



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE COMPUTACIÓN**

**IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL EN
ANDROID UTILIZANDO TECNOLOGÍA BEACON PARA UNA
MEJOR EXPERIENCIA DEL USUARIO DENTRO DEL
JARDIN BOTÁNICO DE QUITO**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Ciencias de la Computación

**AUTORES: LEÓNIDAS ANDRÉS GUEVARA PADILLA
STALIN MAURICIO VEGA CASA**

TUTOR: JOSÉ LUIS AGUAYO MORALES

Quito – Ecuador

2023



CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Leonidas Andres Guevara Padilla con documento de identificación N*1753383775 y Stalin Mauricio Vega Casa con documento de identificación N*1724583446 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 27 de febrero del año 2023

Atentamente,



Leonidas Andres Guevara Padilla
1753383775



Stalin Mauricio Vega Casa
1724583446

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Leonidas Andres Guevara Padilla con documento de identificación N*1753383775 y Stalin Mauricio Vega Casa con documento de identificación N*1724583446, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Implementación de una Aplicación Móvil en Android Utilizando Tecnología Beacon para una Mejor Experiencia del Usuario Dentro del Jardín Botánico de Quito”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final con formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 27 de febrero del año 2023

Atentamente,



Leonidas Andres Guevara Padilla
1753383775



Stalin Mauricio Vega Casa
1724583446

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, José Luis Aguayo Morales con documento de identificación N* 1709562597, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL EN ANDROID UTILIZANDO TECNOLOGÍA BEACON PARA UNA MEJOR EXPERIENCIA DEL USUARIO DENTRO DEL JARDÍN BOTÁNICO DE QUITO, realizado por Leonidas Andres Guevara Padilla con documento de identificación N*1753383775 y Stalin Mauricio Vega Casa con documento de identificación N*1724583446, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 27 de febrero del año 2023

Atentamente,



Ing Jose Luis Aguayo Morales MSc
170956259



DEDICATORIA

Quiero empezar agradeciendo a Dios, por ser quien me acompaña en cada instante de mi vida y mi camino. Agradecer a mis padres por darme todo el apoyo, la fortaleza para no rendirme y seguir avanzo en este que es sueño. A mis herman@s que siempre han creído en mí y han estado a mi lado en las buenas y en las malos, a mi familia y amigos que me han brindado su amistad su apoyo y me han motivado a lo largo de mis estudios en esta prestigiosa universidad.

INDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1.....	3
ESTADO ACTUAL DEL PROBLEMA DE ESTUDIO	3
OBJETIVOS	4
Objetivo General	4
Objetivos Específicos	4
ALCANCE Y LIMITACIONES	4
CAPÍTULO 2.....	5
MARCO METODOLÓGICO Y TEÓRICO	5
MARCO METODOLÓGICO.....	5
MARCO TEÓRICO.....	6
ESTADO DE ARTE	6
Beacon.....	11
Funcionamiento	11
Protocolo	12
Rango.....	14
Ventajas y desventajas	14
Android.....	15
iOS	16

Android Studio	16
Ventajas y Desventajas de Android Studio.....	19
WiFi.....	19
Radiofrecuencia (RFID).....	20
CAPÍTULO 3.....	21
Requerimiento y diseño	21
Requerimientos Funcionales	21
Análisis e Interpretación de los Resultados	27
Diseño, desarrollo e implementación	27
Evidencia del funcionamiento de la aplicación	43
Interfaz del Usuario	43
Pantalla de Ruta	47
CONCLUSIONES	48
RECOMENDACIONES	49
Anexos	50
Referencias.....	53

ÍNDICE DE FIGURAS

FIGURA 1	11
FIGURA 2	12
FIGURA 3	13
FIGURA 4	13
FIGURA 5	14
FIGURA 6	15
FIGURA 7	16
FIGURA 8	17
FIGURA 9	18
FIGURA 10	19
FIGURA 11	19
FIGURA 12	20

RESUMEN

El Jardín Botánico de Quito desea difundir información a los usuarios y por ello se ha propuesto desarrollar una tecnología de servicio denominada beacons que ayuda a identificar beacons que funcionan al recibir datos transmitidos desde un dispositivo bluetooth, para brindar una mejor experiencia. El objetivo del proyecto es implementar una aplicación móvil mediante tecnología beacon para una mejor experiencia de usuario dentro de las instalaciones del Jardín Botánico de Quito, derivando una propuesta de modelo de aplicación informática que cumpla con los requisitos para satisfacer las necesidades de los visitantes, y evaluarla utilizando métricas de software. Luego de una investigación y análisis se desarrolló el proyecto con el framework Scrum Agile, ya que es una metodología que está en tendencia y a la vez es moderna y cada vez es más aceptada en la comunidad global de desarrollo de software. El resultado arrojó que el desarrollo de la aplicación móvil y la implementación de las diferentes herramientas como el iBeacon permite tener una conexión en interiores teniendo mayor precisión en evidencia que la Aplicación Móvil cumple con las expectativas esperadas debido a la metodología scrum lo cual ayudó a tener avances organizados. en las diferentes etapas del desarrollo del proyecto. En conclusión, la aplicación móvil tendrá un efecto social y tecnológico, ya que el documento de investigación y aplicación explica que existen nuevas tecnologías que tienen mucha demanda en el mercado actual y que no están siendo utilizadas en nuestro medio.

Palabras Clave: Tecnología beacon, Aplicaciones móviles, Dispositivos móviles, Bluetooth.

ABSTRACT

The Quito Botanical Garden wishes to disseminate information to users and for this reason it has proposed to develop a service technology called beacons that helps identify beacons that work by receiving data transmitted from a Bluetooth device, to provide a better experience. The objective of the project is to implement a mobile application through beacon technology for a better user experience within the facilities of the Quito Botanical Garden, deriving a proposal for a computer application model that meets the requirements to meet the needs of visitors, and evaluate it using software metrics. After an investigation and analysis, the project was developed with the Scrum Agile framework, since it is a methodology that is in trend and at the same time is modern and is increasingly accepted in the global software development community. The result showed that the development of the mobile application and the implementation of the different tools such as the iBeacon allows to have a connection indoors, having greater precision in evidence that the Mobile Application meets the expected expectations due to the scrum methodology which helped to have organized advances. at different stages of project development. In conclusion, the mobile application will have a social and technological effect, since the investigative and application document explains that there are new technologies that are in great demand in the current market and which are not being used in our environment.

Keywords: Beacon technology, Mobile applications, Mobile devices, Bluetooth.

INTRODUCCIÓN

Antecedentes Las nuevas tecnologías están dando un cambio en la sociedad y evolucionando hacia un futuro conectado de la comunicación y el desarrollo de dispositivos electrónicos que están constantemente "online" a través de aplicaciones, sensores, teléfonos móviles, ordenadores. Pueden contribuir para ser parte de esta red insondable y pueden activarse o manipularse en cualquier momento y en cualquier lugar. Estas tecnologías están entrando cada vez más en campos y sus beneficios están siendo probados.

Problema En el Jardín Botánico (JB) de Quito se ha experimentado dificultad al momento de distribuir la información a los usuarios, para ello se va a desarrollar una tecnología de servicio llamada beacon (balizas) que ayuda a la identificación de balizas o un servicio basado en la localización que funciona mediante la recepción de datos transmitidos desde un dispositivo Bluetooth. Existen varios tipos de aplicaciones que sirven para dispositivos móviles por los cuales se pueden acceder a su posicionamiento, sin embargo, su rendimiento en interiores puede llegar a ser insatisfactorio. (Jae-Hwan & Lee, 2018). Es importante mencionar que la forma en la cual se distribuye la información dentro de las instalaciones del JB, se la realiza mediante parlantes o elementos de señalización los cuales por sus propias limitaciones de construcción no permiten brindarle información necesaria al visitante, así como enriquecer su experiencia mediante elementos multimedia o contenido digital (Atherton, 2019).

Justificación Este proyecto ayudara al JB a aumentar la afluencia de visitantes, mejorando la experiencia del usuario buscando aumentar su interés e ir promocionando el sector turístico, ya que se mejora la percepción de satisfacción, al aumentar la información y que sea de mayor calidad y cantidad que recibe en cada recorrido. Al aumentar la cantidad de visitantes de igual forma, aumenta la cantidad de recaudación de ingresos que obtendrá el JB con el fin de preservar su funcionamiento en el tiempo. La obligación para que los visitantes tengan una mejor experiencia y disminuir el gasto en elementos físicos, con el fin de entregar más información sobre la diversidad de vegetación que se encuentra en el sitio y además informar sobre eventos que se promocionan dentro del JB. La visión del JB incentivar al cuidado de la naturaleza mediante el uso de la tecnología.

Objetivo General

Implementar una aplicación móvil en Android mediante tecnología beacon para el JB, la cual se plantea diseñar una propuesta de modelo de aplicación móvil informativa mediante el uso de tecnología beacon.

Objetivos Específicos

- Diseñar una propuesta de modelo de aplicación móvil informativa mediante el uso de tecnología beacon.
- Determinar los requerimientos con la finalidad de satisfacer la necesidad del JB.
- Evaluar la aplicación utilizando métricas de software y cálculo computacional.

Metodología La metodología elegida para el desarrollo de este proyecto se ajusta al marco Scrum Agile, que es una tendencia moderna y cada vez más aceptada en la comunidad global de desarrollo de software.

CAPÍTULO 1

ESTADO ACTUAL DEL PROBLEMA DE ESTUDIO

El JB tiene dificultades para difundir información a los usuarios y por esta razón se está desarrollando una tecnología de servicio llamada beacons que ayuda a identificar balizas que funciona mediante la recepción de datos transmitidos desde un dispositivo Bluetooth. Existe una variedad de aplicaciones de dispositivos móviles pero su rendimiento en interiores puede no ser satisfactorio, por lo que se han explorado nuevas técnicas de posicionamiento en interiores para lograr una mayor eficacia. (Jae-Hwan & Lee, 2018). Cabe mencionar que la distribución de información en las instalaciones del JB se realiza a través de parlantes o elementos de señalización, los cuales por sus limitaciones estructurales no permiten que los visitantes obtengan la información necesaria, por lo cual se implementaran elementos multimedia y contenido digital para enriquecer la experiencia de los usuarios (Atherton, 2019).

