

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Diseño y prototipado de una pared de escalada controlada

por un App

Proyecto de investigación.

Alvaro Valdivieso Cortes

Ingeniería en Sistemas

Trabajo de titulación presentado como requisito
para la obtención del título de
Ingeniero en Sistemas

Quito, 6 de diciembre de 2018

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO DE CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN**

Diseño y prototipado de una pared de escalada controlada por un App

Alvaro Valdivieso Cortes

Calificación:

Nombre del profesor, Título académico

Fausto Pasmay, Master of Sciences

Firma del profesor

Quito, 6 de diciembre de 2018

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante:

Nombres y apellidos:

Alvaro Valdivieso Cortes

Código:

00110677

Cédula de Identidad:

1714816467

Lugar y fecha:

Quito, 6 de diciembre de 2018

RESUMEN

En este proyecto se propone una solución para un problema presente en gimnasios de Boulder (una modalidad de escalada en roca). Este se presenta en los tiempos de mantenimiento de las paredes de escalada pues los caminos o rutas que se forman con las presas (imitaciones artificiales de rocas naturales para dichas paredes) deben ser constantemente cambiados para presentar nuevos desafíos para el usuario. Todo esto se debe hacer manualmente cada cierto tiempo para lo cual se suelen usar presas o tape del mismo color para determinar un camino. En este caso, la solución a este problema se presenta mediante la implementación de una pared interactiva con una matriz de luces LED. Esta pared será controlada dinámicamente desde un App para que cada usuario cree sus propias rutas, así eliminando la necesidad de mantenimiento. Las luces serán controladas mediante un Raspberry Pi, el cual actuará de servidor bluetooth de baja energía (BLE) para comunicarse con el App.

Palabras clave: Bouldering, Bluetooth Low Energy (BLE), App, Matriz LED.

ABSTRACT

This Project proposes a solution for a frequent problem found in “Bouldering” (a type of rock climbing). That problem being the time it takes to set up new routes on a climbing wall due to the necessity of having new challenges for the people to constantly climb and challenge themselves without it getting old. For this, colored holds (artificially made rock imitations) or colored tape is used to mark routes according to their difficulty; the solution to this issue could be the implementation of an interactive LED matrix that can be controlled from an App so the users can create routes by themselves, therefore, cutting the need for constant maintenance on the wall. The LED light will be controlled by a Raspberry Pi acting as a Bluetooth low energy (BLE) server for the App to communicate with.

Key words: Bouldering, BLE, Raspberry Pi, App, LED Matrix.

TABLA DE CONTENIDO

1. Introducción	8
2. Desarrollo del Tema	10
2.1. HARDWARE	10
2.2. BLUETOOTH LOW ENERGY	14
2.3. SOFTWARE	17
3. Conclusiones	21
4. Trabajo futuro	22
5. Referencias bibliográficas	23

ÍNDICE DE FIGURAS

Figura 1 Luces WS2812b con pines soldados a las conexiones	11
Figura 2 Frente del Prototipo	13
Figura 3 Reverso del Prototipo	13
Figura 4. Proceso de Publicidad BLE	15
Figura 5. Proceso de Transmisión de datos BLE	17
Figura 6 Aplicación	20

1. INTRODUCCIÓN

En el campo de la escalada en roca, una de las variantes más populares es la del Boulder, una modalidad que se basa en escalar en rocas o muros de poca altura, sin más protección que colchones debajo de estos; la dificultad de este deporte se basa en resolver un desafío o secuencia de movimientos que varía en términos de complejidad y número de movimientos. A diferencia de otras modalidades, el Boulder se basa en escalada tanto horizontal como vertical, por lo que se enfoca más en la dificultad en el movimiento de la escalada más que en llegar alto. En el caso de los gimnasios de escalada se utilizan imitaciones artificiales de roca llamadas presas mediante las cuales se marcan caminos de entre 5 y 15 rocas, todo esto para entrenar a las personas de la manera más realista posible para cuando vayan a escalar en paredes naturales (Men's Health, 2018). Para marcar los distintos desafíos en estos gimnasios se suelen utilizar presas o tape de un mismo color para indicar tanto la ruta como la dificultad que esta presenta cada ruta, lo cual genera un problema en términos de mantenimiento de estas paredes, pues se tienen que estar cambiando constantemente los desafíos para que la gente se siga retando a sí misma y experimente con caminos nuevos cada vez y no llegue a memorizar y aburrirse con un set predeterminado cada vez que quiera escalar.

