



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

**“IMPLEMENTACIÓN DE UNA PERCHA INTELIGENTE PARA PUNTOS**  
**DE VENTA EN BOUTIQUES VINCULADAS CON EL INTERNET DE LAS**  
**COSAS (IoT)”**

**Trabajo de titulación**

**Tipo:** Propuesta Tecnológica

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES**  
**INDUSTRIALES**

**AUTOR:** ALEX JAVIER ICHINA LÓPEZ

**DIRECTOR:** ING. EDWIN VINICIO ALTAMIRANO SANTILLÁN

Riobamba – Ecuador

2020

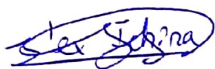
©2019, Alex.Javier Ichina López

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Alex Javier Ichina López, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

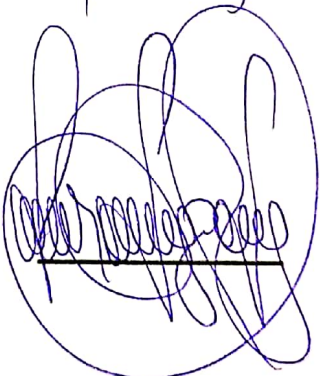
Riobamba, 22 de Julio del 2020.



**Alex Javier Ichina López**  
**C.I: 180470184-3**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

El Tribunal de Trabajo de Titulación certifica que: El trabajo de titulación: Tipo: Propuesta Tecnológica: "IMPLEMENTACIÓN DE UNA PERCHA INTELIGENTE PARA PUNTOS DE VENTA EN BOUTIQUES VINCULADAS CON EL INTERNET DE LAS COSAS (IoT)" de responsabilidad del señor Alex Javier Ichina López, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

	FIRMA	FECHA
Dr. Geovanny Estuardo Vallejo Vallejo. <b>PRESIDENTE DEL TRIBUNAL</b>	 <hr/>	2020-07-22
Ing. Edwin Vinicio Altamirano Santillán Msc. <b>DIRECTOR DEL TRABAJO DE</b> <b>TITULACIÓN</b>	 <hr/>	2020-07-22
Ing. Jorge Luis Paucar Samaniego Msc. <b>MIEMBRO DEL TRIBUNAL</b>	 <hr/>	2020-07-22

## **DEDICATORIA**

A Dios por brindarme paciencia y sabiduría para culminar con éxito mis metas propuestas. A mis padres Víctor y María Isabel, mis abuelitos Luis y María (+), mis tías Elena y Rosita, mis tíos, mis primos y primas por su confianza y apoyo incondicional en todo momento. A mis hermanos Danilo y Alejandra, por sus consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Al Ing. Edwin Vinicio Altamirano, director del trabajo de titulación, por su amistad, guía y asesoramiento en la realización de la misma. Al Ing. Jorge Luis Paucar, miembro del tribunal de titulación, por su experiencia y guía en la realización del proyecto. A la Escuela Superior Politécnica de Chimborazo por haberse convertido en mi segundo hogar y a mis Maestros por guiarme en el desarrollo de mi formación profesional.

Alex

## **AGRADECIMIENTO**

A mis padres Víctor y María Isabel, por su confianza, apoyo, cuidado y enseñarme el valor de luchar día a día para poder cumplir mis metas. A mi tía Elena y a toda mi familia, por brindarme apoyo día a día en el transcurso de mi carrera universitaria.

Alex

## TABLA DE CONTENIDO

<b>DEDICATORIA</b> .....	v
<b>AGRADECIMIENTO</b> .....	vi
<b>ÍNDICE DE TABLAS</b> .....	xi
<b>ÍNDICE DE FIGURAS</b> .....	xii
<b>ÍNDICE DE ANEXOS</b> .....	xv
<b>RESUMEN</b> .....	xvii
<b>SUMMARY</b> .....	xviii

### CAPÍTULO I

<b>1. INTRODUCCIÓN</b> .....	1
<b>1.1 Planteamiento del Problema</b> .....	1
<b>1.2 Justificación del proyecto</b> .....	3
<b>1.3 Objetivos</b> .....	4
<b>1.3.1 Objetivo General</b> .....	4
<b>1.3.2 Objetivos Específicos</b> .....	4
<b>1.4 Alcance</b> .....	4

### CAPÍTULO II

<b>2. MARCO TEÓRICO</b> .....	6
<b>2.1 Internet de las cosas (IoT)</b> .....	6
<b>2.1.1 Beneficios del Internet de las cosas</b> .....	7
<b>2.1.2 “Entornos” para aplicaciones de la IoT</b> .....	7
<b>2.1.3 Componentes básicos de un Sistema IoT</b> .....	8
<b>2.2 Punto de venta – Retail</b> .....	9

2.2.1	<i>Tecnologías aplicadas a puntos de venta</i> .....	10
2.3	<b>Plataforma LabVIEW</b> .....	11
2.3.1	<i>Recursos para la creación de programas</i> .....	11
2.4	<b>Red Inalámbrica de Sensores (WSN)</b> .....	12
2.4.1	<i>Radiofrecuencia</i> .....	13
2.4.1.1	<i>Módulos de radiofrecuencia</i> .....	14
2.5	<b>Tarjeta Arduino</b> .....	14
2.6	<b>Microcontroladores</b> .....	15
2.7	<b>Raspberry</b> .....	16
2.8	<b>Video Wall</b> .....	17

### CAPITULO III

3.	<b>METODOLOGÍA</b> .....	18
3.1	<b>Planteamiento de requerimientos específicos</b> .....	18
3.2	<b>Determinación de Hardware y Software</b> .....	20
3.2.1	<i>Hardware - Comunicación Inalámbrica</i> .....	20
3.2.2	<i>Hardware &amp; Software – Soporte para Aplicación Informática</i> .....	22
3.2.3	<i>Hardware – Recursos Visuales</i> .....	23
3.2.4	<i>Hardware – Interfaz de adquisición</i> .....	26
3.2.5	<i>Software – Plataforma IoT</i> .....	28
3.2.6	<i>Software – Video Wall</i> .....	29
3.3	<b>Programación y configuración de software y hardware</b> .....	29
3.3.1	<i>Adecuación de emisor – receptor de RF</i> .....	29
3.3.2	<i>Adquisición de señales del conjunto emisor receptor</i> .....	30
3.3.3	<i>Monitoreo local del estado de las prendas en la percha</i> .....	32
3.3.3.1	<i>Módulo de Transmisión</i> .....	32
3.3.3.2	<i>Módulo de recepción de información</i> .....	33
3.3.3.3	<i>Desarrollo de Interfaz gráfica para el monitoreo local</i> .....	34



<b>3.3.4</b>	<b><i>Monitoreo remoto del estado de prendas en percha en la plataforma IoT</i></b> .....	35
<b>3.3.5</b>	<b><i>Configuración e implementación del Video Wall</i></b> .....	41
3.3.5.1	<i>Configuración de la red</i> .....	43
3.3.5.2	<i>Configuración de la Tarjeta Raspberry</i> .....	44
3.3.5.3	<i>Instalación de las librerías en Python</i> .....	47
3.3.5.4	<i>Configuración de Tarjetas Esclavos</i> .....	47
3.3.5.5	<i>Test de comunicación Raspberry Master - Esclavos</i> .....	50
<b>3.3.6</b>	<b><i>Vinculación Piwall con la percha inteligente</i></b> .....	51
3.3.6.1	<i>Lectura de señales de control para el Piwall en el Master</i> .....	51
3.3.6.2	<i>Envío de señales para el control del Piwall desde LabVIEW</i> .....	53
<b>3.3.7</b>	<b><i>Vinculación de la percha inteligente al IoT</i></b> .....	56
<b>3.3.8</b>	<b><i>Desarrollo de la aplicación para la gestión de la Boutique</i></b> .....	58
3.3.8.1	<i>Diseño de la base datos</i> .....	59
3.3.8.2	<i>Vinculación de la Base de Datos con LabVIEW</i> .....	61
<b>3.4</b>	<b><i>Implementación y pruebas de funcionalidad</i></b> .....	68
<b>3.4.1</b>	<b><i>Circuito electrónico de adquisición de datos</i></b> .....	68
<b>3.4.2</b>	<b><i>Circuito señales de control para la tarjeta Master del Piwall</i></b> .....	70
<b>3.4.3</b>	<b><i>Pruebas de conectividad entre los módulo emisores y receptores</i></b> .....	72
<b>3.4.4</b>	<b><i>Implementación de la red de Raspberry</i></b> .....	72
<b>3.4.5</b>	<b><i>Implementación del Piwall</i></b> .....	74
<b>3.4.6</b>	<b><i>Pruebas para verificación del funcionamiento del Piwall</i></b> .....	75
<b>3.4.7</b>	<b><i>Instalación de la aplicación para el monitoreo de la percha inteligente</i></b> .....	75
<b>3.4.8</b>	<b><i>Prueba de verificación del monitoreo local</i></b> .....	76
<b>3.4.9</b>	<b><i>Montaje del monitoreo de la percha inteligente en Ubidots</i></b> .....	77
<b>3.4.10</b>	<b><i>Pruebas de verificación del monitoreo remoto</i></b> .....	77
<b>3.4.11</b>	<b><i>Aplicación informática Bella Boutique</i></b> .....	78
<b>3.4.12</b>	<b><i>Medición de corriente para determinación del consumo del sistema</i></b> .....	80

## CAPITULO IV

<b>4.</b>	<b>GESTIÓN DEL PROYECTO</b> .....	83
<b>4.1</b>	<b>Cronograma</b> .....	83
<b>4.2</b>	<b>Recursos y materiales: humanos, equipos, financiamiento</b> .....	83
<b>4.2.1</b>	<i>Costos</i> .....	83
<b>4.2.2</b>	<i>Talento Humano</i> .....	84
<b>4.2.3</b>	<i>Recursos Materiales</i> .....	84
	<b>CONCLUSIONES</b> .....	85
	<b>RECOMENDACIONES</b> .....	86
	<b>GLOSARIO</b>	
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-2:</b> Entornos para aplicaciones de IoT.....	8
<b>Tabla 2-3:</b> Entornos para aplicaciones de IoT.....	18
<b>Tabla 3-3:</b> Requerimientos de hardware para instalación de LabVIEW.....	23
<b>Tabla 4-3:</b> Características de la Raspberry Pi3 .....	25
<b>Tabla 5-3:</b> Características de los monitores a usarse .....	26
<b>Tabla 6-3:</b> Características plataforma Arduino UNO .....	27
<b>Tabla 7-3:</b> Tablas para distribución de la información .....	59
<b>Tabla 8-3:</b> Resumen Pruebas de funcionalidad y respuestas del sistema.....	80
<b>Tabla 9-3:</b> Costos directos e indirectos del proyecto. ....	80
<b>Tabla 10-4:</b> Cronograma. ....	83
<b>Tabla 11-3:</b> Costos directos e indirectos del proyecto. ....	83

## ÍNDICE DE FIGURAS

<b>Figura 1-2.</b> Internet de las cosas.....	6
<b>Figura 2-2.</b> Beneficios del IoT .....	7
<b>Figura 3-2.</b> Componentes del IoT .....	9
<b>Figura 4-2.</b> Punto de venta físico de una tienda de ropa. ....	10
<b>Figura 5-2.</b> Probadores de una tienda con realidad aumentada.....	11
<b>Figura 6-2.</b> Componentes de LabVIEW – National Instrument. ....	12
<b>Figura 7-2.</b> Topologías WSN.....	13
<b>Figura 8-2.</b> Sistema de Radiofrecuencia. ....	13
<b>Figura 9-2.</b> Esquema de un Sistema electrónico. ....	14
<b>Figura 10-2.</b> Tipos de Placas Arduino. ....	15
<b>Figura 11-2.</b> Estructura de un microcontrolador. ....	16
<b>Figura 12-2.</b> Tarjeta Raspberry Pi.....	16
<b>Figura 13-2.</b> Video Wall instalado en una boutique. ....	17
<b>Figura 14-3.</b> Software libre y propietario.....	20
<b>Figura 15-3.</b> Módulos de Radio Frecuencia.....	21
<b>Figura 16-3.</b> Receptor RF – Pines de uso en la aplicación.....	21
<b>Figura 17-3.</b> LabVIEW y la conexión con dispositivos externos. ....	22
<b>Figura 18-3.</b> Características de hardware para desarrollo de la aplicación informática.....	23
<b>Figura 19-3.</b> Sistema de Video Wall comercializado.....	24
<b>Figura 20-3.</b> Raspberry Pi3 .....	25
<b>Figura 21-3.</b> Monitor.....	26
<b>Figura 22-3.</b> Módulo de Relés, Arduino.. ....	27
<b>Figura 23-3.</b> Plataforma Arduino UNO .....	28
<b>Figura 24-3.</b> Plataforma de Ubidots IoT. ....	28
<b>Figura 25-3.</b> Plataforma de Python .....	29
<b>Figura 26-3.</b> Adecuación módulos RF. ....	30
<b>Figura 27-3.</b> Pines de codificación módulos RF. ....	30
<b>Figura 28-3.</b> Diagrama de conexiones adquisición de señales del receptor al Arduino UNO ..	31
<b>Figura 29-3.</b> Programación del Arduino UNO A para lectura y envío de información. ....	32
<b>Figura 30-3.</b> Adquisición y procesamiento de información del puerto serial. ....	33
<b>Figura 31-3.</b> Diseño de Interfaz gráfica en LabVIEW.....	35
<b>Figura 32-3.</b> Creación de cuenta en Ubidots.....	35
<b>Figura 33-3.</b> Listado de dispositivos existentes en Ubidots.....	36
<b>Figura 34-3.</b> Adición de un dispositivo en Ubidots. ....	37

<b>Figura 35-3.</b> Listado de variables del dispositivo en Ubidots.....	37
<b>Figura 36-3.</b> Variables declaradas dentro del dispositivo en Ubidots.....	38
<b>Figura 37-3.</b> Creación del Dashboard en Ubidots.....	38
<b>Figura 38-3.</b> Adición recursos gráficos en Ubidots.....	39
<b>Figura 39-3.</b> Ubidots – Selección variable vinculada al indicador.....	39
<b>Figura 40-3.</b> Ubidots – Indicador anexo y vinculado a una variable en el Dashboard.....	40
<b>Figura 41-3.</b> Ubidots – Llave de enlace con otras aplicaciones, Tokens.....	40
<b>Figura 42-3.</b> Ejecución comando sudo raspi-config en el terminal.....	41
<b>Figura 43-3.</b> Configuración ajustes avanzados en la Raspberry.....	42
<b>Figura 44-3.</b> Expansión de memoria en ajustes avanzados en la Raspberry.....	42
<b>Figura 45-3.</b> Ingreso a la configuración parámetros de la red.....	43
<b>Figura 46-3.</b> Configuración parámetros de red para el Master.....	44
<b>Figura 47-3.</b> Preferencias del sistema.....	45
<b>Figura 48-3.</b> Cambio del hostname Raspberry Master.....	46
<b>Figura 49-3.</b> Ampliación de la memoria GPU.....	46
<b>Figura 50-3.</b> Edición sentencias en el archivo Piwall.....	48
<b>Figura 51-3.</b> Asignación del nombre al esclavo en el archivo potile.....	48
<b>Figura 52-3.</b> Edición archivo autoslave para asignación de posiciones de los esclavos.....	49
<b>Figura 53-3.</b> Sentencia para iniciar el sistema o el terminal con el archivo autoslave.sh.....	50
<b>Figura 54-3.</b> Sección (a) código Thonny Python.....	52
<b>Figura 55-3.</b> Sección (b) código Thonny Python.....	53
<b>Figura 56-3.</b> Escritura en el puerto Serial desde LabVIEW.....	54
<b>Figura 57-3.</b> Lectura y procesamiento del puerto serial en el Arduino UNO A.....	54
<b>Figura 58-3.</b> Diagrama de conexiones señales de control para conmutación del Piwall.....	55
<b>Figura 59-3.</b> Diagrama de conexiones plataformas Arduino UNO.....	56
<b>Figura 60-3.</b> Bloques de programación vinculación LabVIEW - Ubidots.....	57
<b>Figura 61-3.</b> Modelo entidad relación.....	60
<b>Figura 62-3.</b> Desarrollo Interfaz gráfica para ingreso de productos.....	61
<b>Figura 63-3.</b> Creación del enlace de datos.....	62
<b>Figura 64-3.</b> Enlace e incrustación de objetos para base de datos (OLE DB).....	62
<b>Figura 65-3.</b> Asignación del .mdb en el OLE DB.....	63
<b>Figura 66-3.</b> Recursos librería Database.....	63
<b>Figura 67-3.</b> Bloque de programación – proceso de visualización.....	64
<b>Figura 68-3.</b> Bloque de programación – proceso de búsqueda.....	65
<b>Figura 69-3.</b> Bloque de programación – Mensajes para el usuario.....	67
<b>Figura 70-3.</b> Bloque de programación – proceso de búsqueda.....	67
<b>Figura 71-4.</b> Circuito electrónico del sistema de percha inteligente.....	69

<b>Figura 72-4.</b> Circuito electrónico del sistema .....	69
<b>Figura 73-4.</b> Circuito electrónico del sistema .....	70
<b>Figura 74-4.</b> Circuito electrónico del sistema .....	71
<b>Figura 75-4.</b> Conjunto electrónico de adquisición, procesamiento y envío de información.....	71
<b>Figura 76-4.</b> Conexión en red del conjunto de Raspberry .....	72
<b>Figura 77-4.</b> Implementación de la Red de Raspberry.....	73
<b>Figura 78-4.</b> Implementación de la Red de Raspberry.....	73
<b>Figura 79-4.</b> Diagrama de integración de monitores a la Red de Raspberry .....	74
<b>Figura 80-4.</b> Adición de monitores al sistema .....	74
<b>Figura 81-4.</b> Imagen dividida en el Piwall.....	75
<b>Figura 82-4.</b> Interfaz gráfica para el monitoreo de la percha inteligente .....	76
<b>Figura 83-4.</b> Interfaz gráfica para el monitoreo de la percha inteligente .....	76
<b>Figura 84-4.</b> Interfaz gráfica para el monitoreo remoto desde Ubidots .....	77
<b>Figura 85-4.</b> Interfaz gráfica para el monitoreo remoto desde Ubidots .....	77
<b>Figura 86-4.</b> Aplicación informática pantalla de inicio .....	78
<b>Figura 87-4.</b> Aplicación informática pantalla de ingreso parámetros de productos.....	79
<b>Figura 88-4.</b> Aplicación informática producto implementado.....	79
<b>Figura 89-4.</b> Medición de corrientes de consumo Módulo de Relé y Receptores RF.....	81
<b>Figura 90-4.</b> Sistema electrónico e interfaces de monitoreo local y remoto .....	81
<b>Figura 91-4.</b> Sistema Video Wall.....	82

## INDICE DE GRÁFICOS

**Gráfico 1-3.** Diagrama de bloque de los requerimientos para ejecución del prototipo ..... 19

## ÍNDICE DE ANEXOS

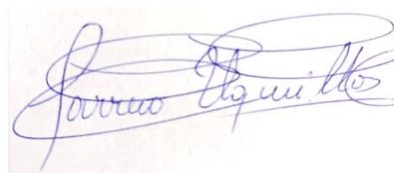
<b>ANEXO A:</b> DATASHEET ARDUINO UNO.....	5
<b>ANEXO B:</b> DATASHEET RASPBERRY PI3 .....	13
<b>ANEXO C:</b> DATASHEET EMISOR RECEPTOR RF .....	21
<b>ANEXO D:</b> PROGRAMACIÓN ARDUINO UNO A.....	23
<b>ANEXO E:</b> PROGRAMACIÓN ARDUINO UNO B .....	24
<b>ANEXO F:</b> PROGRAMACIÓN THONNY PYTHON .....	25
<b>ANEXO G:</b> PROGRAMACIÓN COMPLETA EN LABVIEW VI MONITOREO .....	26
<b>ANEXO H:</b> PROGRAMACIÓN COMPLETA EN LABVIEW VI DE GESTION DE INFORMACIÓN .....	1
<b>ANEXO I:</b> REGISTRO DE EXPERIMENTACIÓN Y CÁLCULO DE EFICIENCIA DEL SISTEMA .....	1



## RESUMEN

El presente trabajo describe la elaboración de una percha inteligente para puntos de venta en boutiques. Se planteó dotarla de una autonomía que interactúe directamente con el cliente, asignar recursos visuales que realcen la interacción con el entorno y se relacione con el IoT para su monitoreo. Se utilizó una red inalámbrica de sensores basada en radiofrecuencia, donde para conocer el estado de una prenda dentro de la percha, si está colgada o la levantó el cliente. Se adaptó a cada gancho de ropa un sensor con una señal digital, siendo este el emisor y por medio de los receptores se gestiona un banco de relés A, donde el estado de sus contactos abiertos son procesados como señales de entrada en un Arduino A, que se encuentra vinculado mediante comunicación serial a una aplicación informática desarrollada en LabVIEW, el cual por medio de interacciones visuales permite el monitoreo local de la percha y a su vez proporciona información a la plataforma de soporte IoT - Ubidots para un monitoreo remoto, adicionalmente en LabVIEW se gestiona la retroalimentación de señales hacia el Arduino A para que éste proporcione señales al banco de relés B, las mismas son receptadas por un segundo Arduino B que está enlazado a una Raspberry Pi3 en calidad de Master a un sistema de Video Wall formado por 4 Raspberry Pi3 enlazados al Master mediante protocolo TCP/IP. La programación y configuración de los elementos para la generación del Video Wall se elaboró en Python, la aplicación cuenta también con un sistema de gestión de productos, inventarios y ventas. En conclusión, se obtuvo un sistema robusto flexible a adaptarse a otros entornos con un porcentaje del 98% en eficiencia en pruebas de funcionalidad. Se recomienda un sistema de alimentación constante y uno de refrigeración adecuado para las Raspberry.

**Palabras clave:** <PERCHA INTELIGENTE>, <INTERNET DE LAS COSAS>, <PIWALL>, <LABVIEW (SOFTWARE)>, <BASE DE DATOS>, <RASPERRY (TARJETA ELECTRÓNICA)>, <PYTHON (SOFTWARE)>



04-05-2020

## SUMMARY

The present work describes the elaboration of an intelligent hanger for points of sale in boutiques. It was proposed to provide it with an autonomy that interacts directly with the client, assign visual resources that enhance interaction with the environment and relate to the IoT for monitoring. A radio frequency-based wireless sensor network was used, where to know the state of a garment inside the hanger, if it is hung or lifted by the client. A sensor with a digital signal was adapted to each clothes hook, this being the emitter and through the receivers a bank of relays A is managed, where the state of its open contacts are processed as input signals in an Arduino A, which is linked by serial communication to a computer application developed in LabVIEW, which through visual interactions allows local monitoring of the hanger and in turn provides information to the IoT support platform for remote monitoring, additionally in LabVIEW signal feedback to the Arduino A is managed so that it provides signals to relay bank B, they are received by a second Arduino B that is linked to a Raspberry Pi3 as a Master to a Video Wall system consisting of 4 Raspberry Pi3 linked to the Master using TCP / IP protocol. The programming and configuration of the elements for the generation of the Video Wall was made in Python, the application also has a product, inventory and sales management system. In conclusion, a flexible robust system was obtained to adapt to other environments with a 98% efficiency percentage in functionality tests. A constant power supply and a suitable cooling system are recommended for Raspberrys.

**Keywords:** <INTELLIGENT HANGER>, <INTERNET OF THINGS>, <PIWALL>, <LABVIEW (SOFTWARE)>, <DATABASE>, <RASPBERRY (ELECTRONIC CARD)>, <PYTHON (SOFTWARE)>

# CAPÍTULO I

## 1. INTRODUCCIÓN

El progreso e implementación de los sistemas automatizados, hoy en día ha dispuesto la necesidad de comunicación, envío y recepción de datos e información en tiempo real para tomar acciones sobre los mismos en determinados procesos locales o industriales. A través del desarrollo de la Industria 4.0 y el uso de robots en diversas actividades han dado resultados positivos, por ejemplo: a través de la digitalización de los procesos, mediante el uso de sensores y sistemas de información como herramienta de ayuda para la productividad. Los avances tecnológicos, digitalización y automatización de procesos hoy en día se inician e implementan rápidamente, si no nos acoplamos bien a ello. No existirá un correcto proceso de aceptación, como es el caso del Internet de las cosas (IoT), que es una consolidación a través de una red de redes la cual aloja una gran cantidad de dispositivos y pueden interconectarse con interfaces de conexión simples tales como electrodomésticos, dispositivos tecnológicos, hasta torres de edificios con capacidad de comunicación. (López Garzón y Cárdenas López, 2019, p. 75)

Debido a la importancia que han tomado las redes inalámbricas y todo lo que con ellas se relaciona, nace el interés de analizar una temática importante, como son las Redes de Sensores Inalámbricos (WSN), cuya finalidad es entender el mecanismo de monitorización y control de los fenómenos que ocurren en el mundo físico. (Córdoba y Santamaría Buitrago, 2013, p. 5)

Siendo el caso de las perchas inteligentes, las cuales interactúan con los clientes a través del uso de la tecnología, esta nueva forma de comercialización de prendas de vestir, para que las tiendas tradicionales mediante el uso de la tecnología puedan competir con las tiendas en línea, adquiriendo productos de forma dinámica, interesante y divertida. Con esta implementación se permite ahorrar recursos humanos y materiales al momento de desarrollar la actividad comercial en beneficio de los usuarios y propietarios de estas tiendas, haciendo uso de las nuevas tecnologías, lo que contribuye a mejorar la experiencia de comprar y vender productos.

### 1.1 Planteamiento del Problema

En la actualidad, la sociedad se expresa de manera digital, se considera en un apogeo la aplicación de tecnologías contactless como: la comercialización electrónica, los métodos de pago

simplificados, entre otros procesos, debido a que facilitan la vida cotidiana de los seres humanos. No obstante, en referencia a la recolección de información del proceso que llevan a cabo las empresas actuales es difícil, si no imposible, porque al visualizar los diferentes tipos de información en el mismo ordenador, compilarlos y sincronizarlos dentro de un informe que sirva como una base inteligente de recursos para la toma de decisiones. Incluso cuando los sistemas son accesibles desde el mismo terminal, éste normalmente requiere de diferentes programas que usan distintos lenguajes. Lo ideal y lo que las organizaciones requieren, es crear una plataforma que cubra todas sus necesidades a la hora de realizar este tipo de tareas, integrando tanto la información que se recibe en tiempo real, como la información estática necesaria, ofreciendo interfaces que permitan acceder de forma remota a todos los servicios de la plataforma. (Fernández Martínez et al., 2009, p. 15).

Una aplicación del uso de las nuevas tecnologías en el mercado está plasmado hoy en día en la forma tradicional de venta en una tienda, las cuales van sufriendo sus estrategias de mercadeo por tecnologías digitales, exportando al mundo físico las mejores ventajas del mundo online, proporcionando al consumidor todas las ventajas y comodidades para realizar su compra a través del uso de “plataformas” que se refieren a una forma específica de crear valor, aprovechando tanto la tecnología como los negocios; y denota las elecciones específicas que gerentes y empresarios hacen en ambas dimensiones, reforzándose mutuamente. Una característica fundamental de las plataformas es la presencia de los denominados efectos de red: a medida que más usuarios interactúan con ella, esta se vuelve más atractiva para potenciales nuevos usuarios. (Gawer et al., 2019, p. 8).

En el caso de las boutiques de ropa las cuales expone sus prendas de temporada, con el limitante que la prenda a ser comercializada es exhibida por un tiempo considerable llegando a acumular una colección específica en la sección dedicada para la exposición de una determinada prenda, limitando así la exhibición de más colecciones de temporada, traduciéndose en una pérdida de tiempo y recursos financieros para las demás prendas que esta destinadas para el comercio. Una solución para esta problemática ha sido el diseño y utilización de un Avatar 3D, el cual mediante un modelo tridimensional a través de una grabación sencilla por un teléfono móvil hacia un objeto o persona y a través de software se realiza una reconstrucción tridimensional. En el campo profesional podría usarse para el peritaje remoto, tasaciones de casas, simuladores de cirugía, de vuelo, nuevas operaciones quirúrgicas, virtual shopping, diseño industrial o turismo. (García de Paso, 2011, p. 29).

Por lo tanto el proyecto planteado, pretende desarrollar un sistema de percha inteligente para puntos de venta en una boutique, siendo que esto no representa limitación para que se lo pueda replicar para entornos diferentes de comercialización. El valor agregado del sistema propuesto radica en el desarrollo de una herramienta informática enlazada con la percha inteligente para

manejo de recursos visuales locales y remotos conectada al internet de las cosas (IoT), permitiendo de esta manera ayudar a los procesos operativos y a pronosticar el comportamiento del cliente para la toma de decisiones y habilidades de marketing para la mejora en la rentabilidad.

## **1.2 Justificación del proyecto**

La propuesta se fundamenta en la tecnificación de un punto de venta considerando la situación actual del mundo globalizado, donde los consumidores cada vez se encuentran más conectados y se facilita el acceso a cualquier tipo de producto o servicio a través de una red. Se considera como caso específico la generación de un prototipo de percha inteligente para un punto de venta en boutiques, bajo la percepción de que, a pesar de poder tener total acceso a las tiendas de manera virtual, las personas aún visitan tiendas físicas, en el placer de la experiencia que esto representa.

La imagen en un local comercial es influyente en la preferencia del cliente y en el caso de instalaciones eléctricas con cables expuestos a la vista resulta incómodo. Es por eso que, se pretende el uso de tecnologías de comunicación inalámbrica para direccionar la información a una aplicación informática desarrollada en LabVIEW, la misma que dispone de una interfaz gráfica que permite el monitoreo del estado de las variables definidas dentro del sistema como recursos para interactuar con el usuario.

Distintos expertos en marketing afirman que el cliente tiene preferencia a comprar si los productos tienen gran exposición y visualización, por el cual se pretende convertir una percha normal en una percha inteligente que pueda interactuar con el cliente de manera que si éste retira una prenda de vestir de la percha, el sistema refleje en una pantalla datos del producto seleccionado conjugado con recursos visuales (multimedia – imagen – video) que permiten observar cómo luce la prenda en una persona (modelo).

El manejo de la información en un local comercial resulta necesario, muchas de las veces esta responsabilidad se la asigna a una persona siendo que el ser humano es más propenso al error en comparación con un sistema automatizado para el manejo de datos. Se considera la implementación de una base de datos en la que se maneja toda la información de inventarios, control de existencias y ventas que se genera en el entorno específico de una boutique.

En síntesis, el proyecto pretende innovar con una propuesta de percha inteligente para un punto de venta en boutiques, que sirva como base para relacionarlo en otros entornos comerciales inmersos en el mundo y uso de la tecnología, así como su monitoreo.

## **1.3 Objetivos**

### ***1.3.1 Objetivo General***

Implementar una percha inteligente para puntos de venta en boutiques vinculada con el Internet de las cosas (IoT)

### ***1.3.2 Objetivos Específicos***

Determinar los requerimientos funcionales que debe cumplir una percha en puntos de venta en una boutique para definirla como inteligente.

Generar una base de datos para el control del inventario, existencias y venta de productos en una boutique, dos pantallas las cuales interactúen con el comprador y al menos cuatro productos seleccionados.

Desarrollar, empleando LabVIEW, una aplicación que relacione la administración de la base de la base de datos y gestione recursos para la percha, pantallas inteligentes y enlace al sistema IoT.

Establecer el hardware y software necesarios para la adquisición y vinculación de la información de la percha inteligente a la aplicación desarrollada en LabVIEW.

Evaluar el cumplimiento de las especificaciones planteadas para el manejo del prototipo implementado.

## **1.4 Alcance**

El trabajo de titulación está encaminado a desarrollar un sistema de percha inteligente para puntos de venta en boutiques en base a los requerimientos funcionales que se definen para dotarla de independencia y autonomía. Se relaciona la boutique con prendas de vestir exhibidas sobre una percha, donde estarán sujetas mediante ganchos y se planteó la adecuación para la interacción con el cliente. Se emite inalámbricamente una señal de aviso cuando la prenda se encuentra colgada o fue retirada de la percha y se limitó la prueba a cuatro prendas. Se presenta una aplicación informática desarrollada en la plataforma LabVIEW que cumple con varias funciones, como: gestionar el uso de una base de datos a través de una interfaz gráfica para el manejo de existencias por el registro de inventarios y ventas de productos para el propietario de la tienda. Permite el

monitoreo local del estado de las prendas sobre la percha y realiza el enlace para un monitoreo remoto a través de una plataforma de soporte del internet de las cosas (IoT). Gestiona los recursos visuales (Video Wall con dos pantallas) que interactúan con el comprador y cuatro productos seleccionados.

## CAPÍTULO II

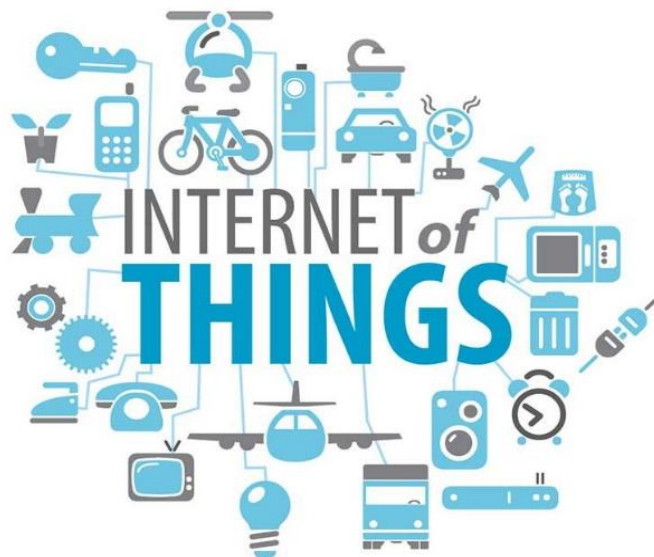
### 2. MARCO TEÓRICO

En este capítulo se hace referencia a los conceptos teóricos que servirán de fundamento para entender la tecnología del internet de las cosas (IoT), sus beneficios y campos de aplicación. También se contempla la conceptualización general del hardware y software que se empleó en el desarrollo del proyecto.

#### 2.1 Internet de las cosas (IoT)

Cisco Internet Business Solutions Group (IBSG) afirma que “el Internet de las cosas (IoT) es sencillamente el punto en el tiempo en el que se conectan a Internet más cosas u objetos que personas”. (Evans Dave, 2011, p. 1).

IoT es una infraestructura de red global dinámica con capacidades de autoconfiguración basadas en protocolos de comunicación estándar, donde abarca tecnologías de comunicación inalámbrica como son las redes de sensores inalámbricos (WSN), el cual están constituidos por sensores autónomos que son capaces de comunicarse entre sí con un mínimo de consumo de energía para la monitorización y control de los dispositivos. (Llaneza González, 2018, p. 20). En la Figura 1-2 se muestra que todo objeto o casa puede anclarse al internet de las cosas.



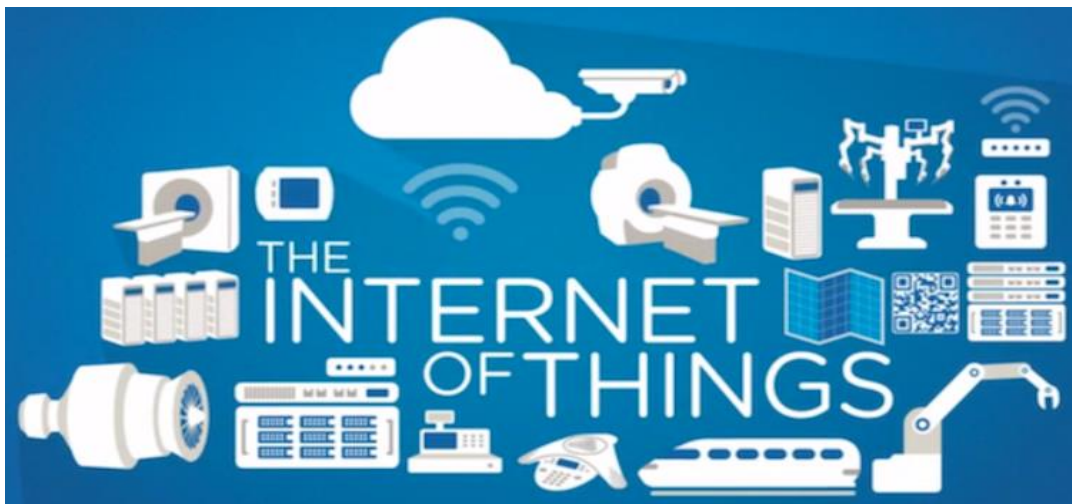
**Figura 1-2.** Internet de las cosas.

Fuente: <https://bit.ly/2Sz5Zia>.



### ***2.1.1 Beneficios del Internet de las cosas***

La primera consecuencia directa de la IoT es la generación de grandes cantidades de datos, donde cada objeto físico o virtual conectado a IoT puede tener una doble digital en la nube, que podría generar actualizaciones periódicas en tiempo real. Como resultado, el volumen de mensajería relacionada con IoT podría alcanzar entre 1.000 y 10.000 mensajes por persona al día. No hace falta decir, que esto es una gran cantidad de mensajes que es necesario gestionar. (Bonilla Fabela et al., 2015, p. 2316). La figura 2-2 permite observar la aplicación del IoT en varios campos de la ingeniería: sector automotriz e industrial.



**Figura 2-2.** Beneficios del IoT

Fuente: <https://bit.ly/2Q46xer>.

### ***2.1.2 “Entornos” para aplicaciones de la IoT***

El uso de la tecnología IoT se encuentra aún en desarrollo, en un futuro próximo se avecina el uso de dicha tecnología en los objetos cotidianos que nos rodean. Dichos objetos dispondrán de una conexión a internet los mismos que proporcionarían una gran cantidad de información brindando a los fabricantes el uso de servicios y productos predictivos que mejoren la vida de la sociedad moderna, el resultado será un mundo hiper conectado, que no impondrá limitaciones a las aplicaciones o servicios que pueden hacer uso de la tecnología. (Karen Rose y Lyman Chapin, 2015, pp. 15-16). La tabla 1-2 enuncia varios campos o entornos en los que se aplica el IoT y se citan algunos ejemplos.

**Tabla 1-2:** Entornos para aplicaciones de IoT.

ENTORNO	EJEMPLOS
CUERPO HUMANO	Dispositivos unidos al cuerpo humano o colocado dentro del mismo.
HOGAR	Edificios de vivienda Controladores y sistemas de seguridad para el hogar.
PUNTOS DE VENTA	Espacios comerciales: Tiendas, bancos, restaurantes, estadios, cualquier lugar donde los consumidores consideren y compren; sistemas de autopago, ofertas en compras presenciales, optimización del inventario.
OFICINAS	Espacios donde operan trabajadores del conocimiento. Gestión de la energía y la seguridad en los edificios de oficinas; mejora de la productividad, incluso para los empleados móviles.
FÁBRICAS	Entornos de producción estandarizados. Lugares con rutinas de trabajo repetitivas, como hospitales y granjas; eficiencia operativa, optimización del uso de los equipos y el inventario.
OBRAS	Entornos de producción a medida, minería, petróleo y gas, construcción; eficiencia operativa, mantenimiento predictivo, salud y seguridad.
VEHÍCULOS	Sistemas dentro de vehículos en movimiento. Vehículos, incluyendo automóviles, camiones, barcos, aviones y trenes; mantenimiento basado en la condición, diseño, basado en el uso, análisis de preventa.
CIUDADES	Entornos urbanos Espacios públicos e infraestructura en entornos urbanos; sistemas de control adaptativo de tráfico, contadores inteligentes, monitoreo ambiental, gestión de recursos.
EXTERIORES	Entre entornos urbanos (y fuera de otros entornos) Los usos exteriores incluyen las vías de ferrocarril, los vehículos autónomos (fuera de los centros urbanos) y la navegación aérea; el enrutamiento en tiempo real, la navegación conectada, el seguimiento de envíos.

