

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Electricidad

IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE CONTEO DE PASAJERO EN BUSES URBANOS PARA CONTROL DE AFORO

Trabajo de grado presentado ante la ilustre Universidad Técnica del Norte previo a la obtención del título de Ingeniero Eléctrico

Autor:

Edwin Santiago Imbaquingo Dávila

Director:

Ing. Francisco Roberto Naranjo Cobo MSc.

Ibarra – Ecuador

2023



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACION DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100324786-1		
APELLIDOS Y NOMBRES:	Edwin Santiago Imbaquingo Dávila		
DIRECCIÓN:	Atuntaqui		
EMAIL:	esimbaquingod@utn.edu.ec		
TELÉFONO FIJO:	2530-190	TELÉFONO MÓVIL:	0979637193

DATOS DE LA OBRA	
TÍTULO:	IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE CONTEO DE PASAJERO EN BUSES URBANOS PARA CONTROL DE AFORO
AUTOR (ES):	Edwin Santiago Imbaquingo Dávila
FECHA: DD/MM/AAAA	10/07/2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input type="checkbox"/> POSGRADO <input checked="" type="checkbox"/> PREGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Electricidad
ASESOR /DIRECTOR:'	Ing. Francisco Naranjo Cobo MSc.

CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de este y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 10 días del mes de Julio de 2023

EL AUTOR:



Nombre: Edwin Santiago Imbaquingo Dávila

C.C: 100324786-1

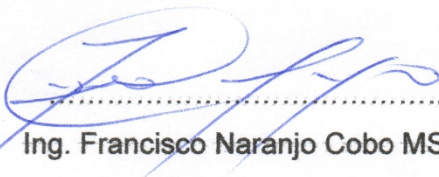


UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE GRADO

Yo, Francisco Roberto Naranjo Cobo en calidad de tutor del señor, Edwin Santiago Imbaquingo Dávila, certifico que ha culminado con las normas establecidas en la elaboración del trabajo de investigación titulado: **“IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE CONTEO DE PASAJERO EN BUSES URBANOS PARA CONTROL DE AFORO”**. Para la obtención del título de Ingeniero en Electricidad; aprobado la defensa, impresión y empastado.



Ing. Francisco Naranjo Cobo MSc.

DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

El presente trabajo de investigación lo dedico a toda mi familia. Sobre todo, a mis padres que me apoyaron para alcanzar uno de mis objetivos en mi formación profesional. Gracias por enseñarme a afrontar las dificultades que se pueden presentar en la vida. Me han enseñado a ser la persona que soy hoy, mis principios, mis valores, mi perseverancia y mi empeño. Todo esto con su amor y sin pedir nada a cambio. También quiero dedicarle este trabajo a mi novia Pamela Cevallos. Por tu paciencia, por tu comprensión, por tu fuerza, por tu amor. De hecho, ella me ayudó a lograr el equilibrio y me permitió alcanzar mi máximo potencial. Nunca dejaré de estar agradecido por eso. Por último, a mis hermanos, por apoyarme incondicionalmente durante todo este trayecto y por estar conmigo en todo momento.

AGRADECIMIENTO

En primer lugar, les agradezco a mis padres que siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Ellos son los que con su cariño me han impulsado siempre a perseguir mis metas y nunca abandonarlas frente a las adversidades. También son los que me han brindado el soporte material y económico para poder concentrarme en los estudios y nunca abandonarlos. Les agradezco a los docentes de la carrera de Ingeniería en Electricidad de la Universidad Técnica del Norte, por haberme compartido sus conocimientos a lo largo de la preparación profesional, de igual manera a mi director de tesis Ing. Naranjo Cobo Francisco Roberto MSc, por la orientación, apoyo, paciencia, guía y la ayuda brindada para el desarrollo de este trabajo de grado.

ÍNDICE DE CONTENIDO

IDENTIFICACIÓN DE LA OBRA	II
CONSTANCIA	III
CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE GRADO	¡Error! Marcador no definido.
DEDICATORIA	V
AGRADECIMIENTO	VI
ÍNDICE DE CONTENIDO	VII
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XIV
RESUMEN	XV
ABSTRACT	XVI
INTRODUCCIÓN	XVII
PLANTEAMIENTO DEL PROBLEMA.....	XVIII
FORMULACIÓN DEL PROBLEMA.....	XVIII
OBJETIVOS.....	XIX
Objetivo General	XIX
Objetivos Específicos.....	XIX
ALCANCE	XIX
JUSTIFICACIÓN.....	XX
CAPÍTULO 1	1
DESCRIBIR LOS DIFERENTES SISTEMAS DE CONTEO DE PASAJEROS	1
1.1. Tecnologías de Control de Acceso y Salida	1
1.2. Tecnologías Mecánicas.....	1
1.2.1. Torniquetes Mecánicos de Acceso	1
1.2.2. Torniquetes de Altura Completa.....	2
1.2.3. Torniquetes Rotativos.....	3

1.3.	Tecnologías con Sensores.....	3
1.3.1.	Torniquetes Ópticos.....	3
1.3.2.	Barras Contadoras de Pasajeros	4
1.4.	Tecnologías con Tarjetas	5
1.4.1.	Tarjetas de Contacto	5
1.4.2.	Tarjetas sin Contacto.....	5
1.4.3.	Tarjetas Inteligentes.....	6
1.5.	Tecnologías con Aplicaciones Móviles	6
1.6.	Tecnologías con Visión Artificial	7
1.7.	Visión Artificial	8
1.8.	Fases de la Visión Artificial	8
1.8.1.	Iluminación	9
1.8.2.	Adquisición.....	9
1.8.3.	Procesamiento	9
1.8.4.	Actuadores	9
1.8.5.	Microcontrolador	9
1.8.6.	Pantalla	9
1.9.	OpenCV	10
1.10.	Técnicas de Detección de Objetos	10
1.10.1.	Sustracción de fondos.....	10
1.10.2.	Detección de Bordes Canny	11
1.10.3.	Detección de Haar-Cascade.....	12
CAPÍTULO 2		15
DISEÑO DEL PROTOTIPO DE CONTEO DE PASAJEROS CON CONTROL DE AFORO		15
2.1.	Descripción del Lugar.....	15
2.2.	Metodología	15
2.2.1.	Análisis.....	16
2.2.2.	Diseño	17

2.2.3.	Implementación.....	17
2.2.4.	Pruebas.....	17
2.3.	Diseño Electrónico.....	18
2.3.1.	Microprocesadores	18
2.3.2.	Microcontrolador	19
2.3.3.	Matriz de LEDS.....	21
2.3.4.	Cámara	22
2.3.5.	Método Haar Cascade.....	23
2.3.5.1.	Imagen Integral.....	24
2.3.5.2.	Extracción de Características.....	25
2.3.5.3.	Clasificador	26
2.3.6.	Pantalla HMI.....	26
2.3.7.	Sensor de distancia	28
2.3.8.	Modulo Relé.....	29
2.3.9.	Electroválvula.....	29
2.4.	Diseño Eléctrico.....	30
2.4.1.	Batería del Autobús	30
2.4.2.	Regulador Reductor.....	30
2.4.3.	Toma de mechero.....	31
2.5.	Software.....	31
2.5.1.	Lenguaje de Programación.....	31
2.5.2.	IDE Visual Studio Code	32
2.5.3.	IDE Arduino.....	32
2.5.4.	IDE Nextion Editor	32
2.6.	Resultados.....	32
2.6.1.	Diagrama de bloques y descripción funcional.....	33
2.6.2.	Esquema de conexión	33
2.6.3.	Programación de las Raspberry Pi 4.....	35

2.6.4.	Programación de la Pantalla HMI.....	36
2.6.5.	Programación del Arduino Mega.....	37
CAPÍTULO 3		39
CONSTRUCCIÓN E IMPLENTACIÓN DEL PROTOTIPO DE CONTEO DE PASAJEROS		
CON CONTROL DE AFORO Y PRUEBAS DE FUNCIONAMIENTO.....		
3.1.	Materiales utilizados para la elaboración del prototipo.....	39
3.2.	Distribución y montaje de los compones electrónicos en el autobús.....	40
3.2.1.	Montaje del sistema de conteo en la puerta de entrada.....	41
3.2.2.	Montaje de Sistema de conteo en la puerta de salida	43
3.2.3.	Montaje de las Matrices Leds.....	45
3.2.4.	Montaje del Actuador en la puerta de entrada.....	48
3.3.	Software.....	49
3.3.1.	Interfaz de la Pantalla HMI	50
3.3.2.	Matriz Leds con espacios restante.....	52
3.3.3.	Matriz Leds con las rutas.....	53
3.3.4.	Reconocimiento facial.....	53
3.4.	Pruebas del Prototipo.....	56
3.4.1.	Pruebas Sistema Eléctrico.....	56
3.4.2.	Pruebas Sistema Electrónico	56
3.4.3.	Pruebas de Software	59
3.5.	Resultados.....	60
3.5.1.	Funcionalidad del Sistema de Conteo.....	61
3.5.2.	Funcionalidad del Control de Aforo	61
CONCLUSIONES		
RECOMENDACIONES.....		
REFERENCIAS		
ANEXOS		
	Hojas de datos de la Raspberry PI 4.....	71

Hoja de datos Arduino Mega	74
Hoja de datos Pantalla HMI	77
Hoja de datos Cámaras	79
Hoja de datos Matriz de Leds	81
Hoja de datos Matriz de Leds	82
Código Raspberry Principal	84
Código Raspberry Secundario.....	91
Código App cambio de aforo y rutas	93
Código Arduino Mega Aforo	94
Código Arduino Mega Rutas.....	96
Manual de Usuario	98
Manual Técnico.....	111

ÍNDICE DE FIGURAS

Figura 1 Tornos o Torniquetes de control de acceso	2
Figura 2 Torniquetes de seguridad para peatones	2
Figura 3 Control de acceso con torniquetes rotativos.....	3
Figura 4 Control de acceso con torniquetes ópticos.....	4
Figura 5 Barras contadoras de pasajeros	4
Figura 6 Tarjetas con chip de contacto	5
Figura 7 Tarjeta de acceso sin contacto	6
Figura 8 Tarjeta de control de acceso MIFARE.....	6
Figura 9 Interfaz de la Aplicación Movilflix	7
Figura 10 Cámara de conteo de personas para trenes y autobuses	7
Figura 11 Proceso de la Visión Artificial.....	8
Figura 12 Logo de OpenCV.....	10
Figura 13 Modelos de Sustracción de fondos.....	11
Figura 14 Ejemplo de detección de Bordes Canny.....	12
Figura 15 Los Haar Features.....	13
Figura 16 Ubicación de la Cooperativa Orozcotola	15
Figura 17 Esquema del desarrollo de la metodología en espiral	16
Figura 18 Raspberry PI 4.....	19
Figura 19 Arduino Mega 2560	20
Figura 20 Matriz led 64X32 RGB.....	22
Figura 21 Cámara Logitech C270	23
Figura 22 Diagrama de bloques de la detección Haar-Cascade.....	24
Figura 23 Cálculo de la integral.....	25
Figura 24 Ejemplos de rasgos de detección de rostros.....	25
Figura 25 Esquema del clasificador en cascada	26
Figura 26 Pantalla Nextion de 5 pulgadas	28
Figura 27 Sensor ultrasonido HC-SR04.....	28
Figura 28 Modulo relé de un canal.....	29
Figura 29 Electroválvula de 12V.....	29
Figura 30 Cargador de carro	30
Figura 31 Toma de mechero	31
Figura 32 Diagrama Funcional de Bloques del Sistema.....	33
Figura 33 Esquema de Conexión	34
Figura 34 Conexión Arduino Mega y Matriz de Leds 64x32.....	35

Figura 35 <i>Flujograma de Programación de las Raspberry Pi 4</i>	36
Figura 36 <i>Flujograma de la Programación de la Pantalla HMI</i>	37
Figura 37 <i>Flujograma de la Programación del Arduino Mega</i>	38
Figura 38 <i>Autobús de la Cooperativa Orozcotola</i>	40
Figura 39 <i>Caja principal del sistema</i>	41
Figura 40 <i>Diagrama de conexión de los componentes de la caja principal</i>	41
Figura 41 <i>Ensamble en la caja principal</i>	42
Figura 42 <i>Implementación del prototipo en el autobús</i>	43
Figura 43 <i>Caja secundaria del sistema</i>	43
Figura 44 <i>Diagrama de conexión de los componentes de caja secundaria</i>	44
Figura 45 <i>Ensamble en la caja secundaria</i>	45
Figura 46 <i>Implementación del Prototipo en el Autobús</i>	45
Figura 47 <i>Cajas para las Matrices Leds</i>	46
Figura 48 <i>Diagrama de conexión de los componentes de las cajas de las matrices leds</i>	46
Figura 49 <i>Ensamble de las Matrices leds</i>	47
Figura 50 <i>Implementación del prototipo en el autobús</i>	48
Figura 51 <i>Diagrama de conexión del Actuador</i>	48
Figura 52 <i>Implementación del prototipo en el autobús</i>	49
Figura 53 <i>Esquema de conexión general</i>	49
Figura 54 <i>Interfaz de la Pantalla HMI Nextion</i>	50
Figura 55 <i>Distribución de los colores de la barra de Estado</i>	51
Figura 56 <i>Interfaz de la Matriz led con espacios restantes</i>	53
Figura 57 <i>Interfaz de la matriz con las rutas</i>	53
Figura 58 <i>Diagrama de flujo general del Prototipo</i>	55
Figura 59 <i>Conexión eléctrica del Prototipo</i>	56
Figura 60 <i>Pruebas del Prototipo</i>	60
Figura 61 <i>Pruebas del Prototipo en una jornada de trabajo</i>	61

ÍNDICE DE TABLAS

Tabla 1 <i>Modelos de microprocesadores del mercado</i>	18
Tabla 2 <i>Modelos de microcontroladores del mercado</i>	20
Tabla 3 <i>Modelos de las Matrices de LEDS del mercado</i>	21
Tabla 4 <i>Modelos de Cámaras Compatibles con Raspberry Pi 4</i>	22
Tabla 5 <i>Pantallas HMI</i>	27
Tabla 6 <i>Materiales utilizados en el Prototipo</i>	39
Tabla 7 <i>Conexión de pines caja principal</i>	42
Tabla 8 <i>Conexión de pines caja secundaria</i>	44
Tabla 9 <i>Conexión de pines caja de matrices</i>	47
Tabla 10 <i>Interfaz Pantalla HMI</i>	52
Tabla 11 <i>Pruebas realizadas en el autobús</i>	57

RESUMEN

El conteo de personas en las unidades de transporte urbano es algo muy demandado para el sector del transporte, para poder brindar un mejor servicio a las personas y conocer las ganancias del autobús en una jornada de trabajo, igualmente, debido a que no existe un control de aforo los conductores de los autobuses tienen la facilidad de llevar una gran cantidad de personas que no va acorde con las dimensiones del autobús y sin preocuparse por la comodidad y seguridad de los usuarios.

En el presente trabajo de grado, se desarrolló un prototipo de un sistema de conteo de pasajeros con control de aforo, que permita controlar el exceso de pasajeros en la unidad de transporte bloqueando la puerta de ingreso del autobús, además, de tener la capacidad de contar a las personas mediante visión artificial con la ayuda de dos cámaras colocadas en las puertas del autobús, con me método Haar-Cascade, las personas y el conductor tendrán la facilidad de observar la cantidad de usuarios en la unidad mediante una matriz leds y pantalla HMI respectivamente, de esta manera poder brindar un mejor servicio a la ciudadanía.

El trabajo concluye mediante la implementación en un autobús de la Cooperativa Orozcotola del cantón Antonio Ante, y realizar las pruebas respectivas del funcionamiento de este, mediante una prueba realizada en las jornadas de trabajo normal de este autobús para poder evidenciar el correcto funcionamiento del prototipo y verificar si es capaz de monitorear y controlar el aforo.

Palabras Clave: prototipo, visión artificial, Haar-Cascades, sistema de conteo, autobús.

ABSTRACT

The counting of people in the urban transport units is something highly demanded for the transport sector, in order to provide a better service to people and to know the earnings of the bus in a working day, also, because there is no control capacity, bus drivers have the facility to carry a large number of people that are not in accordance with the dimensions of the bus and without worrying about the comfort and safety of the users.

In the present degree work, a prototype of a passenger counting system with capacity control was developed, which allows controlling the excess of passengers in the transport unit by blocking the entrance door of the bus, in addition to having the capacity to counting people by artificial vision with the help of two cameras placed on the bus doors, with the Haar-Cascade method, people and the driver will have the facility to observe the number of users in the unit through an LED matrix and screen HMI respectively, in this way to be able to provide a better service to citizens.

The work concludes through the implementation in a bus of the Cooperativa Orozcotola of the Antonio Ante canton, and carry out the respective tests of its operation, through a test carried out in the normal working days of this bus in order to demonstrate the correct functioning of the prototype and verify if it is capable of monitoring and controlling the capacity.

Keywords: prototype, artificial vision, Haar-Cascades, counting system, bus.