El problema con el mantenimiento de las paredes de escalar, es que consume tiempo y recursos desde el diseño de nuevos desafíos hasta el cambio físico de las presas o el costo del tape, por lo cual para este proyecto se propone la creación de una matriz de luces LED que se pueda instalar en una pared de Boulder y mediante la cual se pueda crear y visualizar rutas marcando el camino a través de luces que indicarán las presas a escalar en la pared. De esta manera los usuarios de esta podrán crear y proponer rutas para escalar, así logrando reducir los costos de mantenimiento de desafíos y logrando una mayor participación del usuario en

los mismos. Esta Matriz LED vendría a ser controlada desde un App, desarrollada en Android, la cual se encargará de la comunicación con la matriz. Esta App contará con una interfaz sencilla e intuitiva para controlar la pared y poder guardar y cargar rutas.

En estos últimos años la influencia y la popularidad del Boulder ha empezado a crecer en Ecuador. Gracias a esto cada vez más lugares especializados en este deporte están abriendo y de igual manera el número de gente interesada en el tema está creciendo. Este proyecto toma en cuenta a estos gimnasios de escalada como posibles interesados en el producto final, pues de momento el mercado ecuatoriano no cuenta con un producto similar y con esto se presenta una oportunidad para la comunidad de Boulder.

2. DESARROLLO DEL TEMA

2.1.HARDWARE

Para desarrollar este proyecto de manera adecuada el primer paso a tomar en cuenta es el hardware que se debe usar para realizar la matriz de luces led. Esto se puede dividir en varias partes:

Pared:

En un comienzo se planeó hacer una instalación directa de las luces en una pared prestada por Campo 4, debido a problemas que se explicarán en la sección de construcción no fue posible realizar una instalación completa, por lo cual se pasó a usar como pared una tabla de madera de 1,40m x 0.9m x 0,02m con huecos para las luces con una separación de 10cm en la cual se realizará un prototipo de 96 luces, con 8 luces por fila y 12 columnas.

Fuente de poder:

Para utilizar el menor número de conexiones eléctricas, se prefirió el uso de una fuente de voltaje de 5V 40A para alimentar a todo el circuito de 400 luces que iban a ser instaladas en un comienzo. Otra posibilidad requería tener varias fuentes de poder para alimentar a la pared desde varios puntos lo largo de la misma.

Luces:

Hubo 2 propuestas al momento de escoger las luces:

1. Usar luces LED de 1 solo color controladas desde un circuito compuesto por demultiplexores, de los cuales se controlaría las señales de poder y tierra para prender y apagar las luces rápidamente en sucesión para generar las rutas deseadas.
2. Usar luces LED ws2812b; estas cuentan con RGB y con circuito integrado que se maneja por PWM. Estas luces cuentan con 6 conexiones: 2 de poder, 2 de tierra, data in y data out, por lo cual el uso de un circuito externo para su control se vuelve innecesario, facilitando así la instalación.

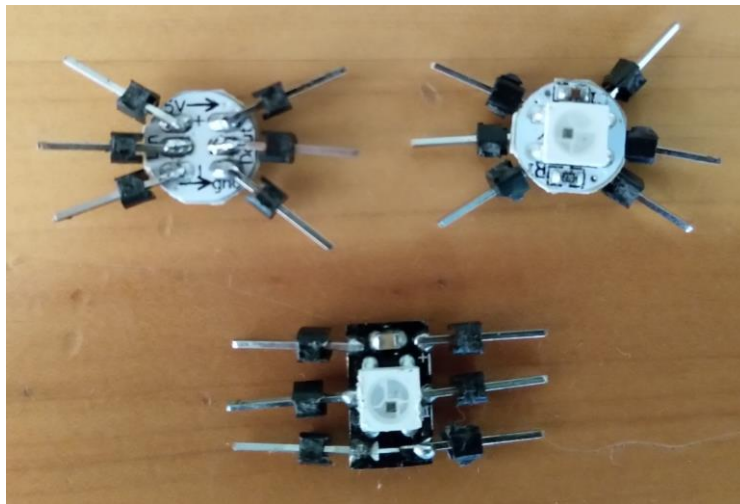


Figura 1 Luces WS2812b con pines soldados a las conexiones

Servidor:

En este caso se va a utilizar un Raspberry Pi modelo 3b como servidor para recibir datos desde el App y para controlar las luces. Se lo escogió debido a la versatilidad que dispone en relación a su tamaño, pues cuenta con Bluetooth integrado y gracias al GPIO se pueden controlar señales tanto de poder como PWM las cuales se pueden controlar de manera relativamente sencilla gracias a Rapsbian, el OS basado en Linux que el Raspberry usa.