La tecnología beacon está siendo utilizada en museos y en marketing a nivel mundial ya que está relacionada con Big-data, gestión y análisis de datos que muchas empresas utilizan para interactuar mejor con sus clientes, también ofrece grandes beneficios de geolocalización en los espacios cerrados permitiendo personalizar al máximo las comunicaciones con el usuario e incluso realizar tracking de comportamientos para conocer mejor las necesidades y preferencias los usuarios (Cleve & Mulzer, 2020). Se han realizados estudios sobre el uso del beacon con una orientación en el marketing, ya que es una estrategia de mercado que utiliza la tecnología de dispositivos móviles para llegar a los usuarios de manera más directa, que les permite enviar mensajes personalizados dentro del radio de la ubicación (Perera, Aghaee, Faragher, Harle, & Blackwell, 2018). Actualmente en Ecuador el JB, se ha propuesto implementar nuevas

herramientas tecnológicas basadas en beacon que permitan aumentar las visitas que recibe diariamente, sin embargo lo más importante es mejorar la experiencia de los usuarios al momento de realizar los diversos recorridos que ofrece (Atherton, 2019).

OBJETIVOS

Objetivo General

Implementar una aplicación móvil en Android mediante tecnología beacon para mejorar la experiencia del usuario dentro de las instalaciones del Jardín Botánico de Quito.

Objetivos Específicos

- Diseñar una propuesta de modelo de aplicación móvil informativa mediante el uso de tecnología beacon.
- Determinar los requerimientos con la finalidad de satisfacer la necesidad del Jardín Botánico.
- Evaluar la aplicación utilizando métricas de software y cálculo computacional.

ALCANCE Y LIMITACIONES

Alcance

El presente proyecto propone analizar, investigar y recopilar información para el desarrollo de un modelo de aplicación móvil el cual va a funcionar en el JB y brindar al visitante una mejor experiencia, cumpliendo las métricas de software para garantizar la calidad de este, también busca la forma de contribuir y compartir la información más personalizada para así ofrecer más oportunidades y aportar en el aprendizaje.

Limitaciones

- Android es un sistema operativo en el cual se va a realizar la Aplicación móvil.
- Funcionalidad incorrecta de las balizas instaladas en el entorno del usuario.

CAPÍTULO 2

MARCO METODOLÓGICO Y TEÓRICO

MARCO METODOLÓGICO

El método elegido para el desarrollo de este proyecto se ajusta al marco Scrum Agile, que es una tendencia moderna y cada vez más aceptada en la comunidad global de desarrollo de software. Esta metodología se lo dividió en 5 fases para tener un mejor orden y poder culminar la aplicación:

El enfoque se divide en 5 fases para completar el desarrollo de la aplicación:

Escena1

Inicio del Proyecto: En esta fase se definen las historias inconclusas, donde se definen los requisitos en función de los criterios de aceptación de las actividades generadas luego de entender los requisitos del usuario final.

Escena2

Sprint Planning: Esta fase se trata de construir la definición hasta obtener el producto que el cliente necesita, para evitar errores dentro de la historia y mostrar lo más claramente posible lo que los desarrolladores deben hacer.

Escena3

Daily Scrum: esta fase incluye se realizaran reuniones periódicamente, se auto gestiona y proporciona retroalimentación en función de las actividades en curso para planificar las actividades de la semana.

Escena4

Sprint Review: esta fase se realiza al final de cada sprint para discutir lo que se ha

hecho, las dificultades encontradas y cómo superarlas para mantener la alta calidad del producto en desarrollo y aumentar la productividad.

Escena5

Demostraciones: las demostraciones son esencialmente "hora del espectáculo"; para proyectos ágiles. En ausencia de modelos y especificaciones detallados, las demostraciones son donde el equipo técnico determina si el producto entregado cumple con las expectativas.

MARCO TEÓRICO

ESTADO DE ARTE

Este capítulo informa sobre las investigaciones encontradas del tema de estudio.

Bocado, (Boccardo & Rie, 2016) en su investigación de maestría de 2016, indico que existen diseños para el desarrollo de aplicaciones móviles que están vinculadas con los dispositivos beacon. La investigación tuvo como objetivo principal brindar información contextual al usuario, brindando a los desarrolladores herramientas útiles para incorporar esta información en la aplicación. La información obtenida de los usuarios es guardada para tener en cuenta las necesidades más tradicionales que existen en cuanto a las actividades de localización. Como resultado, Bocado presento públicamente unas aplicaciones móviles que utilizan diseños con patrones similares. Sin embargo, sin ningún tipo de métrica que explique claramente el beneficio de utilizar esos diseños (Boccardo & Rie, 2016). En resumen, concluimos que el desarrollo de la aplicación de Bocado fue bastante simple para favorecer a los beacon y así su programación sea más sencilla. El diseño era muy simple y no respondía con la información adecuada para esto, la implementación tomó en cuenta la distribución de entidades y la lógica proporcionada por Bocado, pero no encontró una forma práctica de reutilizar dicha API y creó un nuevo código para este artículo.

Caballero de Tineo (Tineo, 2016), en su trabajo de final de masterado del 2016, propone una solución para la industria turística basada en Beacons y objetos turísticos de comunicación Bluetooth Low Energy (BLE) y proporciona servicios de información que aportan valor añadido a las actividades de turismo cultural. El motivo del estudio es la problemática de los turistas que afirman que no existe información clara y oportuna que les permita aprovechar la riqueza cultural existente. Para ello, Caballero de Tineo marcó el flujo de turistas (eventos y atractivos turísticos), lo que se tradujo en la instalación de una baliza en el entorno exterior e interior. La investigación de Caballero de Tineo (Tineo, 2016) examina principalmente la "experiencia de proximidad" porque los conceptos de "lejos" y "cerca" se utilizan para determinar si un turista pasa por un atractivo turístico lejos o cerca. Un turista clasificado recibe información sobre los atractivos destinos que se pueden visitar. Los turistas llamados clasificados o "cerca" pueden ser considerados ayudantes siempre que se encuentren a menos de 20 metros de la baliza durante al menos 5 minutos. El visitante recibe información sobre el lugar turístico mencionado. Este proyecto utiliza el formato Eddystone para implantar 45 Beacons, permitiendo la participación de visitantes con teléfonos móviles iOS y Android (Peña Parra, 2017). Caballero de Tineo hace referencia a la intensidad con la que un beacon envía un mensaje en un intervalo de tiempo a un número de visitantes. Para lograr un mayor aforo de visitantes se necesita que los beacon tengan más potencia y menor tiempos de intervalos de transmisión para recibir mensajes a una distancia de 70 metros. Sin embargo, estas configuraciones que se pueden realizar llevan a un consumo de energía más alto. En conclusión, se demuestra que los beacon son de utilidad ya que tienes varios tipos de

configuraciones para permitir distintos tipos de estrategias de control para ver posibles usuarios potenciales y que información es la que se podría enviar (Tineo, 2016).

En un artículo de 2016 titulado "S-Beacon: solución de baliza BLE de próxima generación para una personalización mejorada" (Byun, Cho, & Hong, 2016), los autores señalaron que las balizas de próxima generación no solo se comunicarán con smartphones, sino también con varios portátiles. Los dispositivos suelen tener más información sobre sus usuarios, como: frecuencia cardíaca, temperatura, etc. Esto permite que se evalúen las preferencias del usuario que ayudara a tener una mejora en los servicios a ofrecer. Además, existen más mejoras sugeridas como es la comunicación bidireccional entre Beacon y el dispositivo móvil para una mejor precisión de ubicación. Los autores son miembros de una empresa global de electrónica, por lo que sus recomendaciones se basan en mejorar las balizas a nivel de componentes físicos y de recursos. Estos beacons con excelentes recursos no se encuentran comercialmente en las búsquedas actuales (Byun, Cho, & Hong, 2016).

Museos un contexto general

De hecho, la tecnología se utiliza más en lugares abiertos al público y privados. El beneficio que se va a obtener al usar los sistemas inteligentes es para ir mejorando las experiencias de los usuarios e ir promocionando dichos lugares turísticos (Piccialli & Chianese, 2017). Para los Museos "Los museos son una institución que no tienen ánimo de lucro que están al servicio de la sociedad favoreciendo al desarrollo que está abierto al visitante para conocer, conservar, investigar, difundir y exhibir con fines educativos" (Sharif-Askari & Abu-Hijleh, 2018).

Los museos tienen como finalidad 3 labores esenciales: los cuales comunicar, informar y apoyar al sistema de enseñanza. Aun cuando, su trabajo se extiende más de estos tres puntos principales, ya que sirven como parte importante dinámica cultural que contribuye al desarrollo de la identidad. Desde el punto social, a las personas les interesa más conservar el patrimonio cultural por motivos que nos representan culturalmente y es algo que se debe transmitir a las futuras generaciones para que tengas una identidad cultural. Muchos de los turistas que vienen a visitar nuestros museos se dan cuenta que son centros importantes de atracción para comprender las culturas e identidades locales (Correa & Jiménez, 2011). Los académicos han enfatizado repetidamente la necesidad de que los museos proporcionen programas educativos y espacios de aprendizaje (Allen, 2004). De esta manera, se puede crear un espacio equilibrado entre los métodos de enseñanza puramente teóricos y los métodos de enseñanza generados en el campo.

Museos con nuevas tecnologías y aplicación

Las nuevas tecnologías han ido evolucionando y popularizándose por ellos varios investigadores se han propuesto evaluar cómo usar estas nuevas herramientas con el fin de brindar una mejor experiencia y una visita más atractiva a los visitantes (Ardito, Buono, Desolda, & Matera, 2018). En los últimos años habido un mayor interés en las tecnologías para museos llamándolos museos inteligentes que son capaces de dar recursos ya sea infraestructura, tecnología, comprometiendo a continuar su desarrollo para mejorar la experiencia de los visitantes y ser más sostenible (EVE, 2018-b). El uso de las nuevas tecnologías en los museos no es casual. En cambio, existe un

conjunto de precedentes que motivan a los administradores de museos a implementar dichas herramientas.

