

**Diseño e implementación de enlace punto a punto y su sistema de adquisición de datos
para la transmisión de imágenes de un cultivo de guanábana**

Fernando Rojas Téllez

Universitaria Agustiniana
Facultad de Ingeniería
Programa Ingeniería en Telecomunicaciones
Bogotá, D.C.
2020

**Diseño e implementación de enlace punto a punto y su sistema de adquisición de datos
para la transmisión de imágenes de un cultivo de guanábana**

Fernando Rojas Téllez

Director

Ing. Martha Isabel Villarreal López

Trabajo de grado para optar al título de Ingeniero en Telecomunicaciones

Universitaria Agustiniana

Facultad de Ingeniería

Programa Ingeniería en Telecomunicaciones

Bogotá, D.C.

2020

Resumen

A continuación, a lo largo del documento, se pretende abarcar la temática referente al desarrollo de un enlace punto a punto para la transmisión de imágenes de un cultivo, en este caso de guanábana, sus componentes y fundamentos necesarios para su funcionamiento. Se incluye esencialmente el sistema de adquisición de datos, indispensable para la recolección de la información pertinente para determinar la presencia o no de las principales plagas que afectan el cultivo, lo que garantizar un control eficaz del mismo; por ello se describirán también a detalle los métodos y equipos empleados para garantizar la fidelidad del enlace y la información que contendrá tanto en la transmisión como en la recepción, así como las pruebas ejecutadas que servirán para realizar un análisis para su tratamiento dependiendo de los resultados obtenidos en un proyecto futuro.

Abstract

Next, throughout the document, we will try to cover the subject related to the development of a point-to-point link for the transmission of images of a crop, in this case soursop, their components and necessary foundations for the operation. It essentially includes the data acquisition system, which is essential for the collection of relevant information to determine the presence or not of the main pests that affect the crop, which guarantees an effective control of it; therefore, it will also describe in detail the methods and equipment used to ensure the fidelity of the link and the information it will contain both in transmission and reception, as well as the tests performed that will serve to make an analysis for its treatment depending on the results obtained in a future project.

Tabla de contenidos

Planteamiento del problema	9
Justificación.....	10
Objetivos	11
Objetivos específicos.....	11
Marco referencial	12
Estado del arte	12
“Sistema de análisis de imágenes para evaluar el color del café tostado y su relación con las propiedades organolépticas.....	12
IOT and agriculture data analysis for smart farm.	12
“Sistema de adquisición, almacenamiento y análisis de información fenológica para el manejo de plagas y enfermedades de un duraznero mediante tecnologías de precisión.	12
Marco teórico.....	13
Marco conceptual	14
Sistema de adquisición de datos.	14
Dispositivo DAQ.	14
Convertidor Analógico Digital (ADC).	14
Almacenamiento de datos.....	15
Transmisión de la información.	15
Pérdida de paquetes.....	15
Marco legal	15
Norma 802.11.	16
Norma 802.15.	16
Norma 802.16.	16
Resoluciones 2016.	16
Resoluciones 2013.	17
Metodología	18
Desarrollo	19
Requerimientos del proyecto.....	19
Almacenamiento transitorio y acondicionamiento de información.....	40
Transmisión y recepción.....	46

Router o Transmisor.	46
Dispositivo terminal o receptor.....	46
Trabajos futuros.....	52
Conclusiones	53
Referencias	55

Lista de figuras

Figura 1. Ejemplo de un formato de trama de datos.	14
Figura 2. Convertidor analógico digital.	15
Figura 3. Esquema metodológico.....	18
Figura 4. Esquema general.....	19
Figura 5. Broca del fruto (<i>Cerconata anonella</i>); Broca del tallo (<i>Croatosomus bombina</i>); Perforador de las semillas (<i>Bephrata naculicollis</i>) respectivamente.	20
Figura 6. Distribución de plantas del cultivo	21
Figura 7. Esquema hipotético del sistema de cámara.....	21
Figura 8. Esquema hipotético del sistema de cámara.....	22
Figura 9. Esquema hipotético del sistema de cámara.....	22
Figura 10. Esquema hipotético del sistema de cámara.....	23
Figura 11. Esquema distribución del cultivo.....	23
Figura 12. Pixy 2 CMUcam5	27
Figura 13. Pan/Tilt Servo Motor Kit.....	27
Figura 14. Características modulo Xbee Pro S1.....	28
Figura 15. Características modulo Xbee Pro S1.....	28
Figura 16. Características físicas modulo Xbee Pro S1	28
Figura 17. Aspecto real Xbee Pro S1 y adaptador Xbeexplorer.....	29
Figura 18. Equivalencia en PW.....	30
Figura 19. Esquema hipotético de la cámara.....	31
Figura 20. Esquema de puntos estratégicos.....	31
Figura 21. Caja Raspberry	33
Figura 22. Perspectiva de estructura de cámara hipotética.....	33
Figura 23. Estructura punto origen de alimentación (Sujeto a cambios por dimensiones panel).....	34
Figura 24. Estructura punto origen de alimentación	34
Figura 25. Visualización puntos estratégicos en recorrido del riel.	36
Figura 26. Esquema de transmisión módulos Xbee.	36
Figura 27. Esquema enlace punto a punto y distancia	37
Figura 28. Esquema enlace punto a punto y distancia, perspectiva.	37
Figura 29. Esquema locación de receptor, punto cercano al cultivo.	38
Figura 30. Punto de ubicación receptor.....	38

Figura 31. Punto de ubicación receptor, perspectiva de visión hacia el cultivo.....	39
Figura 32. Punto de ubicación transmisor, perspectiva de visión hacia punto receptor en casa.	39
Figura 33. Distribución almacenamiento.	40
Figura 34. Aspecto Raspberry e interfaz gráfica.....	40
Figura 35. Montaje de sistemas.....	41
Figura 36. Instalación MariaDB MySQL.....	41
Figura 37. Estado del servicio MySQL.....	42
Figura 38. Asignación de permisos y creación DB.....	42
Figura 39. Creación tablas y parámetros.....	43
Figura 40. Python, conexión con base de datos.	44
Figura 41. Python, Inserción de datos.	44
Figura 42. Python, Inserción de datos.	45
Figura 43. Python, Visualización salida de código.	45
Figura 44. Xbee configuración transmisor o router..	46
Figura 45. Xbee configuración receptor o dispositivo terminal.	47
Figura 46. Xbee prueba transmisión.	47
Figura 47. Xbee prueba recepción.....	48
Figura 48. Python, código para envío de información codificada mediante el Xbee.....	48
Figura 49. Tamaño de la imagen en KB.....	49
Figura 50. Tamaño de la imagen en caracteres del código.....	49
Figura 51. División del código.	49
Figura 52. Debian OS, ejecución del código XBEE.py y su visualización de información.....	50
Figura 53. XCTU, recepción de código en el receptor Xbee	50
Figura 54. Validación del código.	51
Figura 55. Código unificado traducido en imagen.	51

Planteamiento del problema

Actualmente, el uso de herramientas tecnológicas para la interacción con la vida diaria y el trabajo, toman mayor fuerza, más aún cuando se empiezan a tratar ámbitos tales como el uso de estas en áreas donde el alcance de la tecnología no ha tenido mayor impacto, para este caso, las áreas rurales. Puntualmente, se aborda entonces el tema del monitoreo de un cultivo de Guanábana, el cual, se ve seriamente comprometido por plagas como la polilla, larva *Spodoptera frugífera*, la Broca del fruto (*Cerconata anonella*), la Broca del tallo (*Croatosomus bombina*), el perforador de las semillas (*Bephrata naculicollis*), entre otras, para lo cual debe hacerse un control constante, puesto que el avance de estas plagas puede comprometer decisivamente la calidad y comportamiento de la fruta.

De esta manera, se pretende plantear una alternativa para garantizar un control constante del estado del cultivo, actividad que se realiza normalmente física y presencial, lo que no garantiza constancia por disponibilidad de tiempo y transporte, requiriéndose entonces una alternativa que pudiese brindar facilidades para el monitoreo a distancia, que pueda transmitir información en imágenes y almacenarlas para su utilidad, con información para el usuario y un análisis sobre la conectividad y calidad del servicio.

Justificación

Se parte de la premisa de la situación existente en el área agrícola, respecto al monitoreo del estado de un cultivo, donde tradicionalmente, se realiza mediante un proceso presencial en sitio. La implementación de herramientas tecnológicas en distintas áreas de trabajo, más específicamente, en las áreas rurales y agrícolas, se ha visto cuestionada, según (Cruz, 2010; Nagel, 2012; Vega & Baena, 2014), puesto que se tiene la percepción de que hay muy poca presencia de las mismas en estas áreas, por ello es importante recalcar la promoción de la optimización de procesos, representados en ahorro de tiempo, costos y estándares de calidad que traen consigo (Claudio Mourgues, 2001). Es en este momento, cuando se da la inclusión de dichos sistemas, cuya finalidad es lograr la interacción digital con el mundo físico, mediante distintos medios de infraestructura, que permitan la extracción de información útil para uno o varios usuarios.

Dicho esto, se pretende diseñar un sistema de adquisición y almacenamiento de datos, representados en imágenes de un cultivo de guanábana, mediante la conexión de nodos principales, permitiendo así el análisis de información de manera útil. Cabe resaltar que se pretende tener en cuenta aspectos fundamentales e imperativos como la calidad del servicio, pérdidas, disponibilidad de servicio o posibles retardos, características participes en una red, y que determinan el rendimiento óptimo o no de la misma.

Objetivos

Objetivo general

Desarrollar un enlace punto a punto que proporcione conectividad para la transmisión de imágenes que permitan identificar plagas en un cultivo de guanábana, incluyendo el diseño de su correspondiente sistema de adquisición de datos.

Objetivos específicos

- Diseñar el sistema de adquisición de datos para el almacenamiento y transmisión de imágenes que permitan identificar las plagas del cultivo, teniendo en cuenta las especificaciones, requerimientos de montaje y capacidades del sistema.
- Implementar el sistema de adquisición de datos en el nodo esclavo para la transmisión hacia el nodo central o receptor.
- Realizar el establecimiento de conectividad entre los dos puntos o nodos en conjunto con el protocolo de pruebas de la red, para la validación de funcionamiento bajo requerimientos de conectividad y calidad de servicio.

Marco referencial

Estado del arte

“Sistema de análisis de imágenes para evaluar el color del café tostado y su relación con las propiedades organolépticas.

En este trabajo se presentan los resultados obtenidos del procesamiento digital de imágenes, para evaluar el color y los matices del café tostado y proponer su relación cuantitativa con sus propiedades organolépticas. Dado que, para una misma muestra de café con diferentes perfiles de tostado, se obtienen resultados diferentes y casi irreconocibles, se implementó un algoritmo de procesamiento digital de imágenes del café tostado variedad Borbon Pointu, de la unidad productiva de Supracafé, para comparar cuantitativamente los matices y el color del café tostado bajo 8 perfiles de tostado. Las muestras de café fueron tostadas bajo los esquemas de la escala AGTRON GOURMET, para posteriormente adquirir las percepciones de color, a través de fotografías de cada muestra y extraer los índices de color y matiz (Tono) en el espacio de color RGB y CIELAB con procesamiento digital de imágenes.” (Universidad Distrital Francisco José de Caldas, 2018)

IOT and agriculture data analysis for smart farm.

“Aquí, se plantea el desarrollo de un sistema de riego óptimo de cultivos agrícolas basado en un sensor inalámbrico red. Inicialmente se diseña y desarrolla un sistema de control mediante sensores de nodos en el campo de cultivo con datos a través de un teléfono inteligente y una aplicación web. El hardware fue diseñado e implementado en la caja de control hardware conectada a recoger datos sobre los cultivos. Los sensores de humedad del suelo se utilizan para monitorear el campo, conectándose a la caja de control. Posteriormente la aplicación basada en la web que fue diseñada e implementada para manipular los detalles de datos de cultivos e información de campo. Es así como la transmisión y almacenamiento de estos datos se usaron para análisis con el fin de predecir la temperatura, humedad y humedad del suelo para un manejo eficaz del crecimiento de los cultivos” (Jirapond Muangprathuba N. B., 2019)

“Sistema de adquisición, almacenamiento y análisis de información fenológica para el manejo de plagas y enfermedades de un duraznero mediante tecnologías de precisión.

Se explica el funcionamiento y aspectos generales de un sistema que es capaz de seguir el comportamiento de las enfermedades que afectan al durazno mediante tecnologías de arquitectura de precisión. El sistema allí descrito emplea una red inalámbrica para poder monitorear el clima además de un sistema con Asistente Digital Personal, procesamiento y visualización. El sistema de

adquisición de datos para este caso, permite adquirir la información que proviene del conteo de hojas y los frutos del durazno para luego ser transferidos al centro de seguimiento, donde se almacena la información para realizar los cálculos y dimensionar espacialmente la enfermedad que este atacando al duraznero. Se emplea un software llamado Cybertracker.” (Jiménez, Ravelo, & Gómez, 2010)

Marco teórico

“Las tecnologías avanzadas pueden aportar beneficios a la mayoría de las personas. En los últimos años, la Internet de los objetos (IO) ha empezado a jugar un papel importante en el desarrollo de la sociedad de la información, en la vida cotidiana, ampliando nuestras percepciones y nuestra capacidad para modificar el entorno que nos rodea. En particular, el sector agroindustrial y de los campos ambientales aplican IOs tanto en el diagnóstico como en el control. Además, puede proporcionar información al usuario final/consumidor sobre el origen y las propiedades del producto (Talavera et al., 2017).

Puede ayudar a evaluar variables de campo como el suelo estado, condiciones atmosféricas y biomasa de plantas o animales. Puede también se puede utilizar para evaluar y controlar variables como la temperatura y la humedad, vibraciones o choques durante el transporte del producto (Pang et al., 2015). Además, el ruido blanco gaussiano puede ser utilizado para monitorear y controlar factores que influyen en el crecimiento y el rendimiento de los cultivos. También se pueden utilizar para determinar el momento óptimo para cosechar, qué agricultor es más adecuado para qué de control, etc., (Ndzi et al., 2014).” (Jirapond Muangprathuba *. N., 2019)

Es de entenderse que se debe analizar inicialmente el tema del envío de imágenes, puesto que de esto depende el funcionamiento principal del sistema, ya que es el dato que se interpretara, para ello se analiza él envío de tramas, el cual, recordando su definición, sería “La unidad de intercambio de información en los protocolos de enlace de datos. Una trama es un bloque de datos que además contiene información de control, empleada por el protocolo para identificar a la trama. Cuando el método de transmisión utilizado está orientado a bloques (como muchas transmisiones síncronas) la sincronización de trama ya está resuelta por el método de transmisión. Por el contrario, al usar métodos de transmisión orientados a caracteres, como la transmisión serie asíncrona, la sincronización de trama debe ser resuelta por algún procedimiento adicional.” (neo.lcc.uma.es, 2019) Ya más adelante profundizando se definirá el tipo de detección inicial y final de cada trama, que puede ser conteo de caracteres, secuencia de bits entre otras.

