea complicada, debido a que hay que tomar en cuenta ciertos factores, tal como la distancia entre el punto de inicio y el destino, los obstáculos fijos y móviles, el acceso que tiene cada usuario, entre otros. IndoorGML soluciona este problema debido a su estructuración de información por capas, en donde se pueden definir los nodos y las rutas posibles de seguir, así como objetos móviles como lo son las puertas [18]. Existen distintos desarrollos que proponen otros métodos para la definición de rutas [19], así como técnicas para la optimización de las mismas [20].

Actualmente podemos encontrar empresas que ofrecen sistemas de navegación en interiores como Senion que trabaja con una arquitectura de Wi-Fi, Beacons BLE y los sensores incluidos en los dispositivos inteligente, como el altímetro, la brújula, el giroscopio y el acelerómetro [21]. Accuware [22] ofrece un sistema muy similar al de Senion. INSOFT ofrece un sistema que integra Wi-Fi, Bluetooth, RFID y UWB [23]. Indoor Atlas utiliza tecnología geomagnética [24]. En este proyecto se tomó en cuenta el análisis realizado sobre estos sistemas para ofrecer una solución eficaz a un menor costo.

## 2. GENERACIÓN DE MAPAS PARA NAVEGACIÓN EN INTERIORES

Los mapas para los sistemas de navegación en interiores pueden ser generados de distintas maneras. La opción más común es el uso de las herramientas de Google [25], Open Street Map (OSM) [26] o los Gráficos Vectoriales Escalables (SVG). En el sistema desarrollado consideramos la necesidad de crear una herramienta para generar el mapa que permita utilizar la información de los planos de edificios, que por lo

general son elaborados utilizando paquetes comerciales como Autocad, por lo que se decidió utilizar SVG ya que es posible exportar capas de un plano digital a este formato.

### 2.1. Editor de Gráficos Vectoriales

Para crear un mapa en formato SVG se pueden utilizar distintas herramientas, Inkscape es una de ellas. Este es un editor de gráficos vectoriales que utiliza los SVG como formato nativo. Inkscape es utilizado para crear una amplia variedad de gráficos como ilustraciones, logotipos, diagramas, mapas, gráficos web. Inkscape es gratuito y de código abierto, tiene una interfaz muy amigable, lo cual hace sencilla y rápida la creación y edición del mapa [27]. Se utilizaron solo cuadros con las dimensiones de las oficinas del plano del edificio y se usó una función para agrupar los elementos. El archivo en formato SVG se va generando automáticamente al ir colocando elementos en el editor. El archivo SVG se puede visualizar con un procesador de textos, pero es muy complicado realizar ajuste en el gráfico modificando el código ya que es difícil identificar elementos específicos. Por lo tanto, es conveniente crear el mapa llevando un orden específico al crear los elementos y poder efectuar modificaciones en el código rápidamente. La figura 1 muestra el mapa del edificio de prueba creado en Inkscape.

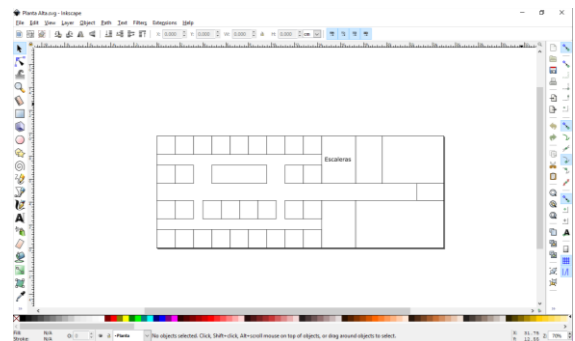


Figura 1.- Mapa SVG creado en Inkscape.

### 2.2. Generación del Mapa por Código

Para realizar el mapa SVG por código se usó la herramienta web llamada CodePen [28]. Esta muestra en tiempo real las modificaciones que se van realizando en el código. La figura 2 muestra la interfaz de CodePen y parte del desarrollo del mapa.

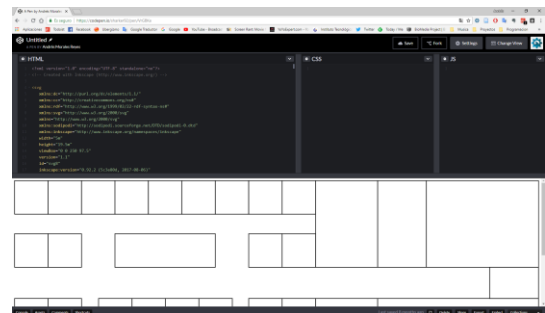


Figura 2.- Mapa creado con CodePen.

