UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
Ingenieros de Sistemas

TEMA:
COMPRESIÓN DE DATOS CON ALGORITMO DE HUFFMAN PARA TRANSMISIÓN DE DATOS MEDIANTE LI-FI UTILIZANDO HARDWARE LIBRE.

AUTORES:
STALIN JOSÉ MINA CHILLA
RENATTO RICARDO RUGEL BARRIGA

TUTOR:
MANUEL RAFAEL JAYA DUCHE

Quito, septiembre del 2017

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Stalin José Mina Chilla y Renatto Ricardo Rugel Barriga, con documento de identificación 1725447104 y 1720507969, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del trabajo de titulación con el tema "COMPRESIÓN DE DATOS CON ALGORITMO DE HUFFMAN PARA TRANSMISIÓN DE DATOS MEDIANTE LI-FI UTILIZANDO HARDWARE LIBRE", mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

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

STALIN JOSÉ
MINA CHILLA

CI: 1725447104

RENATTO RICARDO
RUGEL BARRIGA

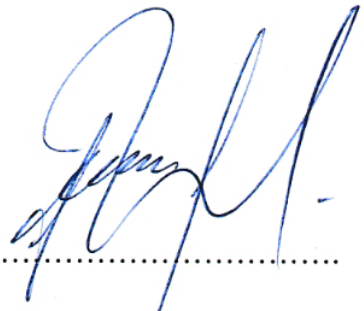
CI: 1720507969

Quito, septiembre del 2017

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación: Compresión de Datos con Algoritmo de Huffman para Transmisión de Datos mediante Li-Fi utilizando Hardware Libre, realizado por Stalin José Mina Chilla y Renatto Ricardo Rugel Barriga, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, septiembre del 2017



.....

Ing. Manuel Rafael Jaya Duche, Msc
CI: 1710631035

DEDICATORIA

Dedico este proyecto de titulación a Dios, a mis padres y a mis hermanas. A Dios porque ha sabido guiar mi vida a cada paso que he dado hasta hoy, llenándome de fuerzas y de fe.

A mis padres, por ser mi apoyo y mi motor, siempre guiando mis vida y dando su sustento para para poder ser alguien en la vida con su ejemplo y su amor. A mis hermanas por su paciencia y su ayuda para poder terminar con éxito este gran paso de mi vida. Gracias a ellos por proporcionar su entera confianza sin dudar ni un solo momento en mi inteligencia y capacidad.

Los amo con mi vida.

Stalin J. Mina Ch.

Dedico este logro a mis padres, a mi hermana y a mi abuelita pues han sido el más grande apoyo que he tenido en esta vida y con el cual he podido avanzar en este camino, me han permitido crecer como persona día a día con su ejemplo y valores y sobre todo por estar ahí cuando muchas veces quise darme por vencido.

A mis tíos, que han sido como mis hermanos, por su constante preocupación, atención y compañía en cada uno de los momentos de mi vida, por sus palabras de aliento y el sacrificio de su propio tiempo, los cuales han hecho posible alcanzar muchas de las metas que me he propuesto.

De igual manera este logro se lo dedico a todas aquellas personas que me han brindado su conocimiento y ayuda durante todos estos años de estudio pues sin ellos no estuviese llegando a la culminación de mi carrera.

Renatto R. Rugel B.

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana que gracias a su misión y docentes, ayudaron a nuestro crecimiento como personas y como profesionales; a nuestro tutor Ing. Rafael Jaya, por darnos su apoyo, conocimiento y guía para realizar este trabajo.

Stalin J. Mina Ch.

Renatto R. Rugel B.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

La comunicación mediante la luz visible se ha convertido en los últimos años en uno de los avances más importantes dentro de las telecomunicaciones, pues permitirá alcanzar mejoras significativas con respecto a la velocidad en la que se transmite la información y su seguridad.

Teniendo en cuenta esto, el presente proyecto se centra en la construcción de un prototipo de comunicación Li-Fi unidireccional que junto con la implementación del algoritmo de codificación de Huffman, permita disminuir el tamaño total de los datos a transmitir y la utilización del espectro dentro del enlace de comunicación, aumentando así, las alternativas que permitan optimizar el uso del canal de comunicación.

Se presenta también un análisis comparativo entre Arduino y Raspberry Pi así como el software disponible para cada uno de estos, los cuales permitan obtener los mejores resultados y sobre todo, que sean accesibles económicamente.

Finalmente, se muestran las distintas pruebas realizadas con respecto a parámetros de velocidad y utilización del espectro de comunicación, así como sus respectivos análisis dentro de dos escenarios, el primero sin implementar la codificación Huffman y, el segundo escenario haciendo uso de dicha codificación; para tales pruebas se ha hecho uso del osciloscopio Agilent.

Abstract

In recent years, communication through visible light has become one of the most important developments in telecommunications, as it will allow significant improvements in terms of the speed at which information is transmitted and its security.

Considering this, the present project focuses on the construction of a unidirectional Li-Fi communication prototype which, together with the implementation of the Huffman coding algorithm, allows to reduce the total size of the data to be transmitted to be reduced and also the use of the spectrum within the communication link, thus increasing the alternatives that allow optimizing the use of the communication channel.

It is also presented a comparative analysis between Arduino and Raspberry Pi as well as the software available for each one, which allow us to obtain the best results and above all, that are economically accessible.

Finally, the different tests performed regarding speed parameters and communication spectrum utilization are shown, as well as their respective analysis within two scenarios, the first one without implementing the Huffman coding algorithm and the second scenario making use of said coding ; for such tests the use of the Agilent oscilloscope has been made.

INTRODUCCIÓN

Planteamiento del Problema

Una de las actividades más importantes en el día a día en este mundo tecnológico es la transferencia de información. Para lo cual se busca mecanismos que permitan el envío y recepción de información desde cualquier lugar, de una manera rápida, segura y manteniendo la integridad de la misma. (Gupta, 2015)

Entre los principales problemas que se tiene en la comunicación inalámbrica Wi-Fi, se encuentran las interferencias que pueden ser ocasionadas por teléfonos inalámbricos que operan a la misma frecuencia, o debido a otras redes inalámbricas cercanas o incluso por otros equipos conectados a la misma red.

La transmisión en Wi-Fi es un medio compartido en el cual, cualquier dispositivo que se encuentre en el alcance de la señal pueda escuchar o interferir la comunicación, además el coste de los elementos de hardware necesarios para captar o interferir la transmisión y realizar actividades de piratería informática es realmente bajo y, cualquiera pueda tener acceso a ella e incluso desde dispositivos celulares. Esto ha hecho que las medidas de seguridad encaminadas al control de acceso a edificios, hogares, salas, etc; resulten ineficientes de cara a proteger la red. (Castro, 2005)

Así mismo es necesario tener en cuenta el consumo de energía de la que hace uso Wi-Fi y es que tal como dice (Jaya, Vizcaíno, & Acosta, 2014) en su investigación, a mayor velocidad de transferencia de datos se tendrá mayor consumo de energía y si bien los dispositivos actuales ya vienen dotados con un sistema a modo de ahorro de energía en estas redes, se disminuye la cobertura y los dispositivos se vuelven susceptibles a ruidos y hasta podría causar interferencia con otros dispositivos inalámbricos.

En función de esto se procederá a crear un prototipo de comunicación Li-Fi con algoritmo de compresión de Huffman, que permita la optimización de recursos.

Justificación

El prototipo de comunicación Li-Fi que se va a diseñar plantea muchas ventajas, ya que se puede emplear la infraestructura de luz LED existente a la que solo habría que instalar moduladores en las bombillas o focos para que éstas pudiesen transmitir información. Como se indica en la página de pureLiFi (pureLiFi, 2014): Li-Fi es único ya que la misma energía de luz visible usada para la iluminación también se puede utilizar para la comunicación.

La seguridad es una de las características más importantes en Li-Fi, ya que al ser luz, no traspasa objetos sólidos, por lo que existe mayor privacidad y seguridad de la información. Como no hay ondas iguales o que interfieran, el protocolo es invulnerable para los hackers ya que no podrán captar las señales que se emite. (Dimitrov & Haas, 2015)

La necesidad de la compresión de datos surge a raíz de que más y más información que se genera y utiliza está en formato digital y el número de bytes necesarios para representarlos puede ser enorme. (Sayood, 2012). Por tal razón con la implementación del algoritmo de Huffman se pretende realizar la transmisión de información, manteniendo la integridad de la misma, utilizando el menor número de bits posible, con ello las limitaciones actuales en cuanto a velocidad de transmisión en Li-Fi no sean un inconveniente y que el algoritmo de compresión sea transparente para el usuario final, logrando así una transmisión óptima y con el menor número de bits y en un tiempo proporcional.

Objetivos

Objetivo General

Realizar la compresión de datos durante la transmisión de información, dentro de un sistema de comunicación Li-Fi unidireccional utilizando hardware libre, mediante el algoritmo de Huffman logrando así que dicha comunicación ocupe la menor cantidad del espectro disponible.

Objetivos Específicos

Investigar información relacionada con Li-Fi y Algoritmo de Huffman que permita contar con las bases necesarias para el desarrollo del tema.

Analizar documentación técnica de Hardware y Software, para elegir las características adecuadas de cada elemento para la implementación del escenario de comunicación.

Diseñar y desarrollar el escenario de comunicación Li-Fi, conjuntamente con la implementación del algoritmo de Huffman.

Efectuar pruebas en el escenario de comunicación para la transmisión de datos con y sin el Algoritmo de Huffman, con el fin de recoger datos para realizar un análisis comparativo de los mismos.

Alcance

El presente Proyecto Técnico, consiste en implementar el Algoritmo de Huffman en un sistema Li-Fi, para determinar la reducción de tiempo durante la transmisión de datos en dicho sistema.

El alcance del proyecto es el análisis de los distintos tiempos que conlleva el transmitir la información sin codificar contra la información ya codificada mediante Huffman y los beneficios que esto con lleva para el sistema.

Metodología

La metodología a utilizarse es la experimental ya que se levantará un escenario de comunicación Li-Fi conjuntamente con la implementación del algoritmo de compresión de datos Huffman, que permita optimizar el espectro de Ancho de Banda, y se realizarán pruebas que permitan apreciar la codificación de los datos en la transmisión con un menor número de bits.

Capítulo 1

Marco Teórico

En este capítulo, se presentan los conceptos más relevantes que permitirán entender de mejor manera el presente proyecto de investigación acerca de la tecnología Visible Light Communication (Li-Fi), conjuntamente con la utilización del algoritmo de codificación Huffman como mecanismo de compresión de los datos a ser transmitidos en el sistema Li-Fi. Se inicia con las definiciones y características de Li-Fi seguidos por un análisis de los elementos y parámetros que intervienen en dicha tecnología, finalizando con la descripción de los aspectos más sobresalientes del algoritmo de Huffman.

1.1 Visible Light Communication (VLC)

Comunicación por luz visible (VLC) IEEE 802.15.7, es un medio de comunicación para la transmisión de datos que utiliza el espectro de luz visible, dando velocidades 10.000 veces más que en el espectro radioeléctrico, evitando así las saturaciones en los sistemas de Radiofrecuencia (Combariza, Vargas Roa, & Acosta, 2015).

Como se indica en la figura 1, VLC utiliza el espectro de luz en el rango de 400-800 THz, esto quiere decir que podrá trabajar en cualquier intervalo de este rango; que comprende desde el ultravioleta hasta el infrarrojo.

Espectro de la Luz Visible

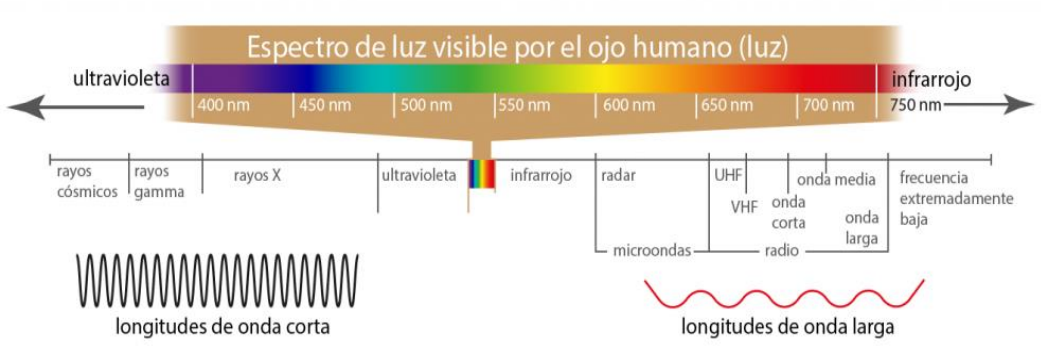


Figura 1. Espectro Electromagnético
Elaborado por: (Calzadilla, 2013)

VLC aprovecha la evolución de la tecnología LED[1] (Light-Emitting Diode o Diodo Emisor de Luz), ya que permite que las bombillas utilizadas en hogares sirvan no solo como iluminación sino que se aproveche para la transmisión de información, como se mencionó antes, VLC transmite datos 10000 veces más que otras tecnologías de comunicaciones inalámbricas. Uno de los aspectos importantes que ofrece es su ancho de banda ya que no se comparte ni se divide, supóngase que en una oficina se utiliza VLC con una velocidad de 100Mbps y son 3 usuarios; entonces cada usuario contara con dicha velocidad. Y más importante, no afecta al ojo humano ni a la salud en general. Como se observa en la figura 2.

Escenario de una Oficina utilizando tecnología VLC.

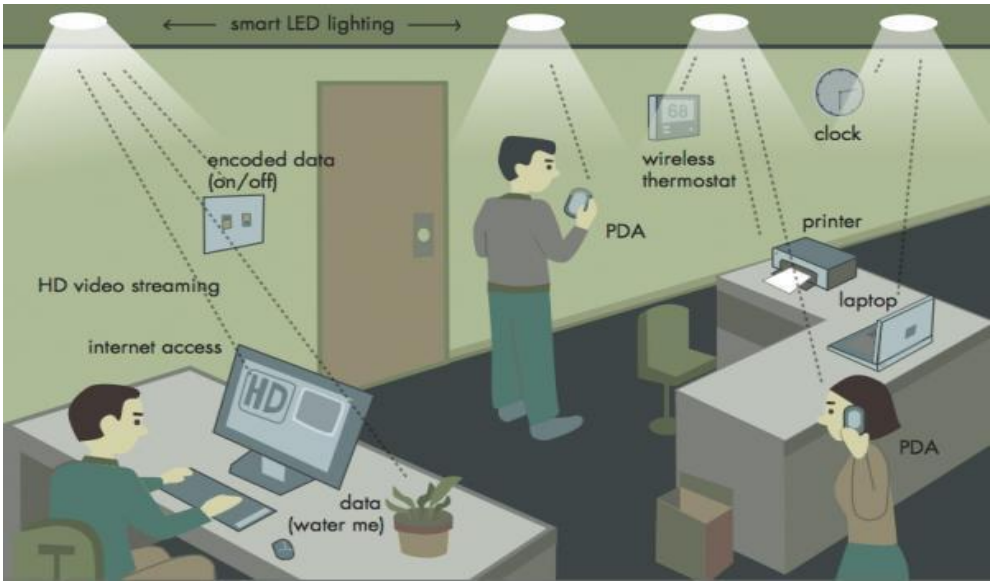


Figura 2. VLC en una oficina, todos los dispositivos están conectados a través de la luz. Elaborado por: (Boston, 2008)

1.1.1 Ventajas de VLC

Como se ha mencionado esta tecnología puede transmitir datos hasta 10000 veces más con respecto a otras tecnologías.

[1] "Se trata de un cuerpo semiconductor sólido de gran resistencia que al recibir una corriente eléctrica de muy baja intensidad, emite luz de forma eficiente y con alto rendimiento."((Educación, 2012)

Con respecto a seguridad, debido a que las ondas de luz no pueden traspasar paredes, la información no puede ser interceptada más allá de su cobertura. Es particularmente especial para ser implementado lugares en donde no afecte el uso de otros dispositivos electrónicos como por ejemplo en hospitales. (Combariza, Vargas Roa, & Acosta, 2015)

1.1.2 Limitación de VLC

VLC a pesar de tener un gran ancho de banda y de la cual un usuario puede disponer en su totalidad, la distancia es muy corta por la cobertura de la luz. Se ha llevado a cabo pruebas de funcionamiento en pequeñas oficinas, teniendo una cobertura de 10 metros desde la fuente de luz LED. (González & Aravena, 2013)

1.2 Tecnología Li-Fi

Esta tecnología es relativamente nueva pudiendo decir que aún se encuentra en etapa experimental, se dio a conocer en el 2011, en donde Harald Haas en una charla TEDGlobal mencionó todos los beneficios que puede ofrecer esta tecnología al utilizar el espectro de luz visible, indico que se puede transmitir información a grandes velocidades y que no afecta a la salud del ser humano ya que es perceptible al ojo humano. También se llevó a cabo un experimento utilizando una linterna LED, con ello se logró transmitir 10Mbps a su computador personal. Es así que bautizó a esta tecnología como Li-Fi. (González & Avarena, 2013 y PureLiFi, 2014)

1.2.1 Diferencia entre VLC y Li-Fi

VLC no es igual que Li-Fi, ya que VLC es un concepto global. Li-Fi está orientado a permitir la conexión a Internet, para ello se pretende transmitir información por medio de la luz y la interconexión entre varios dispositivos que estén en el sistema Li-Fi. VLC sirve solo para la transmisión unidireccional de datos. (pureLiFi, 2014). Li-Fi está basado en el modelo de VLC, como se observa en la figura 3.

Li-Fi tienes dos características importantes:

- **Itinerantica:** El usuario puede moverse entre las luces sin perder conectividad.
- **Acceso Múltiple:** Diferentes personas pueden navegar en internet al mismo tiempo utilizando la misma luz del LED.

Li-Fi está basado en el modelo de VLC, como se observa en la figura 3.

Modelo de Diseño de VLC básico

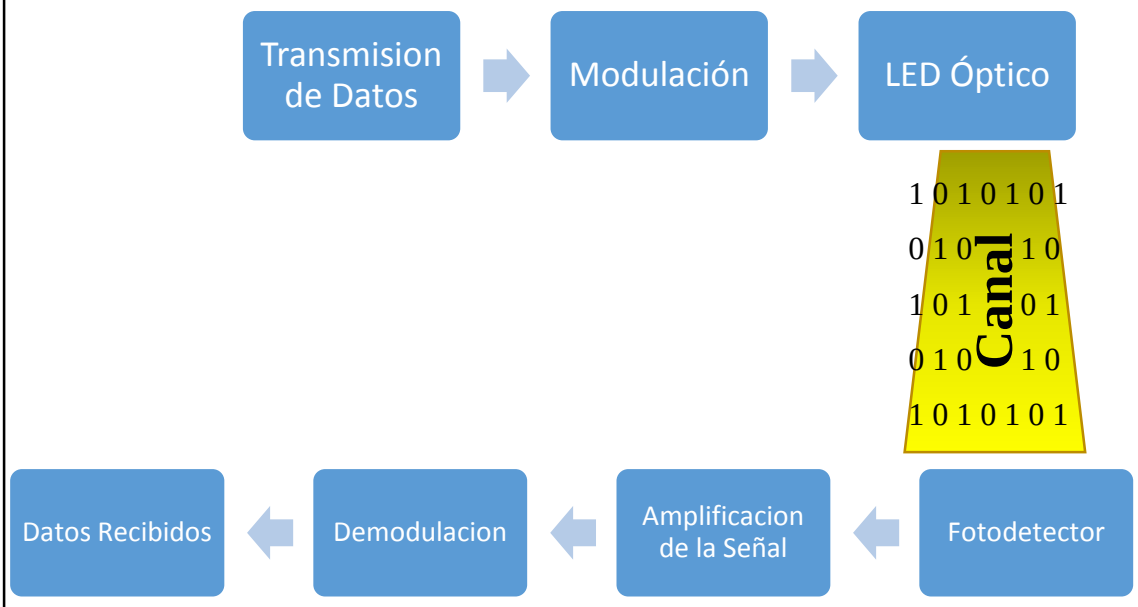


Figura 3. Modelo VLC básico: Empieza con un transmisor de datos, cuya información es modulada para convertirlas en 0's y 1's para que pase por el canal de luz y sea detectada por el Fotodetector. La señal será amplificada y demodulada para que en el receptor pueda obtener la información enviada sin errores. Elaborado por: Stalin Mina, Renatto Rugel

1.2.2 Li-Fi

Li-Fi significa Light Fidelity, traducido al español es Fidelidad de la Luz, dicho término fue creado por el profesor Harald Haas. Es una nueva tecnología perteneciente a las comunicaciones ópticas inalámbricas (OWC[2]), la cual pretende sustituir a Wi-Fi.

La principal característica de Li-Fi es el uso de la luz LED para la transmisión de señales inalámbricas por la cual se transmite información. Codifica información basándose en la frecuencia del LED, la cual transmite luz y su característica principal es de encenderse y apagarse altas velocidades hasta 10 mil millones de veces por segundo, investigado en los

[2]Comunicación óptica inalámbrica la cual es utilizada para llevar una señal y por ende información, es también conocida como VLC, la característica principal es el uso de LED´s, que permite una gran velocidad de transmisión.

laboratorios de Svilen Dimitrov (2015), que el ojo humano no podría percibirlo. Esta característica genera códigos binarios de unos ‘1’y ceros ’0’.

“La velocidad de transmisión tiene directa relación con el color de las luces LED, ya que entre más apartada del blanco cálido, más rápida es la transmisión, debido a que puede aprovecharse el espectro de frecuencia.” (Dimitrov & Harald, Principios de las Comunicaciones de Luz LED, 2015)

El matemático Arturo Campos en una entrevista para el Diario El País de México asegura que: “Li-Fi no sólo sirve para tener acceso a Internet, sino también para compartir datos en una oficina u hogar, en pruebas preliminares ya se logró una velocidad de transferencia de archivos de hasta 224 Gb por segundo. Esto supondría que se puede descargar 18 películas de alta definición en 35 segundos.” (Campos, 2016)

1.2.3 Características

Li-Fi cuenta con beneficios muy importantes como capacidad, eficiencia y seguridad que en otros sistemas de comunicación inalámbrica no encontramos completamente desarrollados, a continuación se citara dichas características.

1.2.3.1 Capacidad:

- **Ancho de Banda:** El espectro a ser utilizado para la transmisión de datos es 10.000 veces más que en radio frecuencia utilizada en la mayoría de sistemas de comunicación inalámbrica, no se necesita licencia para su uso.
- **Densidad de datos:** La luz visible puede estar en cualquier parte, permitiendo así cubrir la gran demanda y densidad de datos 1000 veces más que Wi-Fi.
- **Velocidad:** Se puede conseguir hasta 10Gbps en teoría, ya que no existe interferencia con otros dispositivos y se aprovecha muy bien el ancho de banda.

1.2.3.2 Eficiencia:

- **Económico:** No se utiliza muchos componentes, para obtener un sistema Li-Fi.
- **Energía:** Para transmitir datos requiere energía adicional insignificante, basta con la iluminación que produce el LED.

- **Medio Ambiente:** No produce ningún daño al medio ambiente ni a la salud de los seres humanos y algo importante Li-Fi funciona bastante bien en el agua. (pureLIFI, 2014)

1.2.3.3 Seguridad:

- **Interferencia:** Como no se usa frecuencias de radio, se evita la interferencia con algunos aparatos electrónicos, por ejemplo en los Aviones y los hospitales.
- **Limitación:** La señal e iluminación que se emite en Li-fi cubre un área pequeña y no atraviesa paredes, dificultando que dichas señales sean pirateadas y se exista la posibilidad de robo de datos.
- **Control:** Los datos transmitidos se pueden monitorear y pueden ser dirigidos a que dispositivos se pueden se debe enviar, no se necesita seguridad adicional

1.2.4 Comparación entre Li-Fi y Wi-Fi

Li-Fi tiene una serie de beneficios sobre los sistemas de comunicaciones Wi-Fi, se podría decir que Li-Fi seria el sustito, pero no es así según pureLiFi (2014), dice que la tecnología de Li-Fi seria complementaria y por ahora no será sustituto de Wi-Fi, sobre todo si su uso es para transmitir gran cantidad de datos.