INTRODUCCIÓN

La Universidad de Malaysia Perlis (UniMAP), realizó un proyecto de control de aforo para el uso de la ciudadanía, con el fin de utilizar los transportes de acuerdo a la capacidad y número de pasajeros que se recogerían por cada ruta donde los buses fueran designados, su principal objetivo, fue determinar cuántas personas se subieron a los buses por cada ruta, y de esta forma, no utilizar los buses grandes dejando dentro del transporte muchos asientos vacíos, sino, seguir la misma ruta pero cambiando a un transporte más pequeño, donde sí entran el mismo número de personas pero no sobrarían asientos de forma innecesaria (Nasir et al., 2018). Pero debido a los pasajeros que entran y salen del autobús sin orden, en ocasiones se logran esconder unos detrás de otros, siendo un gran desafío para los sistemas de conteo de pasajeros, debido a que no pueden contabilizar automáticamente a los pasajeros a través de una cámara (Liu et al., 2020).

Existe una gran variedad de tecnologías para el conteo automático, sean estos manuales, o sistemas digitales muy avanzados. La contabilización de objetos se ha transformado en un importante indicativo en diferentes campos, incluso en las industrias productoras, donde es fundamental realizar un conteo de los productos elaborados; del mismo modo, se considera necesario llevar un control del aforo en las unidades de transporte público; es por esto, que se ha evidenciado la necesidad de investigar sobre el diseño de sistemas de conteo más precisos y a costos accesibles (INFODEV, 2016).

Según Morales (2017), en su tesis de grado de la Universidad Técnica de Ambato "*Sistema de Visión Artificial con Monitoreo Web para Conteo de Pasajeros en Buses de Transporte Público*" tuvo como objetivo brindar un excelente servicio a las personas al momento de subirse a los autobuses, y darle al conductor la información exacta de su ganancia en sus horarios de trabajo. En este proyecto, se investigó los diferentes sistemas de conteo de pasajeros que tiene las cooperativas de transporte en la ciudad de Ambato. Este sistema de conteo de pasajeros permite registrar el ingreso y salida de los usuarios; el sistema es capaz de identificar a los pasajeros mediante una cámara, esta capta y envía los registros usando la red GSM/GPRS, la construcción de este sistema de conteo se basó en software y hardware libre (Morales, 2017).

Según Ortega (2019) en su Reporte Técnico de Investigación de la Universidad Autónoma de Ciudad Juárez "*Prototipo para el conteo de personas mediante la detección de rostros en el transporte Público*" tuvo como enfoque el disminuir la mala administración y gestión del transporte público, brindando un prototipo con la capacidad de detectar el rostro de las personas cuando se suben y bajan del transporte público, guardando un historial de

las personas que estuvieron en las unidades de transporte. En las pruebas prácticas del sistema de conteo de personas que incorpora el algoritmo de seguimiento Median Flow para rastrear personas, este sistema tuvo algunos inconvenientes al momento del conteo, detectando rostros falsos en ambientes con variaciones de iluminación dando un conteo falso en algunas ocasiones al momento de su funcionamiento (Ortega, 2019).

Hsu Ya-Wen (2020) en su artículo de investigación *“Passenger flow counting in buses based on deep learning using surveillance video”* tuvo como objetivo el conteo de pasajeros en un autobús, basándose en un sistema donde al abrirse la puerta de forma automática, el detector determinará las características de los pasajeros y finalmente se utiliza un filtro de partículas mediante un esquema de asociación para el seguimiento de pasajeros. Según sus resultados experimentales, este sistema tiene una efectividad del 91,95 %. Es importante recalcar que cuando el transporte tiende a tener un exceso de pasajeros no puede contar con precisión debido a la multitud, reduciendo su tasa de efectividad (Hsu et al., 2020).

PLANTEAMIENTO DEL PROBLEMA

En los últimos años, los sistemas de conteo de pasajeros de los transportes urbanos han tenido un avance tecnológico muy importante, mediante el uso de la visión artificial se ha logrado un cambio muy significativo en la vida diaria de las personas, permitiendo monitorear la cantidad de usuarios en un transporte público.

En la actualidad, los transportes públicos urbanos del cantón Antonio Ante, no cuentan con un sistema de conteo de pasajeros, sino, que tuvieron un sistema de caja común para ver cuantas unidades se necesitan para cumplir con la demanda de pasajeros, pero los conductores de las unidades tienden a llevar a una cantidad excesiva de pasajeros sin preocuparse de la comodidad y seguridad de los usuarios que se suben a las unidades.

Estas complicaciones se generan por no tener un monitoreo que permita alertar sobre el número de pasajeros que ingresan; en este contexto, este sistema brindará a la sociedad la posibilidad de controlar, monitorear y cumplir con el aforo permitido, mejorando incluso las condiciones de seguridad en el interior de las unidades.

FORMULACIÓN DEL PROBLEMA

¿Cómo implementar un prototipo de un sistema de conteo de pasajeros, que permita monitorear el aforo de personas en las unidades de transporte, por medio de visión artificial?

OBJETIVOS

Objetivo General

Implementar un prototipo de sistema de conteo de pasajeros en los buses de transporte urbano en el cantón Antonio Ante, para disminuir la cantidad excesiva de personas en el interior de las unidades, a través del uso de visión artificial.

Objetivos Específicos

- Describir la situación actual de los diferentes sistemas de conteo de pasajeros en las unidades de transporte público.
- Diseñar un prototipo que realice el conteo de pasajeros y monitoreo del aforo en las unidades de transporte urbano.
- Implementar el prototipo de un sistema de conteo de pasajeros y control de aforo en el transporte urbano del cantón Antonio Ante.

ALCANCE

El presente trabajo de titulación está enfocado en la realización de un sistema de conteo de pasajeros con visión artificial para control de aforo, mediante la automatización de la puerta de entrada en las unidades de transporte urbano, mismo que se implementará en el cantón Antonio Ante, en uno de los buses urbanos de la Cooperativa Orozcotola, con la finalidad de monitorear y controlar el aforo en su interior y brindar un excelente servicio a la ciudadanía.

Para el diseño del prototipo se realizará una investigación y análisis de los diferentes sistemas de conteo de pasajeros con visión artificial que existen en la actualidad, de modo que esto aporte en la creación e implementación de un sistema confiable. El sistema tendrá la facilidad de adaptarse a cualquier capacidad del autobús, mediante el ingreso del número de personas que soporta la unidad, en una pantalla táctil la misma que indicará el número de lugares disponibles al conductor y el total de personas que ingresaron a la unidad.

Por medio del estudio de los sistemas de conteo, se establecerá un prototipo que sea capaz de detectar y cuantificar el número de personas que suben y bajan de las unidades de transporte utilizando dos cámaras, dando lugar a un registro preciso para el respectivo control de aforo, mediante matrices leds se indicara el número de personas y lugares disponibles en la unidad de transporte, con la finalidad que la personas conozcan la cantidad de lugares

disponibles. Cuando el bus complete el aforo, la puerta automatizada no permitirá el ingreso de más pasajeros, hasta que disponga nuevamente de lugares disponibles.

Para corroborar el adecuado funcionamiento del sistema, se efectuarán las pruebas respectivas en el autobús para así identificar si el prototipo permite realmente monitorear y controlar el aforo.

JUSTIFICACIÓN

Es importante que los transportes urbanos del cantón Antonio Ante, cuenten con un sistema de conteo de pasajeros y control de aforo en el autobús, permitiendo así, que las personas viajen con tranquilidad y comodidad a su lugar de destino.

Existen diversos sistemas de conteo de pasajeros, el prototipo planteado en este trabajo, y su implementación, significará una mejora notable siendo un mecanismo amigable con los usuarios aportando a que el servicio de transporte brindado sea mejor. El sistema deberá controlar el aforo de pasajeros, mediante una adaptación en la puerta de entrada, que permita el ingreso de personas cuando el transporte cuente con lugares disponibles; a su vez, cuando el transporte no tenga lugares disponibles, no se permitirá el ingreso a más personas en la unidad de transporte; permitiendo al conductor, llevar la cantidad exacta de pasajeros según la capacidad de la unidad, asegurando que las personas viajen de forma segura.

La implementación de este sistema representa un avance tecnológico para mantener el control de aforo, permitiendo que la información sea accesible tanto para los pasajeros como para el conductor, de modo que se pueda controlar y evitar la aglomeración en el interior de las unidades, cumpliendo con los parámetros establecidos para brindar un servicio de calidad a los usuarios.

CAPÍTULO 1

DESCRIBIR LOS DIFERENTES SISTEMAS DE CONTEO DE PASAJEROS

En este apartado se detallaron las diversas tecnologías empleadas en los sistemas de conteo; también se analizaron las diferencias técnicas de visión artificial para determinar así el sistema más adecuado a implementar en el prototipo con la finalidad de que cumpla con los parámetros requeridos y funcione correctamente.

1.1. Tecnologías de Control de Acceso y Salida

En la actualidad existen diferentes métodos, equipos y dispositivos capaces de detectar la cantidad de personas que ingresan a edificios, transportes y centros comerciales, estos mecanismos de conteo de personas utilizan una gran variedad de actuadores como: torniquetes, tarjetas, cámaras, sensores, para determinar la cantidad de personas que ingresan y salen, a continuación, se describe algunos de los sistemas de conteo que fueron evolucionado a lo largo de los años.

1.2. Tecnologías Mecánicas

Las tecnologías mecánicas, son aquellos dispositivos de conteo de personas que están compuestos por una operación mecánica, dentro de este grupo tenemos a los torniquetes, estos elementos son capaces de contar personas, cuando estas cruzan entre ellos, teniendo un registro del número de personas que ingresan y salen, de los diferentes lugares donde estén instalados estos dispositivos como: autobuses, estadios, estaciones de metro entre otros.

1.2.1. Torniquetes Mecánicos de Acceso

Los torniquetes son dispositivos que incorporan tres brazos giratorios como se puede observar en la Figura 1, el funcionamiento de estos torniquetes se basa en el empuje de las personas para iniciar el movimiento de los brazos del torniquete y a partir de ahí este proporciona el giro suave y preciso permitiendo el ingreso de las personas, estos torniquetes son empleados en ingresos a oficinas, estadios, transporte público entre otros (Gizatullin et al., 2022).

Figura 1

Tornos o Torniquetes de control de acceso



Nota. Tomado de "Experimental Studies of Building Access Control Systems Under the Influence of Electrostatic Discharge" (p.1350), por M. Gizatullin, 2022, IEEE.

1.2.2. Torniquetes de Altura Completa

Estos torniquetes tienen mayor nivel de seguridad que otros, debido a su construcción con barreras metálicas de pies a cabeza, es muy difícil que una persona sin autorización pueda pasar a través de este control de acceso como se puede observar en la Figura 2, los torniquetes de altura completa son similares a una puerta giratoria que restringe el paso por persona, debido a que no pueden pasar varias personas al mismo tiempo (*Control de Accesos*, 2022).

Figura 2

Torniquetes de seguridad para peatones



Nota. Tomado de Torniquetes de alturas completa, por BOON EDAM, 2022, (<https://www.boonedam.com/es/productos/torniquetes-de-altura-completa>).

1.2.3. Torniquetes Rotativos

Los Torniquetes rotativos tiene un mecanismo de funcionamiento más eficaz brindando un control de acceso confiable, permitiendo el ingreso de personas mediante el giro de sus puertas donde da el acceso de entrada y salida como se observa en la Figura 3, además, estos torniquetes son de media altura facilitando el uso a las personas (Derek, 2020).

Figura 3

Control de acceso con torniquetes rotativos



Nota. Tomado de Control de acceso con torniquetes, por A. Derek, 2022, eurossoft (<https://solucioneseurossoft.com/control-acceso/control-de-acceso-con-torniquetes/>).

1.3. Tecnologías con Sensores

Las tecnologías con sensores son sistemas que utilizan una amplia variedad de actuadores, estos pueden ser infrarrojos, ultrasonidos, sensores de distancia o movimiento entre muchos otros, estos permiten el ingreso de las personas de una forma más fácil, debido a que estos sensores no tienen contacto con las personas como en el caso de los torniquetes, ya que, los sensores solo actúan cuando detectan a las personas, de esta manera realizan el conteo respectivo de ingreso y salida.

1.3.1. Torniquetes Ópticos

Los torniquetes ópticos son modelos que funcionan de una forma más eficaz, estas puertas están diseñadas solo para el ingreso o salida de las personas, pero también pueden ser utilizadas en doble sentido, están formados por dos gabinetes con una cierta distancia entre ellos como se aprecia en la Figura 4, estos gabinetes están conectados por rayos infrarrojos, ayudando a detectar cuando una persona ha pasado por el torniquete, las puertas son idóneas como paso de salida libre, estando las puertas en posición abierta por defecto y

cerrándose en caso de que los sensores detecten un usuario intentando entrar por la salida, su aplicación es en estaciones de metro o ingreso a edificios (A3M, 2019).

Figura 4

Control de acceso con torniquetes ópticos



Nota. Tomado de como ayudan los torniquetes de acceso a la seguridad, por Revista Seguridad 360, 2022, (<https://revistaseguridad360.com/destacados/torniquetes-de-acceso/>).

1.3.2. Barras Contadoras de Pasajeros

Es un sistema capaz de contar pasajeros que consta de una barra de filas leds infrarrojos emisores y otra barra de leds receptores posicionados unos al frente del otro, estos dispositivos se colocan tanto en la puerta delantera como en las puertas traseras con su correspondiente fuente de alimentación y baterías de respaldo como se observa en la Figura 5. La barra que hace la función de receptor posee una tarjeta de procesamiento, memorias, y puertos para periféricos con el fin de visualizar el contador (BEA-RT, 2018).

Figura 5

Barras contadoras de pasajeros



Nota. Tomado de sistemas automáticos de conteo de pasajeros, por CVX-R SISTEMAS DE CONTROL, 2022, (<https://cvx-r.cl/servicios/contador-de-pasajeros/>).

1.4. Tecnologías con Tarjetas

Las tecnologías con tarjetas son sistemas utilizados principalmente para cobros de pasaje o ingreso a edificios, también pueden llevar un registro de las personas que ingresan a las unidades de transporte, ya que, su funcionamiento es solo pasar la tarjeta por el lector respectivo, para que permita el ingreso de las personas y así las contabilice. Estas tarjetas pueden ser de diversos tipos como: Tarjetas de Contacto, Tarjetas sin Contacto Tarjetas Inteligentes entre otras.

1.4.1. Tarjetas de Contacto

Estas tarjetas deben colocarse a un sensor para que este registre el ingreso o salida de las personas, esta tarjeta tiene dos diferentes tipos como: tarjetas síncronas y asíncronas, las tarjetas síncronas son las que tiene únicamente la función de memoria, esta memoria es similar a la tarjeta de celulares que se las conoce como chip como se observa en la Figura 6, también estas pueden ser de memoria libre o protegida, las tarjetas asíncronas son las que tiene un microprocesador y una cinta magnética en la parte posterior dentro de estas pueden contener RAM, ROM y EEPROM (Derek, 2020).

Figura 6

Tarjetas con chip de contacto



Nota. Tomado de Tarjetas de contacto con chip para el ingreso de personas, por A. tarjesol soluciones con tarjetas plasticas, 2019, tarjesol (<https://www.tarjesol.com/tarjeta-con-chip-de-contacto/>).

1.4.2. Tarjetas sin Contacto

Este tipo de tarjetas es muy similar a las tarjetas con contacto, como se observa en la Figura 7. Tienen la misma función, pero la única diferencia es que no debe haber contacto directo entre el sensor de acceso y la tarjeta sino una aproximación. Además, cuenta con una bobina de antena metálica integrada que transporta la información; entonces, al ser

escaneado, el dispositivo leerá los datos y permitirá su ingreso. Esto puede ser muy útil para la salud e higiene de las personas (Roger, 2021).

Figura 7

Tarjeta de acceso sin contacto



Nota. Tomado de Tarjetas de acceso legible con etiqueta inteligente, por Hilitand, 2019, Amazon (<https://www.amazon.com/-/es/tarjetas-proximidad-contacto-etiqueta-inteligente/dp/B083B79THH>).

1.4.3. Tarjetas Inteligentes

Estas tarjetas tienen la misma función que las demás, pero con el avance tecnológico que les permite incluir una gran cantidad de información para poder controlar los accesos, también, estas tarjetas no solo sirven para el ingreso de personas sino para pagar productos o incluso para pagar el transporte público de una manera más confiable y efectiva como se aprecia en la Figura 8, siendo una de las tarjetas inteligentes más económicas y utilizadas para los sistemas de control de acceso en transporte es la tarjeta MIFARE (Veriddica, 2022).

Figura 8

Tarjeta de control de acceso MIFARE.



Nota. Tomado de Tarjeta MIFARE Plus, por A3M, 2020, A3M (<https://a3m.eu/es/tarjeta-mifare-plus>).