Construcción

Ya al momento de construir y probar la matriz de luces en la pared, en un comienzo se realizó la instalación de la matriz en una pared prestada por Campo 4, donde se soldaron las luces directamente a cable de calibre AWG 14, y se las pegó a la pared con cinta doble faz para después ser ajustadas con grapas para cable. Este método de instalación resultó ser desastroso pues aparte de que se demoró bastante tiempo, se descubrió que debido a las vibraciones que se producen en la pared al momento de escalar, el cable que no está protegido o por estaño o por plástico se acaba rompiendo gracias al estrés generado por el ligero movimiento de las vibraciones; además de esto, se descartó la idea de soldar cable directo a las luces debido al alto costo de mantenimiento que se tendría al tener que reemplazar una luz instalada de esta manera.

Para el prototipo se pasó a soldar pines directamente a las luces para facilitar la reparación del prototipo en caso de que algún LED se rompa; para las conexiones entre luces se utilizaron jumpers para Arduino y grapas para asegurar los cables. Al armar este prototipo, el problema que se presentó fue muy distinto al que se presentó en Campo 4, pues en este caso la resistencia de los cables de Arduino es bastante más alta que la del cable 14, por lo que después de 8 luces, el voltaje ya no era suficiente para alimentar al resto de luces; debido a este problema se tuvo que usar borneras para tener una conexión directa a la fuente de poder en cada fila de luces.



Figura 2 Frente del Prototipo

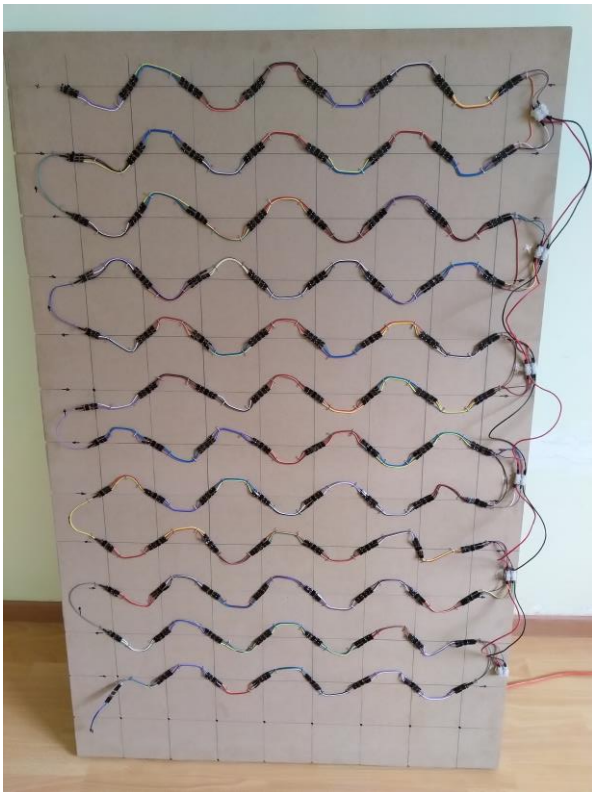


Figura 3 Reverso del Prototipo

2.2.BLUETOOTH LOW ENERGY

Se decidió usar Bluetooth de baja energía (BLE) para este proyecto debido a que iOS no permite utilizar librerías para conexión con bluetooth clásico a menos que se esté registrado en un programa específico para desarrollar esta tecnología con Apple, tras lo cual recién se recibiría la tecnología y documentación para desarrollar. BLE por otro lado, es un estándar que no requiere de nada más que las librerías y documentación ya disponible para iOS.

Para entender mejor cómo funciona el software en este proyecto, hay que tener claro ciertos conceptos que forman la base de las conexiones de bluetooth que se van a usar, pues el protocolo de baja energía de bluetooth es diferente del funcionamiento del bluetooth clásico.

El estándar Bluetooth LE se introduce como parte de la versión 4.0 de Bluetooth. Esta es una versión más liviana que el protocolo Bluetooth normal que además utiliza menos energía. La diferencia clave que tiene con el estándar clásico de bluetooth, el cual está diseñado para transmisiones continuas de datos, es que este fue diseñado para transmisiones pequeñas de datos, cortando el uso de energía tanto para el dispositivo maestro como para el esclavo.