Fuente: (KAREN ROSE ET ALL, 2015)

### 2.1.3 Componentes básicos de un Sistema IoT

- **Sensores:** combinan flujos de datos provenientes de sensores virtuales o dispositivos de detección física.
- **La infraestructura de computación en la nube:** permite el almacenamiento de la información sensor, actuando, así como una base de datos en la nube. La infraestructura de la

nube también almacena metadatos para los diversos servicios, como parte del proceso de programación.

- **El servicio de directorio:** almacena información sobre todos los sensores disponibles en la plataforma IoT abierta.
- **El Planificador Global:** procesa todas las solicitudes de despliegue de servicios a pedido y garantiza su acceso adecuado a los recursos que requieren
- **El componente de configuración y monitoreo:** permite las funcionalidades de administración y configuración sobre los sensores, y los servicios IoT que se implementan dentro de la plataforma. (Buyya y Dastjerdi, 2016, pp. 33-34)

La Figura 3-2 representa la estructura de un sistema con IoT con sus componentes.



**Figura 3-2.** Componentes del IoT

Fuente: <https://bit.ly/2MDegOq>

## 2.2 Punto de venta – Retail

El concepto del retailing, es una orientación gerencial que se concentra en conocer las necesidades de su mercado objetivo y satisfacerlas eficiente y efectivamente. Aun tratándose de un modelo exitoso, el retailer tendrá que definir siempre nuevas estrategias para mantener a sus clientes. La estrategia modular de un retailer es construir una ventaja competitiva sostenible en el tiempo en comparación con otras empresas del ramo. Esto significa que el retail deberá construir una muralla

lo suficientemente alta con la finalidad de mantener a sus clientes o ser líder en su mercado objetivo. Es usual que, dentro de esta estrategia, los retailers mantengan:

- a) Lealtad del cliente al retailer.
- b) Ubicación idónea al mercado objetivo.
- c) Equipo gerencial altamente calificado.
- d) Sistemas informáticos de clase mundial.
- e) Productos únicos.
- f) Estrecha colaboración con sus proveedores.
- g) Servicio al cliente. (Guerrero-Martínez, 2012, pp. 194-195)

La figura 4-2 representa el entorno de una tienda física que carece de recursos tecnológicos.



**Figura 4-2.** Punto de venta físico de una tienda de ropa.

Fuente: <https://bit.ly/3508Weg>

### ***2.2.1 Tecnologías aplicadas a puntos de venta***

La selección del tipo de tecnología va orientada hacia el usuario que se pretende alcanzar. La realidad aumentada hace referencia a la visión desde un dispositivo tecnológico (ordenador, Smartphone, Tablet, videoconsola) del mundo físico mezclado con imágenes virtuales. La realidad aumentada añade contenidos virtuales a la visualización del mundo real, la mayoría del

software creado utiliza imágenes bidimensionales superpuestas al usuario. (Meca López, 2014, pp. 5-6).

La figura 5-2 indica el funcionamiento de la realidad aumentada en una tienda física.



**Figura 5-2.** Probadores de una tienda con realidad aumentada.

Fuente: <https://bit.ly/39pQJKK>

## 2.3 Plataforma LabVIEW

LabVIEW es una herramienta de programación gráfica. Originalmente este programa estaba orientado para aplicaciones de control de equipos electrónicos usados en el desarrollo de sistemas de instrumentación, lo que se conoce como instrumentación virtual. (Vizcaíno y Sebastián, 2011, p. 22)

Cada uno de los programas creados en LabVIEW se almacenará en ficheros llamados VI (Virtual Instrument), aunque en la actualidad se han integrado en una librería llamada VISA, en donde se construye un interface de usuario, o panel frontal, con controles e indicadores (Vizcaíno y Sebastián, 2011, p. 22).

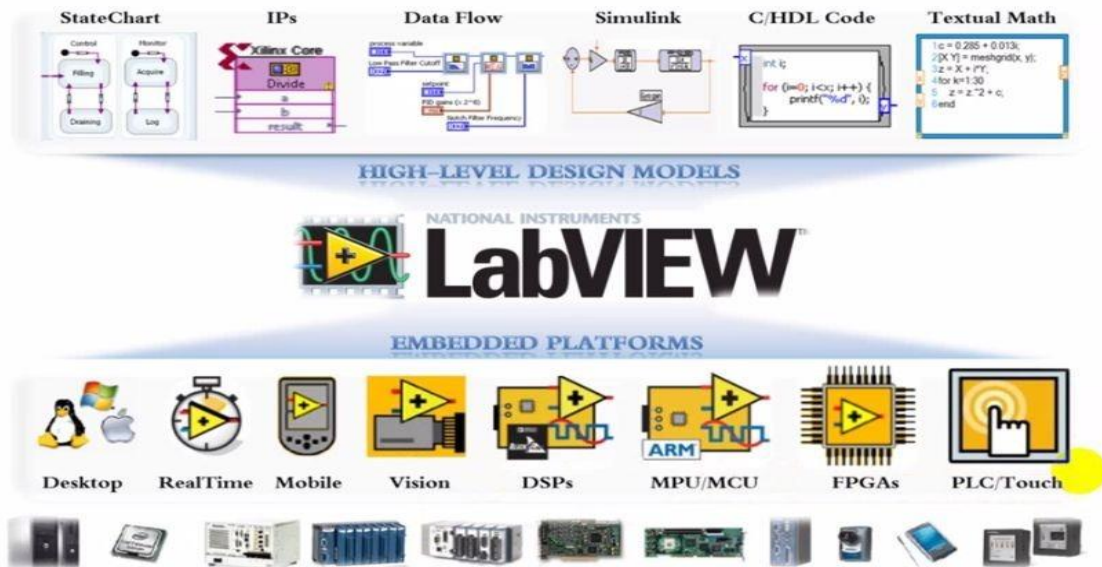
### 2.3.1 Recursos para la creación de programas

En LabVIEW la programación se realiza en el Diagrama de Bloques. Un Programa habitualmente está formado por:

- **Controles:** sirven de entrada para los datos.
- **Funciones, VI y estructuras:** realizan una o varias operaciones con esos datos.
- **Indicadores:** sirven de salida para los datos.

Los datos ‘circulan’ por el programa mediante cables, que sirven para unir unos elementos con otros mediante la herramienta Connect Wire. (Lajara y Pelegri, 2011, p. 36)

La Figura 6-2 muestra varios de los componentes que dispone LabVIEW adaptables a varias soluciones.



**Figura 6-2.** Componentes de LabVIEW – National Instrument.

Fuente: <https://bit.ly/2u2kAsn>

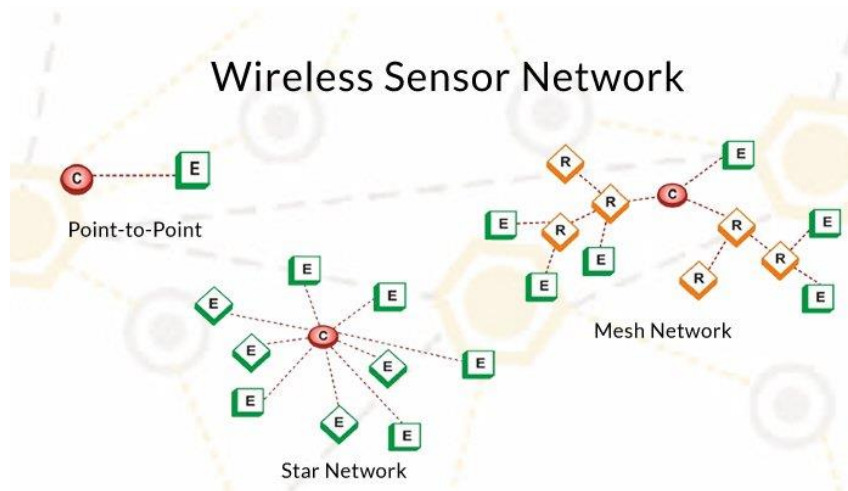
## 2.4 Red Inalámbrica de Sensores (WSN)

Se define como WSN (Wireless Sensor Network) a una red inalámbrica de dispositivos de bajo consumo para la adquisición de información o sensoramiento de variables. A los dispositivos que forman parte de dichas redes se los conoce como nodos sensores o motas (motes) y están limitados en su capacidad computacional y de comunicación. Mediante una interacción en conjunto estos dispositivos transportan la información de un punto a otro de la red transmitiendo pequeños mensajes de un nodo a otro. (Fernández Martínez et al., 2009, p. 26)

La auto configuración y la facilidad de despliegue caracterizan este tipo de redes, pudiendo registrar datos relativos a los sensores de cada punto nodal para una comunicación broadcast y convertirse en todo momento en emisor, receptor, ofrecer servicios de encaminamiento entre nodos sin visión directa (Fernández Martínez et al., 2009, p. 26)

El desarrollo tecnológico permitió el desarrollo de ordenadores pequeños y de bajo costo que manejan comunicación inalámbrica y organización autónoma. La finalidad de estas redes es distribuir aleatoriamente estos nodos en grandes extensiones. (Fernández Martínez et al., 2009, p. 26)

La Figura 7-2 representa las formas de conexión que pueden tener las redes inalámbricas de sensores, se las denomina topologías de red.

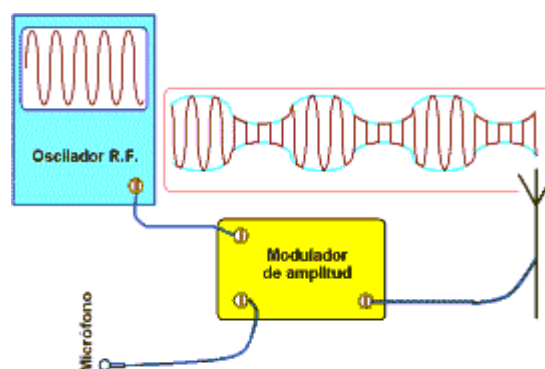


**Figura 7-2.** Topologías WSN.

Fuente: <https://n9.cl/vipl>

### 2.4.1 Radiofrecuencia

La radiofrecuencia se la define como una parte del espectro electromagnético, específicamente hace referencia a un segmento de baja energía. El enlace de comunicación se genera en el ciclo de transmisión de una señal al hacer circular corriente a través de un conductor y la recepción por medio de una antena. (Universidad de Valencia, 2018). La Figura 8-2 representa un ciclo de emisión de señales por radio frecuencia.



**Figura 8-2.** Sistema de Radiofrecuencia.

Fuente: Arduino, 2020.

Son las ondas o señales de radio que emplea la radiofrecuencia como medio o interfaz para canalizar la información y comunicar dispositivos como los conocidos walkie talkies o

extendiéndose a su uso en todos los dispositivos de comunicación existentes como la televisión, sistemas de GPS o redes móviles. (Universidad de Valencia, 2018)

#### 2.4.1.1 Módulos de radiofrecuencia

Sus aplicaciones se extienden a diferentes campos incluso con módulos ideales para la implementación de prototipos. La plataforma Arduino cuenta con módulos de radiofrecuencia basados en un conjunto emisor – transmisor.

Cuando se tiene como requerimiento realizar una aplicación de bajo costo con la que se pueda efectuar control a una distancia no muy grande, inferior a 15 metros Arduino proporciona módulos eficientes basados en chips comerciales como lo que se observa en la Figura 9-2 como son el PT2262 y el PT2272.



**Figura 9-2.** Esquema de un Sistema electrónico.

Fuente: <https://n9.cl/tyk3>

PT2262 PT2272 corresponde a la codificación de dos chips tipo codificador y decodificador respectivamente basados en tecnología CMOS, permiten una comunicación unidireccional que embebidos en módulos de la plataforma Arduino son herramientas útiles para conseguir manos a distancia. (Prometec, 2019)

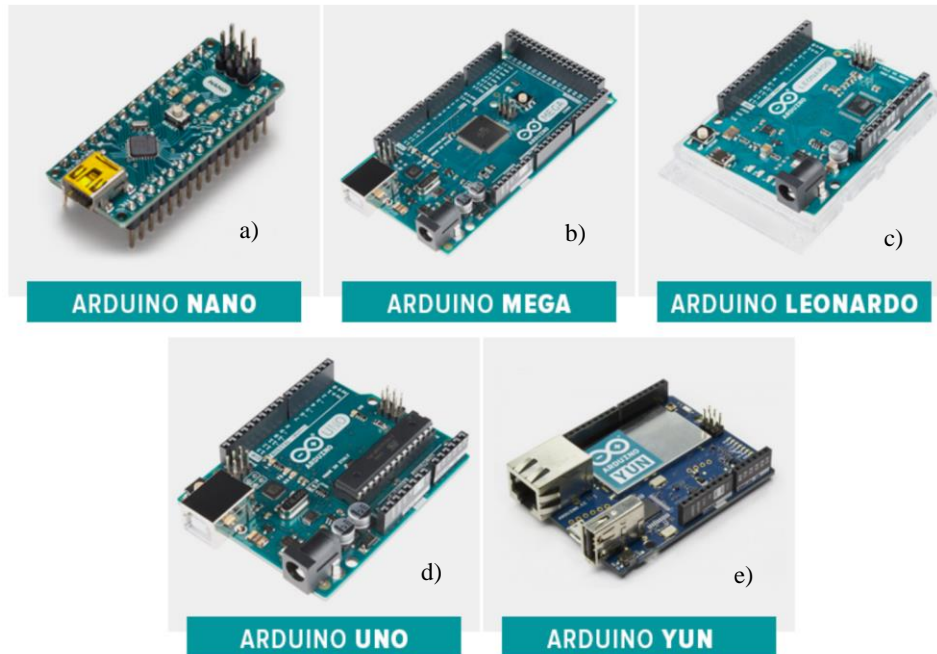
## 2.5 Tarjeta Arduino

Arduino es una plataforma embebida que contiene como elemento principal un microcontrolador reprogramable, el mismo que permite la adquisición de señales, procesamiento mediante sentencias programadas y la emisión de señales de control. Este sistema embebido además se acopla a una amplia gama de sensores de la misma marca u otra que proporcionen señales del tipo



digital y/o analógico para ser procesadas y según su evaluación por medio de interfaces de potencia poder realizar el control de actuadores sin importar la carga que manejen. (Arduino, 2020)

En la figura 10-2 se muestra las diferentes versiones o modelos de placas de Arduino que se diferencian principalmente por el número de pines digitales y analógicos, capacidad de memoria, velocidad de procesamiento, entre los parámetros principales.



**Figura 10-2.** Tipos de Placas Arduino.

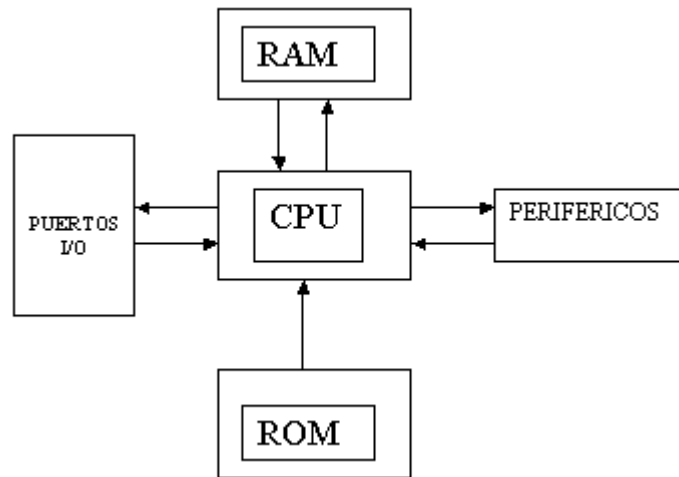
Fuente: Arduino, 2020.

## 2.6 Microcontroladores

Se define como un dispositivo electrónico embebido o un circuito integrado que encapsula varios recursos, la ventaja de este elemento que se puede programar de acuerdo a la necesidad que suscite el entorno en el que se lo va a emplear, es capaz de ejecutar con total autonomía previo a la carga de instrucciones predefinidas cualquier tipo de trabajo que se lo encomiende.

El microcontrolador en su estructura interna consta de tres partes principales que definen su capacidad para un trabajo específico, estas son el CPU, Diferentes tipos de memorias y periféricos de entradas y salidas (Artero, 2013, p. 62)

La Figura 11-2 representa la estructura interna del microcontrolador con sus principales componentes.



**Figura 11-2.** Estructura de un microcontrolador.

Fuente: SANDOROBOTICS, 2019

## 2.7 Raspberry

Raspberry Pi es una minicomputadora de la gama de hardware libre que, constituida de una placa simple empleada en el ámbito educativo para estudiantes del área de informática y electrónica principalmente, el soporte de software para este dispositivo es de código abierto. Está constituida esencialmente por un procesador, memoria RAM, puertos USB, GPU, HDMI, ETHERNET, requiere de una micro SD pues no incluye memoria interna. La potencialidad de la Raspberry Pi la hace suficientemente potente como para poder tomar el lugar de una computadora de escritorio. (A, B rev 1, B rev 2, B+). (Salcedo & Cendrós, 2006, pp. 63-64) La Figura 12-2 representa la vista física de la Raspberry modelo Pi3 B.



**Figura 12-2.** Tarjeta Raspberry Pi.

Fuente: (Raspberry PI, 2019)

Las características principales de la Raspberry son, un tamaño de 85mm x 56 mm aproximadamente con un peso Menor a 50 gramos disponibles en 4 modelos (A, B rev 1, B rev 2, B+). (Salcedo & Cendrós, 2006, pp. 63-64)

## 2.8 Video Wall

Se entiende por Video Wall como una configuración especial de pantallas o monitores profesionales sincronizados para mostrar contenidos y simular una pantalla de gran tamaño. Se lo puede crear utilizando tantos monitores o pantallas como sean necesarios para cubrir un área deseada. Se los puede diseñar en forma simétrica o no y su tamaño. (Video Wall, 2020)

El fin de la implementación de un Video Wall es dar la apariencia de una gigantesca pantalla en una gran área lo que resulta imposible hacerlo como una sola pantalla, a pesar de que se la podría fabricar, pero el costo sería y el método de transporte sería demasiado complicado. Por este motivo, un Video Wall es una excelente alternativa costo-beneficio cuando se requiere mostrar contenidos de gran tamaño. (Video Wall, 2020)

Son varios los campos de aplicación de los Video Wall como ejemplo en este caso están los utilizados con fines publicitarios y señaléticas, este tipo están diseñados para reproducir contenido multimedia provenientes de un punto administrador, son una estrategia elegante para impactar al espectador. (Video Wall, 2020)

La Figura 13-2 muestra un Video Wall instalado en una boutique



**Figura 13-2.** Video Wall instalado en una boutique.

Fuente: SANDROBOTICS, 2019

## CAPITULO III

### 3. METODOLOGÍA

El desarrollo de una percha inteligente para puntos de venta en boutiques, es el innovar dentro del mercado físico por medio de los recursos tecnológicos, el captar la atención de los clientes en la compra virtual como un recurso e inserción a las tiendas online.

Para el diseño de una percha inteligente, se partió de la revisión bibliográfica de los requerimientos generales necesarios para cumplir con el objetivo de generar un prototipo inteligente con recursos tecnológicos que lo hagan innovador y actual para obtener el interés de los clientes y propietarios.

Se establece un conjunto de etapas como recurso metodológico para la ejecución ordenada del proyecto. En la Tabla 2-3 se detalla el listado de etapas a seguir:

**Tabla 2-3:** Entornos para aplicaciones de IoT.

#Etapa	DESCRIPCIÓN
1	Planteamiento de requerimientos específicos para el prototipo
2	Determinación de hardware y software
3	Programación y configuración de software y hardware
4	Implementación recursos informáticos y físicos
5	Análisis de resultados y pruebas de funcionalidad

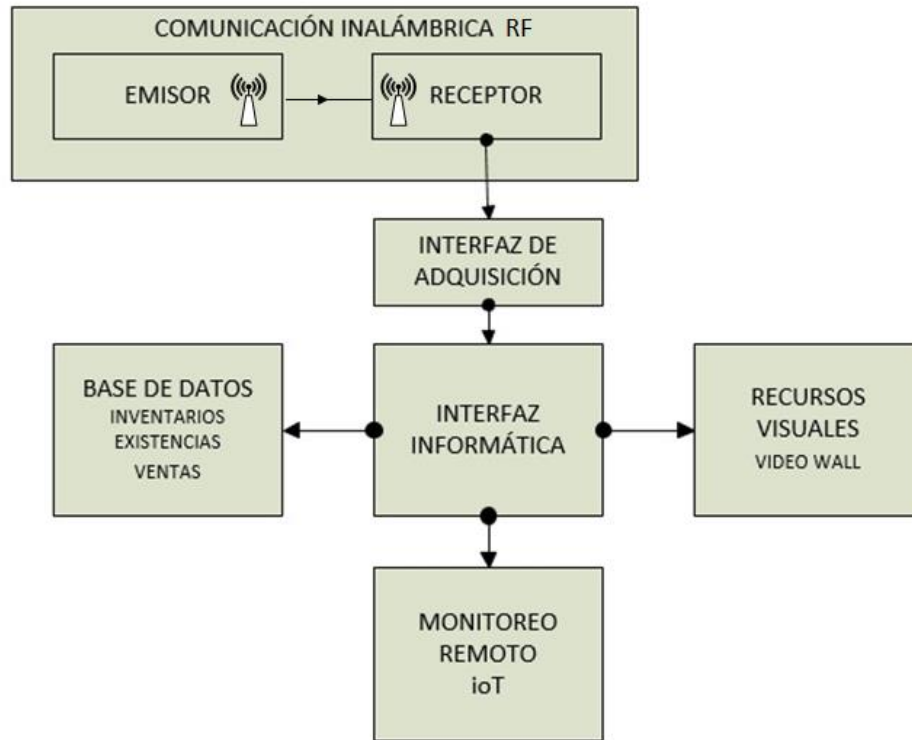
**Realizado por:** Ichina, Alex, 2020.

La ejecución de cada una de las etapas planteadas se relaciona con sub etapas que se desarrollarán a lo largo de este trabajo.

#### 3.1 Planteamiento de requerimientos específicos

Se ha formulado el proyecto para un entorno específico como lo es una boutique donde lo que se desea es generar una percha con procesos de inteligencia como estrategia para alcanzar el interés del cliente y de sus propietarios.

En el Gráfico 1-3, se muestra un diagrama de bloques en los que se asocian los requerimientos planteados para el prototipo de percha inteligente implementada como una propuesta innovadora.



**Gráfico 1-3.** Diagrama de bloque de los requerimientos para ejecución del prototipo

Realizado por: Ichina, Alex, 2020

Para el entorno en que se desarrolla el trabajo se planteó como requerimientos:

- Dotar de independencia a los ganchos que contienen la ropa en una percha de exhibición, para pruebas del prototipo se plantea realizar esta implementación a una muestra de cuatro ganchos.
- Proporcionar desde los ganchos una señal, que permita conocer si se encuentran colgados con la prenda o han sido retirados por un cliente, la comunicación de la señal es inalámbrica.
- El desarrollo de una aplicación informática que permita:

La gestión de las señales recibidas desde los ganchos (emisores).

El monitoreo del estado de los emisores a través de una interfaz gráfica.

Proporcionar información para el monitoreo remoto a través de una plataforma IoT.

Permitir el manejo de un sistema de inventarios, existencias y ventas de productos en una boutique.

- Se propuso también la inserción de recursos visuales para interacción del cliente con la percha inteligente por medio de la vinculación de un Video Wall.

### 3.2 Determinación de Hardware y Software



**Figura 14-3.** Software libre y propietario

Fuente: <https://n9.cl/u37tj>

Planteados los requerimientos que debía cumplir el sistema, se procedió con la determinación de los elementos hardware y software entre recursos libres y de propietarios disponibles. La figura 14-3 describe varios software entre licenciados y libres.

#### 3.2.1 Hardware - Comunicación Inalámbrica

Siendo los ganchos de ropa los elementos que están en contacto directo con el cliente resultan necesario que mantenga un tipo de comunicación inalámbrica, para este trabajo se optó por el uso de una interfaz inalámbrica por radio frecuencia, por la disponibilidad de equipamiento en el mercado y de precio accesible, se usó para este trabajo los módulos RF de Arduino.

En la Figura 15-3 se puede observar el conjunto emisor - receptor donde, el emisor posee una forma de llavero con antena retráctil, por lo que en éste se deberá realizar un tipo de adaptación para colocarlo en los ganchos de ropa del prototipo.

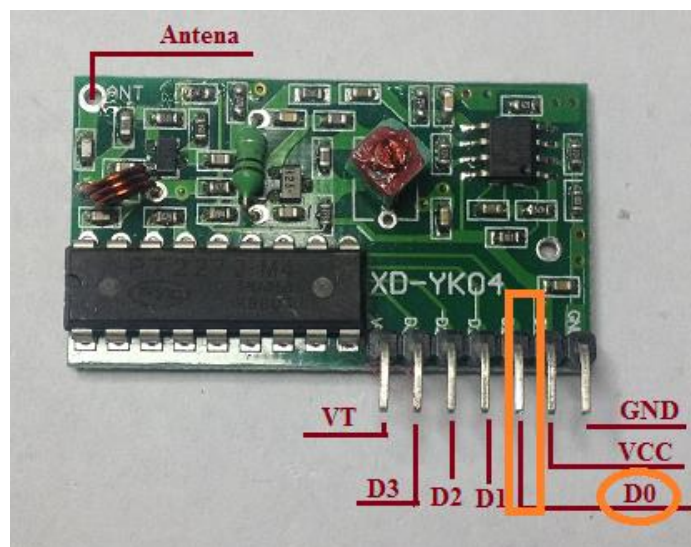


**Figura 15-3.** Módulos de Radio Frecuencia.

Fuente: <https://n9.cl/u37tj>

Al analizar los requerimientos y las características de estos módulos de radio frecuencia se establece que cubren la necesidad propuesta, al ser que se realizará una comunicación unidireccional pues el gancho envía información, más no se recibe ninguna señal de retroalimentación. Los módulos de RF trabajan a una frecuencia de 315 MHz.

En la Figura 16-3 se tiene la distribución de los pines del receptor: Se observa que maneja cuatro señales digitales, éstas se relacionan a los cuatro botones que dispone el emisor. Para el desarrollo de la aplicación se utilizó uno solo, pues el estado del gancho representa una sola señal discreta, o “está colgada la prenda” o “la retiró el cliente”.



**Figura 16-3.** Receptor RF – Pines de uso en la aplicación.

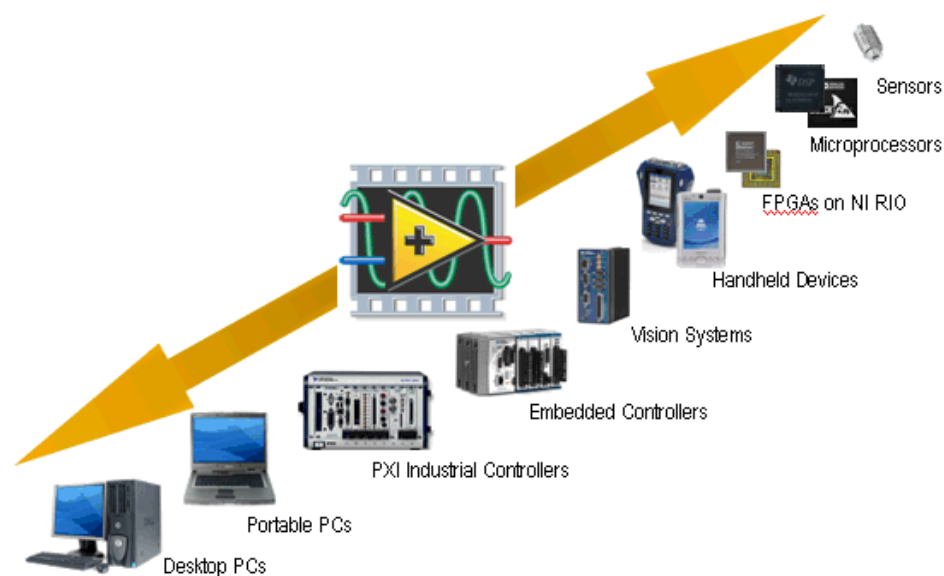
Fuente: <https://n9.cl/5snv>

En el receptor se observa que existe un pin de conexión para una antena, la misma que debe ser colocada para ganar distancia de cobertura.

### 3.2.2 Hardware & Software – Soporte para Aplicación Informática

El desarrollo de la aplicación informática se lo realizó sobre la plataforma de LabVIEW por sus prestaciones para el manejo de interfaces de comunicación que le permiten la interacción con elementos externos, pues en este caso se requiere gestionar las señales de los módulos a instalarse en los ganchos para conocer su estado dentro de la percha inteligente. Estos resultan para la aplicación dispositivos externos.

En la figura 17-13 se representa la flexibilidad de LabVIEW para relacionarse con varios tipos de dispositivos.



**Figura 17-3.** LabVIEW y la conexión con dispositivos externos.

Fuente: <https://n9.cl/w13y>

LabVIEW permite la interacción de su plataforma con otras aplicaciones, en este trabajo el objetivo es relacionarlo con una plataforma de soporte IoT para cubrir el requerimiento de establecer un monitoreo remoto de la percha inteligente.

Además, LabVIEW proporciona herramientas para la administración de información mediante el enlace a gestores de bases de datos justificando su uso al ser que en la aplicación se manejará un sistema de inventarios para existencias y ventas de productos en una boutique.

Los requisitos de hardware para la instalación y normal desarrollo del Software LabVIEW se muestran en la Tabla 3-3.



**Tabla 3-3:** Requerimientos de hardware para instalación de LabVIEW

CARACTERÍSTICA	DESCRIPCIÓN
Procesador	Pentium III/Celeron 866 MHz o equivalente mínimo, Pentium 4/M o equivalente recomendado.
RAM	256 MB mínimo, 1 GB recomendado
Resolución de Pantalla	1024 x 768 píxeles mínimo, 1024 x 768 píxeles recomendado
Sistema Operativo	Windows 8.1/8/7/Vista/XP SP3 mínimo, Windows 8.1/8/7/Vista recomendado
Espacio en Disco	500 MB - 5 GB (instalación completa)

Fuente: (Vizcaíno y Sebastián, 2011)

Realizado por: Ichina, Alex, 2020

En virtud a estos requerimientos se empleó una computadora portátil con las características que se muestran en la Figura 18-3.

[Ver información básica acerca del equipo](#)

The screenshot displays the Windows 8 system information page. At the top, it shows 'Edición de Windows' as 'Windows 8.1 Single Language' with copyright information for 2013 Microsoft Corporation. The 'Sistema' section lists the following specifications: Processor: Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz 2.60 GHz; Memory installed (RAM): 8,00 GB (7,88 GB utilizable); System type: Sistema operativo de 64 bits, procesador x64; and Touch and pen input: La entrada táctil o manuscrita no está disponible para esta pantalla. The 'Configuración de nombre, dominio y grupo de trabajo del equipo' section shows: Nombre de equipo: DELL7; Nombre completo de equipo: DELL7; Descripción del equipo: WORKGROUP. The 'Activación de Windows' section indicates that Windows is activated and provides a link to the Microsoft software license terms, along with the product ID: 00262-30227-15264-AAOEM. Logos for Dell, Windows 8, and Microsoft are visible on the right side of the page.

**Figura 18-3.** Características de hardware para desarrollo de la aplicación informática.

Fuente: <https://n9.cl/w13y>

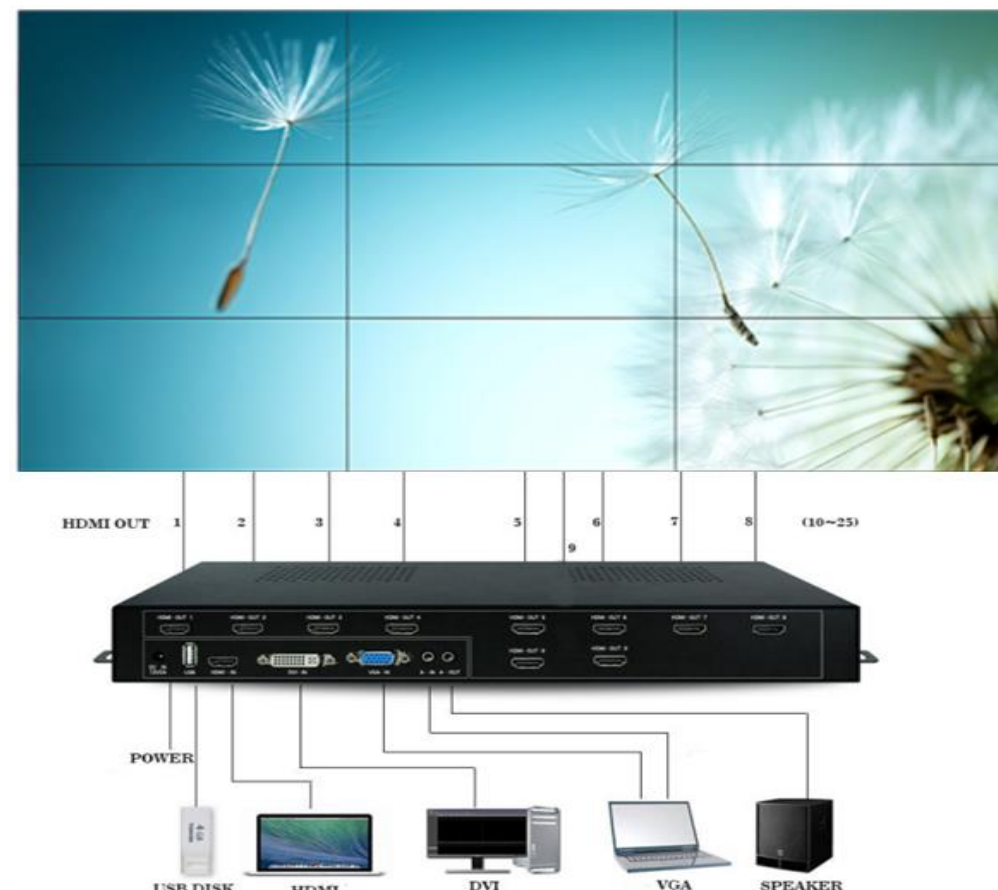
### 3.2.3 Hardware – Recursos Visuales

Como estrategia para el manejo de recursos visuales se programó el uso de un conjunto de monitores interactuando entre sí en forma de mosaico para generar una pantalla más grande sin la inversión de un gran capital.

En una descripción más específica del requerimiento se puede mencionar que el Video Wall tiene dos aspectos importantes:

- El primero, en estado de reposo de la percha inteligente, entendiéndose a este como el instante en el que el cliente no tiene contacto con las prendas en exhibición, el Video Wall proyecta videos de publicidad en general de la boutique.
- El segundo, cuando el sistema requiera interactuar con el cliente por medio del Video Wall lo hace mediante la evaluación de la señal del gancho correspondiente a la prenda que fue levantada, y ocurre una conmutación de fuentes para el Video Wall. La conmutación de fuentes para el Video Wall consiste que, al ser levantado un gancho específico éste activará un video correspondiente a la prenda que alberga, de esta manera el cliente podrá apreciar la prenda montada en una persona real.

Para la creación del Video Wall se requieren muchas computadoras o a su vez de varias tarjetas de video, lo que involucra una gran inversión. También existen sistemas embebidos para la generación del Video Wall, pero su costo es elevado. En la figura 19-3, se muestra un sistema de Video Wall comercializado.



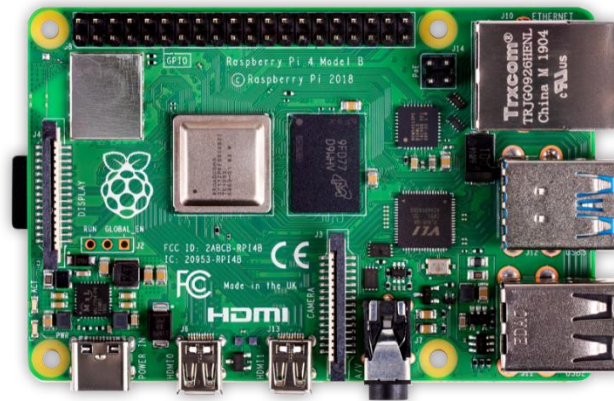
**Figura 20-3.** Sistema de Video Wall comercializado.

Fuente: <https://n9.cl/mtqg>

Para la implementación del Video Wall a emplearse en el prototipo se consideró el uso del minicomputador Raspberry Pi3 que actualmente se ha constituido en un dispositivo apto para adaptarlo en diferentes entornos, sirve de base para la generación de un servidor web o un centro de multimedia casero entre sus destacadas aplicaciones.

Se consideró para la experimentación generar el Video Wall con cuatro monitores por lo que surge como requerimiento el uso de cinco Raspberry, donde cuatro de ellas actuarán en el papel de esclavos y una como maestro.

Se seleccionó la Raspberry Pi3 con las características expuestas en la Tabla 4-3.



**Figura 21-3.** Raspberry Pi3

Fuente: <https://n9.cl/ia8kl>

**Tabla 4-3:** Características de la Raspberry Pi3

CARACTERÍSTICA	DESCRIPCIÓN
Marca	Raspberry Pi
Serie	Raspberry PI 3 Model B
Peso del producto	45,4 g
Dimensiones del producto	12,2 x 7,6 x 3,4 cm
Pilas	3 9 V (Tipo de pila necesaria)
Tipo de procesador	Core 2 Quad
Velocidad del procesador	1.20 GHz
Capacidad de la memoria RAM	1 GB
Interfaz del disco duro	ATA-4
Tipo de conectividad	WiFi
Tipo de conexión inalámbrica	802.11bgn
Número de puertos USB 2.0	4
Voltaje	5 voltios DC
Plataforma de Hardware	Linux
Sistema operativo	Linux

Fuente: (Raspberry, 2020).

Realizado por: Ichina, Alex, 2020

Para proyección del Video Wall se dispone de monitores de 19 pulgadas en la Tabla 5-3 se exponen las características.

**Tabla 5-3:** Características de los monitores a usarse

CARACTERÍSTICA	DESCRIPCIÓN
Tamaño	19 pulgadas
Pantalla	LCD
Formato	16:9
Resolución	1366 x 768
Brillo	300 (cd/m <sup>2</sup> )
Radio de contraste	30.000:1
Tiempo de respuesta	5 ms
Ángulo de Visión	170°
Número de Colores	16.7(Millones)

Fuente: (LG, 2020).

Realizado por: Ichina, Alex, 2020



**Figura 22-3.** Monitor

Fuente: <https://n9.cl/ecn4i>

### 3.2.4 Hardware – Interfaz de adquisición

La comunicación inalámbrica de los receptores proporciona señales digitales, que expresados como niveles de voltaje se habla de un nivel de tensión de 5 VCC que mediante una pasarela deben llegar hacia la aplicación informática. Si bien es cierto, estas señales pueden ser direccionadas directamente a un microcontrolador para su procesamiento, pero en esta ocasión se dispone a usar relés con bobinas de 5 VCC como elementos de protección. La Figura 22-3 representa un módulo de 8 relés de la gama de Arduino.