CodePen permite tener un código más simple y ordenado pero se requiere mayor tiempo de desarrollo. La funcionalidad de CodePen permite realizar animaciones en el mapa SVG para agregar elementos que representen el dispositivo móvil y trazar rutas de un punto a otro.

### 2.3 Visualización del Mapa SVG en la Aplicación Móvil

Se desarrolló una aplicación móvil de prueba para dispositivos Android utilizando el ambiente de desarrollo *Android Studio*. Se utilizó el *Vector Asset Studio* para convertir el mapa SVG a XML ya que Android no soporta nativamente SVG. Durante el proceso de conversión se perdió la animación creada en CodePen, dejando sin esa funcionalidad al mapa, por lo que será necesario crear el mapa directamente en XML para evitar la conversión. El mapa se insertó en la actividad que se ejecuta al presionar el botón "MAP" de la aplicación. La figura 3 muestra la interfaz de la aplicación móvil.

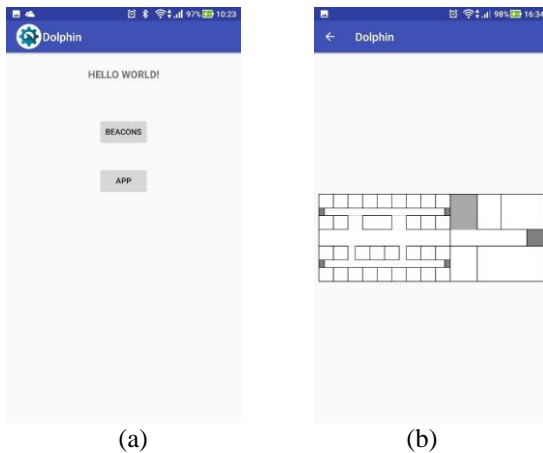


Figura 3.- Interfaz de la aplicación.(a) Vista principal (b)Vista del mapa con la pantalla en posición vertical.

Al igual que el SVG, el XML no pierde calidad al cambiar sus dimensiones, lo cual se puede comprobar observando la figura 4, donde se aprecia el mapa en la pantalla con posición horizontal. El uso de gráficos vectoriales escalables permite girar el celular, ampliar o reducir el mapa sin perder calidad en la visualización.

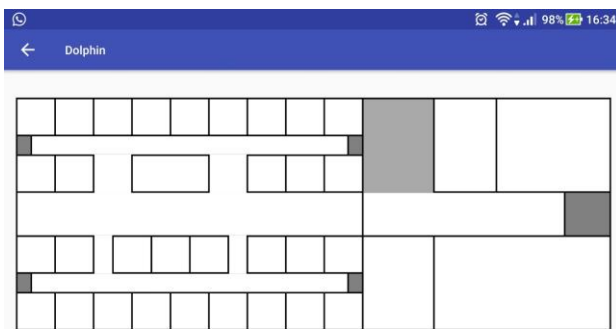


Figura 4.- Vista horizontal del mapa en la aplicación móvil.

### 2.4 Animación del Mapa en la Aplicación Móvil

Se realizaron pruebas de animación con las funciones de Android Studio, se probó con las funciones *Property Animation*, *Frame Animations*, *AnimatedVectorDrawable* y sobrecargando la función *onDraw* con dibujos personalizados, y se logró modificar la posición de un objeto con coordenadas absolutas y relativas, rotarlo, cambiar su tamaño, su color, su opacidad, y hacerlo recorrer una trayectoria, pero todas estas acciones fueron preprogramadas y se requiere de un evento específico para iniciar la animación. No fue posible realizar animaciones con cambios en tiempo real ya que las librerías son privadas y no permiten el acceso a los elementos internos del vector para modificarlo. Se porbaron dos librerías externas para realizar la animación en tiempo real, estas son Vector Master y RichPath. La librería de RichPath contiene más métodos y funciones que Vector Master, esto permite un mayor control sobre los gráficos vectoriales. Usando RichPath se logró modificar los valores internos de los gráficos vectoriales y hacer que se desplazaran por el mapa, pero se tuvo problemas con el manejo de coordenadas relativas y absolutas. El mayor inconveniente fue al cambiar el tamaño de la pantalla, la traslación no se escala al realizar el cambio de tamaño, por lo que los resultados son distintos en pantallas de tamaño diferentes.

