IMPLEMENTACIÓN DE UNA ESTANTERÍA INTELIGENTE SOPORTADA EN TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA PARA APORTAR SOLUCIONES A LA LOGÍSTICA MINORISTA.

ALEX FERNAN PARRA MERA
JHOIMAR EDUARDO SALAMANCA RODRÍGUEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2015

IMPLEMENTACIÓN DE UNA ESTANTERÍA INTELIGENTE SOPORTADA EN TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA PARA APORTAR SOLUCIONES A LA LOGÍSTICA MINORISTA.

JHOIMAR EDUARDO SALAMANCA RODRÍGUEZ
ALEX FERNÁN PARRA MERA

Trabajo de Grado para Optar el Título de
Ingeniero Electrónico

Director
M.I.E. (C) EFREN DARIO ACEVEDO CÁRDENAS
Ingeniero Electrónico

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2015

A Dios, que siempre me guía y me fortalece.

A mi familia por su compresión, amor y apoyo incondicional.

A mi novia y amigos porque siempre han sido como una familia para mí.

Alex Fernán Parra Mera.

A Dios y mis padres que me han permitido llegar hasta aquí.

A mi director de grado Efrén Darío Acevedo a mi compañero Alex Fernan Parra Mera y a todas las personas que apoyaron este proceso

Jhoimar Eduardo Salamanca Rodríguez.

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

- Dios, por permitirnos realizar y terminar este proyecto.

- Ing. Efrén Darío Acevedo Cárdenas por su colaboración y orientación durante el proyecto.

- Ing. Diana Carrillo, por su colaboración, orientación y paciencia durante la realización del proyecto.

- Ing. Jorge Hernando Ramón Suarez, por su colaboración, orientación y tiempo compartido en el desarrollo de este proyecto, a él mil gracias.

- Nuestros padres y amigos con quienes hemos contado siempre y a quienes les debemos quienes somos.

CONTENIDO

LISTAS DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

TITULO: IMPLEMENTACIÓN DE UNA ESTANTERÍA INTELIGENTE SOPORTADA EN TECNOLOGÍA DE IDENTIFICACIÓN POR RADIOFRECUENCIA PARA APORTAR SOLUCIONES A LA LOGÍSTICA MINORISTA*.

AUTORES: JHOIMAR EDUARDO SALAMANCA RODRÍGUEZ, ALEX FERNÁN PARRA MERA**

PALABRAS CLAVE: Piloto Electrónico, Plataforma software, Interfaz Gráfica, Tecnología RFID.

DESCRIPCIÓN:

Con este proyecto se busca implementar un piloto electrónico de captura de información en productos contenidos dentro de una estantería. El trabajo de investigación busca diseñar e integrar una plataforma software que adquiera información en tiempo real de los elementos y su posición dentro del espacio de almacenamiento sobre una interfaz gráfica, además, el proyecto se soportará en un profundo análisis de los desafíos presentes en la comunicación de datos con la tecnología RFID que se puedan presentar en el desarrollo de éste.

El objetivo de la investigación consistió en brindar a las pequeñas y medianas empresas en la región una herramienta para realizar el inventariado de sus productos, y llevar control de lo que está puesto en estantería, haciendo más rápido, eficiente y confiable este proceso de manera alternativa, usando las antenas de tecnología RFID, buscando así, ser más competente este sector económico frente a nuevos comercios que podrían llegar pronto a competir por el mercado.

El trabajo hace parte del diseño e implementación de un prototipo de laboratorio para evaluación de la tecnología RFID en escenarios logísticos planteado dentro del proyecto de innovación y desarrollo tecnológico patrocinado por Colciencias "*Diseño e implementación de mecanismos y herramientas conceptuales y tecnológicas para el mejoramiento del sistema logístico de la empresa COMERTEX S.A*" en el cual participan los grupos de investigación CPS y OPALO de las escuelas de Ingeniería Eléctrica, electrónica y Telecomunicaciones y Escuela de Estudios Industriales y Empresariales, respectivamente.

---

*Trabajo de Grado

**Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones (E^3T), Director MIE(c). Efrén Darío Acevedo Cárdenas, Codirector M.Sc. Oscar Mauricio Reyes Torres.

ABSTRACT

TITLE: IMPLEMENTATION OF AN INTELLIGENT RACK SUPPORTED IN RADIO FREQUENCY IDENTIFICATION TECHNOLOGY TO PROVIDE SOLUTIONS TO RETAIL LOGISTICS*.

AUTHORS: JHOIMAR EDUARDO SALAMANCA RODRÍGUEZ, ALEX FERNÁN PARRA MERA**

KEY WORDS: Electronic Pilot, software platform, Graphic Interface (GUI), RFID Technology.

DESCRIPTION:

This project seeks to implement an electronic-pilot to capture information of products contained within a bookshelf. The research aims to design and integrate a software platform that acquires real-time information of the elements and their position within the storage space on a GUI, besides, the project is supported by the analysis of the challenges present in the data communication with the RFID technology that may arise in the development of this.

The aim of the research is to provide small and medium enterprises in the region a tool to make an inventory of their products, having more control about the products in the shave, making this process faster, efficient and reliable in an alternatively way, using RFID antennas, looking well, be more competent this industry against new businesses that could soon come to compete for the market.

The work is part of the design and implementation of a laboratory prototype for evaluation of RFID technology in logistics scenarios presented within the project of innovation and technological development sponsored by Colciencias, "Design and implementation of mechanisms conceptual and technology tools for the improving of company's logistic system, COMERTEX.S.A", in which CPS and OPAL research groups from schools of Electrical, Electronical and Telecommunications, School of Industrial Studies and Management, respectively have been involved.

---

*Bacherlor Thesis

**Physical-Mechanical Engineering Faculty. Electrical, Electronics and Telecommunications (E^3T) Engineering School, Director MIE(c). Efrén Darío Acevedo Cárdenas, Codirector M.Sc. Oscar Mauricio Reyes Torres.

INTRODUCCION

Hoy en día el nivel de producción de una empresa está relacionado con cuan atractivo se es para los clientes, entre más competitiva sea una compañía en el mercado, aumentarán las probabilidades de producción y consecuentemente mayores serán sus ingresos. Uno de los métodos para lograr esto consiste en desarrollar procesos logísticos eficientes, revisando y controlando permanentemente el estado del producto en diferentes partes de la cadena de abasto. Actividades tales como el inventariado del producto, control de salida e ingreso de la mercancía, pueden ser mejoradas y optimizadas implementadas tecnologías dirigidas en la identificación electrónica del producto.

El método de identificación electrónica de productos más usada a nivel mundial ha sido el código de barras, sin embargo, en la actualidad han venido surgiendo nuevas tecnologías que son consideradas la evolución de esta misma, ejemplo de esto es el código electrónico de producto, la codificación 3D, la logística manejada por voz, la identificación por radio frecuencia, entre otras.

La identificación por radiofrecuencia (RFID, por su sigla en inglés) se ha ido haciendo más popular entre las grandes empresas debido a su simple implementación y las diversas aplicaciones a las que se puede adaptar, “Los sistemas de RFID están transformando los procesos logísticos alrededor del mundo” [1]. A través de señales inalámbricas similares a las que utilizan los teléfonos celulares, esta tecnología permite dar seguimiento a los productos a través de toda la cadena de suministro desde la fabricación de un producto hasta su servicio de posventa, de forma rápida y automática; debido a las enormes posibilidades que ofrecen las etiquetas RFID en las que es posible incluir una gran cantidad de datos relacionados al producto, sumado a otras serie de características como la durabilidad de los tags y la facilidad de uso, esta tecnología ha logrado insertarse en el mercado empresarial cada vez con más presencia.

En este proyecto, se busca realizar a partir de técnologia RFID la implementación de una Estantería Inteligente aprovechando los beneficios que ésta brinda, para ello se utilizó dipositivos hardware tales como Antenas Planas, Readers, Tags y otros recursos software. El funcionamiento a groso modo del sistema, empieza una vez colocado el producto en los anaqueles del estante en donde hay una antena plana en la base de ésta, cada artículo con su respectivo Tag es detectado y esta información es enviada al Reader, una vez ahí, es llevada al computador para que este los guarde.

Para archivar los datos, es necesario un sofftware especial que tenga la capacidad necesaria para la aplicación, el uso de bases de datos para la Estantería Inteligente es fundamental para su desarrollo, ya que en la realización de la Interfaz gráfica, el usuario podrá acceder a toda la información acerca de los productos de forma fácil, además de brindar la herramienta de estantería virtual en donde se puede observar los productos en tiempo real.

1. PLANTEAMIENTO DEL PROBLEMA

Dar el siguiente paso es el propósito de las empresas para ser competitivos en el mercado, día a día están en búsqueda de mejoras en los procesos logísticos para avanzar y obtener mayores resultados en sus respectivos planes estratégicos. Uno de los principales problemas que se han evidenciado son la manipulación de procesos de administración de mercancías, que a pesar de ser realizado por personal capacitado presenta inconvenientes en términos de eficiencia y vulnerabilidad a errores humanos respecto a la localización, verificación, revisión e información del producto. Con el fin de cumplir los objetivos y alcanzar las metas propuestas, las empresas se ven impulsadas a una continua investigación y desarrollo de nuevas tecnologías aplicadas a los diferentes procesos logísticos para optimizar y mejorar los resultados [2].

Como solución a las problemáticas presentadas anteriormente, se propone la implementación de la Estantería Inteligente, con la cual tareas como inventariado, verificación de la ubicación de algún elemento presente en la empresa o tienda, información del producto a la mano como la fecha de vencimiento, fecha de ingreso entre otros se pueden automatizar y optimizar reduciendo tiempos en la realización de dichas actividades.

En el caso concreto de este proyecto, se busca solucionar aspectos de control y flujo de productos en la estantería de las diferentes tiendas y almacenes, brindando la posibilidad de acceder a la información oportuna en una base de datos, optimizar tiempos de inventariado y venta, además de observar por medio de una plataforma gráfica el comportamiento de los productos en los anaqueles.

2. JUSTIFICACIÓN

El crecimiento de una empresa es vital, día a día deben innovar para aumentar el nivel competitivo en el mercado y de tal manera obtener beneficios tanto económicos como culturales, de tal manera que se debe implementar tecnología para optimizar la manera de realizar los procesos logísticos, teniendo como resultado un aumento en su rendimiento, disminución del tiempo de los procesos, costos y mejorar su productividad. La tecnología utilizada para la implementación es la identificación de productos por radiofrecuencia (RFID), se busca reducir tramites de documentación, disminuir los tiempos de algunos procesos logísticos, bajar costos producidos por el inventariado y corregir lo que llamamos error humano a la hora de recolectar datos, otorgarle información específica a cada etiqueta del producto implementado.

Con este proyecto se busca implementar un piloto electrónico de captura de información en productos contenidos dentro de una estantería, proporcionando toda clase de datos deseados por medio de una plataforma software que adquiera información en tiempo real de los elementos y su posición dentro del espacio de almacenamiento sobre una interfaz gráfica.

El proyecto espera optimizar el proceso logístico de inventariado suministrando toda la información de los productos en menor tiempo y con certeza; proporcionar información inalámbrica al usuario de la ubicación del producto en tiempo real y otorgar una aplicación innovadora con el uso de la estantería inteligente.

3. OBJETIVOS

3.1. Objetivo General.

Diseñar e implementar una estantería inteligente para aportar soluciones a la logística minorista.

3.2. Objetivos Específicos.

- Implementar el desarrollo de una estantería inteligente soportada en tecnología de identificación por radiofrecuencia (RFID).

- Realizar una interfaz gráfica en donde el usuario pueda obtener información virtual de la estantería inteligente en tiempo real.

- Analizar los desafíos presentes en la comunicación de datos con la tecnología RFID pasiva en espacios con alta densidad de productos.

- Analizar y almacenar las lecturas en tiempo real otorgados por el lector RFID speedway R420 de las etiquetas RFID presentes en la Estantería Inteligente.

- Documentar el desarrollo de la investigación, para que sea punto de referencia de trabajos futuros.

4. ALCANCES

En la consecución de este proyecto se implementará tecnología RFID en el desarrollo de la Estantería Inteligente, estos equipos permitirán realizar la recolección de los datos y su almacenamiento. El principal dispositivo a utilizar es el Reader Speedway R420, el cual limita los demás elementos hardware tales como antenas, tipo de cable y elementos de expansión como los Hubs y GPIO; que busca convertir un simple estante en un elemento inteligente.

Se realizará pruebas para establecer que limitantes se puede presentar a la hora de realizar lectura de una cantidad considerable de productos, las cuales se llevaran a cabo utilizando los protocolos disponibles en el Reader Speedway R420, brindando una mejor caracterización del sistema y funcionamiento óptimo de los elementos.

Una vez el sistema en marcha, la encargada de almacenar los datos y su respectiva administración será hecha por la base de datos, donde se guardarán aspectos de lectura tales como el puerto de la antena en el que está ubicado el producto, EPC que contiene la información correspondiente al Tag, Potencia el cual permite saber que tan próximo a la antena está ubicado dicho elemento en la Estantería Inteligente en tiempo real.

Finalmente, este proyecto busca realizar un piloto que contará con una plataforma software que adquiera la información en tiempo real de la base de datos, desde esta interfaz el usuario podrá acceder a las lecturas realizadas por el Reader, además de observar desde una representación virtual 2D si un elemento se encuentra dentro o fuera de la Estantería Inteligente.

5. DESCRIPCION DEL PROCESO

5.1. SISTEMAS RFID

Un sistema de RFID (*Radio Frequency Identification*) es la tecnología inalámbrica que permite establecer la comunicación entre un lector y una etiqueta adherida a un producto. Estos sistemas permiten almacenar información en sus etiquetas mediante comunicaciones de radiofrecuencia. Esta información puede ir desde un bit hasta un Kbyte. Dependiendo principalmente del sistema de almacenamiento que posea el transponder [2].

Un Tag o etiqueta electrónica contiene un microchip y una antena, que puede adherirse a cualquier producto. Incluso se están desarrollando tags que son de un tamaño tan pequeño que pasarían inadvertidos en algunos objetos. El microchip almacena un número de identificación único e inequívoco, existen diferentes tipos de estándares propuestos para estos números, por ejemplo, el Electronic Product Code (EPC) diseñado por Auto-ID center. Es decir, cada objeto tendrá un código único que lo diferenciara e identificará no solo de otros tipos de productos, sino de productos iguales [3].

El funcionamiento del sistema, es a priori, bastante sencillo, primero el lector envía una serie de ondas de radiofrecuencia al Tag, que son captadas por una antena puesta en él, dichas ondas activan el microchip, segundo, el lector envía de vuelta al lector la información que tenga en su memoria, finalmente el lector recibe la información que tiene el Tag y lo envía a una base de datos en la que previamente se han registrado las características del producto o puede procesarlo según convenga a cada aplicación [4].

La comunicación entre el lector y la etiqueta se realiza mediante señales de radiofrecuencia a una determinada frecuencia que generan las antenas del lector y la etiqueta. La comunicación entre ellas tiene unas determinadas características de alcance, velocidad y seguridad según el rango de frecuencia, el tipo de antenas utilizadas, el tipo de etiquetas y demás parámetros que se puedan configurar para una aplicación u otra [5].

En equipos de RFID es posible observar protocolos anticolisión que permiten leer varias tarjetas al mismo tiempo. En caso de que varias tarjetas estén en el rango de alcance del interrogador y dos o más quieran transmitir al mismo tiempo, se produce una colisión. El interrogador detecta la colisión y manda parar la transmisión de las tarjetas durante un tiempo. Después irán respondiendo cada una por separado por medio de un algoritmo bastante complejo. Obviamente a mayor capacidad de la etiqueta y el lector, más efectivos serán estos algoritmos [6].

El funcionamiento de los dispositivos RFID se realiza entre los 50 KHz y 2.5GHz. Las unidades que funcionan a bajas frecuencias (50 KHz – 14 MHz) son de bajo coste, corto alcance y resistentes al ruido en tres otras características. No se requiere licencia para operar en este rango de frecuencia. Las unidades que operen a frecuencias más altas (14 MHz – 2.5 GHz), son sistemas de mayor coste y tecnología más compleja [7].

La etiqueta contiene información que puede ser solo leída o puede permitir la escritura, dependiendo del tipo de memoria que posea el transponder. La mayor parte de los sistemas tienen memoria EEPROM (Electically Erasable Programmable Read-Only Memory). En algunos casos llevan datos guardados de fábrica y en otros se puede grabar por parte del usuario. El usuario habitualmente recibe esta información en un lector portátil con un display alfanumérico o puede pasar directamente a un ordenador que procese los datos obtenidos [8].

Para la creación de un sistema RFID hay que tener en cuenta diversos factores de diseño como el rango de alcance donde se puede mantener la comunicación, la cantidad de información que puede almacenar el transponder. La velocidad de flujo de datos que se puede obtener entre el lector y etiqueta, el tamaño físico de la etiqueta, la habilidad del lector para mantener la comunicación con varias etiquetas a la vez o la robustez que ofrece la comunicación a posibles interferencias de materiales entre el lector y etiqueta [9].

Los sistemas RFID tienen la ventaja de su total funcionamiento sin visibilidad directa entre lector y etiqueta. En este aspecto es donde claramente supera al código de barras y otros sistemas ópticos. Pero debido a su coste, que aunque ha ido reduciéndose progresivamente, siempre será superior al del código de barras, no se ha implementado en aplicaciones sencillas donde el código de barras sigue dominando el mercado. Pero es en las aplicaciones donde el código de barras y la tecnología óptica es más limitada y no resultan efectivos, donde el crecimiento de la tecnología RFID es más notorio [10].

5.2. ESTANTERIA INTELIGENTE

Una estantería inteligente es un estante que se ha equipado con lectores RFID. Al escanear continuamente las etiquetas RFID implementadas en los diferentes artículos, el lector RFID notifica consistentemente información de dichos artículos, por ejemplo, ubicación, identificación de los elementos que no pertenezcan al estante como “elementos fuera de lugar”, dando como resultado el estudio no solo de las decisiones de los clientes sino también de los mismos productos [11].

El fin de la Estantería Inteligente es mejorar la forma en que las pequeñas empresas realizan actividades como reposición de productos, inventarios y productos vendidos, en relación a lo anterior, se presentará a continuación los aspectos relacionados con el funcionamiento de la Estantería Inteligente y sus componentes [11].

5.2.1. Estantería. La parte Física de la Estantería Inteligente consta de un mueble con anaqueles o entrepaños, y generalmente sin puertas, que sirve para colocar libros, papeles u otras cosas, en estos compartimentos se colocan los productos al cliente, cada producto tiene la respectiva etiqueta con la cual la antena puesta en la base del compartimento leerá si este está o no esta en la Estantería Inteligente. Una vez hecha la lectura, esta será enviada hacia los dispositivos en los cuales el usuario pueda recibir, organizar y guardar la información para su uso eficiente en para la empresa.

5.2.2. Conexiones. Todas las antenas que están en la base de los compartimentos, tienen puertos para la conexión hacia el Reader, sin embargo, en un caso práctico en el que se utilicen más de una Estantería Inteligente también se utilizará más de una antena. Un Reader consta solo de 4 puertos, para dar solución a lo anterior, la compañía de tecnología RFID IMPINJ ofrecen la posibilidad de conectar hasta 32 antenas al Reader utilizando periféricos de entrada/salida, así un solo Reader puede manejar la información proveniente de diferentes Estanterías Inteligentes.

5.2.3. Lecturas. Luego de tener a punto la Estantería Inteligente, es decir, tener las antenas puestas en la base del compartimento, tener los productos con su respectivo Tag, los productos empezaran a enviar información al Reader, para esto es necesario saber que método de recolección de la información entre la base de datos y el Reader, teniendo en cuenta que se tiene más de una forma para el acceso

a la información de las lecturas realizadas por él. Para la recibir esta información es necesario utilizar un método que permita la administración de los datos de manera que se puedan almacenar y procesar.

5.2.4. Procesamiento de los datos. Luego de obtener los datos mediante una de las formas de recolección de información establecidas por el Reader Speedway Revolution, estos se tendrán que guardar ya sea en un documento o en base de datos; esto permite que el usuario tenga acceso a la información en cualquier momento que sea necesario, ya sea para verificar la venta de productos o para observar el comportamiento de los productos en la Estantería Inteligente.

5.2.5. Interfaz Gráfica. Es muy importante en un proyecto llevar todo a la simplicidad, por ende para el uso efectivo de la información suministrada por el Reader y para una mejor comprensión de lo que se tiene en la base de datos, es necesario crear un ambiente gráfico en el que el usuario se sienta cómodo. La interfaz gráfica de la Estantería Inteligente permite a la persona verificar en la base de datos la información en ella, de una forma intuitiva, también brinda información sobre los productos que el cliente desea, y por último, una representación 2D de lo que está ocurriendo en la Estantería Inteligente, esto último permite llevar un mayor control sobre los productos puestos para el cliente, brindando más seguridad y cobertura [12].

6. ARQUITECTURA Y SOFTWARE

El uso de elementos apropiados para la ejecución de una labor es tan importante como la planeación, en la Estantería Inteligente que se propone se implementará tecnología RFID en el etiquetado de los productos, para ello, es necesario la utilización de antenas, dispositivos y programas software utilizados para la captura e identificación de las antenas puestas en el producto [13].

¿Por qué elegir una tecnología específica?, muchas veces al escoger uno de los dispositivos acota la selección de otros dispositivos que se deben usar en relación con la compatibilidad o facilidad de uso, en este caso, todo el hardware necesario estaba ya estipulado y el Reader escogido para esta tarea, limitó los componentes y métodos a utilizar para la consecución de la Estantería Inteligente.

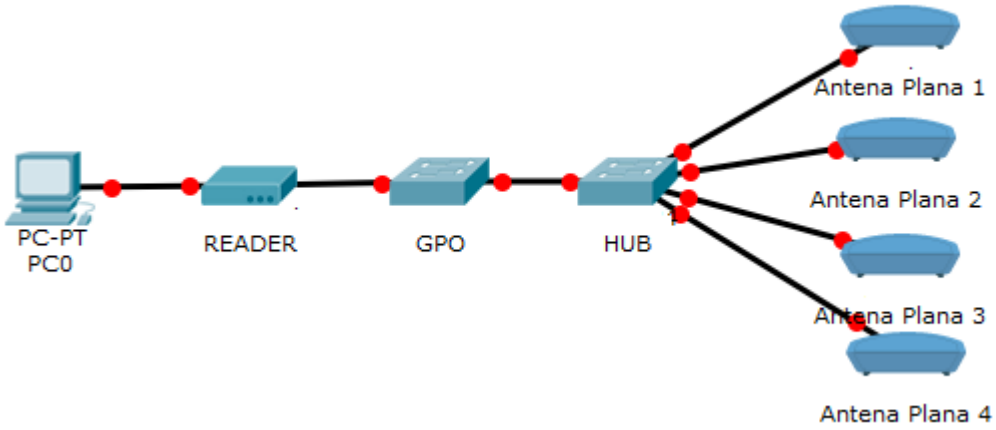
Entre los proveedores de tecnologías RFID, IMPINJ ofrece los equipos y todo lo necesario para el funcionamiento de las diferentes aplicaciones desarrolladas por sus clientes, dando soporte bibliográfico y técnico. IMPINJ ofrece el Reader Speedway Revolution, que puede ser utilizado para el fin de una Estantería Inteligente, el Reader de 4 puertos de lectura para antenas mono-estáticas, puertos de comunicaciones: RS-232, RJ45, 4 inputs y 8 outputs. Soporta protocolo LLRP Versión 1.0.1. Su aspecto puede verse en la Figura 1. Para su funcionamiento, el Reader posee su propio software provisto por IMPINJ [14].