- Li-Fi puede ofrecer hasta 1Gbps, incluso más rápido que la fibra óptica empleada en Wi-fi.
- Cualquier bombilla con tecnología LED, puede convertirse en un Router.
- Es más barato que Wi-Fi, inclusive 10 veces más barato. (Campos, 2016)
- Es seguro ya que la luz no atraviesa las paredes, así que no existiría robo de información.
- No se requiere las famosas frecuencias que se utiliza en Wi-Fi, ya que el rango que se utiliza en Li-Fi son libres.
- No se requiere muchos elementos para obtener un sistema de comunicación Li-Fi.

Tabla 1. Comparación Li-Fi vs Wi-Fi

Parámetros	Li–Fi	Wi-Fi
Velocidad Máxima	Más de 1Gbps	150 Mbps
Frecuencia	Más de 100 THz	2,4 GHz - 5 GHz
Medio de Transmisión	Luz	Aire
Alcance	Menos de 10 metros	Más de 10 metros
Costo	Económico: No requiere contratar licencias. La instalación no es complicada ni costosa	Costoso a comparación de Li-Fi: Se requiere licencias para bandas de radio. Su instalación puede ser costosa, dependiendo su topología.
Privacidad	Las paredes, muebles, techo, bloquea el paso de la luz, logrando así una transmisión más segura y privada.	Las señales RF traspasan, paredes y techos disminuyendo la seguridad de los datos.
Seguridad	Alta	Media
Impacto Ambiental	Bajo	Medio
Recursos /Dispositivos	Bajo	Medio

Nota: Cuadro comparativo de los parámetros más importantes entre Li-Fi y Wi-Fi, a pesar de que Li-Fi tiene bastantes beneficios, por ahora no sustituiría a Wi-Fi
Elaborado por: (pureLIFI, 2014)

1.2.5 Funcionamiento

Se aplica corriente constante a los LED's, el cual permite la luz visible, se utiliza esto para la transmisión de datos.

El LED es semiconductor, permitiendo la salida óptica o salida de los datos codificados a velocidades muy altas.

La información es transmitida de forma binaria 0's y 1's, esto se da por el encendido y apagado de millones de veces por segundo del LED, a la vez brinda iluminación donde esté ubicado.

Un fotodetector colocado en el receptor en otras palabras Laptop o celular captara la información convertida en 0's y 1's y será el encargado de convertirlo nuevamente a corriente eléctrica para que pueda ser decodificado y obtener los datos transmitidos. Todo este proceso es imperceptible al ojo humano. Como se observa en la figura 4.

Funcionamiento Li-Fi

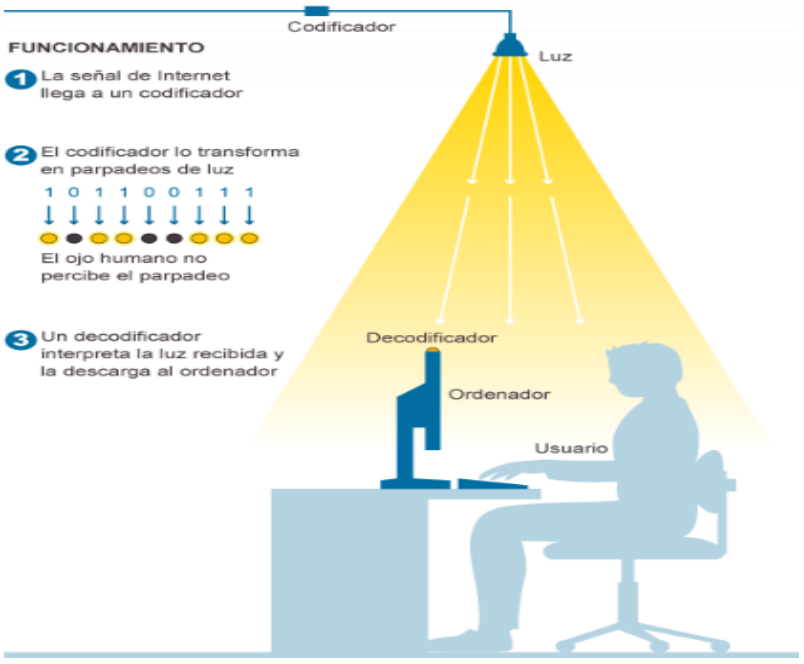


Figura 4. Funcionamiento de Li-Fi.
Desde la codificación de la información hasta su decodificación.
Elaborado por: (Longás, 2016)

La distancia entre el emisor y el receptor no debe pasar los 10 metros.

Si existe o se aplica corriente variada al LED la intensidad de la luz se atenúa y afectara a la transmisión e inclusive a la perdida de datos. (pureLIFI, 2014).

Algo importante es que no existe interferencia con otros sistemas de comunicaciones ya que los dispositivos actuales trabajan en otras frecuencias, resultando en un menor consumo de energía eléctrica y en un menor costo económico.

1.2.6 Ventajas de Li-Fi

- Li-Fi no tiene interferencias con las frecuencias RF que utilizan la mayoría de aparatos electrónicos como celulares, Smart TV, etc; pudiendo ser utilizado con Wi-Fi.
- Su velocidad de transferencia es impresionante, va desde los 15Mbps hasta los 20 Gbps, más rápido que utilizar Fibra Óptica que tiene una velocidad de hasta 10Gbps.

- En la infraestructura de Li-Fi no se necesita cables, esto resulta económico en caso de mantenimientos o reparaciones.
- El ancho de banda es un punto importante ya que no se divide entre los usuarios, así que si el sistema Li-Fi es de un 1Gbps, cada usuario tendrá dicha velocidad.
- Ahorro económico ya que aparte de tener iluminación en su casa u oficina, se cuenta con señal Li-Fi.
- Li-Fi se puede utilizar en ambientes donde ahora no es posible por problemas de interferencias o la tecnología no soporta como es los aviones y agua. (pureLIFI, 2014)

1.2.7 Desventajas de Li-Fi

- El alcance es muy corta ya que la iluminación del LED, no cubre más allá de los 10 metros.
- Si se coloca un objeto entre el emisor (LED) y el Receptor (Celular), la señal se corta y se perdería la conexión. Las ondas de luz no traspasan objetos ni paredes.
- Si existe luz solar, Li-Fi no funcionara en su totalidad y podría tener desconexiones o si se está transmitiendo o descargando datos se podría perder o llegar incompleta. (Campos, 2016)

1.2.7 Mitos de Li-Fi

- Las velocidades que se pueden obtener va a depender mucho del Proveedor del servicio de Internet ISP, ya que ellos limitaran la velocidad al crear planes de internet.
- En los laboratorios de PureLiFi se indica que Li-Fi: No es “100 veces más rápido que la conexión Wi-Fi.”. El último estándar WiFi (IEEE 802.11ad) o Wi-GIG soporta velocidades de datos de 7 Gbps. (pureLiFi, 2014)

1.3 Estándar IEEE 802.15.7

La IEEE establece la norma 802.15.7 para los sistemas de comunicación inalámbrica óptica de manera global, ofreciendo en el espectro de la luz visible para el ojo humano, un rango de 400-800 THz sin licencia, para ser utilizados y así evitar la interferencia y saturación que existe en los sistemas de comunicación que utilizan Radio Frecuencia.

El Estándar IEEE 802.15.7 define muy claramente dos aspectos importantes como es la Capa Física PHY y la capa de Control de Acceso conocida también como MAC. Con las cuales se logra altas velocidades para poder ofrecer servicios de audio, video y todo tipo de dato. (Quinatana Sánchez, 2012)

En el estándar se establece 3 tipos de capa física (PHY) cada uno con una característica en particular, que se dará a detalle a continuación:

- **PHY 1:** Creada para trabajar al aire libre con velocidades bajas, que va desde 11,67 kbits a 267,6 kbits.
- **PHY II:** Trabaja con velocidades a partir de: 1,25 Mbits a 96 Mbits, para ser empleados en lugares internos.
- **PHY III:** Trabaja en velocidades desde los 12 Mbits hasta los 96 Mbits. Utiliza un sistema MIMO múltiples entradas con múltiples salidas. Esto quiere decir que podrá trabajar en sistemas donde exista varias fuentes de luz como transmisores y varios receptores que recogerán la señal con información. (Rajagopal, Roberts, & Lim, 2012)

En PHY I y PHY II se trabaja con dos tipos de Modulación digital de Amplitud como son: Modulación Encendido-Apagado (OOK) y Modulación por posición de Pulso (PPM). Adicional utilizan la comunicación SISO, en otras palabras una entrada y una salida (Un transmisor un receptor), como se puede ver en la figura 5.

Se utiliza codificación Manchester [3]en las Capas PHY I y PHY II, en la cual se incluye un reloj para poder transmitir datos y no se pierda en el paso del emisor al receptor.

[3] Es un método de codificación eléctrica de una señal binaria en el que en cada tiempo de bit hay una transición entre dos niveles de señal. Es una codificación auto sincronizada, ya que en cada bit se puede obtener la señal de reloj, haciendo posible una sincronización precisa del flujo de datos. (Quinatana Sánchez, 2012)

Comunicación SISO

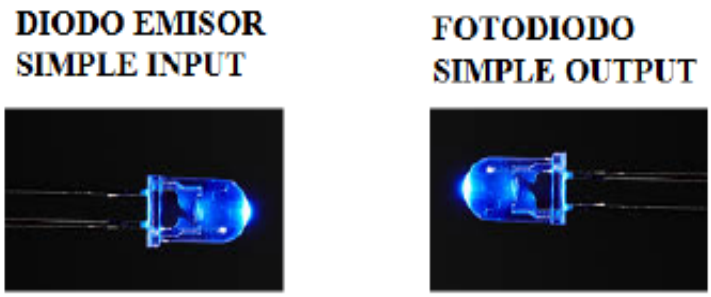


Figura 5. Comunicación SISO.
En un solo receptor. Se utiliza un diodo en el emisor y un fotodiodo que capte la señal de luz en el Receptor.
Elaborado por: (Haro Soledispa & Buenaño Medina, 2016)

1.4 Elementos en un Sistema Li-Fi

1.4.1 Diodo LED

El diodo es un componente electrónico semiconductor, por el cual pasa corriente eléctrica. El LED es un tipo especial de diodo, que al ser polarizada correctamente y pasar corriente continua (CC), produce luz. (Santillo, 2012)

Generalmente un LED está cubierto de un plástico resistente más fuerte que el vidrio, que en mucho de los casos están pintados de diferentes colores. Como se observa en la figura 6.

Diodo LED

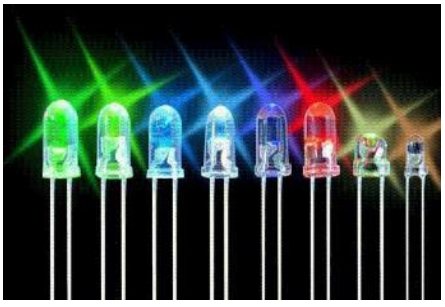


Figura 6. Diodos de Diferentes Colores
Elaborado por: (Greenpeace, 2012)

La intensidad de luz va depender de la corriente que atraviesa por el LED, que debe tener un voltaje desde 1,8 hasta 3.8 voltios, si se sobrepasa este rango podría quemar este

componente, todo va depender del material con el cual se fabricó, generalmente están elaborados de materiales semiconductores como arseniuro de aluminio y galio (AlGaAs) o el arseniuro fosfuro de galio (GaAsP) y el color de luz que emite. (Haro Soledispa & Buenaño Medina, 2016)

1.4.1.1 Características del LED

Las características de un LED, depende mucho del material con la que se fabricó. A continuación se detallara en la tabla 2. Las características en función al material y su color.

Tabla 2. Características Generales del Diodo Led

Color	Material	Voltaje	Consumo
Infrarrojo	Arseniuro de Galio (GaAs)	1,8 V – 2.1 V	10 mA
Rojo	Arseniuro de Galio y Aluminio (AlGaAs)	1,8 V – 2.1 V	10 mA
Anaranjado y Amarillo	Arseniuro fosfuro de Galio (GaAsP)	1,8 V – 2.1 V	10 mA
Verde	Fosfuro de Galio (GaN)	1,8 V – 2.1 V	10 mA
Azul	Seleniuro de Cinc (SiC)	3 V 3,8 V	10 mA
Blanco	Arseniuro de Galio (GaAs)	3 V 3,8 V	10 mA

Nota: Características del Led en función al material y su color
Elaborado por: (Greenpeace, 2012 y Peñafiel Peñafiel, 2015)

1.4.1.2 Ventajas del LED

- Bajo consumo de energía
- Mayor tiempo de vita útil
- No se recalienta, no sufriría daño ni el LED ni su alrededor.
- No contiene mercurio, no afecta a la salud de los seres humanos

1.4.2 Arduino Mega 2560

Arduino Mega es una tarjeta open-source, que cuenta con un microcontrolador modelo Atmega2560, es muy utilizado para proyectos autónomos o proyectos que necesiten la comunicación con una PC, el único límite para desarrollar proyectos es la imaginación.

En la figura 7 se puede visualizar la tarjeta Arduino. (Arduino, arduinoBoardMega, 2017)

Tarjeta Arduino Mega2560

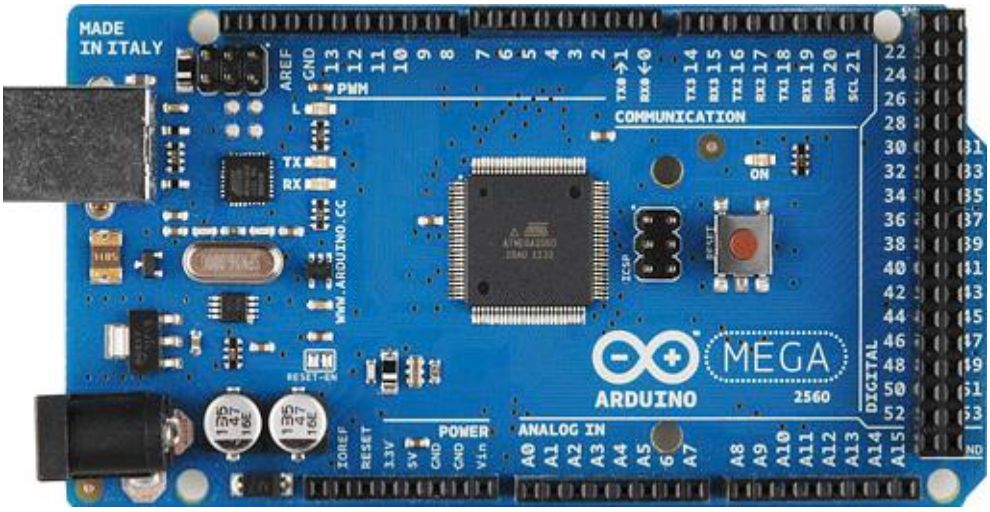


Figura 7. Tarjeta Arduino Mega2560
Elaborado por: (Arduino, arduinoBoardMega, 2017)

1.4.2.1 Características Arduino Mega 2560

Las características que tiene este modelo de Arduino el Mega 2560, permite mayores velocidades de transmisión, en la tabla No 3 se enumera las más importantes.

Tabla 3. Características Arduino Mega 2560

Parámetros	
Microcontrolador	ATmega1280
Fuente de Alimentación	5v para dar energía al microcontrolador y componentes de la tarjeta.
Pines Digitales I/O	54 Pines
Memoria Flash	128 Kb para almacenar el código, del cual utiliza 4Kb para el arranque
Longitud	101.52 mm
Peso	37 g
Programación	Su línea de código está basado en C, para la cual se compila en un software Monitor propietario de Arduino.

Nota: Características Arduino Mega 2560
Elaborado por: (Arduino, arduinoBoardMega, 2017)

1.4.3 Módulo Wi-Fi ESP8266

Este módulo fue diseñado para ser utilizado en proyectos para IOT, ya que es muy sencillo de utilizar y dispone de las características necesarias para conectarse a Internet mediante comandos de texto AT (instrucciones de comunicación entre el hombre y terminal). Adicional el costo comprende entre el $7 y $9 que es mucho más económico que otros módulos de acceso de Internet, utilizados para Arduino. (PromeTec, 2015)

El ESP8266 ya es catalogado un micro ya que cuenta con un procesador interno, con la cual ya puede trabajar de manera autónoma, ya que sus pines no tienen un propósito en general y pueden ser programadas a conveniencia.

Módulo Wi-Fi ESP8266

Figura 8. Vista frontal del Módulo ESP8266
Elaborado por: (PromeTec, 2015)

1.4.3.1 Características de Wi-Fi ESP8266

- El módulo debe ser alimentado a 3,3v, para que se pueda activar el mismo.
- Tiene integrado la pila de protocolos TCP/IP.
- Ahorra energía, ya que consume energía <10uA.
- Tiene procesador de 32 bits. (PromeTec, 2015)
- Soporta seguridad WPA/WPA2
- Trabaja en temperaturas -40C° - 125C°

1.4.4 Fotodiodo

El fotodiodo es utilizado en aplicaciones para detectar luz, ya que convierte la luz en electricidad. Es un semiconductor sensible a la presencia de la luz visible inclusive de la infrarroja. Para su correcto funcionamiento va depender mucho de su polarización que debe ser inversa (Polarización PN).

La corriente en un fotodiodo pasa de manera inversa, para la cual la corriente aumenta cuando se aumenta la luz en el fotodiodo. (Unicrom, 2016)

1.4.4.1 Diferencia entre un Fotodiodo y LDR

Los fotodiodos son más rápidos para detectarla luz visible, además detecta con rapidez a los cambios de ausencia de luz. Los fotodiodos son utilizados en lectores de CD, lectores de códigos de barras y transmisión rápida de datos. (Unicrom, 2016)

Fotodiodo

Figura 9. Fotodiodo
Elaborado por: (Wikipedia, 2011)

1.5 Comunicación Serial

La Comunicación serial no es más que él envió o recepción de datos un bit a la vez continuamente en un canal o medio de comunicación. Generalmente es utilizado para envió de datos en formato ASCII.

1.5.1 Transmisión Asíncrona

En este tipo de comunicación el receptor no sabe a qué momento recibe la información enviada por el emisor ya que él decide cuándo empieza la transmisión de datos por el

canal. Cada carácter transmitido utiliza un bit de cabecera al inicio y uno o dos bits de parada para determinar el final del carácter. (Stallings, 2012, págs. 565-566)

El objetivo de estos bits extras es que el emisor y el receptor puedan sincronizar sus relojes y no se pierdan bits al momento de decodificar. Como se puede ver en la figura 10.

Transmisión Asíncrona

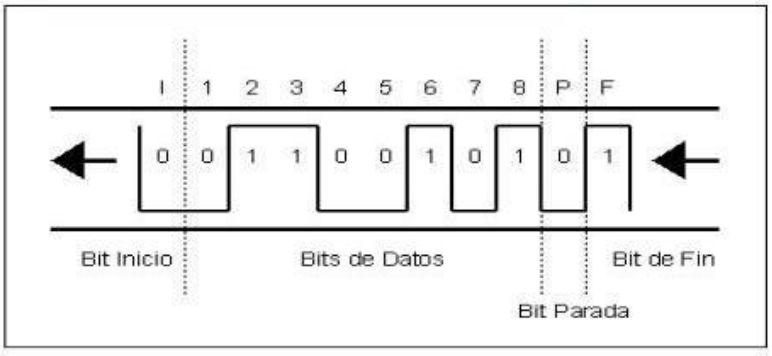


Figura 10. Transmisión Asíncrona
Elaborado por: (Rodriguez Jimenez, 2011)

1.5.1.1 Bit de Inicio

Bit adicional que va al principio de la transmisión para indicarle al receptor que empezó el envío de la información, es ahí cuando el receptor debe empezar a recibir e interpretar los datos que envía el receptor.

1.5.1.2 Bit de Parada

Es el bit que va al final de la transmisión para indicar que finalizo el mismo al receptor, puede ser 1 a 2 bits.

1.5.1.3 Bit de Paridad

Este bit permite detectar errores al momento de la transmisión.

1.6 Parámetros incidentes dentro del Sistema Li-Fi

1.6.1 Ancho de Banda

Se refiere a la cantidad de datos que se transmiten, por el medio de comunicación en un periodo de tiempo. El ancho de banda no es más que la diferencia o el intervalo entre la frecuencia más alta y la más baja que se encuentran en la señal. Se mide en Hz.

1.6.2 Banda Base

Se conoce al conjunto de señales que no necesitan ser moduladas a la salida del emisor, en otras palabras son señales transmitidas en su señal original. Son utilizadas para transmisiones a cortas distancias.

1.6.3 Velocidad de Transmisión

Diferencia entre la cantidad de información transmitida y el tiempo que se emplea para la misma. La velocidad de transmisión va depender de dos factores: Ancho de Banda y calidad de canal. Generalmente se mide en bits/segundo bps. (Stallings, 2012)

1.6.4 Tasa de Error Binario (BER)

El BER no es más que el número de bits errados de un conjunto de bits recibidos, dividió para el número total de bits transmitidos en un intervalo de tiempo. (International Centre for Theoretical Physics, 2011)

Mediante este cálculo se permite analizar el rendimiento de un sistemas de comunicaciones, frecuentemente este tipo de error se da por el ruido.

1.7 Lenguaje de Programación C++

El lenguaje de programación C++ es definido como un lenguaje de nivel medio, ya que manipula datos como caracteres, números, bits y bytes. Este lenguaje es muy escrito en su estructura ya que diferencia mayúsculas de minúsculas y para finalizar una línea de código se coloca el signo de ‘;’.

Se desarrolló el programa de compresión de Huffman en el lenguaje C++, ya que el IDE de Arduino solo soporta líneas de programación en C o C++.

1.7.1 Estructura de Datos

La estructura de datos permite guardar en una sola variable, varios valores ordenados, adicional permite guardar los datos en memoria y así se podría determinar en qué posición y orden se guardó dicho dato y extraerlo conociendo su posición sin utilizar mucho memoria del sistema.

Dentro de las estructuras de datos existen otro tipo de estructuras, que se las conoce como nodos, los cuales agrupan los datos y apuntadores, utilizados en todo el programa. En la figura 11 se ve el ejemplo de la estructura de un nodo. En donde *nodo_izq, apunta a un nodo, así se podrán utilizar en cualquier tipo de estructura de datos.

Estructura de un Nodo

```
struct nodo_t {
struct nodo_t *nodo_izq, *nodo_der;
int frec;
} *nodo;
```

Figura 11. Ejemplo Estructura de un Nodo
Elaborado por: Stalin Mina, Renatto Rugel

1.7.2 Apuntador

Los apuntadores nos permite crear Arrays con tamaños dinámicos, en otras palabras el tamaño se establecerá mientras se compila el programa en tiempo real y según la necesidad.

1.7.2.1 Características de un Apuntador

Puntos importantes al momento de crear un apuntador:

- Hay que tomar en cuenta que el tipo de apuntador debe coincidir con la variable que se va apuntar.
- Se utiliza asterisco (*), para indicar que se utilizar el valor que se guardó en dicha posición de memoria. Que es uno de los más utilizados al momento de recorrer un Array.

En la figura 12, se tiene un ejemplo que se crea un Array tipo char llamado código en la cual se extraerá el valor de la posición 128.