**Figura 23-3.** Módulo de Relés, Arduino..

**Fuente:** <https://n9.cl/qd5ms>

El papel que desempeñarán los relés, es la conmutación de sus contactos proporcionar señales discretas hacia un microcontrolador albergado en la plataforma Arduino UNO, el mismo que se convierte en una pasarela de información que comunique el entorno físico con la aplicación informática.

Se expone a continuación en la Tabla 6-3 las características principales de un Arduino UNO.

**Tabla 6-3:** Características plataforma Arduino UNO

CARACTERÍSTICA	DESCRIPCIÓN
Microcontrolador:	ATmega328
Voltaje Operativo	5V
Voltaje de Entrada	7-12V
Voltaje de Entrada (límites)	6-20V
Pines digitales de Entrada/Salida:	14 (6 proveen salida PWM)
Corriente DC por cada Pin Entrada/Salida	40 mA
Corriente DC entregada en el Pin	3.3V: 50 mA
Memoria Flash	32 Kb de los cuales 0,5 KB usados por el bootloader
Memoria SRAM	2KB
Memoria EEPROM	1KB
Velocidad de Reloj	16 MHz
Forma de alimentación	Fuente externa o por USB
Interfaz de comunicación	USB/Serial

**Fuente:** (Sabika, 2010).

**Realizado por:** Ichina, Alex, 2020

Analizando las características del sistema embebido Arduino UNO se puede deducir que resulta ideal para la aplicación, según el número de puertos digitales que dispone se encuentra sobredimensionado en primera instancia, pues se planteó cuatro como requerimiento.

Además, al emplear comunicación serial, lo hace ideal para enlazarlo con la aplicación informática desarrollada con LabVIEW, donde éste se había dicho permite la comunicación con dispositivos externos. El consumo de energía es bajo.

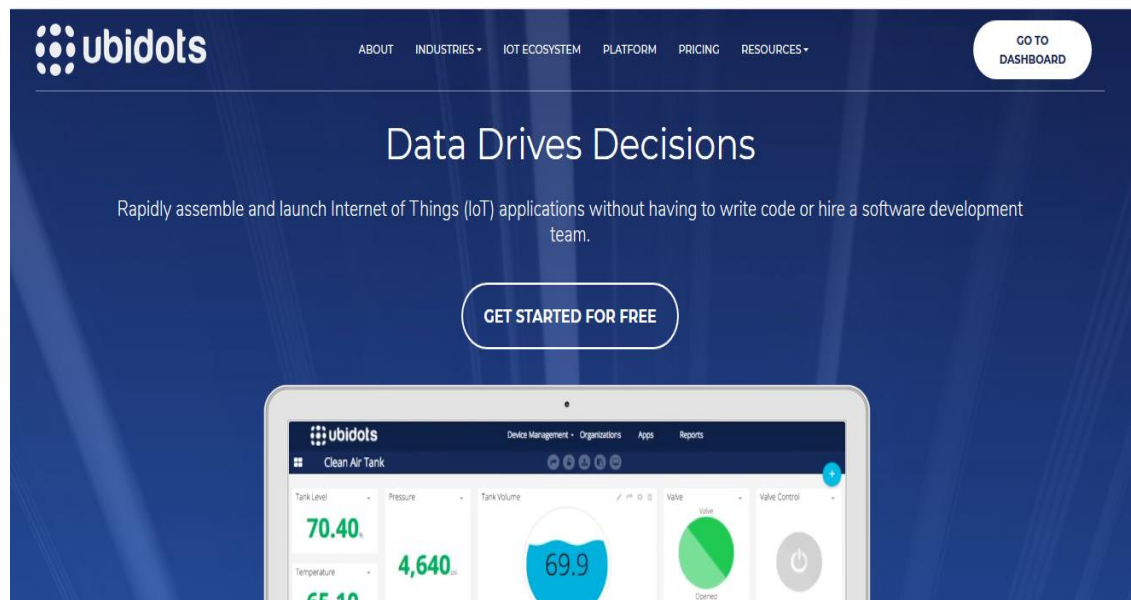


**Figura 24-3.** Plataforma Arduino UNO

Fuente: <https://n9.cl/bw3q>

### 3.2.5 Software – Plataforma IoT

La figura 25-3, muestra la Plataforma de Ubidots IoT.



**Figura 24-3.** Plataforma de Ubidots IoT.

Fuente: <https://ubidots.com/>

Con la finalidad de innovar con la ejecución de este proyecto se plantea que la percha puede conectarse al internet de las cosas para la realización del monitoreo remoto del estado de la percha inteligente.

El uso de Ubidots, una plataforma de IoT permite el monitoreo de variables de diferente tipo, incluidas las discretas para el caso de estudio.

### ***3.2.6 Software – Video Wall***

Al determinar el uso de Raspberry Pi3 como elementos para la implementación del Video Wall se relaciona directamente con el uso del sistema operativo Linux, esto direcciona de igual manera al uso de Python para la generación del mismo que sirve de plataforma para generar el código necesario para controlar la conmutación de las fuentes en el Video Wall.

La Raspberry para reproducción de videos sirve únicamente el reproductor omxplayer por lo que se puede considerar que forma parte del paquete de software necesario para la implementación del Video Wall.



**Figura 26-3.** Plataforma de Python

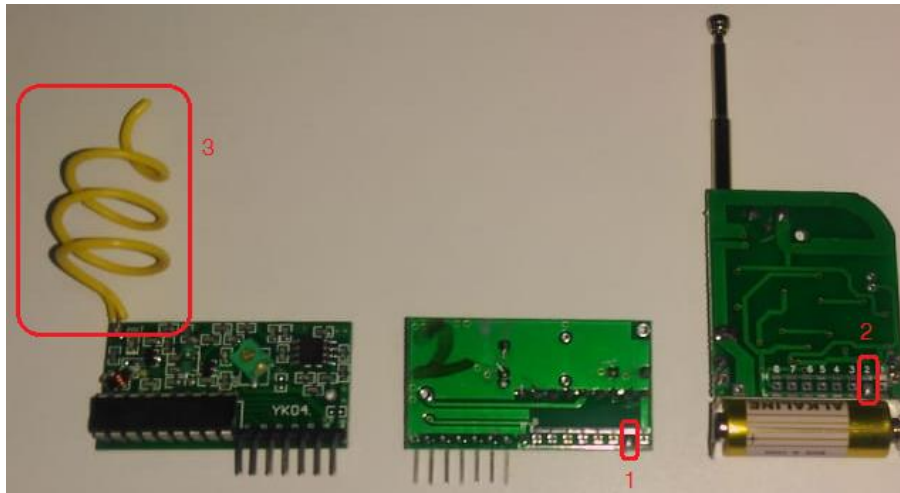
Fuente: <https://ubidots.com/>

## **3.3 Programación y configuración de software y hardware**

### ***3.3.1 Adecuación de emisor – receptor de RF***

De fábrica el conjunto emisor – receptor de RF viene sin codificación, lo que requiere de un proceso de adecuación.

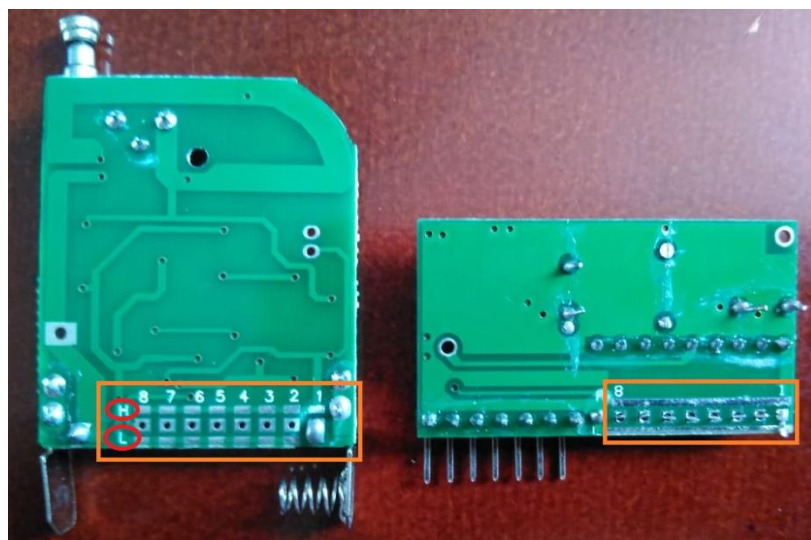
La figura 26-3 representa las acciones que se deben tomar en cuenta para poner operativos los módulos RF. El cuadro rojo 1 y 2 de la figura representan la codificación que se realiza para que el receptor reconozca al emisor como único y el cuadro 3 indica que para aumentar el alcance del receptor se debe soldar un cable espiral para que actúe como antena.



**Figura 27-3.** Adecuación módulos RF.

Realizado por: Ichina, Alex, 2020

Para la codificación se dispone en ambos módulos de 8 pines y dos líneas, una de HIGH y una de LOW como se indica en la Figura 27-3, siendo necesario que la codificación del emisor sea la misma del receptor para que puedan identificarse y tener comunicación. En el caso citado como ejemplo en la Figura 26-3 se puede observar que la codificación usada es el pin 2 a LOW en los módulos.



**Figura 28-3.** Pines de codificación módulos RF.

Realizado por: Ichina, Alex, 2020

### 3.3.2 Adquisición de señales del conjunto emisor receptor

El microcontrolador Atmega328 sobre la plataforma Arduino UNO A se lo empleó como medio de adquisición de la información de los receptores.

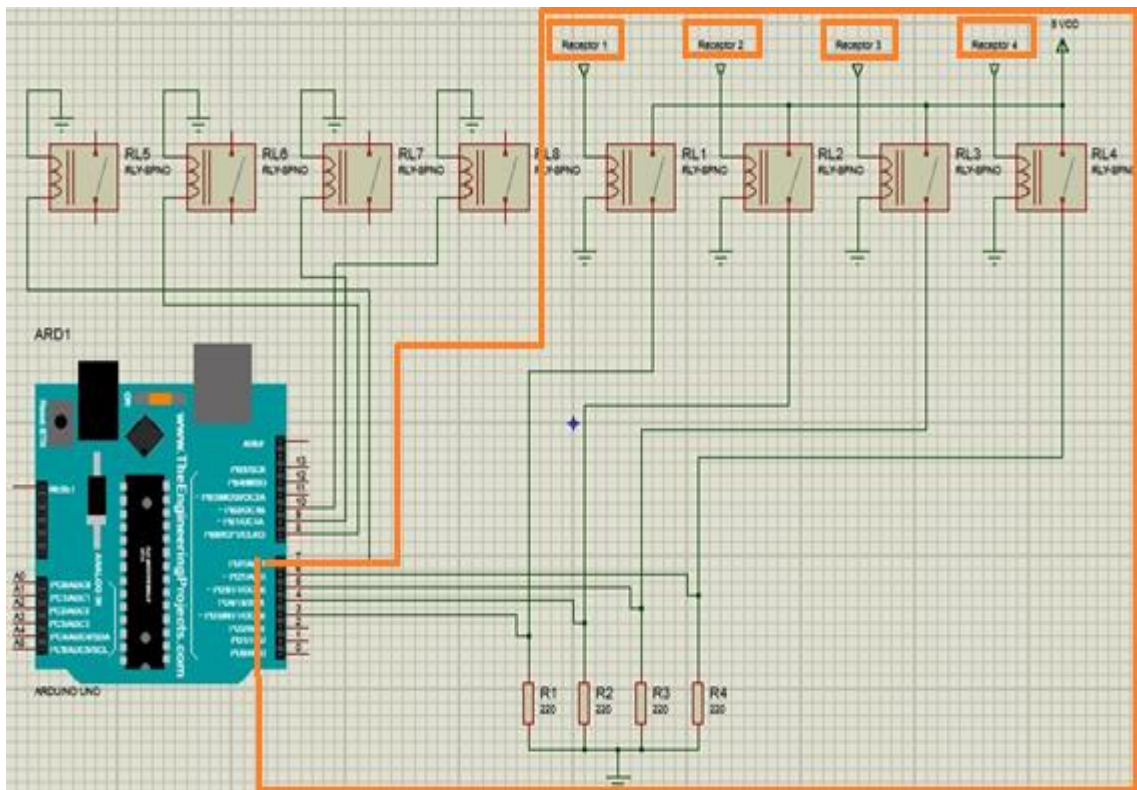


Los receptores como se explicó anteriormente entregan un nivel lógico de HIGH al recibir la señal del emisor, como que se observa en la Figura 28-3 es la señal proveniente del receptor la que se emplea para la polarización de las bobinas de los relés.

El módulo de relé por medio de su contacto normalmente abierto es aprovechado para generar una señal digital, la misma que se direcciona al microcontrolador en el que se la declara como entrada mediante programación.

El ciclo de funcionamiento es: - envío de una señal desde el emisor, - recepción de la señal y polarización del relé a través del receptor - polarización de la entrada del microcontrolador.

La información proporcionada desde el puerto serial del Arduino UNO A, se la realiza a través del bloque de programación en LabVIEW expuesto en la Figura 30-3.



**Figura 29-3.** Diagrama de conexiones adquisición de señales del receptor al Arduino UNO A.

Realizado por: Ichina, Alex, 2020

### 3.3.3 Monitoreo local del estado de las prendas en la percha

#### 3.3.3.1 Módulo de Transmisión

La aplicación informática, presenta la opción de realizar un monitoreo local sobre el estado de las prendas dentro de la percha, mediante la creación de un bloque de programación en LabVIEW, donde se parte de la adquisición de las señales por parte de los elementos del entorno físico que faciliten la información.

En la figura 28-3, se muestra el diseño de la interfaz para la adquisición de la información del conjunto emisor – receptor donde se tiene un Arduino UNO A, que actúa como pasarela entre el entorno físico y la aplicación informática, mediante una comunicación serial.

La Figura 29-3 representa una sección de la programación del microcontrolador ejecutada sobre el IDE de la plataforma Arduino. El bloque de la función setup () contiene cuatro variables que corresponden a las señales que se maneja a través de los relés según la acción del conjunto emisor - receptor RF.

```
void setup()
{
  Serial.begin(9600);
  pinMode(pin1, INPUT);
  pinMode(pin2, INPUT);
  pinMode(pin3, INPUT);
  pinMode(pin4, INPUT);
}

void loop()
{
  var1=digitalRead(pin1);
  var2=digitalRead(pin2);
  var3=digitalRead(pin3);
  var4=digitalRead(pin4);
  if(var1==HIGH)
  {s[0]='e';}
  else{s[0]='o';}
  if(var2==HIGH)
  {s[1]='f';}
  else{s[1]='o';}
  if(var3==HIGH)
  {s[2]='g';}
  else{s[2]='o';}
  if(var4==HIGH)
  {s[3]='h';}
  else{s[3]='o';}
  Serial.println(s);
}
```

**Figura 30-3.** Programación del Arduino UNO A para lectura y envío de información.

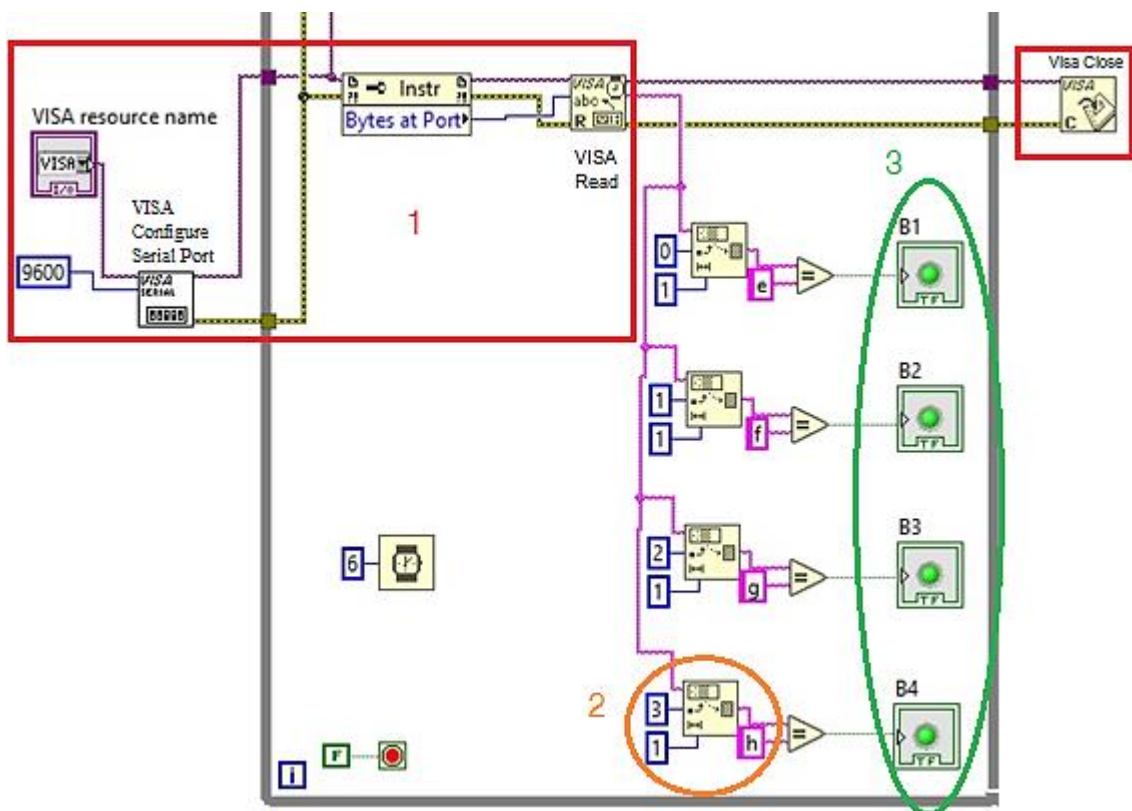
Realizado por: Ichina, Alex, 2020

La segunda sección del programa representa un bloque de código incluido en la función loop (), en la que inicialmente se realiza la lectura del estado de los pines digitales declarados como entradas, para luego ser procesados. De acuerdo al estado de la variable digital, se asigna un carácter: ‘e’ (gancho 1- activo), ‘f’ (gancho 2- activo), ‘g’ (gancho 3- activo), ‘h’(gancho 4- activo), cuando la entrada ha sido activada, es decir, otorga un valor de nivel lógico *HIGH* en su lectura y una ‘o’ (ningún gancho activo) cuando la entrada no ha sido manipulada, donde otorga un valor de nivel lógico *LOW* en su lectura.

En el código el vector s condensa la información de todas las variables codificada mediante caracteres, y éste vector que se lo imprime hacia al puerto serial para su lectura y procesamiento en la aplicación informática.

### 3.3.3.2 Módulo de recepción de información

El módulo de recepción de la percha inteligente facilita la lectura de la información proporcionada desde el puerto serial del Arduino UNO A, como se muestra en la figura 30-3.



**Figura 31-3.** Adquisición y procesamiento de información del puerto serial.

Realizado por: Ichina, Alex, 2020

Los cuadros rojos montados sobre la Figura 30-3, contienen los bloques de programación para la lectura del puerto serial al que se conecta el Arduino UNO A. Parte de la inserción de un bloque para configuración del puerto serial **VISA Configure Serial Port** en el que asigna como atributo un **VISA Resource Name** que es un elemento de entrada que se refleja en el panel frontal de LabVIEW donde se diseña la interfaz gráfica, se especifica el puerto COM al que se ha conectado el Arduino UNO para su lectura. Se añade también una constante numérica que indica la velocidad a la que se comunicará el Arduino UNO con la aplicación informática. Posterior a la configuración del puerto serial se añade el conjunto de bloques para la lectura del puerto **Visa Read**, que entrega como resultado información en forma de un arreglo. Para cerrar el ciclo de la comunicación se añade también un **VISA Close**.

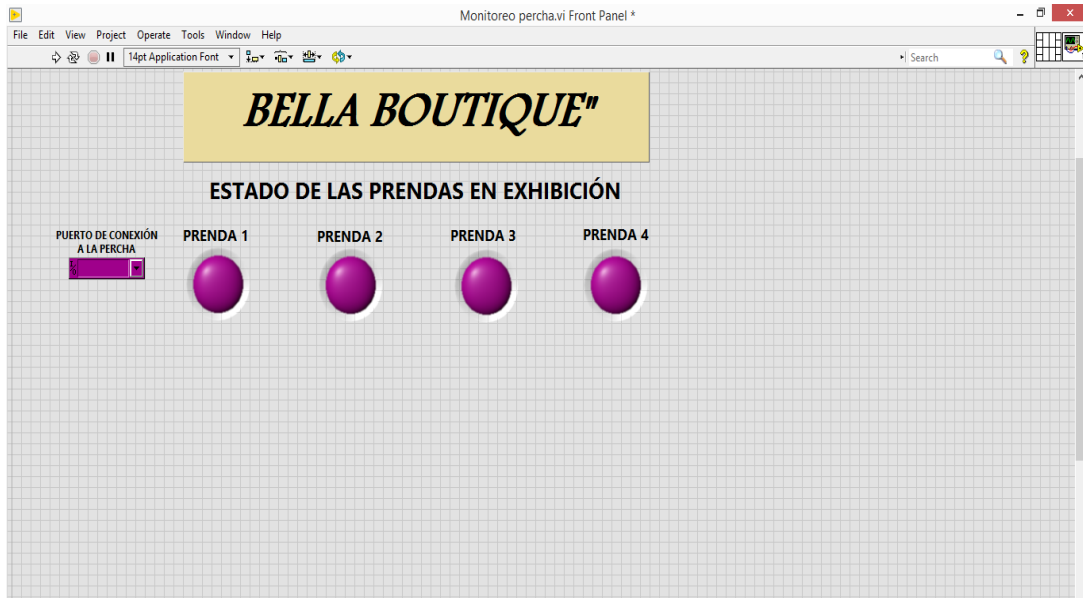
El arreglo entregado por el Visa Read contiene una trama de 4 caracteres como se los envió desde el Arduino UNO A en forma de vector y por medio de un bloque **String Subset** (círculo naranja). Este vector es tratado como una cadena y permite la separación de sus elementos, en este caso caracteres según la posición que se asigne dentro de sus atributos de configuración.

En el círculo de color naranja, una vez que se separan los cuatro elementos del vector se realiza una comparación entre el carácter recibido y constantes string relacionadas con la codificación empleada en el Arduino UNO A. El resultado de la comparación determinará un valor booleano que será empleado para el control del estado de indicadores digitales que se pueden apreciar en el óvalo verde.

### *3.3.3.3 Desarrollo de Interfaz gráfica para el monitoreo local*

Mediante el panel frontal de LabVIEW, se trabaja la interfaz gráfica para el monitoreo local que se posicionará en las instalaciones de la boutique, más no en la misma percha. La implementación del Video Wall instalada en la percha, se ejecuta mediante una interacción directa con el cliente. El monitoreo desde LabVIEW es exclusivo del personal de la boutique para fines pertinentes.

La Figura 31-3, muestra el diseño de una interfaz gráfica de alto nivel para interacción del prototipo con el usuario, usando recursos gráficos que permitan relacionar fácilmente el estado de las variables que representan.

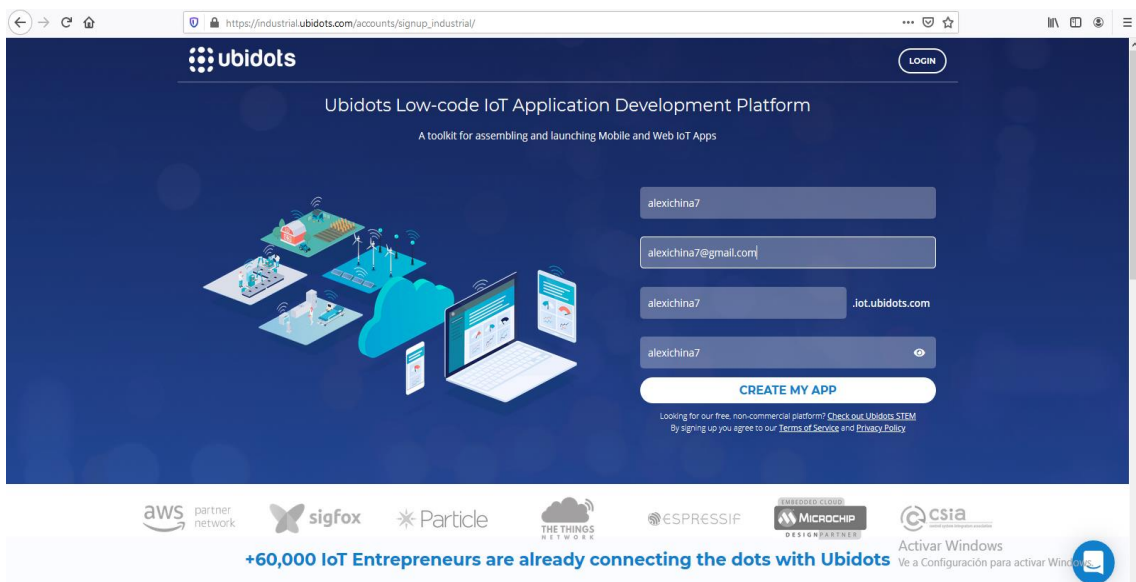


**Figura 32-3.** Diseño de Interfaz gráfica en LabVIEW.

Realizado por: Ichina, Alex, 2020

### 3.3.4 Monitoreo remoto del estado de prendas en percha en la plataforma IoT

Para realizar el monitoreo remoto del prototipo se optó por el uso de Ubidots, para lo cual se debe crear una cuenta en esta plataforma, existe dos opciones: suscribirse para prueba de la plataforma o suscribirse con un costo para uso extendido. Para el caso de estudio al tratarse de la implementación de un prototipo se optó por la suscripción para prueba con una duración de 30 días.



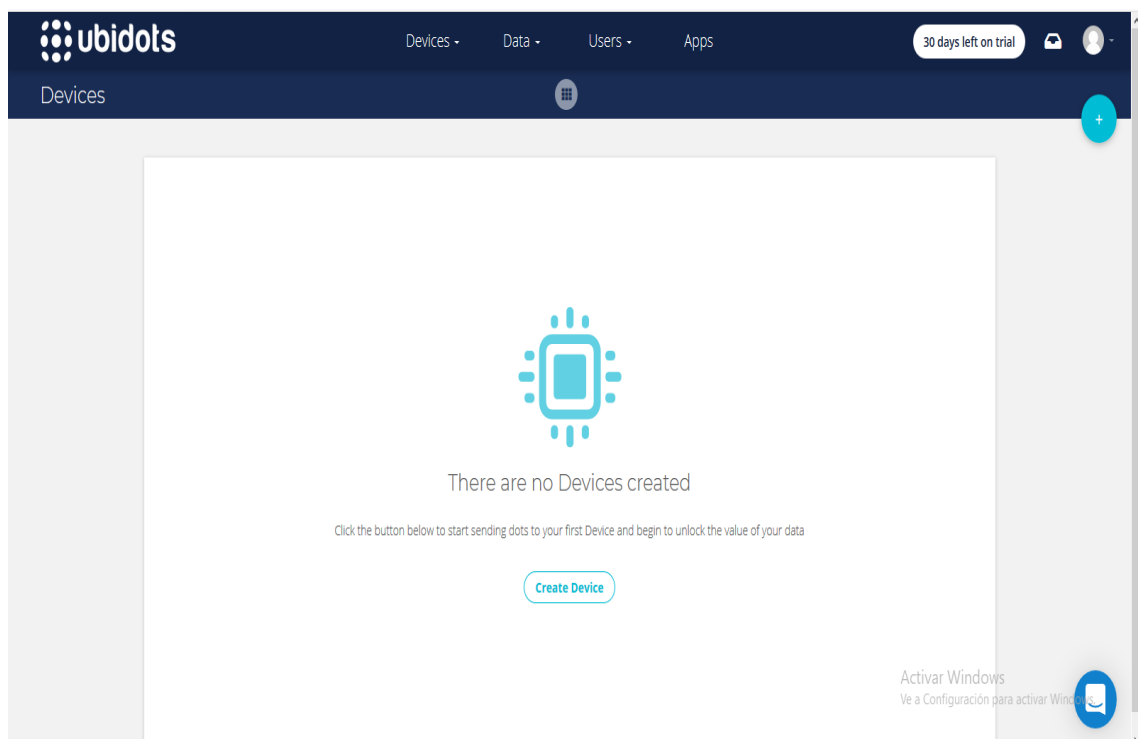
**Figura 33-3.** Creación de cuenta en Ubidots.

Realizado por: Ichina, Alex, 2020

Una vez creada la cuenta se accede a la plataforma donde en el menú de Ubidots presenta las siguientes opciones:

- **Device:** Hace referencia a la creación de dispositivos que se van a vincular a la plataforma o lo que se denominaría las cosas dentro de la tecnología IoT, Dashboard y User.
- **Dashboard;** Permite montar entornos para el monitoreo y control de las variables que se asignen dentro de cada dispositivo.
- **Users:** Da la posibilidad de otorgar roles a los usuarios en la administración o monitoreo de un dispositivo.

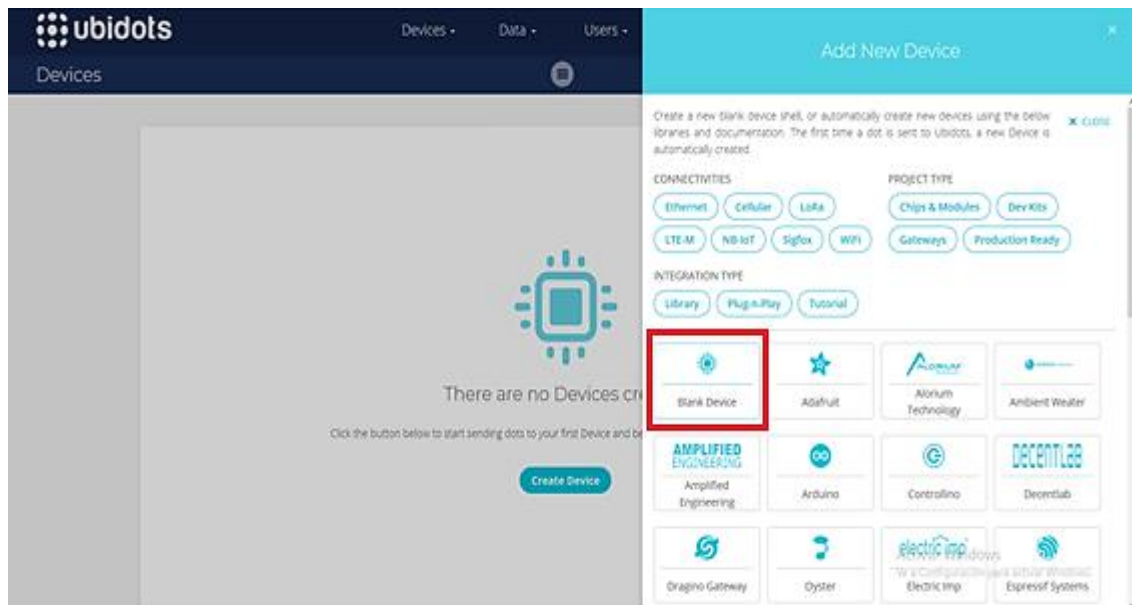
Para el caso de estudio, se requiere crear un dispositivo en la plataforma que represente la percha, por lo que al ingresar en la pestaña de Device se apertura una nueva pantalla Figura 33-3, en la que inicialmente se observa que no se dispone de ningún dispositivo.



**Figura 34-3.** Listado de dispositivos existentes en Ubidots.

**Realizado por:** Ichina, Alex, 2020

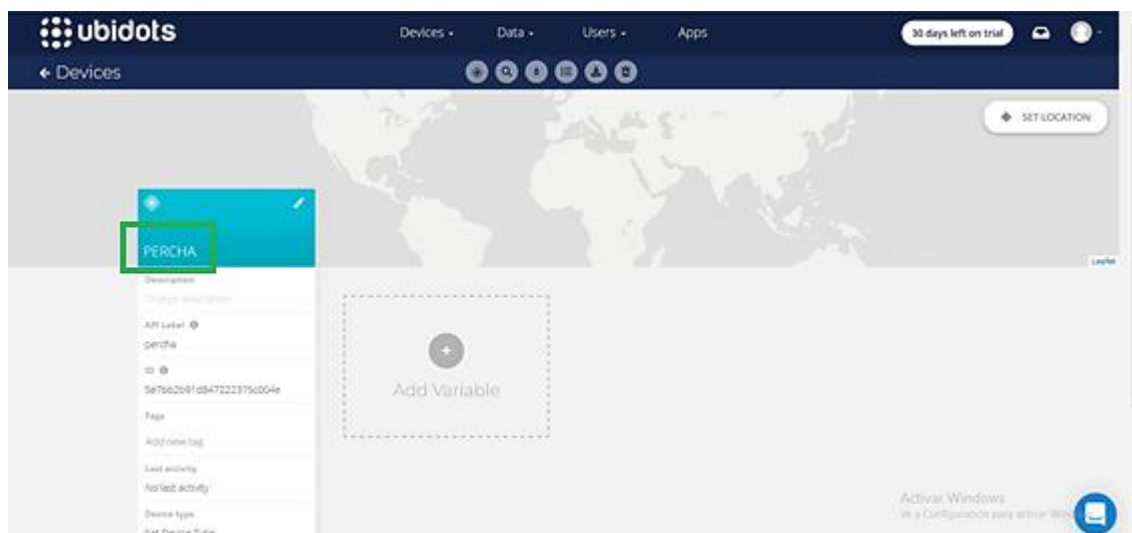
Para añadir un nuevo dispositivo vinculado con la cuenta creada se da click en **Create Device** y aparece la pantalla mostrada en la Figura 34-3 que muestra la adición de un dispositivo en blanco, es decir libre para configurarse de acuerdo a la necesidad que se tenga.



**Figura 35-3.** Adición de un dispositivo en Ubidots.

Realizado por: Ichina, Alex, 2020

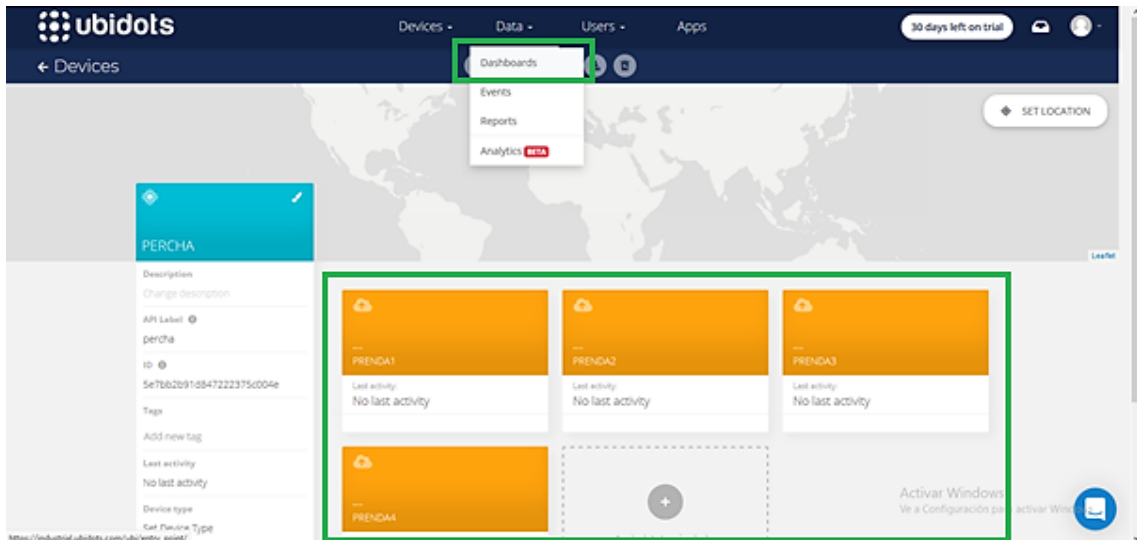
Una vez creado el dispositivo aparece en el listado de la pestaña Device. Para este trabajo se creó el dispositivo **PERCHA**, que al ingresar a revisar sus atributos se encuentra vacío con la leyenda **Add Variable** como se puede observar en la Figura 35-3.



**Figura 36-3.** Listado de variables del dispositivo en Ubidots.

Realizado por: Ichina, Alex, 2020

Las variables que se agregó al dispositivo creado corresponden a las cuatro señales discretas que proporcionarán los ganchos para determinar su estado dentro de la percha. En la figura 36-3 se observa las variables creadas en el recuadro verde grande.

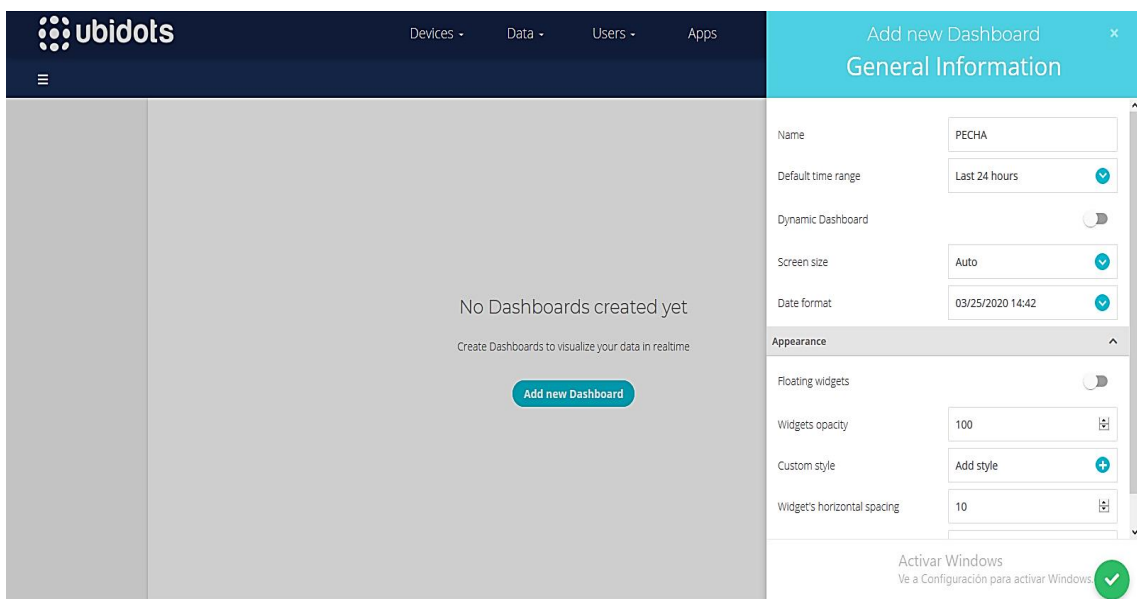


**Figura 37-3.** Variables declaradas dentro del dispositivo en Ubidots.

**Realizado por:** Ichina, Alex, 2020

Posterior a la generación de variables, se crea la interfaz gráfica para poder monitorear el estado de estas desde un entorno gráfico, para esto se debe ubicar sobre la pestaña Dashboard que se observa en el recuadro verde pequeño de la Figura 36-3.

La Figura 37-3 señala el procedimiento para la creación del nuevo Dashboard en el que de igual manera se solicita agregar recursos, los mismos que vinculan a las variables del respectivo dispositivo que se desea monitorear o controlar. Se creó el Dashboard denominado PERCHA.