En el desarrollo de la Estantería Inteligente se utilizaron diferentes herramientas tanto hardware como software, que permitieron realizar las tareas de recolección, envío y almacenamiento de datos. A continuación en este capítulo se describe los dispositivos, gestores y lenguajes utilizados en la consecución del proyecto.

6.1. HARDWARE

En la siguiente sección, se presenta los elementos físicos que hacen parte de la Estantería Inteligente, cada uno de ellos se describe tanto en funcionalidad e implementación en el desarrollo del proyecto.

Figura 1. Esquema de Conexión de los Dispositivos.

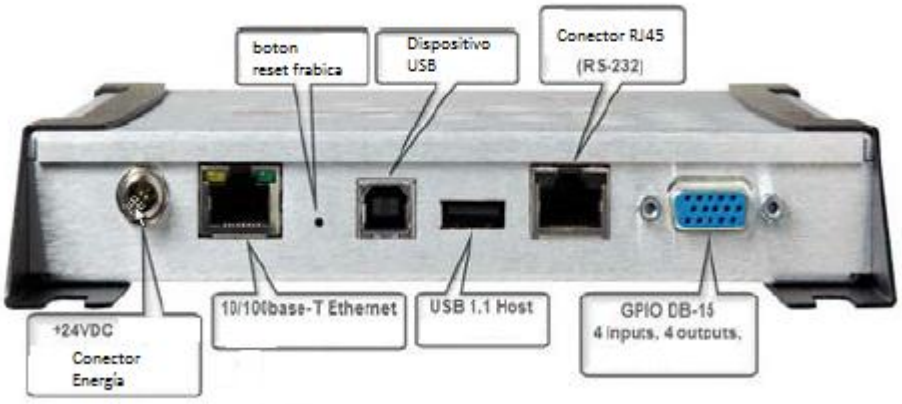


En la Figura 1 se puede observar la conexión general de los dispositivos que hacen parte de la Estantería Inteligente. En primer lugar, las antenas planas 1, 2 3, y 4 realizan la lectura de las etiquetas en la cajonera, la antena HUB funciona como un multiplexor en tiempo, el GPIO realiza la tarea de sincronización de las antenas HUB para enviar dicha información al Reader, una vez en el Reader este la envía al computador para su posterior almacenamiento y uso.

6.1.1. Lector Speedway Revolution. Un lector RFID es un dispositivo electrónico que convierte información de un soporte determinado en otro tipo de señal, para procesarla informáticamente o reproducirla por otros medios, dicho aparato contiene un módulo de RF, el cual actúa como un transmisor y receptor de señales de frecuencia de radio. El transmisor consta de un oscilador para crear la frecuencia portadora; un modulador incida comandos de datos sobre esta señal portadora y un amplificador para aumentar la señal suficiente para despertar la etiqueta.

El Reader Speedway Revolution posee cuatro puertos para antenas RFID, un puerto Ethernet que permite la conexión y comunicación cableada entre el Reader y el computador, un puerto GPIO DB-5 que permite una conexión serial entre el Reader y el computador para el envío de datos, también sirve para la sincronización y uso de antenas HUB en el Reader , un puerto USB para el envío de datos por cable USB, un puerto USB Device para el envío de datos por Keyboard desde el Reader al computador , y su respectivo puerto para conectar la energía [14], y se puede ver en la Figura 2.

Figura 2. Puertos Reader Speedway Revolution



Provisto por IMPINJ. Configuración de Puertos Speedway Revolution. Tomado de [14].

6.1.1.1. Configuración Física. La conexión del Reader Speedway Revolution para obtener información de etiquetas se debe realizar el siguiente procedimiento [15].

1. Se conectan las antenas al puerto correspondiente detrás del Reader.
2. Se conecta el Reader, esto hará que el automáticamente encienda.
3. Para conectar al computador, se puede hacer de dos formas. Primero, se puede conectar directamente el Reader al computador por medio del cable y puerto Ethernet. Segundo, se conecta el Reader al modem de internet, esto hará que cualquier computador con acceso a la red pueda establecer conexión con el Reader.

6.1.2. Conexión Reader-Computador. Para establecer la conexión funcional entre el Reader y el computador, es necesario conocer el nombre del Reader para crear la vía de datos, en este caso, suele ser “speedway” seguido de los últimos 6 dígitos de la MAC del Reader que vienen en la parte lateral en la etiqueta separados por el signo menos (-) seguido de la palabra “.local”; Para el caso en particular, este nombre es speedwayr-10-a0-70.local.

6.1.3. Speedway Reader Antena HUB. La Antena HUB surge de la necesidad de tener más puertos para el manejo de antenas; el Reader solo cuenta con 4 puertos para antenas. Cada Antena HUB permite la conexión de ocho antenas, para un máximo de 32 antenas por cada Reader [16].

Figura 3. Antena Hub.

Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [14].

La instalación de una antena Hub es sencilla, ya que esta no necesita de una configuración de Software, simplemente se conecta de manera correcta y esta funcionará. Para el funcionamiento es necesario la instalación de un adaptador GPIO que sincroniza la conexión de máximo cuatro antenas Hub [16].

6.1.4. Adaptador GPIO. El adaptador GPIO, es necesario para la conexión exitosa entre las antenas Hub y el Reader, una vez las antenas Hub están conectadas por medio del cable coaxial éste permite un acceso conveniente al puerto GPIO, realizando la tarea de sincronización de los datos provenientes de las antenas en las cajoneras [16].

El adaptador GPIO consta de 5 puertos, 4 de ellos para conexión Ethernet con las antenas HUB para sincronización de datos, y un puerto serial para la conexión entre adaptador y Reader.

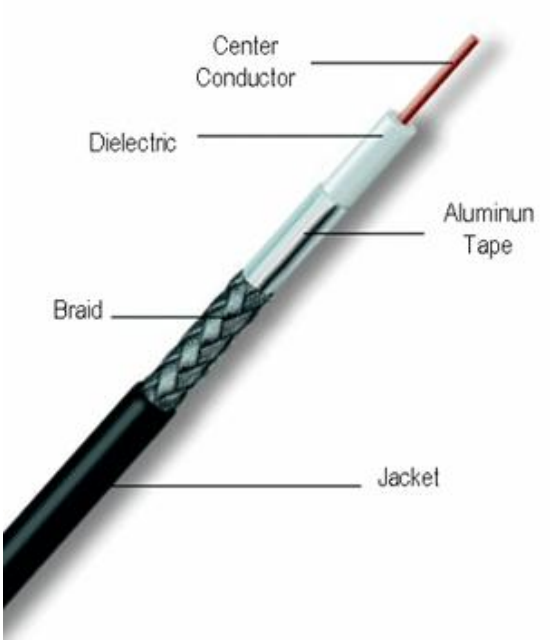
Figura 4. Adaptador GPIO.

Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [14].

6.1.5. Cable Coaxial y Conectores. El cable coaxial, es usado para la conexión de las antenas puestas en la Estantería Inteligente con la antena Hub, una vez conectadas las antenas es necesario conectar la antena Hub con uno de los puertos Reader, para esto también se necesita realizar la conexión con cable coaxial para tener todo debidamente conectado. Ver Figura 5.

Los conectores usados RP-SMA los cuales son usados en cable coaxial para aplicaciones RFID, tipo plug con terminación soldada, posee una impedancia de 50 Ohm y una frecuencia de trabajo máxima de 6 [GHz]. Ver Figura 6.

Figura 5. Cable RFID.



Provisto por Perfect RFID. Tomado de [17].

Figura 6. Conectores RP-SMA

Provisto por Perfect RFID. Tomado de [17].

6.1.6. Antenas de la Estantería Inteligente. Para la Estantería Inteligente es necesario utilizar una antena plana que se pueda colocar en la base del compartimento; se ajustan a secciones con poco espacio y cubren mayor área que las antenas tradicionales. Existen diferentes compañías que proveen antenas planas tal es el caso de IMPINJ que para el mejor funcionamiento de los Readers, aconsejan utilizar la antena Slimline A7075 de TIMES-7 [17].

La antena permite percibir las etiquetas RFID que están próximas a ella, es decir, todos los productos colocados dentro del compartimiento.

Figura 7. Antena Slimline A7075 Linear Polarised UHF RFID Shelf Antenna

Provisto por Perfect RFID. Tomado de [17].

6.2. SOFTWARE

6.2.1. Sistema operativo del computador. Como es conocido, hoy día las personas están acostumbradas a trabajar en sus computadoras con el sistema operativo Windows, ya que es mucho más visual y amable con el usuario. La principal desventaja de Windows como sistema operativo es el pago de licencias para poder tener uso de sus servicios. Existen sistemas operativos de uso libre, es decir, la persona no paga una licencia, están hechos para que cualquier persona pueda utilizarlo, solo que sus entornos son un poco más técnicos que el de Windows. Sistemas operativos como estos los brindan compañías como Linux, el

cual ha creado varios sistemas operativos como UBUNTU soportado en su propio lenguaje.

Hoy en día, más y más empresas están empezando a utilizar software libre, a tal punto que la mayoría de los dispositivos diseñados para la Industria, están pensados para uso en sistemas operativos Linux, lo que permite mayor interacción con el dispositivo, IMPINJ no es la excepción, aunque también se puede trabajar en sistemas operativos Windows, la mayoría de las aplicaciones necesitan de un software con licencia para usarlas, por ende es más cómodo y económico trabajar en una plataforma Linux. Para el caso en particular, se trabaja con sistema operativo Ubuntu, una de las múltiples plataformas software de Linux. Ubuntu permite por medio de línea de comandos, realizar las configuraciones necesarias para manejar el dispositivo, además no necesita de software secundario para poder ejecutar las aplicaciones del Reader. Por los motivos expresados anteriormente, Ubuntu es el sistema operativo escogido para el desarrollo de la Estantería Inteligente.

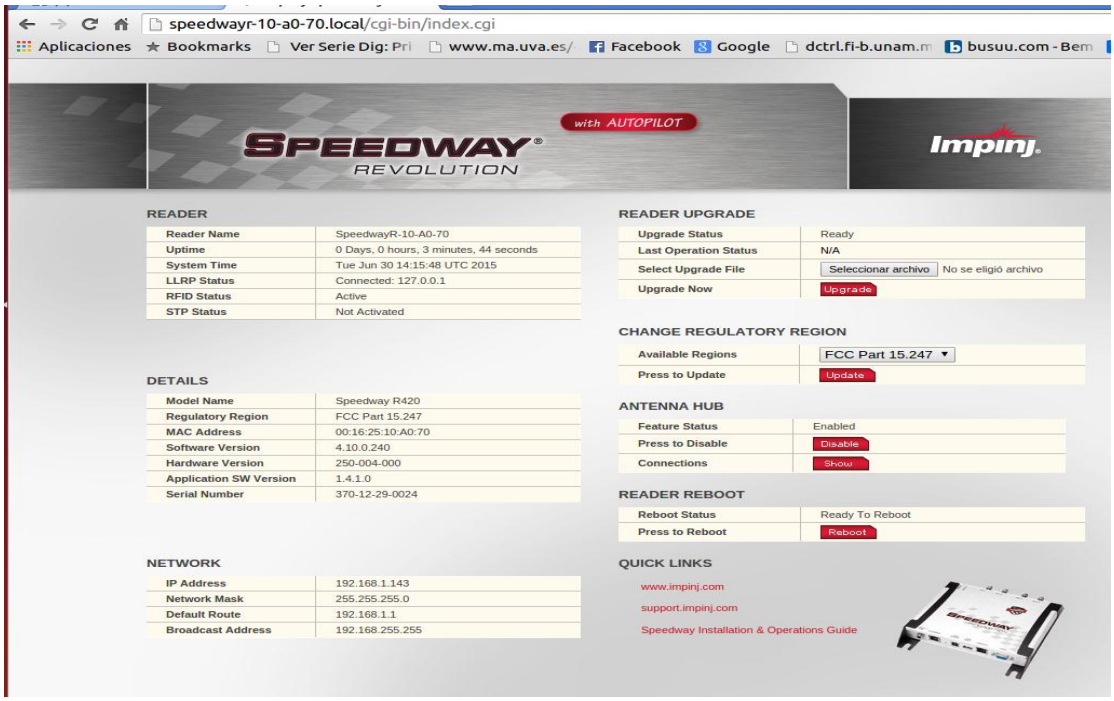
6.2.2. Configuración del Reader. Una vez realizada la conexión del Reader al computador, este debe ser configurado para empezar a funcionar. En este proceso el Reader cuenta con un archivo de configuración SpeedwayConnect_1.0.upg desarrollado por IMPINJ. Para ejecutar el archivo se debe cumplir con una serie de pasos, que se describen a continuación:

- En el navegador de internet se digita speedwayr-10-A0-70.local, lo cual nos pedirá un usuario y una contraseña.

- Aparecerá la plataforma de Speedway Connect como se ver en la Figura 8, en donde se cargara el archivo. En la sección READER UPGRADE, se busca en el computador el archivo SpeedwayConnect_1.0.upg

- Luego de cargar el archivo, se da clic en el botón reboot, asegurando de esta forma que la configuración sea realizada y que el Reader funcione correctamente.

Figura 8. Plataforma Speedway Connect



Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [14].

6.2.3. Gestores de bases de datos (SGDB). PostgreSQL. Un gestor de bases de datos es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos, algunos son pagos, otros son libres, a continuación en la Tabla 1 se mostraran los gestores de bases de datos más conocidos haciendo una comparación de los servicios que brindan, sus ventajas y desventajas [18].

Tabla 1. Comparación entre gestores de bases de datos.

SGDB	Características	Ventajas	Desventajas
Oracle	Es una herramienta de administración gráfica que es intuitiva y cómoda de utilizar. Ayuda a analizar datos y efectuar recomendaciones para mejorar el rendimiento del manejo de datos que se encuentran almacenados.	Es el motor de bases de datos relacional y es el más utilizado a nivel mundial. Tiene un excelente soporte.	Es un servicio pago.

	Apoya en el diseño y optimización de modelos de datos. Apoya en la definición de estándares de diseño y nomenclatura de objetos.	Es la base de datos con más orientación a Internet.	
MySQL server.	Uso personalizado y portabilidad. Funciona en diferentes plataformas. Aprovecha la potencia de los sistemas multiprocesador, gracias al funcionamiento de multihilo. Se puede utilizar con varios lenguajes, C, C++, Java, PHP, Python.	Es un gestor de bases de datos gratuito. El servidor de bases de datos relacionales es rápido y fácil de usar. Gran portabilidad en distintos sistemas y plataformas.	Hay limitaciones con lo que se pueda hacer para el soporte de señales emergentes. Tiene una capacidad limitada a un millón de registros, solo se puede utilizar para aplicaciones pequeñas. Tiene poca seguridad.
SQL server	Facilidad de instalación y distribución ya que posee gran variedad administrativas. Puede utilizarse el mismo motor de bases de datos a través de distintas plataformas. Incluye herramientas para extraer y analizar los datos resumidos para el proceso analítico en línea.	Puede ser útil para manejar y obtener datos de la red de redes. Ofrece una potente forma de unir SQL e Internet. Es seguro.	Bloqueo a nivel de página. Utiliza gran cantidad de memoria RAM. Funciona en sistemas operativos Windows. Es un servicio pago.
PostgreSQL	Se destaca por tener una amplia lista de prestaciones que lo hacen capaz de competir con cualquier SGBF comercial. Cuenta con un conjunto de tipos de bases de datos, permitiendo su extensión mediantes operadores definidos por el usuario.	Tiene gran capacidad para almacenamiento de datos, no tiene un límite de registros para el guardado de la información. Es altamente confiable y tiene versión libre. .	El soporte a orientación a objetos es una simple extensión no un soporte completo.

Tomado de [19].

A partir de lo detallado anteriormente y teniendo en cuenta que la estantería inteligente necesita un soporte robusto en la entrega y almacenamiento de datos. Por otra parte, se busca un software libre, por tales motivos se decide utilizar como gestor de bases de datos PostgreSQL.

6.2.4. Lenguaje de Programación Python. Es importante escoger el lenguaje de programación que se adapte a las necesidades de la interfaz gráfica; dependiendo de las aplicaciones a las cuales se quiera someter, un lenguaje u otro puede brindar diferentes librerías y mejoras. Además de recibir los datos y llegar a lo planteado en el desarrollo de la Estantería Inteligente, es necesario un lenguaje que permita tener acceso a una base de datos, en este caso una base de datos PostgreSQL, por tanto debe tener librerías que permita la conexión con este gestor de datos, borrar, seleccionar e insertar información.

En este último aspecto, muchos lenguajes se quedan cortos, ya que están orientados a aplicaciones computacionales y de sistemas operativos. Python es un lenguaje de programación que cada vez está tomando más fuerza entre los desarrolladores, gracias a su practicidad y flexibilidad ante los otros lenguajes.

Para el desarrollo de la Estantería Inteligente propuesta anteriormente, se necesita llevar la información a una base de datos y Python brinda la posibilidad de llevar inmediatamente la información recibida por Socket a una base de datos. Esto genera practicidad al tener todo el proceso de recolección y almacenamiento de la información en la base de datos en un solo Script. Por tal motivo se toma como mejor opción la programación en Python para el uso de TCP/IP Socket en la recolección de los datos del Reader [20].

6.2.5. Servidor Independiente de Plataforma. Los servidores independientes de plataforma hacen parte de la creación y ejecución de lenguajes en la red. Lenguajes como PHP, HTTP, SQL, entre otros; no pueden actuar si no tienen un Servidor de Plataforma que le permita realizar las tareas. Estos servicios pueden activarse de forma gratuita a partir de software libre, como Apache, Mercury, entre otros. Existen servidores que permiten tener todo en una sola aplicación, es decir, servidores como Apache o Mercury para desarrollar aplicaciones web sobre lenguaje PHP y HTML.

XAMPP, el Servidor Independiente de Plataforma escogido, brinda todos los servicios Apache tales como HTTP, PHP, MySQL, Mercury, entre otros además de ser un software libre, su uso es bastante sencillo y conveniente para aplicaciones web [21].

6.2.6. Lenguajes de Interfaz Gráfica. Para el desarrollo de interfaces gráficas, los lenguajes más usados son los lenguajes XML, estos se representan en un formato de etiquetas que describen la forma en que está compuesta la interfaz gráfica, más que un método de programación es un lenguaje de descripción, en donde la persona va desarrollando la aplicación con códigos que dan formato a cada sección o área de trabajo.

En busca de un lenguaje sencillo y fácil de aplicar a la hora de desarrollar la Interfaz Gráfica de la Estantería Inteligente en un lenguaje XML; HTML es un lenguaje de realización de interfaz gráfica libre, es decir que cualquier persona puede utilizar sin necesidad de pagar por él, sin embargo, está abierto a brindar donaciones a las personas que lo crearon.

El lenguaje HTML esta soportado en el desarrollo por referenciación, es decir, para añadir un elemento externo a la página (imagen, vídeo, script, entre otros), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. Las aplicaciones sobre el lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en Linux, el Bloc de notas de Windows, o cualquier otro editor que admita texto sin formato como GNU Emacs; de esta forma es muy fácil crear scripts HTML, y poderlos ejecutar desde un navegador a Internet [22].

Entre las aplicaciones que ofrece HTML es la creación de formularios, un formulario tiene como fin realizar acciones de consulta, borrado, inserción o selección de información en una base de datos, dependiendo de lo que requiera el usuario; esto es una ventaja de HTML, ya que permite utilizar las propiedades de otros lenguajes. El método POST es utilizado para aplicar la información insertada en un formulario HTML, el cual envía a un script PHP lo solicitado por el formulario. En el script PHP se encuentra el código necesario para realizar la conexión a la base de datos, dando así las posibilidades de insertar, borrar, cambiar y consultar los datos [22].

PHP es un lenguaje de programación orientado a objetos, tal vez el más popular entre desarrolladores ya que brinda un sin número de posibilidades y gran flexibilidad entre librerías y otros lenguajes para el avance de aplicaciones a objetos. Entre las muchas opciones que tiene PHP, está la de acceder a bases de datos ya que este permite el uso de lenguaje SQL, facilitando el uso aplicado a formularios HTML, bases de datos, entre otras [23].

7. DISEÑO DE LA ESTANTERIA INTELIGENTE Y PRUEBAS PILOTO

En los capítulos anteriores se planteó el desarrollo de la Estantería Inteligente, desde la descripción de hardware utilizado hasta llegar a la descripción de software necesario para la aplicación. En el presente capitulo se describe la interconexión de los dispositivos hardware de la Estantería Inteligente, luego se detalla el desarrollo de la base de datos y el método de adquisición de información, por último se describe el desarrollo de la interfaz gráfica.

7.1. CONEXIÓN FÍSICA DE LOS DISPOSITIVOS.

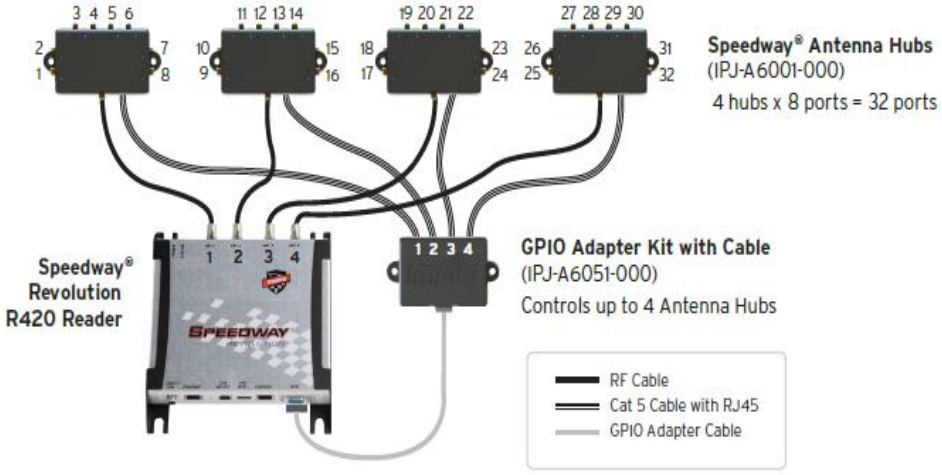
Para la creación de una o varias Estanterías Inteligentes, es necesario una antena por cada cajonera, por ende es necesario más de una antena, a continuación se presentan las soluciones para la interconexión de los dispositivos necesarios en la construcción de la estantería inteligente.

7.1.1. Estantería y Antenas. La Estantería está hecha de madera con el fin de no generar interferencias entre el Tag y la antena producto de la reflexión de ondas electromagnéticas. Cada cajonera de la Estantería tendrá en su base la antena Slimline A7075, sobre la cual se posicionan los productos, para este fin cada producto tiene asignado un Tag con identificación única. Como se puede apreciar en la Figura 9, la Estantería necesita de más antenas de la que puede brindar el Reader, por ende es necesario conectar a la antena Hub, para tener la posibilidad de tener más puertos de conexión.

7.1.2. Conexión Antenas Hub – Reader. Luego de conectar cada una de las antenas al antena Hub, se debe conectar la antena Hub con el Reader utilizando el cable coaxial, para que el Reader energice las antenas planas, también se debe conectar la antena Hub al adaptador GPIO y este al puerto GPIO del Reader. Las conexiones descritas se pueden observar en la Figura 10.

Figura 9. Modelo de Estantería de Madera usada.