- Se utiliza ampersand (&), para conocer la posición de memoria de cualquier valor.

Ejemplo de Apuntador

```
char *codigo[128]
```

Figura 12. Ejemplo de Apuntador
Elaborado por: Stalin Mina, Renatto Rugel

1.7.3 Arrays

Los Arrays permiten almacenar valores (números, caracteres, letras), en posiciones de memorias de acuerdo al orden que se almacena y permite acceder a dichos valores de manera rápida y optimizando recursos. (Deitel & Deitel, 2012)

Este tipo de estructura es necesario en nuestro proyecto ya que nos permite ir guardando cada carácter ingresado con su posición, y así recorrer para determinar si se repite. Con ello al momento de decodificar nos permitirá comparar en qué posición van cada carácter. En la figura 13 se tiene un ejemplo de un Array llamado buf, en el cual se podrá ingresar 1024 elementos de tipo char.

Ejemplo de Arrays

```
char buf[1024];
```

Figura 13. Ejemplo de Arrays
Elaborado por: Stalin Mina, Renatto Rugel

1.7.4 Cola de Prioridad

En este tipo de estructura de datos, se atiende a los valores de acuerdo al orden establecido por una prioridad asociada a cada valor. Si algunos valores tienen una prioridad igual, se atiende según el orden en que se ingresó u ocupen. (Cruz, 2015).

En el programa se utiliza cola de prioridad para insertar cada carácter, según la frecuencia que tenga, para establecer cuantas veces aparece un carácter en el texto ingresado. E ir ordenando de mayor a menor su frecuencia como se observa en la figura 14.

Cola de Prioridad

```
void insertar_nodo_cola(nodo n) {
        int j;
    int i = fin_cola++;

    if (cola_prin[j]->frec <= n->frec) {
    cola_prin[i] = cola_prin[j], i = j; } }
```

Figura 14. Ejemplo de Cola de Prioridad
Elaborado por: Stalin Mina, Renatto Rugel

1.7.5 Arboles Binarios

Se trata de árboles que contiene como máximo dos nodos, que son conocidos como nodo hijo izquierdo y nodo hijo derecho; que pueden tener valores o pueden ser nulos. (Fernández, 2012). Cada nodo es apuntado por un nodo principal, cumpliéndose así la característica principal de ser una estructura jerarquizada. En el capítulo 3 apartado 3.2 Algoritmo de Codificación Huffman, se explica cómo se crea el árbol.

Una parte fundamental dentro de la construcción del árbol de Huffman son cada una de las hojas y nodo raíz, para su representación dentro de la programación se utilizan las denominadas Estructuras dentro del lenguaje C, estas permiten crear objetos que contengan más de un tipo de dato, pues almacenarán el carácter a codificar, su frecuencia y sus nodos hijos en el lado izquierdo y derecho, que permitirán las construcción de las ramas restantes del árbol.

1.7.5.1 Tipos de Nodos

- **Nodo Raíz:** Nodo principal que tiene hijos nodos.
- **Nodo Hoja:** Nodo que no tiene hijos y depende el nodo raíz.

1.7.6 Librerías

Las librerías contienen códigos de funciones, que permiten hacer uso del mismo sin necesidad de escribir muchas líneas de programación, solo con importar el fichero podemos hacer uso de sus funciones como son: manejo de ficheros, escribir en pantalla, manejo de datos, funciones matemáticas, es posible crear librerías uno mismo según la necesidad.

Las extensiones más comunes son: .lib, .dll, .h.

Para el proyecto se creó una librería llamada "huffman.h", en la cual se incluye las funciones para crear el árbol, codificar, decodificar, para que en el programa principal no ocupe mucha memoria al momento de compilar.

1.7.6.1 Sintaxis para declarar una Librería

Generalmente se declaran las librerías al principio del programa, antes de declarar variables o funciones, para que al momento de correr el programa el compilador sepa que librerías se usaran y debe exportar. A continuación se indica las formas correctas para llamar una librería:

Sintaxis para declarar una Librería en C++

```
#include <stdio.h>
#include "string.h"
```

Figura 15. Sintaxis para declarar una Librería en C++
Elaborado por: Stalin Mina, Renatto Rugel

1.7.6.2 Librerías implementadas

A continuación se describirán las librerías utilizadas en el proyecto:

- **Fstream:** Permite la manipulación de ficheros; abrir, leer escribir, etc.
- **Stadio:** Permite manipular cualquier tipo de datos.
- **String:** Permite manipular datos de tipo alfanuméricas, manejo de cadenas de objetos. Adicional permite la búsqueda de caracteres en cadenas.

- **Arduino:** Contiene todas las funciones nativas para poder hacer uso de la placa de Arduino: inicialización, manejo de datos, envió y recepción de datos.
- **Huffman**: Fichero creado para hacer uso de las funciones para poder codificar y decodificar los datos ingresados.

1.7.7 Funciones

Las funciones son de gran utilidad ya que permiten reutilizar, ahorrar líneas de código y hacer más estable al programa, pues al crear una función, esta puede ser utilizada las veces necesarias con solo llamarlas. Usualmente las funciones reciben parámetros, para ser usados ya sea para operaciones matemáticas o manejo de cadenas para luego retornar un valor.

Capítulo 2

Análisis del Algoritmo Compresión de Huffman

En este capítulo se realizar un breve análisis sobre el algoritmo de Compresión de Huffman, para la cual se realizara explicara su codificación y decodificación con un pequeño ejemplo.

2.1 Codificación Huffman

Es una técnica que fue desarrollada por David Huffman para la compresión de datos en la cual se generan códigos llamados códigos Huffman para cada uno de los caracteres contenidos en una fuente a codificar mediante la construcción previa de un árbol binario. Los códigos generados tienen la característica de ser códigos prefijos.

El procedimiento de Huffman se basa en dos observaciones respecto a los códigos prefijos óptimos:

2.1.1 Código Prefijo

Es aquel código de tamaño variable cuya combinación de bits, encargado de representar a un carácter, no es prefijo de otro, esto quiere decir que su combinación no es la base para la combinación de otro carácter, aun cuando todos pertenecen a un mismo conjunto.

2.1.2 Arboles Binarios

Estructuras jerárquicas en las que podemos identificar a un nodo raíz para cada subárbol y nodos hijos llamados subárboles derecho e izquierdo, respectivamente.

2.1.3 Código Óptimo

- En un código óptimo, los caracteres que se producen con mayor frecuencia de ocurrencia (tienen una mayor probabilidad de frecuencia) tendrán palabras código más cortas que aquellos caracteres que ocurren con menos frecuencia.
- En un código óptimo, los dos caracteres que ocurren con menor frecuencia tendrán la misma longitud.

Si los caracteres que ocurren más a menudo tienen palabras código más largas que las palabras código para caracteres que ocurren con menor frecuencia, el número promedio de bits por carácter sería mayor que si las condiciones se invirtieran. Por esta razón, un código que asigna palabras código más largas a caracteres que ocurren con más frecuencia no puede ser óptimo. (Sayood, Introduction to Data Compression, 2012, págs. 54-55)

2.2 Algoritmo de Codificación Huffman

El proceso que sigue el algoritmo se inicia mediante la creación de una lista con todos los símbolos alfabéticos en orden ascendente respecto a su frecuencia de ocurrencia. A continuación, se procede a construir el árbol con un símbolo en cada hoja, de abajo hacia arriba.

En cada paso se seleccionan los dos símbolos con la frecuencia de ocurrencia más pequeña; se añaden a la parte superior del árbol parcial, se eliminan de la lista y se reemplazan por un símbolo auxiliar que representa a los dos símbolos originales.

Cuando la lista se reduce a un sólo símbolo auxiliar, el cual representa a todo el alfabeto, se considera al árbol como completo.

Posteriormente, se recorre el árbol de arriba hacia abajo, como si de un camino se tratase, para determinar los códigos a asignar a cada uno de los símbolos.

El proceso se ilustra a continuación mediante un ejemplo. Se inicia con un conjunto de 5 símbolos de un alfabeto:

A = {a, r, t, b, u}, cuya frecuencia de ocurrencia es la siguiente:

F (a) = 4; F (b) = 2; F (t) = 2; F (r) = 1; F (u) = 1

Cadena= aaaabbttru

Se procede a ordenar los símbolos respecto a su frecuencia de ocurrencia y se procede a colocar su frecuencia dentro de cada hoja como se muestra en la figura 16.

Símbolos ordenados según su frecuencia

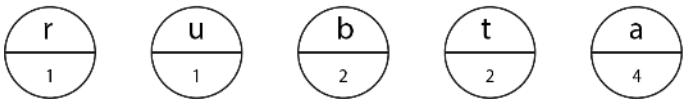


Figura 16. Hojas ordenadas respecto a la frecuencia de cada símbolo alfabético del conjunto.
Elaborado por: Stalin Mina, Renatto Rugel.

Después, es posible la agrupación como se muestra en las siguientes figuras:
Combinamos dos de las hojas cuyas frecuencias sean las más bajas en una, en este caso r se combina con u al poseer la frecuencia más pequeña (1), ambas se sustituyen por el símbolo auxiliar 0 o Null y con una frecuencia de 2.

Combinación de hojas

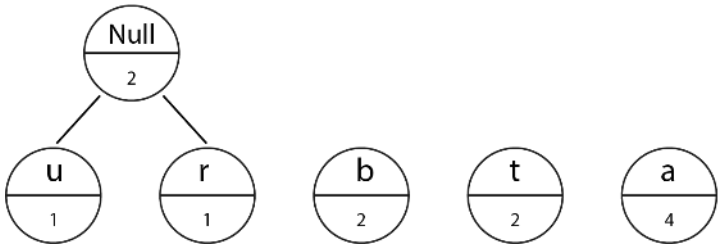


Figura 17. Combinación de las dos primeras hojas de menor frecuencia.
Elaborado por: Stalin Mina, Renatto Rugel.

Ahora quedan cuatro símbolos disponibles:

• a con una frecuencia de ocurrencia de 4
• b y t con una frecuencia de ocurrencia de 2
• Null con una frecuencia de ocurrencia de 2

Se seleccionan y agrupan las hojas con la frecuencia de ocurrencia menor nuevamente siendo estas b y Null, y reemplazando por el símbolo auxiliar Null cuya frecuencia es ahora 4. Como se observa en la figura 18.

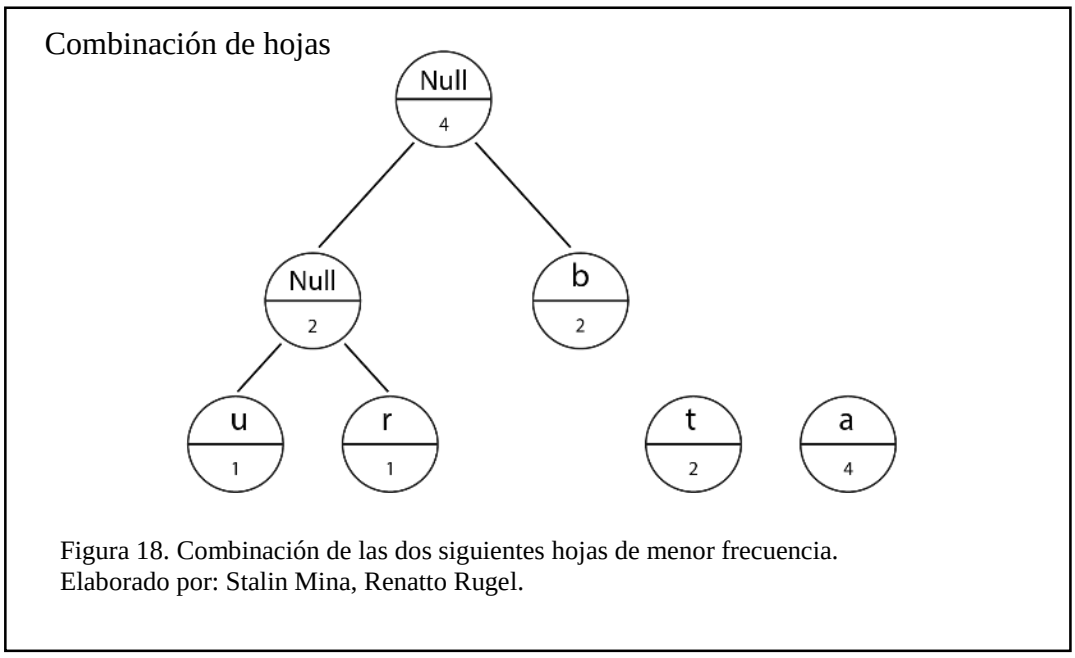


Figura 18. Combinación de las dos siguientes hojas de menor frecuencia.
Elaborado por: Stalin Mina, Renatto Rugel.

En este momento tenemos tres símbolos, a, t y Null con frecuencias de ocurrencia de 4, 2 y 4 respectivamente. Se procede ahora a seleccionar los dos nodos de menor frecuencia, no se tiene en cuenta el último nodo creado, t y a, combinándolos y reemplazándolos por el símbolo auxiliar Null cuya frecuencia de ocurrencia es ahora 6.

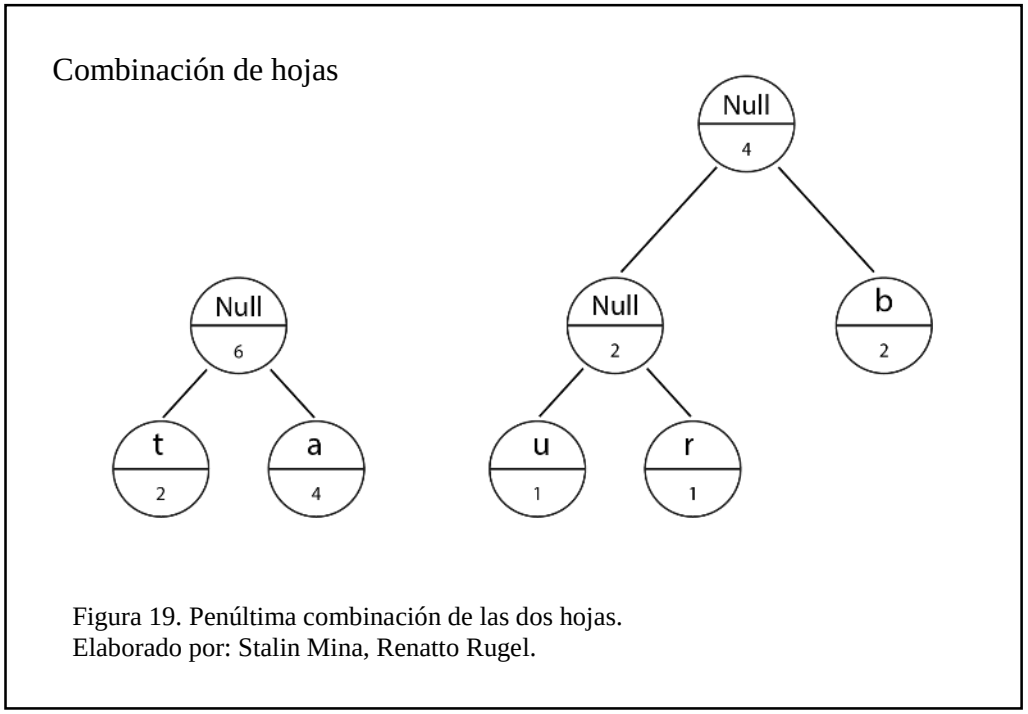


Figura 19. Penúltima combinación de las dos hojas.
Elaborado por: Stalin Mina, Renatto Rugel.

Finalmente, se combinan los dos subárboles restantes, Null con una frecuencia de 6 y un segundo subárbol Null con una frecuencia de 4, nuevamente se utiliza el carácter Null para representar la unión de los dos subárboles finales y cuya frecuencia es ahora de 10, completándose así el árbol como se indica en la figura 20.

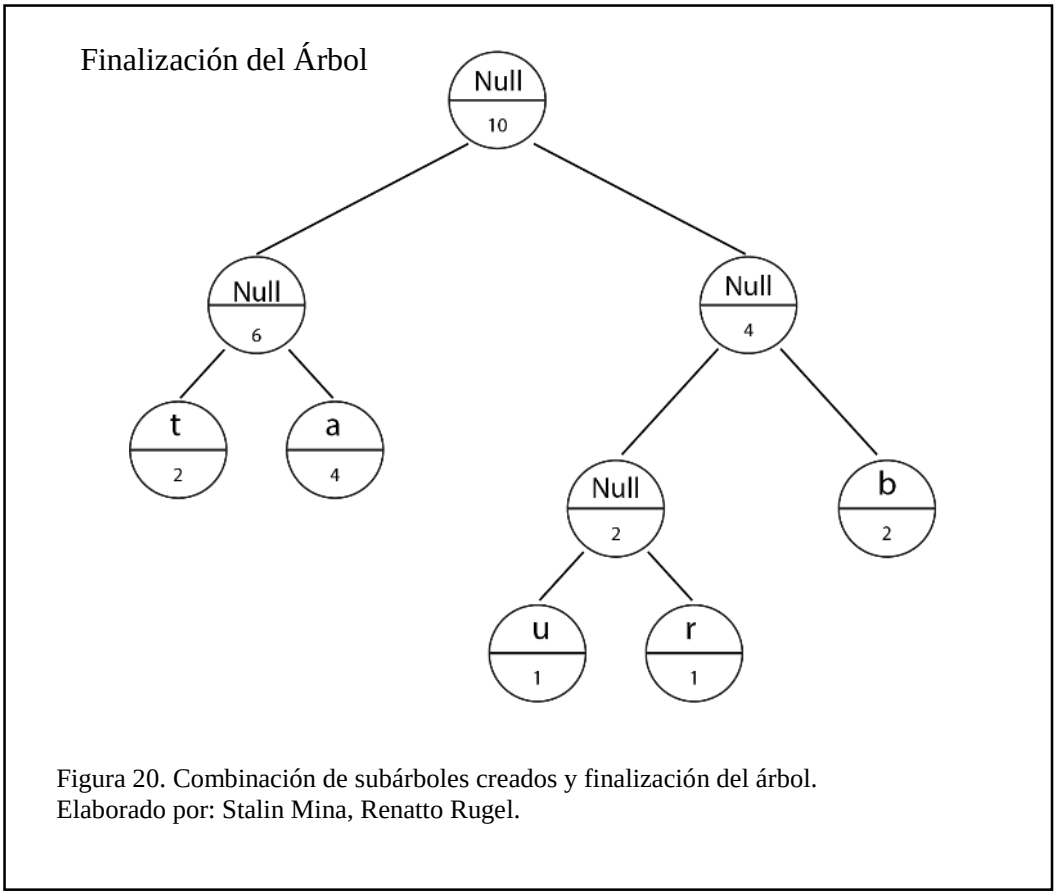


Figura 20. Combinación de subárboles creados y finalización del árbol.
Elaborado por: Stalin Mina, Renatto Rugel.

Es así que se obtiene la raíz principal en la parte superior y descendiendo, sus ramas de izquierda y derecha, que a su vez pueden ser la raíz de otro subárbol o una hoja.

Cada símbolo alfabético está acompañado por su frecuencia de aparición.

Para finalizar se procede a colocar 0s a las ramas del lado izquierdo y 1s a las ramas del lado derecho, las cuales permitirán construir cada uno de los códigos para los símbolos como se indica en la figura 21.

Árbol y sus hojas con su respectivo código

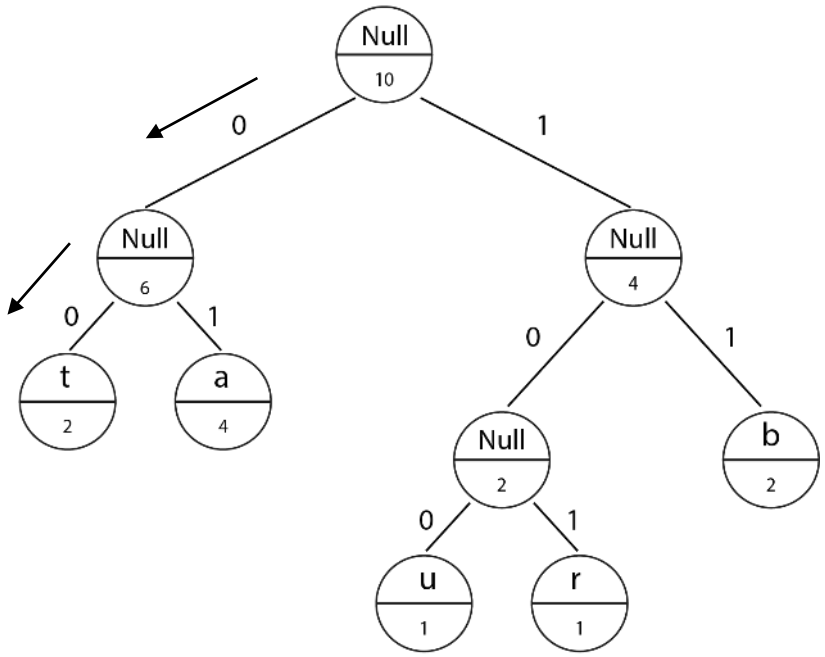


Figura 21. Árbol terminado colocados 0s y 1s a cada una de sus ramas.
Elaborado por: Stalin Mina, Renatto Rugel.

Para ilustrar la construcción de todos códigos prefijo que se generarán, se toma como ejemplo el camino mostrado por las flechas para el símbolo *t*, en el cual, se construye uno de los códigos deseados y que permitirá realizar la compresión de los datos, gracias al árbol construido como se aprecia en la figura 21.

Finalmente, se presentan todos los códigos generados para el conjunto inicialmente dado que son: 01, 11, 00, 101 y 100, para a, b, t, r y u, respectivamente:

Tabla 4. Símbolos con su respectiva frecuencia y código

Símbolo	Frecuencia	Palabra de Código
a	4	01
b	2	11
t	2	00
r	1	101
u	1	100

Nota: Símbolos, frecuencias y palabras código obtenidas.
Elaborado por: Stalin Mina, Renatto Rugel.

Y dándonos como resultado el código general siguiente del texto aaaabbttru recordando sus frecuencias: 010101011111000010110 0.

2.3 Decodificación Huffman

Para la decodificación del flujo de datos codificados mediante Huffman, existen algunas alternativas, siendo la primera la de ellas, la de obtener previamente los códigos prefijo que servirán para representar a cada uno de los símbolos a codificar de acuerdo a su frecuencia de ocurrencia, antes de empezar por la codificación completa de los datos; dichas frecuencias deberán ser enviadas como información adicional junto con los datos codificados, para que cualquier decodificador Huffman se capaz de recuperar los datos originales, mismos que pueden ir de desde unos cuantos bytes hasta cientos de ellos.

Un enfoque diferente es el de agregar cada uno los códigos prefijo generados, al flujo de datos codificado, algo que puede resultar difícil ya que cada código puede llegar a tener una longitud diferente; una alternativa un tanto más compleja es la de enviar el árbol de Huffman junto a los datos codificados, esto implica un aumento significativo de bytes al flujo de información que se está enviando.

Al emplear la última alternativa, se debe empezar desde la raíz del árbol, leer el primer bit del flujo de datos codificado y dependiendo de si se encuentra un 0, dirigirse a la rama asociada al 0 y si es un 1 continuar por la rama contraria hasta llegar a una hoja, en la cual se encuentra el código original sin codificar del símbolo (un código ASCII) que es devuelto por el codificador, el proceso se vuelve a repetir desde la raíz del árbol.

Capítulo 3

Análisis de Hardware y Software

En este capítulo se realiza el análisis de dos de los dispositivos candidatos a ser utilizados en el presente proyecto, Arduino Mega 2560 y Raspberry Pi 3, para el desarrollo del prototipo Li-fi, se presentan las principales características técnicas junto con los distintos pesos al momento de la toma de decisión.