### 2.5 Visualización de Mapas 3D con Unity

Se decidió desarrollar la aplicación para desplegar el mapa utilizando Unity. Este es un software multiplataforma muy popular para la programación de videojuegos en 2D y 3D, pero también es utilizado para desarrollar simuladores y videojuegos de realidad virtual o aumentada, entre otros proyectos. Existen otras plataformas de software para la programación de videojuegos, como Unreal Engine y CryEngine, se eligió Unity debido a que es gratuito y permite realizar aplicaciones compatibles con Android e iOS. Una de las ventajas es la facilidad para implementar las colisiones y definir las áreas por las que se puede desplazar el objeto gráfico que representa la ubicación del dispositivo móvil. Para tomar la decisión de migrar a Unity primero se investigó la posibilidad de que la aplicación pudiera interactuar con las librerías de los beacons BLE. Se encontró una solución ofrecida en la tienda de activos de Unity llamada iBeacon, contiene funciones que permiten conectarse con los beacons.

La aplicación se desarrolló en 3D ya que en Unity es muy fácil indicar que objeto en el mapa puede ocasionar colisiones, además de los beneficios de visualización como los cambios de perspectiva. Existen diferentes programas que permiten crear mapas en 3D como Blender, Maya y 3Ds Max. Se utilizó este último ya que está orientado al diseño arquitectónico y facilita la creación de estructuras en 3D con paredes, puertas y ventanas. La figura 5 muestra el mapa del edificio de prueba construido con 3Ds Max.

El mapa 3D fue exportado directamente a la aplicación en Unity donde se crearon las animaciones necesarias para mostrar la ubicación del dispositivo móvil por medio de una esfera roja que se puede mover por los pasillos del edificio.

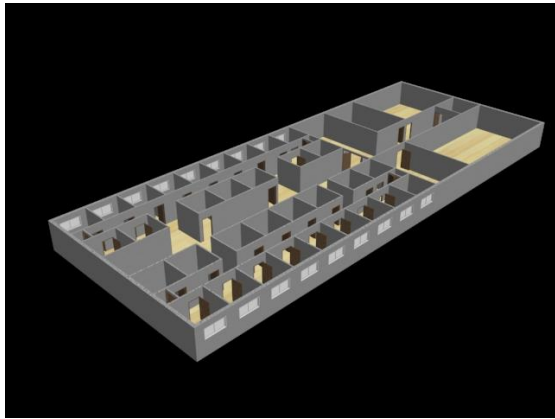


Figura 5.- Mapa 3D del edicio de prueba creado con 3Ds Max.

### 3. LOCALIZACIÓN DE DISPOSITIVOS MÓVILES

Se realizó una comparación entre las tecnologías disponibles en los dispositivos móviles con las que se podía lograr la localización en interiores. Las cuatro opciones son:

**WLAN.-** Utiliza redes Wi-Fi para calcular la posición del dispositivo móvil utilizando principalmente la técnica de trilateración [8,19].

**Bluetooth.-** Específicamente bluetooth 4.0 que se le conoce como Bluetooth de baja energía (BLE). Se ha convertido en el estándar de comunicación más usado para el Internet de las cosas (IoT). También se utiliza la trilateración para calcular la posición del dispositivo móvil [2].

**Visión.-** Utiliza la cámara frontal del dispositivo móvil para recibir información de lámparas LED y de esta manera poder determinar en donde se encuentra [29].

**Magnetismo.-** Todos los edificios tienen una huella magnética única, la cual se aprovecha utilizando el sensor magnético del dispositivo móvil para estimar en que parte del edificio se encuentra [26].

Se realizó un análisis comparativo de las cuatro tecnologías tomando en cuenta aspectos como disponibilidad en el mercado, precisión, escalabilidad, seguridad y costo, siendo BLE la tecnología mejor evaluada.