Figura 10. Conexión Adaptador GPIO – Antenas Hub - Reader



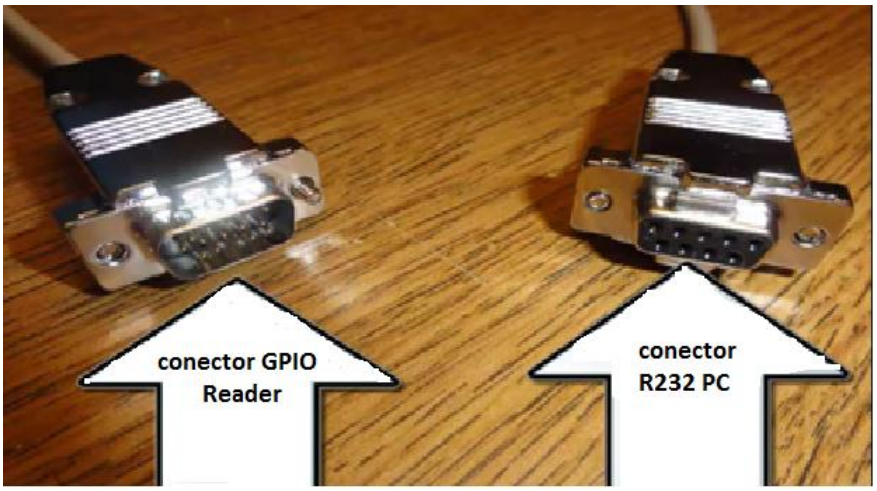
Provisto por IMPINJ. Conexión Antenas Hub con el Reader. Tomado de [14].

7.2. RECOLECCION DE DATOS – PRUEBAS PILOTO.

El Reader Speedway Connect brinda la posibilidad de obtener los datos de las lecturas realizadas por las antenas de diferentes formas. A continuación se presentan las distintas formas de recolección de datos, sus ventajas y desventajas a la hora de utilizar cada uno de los métodos.

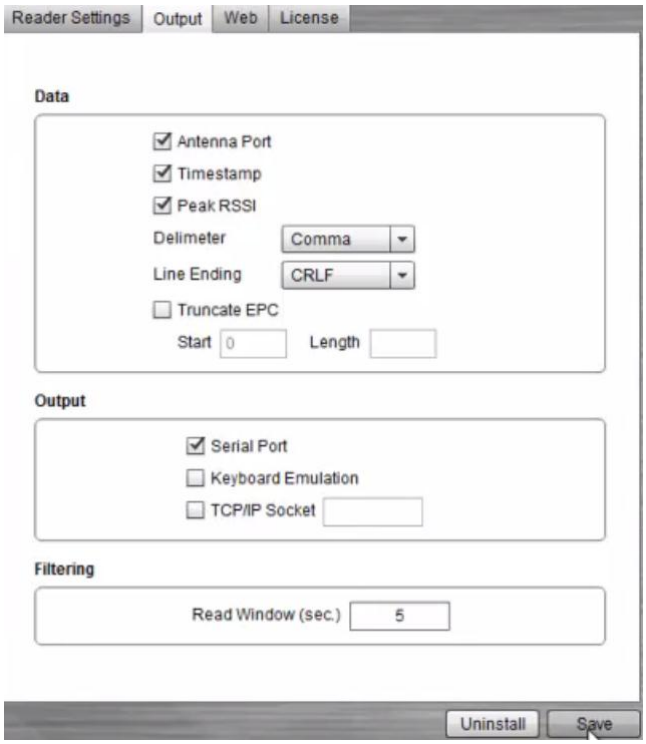
7.2.1. Serial Port. Para obtener los datos a través del puerto Serial del computador, es necesario conectar el Reader a través del puerto GPIO utilizando un cable RS232 como se muestra en la Figura 11.

Figura 11. Cable RS232 para conexión serial Reader-Computador



Una vez realizada la conexión entre el computador y el Reader, se debe configurar la forma como se reciben los datos. Desde un navegador se digita la extensión https://(IP del Reader), en este caso particular se digitó en el navegador https://192.168.1.143, luego en la pestaña Output se seleccionan los datos que se desean obtener de las lecturas, en delimiter se escoge como se requiere que vayan separados los datos, el Line ending genera un salto de línea entre dato y dato, finalmente se selecciona el modo de recepción de datos. Véase Figura 12.

Figura 12. Configuración de la entrega de los datos del Reader.



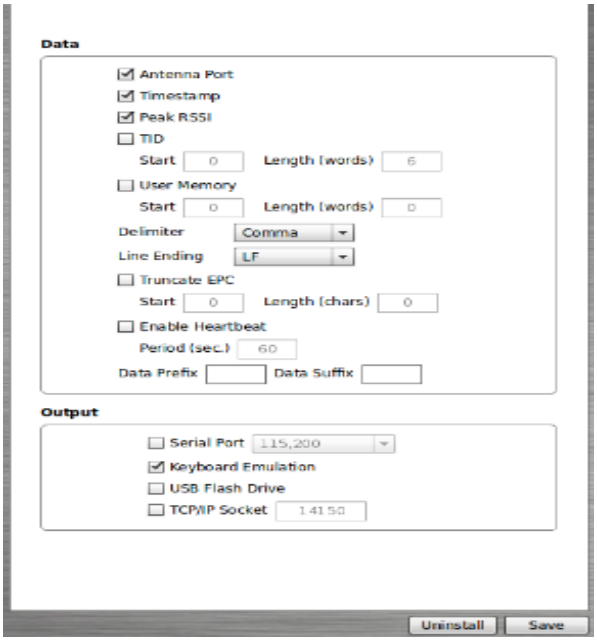
Una de las razones por las que NO se escogió esta forma de recolección de datos, es porque utiliza el puerto GPIO para la conexión serial, esto inhabilita el uso de antenas Hub, y como se estipulo en el apartado 3.1.2. El puerto GPIO es necesario para la configuración de la Estantería Inteligente. Este Método puede ser usado para observar si el Reader está funcionando correctamente.

7.2.2. Keyboard Emulation. Este método permite obtener los datos enviados por las lecturas del Reader por medio del puerto USB tanto del computador como del Reader, una vez realizada la conexión entre el Reader y el computador, se debe configurar el Reader para que la obtención de los datos, se va a la página https.//IP del Reader que en el caso particular es https://192.168.1.143. A continuación se procederá a configurar de la forma en que se observa en la Figura 13.

Luego de realizar la conexión, los datos pueden ser observados desde la terminal Ubuntu y guardarlos en un archivo de texto plano. La gran ventaja de tener los datos en un archivo de texto plano es la facilidad de procesar los datos desde cualquier lenguaje de programación. Para poder ver los datos y guardarlos en un archivo de texto plano, es necesario utilizar el comando nc en la terminal de Ubuntu, en donde se especifica la IP del Reader, el puerto al que está conectado que por defecto es

el 22 y el nombre del archivo de texto en donde se va a guardar, este archivo debe existir y estar en la carpeta de usuario del computador. Para el caso particular se digita el comando: **nc 192.168.1.143 22 < leyendo.txt**

Figura 13. Configuración del Reader para datos por Keyboard Emulation



La gran ventaja de poder tener los datos de información provenientes del Reader en documentos de texto plano puede convertirse en un problema, por una parte tenemos que al utilizar archivos de texto plano se puede utilizar cualquier lenguaje de programación para su manipulación e ingreso a una base de datos, por otra parte tenemos que cuando se realicen lecturas en cantidades masivas de productos o en aplicaciones que se necesite realizar gran número de lecturas se crearan archivos de grandes dimensiones que van a llenar rápidamente la memoria del disco del computador desperdiciando recursos.

7.2.3. HTTP-Post. Este método de adquisición de datos, se basa en la operación de las páginas Web, aprovechando la forma de adquisición y envió de datos entre páginas Web, el Reader puede enviar la información de las lecturas hechas a una página PHP utilizando el método de recepción de datos _POST, del lenguaje de programación PHP. A continuación se explicara de una forma más detallada este método de envío:

- Configurar el Reader para la adquisición de datos por medio de HTTTP – Post, para esto, se dirige a la pestaña Web de la página de configuración del Reader, ahí se habilita el servicio de HTTP – Post.
- Especificar la ruta de la Pagina PHP en donde se enviaran los datos. Ver Figura 14.
- Crear el código de la página PHP donde se va a recibir los datos. Cada dato enviado a la página PHP debe ser guardado en una variable, una vez en la variable se podrá realizar toda la manipulación de la variable en el lenguaje PHP. Ver Figura 15.

Figura 14. Servicios de HTTP - Post.

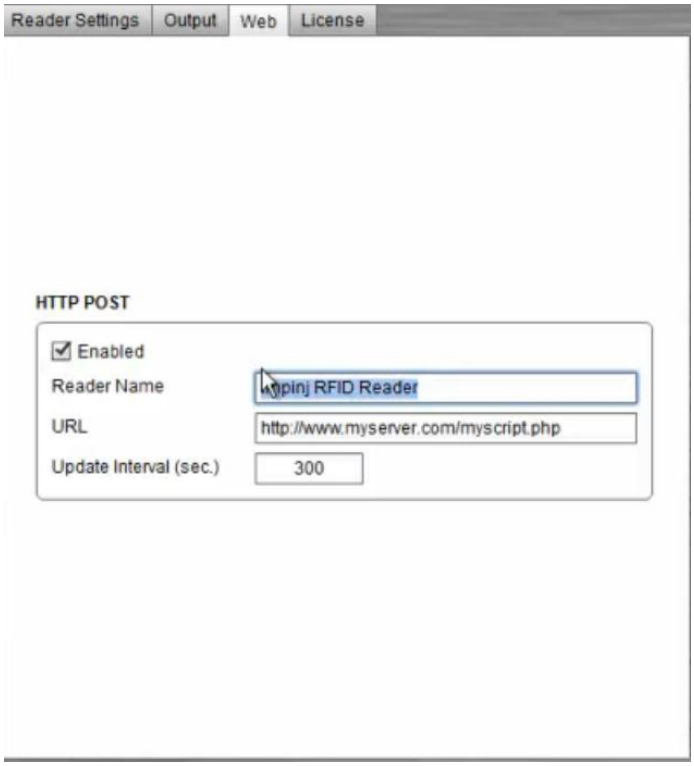
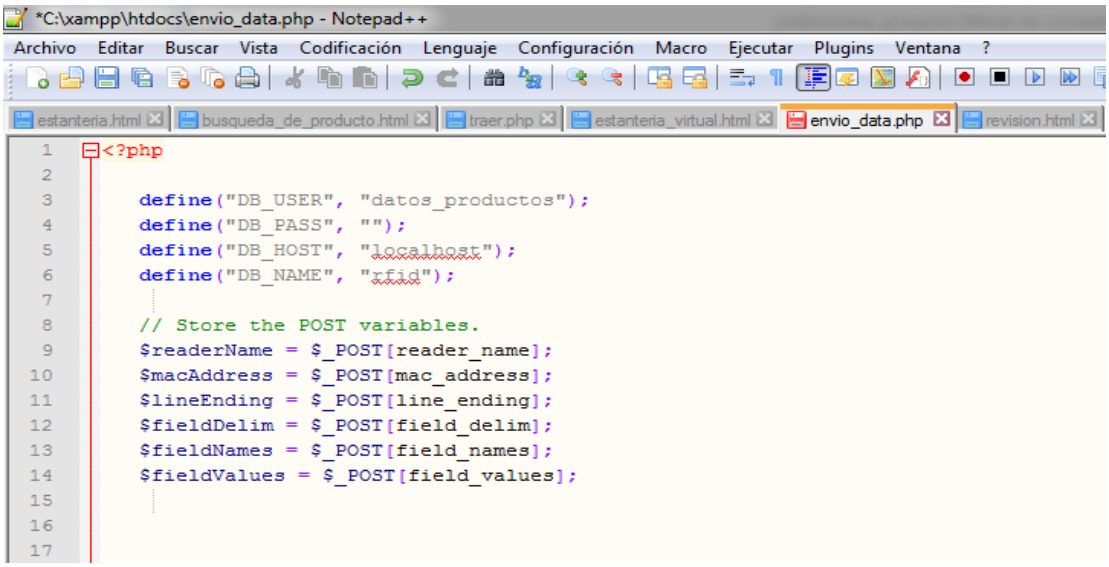


Figura 15. Guardado de los datos en variables PHP mediante la función HTTP – Post.



La gran ventaja de usar el método HTTP – Post es la fácil manipulación de los datos, ya que al tener los datos como variables en el lenguaje de programación PHP, es muy fácil insertar los datos en una base de datos usando lenguaje de gestor de bases de datos SQL. Otra gran ventaja es que no genera archivos inmensos que llenen la memoria del computador ya que PHP realiza un manejo interno de los datos.

Una de las desventajas de usar HTTP – Post es que para que los datos sean insertados en la base de datos, se debe estar refrescando cada cierto tiempo automáticamente la página PHP, esta acción genera deficiencias en tiempo. Otra desventaja es que el servidor solo es capaz de cargar una página PHP un cierto lapso de tiempo, entonces se tendría que ejecutar nuevamente la página PHP. Además para que el Reader envíe los datos Web por HTTP – Post necesita un tiempo para organizar la información y esperar a que la página PHP haga la petición.

Para una aplicación en tiempo real en la cual se manejen grandes cantidades de información y datos, no es una buena opción este método de recolección de datos, ya que genera grandes retrasos a la hora de obtener las lecturas, tampoco permite un trabajo continuo de la Estantería Inteligente.

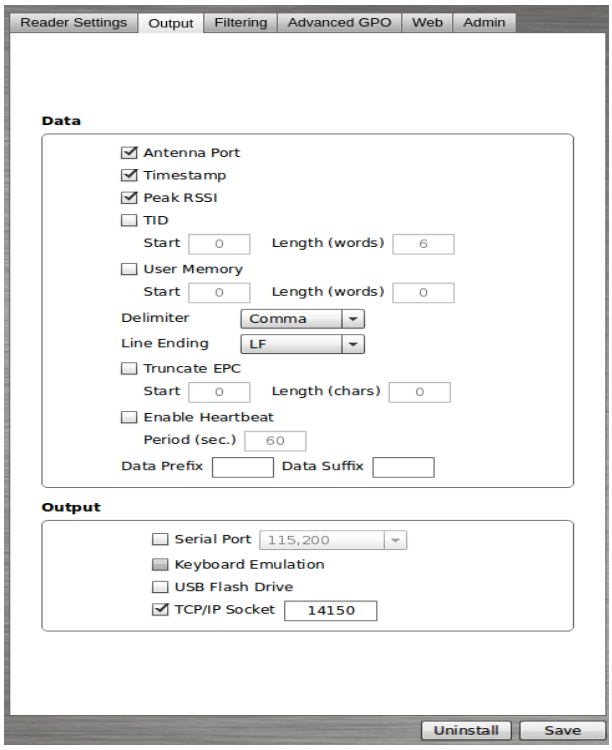
7.2.4. TCP/IP Socket. Este método de adquisición de datos que brinda el Reader Speedway Connect, es el más confiable. La adquisición de los datos se hace a partir de un lenguaje de programación que permita obtener datos a partir del Socket, entre algunos lenguajes de programación esta Ruby, C#, C, Python, cada uno de ellos con posibilidades y librerías diferentes que permiten al usuario manejar de una u otra forma la información envidada desde el Reader.

Python como se ha dicho antes, es un lenguaje de programación que cada vez está tomando más fuerza entre los desarrolladores, gracias a su practicidad y flexibilidad entre otros lenguajes. Python es una muy buena opción para el desarrollo de la Estantería Inteligente propuesta, ya que es capaz de llevar los datos por Socket a una base de datos. Para utilizar TCP/IP Socket, es necesario configurar el Reader desde la página de configuración, seleccionando la información que se necesita y el formato de envío. Ver Figura 16.

Una vez configurado el Reader, se debe crear el Script para la recolección de la información de los productos, Python brinda la posibilidad de trabajar con Socket usando la librería correspondiente, con la función de la librería se puede tener acceso a la lectura del puerto 14150 del Speedway Connect y guardar los datos en variables para su manipulación en Python. Ver Figura 17.

Para la visualización de los datos enviados por el Reader, se puede hacer desde la terminal de Ubuntu, así el usuario podrá observar la evolución de las lecturas provenientes de las antenas, simplemente se coloca como línea de comando la palabra Python seguido del nombre del script.py.

Figura 16. Configuración TCP/IP Socket para la recolección de datos.



La programación en Python para la recolección de datos brinda la ventaja de obtener la las lecturas en tiempo real e inmediatas, además de mayor manejo de los datos adquiridos. Otra gran ventaja que ofrece Python es facilidad de inserción de las lecturas en bases de datos. Por estas razones, TCP/IP Socket es la manera de recolección de datos escogida para la realización de la Estantería Inteligente.

Figura 17. Código Python, Lectura Socket.

```
# Definiendo variables de conexion.

#TCP_IP = 'speedwayr-10-a0-70.local'
TCP_IP = '192.168.1.143'
TCP_PORT = 14150
BUFFER_SIZE = 1024
Path='/home/leyendo.txt'
try:
    print("Abriendo el puerto.")
    # Creacion del socket.
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Definicion de timeout de recepcion y tamano del socket.
    s.settimeout(120)

    # Conexion con el socket.
    s.connect((TCP_IP, TCP_PORT))
    print("Puerto abierto.")

    #s.send(MESSAGE)
    time.sleep(1)

    # Recepcion continua de informacion de consumos, almacenan
    data = s.recv(BUFFER_SIZE)
    Fsave=open(Path,"a")
    Fsave.write(data)
    print data

    # NOTA: La cantidad de datos recibidos supera los 1024 byt
    # trama deber ser recursivo.
    while(data.find("H")==-1):
        data = s.recv(BUFFER_SIZE)
        Fsave.write(data)
        print data
    Fsave.close()
    s.close()
    print("Termino")
```

7.2.5. Conexión Reader – Computador. Para la conexión entre el computador y el Reader, siempre debe existir la conexión por medio de los puertos Ethernet de ambos dispositivos, luego dependiendo de la forma de recolección de datos, se utilizará una conexión física demás, por ejemplo si se usa recolección de datos serial, se necesitara un cable RS232, si se usa recolección de datos por Keyboard, se necesitará un cable USB.

7.3. DISEÑO DE LA BASE DE DATOS.

Las bases de datos son los recursos utilizados para el almacenamiento de información, cualquier aplicación que necesite tener un flujo de variables, imágenes, texto, debe tener una base de datos que soporte toda esta información. Existen gestores de bases de datos que ofrecen al usuario un número determinado de beneficios, servicios y un lenguaje para el manejo de ellas.

7.3.1. Bases de datos. Son bancos de información que contienen datos relativos a diversas temáticas y categorizados de manera distinta, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjuntos.

La necesidad tener archivado todo lo referente a una actividad llevo a las personas a crear bases de datos, hoy en día, los gestores de bases de datos, brindan la posibilidad de interactuar con estas, cada gestor tiene un lenguaje para insertar, seleccionar o eliminar información de una base de datos, pero todas están basados en un lenguaje estándar, SQL.

7.3.2. Diseño de la base de datos. Una vez instalado PostgreSQL, se procede a crear la base de datos en donde se guardaran las lecturas realizadas por el Reader. Una base de datos consta de tablas, en cada tabla se guardara la información deseada, y cada base de datos puede tener cuantas tablas desee, en las tablas se crean los campos que son los datos guardados y enviados desde el Reader por medio de Python, por ejemplo en una base de datos llamada estudiantes, se crea una tabla llamada datos que tendrá los campos código, nombre de estudiante y calificación.

Para la base de datos de la Estantería Inteligente, al ver las Figura 18 y **¡Error! No se encuentra el origen de la referencia.**, se determina el número de los campos y el nombre los cuales serán Antenna Port como primer campo de la tabla, EPC como segundo campo de la tabla, Timestamp como tercer campo de la tabla y como último campo de la tabla Peak RSSI. Ahora se caracteriza el formato de los datos, es decir, si son datos string, texto, entero etc., y el tamaño de estos.

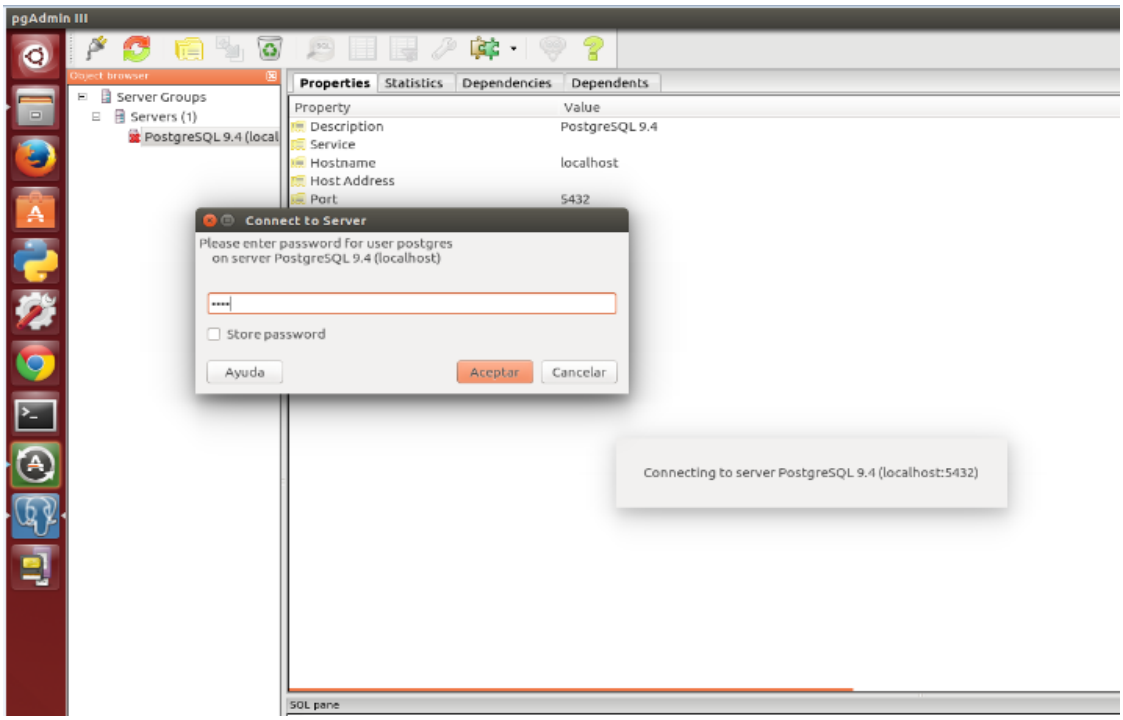
Ya con un vistazo de lo que será la base de datos, se diseña la base de datos en PostgreSQL, en un principio el gestor pedirá a la persona una contraseña y un usuario, por el cual podrá acceder a la información cada vez que necesite. Ver Figura 19. Una vez creado el usuario de la base de datos y de haber dado un nombre a la base de datos, se selecciona Databases, se da clic en Schemas, y por último en Public se pueden crear las tablas en donde se guardara la información.

Para el almacenamiento de los datos se crearon cinco tablas dentro de la base de datos, cuatro que corresponden a cada una de las antenas de la Estantería Inteligente y otra para los productos vendidos, esto con el fin de saber si el producto retirado de la Estantería fue vendido o no.

Así como se planteó anteriormente, entre las opciones de las tablas, está el de crear los campos, Antenna Port formato numérico, EPC formato texto, Timestamp formato numérico y por ultimo Peak RSSI formato numérico. Ver Figura 18.

Figura 18. Campos de las tablas de la Base de Datos.

Figura 19. PostgreSQL, creación de base de datos.