3.1 Diagrama de bloques del prototipo

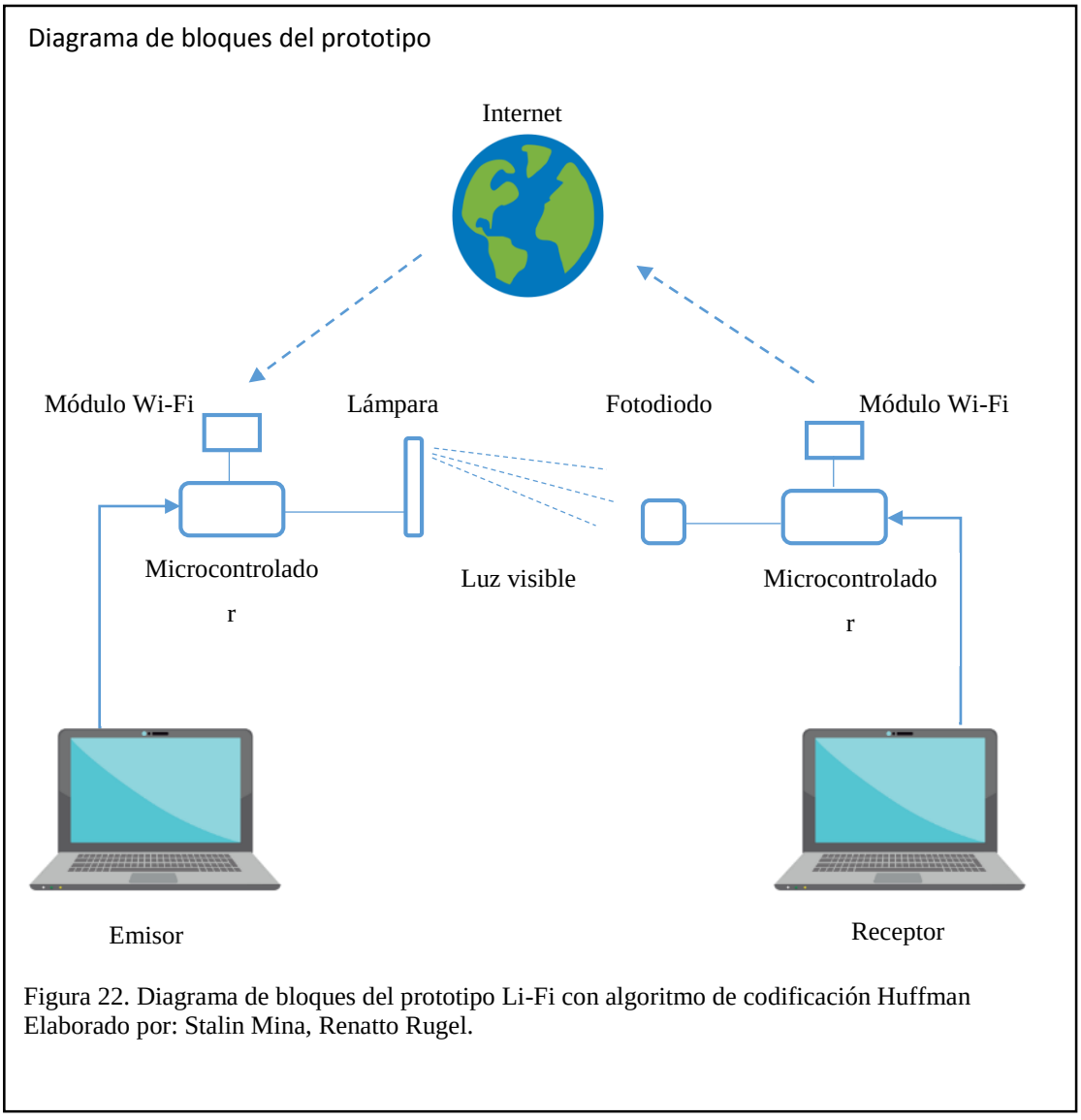


Figura 22. Diagrama de bloques del prototipo Li-Fi con algoritmo de codificación Huffman
Elaborado por: Stalin Mina, Renatto Rugel.

3.2 Descripción de las características técnicas de cada dispositivo

3.2.1 Arduino Mega 2560

Es un microcontrolador de placa de código abierto, ha sido diseñado para trabajar en aplicaciones de robótica y automatización pues su utilización y manipulación es relativamente fácil con una dimensión de 108 x 53 x 15 mm.

Cuenta con una amplia gama de módulos que no requieren de complejas configuraciones para su integración, utiliza los lenguajes C y C ++ para su programación y tienen la capacidad de trabajar conjuntamente con sistemas más grandes.

Arduino Mega 2560

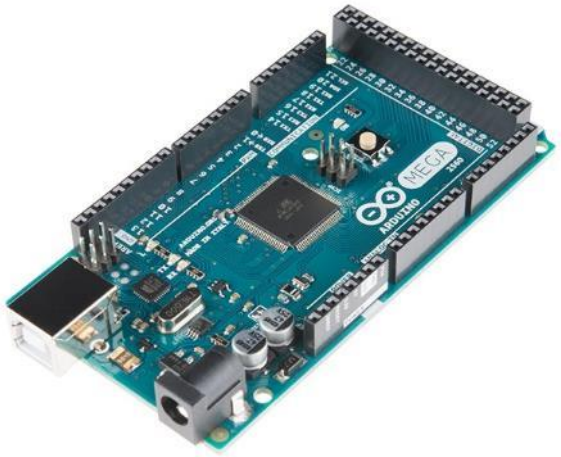


Figura 23. Arduino Mega 2560
Elaborado por: (sparkfun.com, s.f.)

3.2.1.1 Microcontrolador ATmega2560

Un microcontrolador de placa es un sistema independiente que cuenta con periféricos de entrada y salida, memoria, procesador y una unidad de almacenamiento integrados que conectado a una fuente de energía relativamente baja, es capaz de apoyar el funcionamiento del microcontrolador que posee.

3.2.1.2 Consumo de energía

Este tipo de microcontrolador es capaz de trabajar con solo 5 voltios de energía, ya sea mediante una conexión USB o una fuente de alimentación externa la cual puede provenir

desde un adaptador AC-DC (enchufe en la pared) o una batería, sin embargo es recomendable que el voltaje de entrada se encuentre entre los 5 y 12 voltios ya que pasado estos límites se puede sobrecalentarse y dañar la placa.

3.2.1.3 Memoria

Dentro del microcontrolador ATmega2560 son utilizados tres tipos de memoria:

- La primera es la denominada memoria Flash la cual se encarga de almacenar el denominado sketch o programa en Arduino con una capacidad 256 KB, este tipo de memoria es no volátil.
- Se cuenta también con una memoria estática de acceso aleatorio (SRAM) en la que se crean y manipulan las variables creadas por el programa o sketch y cuenta con una capacidad de 8 KB. Este tipo de memoria es volátil, la información se perderá cuando se retire la energía.
- Finalmente se cuenta con la memoria EEPROM con una capacidad de 4 KB, la cual se encarga de almacenar por largos periodos de tiempo la información deseada, este tipo de memoria es no volátil.

3.2.1.4 Comunicación

Para el envío y recepción de datos desde y hacia esta placa, Arduino cuenta con un software de monitorización serial que conjuntamente con dos LED para notificación, ya sea para recepción o para transmisión, se iluminan cuando se reciben o envían datos mediante el chip ATmega8U2 y la conexión USB al computador.

Para la comunicación con otros dispositivos, ya sean computadores, otras placas Arduino o incluso otros microcontroladores, el Arduino Mega 2560 cuenta con 4 dispositivos Transmisor-Receptor Asíncrono Universal (UART).

Si bien no cuenta con un puerto Ethernet básico, es posible agregar el módulo ESP8266, el cual es un módulo que cuenta con el protocolo TCP/IP integrado, permitiendo así la conectividad inalámbrica Wi-Fi con otros dispositivos e Internet.

3.2.1.5 Software

Para la programación se utiliza el Entorno de Desarrollo Integrado (IDE) conocido como Arduino Software, de código abierto, formado por un editor de texto, área de notificaciones, barra de herramientas propia para funciones comunes y opciones adicionales para la escritura de toda la programación.

El IDE permite la actualización y transferencia de nuevo código una vez conectado a Arduino.

Figura 24. Arduino IDE
Elaborado por: Arduino Software (IDE)

3.2.1.6 Precio

A continuación se presenta el precio de la placa Arduino y del módulo adicional que se requeriría en caso de ser seleccionado este tipo de tecnología.

Tabla 5. Precios de Arduino, Módulo y accesorios

Dispositivo	Cantidad	Precio
Arduino Mega 2560	1	$38,65
Módulo Wi-fi ESP8266	1	$8,40
Cable USB (A/B)	1	$4,26
Total		$51,31

Nota: Precios de Arduino y sus Módulos
Elaborado por: Stalin Mina, Renatto Rugel.

3.2.1.7 Proyectos Vinculados

Uno de los aspectos que se han tomado en cuenta para la elección del hardware y software, son los distintos casos de uso en los que se ha utilizado dichos dispositivos; a continuación se listan aquellos trabajos en los que han sido utilizados el hardware Arduino Mega 2560 en nuestro país con resultados satisfactorios y que permiten tener una guía mucho más clara y concisa para implementaciones y trabajos futuros.

Tabla 6. Casos de Uso

Proyecto	Autores	Año
Diseño e Implementación de un Prototipo de Tecnología Visible Light Communication (Li-Fi) para transmitir datos de manera Unidireccional	Karla Haro & Arturo Buenaño	2016
Desarrollo E Implementación del Algoritmo de Codificación de línea BNZS para optimizar el uso de canal de transmisión De Light Fidelity (Li-Fi)	Johanna Cumbajín & Paola Rivadeneira	2016

Nota: Casos de Uso utilizando Arduino
Elaborado por: Stalin Mina, Renatto Rugel.

3.2.2 Raspberry Pi 3

Raspberry Pi 3 es un computador de placa reducida, del tamaño de una tarjeta de crédito, que cuenta con el potente procesador Cortex-A53 de ARM, una cantidad razonable de puertos USB para la comunicación con otros dispositivos y el envío y recepción de información; una memoria RAM de 1GB, capacidad de conectarse a la red, salida de audio y video HDMI, entre otros; con tan solo contar con 85 x 56 x 17mm de dimensión.

Raspberry Pi 3

Figura 25. Raspberry Pi 3
Elaborado por: (sparkfun.com, s.f.)

3.2.2.1 Computador de Placa Reducida (SBC)

Este tipo de dispositivo es considerado como un computador completo caracterizado principalmente por ser de un tamaño relativamente pequeño, en el cual sus componentes principales tales como microprocesador y memoria entre otros, se encuentran integrados en una sola placa.

Si bien no cuentan con la misma capacidad que computadores estándar, son un elemento muy utilizado en la industria para tareas de automatización, monitorización o como equipos de control, debido a su bajo consumo de energía, capacidad, tamaño y eficiencia. Los computadores de placa reducida también cuentan con una capacidad de expansión diferente, mientras que para computadores de escritorio todo se reduce a conectar dispositivos periféricos externos en las ranuras de la placa madre, ya sea para audio, video de alta calidad, etc, para aquellos de placa reducida, los mismos ya se encuentran integrados a la placa.

3.2.2.2 Procesador Quad-Core ARM Cortex-A53

El procesador utilizado por este tipo de sistema cuenta con una velocidad de 1250 MHz y permite ejecutar la gama de distribuciones GNU/Linux para ARM, así como la versión de

Windows 10 de 32 bits debido a que este procesador ofrece un buen balance entre rendimiento y consumo de energía, conjuntamente con técnicas avanzadas para la búsqueda y acceso a los datos.

Para aplicaciones que han sido desarrolladas para aprovechar procesadores de varios núcleos, el rendimiento es capaz de mejorar un promedio de 4 veces más, especialmente para aplicaciones que trabajan ejecutando múltiples hilos.

3.2.2.3 Consumo de Energía

Este tipo de placa requiere de una mayor cantidad de energía, siendo que en funcionamiento inactivo su consumo es de aproximadamente 200mA, al utilizar la comunicación para red, ya sea mediante el módulo Wi-Fi, que representa un aumento de 170mA, o Ethernet con 40mA, el consumo va en aumento.

Al tomar en consideración la ejecución de tareas muy complejas, el consumo aumenta aún más, siendo el consumo mínimo de 200-250mA; llegando a requerir una fuente de alimentación de 5V.

3.2.2.4 Memoria

Raspberry Pi 3 cuenta con un chip de memoria RAM de baja potencia, soldada en el fondo de la placa del circuito con una capacidad de 1GB del tipo LPDDR2 diseñada para equipos móviles y con un consumo de 1,8 voltios en comparación con los tradicionales 2,5 voltios.

3.2.2.5 Comunicación

Cuenta con 4 puertos USB integrados a la placa para permitir la conectividad con dispositivos externos tales como mouse, teclado e incluso para concentradores de puertos USB y una ranura Ethernet 10/100 Base T.

No existe la necesidad de conectar antenas externas puesto que se incluye una, soldada directamente a la placa y de un tamaño suficientemente pequeño que permite adaptarse a las dimensiones de la placa pero, al mismo tiempo capaz de captar señales inalámbricas LAN y Bluetooth incluso a través de las paredes.

Es posible utilizar una comunicación serial para el envío y recepción de datos, ya sea mediante una conexión a un computador, para permitir el acceso a la consola de Linux, o

una conexión a un microcontrolador u otro periférico que cuente con una interfaz serial, para permitir a Raspberry Pi controlar dicho dispositivo.

3.2.2.6 Software

El software disponible para este mini computador es relativamente extenso ya que permite la ejecución de varias distribuciones Linux pensadas para Raspberry Pi 3 y el sistema operativo Windows 10 desde la tarjeta Micro SD.

Entre las distribuciones más populares se tiene a Raspbian, Snappy Ubuntu Core y Fedora, entre otros, siendo Raspbian, sistema basado en Debian, la más popular debido a la gran comunidad de usuarios que posee, una extensa variedad de paquetes listos para ser utilizados.

Esto brinda la posibilidad de utilizar entornos de programación tales como Eclipse y Netbeans, con la instalación previa del SDK de Java y los plugin para cada entorno.

3.2.2.7 Precio

A continuación se presenta el precio de la placa Raspberry Pi 3 y de los dispositivos adicionales que se requeriría en caso de ser seleccionado este tipo de tecnología.

Tabla 7. Precios Raspberry Pi 3 y Módulos

Dispositivo	Cantidad	Precio
Raspberry Pi 3	1	$ 46,16
Tarjeta microSD con adaptador - 16GB (Clase 10)	1	$19,95
Fuente de alimentación - 5.1V DC 2.5A (USB Micro-B)	1	$7,95
Pantalla LCD Raspberry Pi – 7’’	1	$64,95
Total		$139,01

Nota: Precios Raspberry Pi 3 y Módulos
Elaborado por: Stalin Mina, Renatto Rugel.

3.2.2.8 Proyectos Vinculados

Desafortunadamente no se cuenta con casos de éxito y uso documentados en los que Raspberry Pi 3 haya sido utilizado para prototipos de sistemas de comunicación por luz visible Li-Fi, principalmente debido a la falta de investigaciones previas en nuestro país.

3.3 Jerarquización de las alternativas

Se presentan las distintas puntuaciones asignadas a cada una de las características analizadas de los dos dispositivos elegibles, en base a los criterios de costos, características técnicas y casos de uso disponibles.

Tabla 8. Análisis de Arduino y Raspberry Pi 3

	Arduino			Raspberry Pi 3		
Criterio	Puntaje	Peso	Total	Puntaje	Peso	Total
Costos	8	0,4	3,2	5	0,4	2
Características Técnicas	7	0,3	2,1	9	0,3	2,7
Casos de éxito disponibles	8	0,3	2,4	0	0,3	0
Puntaje Total			7,7			4,7

Nota: Análisis de Arduino y Raspberry Pi 3, para determinar el idóneo para el proyecto. Dando importancia al Costo.
Elaborado por: Stalin Mina, Renatto Rugel.

Opción 2:

Tabla 9. Segundo Análisis de Arduino y Raspberry Pi 3

	Arduino			Raspberry Pi 3		
Criterio	Puntaje	Peso	Total	Puntaje	Peso	Total
Costos	8	0,3	2,4	5	0,3	1,5
Procesamiento	6	0,1	0,6	9	0,1	0,9
Consumo de energía	8	0,1	0,8	8	0,1	0,8
Memoria	5	0,1	0,5	9	0,1	0,9
Comunicación	7	0,1	0,7	9	0,1	0,9
Software	5	0,1	0,5	9	0,1	0,9
Casos de éxito	8	0,2	1,6	0	0,2	0
Puntaje Total			7,1			5,9

Nota: Segundo análisis de Arduino y Raspberry Pi 3, basándose en sus características más importantes para determinar el idóneo para el proyecto. Dando importancia al costo y al caso de uso.
Elaborado por: Stalin Mina, Renatto Rugel.

3.4 Resultados y Elección del Hardware para el prototipo

En base a la tabla anterior, se ha elegido como ganadora, a la alternativa del microcontrolador de placa Arduino Mega 2560, es la opción más factible debido a las características presentadas anteriormente y, al tratarse de un prototipo, el cual permitirá

dar a conocer los primeros resultados del trabajo conjunto de la tecnología Li-Fi con el algoritmo de codificación Huffman, no se requiere una gran cantidad de recursos, que de cierta manera, serían desperdiciados para un prototipo el cual ha presentado buenos resultados con la tecnología Arduino, adicional por el costo de sus accesorios y módulo.

Capítulo 4

Diseño e Implementación

En el presente capítulo, se explicara el diseño del escenario Li-Fi para la transmisión de datos unidireccional, implementando el algoritmo de compresión de Huffman. Se utilizaran módulos Wi-Fi para Arduino, para la conexión a la Página Web, creada para él envió de datos. En el transmisor: el Arduino codificara los datos enviados y se transmitirán por el Led; en el receptor: el fotodiodo recoge los datos codificados, pasa al Arduino decodifica los datos y sube la información a la Página Web de recepción. Anexo 1.

4.1 Diagrama del escenario Li-Fi

El diseño experimental de este proyecto técnico, utilizando la tecnología Li-Fi; es unidireccional trabajando con comunicación serial. En la cual se implementará el algoritmo de Compresión de Huffman, para mejorar el rendimiento del sistema y reducir el tiempo de transmisión de datos.

Se ha creado dos páginas Web para el envío y recepción de datos (http://unichat.ddns.net/index.php), colgados a la red gracias a un hosting gratuito.

Como se explicó en el anterior capitulo el microcontrolador elegido es el Arduino Mega 2560. Se adquirió dos Arduino en los cuales se programó tanto la codificación en el transmisor y la decodificación en el receptor, basados en el algoritmo de Huffman.

Se conectaran a las páginas Web mediante los módulos Wi-Fi ESP8266, los cuales son montados en los Arduinos y programados para que tengan salida a Internet.

Para crear el canal de transmisión por luz visible principio de Li-Fi, en el transmisor se colocará un Led y en el receptor un fotodiodo que se encargará de receptar la información y mostrarla en la Web. Como se observa en la figura 26.

Diagrama del escenario Li-Fi

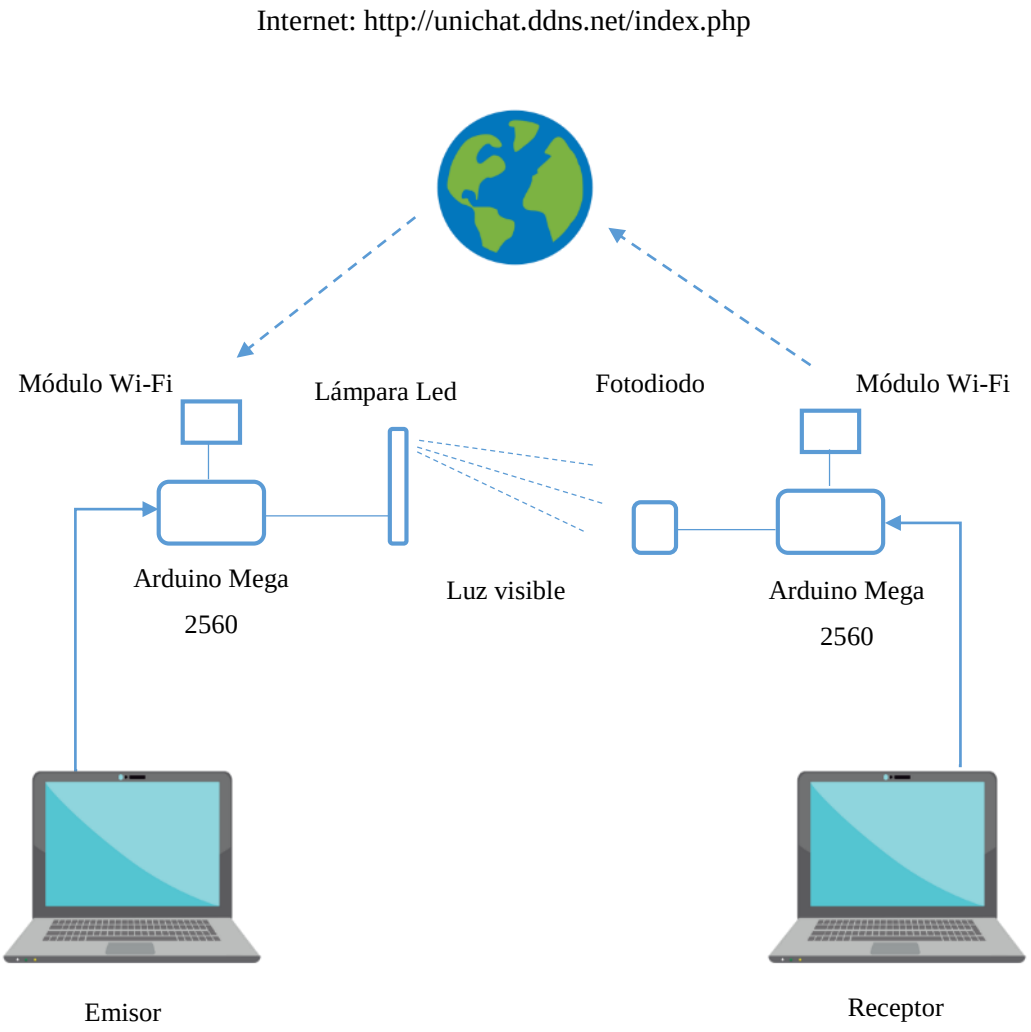


Figura 26. Diagrama de proceso Li-Fi.
Elaborado por: Stalin Mina, Renatto Rugel.