#### 3.1 Beacons BLE

Para la implementación de un sistema de posicionamiento en interiores basado en BLE se requieren dispositivos llamados "Beacons". Estos dispositivos emiten señales donde se envían paquetes de información a través de Bluetooth en un intervalo de tiempo definido. La información que envían varía dependiendo del fabricante. Los dos principales fabricantes son *Estimote* [30] y *Kontakt* [31]. A la fecha del desarrollo el Beacon más completo de Kontakt es el *Beacon Pro* y el de Estimote es el *Location Beacon*, mostrados en la figura 6. Después de analizar sus propiedades y disponibilidad en el mercado nacional se seleccionó el Beacon Pro de Kontakt. Este cuenta con una interfaz web para visualizar y modificar sus parámetros. Algunos de los parámetros que se pueden modificar son los perfiles *iBeacon* y *Eddystone*, nombre de

identificación, potencia de transmisión, intervalo de señal, etiquetas, alertas de batería baja, ahorro de energía y configuraciones del perfil de seguridad. También se pueden obtener parámetros fijos como la versión de firmware, porcentaje de la batería, dirección MAC, fecha y hora de la última conexión e identificador único. Otra forma de modificar los parámetros es por medio de una aplicación para Android o iOS desde un dispositivo móvil, que se conecta con los beacons disponibles por medio de Bluetooth.

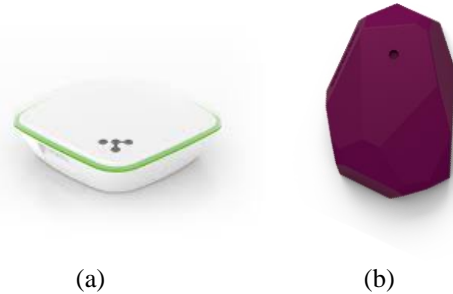


Figura 6.- Beacons BLE (a) Beacon Pro de Kontakt y (b) Location Beacon de Estimote.

Se decidió utilizar la plataforma Android para interactuar con los beacons ya que es la de mayor crecimiento en el mercado de dispositivos móviles. La tecnología BLE es soportada por el sistema operativo Android a partir de la versión 4.3. Para el desarrollo de aplicaciones de interacción con los beacons se utilizó Android Studio. La interfaz de la aplicación mostrada en la figura 3a contiene un botón que permite el acceso a una interfaz donde se listan los beacons detectados por medio de señales BLE con su respectiva información, esta permite el cálculo de la distancia entre el beacon y dispositivo móvil. La aplicación actualiza la información en tiempo real agregando nuevos beacons detectados o eliminando los que quedan fuera de alcance de la señal BLE. El Beacon Pro de Kontakt soporta los protocolos iBeacon, Eddystone y el protocolo seguro de Kontakt.io. Después de algunas pruebas se decidió utilizar el protocolo iBeacon pues fue el que mejor se comportó al conectar, actualizar y desconectar los beacons de acuerdo a las señales BLE detectadas.

#### 3.2. Cálculo de la Distancia del Beacon al Dispositivo Móvil

Existen diferentes procedimientos para el cálculo de la distancia entre el beacon y el dispositivo móvil basados en el indicador de intensidad de la señal recibida (RSSI) que es obtenido del beacon mediante el protocolo de comunicación. El SDK de Kontakt proporciona una función que calcula la distancia, pero no hay información en la documentación sobre la fórmula que utiliza. Otro procedimiento para el cálculo de la distancia desarrollado por los creadores de AltBeacon utiliza la fórmula de la Ec. 1.

$$D = (0.89976) \times \frac{RSSI^{7.7095}}{TxP} + 0.11 \quad (1)$$

Donde D es la distancia calculada, RSSI es el indicador de intensidad recibida del beacon y TxP es la potencia de transmisión del dispositivo móvil. Los valores 0.89976, 7.7095 y 0.111 son constantes calculadas por sus desarrolladores para generar una curva que se ajuste mejor al comportamiento de un dispositivo específico. Existe un tercer procedimiento publicado en [32] basado en la fórmula de la Ec. 2.

$$D = \sqrt{10^{\frac{TxP-RSSI}{10}}} \quad (2)$$

Se realizaron pruebas de medición de distancias en exteriores colocando los beacons a diferentes alturas, cerca del suelo y a 2m de altura. Se utilizaron los tres procedimientos mencionados anteriormente. Se determinó que los dos primeros procedimientos dan los mismos resultado y el comportamiento es mejor a 2m del suelo. Al revisar diferentes estudios nos percatamos que estas fórmulas no funcionan adecuadamente y su formulación depende de un dispositivo específico, por lo que se requieren varias pruebas para ajustarlas. Entonces se realizó un estudio en interiores para determinar el comportamiento de valor del RSSI de los beacons a diferentes distancias y ver la posibilidad de obtener una caracterización matemática que permita conseguir el objetivo de posicionamiento en interiores. Se utilizaron 4 beacons (1101, 1102, 1103 y 1104), los tres primeros con una potencia de trasmisión de -72dBm y el cuarto a -77dBm. Se tomaron mediciones del RSSI durante 30 segundos a diferentes distancias. La figura 7 muestra la gráfica con los promedios de los datos medidos. Se puede observar que existen variaciones en las mediciones de cada beacon. Como los valores de RSSI aumentan y disminuyen de una medición a otra, es difícil utilizarlos para establecer una relación que permita aproximar un valor de distancia aceptable.