Este estándar funciona en base a Perfiles de Atributos Genéricos (GAP) y Atributos Genéricos (GATT); para esto se utilizará el trabajo de Kevin Townsend (Introduction to Bluetooth Low Energy, 2014).

GAP:

El propósito de GAP es definir un perfil genérico que facilite los procedimientos de descubrimiento, conexión de dispositivos, y distintos niveles de seguridad.

En este caso, se necesita describir 2 roles que van a ser importantes más adelante:

Periférico: Estos dispositivos son los encargados de publicitar su señal y que generalmente se conectan a dispositivos más potentes.

Central: Comúnmente dispositivos móviles o computadoras, estos dispositivos se encargan de procesar datos para transmitir a los dispositivos periféricos o leer información de estos.

Para el descubrimiento del dispositivo periférico este envía paquetes de datos de hasta 31 bits de datos, comunicando el tiempo de envío de pedido al dispositivo central, tras lo cual empieza un tiempo de respuesta en el cual el dispositivo central puede pedir más información lo cual extenderá el periodo de publicidad; esta información será enviada después del tiempo estipulado en el paquete de pedido y esto acabará con el intervalo de publicidad, tras lo cual, el proceso se repetirá nuevamente.

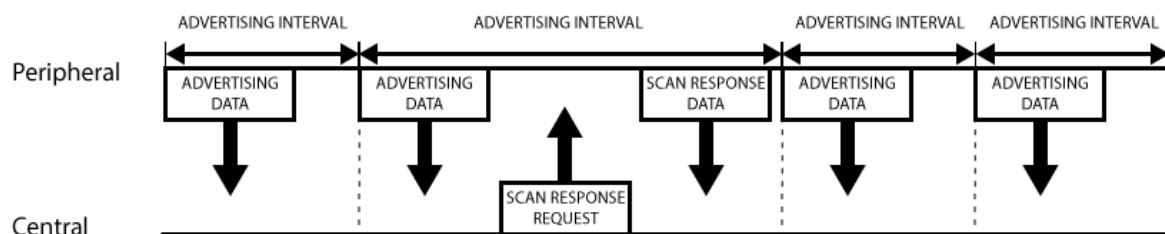


Figura 4. Proceso de Publicidad BLE (Townsend, Introduction to Bluetooth Low Energy)

GATT:

GATT por otro lado, se encarga de proporcionar acceso a una jerarquía de datos que se expone a conexiones de BLE; eso se logra a través de protocolos de atributos que generan servicios y características para dar acceso a los distintos servicios que pueda ofrecer el dispositivo periférico. Es importante entender bien que son estos servicios y características.

Servicios: Los servicios se usan para dividir los datos en entidades lógicas, cada una conteniendo características. Cada servicio puede tener varias características y siempre debe contar con al menos una. Los servicios se distinguen entre sí gracias a un UUID (Universal Unique Identity), un identificador numérico de 128 bits en el caso de servicios personalizados y 16 bit en el caso de servicios oficiales de BLE.

Características: El concepto de menor nivel en GATT, también requiere de un UUID y permite al dispositivo central interactuar con el periférico ya sea para leer datos, recibir datos o suscribirse a notificaciones del dispositivo.

Cada dispositivo periférico crea un servidor GATT en el cual auspicia sus servicios y características, las últimas siendo mostradas solo después de la conexión al servicio; el dispositivo central para realizar esta conexión crea un cliente GATT, el cual almacena las características y servicios enviadas por el servidor para ser utilizadas después. Estas conexiones se dan en intervalos definidos entre el cliente y servidor.

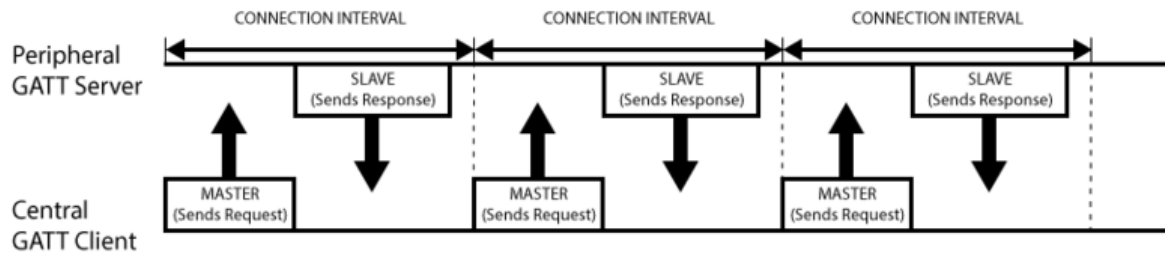


Figura 5. Proceso de Transmisión de datos BLE (Townsend, Introduction to Bluetooth Low Energy)