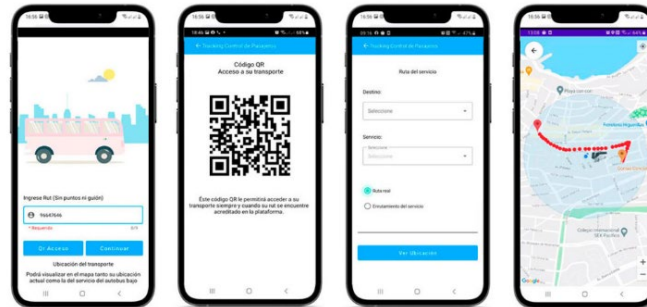
1.5. Tecnologías con Aplicaciones Móviles

Estas tecnologías se encargan de contar a los pasajeros, realizar el cobro del pasaje, y tienen otras facilidades como saber las rutas del autobús, también se puede recargar los boletos electrónicos con transferencia y tarjetas de débito o crédito. Las aplicaciones tienen

códigos QR como se observa en la Figura 9, permitiendo acceder a las personas a los autobuses. Debido a esto los sistemas son más seguros e interesantes para las personas que conocen sobre estas aplicaciones móviles.

Figura 9

Interfaz de la Aplicación Moviflix



Nota. Tomado de Control e Identificación de Pasajeros de Buses, por Moviflix, 2020, (<https://moviflix.com/control-e-identificacion-de-pasajeros-de-buses/>).

1.6. Tecnologías con Visión Artificial

Consiste en sistemas para contabilizar a los pasajeros en tiempo real mediante cámaras que los detectan uno por uno, sus movimientos y establecen así el ingreso o salida del transporte o estación. En la Figura 10 se observa un ejemplo de las cámaras utilizadas en estos sistemas que permiten lograr un control de usuarios más preciso, fácil y eficiente, de tal modo que se evita aglomeraciones (SmartCounter, 2019).

Figura 10

Cámara de conteo de personas para trenes y autobuses



Nota. Tomado del conteo de personas de Hikvision probado en SGSE, por Soluciones Globales de Seguridad Electrónica, 2020, (<http://www.sgse.eu/el-conteo-de-personas-de-hikvision-probado-en-sgse/>).

Tomando en cuenta todas las tecnologías descritas anteriormente y como estamos en una época donde la visión artificial es algo muy novedoso, el diseño del prototipo de un

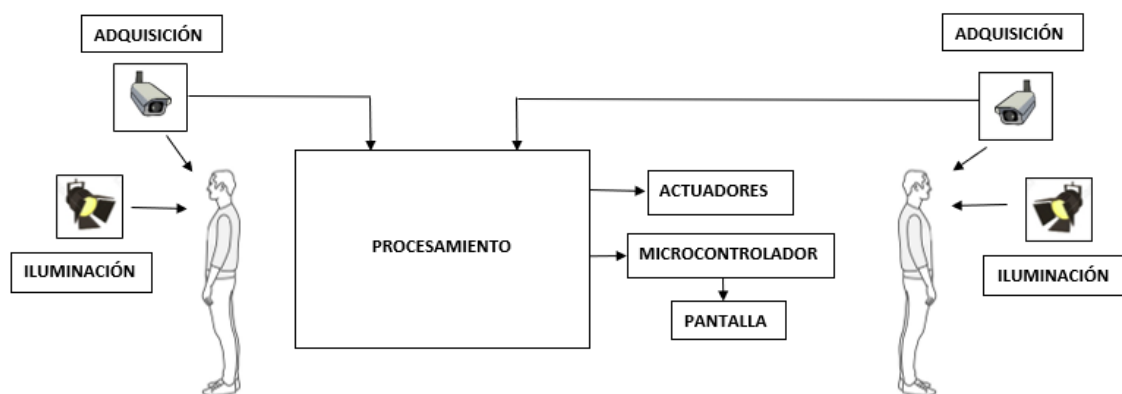
sistema de conteo de personas con control de aforo se implementó con la tecnología de visión artificial, la cual permitió un enfoque más creativo. De esta forma, obtener un reconociendo más preciso de las personas que suben y bajan del medio de transporte, además, facilitar el control de aforo en la unidad de transporte. Por otra parte, este sistema es el que mejor se acopla con los buses urbanos del cantón Antonio Ante, ya que, la infraestructura del bus no permite la instalación de las otras tecnologías mencionadas de conteo de personas, además, tiene mayor adaptabilidad para todos los usuarios.

1.7. Visión Artificial

Es el proceso por el que un computador es programado para procesar imágenes y videos con la finalidad de extraer sus datos y utilizarlos para ejecutar alguna acción determinada. A continuación, la Figura 11 detalla un esquema de visión artificial con cada una de sus etapas: desde la obtención de la imagen, su procesamiento y, finalmente, la activación de la acción (Alvear et al., 2016).

Figura 11

Proceso de la Visión Artificial



Nota. Elaboración Propia.

1.8. Fases de la Visión Artificial

La visión artificial tiene una variedad de componentes que se encargan de un proceso para que todos ellos interactúen y funcionen de una forma correcta, pero los principales son la iluminación, adquisición, procesamiento y actuadores para tener un mejor desempeño en prototipos que utilizan visión artificial.

1.8.1. Iluminación

Una adecuada iluminación es necesaria para que la visión artificial identifique los elementos, por lo que debe emplearse eficientemente con adecuadas fuentes de luz ubicadas estratégicamente. El propósito es que puedan visualizarse todas las características necesarias para que la cámara las capte correctamente y así se procesen sin dificultad ni errores (bcnvision, 2017).

1.8.2. Adquisición

Este proceso implica captar los rayos reflejados por los elementos y convertirlos en señales eléctricas con la finalidad de que puedan procesarse por la cámara, rayos X, equipos de ultrasonido o cualquier otro utilizado para este fin (Lozano & Orduz, 2018).

1.8.3. Procesamiento

Consiste en el instrumento usado para recabar los datos de una imagen digital, ya sea mediante un sistema externo ubicado en un computador o directamente a través del sistema de visión. Para ello se implementan algoritmos que controlan la imagen, capturan los datos necesarios, los inspeccionan y ejecutan una decisión (González, 2006).

1.8.4. Actuadores

Por lo general, la visión artificial suele ser parte de otros procesos y sistemas mayores que la contienen. Entonces, la información es enviada a robots, cintas transportadoras y cualquier otra clase de actuador encargado del procedimiento mecánico como tal, por ejemplo activar o desactivar puertas para permitir o no el paso de usuarios (Azuela & Humberto, 2013).

1.8.5. Microcontrolador

El microcontrolador consiste en un sistema que cuenta por lo menos con un microprocesador, memoria del software y de datos, así como puertos de entrada y salida; en ciertos casos hay elementos adicionales, por ejemplo, contadores. Este sistema puede implementarse a gran escala en discos duros o estar integrados en un chip (Dogán, 2008).

1.8.6. Pantalla

Una pantalla es el elemento e interfaz para controlar y visualizar información; constituye entonces la manera de interacción entre el usuario, el proceso y la máquina. La

imagen se muestra en píxeles (unidad de la imagen), y la resolución varía según la cantidad de píxeles (GCFGlobal, s.f.).

1.9. OpenCV

OpenCV es una librería multiplataforma inicialmente creada por Intel como apoyo para los primeros compiladores (Intel C++ y Microsoft Visual C++), pero hoy en día se la usa para procesar imágenes en torno a la visión artificial. Su lenguaje de programación es C optimizado y está escrito de tal modo que aprovecha los procesadores multinúcleo, cuenta con una estructura ágil de utilizar y el usuario es capaz de desarrollar rápidamente aplicaciones de visión artificial muy eficientes. Tiene incluso más de 400 funciones relacionadas a distintos aspectos: seguridad, calibración de cámaras, robótica, etc. (Bradski & Kaehler, 2008). A continuación, la Figura 12 expone el logo.

Figura 12

Logo de OpenCV



Nota. Tomado de “¿Como leer, visualizar y guardar una imagen en Python usando OpenCV? “, por OMES, 2019, (<http://www.sgse.eu/el-conteo-de-personas-de-hikvision-probado-en-sgse/>).

1.10. Técnicas de Detección de Objetos

Estos métodos son aplicados en prototipos para contar objetos determinados y así, poder realizar prototipos de una forma más eficiente, estos métodos facilitan el funcionamiento permitiendo detectar objetos determinados con mayor presión.

1.10.1. Sustracción de fondos

La sustracción de fondo es un paso fundamental en varias aplicaciones donde se requiere detección de movimiento e identificación y seguimiento de objetos en secuencias de imágenes dinámicas. Estas aplicaciones incluyen videovigilancia como: detección y captura de flujo de movimiento, interacción hombre computadora o codificación de video basada en contenido. Para detectar objetos en movimiento, cada fotograma de entrada se compara con el modelo de fondo obtenido del fotograma anterior para clasificar la escena en zonas de

fondo o estáticas y zonas dinámicas correspondientes al primer plano (Sakpal & Sabnis, 2018).

Existen tres modelos diferentes para aplicar este método de detección de objetos con sustracción de fondos en OpenCV, en la Figura 13, se puede observar cómo es la segmentación de fondo en estos modelos.

- **Modelo BackgroundSubtractorMOG:** es un algoritmo de segmentación de fondos en primer plano basado en una mezcla gaussiana.
- **Modelo BackgroundSubtractorMOG2:** es un algoritmo de segmentación de fondo de primer plano basado en una mezcla gaussiana, esta tiene una mejor adaptabilidad en diferentes escenarios con cambio de iluminación.
- **Modelo BackgroundSubtractorGMG:** este algoritmo combina la estimación estadística de la imagen de fondo y la segmentación bayesiana por píxel.

Figura 13

Modelos de Sustracción de fondos



Nota. Tomado de Sustracción de fondo | Python - OpenCV, por Gabriela Solano, 2020, OMES (<https://omes-va.com/sustraccion-de-fondo-opencvpython/#:~:text=La%20sustracci%C3%B3n%20de%20fondo%20oya%20que%20permanecer%C3%A1%20sin%20movimiento.>).

1.10.2. Detección de Bordes Canny

Canny (1986), propuso un método para la detección de bordes que ofrecían mejores resultados con una mayor complejidad computacional. Este método se basa en tres criterios principales, el criterio de detección que expresa el hecho de evitar la eliminación de bordes importantes así como no suministrar falsos bordes, el criterio de localización que establece que la distancia entre la posición real y la posición localizada para el borde debe ser minimizada, el criterio de respuesta única que establece la necesidad de que el detector

retorne un único punto por cada punto de borde verdadero, esto implica que el detector no debe encontrar múltiples píxeles de borde donde solo existe uno (Currás & Lola, 2012).

El algoritmo de Canny tiene tres pasos fundamentales a seguir para la ejecución de su funcionamiento como se observa en la Figura 14, estos pasos son los siguientes:

- Obtención del gradiente: se aplica el filtro gaussiano a la imagen original para suavizar la imagen y eliminar el ruido que exista en la misma, obteniendo dos imágenes.
- Supresión no máxima al resultado del gradiente: las dos imágenes obtenidas anteriormente sirven para generar una imagen con bordes adelgazadores, realizando la supresión no máxima de puntos de borde.
- Histéresis de umbral a la supresión no máxima: consiste en tomar la imagen anterior y orientar los puntos del borde de la imagen, así tomar dos umbrales de forma que uno sea más pequeño que el otro reduciendo los bordes falsos.

Figura 14

Ejemplo de detección de Bordes Canny



Nota. Tomado de Detección de Bordes, por Carlos Cauich, 2012, Academia (https://www.academia.edu/24190825/Detecci%C3%B3n_de_Bordes_Vi%20si%C3%B3n_Artificial_13_03_2012).

1.10.3. Detección de Haar-Cascade

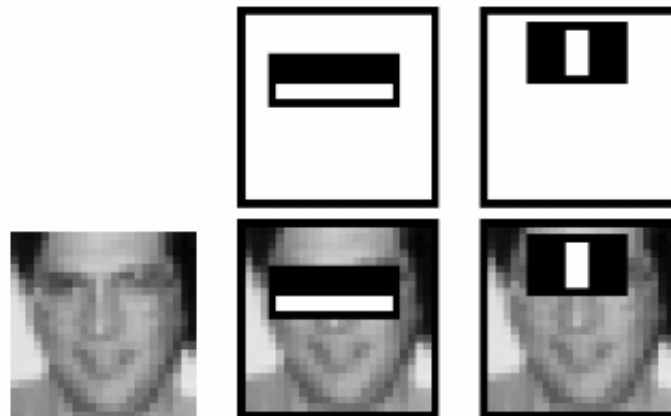
La documentación de OpenCV revela que la detección de objetos mediante clasificadores en cascada basados en características de Haar, es un método eficaz de

detección de objetos. Propuesto por Paula Viola y Michael Jones (2001), en su artículo Detección rápida de objetos usando una cascada. Donde se mostró que las cascadas funcionales se entrenan a partir de muchas imágenes positivas y negativas.

Para manejar el reconocimiento facial, el algoritmo primero necesita muchas imágenes positivas (estas imágenes contienen el objeto que queremos detectar) e imágenes negativas (imágenes de todo lo demás, que no contienen el objeto que queremos detectar) para entrenar al clasificador, así que tienes que extraer la función. Utiliza funciones de Haar, que representan núcleos de convolución, cada función es un valor único que se obtiene al restar la suma de píxeles debajo del cuadrado blanco de la suma de píxeles debajo del cuadrado negro como se puede ver en la Figura 15 (Nolasco Valenzuela, 2018).

Figura 15

Los Haar Features



Nota. Tomado de Detección de Rostros a la Vieja Usanza con Haar Cascades, por Jesús Martínez, 2011, DATASMARTS, (<https://datasmarts.net/es/deteccion-de-rostros-a-la-vieja-usanza-con-haar-cascades/>).

Básicamente, en lugar de aplicar todas las funciones a una sola ventana, el algoritmo las agrupa en etapas y las aplica una a la vez. Si una ventana falla en la primera etapa, se filtra inmediatamente y no se le aplican el resto de las características. De lo contrario, si la ventana pasa a la primera etapa, se debe aplicar la segunda etapa para continuar con el proceso. Se considera que un rostro se encuentra en la región analizada si la ventana en cuestión pasa por todos los niveles.

De los tres métodos para detectar objetos con visión artificial expuestos anteriormente, el método que se utilizó en el prototipo es el de Haar-Cascade, debido a que el prototipo se instaló en un autobús, donde existe mucho movimiento y puede tener sombras

que pueden ser mal interpretadas por los otros dos métodos, haciendo un conteo incorrecto de las personas.

Por esta razón el método Haar-Cascade es la mejor opción debido a que este método se encarga de contabilizar los rostros de las personas que ingresan y salen del autobús, permitiendo contar de forma correcta y sin contar otros objetos que no sean rostros de personas, facilitando de este modo los requerimientos de desarrollo del prototipo. En los siguientes capítulos se realizó el diseño e implementación del sistema de conteo de pajareros utilizando este método.

CAPÍTULO 2

DISEÑO DEL PROTOTIPO DE CONTEO DE PASAJEROS CON CONTROL DE AFORO

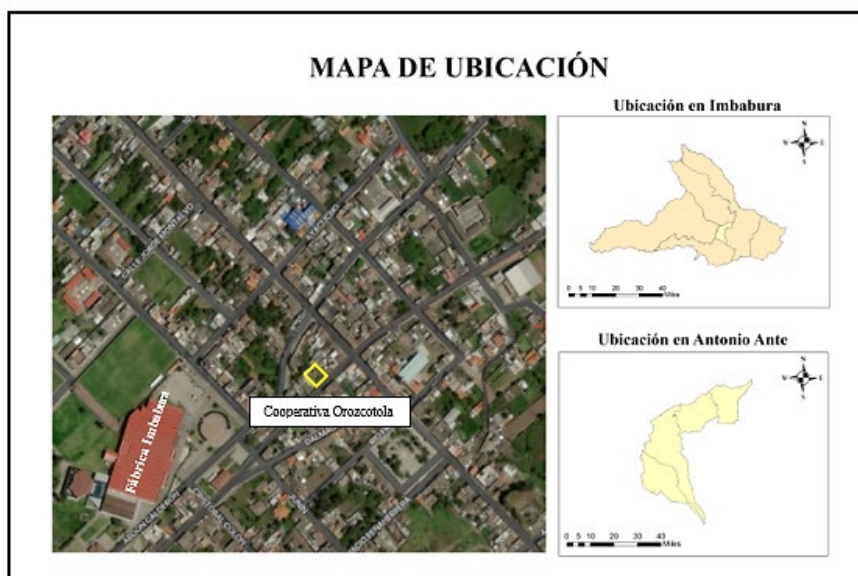
Con la descripción de los diferentes sistemas y métodos de conteo, se realizó el diseño eléctrico, electrónico, mecánico y de software del prototipo, en el cual, se analizan y desarrollan los requerimientos correspondientes para la posterior construcción de un sistema de conteo de pasajeros con control de aforo, que posteriormente se implementó en un autobús de la Cooperativa Orozcotola del Cantón Antonio Ante.

2.1. Descripción del Lugar

La implementación del prototipo se realizó en el cantón Antonio Ante, en la parroquia Andrade Marín, en un autobús de la Cooperativa Orozcotola, que se encuentra en las calles 21 de noviembre y General Enríquez. En la Figura 16 se puede observar la ubicación exacta de esta Cooperativa.

Figura 16

Ubicación de la Cooperativa Orozcotola



Nota. Elaboración Propia.