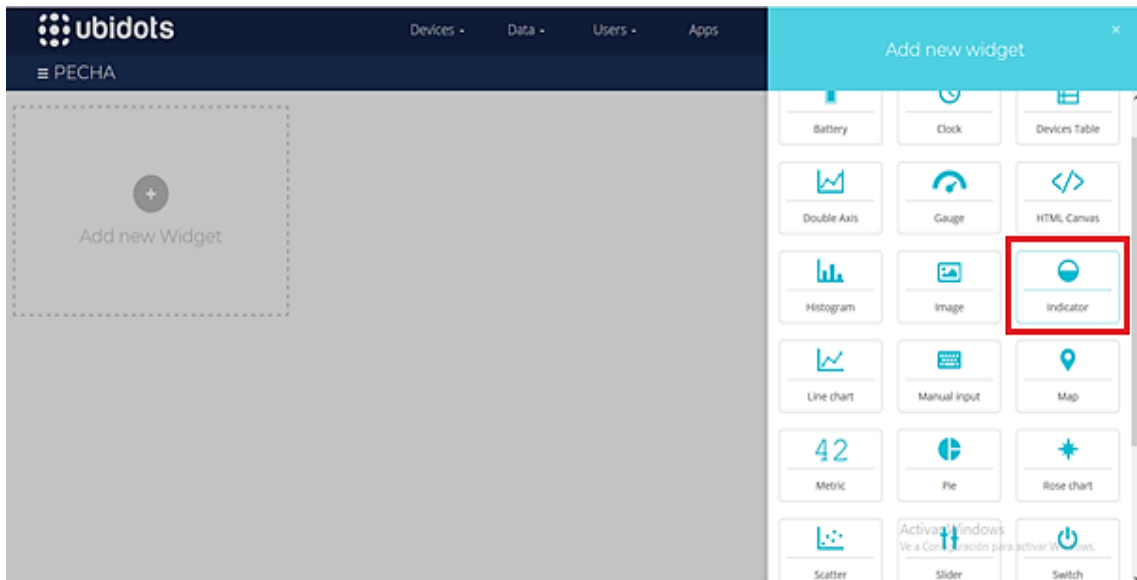


**Figura 38-3.** Creación del Dashboard en Ubidots.

**Realizado por:** Ichina, Alex, 2020



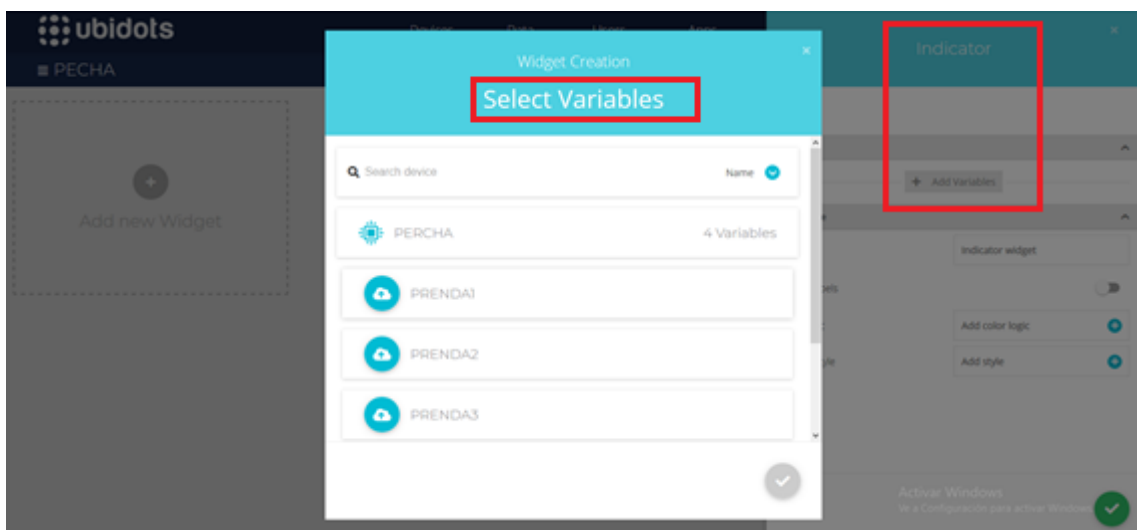
Establecido el Dashboard, se asigna el tipo de recurso gráfico que se desea añadir a la interfaz gráfica para representación de una variable, en este caso, al tratarse de variables discretas se seleccionó el recurso **Indicador** que consiste en la representación de un foco que maneja los estados de prendido y apagado para relacionarlo con el estado de la variable a la hora de monitorearla.



**Figura 39-3.** Adición recursos gráficos en Ubidots.

Realizado por: Ichina, Alex, 2020

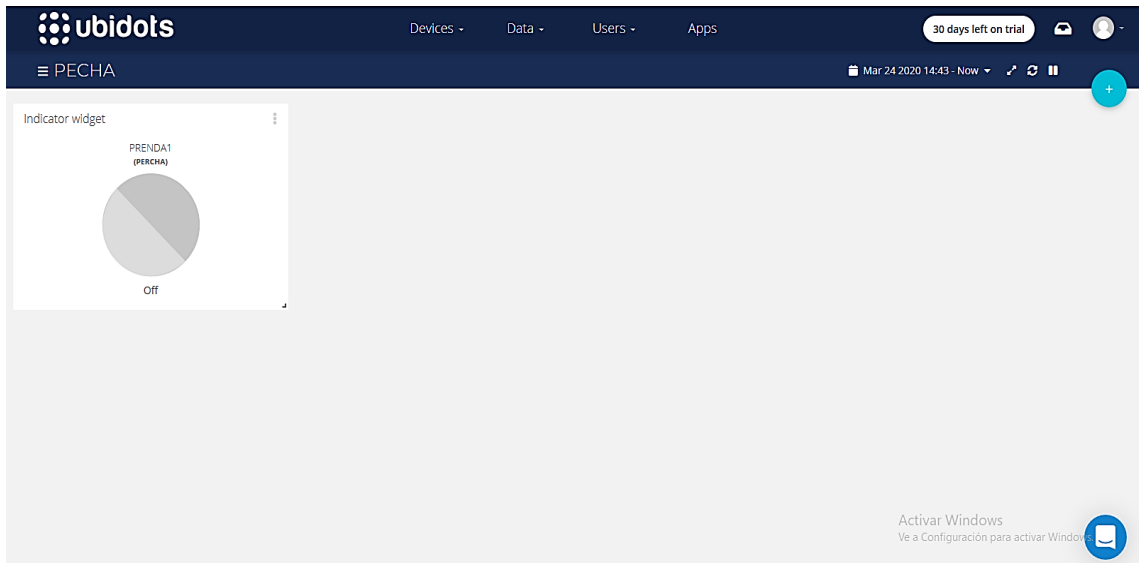
Seleccionado el recurso gráfico, se debe vincular a la variable del sistema a la que representa por lo que se proyecta en una nueva ventana, Figura 39-3 el listado de dispositivos disponibles con sus respectivas variables para realizar la selección.



**Figura 40-3.** Ubidots – Selección variable vinculada al indicador

Realizado por: Ichina, Alex, 2020

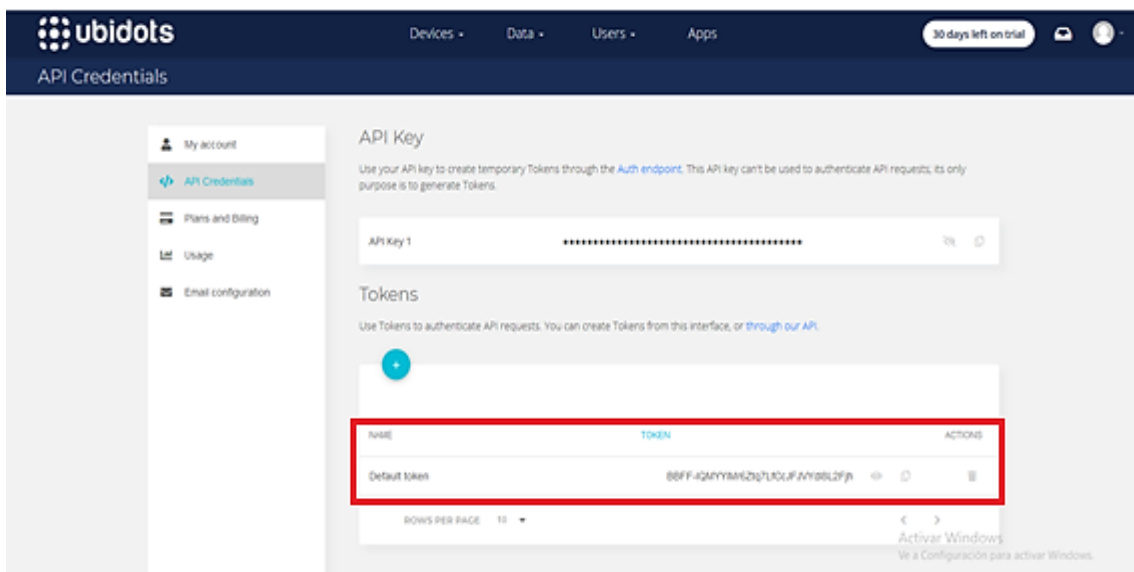
Una vez seleccionada la variable que se vincula al indicador se obtendrá el modelo de la Figura 40-3 donde se puede apreciar el recurso gráfico para la interacción del estado de la variable discreta.



**Figura 41-3.** Ubidots – Indicador anexo y vinculado a una variable en el Dashboard

**Realizado por:** Ichina, Alex, 2020

La adición de recursos a la Dashboard será de acuerdo al número de variables que se deseen monitorear o controlar desde esta interfaz, el proceso para anexarlos será el mismo acorde a lo descrito anteriormente.



**Figura 42-3.** Ubidots – Llave de enlace con otras aplicaciones, Tokens

**Realizado por:** Ichina, Alex, 2020

Para la vinculación de esta plataforma hacia otras aplicaciones, en este caso específico para a la aplicación informática a desarrollarse en LabVIEW se requiere de una identificación única de la cuenta de Ubidots, la misma que se ubica en la sección de **My Profile** en la sección denominada **API Credentials** donde la dirección escrita en el campo **Tokens** es la llave para la ejecución de cualquier tipo de enlace. En la Figura 41-3 se puede visualizar el entorno en el que se halla el código Tokens.

### 3.3.5 Configuración e implementación del Video Wall

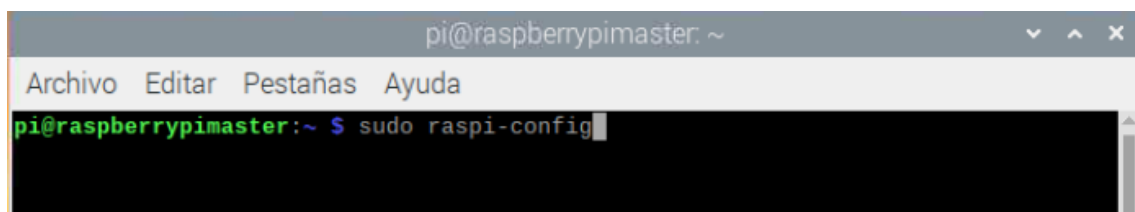
La configuración e implementación del Video Wall, brinda un realce interesante al proyecto pues la propuesta involucra el uso de hardware y software libre. Existen dispositivos embebidos que permiten el manejo de un Video Wall pero tienen costos elevados, en este proyecto se pone a disposición la información para la construcción de un Video Wall 2x2 de bajo costo y de alto rendimiento.

Inicialmente se debe cargar el sistema operativo en las cinco tarjetas Raspberry del proyecto, en la página oficial [www.raspberrypi.org](http://www.raspberrypi.org) se puede descargar en diferentes versiones de forma gratuita, en este caso se empleó el sistema operativo Raspbian.

Para la instalación del sistema operativo se descargado en formato de imagen de disco y se requiere de una SD Card de clase 10, la misma que previamente debe ser formateada y empleando un software auxiliar como WinDiskImage para montar la imagen sobre ella.

El modo de funcionamiento del Video Wall, puede ser como Maestro para gestionar las instrucciones de reproducción del video a través de monitores o pantallas conectadas a la salida de HDMI de las otras Raspberry que actúan en situación de esclavos. Todas las Raspberry están enlazadas por una interfaz Ethernet sobre una red de área local a través de un switch.

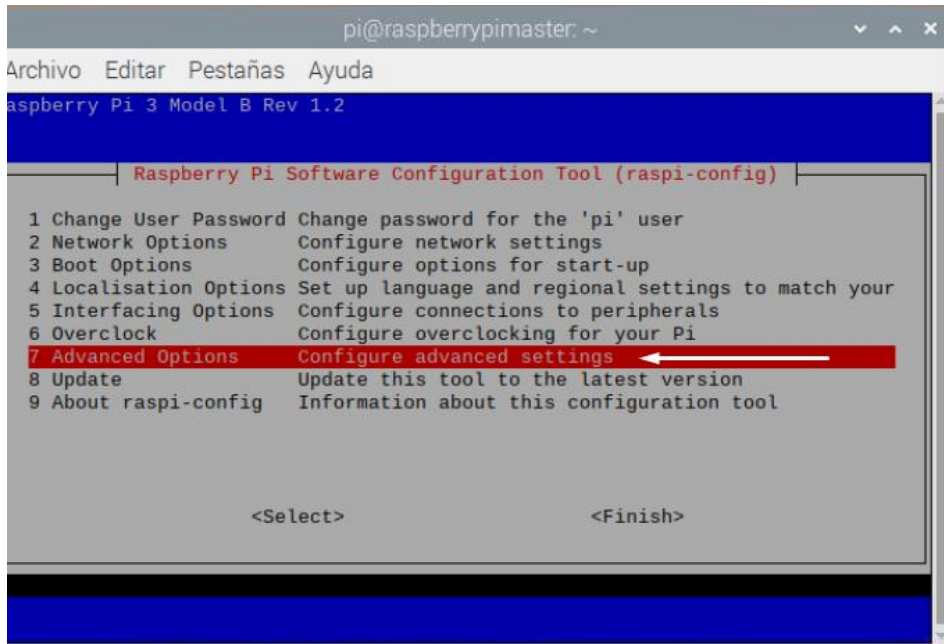
Es recomendable que antes de iniciar el proyecto, ejecutar en la Raspberry desde el terminal de la minicomputadora ciertos comandos como el “*sudo raspi-config*” para acceder a cambiar o configurar ciertos parámetros, como se muestra en la Figura 42-3.



**Figura 43-3.** Ejecución comando sudo raspi-config en el terminal

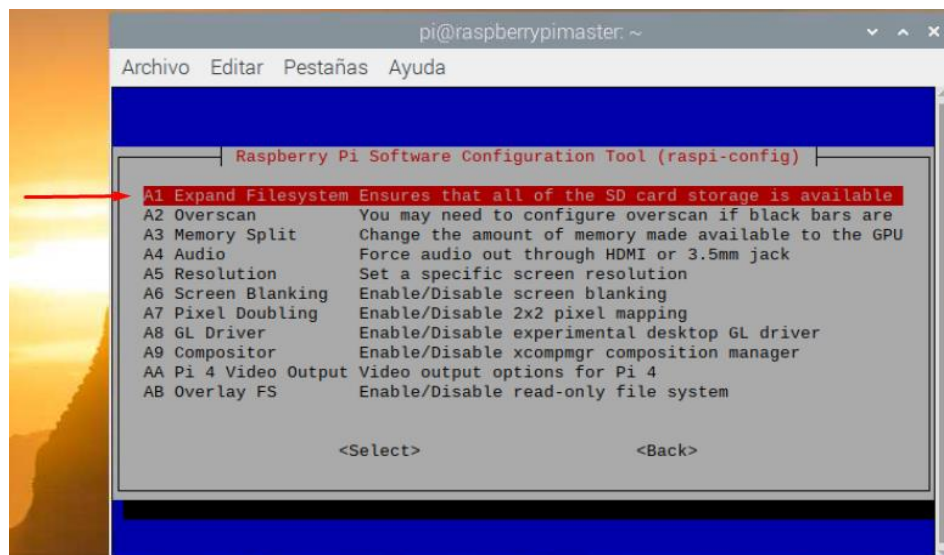
Realizado por: Ichina, Alex, 2020

Tras la ejecución del comando, se despliega una nueva ventana con las opciones de configuración, se selecciona la opción 7 Advance Options (Figura 43-3) que despliega un nuevo listado de opciones (Figura 44-3) donde se selecciona la expansión del sistema de archivos, es decir, liberar la memoria asegurando que todo el almacenamiento de la SD Card esté disponible.



**Figura 44-3.** Configuración ajustes avanzados en la Raspberry

Realizado por: Ichina, Alex, 2020



**Figura 45-3.** Expansión de memoria en ajustes avanzados en la Raspberry

Realizado por: Ichina, Alex, 2020

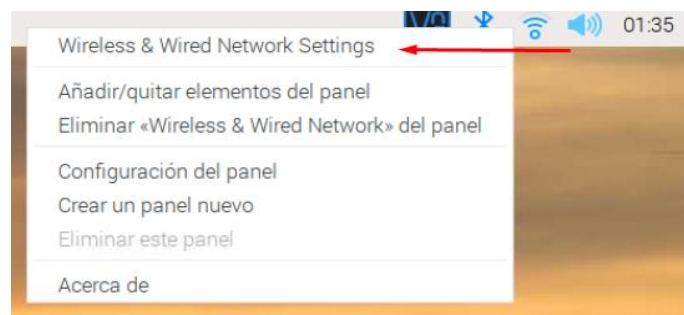
Otros de los comandos necesarios para preparar a la Raspberry son los que a continuación se citan:

- **“sudo rpi-update”** Permite realizar una actualización del Sistema Operativo en conjunto con el kernel y sus módulos a su última versión disponible. Ocurre que en ocasiones pide una confirmación en el transcurso de la instalación la cual se debe autorizar.
- **“sudo ldconfig”** Esta instrucción facilita la creación de enlaces y asignación de memoria caché necesarios para las bibliotecas compartidas más recientes ubicadas en los directorios especificados en la línea de comando.
- En este punto se debe reiniciar la Raspberry para luego volver a aperturar el terminal y continuar ejecutando instrucciones.
- **“sudo apt-get update”** Recoge todas las librerías y parches actualizados.
- **“sudo apt-get dist-upgrade”** Actualiza e instala paquetes actualizados para todo el conjunto de librerías y el sistema operativo de la Raspberry.

### 3.3.5.1 Configuración de la red

Una comunicación de maestro y esclavo en este caso se refiere a un tipo de red de conexión de un conjunto de Raspberry Pi3 y monitores para la implementación del Video Wall que se lo ha denominado **Piwall** como contracción empleada por varios autores.

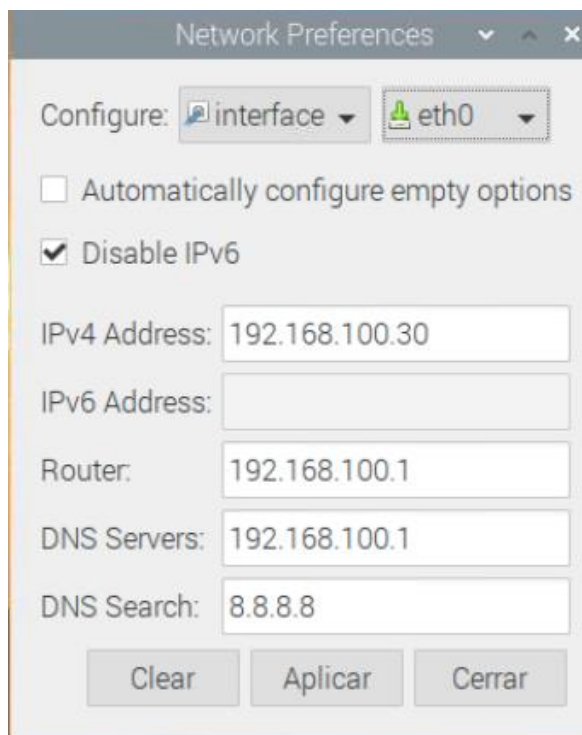
Se emplea una red de área local, con una topología de estrella donde el punto central lo representa el Master y los elementos finales los esclavos. La Figura 45-3 muestra el acceso a la configuración de red.



**Figura 46-3.** Ingreso a la configuración parámetros de la red

Realizado por: Ichina, Alex, 2020

Donde la configuración de la Raspberry Máster se la puede apreciar en la Figura 46-3.



**Figura 47-3.** Configuración parámetros de red para el Master.

**Realizado por:** Ichina, Alex, 2020

El direccionamiento del equipo Master corresponde a la asignación de una dirección ip en el formato IPv4 donde por consecuencia las Raspberry esclavos poseerán una dirección asociada a la misma red obteniéndose así:

RPi 1: 192.168.100.31

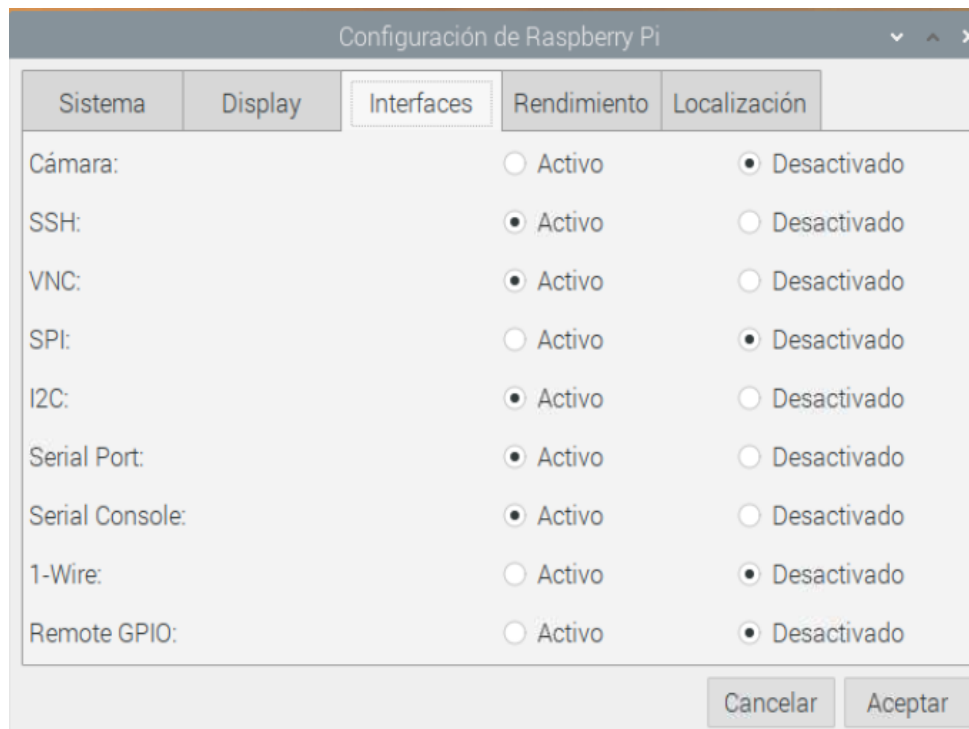
RPi 2: 192.168.100.32

RPi 3: 192.168.100.33

RPi 4: 192.168.100.34

### *3.3.5.2 Configuración de la Tarjeta Raspberry*

Para el funcionamiento correcto y la libre asociación de las Raspberry se recomienda determinar ciertos parámetros de preferencias en su configuración, la Figura 47-3 señala ítems que deben activarse y los que no se deben activar.



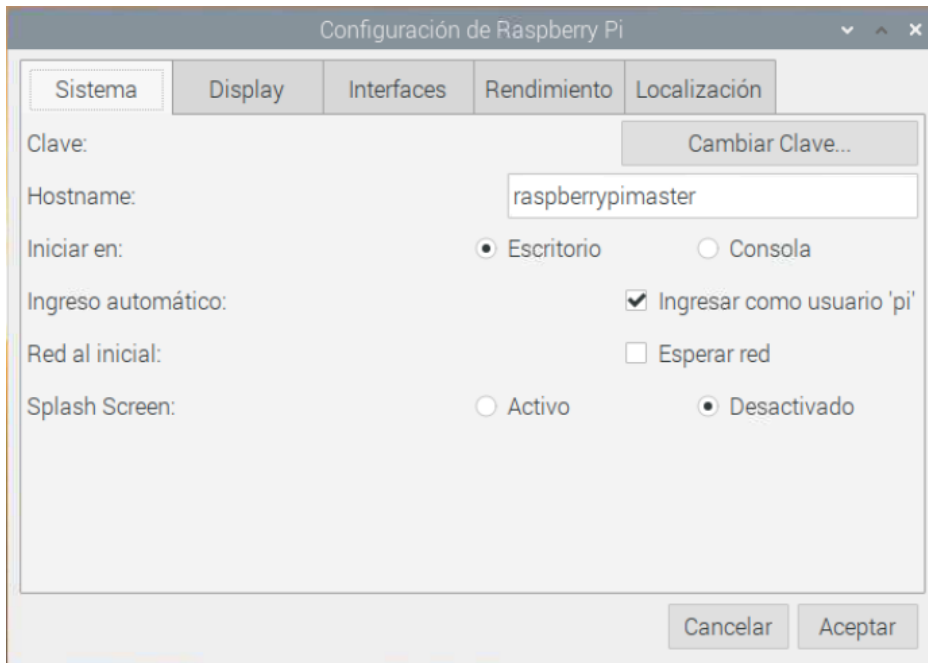
**Figura 48-3.** Preferencias del sistema.

Realizado por: Ichina, Alex, 2020

A continuación, se explica la función de los elementos activados como preferencias:

- **SSH:** Permite la conexión de los usuarios a un host remotamente, además permite mantener comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor.
- **VNC:** Este modo resulta conveniente con el uso de la Raspberry cuando no se desea interactuar con ella sin un monitor, pues se puede observar las acciones del ordenador servidor remotamente a través de un ordenador cliente.
- **I2C:** representa un puerto y protocolo de comunicación serial, permite la transferencia de bits entre dos dispositivos digitales estableciendo la trama de datos y las conexiones físicas.
- **Serial Port:** Habilita el tráfico bidireccional de información por el puerto serial.

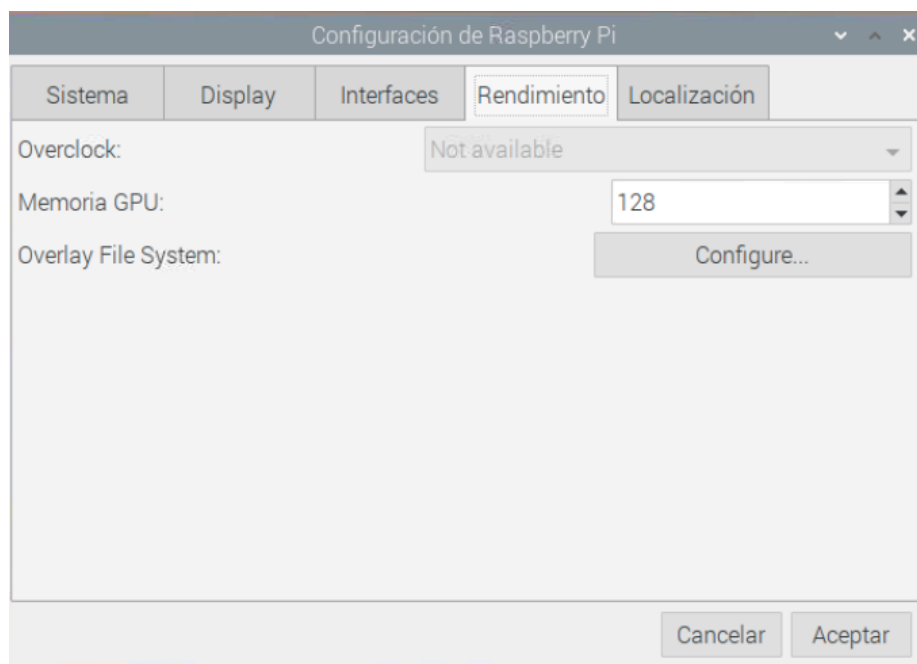
Dentro de la configuración del sistema se debe cambiar el Hostname a todas las Raspberry para poder identificarlas de mejor manera como se muestra en la figura 48-3.



**Figura 49-3.** Cambio del hostname Raspberry Master.

**Realizado por:** Ichina, Alex, 2020

En los parámetros de configuración sobre el Rendimiento del dispositivo se debe modificar los valores predeterminados de memoria GPU a 128 como se indica en la Figura 49-3 debido a que tienen que reproducir videos consecutivamente. En el caso de no hacerlo es posibles caídas de frames, o generación de pantallas verdes.



**Figura 50-3.** Ampliación de la memoria GPU

**Realizado por:** Ichina, Alex, 2020



### 3.3.5.3 Instalación de las librerías en Python

Para poder desarrollar este proyecto es necesaria la instalación de dos librerías en todas las tarjetas Raspberry para que puedan hacer el Video Wall correctamente y reproducir los videos coordinadamente.

Se puede ubicar dentro de la página de <http://www.Piwall.co.uk/> información acerca de la instalación y archivos de configuración. En el enlace <http://www.Piwall.co.uk/information/installation> se encuentran archivos para descargarlos desde el navegador de las Raspberry. Se deben seguir las siguientes instrucciones:

- En el navegador de la Raspberry, descargar el paquete pwlibs y guardarlo.
- Ejecutar en el terminal la instrucción "sudo dpkg -i /home/pi/pwlibs1\_1.1\_armhf.deb"
- En el navegador de la Raspberry descargar el paquete pwomxplayer y guardarlo.
- Ejecutar en el terminal la instrucción  
"sudo dpkg -i /home/pi/pwomxplayer\_20130815\_armhf.deb"

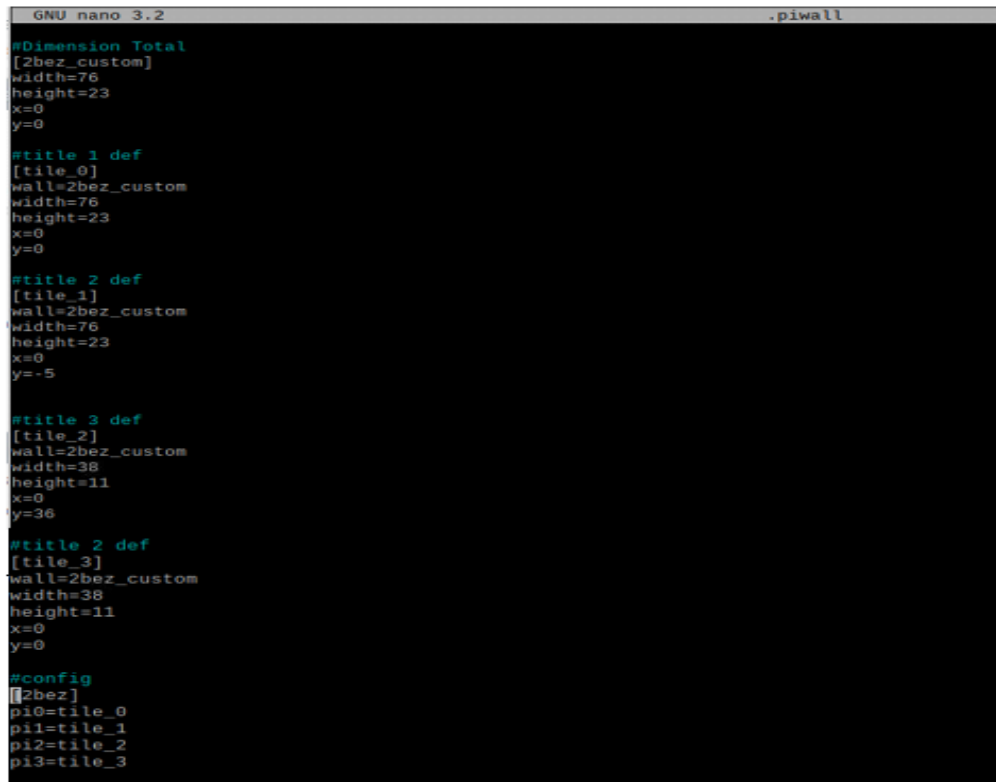
Para finalizar con la configuración de las Raspberry se debe ejecutar las siguientes acciones solo en el Master:

- "**sudo apt-get install libav-tools**" al ejecutar esta instrucción se visualizará un mensaje de que dicha librería ya está instalada y se instaló con el nombre de ffmpeg por lo que no es necesaria, esto se da en las versiones actualizadas de Raspbian, Si el comando se envía de esa manera a los esclavos en todas reflejará el mismo error por lo que se debe instalar manualmente ejecutando:
- "**sudo apt install ffmpeg**"
- "**wget http://launchpadlibrarian.net/348889634/libav-tools\_3.4.1-1\_all.deb**"
- "**sudo dpkg -i libav-tools\_3.4.1-1\_all.deb**"

### 3.3.5.4 Configuración de Tarjetas Esclavos

Se ejecuta en el terminal la instrucción "**sudo nano .Piwall**", en la que se crea un archivo; la sintaxis del punto antes del nombre hace que el archivo sea del tipo oculto. En este archivo se

registrarán las dimensiones de las pantallas y la identificación de cada una de ellas como se muestra en la Figura 50-3.



```
GNU nano 3.2 .piwall
#Dimension Total
[2bez_custom]
width=76
height=23
x=0
y=0

#title 1 def
[tile_0]
wall=2bez_custom
width=76
height=23
x=0
y=0

#title 2 def
[tile_1]
wall=2bez_custom
width=76
height=23
x=0
y=-5

#title 3 def
[tile_2]
wall=2bez_custom
width=38
height=11
x=0
y=36

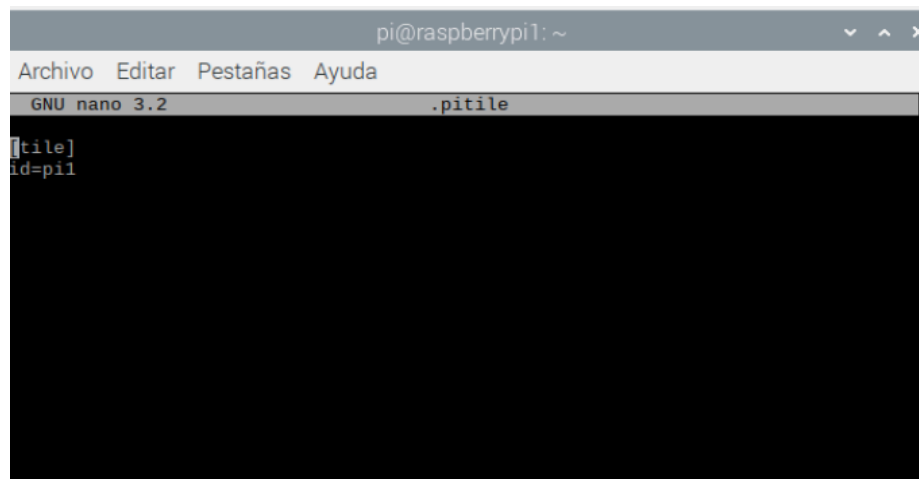
#title 2 def
[tile_3]
wall=2bez_custom
width=38
height=11
x=0
y=0

#config
[[2bez]
pi0=tile_0
pi1=tile_1
pi2=tile_2
pi3=tile_3
```

**Figura 51-3.** Edición sentencias en el archivo Piwall

Realizado por: Ichina, Alex, 2020

Se crea otro archivo denominado *pitile* en el que se identifica a cada esclavo con un nombre específico, se ejecuta en el terminal la instrucción “*sudo nano .pitile*”, se visualizará la ventana mostrada en la Figura 51-3, en la que se editará el nombre del esclavo y se guardará.



```
pi@raspberrypi1: ~
Archivo Editar Pestañas Ayuda
GNU nano 3.2 .pitile
[tile]
id=pi1
```

**Figura 52-3.** Asignación del nombre al esclavo en el archivo pitile

Realizado por: Ichina, Alex, 2020

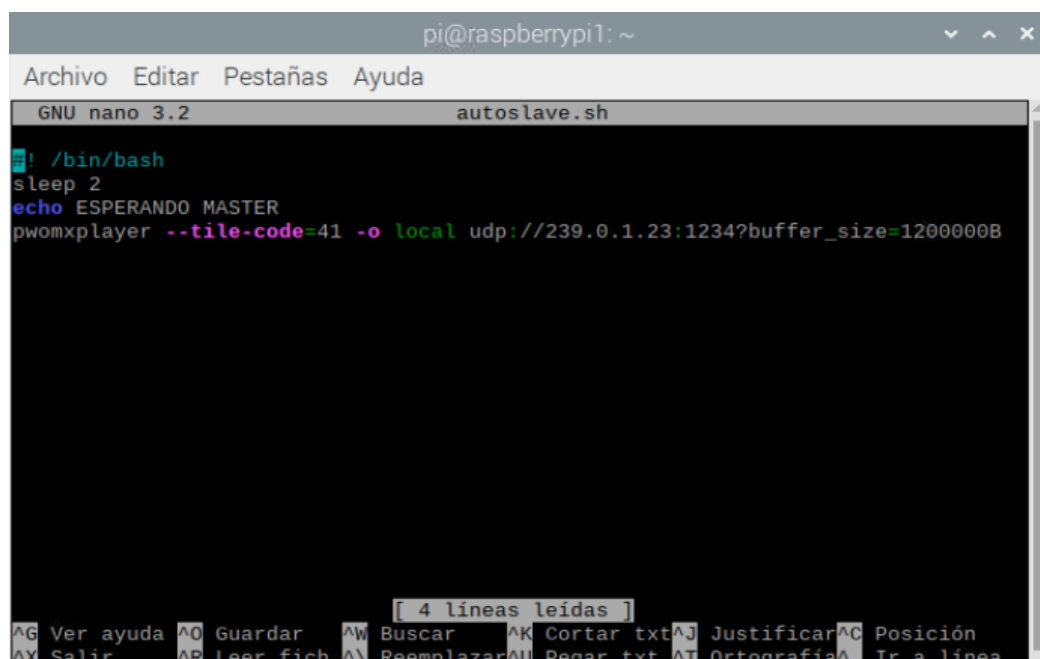
Ejecutado la sentencia “*sudo autoslave.sh*”, se crea un nuevo archivo en el que se tendrá acorde a la codificación que se observa en la Figura 52-3: el esclavo espera 2 segundos y envía un mensaje por el terminal indicando que está en espera de la Raspberry Master, posteriormente se ejecuta el comando que hará que se reproduzca la parte correspondiente del video asignado para ese esclavo.

Para el caso del ejemplo en la Figura 53-3, en la línea de código el número 41 indica que es el esclavo1 (Pi1) de cuatro esclavos en el que se muestra la esquina superior izquierda del video.

Para los otros esclavos la línea de código expresa en términos de:

- 42 indica que es el esclavo2 (Pi2) de cuatro esclavos en el que se mostrará la esquina superior derecha del video.
- 43 indica que es el esclavo3 (Pi3) de cuatro esclavos en el que se mostrará la esquina inferior izquierda del video.
- 44 indica que es el esclavo4 (Pi4) de cuatro esclavos en el que se mostrará la esquina inferior derecha del video.

La dirección udp se deja asignada por defecto, dentro del tamaño del buffer puede variar los Bytes dependiendo de la duración del video.



```
pi@raspberrypi1: ~
Archivo Editar Pestañas Ayuda
GNU nano 3.2 autoslave.sh
#!/bin/bash
sleep 2
echo ESPERANDO MASTER
pwomxplayer --tile-code=41 -o local udp://239.0.1.23:1234?buffer_size=1200000B
```

**Figura 53-3.** Edición archivo autoslave para asignación de posiciones de los esclavos

Realizado por: Ichina, Alex, 2020

```
pi@raspberrypi1: ~
Archivo  Editar  Pestañas  Ayuda
GNU nano 3.2 /home/pi/.bashrc Modificado
. ~/.bash_aliases
fi
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
bash /home/pi/autoslave.sh
```

**Figura 54-3.** Sentencia para iniciar el sistema o el terminal con el archivo autoslave.sh

Realizado por: Ichina, Alex, 2020

Como última acción de configuración en los esclavos para que el archivo que contiene el comando de la Piwall se ejecute al inicio o cuando se abre el terminal se debe editar el archivo “*bashrc*” con permisos de usuario.