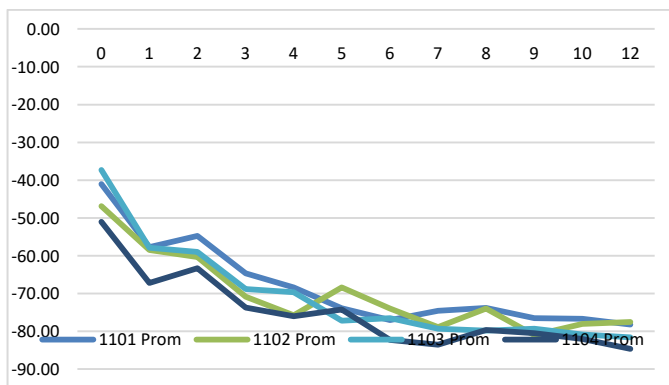


Figura 7.- Gráfica del promedio del RSSI (eje vertical) de 4 beacons a diferentes distancias (eje horizontal).

### 3.3. Localización de los Beacons en el Edificio

Para determinar la potencia de transmisión de los beacons y poder continuar con las pruebas con el objetivo de obtener un comportamiento de la señal que permita calcular la distancia, fue necesario ubicar los beacons de manera efectiva, buscando

tener solo un mínimo traslape entre sus áreas de cobertura para evitar interferencias. Se distribuyeron los 10 beacons disponibles para cubrir el edificio de prueba. Se determinó que con una potencia de -81dBm pueden cubrir toda la superficie, aunque con algunos traslapes. La figura 8 muestra el mapa del edificio de prueba y la localización de los beacons con su área de cobertura de 10m para la potencia de transmisión de -81dBm.

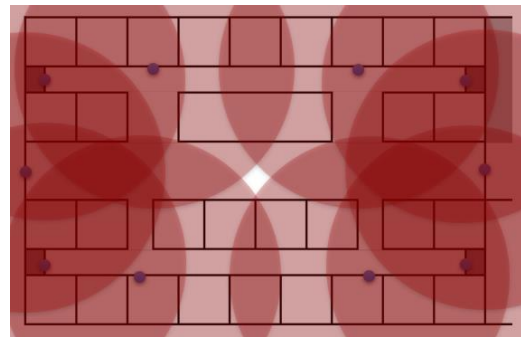


Figura 8.- Localización de los beacons con su área de cobertura a -81dBm.

Se realizó una prueba utilizando los 10 beacons con la intención de obtener una curva característica. Se fijó una potencia de -81dBm en los 10 beacons y se tomaron mediciones por 30 segundos en distancias de 1 a 10m. Los resultados obtenidos fueron similares a los experimentos anteriores. La figura 9 muestra la gráfica con las mediciones promedio del RSSI para los 10 beacons a diferentes distancias. Se puede observar que la tendencia general es la misma, aunque el comportamiento individual es diferente. Se procedió a realizar el ajuste por regresión de una función logarítmica, que es la que mejor se ajusta al comportamiento de la señal RSSI [2], la función resultante fue

$$RSSI = -11.01 \cdot \ln(dist) - 63.98 \quad (3)$$

Despejando la Ec. 3 obtenemos la fórmula para el cálculo de la distancia en base al RSSI medido.

$$dist = e^{\frac{RSSI+63.98}{-10.81}} \quad (4)$$

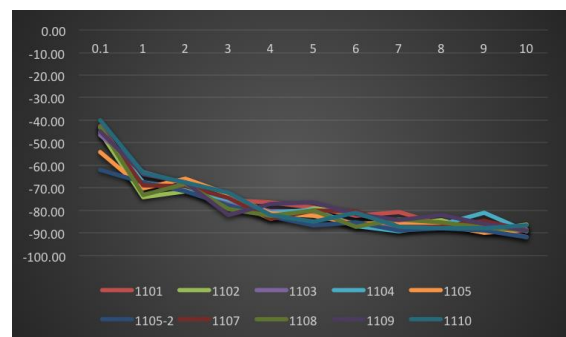


Figura 9.- Gráfica del promedio de RSSI de los 10 beacons a diferentes distancias con una potencia de -81dBm.