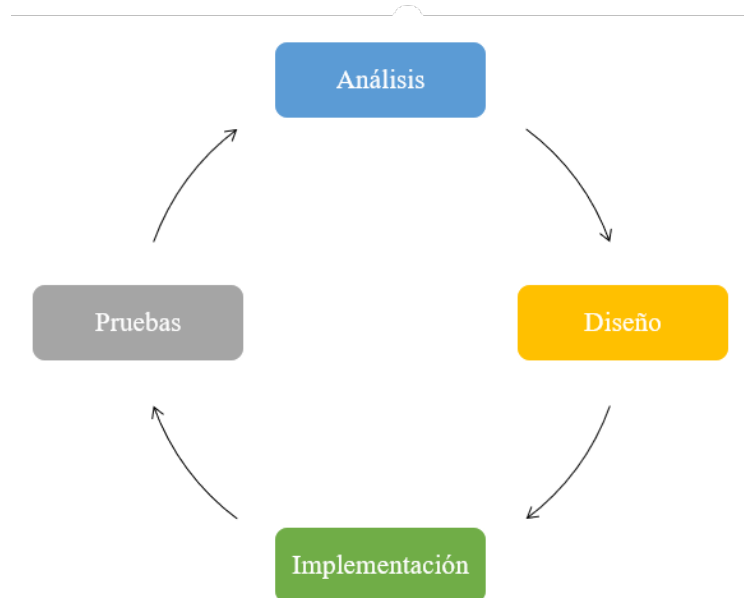
2.2. Metodología

Para el desarrollo del trabajo se utilizó la metodología en espiral desarrollada por Barry W. Boehm (1986), que se puede observar en la Figura 17. Esta metodología sirve para

realizar ajustes conforme se esté desarrollando el prototipo con el fin de seleccionar los componentes más adecuados; de esta manera, el diseño se adaptará a los requerimientos y necesidades para su correcto funcionamiento.

Figura 17

Esquema del desarrollo de la metodología en espiral



Nota. Adaptado de A Spiral model of software development and enhancement, por Barry W. Boehm, 1988, IEEE (<https://ieeexplore.ieee.org/abstract/document/59/authors#authors>)

2.2.1. Análisis

Para el desarrollo del prototipo se tomó en cuenta los siguientes aspectos principales para el diseño del sistema de conteo de pasajeros con control de aforo como son:

- Estudio de los diferentes sistemas de conteo de personas.
- Descripción de los diferentes métodos de detección de objetos.
- Facilidad de adquisición de los equipos y componentes necesarios para la elaboración del prototipo.
- Tener un conteo más exacto de las personas para poder controlar el aforo adecuadamente.
- Lograr un conteo preciso dentro de un autobús tomando en cuenta los factores de iluminación.

2.2.2. Diseño

Para la elaboración del prototipo se toma en cuenta el diseño eléctrico, electrónico, mecánico y de software, seleccionando los materiales óptimos para el correcto desempeño del prototipo, tomando en cuenta los siguientes puntos:

- Se realizó una investigación de los principales métodos de conteo de personas, que utilizan visión artificial para tener una mayor noción para el diseño y construcción del prototipo.
- Para el diseño eléctrico se toma en cuenta, la capacidad de la batería del autobús para poder adaptarnos a esa cantidad, para que funcione correctamente el prototipo.
- Para el diseño mecánico se debe conocer cómo funciona el mecanismo de la puerta del autobús, para proceder a manipular ese mecanismo y poder controlar el aforo del autobús.
- Para el diseño electrónico, los dispositivos o elementos deben ser lo más adecuados para implementar en la unidad de transporte
- Para el diseño del software se desarrolló un código de programación que permite realizar el monitoreo de las personas que están en el autobús mediante 2 cámaras, también para poder indicar en las pantallas la capacidad disponible del autobús.

2.2.3. Implementación

Para la implementación del prototipo es necesario saber las dimensiones del autobús para poder colocar los diferentes equipos materiales y tener un funcionamiento más óptimo debido a eso tenemos que seguir los siguientes pasos:

- Se realizará la implementación en el autobús con materiales resistentes para que no haya ningún fallo del prototipo.
- Se implementan los sistemas eléctricos, electrónicos y mecánicos para realizar las pruebas respectivas de funcionamiento en el autobús.
- En la minicomputadora debe estar instalado todos los programas y codificaciones para que interactúe con todos los equipos colocados en el autobús.

2.2.4. Pruebas

Una vez que todos los sistemas estén implementados en el autobús, se realizó las pruebas respectivas de funcionamiento, revisando que todo funcione correctamente.

- El prototipo tiene dos cámaras una en la puerta de ingreso y otra cámara en la puerta de salida, para contar a las personas que entran y salen del autobús.
- El monitoreo de las personas se muestra en la pantalla táctil al conductor y en una matriz leds a los usuarios que vayan a ingresar al autobús.
- Cuando el autobús cumpla con su aforo la puerta de ingreso no se abre para que el conductor no suba una cantidad excesiva de pasajeros, excepto en caso de emergencia.

Después de haber hecho el análisis metodológico, a continuación, se indicó cuáles son los equipo, materiales y elementos utilizados para el desarrollo del prototipo, se realizó las respectivas comparativas entre los distintos equipos que existen en el mercado y se escogió al más adecuado para que el prototipo funcione de la mejor forma posible.

2.3. Diseño Electrónico

Acorde a la Figura 11 del Capítulo 1 se procedió con la selección de los equipos y elementos para el diseño electrónico del sistema de conteo de personas con control de aforo, teniendo en cuenta las características y especificaciones de cada uno de estos, para que se adapten al autobús.

2.3.1. Microprocesadores

Un Sistema en chip o SoC es un circuito integrado que se ocupa de la comunicación entre la CPU, la RAM, el almacenamiento y otros periféricos que integra algunos componentes como uno a cuatro chips y cuentan con controladores para periféricos de uso común, como el teclado, el mouse o el monitor, en un solo chip de silicio, junto con un procesador, el sistema integrado en chip suele contener una GPU (procesador de gráficos), memoria, controlador USB, circuitos de administración de energía y radios inalámbricas (Alonso, Hard Zone, 2020). Es un dispositivo que se caracteriza por ser utilizado para desarrollar prototipos en el área de la informática y electrónica, en la Tabla 1 se puede observar los microprocesadores del mercado más utilizados para proyecto con visión artificial, para la selección del microprocesador se muestra las diferentes características que existen entre ellos.

Tabla 1

Modelos de microprocesadores del mercado

pcDuino	Raspberry Pi 3 B+	Raspberry Pi 4
---------	-------------------	----------------

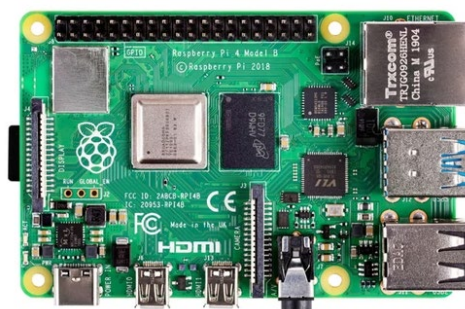
Dimensiones	125 x 52 mm	85 x 53 mm	85 x 53 mm
CPU	1 GHz ARM Cortex A7 Dual Core	ARM Quad-core 64 bits a 1.4 GHz (A53)	ARM Quad-core 64 bits a 1.5 GHz (A72)
RAM	1 GB	1 GB	1/2/4/8 GB
Conectividad	Ethernet 10/100/100	Ethernet 10/100/100 Bluetooth 4.2 Wifi 2.4 / 5 GHz	Ethernet 10/100/100 Bluetooth 5.0 Wifi 2.4 / 5 GHz
Puertos	1 HDMI, 2 USB 2.0 + 1 mini-USB OTG	4 x USB 2.0, 1 HDMI, 3.5 mm audio	2 x USB 3.0, 2 x USB 2.0, 3.5 mm audio
Sistema Operativo	Linux, Ubuntu, Android	Linux	Linux

Nota. Fuente: (PcDuino, 2022), (Raspberry, 2022), (Alonso, 2020).

De acuerdo con las características mostradas en la Tabla 1, se seleccionó el Raspberry Pi 4 de 8 RAM, uno para la puerta de entrada y otro para puerta de salida, debido a la capacidad del procesamiento y la velocidad de RAM necesarias para la implementación de la capa de IA. En la Figura 18 se puede observar una imagen de este microprocesador.

Figura 18

Raspberry Pi 4



Nota. Tomado de Raspberry Pi 4, por Raspberry Pi, 2022, Raspberry Pi, (<https://www.raspberrypi.com/productos/raspberry-pi-4-model-b/>)

2.3.2. Microcontrolador

Se debe utilizar un microcontrolador para los matrices leds que se implementó en el prototipo, debido a que este será el encargado de interpretar la cantidad de personas que

existan en el autobús. A continuación, se muestra una tabla con algunos modelos de microcontroladores que existen en el mercado.

Tabla 2

Modelos de microcontroladores del mercado

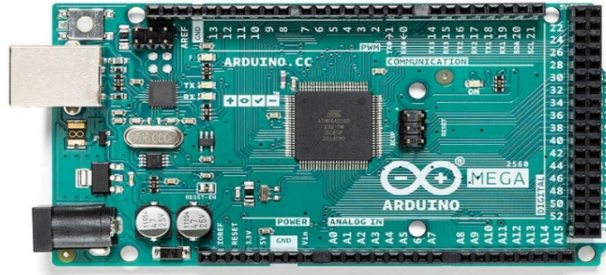
	Arduino Uno	Arduino Mega	ESP32 WROOM
Dimensiones	68.6 x 53.4 mm	101.52 x 53.3 mm	54.7 x 71.5 mm
Microcontrolador	Atmega328P	Atmega2560	ESP32
Velocidad de reloj	16 MHz	16 MHz	240 MHz
Voltaje de trabajo	5 V	5 V	3,3 V
Voltaje de entrada	7,5 - 12 V	7,5 - 12 V	3 – 3.6 V
Pines Digitales	14 – 6 PWM	54 – 15 PWM	24
Pines Analógicos	6	16	12
Memoria	32 KB Flash y 2 KB Ram y 1 KB Eeprom	256 KB Flash y 8 KB Ram y 4 KB Eeprom	32 KB

Nota. Adaptado de (Arduino, 2022), (Guerrero, 2014), (330ohms, 2016).

De acuerdo con las características mostradas en la Tabla 2, y cómo debe adaptarse a un panel de matriz LED RGB de 64 x 32, se recomienda utilizar un microcontrolador con gran RAM y alta velocidad para controlar estos leds, por este motivo se decidió trabajar con el Arduino Mega 2560 que se observa en la Figura 19, ya que sus especificaciones son adecuadas para el prototipo, además este microprocesador tiene un voltaje de trabajo de 5 V, que necesitamos para implementarlo en el autobús.

Figura 19

Arduino Mega 2560



Nota. Tomado de Arduino Mega 2560 Rev3, por Arduino, 2022, Arduino, (<https://store.arduino.cc/products/arduino-mega-2560-rev3?queryID=undefined>)

2.3.3. Matriz de LEDs

Este módulo de matriz de leds debe ser de un tamaño considerable para que las personas puedan leer lo que dice en la parte delantera del autobús, se realizó una comparativa de los diferentes matrices leds que existe en el mercado.

Tabla 3

Modelos de las Matrices de LEDs del mercado

	Modulo MAX7219	Matriz p4 32x32 RGB	Matriz P4 64x32 RGB
Dimensiones	32 x 128 x 15 mm	960 x 960 x 150 mm	1940 x 960 x 150 mm
Voltaje de trabajo	5 V	5 V	5 V
Niveles de brillo	16 candela por m^2	2200 candela por m^2	2200 candela por m^2
pixeles	8x8	32x32	64x32
peso	100 g	124,74 g	249,48 g
Color	Rojo	RGB	RGB
Distancia de visualización	1 m	4 m – 30 m	3 m – 30 m

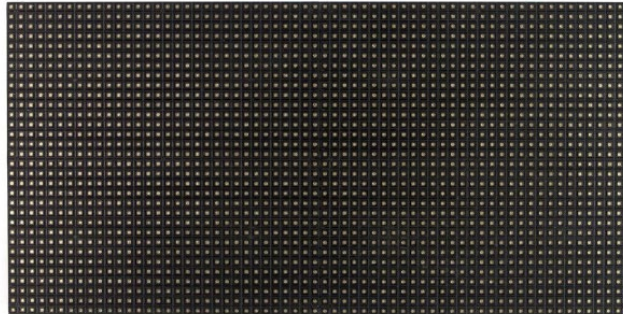
Nota. Adaptado de (Maxim, 2022), (AZERONA, 2022).

Debido a que la pantalla debe estar colocada en la parte delantera del autobús para que las personas puedan visualizar de mejor manera la cantidad de lugares disponibles, se eligió la Matriz P4 64 x 32 como se puede observar en la Figura 20, así las personas no

tendrán ningún problema al momento de observar la Matriz desde donde ellos estén parados. Además, en la otra matriz se pueden observar los lugares por donde hace el recorrido el autobús para un mejor servicio para los usuarios.

Figura 20

Matriz led 64X32 RGB



Nota. Tomado de Panel de matriz LED RGB, por AZERONA, 2022, PIMORONI, (<https://shop.pimoroni.com/products/rgb-led-matrix-panel?variant=19321740999>)

2.3.4. Cámara

La cámara para implementar en el prototipo debe ser compatible con la Raspberry Pi 4, debido a eso se realizó una búsqueda de cámaras compatible que tengan una resolución adecuada para que cuente de una forma precisa al momento de reconocer a las personas en el autobús. A continuación, en la Tabla 4 están las características de las cámaras compatibles con las Raspberry Pi 4.

Tabla 4

Modelos de Cámaras Compatibles con Raspberry Pi 4

	Cámara V2 IMX219	Cámara OV5647	Cámara logitech C270
Dimensiones	25 x 23 x 9 mm	25 x 24 x 19 mm	72,91 x 31,91 x 66,64 mm
Captura de video	1080p30 / 720p60	1080p	720p30
Voltaje de operación	1 V - 5V	3.3 V	5 V
Megapíxel	8 mp	5 mp	3 mp

Campo visual	62,2 °	70 °	55 °
Longitud de cable	0,15 m	0,15 m	1, 5 m
Peso	27 g	35 g	75 g
Tipo de Conexión	Puerto CSI	Puerto CSI	USB

Nota. Adaptado de (Maxim, 2022), (AZERONA, 2022).

Se decidió trabajar con la cámara logitech C270, que se observa en la Figura 21, debido a que se conecta mediante un puerto USB, y las otras 2 cámaras tiene un puerto CSI que puede soltarse al momento que el autobús este en movimiento mientras que el USB es más seguro y no se desconecta con facilidad, además esta cámara tiene una longitud del cable de 1,5 m, esto nos permite colocarle la cámara en una posición adecuada, brindando un conteo más exacto de las personas que utilizan el autobús.

Figura 21

Cámara Logitech C270



Nota. Tomado de C270 HD WEBCAM, por logitech, 2022, PIMORONI, (<https://www.logitech.com/es-roam/products/webcams/c270-hd-webcam.960-000947.html>)

2.3.5. Método Haar Cascade

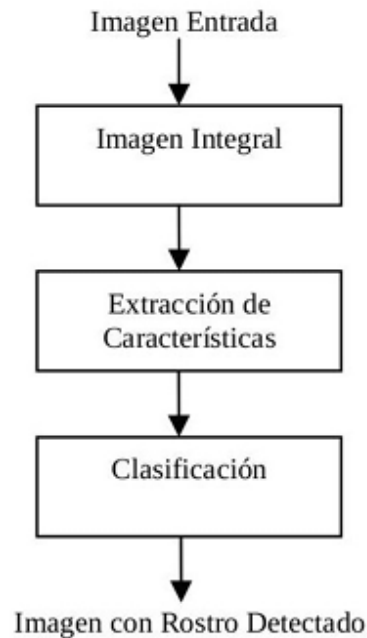
Como se manifiesta en el primer capítulo, el clasificador a utilizar es el Haar Cascade, este clasificador recorrerá toda la imagen compuesta de píxeles, sumando los píxeles positivos y negativos respectivamente, luego se realiza una diferencia de que se denomina valor de la característica. Este valor se puede representar a través de la siguiente ecuación:

$$H(x, y) = \sum_p I(x, y) - \sum_n I(x, y)$$

Donde: $I(x, y)$ = Imagen a evaluar, p y n = región positiva y negativa, $H(x, y)$ = valor de la característica Haar en el punto x y y . En la Figura 22 se observa el diagrama de bloques de como realiza la detección de rostros este método.

Figura 22

Diagrama de bloques de la detección Haar-Cascade



Nota. Tomado de Análisis e Implementación del Algoritmo de Detección Facial de Viola-Jones (p.8), por Rodríguez Karen, 2017, Repositorio Universitario, Universidad Técnica del Norte.

Primero se necesita la imagen de entrada, esta imagen es donde se encuentran los rostros de las personas que suben y bajan del autobús, el método Haar-Cascade nos dirá si existe un rostro, permitiendo el monitoreo del aforo en el autobús.