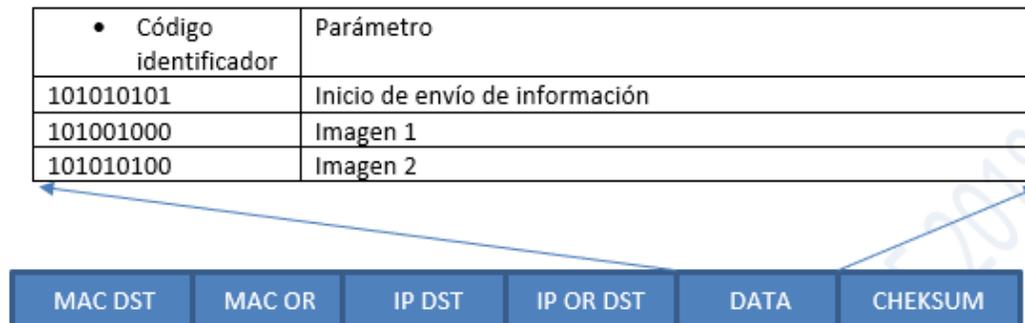


Figura 1. Ejemplo de un formato de trama de datos (neo.lcc.uma.es, 2019).

Marco conceptual

Sistema de adquisición de datos.

La adquisición de datos (DAQ) es el proceso de medir mediante un software de una PC un cierto tipo de información, pudiendo ser un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido, así como también información digital. Un sistema DAQ consta o de sensores recopiladores o principales, hardware de medidas DAQ y una PC con software programable. Comparados con los sistemas de medidas tradicionales, los sistemas DAQ basados en PC aprovechan la potencia del procesamiento, la productividad, la visualización y las habilidades de conectividad de las PCs estándares. (National Instruments, 2019)

Dispositivo DAQ.

El hardware DAQ actúa como la interfaz entre una PC y señales del mundo exterior. Funciona principalmente como un dispositivo que digitaliza señales analógicas o digitales entrantes para que una PC pueda interpretarlas. Los tres componentes clave de un dispositivo DAQ usado para medir una señal son el circuito de acondicionamiento de señales, convertidor analógico-digital (ADC) y un bus de PC. (National Instruments, 2019)

Convertidor Analógico Digital (ADC).

Las señales analógicas de los sensores deben ser convertidas en digitales antes de ser manipuladas por el equipo digital como una PC. Un ADC es un chip que proporciona una representación digital de una señal analógica en un instante de tiempo. En la práctica, las señales analógicas varían continuamente con el tiempo y un ADC realiza "muestras" periódicas de la señal a una razón predefinida. Estas muestras son transferidas a una PC a través de un bus, donde la señal original es reconstruida desde las muestras en software. (National Instruments, 2019)



Figura 2. Convertidor analógico digital. (National Instruments, 2019).

Acondicionamiento de señales.

Las señales de los sensores o del mundo exterior pueden ser ruidosas o demasiado peligrosas para medirse directamente. El circuito de acondicionamiento de señales manipula una señal de tal forma que es apropiado para entrada a un ADC. Este circuito puede incluir amplificación, atenuación, filtrado y aislamiento. Algunos dispositivos DAQ incluyen acondicionamiento de señales integrado diseñado para medir tipos específicos de sensores. (National Instruments, 2019)

Almacenamiento de datos.

“El almacenamiento de datos es el proceso mediante el cual la tecnología de la información archiva, organiza y comparte los bits y bytes que conforman los sistemas de los que dependemos todos los días, desde las aplicaciones hasta los protocolos de red, los documentos, el contenido multimedia, las libretas de direcciones y las preferencias del usuario.” (RedHat, 2019)

Transmisión de la información.

Un medio de transmisión es el canal que permite la transmisión de información o datos, digitales o analógicos, entre dos terminales de un sistema de transmisión. La transmisión se realiza habitualmente empleando ondas electromagnéticas que se propagan a través del canal. A veces el canal es un medio físico y otras veces no, ya que las ondas electromagnéticas son susceptibles de ser transmitidas por el vacío.

Pérdida de paquetes.

Es el porcentaje de paquetes transmitidos que se descartan en la red, debido a una alta tasa de error en alguno de los medios de enlace o por sobrepasarse la capacidad de un buffer de una interfaz en momentos de congestión. (sarenet.es, 2019)

Marco legal

Leyes sobre las comunicaciones inalámbricas y su regulación

Norma 802.11.

Ahora hablemos sobre la norma IEEE 802.11, esta es la norma que regula todo lo que se trata de redes inalámbricas locales (WLAN), esta norma regula en general tres aspectos importantes, la frecuencia, velocidad y alcance que tiene la red inalámbrica local.

Se puede agregar que en la actualidad las normas a,b,g y n trabajan juntas sin conflictos. Otro punto a destacar es que con ayuda de CSMA/CD (Acceso múltiple con escucha de portadora y detección de colisiones) que es un sistema que permite que los datos no colisionen unos a otros dentro de la red inalámbrica. Otro aspecto importante es la cantidad de accesos, el cual puede aceptar un número de 25-30 personas (o más) antes de que se degrade la conectividad, aunque esto depende del equipamiento y distribución de la red por medio de otros puntos de acceso o servidores.

Norma 802.15.

Fue diseñada con el fin de lograr una transferencia eficiente y rápida de datos en WPAN's (Wireless Personal Area Network o redes inalámbricas de área personal). Existen diferentes tipos de esta red en la que están incluidas Bluetooth y Zigbee. Pero nos enfocaremos en la primera por ser la más utilizada en la actualidad. Entre los aspectos más importantes que ve podemos mencionar la transferencia, el alcance por tipo y la frecuencia. Bluetooth trabaja bajo la norma 802.15.1 y sus estándares están orientados a velocidades de hasta 16Mbps, trabaja en las frecuencias de 300MHz, 2.4 GHz y 3 GHz, finalmente hay tres tipos de redes inalámbricas bluetooth, tipo 3 que abarca 1 metro, tipo 2 que abarca 10 metros (Esta es ampliamente utilizada) y tipo 1 que abarca 100 metros. También se puede destacar otro punto que es el acceso de usuarios el cual es de alrededor de 8 conectados.

Norma 802.16.

Estándar creado para las WMAN (Redes inalámbricas de área metropolitana), con el nombre de WiMAX, permite accesos de unos 50 a 80 kilómetros y velocidades que pueden llegar a 1Gbps, funcionan en las frecuencias de 2 hasta 11 Ghz y 11 a 66 Ghz. Se puede dar a entender WiMAX con una gran red inalámbrica constituida de pequeñas WLAN. (Fabio Asecio, 2019)

Bandas de frecuencia de libre utilización - normativa vigente**Resoluciones 2016.**

Agencia nacional del espectro – ANE. Resolución 711 de 2016 ANE - Por la cual se establecen las bandas de frecuencia de libre utilización dentro del territorio nacional y se derogan algunas disposiciones.

Bandas de frecuencia, cuadro nacional de atribución de leyes 2009. Ley 1341 de 2009 - Por la cual se definen principios y conceptos sobre la sociedad de la información y la organización de las tecnologías de la información y las comunicaciones -tic-, se crea la agencia nacional de espectro y se dictan otras disposiciones; arts. 4o. núm. 7o.; 17 núm. 4o.; 18 num. 6o., 7o.

Resoluciones 2013.

Agencia nacional del espectro – ANE. Resolución 357 de 2013 ane - Por la cual se cambia la atribución de los rangos de frecuencia 894-905mhz y 942.5-950mhz y se dictan otras disposiciones.

Ministerio de tecnologías de la información y las comunicaciones 2010. Resolución 129 de 2010 mtic - Por la cual se establecen medidas en materia de ordenación técnica del espectro radioeléctrico, se actualiza el cuadro nacional de atribución de bandas de frecuencias y se adopta su contenido. (Ministerio de Telecomunicaciones, 2019)

Metodología

Para este proyecto, es acertado hablar de una metodología cuantitativa, ya que, debido a las características de esta, contribuirá al que el resultado del análisis sea más conciso y con datos acertados respecto a las características del comportamiento de la red. Esta metodología se basa en el estudio de las propiedades y los fenómenos cuantitativos y su variabilidad para la formulación de una hipótesis y posterior teoría, mediante modelos matemáticos que competen a los eventos y fenómenos, representados en el análisis de datos obtenidos de las pruebas realizadas en la red.

Más específicamente, dentro de esta metodología, se encuentra una investigación analítica, que consiste en establecer la comparación de datos obtenidos y sus variables en diferentes grupos de estudio, estudiándolas para la proposición de la hipótesis. Planteado lo anterior se procede con un esquema a grandes rasgos, de la estructura metodología que se tendría que emplear, teniendo en cuenta los requerimientos que demandaría el sistema.

- Diseño de sistema de almacenamiento y transmisión de datos, el cual permita el envío de la trama de imágenes sobre la banda de 2,4 Ghz, que permita la conectividad con dos puntos de manera simultánea para transmitir las imágenes almacenadas en el sistema de adquisición de datos con un tamaño promedio de 62,6 KB,
- Programación o implementación del sistema de adquisición de datos con su transmisor.
- Generación de conectividad entre los dos puntos de la red inalámbrica en una zona con un rango de distancias de pruebas unitarias, de ambos nodos a diferentes distancias en cuanto a transmisión y pérdida de datos.

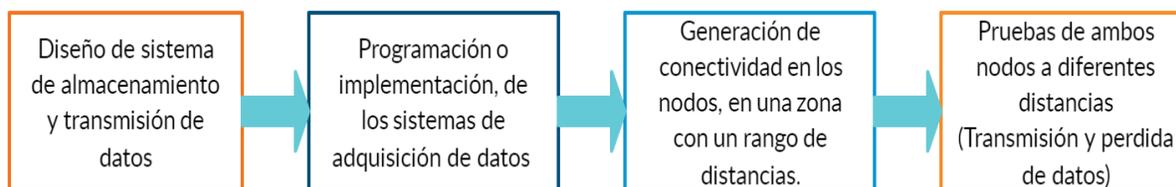


Figura 3. Esquema metodológico. (Autoría propia, 2020).

Desarrollo

Requerimientos del proyecto

Es factible el planteamiento de un esquema principal que servirá para impartir una guía durante el desarrollo del proceso de construcción del sistema. Por ello en la siguiente figura, se establece el diagrama de lo que sería el proyecto y sus componentes, los cuales se irán desglosando a lo largo del documento.

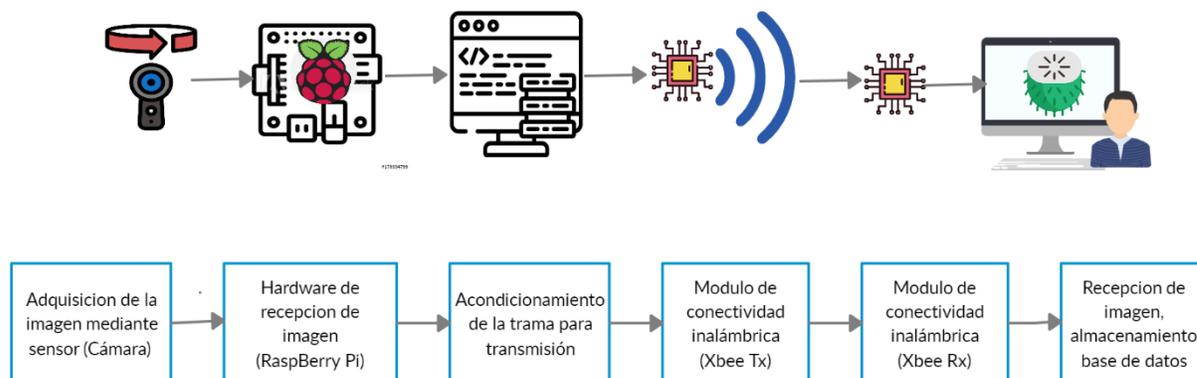


Figura 4. Esquema general. (Autoría propia, 2020).

Partiendo de las condiciones de siembra de un cultivo de guanábana, se tiene que estos cultivos en la región de Colombia, donde se conoce que según estadísticas, gran parte o la mayoría de las áreas sembradas están en los departamentos de Tolima, Valle, Santander, Cundinamarca, Huila, Antioquia, en su mayoría son cultivos de pequeños productores, “Al mercado externo va cerca de 25% de la producción, básicamente a Estados Unidos y es precisamente pionero en el país en el negocio de procesar y exportar la fruta”. Dicho lo anterior, entonces se debe sembrar la fruta entre los 800 y 1.000 metros sobre el nivel del mar en zonas de bajo régimen de lluvia. (La República, 2015).

Estos cultivos, por lo general son cultivos caracterizados por tener un tardío rendimiento y aunque comúnmente un árbol puede durar hasta 30 años produciendo de manera efectiva, luego de sembrado inicialmente, una primera cosecha tarda al menos 4 años después. Siempre y cuando tengas las condiciones propicias de índole climática, mantenimiento y fertilización, al quinto año ya produce aproximadamente un 70% y al sexto un 100%. (La República, 2015)

En un enfoque al proyecto en cuestión, es de saberse entonces que el monitoreo debe realizarse en cultivos en los que ya el árbol este establecido como productor, posterior a la primera cosecha, en fase de crecimiento y de maduración, el punto crítico en presencia de plagas. En esta estancia

se recalca entonces que según el territorio colombiano y su caracterización climatológica, la distribución de floración en los cultivos de guanábana se presenta principalmente durante dos épocas del año, la primera de ellas se da en los meses de Febrero hasta Mayo y la segunda del mes de Octubre al de Noviembre, lo que consecuentemente se ve reflejado en la cosecha de la primera floración, que se recolecta en los meses de Noviembre hasta Enero y la segunda cosecha recolectada en el mes de Mayo. (Miranda, Barragan, Barreto, & Caicedo, 2020)

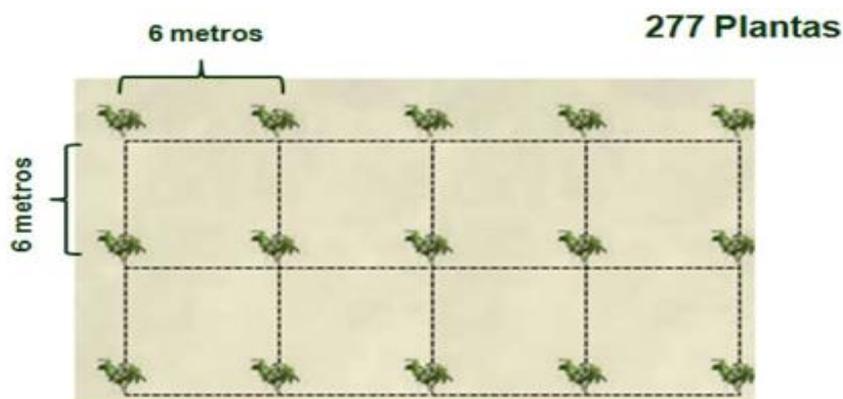
La oportuna identificación de las plagas de la guanábana, contribuirá al óptimo desarrollo y producción del fruto, para ello a continuación se relaciona cuáles son las más importantes, las que requieren de especial atención, y que son visualmente identificables de manera física en la planta.