- Primero, no es posible mantener guías humanos durante todo el día debido a los recursos humanos limitados en los museos. Esto conducirá a mayores costos de personal, que el museo no puede permitirse en muchos casos.
- Segundo, las exposiciones de los museos suelen presentar grandes colecciones que los visitantes no pueden ver en un tiempo razonable. Como resultado, las personas que van a al museo no puede visualizar toda la exposición y se focalizan en subconjuntos de exhibiciones (Kirchberg & Trondle, 2012).
- Tercero, los museos tienen exhibiciones complejas con un valor emocional ya sea social, científico tiene un gran significado, que no son fáciles de entender para los turistas, y por esta razón muchos de los visitantes no se toman el tiempo suficiente para apreciar lo disponible (Sparacino, 2002).

Por los problemas antes mencionados los museos han propuesto realizar cambios en los centros agregando nuevas herramientas para que crezca la relación entre los visitantes y el museo. Abad dijo que el valor que agregaban los visitantes a los museos es irremplazable, por ello son las herramientas que vamos a utilizar las que se deben moldear a los perfiles de cada usuario que este dentro del museo (Juchnowicz & Abad, 2011).

Con los smartphones, la mayoría de los museos ha optado por adquirir nuevas tecnologías como son las aplicaciones o pantallas táctiles, pero todo tiene una razón la cual fue crear espacios de entretenimiento con realidad virtual, realidad aumentada, juegos (Angelaccio, Basili, & Liguori, 2012).

Beacon

Bluetooth Low Energy (BLE) es una característica del Bluetooth estándar, tiene una versión 4.0 la cual fue la primera en Bluetooth Smart, esa diseñado esencialmente para tener conexiones entre dispositivos y tiene una alta transmisión de datos.

Está distribuida en distintas capas, como se muestra a continuación en la figura1.

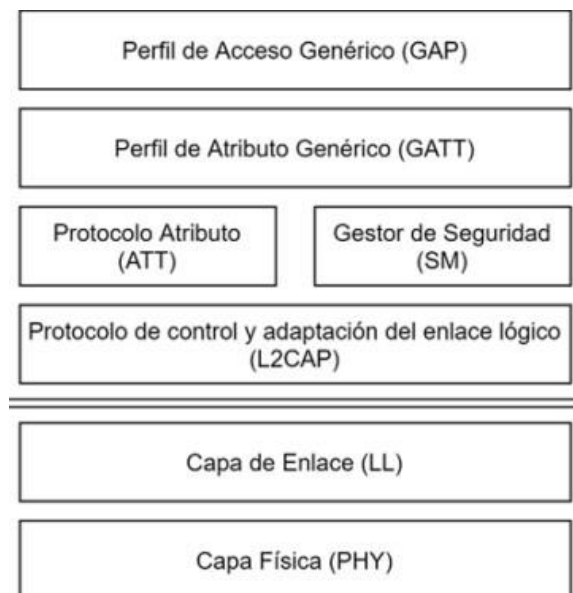


Figura 1. Capas del Beacon

Funcionamiento

BLE funciona en la banda de 2,4 GHz con 40 canales de radiofrecuencia espaciados a intervalos de 2 MHz y, como es común que varios dispositivos ocupen los mismos canales de frecuencia podría ocurrir una interferencia. Aquí es donde aparece el Bluetooth 5.0 para dar una solución con el salto de frecuencia lo que favorece a BLE con mayor compatibilidad con otros dispositivos y la transmisión de los datos aumentaría hasta 2Mbps.

Los canales BLE se dividen en dos categorías:

- Publicidad (canales 37, 38 y 39)
- Transmisión de datos. (canales 0 a 36).

Los datos pasan por un canal de transmisión y se utilizan para la comunicación bidireccional entre los dispositivos que se conectan en Bluetooth, mientras que el canal de publicidad se utiliza para la conexión y transmisión de difusión. El beneficio de los canales publicitarios es que su estructura ayuda a no tener interferencias en los canales Wi-Fi 1, 6 y 11 (estándar 802.11).



Figura 2. Funcionamiento Beacon

Protocolo

Beacon tiene dos protocolos: Ibeacon y Eddystone. Ambos son similares a los paquetes de software de transmisión apropiados y las cargas válidas, pero son ligeramente diferentes para identificar la distribución de bytes (Caballero, 2016).

iBeacon

El protocolo de comunicación desarrollado en primera instancia particularmente para la tecnología beacon fue creada por Apple para indagar nuevas tecnologías en el ambiente de localización empleando BLE. iBeacon es un protocolo diseñado para relacionarse con dispositivos de la marca Apple como: iPhone y iPad, pero se puede

utilizar con otros dispositivos que ejecutan el sistema operativo Android, como se muestra en la figura3:

- **UUID:** Es un identificador único que consiste en un marco de 16 bytes destinado a ser reconocido por la aplicación final.
- **MAJOR:** Una trama de 2 bytes que identifica un grupo de balizas dentro de una empresa.
- **MINOR:** Un marco de 2 bytes que identifica una baliza específica dentro del grupo de balizas de una empresa (Gilchrist, 2014).



Figura 3. Protocolo

Los protocolos Eddystone son utilizados para la comunicación y fue desarrollado por Google, este tipo de protocolo son realizados para ser utilizado Android e iOS, Eddystone que tiene diferentes tipos de marcos que se pueden usar individualmente o en conjunto (SiIcon, 2017):

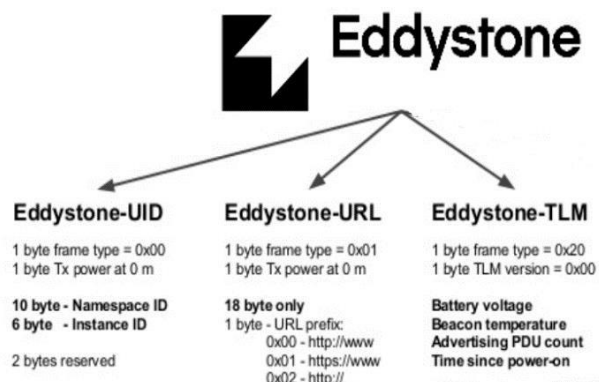


Figura 4. Protocolo Eddystone

Rango

La mayoría de las balizas del mercado tienen actualmente un alcance de 70m, como se muestra en la figura 4.

Las aplicaciones de los *beacons* tienen tres rangos de distancia:

- Lejos: está configurado para cuando detecte un beacon lance una acción.
- Cerca: está configurado para cuando detecte un beacon cerca y lance una acción.
- Inmediato: está configurado para cuando detecte un beacon muy cerca lance una acción.

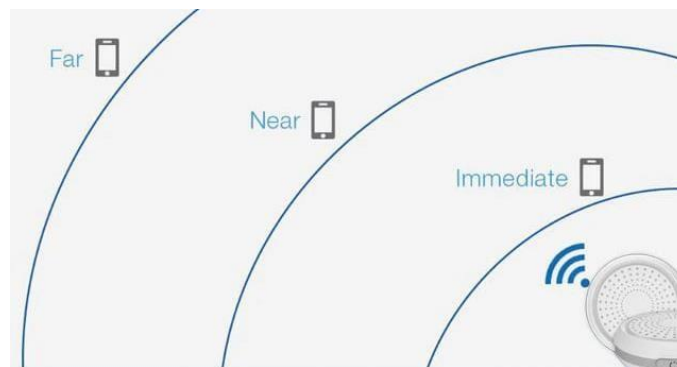


Figura 5. Rango de Alcance

Ventajas y desventajas

Las beacons son una tecnología que se puede utilizar en una variedad de situaciones, pero como todas las tecnologías, son buenas y malas, las cuales se describen (Padron, 2016).

Las principales ventajas del beacon son:

- El Bluetooth tiende a consumir una menor cantidad de energía a comparación de la tecnología GPS.

- El beacon tiene una mayor precisión en interiores ya sea de edificios u otros lugares a diferencia del GPS.

Las desventajas se destacan lo siguiente:

- El rendimiento depende del uso final.
- Consumo de energía cuando Bluetooth está activo.

Android



Figura 6. Logo Android

Todo dispositivo móvil necesita de un sistema operativo para eso creo Android, que fue adquirida por Google en 2005. En la actualidad, la empresa continúa el desarrollo del sistema operativo en forma de código libre. El kernel de Android se basa en Linux y el software gratuito está orientado a Unix. Este es un entorno de desarrollo para implementar aplicaciones móviles y no solo es uno de los editores de código abierto más potentes sino se caracteriza por aumentar la productividad de desarrollo (Molina, Sandoval, & Toledo, 2011).

Algunas de estas características son:

Herramienta para identificar problemas de versiones.

- Compatible con C++.
- Integre con plantillas de código o GitHub para crear una funcionalidad común en todas las aplicaciones.

iOS

Es un sistema que fue desarrollado Apple, principalmente fue diseñado para el uso de los iPhone, pero desde entonces se ha utilizado en los dispositivos más nuevos. IOS tiene una gran desventaja la cual consiste en que no se puede instalar en otros dispositivos que no sean de Apple (Pedrozo, 2012). La interfaz consiste en la manipulación directa mediante controles deslizantes. La interacción con el sistema operativo implica gestos como arrastrar, tocar y pellizcar, que se definen de manera diferente según el contexto de la interfaz. Tiene cuatro capas de abstracción (Pedrozo, 2012) como se puede observar a continuación en la figura 5.

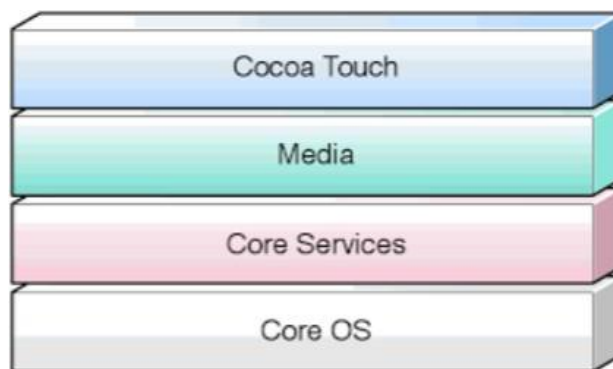


Figura 7. Capas de IOS

Android Studio

Android Studio (AS) es el entorno de desarrollo integrado (IDE) mediante el cual se puede ejecutar aplicaciones de Android basadas en IntelliJ IDEA. Conjuntamente con el editor de código y las herramientas para desarrolladores de IntelliJ, Android Studio contiene diversas características para elevar la productividad al idear aplicaciones de Android, como se muestra en la figura 6.