2.3.5.1. Imagen Integral

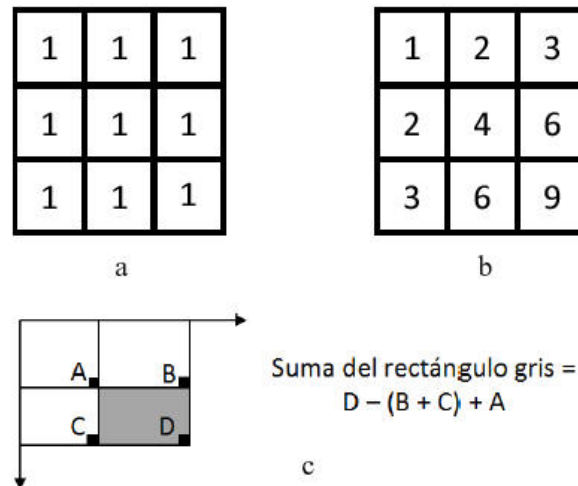
La imagen integral es la suma de píxeles de la parte superior izquierda de la posición (x, y) , la suma de todos los píxeles de cualquier rectángulo en la imagen (Rodríguez, 2017). En la Figura 23 se indica un ejemplo del cálculo de la imagen integral, y a continuación se observa la ecuación para el cálculo de la imagen integral y su respectivo análisis:

$$II(x, y) = \sum_{x' \leq x, y' \leq y} I m(x', y') \quad (2.2)$$

Donde $I(x,y)$ es la imagen integral y $Im(x,y)$ es la imagen original, además debemos tener en cuenta que, para el cálculo de la imagen integral, la imagen original debe estar con escalas de grises.

Figura 23

Cálculo de la integral



Nota. En la el grafico se interpreta como: (a) imagen original, (b) imagen integral y (c) Calculo de la suma en una imagen integral. Tomado de Implementación Híbrida Hardware Software del Algoritmo de Detección de Rostros de Viola-Jones sobre FPGA, por Hernández Ernesto, 2012, SciELO.

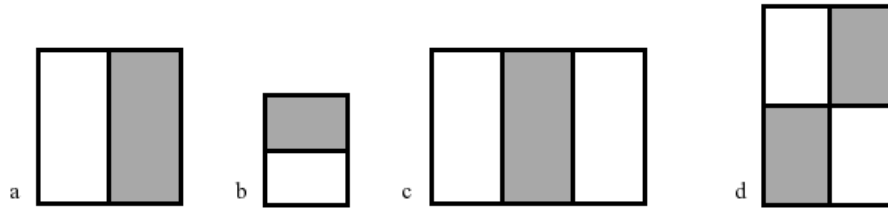
2.3.5.2. Extracción de Características

En imágenes las características de cada objeto se extraen al aplicar ciertas funciones que permitan la representación y descripción de los objetos de interés de la imagen (Rodríguez, 2017). La extracción de caracteres es el reconocimiento de patrones, que permiten encontrar distintos rasgos que pueden ayudar para la detección del rostro, esto se lo realiza aplicando filtros a la imagen.

En la Figura 24 se puede evidenciar algunos ejemplos de los rasgos utilizados para la detección de rostros.

Figura 24

Ejemplos de rasgos de detección de rostros



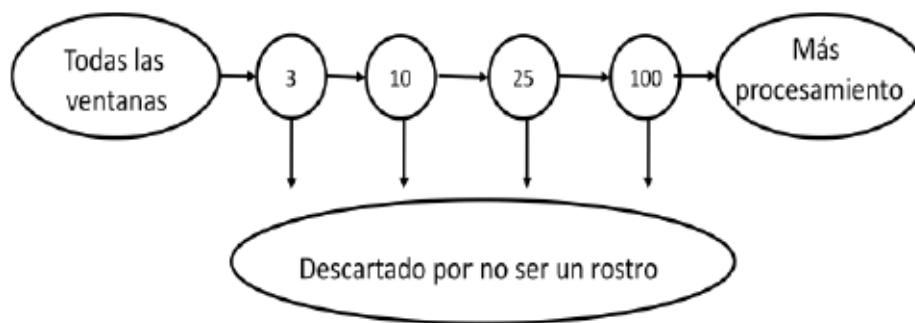
Nota. En la el grafico se indica que, (a) imagen original, (b) imagen integral y (c) Calculo de la suma en una imagen integral. Tomado de *Implementación Híbrida Hardware Software del Algoritmo de Detección de Rostros de Viola-Jones sobre FPGA*, por Hernández Ernesto, 2012, SciELO.

2.3.5.3. Clasificador

Esta etapa dentro del algoritmo de detección se encarga de asignar un conjunto de características dado a una clase con la que se encuentra una mayor similitud, de acuerdo con un modelo inducido durante el entrenamiento (Rodríguez, 2017). La construcción de los clasificadores para realizar la detección de rostros está basada en la selección, utilizando el algoritmo de entrenamiento denominado AdaBoost (Hernández, Cabrera, & Sánchez, 2012). En la Figura 25 se puede evidenciar el clasificador en cascada, para la detección de rostros.

Figura 25

Esquema del clasificador en cascada



Nota. Tomado de *Implementación Híbrida Hardware Software del Algoritmo de Detección de Rostros de Viola-Jones sobre FPGA*, por Hernández Ernesto, 2012, SciELO.

2.3.6. Pantalla HMI

Son pantallas que proporciona una interfaz de control y visualización de datos entre un humano, maquina y un procesador (Lozano R. , 2021). Estas pantallas pueden remplazar a las LCD, además están tienen la particularidad de que son Touch permitiendo tener mayores formas de aplicación en diversos proyectos, en la Tabla 5, están las pantallas HMI más utilizadas.

Tabla 5*Pantallas HMI*

	NX3224T024	NX4832T035	NX8048T050
Tamaño	2,4	3,5	5,0
Dimensiones	74.4 x 42.9 mm	73.44 x 48.96 mm	133.5 x 84 mm
Resolución	320 x 240	480 x 320	800 x 480
Voltaje de operación	4.75 V (mín.), 5 V(típico), 7V(máx.)	4.75 V (mín.), 5 V(típico), 7V(máx.)	4.75 V (mín.), 5 V(típico), 7V(máx.)
Corriente de operación	100 % de brillo 90mA Modo Sleep: 15 mA	100 % de brillo 145mA Modo Sleep: 15 mA	100 % de brillo 410mA Modo Sleep: 15 mA
Memoria Flash	4 MB	16 MB	16 MB
Memoria RAM	3584 byte	3584 byte	3584 byte
Interfaz	4 pines (+5, TX, RX, GND)	4 pines (+5, TX, RX, GND)	4 pines (+5, TX, RX, GND)
Puerto SD	Si, Formato FAT32, Micro SD 32G máximo	Si, Formato FAT32, Micro SD 32G máximo	Si, Formato FAT32, Micro SD 32G máximo
Temperatura de Trabajo	-20 ° C a 70 °C	-20 ° C a 70 °C	-20 ° C a 70 °C
Fuente de alimentación	5 V, 500mA, DC	5 V, 1A, DC	5 V, 1A, DC

Nota. Adaptado de (UNIT Electronics, s.f.)

La pantalla que se utilizó fue la Nextion (NX8048T050) como se observa en la Figura 26. Debido a que es de un tamaño de 5 pulgadas y resolución 800 x 480, para que el conductor del autobús tenga la facilidad de observar la cantidad de lugares disponibles,

también se obtiene un registro de las personas que ingresan a la unidad por cada jornada de trabajo.

Figura 26

Pantalla Nextion de 5 pulgadas



Nota. Tomado de NX8048T050 Nextion Display 5", por NEXTION, 2018, UNIT ELECTRONICS, (<https://uelectronics.com/producto/nx8048t050-nextion-display-5-pulgadas/>)

2.3.7. Sensor de distancia

Los sensores de distancia son sensores con diferentes tecnologías para realizar la medida de posición lineal, medida de distancia lineal o medida de desplazamiento lineal (Sensing, s.f.). Con el fin de que las cámaras no estén encendidas en todo el tiempo, sino solo cuando entren los usuarios al autobús, se utilizó dos sensores de distancia. El sensor que se utilizó fue el HC-SR04 como se puede observar en la Figura 27, debido que es un sensor de distancia que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm, destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio (NAYLAMP MECHATRONICS, s.f.).

Figura 27

Sensor ultrasonido HC-SR04



Nota. Tomado de Sensor Ultrasonido HC-SR04, por ELECTRONICA MADE, 2022, MADE, (<https://electronicamade.com/sensor-ultrasonico/>)

2.3.8. Modulo Relé

Este módulo es el encargado del control de la puerta permite que se abra o se quede cerrada y así impidiendo el ingreso de personas en el autobús. El módulo posee 1 Relays de alta calidad como se observa en la Figura 28, capaz de manejar cargas de hasta 250V/10A., cada canal posee aislamiento eléctrico por medio de un optoacoplador y un led indicador de estado.

Figura 28

Modulo relé de un canal



Nota. Tomado de Modulo relé 1 canal, por Oiyagay, 2018, AV Electronics, (<https://avelectronics.cc/producto/modulo-rele-1-canal/>)

2.3.9. Electroválvula

Una válvula eléctrica es una válvula electromecánica diseñada para controlar el flujo de fluido a través de un canal o tubería. La válvula es accionada por una bobina de solenoide. Suele tener sólo dos posiciones: abierto y cerrado, o todos y ninguno. Las válvulas solenoides como se puede observar en la Figura 29, se utilizan en muchas aplicaciones para controlar el flujo de todo tipo de fluidos.

Figura 29

Electroválvula de 12V



Nota. Tomado de Electroválvula, por OKYBOLIVIA, 2023, (<https://www.okybolivia.com/product/electrovalvula-de-3-4-pulgada-de-12v/>)

2.4. Diseño Eléctrico

Se aspira poder alimentar el prototipo con la batería del autobús, mediante un reductor de voltaje, para que suministre voltaje a los diferentes equipos como la Raspberry Pi 4, Arduino Mega, Pantalla Nextion y las Matrices de Leds, además se realizó el cálculo de un batería que alimente todo el prototipo en caso no se pueda conectar a la batería del autobús.

2.4.1. Batería del Autobús

Una batería de autobús, almacena electricidad para cederla en el momento que los consumos la demanden, transforma la energía eléctrica que recibe en energía química a partir de la reacción que se produce cuando dos materiales diferentes, como los de las placas positiva y negativa, se tiene 2 voltios por celda, con un total de 12 a 24 voltios correspondiente a la batería de un autobús (VARTA, s.f.).

2.4.2. Regulador Reductor

Este dispositivo permite cargar teléfonos o Tablet, desde un carro, debido a que necesitamos que la alimentación de las Raspberry PI 4 sea contante y no haya subida ni bajas de corriente, se trabajó con este regular por la facilidad de instalación en el autobús y sin la necesidad de colocar más elementos al prototipo, como se muestra en la Figura 30, este cargador para carro cuenta con un voltaje de salida de 5V y con una corriente que puede variar de 1.5 A hasta los 4.8 A, estas características permitieron que las dos Raspberry PI 4 y las dos matrices leds funcionen de la mejor manera.

Figura 30

Cargador de carro



Nota. Tomado de Cargador de coche rápido USB y tipo C con cable de carga (4,8 A), por Vidvie, 2022, Tanzilaad, (<https://www.tanzilaad.com/product/vidvie-usb-type-c-fast-car-charger-with-charging-cable-4-8a/>)

2.4.3. Toma de mechero

Es un conector eléctrico utilizado en automóviles, permiten prender cigallos y conectar dispositivos como radios, cargadores, divisores entre otros. Este adaptador debe tener una entrada de alimentación de 12 - 24 V dependiendo de la batería del automóvil, como se puede observar en la Figura 31, existen diferentes modelos, divisores y formas de esta toma de mechero.

Figura 31

Toma de mechero



Nota. Tomado de Divisor de encendedor de cigallos para automóvil de 3 enchufes (12 – 24 V) / Cargador USB (3 puertos), por Gadgets 4 Frikis, 2022, Cadgets4geeks, (<https://www.gadgets4geeks.com.au/3-socket-cigarette-lighter-port-splitter-usb-car-charger>)

2.5. Software

Para el desarrollo del prototipo se utilizó algunos programas y lenguajes de programación para realizar la codificación para el funcionamiento, como son los siguientes: Python, OpenCV, Arduino, Visual Studio Code y Nextion Editor. Estos programas permitirán elaborar un sistema de conteo de personas con control de aforo.

2.5.1. Lenguaje de Programación

Python es un lenguaje de programación, que posee una licencia de código abierto, gratuita para sistemas con Windows, Linux, Mac. Es un lenguaje interpretado, lo que significa que no necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad de ejecución (Alvarez, 2003). Debido a esto se decidió trabajar con Python, ya que es un lenguaje fácil de

comprender, además es uno de los lenguajes más utilizados en el mundo considerando que vamos a ocupar las librerías de OpenCV siendo más factible y más práctico trabajar con este lenguaje de programación.

2.5.2. IDE Visual Studio Code

Para la programación del microprocesador Raspberry Pi 4, se hará uso de la plataforma de desarrollo integrado (IDE) Visual Studio Code, este editor de código fue desarrollado por Microsoft, es un software libre, está disponible para Windows, Linux y MacOS, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación (Flores, 2022).

2.5.3. IDE Arduino

Para la programación del microcontrolador Arduino Mega, se hizo uso de la plataforma de desarrollo integrado (IDE) Arduino, que es de software libre, también es compatible con los sistemas operativos Windows, Mac OS y Linux, posee adaptabilidad a una infinidad de aplicaciones, su lenguaje de programación se denomina Arduino Programming Language, que está basado en Lenguaje C y algunas funciones de C++ (Fernandez, 2022).

2.5.4. IDE Nextion Editor

Nextion Editor es un software gratuito de desarrollo de GUI de interfaz hombre máquina (HMI) para todos los productos de la serie Nextion, es más fácil realizar funciones de animación, menú desplegable y diapositiva táctil de texto (NEXTION, s.f.). El editor Nextion tiene componentes masivos tales como botones, texto, barra de progreso, slider, panel de instrumentos, etc. para enriquecer el diseño de su interfaz (Torres, s.f.).

2.6. Resultados

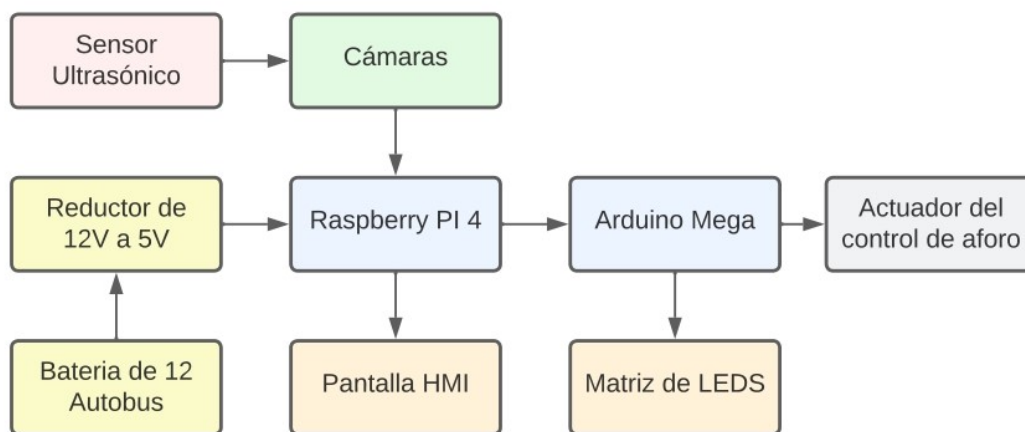
Una vez seleccionados los materiales y lenguaje de programación anteriormente se realizó el diseño del prototipo. A continuación, se muestra el diagrama de bloques del contador de pasajeros con control de aforo y el esquema de conexión con todos los dispositivos utilizados.

2.6.1. Diagrama de bloques y descripción funcional

Como se puede observar en la Figure 32, se muestra el diagrama de bloques del prototipo del sistema de conteo de pasajeros con control de aforo, permitiendo interpretar de manera grafica como se elaboró el prototipo.

Figura 32

Diagrama Funcional de Bloques del Sistema



Nota. Elaboración Propia.