Para abrir el archivo se ejecuta en el terminal “*sudo nano bashrc*” ubicándose en la última línea del código se debe añadir la sentencia *bash/home/pi/autoslave.sh* como se muestra en la figura 53-3.

### 3.3.5.5 Test de comunicación Raspberry Master - Esclavos.

Para realizar el test de comunicación se ejecuta en el terminal del Master la sentencia: “*avconv -re -i testeo3.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234*”, donde en el caso de haber configurado correctamente en los pasos anteriores se podrá visualizar la ejecución del video dividido en partes.

Resulta importante que todos los archivos se encuentren en la Raspberry Master en el path raíz /home/pi.

Con el resultado positivo del test se procede a programar las acciones del Piwall interactuando con las señales de la percha inteligente.

### ***3.3.6 Vinculación Piwall con la percha inteligente***

En esta sección se describe la configuración para agregar el Piwall como recurso de la percha inteligente.

#### ***3.3.6.1 Lectura de señales de control para el Piwall en el Master***

Mediante la descripción del código elaborado en el IDE de Thonny Python de las Figuras 54-3 y 55-3 se expone las acciones realizadas mediante programación para permitir la vinculación del Piwall a la percha inteligente. Se describe la etapa de lectura de señales de control provenientes de la aplicación informática para la conmutación de videos en el Piwall.

Es necesario importar las siguientes librerías:

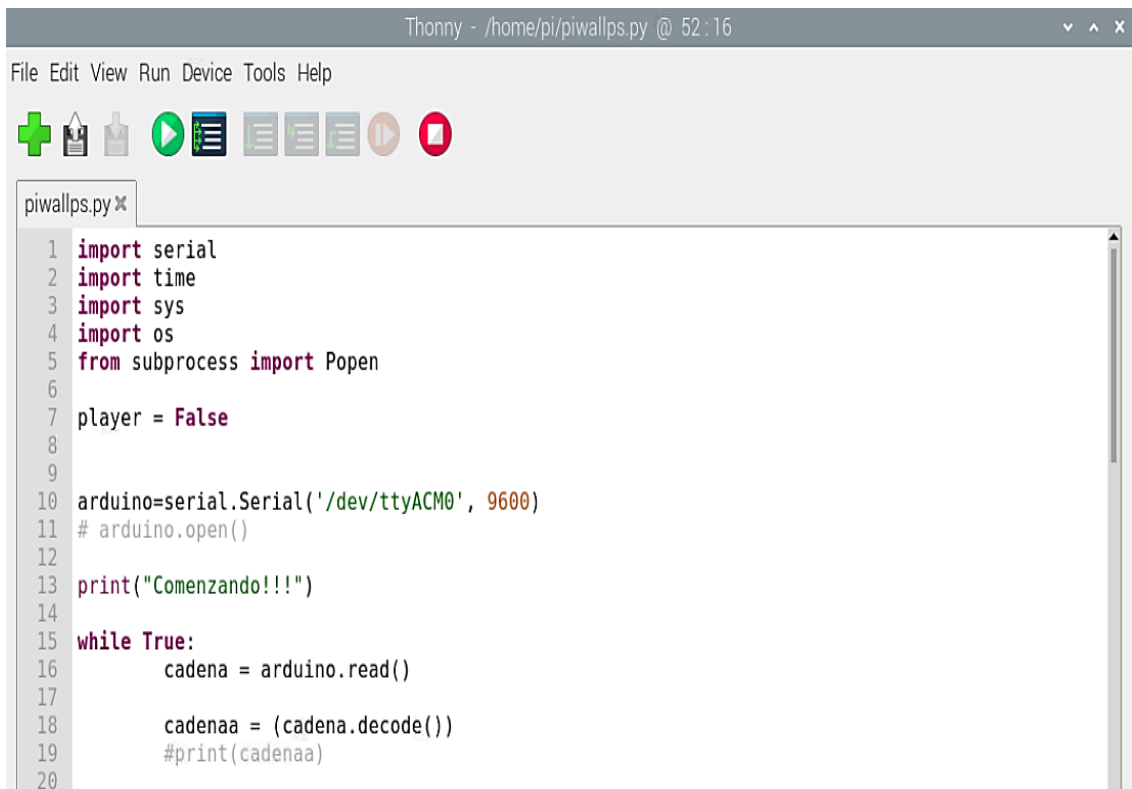
- ***Serial:*** para establecer una comunicación con un Arduino UNO B por el puerto Serial, el cual se planifica se encargará de gestionar señales de habilitación que lleguen de forma directa o indirecta desde la aplicación informática para generar la conmutación de videos fuente para el Piwall.
- ***Time:*** por si resulta necesario manejar algún tipo de retraso.
- ***Sys:*** para manipular comandos del sistema, o para enviar comandos en el terminal del sistema.

Se asigna a una variable denominada Arduino la instrucción de lectura del puerto serial incluyendo el puerto COM al que está conectado el Arduino UNO B, en este caso se establece una velocidad de comunicación de 9600 Baudios.

Se imprime un mensaje de verificación “Comenzando...!!!”.

Dentro del ciclo While, se crea una variable “cadena” a la que se le asigna el valor leído en la variable Arduino desde el puerto serial; se crea otra variable llamada “cadenaaa” en la que se

igual a al valor decodificado del mensaje que contiene la variable “cadena” para que lo muestre en su forma original, es decir como la envía el Arduino.



```
Thonny - /home/pi/piwallps.py @ 52:16
File Edit View Run Device Tools Help
+ [Icons]
piwallps.py x
1 import serial
2 import time
3 import sys
4 import os
5 from subprocess import Popen
6
7 player = False
8
9
10 arduino=serial.Serial('/dev/ttyACM0', 9600)
11 # arduino.open()
12
13 print("Comenzando!!!")
14
15 while True:
16     cadena = arduino.read()
17
18     cadenaaa = (cadena.decode())
19     #print(cadenaaa)
20
```

**Figura 55-3.** Sección (a) código Thonny Python

**Realizado por:** Ichina, Alex, 2020

En la primera sección del código se realizan las respectivas comparaciones de acuerdo con los caracteres que se esperan ser leídos desde el Arduino UNO B propuesto como pasarela para enlazar la aplicación informática desarrollada en LabVIEW con el Piwall.

Se especificó que las señales gestionadas por el Arduino UNO B servirán para conmutar videos en el Piwall, son 5 videos los asignados testeo1, testeo2, testeo3, testeo4, testeo5 como se los ha denominado. Donde el primero corresponde a un video de publicidad de la boutique y los otros cuatro serán los correspondientes a cada una de las prendas en exhibición, en este caso se cierra a cuatro prendas como se especificó en el alcance de la ejecución del proyecto.

Tomando una de las comparaciones ejecutadas en el código se dice que si el valor leído en el puerto es “5” primero se imprima un aviso de que la señal corresponde al video de promoción o publicidad y luego ejecute en el terminal la sentencia con la que la Raspberry Master indicará a los esclavos que reproduzcan su aparte asignada del video y terminado esto se cierre el reproductor omxplayer.

El mismo proceso se repite para las otras comparaciones con otros caracteres, los cuales representan otros videos; y por último se cierra la comunicación serial con el Arduino.

```
21     if cadenaaa=='5':
22         #os.system('killall omxplayer.bin')
23         print("Video Promocion")
24         os.system('avconv -re -i testeo1.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
25         os.system('killall omxplayer.bin')
26         player = False
27     elif cadenaaa=='1':
28         #os.system('killall omxplayer.bin')
29         print("Video Blusa")
30         os.system('avconv -re -i testeo2.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
31         os.system('killall omxplayer.bin')
32         player = False
33     elif cadenaaa=='2':
34         #os.system('killall omxplayer.bin')
35         print("Video Panton")
36         os.system('avconv -re -i testeo3.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
37         os.system('killall omxplayer.bin')
38         player = False
39     elif cadenaaa=='3':
40         #.system('killall omxplayer.bin')
41         print("Video Interiores")
42         os.system('avconv -re -i testeo4.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
43         os.system('killall omxplayer.bin')
44         player = False
45     elif cadenaaa=='4':
46         #os.system('killall omxplayer.bin')
47         print("Video Zapatos")
48         os.system('avconv -re -i testeo5.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
49         os.system('killall omxplayer.bin')
50         player = False
51
52     arduino.close()
```

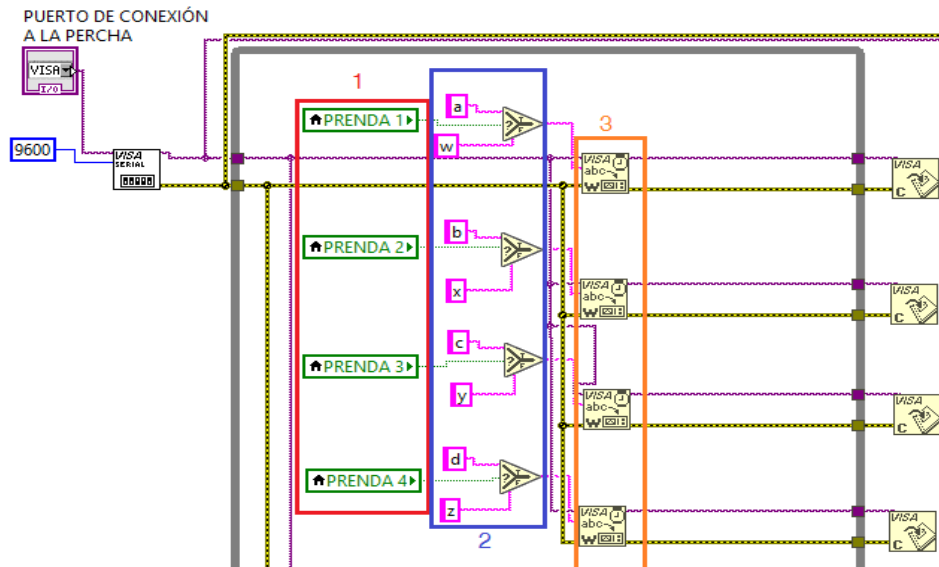
**Figura 56-3.** Sección (b) código Thonny Python

Realizado por: Ichina, Alex, 2020

### 3.3.6.2 Envío de señales para el control del Piwall desde LabVIEW

El Arduino UNO A toma las señales desde el entorno físico de la percha inteligente y se las direcciona a la aplicación informática desarrollada en LabVIEW, en este punto se establece el procedimiento desde LabVIEW para gestionar señales digitales que sirvan como variables de control en el proceso de conmutación de los videos en el Piwall, éstas replican el estado de los ganchos para sincronizar el funcionamiento del sistema.

En la Figura 56-3 se puede apreciar que se toman las señales booleanas de los indicadores de la etapa de monitoreo local en forma de variables locales (1) PRENDA 1, PRENDA 2, PRENDA 3, PRENDA 4 como habilitantes de un bloque de decisión que entrega por verdadero un carácter y por falso otro (2). Los caracteres son direccionados hacia un bloque **Visa Write** que permite escribir en el puerto serial (3), en este caso al COM al que está iniciada la comunicación con el Arduino UNO A. Por último con el **Visa Close** se cierra el bucle.



**Figura 57-3.** Escritura en el puerto Serial desde LabVIEW

Realizado por: Ichina, Alex, 2020

De esta manera la aplicación informática proporciona por medio de la codificación de caracteres el estado de las variables de los ganchos a través del puerto serial, la misma que se direcciona al Arduino UNO A para procesarla.

Mediante las líneas de código expuestas en la Figura 57-3 en primera instancia se gestiona un conjunto de cuatro variables digitales declaradas como salidas para luego realizar la lectura del puerto serial y obtener la información que está siendo enviada desde LabVIEW para procesarla.

```

void setup()
{
  Serial.begin(9600);
  pinMode(pinled1, OUTPUT);
  pinMode(pinled2, OUTPUT);
  pinMode(pinled3, OUTPUT);
  pinMode(pinled4, OUTPUT);
}
void loop()
{
  if(Serial.available())
  {
    var=Serial.read();
    if(var=='a')
    {digitalWrite(pinled1, HIGH);}
    if(var=='w')
    {digitalWrite(pinled1, LOW);}
    if(var=='b')
    {digitalWrite(pinled2, HIGH);}
    if(var=='x')
    {digitalWrite(pinled2, LOW);}
    if(var=='c')
    {digitalWrite(pinled3, HIGH);}
    if(var=='y')
    {digitalWrite(pinled3, LOW);}
    if(var=='d')
    {digitalWrite(pinled4, HIGH);}
    if(var=='z')
    {digitalWrite(pinled4, LOW);}
  }
}

```

**Figura 58-3.** Lectura y procesamiento del puerto serial en el Arduino UNO A

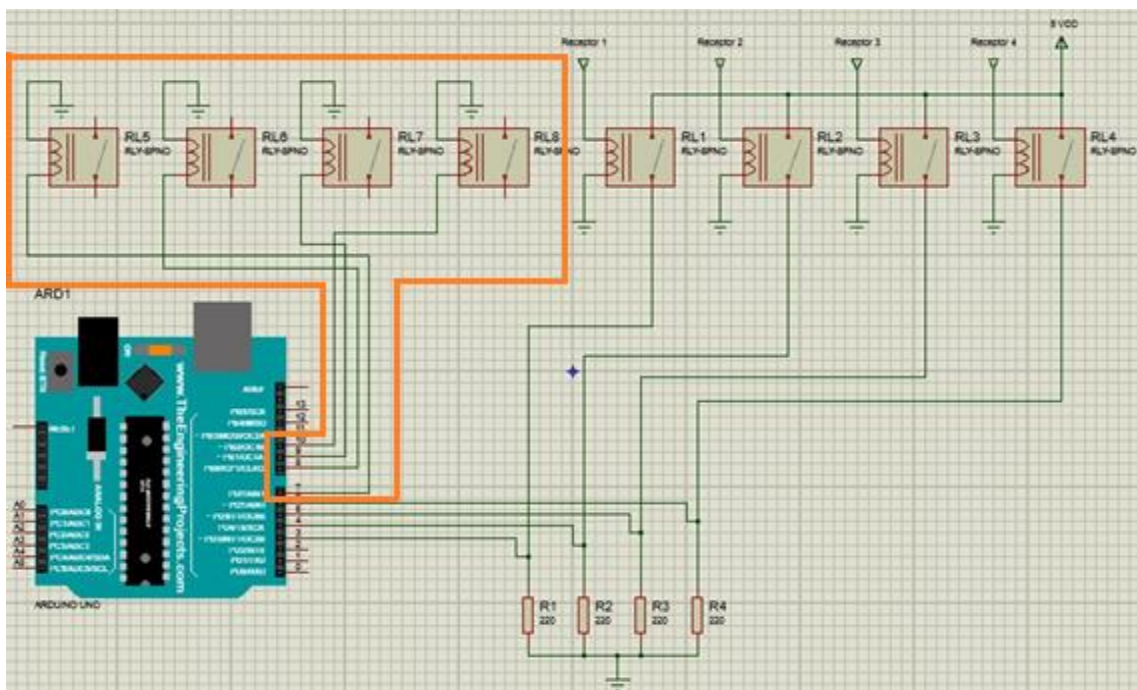
Realizado por: Ichina, Alex, 2020



Se observa que, a través del puerto serial se obtendrán los caracteres que envía LabVIEW y por medio de comparaciones según el caracter codificado se gestionará el estado de las salidas para escribir un *HIGH* o un *LOW* en el pin correspondiente del Arduino UNO A.

La Figura 58-3, presenta la segunda sección del circuito, en el que se observa que los pines del Arduino UNO B declarados como salidas gestionarán las señales de polarización para un segundo bloque de relés, donde la conmutación de los contactos de éstos relés son direccionados y configurados como entradas hacia un Arduino UNO B.

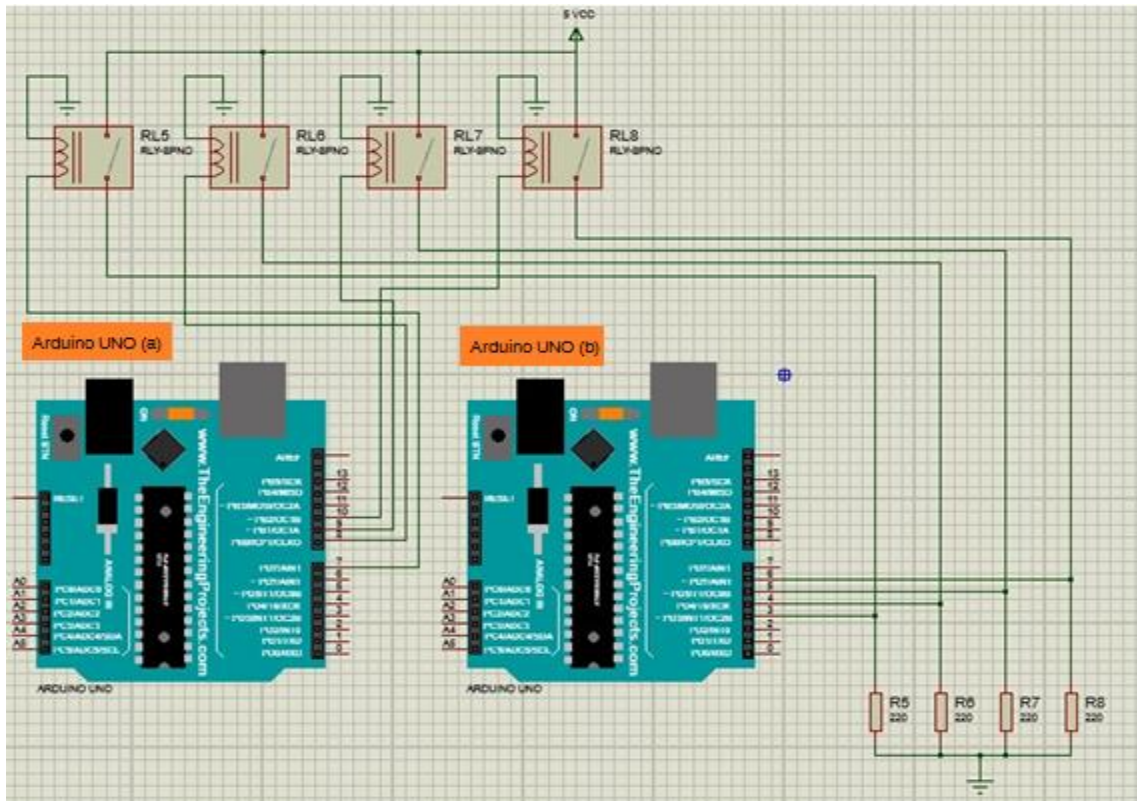
La Figura 58-3 representa el recurso que se propuso como recurso para vincular el control de la aplicación informática sobre el Piwall para la gestión de los videos en las pantallas.



**Figura 59-3.** Diagrama de conexiones señales de control para conmutación del Piwall

**Realizado por:** Ichina, Alex, 2020

La figura 59-3 muestra los dos Arduino UNO empleados en el proyecto, donde se puede considerar que el Arduino UNO A es la fuente de información para el Arduino UNO B en conjunto funcionan como medios de adquisición y pasarelas para el tráfico de información.



**Figura 60-3.** Diagrama de conexiones plataformas Arduino UNO

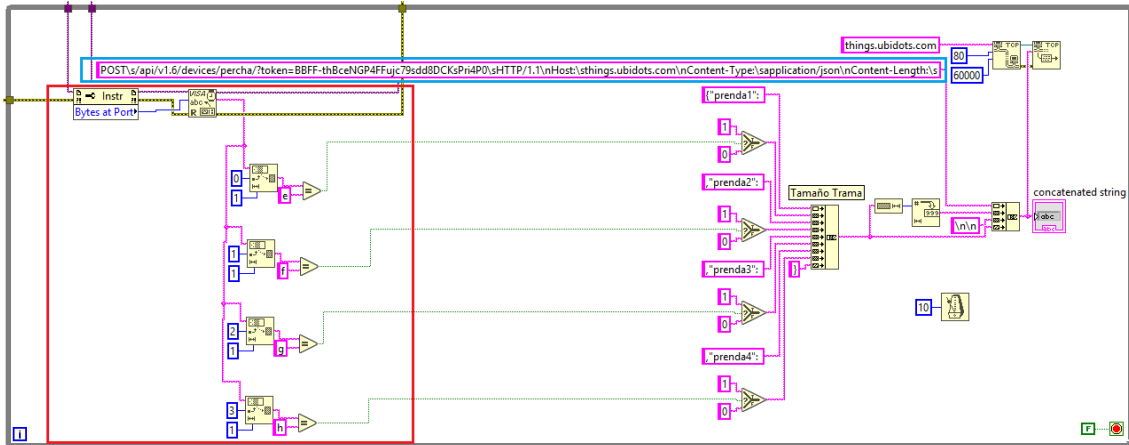
Realizado por: Ichina, Alex, 2020

El código empleado en la programación del Arduino UNO B se fundamenta en las mismas condiciones del Arduino UNO A pues el ciclo se repite, se adquiere una señal digital gestionada a través de los relés, según el análisis de éstas se imite información codificada al puerto serial, la misma que es leída a través del serial del Master como se lo reviso en el ítem 2.3.6.1.

### 3.3.7 Vinculación de la percha inteligente al IoT

Una vez que en Ubidots se ha montado el dispositivo que representa la percha inteligente física con las respectivas variables se realiza desde la aplicación informática la programación necesaria para vincularla con dicha plataforma.

La figura 60-3 describe los bloques de programación que intervienen para realizar el anclaje de la aplicación desarrollada en LabVIEW con la plataforma Ubidots.



**Figura 61-3.** Bloques de programación vinculación LabVIEW - Ubidots

**Realizado por:** Ichina, Alex, 2020

La sección resaltada en el cuadro rojo de la Figura 60-3, es una réplica del proceso de adquisición de datos del Arduino UNO A hacia LabVIEW. El estado de las variables digitales que se leen en el microcontrolador representan el estado de los ganchos de la percha inteligente y son éstos los que se desean monitorear a través de Ubidots, es así que esta sección se considera es la fuente de información que deberá ser montada a la plataforma IoT.

Siguiendo con el análisis del bloque de programación se observa se usan dos nuevos recursos el **TCP Open Connection** que abre una conexión de red TCP con la dirección y el puerto remoto o el nombre del servicio y **TCP Write** que escribe datos en una conexión de red TCP. Cada uno de estos bloques maneja parámetros específicos que se los detalla a continuación.

Para la configuración del bloque TCP Open Connection se añaden los siguientes parámetros:

- La dirección con la que se desea establecer una conexión que puede estar en formato de punto ip o ser el nombre de un host, en este caso se emplea el host things.ubidots.com.
- Puerto remoto de comunicación que admite valores numéricos o de cadena en este caso se asigna el puerto 80 para uso de interfaz Ethernet.
- Timeout representa un tiempo en milisegundos que la función espera completar y devolver un error. El valor predeterminado es 60000 ms.

Para la configuración del bloque TCP Write se añaden los siguientes parámetros:

- Identificador de Conexión define una referencia de conexión de red que identifica de forma exclusiva la conexión TCP. Como se observa este pin se conecta al bloque TCP Open Connection que tiene como salida el mencionado identificador.
- Datos de ingreso, en este terminal se añade lo que se desea escribir en el puerto de conexión y pueden existir varias técnicas para realizarlo, en este caso se emplea el envío de un mensaje predeterminado con un encabezado de tamaño fijo que describa el mensaje.

El método empleado para armar la trama al host de Ubidots se denomina POST. La trama para ser montada en el bloque TCP Write requiere expresarse como una cadena de caracteres, la misma que se arma de la siguiente manera con la ayuda de un bloque **Concatenate String**:

- **Encabezado**  
 POST /api/v1.6/devices/percha/?token=BBFF-thBceNGP4FFujc79sdd8DCKsPri4P0  
 HTTP/1.1
- **Denominación del Host** al que se escribe  
 Host: things.ubidots.com
- **Tipo de contenido**  
 Content-Type: application/json
- **Dimensión de la trama**  
 Content-Length: 53
- **Estado de las variables:** en este último parámetro se puede observar que entre comillas se encuentran los nombres de las variables creadas en el dispositivo PERCHA en Ubidots y a continuación se asocia un 1 o 0 para definir el estado de la variable, depende claramente de la señal adquirida por el Arduino UNO A.  
 {"prenda1": 0,"prenda2": 0,"prenda3": 0,"prenda4": 0}

De esta manera se arma la trama que llega a Ubidots para el análisis del estado de las variables enviadas y actualizar las residentes conjugando los recursos del Dashboard para crear un ambiente gráfico de monitoreo remoto.

### ***3.3.8 Desarrollo de la aplicación para la gestión de la Boutique***

En este punto se revisará el procedimiento para la creación de la base de datos y su vinculación a LabVIEW.

### 3.3.8.1 Diseño de la base datos

Como uno de los requerimientos planteados para desarrollo del sistema es la necesidad de manejar existencias por medio del registro de un inventario y ventas de productos, esto involucra la creación de una base de datos razón por la cual para el diseño de la misma se empleó el modelo entidad relación y se planteó el uso de Microsoft Access como gestor de base de datos.

Para el diseño de la base de datos se consideró manejar la información mediante tablas relacionadas como manifiesta el modelo de entidad relación, de esta forma se crearon las siguientes tablas.

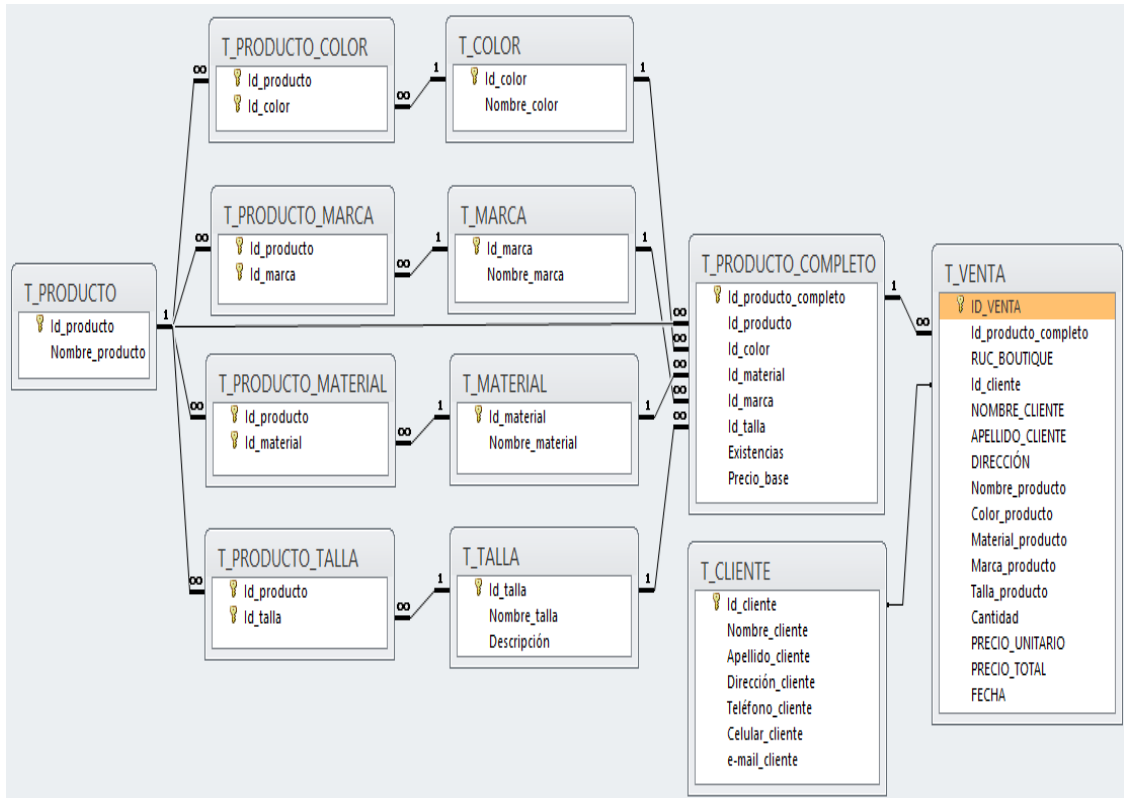
**Tabla 7-3:** Tablas para distribución de la información

<b>Tablas principales</b>	<b>Descripción</b>
T_producto	Contiene la denominación de las prendas de vestir existentes. Ejemplo: Blusa
T_color	Contiene la gama de colores. Ejemplo: Rojo
T_marca	Contiene los nombres de las diferentes marcas proveedoras de productos. Ejemplo: Hollister
T_material	Contiene los posibles materiales empleados en la confección de prendas de vestir. Ejemplo: Algodón
T_talla	Contiene las asignaciones de tallas en las que se puede ofrecer las prendas de vestir. Ejemplo: Large.
T_producto_color	Conjuga los atributos de la tabla de productos y color generando información más específica. Ejemplo: Blusa roja.
T_producto_marca	Conjuga los atributos de la tabla de productos y marca generando información más específica. Ejemplo: Blusa NIKE.
T_producto_material	Conjuga los atributos de la tabla de productos y material generando información más específica. Ejemplo: Blusa algodón
T_producto_talla	Conjuga los atributos de la tabla de productos y talla generando información más específica. Ejemplo: Blusa large
T_producto_completo	Contiene la información específica de cada uno de los productos de la boutique incluyendo las existencias de cada uno de ellos. Ejemplo: Blusa, roja, NIKE, algodón, large, stock 10.
T_Cliente	Contiene los datos informativos de los clientes de la boutique

T_venta	Contiene los datos de las ventas realizadas en la boutique.
---------	-------------------------------------------------------------

Realizado por: Ichina, Alex, 2020

La figura 61-3 representa el modelo entidad relación para la base de datos fundamentado en la Tabla 7-3, donde se encuentran asignados los atributos de cada uno de ellas. Además se visualiza la asignación de las diferentes claves PK en cada tabla para asegurarse de la veracidad de la información que se almacena.



**Figura 62-3.** Modelo entidad relación

Realizado por: Ichina, Alex, 2020

De esta manera se tiene creada la base de datos independientemente, pero el objetivo de esto es vincularla a la aplicación informática desarrollada.

### 3.3.8.2 Vinculación de la Base de Datos con LabVIEW

La potencialidad de LabVIEW para vincularse con otras aplicaciones es utilizada en este caso para enlazar la base de datos creada en Microsoft Access para gestionarla desde la aplicación informática desarrollada en LabVIEW.

La Figura 62-3 contiene el panel frontal del VI en el que se desarrolló la interfaz gráfica para el ingreso de los productos, si se observa detenidamente se puede relacionar los campos de la tabla T\_producto\_completo con los recursos gráficos que se incluyen en la pantalla.

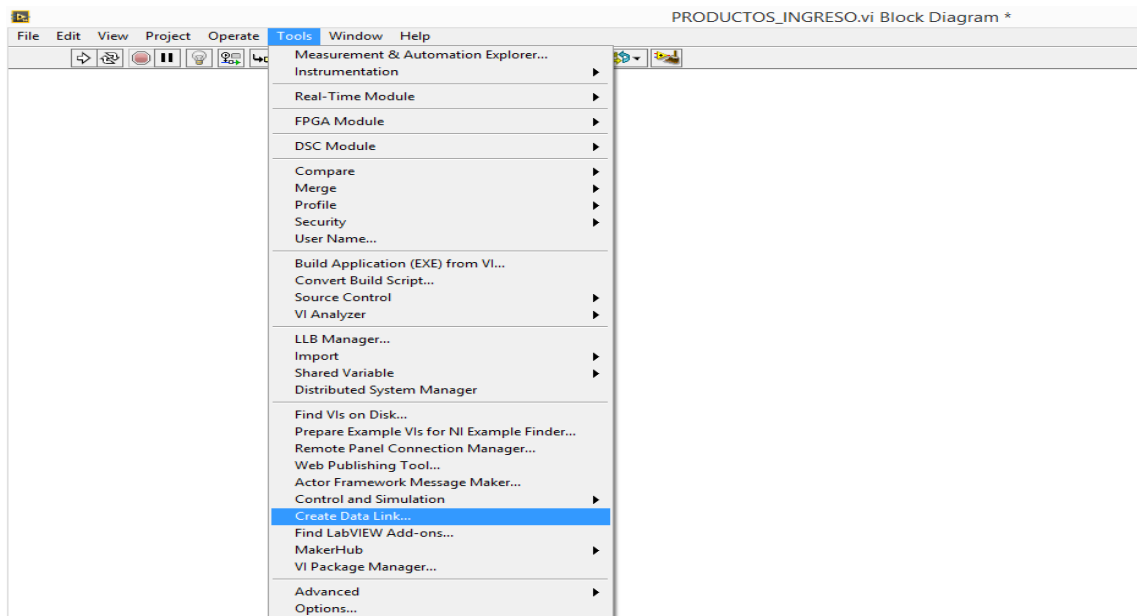


**Figura 63-3.** Desarrollo Interfaz gráfica para ingreso de productos

**Realizado por:** Ichina, Alex, 2020

Al generar la base de datos en Microsoft Access, se guarda en formato .mdb, el mismo que no se puede vincular directamente con LabVIEW, se requiere generar un archivo de la base de datos en formato .udl.

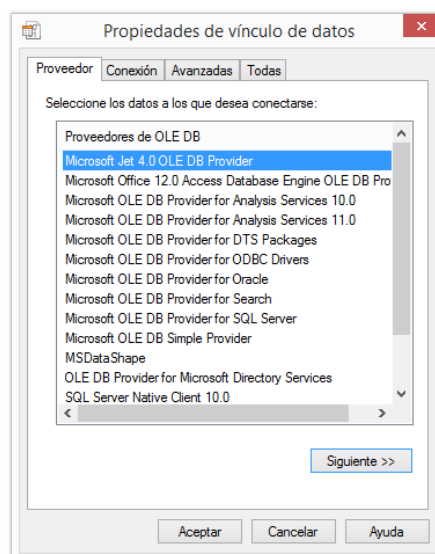
LabVIEW permite configurar un puente para el tráfico de información con la base de datos y crea el archivo .udl, para ello como se muestra en la Figura 63-3 se ingresa en la pestaña de **Tools** y se ubica la opción **Create Data Link** es decir se inicia la creación de un enlace de datos.



**Figura 64-3.** Creación del enlace de datos

**Realizado por:** Ichina, Alex, 2020

Una vez seleccionada la opción para creación del enlace de datos se debe establecer el tipo de OLE DB, es decir la forma de enlace o inserción de objetos en la base de datos diseñada para el proyecto. Esta vez se seleccionó Microsoft Jet 4.0 OLE DB Provider compatible con Microsoft Access.



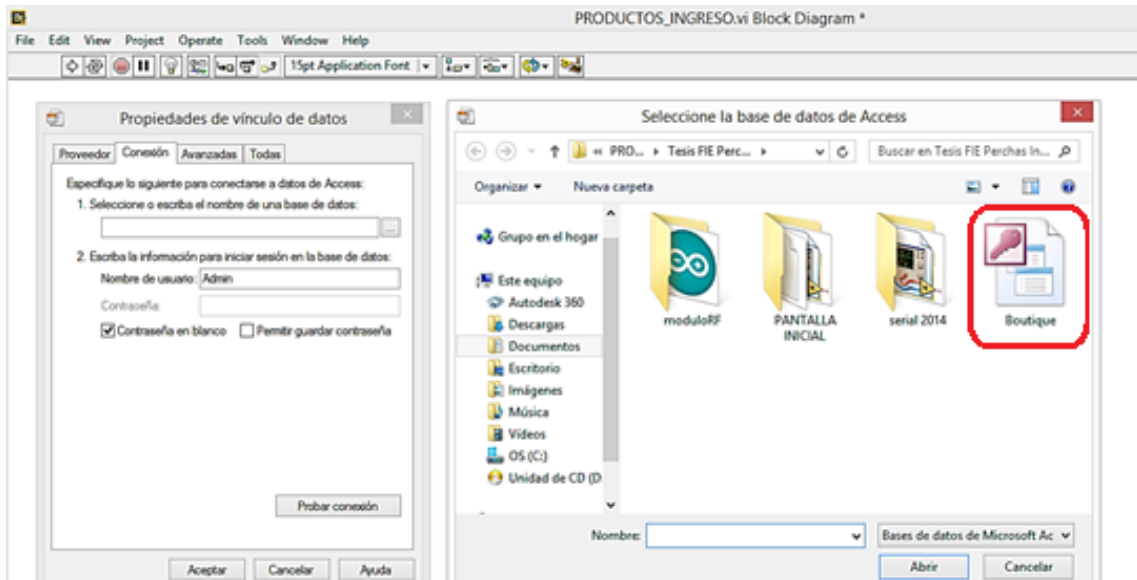
**Figura 65-3.** Enlace e incrustación de objetos para base de datos (OLE DB)

**Realizado por:** Ichina, Alex, 2020

El OLE DB seleccionado apertura una nueva ventana como se muestra en la Figura 65-3 donde se pide seleccionar la base de datos creada en Microsoft Access con .mdb y se guarda el path de



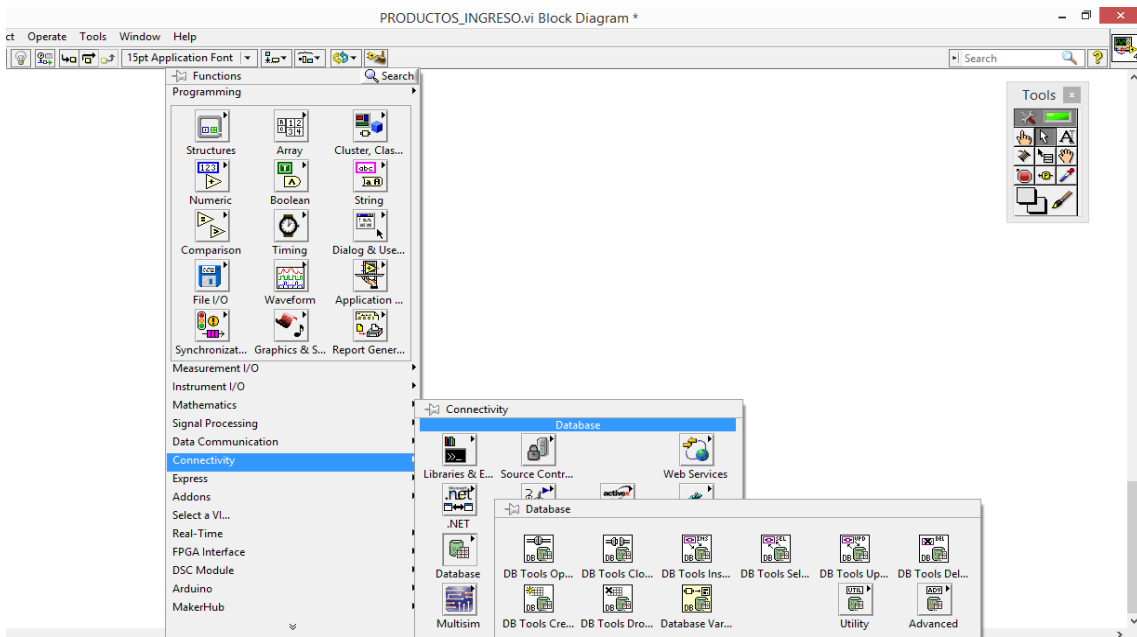
su ubicación, existe la posibilidad de probar la conexión si esta resulta correcta se la acepta y se genera el archivo .udl.