Para obtener la posición del dispositivo móvil dentro del edificio de prueba se utilizó la distancia medida a los tres beacons mas cercanos en un algoritmo de trilateración [33] que esta basado en el cálculo del centro radical de tres círculos. Las distancias a los beacons sería el radio de cada círculo. El algoritmo de trilateración no tiene puntos ciegos, ni casos especiales y tiene una alta relación geométrica, por lo que su resultado esta intrínsecamente relacionado con la calidad de las distancias medidas.

#### 4. NAVEGACIÓN EN INTERIORES

La navegación en interiores se refiere a que una vez establecida la posición del dispositivo móvil en el mapa de edificio se pueda utilizar la aplicación para seleccionar un punto destino y que un algoritmo busque las posibles rutas para ayudar al usuario del dispositivo móvil a llegar a su destino.

##### 4.1 Adecuación de la aplicación en Unity para la navegación en interiores

Se creó una nueva escena en la aplicación de Unity y se insertó el modelo del mapa desarrollado en 3Ds Max. Luego se agregó una esfera roja que representa el dispositivo móvil, píldoras azules representando la ubicación de los *beacons* y se agregaron los atributos de colisión y gravedad a los elementos correspondientes. Se creo un script que toma en cuenta los tres beacons más cercanos de acuerdo a la medición del RSSI, se programó la fórmula para el cálculo de distancias y el algoritmo de trilateración. También se creo un script que cambia de posición la esfera roja al existir cambios en los valores RSSI de los beacons.

##### 4.2. Algoritmo de generación de rutas

Para la generación de rutas de un punto a otro se implementó el algoritmo denominado A\* [19]. Este algoritmo define una ruta uniendo múltiples puntos conocidos como “nodos”. Fue necesario colocar los nodos en los puntos del mapa en donde se requiere un cambio de dirección para llegar de un punto a otro. El algoritmo revisa la posición actual y los nodos a los que se puede mover agragando a la ruta aquellos nodos minimizen la distancia al punto destino. Este proceso se repite mientras el dispositivo se encuentre en movimiento o se seleccione un punto destino diferente. La ruta se actualiza continuamente, en caso de que se tome otro trayecto, el sistema ajustará la ruta trazada. En la figura 10 se pueden observar los nodos (puntos verdes) colocados en las esquinas y en las entradas de las puertas de las oficinas, los cuales permiten trazar la ruta (línea morada) a cualquier punto seleccionado en el mapa.

Para mejorar el algoritmo se definieron puntos para restringir el movimiento de la esfera roja y evitar oscilaciones provocadas por la variación en las mediciones de RSSI de los beacons. Estos puntos se colocaron en el centro de los pasillos. Una vez que se calcula la posición con las señales de los beacons se selecciona el punto mas cercano para colocar la esfera roja.

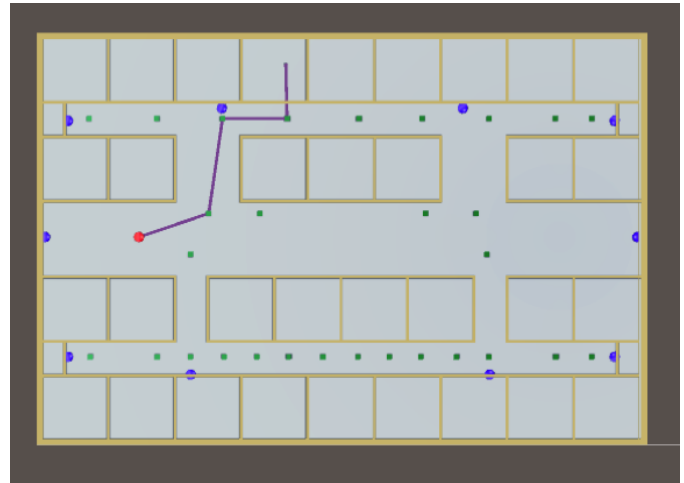


Figura 10.- Algoritmo de navegación implementado en la aplicación desarrollada en Unity.

#### 5. CONCLUSIONES

Se desarrolló una aplicación para dispositivos móviles sobre la plataforma Android que permite navegar dentro de un edificio.