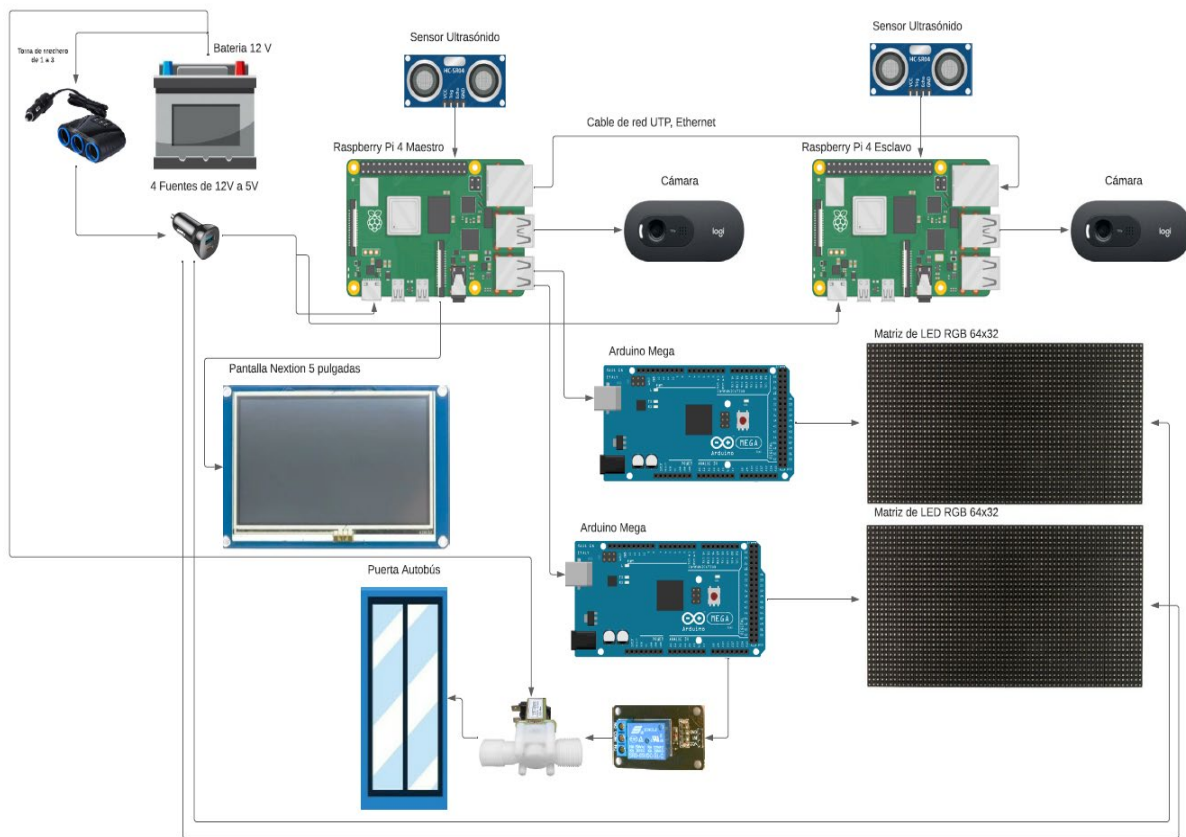
El sistema de conteo de pasajeros con control de aforo se conectó a la batería del autobús, mediante un regulador reductor, además cuenta con una batería de respaldo en caso de no poder conectarse a la batería del autobús, esto permitió que los elementos como la Raspberry Pi 4, Arduino Mega, Pantalla Nextion y a las Matriz de Leds, tengan la energía suficiente para funcionar correctamente. El prototipo trabajo de la siguiente manera, cuando una persona ingresa por la puerta de entrada, activa el sensor ultrasonido, el cual prende la cámara delantera del autobús para realizar el conteo de la personas, una vez realizado el conteo mostrará en la pantalla Nextion la cantidad de aforo en el autobús y en la matriz de leds la cantidad de lugares disponibles en la unidad de transporte, esto permitió tanto al conductor como a los usuarios, que se van a subir a este medio de transporte conocer el aforo del autobús, antes de entrar al mismo, cuando el aforo llegue a su máxima capacidad, la puerta de entrada no se abrirá, impidiendo el ingreso a las personas que van a subirse al autobús.

2.6.2. Esquema de conexión

A continuación, se observa en la Figura 33 el esquema de conexión del conteo de pasajeros con control de aforo.

Figura 33

Esquema de Conexión



Nota. Elaboración Propia.

Primero se energizó el prototipo, luego se conectó a la batería del autobús el regulador reductor LM2596S, para tener una salida de voltaje de 5 V para conectar a las dos Raspberry Pi 4, y estas estarán conectados los demás elementos.

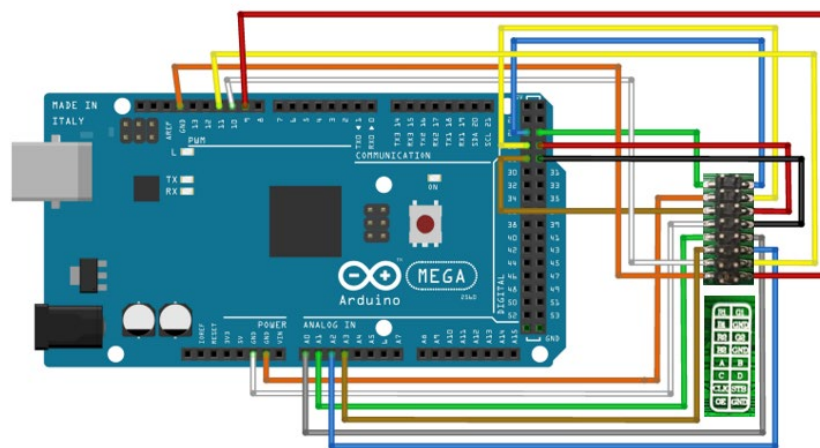
Las dos Raspberry Pi 4 están conectadas mediante un cable cruzado Ethernet permitiendo una comunicación más rápida, la primera Raspberry Pi 4 está conectada a un sensor ultrasonido en el pin 18 está conectado al pin ECHO para recibir la señal ultrasónica y en el pin 16 está conectado con el pin TRIG para recibir el rebote de la señal ultrasónica, además estará conectados los pines 4 y 6 para alimentar con 5V al sensor. También está conectada la Pantalla Nextion de 5 pulgadas, al pin 8 está conectado al pin TX para la salida de la interfaz en serie y al pin 10 está conectado al pin RX para la entra de la interfaz en serie. Los pines 2 y 9 están conectados para alimentar con 5 V a la pantalla. Por último, está conectado un puerto USB a una cámara, otro puerto USB a un Arduino Mega y otro puerto USB al otro Arduino Mega.

El segundo Raspberry Pi 4, está conectado un sensor ultrasonido en el pin 18 está conectado al pin ECHO para recibir la señal ultrasónica y en el pin 16 está conectado con el pin TRIG para recibir el rebote de la señal ultrasónica, además esta conectados los pines 4 y 6 para alimentar con 5 V al sensor. Por último, está conectado mediante el puerto USB a la otra cámara.

Los dos Arduino Mega están conectados a la primera Raspberry Pi 4 por los puertos USB, para poder manejar las Matriz de Leds 64x32 debemos realizar la conexión como se muestra en la Figura 34.

Figura 34

Conexión Arduino Mega y Matriz de Leds 64x32



Nota. Elaboración Propia.

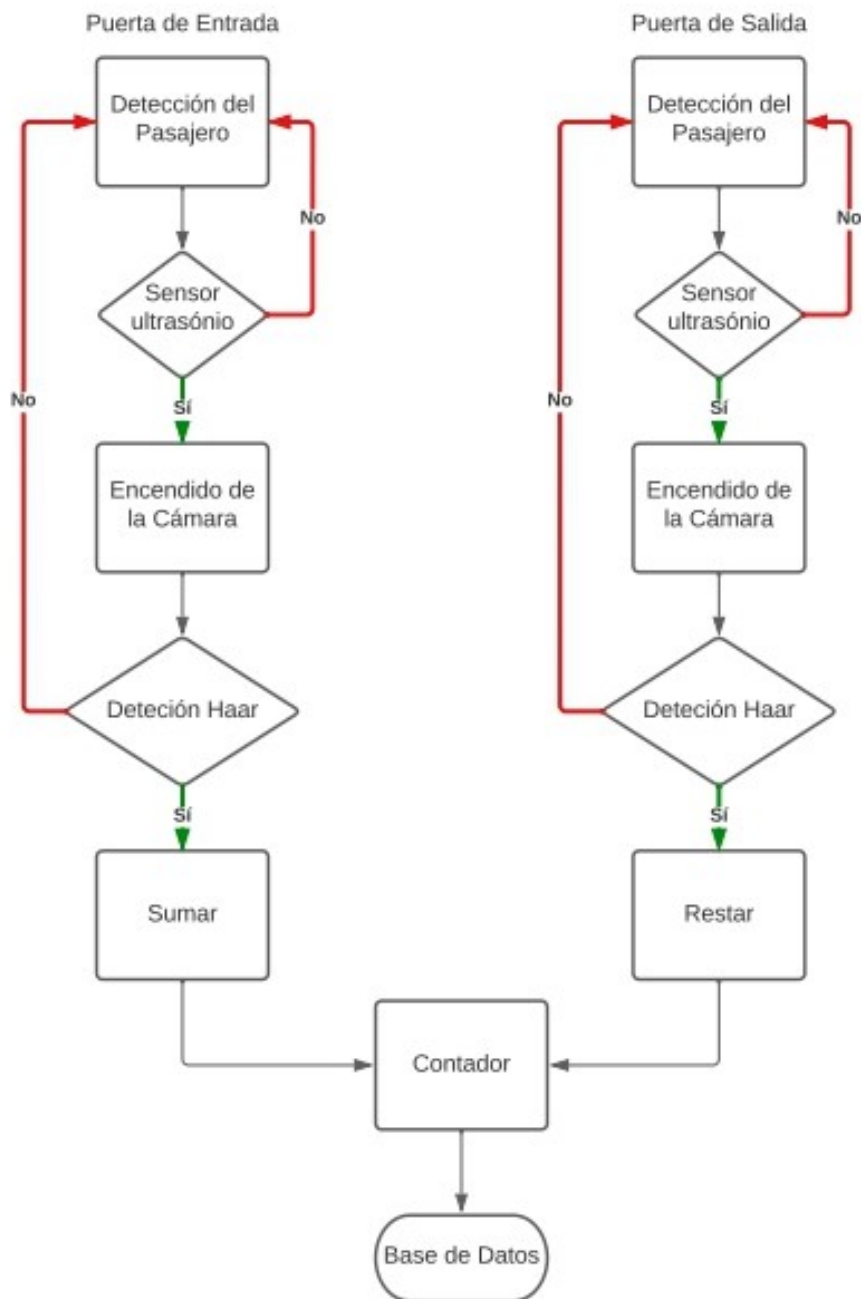
También conectamos a un Arduino Mega el Módulo relé, mediante el pin 7 del Arduino Mega lo conectamos al pin IN del Módulo relé, además estará conectados los pines 5 V y GND para alimentar con 5V al relé. Por último, conectamos las salidas del relé al actuador para poder tener el control del aforo de la puerta y así solo permitir que la puerta se habrá si es que hay lugares disponibles en el autobús, caso contrario la puerta permanece cerrada hasta que haya lugares disponibles nuevamente.

2.6.3. Programación de las Raspberry Pi 4

El Raspberry Pi 4 tiene la capacidad de determinar cuando una persona entra y sale del autobús mediante sus cámaras y así poder realizar el conteo de los pasajeros como se puede observar el diagrama de flujo en la Figura 35.

Figura 35

Flujograma de Programación de las Raspberry Pi 4



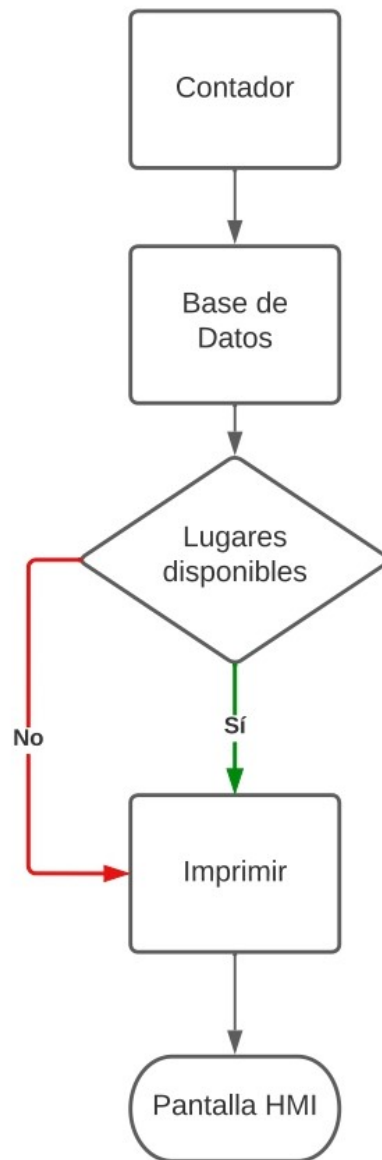
Nota. Elaboración Propia.

2.6.4. Programación de la Pantalla HMI

Para la impresión en la pantalla HMI, el contador se encarga de colocar en una base de datos la cantidad de aforo que hay en el autobús y la pantalla toma este valor para imprimirlo en la pantalla HMI para que el conductor pueda observó el aforo de la unidad como se puede observar el diagrama de flujo en la Figura 36.

Figura 36

Flujograma de la Programación de la Pantalla HMI



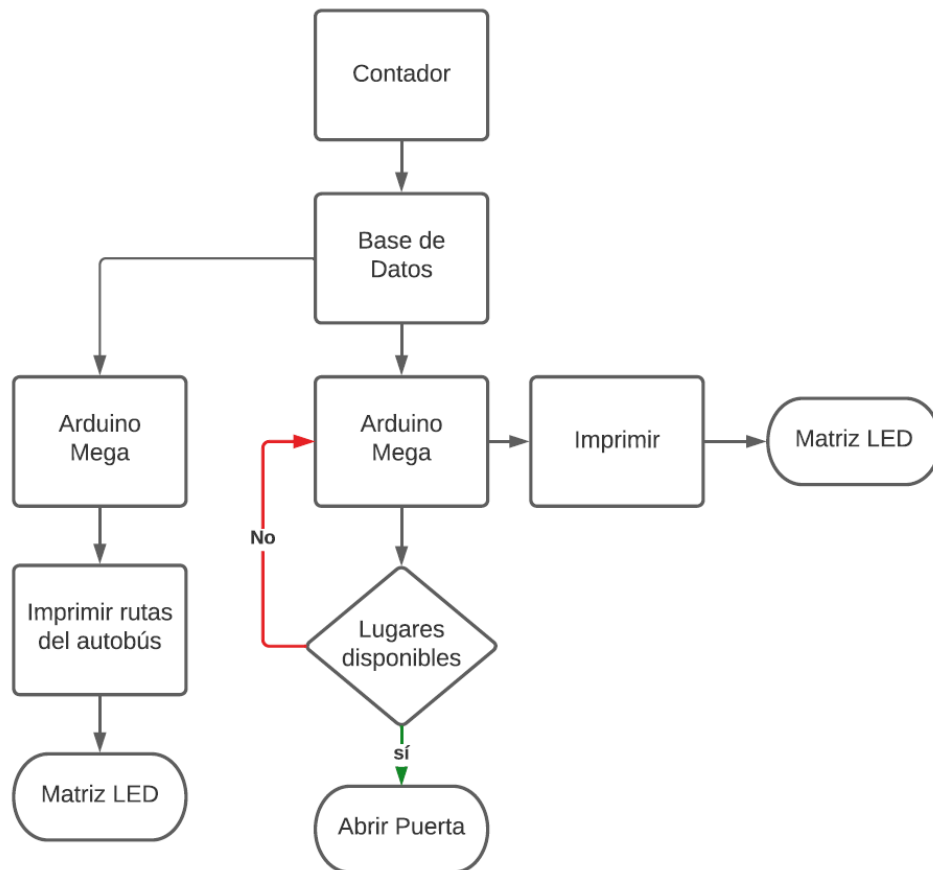
Nota. Elaboración Propia.

2.6.5. Programación del Arduino Mega

Uno de los Arduino Mega imprime las rutas del autobús, y el otro Arduino Mega se encarga de imprimir la cantidad de lugares disponibles que ahí en el autobús, este valor lo saca de la base de datos y así puede determinar si existen o no lugares disponibles para poder abrir o no la puerta, de esta forma, permitiendo el acceso a las personas a la unidad de transporte como se puede observar el diagrama de flujo en la Figura 37.

Figura 37

Flujograma de la Programación del Arduino Mega



Nota. Elaboración Propia.

De la selección adecuada de los diferentes elementos dependió del correcto funcionamiento del prototipo de un sistema de conteo de pasajeros con control de aforo, debido a que el contero de las personas se realizó con visión artificial se tomó en cuenta el lugar donde se colocaron las cámaras para que todo el sistema funcione correctamente, también de la posición de la pantalla y matriz de leds, así el conductor como los usuarios tengas en conocimiento de los lugares disponibles en la unidad de transporte.

CAPÍTULO 3

CONSTRUCCIÓN E IMPLEMENTACIÓN DEL PROTOTIPO DE CONTEO DE PASAJEROS CON CONTROL DE AFORO Y PRUEBAS DE FUNCIONAMIENTO

En esta sección se detalló la construcción del prototipo de conteo de pasajeros con control de aforo, se consideró las dimensiones del autobús igual que la capacidad máxima de personas que pueden estar adentro de este, de esta manera poder adaptarse al autobús y al sistema mecánico de la puerta de entrada, con el fin de realizar las pruebas respectivas y ajustar la posición de las cámaras para poder realizar un conteo más exacto controlando el aforo del autobús.

3.1. Materiales utilizados para la elaboración del prototipo

A continuación, en la Tabla 6 se presenta los diferentes materiales utilizados en la elaboración e implementación del prototipo de un sistema de conteo de personas con control de aforo.

Tabla 6

Materiales utilizados en el Prototipo

Material	Cantidad	Valor Unidad (\$)	Valor Total (\$)
Raspberry Pi 4 8 RAM	2	250	500
Arduino Mega	2	25	50
Pantalla Nextion NX8048T050	1	130	130
Cámara Logitech C270	2	57,50	115
Matriz Led 64x32	2	40	80
Sensor ultrasonido HC-SR04	2	3	6
Electroválvula	1	15	15
Modulo Relé	1	3	3

Fuentes de alimentación	4	4	16
Impresión 3D	4	15	60
Cable de Red	1	10	10
Cable de Timbre	1	10	10
Otros	1	20	20
Total (\$)			1015

Nota. Elaboración Propia

Como se puede apreciar en la Tabla 6, indica los valores comerciales de cada material utilizado en la elaboración del prototipo, dando un costo total de \$ 1015 dólares.