7.3.3. Inserción de los datos en la Base de Datos. Cuando se va a insertar información en la base de datos cada gestor brinda diferentes posibilidades para hacerlo. PostgreSQL permite realizar la inserción de los datos en los campos de las tablas creadas en una base de datos de forma manual, es decir haciendo clic en cada campo se escribe el valor para el campo, y así sucesivamente llenar la tabla. Aunque se puede utilizar el método anteriormente mencionado este es ineficiente, como solución a esta cuestión, PostgreSQL brinda la posibilidad de insertar los datos, eliminar datos, cambiar datos y seleccionar datos en una base de datos desde distintos motores de programación tales como C, C++, PHP y Python, respetando la sintaxis SQL.

Las principales líneas de código para el trabajo con SQL en bases de datos son:

- SELECT * FROM nombre – tabla
- IINSERT INTO ‘nombre de tabla’ (‘campo de tabla1,…’campo de tabla n’) VALUES (‘valor de campo 1’,…,’’valor de campo n)
- DELETE * FROM nombre – tabla

Antes de poder insertar, o eliminar un valor de una base de datos, se debe establecer la conexión entre la base de datos y el Script, para ello cada lenguaje de programación tiene una manera distinta de hacerlo, en Python se maneja con las reglas estipuladas en la librería para ello. A continuación se muestra en la Figura 20 el procedimiento en cómo se realiza la conexión a la base de datos existente y como se insertó el valor en la base de datos de la Figura 18. Una vez se ejecuta el programa en la terminal de Ubuntu, esta imprimirá en ella si la inserción fue realizada o no.

Figura 20. Conexión e inserción de información en la base de datos.

```
import pprint
import psycopg2
import sys

#

try:
    conn=psycopg2.connect("dbname='inventario_total' user='root' password='0000'")
except:
    print "I am unable to connect to the database."


cursor = conn.cursor()


#(`EPC`,`Antenna Port`,`Timestamp`,`Peak RSSI`)   `"+data_string+"`

sql = "INSERT INTO `estanteria`(`Antenna Port`,`EPC`,`Timestamp`,`Peak RSSI`) VALUES('15','0','1435257848715648','-67')"
cursor.execute(sql)
Base.commit()

print "INSERCION REALIZADA"
```

7.4. DISEÑO DE LA INTERFAZ GRAFICA.

La función principal del desarrollo de la interfaz gráfica es brindar al usuario un conjunto de imágenes y objetos con los cuales pueda interactuar de forma más sencilla con la información guardada en la base de datos. Además brindar información oportuna sobre los productos de la Estantería Inteligente y de los productos de la empresa. Para ello existen diferentes lenguajes orientados a la creación de interfaces. Los más usados son los lenguajes XML, estos vienen en un formato de etiquetas que describen la forma en que está compuesta la interfaz gráfica.

Entre los lenguajes XML para la aplicación de interfaces gráficas esta HTML, es un lenguaje sencillo para la creación de interfaces, que permite desarrollar páginas en la Web. Por la sencillez del lenguaje se ha extendido rápidamente entre los desarrolladores de páginas web y gracias a ello, se puede encontrar bastante información sobre HTML. En su versión más reciente, permite interactuar con otros lenguajes de páginas Web para hacer aún más llamativas las interfaces.

La Interfaz Gráfica de la Estantería Inteligente debe contener el espacio donde se puedan realizar consultas sobre los productos, debe tener un espacio donde se pueda ver que está sucediendo en la Estantería Inteligente y además un espacio donde el cliente pueda observar los productos que existen en la empresa con todos los detalles.

Antes de crear una la Interfaz gráfica se debe tener un boceto de lo que se desea, a continuación se mostrará en las Figura 21, Figura 22,Figura 23 y Figura 24 el diseño respectivo.

Figura 21. Opción 1. Búsqueda de productos vendidos.

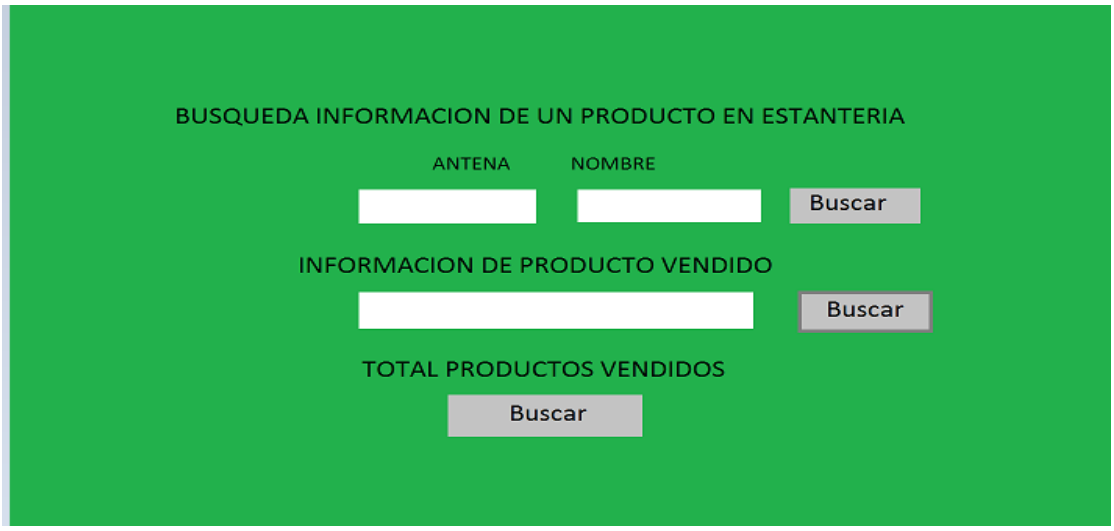
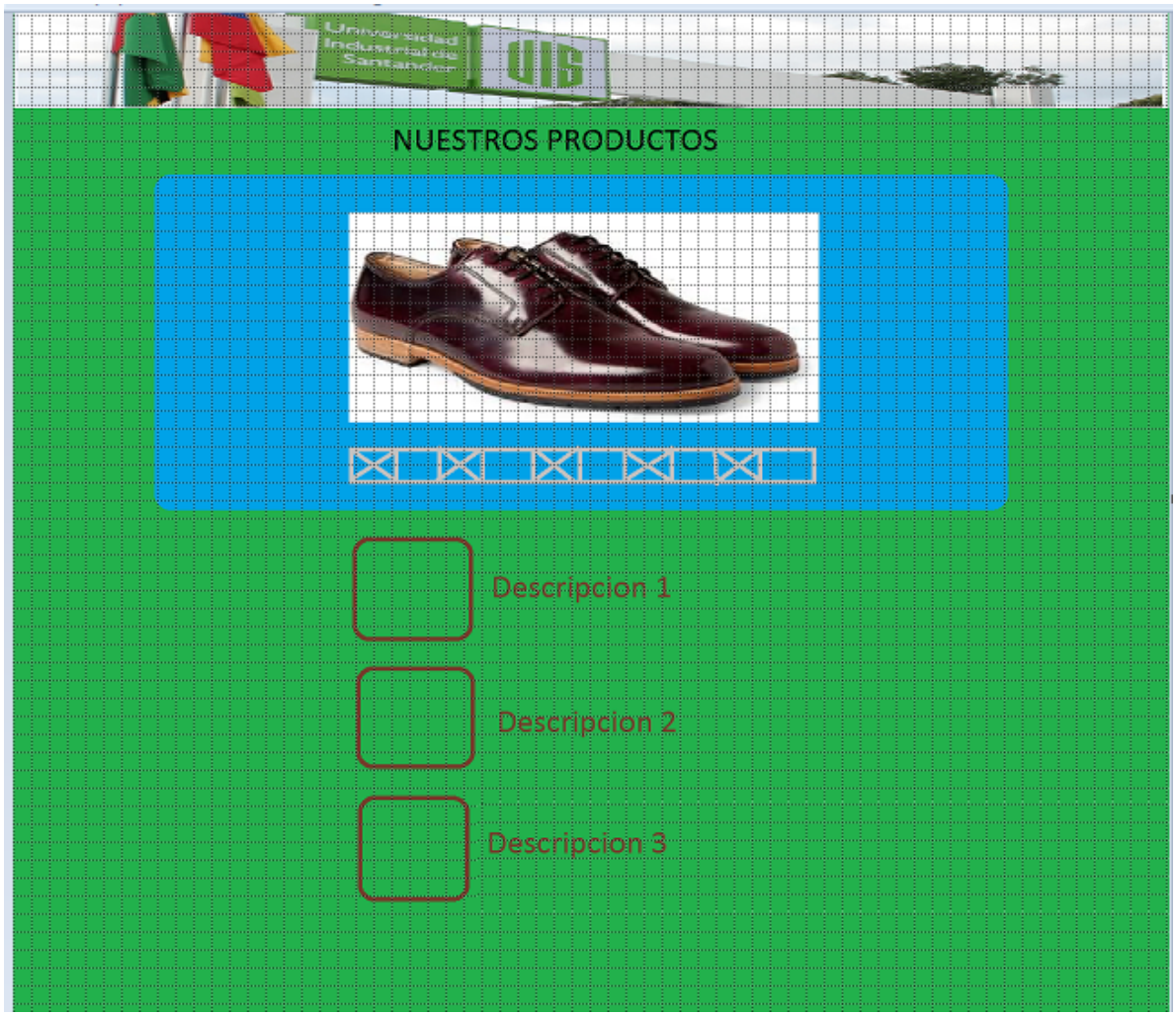


Figura 22. Boceto de página principal de la Interfaz Gráfica

Figura 23. Opción 2. Estantería Virtual.

Figura 24. Opción 3. Productos con descripciones.



8. IMPLEMENTACION Y RESULTADOS

Luego de realizar las pruebas necesarias del funcionamiento de cada una de las partes que conforman a la Estantería Inteligente, en el presente capítulo muestran los resultados y diferentes pruebas que se llevaron a cabo para el completo funcionamiento de todo el conjunto que hizo posible la Estantería Inteligente.

8.1. HARDWARE DE LA ESTANTERIA INTELIGENTE.

Una vez descritos cada uno de los dispositivos provistos para el desarrollo de la Estantería Inteligente en capítulos anteriores, a continuación se muestra en las Figura 25, Figura 26 y Figura 27 como quedo finalmente los componentes esenciales de la ella.

Figura 25. Estantería Inteligente.

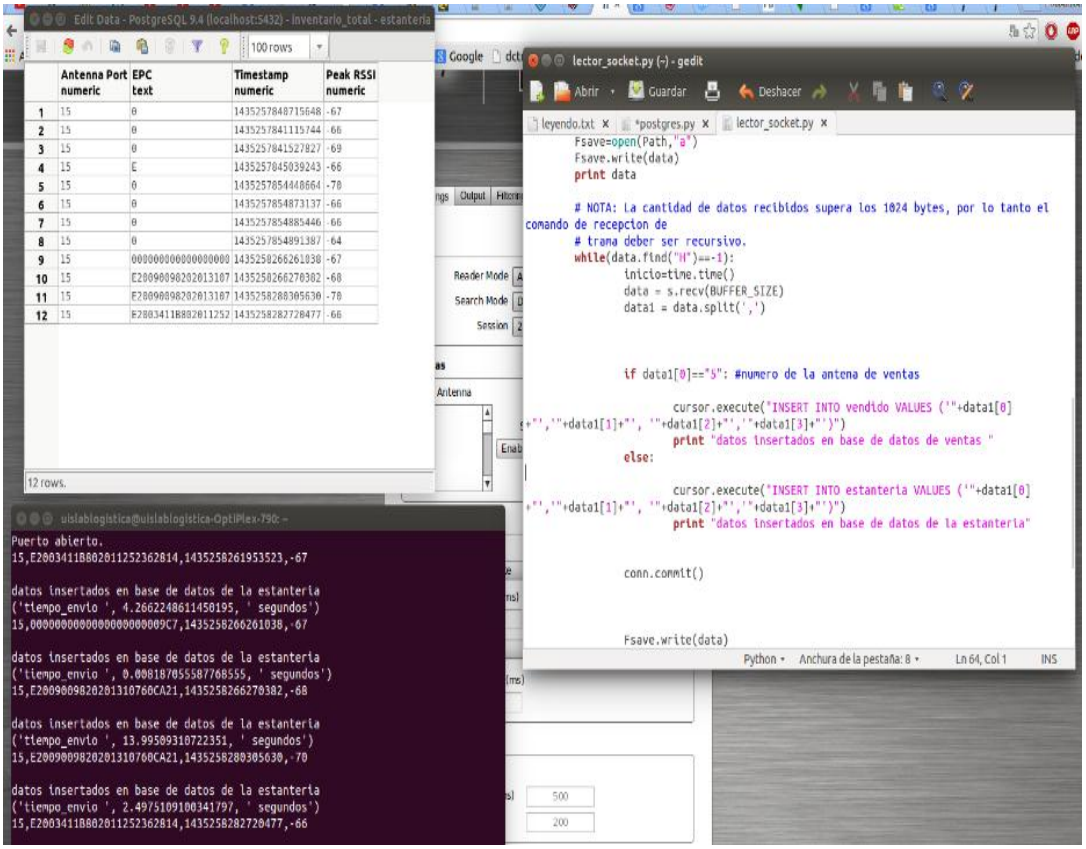
Figura 26. Antena Hub. Estantería Inteligente.

Figura 27. Reader Speedway Revolution. Estantería Inteligente.

8.2. ENVIO DE LECTURAS A LA BASE DE DATOS.

Para el envío de las lecturas a la base de datos se utilizó un código en Python (Ver anexo 2), con el cual se tiene acceso a la base de datos e inmediatamente va insertando los datos provenientes de las antenas conectadas al Reader. El código permite distinguir cual es la antena que esta información e ir colocando las lecturas en la respectiva tabla de la base de datos. Ver Figura 28.

Figura 28. Recolección de lecturas y guardado en la base de datos.



8.3. PRUEBAS DE LECTURAS EN ALTAS DENSIDADES DE PRODUCTOS.

Una vez configurado el Reader para que pueda enviar las lecturas de las antenas por medio de TCP/IP Socket y el código en Python; programa que permite utilizar esta función, luego de tener acceso a la base de datos y a sus diferentes tablas, es importante saber qué modo de operación es el mejor para recibir y guardar los datos. Ver ANEXO G.

El Reader en su configuración previa para la lectura de Tags, brinda al usuario la posibilidad de realizar estas lecturas de forma diferente, es decir, si la persona está trabajando en una aplicación de inventariado, necesitará realizar una sola lectura por producto, a diferencia de alguien que realiza una lectura constante de los productos que este manejando. La posibilidad de manejar los Modos y Sesiones dados en la plataforma de IMPINJ viene dada en el driver del Reader, en donde con conocimiento previo o pruebas de laboratorio se puede definir cuál es la mejor manera para recibir los datos.

A continuación se explicara un poco de cada uno de los ítems anteriormente mencionados y las pruebas realizadas en el laboratorio, esto con el fin de escoger la forma óptima de funcionamiento y caracterización de la Estantería Inteligente.

8.3.1. Ancho de banda del Router. Antes de enviar información hacia el computador, no se puede suponer que el Router pueda recibir y enviar toda la información que el Reader obtenga de las lecturas. En el desarrollo de la Estantería Inteligente, se hizo uso de un Router CISCO Linksys E900, el cual como se puede ver en la Tabla 2 ancho de banda en envió de datos puede llegar a 100 Mbps, limitando así la capacidad de canal, es decir, se debe tener en cuenta que no se puede saturar la Estantería Inteligente con un número infinito de Tags, es decir, de productos.

Tabla 2. Ficha Técnica del Router.

Red	
Estándares de red	IEEE 802.11n, IEEE 802.3, IEEE 802.3u
Seguridad	
Algoritmos de seguridad soportados	WPA2
Características de LAN inalámbrico	
Wi-Fi estándars	802.11n
Wifi	✔
WLAN velocidad de transferencia de datos, soportada	300 Mbit/s
Características de DSL	
DSL conexión	✘
Características de administración	
Botón de restaurar	✔
Características de LAN Ethernet	
Tecnología de cableado	10/100BASE-T(X)
Ethernet	✔
Ethernet LAN, velocidad de transferencia de datos	10, 100 Mbit/s
Contenido del embalaje	
Adaptador AC incluido	✔
Cables incluidos	LAN (RJ-45)
Manual de usuario	✔
Guía de configuración rápida	✔
Antena	
Tipo de antena	Interno

Provisto por CISCO. Tomado de [13].

8.3.2. Información del Tag. Un Tag brinda la posibilidad de guardar información pertinente al producto en el campo EPC, este campo permite almacenar 96 bits empaquetados en números hexadecimales, pero este dato no es la única información proveniente de las lecturas, cuando una antena detecta un Tag, el Reader envía otra información como es Timestamp, Antenna Port y Peak RSSI. Véase Figura 16. Cada uno de estos campos tiene un tamaño en bits específico, y en total por cada lectura en el canal viajan 47 bits de información.

8.3.3. Modos de Lectura. Los modos de lectura son los protocolos anticolisiones que utiliza el Reader Speedway Connect, estos protocolos han sido creados para evitar los problemas que surgen cuando varias etiquetas de RFID pasivo entran en contacto con el área de cobertura del lector. En estos casos, la distancia entre etiqueta y lector admitida por el sistema será mayor cuanto menor sea el consumo de energía del protocolo anti-colisión.

. El Reader Speedway Revolution ofrece los siguientes modos de lectura:

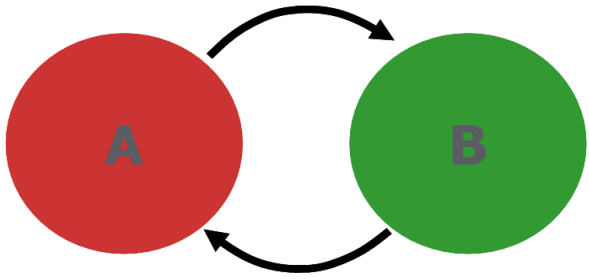
- Auto-Set Dense Reader
- Auto-Set Single Reader
- Dense Reader M4
- Dense Reader M8
- Hybrid
- Max-Miller
- Max Throughput

Para mayor información acerca los modos de lectura véase ANEXO G.

8.3.4. Modos de Búsqueda. Los modos de búsqueda corresponden a la forma en que la antena va inspeccionar el dato y como el Reader va a responder a una eventual energización del Tag al estar próximo a la antena. Existen tres modos de búsqueda, los cuales son Single Target, Dual Target y Single Target with Supression, los cuales se ocupan del modo en que el Tag va a ser energizado, des-energizado y representado como dato frente al Reader. Ver F

8.3.5. y Figura 30F

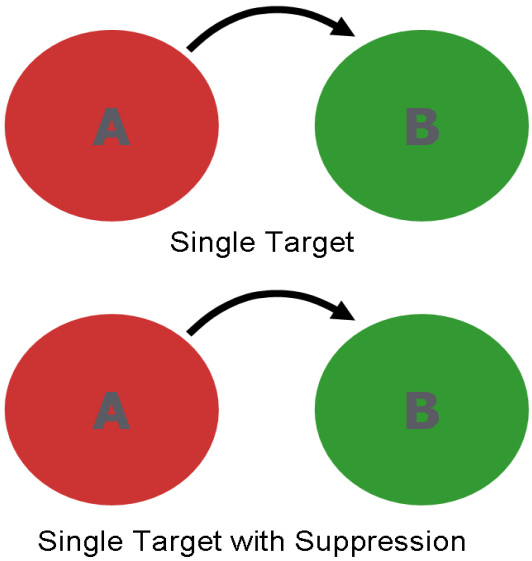
Figura 29. Modo Dual Target.



Dual Target

Provisto por IMPINJ. Dual Target. Tomado de [14]

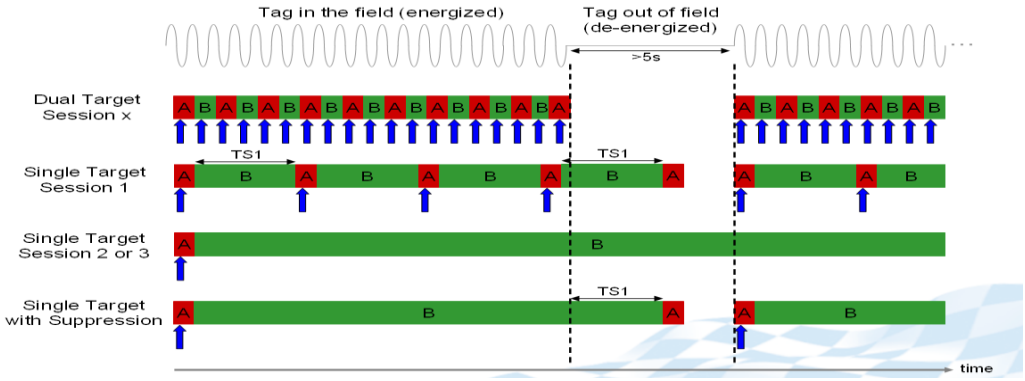
Figura 30. Modo Single Target y Single Target with Suppression.



Provisto por IMPINJ. Single Target and Single Target with Suppression. Tomado de [14]

8.3.6. Sesiones. Cuando se escoge un modo de lectura, se debe saber en qué forma y tiempo este funciona, es decir, una vez energizado un Tag, cuanto le toma al Reader reconocer que este está fuera del rango de la antena, y según la aplicación se puede escoger que sesión utilizar. A continuación la Figura 31 muestra su funcionamiento y en que modos de búsqueda se utiliza.

Figura 31. Sesiones y sus modos.



Provisto por IMPINJ. Sesiones y modos. Tomado de [14].

8.3.7. Pruebas de Laboratorio. Las pruebas de lecturas en el laboratorio fueron hechas para verificar cuál de los modos y sesiones era el más apropiado para la aplicación de la Estantería Inteligente. En busca de obtener una caracterización más detallada del sistema, se realizaron pruebas para determinar el máximo número de lecturas que puede realizar el Reader y a que potencia deben trabajar las antenas.

8.3.7.1. Modos de Lectura. Lo primero a definir es el modo de lectura que se debe utilizar en la Estantería Inteligente, a continuación se mostrarán las diferentes lecturas realizadas en cada uno de los modos provistos por el Reader Speedway Connect, para seleccionar el mejor para la aplicación. Para cada modo se tomó el caso más extremo en la recolección de lecturas, es decir, sin filtro de tiempo.

En la Tabla 3, se puede observar los resultados de utilizar cada uno de los modos de lectura del Reader, para ello se realizaron 500 lecturas que sirvieron como muestra.

Tabla 3. Lecturas y Errores de los Diferentes Modos de Lectura.

Modo de lectura	Número de lecturas	Número de lecturas erróneas	Error
Max Throughput.	500	50	10%
Hibrido	500	49	9,8%
Modo Max Miller	500	39	7,8%
Modo Dense Reader M4	500	33	6,6%
Modo Dense Reader M8	500	33	6,6%
Autoset dense Reader	500	32	6,4%
Auto-Set Single reader	500	27	5,4%

Finalmente, luego de realizadas las pruebas y en base a los resultados, se decidió que el mejor modo de lectura para la aplicación de Estantería Inteligente es Auto-Set Single Dense.

8.3.7.2. Modos de Búsqueda y Sesiones Apropiado. Una vez determinado el modo de lectura a utilizar, a continuación se mostraran las pruebas realizadas de lecturas hechas en la Estantería Inteligente cuando el Reader envía grandes cantidades de información; dichas lecturas fueron tomadas en los diferentes modos y sesiones.