En la primera etapa se desarrolló del mapa del edificio, se probaron diferentes opciones que permitieran realizar al mismo tiempo animaciones, cambios en la orientación y tamaño del gráfico. Se desarrolló el mapa SVG del edificio de prueba y se convirtió a XML para poder mostrarlo en dispositivos Android, pero se complicó el proceso de animación para mostrar la ubicación y movimiento del dispositivo móvil. Se migró al desarrollo del mapa en 3D con 3Ds Max para insertarlo en una aplicación desarrollada en Unity ya que es mas eficiente la programación de las animaciones de los puntos para la posición del dispositivo móvil y la generación de las rutas de navegación.

En la segunda etapa se desarrolló el sistema de posicionamiento en interiores utilizando beacons BLE. Se adquirieron 10 beacons de la marca Kontakt. Se realizaron pruebas para calcular la distancia del beacon al dispositivo móvil utilizando diferentes procedimientos y bajos diferentes condiciones, en exteriores e interiores y con diferentes potencias de transmisión. Ya que los procedimientos para el cálculo de distancia se basan en la medición de la señal RSSI, se caracterizó el comportamiento de los beacons en base a las señales RSSI medidas a diferentes distancias. Se distribuyeron los 10 beacons en el edificio de prueba buscando cubrir toda la superficie. Se utilizaron las medidas de distancia de los tres beacons más cercanos para calcular la posición del dispositivo por medio de trilateración.

Después de realizar pruebas extensivas con lo beacons, tanto en exteriores como en interiores, concluimos que no es posible llegar a una expresión que caracterice de manera correcta el comportamiento de los beacons y que permita obtener una buena aproximación de la medida de distancia del beacon al dispositivo móvil basado en el RSSI. Esto llevó a utilizar los beacons solo como una referencia que permite

establecer grados de proximidad para la localización del dispositivo móvil estableciendo puntos discretos de localización en el mapa del edificio de prueba.

En la tercera etapa se integró al sistema el algoritmo A\* para la determinación de la ruta a seguir por el usuario teniendo definida su ubicación actual y permitiendo la selección del punto destino sobre el mapa en la aplicación del dispositivo móvil desarrollada en Unity.

De esta forma se completó el funcionamiento del sistema de navegación en interiores para dispositivos móviles utilizando beacons BLE para el posicionamiento de dispositivo en el mapa del edificio de prueba. El mapa es desplegado en la aplicación móvil y permite al usuario ver su ubicación y seleccionar un punto de destino, entonces el sistema de navegación mostrará la ruta a seguir.

El trabajo a futuro se centrará en mejorar la exactitud del sistema de localización combinando la tecnología de beacons BLE implementada hasta ahora con señales Wi-Fi y posiblemente otros de los sensores con los que cuentan los dispositivos móviles.