3.2. Distribución y montaje de los compones electrónicos en el autobús

Se realizó la implementación del prototipo en el autobús número 7 de la Cooperativa Orozcotola en la ciudad de Andrade Marín, como se puede observar en la Figura 38, el autobús posee una capacidad máxima de 50 pasajeros, además el autobús tiene las siguientes dimensiones: 8,60 m de largo, 2,40 m de ancho y 3,20 m de alto.

Figura 38

Autobús de la Cooperativa Orozcotola



Nota. Elaboración Propia.

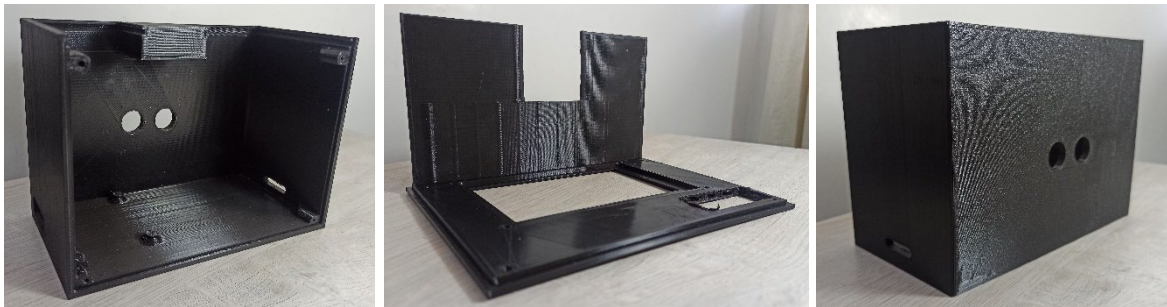
Debido a esto se realizó cuatro cajas hechas con impresión 3D con filamento de PETG, donde van a estar los materiales para que no exista desconexiones o daños en los componentes. De esta manera fue más factible la colocación del prototipo en el autobús.

3.2.1. Montaje del sistema de conteo en la puerta de entrada

Para poder tener una mayor facilidad de conteo el sistema se colocó en una caja hecha con impresión 3D, como se puede observar en la Figura 39. En esta caja están los componentes principales como la Raspberry PI 4, Pantalla HMI Nextion NX8048T050, el sensor ultrasonido HC-SR04 y la cámara.

Figura 39

Caja principal del sistema

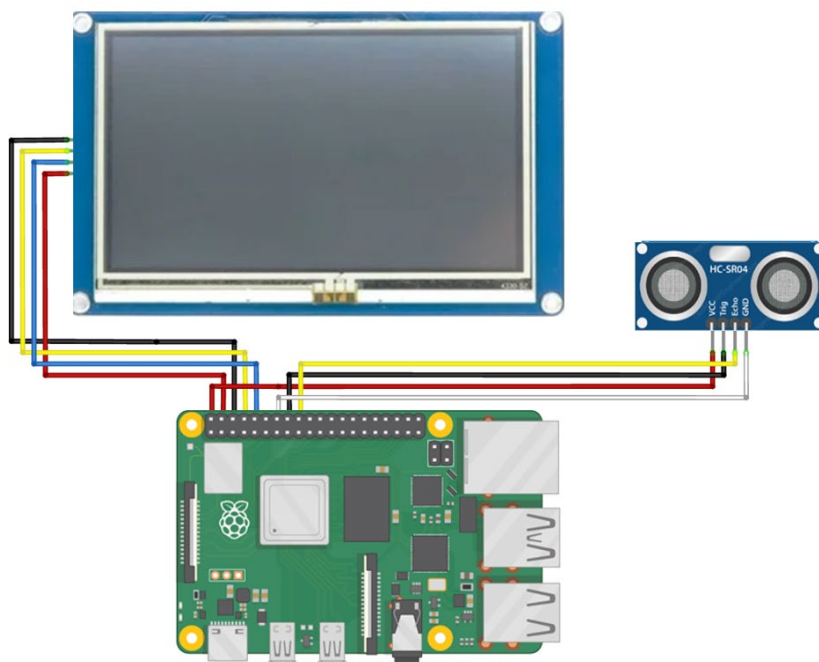


Nota. Elaboración Propia.

A continuación, se puede observar en la Figura 40, el diagrama de conexión de la Raspberry PI 4 con la pantalla Nextion NX8048T050 y el sensor HC-SR04.

Figura 40

Diagrama de conexión de los componentes de la caja principal



Nota. Elaboración Propia.

Como no se logra apreciar en la imagen, la conexión de los diferentes pines que tienen el sensor HCSR-04 y la pantalla Nextion NX8048T050 se los puede evidenciar de mejor manera en la Tabla 7.

Tabla 7
Conexión de pines caja principal

Componentes	Pines	Pines Raspberry PI 4
	+ 5V	4 (5V power)
Pantalla Nextion NX8048T050	RX	8 (GPIO 14 TXD)
	TX	10 (GPIO 15 RTX)
	GND	6 (Ground)
	VCC	2 (5V power)
Sensor Ultrasonido HC-SR04	Tring	16 (GPIO 23)
	Echo	18 (GPIO 24)
	GND	14 (Ground)

Nota. Elaboración Propia

Después de realizar la conexión entre los componentes se ensambló todo en la caja y se atornilló, para que de esta manera, quede bien asegurado como se puede observar en la Figura 41.

Figura 41
Ensamble en la caja principal



Nota. Elaboración Propia.

Una vez ensamblado todos los componentes se procede a colocar en el autobús en la puerta de entrada como se puede observar en la Figura 42. De esta manera, se coloca la caja para que el sensor pueda detectar a la persona y el conductor pueda observar la pantalla desde la posición donde está sentado, además se conectó la cámara en un puerto USB que se colocó de forma que visualice a las personas desde el momento que ingresan al autobús.

Figura 42

Implementación del prototipo en el autobús



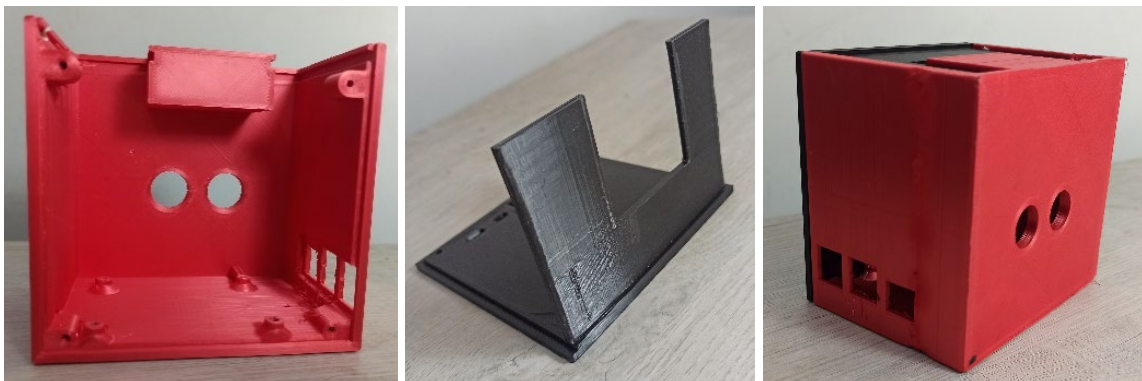
Nota. Elaboración Propia.

3.2.2. Montaje de Sistema de conteo en la puerta de salida

Para la puerta de salida se colocó en una caja hecha con impresión 3D como se puede observar en la Figura 43. Además, la acción que realizó este sistema es contar a las personas que se van a bajar del autobús, siendo de esta manera, que los espacios ocupados vayan disminuyendo.

Figura 43

Caja secundaria del sistema

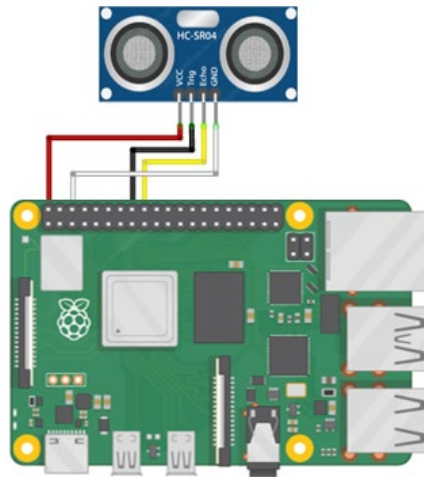


Nota. Elaboración Propia.

Después se realizó las conexiones como se puede observar en la Figura 44, de la Raspberry PI 4 con el sensor ultrasonido HC-SR04 y la cámara.

Figura 44

Diagrama de conexión de los componentes de caja secundaria



Nota. Elaboración Propia.

En la Tabla 8 se puede evidenciar de mejor manera la conexión de los diferentes pines que tienen el sensor ultrasonido HCSR-04 y la Raspberry PI 4, visto que mediante la imagen no se aprecia de una forma óptima.

Tabla 8

Conexión de pines caja secundaria

Componentes	Pines	Pines Raspberry PI 4
Sensor Ultrasonido HC-SR04	VCC	2 (5V power)
	Tring	16 (GPIO 23)
	Echo	18 (GPIO 24)
	GND	14 (Ground)

Nota. Elaboración Propia

Después de realizar la conexión entre los componentes se ensambló todo en la caja de la puerta de salida asegurando y colocando todo, como se puede observar en la Figura 45.

Figura 45

Ensamble en la caja secundaria



Nota. Elaboración Propia.

Una vez ensamblado todos los componentes, se procedió a colocar en el autobús en la puerta de salida como se puede observar en la Figura 46. De esta manera se colocó la caja para que pueda detectar el sensor a la persona que va a salir de la unidad de transporte, además se conectó la cámara en un puerto USB y se colocó de forma que visualice a las personas desde el momento que van a salir del autobús.

Figura 46

Implementación del Prototipo en el Autobús



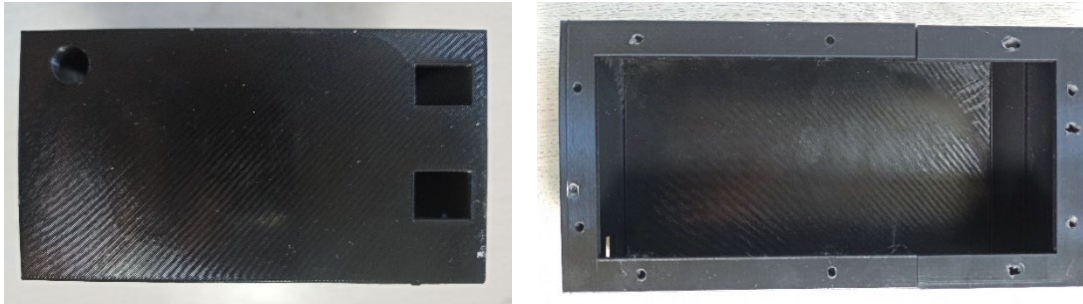
Nota. Elaboración Propia.

3.2.3. Montaje de las Matrices Leds

Para las matrices leds que se colocaron en la parte delantera del autobús, se realizó dos cajas para cada matriz como se puede observar en la Figura 47. De este modo las matrices leds están más seguras y no sufrieron desconexiones de los componentes. Por lo que, de esta manera podrán mostrar la cantidad de espacios disponibles y los lugares por donde pasa la unidad de transporte.

Figura 47

Cajas para las Matrices Leds

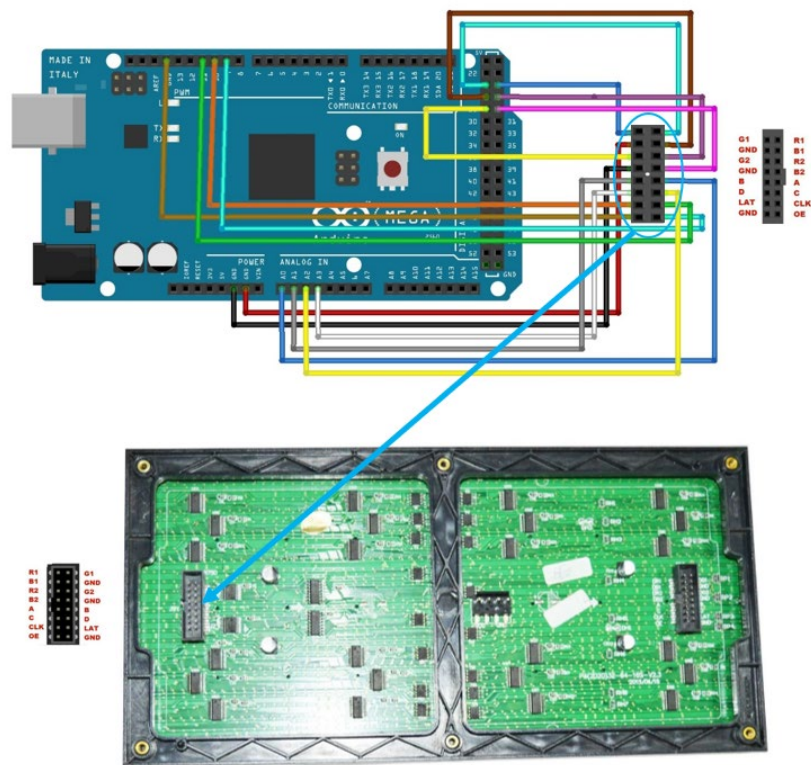


Nota. Elaboración Propia.

Después se realizó las conexiones como se puede observar en la Figura 48, de cada Arduino Mega con una Matriz led 64 x 32.

Figura 48

Diagrama de conexión de los componentes de las cajas de las matrices leds



Nota. Elaboración Propia.

Como no se logra apreciar en la imagen la conexión de los diferentes pines que tienen el Arduino Mega y la matriz leds se los puede evidenciar de mejor manera en la Tabla 9.

Tabla 9

Conexión de pines caja de matrices

Componentes	Pines	Pines Arduino Mega
Matriz Led 64x32	R1	24
	B1	26
	R2	27
	B2	29
	A	A0
	C	A2
	CLK	11
	OE	9
	G1	25
	GND	GND
	G2	28
	GND	GND
	B	A1
	D	A3
STB – LAT	10	
GND	GND	

Nota. Elaboración Propia

Después de realizar la conexión entre los componentes se ensambló todo en la caja de las Matrices y se aseguró todo como se puede observar en la Figura 49.

Figura 49

Ensamble de las Matrices leds



Nota. Elaboración Propia.

Una vez ensamblado todos los componentes se procedió a colocar en el autobús como se puede observar en la Figura 50. Para que las personas puedan visualizar la cantidad de lugares disponibles y los lugares por donde pasa el autobús, además se conectó las dos matrices a dos puertos USB de la Raspberry PI 4, para obtener los datos de los espacios disponibles en la unidad de transporte y poder visualizarlos en una de las matrices.

Figura 50

Implementación del prototipo en el autobús



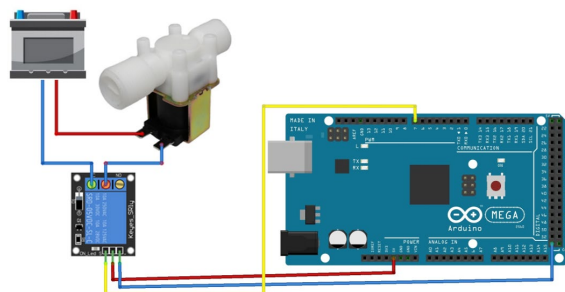
Nota. Elaboración Propia.

3.2.4. Montaje del Actuador en la puerta de entrada

Como el sistema para abrir la puerta resulto ser mediante una electroválvula manual, se instaló otra electroválvula para impedir el paso de aire cuando el autobús este con el aforo completo. En la Figura 51 se puede observar la conexión de esta electroválvula con el relé y el Arduino Mega que indica el número de espacios restantes en el autobús.

Figura 51

Diagrama de conexión del Actuador



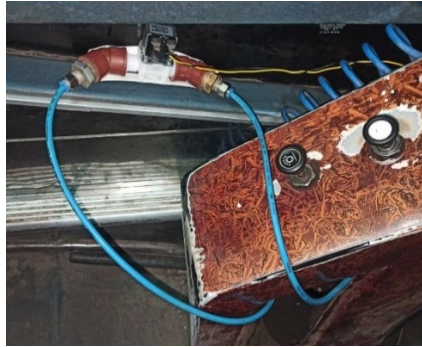
Nota. Elaboración Propia.

Una vez ensamblado todos los componentes se procede a colocar la electroválvula en el autobús como se puede observar en la Figura 52. De esta manera permite realizar el bloqueo de la puerta cuando llegue a la cantidad máxima de aforo. Además, este bloqueo de

la puerta tiene un botón de emergencia que desbloquea la puerta en caso se surgir un accidente.