Como se mencionó en la sección 4.3.4, las sesiones y modo de búsqueda permiten realizar la lectura de maneras diferentes, para el caso de la Estantería Inteligente se utilizó Single Target with Suppression que a su vez es una mejora del Single Target; para la aplicación funciona perfectamente ya que se necesitan lecturas del Tag periódicamente sin saturar la base de datos con información redundante a diferencia del modo Dual Target. Ver Figura 31.

8.3.7.3. Capacidad de Canal y Máximo Número de Lecturas. La capacidad de un canal es la velocidad a la que se pueden transmitir los datos por un medio de comunicación, desde este concepto, se centra el siguiente planteamiento para determinar el número máximo que podría soportar el sistema.

Cuando la antena realiza una lectura, como se estipuló anteriormente el Reader envía un paquete de información de 116 bits, estos empaquetados en forma hexadecimal, para determinar cuántas lecturas máximo puede realizar el sistema, se debe saber cuánto tarda un dato en llegar a la base de datos. Al realizar una prueba con una muestra de 500, el promedio de tiempos concluye que un dato tarda 0.01 segundo, es decir, con un filtro de 1 segundo, el sistema máximo podrá realizar la lectura de 100 Tags,

Los tiempos de envío y recolección de datos pueden ser mejorados realizando modificaciones en el código de recolección de lecturas, ya que el código es ejecutado línea por línea, el uso eficiente de los comandos, el acceso a la base de datos y la forma en que se realiza el almacenamiento de la información puede reducir la carga de procesamiento de los paquetes enviados por el Reader.

8.3.7.4. Lectura de Potencia de Antenas y Filtros de Cajonera. Es necesario saber de dónde proviene la información envía que el Reader de las antenas, por suerte cuando se realiza una lectura, el dato trae incorporado el puerto o antena de donde proviene, haciendo posible separar de esta manera la información de cada cajonera. Para lograr clasificar la información de una antena u otra en las cajoneras de la Estantería Inteligente. Como se denota en la

Figura 32, en el código de adquisición de datos trae consigo un segmento de clasifica de acuerdo al puerto de la antena la información en tablas diferentes de la base de datos utilizada en la aplicación.

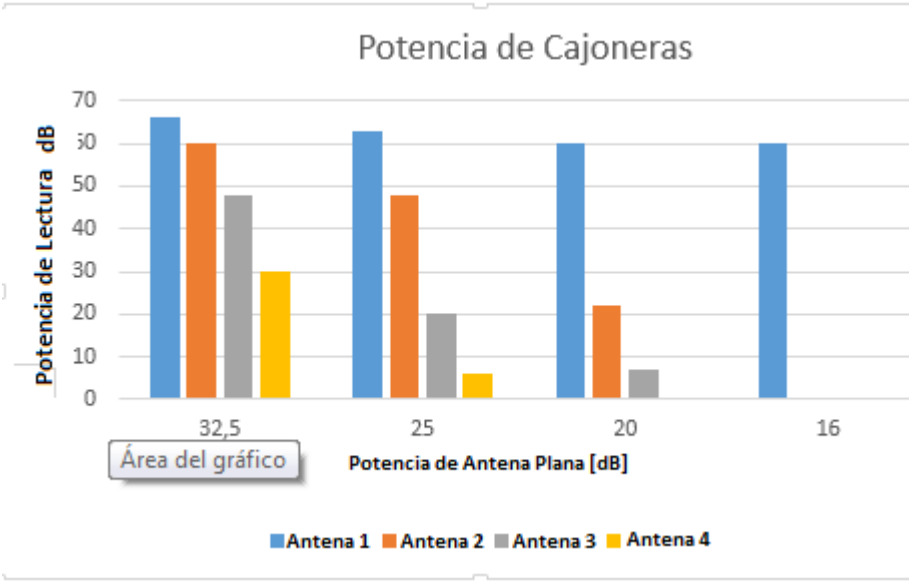
Para determinar la potencia a utilizar en la lectura de Tags en la Estantería Inteligente, se colocó una antena en la primera cajonera a diferentes densidades de potencia para determinar que antenas son capaces de leerla, estos valores fueron 32.5 [dB], 25 [dB], 20 [dB] y 16 [dB] lo cual se puede observar en la Figura 33, luego se obtuvo un valor de potencia a la cual solo la primera antena puede realizar la lectura del Tag.

Figura 32. Fragmento de Código de Adquisición de Datos. Filtro.

```
        if data1[0]=="5": #numero de la antena de ventas

            cursor.execute("INSERT INTO vendido VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
            print "datos insertados en base de datos de ventas "

        elif data1[0]=="1":

            cursor.execute("INSERT INTO antena1 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
            print "datos insertados en base de datos de la estanteria"

        elif data1[0]=="2":

            cursor.execute("INSERT INTO antena2 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
            print "datos insertados en base de datos de la estanteria"

        elif data1[0]=="3":

            cursor.execute("INSERT INTO antena3 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
            print "datos insertados en base de datos de la estanteria"

        elif data1[0]=="4":

            cursor.execute("INSERT INTO antena4 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
            print "datos insertados en base de datos de la estanteria"


        conn.commit()


        Fsave.write(data)
```

Finalmente, luego de todas las pruebas realizadas en la Estantería Inteligente, se concluye que la configuración del Reader es vital para que esta funcione de buena manera según la aplicación que se necesite, para el fin mencionado en este proyecto, el modo de lectura a utilizar es Auto-set Single Dense, el protocolo anticolisiones es Single Target with Suppression y por último la potencia a utilizar para lecturas confiables debe ser 16 dB se muestra en la Figura 33.

Figura 33. Potencia de Tags en las Cajoneras.



8.4. INTERFAZ GRÁFICA.

Como se estipuló en el capítulo 3, la interfaz gráfica consta de una página principal en la cual hay descripciones fundamentales de funcionamiento, luego se describe las opciones que maneja la interfaz gráfica; Opciones de Producto, Estantería Virtual, Nuestros Productos. Para ello se empleó lenguaje HTML (Ver Anexo 3), a continuación se presentara en la Figura 34 el resultado de la página principal de la interfaz.

Figura 34. Interfaz Gráfica. Página Principal.

En parte central de la página principal de la interfaz gráfica, se observa las opciones a las que el usuario puede acceder, cada una de estas opciones se pueden apreciar en las Figura 35, Figura 36, Figura 37 y Figura 38. La forma en cómo fue diseñada cada una de las paginas mostradas se puede encontrar en el capítulo 3 en donde se muestran bocetos de como tendría que quedar. Sus códigos HTML y PHP realizados para la creación de éstas se pueden encontrar en los

ANEXO **D**, ANEXO E y ANEXO F respectivamente.

Figura 35. Opción 1. Opciones de Producto.

Figura 36. Opción 2. Estantería Virtual.

Figura 37. Opción 3. Nuestros Productos Visualizador.

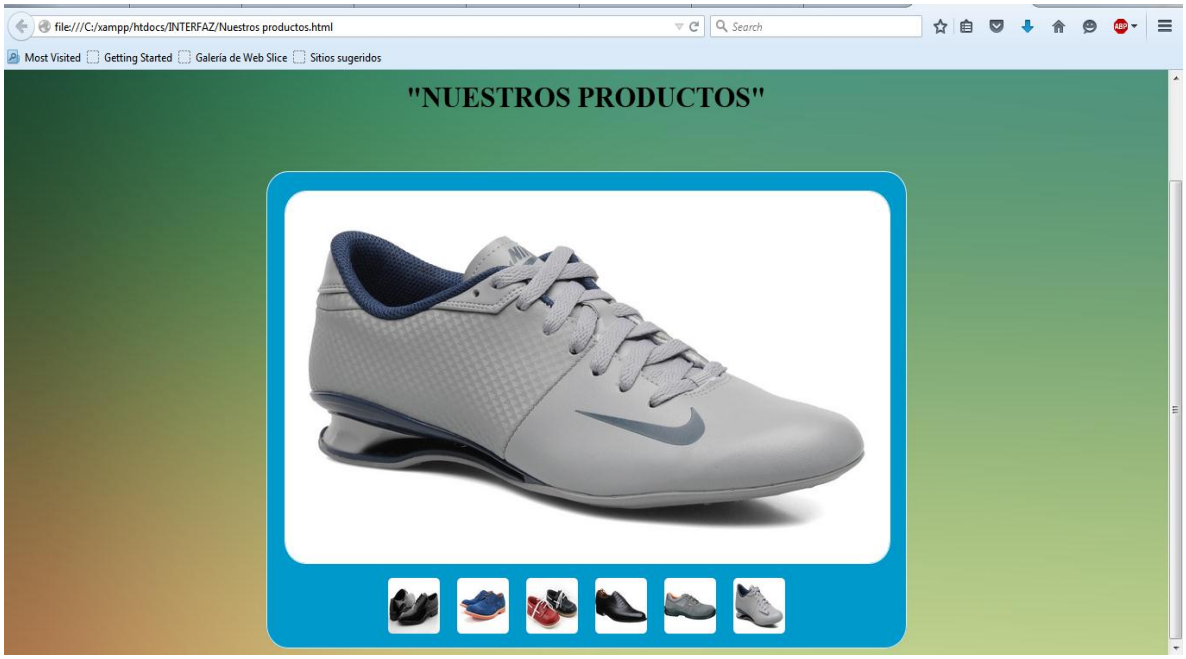
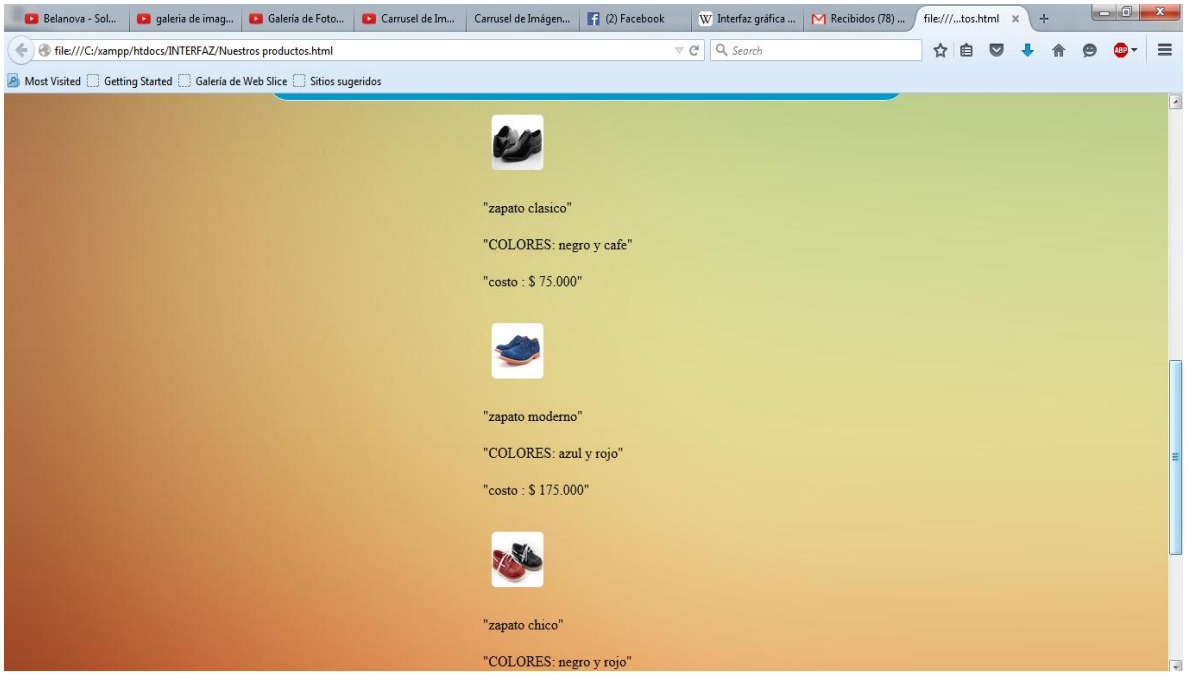


Figura 38. Opción 3. Nuestros Productor. Detalles.



CONCLUSIONES Y OBSERVACIONES

Es importante conocer y aprender sobre la tecnología y los dispositivos que se vayan a utilizar en una aplicación, el conocimiento previo de los implementos usados en la Estantería Inteligente fue de gran ayuda a la hora de diseñar y estudiar las posibilidades que brinda el Reader IMPINJ para la recolección de las lecturas. Se debe tener en cuenta que los archivos de configuración del Reader estén en perfecto estado, ya que si este presenta errores, éste no funcionará de manera correcta presentando inconvenientes a la hora de enviar los datos a algún terminal.

En la base de datos se debe tener en cuenta para que aplicación se va a crear, ya que dependiendo de la necesidad del usuario, esta puede ser creada y configurada para que ejecute acciones automáticamente. El uso de disparadores en una base de datos es un recurso bastante importante y útil, gracias a estos se pueden llevar auditorias de las tablas en las que se trabaja, fechas, inserciones, y otras funciones que junto a la creatividad del diseñador pueden hacer de ésta una gran herramienta.

En el desarrollo de la Interfaz Gráfica, es recomendable que esta esté dirigida a dispositivos móviles, en el caso de hacer uso del lenguaje HTML para aplicaciones en la Web, hacer uso de pixeles para catalogar las dimensiones de las tablas, texto, títulos y demás en la Interfaz puede distorsionar la forma de la información presente en ésta debido que los dispositivos tienen pantallas de diferentes tamaños y resoluciones, para evitar este problema a la hora de dar dimensiones y formato a la Interfaz Gráfica es mejor utilizar porcentajes de pantalla, así sin importar el dispositivo éste se adaptara al tamaño de la pantalla.

Al trabajar con un lenguaje de programación, la persona debe tener en cuenta la aplicación en la cual se va a implementar; la diferencia entre un lenguaje y otro depende de las ventajas y opciones que estos presentan frente a los demás. En el uso del lenguaje de programación Python, este debe tener disponibles las librerías necesarias para que este entienda que tipo de lenguaje y código se está ejecutando, si este no las tiene, descargarlas y habilitarlas; si se trabaja en Ubuntu estas pueden ser adquiridas como software libre desde el Centro de Recursos de Software.

La tecnología de identificación de radiofrecuencia (RFID) tiene un gran potencial de causar impacto en cadenas de abastecimiento, y a pesar que su expectativa es alta aún no ha podido extenderse como método de abastecimiento ni reemplazar el clásico código de barras, por tal razón es necesario realizar mayor investigación e inversión en áreas que apoyen automatización de los procesos de la industria local para que se encuentre en ésta tecnología una opción a la hora de mejorar sus procesos logísticos. El uso de tecnología igualmente eficiente y más económica es

el verdadero reto y paso siguiente en la implementación de tecnología RFID en su adaptación para las pequeñas empresas puesto que este es el factor principal que lleva a los dueño a pensar que aplicarlo en sus locales es costoso y se cierran a las posibilidades y ventajas que ésta tecnología brinda.

Es verdad que esta tecnología permite realizar muchas lecturas por segundo, pero está limitado al procesamiento de estos, ya se ha demostrado que según el modo en que el Reader funcione y reciba los datos el procesamiento de estos será diferente y el sistema también responderá o más rápido o más lento, por eso es importante caracterizar muy bien el sistema, observar que opciones le brinda la tecnología de los dispositivos que se usan y determinar si pueden desarrollar de buena forma la aplicación para que se requiera.

REFERENCIAS BIBLIOGRÁFICAS

[1] B. C. A. S. T. Melanie R. Rieback, «The Evolution of RFID Security,» , *IEEE Pervasive Computing,* vol. 5, nº 1255445, pp. 62-69, January-March 2006.

[2] Z. N. C. A. C. Xianming Qing, «Multi-loop Antenna for High Frequency RFID Smart Shelf Application,» *Antennas and Propagation Society International Symposium IEEE 2007,* vol. 1, nº 978-1-4244-0878-8, pp. 5467-5470, 9-15 June 2007.

[3] J. R. C. C. A. F. Carla R. Medeiros, «RFID Reader Antennas for Tag Detection in Slef-Confined Volumes at UHF,» *Antennas and Propagation Magazine,* vol. 53, nº 12108508, pp. 39-50, April 2011.

[4] Z. N. C. Xianming Qing, «Proximity Effects of Metallic Enviroments on High Frequency RFID Reader Antenna: Study and Applications,» *Antennas and Propagation,* vol. 55, nº 9674031, pp. 3105-311, Nov 2007.

[5] J. S. D. K. Weixin Wang, «Complex Event Processing in EPC Sensor Network Middleware for Both RFID and WSN,» *Object Oriented Real-Time Distributed Computing(ISORC),* nº 997773, pp. 165-169, 5-7 May 2008.

[6] J.-S. K. J.-H. B. G. C. J.-S. C. Wonkyu Choi, «Near-field Antenna for RFID Smart Shelf in UHF,» *Antennas and Propagation Society International Symposium,* nº 10800974, pp. 1-4, 1-5 June "009.

[7] S. M. R. M. S. I. Nemai Chandra Karmakar, «Development of Smart Antenna for RFID Reader,» *RFID, 2008 IEEE,* nº 10039565, pp. 65-73, 17 April 2088.

[8] H. L. D. W. E. R. E. Jae Sung Choi, «Passive UHF RFID-Based Localization Using Detection of Tag Interference on Smart Shelf,» *Transactions on Systems, Man and Cybernetics* , vol. 42, nº 12537542, pp. 268-275, 2 Mar 2012.

[9] Z. N. C. Xianming Qing, «Characteristics of a Metal-Backed Loop Antenna and its Application to a High-Frequency RFID Smart Shelf,» *Antennas and Propagation Magazine,* vol. 51, nº 10764634, pp. 26-38, April 2009.

[10] L. M. Katina Michael, «The Pros and Cons of RFID in Supply Chain Management,» *Mobile Business, 2005. ICMB,* nº 8651819, pp. 623-629, 11 July 2005.

[11] C. W. Y. Y. Ankang Ren, «A Robust UHF Near-Field RFID Reader Antenna,» *Transactions on Antennas and Progragation,* vol. 60, nº 12639403, pp. 1690-1697, 4 April "012.

[12] J. G. J. M. Benjamin B. Bederson, «Toolkit Design for Interactive Structured Graphics,» *Transactions on Software Engineering,* vol. 30, nº 8113541, pp. 535-546, August 2004.

[13] T. L. S. H. J. S. Christ of Borhovd, «Integrating Smart Items with Business Processes,» *Proceedings of the #8th International Conference on System Sciences,* vol. 38, p. 227c, 6-jun-2005.

[14] IMPIN, «IMPINJ Platform,» IMPINJ, 3 Abril 2002-20015. [En línea]. Available: http://www.impinj.com/products/. [Último acceso: 15 Enero 2015].

[15] IMPINJ, «LTK Programmers Guide,» IMPINJ, 3 Abril 2002-2015. [En línea]. Available: http://www.impinj.com. [Último acceso: 25 Enero 2015].

[16] IMPINJ, «Antena Hub,» IMPINJ, 3 Abril 2002-2015. [En línea]. Available: http://www.impinj.com. [Último acceso: 2 Febrero 2015].

[17] D. D. LUCA, HTML5, México Distrito Federal: La Tora, 2011.

[18] Python, «Python.,» Python, 22 Septiembre 2005-2015. [En línea]. Available: https://www.python.org/. [Último acceso: 2015 Abril 4].

[19] RFID, Perfect, «Antenas,» Perfect RFID, 4 Junio 2004-2015. [En línea]. Available: http://www.perfectrfid.com/shelving-cabinet-antennas/slimline-a7075-linear-polarised-uhf-rfid-shelf-antenna/85994.. [Último acceso: Febrero 10 2015].

[20] P. R. y. M. &. V. Scholl, Introduction to Spatial Databases: Applications to GIS, New York: Academic Press, 2001.

[21] D. Web., «Desarrollo Web,» Web Libre, 14 Junio 2007-2015. [En línea]. Available: 2015http://desarrollowebydesarrolloweb.blogspot.com/2015/02/tabla-comparativa-de-los-sistemas.html. [Último acceso: 2015 Marzo 30].

[22] XAMPP, «XAMPP,» Apache, 22 Diciembre 2001-2015. [En línea]. Available: https://www.apachefriends.org/es/index.html. [Último acceso: 2015 Marzo 27].

[23] J. A. RODRÍGUEZ, Tutorial MySQL en PHP, México Distrito Federal: La Tora, 2011.

BIBLIOGRAFÍA

B. C. A. S. T. Melanie R. Rieback. The Evolution of RFID Security, En: IEEE Pervasive Computing, vol. 5, nº 1255445, pp. 62-69, January-March 2006.

C. W. Y. Y. Ankang Ren, A Robust. UHF Near-Field RFID Reader Antenna. En: Transactions on Antennas and Progragation, vol. 60, nº 12639403, pp. 1690-1697, 4 April "012.

D. D. LUCA, HTML5, México Distrito Federal: La Tora, 2011.

H. L. D. W. E. R. E. Jae Sung Choi. Passive UHF RFID-Based Localization Using Detection of Tag Interference on Smart Shelf. En: Transactions on Systems, Man and Cybernetics , vol. 42, nº 12537542, pp. 268-275, 2 Mar 2012.

IMPIN, «IMPINJ Platform,» IMPINJ, 3 Abril 2002-20015. [En línea]. Available: http://www.impinj.com/products/. [Último acceso: 15 Enero 2015].

IMPINJ, «Antena Hub,» IMPINJ, 3 Abril 2002-2015. [En línea]. Available: http://www.impinj.com. [Último acceso: 2 Febrero 2015].

IMPINJ, «LTK Programmers Guide,» IMPINJ, 3 Abril 2002-2015. [En línea]. Available: http://www.impinj.com. [Último acceso: 25 Enero 2015].

J. A. RODRÍGUEZ, Tutorial MySQL en PHP, México Distrito Federal: La Tora, 2011.

J. G. J. M. Benjamin B. Bederson. Toolkit Design for Interactive Structured Graphics. En: Transactions on Software Engineering, vol. 30, nº 8113541, pp. 535-546, August 2004.

J. R. C. C. A. F. Carla R. Medeiros. RFID Reader Antennas for Tag Detection in Slef-Confined Volumes at UHF. En: Antennas and Propagation Magazine, vol. 53, nº 12108508, pp. 39-50, April 2011.

J. S. D. K. Weixin Wang. Complex Event Processing in EPC Sensor Network Middleware for Both RFID and WSN. En: Object Oriented Real-Time Distributed Computing(ISORC), nº 997773, pp. 165-169, 5-7 May 2008.

J.-S. K. J.-H. B. G. C. J.-S. C. Wonkyu Choi. Near-field Antenna for RFID Smart Shelf in UHF. En: Antennas and Propagation Society International Symposium, nº 10800974, pp. 1-4, 1-5 June "009.

L. M. Katina Michael. The Pros and Cons of RFID in Supply Chain Management. En: Mobile Business, 2005. ICMB, nº 8651819, pp. 623-629, 11 July 2005.

P. R. y. M. &. V. Scholl, Introduction to Spatial Databases: Applications to GIS, New York: Academic Press, 2001.

Python, «Python.,» Python, 22 Septiembre 2005-2015. [En línea]. Available: https://www.python.org/. [Último acceso: 2015 Abril 4].

RFID, Perfect, «Antenas,» Perfect RFID, 4 Junio 2004-2015. [En línea]. Available: http://www.perfectrfid.com/shelving-cabinet-antennas/slimline-a7075-linear-polarised-uhf-rfid-shelf-antenna/85994.. [Último acceso: Febrero 10 2015].

T. L. S. H. J. S. Christ of Borhovd. Integrating Smart Items with Business Processes. En: Proceedings of the #8th International Conference on System Sciences, vol. 38, p. 227c, 6-jun-2005.