4.1.1 Escenario del Sistema final Li-Fi

Escenario Real Li-Fi

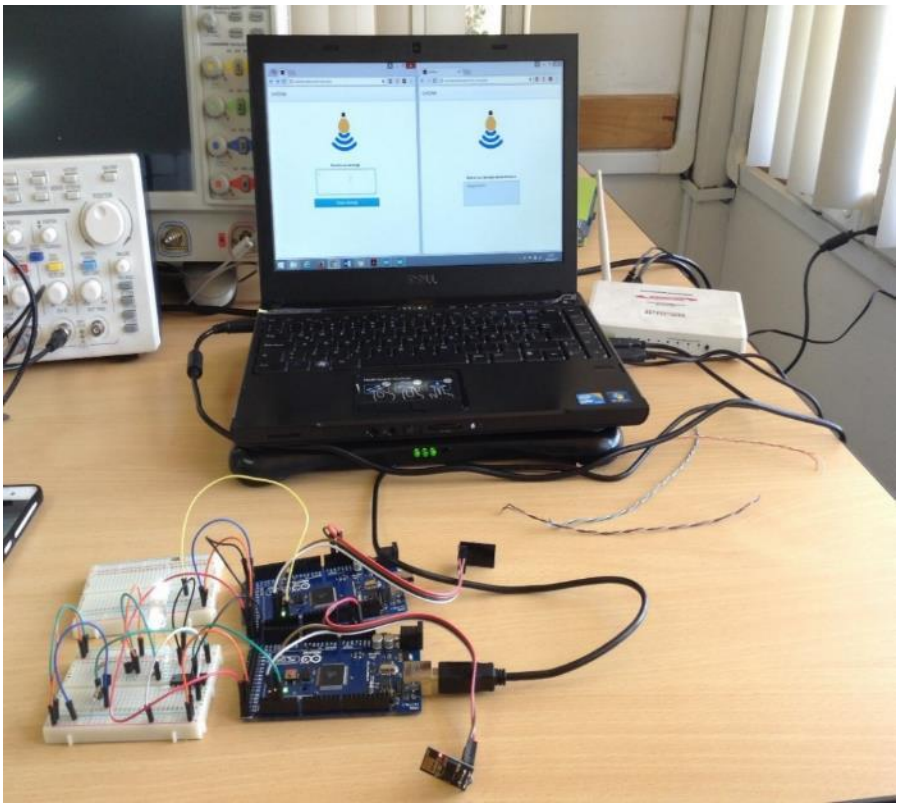


Figura 27. Escenario Real Li-Fi
Elaborado por: Stalin Mina, Renatto Rugel.

En la figura 27, se puede observar el escenario Li-Fi real con sus respectivos componentes tanto en el emisor como en el receptor. El Arduino que corresponde al Receptor está conectada a la Computadora para que alimente de corriente con una fuente de 5v. El mismo que comparte corriente al Arduino del Emisor.

Se tiene un Router que compartirá red Wi-Fi a los módulos ESP8266 y permitirá la salida al internet y la conexión a las páginas Web.

4.1.2 Diseño del Emisor

En la figura 28 se puede observar con mayor claridad los elementos de hardware utilizados en el Emisor.

Diseño del Emisor

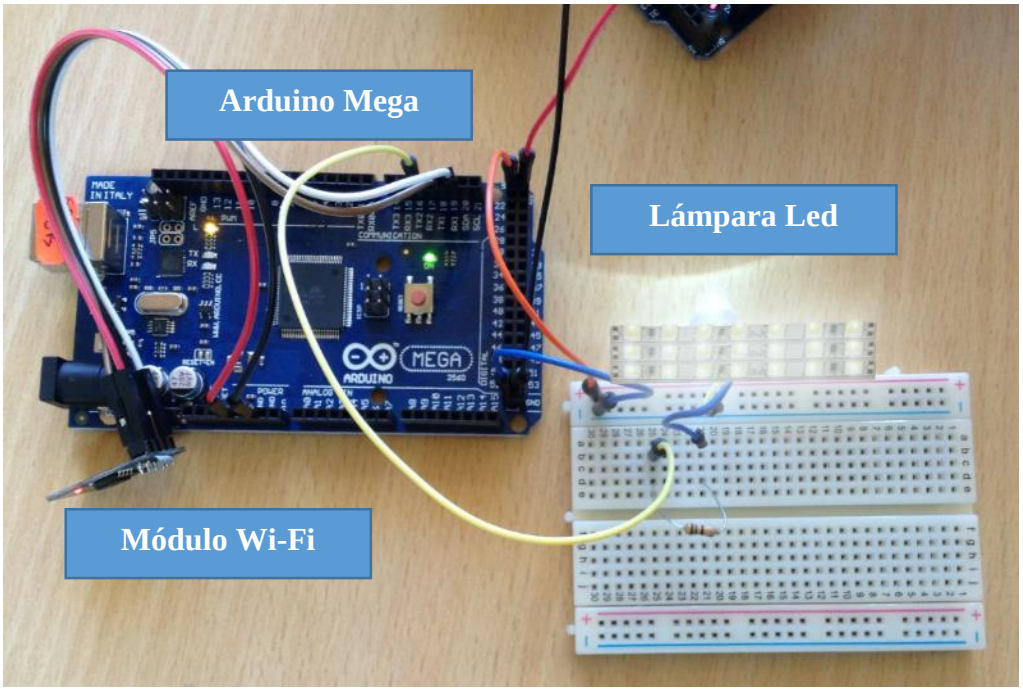


Figura 28. Diseño del Emisor
Elaborado por: Stalin Mina, Renatto Rugel.

4.1.2.1 Detalle de Elementos en el Emisor

- Arduino Mega 2560, cargado el sketch con el código de programación para la codificación de los datos.
- Módulo Wi-Fi ESP8266, conectado al Internet.
- Lámpara led

4.1.3 Diseño del Receptor

En la figura 29 se puede observar con mayor claridad los elementos de hardware utilizados en el Receptor.

Diseño del Receptor

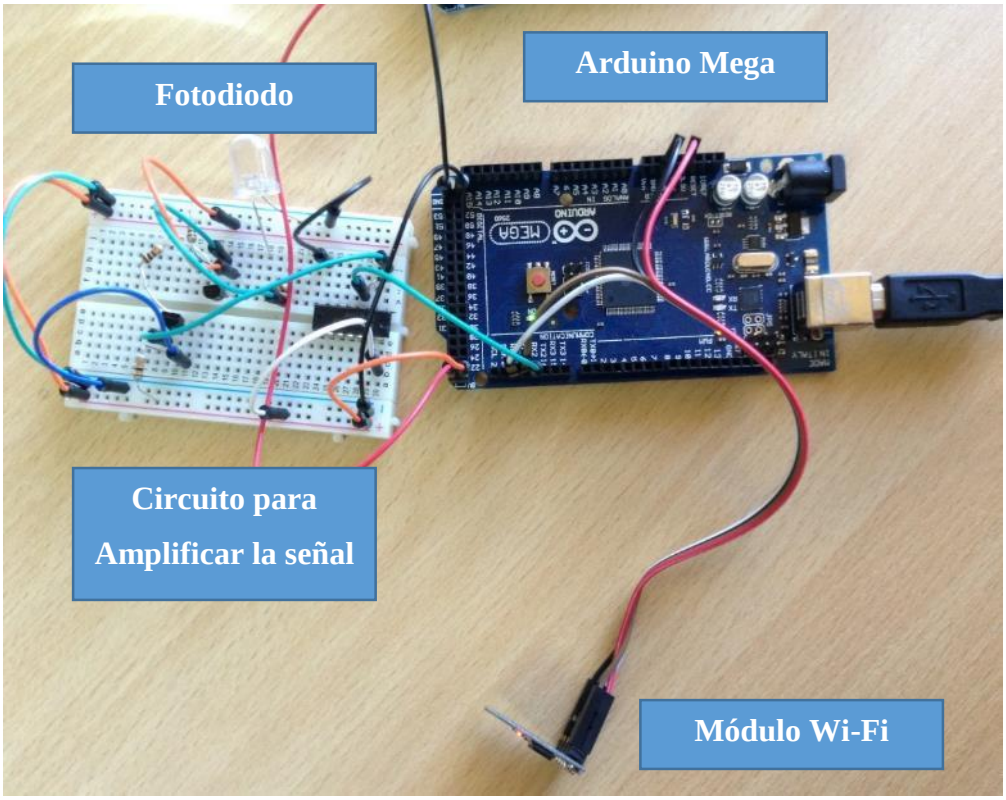


Figura 29. Diseño del Receptor
Elaborado por: Stalin Mina, Renatto Rugel.

4.1.3.1 Detalle de Elementos en el Receptor

- Arduino Mega 2560, cargado el sketch con el código de programación para la decodificación de los datos.
- Módulo Wi-Fi ESP8266, conectado al Internet.
- Fotodiodo, encargado de receptar los pulsos de luz emitidas por la lámpara led
- Circuito para Amplificar la señal, encargada de limpiar y amplificar la señal para evitar perdida de datos. Anexo 2

4.2 Diagrama de Flujo del Escenario Li-Fi

En los siguientes diagramas de flujo, se indica de una manera detalla las conexiones del módulo Wi-Fi a la red para permitir él envió de datos desde la página Web, para que pueda ser codificada y decodificada por el Arduino Mega 2560 tanto en el emisor como el receptor respectivamente, teniendo la comunicación mediante la tecnología Li-Fi y se presente los datos en la página Web de recepción.

4.2.1 Diagrama de Flujo del proceso en el Emisor

A continuación se procede a enlistar cada uno de los pasos que sigue el Emisor para la codificación y transmisión de datos junto con su respectivo diagrama de flujo en la figura 30:

- Inicio del sketch para el Emisor, reconocimiento de librerías. Anexo 3
- Configuración de datos de Red (SSID – PASSWORD - IP- Mascara de Red – Gateway). Estos datos son necesarios para la conexión del módulo Wi-Fi al Internet. Anexo 4
- Se inicializa variables y funciones que van a permitir la conexión y la configuración con el servidor Web (unichat.ddns.net) y puerto (80).
- Se ingresa en la página Web (http://unichat.ddns.net/index.php) y en el cuadro de texto se ingresa los datos a ser transmitidos.
- La página web guarda los datos y envía al Arduino que es recogido por la función leer() y se inicializan las variables que permitirán llamar a la función de codificación de Huffman. Anexo 5.
- Los datos son procesados en la función codificación creada en la librería huffman.h. Explicado en el capítulo 5 apartado 5.2. Anexo 6
- El resultado de la codificación son transmitidos por luz visible. Anexo 7

Diagrama de flujo del proceso en el emisor

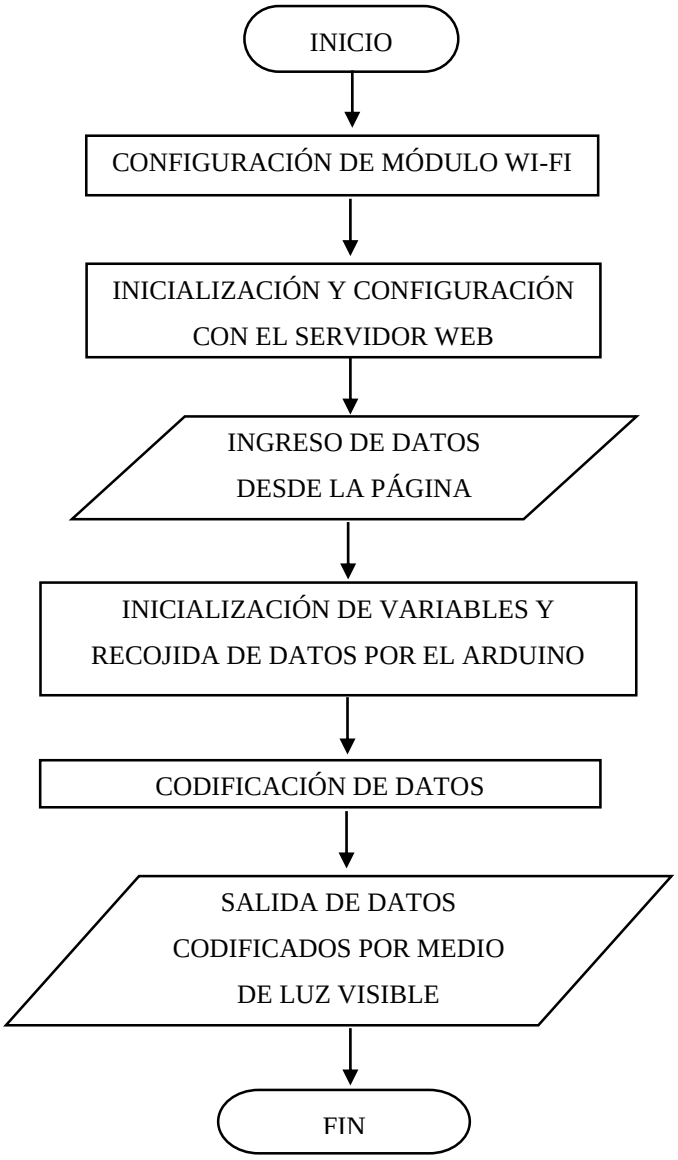


Figura 30. Diagrama de flujo del proceso en el emisor.
Elaborado por: Stalin Mina, Renatto Rugel.

4.2.2 Diagrama de Flujo del proceso en el Receptor

A continuación se procede a enlistar cada uno de los pasos que sigue el Receptor para la respectiva recepción y decodificación de datos junto con su respectivo diagrama de flujo en la figura 31:

- Inicio del sketch para el receptor, reconocimiento de librerías. Anexo 8
- Configuración de datos de Red (SSID – PASSWORD - IP- Mascara de Red – Gateway). Estos datos son necesarios para la conexión del módulo Wi-Fi al Internet. Anexo 9
- Se inicializa variables y funciones que permitirán la conexión y la configuración con el servidor Web (unichat.ddns.net) y puerto (80).
- Los datos codificados, enviados por el transmisor y enviados por luz visible son recogidos por el Arduino en el receptor.
- Se inicializa variables y los datos codificados pasan por la función leer(), que permitirá llamar a la función de decodificación de Huffman y proceder armar los datos correctos para poder ser presentados. Explicado en el capítulo 5 apartado 5.2 Anexo 10
- Los datos decodificados son enviados a la página Web (http://unichat.ddns.net/envio_msn.php) y se podrá ver los datos en un cuadro de texto.

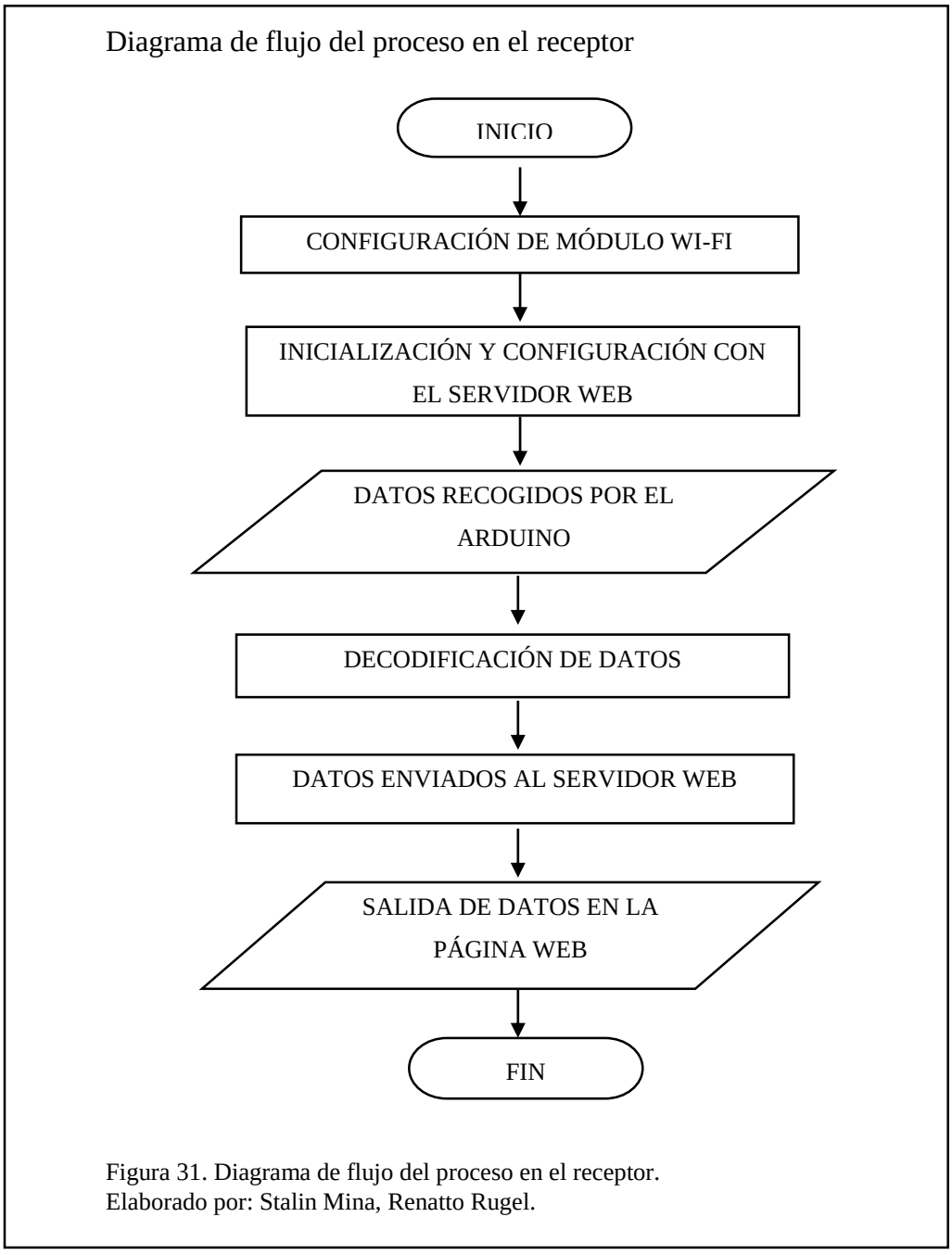


Figura 31. Diagrama de flujo del proceso en el receptor.
Elaborado por: Stalin Mina, Renatto Rugel.

4.3 Desarrollo del código de Programación basado en el Algoritmo de Huffman

Se construyó la programación basándose en programas de Huffman existentes en la web de varios autores y pseudocódigos que se encontró en documentos de Internet (GRC Grupo de Redes de Computadores, 2014; Benitez, 2014 y de Antón, 2015). Dicha

información fue adaptada a la necesidad de cumplir los requerimientos para poder complicar en el IDE de Arduino y basada en el algoritmo visto en el capítulo 3.

4.3.1 Funciones para Codificación y Decodificación

4.3.1.1 Función para la creación de nuevos nodos

Esta función recibe como parámetros un valor entero que representa la frecuencia del símbolo, un valor de tipo carácter que representa el símbolo propiamente dicho, y dos estructuras a y b de tipo nodo que representan ambos lados del nodo, todo esto permitirá la creación de una nueva estructura de este tipo, para cada uno de los símbolos contenidos en la fuente a codificar dentro de la primera parte del bloque de la sentencia if, a medida que se detecta la existencia de una frecuencia dentro del vector frec como se muestra en la figura 32..

Asignación de valores hacia el nuevo nodo n

```
if (frec){
        n -> caracter = caracter;
        n -> frec = frec;
}
else {
        n -> nodo_izq = a, n -> nodo_der = b;
        n -> frec = a->frec + b -> frec;
}
```

Figura 32. Extracto del código de programación para asignación de valores en un nuevo nodo
Elaborado por: Stalin Mina, Renatto Rugel

En el segundo bloque, la sentencia else, permite la construcción de un nuevo nodo, combinación de los dos nodos cuya frecuencia es la más baja y la asignación de la frecuencia total, suma de las frecuencias de los nodos a y b, para dicho nodo.

Una vez realizado cualquiera de los dos procesos se procede a devolver este nuevo nodo n a la función insertar_nodo_cola para su inserción dentro de la cola que contiene la totalidad de los nodos.

4.3.1.2 Función para la extracción de nodos y creación de subárboles

Para la construcción de los subárboles es necesario retirar los dos nodos cuyas frecuencias sean las más bajas, para ello inicialmente se procede a retirar el primer nodo dentro de la cola, después se lleva a cabo un proceso de reordenamiento con los nodos restantes, una vez realizado este proceso, se devuelve este primer nodo, el proceso se repite una segunda vez para retirar el siguiente nodo que formará parte de un subárbol como se muestra en la figura 33.

Extracción de nodos y reordenamiento parcial de los restantes

```
int i, l;
nodo n = cola_prin[i = 1];

while ((l = i * 2) < fin_cola)
{
        if ( (l + 1 < fin_cola) && (cola_prin[l + 1] -> frec < cola_prin[l] -> frec) )
        {
                l++;
        }
        cola_prin[i] = cola_prin[l], i = l;
}
cola_prin[i] = cola_prin[fin_cola];
```

Figura 33. Extracto del código de programación para la extracción de nodos y su reordenamiento parcial.
Elaborado por: Stalin Mina, Renatto Rugel

4.3.1.3 Función para creación del Árbol de Huffman

Para la construcción del árbol de Huffman se inicia con la definición de una variable de tipo entero, un vector de enteros con espacio total de 127 elementos, inicializado con valores en 0.

Después de esto se procede a leer cada uno de los caracteres o símbolos, y a almacenar sus respectivas frecuencias de ocurrencia; una vez realizado este proceso se procede a crear un nodo para cada uno de los caracteres, el cual contiene su frecuencia de ocurrencia, el carácter al que corresponde, y su nodo izquierdo o derecho, inicialmente ninguno cuenta con estos nodos mencionados como se muestra en la siguiente figura.

Creación e inserción de nodos para la construcción del Árbol de Huffman

```
for (i = 0; i < 128; i++)
{
        if (frec[i])
        {
                insertar_nodo_cola (nuevo_nodo (frec[i], i, 0, 0));
        }
}
while (fin_cola > 2)
{
        insertar_nodo_cola (nuevo_nodo (0, 0, quitar_nodo_cola(),
        quitar_nodo_cola()));
}
```

Figura 34. Creación e inserción de nodos y construccion del Arbol de Huffman
Elaborado por: Stalin Mina, Renatto Rugel

Habiendo terminado la creación de los nodos se procede a la creación de los subárboles, proceso que se repite y se controla mediante el bucle while, hasta que solo quede uno, para esto se hace uso de las funciones quitar_nodo_cola, nuevo_nodo e insertar_nodo_cola en ese mismo orden.

4.3.1.4 Función para la decodificación del código Huffman generado

Para el proceso de decodificación se procede a leer cada uno de los caracteres codificados contenidos dentro del apuntador de caracteres s, dependiendo de cada uno de los bits, 0 o 1, se dirigirá hacia el nodo de la izquierda o de la derecha, hasta llegar a un carácter como se muestra en la figura 35, y presentarlo dependiendo de su posición dentro de la tabla ASCII.

Parte del proceso de decodificación

```
if (*s++ == '0')
{
        n = n -> nodo_izq;
}
else n = n -> nodo_der;

if (n -> caracter)
{
        putchar (n->caracter);
        n = t;
}
```

Figura 35. Proceso de decodificación del código Huffman generado.
Elaborado por: Stalin Mina, Renatto Rugel

4.3.1.5 Función para la inserción de nodos dentro de la cola

Esta función realiza la inserción de todos los nodos que se generan por cada símbolo contenido en la fuente a codificar; terminado el proceso de inserción como se muestra en la figura 36, seguido de esto se procede a ordenar cada nodo dependiendo de su frecuencia de ocurrencia de menor a mayor, esto permitirá la formación de subárboles con aquellos nodos cuyas frecuencias son las más bajas.

Parte del proceso de inserción y ordenamiento de nodos dentro de la cola

```
if (cola_prin[j] -> frec <= n -> frec)
        break;
cola_prin[i] = cola_prin[j], i = j;
cola_prin[i] = n;
```

Figura 36. Extracto del código de programación para la inserción y ordenamiento de nodos.
Elaborado por: Stalin Mina, Renatto Rugel

4.3.1.6 Función para la inserción de nodo en un Arreglo tipo Cola

Mediante esta función se van guardando los nodos según su frecuencia de menor a mayor. Se selecciona la frecuencia del nodo, para luego ser sometida a una sentencia while para guardar su nueva posición en el arreglo cola_prin[] como se muestra en la figura 37.