Figura 8. Características de Android Studio

Android Studio es una plataforma la cual cumple con todos lo necesario para el desarrollo de aplicaciones las cuales se pueden adaptar a una amplia gama de dispositivos móviles, cada aplicación se puede probar usando emuladores de dispositivos que nos permiten usar la aplicación (Studio A. , 2017).

AS tiene componentes que ayudan y hacen más dinámico el desarrollo de aplicaciones, consta de un intuitivo diseño que básicamente consiste en arrastrar y soltar, usar Gradle consume un mínimo de recursos, el desarrollo es más sencillo y el estilo de trabajo más ágil (Studio A. , 2014). Tiene un emulador que puede ejecutar la aplicación rápidamente, también le permite construir nuestros prototipos y puede probarlos en cada dispositivo con una amplia gama de configuraciones de dispositivos (Studio A. , 2017).

Puede usar APK Analyzer, que analiza todo el contenido del APK, revela el tamaño de los componentes y, por lo tanto, puede disminuir el peso del APK. También admite agregar o editar proyectos en lenguajes C y C++, y también analiza su sintaxis

y conversión, permite la depuración simultánea de Java y los lenguajes anteriores (FAQ, 2016).

Principales características de AS, se muestran en la figura 9:

- ❖ Soporte para programar aplicaciones para Android Wear (sistema operativo para dispositivos corporales como por ejemplo un reloj).
- ❖ Herramientas Lint (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.
- ❖ Permite la importación de proyectos realizados en el entorno Eclipse, que a diferencia de Android Studio (Gradle) utiliza ANT.
- ❖ Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- ❖ Vista previa en diferentes dispositivos y resoluciones.
- ❖ Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo xml.

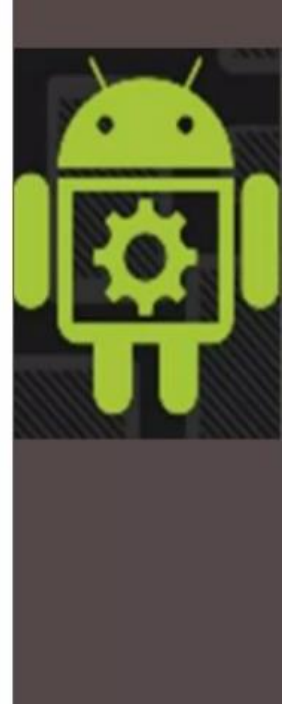


Figura 9. Características de Android Studio

Ventajas y Desventajas de Android Studio

Ventajas	Desventajas
Tiene fluidez de trabajo al ejecutar el código	
El editor te ayuda a escribir un código mejor , trabajar mas rápido y te da sugerencias.	
Tiene un emulador ,con muchas características ,por ejemplo ; pueden simular la ubicación GPS.	Tiene Requisitos de instalación altos.
La estructura de proyectos y las compilaciones te brindan la flexibilidad que necesitas para generar APK para toda clase de dispositivos.	Gasta mucha batería.
Android Studio es completamente compatible con C y C++	

Figura 10. Ventajas y Desventajas

WiFi

WiFi es una red inalámbrica de corto alcance ampliamente necesaria para conexiones a Internet. Ocupa ondas para enviar información y requiere de un transmisor inalámbrico generalmente un enrutador y un adaptador para leer y enviar información a través de la red. Los estándares WiFi han cambiado a lo largo de los años, siendo 802.11n uno de los más utilizados.

Estándar	Veloc. Máx	Banda Frecuencia	Radio cobertura interior	Radio cobertura exterior
802.11a	54 Mbps	5 GHz	85 m	185 m
802.11b	11 Mbps	2.4 GHz	50 m	140 m
802.11g	54 Mbps	2.4 GHz	65 m	150 m
802.11n	>100 Mbps	2.4 GHz ó 5 GHz	90 m	250 m
802.11ac	>1 Gbps	5 GHz	90 m	250 m

Figura 11. Estándares

Radiofrecuencia (RFID)

La identificación por radiofrecuencia (RFID) es considerada la primera tecnología para

la comunicación. Un RFID está formado de dos partes, primero el transpondedor el cual se encarga de adjuntar e identificar al objeto, como segunda parte importante está el lector el cual permite leer o en ciertos casos leer y escribir sobre el objeto.

El rango de distancia se muestra en la siguiente figura 12.

Banda de Frecuencias	Descripción	Rango
125 kHz - 134 kHz	LF (Baja Frecuencia)	Hasta 45 cm.
13,553 MHz - 13,567 MHz	HF (Alta Frecuencia)	De 1 a 3 m.
400 MHz - 1.000 MHz	UHF (Ultra Alta Frecuencia)	De 3 a 10 m.
2,45 GHz - 5,4 GHz	Microondas	Más de 10 m.

Figura 12. Rango de Distancia RFID

La distancia de trabajo de las etiquetas pasivas que operan a bajas frecuencias es de hasta 15 cm y UHF de hasta 15 m. La etiqueta activa también depende de la frecuencia de trabajo. Sin embargo, si el sistema está bien configurado, el alcance máximo puede ser de hasta 100 metros.

CAPÍTULO 3

Requerimiento y diseño

Requerimientos Funcionales

-El permiso del Bluetooth es necesario para la conexión con el dispositivo beacon ya que por esa forma es la comunicación.

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Permiso de Bluetooth.
Características:	La aplicación requiere del permiso del bluetooth.
Descripción del requerimiento:	El permiso de bluetooth es necesario para la funcionalidad y conexión.
Requerimiento NO funcional:	RNF02
Prioridad del requerimiento:	Alta

-La conexión con Beacon es importante para la interacción del usuario con la app.

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Asociar un Beacon a un sitio
Características:	El beacon emerge una notificación cuando estas cerca del mismo.
Descripción del requerimiento:	La aplicación nos muestra información acorde al beacon y su sitio asociado
Requerimiento NO funcional:	
Prioridad del requerimiento:	Alta

-Asociar un Beacon a un sitio es necesario para obtener información sobre el sitio ya que se vincula toda información a la baliza.

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Conexión con Beacon
Características:	El Beacon es un dispositivo que tramite vía bluetooth.
Descripción del requerimiento:	La aplicación se conecta al dispositivo por el bluetooth.
Requerimiento NO funcional:	
Prioridad del requerimiento:	Alta

-La información para el usuario es fundamental para el mejoramiento de la experiencia del usuario.

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Colocar información específica del sitio
Características:	El beacon mostrara la información de un sitio conforme nos acerquemos al beacon, siendo la información de modo texto e imágenes.
Descripción del requerimiento:	La información debe ser específicamente del sitio asociado al beacon para la mejora del usuario.
Requerimiento NO funcional:	
Prioridad del requerimiento:	

Requerimientos No Funcionales

-Es fundamental tener un diseño intuitivo para que el usuario no tenga ningún inconveniente al utilizar la app.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Diseño intuitivo.
Características:	La aplicación presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento:	Media

-Es necesario para mejorar la experiencia de usuario y solventar cualquier inconveniente posible

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Ayuda en el uso de la aplicación.
Características:	La interfaz del usuario deberá de presentar una opción de ayuda para el manejo de la aplicación.
Descripción del requerimiento:	La interfaz debe estar complementada con un buen sistema de ayuda.
Prioridad del requerimiento:	Media

Documentación de la aplicación

Se utilizaron dos métricas las cuales nos ayudaron a evaluar la aplicación móvil.

Métricas de rendimiento: que a través de ellas se consigue medir el desempeño del software, tanto de cada uno de sus módulos, como del sistema al completo.

Métrica de Rendimiento		
Sub característica de Calidad	Nombre de la Métrica	Porcentaje de cumplimiento de la característica de calidad
Promedio se calcula promediando los tiempos de respuesta	Tiempo de respuesta	90.0 %
Seguimiento del uso de la memoria de una aplicación	Uso de la memoria	80.0%

Métricas de usabilidad: para evaluar esta métrica se buscó soluciones intuitivas para ayudar a determinar si la solución cumple con dichos requisitos

Métrica de Usabilidad		
Sub característica de Calidad	Nombre de la Métrica	Porcentaje de cumplimiento de la característica de calidad
Comprensibilidad	Accesibilidad al sistema	50.0 %
Instructibilidad	Eficiencia de la documentación y el usuario o sistemas de ayuda en el uso	75.0%

Diagrama de casos de uso

- En este diagrama como muestra la figura 12 está representando los casos en los cuales el usuario intenta acceder al beacon mediante la aplicación móvil.

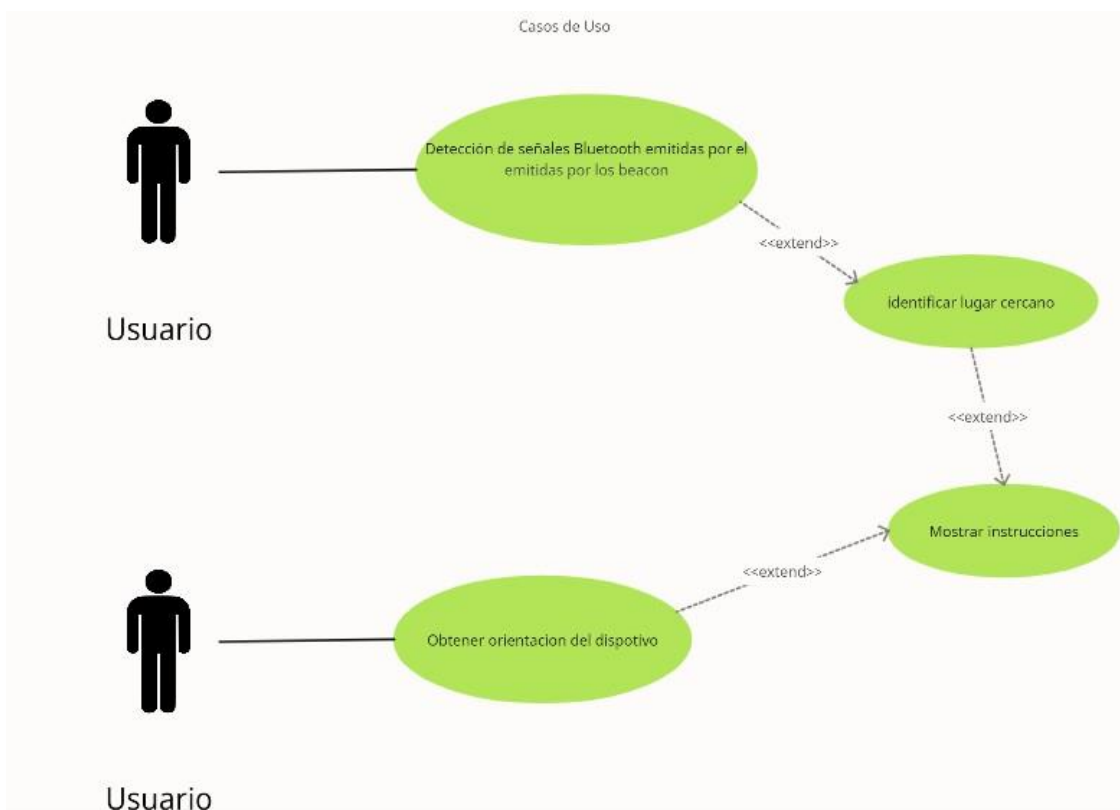


Figura 12. Casos de Uso

Diagramas de secuencia

- En la figura 13 se muestra un diagrama de secuencia el cual está informando sobre el proceso de escaneo y detección del beacon.