2.3.SOFTWARE

Se consideró usar conexiones tanto por wifi como por Bluetooth para controlar la matriz de luces desde el App; se acabó decidiendo por bluetooth para evitar problemas por conexiones simultáneas y por si no se dispone de conexión WLAN en donde se instale la pared.

En cuanto al funcionamiento del Bluetooth hay que tener en cuenta que IOS no soporta conexiones que no sean de baja energía (Bluetooth Low Energy), por lo que se decidió por usar esta misma tecnología para el diseño de la aplicación.

En este caso se puede dividir el software en el que se va a usar para el servidor y el que será usado en el App.

Servidor:

Para hacer uso de las capacidades BLE del Raspberry se comenzó diseñando un programa en Python para el manejo de estas conexiones; después de varias pruebas con distintos lenguajes de programación (Android, IOS, C#, React-Native y Flutter) se determinó que las conexiones eran demasiado inestables como para proceder con este lenguaje, pues no se encontraba la

señal de Bluetooth con suficiente frecuencia como para conectarse en poco tiempo y cuando se conectaba, a veces no se lograba enviar datos al servidor.

La siguiente tecnología que se intentó fue Node.js, con el cual gracias al módulo `bleno`, se logró de manera relativamente sencilla la creación del servidor y el manejo de datos que ya era lo suficiente estable para varios de los lenguajes de programación mencionados anteriormente.

Cliente (App):

En cuanto al cliente se consideraron varios lenguajes de programación para usar, cada uno con sus ventajas y desventajas:

Flutter: Es un lenguaje novedoso que, a pesar de contar con librerías para trabajar con estas conexiones y ser multiplataforma, no cuenta con la documentación ni con una comunidad en línea lo suficientemente interesada como para proveer ejemplos y resolver dudas o errores. No se logró que se conecte al servidor a pesar de poder encontrar la señal del servidor.

C#: Por medio de Xamarin se puede programar tanto para Android como IOS, pero la programación del manejo de BLE se tiene que programar individualmente para cada plataforma y solo se incluye la librería de BLE de Android por defecto, no se llegó a buscar la librería de conexiones BLE para IOS pues la conexión usando esta tecnología causaba desconexiones espontáneas y a veces no se lograba una conexión con el servidor y se determinó que no era apropiado continuar.

React-Native: Lenguaje multiplataforma, aun en beta, se lograba una conexión bastante estable con el servidor y se puede escribir al mismo sin problema; como es un lenguaje en

beta, hacer debugging no es nada sencillo pues algunos errores no son claros y son específicos a la plataforma en la que se trabaje. Este lenguaje basado en javascript aun cuenta con problemas de memory leak, por lo cual uno tiene que tener cuidado con como se diseña el App, pues un diseño que no tome en cuenta eso puede resultar en un App bastante lenta.

Android/IOS: Se tiene que programar un App por plataforma, pero se garantiza que va a funcionar, los métodos de BLE no están muy bien documentados, pero hay librerías para lidiar con eso.

Implementación:

Para la creación del App se decidió programar en Android pues se garantiza que la conexión va a funcionar bien y se va a poder hacer debugging de los errores que surjan sin mayor problema. Para esta App se contará con un diseño sencillo para demostrar la funcionalidad del prototipo. Para facilitar las conexiones de bluetooth se hizo uso de la librería AndroidBluetoothLibrary hecha por Douglas Nassif Roma Junior. El layout de la aplicación se dividirá en 3 partes: una para la selección de color, otra para distintos inputs y para desplegar el estado de conexión y finalmente una matriz de botones que serán los encargados de reflejar las luces que se prendan y apaguen en la pared.