**Figura 66-3.** Asignación del .mdb en el OLE DB

Realizado por: Ichina, Alex, 2020

Creado el archivo .udl desde la plataforma LabVIEW se puede empezar a trabajar en la programación necesaria para interactuar desde la aplicación informática gestionando información para la base de datos. LabVIEW cuenta con una librería específica, Figura 66-3.



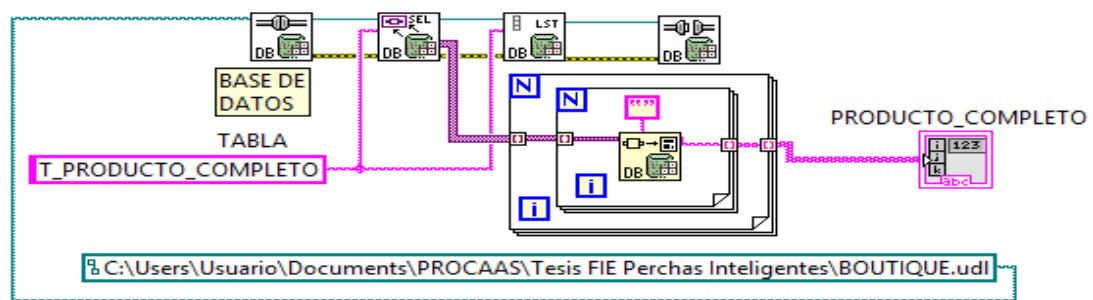
**Figura 67-3.** Recursos librería Database

Realizado por: Ichina, Alex, 2020

El conjunto de bloques disponibles dentro de la librería Database proporciona los recursos necesarios para ejecutar en la base de datos acciones de ingreso, eliminación, búsqueda y modificación de datos.

A continuación, se presentan extractos del programa con algunos de los principales procedimientos:

*Visualización:* Este proceso se lo identifica en la Figura 67-3 y contiene un cierto conjunto de bloques e instrucciones.



**Figura 68-3.** Bloque de programación – proceso de visualización

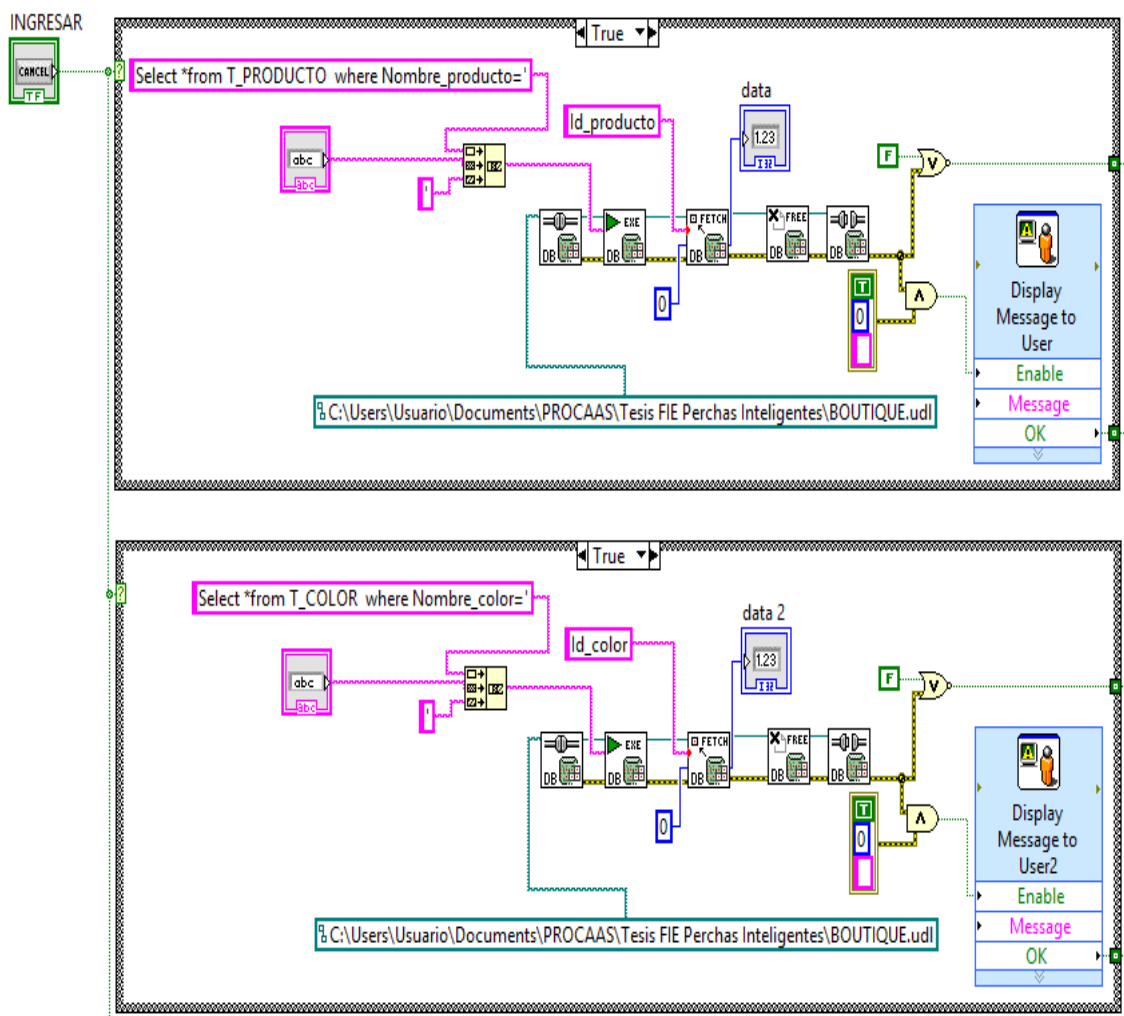
Realizado por: Ichina, Alex, 2020

Dentro del proceso de visualización destacan los bloques:

- **DB Tools Open Connection:** Este bloque es el que se encarga de iniciar la comunicación con la base de datos, se observa que en sus parámetros de ingreso se encuentra el Path de la ubicación de la base de datos en su formato .udl.
- **DB Tools Select Data:** Permite en sus parámetros de configuración establecer de la base de datos que se apertura con que tabla se va a trabajar. En la Figura 67-3 se observa que este bloque de proceso se ha extraído del proceso de visualización de la tabla T\_producto\_completo.
- **DB Tools List Column:** Este bloque entrega el listado de columnas que conforman la tabla seleccionada.
- **Database Variant to Data:** Éste es un bloque especial actúa como un traductor, la información que recibe de la base de datos, la convierte en un formato capaz de ser interpretado por LabVIEW o cualquier otra aplicación, en esta aplicación el bloque trabaja en conjunto con dos estructuras de repetición *for* pues al tratarse de tablas las que se van a visualizar se requiere recorrer filas y columnas y esto se lo logra con dichas estructuras.

- **DB Tools Close Conection:** Al abrir la comunicación con la base de datos se utiliza este bloque para cerrar el ciclo, pues representa el cierre de la comunicación

**Búsqueda:** Este proceso se lo identifica en la Figura 68-3 representa la búsqueda de un producto por su denominación, se lo emplea previo al ingreso. Se verifica su existencia en la base de datos con el fin de no repetir productos de igual denominación y se procede a guardar. Esta comprobación se la hace para todos los parámetros de un producto completo en el ejemplo se han citado dos casos, la verificación del producto y el color. En el caso de no existir uno de los parámetros en las tablas de registro individual no se procede al proceso de ingreso.



**Figura 69-3.** Bloque de programación – proceso de búsqueda

Realizado por: Ichina, Alex, 2020

En este proceso reaparecen los bloques, DB Tools Open Conection y DB Tools Close Conection que se explicó anteriormente su función, además aparecen nuevos bloques que a continuación se los describe:

- **DB Tools Execute Query:** Este bloque emplea una cadena de consulta para interactuar con la base de datos, se conoce a esto como instrucciones Query por ejemplo:

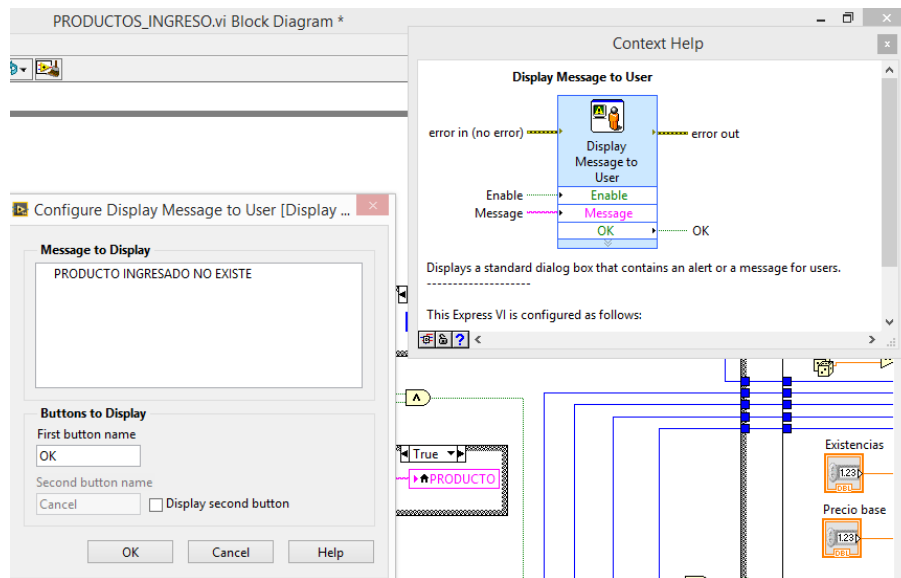
*Select \*from T\_PRODUCTO where Nombre\_producto='.....',*

Interpretando la sentencia diría, Seleccione la tabla T\_producto, el atributo Nombre\_producto que tenga asignado la denominación que sería una variable de ingreso desde la interfaz.

- **DB Tools Fetch Element Data VI:** Si el resultado del Query es positivo, es decir se halló el producto permite extraer de la fila del producto encontrado un campo descriptivo específico. En este caso se observa que en los parámetros de configuración del bloque se coloca el Id\_producto que representa el atributo a extraerse y el 0 representa que el dato extraído será del tipo numérico y como señal de salida en el bloque data que contendrá el dato.

Si se desea obtener más datos de la fila del producto hallado se debe replicar este bloque especificando el atributo que se desea.

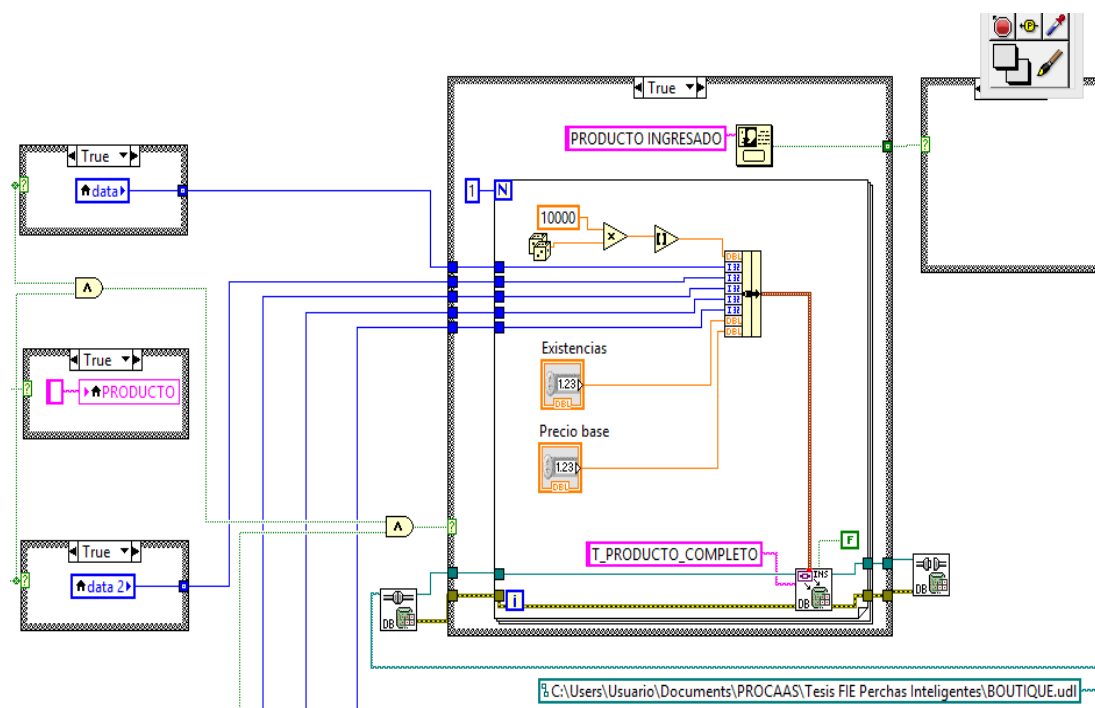
- **DB Tools Free Object VI:** Libera un objeto destruyendo su referencia asociada y devuelve un objeto de referencia diferente.
- **Display message to User:** Con el fin de generar una aplicación informática que interactúe con el usuario se añaden bloques de aviso con mensajes sobre eventos que se den dentro del programa. Es bloque de display message to user en este caso se lo utiliza para que en caso de ingresar un parámetro que no sea válido el usuario pueda saberlo por medio de un aviso, se lo configura como se observa en la Figura 69-3 citando un ejemplo.



**Figura 70-3.** Bloque de programación – Mensajes para el usuario

Realizado por: Ichina, Alex, 2020

**Ingreso:** El procedimiento para el ingreso de valores a la base de datos es el mismo para todas las tablas, la información que llega a este bloque para su registro es garantizada al pasar el filtro del bloque de búsqueda que se expuso anteriormente, en la Figura 70-3 se expone el proceso de ingreso de atributos en la tabla T\_producto completo.



**Figura 71-3.** Bloque de programación – proceso de búsqueda

Realizado por: Ichina, Alex, 2020

Para el ingreso de los atributos de una tabla se requieren bloques específicos que se detallan a continuación:

- ***DB Tools Open Connection, DB Tools Close***
- ***Bundle***: Este bloque es el que concentra la información de los atributos que se desean ingresar a una tabla. Admite todo tipo de datos y en su salida entrega un clúster que se direcciona a un bloque de la librería Database.
- ***DB Tools Insert Data***: Es el encargado de direccionar toda la información encapsulada en el bundle el atributo de entrada señala la tabla a la que direcciona la información.

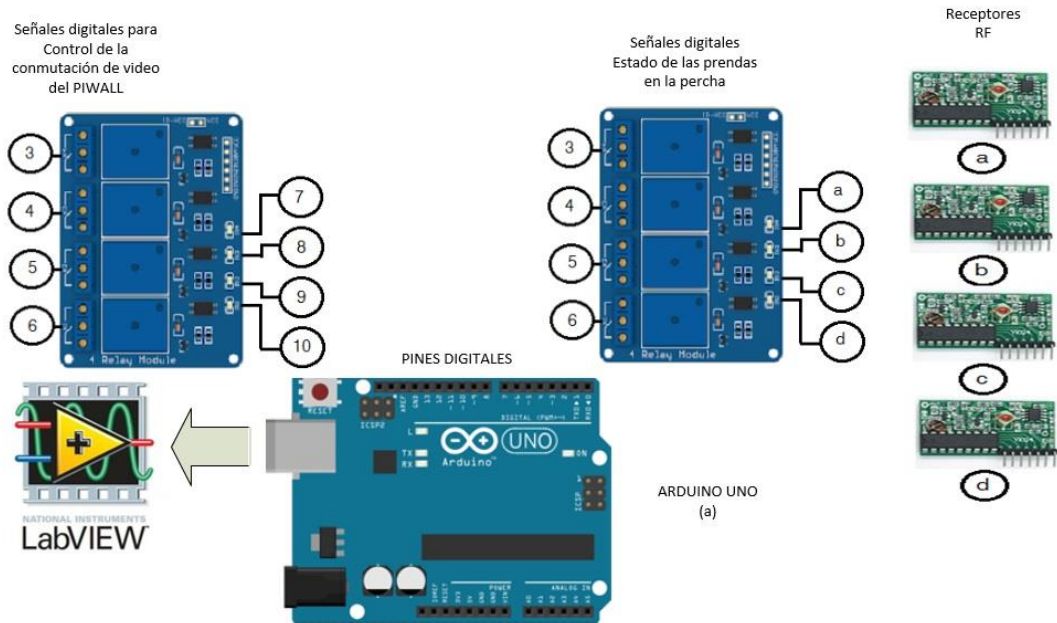
Este proceso de ingreso en el manejo de tipo de datos es importante que sea coordinado con los que se declare en la base de datos, si no hay coincidencia entre el tipo de dato de la variable declarada en LabVIEW con la variable del destino se genera un error.

### **3.4 Implementación y pruebas de funcionalidad**

#### ***3.4.1 Circuito electrónico de adquisición de datos***

El diseño del sistema que se implementó con los circuitos electrónicos; como se muestra en la Figura 71-4 representa el esquema general de las conexiones de los elementos físicos del sistema que da cumplimiento a tres funciones esenciales que son:

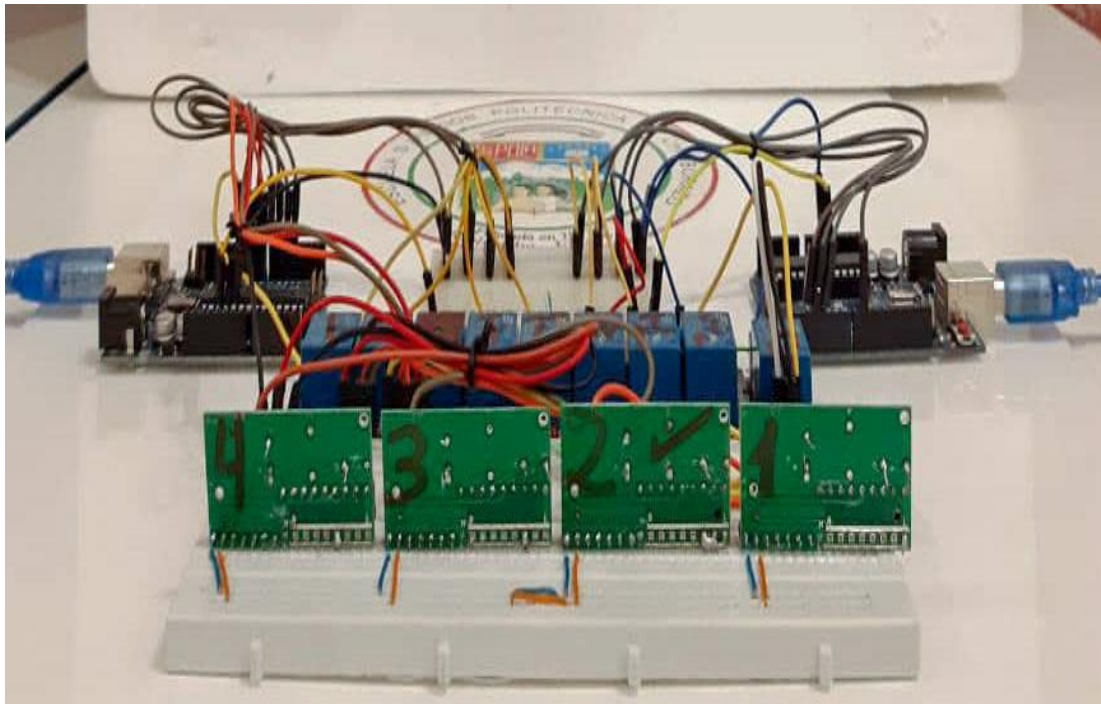
- La adquisición de señales de los receptores a través de la intervención del módulo de relé hacia los pines digitales determinados como entradas en la plataforma de Arduino que contiene el microcontrolador.
- Se representa la forma de conexión por medio del puerto serial hacia la interfaz desarrollada en LabVIEW para el flujo bidireccional de información.
- La adecuación de los puertos del microcontrolador por medio del módulo de relés para gestionar las señales de control para la conmutación de los videos del Piwall.



**Figura 72-4.** Circuito electrónico del sistema de percha inteligente

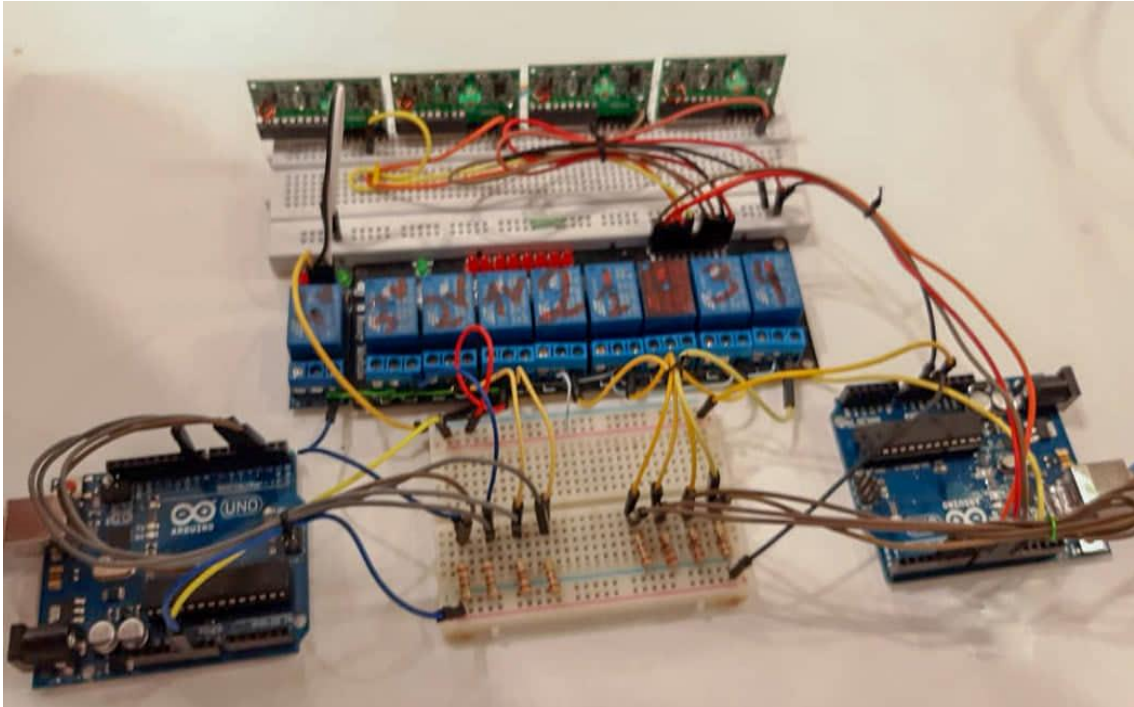
**Realizado por:** Ichina, Alex, 2020

Con la representación gráfica del sistema, se procedió a implementar el circuito como se lo muestra en la figuras 72-4 y 73-4, donde se mantiene un orden de conexiones y ubicación de los elementos como un sistema de etapas fáciles de identificar e implementar.



**Figura 73-4.** Circuito electrónico del sistema

**Realizado por:** Ichina, Alex, 2020



**Figura 74-4.** Circuito electrónico del sistema

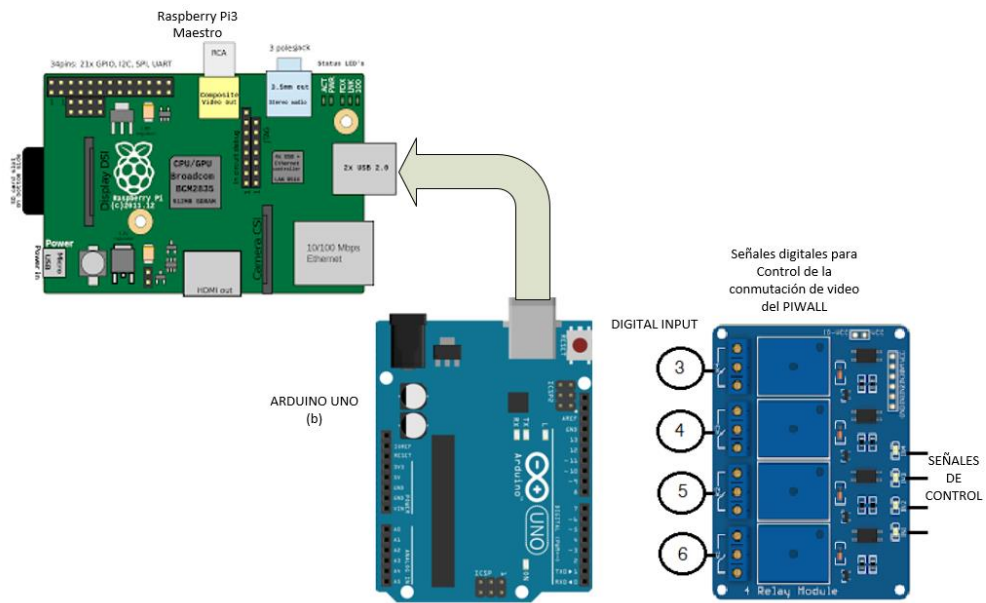
**Realizado por:** Ichina, Alex, 2020

### ***3.4.2 Circuito señales de control para la tarjeta Master del Piwall***

Otra de las etapas definidas en el diseño, se encuentra lo expuesto en la Figura 74-4, siendo sus funciones las siguientes:

- La integración de señales de control para la gestión de video en el Piwall donde el segundo módulo de relés controlado por el primer Arduino UNO A interactúa a través de los contactos abiertos como fuente de señales discretas para el segundo Arduino UNO B que las recibe como entradas para su procesamiento e impresión codificada hacia el puerto serial.
- El Arduino UNO B mantiene comunicación con la Raspberry Master del Piwall a través del puerto serial, con un flujo de información unidireccional.
- La Raspberry Master lee y decodifica la información recibida por medio de la programación elaborada en Thonny Python, además según el análisis de datos se emite a la red de esclavos las sentencias correspondientes para coordinar el trabajo de cada uno de los esclavos.

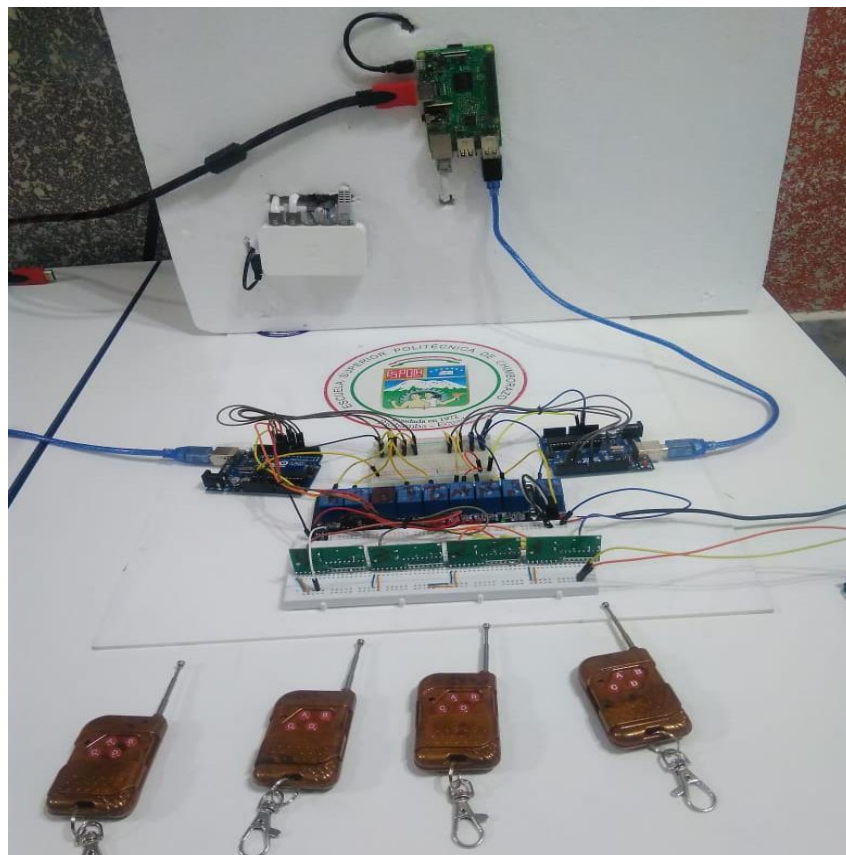




**Figura 75-4.** Circuito electrónico del sistema

Realizado por: Ichina, Alex, 2020

La figura 75-4 representa las dos etapas del sistema expuestas en los literales 3.4.1 y 3.4.2 implementado.



**Figura 76-4.** Conjunto electrónico de adquisición, procesamiento y envío de información

Realizado por: Ichina, Alex, 2020

### 3.4.3 Pruebas de conectividad entre los módulo emisores y receptores

Realizadas las conexiones y alimentando el conjunto de dispositivos electrónicos se realizó las pruebas de conectividad de entre el emisor y receptor. Se codificó para la utilización de un solo botón del control de RF (emisor) por lo tanto se usa también una sola salida en los receptores.

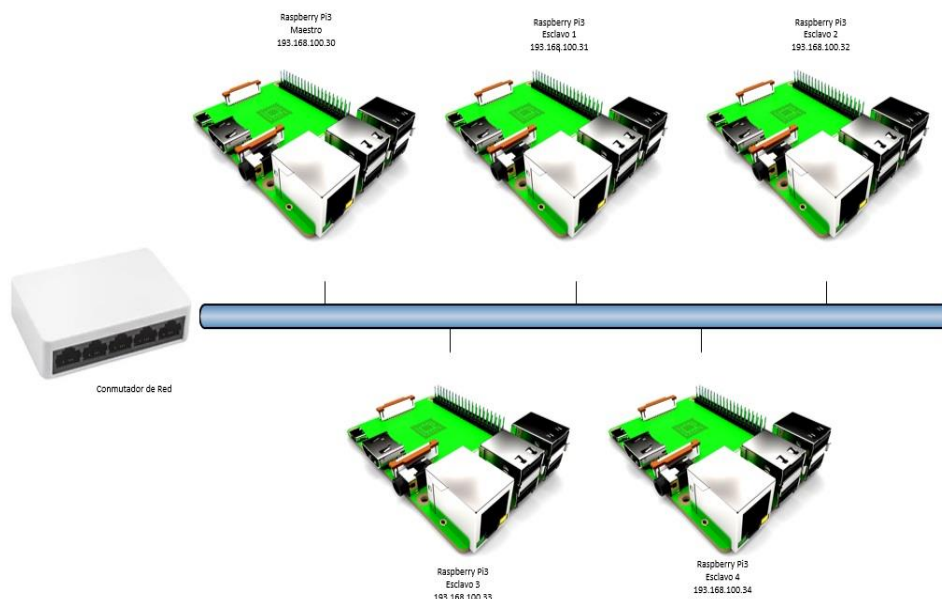
Para valoración de la funcionalidad de esta etapa se efectuó la conmutación del botón del control y se verificó la llegada de la señal al receptor. La forma de verificación de la recepción de la señal fue la conmutación sonora del relé en cada prueba.

Realizando la prueba descrita en un total de 50 repeticiones para que represente una muestra significativa se obtuvo el total de aciertos denotando una eficiencia de la etapa del 100%. ANEXO I

### 3.4.4 Implementación de la red de Raspberry

Bajo el concepto de una red maestro – esclavos, se realizó las conexiones de todos los puertos de la interfaz de Ethernet de las Raspberry hacia un switch empleando cable de red cruzado de conectores RJ-45.

La Figura 76-4 muestra la representación de las conexiones físicas de la Figura 77-4, se establece una conexión de topología estrella.

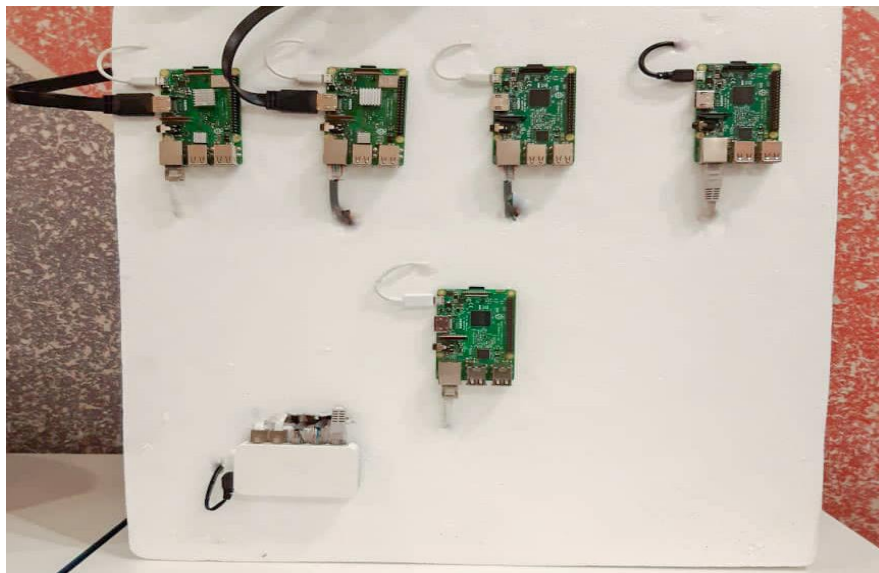


**Figura 77-4.** Conexión en red del conjunto de Raspberry

**Realizado por:** Ichina, Alex, 2020

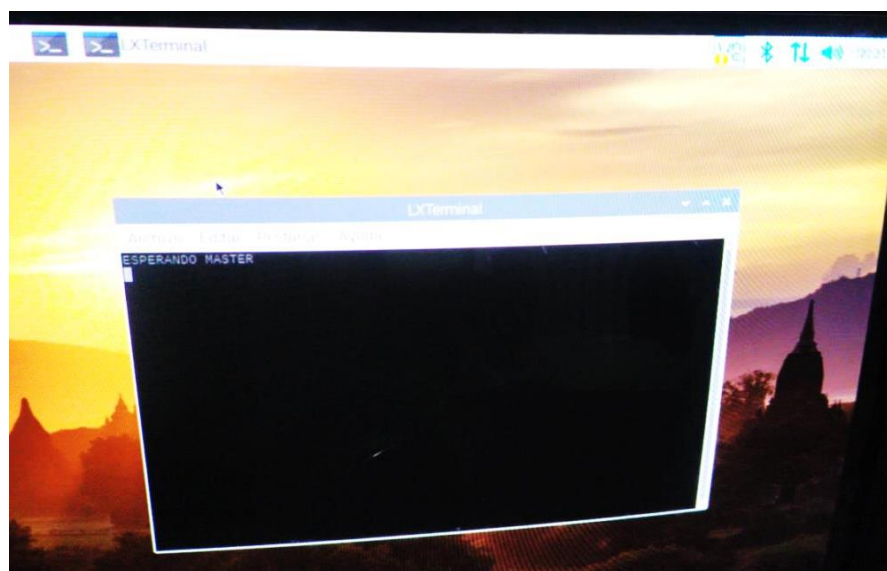
Para la verificación de la conectividad de los nodos esclavos hacia el maestro se realizó una prueba individual, en la que para iniciar se debía tener encendida la Master y luego arrancar el terminal de cada esclavo.

La figura 78-4 muestra la ejecución del terminal en un esclavo donde aparece el mensaje “ESPERANDO MASTER”, esto garantiza la conexión de ese nodo en la red. Para evaluar esta etapa se consideró cronometrar el tiempo de respuesta del esclavo a la red al iniciar el terminal obteniendo un tiempo promedio de 1 segundos, lo que permite considerar que el sistema tiene una buena respuesta, una eficiencia del 100%. ANEXO I



**Figura 78-4.** Implementación de la Red de Raspberry

Realizado por: Ichina, Alex, 2020

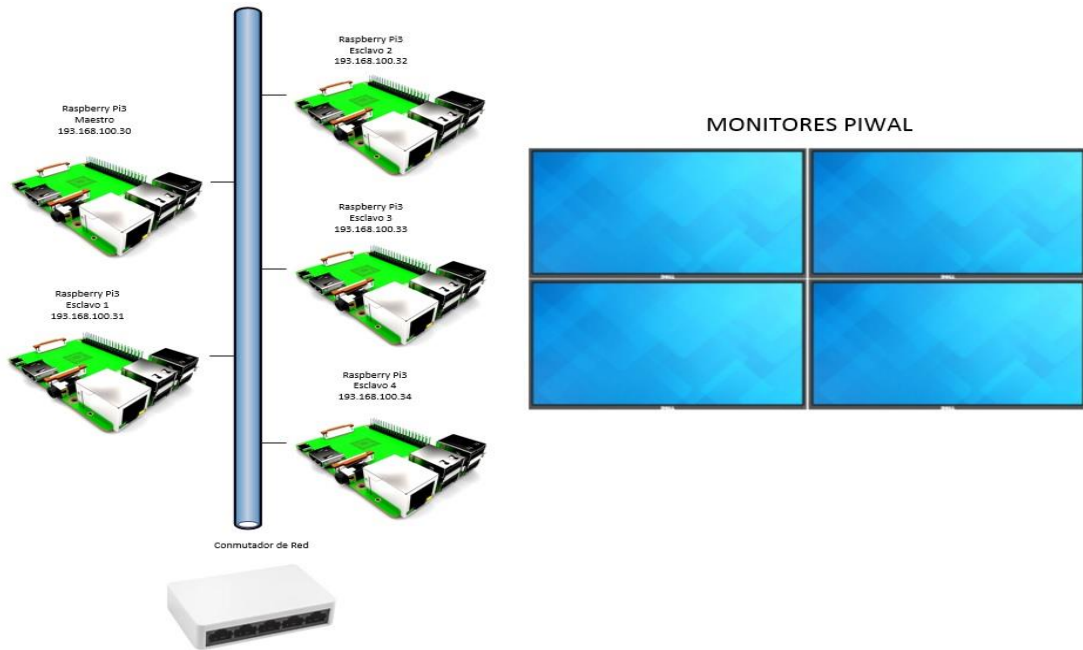


**Figura 79-4.** Implementación de la Red de Raspberry

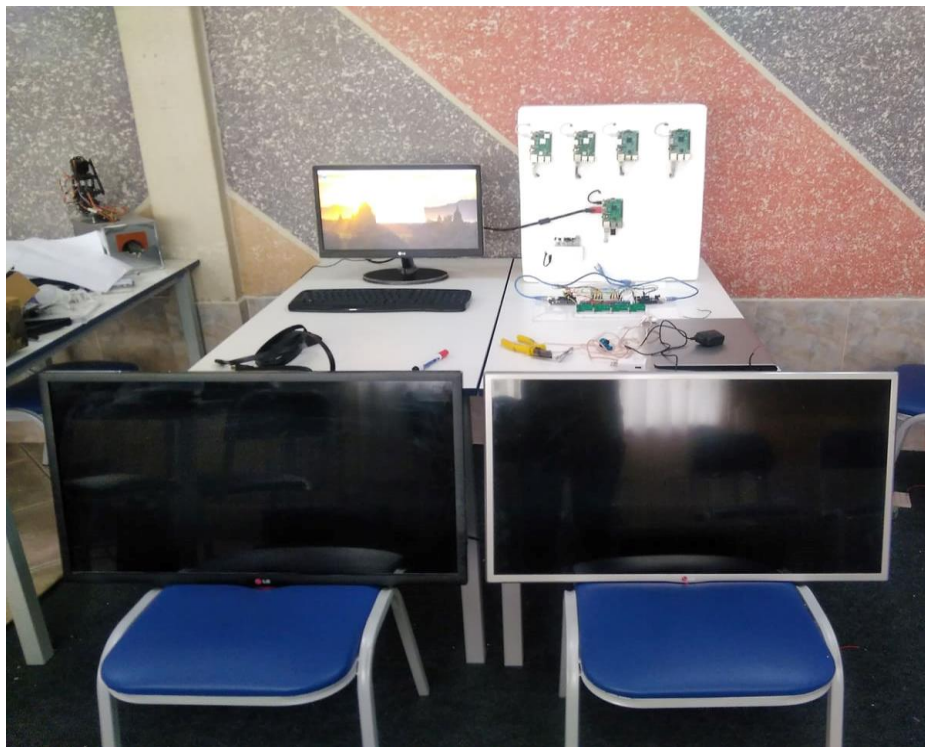
Realizado por: Ichina, Alex, 2020

### 3.4.5 Implementación del Piwall

Montada la red y configurados tanto el Master como los esclavos se procede a la integración de los monitores a cada Raspberry anclándolos en los puertos HDMI.