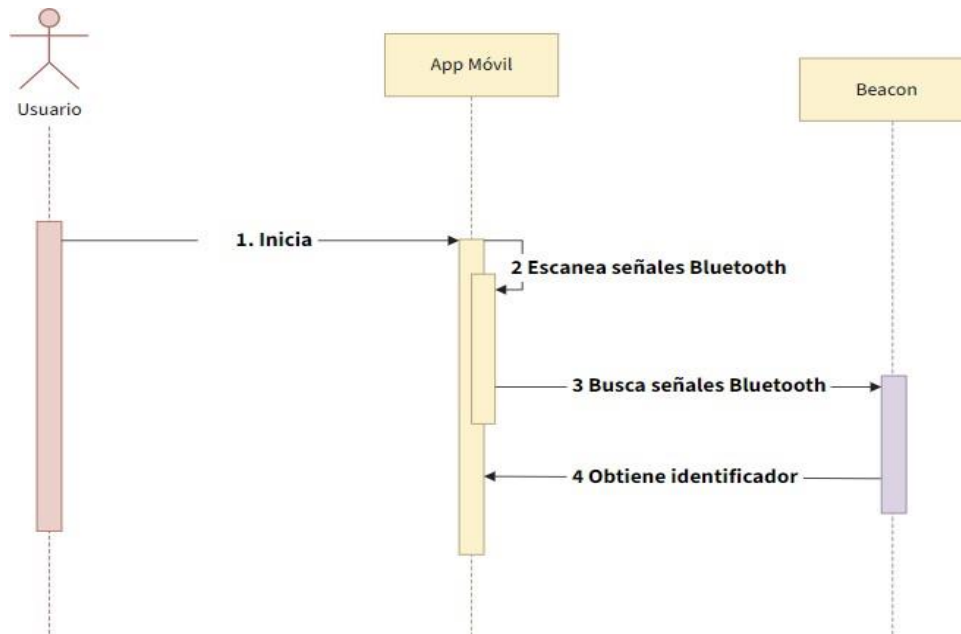
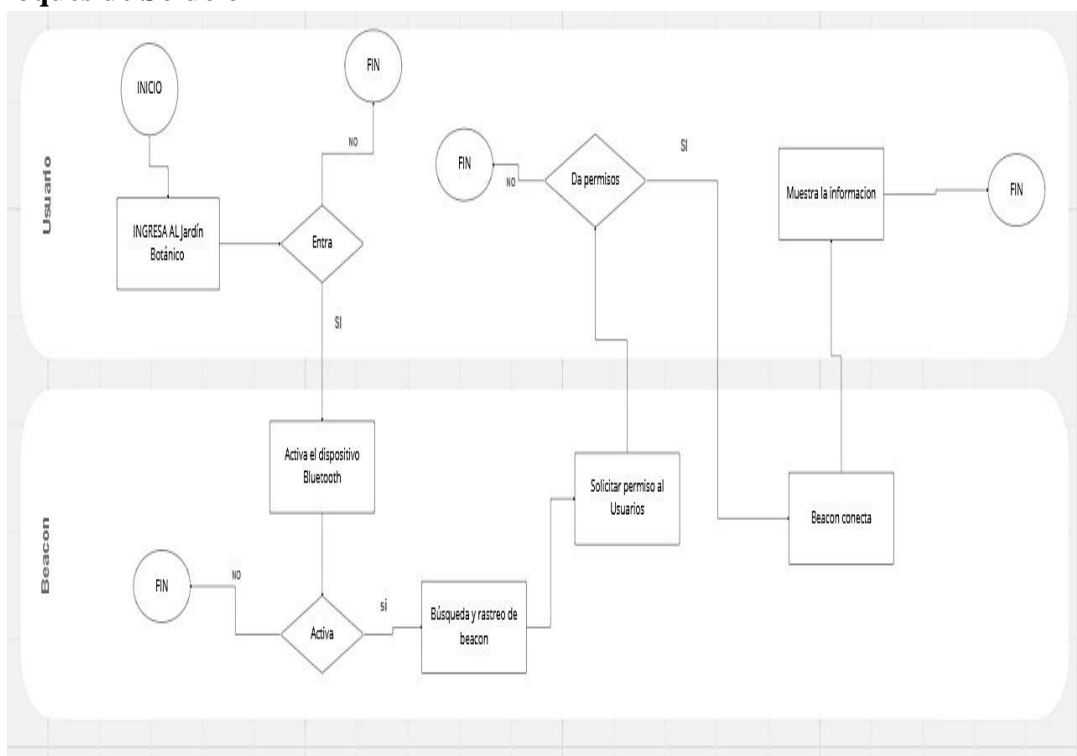


Figura 13. Detector de Beacon

Bloques de Solución

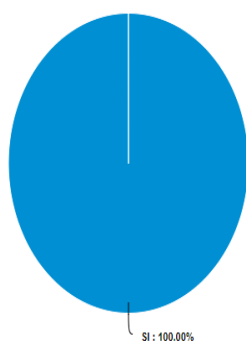
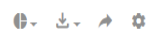


Análisis e Interpretación de los Resultados

Una vez terminado el desarrollo de la aplicación, se procedió a realizar el tratamiento correspondiente para el análisis del mismo, por cuanto el funcionamiento de la app cumplió los requerimientos funcionales y no funcionales con éxito, por lo tanto, mostrará eficiencia para mejorar la experiencia de usuario en el jardín botánico

PREGUNTA 1 – al evaluar a distintas personas con un formulario se dio a conocer que las personas entre 18 a 30 años prefieren utilizar más las aplicaciones móviles para dialogar y tener más amistades.

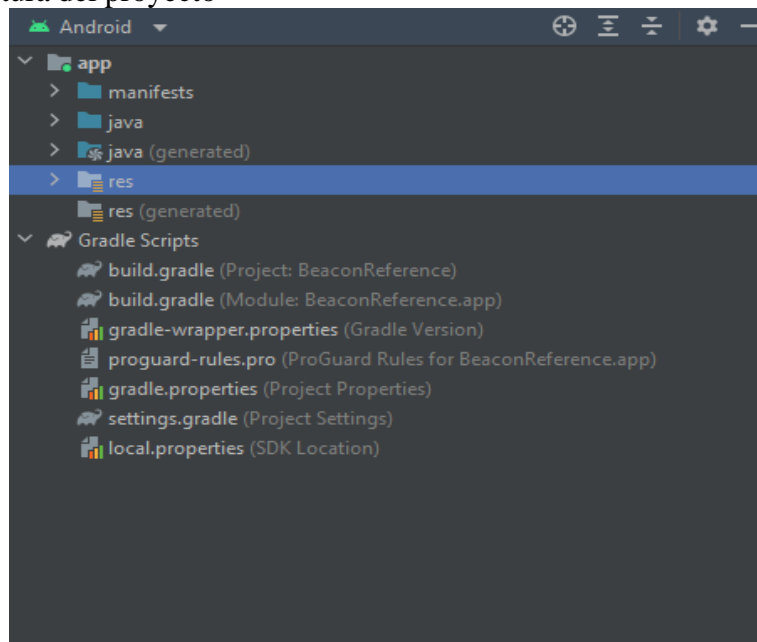
1. ¿Crees que hoy en la actualidad prefieren utilizar más las aplicaciones móviles?



Respuesta	Recuento	Porcentaje	20%	40%	60%	80%	100%
SI	20	100%					
NO	0	0%					
Total	20	n = 20					

Diseño, desarrollo e implementación:

1. Estructura del proyecto



2. Permissions Activity

En esta clase del proyecto se explica al usuario que la función no está disponible porque requiere un permiso que el usuario ha denegado. El mismo tiempo, respetar la decisión del usuario. No vincular al sistema configuración en un esfuerzo por convencer al usuario de que cambie su decisión.

```
3. package org.altbeacon.beacon.permissions

import android.Manifest
import android.annotation.SuppressLint
import android.app.AlertDialog
import android.content.Context
import android.content.pm.PackageManager
import android.graphics.Color
import android.os.Build
import android.os.Bundle
import android.util.Log
import android.view.Gravity
import android.view.View
import android.widget.Button
import android.widget.LinearLayout
import android.widget.TextView
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.content.ContextCompat
import androidx.core.view.setPadding
```

```

open class PermissionsActivity: AppCompatActivity() {

    val requestPermissionsLauncher =
        registerForActivityResult(
            ActivityResultContracts.RequestMultiplePermissions()
        ) { permissions ->

            permissions.entries.forEach {
                val permissionName = it.key
                val isGranted = it.value
                if (isGranted) {
                    Log.d(TAG, "$permissionName permission
granted: $isGranted")
                } else {
                    Log.d(TAG, "$permissionName permission
granted: $isGranted")
                }
            }
        }

    companion object {
        const val TAG = "PermissionsActivity"
    }
}

```

```

class PermissionsHelper(val context: Context) {

    fun isPermissionGranted(permissionString: String): Boolean {
        return (ContextCompat.checkSelfPermission(context,
permissionString) == PackageManager.PERMISSION_GRANTED)
    }

    fun setFirstTimeAskingPermission(permissionString: String,
isFirstTime: Boolean) {
        val sharedPreferences =
context.getSharedPreferences("org.altbeacon.permisisions",
AppCompatActivity.MODE_PRIVATE)
        sharedPreferences.edit().putBoolean(permissionString,
isFirstTime).apply()
    }

    fun isFirstTimeAskingPermission(permissionString: String): Boolean
{
        val sharedPreferences = context.getSharedPreferences(
            "org.altbeacon.permisisions",
            AppCompatActivity.MODE_PRIVATE
        )
        return sharedPreferences.getBoolean(
            permissionString,
            true
        )
    }
}