Función para la inserción de nodo en un Arreglo tipo Cola

```
int i = fin_cola++;
while ((j = i ++)) {
        if (cola_prin[j]->frec <= n->frec)
        cola_prin[i] = cola_prin[j], i = j;
}
cola_prin[i] = n;
```

Figura 37. Código de programación para la inserción de un nodo.
Elaborado por: Stalin Mina, Renatto Rugel

4.3.1.7 Función para Crear Código

Para finalizar, se obtiene cada uno de los códigos Huffman mediante la función crear_codigo, la cual se encarga de colocar los 0's y 1's a cada una de las ramas del árbol. 'O' en el lado izquierdo y '1' en el lado derecho como se indica en la figura 38.

Función para Crear Código

```
If (n->caracter) {
        s[len] = 0;
        strcpy(out, s);
        codigo[n->caracter] = out;
        out += len + 1;
        return;
}
s[len] = '0'; crear_codigo(n->nodo_izq, s, len + 1);
s[len] = '1'; crear_codigo(n->nodo_der, s, len + 1);
```

Figura 38. Función para Crear Código
Elaborado por: Stalin Mina, Renatto Rugel

4.3.1.8 Función para Codificar

Procesa los datos ingresados y el código creado por la función crear_codigo, primero copia el código en un puntero out y luego guarda la longitud del código. Para poder ser presentados en pantalla. Ver figura 39.

Función para Codificar

```
while (*s) {

        strcpy(out, codigo[*s]);
        out += strlen(codigo [*s++] );
}
```

Figura 39. Función para Codificar
Elaborado por: Stalin Mina, Renatto Rugel

Capítulo 5

Pruebas y Resultados

En este capítulo se realizara varias pruebas para verificar la calidad de funcionamiento del sistema Li-Fi. Se someterá a pruebas en varias velocidades y distancias, para analizar ancho de banda, retardo y BER. Estas pruebas serán realizadas en el sistema con y sin el algoritmo de Huffman.

5.1 Pruebas de Transmisión de Datos sin Algoritmo de Huffman

Se someterá a pruebas el sistema Li-Fi, sin implementar la programación de Huffman en los programas sketch de Arduino, tanto en el emisor como en el receptor.

Verificación de Señal en el Emisor y Receptor

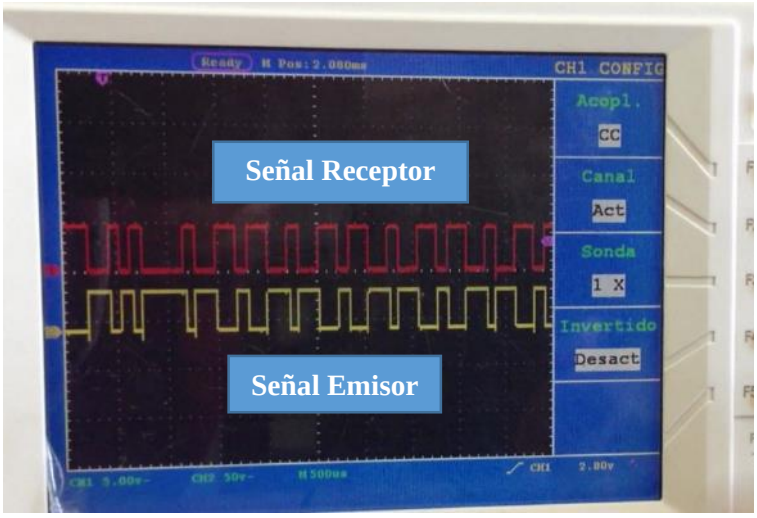


Figura 40. Verificación de Señal en el Emisor y Receptor
Elaborado por: Stalin Mina, Renatto Rugel

En la figura 40, se puede observar que en el emisor y en el receptor la señal es nítida al ser sometida a una velocidad de 9600bps a una distancia de 0,05m, para esto se verifica y

se determina que los datos enviados, llegaron completos desde el envío de la Pagina web hasta su llegada a la página de recepción.

5.1.1 Cálculo de la Velocidad de Transmisión

En el escenario Li-Fi sin la implementación del Algoritmo de codificación Huffman, se realizaron pruebas utilizando 4 velocidades distintas de modulación.

Basándose en la siguiente formula:

$$V_{tx} = V_m * \log_2 N$$

Vm= Velocidad de modulación (baudios)

N= Numero de niveles, quiere decir que 2 bits por cada señal emitida (Haro Soledispa & Buenaño Medina, 2016 y Arduino, 2017)

Tabla 10. Velocidad de Transmisión

Velocidad de Modulación (Baudios)	Velocidad de Transmisión (Kbps)
4800	4,8
9600	9,6
19200	19,2
38400	38,4

Nota: Cálculo de la velocidad de transmisión
Elaborado por: Stalin Mina, Renatto Rugel.

5.1.2 Análisis con Velocidad de Transmisión 4800

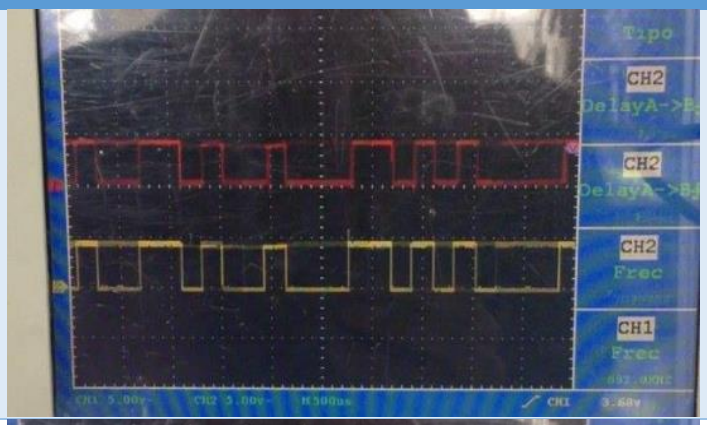
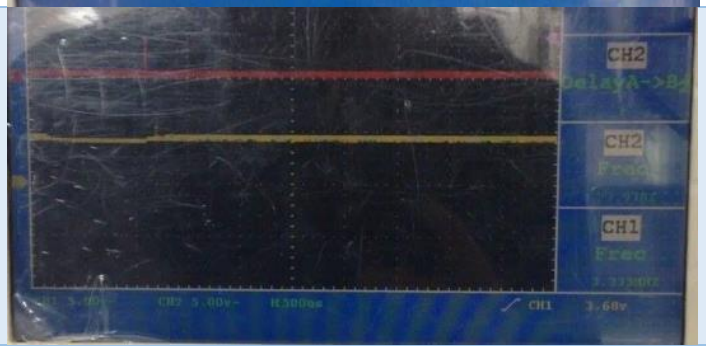
5.1.2.1 Análisis de Velocidad de Transmisión vs Distancia

Se realizó la transmisión de información a una velocidad de 4800bps a 3 distancias para determinar la frecuencia y el retardo. Se utilizó el Osciloscopio PeakTech Storage, el cual permite visualizar los datos antes mencionados.

Señal Amarilla = Emisor

Señal Roja = Receptor

Tabla 11. Gráfica Distancia vs Velocidad de Transmisión 4800bps

Distancia [m]	Velocidad de Transmisión [bps]	
0,05	4800	
0,06	4800	
0,07	4800	

Nota: Gráfica s obtenidas a diferentes distancias a las que se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

Como se puede observar en la tabla No 11, la señal se degrada al aumentar la distancia entre el emisor y el receptor. A una distancia de 0,06m, se realiza la transmisión pero se pierde gran parte de la integridad de los datos. A una distancia de 0,07 el fotodiodo no puede recibir los pulsos de luz, debido a esto no se evidencia señal y no llegan datos legibles al receptor.

Tabla 12. Análisis de datos Distancia vs Velocidad de Transmisión 4800 bps

Distancia (m)	Frecuencia kHz	Retardo (ms)	Observación
0,05	956,80	0,61732	Transmisión Correcta (100%)
0,06	561,8	0,62098	Perdida mínima de datos (10%)
0,07	2270	--	Pérdida total de datos (0%)

Nota: Observaciones realizadas para cada una de las distancias, a las cuales se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

5.1.2.2 Análisis de Distancia vs Retardo

Gráfica Análisis Retardo

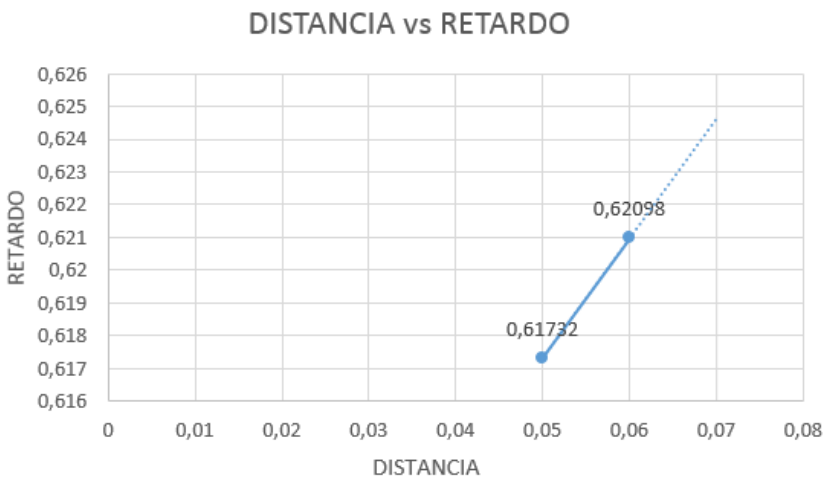


Figura 41. Análisis Retardo 4800 bps
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 41, el retardo es proporcional a la distancia, aunque sufre un leve retraso de 0.003 ms, entre las dos distancias.

5.1.3 Análisis de Velocidad de Transmisión 9600

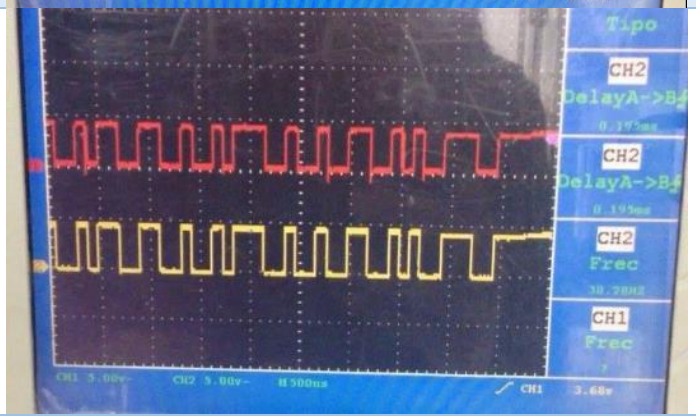
5.1.3.1 Análisis de Velocidad de Transmisión VS Distancia

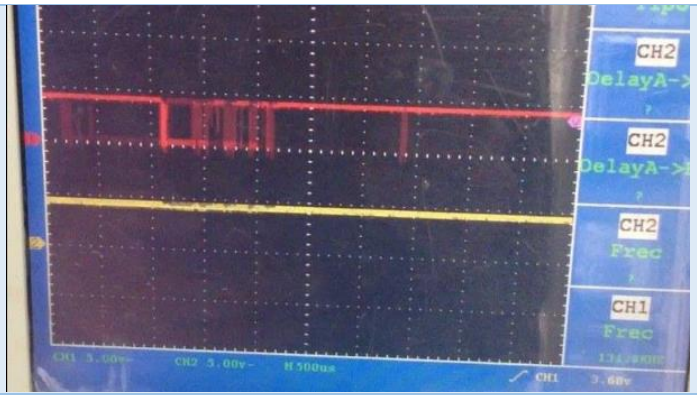
Se realizó la transmisión de información a una velocidad de 9600bps a 3 distancias para determinar la frecuencia y el retardo. Se utilizó el Osciloscopio PeakTech Storage, el cual permite visualizar los datos antes mencionados.

Señal Amarilla = Emisor

Señal Roja = Receptor

Tabla 13. Gráfica Distancia vs Velocidad de Transmisión 9600bps

Distancia [m]	Velocidad de Transmisión [bps]	
0,05	9600	
0,06	9600	

0,07	9600	

Nota: Gráfica s obtenidas a diferentes distancias a las que se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

Como se puede observar en la tabla No 13, la señal se degrada al aumentar la distancia entre el emisor y el receptor. A la distancia de 0,05m se tiene una señal nítida y una transmisión optima de datos. A una distancia de 0,06m, los datos se transmiten, sin embargo se tiene una pérdida en la integridad de los datos. A la distancia de 0,07m el emisor trata de trasmitir los datos con errores y el fotodiodo no puede recibir los pulsos de luz, no evidenciándose señal y los datos que se receptan son incoherentes.

Tabla 14. Análisis de datos Distancia vs Velocidad de Transmisión 9600 bps

Distancia (m)	Frecuencia kHz	Retardo (ms)	Observación
0,05	1,35	0,198	Transmisión Correcta (100%)
0,06	1, 924	0,406	Perdida mínima de datos (10%)
0,07	134,28	--	Pérdida total de datos y distorsión de la misma (0%)

Nota: Observaciones realizadas para cada una de las distancias, a las cuales se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

5.1.3.2 Análisis de Distancia vs Retardo

Gráfica Análisis Retardo

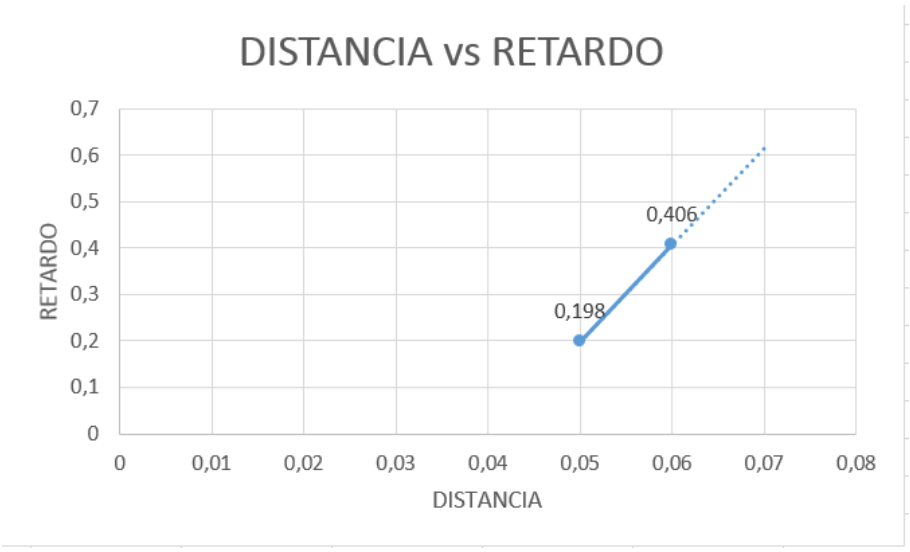


Figura 42. Análisis Retardo 9600bps
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 42, el retardo es proporcional a la distancia y sufre un retraso de 0.208 ms, entre las dos velocidades.

5.1.4 Análisis de Velocidad de Transmisión 19200

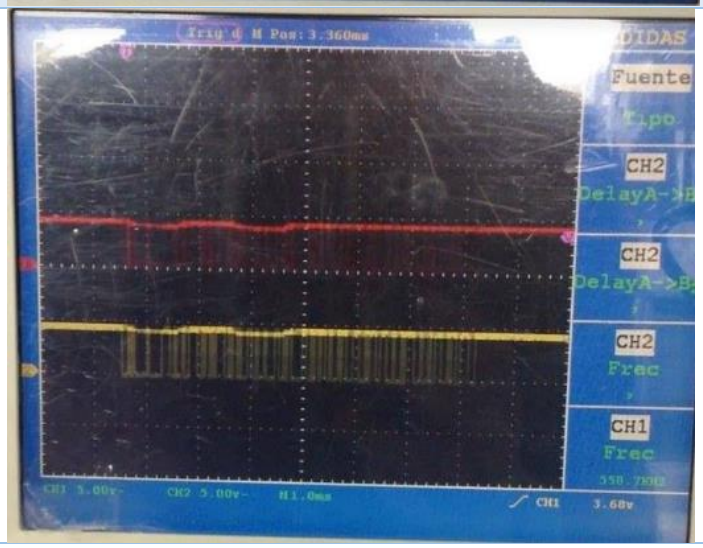
5.1.4.1 Análisis de Velocidad de Transmisión vs Distancia

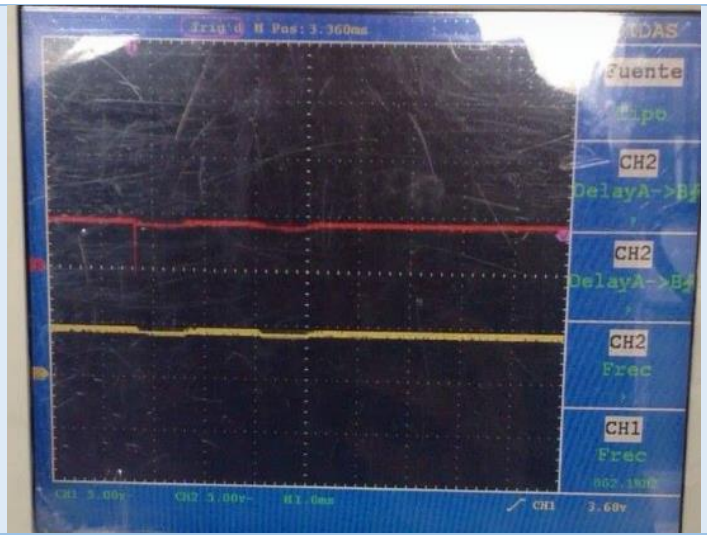
Se realizó la transmisión de información a una velocidad de 19200bps a 3 distancias para determinar la frecuencia y el retardo. Se utilizó el Osciloscopio Peaktech Storage,el cual permite visualizar los datos antes mencionados.

Señal Amarilla = Emisor

Señal Roja = Receptor

Tabla 15. Gráfica Distancia vs Velocidad de Transmisión 19200bps

Distancia [m]	Velocidad de Transmisión [bps]	
0,05	19200	
0,06	19200	

0,07	19200	

Nota: Gráfica s obtenidas a diferentes distancias a las que se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

Como se puede observar en la tabla No 13, la señal se degrada al aumentar la distancia entre el emisor y el receptor. A una distancia de 0,06m, los datos si se transmiten perdiendo la integridad de los datos. A la distancia de 0,07 el emisor trata de trasmitir los datos con errores y el fotodiodo no puede recibir los pulsos de luz, no se evidencia señal y los datos que se receptan son incoherentes.

Tabla 16. Análisis de datos Distancia vs Velocidad de Transmisión 19200 bps

Distancia (m)	Frecuencia kHz	Retardo (ms)	Observación
0,05	1,35	0,198	Transmisión Correcta (100%)
0,06	1, 924	0,406	Perdida mínima de datos (10%)
0,07	134,28	--	Pérdida total de datos (0%)

Nota: Observaciones realizadas para cada una de las distancias, a las cuales se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

5.1.4.2 Análisis de Distancia vs Retardo

Gráfica Análisis Retardo

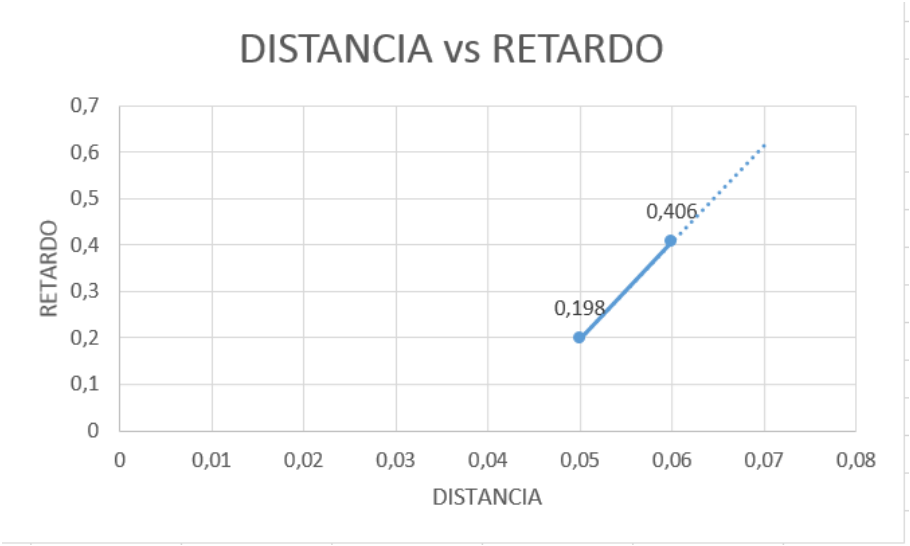


Figura 43. Análisis Retardo 19200bps
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 43, el retardo es proporcional a la distancia y sufre un retraso de 0.208 ms, entre las dos velocidades.

5.1.5 Análisis de Velocidad de Transmisión 38400

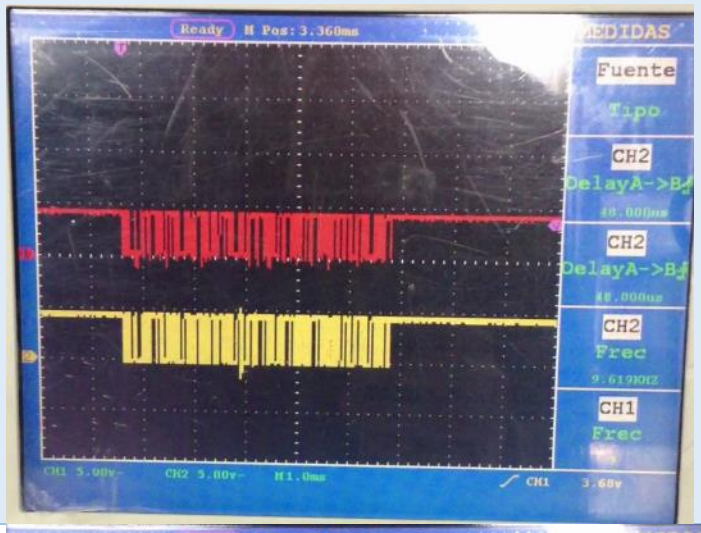
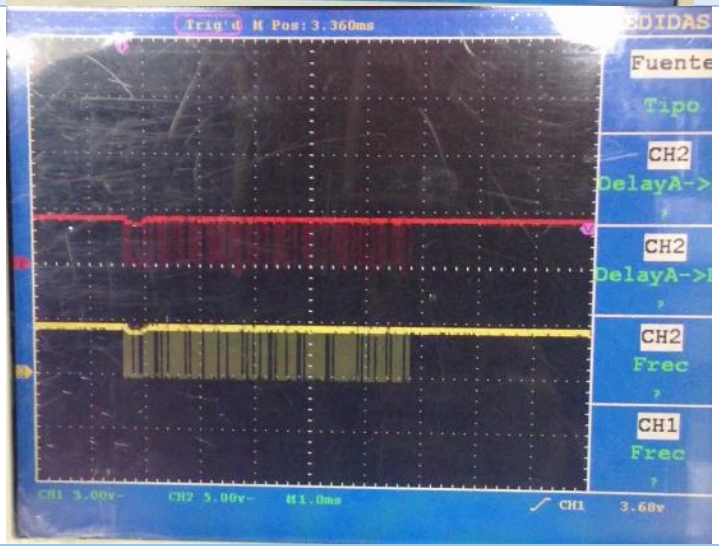
5.1.5.1 Análisis de Velocidad de Transmisión vs Distancia

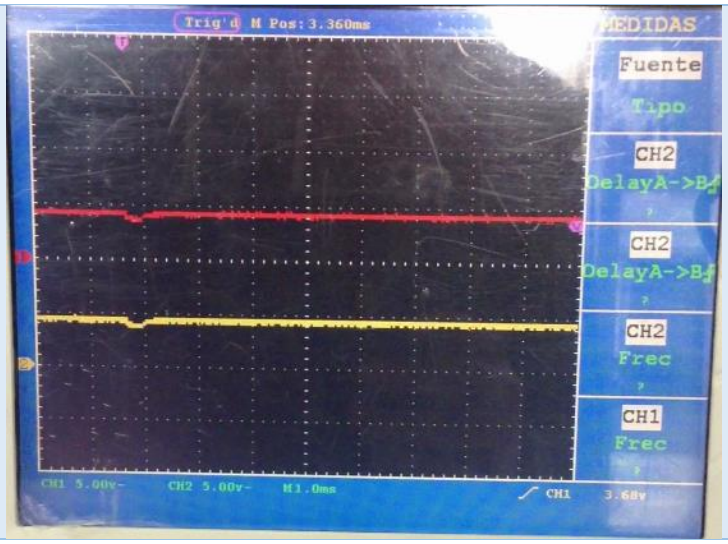
También se efectuaron pruebas de transmisión de información a una velocidad de 38400bps a 3 distancias para determinar la frecuencia y el retardo generado, para ello nuevamente se utilizó el Osciloscopio PeakTech Storage.