“Las más considerables plagas que padece la guanábana son las siguientes: La Broca del fruto (*Cerconata anonella*), controlable eliminando los frutos atacados, embolsando el resto de frutos y aplicando insecticidas a las flores y frutos; La Broca del tallo (*Croatosomus bombina*) permanece dentro del tronco y se puede controlar aplicando una pasta de sulfato de cobre, azufre y diazinón, sellando los orificios en el tallo; El Perforador de las semillas (*Bephrata naculicollis*) se controla recolectando y enterrando los frutos atacados y aplicando una mezcla de Malathion y Dipetrix tan pronto como aparezcan los frutos. Entre las principales enfermedades, la más importante es la antracnosis del follaje y del fruto, causada por *Colletotrichum gloeosporioides*, al que se previene con aplicaciones alternadas de oxiclورو.” (encolombia, 2020)



Figura 5. Broca del fruto (*Cerconata anonella*); Broca del tallo (*Croatosomus bombina*); Perforador de las semillas (*Bephrata naculicollis*) respectivamente. (encolombia, 2020)

Además de tener en cuenta factores como los anteriores, en cuanto a especificaciones técnicas, se debe tener en cuenta la distribución de cada planta y la distancia que se debe tener en cuenta entre una y otra, además de la estructura a implementar, adecuada a la necesidad, una planta puede crecer en promedio 5 a 6 metros de altura; en la figura 8 se muestra la distribución habitual, con un promedio de 6 metros entre cada planta, y el cálculo necesario dependiendo del terreno a emplear.



Formula de Calculo: N° plantas por hectárea

$$n = Su \text{ m}^2 / (d * d)$$

Donde:

n = número de plantas.

Su = superficie del campo, en metros cuadrados (m²).

d = distancia entre plantas, en metros (m), multiplicada por sí misma.

Figura 6. Distribución de plantas del cultivo (Ecuguanábana, 2020)

Dicho lo anterior, realizando un esquema de la ubicación de la cámara, el sensor utilizado para capturar la información o imágenes, se ubicaría en el centro de cada sección en un riel de material metálico cubierto con esmalte de poliuretano para evitar su corrosión, sobre el cual se movilizará la estructura de la cámara, como se ve a continuación. Con una altura promedio de 6 metros, puesto que el árbol llegara a medir probablemente su máximo de altura.

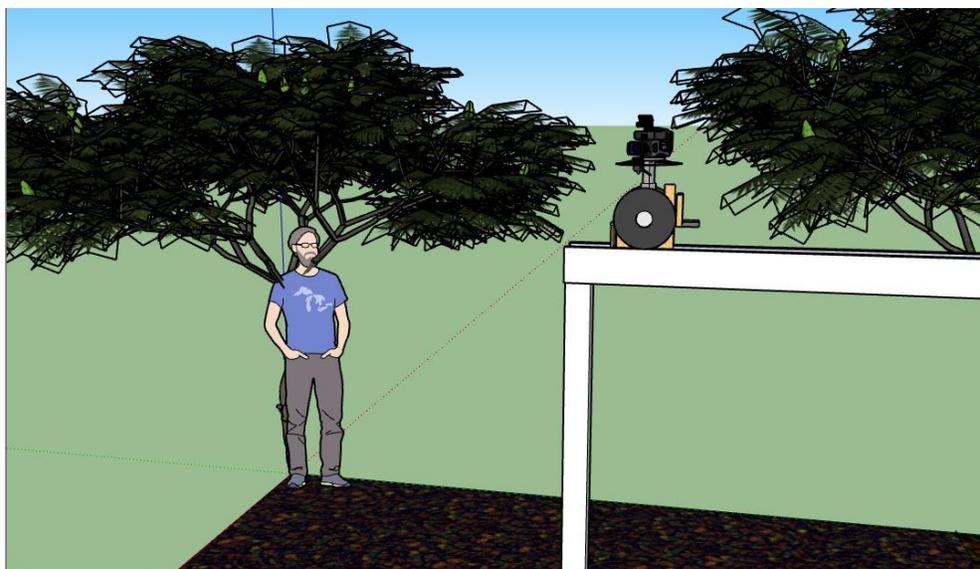


Figura 7. Esquema hipotético del sistema de cámara. (Autoría propia, 2020).



Figura 8. Esquema hipotético del sistema de cámara. (Autoría propia, 2020).

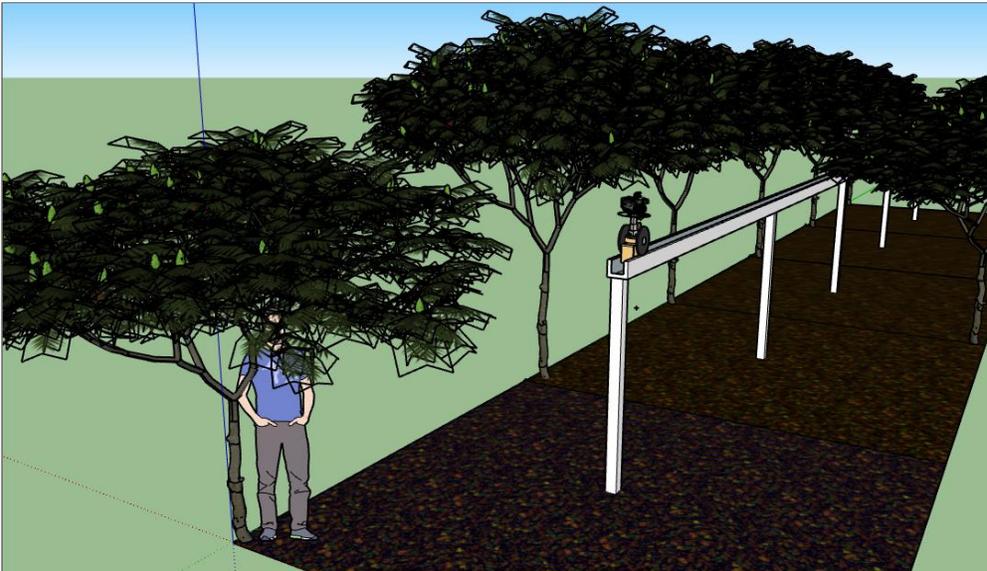


Figura 9. Esquema hipotético del sistema de cámara. (Autoría propia, 2020).



Figura 10. Esquema hipotético del sistema de cámara. (Autoría propia, 2020).

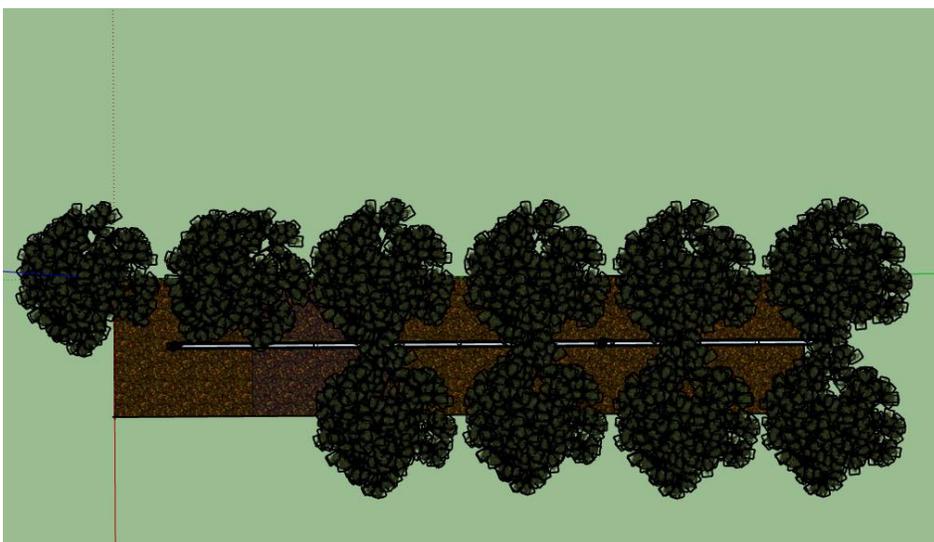


Figura 11. Esquema distribución del cultivo. (Autoría propia, 2020).

Para lograr la identificación de la cámara a emplear, fue necesario realizar un análisis del hardware de adquisición, el cual se describe en el apartado siguiente.

Llegando hacia el desarrollo efectivo del sistema, es imperativo indagar acerca de la mejor placa de desarrollo a emplear, puesto que las funcionalidades y características pueden variar, lo que puede facilitar el uso de herramientas de estas placas en pro del resultado que se desea obtener y su manera de gestionar la información.

Por ello, se analizaron las placas Raspberry Pi, Arduino y Beaglebone Black. Aunque existe una amplia gama de placas base para desarrollo, tales como i.MX8, Jetson TX2 o la Samsung ARTIK

710, (Wendt, 2018) según especificaciones y facilidad, se analizaran las primeras tres mencionadas. Estas plataformas, al permitir el desarrollo de aplicaciones usando electrónica programable constituyen la tendencia de construcción de sistemas con fácil alcance ya sea mediante interfaz gráfica como el Raspberry Pi semejante a una microcomputadora o el Arduino, microcontrolador diseñado para un control orientado a la electrónica externa y el Beaglebone una mezcla de los dos anteriores. (Casco, 2014) Mediante la siguiente tabla, se pretenden analizar los factores determinantes de la elección, para este caso del Raspberry Pi.

Tabla.1.

Comparativa general placas de desarrollo.

Placa	Características	Ventajas	Desventajas
Raspberry Pi	Diseñada como una microcomputadora que usa variados tipos de lenguaje de programación de alto nivel como Phyton, C++ y Java. Posee relativo alto desempeño,	Posee bastante versatilidad en cuanto a su uso, como la posibilidad de uso de puertos USB y HDMI, con una cómoda interfaz gráfica para usos multimedia. El procesador grafico que posee, soporta 1080p; su conexión a internet la hace más versátil al igual que su sistema operativo el cual puede acomodarse a la necesidad de visualización y manejo. Es bastante adaptable con shields.	No posee muchas opciones para la conexión de interfaces con sensores externos, por lo que no es la opción ideal para proyectos esencialmente electrónicos.
Arduino	Plataforma de tipo open source, compuesto por un	Además de su estandarte Arduino	No posee como tal una interfaz gráfica, es una

	<p>microcontrolador software y hardware, posee su propio lenguaje (APL) todos los archivos relacionados a sus aplicaciones están disponibles para uso personal o comercial, ya que no se requiere ningún tipo de licencia para su desarrollo. Se fundamenta en código escrito el cual al ser compilado se traduce en instrucciones entendibles almacenadas en el microcontrolador</p>	<p>Uno, posee una amplia gama de modelos a elegir. Como es de bajo consumo, es ideal para aplicaciones de usos de larga duración, o incluso para uso de baterías. El aspecto más importante es la gran facilidad de encontrar apoyo, documentación sobre proyectos particulares, tutoriales; presenta flexibilidad para distintos tipos de interfaces.</p>	<p>plataforma para principiantes. Normalmente no puede manejar varios procesos al mismo tiempo, por su capacidad y simplicidad, no es muy bueno con proyectos de mayores necesidades de procesamiento.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Beaglebone Black	Posee una serie de capas, que son placas de expansión multifuncionales, cada una con funciones específicas según aplicación requerida. Está equipada para la recepción de conexiones de equipos electrónicos externos, lo cual hace que en su hardware existan multiconexiones con varios tipos de puertos y mayor número de pines. Al no poseer una interfaz de escritorio gráfica, esta también diseñada para desarrolladores de nivel.	Es completamente operativo desde el inicio, puesto que posee ya su sistema operativo. Tiene variedad de entradas y salidas, 69 pines. Tiene opciones de conexión SPI bus, temporizadores, puertos seriales, salidas PWM y entradas analógicas. Posee un procesador de 1 GHz	No posee codificación de video lo que no la hace la mejor opción de minicomputadora; posee limitado número de puertos USB. Toda la documentación referente a proyectos, ejecución, referentes y demás no son muy sencillos de encontrar. Comparando, Pi posee 13 veces más de tráfico en la red.
------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Nota. Información tomada de (Casco, 2014)

Debido a factores vistos anteriormente, la correcta elección para la ejecución del sistema, es el Raspberry Pi3, debido a su flexibilidad y considerable robustez para proyectos relacionados con multimedia y su procesamiento, por lo cual, los esfuerzos y análisis se verán centrados en emplear este sistema de placa para la recepción y tratamiento de las imágenes del cultivo. Definido el hardware de adquisición de datos, el sensor o cámara a emplear fue el de un cámara Pixy 2 CMUcam5, que tiene la posibilidad de incluir un Pan/Tilt Servo Motor Kit para Pixy cmucam que consiste en un dispositivo electrónico robot, para rotación 360 grados con la cámara, para la captura de imágenes a su alrededor.



Figura 12. Pixy 2 CMUcam5 (SandoRobotics, 2020)



Figura 13. Pan/Tilt Servo Motor Kit (Amazon, 2020).

Además de la comparación en cuanto a las placas, se debe recalcar el sistema de conectividad inalámbrica a usar para la transmisión de la información, que para este caso serán los módulos Xbee Pro S1 configurados para obtener la comunicación punto a punto deseada, para lo cual tenemos las siguientes características.