XAMPP, «XAMPP,» Apache, 22 Diciembre 2001-2015. [En línea]. Available: https://www.apachefriends.org/es/index.html. [Último acceso: 2015 Marzo 27].

Z. N. C. A. C. Xianming Qing. Multi-loop Antenna for High Frequency RFID Smart Shelf Application. En: Antennas and Propagation Society International Symposium IEEE 2007, vol. 1, nº 978-1-4244-0878-8, pp. 5467-5470, 9-15 June 2007.

Z. N. C. Xianming Qing. Characteristics of a Metal-Backed Loop Antenna and its Application to a High-Frequency RFID Smart Shelf. En: Antennas and Propagation Magazine, vol. 51, nº 10764634, pp. 26-38, April 2009.

ANEXO A. IMPINJ MULTIREADER SOFTWARE

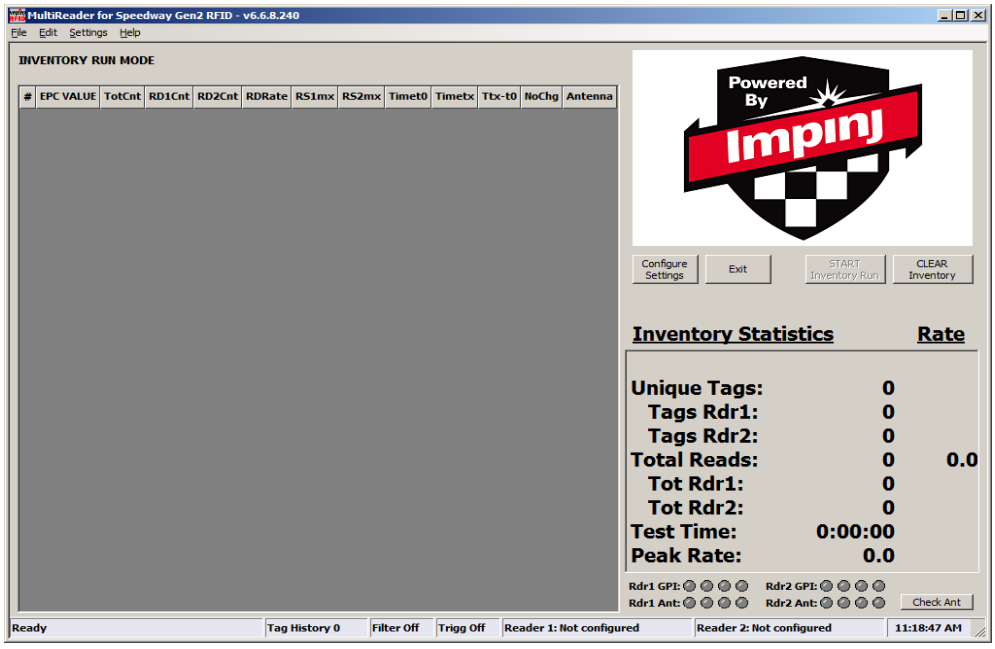
IMPINJ MULTIREADER SOFTWARE

Este software es dado por la empresa IMPIINJ, para sistemas RFID, es de uso sencillo y bastante útil para realizar pruebas de lecturas como identificación y renombramiento de las características de las antenas. Este puede ser adquirido directamente de la página de IMPINJ e instalado en cualquier computador.

A continuación se mostrara una breve explicación del programa y algunas de las aplicaciones y funcionalidades que este brinda.

Una vez iniciado el programa, en este se podrá ver la ventana vista en la Figura 39, en donde se ven los botones Configure Settings, Exit, START Inventory Run, CLEAR Inventory, también se puede ver el recuadro Inventory Statistics, donde se aprecia datos de las antenas y por último, el espacio de INVENTORY RUN MODE, en donde se puede apreciar el nombre de la antena, potencia, frecuencia, tiempo, entre otros.

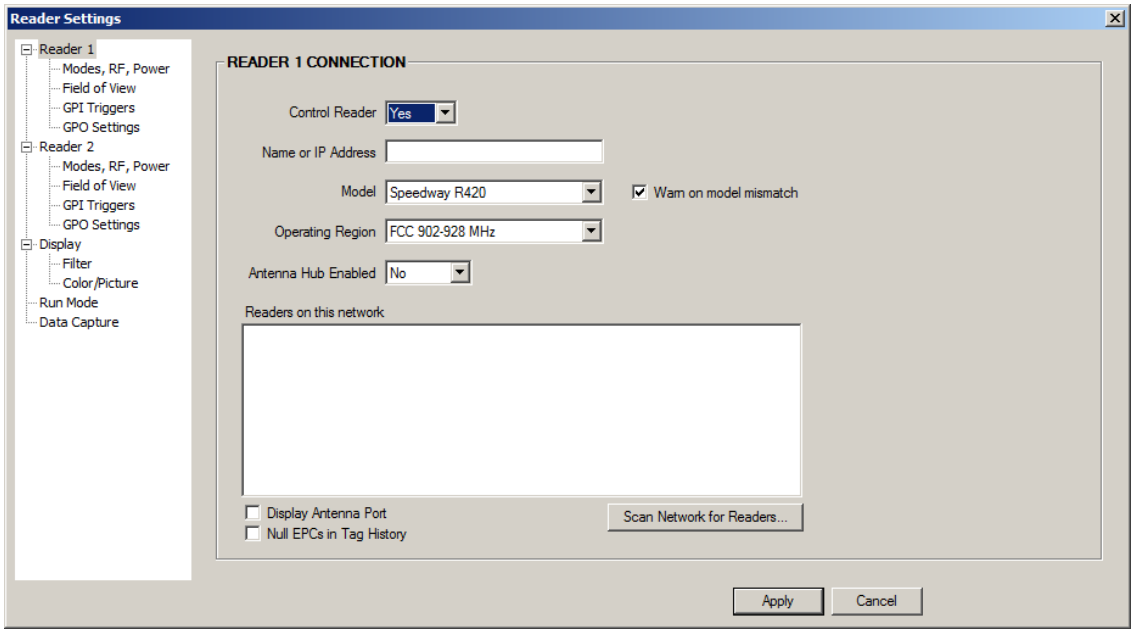
Figura 39. Multi-Reader Software



Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

Una vez iniciado el programa y habiendo ya configurado el Reader de tal forma que ya se haya establecido la conexión entre este y el computador, se pulsa el botón Configure Settings, en la nueva ventana aparecen pestañas que se completaran con los datos del Reader. En esta ventana se podrán configurar las opciones para dos Readers, y para configurar el aspecto visible de los resultados como se muestra en la Figura 40, parte izquierda.

Figura 40. Configure Settings



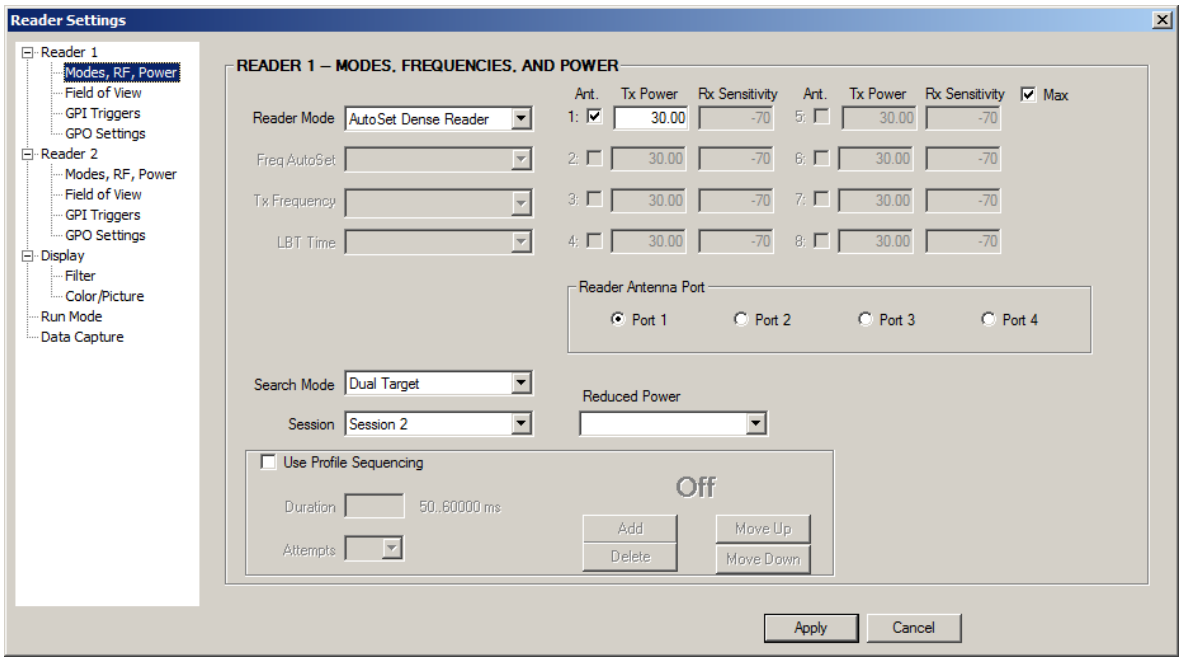
Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

En la opción READER CONNECTION se encuentra la opción Control Reader, el cual es para activar o desactivar el funcionamiento de este, luego Name or IP Address, en donde se coloca la IP dada por la terminal de Ubuntu una vez se inicie la conexión Reader- Computador, en Model se selecciona el modelo del Reader, en Operating Region se da la posibilidad de escoger que frecuencia trabaja el Reader, Antena Hub Enabled se puede activar en caso de tener una Antena de este tipo conectada al Reader. Todo esto puede verse en la Figura 40.

En la parte izquierda de la ventana mostrada en la Figura 40, al pulsar la opción Modes, RF, Power, se podrá apreciar como se muestra en la Figura 41, donde se podrá modificar el modo en que el Reader realizará la lectura, como también la sensibilidad de la antena al modificar la potencia.

De igual forma, en la opción Field of View, se puede configurar la estimación de tags que puede leer la antena por medio del Reader tal como se muestra en la Figura 42. Esto nos permitirá saber la densidad de antenas que puede llegar al leer el Reader en un instante.

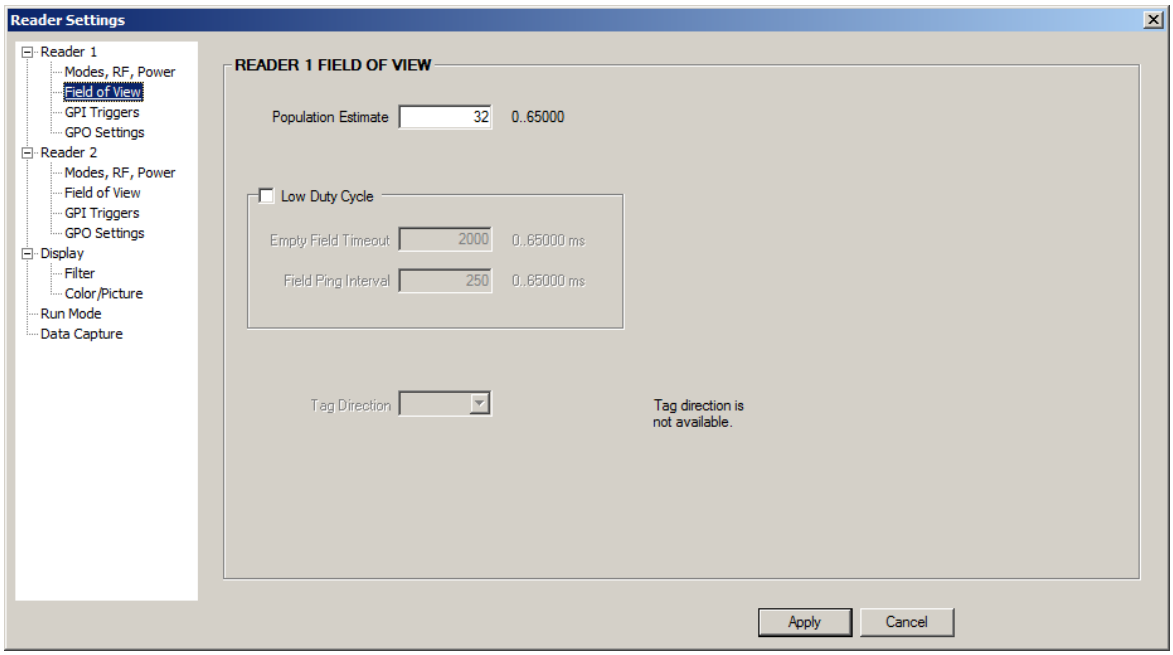
Figura 41. Modos, Frecuencia y Potencia de Antena



Provisto por IMPINJ. Multi-Reader Software. Tomado de [14]

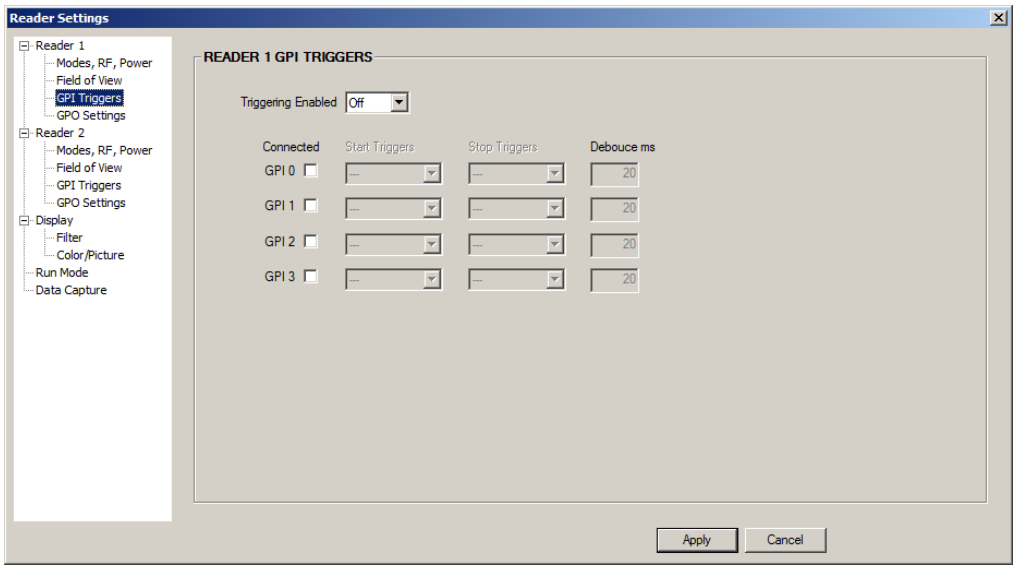
La activación de las funciones con GPI y GPIO, y su configuración vienen dadas en las opciones GPI Triggers y GPIO Settings, en donde se puede activar y configurar a la necesidad y aplicación de la persona. Ver Figura 43 y Figura 44.

Figura 42. Densidad de Antenas



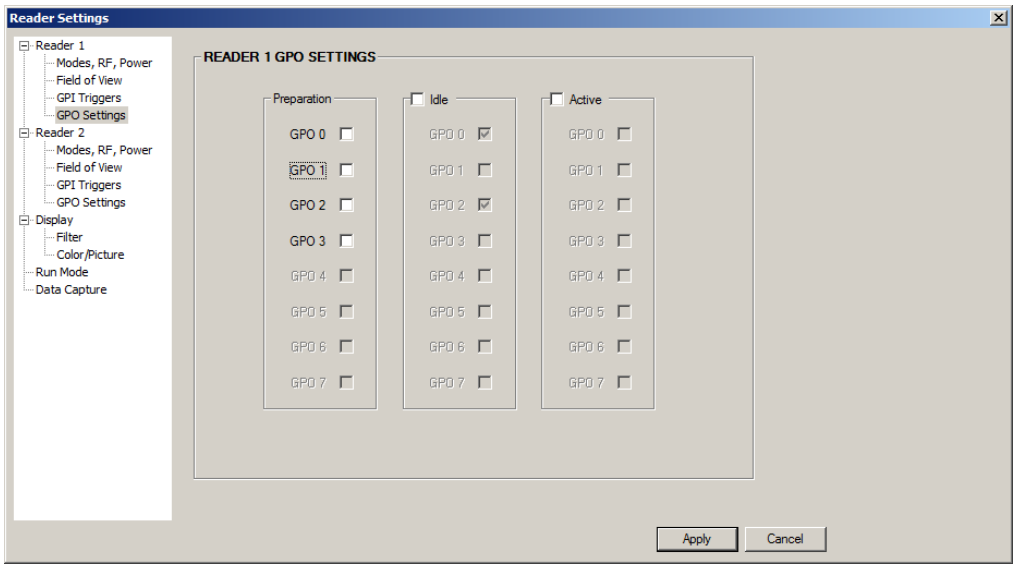
Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

Figura 43. Activación GPI



Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

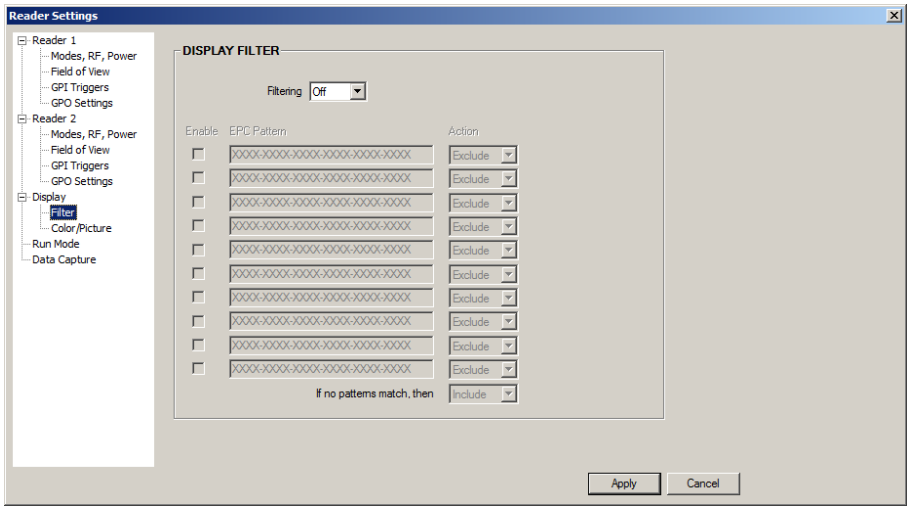
Figura 44. Configuración GPIO



Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

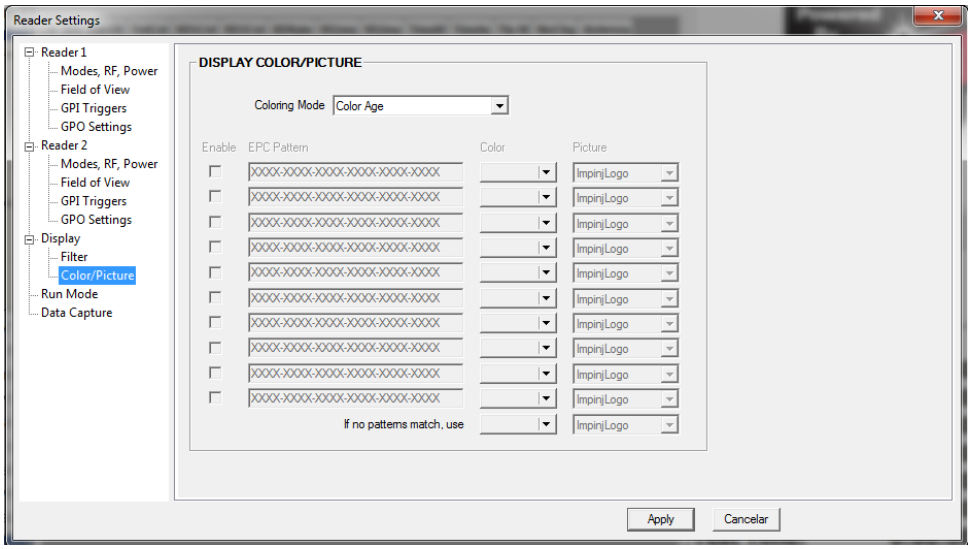
Otras opciones adicionales están dadas en el menú Display, donde se podrán leer solo las antenas que se desee, como cambiar color y formato de la ventaja en donde se muestra las lecturas entre otros, como se muestra en las Figura 45, Figura 46, Figura 47 y Figura 48.

Figura 45. Filtro.



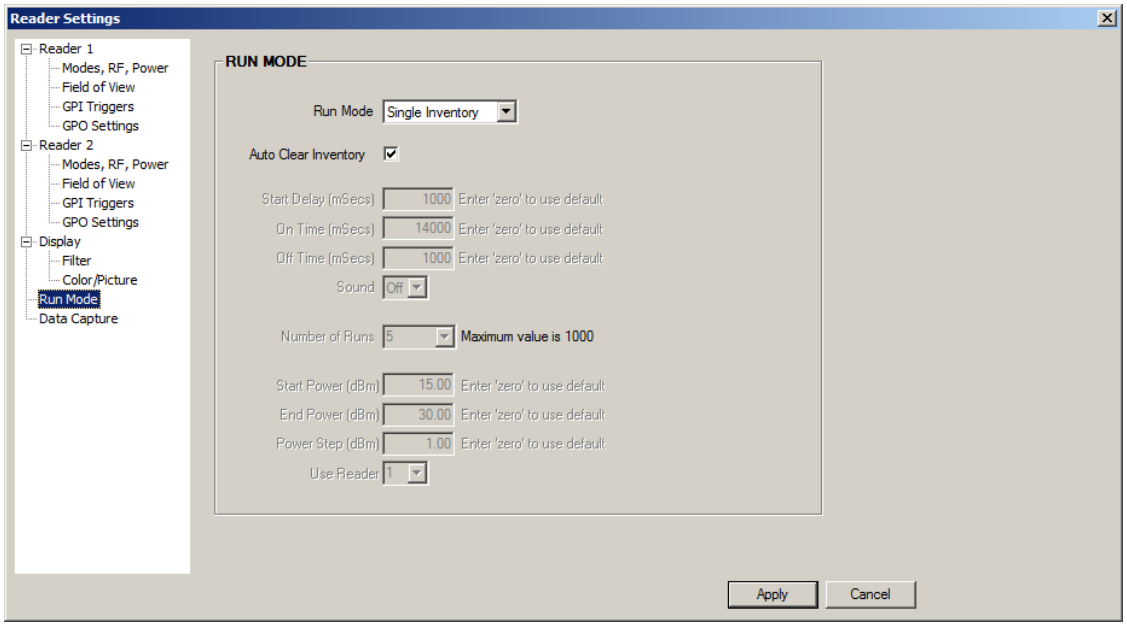
Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

Figura 46. Color



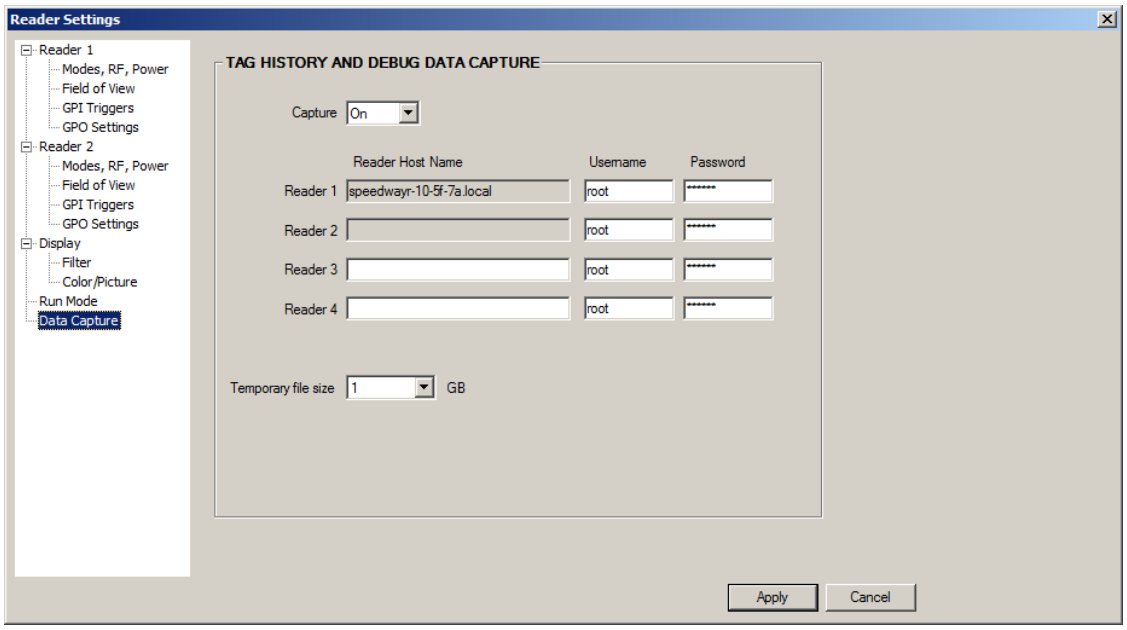
Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

Figura 47. Run Mode.



Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

Figura 48. Captura de Datos.



Provisto por IMPINJ. Multi-Reader Software. Tomado de [14].

Con este software se puede revisar la calidad de los tags, como también el funcionamiento correcto de las antenas y el Reader, también permite obtener lecturas rápidas de productos, mostrando en pantalla las antenas detectadas, permite guardar información referente a los productos logrando diferenciar una antena de otra a pesar que tengan la misma referencia, también guarda historiales de lectura de los productos.

Una de las aplicaciones que permite el software es registrar en orden llegada de un producto, y especificando el tiempo en el que ha sido registrado, pero a pesar de todos los beneficios, opciones y su fácil manejo, este no permite guardar los datos leídos de la antena en tiempo real, si no guardarlos una vez se ha detenido el proceso.

ANEXO B. CODIGO PYTHON DE RECOLECCION Y GUARDADO EN LA BASE DE DATOS.

Figura 49. Código Python. Acceso y guardado en la base de datos. Parte 1.

```
import time
import sys
import os
import socket
import binascii

import psycopg2

#accediendo a la base de datos

conn = psycopg2.connect("dbname='inventario_total' user='postgres' host='localhost' password='0000'")
print "base de datos lista para usar"

cursor = conn.cursor()

# Definiendo variables de conexion.

#TCP_IP = 'speedwayr-10-a0-70.local'
TCP_IP = '192.168.1.143'
TCP_PORT = 14150
BUFFER_SIZE = 1024
Path='/home/leyendo.txt'

#  Definiendo mensaje de envio.
#MESSAGE="MASIVO"

try:
    print("Abriendo el puerto.")
    # Creacion del socket.
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)


try:
    print("Abriendo el puerto.")
    # Creacion del socket.
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Definicion de timeout de recepcion y tamano del socket.
    s.settimeout(120)

    # Conexion con el socket.
    s.connect((TCP_IP, TCP_PORT))
    print("Puerto abierto.")

    #s.send(MESSAGE)
    time.sleep(1)
```

Figura 50. Código Python. Acceso y guardado en la base de datos. Parte 2.

```
# Recepcion continua de informacion de consumos, almacenamiento en un archivo de texto plano.
data = s.recv(BUFFER_SIZE)
Fsave=open(Path,"a")
Fsave.write(data)
print data

# NOTA: La cantidad de datos recibidos supera los 1024 bytes, por lo tanto el comando de recepcion de
# trama deber ser recursivo.
while(data.find("H")==-1):
    inicio=time.time()
    data = s.recv(BUFFER_SIZE)
    data1 = data.split(',')

 if data1[0]=="5": #numero de la antena de ventas

     cursor.execute("INSERT INTO vendido VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
     print "datos insertados en base de datos de ventas "

 elif data1[0]=="1":

     cursor.execute("INSERT INTO antena1 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
     print "datos insertados en base de datos de la estanteria"

 elif data1[0]=="2":

     cursor.execute("INSERT INTO antena2 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
     print "datos insertados en base de datos de la estanteria"

 elif data1[0]=="3":

     cursor.execute("INSERT INTO antena3 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
     print "datos insertados en base de datos de la estanteria"

 elif data1[0]=="4":

     cursor.execute("INSERT INTO antena4 VALUES ('"+data1[0]+"','"+data1[1]+"', '"+data1[2]+"','"+data1[3]+"')")
     print "datos insertados en base de datos de la estanteria"


 conn.commit()

        Fsave.write(data)
        fin=time.time()
        tiempo_total = fin - inicio

        print("tiempo_envio ",tiempo_total," segundos")
        print data

    conn.close()
    cursor.close
    Fsave.close()
    s.close()
    print("Termino")
except Exception,e:
    data=-1
    print("error-- "+str(e))
```

ANEXO C. CODIGO HTML – INTERFAZ GRAFICA – PAGINA PRINCIPAL.

Figura 51. Código HTML. Página Principal.

```
<html>
<head>
 <title> ESTANTERIA iNTELIGENTE UIS </title>
</head>


<body background = "fondo.jpg" text='white'>

 <a href = 'http://www.uis.edu.co/webUIS/es/index.jsp' title = 'Universidad Industrial de Santander'
target = new><img src="entrada.jpg" width="100%" height="15%" alt="paylas"> <a/>


<CENTER>
 <h1><B><FONT SIZE=7> UNIVERSIDAD INDUSTRIAL DE SANTANDER </FONT></B></h1>
 <h1><B><FONT SIZE=6> ESCUELA INGENIERIA ELECTRICA, ELECTRONICA Y TELECOMUNICACIONES </FONT></B></h1>
 <h1><B><FONT SIZE=6> ESTANTERIA INTELIGENTE </FONT></B></h1>
 <CENTER/>
 <hr color= black width=100%>


<P> <FONT SIZE = 4>La Estanteria Inteligente es la solucion con tecnologia RFID
  que permite la deteccion automatica, identificacion y localizacion de cada uno de los articulos
  expuestos en tiempo real y de manera simultanea. Este proyecto ha sido realizado
  bajo la direccion de <a href= 'http://scienti1.colciencias.gov.co:8081/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0001502236'
 target= new <strong> MIE(c). Efren Dario Acevedo Cardenas</strong></a>
   Director y <a href= 'http://scienti1.colciencias.gov.co:8081/cvlac/visualizador/generarCurriculoCv.do?cod_rh=0000216119'
  target= new <strong> M.Sc. Oscar Mauricio Reyes Torres</strong></a>   Codirector </P>
 <BR>

 <a href='http://localhost/busqueda_de_producto.html' <h1><strong>Opciones de Productos</strong></h1></a>
  <BR><BR>
<a href='http://localhost/estanteria_virtual.html' <h1><strong>Estanteria Virtual</strong></h1></a>

 <a href='http://localhost/busqueda_de_producto.html' <h1><strong>Opciones de Productos</strong></h1></a>
 <BR><BR>
<a href='http://localhost/estanteria_virtual.html' <h1><strong>Estanteria Virtual</strong></h1></a>
<BR><BR>
<a href='http://localhost/Nuestros%20productos.html' <h1><strong>Nuestros Productos</strong></h1></a>


<BR><BR>
<hr color= black width=100%>
<CENTER>
<a href ='https://www.facebook.com/fer.mera'
target= new <h1><B><FONT SIZE=4> ALEX FERNAN PARRA MERA </FONT></B></h1></a> <BR>
<a href ='https://www.facebook.com/jhoimareduardo?fref=ts'
target= new<h1><B><FONT SIZE=4> JHOIMAR EDUARDO SALAMANCA RODRIGUEZ </FONT></B></h1></a>
<BR>
<address> pagina creada por estudiantes UIS </address>

<CENTER/>

</body>
</html>
```

ANEXO D. CODIGO INTERFAZ GRAFICA – BUSQUEDA DEL PRODUCTO.

1. CODIGO HTML.

Figura 52. Código HTML. Búsqueda del Producto

```
<html>
<head>
 <title></title>
</head>

<body background = "fondo.jpg">

<center>
 <h1> Busque informacion de un producto en estanteria</h1>

<form action= "traer.php" method="post" name="form">
    <input type="text" name="epc" />
    <input type="submit" value="buscar" />
</form>

 <h1> Informacion de un producto vendido</h1>

<form action= "traer2.php" method="post" name="form">
    <input type="text" name="epc" />
    <input type="submit" value="buscar" />
</form>

 <h1> Total produtos vendidos</h1>

<form action= "traer3.php" method="post" name="form">
    <input type="text" name="epc" />
    <input type="submit" value="buscar" />
</form>
<a href ='file:///var/www/html/estanteria.html' <h1><B><FONT SIZE=4> REGRESAR MENU </FONT></B></h1></a>
</center>
</body>
 </html>
```

2. CODIGO PHP

2.1. Información de Producto en Estantería Inteligente.

Figura 53. Código de Información de Producto en Estantería Inteligente.

```php
<?php

ini_set('display_errors', 'On');
    ini_set('display_errors', 1);

$cadena=" host='localhost' port='5432' dbname='inventario_total' user= 'postgres' password='0000' ";
$contador=0;
$con=pg_connect($cadena)or die("problema con la conexion");
echo "conectado a la base de datos"."<BR><BR>";

$epc=$_POST['epc'];
$antena = $_POSR['antena']

if($antena==1){

$registro= pg_query("SELECT * FROM antena1 ");

while($reg=pg_fetch_array($registro) )
{
    if($reg['EPC']==$epc){
    $contador = $contador +1;
    echo"<BR><BR> ","  Puerto antena= ", $reg['Antenna Port']","       ",",  EPC= ",$reg['EPC']","          ",",  Timestamp= ",$reg['Timestamp']","
    }
}
 echo "<BR>", $contador, " productos vendidos en total"

}elseif($antena==2){
    $registro= pg_query("SELECT * FROM antena2 ");

while($reg=pg_fetch_array($registro) )
    if($reg['EPC']==$epc){
    $contador = $contador +1;
    echo"<BR><BR> ","  Puerto antena= ", $reg['Antenna Port']","       ",",  EPC= ",$reg['EPC']","          ",",  Timestamp= ",$reg['Timestamp']","
    }
}
  echo "<BR>", $contador, " productos vendidos en total"
 }elseif($antena==3){

$registro= pg_query("SELECT * FROM antena3 ");

while($reg=pg_fetch_array($registro) )
{
    if($reg['EPC']==$epc){
    $contador = $contador +1;
    echo"<BR><BR> ","  Puerto antena= ", $reg['Antenna Port']","       ",",  EPC= ",$reg['EPC']","          ",",  Timestamp= ",$reg['Timestamp']","
    }
}
 echo "<BR>", $contador, " productos vendidos en total"

}elseif($antena==4){
    $registro= pg_query("SELECT * FROM antena4 ");

while($reg=pg_fetch_array($registro) )
{
    if($reg['EPC']==$epc){
    $contador = $contador +1;
    echo"<BR><BR> ","  Puerto antena= ", $reg['Antenna Port']","       ",",  EPC= ",$reg['EPC']","          ",",  Timestamp= ",$reg['Timestamp']","
    }
}
 echo "<BR>", $contador, " productos vendidos en total"
}
?>
```

2.2. Información de un Producto Vendido.

Figura 54. Código PHP de Productos Vendidos

```php
<?php

ini_set('display_errors', 'On');
    ini_set('display_errors', 1);


$cadena=" host='localhost' port='5432' dbname='inventario_total' user= 'postgres' password='0000' ";

$contador=0;

$con=pg_connect($cadena)or die("problema con la conexion");

echo "conectado a la base de datos"."<BR><BR>";
//mysql_select_db($db,$con)or die("problema al conectar db");
$epc=$_POST['epc'];

$registro= pg_query("SELECT * FROM vendido"); #     WHERE 'Antenna Port'='4' ");
echo $registro;
while($reg=pg_fetch_array($registro) )
{
    if($reg['EPC']==$epc){
    $contador = $contador +1;
    echo"<BR><BR> ","  Puerto antena= ", $reg['Antenna Port']," ","  EPC= ",$reg['EPC']," "
    }

}

 echo "<BR>", $contador, " productos vendidos en total"
?>
```

2.3. Información Total de Productos Vendidos.

Figura 55.Código PHP de Total de Productos Vendidos.

```php
<?php

ini_set('display_errors', 'On');
    ini_set('display_errors', 1);

$cadena=" host='localhost' port='5432' dbname='inventario_total' user= 'postgres' password='0000' ";
$contador=0;
$con=pg_connect($cadena)or die("problema con la conexion");
echo "conectado a la base de datos"."<BR><BR>";

$epc=$_POST['epc'];

$registro= pg_query("SELECT * FROM vendido"); #     WHERE 'Antenna Port'='4' ");

while($reg=pg_fetch_array($registro) )
{
    $contador = $contador +1;
    echo"<BR><BR> ","  Puerto antena= ", $reg['Antenna Port'],"       ","",  EPC= ",$reg['EPC'],"
}
 echo "<BR>", $contador, " productos vendidos en total"
?>
```

ANEXO E. CODIGO INTERFAZ GRAFICA – ESTANTERIA VIRTUAL

Figura 56. Código PHP-HTML. Estantería Virtual. Parte 1.

```php
<?php

ini_set('display_errors', 'On');
    ini_set('display_errors', 1);
$cadena=" host='localhost' port='5432' dbname='inventario_total' user= 'postgres' password='0000' ";
$contador=0;
$con=pg_connect($cadena)or die("problema con la conexion");

$registro1= pg_query("SELECT * FROM antena1");
while($reg1=pg_fetch_array($registro1) )
{
    $_POST['Antena1']=$reg1['Antenna Port'];
        $_POST['EPC1']=$reg1['EPC'];
        $_POST['potencia1']=$reg1['Peak RSSI'];
        echo"<script>\n";
        echo"Antena1='".$_POST['Antena1']."'\n";
        echo"EPC1='".$_POST['EPC1']."'\n";
        echo"potencia1='".$_POST['potencia1']."'\n";
        echo"</script>\n";
}

$registro2= pg_query("SELECT * FROM antena2");
while($reg2=pg_fetch_array($registro2) )
{
    $_POST['Antena2']=$reg2['Antenna Port'];
        $_POST['EPC2']=$reg2['EPC'];
        $_POST['potencia2']=$reg2['Peak RSSI'];
        echo"<script>\n";
        echo"Antena2='".$_POST['Antena2']."'\n";
        echo"EPC2='".$_POST['EPC2']."'\n";
        echo"potencia2='".$_POST['potencia2']."'\n";
        echo"</script>\n";
}
```

```php
$registro3= pg_query("SELECT * FROM antena3");
while($reg3=pg_fetch_array($registro3) )
{
    $_POST['Antena3']=$reg3['Antenna Port'];
        $_POST['EPC3']=$reg3['EPC'];
        $_POST['potencia3']=$reg3['Peak RSSI'];
        echo"<script>\n";
        echo"Antena3='".$_POST['Antena3']."'\n";
        echo"EPC3='".$_POST['EPC3']."'\n";
        echo"potencia3='".$_POST['potencia3']."'\n";
        echo"</script>\n";
}

$registro4= pg_query("SELECT * FROM antena4");
while($reg4=pg_fetch_array($registro4) )
{
    $_POST['Antena4']=$reg4['Antenna Port'];
        $_POST['EPC4']=$reg4['EPC'];
        $_POST['potencia4']=$reg4['Peak RSSI'];
        echo"<script>\n";
        echo"Antena4='".$_POST['Antena4']."'\n";
         echo"EPC4='".$_POST['EPC4']."'\n";
        echo"potencia4='".$_POST['potencia4']."'\n";
        echo"</script>\n";
}
?>
```

Figura 57. Código PHP-HTML. Estantería Virtual. Parte 2.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta http-equiv=refresh content=5>
<title>Thumbnail Cycle Tutorial</title>
<style>

.item{
    float: left;
    padding: 10px;
    margin: 10px;
}
.item > img{
    width: 180px;
    height: 140px;
    border: #666 1px solid;
    cursor: pointer;
}
.item > span{
    display: block;
    text-align: center;
    margin-top: 10px;
}
</style>
<script language="JavaScript" src="archivo.php"></script>
<script>
window.addEventListener("load", function(){

    zapato1="";
    zapato2="";
    zapato3="";

    var items_container = document.getElementById("items_container");
    var thumbtimer, ti=0, dir="/";

    if(EPC1=="8030A00829000000000A87B1" && potencia1<-48){
    obj1 = { name:zapato1, pics:["6.jpg","7.jpg","8.jpg"] };
    }else{
    obj1 = { name:zapato1, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
    }

        if(EPC1=="0003FB63AC1F3841EC480471" && potencia1<-48){
    obj2 = { name:zapato2, pics:["9.jpg","10.jpg","11.jpg"] };
        }else{
        obj2 = { name:zapato2, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }

        if(EPC1=="0003FB63AC1F3841EC880471" && potencia1<-48){
    obj3 = { name:zapato3, pics:["12.jpg","13.jpg","14.jpg"] };
        }else{
        obj3 = { name:zapato3, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }
```

Figura 58. Código PHP-HTML. Estantería Virtual. Parte 3.

```
    var ary = [obj1,obj2,obj3];
    for(var i=0; i < ary.length; i++){
        var div = document.createElement("div");
        var img = document.createElement("img");
        var span = document.createElement("span");
        div.className = "item";
        img.oi = i;
        img.src = dir + ary[i].pics[0];
        span.innerHTML = ary[i].name;
        items_container.appendChild(div);
        div.appendChild(img);
        div.appendChild(span);
        img.addEventListener("mouseover", function(event){
            thumbtimer = setInterval(function(){
                ti++;
                if(ti == ary[event.target.oi].pics.length){
                    ti = 0;
                }
                event.target.src = dir + ary[event.target.oi].pics[ti];
            }, 700);
        });
        img.addEventListener("mouseout", function(event){
            clearInterval(thumbtimer);
            ti = 0;
            event.target.src = dir + ary[event.target.oi].pics[ti];
        });
    }

});
</script>

<style>
.item{
    float: left;

    padding: 10px;
    margin: 10px;
}
.item > img{
    width: 180px;
    height: 140px;
    border: #666 1px solid;
    cursor: pointer;
}
.item > span{
    display: block;
    text-align: center;
    margin-top: 10px;
}
</style>
<script>
window.addEventListener("load", function(){

    zapato4="";
    zapato5="";
    zapato6="";

    var items_container2 = document.getElementById("items_container");
    var thumbtimer, ti=0, dir="/";
```

Figura 59. Código PHP-HTML. Estantería Virtual. Parte 4.

```
    if(EPC2=="E200900982020131070700D1BF" && potencia2<-48){
    obj1 = { name:zapato4, pics:["3.jpg","2.jpg","3.jpg"] };
        }else{
        obj1 = { name:zapato4, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }

    if(EPC2=="E200900982020131070700D1BF" && potencia2<-48){
    obj2 = { name:zapato5, pics:["3.jpg","2.jpg","3.jpg"] };
        }else{
        obj2 = { name:zapato5, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }

    if(EPC2=="E200900982020131070031BF" && potencia2<-48){
    var obj3 = { name:zapato6, pics:["1.jpg","3.jpg","4.jpg","5.jpg"] };
        }else{
            obj3 = { name:zapato6, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }
    var ary = [obj1,obj2,obj3];
    for(var i=0; i < ary.length; i++){
        var div = document.createElement("div");
        var img = document.createElement("img");
        var span = document.createElement("span");
        div.className = "item";
        img.oi = i;
        img.src = dir + ary[i].pics[0];
        span.innerHTML = ary[i].name;
        items_container.appendChild(div);
        div.appendChild(img);
        div.appendChild(span);
        img.addEventListener("mouseover", function(event){
            thumbtimer = setInterval(function(){
                ti++;
                if(ti == ary[event.target.oi].pics.length){
                    ti = 0;
                }
                event.target.src = dir + ary[event.target.oi].pics[ti];
            }, 700);
        });
        img.addEventListener("mouseout", function(event){
            clearInterval(thumbtimer);
            ti = 0;
            event.target.src = dir + ary[event.target.oi].pics[ti];
        });
    }
});
</script>

<style>
.item{
    float: left;

    padding: 10px;
    margin: 10px;
}
.item > img{
    width: 180px;
    height: 140px;
    border: #666 1px solid;
    cursor: pointer;
}
.item > span{
    display: block;
    text-align: center;
    margin-top: 10px;
```

Figura 60. Código PHP-HTML. Estantería Virtual. Parte 5.

```
</style>
<script>
window.addEventListener("load", function(){

    zapato7="";
    zapato8="";
    zapato9="";

    var items_container3 = document.getElementById("items_container3");
    var thumbtimer, ti=0, dir="/";

        var condicion1=30;
        if(EPC3="0001FB63AC1F3841EC880467" && potencia3<-48){
    obj1 = { name:zapato7, pics:["3.jpg","2.jpg","3.jpg"] };
        }else{
         obj1 = { name:zapato7, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }

        var condicion1=30;
       if(EPC3="0001FB63AC1F3841EC830467" && potencia3<-48){
    obj2 = { name:zapato8, pics:["3.jpg","4.jpg","5.jpg"] };
        }else{
        obj2 = { name:zapato8, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }

        var condicion1=30;
        if(EPC3="0001FB63AC1F3841EC880467" && potencia3<-48){
    var obj3 = { name:zapato9, pics:["1.jpg","3.jpg","4.jpg","5.jpg"] };
        }else{
         obj2 = { name:zapato9, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }
    var ary = [obj1,obj2,obj3];
```

```
    for(var i=0; i < ary.length; i++){
        var div = document.createElement("div");
        var img = document.createElement("img");
        var span = document.createElement("span");
        div.className = "item";
        img.oi = i;
        img.src = dir + ary[i].pics[0];
        span.innerHTML = ary[i].name;
        items_container.appendChild(div);
        div.appendChild(img);
        div.appendChild(span);
        img.addEventListener("mouseover", function(event){
            thumbtimer = setInterval(function(){
                ti++;
                if(ti == ary[event.target.oi].pics.length){
                    ti = 0;
                }
                event.target.src = dir + ary[event.target.oi].pics[ti];
            }, 700);
        });
        img.addEventListener("mouseout", function(event){
            clearInterval(thumbtimer);
            ti = 0;
            event.target.src = dir + ary[event.target.oi].pics[ti];
        });
    }
});
```

Figura 61. Código PHP-HTML. Estantería Virtual. Parte 6.