Señal Amarilla = Emisor

Señal Roja = Receptor

Tabla 17. Gráfica Distancia vs Velocidad de Transmisión 38400bps

Distancia [m]	Velocidad de Transmisión [bps]	
0,05	38400	
0,06	38400	

0,07	38400	

Nota: Gráfica s obtenidas a diferentes distancias a las que se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

Al observar en la tabla No 17, es posible resaltar que la señal entre emisor y receptor, se ha degradado de forma substancial conforme se ha aumentado aún más la distancia y su velocidad de transmisión. Así pues, a una distancia de 0,06m, los datos son transmitidos pero de una forma rápida, pero poco eficaz para que el receptor sea capaz de detectar completamente los datos que se han enviado; ya a una distancia de 0,07 y manteniendo la velocidad anteriormente mencionada, el receptor es incapaz de detectar los pulsos de luz del emisor, razón por la cual no se presentan cambios en el osciloscopio.

Tabla 18. Análisis de datos Distancia vs Velocidad de Transmisión 38400 bps

Distancia (m)	Frecuencia kHz	Retardo (us)	Observación
0,05	9,619	48,00	Transmisión con pérdida mínima de datos (10%)
0,06	14,751	56,7	Pérdida casi total de datos (50%)
0,07	----	----	Pérdida total de datos (0%)

Nota: Observaciones realizadas para cada una de las distancias, a las cuales se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

5.1.5.2 Análisis de Distancia vs Retardo

Gráfica Análisis Retardo

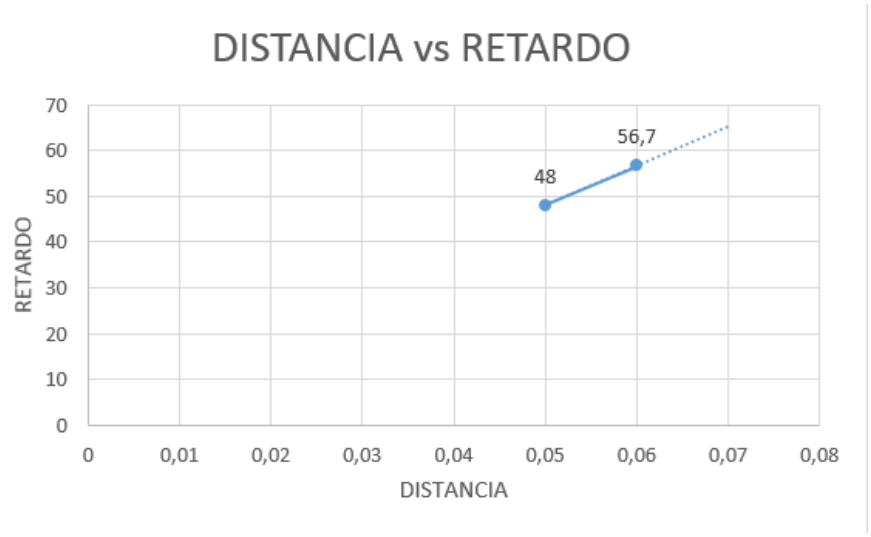


Figura 44. Análisis Retardo 38400bps
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 44, el retardo existente es mucho mayor en comparación con anteriores distancias y velocidades, esto debido a que la señal dentro del sistema actual, no es capaz de viajar de forma íntegra tales distancias, aun cuando la velocidad se ha sido aumentada a tan altos valores.

5.1.6 Análisis Distancia vs BER

Una vez realizado el análisis de transmisión de datos a diferentes velocidades y distancias se determina que la velocidad adecuada para este sistema Li-Fi es a 9600bps.

Por medio del código creado para la transmisión, se sacó los bits del mensaje transmitido:

“El enojo, el orgullo y la competencia son nuestros verdaderos enemigos” (Lama, 1935)

Bits: 01000101 01101100 00100000 01100101 01101110 01101111 01101010 01101111 00101100 00100000 01100101 01101100 00100000 01101111 01110010 01100111 01110101 01101100 01101100 01101111 00100000 01111001 00100000 01101100 01100001 00100000 01100011 01101111 01101101 01110000 01100101 01110100 01100101 01101110 01100011 01101001 01100001 00100000 01110011 01101111

01101110 00100000 01101110 01110101 01100101 01110011 01110100 01110010 01101111 01110011 00100000 01110110 01100101 01110010 01100100 01100001 01100100 01100101 01110010 01101111 01110011 00100000 01100101 01101110 01100101 01101101 01101001 01100111 01101111 01110011

Tabla 19. Análisis de la Distancia vs BER a una Velocidad de Transmisión de 9600bps

Velocidad de Transmisión (bps)	Distancia (m)	Bits Errados	Cálculo del BER	%
9600	0,05	0	0,0	0
9600	0,06	17	0,03035714	3
9600	0,07	483	0,86250000	86

Nota: Valores obtenidos durante el cálculo del BER.
Elaborado por: Stalin Mina, Renatto Rugel.

Gráfica Análisis Distancia vs BER

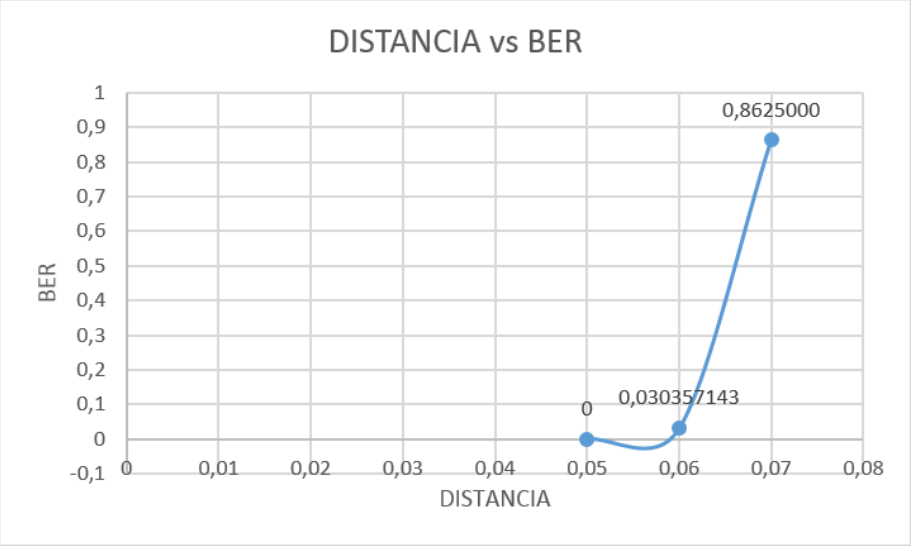


Figura 45. Análisis del cálculo del BER
Elaborado por: Stalin Mina, Renatto Rugel

En la figura 45 se observa que al realizar la transmisión de datos (560 bits); con una distancia inicial de 0,05 m, la tasa de error es completamente nula, sin embargo se puede observar que al aumentar la distancia a 0,06 - 0,07, esta nulidad se pierde y empiezan a existir pérdidas, principalmente debido a que el receptor no es capaz de recibir todos los pulsos de luz que se transmiten a estas últimas distancias.

A la distancia de 0,06 se observa 17 bits errados, que representa el 3% de perdida y a una distancia de 0,07 se tiene una pérdida de 483 bits, generándose así una pérdida de 15%.

5.1.7 Análisis de Ancho de Banda

Una vez realizado el análisis de transmisión de datos a diferentes velocidades y distancias se determina que la velocidad adecuada para este sistema Li-Fi es a 9600bps.

Por medio del codigo creado para la transmisión, se sacó los bits del mensaje transmitido: "El enojo, el orgullo y la competencia son nuestros verdaderos enemigos"

Se realiza el análisis gracias al software Matlab, utilizando la transformada de Fourier que permite analizar las frecuencias de una señal temporal, en la cual se toma los datos de un txt que se tiene guardado los bits del mensaje transmitido.

Basándose en la siguiente formula:

$$AB = f_2 - f_1$$

AB= f2 - f1

AB=1.339 - 1,32

AB=1,33768 kHz

Gráfica Análisis Ancho de Banda

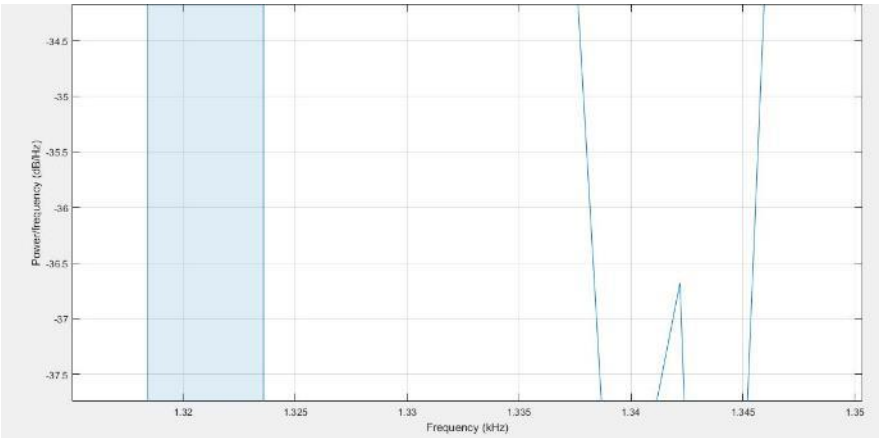


Figura 46. Análisis Ancho de Banda
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 46, se obtienen ondas senoidales para la cual se tiene la onda que se encuentre en la Frecuencia 1,35kHz, dato que se obtuvo del osciloscopio y se procede a seleccionar la onda y determinar su frecuencia mayor y menor.

5.2 Pruebas de Transmisión de Datos con Algoritmo de Huffman

Se someterá a pruebas el sistema Li-Fi, implementando la programación de Huffman en los programas sketch del Arduino, tanto en el emisor como en el receptor.

Verificación de Señal en el Emisor y Receptor

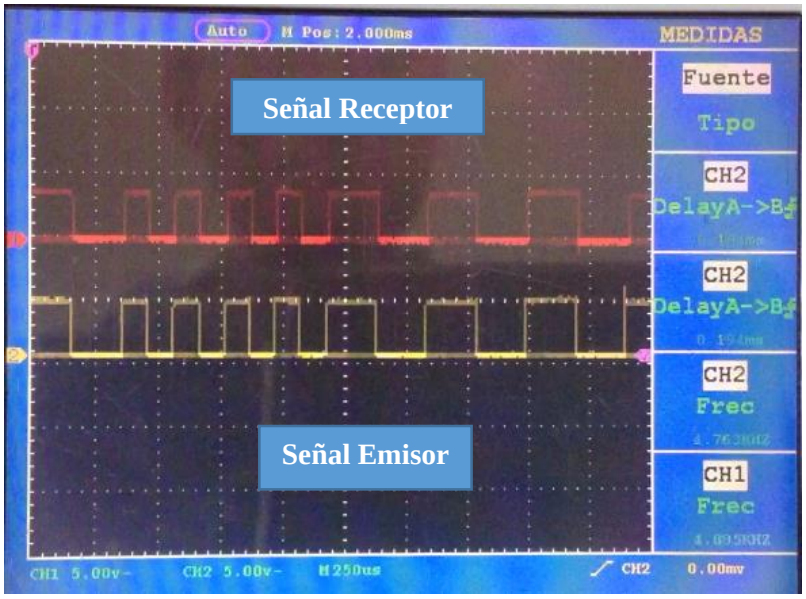


Figura 47. Verificación de Señal en el Emisor y Receptor con Codificación Huffman
Elaborado por: Stalin Mina, Renatto Rugel

En la figura 47, se puede observar que en el emisor y en el receptor la señal es nítida sometida a una velocidad de 9600bps a una distancia de 0,0532m, se verifica y se determina que los datos enviados, llegaron completos a un tiempo de 0,194s; desde el envío de la Pagina web hasta su llegada a la página de recepción.

5.2.1 Cálculo de la Velocidad de Transmisión

En el escenario Li-Fi con la implementación del Algoritmo de codificación Huffman, se realizaron pruebas utilizando 3 velocidades distintas de modulación ya que como se comprobó en el apartado 6.1.5.1 del capítulo 6, a una velocidad de 38400 se tiene perdida de datos en todas las distancias.

Basándose en la siguiente formula:

$$V_{tx} = V_m * \log_2 N$$

Vm= Velocidad de modulación (baudios)

N= Numero de niveles, quiere decir que 2 bits por cada señal emitida (Haro Soledispa & Buenaño Medina, 2016 y Arduino, 2017)

Tabla 20. Velocidad de Transmisión

Velocidad de Modulación (Baudios)	Velocidad de Transmisión (Kbps)
4800	4,8
9600	9,6
19200	19,2

Nota: Cálculo de la velocidad de transmisión
Elaborado por: Stalin Mina, Renatto Rugel.

5.2.2 Análisis con Velocidad de Transmisión 4800

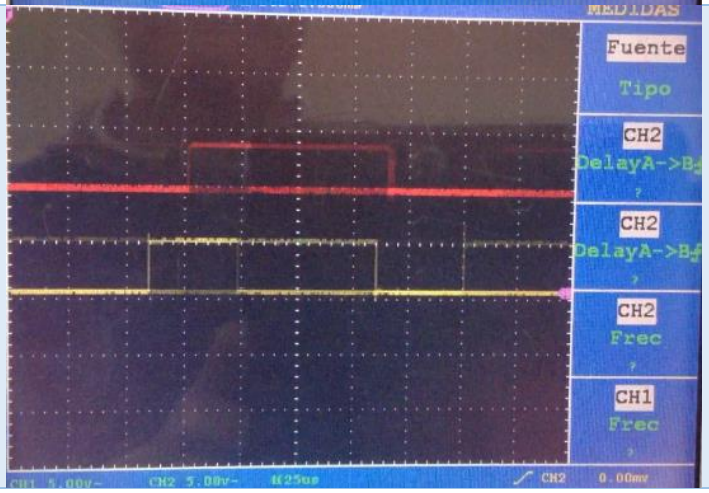
5.2.2.1 Análisis de Velocidad de Transmisión vs Distancia

Se realizó la transmisión de información codificada con Huffman, a una velocidad de 4800bps a 3 distancias para determinar la frecuencia y el retardo. Se utilizó el Osciloscopio PeakTech Storage, el cual permite visualizar los datos antes mencionados.

Señal Amarilla = Emisor

Señal Roja = Receptor

Tabla 21. Gráfica Distancia vs Velocidad de Transmisión 4800bps con Huffman

Distancia [m]	Velocidad de Transmisión [bps]	
0,0532	4800	
0,0674	4800	
0,0772	4800	

Nota: Gráfica s obtenidas a diferentes distancias a las que se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

Como se puede observar en la tabla 21, la señal no tiene mayor degradación al aumentar la distancia entre el emisor y el receptor. A una distancia de 0,0532 y 0,0674 m, se realiza la transmisión con éxito y una señal nítida. A una distancia de 0,077 se puede observar que tiene un retraso considerable en la recepción, ocasionando una pérdida de información.

Tabla 22. Análisis de datos Distancia vs Velocidad de Transmisión 4800 bps con Huffman

Distancia (m)	Frecuencia kHz	Retardo (ms)	Observación
0,0532	4,935	0,195	Transmisión Correcta (100%)
0,0674	9.312	0,294	Transmisión Correcta (100%)
0,0772	3,287	0,392	Perdida media de datos (50%)

Nota: Observaciones realizadas para cada una de las distancias, a las cuales se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

5.2.2.2 Análisis de Distancia vs Retardo

Gráfica Análisis Retardo

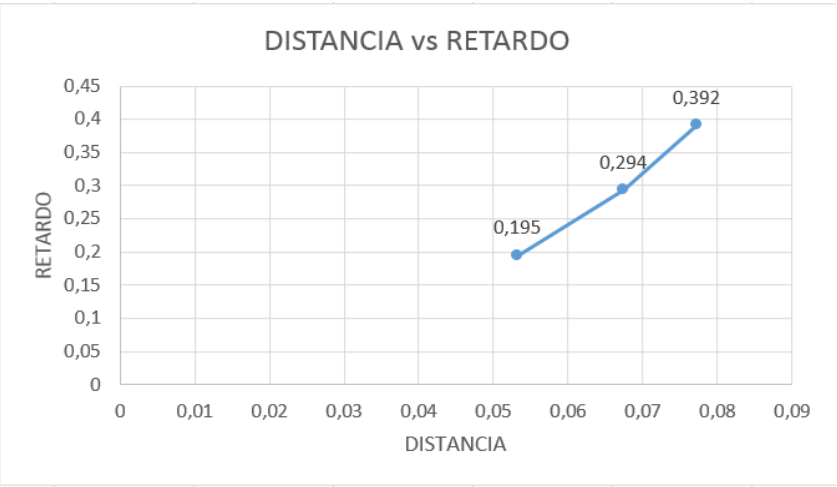


Figura 48. Análisis Retardo 4800 bps con Huffman
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 48, el retardo es proporcional a la distancia, aunque sufre un leve retraso de 0.1 ms, entre las tres distancias.

5.2.3 Análisis de Velocidad de Transmisión 9600

5.2.3.1 Análisis de Velocidad de Transmisión VS Distancia

Se realizó la transmisión de información codificada con Huffman, a una velocidad de 9600bps a 3 distancias para determinar la frecuencia y el retardo. Se utilizó el Osciloscopio PeakTech Storage, el cual permite visualizar los datos antes mencionados.

Señal Amarilla = Emisor

Señal Roja = Receptor

Tabla 23. Gráfica Distancia vs Velocidad de Transmisión 9600bps con Huffman

Distancia [m]	Velocidad de Transmisión [bps]	
0,0532	9600	

0,0674	9600	
0,0772	9600	

Nota: Gráfica s obtenidas a diferentes distancias a las que se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

Como se puede observar en la tabla 23, no existe degradación de la señal al aumentar la distancia entre el emisor y el receptor. A la distancia de 0,0532m y 0,0674m no hay pérdida de datos y la transmisión es exitosa. A la distancia de 0,0772m la señal es nítida, sin embargo hay un pequeño retraso en el receptor y existe una perdida mínima de caracteres al final del texto enviado

Tabla 24. Análisis de datos Distancia vs Velocidad de Transmisión 9600 bps con Huffman

Distancia (m)	Frecuencia kHz	Retardo (ms)	Observación
0,0532	1,613	0,198	Transmisión Correcta (100%)
0,0674	4, 935	0,199	Transmisión Correcta (100%)
0,0772	2,467	0,203	Perdida mínima de datos (5%)

Nota: Observaciones realizadas para cada una de las distancias, a las cuales se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

5.2.3.2 Análisis de Distancia vs Retardo

Gráfica Análisis Retardo

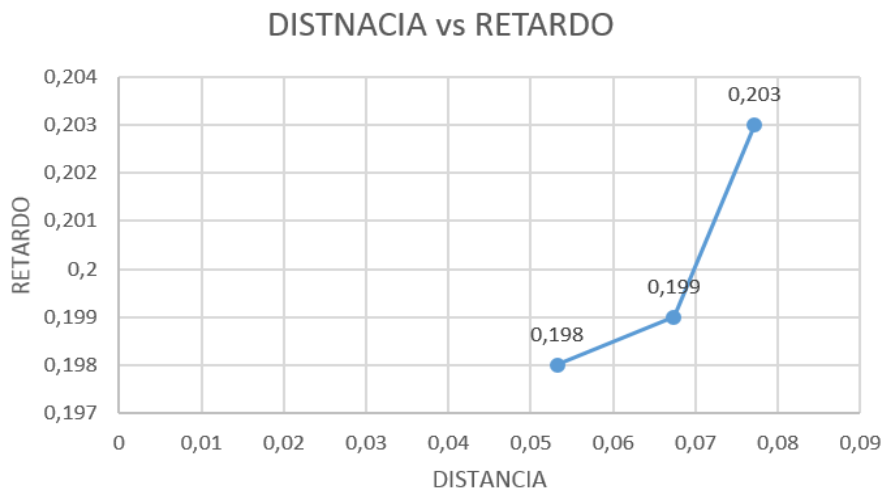


Figura 49. Análisis Retardo 9600bps con Huffman
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 49, el retardo es proporcional a la distancia y sufre un retraso de 0.003 ms, entre las tres velocidades.

5.2.4 Análisis de Velocidad de Transmisión 19200

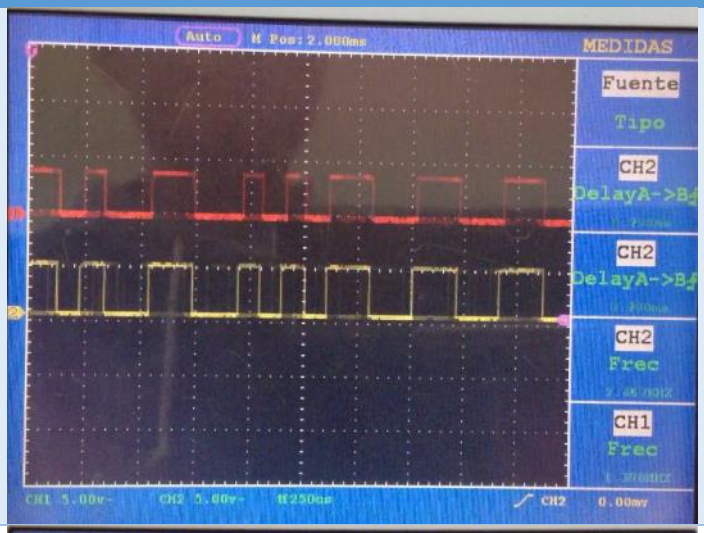
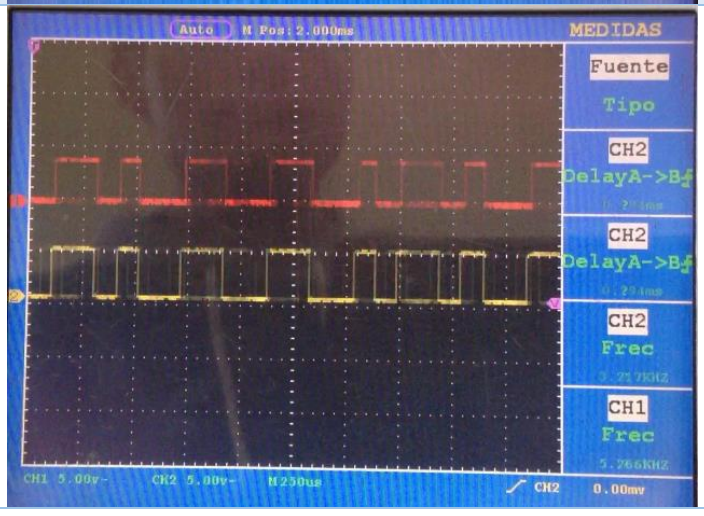
5.2.4.1 Análisis de Velocidad de Transmisión vs Distancia

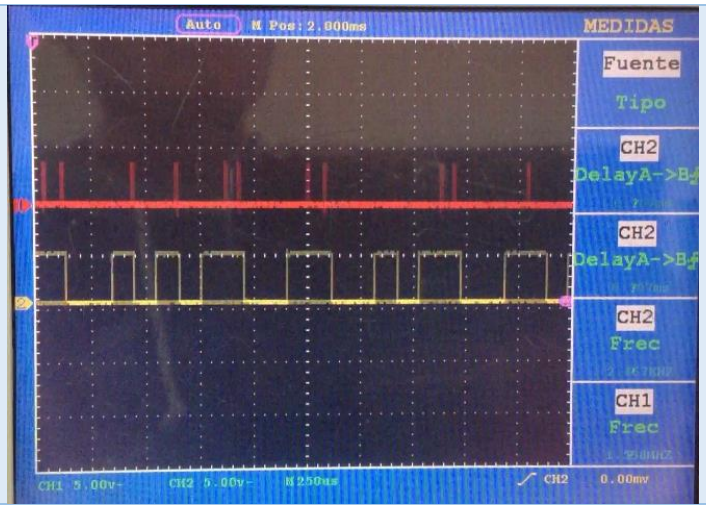
Se realizó la transmisión de información codificada con Huffman, a una velocidad de 19200bps a 3 distancias para determinar la frecuencia y el retardo. Se utilizó el Osciloscopio Peaktech Storage, el cual permite visualizar los datos antes mencionados.