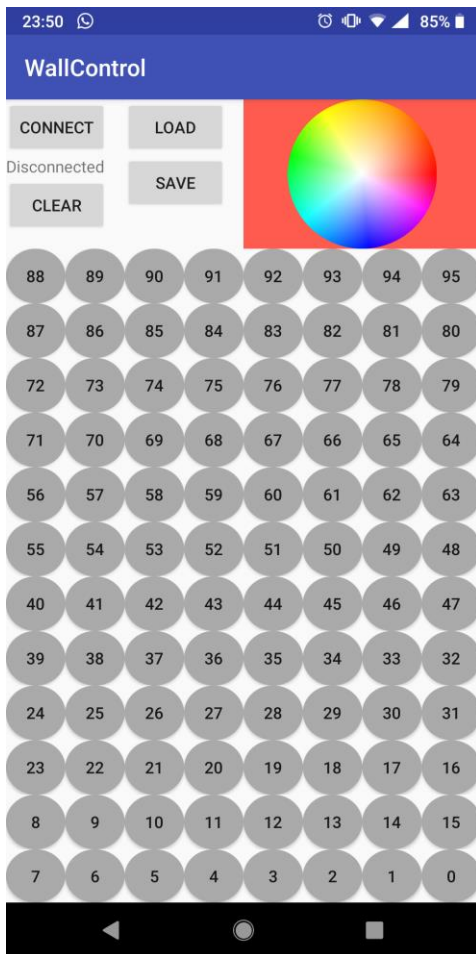


Figura 6 Aplicación

Entre los distintos inputs se encuentra el botón para conectarse a la pared, el cual una vez conectado a la pared, funciona para terminar la conexión; debajo se despliega el estado de la conexión, ya sea si está buscando conectarse, está conectado o desconectado. Bajo este, hay un botón para limpiar la matriz en caso de que el usuario quiera. A su derecha se encuentra un botón para guardar la ruta que está en la pared actualmente y un botón para cargar la ruta previamente guardada.

3. CONCLUSIONES

El prototipo fue desarrollado exitosamente, demostrando el potencial que este tiene para ser implementado en un gimnasio de Boulder una vez que se desarrolle un poco más la idea para hacerla más estable frente a las vibraciones de la pared real. En un futuro se trabajará para alcanzar ese objetivo para lo cual se encontrarán más problemas que no surgieron en este prototipo debido a la escala en la que se lo hizo y las condiciones en las que está.

En cuanto a la aplicación, el desarrollo en Android fue el más exitoso de todos, ayudado por la simplicidad del diseño del App; el único problema que tiene y que se puede trabajar a futuro es el uso de la librería `AndroidBluetoothLibrary` para facilitar las conexiones BLE, pues no se garantiza que a medida que se actualicen los métodos nativos de Android, la librería siga funcionando, por lo cual la siguiente iteración deberá ser desarrollada con los métodos provistos en las librerías de Android.

Los problemas más grandes que se presentaron para este proyecto recaen más que nada en la obtención de las luces, pues no se encuentran en Ecuador y se tienen que importar. En cuanto a las vibraciones que se presentan en la pared real, es un gran problema el cual se va a tener que resolver con más tiempo en un futuro, pues es un requiere de bastante tiempo para solucionar debido a las pruebas en tiempo real que se deben realizar en la pared de Boulder.

4. TRABAJO FUTURO

Debido a que el trabajo quedó como un prototipo nada más a futuro se realizarán más pruebas para averiguar la resistencia de las luces frente a la vibración de la pared ya en una pared real y se procederá a hacer una instalación completa del sistema. Una vez que la pared esté ya funcional, se recolectarán las rutas ingresadas por el usuario y se procederá a crear un algoritmo que use machine learning para generar rutas automáticamente, así volviendo aún más tentadora la idea de este proyecto para la comunidad de Boulder.

En cuanto al App se trabajará en migrar las conexiones BLE hacia los métodos nativos de Android y se agregará la posibilidad de generar rutas automáticamente con ayuda de los algoritmos de machine learning mencionados anteriormente.

5. REFERENCIAS BIBLIOGRÁFICAS

Doman Junior, D. N. (2018, October 11). AndroidBluetoothLibrary. Retrieved from <https://github.com/douglasjunior/AndroidBluetoothLibrary>

Men's Health. (2018, Julio 21). ¿Qué es la escalada en boulder? Recuperado de <https://www.menshealthlatam.com/como-practicar-escalada-boulder/>

Ray, B. (2015, Noviembre 1). Bluetooth Vs. Bluetooth Low Energy: What's The Difference? Recuperado de <https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy>

Townsend, K. (2015). Adafruit - Introduction to Bluetooth Low Energy. 2014-04-17, 1–11. Recuperado de <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction%0Ahttps://learn.adafruit.com/introduction-to-bluetooth-low-energy>