```

```

    fun beaconScanPermissionGroupsNeeded (backgroundAccessRequested:
Boolean = false): List<Array<String>> {
    val permissions = ArrayList<Array<String>>()
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
permissions.add(arrayOf(Manifest.permission.ACCESS_FINE_LOCATION))
    }
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        if (backgroundAccessRequested) {
permissions.add(arrayOf(Manifest.permission.ACCESS_BACKGROUND_LOCATION
))
        }
    }
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
permissions.add(arrayOf(Manifest.permission.BLUETOOTH_SCAN,
Manifest.permission.BLUETOOTH_CONNECT))
    }
    return permissions
}
}

```

```

open class BeaconScanPermissionsActivity: PermissionsActivity() {
    lateinit var layout: LinearLayout
    lateinit var permissionGroups: List<Array<String>>
    lateinit var continueButton: Button
    var scale: Float = 1.0f
    get() {
        return this.getResources().getDisplayMetrics().density
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        layout = LinearLayout(this)
        layout.setPadding(dp(20))
        layout.gravity = Gravity.CENTER
        layout.setBackgroundColor(Color.WHITE)
        layout.orientation = LinearLayout.VERTICAL
        val backgroundAccessRequested =
intent.getBooleanExtra("backgroundAccessRequested", true)
        val title = intent.getStringExtra("title") ?: "Permisos
Necesarios"
        val message = intent.getStringExtra("message") ?: "Para buscar
beacon, esta aplicación requiere los siguientes permisos del sistema
operativo. Toque cada botón para otorgar cada permiso requerido."
        val continueButtonTitle =
intent.getStringExtra("continueButtonTitle") ?: "Continuar"
        val permissionButtonTitles =
intent.getBundleExtra("permissionButtonTitles") ?:
getDefaultPermissionTitlesBundle()

        permissionGroups =
PermissionsHelper(this).beaconScanPermissionGroupsNeeded(backgroundAcc
essRequested)
    }
}

```

```

val params = LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.MATCH_PARENT,
    LinearLayout.LayoutParams.WRAP_CONTENT
)
params.setMargins(dp(0), dp(10), dp(0), dp(10))

val titleView = TextView(this)
titleView.setGravity(Gravity.CENTER)
titleView.textSize = dp(10).toFloat()
titleView.text = title
titleView.layoutParams = params

layout.addView(titleView)
val messageView = TextView(this)
messageView.text = message
messageView.setGravity(Gravity.CENTER)
messageView.textSize = dp(5).toFloat()
messageView.textAlignment = TextView.TEXT_ALIGNMENT_CENTER
messageView.layoutParams = params
layout.addView(messageView)

```

```

var index = 0
for (permissionGroup in permissionGroups) {
    val button = Button(this)
    val buttonText =
        permissionButtonTitles.getString(permissionGroup.first())
    button.id = index
    button.text = buttonText
    button.layoutParams = params
    button.setOnClickListener(buttonClickListener)
    layout.addView(button)
    index += 1
}

continueButton = Button(this)
continueButton.text = continueButtonText
continueButton.isEnabled = false
continueButton.setOnClickListener {
    this.finish()
}
continueButton.layoutParams = params
layout.addView(continueButton)

setContentView(layout)
}

fun dp(value: Int): Int {
    return (value * scale + 0.5f).toInt()
}

```



```

val buttonClickListener = View.OnClickListener { button ->
    val permissionsGroup = permissionGroups.get(button.id)
    promptForPermissions(permissionsGroup)
}

@SuppressLint("InlinedApi")
fun getDefaultPermissionTitlesBundle(): Bundle {
    val bundle = Bundle()
    bundle.putString(Manifest.permission.ACCESS_FINE_LOCATION,
"Ubicación")
bundle.putString(Manifest.permission.ACCESS_BACKGROUND_LOCATION,
"Ubicación en segundo plano")
    bundle.putString(Manifest.permission.BLUETOOTH_SCAN,
"Bluetooth")
    return bundle
}

fun allPermissionGroupsGranted(): Boolean {
    for (permissionsGroup in permissionGroups) {
        if (!allPermissionsGranted(permissionsGroup)) {
            return false
        }
    }
    return true
}

```

```

fun setButtonColors() {
    var index = 0
    for (permissionsGroup in this.permissionGroups) {
        val button = findViewById<Button>(index)
        if (allPermissionsGranted(permissionsGroup)) {
            button.setBackgroundColor(Color.parseColor("#448844"))
        }
        else {
            button.setBackgroundColor(Color.RED)
        }
        index += 1
    }
}

override fun onResume() {
    super.onResume()
    setButtonColors()
    if (allPermissionsGranted()) {
        continueButton.isEnabled = true
    }
}

fun promptForPermissions(permissionsGroup: Array<String>) {
    if (!allPermissionsGranted(permissionsGroup)) {
        val firstPermission = permissionsGroup.first()

        var showRationale = true
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            showRationale =
                shouldShowRequestPermissionRationale(firstPermission)
        }
        if (showRationale ||
            PermissionsHelper(this).isFirstTimeAskingPermission(firstPermission))
        {
            PermissionsHelper(this).setFirstTimeAskingPermission(firstPermission,
                false)
            requestPermissionsLauncher.launch(permissionsGroup)
        }
    }
}

```

```

        else {
            val builder = AlertDialog.Builder(this)
            builder.setTitle("Can't request permission")
            builder.setMessage("This permission has been
previously denied to this app. In order to grant it now, you must go
to Android Settings to enable this permission.")
            builder.setPositiveButton("OK", null)
            builder.show()
        }
    }
}

fun allPermissionsGranted(permissionsGroup: Array<String>):
Boolean {
    val permissionsHelper = PermissionsHelper(this)
    for (permission in permissionsGroup) {
        if (!permissionsHelper.isPermissionGranted(permission)) {
            return false
        }
    }
    return true
}

```

```

    }

    companion object {
        const val TAG = "BeaconScanPermissionActivity"
        fun allPermissionsGranted(context: Context,
backgroundAccessRequested: Boolean): Boolean {
            val permissionsHelper = PermissionsHelper(context)
            val permissionsGroups =
permissionsHelper.beaconScanPermissionGroupsNeeded(backgroundAccessReq
uested)
            for (permissionsGroup in permissionsGroups) {
                for (permission in permissionsGroup) {
                    if
(!permissionsHelper.isPermissionGranted(permission)) {
                        return false
                    }
                }
            }
            return true
        }
    }
}
}
}

```

3. MainActivity

-En esta clase se realizó las validaciones internas para poder mostrar al usuario toda la información correspondiente acorde al Beacon detectado.

```

package org.altbeacon.beaconreference

import android.app.AlertDialog
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.AdapterView
import android.widget.Button
import android.widget.ListView
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.Observer
import org.altbeacon.beacon.Beacon
import org.altbeacon.beacon.BeaconManager
import org.altbeacon.beacon.MonitorNotifier
import
org.altbeacon.beacon.permissions.BeaconScanPermissionsActivity

```

```

class MainActivity : AppCompatActivity() {
    lateinit var beaconListView: ListView
    lateinit var beaconCountTextView: TextView
    lateinit var monitoringButton: Button
    lateinit var rangingButton: Button
    lateinit var beaconReferenceApplication:
BeaconReferenceApplication
    var alertDialog: AlertDialog? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        beaconReferenceApplication = application as
BeaconReferenceApplication

        // Set up a Live Data observer for beacon data
        val regionViewModel =
BeaconManager.getInstanceForApplication(this).getRegionViewModel
        (beaconReferenceApplication.region)

```

```

        // observer will be called each time the monitored regionState
changes (inside vs. outside region)
        regionViewModel.regionState.observe(this, monitoringObserver)
        // observer will be called each time a new list of beacons is
ranged (typically ~1 second in the foreground)
        regionViewModel.rangedBeacons.observe(this, rangingObserver)
        rangingButton = findViewById<Button>(R.id.rangingButton)
        monitoringButton = findViewById<Button>(R.id.monitoringButton)
        beaconListView = findViewById<ListView>(R.id.beaconList)
        beaconCountTextView = findViewById<TextView>(R.id.beaconCount)
        beaconCountTextView.text = "No se detecta nada"
        beaconListView.adapter = ArrayAdapter(this,
android.R.layout.simple_list_item_1, arrayOf("--"))

```

```

    }

    override fun onPause() {
        Log.d(TAG, "onPause")
        super.onPause()
    }

    override fun onResume() {
        Log.d(TAG, "onResume")
        super.onResume()
        // You MUST make sure the following dynamic permissions are
granted by the user to detect beacons
        //
        // Manifest.permission.BLUETOOTH_SCAN
        // Manifest.permission.BLUETOOTH_CONNECT
        // Manifest.permission.ACCESS_FINE_LOCATION
        // Manifest.permission.ACCESS_BACKGROUND_LOCATION // only
needed to detect in background
        //
        // The code needed to get these permissions has become
increasingly complex, so it is in
        // its own file so as not to clutter this file focussed on how
to use the library.