**Figura 80-4.** Diagrama de integración de monitores a la Red de Raspberry  
**Realizado por:** Ichina, Alex, 2020



**Figura 81-4.** Adición de monitores al sistema  
**Realizado por:** Ichina, Alex, 2020

### ***3.4.6 Pruebas para verificación del funcionamiento del Piwall***

Para verificación de la funcionalidad del Piwall se ejecutó el programa elaborado en Thonny Python desde el Master, la lógica de funcionamiento radica en que si no existe señal de los ganchos (emisor), el Piwall reproducirá una y otra vez el video promocional del local comercial.



**Figura 82-4.** Imagen dividida en el Piwall

**Realizado por:** Ichina, Alex, 2020

En este caso, las pruebas de conmutación se realizaron en un número de 50 veces, culminado todos los intentos en aciertos considerando en esta etapa también una funcionalidad del 100%. Sin embargo, se cronometró la respuesta del sistema, es decir se midió el tiempo en que se demora la conmutación del video a partir del pulso generado en el emisor instalado en el gancho, se obtuvo q existe un retardo de 4 segundos en la respuesta. ANEXO I.

### ***3.4.7 Instalación de la aplicación para el monitoreo de la percha inteligente***

Desarrollados los algoritmos de la aplicación informática desarrollada para la prestación del sistema en el monitoreo local de la percha inteligente se realizó el diseño de una interfaz gráfica amigable a la vista del usuario. El entorno con en LabVIEW es editable con el fin de conjugar

fondos, imágenes, gifs, que pueden ser utilizados entre otros locales comerciales similares, acorde al tipo de ambiente en el que se vaya a implementar el sistema.



**Figura 83-4.** Interfaz gráfica para el monitoreo de la percha inteligente

Realizado por: Ichina, Alex, 2020

### 3.4.8 Prueba de verificación del monitoreo local

Se validó la funcionalidad verificando la conmutación de los indicadores montados como recursos visuales en la interfaz, donde al emitir una señal desde los ganchos se pudo verificar el cambio de estado de los indicadores, se realizó 50 pruebas verificando una funcionalidad del 100%, pues en todas las pruebas realizadas se obtuvo el resultado esperado. También se evaluó el tiempo de respuesta de interacción verificando una réplica en tiempo real. ANEXO I.

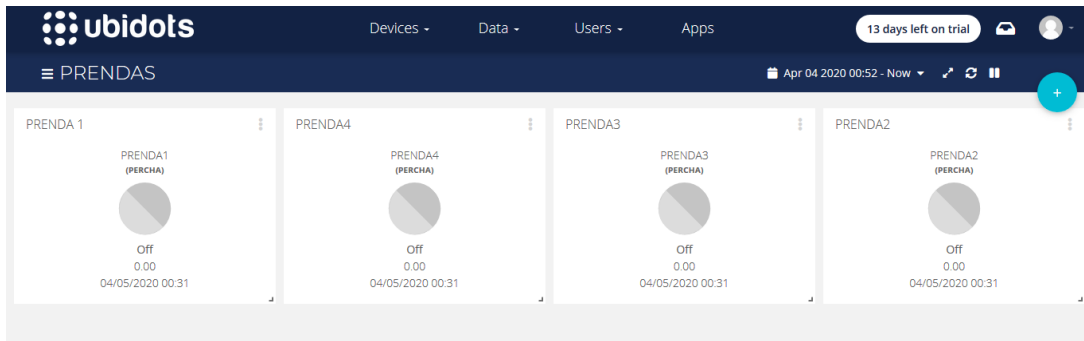


**Figura 84-4.** Interfaz gráfica para el monitoreo de la percha inteligente

Realizado por: Ichina, Alex, 2020

### 3.4.9 Montaje del monitoreo de la percha inteligente en Ubidots

Para tener cobertura total del estado de los ganchos en la percha inteligente se implementó la Dashboard con las variables asignadas a cada uno de ellos como se aprecia en la Figura 84-4, garantizando el monitoreo remoto de todos sus elementos.



**Figura 85-4.** Interfaz gráfica para el monitoreo remoto desde Ubidots

**Realizado por:** Ichina, Alex, 2020

### 3.4.10 Pruebas de verificación del monitoreo remoto

Paralelamente a las pruebas del monitoreo local se evaluó el monitoreo remoto que medido en eficiencia igualmente otorgó un 100% al cumplirse la conmutación del estado de los ganchos en la Dashboard acorde a la generación de las señales en los mismos.

En esta etapa se detectó un retraso de 1 a 2 segundos al cambio de estado respecto a al monitoreo local, debido al ancho de banda del servicio de internet, lo que se podría mejorar con un ancho de banda más amplio. ANEXO I.



**Figura 86-4.** Interfaz gráfica para el monitoreo remoto desde Ubidots

**Realizado por:** Ichina, Alex, 2020

### 3.4.11 Aplicación informática Bella Boutique

Analizadas las etapas del sistema, se verifica la gestión sobre la plataforma para el manejo de existencias de productos por medio de inventarios y ventas.

La figura 86-4 muestra la pantalla con la que inicia la aplicación informática que contiene recursos visuales en movimiento. Un menú representado por botones, donde uno de ellos invita a ingresar al sistema de la percha inteligente o al sistema para gestión de los productos o la opción salir de la aplicación.



**Figura 87-4.** Aplicación informática pantalla de inicio

**Realizado por:** Ichina, Alex, 2020

El ingreso al botón de percha inteligente direcciona a la pantalla previamente analizada en la funcionalidad de la misma.

El botón gestión de productos direcciona a las pantallas creadas para el ingreso de los parámetros descriptivos de las prendas para el ingreso de productos. En la Figura 87-4 y 88-4 se muestran las pantallas resultantes para el ingreso de los parámetros descriptivos de los productos y el ingreso de un producto completo citadas en la parte de diseño.





**Figura 88-4.** Aplicación informática pantalla de ingreso parámetros de productos

Realizado por: Ichina, Alex, 2020



**Figura 89-4.** Aplicación informática producto implementado

Realizado por: Ichina, Alex, 2020

Verificada la funcionalidad del sistema en cada una de sus etapas se presenta una tabla resumen de los resultados obtenidos en cada una de ellas.

**Tabla 8-3:** Resumen Pruebas de funcionalidad y respuestas del sistema

PROCEDIMIENTO	ACIERTOS	FUNCIONALIDAD	RESPUESTA
Comunicación emisor-receptor RF	50 de 50	100%	Tiempo Real
Respuesta a la conexión Maestro - Esclavo	50 de 50	100%	Retraso 1segundo
Conmutación de videos	50 de 50	100%	Retraso 1 segundo
Monitoreo Local	50 de 50	100%	Tiempo Real
Monitoreo Remoto	47 de 50	97%	1 a 2 segundos (depende de la calidad de conexión)

Realizado por: Ichina, Alex, 2020

### 3.4.12 Medición de corriente para determinación del consumo del sistema

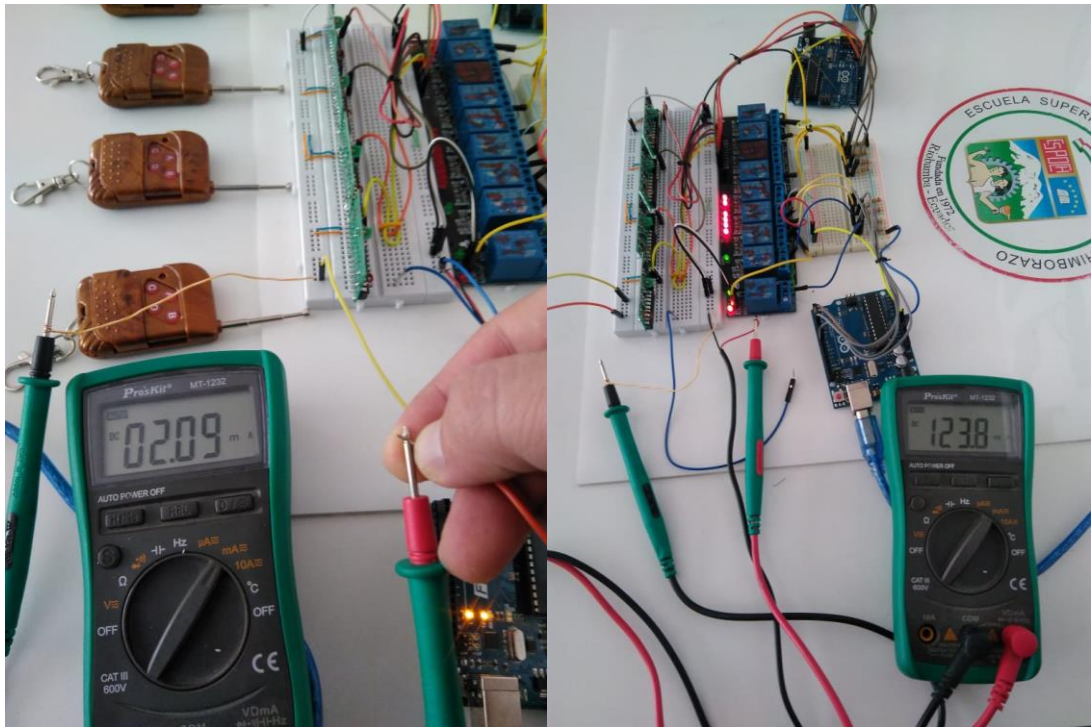
Se consideró importante conocer el consumo de energía de todo el sistema, por lo que a continuación se describen en la Tabla 9-3 todas las cargas del sistema con su consumo individual para determinar el total.

**Tabla 9-3:** Costos directos e indirectos del proyecto.

# DE CARGAS	CARGA	CONSUMO INDIVIDUAL POR HORA (A-Wh)	CONSUMO CONJUNTO POR HORA (A-Wh)	CONSUMO TOTAL (Wh)
5	Raspberry Pi3	350 mA - 1,8 Wh	1,75 A - 9 Wh	9
2	Arduino UNO R3	46 mA - 0,23 Wh	92 mA - 0,46 Wh	0,46
4	Receptores RF	2.09 mA - 10 mWh	8,36mA - 40 mWh	0,040
1	Módulo de Relés	123 mA - 0,615 Wh	123 mA - 0,615 Wh	0,615
1	Ordenador portátil	200Wh	200Wh	200
1	Switch	50mA - 0,6 Wh	50mA - 0,6 Wh	0,6
4	Monitores	50 Wh	200 Wh	200
				Total 410,715 Wh 5,5A

Realizado por: Ichina, Alex, 2020.

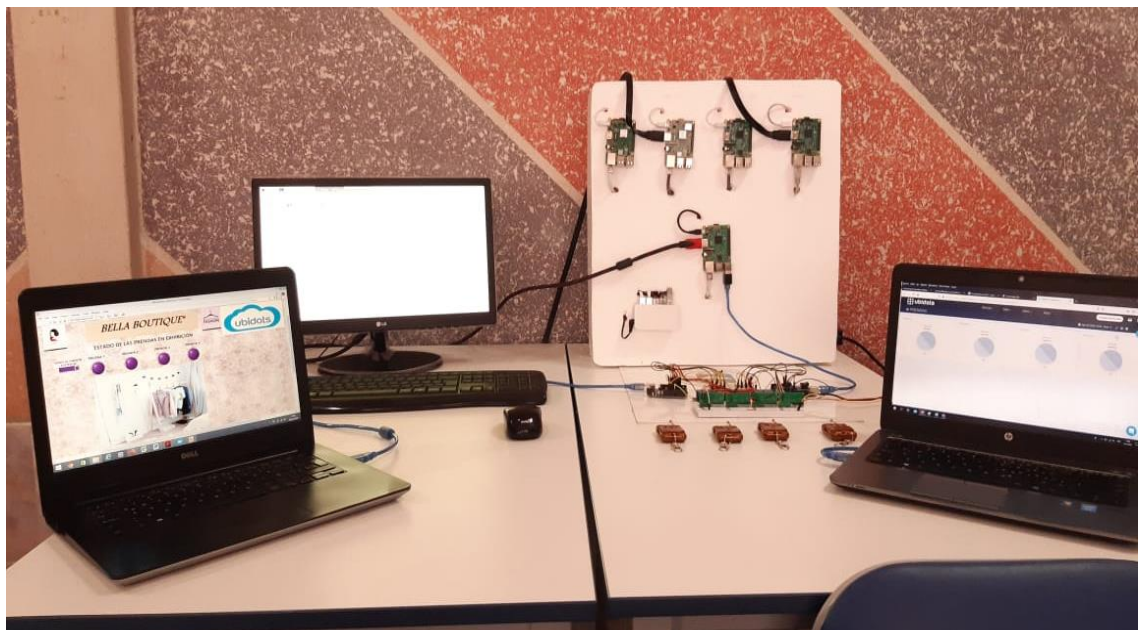
Los valores expuestos se los ha realizado a plena carga del equipo. En el sistema todos los elementos están activos desde el momento de encender, la Figura 89-4 representan el proceso de medición de los consumos de corriente de algunos elementos que no se halló su consumo en el datasheet.



**Figura 90-4.** Medición de corrientes de consumo Módulo de Relé y Receptores RF

**Realizado por:** Ichina, Alex, 2020

La Figura 89-4 y 90-4 muestran todo el sistema implementado y funcional en conjunto de la percha inteligente.



**Figura 91-4.** Sistema electrónico e interfaces de monitoreo local y remoto

**Realizado por:** Ichina, Alex, 2020



**Figura 92-4.** Sistema Video Wall

**Realizado por:** Ichina, Alex, 2020

## CAPITULO IV

### 4. GESTIÓN DEL PROYECTO

#### 4.1 Cronograma

**Tabla 10-4:** Cronograma.

Actividad	Mes 1		Mes 2		Mes 3		Mes 4		Mes 5		Mes 6	
	Semanas		Semanas		Semanas		Semanas		Semanas		Semanas	
	1-2	3-4	1-2	3-4	1-2	3-4	1-2	3-4	1-2	3-4	1-2	3-4
Etapa de Análisis y Recolección de la Información Necesaria												
Etapa de Selección de los Elementos												
Etapa de diseño del prototipo												
Etapa de implementación del prototipo												
Etapa de Análisis de resultados y pruebas												
Etapa de redacción del informe final												
Etapa de Articulación Científica												

Realizado por: Ichina, Alex, 2020.

#### 4.2 Recursos y materiales: humanos, equipos, financiamiento

##### 4.2.1 Costos

La siguiente tabla muestra el costo total para el sistema de control.

**Tabla 11-3:** Costos directos e indirectos del proyecto.

N°	Detalle	Cantidad	Valor Unitario (\$)	Total (\$)
1	Raspberry Pi3	5	80,00	400,00
2	Micro SD	5	12,00	60,00
3	Fuente de poder 5VCC- 2,5A	7	9,00	63,00

<b>4</b>	Switch de red 5 puertos	1	15,00	15,00
<b>5</b>	Cable HDMI 3m	5	7,00	35,00
<b>6</b>	Cable de red 1m	5	5,00	25,00
<b>7</b>	Transmisor – Receptor RF	4	15,00	60,00
<b>8</b>	Arduino UNO R3	2	15,00	30,00
<b>9</b>	Kit de cables H-M, M-H	1	5,00	5,00
<b>10</b>	Placas para montaje de elementos	2	10,00	20,00
			Total	713,00

**Realizado por:** Ichina, Alex, 2020.

#### ***4.2.2 Talento Humano***

Alex Javier Ichina López, estudiante que realizó el trabajo. Para la realización del proyecto se combinaron habilidades para la investigación e implementación del sistema culminándolo con éxito con los resultados esperados.

Ing. Edwin Altamirano Santillán, Ing. Jorge Luis Paucar, director y miembro del trabajo de titulación quienes, con su conocimiento y experiencia colaboraron en el desarrollo del proyecto.

#### ***4.2.3 Recursos Materiales***

Para conseguir los resultados del proyecto se empleó tanto software y hardware detallados en el proceso de metodología.

El financiamiento fue cubierto por el estudiante promotor de la idea, pues el trabajo se lo considera como una idea para seguirla desarrollando y posiblemente generar un emprendimiento que permita replicar el sistema en diferentes entornos.

## **5. CONCLUSIONES**

Para la implementación de una percha inteligente y tenga autonomía en su funcionamiento, esta debe interactuar con los clientes de una forma amigable. La inclusión con el internet de las cosas (IoT) se incorpora al desarrollo tecnológico en el campo de la industrial 4.0.

Los requerimientos software y hardware utilizados en la implementación de la percha inteligente, proporcionan la información necesaria para ser procesada y cumple con las especificaciones de autonomía de gestión y monitoreo local y remoto.

La herramienta LabVIEW por su flexibilidad para vincularse con otras aplicaciones se logró obtener una aplicación informática con una interfaz gráfica que permite la gestión de existencias por medio del registro y control de inventarios y ventas de los productos.

Se vinculó el internet de las cosas al sistema mediante una aplicación para el monitoreo remoto de la percha inteligente; permitiendo el enlace con la plataforma de soporte de Ubidots en tiempo real.

De las pruebas realizadas en cada una de las etapas de la implementación del sistema generaron resultados satisfactorios en tiempo real de un 98% y un segundo en la conmutación de videos, tanto en monitoreo local como remoto, retardo que varía según el ancho de banda de la conexión de internet con la que se trabaje.

## **6. RECOMENDACIONES**

Para personalizar o modificar las pantallas en la percha inteligente se requiere de una persona con conocimiento en el área y el código fuente de la aplicación utilizada en este sistema.

Se recomienda colocar una clave a la base de datos para administrar la gestión de tienda y tener una integridad de los datos.

Para evitar el retraso en la respuesta del sistema en la interacción de la interfaz del Dashboard de Ubidots se recomienda tener a disposición un buen ancho de banda.

Las tarjetas Raspberry que conforman el sistema requieren de un sistema de ventilación para evitar su calentamiento que puede provocar la distorsión de los videos en el Piwall.



## GLOSARIO

**Internet de las Cosas (IoT):** Una infraestructura de red global dinámica con capacidades de autoconfiguración basadas en protocolos de comunicación estándar e interoperables donde las cosas físicas y virtuales tienen identidades, atributos físicos y personalidades virtuales, y usan interfaces inteligentes de manera integrada en el red de información referimos a la biomecánica del pie en el cual está involucrada la anatomía de las extremidades inferiores del mismo, por ende, es necesario tener un conocimiento previo desde el punto de vista anatómico.(7) (Llaneza González, 2018, p. 20)

**LabVIEW:** Es una herramienta de programación gráfica. Originalmente este programa estaba orientado para aplicaciones de control de equipos electrónicos usados en el desarrollo de sistemas de instrumentación, lo que se conoce como instrumentación virtual. (13) (Vizcaíno y Sebastián, 2011, p. 22)

**Red Inalámbrica de Sensores:** Se define como WSN (Wireless Sensor Network) a una red inalámbrica de dispositivos de bajo consumo para la adquisición de información o senso de variables. A los dispositivos que forman parte de dichas redes se los conoce como nodos sensores o motas (motes) y están limitados en su capacidad computacional y de comunicación. Mediante una interacción en conjunto estos dispositivos transportan la información de un punto a otro de la red transmitiendo pequeños mensajes de un nodo a otro. (4) (Fernández Martínez et al., 2009, p. 26)

**Radiofrecuencia (RF):** La radiofrecuencia se la define como una parte del espectro electromagnético, específicamente hace referencia a un segmento de baja energía. El enlace de comunicación se genera en el ciclo de transmisión de una señal al hacer circular corriente a través de un conductor y la recepción por medio de una antena. (Universidad de Valencia, 2018)

**Arduino:** es una plataforma embebida que contiene como elemento principal un microcontrolador reprogramable, el mismo que permite la adquisición de señales, procesamiento mediante sentencias programadas y la emisión de señales de control. Este sistema embebido además se acopla a una amplia gama de sensores de la misma marca u otra que proporcionen señales del tipo digital y/o analógico para ser procesadas y según su evaluación por medio de interfaces de potencia poder realizar el control de actuadores sin importar la carga que manejen. (Arduino, 2020)

**Microcontrolador:** Se define como un dispositivo electrónico embebido o un circuito integrado que encapsula varios recursos, la ventaja de este elemento que se puede programar de acuerdo a la necesidad que suscite el entorno en el que se lo va a emplear, es capaz de ejecutar con total autonomía previo a la carga de instrucciones predefinidas cualquier tipo de trabajo que se lo encomiende. (Artero, 2013, p. 62)

**Raspberry Pi:** Es una minicomputadora de la gama de hardware libre que constituida de una placa simple empleada en el ámbito educativo para estudiantes del área de informática y electrónica principalmente, el soporte de software para este dispositivo es de código abierto. Está constituida esencialmente por un procesador, memoria RAM, puertos USB, GPU, HDMI, ETHERNET, requiere de una micro SD pues no incluye memoria interna. La potencialidad de la Raspberry Pi la hace suficientemente potente como para poder tomar el lugar de una computadora de escritorio. (A, B rev 1, B rev 2, B+). (Salcedo & Cendrós, 2006, pp. 63-64)

**Video Wall:** Configuración especial de pantallas o monitores profesionales sincronizados para mostrar contenidos y simular una pantalla de gran tamaño. Se lo puede crear utilizando tantos monitores o pantallas como sean necesarios para cubrir un área deseada. Se los puede diseñar en forma simétrica o no y su tamaño. (Video Wall, 2020)

## BIBLIOGRAFÍA

**ARDUINO.** *Software Arduino*. [en línea]. Disponible en: <https://www.arduino.cc>.

**ARDUINO.** *¿Qué es Arduino?* [en línea]. Disponible en: <https://arduino.cl/que-es-arduino/>.

**ARTERO, T.Ó.** *ARDUINO Curso básico de formación*. Alfaomega, 2013. DOI 9788494072505. p. 62.

**BONILLA FABELA I,** Tavizon-Salazar A, Morales-Escobar M, GuajardoMuñoz L, Laines-Alamina C. *IoT, el internet de las cosas y la innovación de sus aplicaciones*. 2015.

**BUYYA R,** Dastjerdi AV. *Internet of Things: Principles and Paradigms*. Elsevier; 2016. 380 p.

**CÓRDOBA A,** Santamaría Buitrago FA. *Vista de Las redes de sensores inalámbricos y el internet de las cosas*. [en línea]. 2013 [citado 30 de enero de 2020]. Disponible en: <https://revistascientificas.cuc.edu.co/ingecuc/article/view/253/232>

**CAMA A,** De la Hoz E, Cama D. *Las redes de sensores inalámbricos y el Internet de las cosas*. [en línea]. 2012. Disponible en: <https://revistascientificas.cuc.edu.co/ingecuc/article/view/253/232>

**EVANS DAVE.** *Internet de las cosas, como la próxima evolución de internet lo cambia todo*. 2011. Disponible en: <http://jeuazarru.com/wp-content/uploads/2014/10/Internet-of-Things.pdf>

**FERNÁNDEZ MARTÍNEZ R,** Ordieres Meré J, Martínez de Pisón F, Gonzáles Marcos A, Alba Elías F, Lostado Lorza R. *Redes inalámbricas de sensores: teoría y aplicación práctica*. 2009.

**GARCÍA DE PASO VJ.** *Vestidor Inteligente* [en línea]. [ESPAÑA]: Universidad Politécnica de Cataluña; 2011 [citado 29 de diciembre de 2019]. Disponible en: <https://upcommons.upc.edu/handle/2099.1/13221>

**GAWER A,** Cusumano M, Yoffie D. *El negocio de las plataformas* | Harvard Deusto las revistas. 2019 [citado 29 de diciembre de 2019]; (295). Disponible en: <https://www.harvard-deusto.com/el-negocio-de-las-plataformas>

**GUERRERO-Martínez DG.** *Factores clave de éxito en el negocio del retail*. Ing Ind. 11 de marzo de 2012, 0(030):189-205-205.

**LAJARA JR,** Pelegri J. *LabVIEW: Entorno gráfico de programación* - José Rafael Lajara Vizcaíno, José Pelegrí Sebastía - Google Libros [en línea]. Segunda. Ediciones Técnicas; 2011 [citado 29 de diciembre de 2019]. Disponible en: [https://books.google.com.ec/books?hl=es&lr=&id=ZFQua3-eeQEC&oi=fnd&pg=PA21&dq=COMUNICACION+SERIAL+EN+LABVIEW&ots=qI1xU8SFhu&sig=gfJ14V93tVGnHRzA5eglxa5zsZI&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.ec/books?hl=es&lr=&id=ZFQua3-eeQEC&oi=fnd&pg=PA21&dq=COMUNICACION+SERIAL+EN+LABVIEW&ots=qI1xU8SFhu&sig=gfJ14V93tVGnHRzA5eglxa5zsZI&redir_esc=y#v=onepage&q&f=false).

**LÓPEZ GARZÓN WF,** Cárdenas López JC. *Vista de Tecnología internet of things (IoT) y el big data*. Mare Ingenii Ing [en línea]. 2019 [citado 30 de diciembre de 2019]; Disponible en: <http://cipres.sanmateo.edu.co/index.php/mi/article/view/213/173>

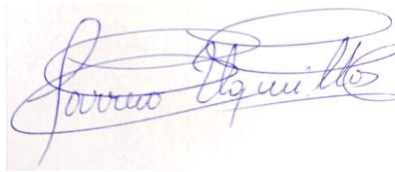
**LORENZANA, V.R.** *Redes Inalámbricas*. Tecnologías Inalámbricas, 2007, pp. 28.

**LLANEZA GONZÁLEZ P.** *Seguridad y responsabilidad en la Internet de las cosas (IoT)*. Primera edición. ESPAÑA: Wolters Kluwer España, S.A; 2018. 360 p.

**MECA LÓPEZ C.** *Probador Virtual: Realidad aumentada sin marcadores* [Trabajo fin de Grado]. Universidad de Alicante; 2014.

**ROSE Karen SE, Lyman Chapin.** *LA INTERNET DE LAS COSAS— UNA BREVE RESEÑA*. Internet Society (ISOC). 2015.

**VIZCAÍNO JRL, Sebastián JP.** *LabVIEW: Entorno gráfico de programación*. Marcombo; 2011. 471 p.



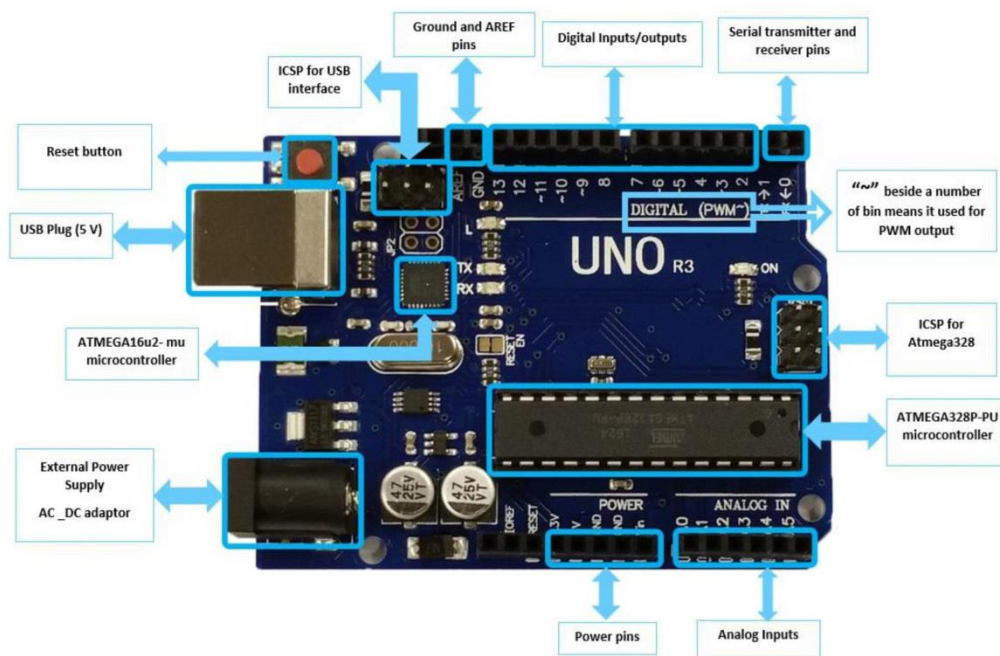
04-05-2020

# ANEXOS

## ANEXO A. DATASHETT ARDUINO UNO



### Arduino Uno R3



### INTRODUCTION

Arduino is used for building different types of electronic circuits easily using of both a physical programmable circuit board usually microcontroller and piece of code running on computer with USB connection between the computer and Arduino.

Programming language used in Arduino is just a simplified version of C++ that can easily replace thousands of wires with words.

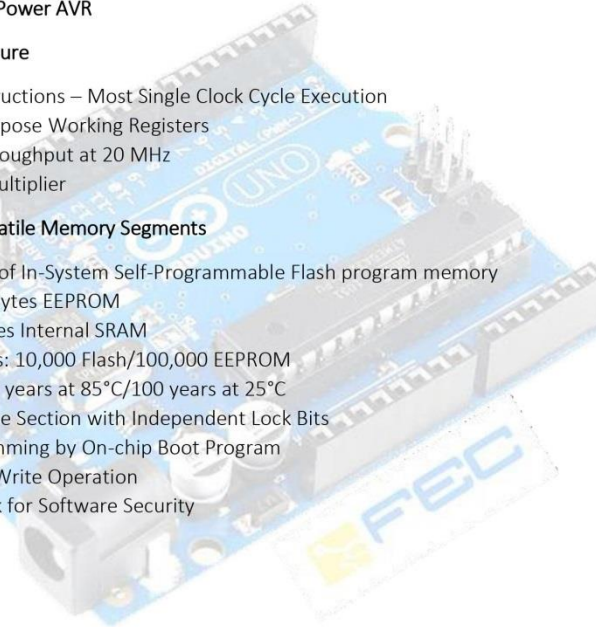


## ARDUINO UNO-R3 PHYSICAL COMPONENTS

### ATMEGA328P-PU microcontroller

The most important element in Arduino Uno R3 is ATMEGA328P-PU is an 8-bit Microcontroller with flash memory reach to 32k bytes. It's features as follow:

- High Performance, Low Power AVR
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1K Bytes EEPROM
  - 512/1K/1K/2K Bytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
  - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
  - Temperature Measurement
  - Programmable Serial USART





- Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface (Philips I2 C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

• **Special Microcontroller Features**

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

• **I/O and Packages**

- 23 Programmable I/O Lines
- 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

• **Operating Voltage:**

- 1.8 - 5.5V

• **Temperature Range:**

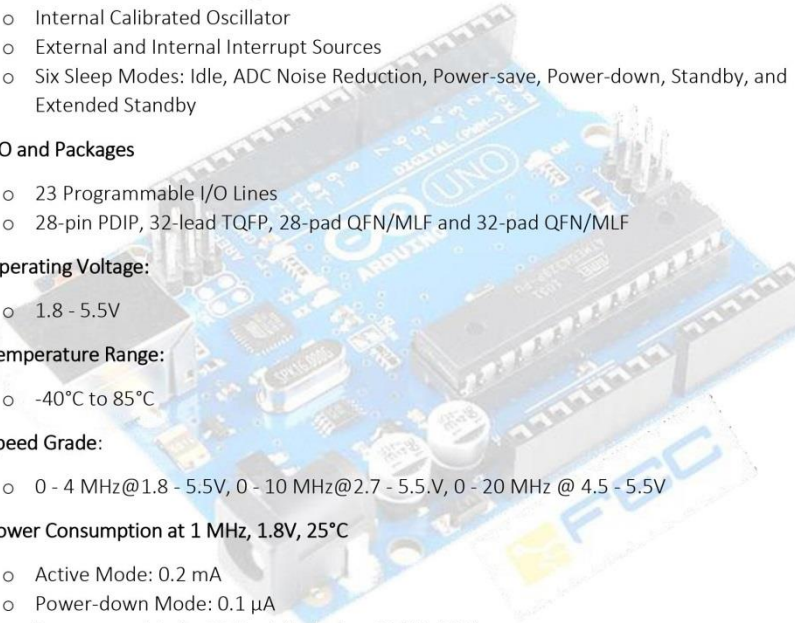
- -40°C to 85°C

• **Speed Grade:**

- 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V

• **Power Consumption at 1 MHz, 1.8V, 25°C**

- Active Mode: 0.2 mA
- Power-down Mode: 0.1  $\mu$ A
- Power-save Mode: 0.75  $\mu$ A (Including 32 kHz RTC)





- Pin configuration

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

#### ATMEGA16u2- mu microcontroller

Is a 8-bit microcontroller used as USB driver in Arduino uno R3 it's features as follow:

- High Performance, Low Power AVR
- Advanced RISC Architecture
  - 125 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
- Non-volatile Program and Data Memories
  - 8K/16K/32K Bytes of In-System Self-Programmable Flash
  - 512/512/1024 EEPROM
  - 512/512/1024 Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
  - Data retention: 20 years at 85°C/ 100 years at 25°C

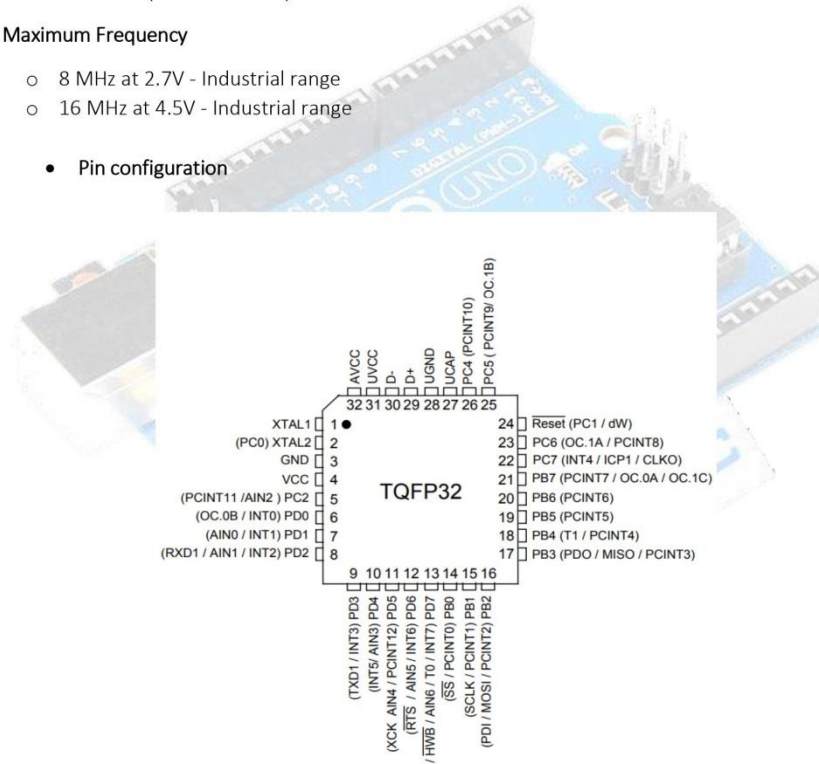




- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by on-chip Boot Program hardware-activated after reset
- Programming Lock for Software Security
- **USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion**
  - Complies fully with Universal Serial Bus Specification REV 2.0
  - 48 MHz PLL for Full-speed Bus Operation: data transfer rates at 12 Mbit/s
  - Fully independent 176 bytes USB DPRAM for endpoint memory allocation
  - Endpoint 0 for Control Transfers: from 8 up to 64-bytes
  - 4 Programmable Endpoints:
    - IN or Out Directions
    - Bulk, Interrupt and Isochronous Transfers
    - Programmable maximum packet size from 8 to 64 bytes
    - Programmable single or double buffer
  - Suspend/Resume Interrupts
  - Microcontroller reset on USB Bus Reset without detach
  - USB Bus Disconnection on Microcontroller Request
- **Peripheral Features**
  - One 8-bit Timer/Counters with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
  - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Mode(three 8-bit PWM channels)
  - USART with SPI master only mode and hardware flow control (RTS/CTS)
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- **On Chip Debug Interface (debug WIRE)**
- **Special Microcontroller Features**
  - Power-On Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby
- **I/O and Packages**
  - 22 Programmable I/O Lines
  - QFN32 (5x5mm) / TQFP32 packages



- Operating Voltages
  - 2.7 - 5.5V
- Operating temperature
  - Industrial (-40°C to +85°C)
- Maximum Frequency
  - 8 MHz at 2.7V - Industrial range
  - 16 MHz at 4.5V - Industrial range
- Pin configuration





## OTHER ARDUINO UNO R3 PARTS

### Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

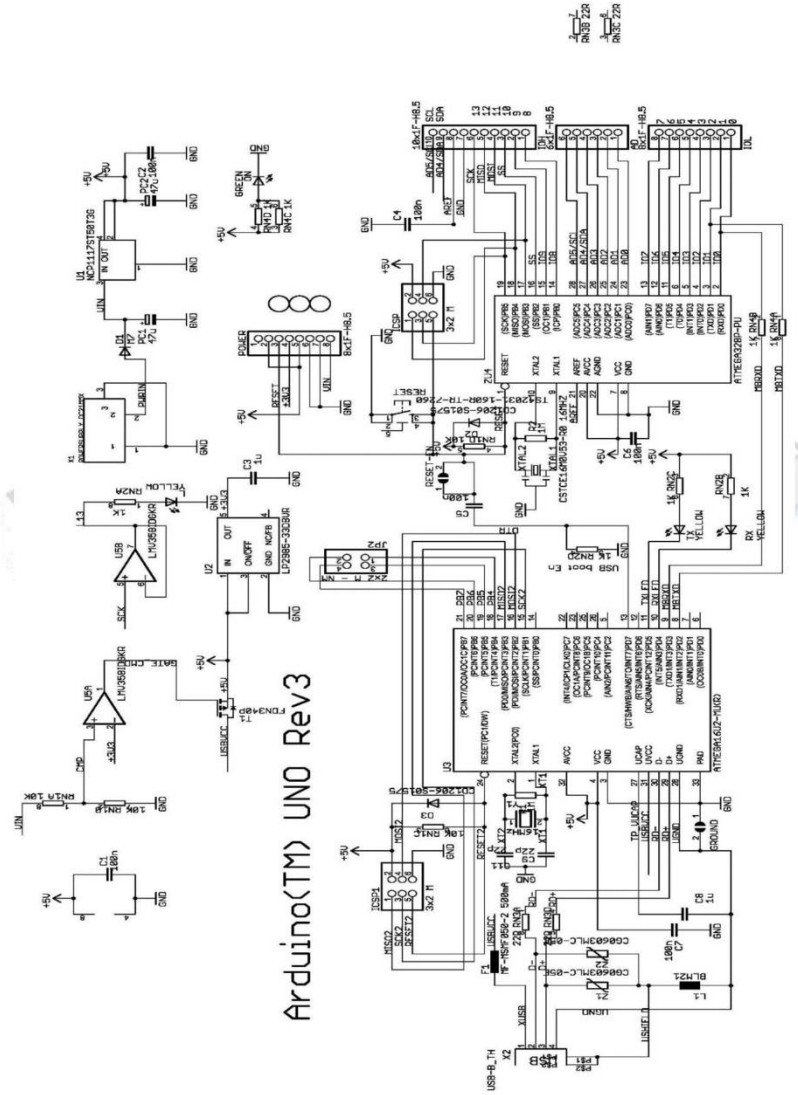
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- AREF: Reference voltage for the analog inputs. Used with `analogReference()`.
- Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



# ARDUINO UNO R3 SCHEMATIC DIAGRAM





Raspberry Pi

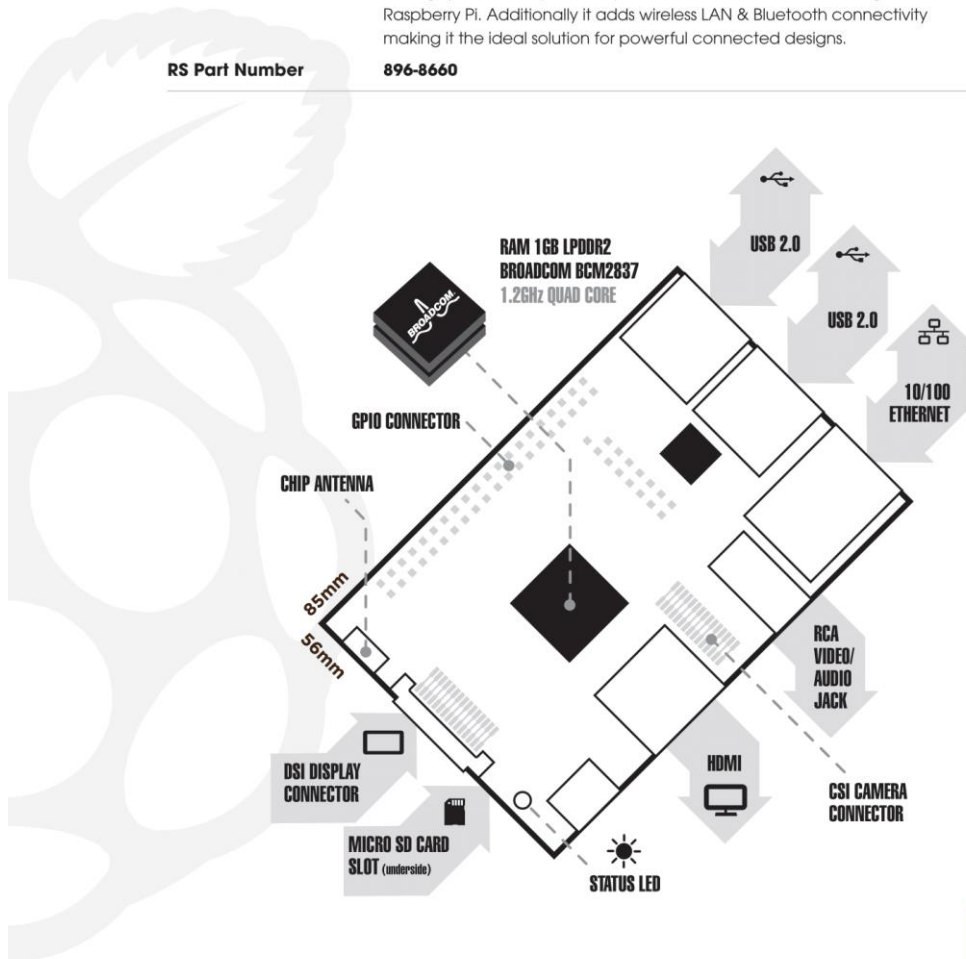


## Raspberry Pi 3 Model B

**Product Name** Raspberry Pi 3

**Product Description** The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processor, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.