PERFORMANCE		
RF DATA RATE	250 kbps	250 kbps
INDOR/URBAN RANGE	100 ft (30 m)	300 ft (100 m)
OUTDOOR/RF LINE-OF-SIGHT RANGE	300 ft (100 m)	1 mi (1.6 km)
TRANSMIT POWER	1 mW (+0 dBm)	60 mW (+18 dBm)*
RECEIVER SENSITIVITY (1% PER)	-92 dBm	-100 dBm
DIGI HARDWARE	S1	
TRANSCEIVER CHIPSET	Freescale MC13212	
FEATURES		
SERIAL DATA INTERFACE	3.3V CMOS UART	
CONFIGURATION METHOD	API or AT Commands, local or over-the-air	
FREQUENCY BAND	2.4 GHz	
INTERFERENCE IMMUNITY	DSSS (Direct Sequence Spread Spectrum)	
SERIAL DATA RATE	1200 bps - 250 kbps	
ADC INPUTS	(6) 10-bit ADC inputs	
DIGITAL I/O	8	
ANTENNA OPTIONS	Chip, Wire Whip, U.FL, & RPSMA	

Figura 14. Características modulo Xbee Pro S1 (DIGI, 2018).

NETWORKING & SECURITY		
ENCRYPTION	128-bit AES	
RELIABLE PACKET DELIVERY	Retries/Acknowledgments	
IDS AND CHANNELS	PAN ID, 64-bit IEEE MAC, 16 Channels	
POWER REQUIREMENTS		
SUPPLY VOLTAGE	2.8 - 3.4VDC	2.8 - 3.4VDC
TRANSMIT CURRENT	45 mA @ 3.3VDC	215 mA @ 3.3VDC
RECEIVE CURRENT	50 mA @ 3.3VDC	55 mA @ 3.3VDC
POWER-DOWN CURRENT	<10 uA @ 25° C	

Figura 15. Características modulo Xbee Pro S1 (DIGI, 2018).

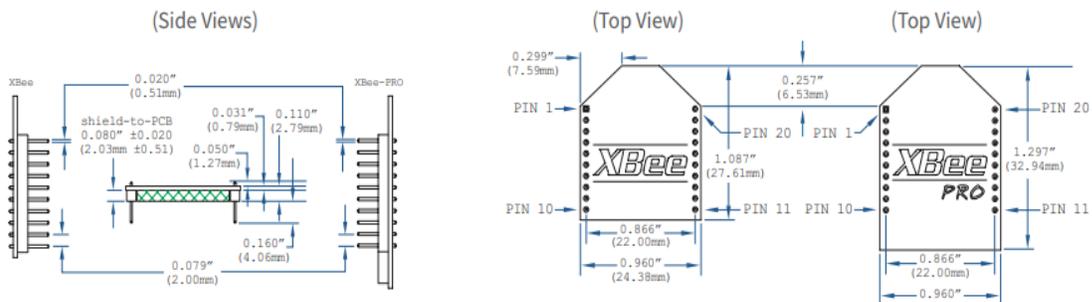


Figura 16. Características físicas modulo Xbee Pro S1 (DIGI, 2018).

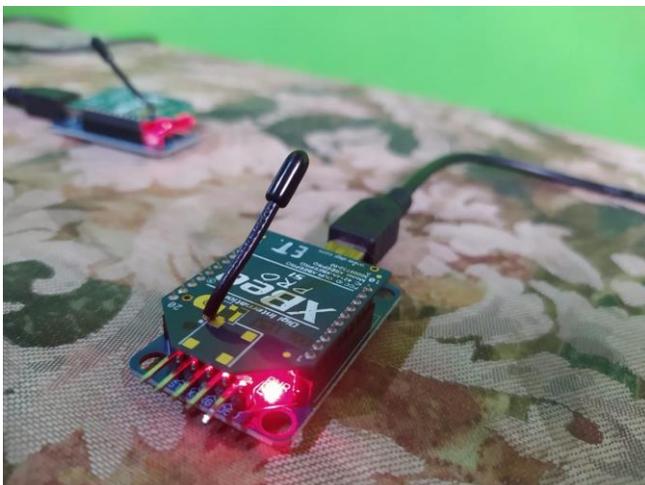


Figura 17. Aspecto real Xbee Pro S1 y adaptador Xbeexplorer. (Autoría propia, 2020).

Para llevar a cabo la instalación de los motores servos, que son aquellos motores de corriente continua que permiten hacer un barrido a un ángulo determinado, involucrando movimiento en el eje horizontal y en el eje vertical, se deben tener en cuenta las siguientes apreciaciones.

Se procede inicialmente con la herramienta raspbian-config la cual se instala mediante código: `sudo raspbian-config`

Posteriormente mediante “Opciones de interfaz” > “Cámara”. “¿Desea que la interfaz de la cámara esté habilitada?” Si. > Shutdown.

Se debe tener en cuenta que la cámara debe establecerse en los ángulos de 45°, 135°, 225° y 315°, estableciendo intervalos de pausa de aproximadamente de 1 segundos entre cada cambio de rotación. Por lo cual se debe calcular su DutyCycle o ciclo de trabajo, teniendo en cuenta que un servo de rotación normal que son los más comunes y económicos, tendrá promedios anchos (PW) como estos ("Chito", 2017):

- a) PW= 1ms mover a 0 grados
- b) PW= 1,5 ms mover a 90 grados
- c) PW= 2 ms mover a 180 grados

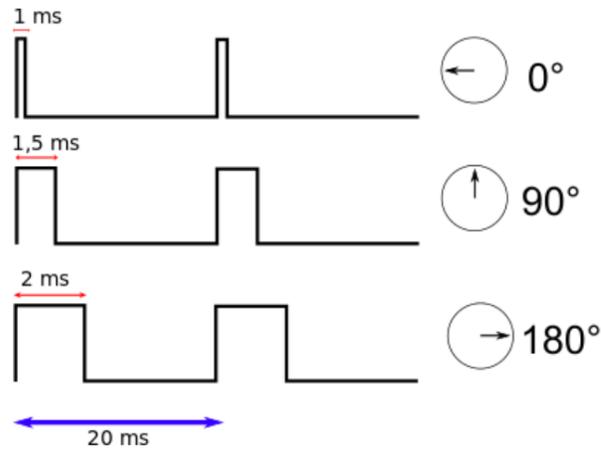


Figura 18. Equivalencia en PW ("Chito", 2017).

Lo que en relación nos arrojaría una equivalencia para cada grado de $45^\circ = 0,75\text{ms}$; $135^\circ = 2,25\text{ms}$; $225^\circ = 3,75\text{ms}$; $325^\circ = 5,25\text{ms}$ de tiempo de pulso en alto.

Teniendo en cuenta que estos cálculos son realizados bajo la premisa de que el motor trabaje con un periodo de 20ms, es decir a una Frecuencia de 50Hz, se procede con el cálculo de su DutyCycle o Ciclo de trabajo, determinado por:

$$DC = PW \times F$$

Para el ángulo de 45°

$$DC = 0,75 \times 10^{-3} \cdot 50 \cdot 100 = 3,75 \quad (1)$$

Para el ángulo de 135°

$$DC = 2,25 \times 10^{-3} \cdot 50 \cdot 100 = 11,25 \quad (2)$$

Para el ángulo de 135°

$$DC = 3,75 \times 10^{-3} \cdot 50 \cdot 100 = 18,25 \quad (3)$$

Para el ángulo de 135°

$$DC = 5,25 \times 10^{-3} \cdot 50 \cdot 100 = 26,25 \quad (4)$$

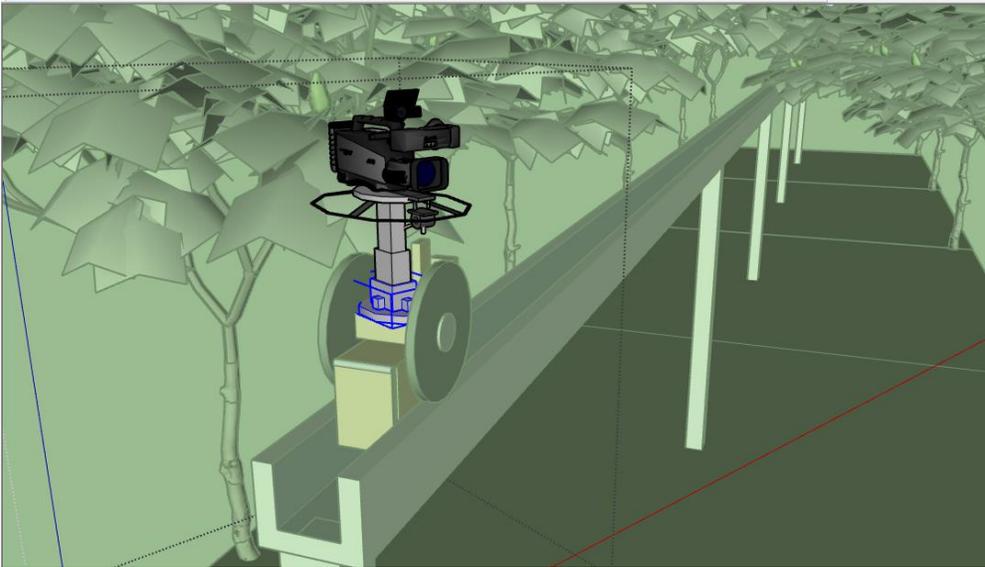


Figura 19. Esquema hipotético de la cámara. (Autoría propia, 2020).

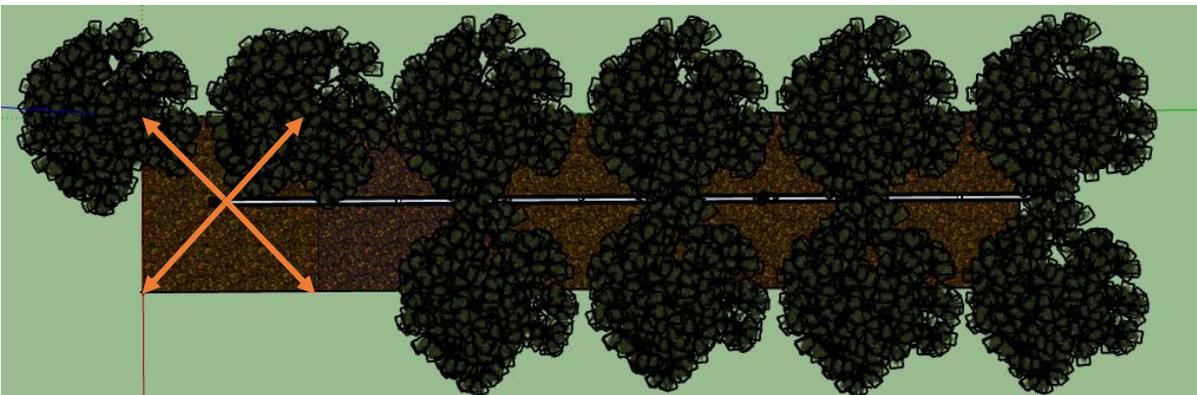


Figura 20. Esquema de puntos estratégicos. (Autoría propia, 2020).

De manera consecuente con lo anterior, se tiene que el código tendría una estructura similar a la evidenciada a continuación, donde se debe realizar el cálculo empleando el ciclo de trabajo o Duty Cycle calculado anteriormente, lo que nos terminara la posición del servo motor en función del ángulo necesario.

```
import RPi.GPIO as GPIO
```

```
import time
```

```

servoPIN = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50) # GPIO 17 for PWM with 50Hz
p.start(0) # Initialization
try:
    while True:
        p.ChangeDutyCycle(3,75)
        time.sleep(1)
        p.ChangeDutyCycle(11,25)
        time.sleep(1)
        p.ChangeDutyCycle(18,25)
        time.sleep(1)
        p.ChangeDutyCycle(26,25)
        time.sleep(1)
except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
("Chito", 2017)

```

Entonces, se debe establecer que, dentro de la estructura móvil, la que va a ir sobre el riel, debe ser capaz de contener al Raspberry, la cámara Pixie2, el Xbee Pro S1, un distribuidor de conexiones y el regulador de corriente. La conexión eléctrica hacia esta caja, se realizará por medio de un cable de poder retráctil, capaz de estirarse o contraerse dependiendo del movimiento necesario; esto consecuentemente lleva a estipular el origen de la alimentación, que será otra estructura ubicada estratégicamente cerca del riel, que contendrá un panel solar en el exterior conectado hacia una batería y un regulador de corriente. En la imagen siguiente se podrá apreciar el aspecto y ubicación de las estructuras en cuestión.

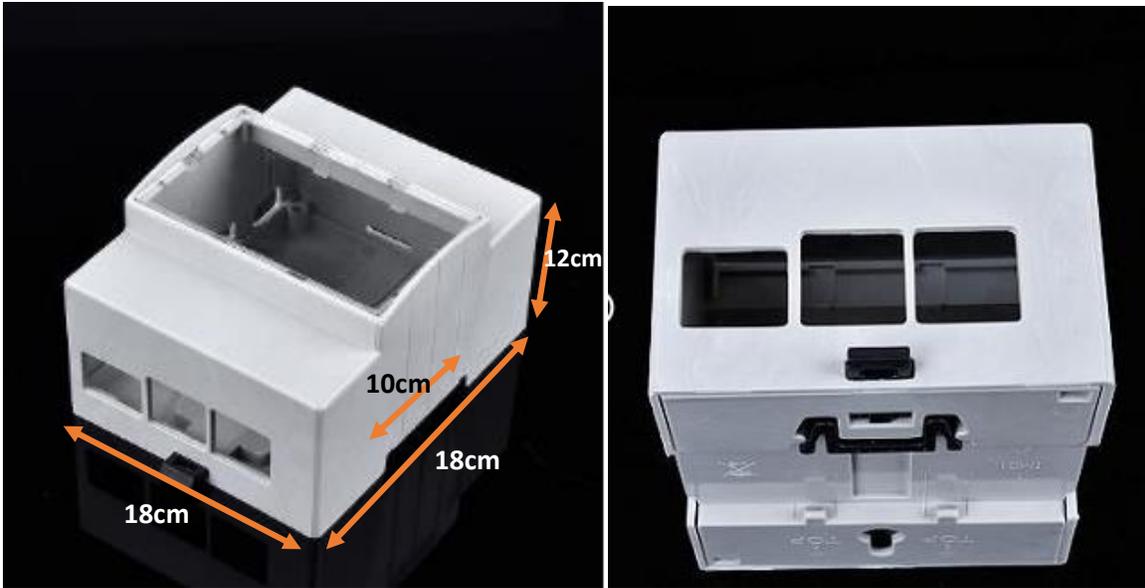


Figura 21. Caja Raspberry (DID Electrónica, 2020).



Figura 22. Perspectiva de estructura de cámara hipotética. (Autoría propia, 2020).