```

```

        if (!BeaconScanPermissionsActivity.allPermissionsGranted(this,
            true)) {
            val intent = Intent(this,
BeaconScanPermissionsActivity::class.java)
            intent.putExtra("backgroundAccessRequested", true)
            startActivity(intent)
        }
    }

    val monitoringObserver = Observer<Int> { state ->
        var dialogTitle = "Beacons detectado"
        var dialogMessage = "didEnterRegionEvent has fired"
        var stateString = "en rango"
        if (state == MonitorNotifier.OUTSIDE) {
            dialogTitle = "Ningun beacon"
            dialogMessage = "didExitRegionEvent has fired"
            stateString = "fuera"
            beaconCountTextView.text = "Outside of the beacon region -
- no beacons detected"
            beaconListView.adapter = ArrayAdapter(this,
android.R.layout.simple_list_item_1, arrayOf("--"))
        }
        else {
            beaconCountTextView.text = "Inside the beacon region."
        }
        Log.d(TAG, "monitoring state changed to : $stateString")
        val builder =
            AlertDialog.Builder(this)
            builder.setTitle(dialogTitle)
            builder.setMessage(dialogMessage)
            builder.setPositiveButton(android.R.string.ok, null)
            alertDialog?.dismiss()
            alertDialog = builder.create()
            alertDialog?.show()
    }

```

```

}

val rangingObserver = Observer<Collection<Beacon>> { beacons ->
    Log.d(TAG, "En rango: ${beacons.count()} beacons")
    if
(BeaconManager.getInstanceForApplication(this).rangedRegions.size > 0)
{
    beaconCountTextView.text = "Ranging enabled:
    ${beacons.count()} beacon(s) detected"
    beaconListView.adapter = ArrayAdapter(this,
    android.R.layout.simple_list_item_1,
    beacons
        .sortedBy { it.distance }
        .map { "${it.id1}\nid2: ${it.id2} id3: rssi:
    ${it.rssi}\nest. distancia: ${it.distance} m" }.toTypedArray())
    }
}

fun rangingButtonTapped(view: View) {
    val beaconManager =
    BeaconManager.getInstanceForApplication(this)
    if (beaconManager.rangedRegions.size == 0) {
    beaconManager.startRangingBeacons (beaconReferenceApplication.region)
    rangingButton.text = "Stop Ranging"
    beaconCountTextView.text = "Ranging enabled -- awaiting
    first callback"
    }
    else {
    beaconManager.stopRangingBeacons (beaconReferenceApplication.region)
    rangingButton.text = "Empezar a oscilar"
    beaconCountTextView.text = "Ranging disabled -- no beacons
    detected"
}
}

```

```

    beaconListView.adapter = ArrayAdapter(this,
    android.R.layout.simple_list_item_1, arrayOf("--"))
    }
}

fun monitoringButtonTapped(view: View) {
    var dialogTitle = ""
    var dialogMessage = ""
    val beaconManager =
    BeaconManager.getInstanceForApplication(this)
    if (beaconManager.monitoredRegions.size == 0) {
    beaconManager.startMonitoring (beaconReferenceApplication.region)
    dialogTitle = "Empezar."
    dialogMessage = "Verá un cuadro de diálogo si se detecta
    un beacon y otro si los beacon dejan de detectarse.."
    monitoringButton.text = "Parar"
    }
    else {
    beaconManager.stopMonitoring (beaconReferenceApplication.region)
    dialogTitle = "parar monitoreo beacon."
    dialogMessage = "Ya no verá cuadros de diálogo cuando las
    señales comiencen o dejen de detectarse."
}
}
}

```

```

        monitoringButton.text = "Empezar"
    }
    val builder =
        AlertDialog.Builder(this)
    builder.setTitle(dialogTitle)
    builder.setMessage(dialogMessage)
    builder.setPositiveButton(android.R.string.ok, null)
    alertDialog?.dismiss()
    alertDialog = builder.create()
    alertDialog?.show()
}

companion object {
    val TAG = "MainActivity"
    val PERMISSION_REQUEST_BACKGROUND_LOCATION = 0
    val PERMISSION_REQUEST_BLUETOOTH_SCAN = 1
    val PERMISSION_REQUEST_BLUETOOTH_CONNECT = 2
    val PERMISSION_REQUEST_FINE_LOCATION = 3
}

//Metodo para ir Acerca de
fun acercaDe(view: View?) {
    val siguiente = Intent(this, AcercaDe ::class.java)
    startActivity(siguiente)
}

//Metodo para ir Ayuda
fun ayuda(view: View?) {
    val siguiente = Intent(this, Ayuda ::class.java)
    startActivity(siguiente)
}
}

```

4. BeaconReferenceApplication

-En esta clase se define de forma predeterminada, AndroidBeaconLibrary solo encontrará Beacon. De manera predeterminada, la biblioteca escaneará en segundo plano cada 5 minutos, que se limitará a trabajos de escaneo programados.

```
5. package org.altbeacon.beaconreference

import android.app.*
import android.content.Context
import android.content.Intent
import android.util.Log
import androidx.core.app.NotificationCompat
import androidx.lifecycle.Observer
import org.altbeacon.beacon.*
import org.altbeacon.bluetooth.BluetoothMedic

class BeaconReferenceApplication: Application() {
    lateinit var region: Region

    override fun onCreate() {
        super.onCreate()

        val beaconManager =
            BeaconManager.getInstanceForApplication(this)
            BeaconManager.setDebug(true)

            beaconManager.getBeaconParsers().clear()
    }
}
```



```

        beaconManager.getBeaconParsers().add(
            BeaconParser().
                setBeaconLayout("m:2-3=0215,i:4-19,i:20-21,i:22-
23,p:24-24"))

        region = Region("all-beacons", null, null, null)
        beaconManager.startMonitoring(region)
        beaconManager.startRangingBeacons(region)

        val regionViewModel =
BeaconManager.getInstanceForApplication(this).getRegionViewModel(region)

        regionViewModel.regionState.observeForever(
            centralMonitoringObserver)

        regionViewModel.rangedBeacons.observeForever(
            centralRangingObserver)
    }

    fun setupForegroundService() {
        val builder = Notification.Builder(this, "BeaconReferenceApp")
        builder.setSmallIcon(R.drawable.ic_launcher_background)
        builder.setContentTitle("Escaneando")
        val intent = Intent(this, MainActivity::class.java)
        val pendingIntent = PendingIntent.getActivity(
            this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT +
PendingIntent.FLAG_IMMUTABLE
    }

```

```

)
    builder.setContentIntent(pendingIntent);
    val channel = NotificationChannel("beacon-ref-notification-
id",
        "My Notification Name",
NotificationManager.IMPORTANCE_DEFAULT)
    channel.setDescription("My Notification Channel Description")
    val notificationManager = getSystemService(
        Context.NOTIFICATION_SERVICE) as NotificationManager
    notificationManager.createNotificationChannel(channel);
    builder.setChannelId(channel.getId());

BeaconManager.getInstanceForApplication(this).enableForegroundServices
canning(builder.build(), 456);
}

val centralMonitoringObserver = Observer<Int> { state ->
    if (state == MonitorNotifier.OUTSIDE) {
        Log.d(TAG, "outside beacon region: "+region)
    }
    else {
        Log.d(TAG, "inside beacon region: "+region)
        sendNotification()
    }
}

val centralRangingObserver = Observer<Collection<Beacon>> {
beacons ->
    Log.d(MainActivity.TAG, "Ranged: ${beacons.count()} beacons")
    for (beacon: Beacon in beacons) {
        Log.d(TAG, "$beacon about ${beacon.distance} meters away")
    }
}
}

```

```

        private fun sendNotification() {
            val builder = NotificationCompat.Builder(this, "beacon-ref-
notification-id")
                .setContentTitle("Jardin Botanico")
                .setContentText("Un Beacon cerca.")
                .setSmallIcon(R.drawable.ic_launcher_background)
            val stackBuilder = TaskStackBuilder.create(this)
            stackBuilder.addNextIntent(Intent(this,
MainActivity::class.java))
            val resultPendingIntent = stackBuilder.getPendingIntent(
                0,
                PendingIntent.FLAG_UPDATE_CURRENT +
                PendingIntent.FLAG_IMMUTABLE
            )
            builder.setContentIntent(resultPendingIntent)
            val notificationManager =
                this.getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
            notificationManager.notify(1, builder.build())
        }

        companion object {
            val TAG = "BeaconReference"
        }
    }
}

```

Evidencia del funcionamiento de la aplicación

Interfaz del Usuario

Pantalla INICAL DE PERMISOS

- Pantalla donde se pide permisos para el correcto funcionamiento de la aplicación.



Pantalla de bienvenida y reconocida de beacon

-Pantalla que nos ayuda para reconocer el beacon para la interacción de una pantalla emergente con la respectiva información



Pantalla de acerca de (Jardín Botánico)

- Pantalla informativa, donde se encuentra información básica del Jardín Botánico.

JARDÍN BOTÁNICO DE QUITO

**UN MARAVILLOSO LUGAR PARA LA
CONTEMPLACIÓN Y DIVERTIDO PARA LA
RECREACIÓN**

MISIÓN

Ejecutar acciones directas e indirectas destinados al conocimiento, protección y conservación del recurso vegetal ecuatoriano, en especial de la

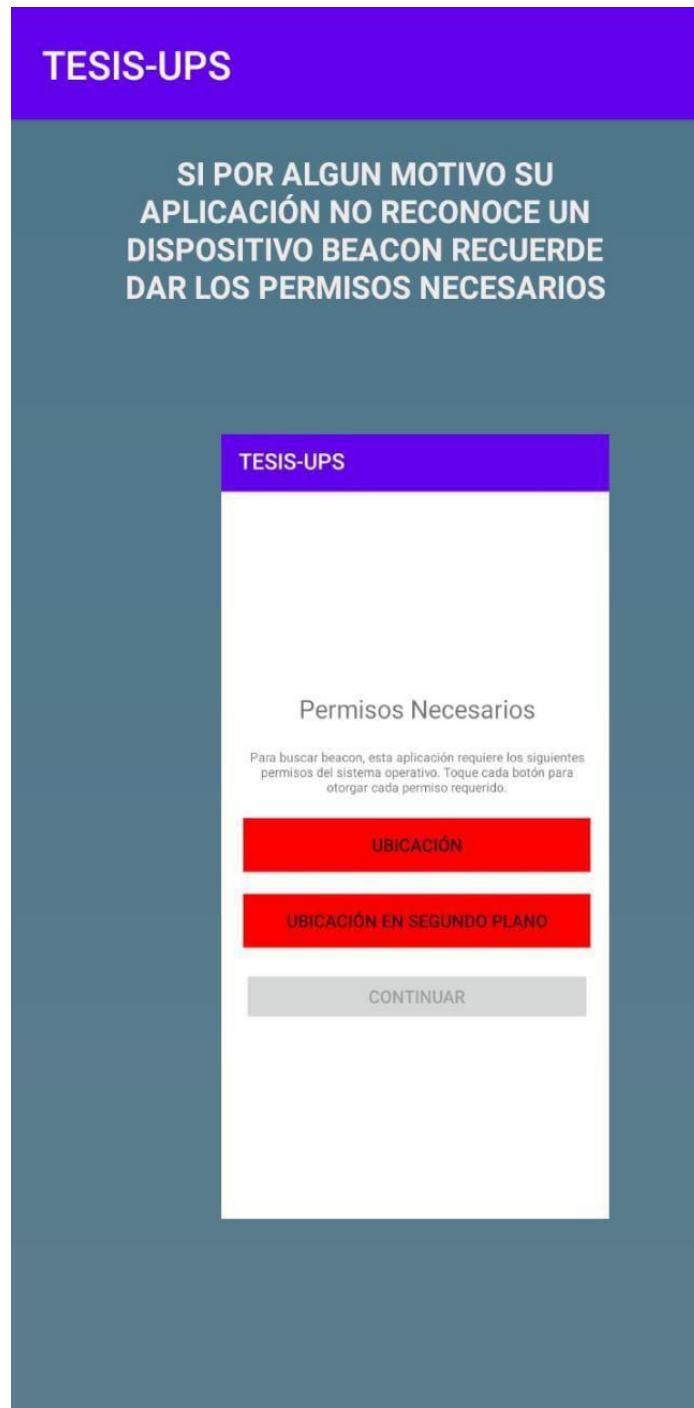
VISIÓN

El Jardín Botánico de Quito orienta sus esfuerzos con dinamismo para convertirse en una organización líder, eficiente y reconocida, en la gestión del patrimonio vegetal de los Andes ecuatorianos. Para el año 2020 se habrá constituido en un centro de investigación, conservación y exhibición de la flora regional nativa, en particular de las especies en riesgo de extinción, así como en un centro educativo sobre estos aspectos.