```
<style>
.item{
    float: left;

    padding: 10px;
    margin: 10px;
}
.item > img{
    width: 180px;
    height: 140px;
    border: #666 1px solid;
    cursor: pointer;
}
.item > span{
    display: block;
    text-align: center;
    margin-top: 10px;
}
</style>
<script>
window.addEventListener("load", function(){

    zapato10="";
    zapato11="";
    zapato12="";

    var items_container4 = document.getElementById("items_container3");
    var thumbtimer, ti=0, dir="/";

        var condicion1=30;
        if(EPC4="0007FB63AC1F3841EC880469" && potencia4<-48){
    obj1 = { name:zapato10, pics:["3.jpg","2.jpg","3.jpg"] };
        }else{
         obj1 = { name:zapato10, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }

        var condicion1=30;
       if(EPC4="0007FB63AC1F3841EC880469" && potencia4<-48){
    obj2 = { name:zapato11, pics:["3.jpg","4.jpg","5.jpg"] };
        }else{
        obj2 = { name:zapato11, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }

        var condicion1=30;
        if(EPC4="0007FB63AC1F3841EC880469" && potencia4<-48){
    var obj3 = { name:zapato12, pics:["1.jpg","3.jpg","4.jpg","5.jpg"] };
        }else{
         obj2 = { name:zapato12, pics:["fondo.jpg","fondo.jpg","fondo.jpg"] };
        }
    var ary = [obj1,obj2,obj3];
    for(var i=0; i < ary.length; i++){
        var div = document.createElement("div");
        var img = document.createElement("img");
        var span = document.createElement("span");
        div.className = "item";
        img.oi = i;
        img.src = dir + ary[i].pics[0];
        span.innerHTML = ary[i].name;
        items_container.appendChild(div);
        div.appendChild(img);
        div.appendChild(span);
        img.addEventListener("mouseover", function(event){
            thumbtimer = setInterval(function(){
                ti++;
                if(ti == ary[event.target.oi].pics.length){
                    ti = 0;
```

Figura 62. Código PHP-HTML. Estantería Virtual. Parte 7.

```
                }
                event.target.src = dir + ary[event.target.oi].pics[ti];
            }, 700);
        });
        img.addEventListener("mouseout", function(event){
            clearInterval(thumbtimer);
            ti = 0;
            event.target.src = dir + ary[event.target.oi].pics[ti];
        });
    }
});
</script>

</head>
<body background = "fondo.jpg">



<CENTER>
<h1>  ESTANTERIA VIRTURAL </H1>
<TABLE align="center" border="2" width="51%" height="80%" bgcolor="A84E1B">

    <TR>
        <td align="center">
            <div id="items_container"></div>
        </TD>
    </TR>

    <TR>
        <td align="center">
            <div id="items_container2"></div>
        </TD>
    </TR>

    <TR>
        <td align="center">
            <div id="items_container3"></div>
        </TD>
    </TR>

        <TR>
        <td align="center">
            <div id="items_container4"></div>
        </TD>
    </TR>
<TABLE>

</CENTER>
<BR><br>
<a href ='file:///var/www/html/estanteria.html' <h1><B><FONT SIZE=4> REGRESAR MENU </FONT></B></h1></a>

</body>

</html>
```

ANEXO F. CODIGO INTERFAZ GRAFICA – NUESTROS PRODUCTOS.

1. CODIGO HTML

Figura 63. Código HTML. Nuestros Productos.

```
</head>
<body background = "fondo.jpg">

 <a href = 'http://www.uis.edu.co/webUIS/es/index.jsp' title = 'Universidad Industrial de Santander'
 target = new><img src="entrada.jpg" width="100%" height="15%" alt="paylas"> <a/>

<center>
 <H1> "NUESTROS PRODUCTOS"</H1>
</center>
 <BR></BR>
<center>
<div id = "galeria">

    <div id= "galeria_base">
    <img src = "1.jpg" id="imgMostrar">

    </div>

    <div id="galeria_miniaturas">

    <img class = "miniatura" src="1.jpg"
    onclick="document.getElementById('imgMostrar').src='1.jpg';"></img>
    <img class = "miniatura" src="2.jpg"
    onclick="document.getElementById('imgMostrar').src='2.jpg';"></img>
    <img class = "miniatura" src="3.jpg"
    onclick="document.getElementById('imgMostrar').src='3.jpg';"></img>
    <img class = "miniatura" src="6.jpg"
    onclick="document.getElementById('imgMostrar').src='6.jpg';"></img>

    </div>
</div>

<div id="galeria_miniaturas">

    <img class = "miniatura2" src="1.jpg"></img>
    <p> "zapato clasico"<BR></BR>"COLORES: negro y cafe"<BR></BR>"costo : $ 75.000"<BR></BR>

    <img class = "miniatura2" src="2.jpg"></img>
    <p> "zapato moderno"<BR></BR>"COLORES: azul y rojo"<BR></BR>"costo : $ 175.000"<BR></BR>

    <img class = "miniatura2" src="3.jpg"></img>
    <p> "zapato chico"<BR></BR>"COLORES: negro y rojo"<BR></BR>"costo : $ 65.000"<BR></BR>

    <img class = "miniatura2" src="6.jpg"></img>
    <p> "tennis"<BR></BR>"COLORES: gris, negro, rojo y azul"<BR></BR>"costo : $ 225.000"<BR></BR>

    </div>
</body>
</html>
```

1. CODIGO .CSS ESTILOS.

Figura 64. Código Java Script. Estilos.

```
body{
 background-color : grey;
}
#galeria {
 border : 1px solid #EAEAEA;
 border-radius: 25px;
 padding: 20px;
 padding-bottom: 0;
 background: #0099CC;
 width : 700px;
 margin: auto;
}
#imgMostrar {
 border: 1px solid #F2F2F2;
 border-radius: 25px;
 width: 700px;
 height: 400px;
}
#galeria_miniaturas{
 display: table;
 margin: 0px auto;
}
.miniatura{
 width: 60px;
 height:60px;
 border-radius: 10px;
 float: left;
 cursor: pointer;
 padding: 5px;
 margin: 10px 5px;
}
.miniatura2{
 width: 60px;
 height:60px;
 border-radius: 10px;
 float: down;
 cursor: pointer;
 padding: 5px;
margin: 10px 5px;}
```

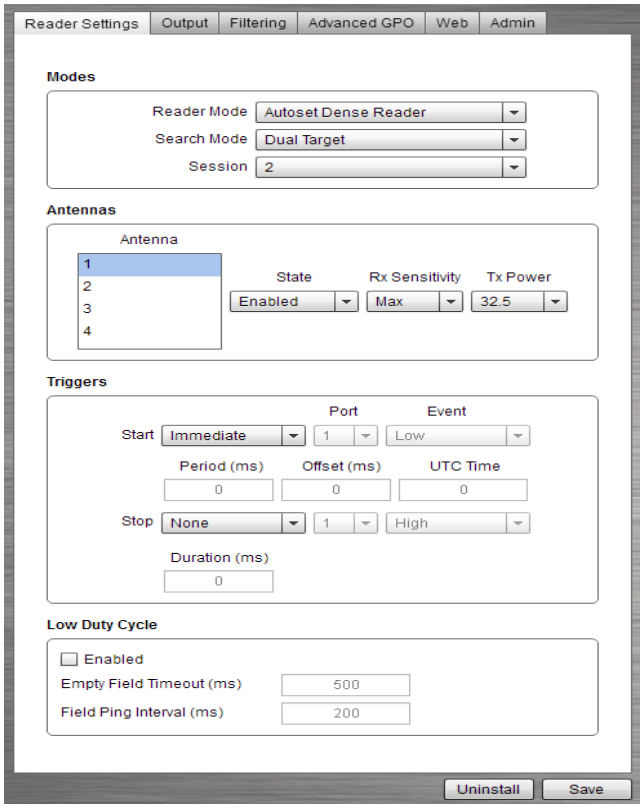
ANEXO G. MODOS DE FUNCIONAMIENTO.

1. MODOS DE FUNCIONAMIENTO.

Una vez configurado el Reader, con el archivo SpeedwayConnect_1.0.upg como se mostró anteriormente, ahora en el navegador se coloca *https://* seguido de la IP del Reader, esto nos llevara a la página de configuración del Speedway Revolution, en donde podemos configurar el Reader de la siguiente forma.

En la primera pestaña como se puede observar en la Figura 65, corresponde a Reader Settings, en donde se puede establecer los modos de lectura para hacerlo de forma eficiente, también se puede activar o desactivar las antenas, cada función se explicara a continuación.

Figura 65. Configuración Reader.



1.1. READER SETTINGS.

1.1.1. Reader Mode. Este parámetro es el que permite decidir el modo de lectura del Reader, ya sea una lectura a gran distancia por la antena o una lectura de proximidad, los modos de lectura y sus características son:

Autoset Dense Reader
<ModeIdentifier>1000</ModeIdentifier>
<DRValue>DRV_8</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_FM0</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>Unknown</SpectralMaskIndicator>
<BDRValue>40000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>6250</MinTariValue>
<MaxTariValue>6250</MaxTariValue>
<StepTariValue>0</StepTariValue>

Autoset Single Reader
<ModeIdentifier>1001</ModeIdentifier>
<DRValue>DRV_8</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_FM0</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>Unknown</SpectralMaskIndicator>
<BDRValue>40000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>6250</MinTariValue>
<MaxTariValue>6250</MaxTariValue>
<StepTariValue>0</StepTariValue>

Max Throughput
<ModeIdentifier>0</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_FM0</MValue>
<ForwardLinkModulation>DSB_ASK</ForwardLinkModulation>

<SpectralMaskIndicator>MI</SpectralMaskIndicator>
<BDRValue>640000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>6250</MinTariValue>
<MaxTariValue>6250</MaxTariValue>
<StepTariValue>0</StepTariValue>

Hybrid Mode
<ModeIdentifier>1</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_2</MValue>
<ForwardLinkModulation>DSB_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>MI</SpectralMaskIndicator>
<BDRValue>640000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>6250</MinTariValue>
<MaxTariValue>6250</MaxTariValue>
<StepTariValue>0</StepTariValue>

Dense Reader (M=4)
<ModeIdentifier>2</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_4</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>DI</SpectralMaskIndicator>
<BDRValue>274000</BDRValue>
<PIEValue>2000</PIEValue>
<MinTariValue>20000</MinTariValue>
<MaxTariValue>20000</MaxTariValue>
<StepTariValue>0</StepTariValue>

Dense Reader (M=8)
<ModeIdentifier>3</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_8</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>DI</SpectralMaskIndicator>
<BDRValue>170600</BDRValue>
<PIEValue>2000</PIEValue>
<MinTariValue>20000</MinTariValue>

<MaxTariValue>20000</MaxTariValue>
<StepTariValue>0</StepTariValue>

MaxMiller
<ModeIdentifier>4</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_4</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>MI</SpectralMaskIndicator>
<BDRValue>640000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>7140</MinTariValue>
<MaxTariValue>7140</MaxTariValue>
<StepTariValue>0</StepTariValue>

Autoset Dense Reader
<ModeIdentifier>1000</ModeIdentifier>
<DRValue>DRV_8</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_FM0</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>Unknown</SpectralMaskIndicator>
<BDRValue>40000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>6250</MinTariValue>
<MaxTariValue>6250</MaxTariValue>
<StepTariValue>0</StepTariValue>

Max Throughput
<ModeIdentifier>0</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_FM0</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>MI</SpectralMaskIndicator>
<BDRValue>436000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>16670</MinTariValue>
<MaxTariValue>16670</MaxTariValue>
<StepTariValue>0</StepTariValue>

Hybrid Mode
<ModeIdentifier>1</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>

<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_2</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>MI</SpectralMaskIndicator>
<BDRValue>436000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>16670</MinTariValue>
<MaxTariValue>16670</MaxTariValue>
<StepTariValue>0</StepTariValue>

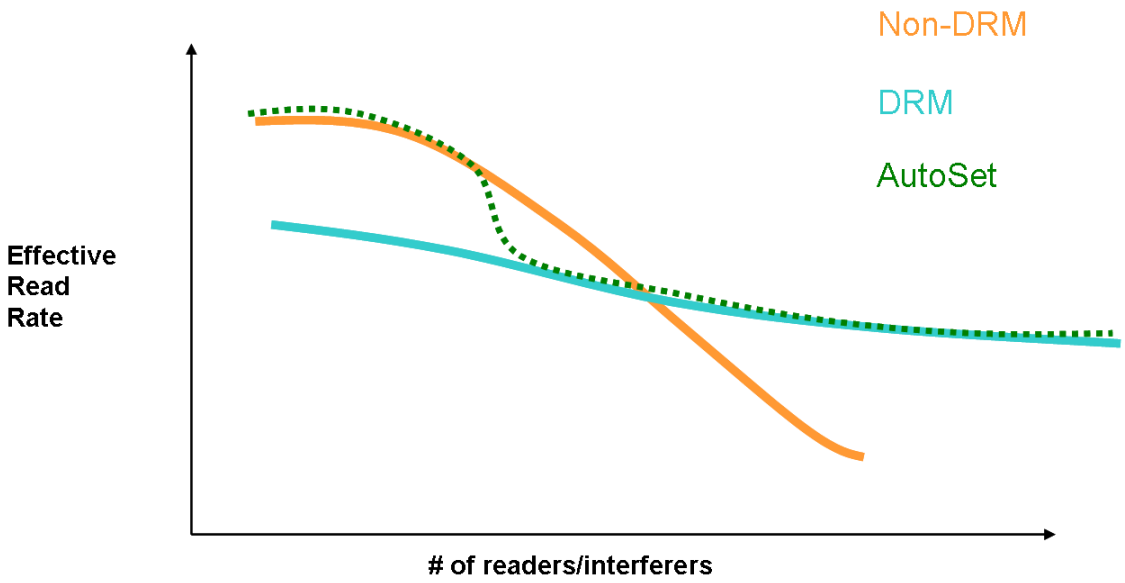
Dense Reader (M=4) 1
<ModeIdentifier>2</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_4</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>DI</SpectralMaskIndicator>
<BDRValue>320000</BDRValue>
<PIEValue>2000</PIEValue>
<MinTariValue>20000</MinTariValue>
<MaxTariValue>20000</MaxTariValue>
<StepTariValue>0</StepTariValue>

Dense Reader (M=4) 2
<ModeIdentifier>5</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_4</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>MI</SpectralMaskIndicator>
<BDRValue>320000</BDRValue>
<PIEValue>1500</PIEValue>
<MinTariValue>16670</MinTariValue>
<MaxTariValue>16670</MaxTariValue>
<StepTariValue>0</StepTariValue>

Dense Reader (M=8)
<ModeIdentifier>3</ModeIdentifier>
<DRValue>DRV_64_3</DRValue>
<EPCHAGTCConformance>false</EPCHAGTCConformance>
<MValue>MV_8</MValue>
<ForwardLinkModulation>PR_ASK</ForwardLinkModulation>
<SpectralMaskIndicator>DI</SpectralMaskIndicator>
<BDRValue>320000</BDRValue>
<PIEValue>2000</PIEValue>

<MinTariValue>20000</MinTariValue>
<MaxTariValue>20000</MaxTariValue>
<StepTariValue>0</StepTariValue>

Figura 66 Gráfica de efectividad de lectura de los modos.



Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [14].

Figura 67. Comparación entre los diferentes modos de lectura

LLRP Mode	Name	Sensitivity	Interference Tolerance	Environment*2
0	Max Throughput	Good	Poor	Multiple
1	Hybrid	Good	Good	Multiple
2	Dense Reader M= 4	Better	Excellent	Dense
3	Dense Reader M= 8	Best	Excellent	Dense
4*1	MaxMiller	Better	Good	Multiple

*1 MaxMiller is not available in all regions
*2 Region dependent (i.e. ETSI region readers all modes are configured for Dense environment)

Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [14]

1.1.2. Search Mode. Este parámetro permite al usuario establecer en que forma será leído los tags por la antena, ya que según sea necesario para la aplicación deseada, se necesita una lectura constante o una lectura lenta por cada Tag. Las opciones brindadas por Search Mode son las siguientes:

1.1.2.1. Dual Target. Este modo permite que una vez una antena percibe un Tag, este muestre su estado ON u OFF, haciendo un seguimiento continuo, generando muchas lecturas, pero permitiendo tener un control mayor sobre lo que está o no leyendo el Reader, es perfecto para pequeñas empresas.

1.1.2.2. Single Target. Este modo permite mostrar los tags que la antena percibe cuando están ON u OFF pero lo realiza de una forma más lenta, de esta forma habrá menos oscilaciones en las lecturas.

1.1.2.3. Single Target with Supression. Este modo permite leer los tags en la antena cuando estos están ON u OFF, lo realiza de forma similar a Single Target, pero de forma más segura. Esta como Single Target es la mejor opción para grandes empresas que manejan gran cantidad de productos.

1.1.3. Session. Esta opción permite establecer la forma en que el Reader responderá a la hora de no recibir señal de la antena.

1.1.4. Antennas. Permite seleccionar las antenas que se van a utilizar para la aplicación, el Reader detecta cuales antenas están conectadas, con esta opción la persona puede seleccionar cuales poner a funcionar.

1.1.5. Triggers. Esta opción permite colocar la configuración de encendido y apagado del Reader para su activación, así la persona sabrá si el Reader comenzará a funcionar inmediatamente y constante o inmediatamente periódico.

1.1.6. State. Permite habilitar o deshabilitar una antena.

1.1.7. Rx Sensitivity. Establece que tan sensible será la antena para recibir la potencia proveniente de los tags.

1.1.8. Low Duty Cycle. Esta opción permite establecer que sucederá en el Reader en caso de no presentar información que leer, este entrará en un estado de Low Duty Cycle de forma que ahorre esfuerzo al Reader, esta opción no está presente si la Antena Hub está funcionando.

1.2. OUTPUT

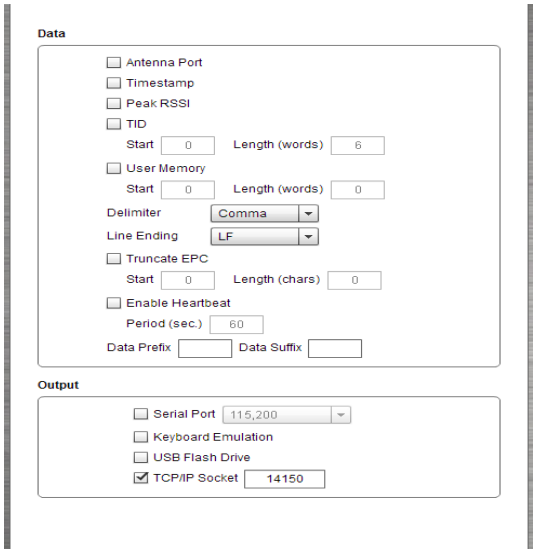
En la siguiente pestaña Output, aparece las opciones mostradas en la Figura 68, esta permite escoger la información que se necesite, el formato y el modo de adquisición de datos.

1.2.1. Data. En este campo, la persona puede escoger que información puede obtener el usuario, ya sea Antenna Port, Timestamp, Peak RSSI, TID que le permite a la persona la información sobre el Tag, las demás opciones de Data, permite dar formato al dato para delimitar cada valor.

1.2.2. Output. En esta opción el usuario selecciona la forma en que va a adquirir los datos del Reader hacia el computador, las diferentes formas que existen son Serial Port, Keyboard Emulation, USB Flash Drive, TCP/IP Socket.

En la siguiente pestaña Filtering, aparece las opciones mostradas en la Figura 69, permite escoger cada cuanto el Reader va a realizar una lectura y envía al computador la información. Esto con el fin de no llenar la memoria con archivos gigantescos o la base de datos con información redundante.

Figura 68. Formato de Salida de los datos



Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [14]

1.3. ADVANCED GPIO

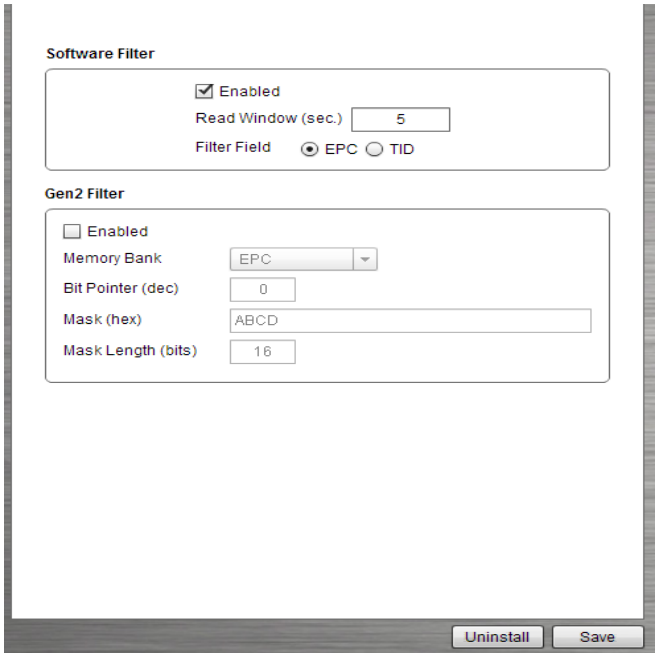
En la siguiente pestaña Advanced GPIO, se puede activar en caso de trabajar con GPIO con el Reader, este permite que puedan trabajar de formas diferentes a preferencia del usuario y a su necesidad como se muestra en la

Figura 70.

1.4. WEB SERVICE

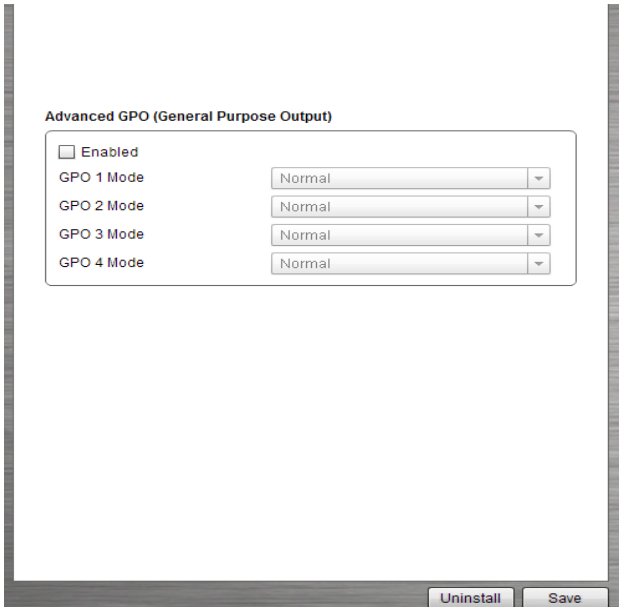
En la siguiente pestaña, Web, permite al usuario disponer de la información brindada por el Reader a través de un servidor Web, para ello se necesita definir la ruta a una página PHP en donde se recolectaran los datos. Para tener una mejor vista de lo dicho, vaya a la Figura 71.

Figura 69. Filtro de datos.



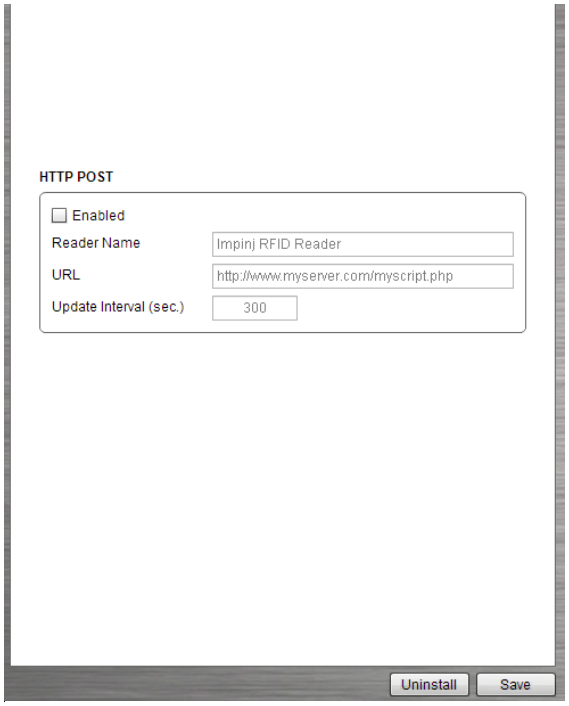
Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [13]

Figura 70. Advanced GPIO.



Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [13]

Figura 71. Servicio Web.



Provisto por IMPINJ. Plataforma Speedway Revolution. Tomado de [13].