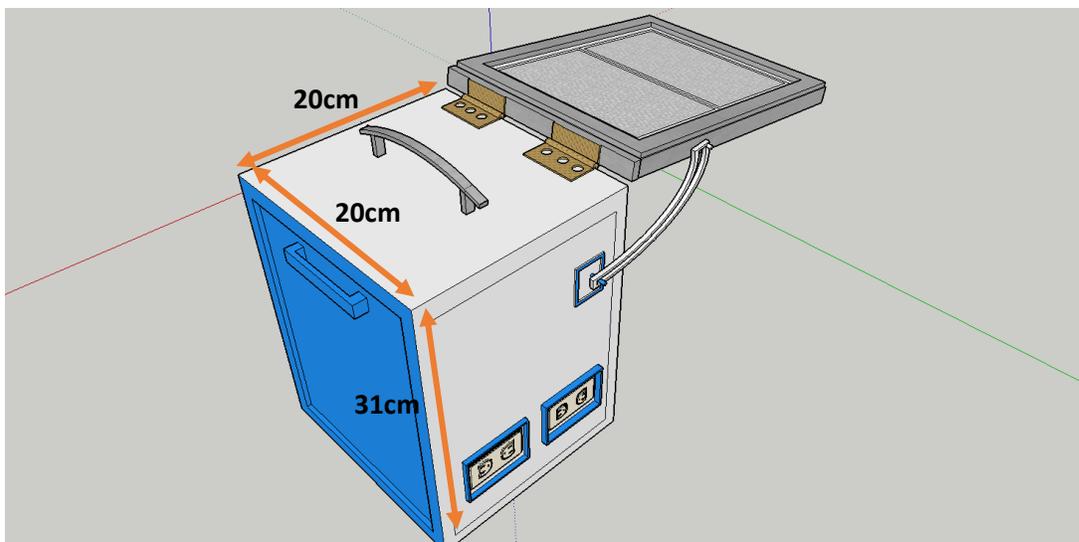


Figura 23. Estructura punto origen de alimentación (Sujeto a cambios por dimensiones panel).
(Autoría propia, 2020).

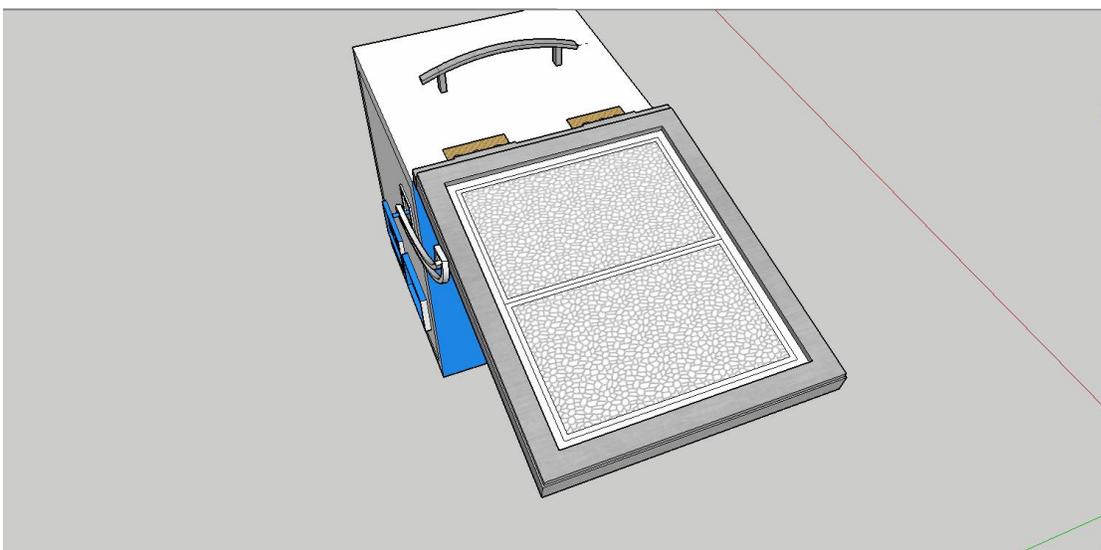


Figura 24. Estructura punto origen de alimentación. (Autoría propia, 2020).

En cuanto al tema de consumo de potencias, se recopiló información de los sistemas vitales que determinaran las características eléctricas del punto origen de alimentación, se debe tener en cuenta que el consumo de los elementos (KWh) se calculó a partir de la premisa de que cuando el sistema requiera su total funcionamiento será por 6 horas al día en promedio, a excepción del Raspberry el cual requiere de un consumo continuo, es decir por 24 horas, además se resalta la importancia del uso de un regulador de corriente. A continuación, se evidencian las relaciones de potencia.

Tabla 2.

Tabla de potencias y consumo

	Voltaje	Corriente	Potencia	Consumo
Cámara	9v	140mA	1,26 W	7,56KWh
	(Independiente)			
	5v	140mA	0,7 W	4,2KWh
Raspberry pi 3 modelo B	5v	2,5A	12,5W	300KWh
Pan/Tilt Servo Motor Kit	9v	1,5A	13,5W	81KWh

Nota. Relación de potencias y consumo de elementos. (Autoría propia, 2020).

Teniendo conocimiento de lo anterior, se puede entonces determinar que se requiere en el punto origen de alimentación, un panel con características para la generación de 13W, 9v y 2,5A, en promedio, lo que significa una batería de mismas características para el almacenamiento cuando el panel no esté en funcionamiento, y la energía solo sea consumida por el Raspberry.

Una parte indispensable de este sistema de la cámara, es el riel, el cual debe emplear un mecanismo de polea y motor, en el que se cumpla la funcionalidad del movimiento uniforme rectilíneo, donde se debe tener en cuenta una velocidad promedio de 3 Kmph para el movimiento de la cámara, y recorridos de a 12 metros puesto que en estos puntos es donde se tomaran las fotos. Por ende y teniendo en cuenta estos valores, se debe realizar un código capaz de controlar el motor de manera que se ubique una vez que se active, cada 12 metros. Para ello se realizó el cálculo del motor en relación con esta distancia y el tiempo que tardaría en recorrer hasta llegar a cada uno de los puntos.

$$Velocidad = 3 \frac{Km}{h} = 0.83 \frac{m}{s} \quad (5)$$

$$Ratio = 2,5 \text{ cm} \quad (6)$$

$$w = \frac{0.833m/s}{0.025m} = 33.33 \text{ rad/s} \quad (7)$$

$$T = \frac{2\pi rad}{33.33 rad/s} = 0.188s \quad (8)$$

$$F = \frac{1}{0.188s} = 5.301Hz \quad (9)$$

$$t = \frac{12m}{0.833m/s} = 14.4s \quad (10)$$

Lo cual no arroja como resultado la necesidad de instaurar un motor de 5,301cps, que de no ser comercial requeriría de la adecuación mediante resistencias, con un radio de 2,5cm, que con una velocidad de 3Km/h tardara en realizar un recorrido cada 12 metros en un total de 14.4 segundos.

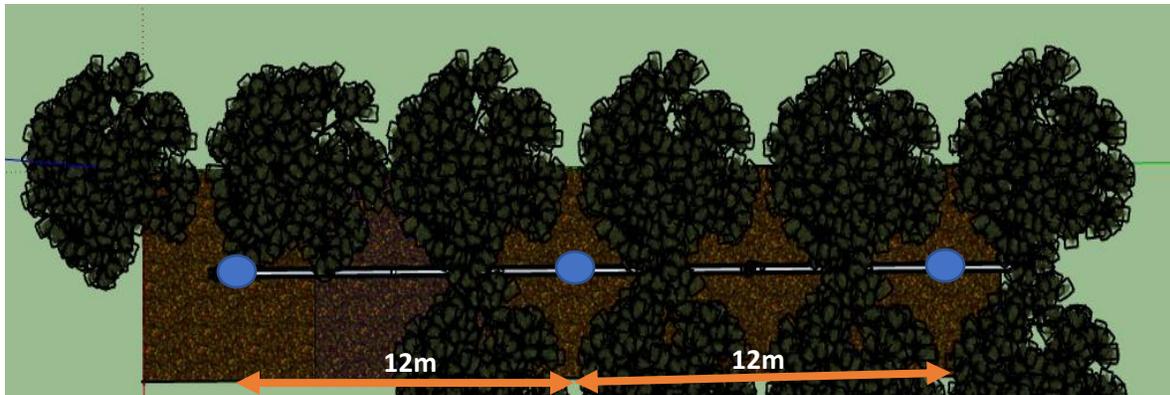


Figura 25. Visualización puntos estratégicos en recorrido del riel. (Autoría propia, 2020).

De acuerdo a lo anterior, se establece la necesidad de aclarar, que cuando la cámara llegue al final del recorrido de su riel, realizará la transmisión de la imagen mediante el módulo Xbee, que estará en línea de vista con su receptor.

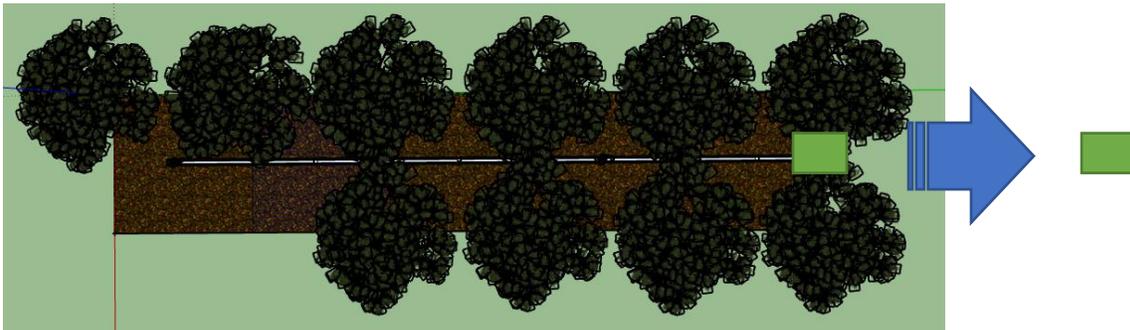


Figura 26. Esquema de transmisión módulos Xbee. (Autoría propia, 2020).

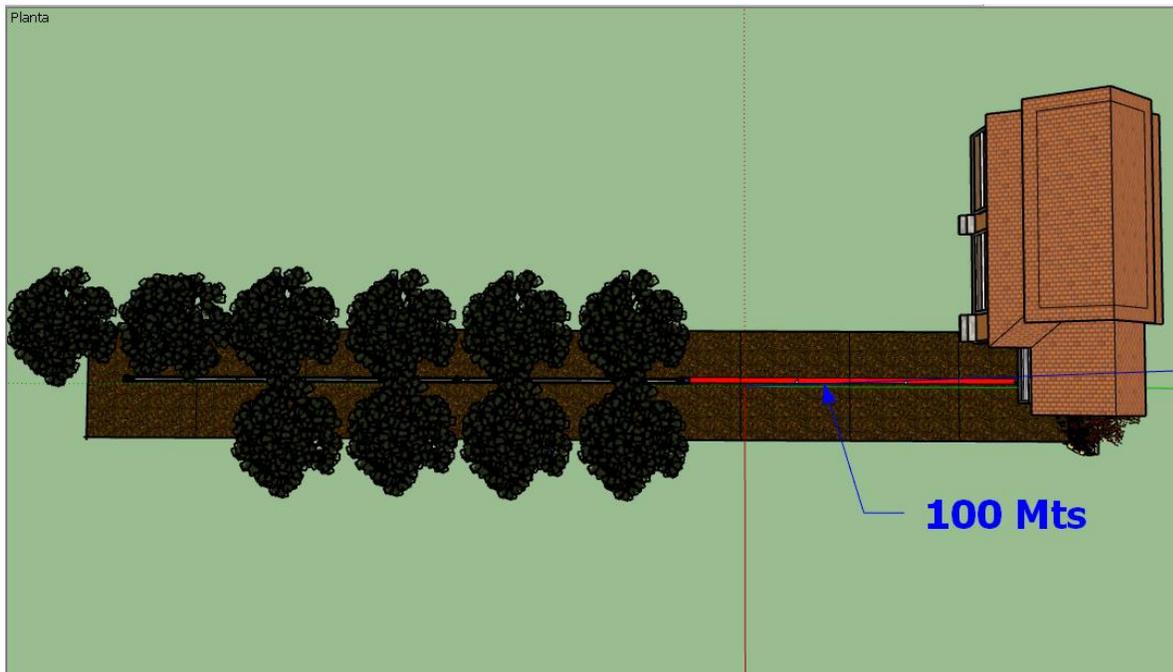


Figura 27. Esquema enlace punto a punto y distancia (Autoría propia, 2020).



Figura 28. Esquema enlace punto a punto y distancia, perspectiva. (Autoría propia, 2020).



Figura 29. Esquema locación de receptor, punto cercano al cultivo. (Autoría propia, 2020).

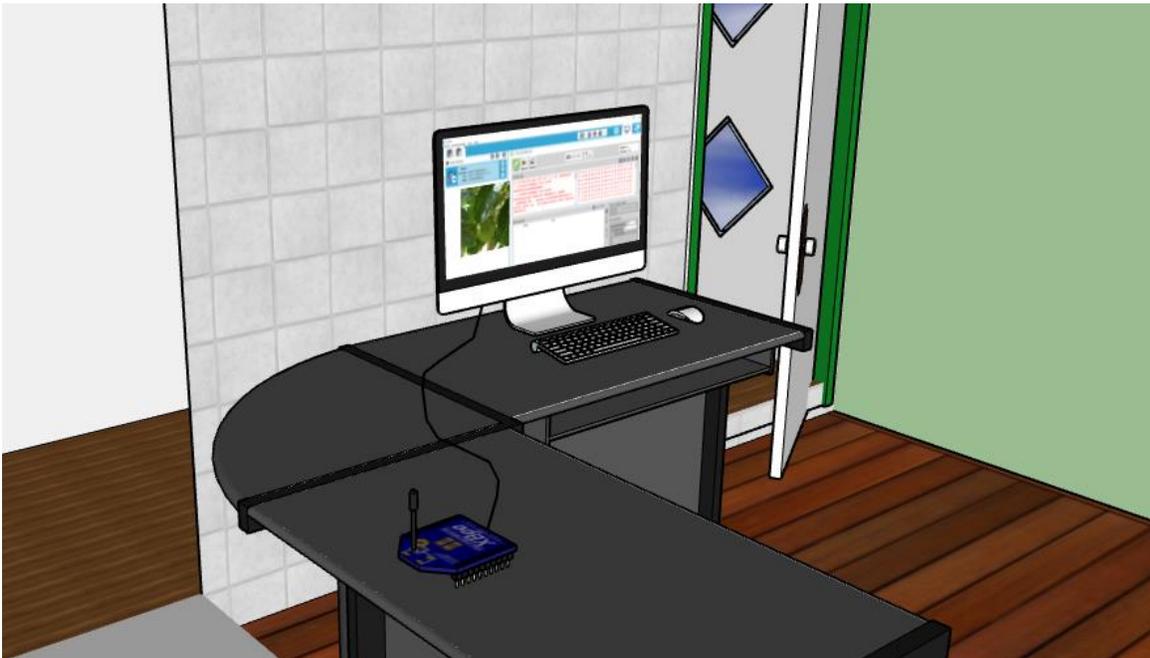


Figura 30. Punto de ubicación receptor. (Autoría propia, 2020).