**RS Part Number** 896-8660





# Raspberry Pi

## Raspberry Pi 3 Model B

### Specifications

<b>Processor</b>	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
<b>GPU</b>	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode.  Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
<b>Memory</b>	1GB LPDDR2
<b>Operating System</b>	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
<b>Dimensions</b>	85 x 56 x 17mm
<b>Power</b>	Micro USB socket 5V1, 2.5A

### Connectors:

<b>Ethernet</b>	10/100 BaseT Ethernet socket
<b>Video Output</b>	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
<b>Audio Output</b>	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
<b>GPIO Connector</b>	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
<b>Camera Connector</b>	15-pin MIPI Camera Serial Interface (CSI-2)
<b>Display Connector</b>	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
<b>Memory Card Slot</b>	Push/pull Micro SDIO

### Key Benefits

- Low cost
- 10x faster processing
- Consistent board format
- Added connectivity

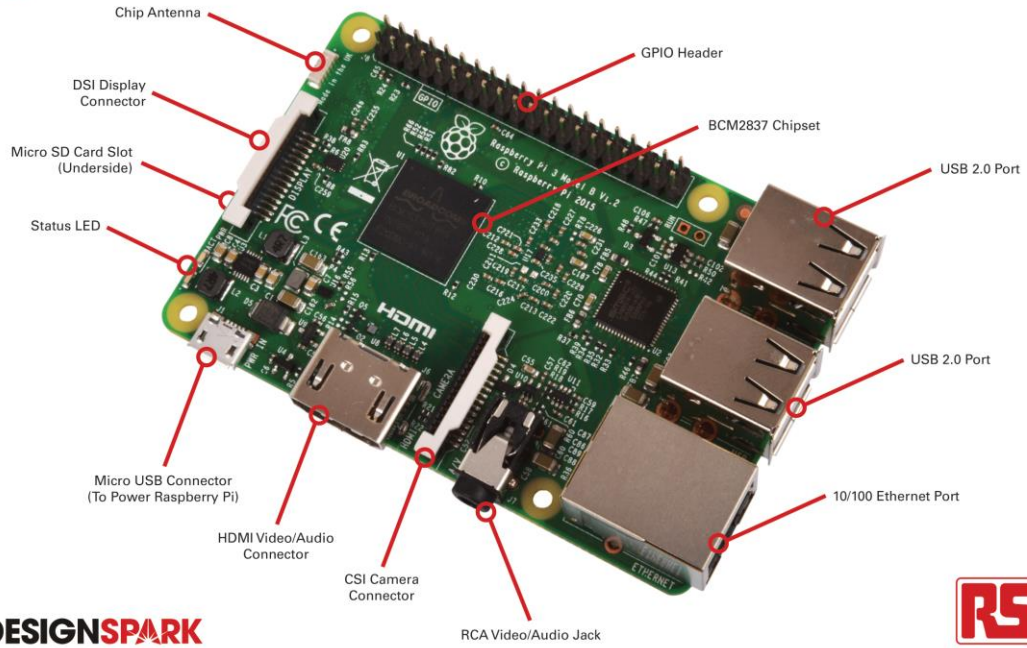
### Key Applications

- Low cost PC/tablet/laptop
- Media centre
- Industrial/Home automation
- Print server
- Web camera
- Wireless access point
- Environmental sensing/monitoring (e.g. weather station)
- IoT applications
- Robotics
- Server/cloud server
- Security monitoring
- Gaming





# Raspberry Pi 3 Model B



DESIGNSPARK





# Raspberry Pi Frequently Asked Questions

## What is a Raspberry Pi?

Created by the Raspberry Pi Foundation, the Raspberry Pi is an open-source, Linux based, credit card sized computer board. The Pi is an exciting and accessible means of improving computing and programming skills for people of all ages. By connecting to your TV or monitor and a keyboard, and with the right programming, the Pi can do many things that a desktop computer can do such as surf the internet and play video. The Pi is also great for those innovative projects that you want to try out - newer models are ideal for Internet of Things projects due to their processing power. With Pi 3, Wireless LAN and Bluetooth Low Energy are on-board too.

## What are the differences between the models?

Current versions of the Raspberry Pi are the Pi A+, Pi B+, Pi 2 B, Pi 3 B and Compute Module.

	Pi A+	Pi B+	Pi 2 B	Pi 3 B	Compute Module
Dimensions	66 x 56 x 14mm	85 x 56 x 17mm	85 x 56 x 17mm	85 x 56 x 17mm	67.5 x 30mm
SoC	BCM2835	BCM2835	BCM2836	BCM2837	BCM2835
Processor Core	ARM11	ARM11	ARM Cortex-A7	ARM Cortex-A53	ARM11
Processing Power	700 MHz	700 MHz	900 MHz	1.2 GHz	700 MHz
Memory	256 MB	512 MB	1 GB	1GB LPDDR2	512 MB
Ports	1x USB 2.0	4x USB 2.0 1x 10/100 Ethernet	4x USB 2.0 1x 10/100 Ethernet	4x USB 2.0 1x 10/100 Ethernet	N/A
GPIO	40	40	40	40	N/A

## What do I get with my Raspberry Pi?

A Raspberry Pi board only.

Each Raspberry Pi customer is unique. You may already have cables, power supplies, keyboards, SD memory cards or monitors. However, if you do require additional products to start with your Pi or to really get creative, we can help.

Our expanding range of accessories includes:

Protective Cases	Power Supplies	NOOBS microSD Cards	Keyboards & Mice	Printers
Cables	Displays & Camera Boards	Wireless Connectivity	Add-on Boards	RS Pi Bundles







## How do I get connected?

To get started with your Pi you will need;

- A monitor or TV screen to set-up your Pi
- A keyboard to interact with your Pi
- A mouse to navigate your Pi
- A power supply
- An SD card with the latest version of New Out Of Box Software (NOOBS), to install the operating system that you would like to use.

To get **sound** and **video** you will need cables to suit what your screen or monitor accepts. For those with monitors that accept VGA, a HDMI to VGA adaptor is needed in addition to a HDMI cable, unless you use the composite video output from the Pi.

For an **internet connection**, the Pi B+ and Pi 2 B have an ethernet port. You also have the option of adding a WiFi Adapter/Dongle which may mean that you need a USB Hub if you have run out of USB ports. The Pi 3 already has 802.11 b/g/n wireless LAN and Bluetooth 4.1 (Bluetooth Classic and Low Energy).

---

## Powering my Pi

The Pi has a 5 V microUSB power socket, located on the bottom left hand corner of your Pi board.

Version	Recommended Power Supply Current Capacity
Pi B	1.2 A
Pi A+	700 mA
Pi B+	1.8 A
Pi 2 B	1.8 A
Pi 3 B	2.5 A

Generally, the more USB ports and interfaces you use on your Pi, the more power you are going to need - be careful.

We advise to look at buying a powered USB hub - this means less pressure on your Pi whilst still being able to incorporate all the features and functionality that you want to. When connecting any devices to your Pi, it is advisable to always check the power rating.

Batteries are not a recommended power supply for your Pi.

Note: The Official Raspberry Power Supply Unit for Pi 3 is not a general purpose power supply and must only be used for the Pi 3.





## What is the user name and password for the Raspberry Pi?

The user name for Raspbian is **pi**

The password for Raspbian is **raspberrypi**

---

## Operating Systems, Programming Languages & SD Cards

You will need an **operating system** to start using your Pi. An operating system is vital software that acts as a computer manager.

To download an operating system you will need an **SD card** between 8-32 GB. We have SD cards with New Out Of Box Software (NOOBS) pre-installed, so you don't have to do all of the work. NOOBS helps you to set up your Pi and has six operating systems that you can download;

Raspbian (recommended)	Pidora	OpenElec	Windows 10 IoT
RaspBMC	RISC OS	Arch Linux	

Of course, you don't have to use NOOBS. The Raspberry Pi Foundation regularly updates other available 'distros' in the downloads section of their website.

### The Raspberry Pi Foundation - Downloads

#### Python

Python is the recommended **programming language** — particularly if you are new to programming or want to refresh your programming knowledge.

#### Scratch

Scratch is a great interactive programming language for children who want to learn to code through creating games, stories and animations.

Other programming languages you can get on your Pi include C, C++, Java and Ruby.





## Raspberry Pi Frequently Asked Questions

### What educational material/resources can I use?

There is so much information out there to support you with Raspberry Pi due to it's collaborative nature.

#### DESIGNSPARK

Here at RS, we recommend DesignSpark, our own support gateway filled with blogs, forums, useful tools, product reviews and much more. You can also let us know how you get on with your projects.

[Visit DesignSpark](#)

We have a range of Raspberry Pi support books, written by Pi experts such as it's co-founder Eben Upton and Carrie Philbin.

[See our Range of Books](#)

---

### Other great Pi resources

[Raspberry Pi Foundation](#)

[MagPi - The official Pi magazine](#)

[Piweekly - Newsletter you can subscribe to](#)

[The Raspberry Pi Guy](#)

[geekurldiaries](#)

---

### Not answered your query?

DesignSpark or The Raspberry Pi Foundation website may be able to help you further.

[Visit DesignSpark](#)

[Visit The Raspberry Pi Foundation website](#)





# T5875DV Raspberry Pi Power Supply



## Features:

- Official Raspberry Pi Power Supply
- 1.5M Micro USB B Lead
- ErP Level 6 Efficiency Rating
- 50,000 Hour MTBF
- 1 Years Warranty



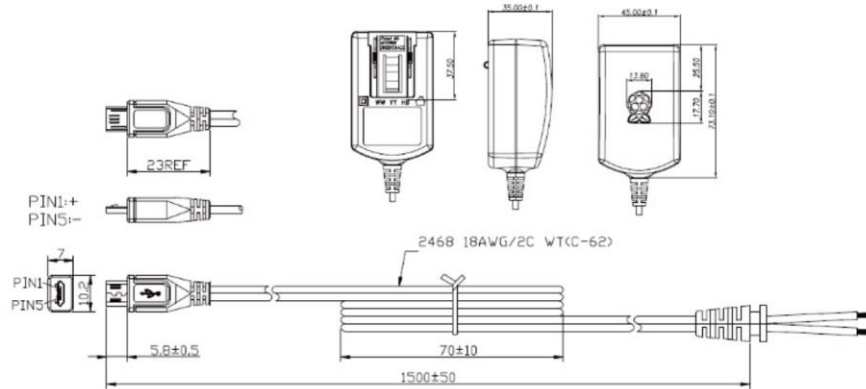
Output	
Output Voltage	+5.1Vdc
Minimum Load Current	0A
Nominal Load Current	2.5A
Nominal Output Power	13W
Output Regulation	+/-5%
Line Regulation	+/-2%
Ripple & Noise	120mVp-p Maximum
Rise Time	100mS Maximum at nominal input
Turn-on Delay	3 Seconds Maximum at nominal input
Protection	Short circuit, over current, over voltage
Efficiency	80.86%
Output Cable	1500mm Micro USB B 5 Pin

Input	
Input Voltage Range	90-264VAC
Input Frequency	47-63Hz
Input Current	0.5A Max
Inrush Current	No damage and IP fuse will not blow
AC Inlet	UK, Euro, Aus & US changeable heads

Other	
Dimensions	73.2 (L) * 45.1 (W) * 35.1 (H) mm
Weight	Approx 150g
Operating Temp.	0 °C to 40 °C
Storage Temp.	-20 °C to +60 °C
Operating Humidity	20 ~ 85 % RH. Non-Condensing
MTBF	50,000 Hours

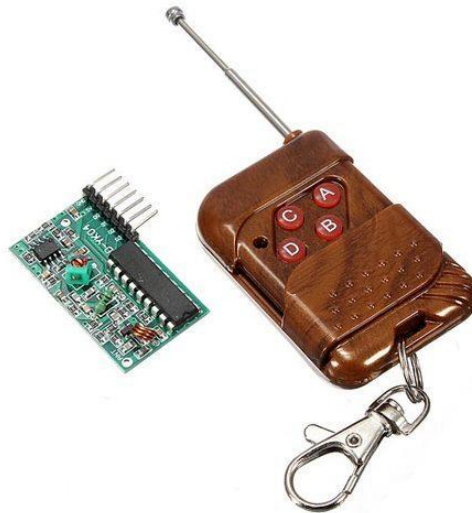
## Diagrams



## STONTRONICS

Chancerygate Business Centre, Cradock Road, Reading, Berkshire, RG2 0AH.  
 Tel: +44 (0) 118 931 1199 • Fax: +44 (0) 118 931 1145 • Email: [info@stontronics.co.uk](mailto:info@stontronics.co.uk) • Web: [www.stontronics.co.uk](http://www.stontronics.co.uk)  
 Please Note: Image shown is representative of entire range. Individual PSU image &/or drawings or data sheets available on request.  
 Stontronics Ltd accepts no responsibility for typographical errors in the production of this leaflet. Product specifications are subject to change without notice.

## ANEXO C. DATASHETT EMISOR RECEPTOR RF



### Control remoto RF 4CH

Controla tus proyectos inalámbricamente utilizando este módulo de radiofrecuencia. Ideal para controlar warbots, puertas levadizas, luminarias.

En muchas ocasiones es necesario establecer un enlace de comunicación inalámbrico de forma rápida y sencilla, siendo este control remoto RF la solución ideal. Para utilizar el control remoto RF no es necesario programar, ni utilizar algún microcontrolador adicional. Para poder utilizarlo basta con alimentar el control remoto y el receptor.

Posee 4 salidas que pueden ir conectadas a transistores de conmutación o opto acopladores y luego conectadas a relays. La frecuencia de trabajo es de 433 MHz, debido a que es una banda de libre uso.

La antena tiene una gran influencia en el módulo receptor, por lo que se recomienda conectar un cable de cobre de unos 17cm de largo para alcanzar la frecuencia de 433 MHz. La posición de la antena también afecta la recepción de datos. El cable de la antena debe estar debidamente aislado.

### ESPECIFICACIONES TÉCNICAS

#### Control Remoto:

Voltaje de operación: 12V DC (Batería modelo 27A/12V)

Corriente de trabajo: 10mA a 12V

Potencia radiada: 10mW a 12V

Modo de Modulación: ASK

Frecuencia de transmisión: 433 MHz (estabilización de frecuencia SAW)

Distancia de transmisión: 50 a 100 metros en campo abierto con sensibilidad del receptor de -100dbm

Encoder de código fijo

### **Receptor RF:**

Voltaje de operación: 5V DC

Sensibilidad del receptor: - 98db

4 Salidas (channels)

Pines: GND/VCC/SALIDAS: D0 D1 D2 D3

Antena (no incluida): Alambre de cobre de 25cm~32cm (recomendado en forma de espiral)

Dimensiones: 6.6 x 22 x 41mm

### **FUNCIONAMIENTO**

El control remoto RF posee 4 pulsadores, cada pulsador se corresponde con uno de los bits del receptor (D0, D1, D2 y D3). Cuando se presiona un pulsador, el control transmite la señal de activación del bit correspondiente, y el voltaje en el bit del receptor se eleva a 5V. Cuando se deja de presionar el pulsador, la salida del receptor retorna al estado de reposo de 0V.

### **LINKS**

Manual de uso

Tutorial

Tutorial Prometec

Modulación ASK

## ANEXO D. PROGRAMACIÓN ARDUINO UNO A

```
void setup()
{
  Serial.begin(9600);
  pinMode(pinled1,OUTPUT);
  pinMode(pinled2,OUTPUT);
  pinMode(pinled3,OUTPUT);
  pinMode(pinled4,OUTPUT);
}
void loop()
{
  if(Serial.available())
  {
    var=Serial.read();
    if(var=='a')
    {digitalWrite(pinled1,HIGH);}
    if(var=='w')
    {digitalWrite(pinled1,LOW);}
    if(var=='b')
    {digitalWrite(pinled2,HIGH);}
    if(var=='x')
    {digitalWrite(pinled2,LOW);}
    if(var=='c')
    {digitalWrite(pinled3,HIGH);}
    if(var=='y')
    {digitalWrite(pinled3,LOW);}
    if(var=='d')
    {digitalWrite(pinled4,HIGH);}
    if(var=='z')
    {digitalWrite(pinled4,LOW);}
  }
}
```

## ANEXO E. PROGRAMACIÓN ARDUINO UNO B


```
void setup()
{
  Serial.begin(9600);
  pinMode(pinled1, OUTPUT);
  pinMode(pinled2, OUTPUT);
  pinMode(pinled3, OUTPUT);
  pinMode(pinled4, OUTPUT);
}
void loop()
{
  if(Serial.available())
  {
    var=Serial.read();
    if(var=='a')
    {digitalWrite(pinled1, HIGH);}
    if(var=='w')
    {digitalWrite(pinled1, LOW);}
    if(var=='b')
    {digitalWrite(pinled2, HIGH);}
    if(var=='x')
    {digitalWrite(pinled2, LOW);}
    if(var=='c')
    {digitalWrite(pinled3, HIGH);}
    if(var=='y')
    {digitalWrite(pinled3, LOW);}
    if(var=='d')
    {digitalWrite(pinled4, HIGH);}
    if(var=='z')
    {digitalWrite(pinled4, LOW);}
  }
}
```



## ANEXO F. PROGRAMACIÓN THONNY PYTHON

Thonny - /home/pi/piwallps.py @ 52:16

File Edit View Run Device Tools Help



piwallps.py ✖

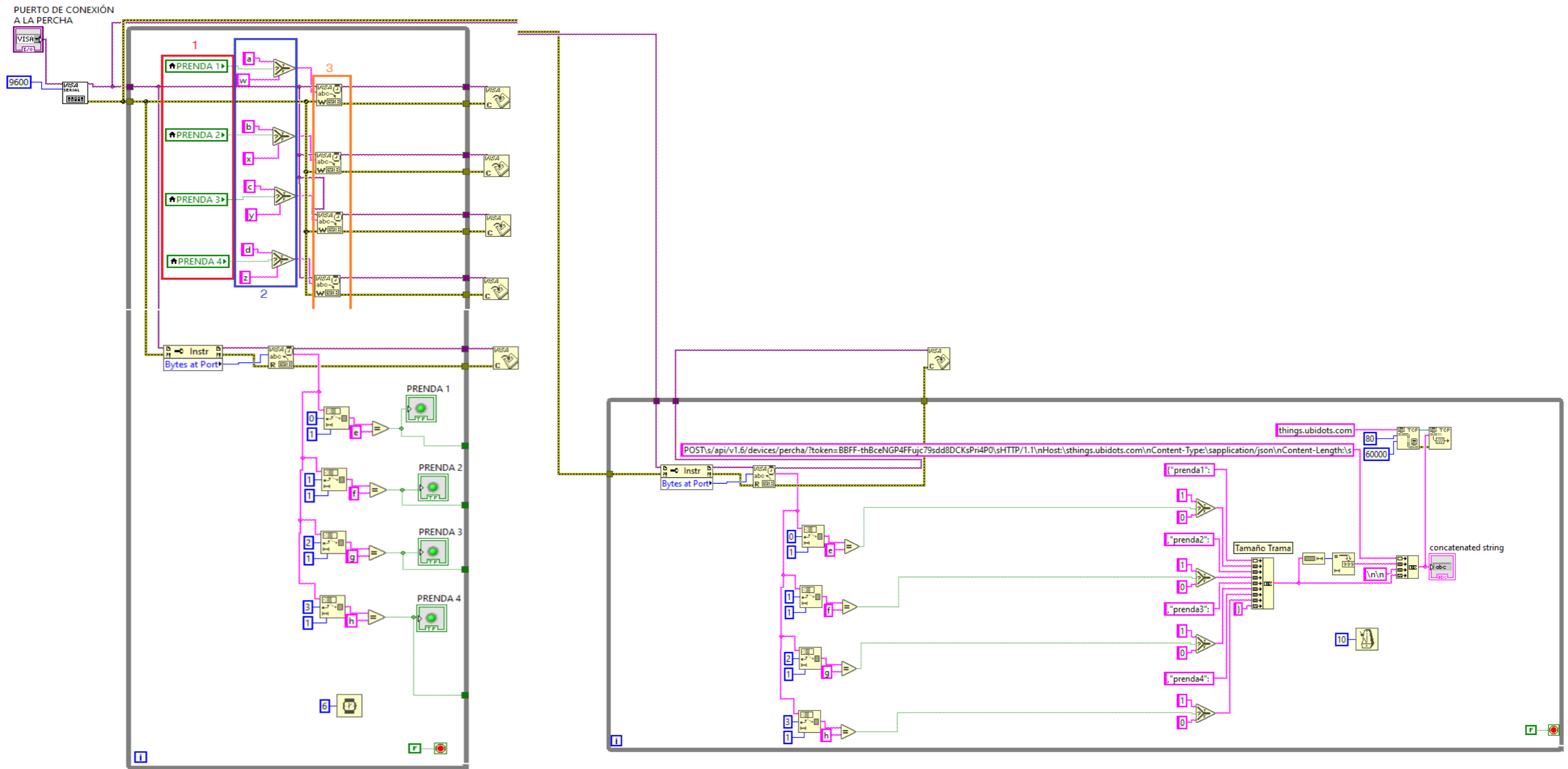
```

1 import serial
2 import time
3 import sys
4 import os
5 from subprocess import Popen
6
7 player = False
8
9
10 arduino=serial.Serial('/dev/ttyACM0', 9600)
11 # arduino.open()
12
13 print("Comenzando!!!")
14
15 while True:
16     cadena = arduino.read()
17
18     cadenaaa = (cadena.decode())
19     #print(cadenaaa)
20
21     if cadenaaa=='5':
22         #os.system('killall omxplayer.bin')
23         print("Video Promocion")
24         os.system('avconv -re -i testeo1.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
25         os.system('killall omxplayer.bin')
26         player = False
27     elif cadenaaa=='1':
28         #os.system('killall omxplayer.bin')
29         print("Video Blusa")
30         os.system('avconv -re -i testeo2.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
31         os.system('killall omxplayer.bin')
32         player = False
33     elif cadenaaa=='2':
34         #os.system('killall omxplayer.bin')
35         print("Video Pantalón")
36         os.system('avconv -re -i testeo3.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
37         os.system('killall omxplayer.bin')
38         player = False
39     elif cadenaaa=='3':
40         #os.system('killall omxplayer.bin')
41         print("Video Interiores")
42         os.system('avconv -re -i testeo4.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
43         os.system('killall omxplayer.bin')
44         player = False
45     elif cadenaaa=='4':
46         #os.system('killall omxplayer.bin')
47         print("Video Zapatos")
48         os.system('avconv -re -i testeo5.mp4 -vcodec copy -f avi -an udp://239.0.1.23:1234')
49         os.system('killall omxplayer.bin')
50         player = False
51
52 arduino.close()

```

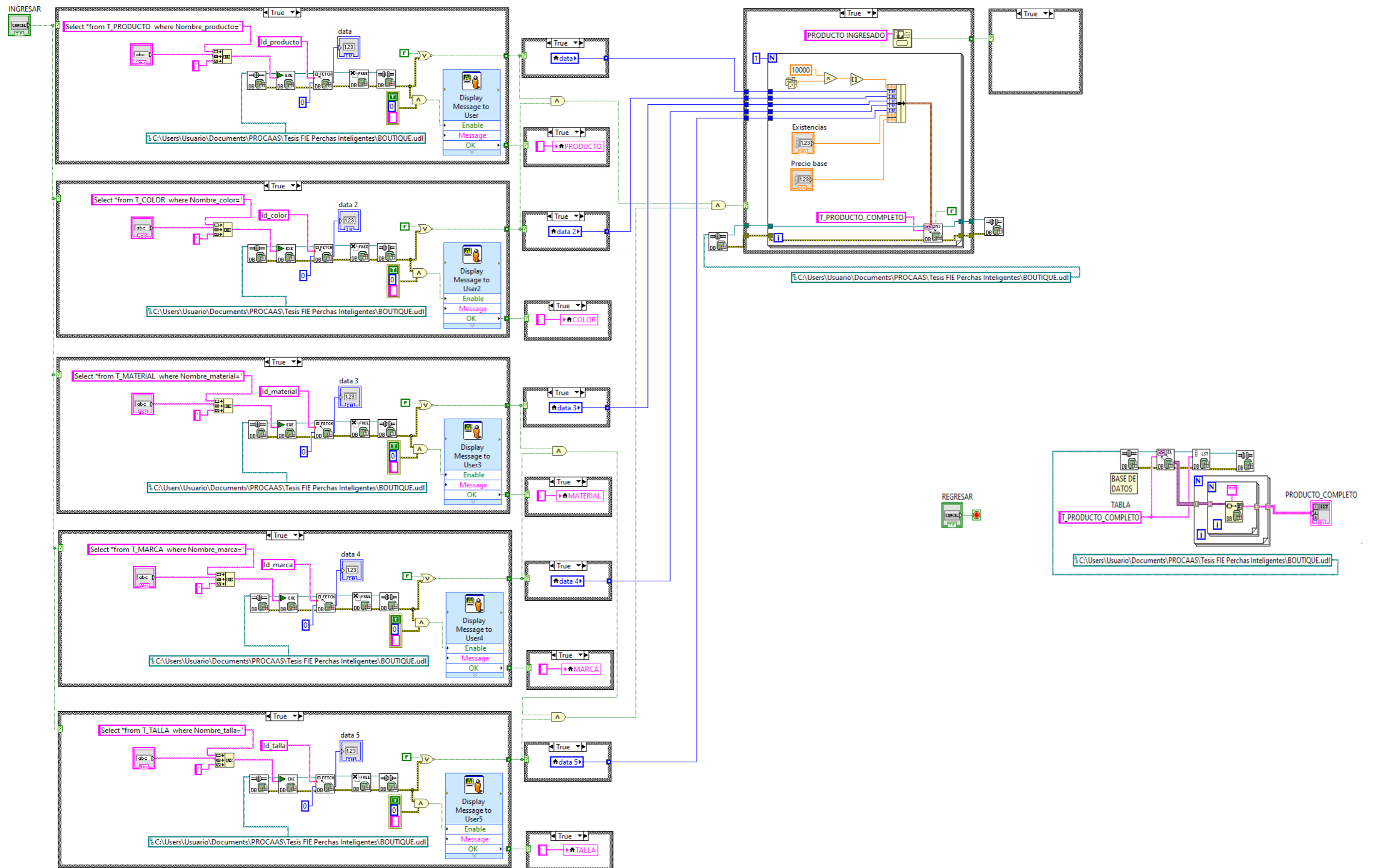
## **ANEXO G. PROGRAMACIÓN COMPLETA EN LABVIEW VI MONITOREO**

# ANEXO G. PROGRAMACIÓN COMPLETA EN LABVIEW VI MONITOREO



**ANEXO H. PROGRAMACIÓN COMPLETA EN LABVIEW VI DE GESTION DE INFORMACIÓN**

## ANEXO H. PROGRAMACIÓN COMPLETA EN LABVIEW VI DE GESTION DE INFORMACIÓN



**ANEXO I. REGISTRO DE EXPERIMENTACIÓN Y CALCULO DE EFICIENCIA DEL SISTEMA**

<b>PROCEDIMIENTO</b>	<b>ACIERTOS</b>	<b>FUNCIONALIDAD</b>	<b>RESPUESTA</b>
Comunicación emisor-receptor del Módulo de Radio Frecuencia	50 de 50	100%	Tiempo Real
Respuesta a la conexión Maestro – Esclavo de Raspberry Pi3	50 de 50	100%	Retraso 1segundo
Conmutación de videos Piwall	50 de 50	100%	Retraso 1 segundo
Monitoreo Local	50 de 50	100%	Tiempo Real
Monitoreo Remoto	47 de 50	97%	1 a 2 segundos (depende de la calidad de conexión)

<b>PRUEBA</b>	<b>COMUNICACIÓN EMISOR-RECEPTOR DEL MÓDULO DE RADIO FRECUENCIA</b>		
	<b>ACIERTO</b>	<b>NO ACIERTO</b>	<b>RESPUESTA</b>
1	X		Tiempo Real
2	X		Tiempo Real
3	X		Tiempo Real
4	X		Tiempo Real
5	X		Tiempo Real
6	X		Tiempo Real
7	X		Tiempo Real
8	X		Tiempo Real
9	X		Tiempo Real
10	X		Tiempo Real
11	X		Tiempo Real
12	X		Tiempo Real
13	X		Tiempo Real
14	X		Tiempo Real
15	X		Tiempo Real
16	X		Tiempo Real
17	X		Tiempo Real
18	X		Tiempo Real
19	X		Tiempo Real
20	X		Tiempo Real
21	X		Tiempo Real
22	X		Tiempo Real
23	X		Tiempo Real
24	X		Tiempo Real
25	X		Tiempo Real
26	X		Tiempo Real
27	X		Tiempo Real
28	X		Tiempo Real
29	X		Tiempo Real
30	X		Tiempo Real
31	X		Tiempo Real

32	X		Tiempo Real
33	X		Tiempo Real
34	X		Tiempo Real
35	X		Tiempo Real
36	X		Tiempo Real
37	X		Tiempo Real
38	X		Tiempo Real
39	X		Tiempo Real
40	X		Tiempo Real
41	X		Tiempo Real
42	X		Tiempo Real
43	X		Tiempo Real
44	X		Tiempo Real
45	X		Tiempo Real
46	X		Tiempo Real
47	X		Tiempo Real
48	X		Tiempo Real
49	X		Tiempo Real
50	X		Tiempo Real
	100% Funcionalidad		RESPUESTA EN TIEMPO REAL

PRUEBA	RESPUESTA A LA CONEXIÓN MAESTRO – ESCLAVO DE RASPBERRY PI3		
	ACIERTO	NO ACIERTO	RESPUESTA (seg)
1	X		1
2	X		1
3	X		1,05
4	X		1
5	X		1
6	X		0,9
7	X		1
8	X		1
9	X		1,08
10	X		1
11	X		1
12	X		0,85
13	X		0,9
14	X		1
15	X		0,95
16	X		1
17	X		1
18	X		1
19	X		0,9
20	X		0,9
21	X		0,9

22	X		1
23	X		1
24	X		1
25	X		1,03
26	X		1
27	X		1
28	X		1
29	X		1
30	X		0,96
31	X		1
32	X		0,87
33	X		1
34	X		1,1
35	X		1
36	X		0,9
37	X		0,95
38	X		1
39	X		0,95
40	X		1
41	X		0,9
42	X		1
43	X		1
44	X		1
45	X		1
46	X		1
47	X		1
48	X		1
49	X		1
50	X		1
	100% Funcionalidad		RESPUESTA 1S DE RETRASO

PRUEBA	Conmutación de videos Piwall		
	ACIERTO	NO ACIERTO	RESPUESTA (seg)
1	X		0,9
2	X		1
3	X		1,2
4	X		1
5	X		1
6	X		1,05
7	X		1
8	X		1,1
9	X		1,3



10	X		1,05
11	X		1
12	X		0,95
13	X		0,9
14	X		1
15	X		0,95
16	X		1
17	X		1
18	X		1,2
19	X		1
20	X		1,05
21	X		1
22	X		1
23	X		1,4
24	X		1
25	X		1
26	X		1,08
27	X		1
28	X		1
29	X		1
30	X		0,96
31	X		1
32	X		0,87
33	X		0,9
34	X		0,7
35	X		1,1
36	X		0,9
37	X		0,85
38	X		1
39	X		0,95
40	X		1
41	X		0,9
42	X		1
43	X		1,2
44	X		1
45	X		1
46	X		1,05
47	X		1

48	X		1,1
49	X		1
50	X		1
	100% Funcionalidad		RESPUESTA 1S DE RETRASO

PRUEBA	MONITOREO LOCAL		
	ACIERTO	NO ACIERTO	RESPUESTA
1	X		Tiempo Real
2	X		Tiempo Real
3	X		Tiempo Real
4	X		Tiempo Real
5	X		Tiempo Real
6	X		Tiempo Real
7	X		Tiempo Real
8	X		Tiempo Real
9	X		Tiempo Real
10	X		Tiempo Real
11	X		Tiempo Real
12	X		Tiempo Real
13	X		Tiempo Real
14	X		Tiempo Real
15	X		Tiempo Real
16	X		Tiempo Real
17	X		Tiempo Real
18	X		Tiempo Real
19	X		Tiempo Real
20	X		Tiempo Real
21	X		Tiempo Real
22	X		Tiempo Real
23	X		Tiempo Real
24	X		Tiempo Real
25	X		Tiempo Real
26	X		Tiempo Real
27	X		Tiempo Real
28	X		Tiempo Real
29	X		Tiempo Real
30	X		Tiempo Real
31	X		Tiempo Real
32	X		Tiempo Real
33	X		Tiempo Real
34	X		Tiempo Real
35	X		Tiempo Real
36	X		Tiempo Real
37	X		Tiempo Real

38	X		Tiempo Real
39	X		Tiempo Real
40	X		Tiempo Real
41	X		Tiempo Real
42	X		Tiempo Real
43	X		Tiempo Real
44	X		Tiempo Real
45	X		Tiempo Real
46	X		Tiempo Real
47	X		Tiempo Real
48	X		Tiempo Real
49	X		Tiempo Real
50	X		Tiempo Real
	100% Funcionalidad		RESPUESTA EN TIEMPO REAL

PRUEBA	MONITOREO REMOTO		
	ACIERTO	NO ACIERTO	RESPUESTA (seg)
1	X		1,1
2	X		1
3	X		1,3
4	X		1,5
5	X		1,8
6	X		1,2
7	X		1,3
8	X		1,5
9	X		1,8
10	X		1,7
11	X		1,5
12	X		1,9
13	X		1,4
14	X		1,8
15	X		1,4
16	X		1,5
17	X		1,6
18	X		1,8
19	X		1,4
20	X		1,1
21	X		2
22	X		1,8
23	X		1,9
24	X		1,5
25	X		1,7
26	X		2
27	X		1

28	X		1
29	X		1,8
30	X		1,8
31	X		1,6
32	X		1,2
33	X		1,5
34	X		1,9
35	X		2
36	X		1,1
37	X		1,5
38	X		1,8
39	X		1,5
40	X		1,9
41	X		1,8
42	X		1,2
43	X		1,9
44	X		1,9
45	X		1,5
46	X		2
47	X		1,5
48	X		1,8
49	X		1,8
50	X		2



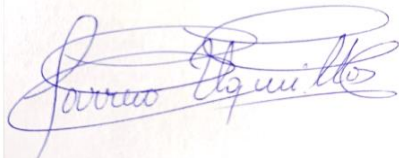

**ESCUELA SUPERIOR POLITÉCNICA DE  
CHIMBORAZO**



**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS  
PARA EL APRENDIZAJE Y LA INVESTIGACIÓN**

**UNIDAD DE PROCESOS TÉCNICOS**  
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

**Fecha de entrega:** 19 / Mayo / 2020

<b>INFORMACIÓN DEL AUTOR/A (S)</b>
Alex Javier Ichina López
<b>INFORMACIÓN INSTITUCIONAL</b>
<b>Facultad:</b> Informática y Electrónica - FIE
<b>Carrera:</b> Ingeniería Electrónica en Control y Redes Industriales
<b>Título a optar:</b> Ingeniero en Electrónica, Control y Redes Industriales
<b>f. Analista de Biblioteca responsable:</b> Jhonatan Rodrigo Parreño Uquillas
 
04-05-2020

0025-DBRAI-UPT-2020