Pantalla de ayuda

-Pantalla de ayuda, aquí se encontrará ayuda el posible error de mal funcionamiento de la app.



Pantalla de Ruta



CONCLUSIONES

- El objetivo de este proyecto no es sustituir las herramientas tradicionales, sino aportar e integrar nuevas tecnologías para complementar y así ofrecer una mejor experiencia a los usuarios dentro del Jardín Botánico de Quito.
- El desarrollo de la aplicación Beacon fue desarrollado para permitir ofrecer a turistas o residentes, una nueva forma de obtener información e historia sobre los distintos puntos de interés de manera fácil y oportuna, sin la necesidad de depender de un guía especializado.
- Usar SCRUM como metodología de desarrollo permite lograr resultados en poco tiempo, dando mayor prioridad a los requerimientos, lo cual es una ventaja, ya que la aplicación y sus nuevas funcionalidades se pueden probar mucho más rápido que cuando se desarrolla con otras metodologías.
- Usar una plataforma de código abierto nos permite crear aplicaciones de calidad sin preocuparnos de las licencias, sus actualizaciones son periódicas por ser un software libre y esto hace que Android sea plataforma más atractiva para desarrollar aplicaciones móviles reduciendo los presupuestos de los proyectos.
- El proyecto realizado contribuye al desarrollo social y tecnológico, ya que mediante este documento investigativo y aplicativo podemos entender que existen nuevas tecnologías las cuales tienen mucha demanda y que actualmente no se están desarrollando en nuestro país.

RECOMENDACIONES

- Para el desarrollo de aplicaciones con Android es recomendable tener un conocimiento básico en lenguajes de programación y programación orientada a objetos.
- En cuanto al diseño de la aplicación es recomendable que los botones no seas pequeños ni que estes en la parte superior izquierda ya que la mayoría de usuarios prefieren manejar con una mano el móvil y presionan sin querer cualquier botón.
- Siempre que vayamos a empezar un proyecto de desarrollo tener en cuenta las versiones en las que está trabajando los dispositivos ya que suele suceder que al momento de querer probar he instalar, las dichas versiones no dejan tener una buena satisfacción.

Referencias

- Allen, S. (2004). *Diseños para el aprendizaje: Estudio de las exposiciones de los museos de ciencias que hacen algo más que entretener*. doi:<https://doi.org/10.1002/sce.20016>
- Angelaccio, M., Basili, A., & Liguori, W. (Marzo de 2012). *Smart and Mobile Access to Cultural Heritage Resources: A Case Study on Ancient Italian Renaissance Villas*. doi:<https://doi.org/10.1109/WETICE.2012.36>
- Arbañil, H., Barrientos, M., & Maguiña, S. (Agosto de 2018). *Innovación del Modelo de Negocio para Mejorar la Experiencia de Compra de los Clientes de un Supermercado*.
- Ardito, C., Bueno, P., Desolda, G., & Matera, M. (19 de Junio de 2018). *De los objetos inteligentes a las experiencias inteligentes: Un enfoque de desarrollo para el usuario final*. doi:doi.org/10.1016/j.ijhcs.2017.12.002
- Atherton, C. (18 de Octubre de 2019). *Web de la ciencia*. Obtenido de <https://bibliotecas.ups.edu.ec:3106/wos/woscc/full-record/WOS:000489102300001>
- Azar, E., & Merkt, M. (20 de Abril de 2016). *Uso de guías digitales en galerías de museos: determinantes de la selección de información*. Obtenido de ScienceDirect: <https://www.sciencedirect.com/science/article/abs/pii/S074756321530306X?via%3Dihub>
- Baz, A., Ferreira, A., & Álvarez, R. (2011). *Dispositivos Móviles*.
- Bermúdez, M., & López, H. (2011). *Análisis comparativo entre Sistemas Operativos de dispositivos móviles Android*.
- Bitgood, S. (Agosto de 2009). *A Critical Review. Visitor Studies*.
- Bitgood, S. (Agosto de 2009). *Museum Fatigue: A Critical Review. Visitor Studies*.
- Bitgood, S. (Julio de 2013). *Atención y valor. Claves para entender a los visitantes de museos*.
- Boccardo, Y., & Rie, D. (1 de Octubre de 2016). *Patrón de Diseño Beacon Action Manager*. Obtenido de Patrón de Diseño Beacon Action Manager.
- Byun, D., Cho, J., & Hong, D. (20 de Noviembre de 2016). *S-Beacon :Next Generation BLE Beacon Solution for Enhanced Personalization*, de *IEEE International Conference on Consumer Electronics (ICCE)*.
- Caballerro, T. (2016). *Beacons BLE (Bluetooth Low Energy) en el sector turístico, control de afluencia y servicios de valor añadido*.
- Cleve, J., & Mulzer, W. (05 de Agosto de 2020). *Web de la Ciencia*. Obtenido de <https://bibliotecas.ups.edu.ec:3106/wos/woscc/full-record/WOS:000551631800002>
- Copeland, T. (s.f.). doi:<https://doi.org/10.4324/9780203646052>
- Correa, J., & Jiménez, E. (2011). *Proyectos Museos vivos: Participación social y alfabetización digital*.
- Davey, G. (Enero de 2005). *¿Qué es la fatiga museística? Visitor Studies Today*.
- Dieguez, S. (14 de Febrero de 2015). *¿Por qué la gente no visita los museos de arte?*

- Eddystone Protocol Specification*. (2016).
- Eddystone Protocol Specification*. (2016). Obtenido de eddystone/protocol-specification.md at master · google/eddystone · GitHub
- EVE, M. (23 de Agosto de 2018-b). *Museos Móviles | EVE Museos e Innovación*.
- FAQ, S. (31 de Mayo de 2016). Obtenido de <http://androidstudiofaqs.com/conceptos/android-studio-historia>
- Gilchrist, C. (2014). *Learning IBeacon*.
- Isabel, M. (2014). *Servicios Web y SOA*.
- Jae-Hwan, J., & Lee, M.-J. (11 de mayo de 2018). Web de la ciencia. doi:10.14257/ijgdc.2018.11.4.05
- Juchnowicz, S., & Abad, M. (2011). *Itinerarios, Inalámbricos, Interactivos, In situ: Un modelos de difusión cultural basado en el uso de las TICs*.
- Kirchberg, V., & Trondle, M. (2012). *La experiencia de las exposiciones: A Review of Studies on Visitor Experiences in Museums*. doi:<https://doi.org/10.1111/j.2151-6952.2012.00167>.
- Molina, R., Sandoval, C., & Toledo, F. (2011). *Sistema Operativo Android: características y funcionalidad para dispositivos móviles*.
- Padron, J. (2016). *Tecnología beacon en entornos universitarios*.
- Pedrozo, G. (2012). *Sistemas Operativos en Dispositivos Móviles*.
- Peña Parra, M. (14 de Febrero de 2017). *Implementacion de los Servicios de proximidad para el Desarrollo de Ambientes Inteligentes usando BLE Beacons*.
- Perera, C., Aghaee, S., Faragher, R., Harle, R., & Blackwell, A. (30 de Noviembre de 2018). *IEEE Xplore*. doi:doi:10.1109/JSYST.2018.2878837
- Piccialli, F., & Chianese, A. (22 de Febrero de 2017). *The Internet of Things Supporting Context-Aware Computing: A Cultural Heritage Case Study. Mobile Networks and Applications*. doi:<https://doi.org/10.1007/s11036-017-0810-4>
- Ppandrea, M., & Salvatore, V. (Junio de 2010). *Proximity Marketing Solution Tailored to User Needs,» de IEEE International Symposium on" A World of Wireless, Mobile and Multimedia Networks*.
- Santos, A. (2009). *Revisión de los Servicios Web, Características y Rendimiento*.
- Schaffer, H., Pallot, M., & Oliveira, A. (Abril de 2011). *Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation BT - The Future Internet*.
- Sharif-Askari, H., & Abu-Hijleh , B. (2018). *Review of museums' indoor environment conditions studies and guidelines and their impact on the museums' artifacts and energy consumption*. doi:<https://doi.org/10.1016/j.buildenv.2018.07.012>
- Silicon, L. (2017). *Developing Beacons with Bluetooth® Low Energy (BLE)*.

- Sparacino, F. (2002). *The museum wearable: Real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences.* . Obtenido de Scopus.
- Studio, A. (11 de Diciembre de 2014). *Academia Android.* Obtenido de <http://academiaandroid.com/android-studio-v1-caracteristicas-comparativaeclipse/>
- Studio, A. (2017). Obtenido de <https://developer.android.com/studio/intro/index.html?hl=es-419>
- Tineo, A. C. (16 de Junio de 2016). *Beacons BLE (Bluetooth Low Energy) en el sector turistico, control de afluencia y servicios de valor añadido.*
- Zanele, N. (2016). *IoT based Proximity Marketing, de 2nd International Conference on Internet of Things, Big Data and Security.*