Figura 31. Punto de ubicación receptor, perspectiva de visión hacia el cultivo. (Autoría propia, 2020).

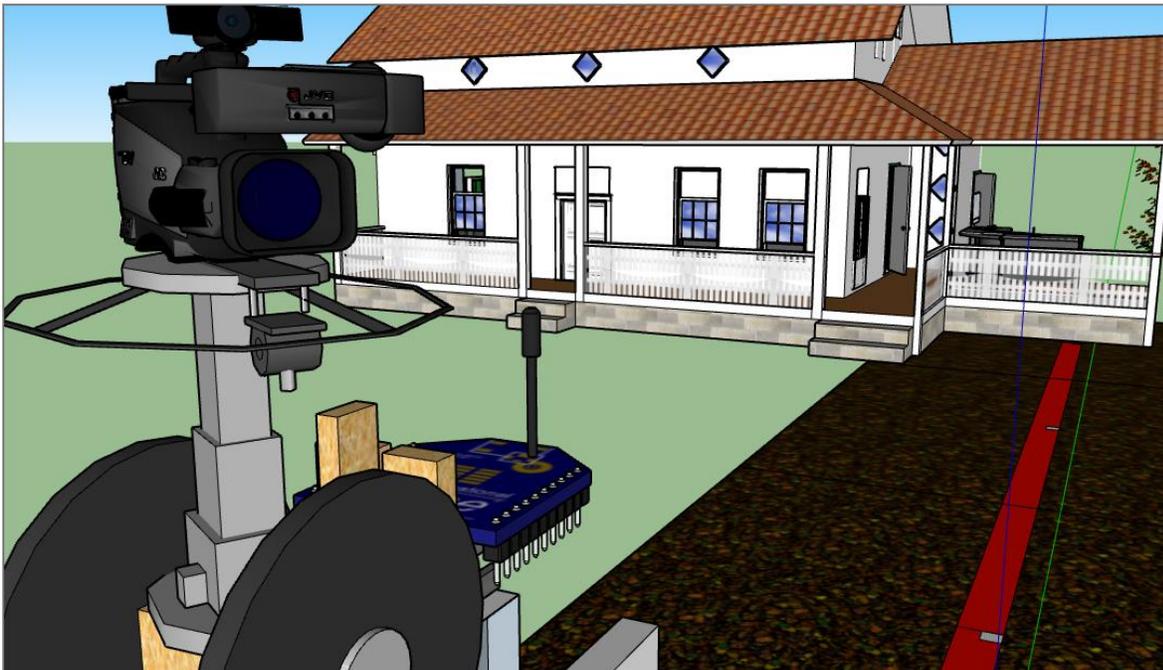


Figura 32. Punto de ubicación transmisor, perspectiva de visión hacia punto receptor en casa. (Autoría propia, 2020).

Almacenamiento transitorio y acondicionamiento de información

Ahondando un poco más en el desarrollo, se define que el sistema debe funcionar durante el inicio de la floración y el final de la cosecha dependiendo de cuál sea el caso, lo que significaría la totalidad del año, puesto que el fruto empieza a desarrollarse según reproducción en el tiempo de floración. Es así entonces, como se pretende tomar una muestra de 4 fotos semanales distribuidas a preferencia, las cuales se eliminarán de la memoria del sistema de adquisición (RaspBerry) en un periodo de 1 semana igualmente para evitar precisamente la saturación de memoria del mismo y no se afecte el rendimiento. La memoria del dispositivo en uso para este caso es de 32GB, de las cuales en la imagen siguiente se describe cuáles son consumidos para el sistema operativo y archivos del sistema, el resto para operatividad y almacenamiento, se tienen destinadas al menos 10 GB para el almacenamiento de imágenes las cuales en promedio tienen un peso de 62 KB, además de la ocupación necesaria por la trama al momento de realizar el acondicionamiento de la misma para ser transmitida.

```

pi@raspberrypi:~$ df -h
S.ficheros      Tamaño Usados  Disp  Uso% Montado en
/dev/root       29G    3,1G   25G   12% /
devtmpfs        459M     0    459M   0% /dev
tmpfs           464M    80M   384M  18% /dev/shm
tmpfs           464M    6,3M   457M   2% /run
tmpfs           5,0M    4,0K   5,0M   1% /run/lock
tmpfs           464M     0    464M   0% /sys/fs/cgroup
/dev/mmcblk0p1  253M    53M   200M  21% /boot
tmpfs           93M     4,0K   93M   1% /run/user/1000
pi@raspberrypi:~$

```

Figura 33. Distribución almacenamiento. (Autoría propia, 2020).



Figura 34. Aspecto Raspberry e interfaz gráfica. (Autoría propia, 2020).

Con lo recolectado anteriormente, se procede con el montaje para la realización de las pruebas correspondientes de procesamiento, almacenamiento y transmisión de la información. Todos los procesos necesarios ser irán desglosando a continuación.



Figura 35. Montaje de sistemas. (Autoría propia, 2020).

Inicialmente, se parte de la instalación de MySQL para la instauración de una base de datos en el RaspBerry, que permitirá la organización y almacenamiento de la información. En la figura siguiente, se observa que debido a la versión Debian Buster 10 del RaspBerry, no es posible la instalación de algunos paquetes propios de la librería de MySQL por lo que en candidato más óptimo para suplir las necesidades es la librería MariaDB fundamentada igual que MySQL.

```

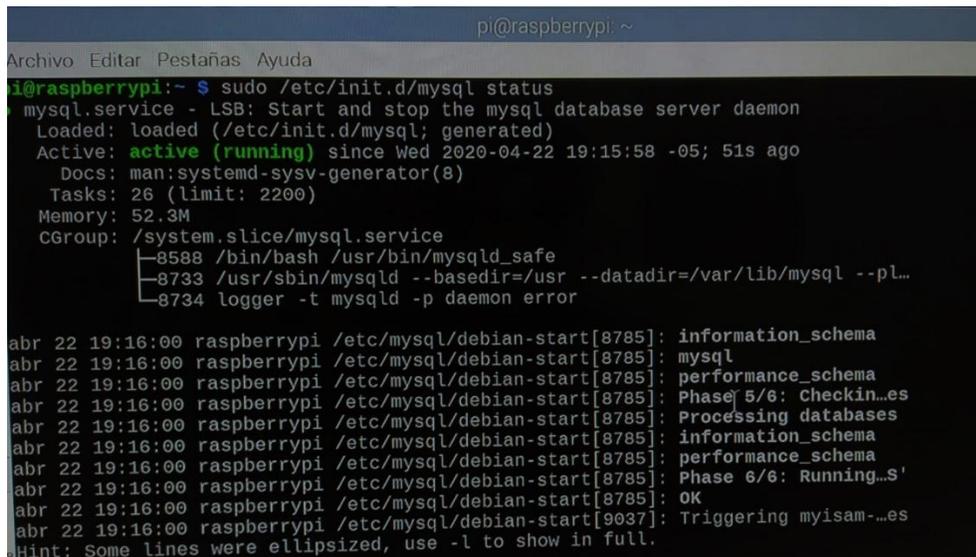
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo apt-get install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete mysql-server no está disponible, pero algún otro paquete hace referencia
a él. Esto puede significar que el paquete falta, está obsoleto o sólo se
encuentra disponible desde alguna otra fuente
Sin embargo, los siguientes paquetes lo reemplazan:
  mariadb-server-10.0

E: El paquete «mysql-server» no tiene un candidato para la instalación
pi@raspberrypi:~$ sudo apt-get install mariadb-server-10.0
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libaio1 libcgi-fast-perl libcgi-pm-perl libdbd-mysql-perl libdbi-perl
  libencode-locale-perl libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libncurses5 libreadline5 libterm-readkey-perl
  libtimedate-perl liburi-perl mariadb-client-10.0 mariadb-client-core-10.0
  mariadb-server-core-10.0
Paquetes sugeridos:
  libclone-perl libltdbm-perl libnet-daemon-perl libsql-statement-perl

```

Figura 36. Instalación MariaDB MySQL. (Autoría propia, 2020).

Posteriormente, mediante los comandos introducidos como se ve en la pantalla de la figura 10, se evidencia el estatus del servidor MySQL instalado de manera local en la RaspBerry.



```

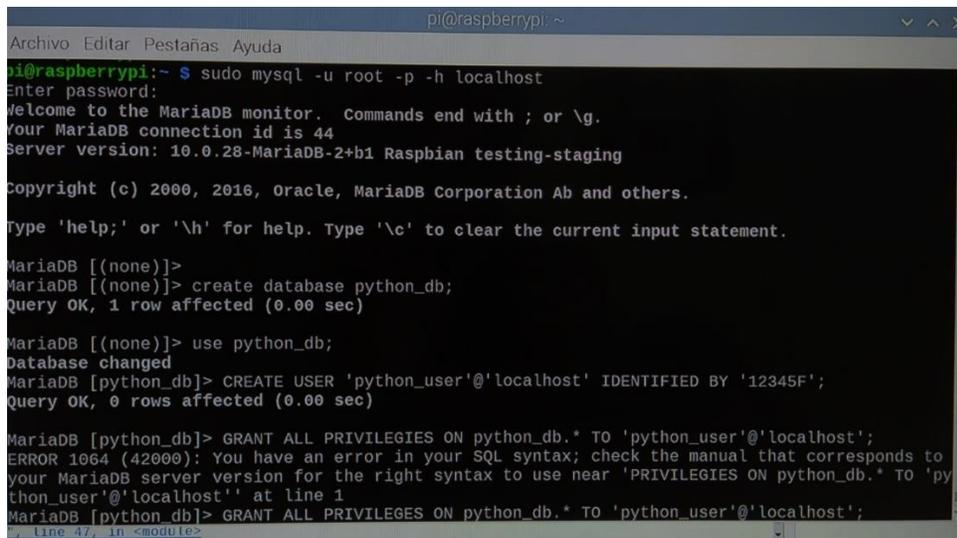
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo /etc/init.d/mysql status
mysql.service - LSB: Start and stop the mysql database server daemon
Loaded: loaded (/etc/init.d/mysql; generated)
Active: active (running) since Wed 2020-04-22 19:15:58 -05; 51s ago
Docs: man:systemd-sysv-generator(8)
Tasks: 26 (limit: 2200)
Memory: 52.3M
CGroup: /system.slice/mysql.service
├─8588 /bin/bash /usr/bin/mysqld_safe
├─8733 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --pl...
└─8734 logger -t mysqld -p daemon error

abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: information_schema
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: mysql
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: performance_schema
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: Phase 5/6: Checkin...es
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: Processing databases
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: information_schema
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: performance_schema
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: Phase 6/6: Running...s'
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[8785]: OK
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[9037]: Triggering myisam-...es
abr 22 19:16:00 raspberrypi /etc/mysql/debian-start[9037]: OK
Hint: Some lines were ellipsized, use -l to show in full.

```

Figura 37. Estado del servicio MySQL. (Autoría propia, 2020).

En el siguiente escenario, es necesario la asignación de un usuario y contraseña, además de la asignación de permisos totales para la gestión de la base de datos, en este caso al usuario `python_user`.



```

pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo mysql -u root -p -h localhost
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.0.28-MariaDB-2+b1 Raspbian testing-staging

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
MariaDB [(none)]> create database python_db;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> use python_db;
Database changed
MariaDB [python_db]> CREATE USER 'python_user'@'localhost' IDENTIFIED BY '12345F';
Query OK, 0 rows affected (0.00 sec)

MariaDB [python_db]> GRANT ALL PRIVILEGIES ON python_db.* TO 'python_user'@'localhost';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'PRIVILEGIES ON python_db.* TO 'python_user'@'localhost'' at line 1
MariaDB [python_db]> GRANT ALL PRIVILEGES ON python_db.* TO 'python_user'@'localhost';
, line 47, in <module>

```

Figura 38. Asignación de permisos y creación DB. (Autoría propia, 2020).

Una vez se tenga la base de datos creada, como se realizó en la pantalla anterior, se procede con la creación de una tabla que garantice la organización, identificación y almacenamiento de la información, que en este caso es la imagen. Por lo cual se crea la tabla con los campos descritos continuación en la figura 36.

```

pi@raspberrypi ~
Archivo Editar Pestañas Ayuda
MariaDB [(none)]> use python_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [python_db]> show tables;
+-----+
| Tables_in_python_db |
+-----+
| python_image        |
+-----+
1 row in set (0.00 sec)

MariaDB [python_db]> describe python_image;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)| NO   | PRI | NULL    | auto_increment |
| name  | text   | NO   |     | NULL    |                 |
| photo | blob   | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

MariaDB [python_db]>

```

Figura 39. Creación tablas y parámetros. (Autoría propia, 2020).

Más detalladamente, se evidencia que para el id de la foto, se establece una variable de tipo INT o entera, para el nombre una variable de tipo TEXT que incluirá el nombre denotado por la fecha, y para la foto, una variable de tipo BLOB empleada para el manejo de volúmenes grandes de información. Mediante el escritor de código del RaspBerry llamado PythonThonny se procede con la instauración del código para el almacenamiento de la información, además de la conversión de la imagen a código hexadecimal.

Inicialmente incursionando en la conexión, en la pantalla siguiente se observa la inscripción de permisos para la conexión con la base de datos y así permitir que el código en desarrollo trabaje en conjunto con la base de datos.

```

import mysql.connector
from mysql.connector import Error

try:
    connection = mysql.connector.connect(host='localhost',
                                        database='python_db',
                                        user='python_user',
                                        password='12345F')

    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version ", db_Info)
        cursor = connection.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)

except Error as e:
    print("Error while connecting to MySQL", e)
finally:
    if (connection.is_connected()):
        cursor.close()
        connection.close()
        print("MySQL connection is closed")

```

Figura 40. Python, conexión con base de datos. (Autoría propia, 2020).