## REFERENCIAS

- [1] E. Kaplan y C. Hegarty, *Understanding GPS: Principles and Applications*. Artech House, 2005.
- [2] [S. Sandve, "Indoor Positioning System Using BLE Beacons", jun. 2016.
- [3] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, y M. Wiseman, "Indoor Location Sensing Using Geo-magnetism", en *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, New York, NY, USA, 2011, pp. 141–154.
- [4] M. A. Al-Ammar *et al.*, "Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms", en *2014 International Conference on Cyberworlds*, 2014, pp. 245–252.
- [5] P. Davidson y R. Piche, "A Survey of Selected Indoor Positioning Methods for Smartphones", *IEEE Commun. Surv. Tutor.*, vol. PP, núm. 99, pp. 1–1, 2016.
- [6] L. Mainetti, L. Patrono, y I. Sergi, "A survey on indoor positioning systems", en *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2014, pp. 111–120.
- [7] "Csci 8715 Pp6: Indoor Positioning Systems Group8 Nuosang Du, Sara Abouelella - Semantic Scholar". [En línea]. Disponible en: /paper/Csci-8715-Pp6-Indoor-Positioning-Systems-Group8-Du-Abouelella/1ec47a66e6f96b32951be200c1d254c585b42f26. [Consultado: 20-mar-2017].
- [8] J. Jiao, F. Li, Z. Deng, y W. Liu, "An indoor positioning method based on wireless signal and image", en *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2016, pp. 656–660.
- [9] Y. Gu, Q. Song, M. Ma, Y. Li, y Z. Zhou, "Using iBeacons for trajectory initialization and calibration in foot-mounted inertial pedestrian positioning systems", en *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–7.
- [10] S. Ye, Y. Chen, y T. Hu, "Evolutionary algorithmic deployment of radio beacons for indoor positioning", en *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 2829–2835.
- [11] S. S. Chawathe, "Beacon Placement for Indoor Localization using Bluetooth", en *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 980–985.
- [12] T. Mori, S. Kajioka, T. Uchiya, I. Takumi, y H. Matsuo, "Experiments of position estimation by BLE beacons on actual situations", en *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, 2015, pp. 683–684.
- [13] "La historia a través de los mapas", *www.nationalgeographic.com.es*. [En línea]. Disponible en: [http://www.nationalgeographic.com.es/historia/grandes-reportajes/la-historia-a-traves-de-los-mapas\\_6991](http://www.nationalgeographic.com.es/historia/grandes-reportajes/la-historia-a-traves-de-los-mapas_6991). [Consultado: 20-mar-2017].
- [14] "Projects | Humanitarian OpenStreetMap Team". [En línea]. Disponible en: <https://www.hotosm.org/projects>. [Consultado: 20-mar-2017].
- [15] H. Li y X. Gong, "An approach to integrate outdoor and indoor maps for books navigation on the intelligent mobile device", en *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 460–465.
- [16] T. Calle-Jimenez, S. Sanchez-Gordon, C. Rivera-Pastrano, y S. Luján-Mora, "A practical example of a collaborative learning experience for engineering students: How to build accesible indoor maps", en *2015 International Conference on Interactive Collaborative and Blended Learning (ICBL)*, 2015, pp. 35–43.
- [17] W. Jeamwathanachai, M. Wald, y G. Wills, "Map data representation for indoor navigation a design framework towards a construction of indoor map", en *2016 International Conference on Information Society (i-Society)*, 2016, pp. 91–96.
- [18] "OGC® IndoorGML | OGC". [En línea]. Disponible en: <http://www.opengeospatial.org/standards/indoorgml#overview>. [Consultado: 24-abr-2017].
- [19] O. M. Elfadil, Y. M. Alkasim, y E. B. Abbas, "Indoor navigation algorithm for mobile robot using wireless sensor networks", en *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, 2017, pp. 1–5.
- [20] P. Wilk y J. Karciarz, "Optimization of map matching algorithms for indoor navigation in shopping malls", en *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 661–669.
- [21] A. Tyrberg, "Senion | Indoor Positioning System | Indoor navigation", *Senion | Indoor Positioning System*. [En línea]. Disponible en: <https://senion.com/>. [Consultado: 21-mar-2017].
- [22] "Locate, Track and Monitor people and assets in the real world", *Accuware*. [En línea]. Disponible en: <https://www.accuware.com/>. [Consultado: 21-mar-2017].
- [23] "Indoor Navigation, Indoor Positioning and Indoor Tracking by infsoft". [En línea]. Disponible en: <https://www.infsoft.com/>. [Consultado: 21-mar-2017].
- [24] "IndoorAtlas". [En línea]. Disponible en: <http://www.indooratlas.com/>. [Consultado: 21-mar-2017].
- [25] "About - Google Maps". [En línea]. Disponible en: <https://www.google.com/maps/about/partners/indoormaps/>. [Consultado: 08-ago-2018].
- [26] "Indoor Mapping - OpenStreetMap Wiki". [En línea]. Disponible en: [https://wiki.openstreetmap.org/wiki/Indoor\\_Mapping](https://wiki.openstreetmap.org/wiki/Indoor_Mapping). [Consultado: 08-ago-2018].
- [27] "Draw Freely | Inkscape". [En línea]. Disponible en: <https://inkscape.org/en/>. [Consultado: 08-feb-2018].
- [28] "About CodePen". [En línea]. Disponible en: <https://codepen.io/about/>. [Consultado: 08-feb-2018].
- [29] F. Dwiyasa y M. H. Lim, "A survey of problems and approaches in wireless-based indoor positioning", en *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–7.
- [30] "Estimote". [En línea]. Disponible en: <http://estimote.com/>. [Consultado: 21-mar-2017].
- [31] "Kontakt.io Store". [En línea]. Disponible en: <https://store.kontakt.io/>. [Consultado: 21-mar-2017].
- [32] Q. Dong y W. Dargie, "Evaluation of the reliability of RSSI for indoor localization", en *2012 International Conference on Wireless Communications in Underground and Confined Areas*, 2012, pp. 1–6.
- [33] V. Pierlot, M. Urbin-Choffray, y M. V. Droogenbroeck, "A New Three Object Triangulation Algorithm Based on the Power Center of Three Circles", en *Research and Education in Robotics - EUROBOT 2011*, 2011, pp. 248–262.