Señal Amarilla = Emisor

Señal Roja = Receptor

Tabla 25. Gráfica Distancia vs Velocidad de Transmisión 19200bps con Huffman

Distancia [m]	Velocidad de Transmisión [bps]	
0,0532	19200	
0,0674	19200	

0,0772	19200	

Nota: Gráfica s obtenidas a diferentes distancias a las que se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

Como se puede observar en la tabla 25, la señal se degrada al aumentar la distancia entre el emisor y el receptor, adicional se puede observar que en las tres distancias existe un ligero retraso en el receptor. A una distancia de 0,0532m los datos se transmiten con éxito, a 0,0674m la señal es casi nítida pero existen perdidas de datos. A la distancia de 0,0772 la señal en el receptor no se puede visualizar y el fotodiodo no puede recibir los pulsos de luz, ocasionando que los datos que se receptan son incoherentes.

Tabla 26. Análisis de datos Distancia vs Velocidad de Transmisión 19200 bps con Huffman

Distancia (m)	Frecuencia kHz	Retardo (ms)	Observación
0,0532	2,567	0,290	Transmisión Correcta (100%)
0,0674	3,217	0,294	Perdida mínima de datos (5%)
0,0772	4,935	0,298	Pérdida total de datos (0%)

Nota: Observaciones realizadas para cada una de las distancias, a las cuales se realizó la transmisión de datos.
Elaborado por: Stalin Mina, Renatto Rugel.

5.2.4.2 Análisis de Distancia vs Retardo

Gráfica Análisis Retardo con Huffman

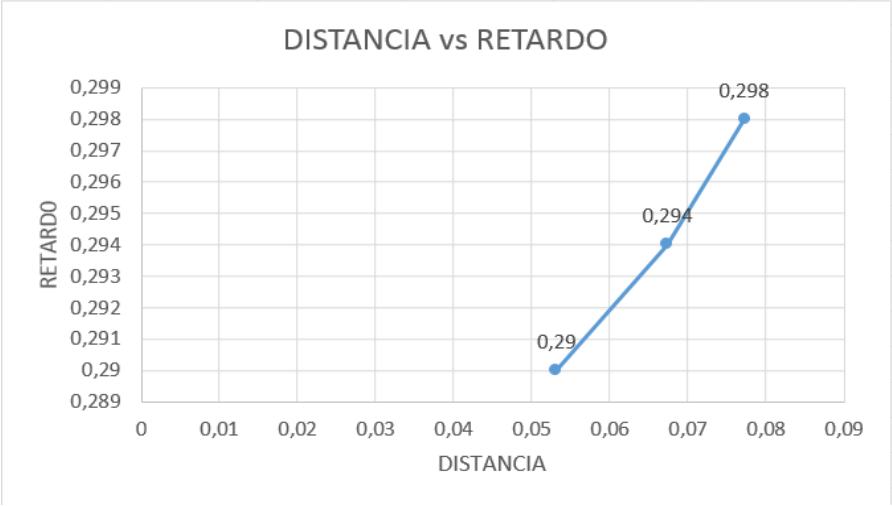


Figura 50. Análisis Retardo 19200bps con Huffman
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 49, el retardo es proporcional a la distancia y sufre un retraso de 0.004 ms, entre las tres velocidades.

5.2.5 Análisis Distancia vs BER

Una vez realizado el análisis de transmisión de datos implementado el Algoritmo de Huffman, a diferentes velocidades y distancias. Se observa que las señales son nítidas a velocidades de 4800-9600bps, se tiene un pequeño retraso en la recepción de la señal. Se determina que la velocidad adecuada para este sistema Li-Fi con Huffman es a 9600bps.

Por medio del código de Huffman creado para la transmisión y recepción, se sacó los bits del mensaje transmitido: “El enojo, el orgullo y la competencia son nuestros verdaderos enemigos” (Lama, 1935)

Bits:

10111100010111100000001100000000110000101111000010110011001111011010000100010010111111000110001010010110100000111111101110110100011100000010001

1100010010110101001000001100001010110010110001100100101010111111011101001101100100110110110100100101010111100000110111111110011101001010101

Tabla 27. Análisis de la Distancia vs BER a una Velocidad de Transmisión de 9600bps

Velocidad de Transmisión (bps)	Distancia (m)	Bits Errados	Cálculo del BER	%
9600	0,0532	0	0,0	0
9600	0,0674	0	0,0	0
9600	0,0772	20	0,071685	7

Nota: Valores obtenidos durante el cálculo del BER.
Elaborado por: Stalin Mina, Renatto Rugel.

Gráfica Análisis Distancia vs BER con Huffman

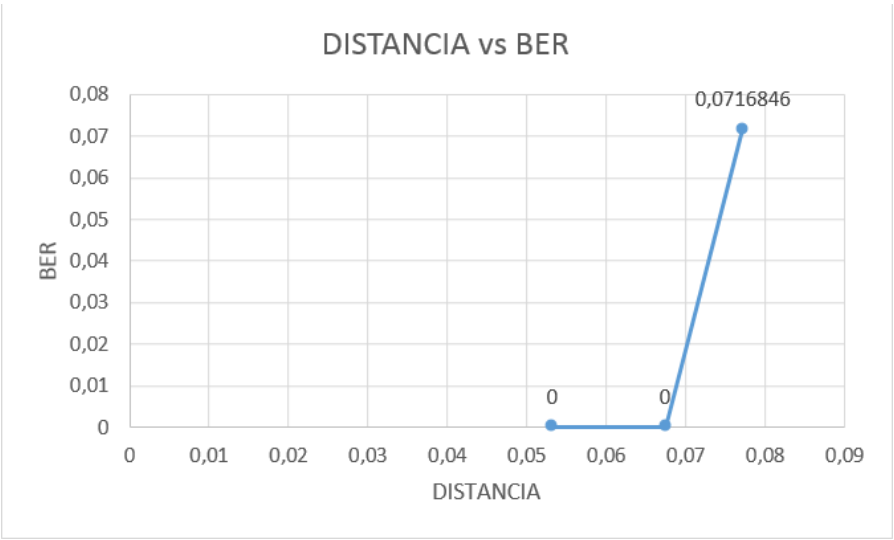


Figura 51. Análisis del cálculo del BER con Huffman
Elaborado por: Stalin Mina, Renatto Rugel

En la figura 51 podemos observar que al realizar la transmisión de datos (279 bits); con una distancia inicial de 0,0532m y 0,0674m, la tasa de error es completamente nula, sin embargo se puede observar que al aumentar la distancia a 0,0772, esta nulidad se pierde y empiezan a existir pérdidas, principalmente debido a que el receptor no es capaz de recibir todos los pulsos de luz que se transmiten a esta última distancia, adicional se observa retrasos en la señal del receptor.

A la distancia de 0,0772 se observa 20 bits errados, que representa el 7% de pérdida.

5.2.6 Análisis de Ancho de Banda

Una vez realizado el análisis de transmisión de datos a diferentes velocidades y distancias se determina que la velocidad adecuada para este sistema Li-Fi es a 9600bps.

Por medio del código creado para la transmisión, se sacó los bits del mensaje transmitido: "El enojo, el orgullo y la competencia son nuestros verdaderos enemigos"

Se realiza el análisis gracias al software Matlab, utilizando la transformada de Fourier que permite analizar las frecuencias de una señal temporal, en la cual se toma los datos de un txt que se tiene guardado los bits del mensaje transmitido.

Basándose en la siguiente formula:

$$AB = f_2 - f_1$$

AB= f2-f1

AB=1,623-1.605

AB=0,018 kHz

Gráfica Análisis Ancho de Banda

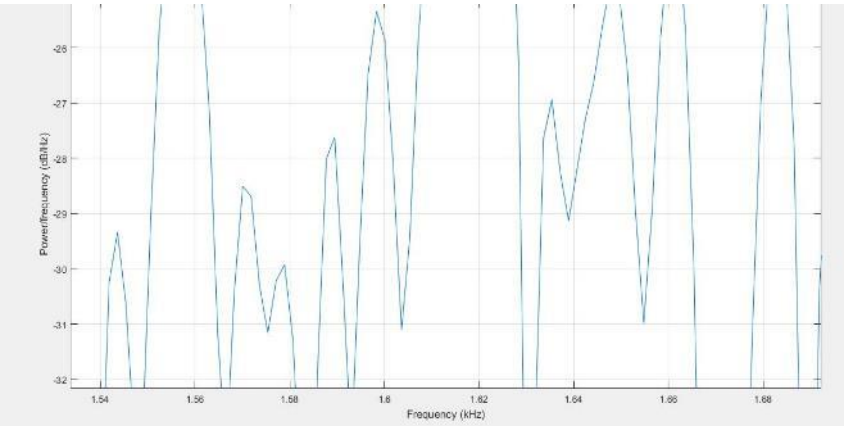


Figura 52. Análisis Ancho de Banda
Elaborado por: Stalin Mina, Renatto Rugel

Como se observa en la figura 52, se obtiene ondas senoidales para la cual se tiene la onda que se encuentre en la Frecuencia 1,613kHz, dato que se obtuvo del osciloscopio y se procede a seleccionar la onda y determinar su frecuencia mayor y menor.

5.3 Análisis de datos obtenidos

Considerando las pruebas realizadas en los apartados anteriores se realizara un análisis más amplio, teniendo en cuenta que la velocidad óptima para utilizar en el sistema Li-Fi con y sin Algoritmo de Huffman es 9600bps.

5.3.1 Análisis Distancia

En la tabla 28, se puede evidenciar que las distancias alcanzadas mediante la implementación del algoritmo de Huffman, aumentaron en una forma considerable. Y se determina que la distancia adecuada es de 0,05 a 0,0674.

Tabla 28. Análisis de la Distancia

Distancia (m)	Distancia (m) Huffman
0,05	0,0532
0,06	0,0674
0,07	0,0772

Nota: Distancias obtenidas.
Elaborado por: Stalin Mina, Renatto Rugel.

5.3.2 Análisis de Frecuencias

En la tabla 29, se observa que la frecuencia aumenta con la implementación del algoritmo de Huffman.

Tabla 29. Análisis de Frecuencias

Frecuencia (Hz)	Frecuencia(Hz) Huffman
1,35	1,613
1,924	4,935
134,28	2,467

Nota: Frecuencias obtenidas.
Elaborado por: Stalin Mina, Renatto Rugel.

5.3.3 Análisis de Retardo

En la tabla 30, se observa que el retardo es proporcional a la distancia, adicional con la implementación del algoritmo de Huffman disminuye el retardo a partir de la distancia 0,06m, inclusive se evidencia transmisión de datos en la distancia 0,07m con un retardo de 0,203ms.

Tabla 30. Análisis de Retardo

Distancia (m)	Retardo (ms)	Distancia (m)	Retardo (ms) Huffman
0,05	0,198	0,0532	0,198
0,06	0,406	0,0674	0,199
0,07	--	0,0772	0,203

Nota: Retardos obtenidos
Elaborado por: Stalin Mina, Renatto Rugel.

5.3.4 Análisis de Transmisión de Datos con éxito

En la tabla 31, se podría decir que la implementación del algoritmo de Huffman ayuda a mejorar la transmisión de datos dentro del canal de comunicación ya que a las tres distancias existe envió correcto de información a pesar que en la distancia 0,0772 existe una perdida mínima de datos, mientras que en el sistema sin algoritmo la perdida es total en los datos.

Tabla 31. Análisis de Transmisión de Datos

Distancia (m)	Observaciones	Distancia (m)	Observaciones con Huffman
0,05	Transmisión Correcta	0,0532	Transmisión Correcta (100%)
0,06	Perdida mínima de datos	0,0674	Transmisión Correcta (100%)
0,07	Pérdida total de datos y distorsión de la señal	0,0772	Perdida mínima de datos (10%)

Nota: Análisis de Transmisión de Datos
Elaborado por: Stalin Mina, Renatto Rugel.

5.3.5 Análisis de Bits transmitidos

En la tabla 33, se evidencia que la implementación del algoritmo redujo la cantidad de bits necesarios para transmitir la cadena de texto “El enojo, el orgullo y la competencia son nuestros verdaderos enemigos”, como prueba. Y como resultado se redujo en un 49,8% el volumen de los datos.

Tabla 32. Análisis Bits transmitidos

Total bits	Código	Total, bits Huffman	Código
560	01000101011011000010000001 10010101101110011011110110 10100110111100101100001000	279	10111100010111100000011 00000001100001011110000 1 01100110011110110100001 0

	0001100101011011000010000 0110111101110010011001111 1010101101100011011000110 1110010000001111001001000 0011011000110000100100000 1100011011011110110110101 1000001100101011101000110 1010110111001100011011010 1011000010010000000111001 1101111011011100010000001 0111001110101011001010111 0110111010001110010011011 1011100110100000001110110 1001010111001001100100011 0001011001001100101011100 0011011110111001100100000 1001010110111001100101011 1101011010010110011101101 1101110011		0010010111111100011000101 0010110100000111111110111 0110100011100000001000111 0001001011010100100000011 0000101011001011000110011 0010101011111101110100111 0110010011011011010010011 0101011110000011011111111 100111010010101

Nota: Análisis Bits transmitidos
Elaborado por: Stalin Mina, Renatto Rugel.

5.3.6 Análisis del Ancho de Banda

En la tabla 32, se podría decir que la implementación del algoritmo de Huffman se reduce en 1% del ancho de banda utilizado, mientras que sin el algoritmo utiliza el 98% y como se indica en el apartado 6.3.5 del presente capitulo el volumen de bits transmitidos se reduce al 49,8%

Tabla 33. Análisis del Ancho de Banda

	Hz	**%**
Ancho de banda	1,33768	98%
Ancho de banda Huffman	0,018	1%

Nota: Análisis del Ancho de Banda
Elaborado por: Stalin Mina, Renatto Rugel.

CONCLUSIONES

- La programación del algoritmo de Huffman, dentro del IDE de Arduino, requirió un 50% más de memoria RAM, debido a esto se tuvo que dividir en varios módulos para su compilación, ya que adicional se necesitó líneas de programación para el funcionamiento de los módulos de Arduino.

- La implementación del algoritmo de codificación de Huffman dentro del sistema Li-Fi, permitió la transmisión de la información con un total de 279 bits, a comparación de la transmisión sin codificación Huffman en la cual se transmitió 560 bits, reduciendo a un 49%, es posible afirmar que existe una mejora en cuanto a la cantidad y velocidad con la que se pueden trasmitir los datos mediante la luz visible, adicional ocasionó que las frecuencias aumentaran, debido a que se necesita un mayor espacio para el envío de datos extras necesarios para la decodificación del código en el lado del receptor.

- Mediante las distintas distancias a las que fueron ubicados el emisor del receptor se pudo constatar el alcance máximo del sistema para la transmisión de datos mediante luz, la cual es de 0,0772m ya que a una distancia superior se pierde la integridad de los datos. Adicional se constató un incremento de (0,0032m - 0,0072m), en las distancias para transmisión de datos entre emisor y receptor debido a la utilización del algoritmo, aumentando así las capacidades del sistema.

- El porcentaje de ancho de banda consumido durante la trasmisión de los datos sin la implementación del algoritmo de Huffman es del 98% disponible dentro del sistema, mientras que, al implementar el mencionado algoritmo, el porcentaje del ancho de banda consumido disminuyó considerablemente hasta representar el 1% total disponible en el sistema.

RECOMENDACIONES

- Para el desarrollo del prototipo fueron utilizadas placas Arduino, las cuales tienen como limitante la cantidad de memoria disponible para el procesamiento y codificación de grandes cantidades de información, por ello se recomienda la investigación de placas como Raspberry Pi para su utilización a futuro en este tipo de proyecto.

- Un fotodiodo mucho más sensible a la luz permitiría alcanzar mayores distancias para la transmisión de información entre el emisor y receptor.

- Se deberían realizar los estudios necesarios que permitan conocer el impacto que este tipo de comunicación inalámbrica tendría dentro de nuestra universidad, empezando por el campus sur, los materiales necesarios para su implementación, costos y beneficios que permitan maximizar/optimizar el acceso a Internet dentro de la institución.

- Lamentablemente no se pudo medir el espectro de luz disponible y el utilizado, debido a la ausencia de equipamiento adecuado para este tipo de medición en la Universidad. Se recomienda adquirir los equipos para futuros proyectos de Tesis o prácticas de laboratorios.

- La medición, tanto de la velocidad como del retardo entre el diodo emisor como en el receptor de luz dentro del sistema, no es posible debido a que la velocidad es una constante, aproximadamente $3x10^8$ m/s, motivo por el cual se realizó la medición desde el momento en que se enviaron los datos hasta el momento en estos llegaron a la página web.

LISTA DE REFERENCIAS

Arduino. (2017). *arduinoBoardMega*. Obtenido de Arduino: https://www.arduino.cc/en/Main/ArduinoBoardMega2560#

Boston, U. (16 de Octubre de 2008). *Science News*. Obtenido de Science Daily: https://www.sciencedaily.com/releases/2008/10/081006130548.htm

Calzadilla, C. (7 de Noviembre de 2013). *Blogthinkbig*. Recuperado el 07 de Junio de 2017, de ALFALITE: http://blogthinkbig.com/vlc-comunicaciones-por-luz-visible/

Campos, A. (26 de Enero de 2016). La conexión 100 veces más rápida que el WiFi, disponible en México. (E. Reina, Entrevistador) Diario El País. DF. México.

Castro, R. (2005). Avanzando en la seguridad de las redes WIFI. *ENFOQUES*, 23-25.

Combariza, L. A., Vargas Roa, C. A., & Acosta, J. D. (2015). *Comunicaciones Inalámbricas.* Recuperado el 07 de Junio de 2017, de Fundación Universitaria Unipanamericana: http://www.profamilia.org.co/docs/docs_blog/Articulo%20-%20La%20tecnologia%20Li-Fi.pdf

Cruz, M. (2015). Obtenido de Blogueando sobre programación!: http://blog.martincruz.me/2012/11/cola-con-prrioridad-en-c.html

Deitel, H. M., & Deitel, P. J. (2012). *Cómo Programar en C++* (Sexta ed.). (L. M. Cruz Castillo, Ed.) México: Pearson Educación .

Dimitrov, S., & Harald, H. (2015). *Principios de las Comunicaciones de Luz LED.* Reino Unido. Recuperado el 16 de Mayo de 2017

Educación, T. y. (30 de Mayo de 2012). Obtenido de Tecnología y Educación: http://www.tecnologiayeducacion.com/%C2%BFque-es-luz-led/

Fernández, C. (2012). Obtenido de Estructura de Datos: http://nuyoo.utm.mx/~caff/doc/Notas%20de%20Estructura%20de%20datos.pdf

González, A., & Aravena, E. (Septiembre de 2013). *Proyectos.* Recuperado el 07 de Junio de 2017, de Universidad Técnica Federico Santa María: http://profesores.elo.utfsm.cl/~agv/elo322/1s13/project/reports/Desafios_de_las_nuevas_tecnologias.pdf

Greenpeace. (Abril de 2012). *Global.* Obtenido de Greenpeace: http://www.greenpeace.org/argentina/Global/argentina/report/2012/4/leds.pdf

Gupta, S. U. (6 de Junio de 2015). *International Journal of Advanced Research in Computer Science and Software Engineering.* Obtenido de ijarcsse: http://www.ijarcsse.com/docs/papers/Volume_5/6_June2015/V5I6-0175.pdf

Haro Soledispa, K. L., & Buenaño Medina, A. D. (2016). *Diseño e Implementación de un prototipo de Tecnología Visible Light Comunication (LI-FI) para transmitir datos de manera unidireccional.* Universidad Politécnica Salesiana, Quito. Obtenido de http://dspace.ups.edu.ec/bitstream/123456789/7770/1/UPS-ST002144.pdf

International Centre for Theoretical Physics. (01 de Noviembre de 2011). *Es la RED.* Obtenido de http://www.eslared.org.ve/walc2012/material/track1/12_Telecommunicaciones%20Basicas_v1.pdf

Jaya, M., Vizcaíno, I., & Acosta, F. (2014). *MONITORIZACIÓN DE CONSUMO ENERGÉTICO EN REDES INALÁMBRICAS WI-FI CON DIFERENTES ESTÁNDARES DE COMUNICACIONES Y VOLÚMENES DE TRÁFICO.* Quito.

Lama, D. (1935).

Longás, H. (31 de Enero de 2016). *Tecnología*. Obtenido de EL PAÍS: http://elpais.com/elpais/2015/12/24/media/1450971049_954491.html

PromeTec. (11 de Noviembre de 2015). Obtenido de PromeTec: http://www.prometec.net/arduino-wifi/

pureLIFI. (2014). *Features*. Obtenido de PURELIFI: http://purelifi.com/what_is_li-fi/li-fi-features/

pureLiFi. (2014). *The LIFI STORY*. Recuperado el 09 de Junio de 2017, de pureLiFi: http://purelifi.com/what_is_li-fi/the-lifi-story/

Quinatana Sánchez, C. (2012). *Contribución al estudio de estrategias de Acesso al Medio en Redes Óticas no guiadas.* Las Palmas de Gran Canaria: ULPGC.

Rajagopal, S., Roberts, R., & Lim, S.-K. (3 de Marzo de 2012). *IEEE XPLORE.* doi:10.1109/MCOM.2012.6163585

Rodriguez Jimenez, C. A. (5 de Febrero de 2011). *Cisco*. Obtenido de http://ciscocarj2011.blogspot.com/2011/02/transmision-sincrona-y-asincrona.html

Santillo, W. (Abril de 2012). *Iluminación Eficiente.* Recuperado el 10 de Junio de 2017, de Greenpeace: http://www.greenpeace.org/argentina/Global/argentina/report/2012/4/leds.pdf

Sayood, K. (2012). *Introduction to Data Compression.* Nebraska: Elsevier, Inc.

sparkfun.com. (s.f.). Obtenido de SparkFun Electronics Inc: https://www.sparkfun.com/products/11061

Stallings, W. (2012). *Comunicaciones y Redes de Computadoras* (Sexta ed.). (A. Prieto, Trad.) Madrid: Prentice Hall.

Unicrom, E. (2016). Obtenido de Electrónica Unicrom: http://unicrom.com/fotodiodo/

Wikipedia. (8 de Mayo de 2011). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Fotodiodo