En esta segunda parte, se observa una sección fundamental, en la que la imagen, se convierte a código hexadecimal, amigable posteriormente con los módulos Xbee. Se define entonces la inserción del id, fecha como nombre y la ubicación de la imagen que será codificada.

```

27 import mysql.connector
28 from mysql.connector import Error
29 import base64
30
31
32
33 def convertToBinaryData(filename):
34
35     with open(filename, 'rb') as file:
36         binaryData = file.read()
37     return binaryData
38
39 def insertBLOB(emp_id, name, photo):
40     print("Inserting BLOB into python_image table")
41     try:
42         #Your credentials of connection
43         connection = mysql.connector.connect(host='localhost',
44                                             database='python_db',
45                                             user='python_user',
46                                             password='12345F')
47
48         cursor = connection.cursor()
49         sql_insert_blob_query = """ INSERT INTO python_image
50                                 (id, name, photo) VALUES (%s,%s,%s)"""

```

Figura 41. Python, Inserción de datos. (Autoría propia, 2020).

```

51     empPicture = convertToBinaryData(photo)
52
53     # Convert data into tuple format
54     insert_blob_tuple = (emp_id, name, empPicture)
55     result = cursor.execute(sql_insert_blob_query, insert_blob_tuple)
56     connection.commit()
57
58     print("Image and file inserted successfully as a BLOB into python_image table", result)
59
60
61 except mysql.connector.Error as error:
62     print("Failed inserting BLOB data into MySQL table {}".format(error))
63
64 finally:
65     if (connection.is_connected()):
66         cursor.close()
67         connection.close()
68         print("MySQL connection is closed")
69
70
71 insertBLOB(1, "24-04-2020", '/home/pi/Desktop/Fotos_cultivo/feb23_2020_1.jpeg')
72
73

```

Figura 42. Python, Inserción de datos. (Autoría propia, 2020).

Para evidenciar que la información fue satisfactoriamente instaurada y que la imagen fue codificada, se tiene la siguiente parte del código, encargada de leer la salida de información, donde mediante una ventana emergente se puede observar la codificación.

```

75 import mysql.connector
76 from mysql.connector import Error
77
78 def write_file(data, filename):
79     # Convert binary data to proper format and write it on Hard Disk
80     with open(filename, 'wb', encoding="rb", errors='ignore') as file:
81         file.write(data)
82
83 def readBLOB(img_id, photo):
84     print("Reading BLOB data from python_image table")
85
86     try:
87         #Your credentials of connection
88         connection = mysql.connector.connect(host='localhost',
89                                             database='python_db',
90                                             user='python_user',
91                                             password='12345F')
92
93         cursor = connection.cursor(buffered=True)
94         sql_fetch_blob_query = """select * from python_image where id = %s"""
95
96         cursor.execute(sql_fetch_blob_query, (img_id,))
97         record = cursor.fetchall()
98         print(record)
99
100    except mysql.connector.Error as error:
101        print("Failed read BLOB data into MySQL table {}".format(error))
102
103    finally:
104        if (connection.is_connected()):
105            cursor.close()
106            connection.close()
107            print("MySQL connection is closed")
108
109    readBLOB(1, '/home/pi/Desktop/Fotos_cultivo/feb23_2020_1.jpeg')
110

```

Figura 43. Python, Visualización salida de código. (Autoría propia, 2020).

Transmisión y recepción

En cuanto a lo que refiere la configuración, los módulos Xbee son una herramienta bastante útil cuando se trata de transmisión mediante RF, para este caso de estudio se emplearán para realizar la transmisión de la imagen mediante su codificación. Para ello, se parte de configurar en modo AT o modo transparente, debido a que existe una restricción importante en cuanto a la recepción y es el limitado número de caracteres, lo que imposibilita el uso del modo API el cual se caracteriza por el uso de frames para la transmisión, pero que, de igual manera en su estructura, el cuerpo del mensaje o los datos, consta de solo 3 bytes de información lo que causaría ineficiencia en su operación. Se debe entonces tener en cuenta que la frecuencia a emplear para el enlace punto a punto es de 9600 Baudios.

Como primera instrucción se debe entender el Transmisor como Router, encargado de enviar la información, por lo que en la siguiente pantalla se muestra su configuración, se necesita usar la herramienta XCTU para la configuración de los dos módulos, los cuales se conectan mediante adaptador USB Xbeexplorer, diseñados para esto.

Router o Transmisor.

De los factores más importantes a configurar es precisamente el CE donde se le asigna la función de Router, y en cuanto a las configuraciones de direccionamiento, en los campos DH y DL, se le deben insertar los datos que posee el dispositivo terminal o receptor.

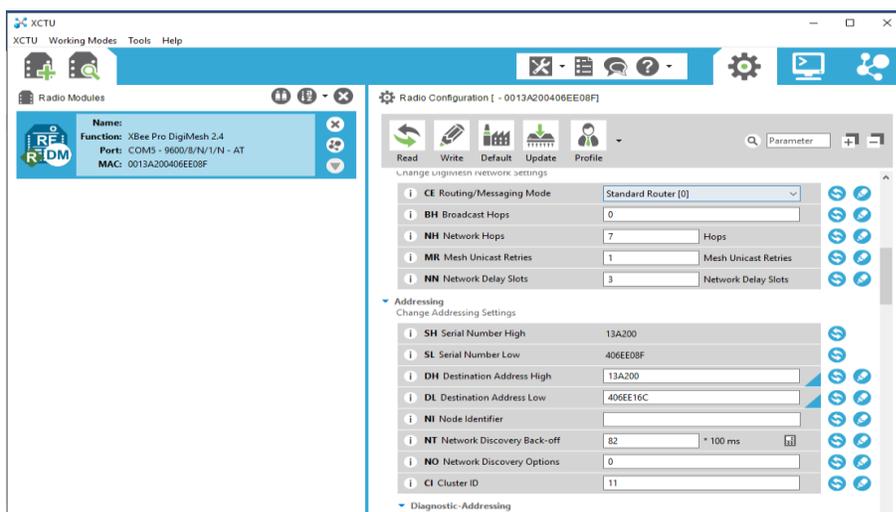


Figura 44. Xbee configuración transmisor o router. (Autoría propia, 2020).

Dispositivo terminal o receptor.

Partiendo de lo anterior, en el parámetro CE, se debe seleccionar la opción de dispositivo terminal para designar este módulo como receptor, donde igualmente los parámetros DH y DL,

deben contener las direcciones del otro modulo, el router, para garantizar la comunicación entre ellos.

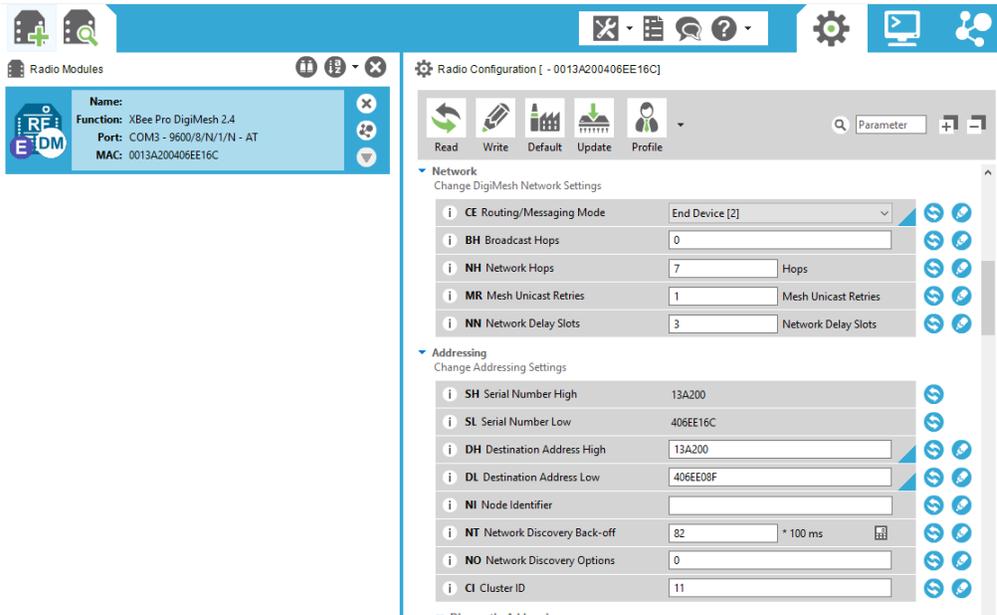


Figura 45. Xbee configuración receptor o dispositivo terminal. (Autoría propia, 2020).

Para la verificación de una comunicación efectiva, se procede con la prueba mediante la consola que ofrece la interfaz XCTU como se muestra en las dos imágenes siguientes, donde el mensaje escrito, Uniagustiniana_2020 en azul, se puede ver recibido por el receptor, de color rojo.

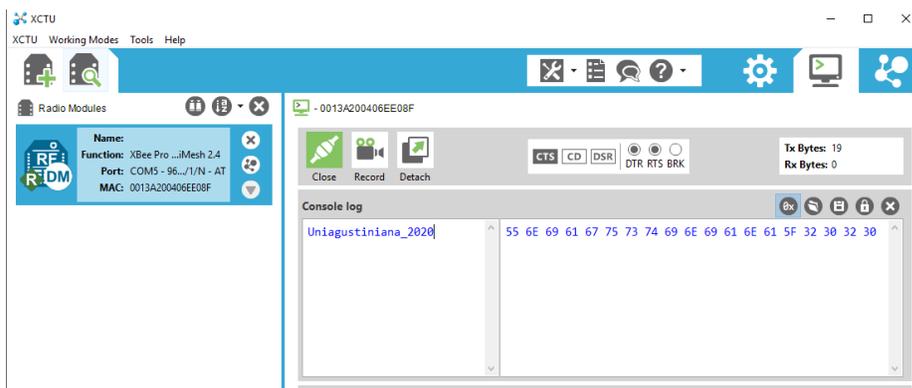


Figura 46. Xbee prueba transmisión. (Autoría propia, 2020).

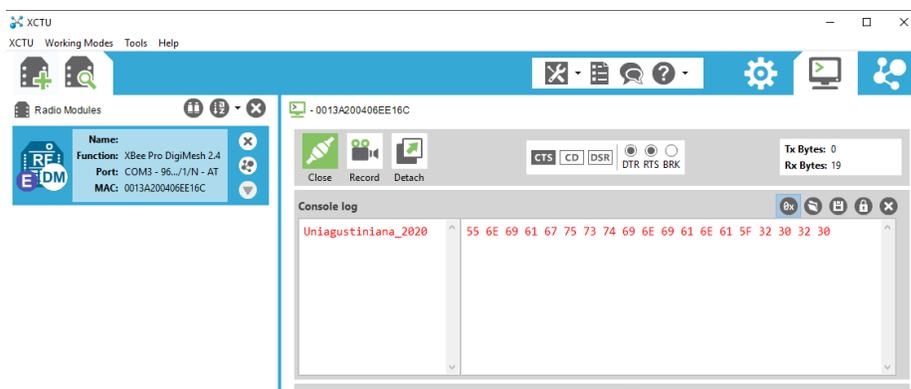


Figura 47. Xbee prueba recepción. (Autoría propia, 2020).

Vale la pena aclarar, que la interfaz usada anteriormente, XCTU, fue usada para efectos de configuración en un sistema operativo Windows, puesto que la configuración mediante interfaz gráfica no está disponible para el sistema operativo Debian Buster 10, propio de el RaspBerry. Por tal motivo, para el reconocimiento del módulo transmisor Xbee en el RaspBerry se requiere del siguiente código, el cual lee el puerto USB donde se conecta el módulo Xbee y posteriormente se instaura una variable que contendrá el código de información de la imagen.

```

1 import serial
2
3 ser = serial.Serial('/dev/ttyUSB0', 9600)
4 string = '[(1,24-04-2020,\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00\x00\x01\x00\x01\
5 print 'Sending "%s"' % string
6 ser.write('%s\n' % string)
7
8 while True:
9     incoming = ser.readline().strip()
10    print 'received %s' %incoming
11    ser.write('Rpi Received: %s/n' % incoming)]

```

Figura 48. Python, código para envío de información codificada mediante el Xbee. (Autoría propia, 2020).

Analizando el uso de recursos del Xbee, se evidencio que, al tener un número limitado de caracteres, la única manera posible de recopilar el gran volumen de información representada en código es mediante la división del mismo en varias partes que al ser reunidas en la recepción conformaran la imagen del estado del cultivo. Se constituye que cada imagen cuenta con un tamaño promedio de 62,7KB lo que arroja como resultado un código de 184990 caracteres, los cuales por consiguiente deben ser divididos en un aproximado de 36 partes para ser recibidos sin perdidas en la recepción por parte del Xbee.

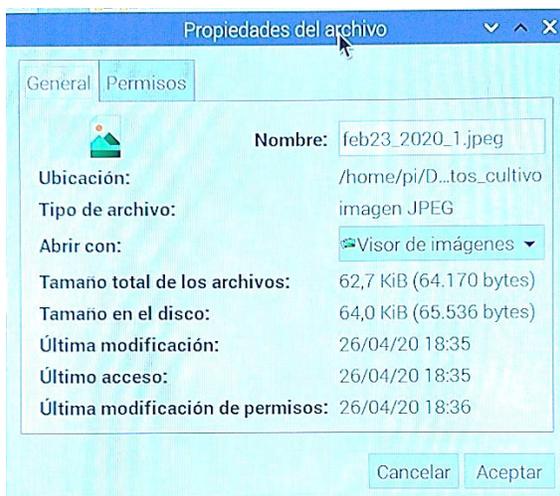


Figura 49. Tamaño de la imagen en KB. (Autoría propia, 2020).

```

Connected to MySQL Server version 5.5.5-10.0.28-Ma
You're connected to database: ('python_db',)
MySQL connection is closed
Inserting BLOB into python_image table
Image and file inserted successfully as a BLOB into
MySQL connection is closed
Reading BLOB data from python_image table
[(1, '24-04-2020', b'\xff\xd8\xff\xe0\x01\x01JFIF\x00\x01\
MySQL
184990 characters squeezed. Click for details.
>>>

```

Figura 50. Tamaño de la imagen en caracteres del código. (Autoría propia, 2020).

Es así como a continuación, en la imagen siguiente se evidencia el código para la realización de la división por el número de caracteres, dividiendo cada parte en 5285 caracteres para ser enviados.

```

1  img_64 = '[(1,24-04-2020,\xff\xd8\xff\xe0\x01\x01JFIF\x00\x01\
2
3  part0= img_64[0:5285]
4  part1= img_64[5285:10570]
5  part2= img_64[10571:15855]
6  part3= img_64[15856:21140]
7  part4= img_64[21141:26425]
8  part5= img_64[26426:31710]
9  part6= img_64[31711:36995]
10 part7= img_64[36996:42280]
11 part8= img_64[42281:47565]
12 part9= img_64[47566:52850]
13 part1= img_64[52851:58135]
14 part11= img_64[58136:63420]
15 part12= img_64[63421:68705]
16 part13= img_64[68706:73990]
17 part14= img_64[73991:79275]
18 part15= img_64[79276:84560]
19 part16= img_64[84561:89845]

```

Figura 51. División del código. (Autoría propia, 2020).

Para la ejecución del código, se sigue su ruta, identificada como `cd/home/pi/Documents` y se ejecuta con el nombre del archivo compilado, en este caso `XBEE.py`. De esta manera el código empieza a ser transmitido en código ASCII.

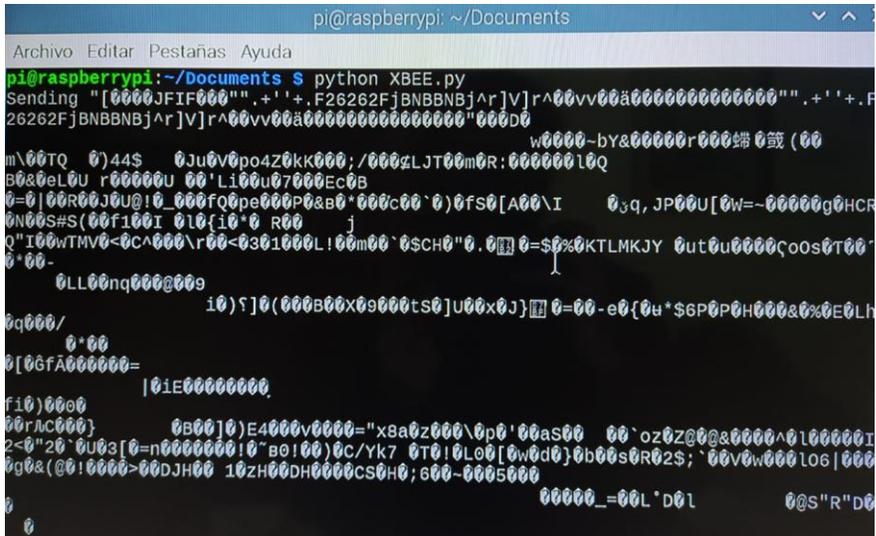


Figura 52. Debian OS, ejecución del código XBEE.py y su visualización de información. (Autoría propia, 2020).

A continuación, se tiene la recepción mediante código ASCII, y paralelamente en Hexadecimal, allí se ve la primera de las 36 partes que deben ser recibidas para totalizar la composición de la imagen.

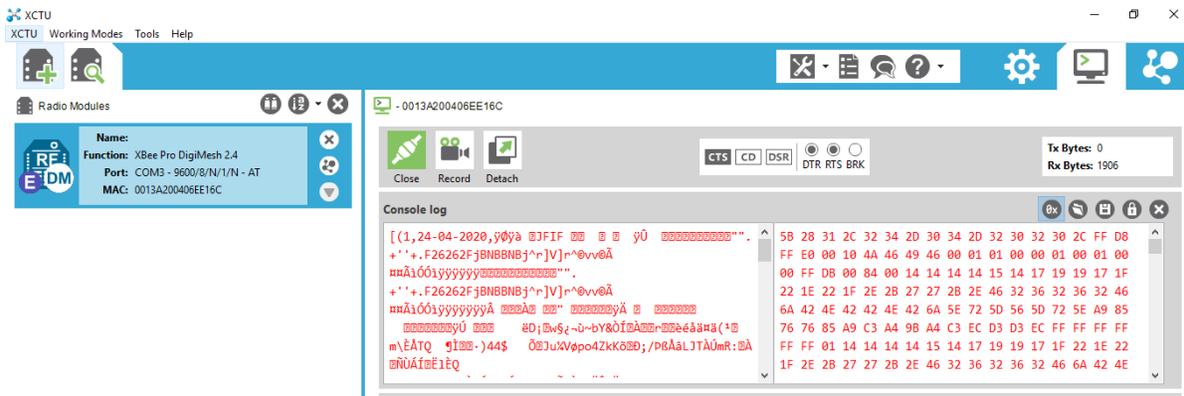


Figura 53. XCTU, recepción de código en el receptor Xbee. (Autoría propia, 2020).

En el momento de recibir la unificación de todas las partes, se tiene que el código ASCII traducido en Hexadecimal o Base64 genera la composición total de la imagen. A continuación, se ve la relación del código recibido traducido en varios tipos de codificación.

Trabajos futuros

Una vez comprobado el funcionamiento del sistema con evidencia de la recepción de datos, se establece que para una proyección a futuro, se plantea que la información o la imagen que llega al punto receptor, al ser recibida, deberá ser unificada y analizada según pudiese ser la necesidad y según los métodos más convenientes, para así determinar las acciones correspondientes dependiendo de la presencia de plaga, todo esto se plantea como un trabajo aparte puesto que va fuera del marco o delineamiento estipulado y del alcance.

Conclusiones

Una vez realizado el estudio, de manera analítica y exhaustiva se pudieron evidenciar varios aspectos importantes en lo que fue la constitución del proyecto y como sus diversos componentes jugaron parte fundamental en el proceso. Inicialmente se parte de analizar el escenario, un escenario rural en donde en el panorama Colombiano se tiene conocimiento del poco cubrimiento de lo que a comunicaciones se refiere, incluso de servicios como la electricidad o demás básicos, por ello es importante recalcar el uso de sistemas que contribuyan a la mejora de labores y además en conjunto, el posible uso de energías renovables para su apoyo, puesto que el sector agrícola ha sentado las bases como un sector productivo de fuerte presencia en el país y del que se pueden optimizar varios procesos. En lo que refiere al meollo de este proyecto, se evidencio que un sistema de adquisición de datos requiere de un sostenible respaldo de alimentación para su implementación, puesto que la fidelidad de la información dependerá de ello, en este caso puntual el Raspberry sería el cerebro de la operación, quien en la primera etapa se encarga de la recepción de las imágenes, donde el almacenamiento transitorio juega un rol fundamental. El Raspberry y su interfaz gráfica evidenciaron un comportamiento estable en lo que respecta al procesamiento de la información o imágenes que recibe por parte de la cámara, durante su uso constante, genero algunas complicaciones por el rendimiento cuando se manejaban las grandes cantidades de texto reflejadas en el código, pero nada que realmente truncara el desarrollo de la operación. Se resalta el uso del lenguaje Python, que al ser un lenguaje orientado a objetos ofrece múltiples funcionalidades, lo que garantiza flexibilidad en muchas funciones con variados fines, aunque se caracteriza por especial legibilidad, es un lenguaje que requiere de algunas bases de conocimiento previas, en lo que a estructura y escritura se refiere. Se logro establecer un código estructurado para la adecuación de la imagen en caracteres, además de su oportuno almacenamiento en base de datos MySQL, que aunque no posee una interfaz gráfica para el sistema operativo Debian Buster 10, propio de este modelo de Raspberry, se pudo manejar vía consola, tal vez de los inconvenientes más significativos es la incompatibilidad con algunas funciones y aplicaciones en este sistema operativo, aunque ofrece alternativas, se desperdicia un poco la amigable interfaz gráfica en el momento que algunas aplicaciones no operan, muchas veces debido a la arquitectura de proceso.

En concordancia con este último tema, anteriormente expuesto, sucede algo similar en cuanto al módulo Xbee transmisor, es de importancia recalcar que la interfaz gráfica XCTU para la configuración y administración de los módulos, no está disponible para Debian Buster 10 Raspbian por lo que todo se debe realizar por consola; aun y con este impase, se logró configurar el módulo Xbee mediante su adaptador al puerto USB sin presentar problema a la hora de reconocerlo, de igual manera como se explicó en el documento, la configuración para mayor facilidad y entendimiento se hizo en una interfaz gráfica XCTU de un sistema operativo Windows, puesto que es de vital importancia que los módulos se reconozcan entre ellos en cuanto a direccionamiento, además de la definición de sus roles como router o transmisor y el terminal o receptor.

Al realizar la transmisión de la información, es decir de las imágenes en código, se observaron algunos comportamientos. En primera instancia se constató que aunque mediante el código para la transformación de la imagen en caracteres, aunque se toma como referencia el tipo Base64, el Raspberry por alguna razón lo convierte a código hexadecimal primero, y al momento de ser transmitido mediante el Xbee se transfiere en código ASCII, lo cual representa la multifuncionalidad y versatilidad sin generar inconvenientes; y por parte de los módulos Xbee algo indispensable para tener en cuenta, es su limitada capacidad de recibir un alto número de caracteres en cada mensaje. Lo que en consecuencia significa la división obligatoria del código en el número descrito en el apartado correspondiente. Se realizaron varias pruebas para determinar cuál era el número de caracteres permitido por cada parte, puesto que la documentación respecto a ello no es muy nutrida, sin embargo, se estableció el numero necesario para que no se perdiera ningún carácter que pudiese afectar la fidelidad de la información. Se realizaron pruebas con algunas distancias sin afectación, lo que hace propensa la pérdida de información es la interrupción significativa de la línea de vista y se debe tener en cuenta que la distancia aconsejable máxima es de 70 a 100 metros para el caso de estos módulos con los que se trabajó, en caso de requerir otras características existen otros módulos ofrecidos por la compañía DIGI, que poseen más funcionalidades y mejoras en cuanto a distancia y banda de frecuencia, entre otras.

Referencias

- "Chito", C. Á. (5 de Junio de 2017). *El blog de chito de tecnología*. Obtenido de <https://chitoraspberrypi.blogspot.com/2017/06/calculando-el-movimiento-del-servo-con.html>
- Amazon. (2020). *Amazon.com*. Obtenido de <https://www.amazon.com/-/es/Servo-Motor-cmucam5-Robotic-Camera/dp/B00IVOEN1Y>
- Casco, S. M. (Septiembre de 2014). *jeuazarru.com*. Obtenido de Raspberry Pi, Arduino y Beaglebone Black, Comparación y Aplicaciones: <http://jeuazarru.com/wp-content/uploads/2014/10/MiniPCs.pdf>
- Claudio Mourgues, M. F. (Enero de 2001). *Investigaciones en Tecnologías de Información Aplicadas a la Industria A/E/C*. Obtenido de <https://stacks.stanford.edu:https://stacks.stanford.edu/file/druid:ck956zv9566/TR124.pdf>
- Code Beautify. (2020). *Convert Your Base64 to Image*. Obtenido de <https://codebeautify.org/base64-to-image-converter>
- conv.darkbyte.ru. (2020). *Online converter*. Obtenido de <https://conv.darkbyte.ru:https://conv.darkbyte.ru/>
- Cruz, Y. M. (18 de 11 de 2010). *Tecnologías de información y comunicación para el desarrollo rural en Colombia*. Obtenido de [file:///C:/Users/RojasTellez/Downloads/387-Texto%20del%20art%C3%ADculo-607-1-10-20121002%20\(1\).pdf](file:///C:/Users/RojasTellez/Downloads/387-Texto%20del%20art%C3%ADculo-607-1-10-20121002%20(1).pdf)
- DID Electrónica. (2020). <https://www.didacticaselectronicas.com/>. Obtenido de <https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/raspberry/accesorios-raspberry/cajas/caja-carcasa-cubierta-riel-din-carril-din-box-rasp-din-detail>
- DIGI. (2018). *Digi XBee S1 802.15.4 RF Modules datasheet*. Obtenido de https://www.digi.com/pdf/ds_xbeemultipointmodules.pdf
- Ecuguanábana. (2020). <http://www.ecuaguanabana.com/>. Obtenido de <http://www.ecuaguanabana.com/manual>
- encolombia. (2020). <https://encolombia.com/>. Obtenido de <https://encolombia.com/economia/agroindustria/cultivo/cultivodeguanabana/>

- Fabio Asecio, P. C. (2019). *wikiredes Inalambricas*. Obtenido de Wiki creada para clase de Telecomunicaciones, conectividad y redes de INACAP Osorno.:
<https://sites.google.com/site/wredwiki/normas/normas-de-redes-inalambricas>
- Jiménez, A., Ravelo, D., & Gómez, J. (Julio de 2010). *Tecnura*. Obtenido de
<https://www.redalyc.org/pdf/2570/257019633005.pdf>
- Jirapond Muangprathuba, *. N. (2019). IoT and agriculture data analysis for smart farm.
Computers and Electronics in Agriculture, 467-474.
- Jirapond Muangprathuba, N. B. (2019). *ELSEVIER*. Obtenido de
www.elsevier.com/locate/compag
- La República. (9 de Julio de 2015). *Agronegocios*. Obtenido de
<https://www.agronegocios.co/agricultura/guanabana-un-cultivo-al-que-muy-pocos-se-le-miden-2621218>
- Ministerio de Telecomunicaciones. (2019). *mintic.gov.co*. Obtenido de
<https://www.mintic.gov.co/portal/inicio/Normatividad/Leyes/>
- Miranda, D., Barragan, E., Barreto, D., & Caicedo, A. (2020). *Manejo integrado del cultivo de la guanábana*. Obtenido de <http://bibliotecadigital.agronet.gov.co/>:
<http://bibliotecadigital.agronet.gov.co/bitstream/11348/6765/1/160.1.pdf>
- Nagel, J. (Noviembre de 2012). *Principales barreras para la adopción de las TIC en la agricultura y en las áreas rurales*. Obtenido de
https://repositorio.cepal.org/bitstream/handle/11362/4011/S2012079_es.pdf?sequence=1&isAllowed=y
- National Instruments. (2019). *National Instruments*. Obtenido de <http://www.ni.com/data-acquisition/what-is/esa/>
- neo.lcc.uma.es. (2019). *Herramientas web para la enseñanza de protocolos de comunicacion*. Obtenido de <http://neo.lcc.uma.es/evirtual/cdd/tutorial/enlace/Entramado.html>
- RedHat. (2019). *redhat.com*. Obtenido de <https://www.redhat.com/es/topics/data-storage>
- SandoRobotics. (2020). *SanDoRobotocs - tecnologia en electronica*. Obtenido de
<https://sadorobotics.com/producto/sen-14678/>
- sarenet.es. (2019). *sarenet*. Obtenido de <https://www.sarenet.es/ayuda/glosario/perdida-paquetes.html>

Universidad Distrital Francisco José de Caldas. (8 de Noviembre de 2018). *XIII Congreso Internacional de Electrónica*. Obtenido de Libro de Memorias.

Vega, O. A., & Baena, D. L. (2014). *Inclusión Digital de comunidades rurales colombianas*. Obtenido de

https://www.researchgate.net/profile/Omar_Vega3/publication/298061436_Inclusion_Digital_de_comunidades_rurales_colombianas/links/56e5753508aedb4cc8ae65c9.pdf

Wendt, Z. (22 de Marzo de 2018). *Arrow Electronics*. Obtenido de <https://www.arrow.com/es-mx/research-and-events/videos/the-top-10-development-platforms-dev-kits-2018>