Figura 52

Implementación del prototipo en el autobús

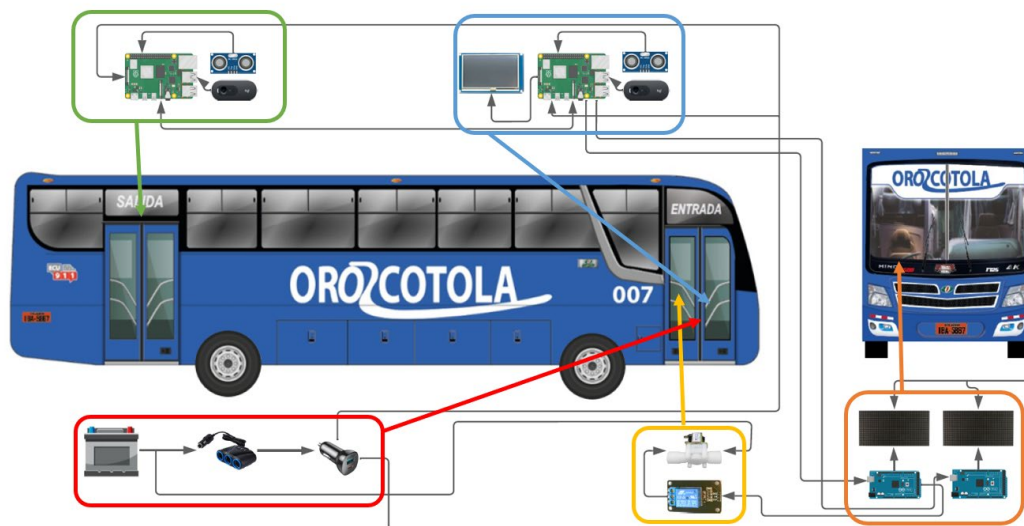


Nota. Elaboración Propia.

A continuación, en la Figura 53 se muestra un esquema general de cómo va a ir todo implementado en el autobús.

Figura 53

Esquema de conexión general



Nota. Elaboración Propia.

3.3. Software

Los softwares utilizados en el prototipo son: Visual Studio Code para la programación del conteo de personas, Arduino para la programación de las matrices leds y Nextion Editor para la programación de las pantalla Nextion NX8048T050, debido a que se va a implementar

en un autobús, el prototipo se diseñó para que se active cuando el conductor encienda el autobús, esto quiere decir que el software del prototipo se inicia automáticamente una vez el autobús este encendido.

3.3.1. Interfaz de la Pantalla HMI

En esta sección se detalla el funcionamiento de la pantalla HMI, que está colocada en el autobús. Como se puede observar en la Figura 54, la pantalla tiene algunas funciones principales como el total de pasajeros actuales, el total de pasajeros, el aforo y la información, todo esto ayuda al conductor a conocer cuántas personas están en el autobús.

Figura 54

Interfaz de la Pantalla HMI Nextion



Nota. Elaboración Propia.

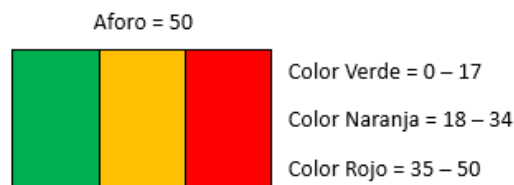
A continuación, se detalla la utilización de la *Interfaz de la Pantalla HMI Nextion* como son:

- El cuadro azul está el Total de Pasajeros Actuales, la función de este es indicar la cantidad de pasajeros que están dentro del autobús.
- En el cuadro verde está el Total de Pasajeros, la función de este es indicar el número de pasajeros que se subieron al autobús durante toda la jornada de trabajo del conductor.
- En el cuadro plomo está el Total de Pasajeros, la función de este es indicar el número de pasajeros que se salieron del autobús durante toda la jornada de trabajo del conductor.

- En el cuadro amarillo está el Aforo, la función de este es indicar el aforo máximo del autobús.
- En el cuadro negro está el Estado, la función de este es indicar mediante colores el aforo del autobús, tiene 3 colores que son verde (bajo), naranja (medio) y rojo (máximo), como se puede observar en la Figura 54.

Figura 55

Distribución de los colores de la barra de Estado



Nota. Elaboración Propia.

- En el cuadro tomate esta un botón de información, al momento de aplastarlo muestra el nombre de la universidad, la facultas, la carrera, el tema, el autor, el tutor, el asesor y el año electivo como se puede observar en la Figura 55. En la nueva interfaz en el cuadro rojo esta un botón de regresar, cuando se aplasta regresara a la interfaz original donde esta los datos del monitoreo.

Figura 56




Distribución de los colores de la barra de Estado



Nota. Elaboración Propia.

A continuación, se indica como estará la interfaz de la Pantalla HMI Nextion durante el funcionamiento del prototipo.

Tabla 10
Interfaz Pantalla HMI

Interfaz Pantalla HMI	Descripción
	<p>Interfaz de la Pantalla HMI, cuando el autobús esté con un aforo bajo (0 - 17).</p>
	<p>Interfaz de la Pantalla HMI, cuando el autobús esta con la mitad de aforo (18-34).</p>
	<p>Interfaz de la Pantalla HMI, cuando el autobús esta con el aforo máximo (35 - 50).</p>

Nota. Elaboración Propia.

Para poder regresar a la interfaz que tiene la información del aforo del autobús, se debe aplastar el botón regresar y automáticamente se muestra la otra interfaz.

3.3.2. Matriz Leds con espacios restante

Esta matriz de leds se encarga de indicar los espacios disponibles que existen en el autobús a las personas que se van a subir como se puede observar en la Figura 56.

Figura 57

Interfaz de la Matriz led con espacios restantes



Nota. Elaboración Propia.

3.3.3. Matriz Leds con las rutas

Esta matriz se encarga de indicar los lugares por donde es la ruta del autobús, la ruta tiene alrededor de 6 lugares para realizar paradas, por esta razón se muestran 3 lugares por 5 segundos y después se muestran los otros 3 lugares como se puede observar en la Figura 57, facilitando de esta manera indicar los 6 lugares por donde circula el autobús.

Figura 58

Interfaz de la matriz con las rutas



Nota. Elaboración Propia.

3.3.4. Reconocimiento facial

En esta sección se explica cómo se realizó la detección y el conteo de las personas, además de cómo interactúa con los demás equipos para el funcionamiento de todo el

prototipo. El lenguaje utilizado es Python en su versión 3.6.13, las librerías utilizadas en el prototipo fueron OpenCV debido a que se especializa en sistemas con visión artificial, Sqlite3 para la base de datos, Pyserial para la comunicación con los Arduino mega y la pantalla HMI.

Primero el sensor ultrasonido detecta a la persona que debe estar en el rango de detección menor a un metro y automáticamente se enciende la cámara y realiza la detección del rostro, para esto debemos utilizar un aprendizaje automático se necesita una gran cantidad de imágenes, en este caso de rostros para entrenar a un clasificar de este modo solo detectar los rostros de personas. Para esto se necesita imágenes positivas de rostros e imágenes negativas de objetos que no son rostros, después se realiza la extracción de características de las imágenes, después se procede al entrenamiento y finalmente obtener un clasificador.

OpenCV nos ofrece clasificadores pre entrenados para la detección de rostro, sonrisas, ojos, nariz, entre otros objetos, pero el clasificador que se utilizó fue haarcascade_frontalface_alt.xml, este archivo se lo puede encontrar en el repositorio github en el enlace: <https://github.com/opencv/opencv/tree/master/data/haarcascades>, a continuación, está la codificación de la detención de rostro en el autobús.

```
if distance <= 100: # Si la distancia es menor a un metro
    # Intentar detectar un rostro
    ret, frame = video_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

    # Si se ha detectado un rostro y no estamos en estado de bloqueo, incrementar el contador y agregarlo a la base de datos
    if len(faces) > 0 and not lock:
        count += 1

        #print('llego')

        # Conectar a la base de datos
        conn = sqlite3.connect('autobus.db')
        c = conn.cursor()

        # Insertar un nuevo registro
        c.execute("INSERT INTO pasajeros (personas_en_el_bus, personas_al_dia, aforo) VALUES (?, ?, ?)", (countG, count, aforo1))
        # Guardar (commit) los cambios
        conn.commit()
        # Cerrar la conexión a la base de datos
        conn.close()
        # Activar el bloqueo
        lock = True
        GPIO.output(CAM1, False)

    elif len(faces) == 0:
        # Si no se detecta ningún rostro, desactivar el bloqueo
        lock = False
        GPIO.output(CAM1, True)

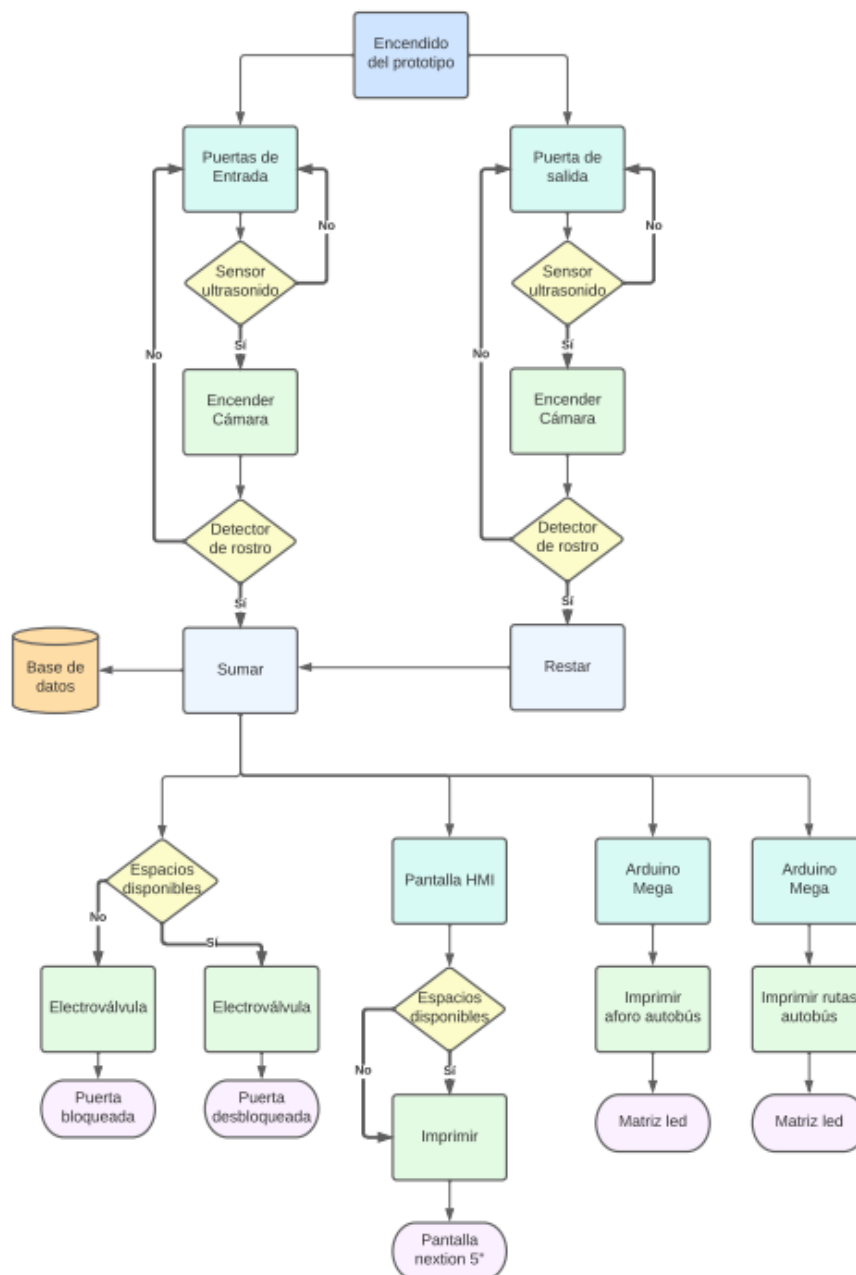
countG = count - count2
if countG < 0:
    countG = 0
gent = aforo1 - countG
if gent < 0:
    gent = 0
```

Una vez el sensor detecta a la persona que está a una distancia menor a un metro, se activa la cámara y se manda a llamar al clasificador haarcascade_frontalface_alt.xml que está en los archivos de la raspberry pi 4 principal, cuando el clasificador detecte el rostro de una persona, se sumará o se restará dependiendo si está la persona en la puerta de entrada o en

la puerta de salida, si existen más rostros de personas no se realizara el suma o resta, además se envía los datos de las personas que ingresaron o salieron del autobús y se mostrara en la pantalla HMI y en la matriz de led el número de personas que están en el autobús. A continuación, en el Figura 58 se puede observar el Flujograma del funcionamiento del Prototipo.

Figura 59

Diagrama de flujo general del Prototipo



Nota. Elaboración Propia.

3.4. Pruebas del Prototipo

A continuación, se muestran las pruebas del sistema eléctrico, electrónico y de software antes de que el prototipo entre en funcionamiento en la ruta de trabajo del autobús.

3.4.1. Pruebas Sistema Eléctrico

Como la batería del autobús es de 12 V, para no poner cables desde la batería al prototipo se utilizó la toma de mechero o más conocido como puerto para encender cigarrillos, el autobús solo tenía una toma de mechero se le adaptó para que tenga 3 tomas de mechero y de esta manera conectar 3 fuentes de 5 V de 3 A hasta 4.8 A, a las tomas de mechero como se puede observar en la Figura 59, de esta manera se obtuvo el voltaje y corriente suficiente para que funcionen las 2 Raspberry, 2 Arduino Mega y las 2 matrices Leds.

Figura 60

Conexión eléctrica del Prototipo



Nota. Elaboración Propia.

Una vez todo implementado en el autobús se realizó la prueba respectiva y se pudo evidenciar que el prototipo tuvo la energía suficiente y no se alteró nada del sistema eléctrico del autobús, facilitando la conexión del prototipo.




3.4.2. Pruebas Sistema Electrónico

Durante las pruebas realizadas, en la Tabla 10, no existió ningún problema el sistema electrónico del prototipo solo que el sensor ultrasonido HC-SR04, hacia que la cámara se activará en algunas ocasiones cuando se cerraba la puerta de entrada del autobús, debido a

que la puerta se movía o existían sombras dando por error la presencia de una persona, pero como la cámara no detecta ningún rostro no afectaba al sistema de conteo en ninguna forma. La solución a este problema es cambiar la distancia del sensor ultrasonido HC-SR04. Las pruebas realizadas participaron mujeres y hombres de distintas edades, de este modo se evidencio que el sistema de conteo detectaba el rostro de las personas sin importan su edad o género.

Tabla 11

Pruebas realizadas en el autobús

Pruebas en el autobús	Descripción	Conteo
	Niño	Correcto
	Adolescente	Correcto
	Adolescente	Correcto



Señor

Correcto



Adulto Mayor

Correcto



Señora

Correcto



Señora

Correcto



Señora

Correcto



Señora Correcto



Adulta Mayor Correcto



Señora con
Mascarilla Incorrecto

Nota. Elaboración Propia.

3.4.3. Pruebas de Software

Como el prototipo debía estar instalado en el autobús se realizó las pruebas con el autobús parado para evidenciar que exista una detención y conteo de los rostros de las personas, de esta manera se pudo detectar los rostros de las personas que se subieron y bajaron del autobús como se puede observar en la Figura 60, además se programó las dos Raspberry PI 4 para que cuando el autobús se encienda el prototipo inicie el conteo de las personas que van a subir y bajar del autobús, dando mayor facilidad al conductor de solo mirar la pantalla HMI y saber cuántas personas están en el autobús, también, las dos matrices funcionaron correctamente dando la cantidad de espacios restantes del autobús, dato que obtenían de la base de datos almacenada en la Raspberry PI 4.

Figura 61

Pruebas del Prototipo



Nota. Elaboración Propia.

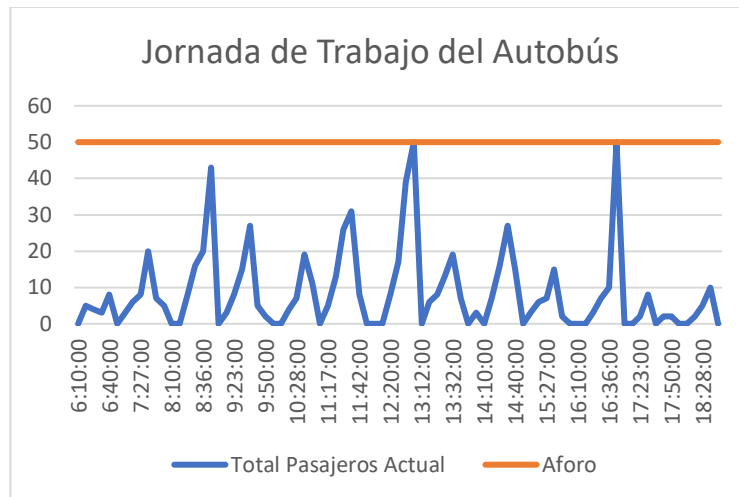
Durante las pruebas hubo algunas personas que todavía utilizan mascarillas, debido a esto las cámaras no detectaron los rostros al momento que se subieron y bajaron del autobús, pero no hubo ningún inconveniente ya que se subieron cuando el autobús aun no estaba en su aforo máximo, así que no pudieron alterar el funcionamiento del prototipo. Para dar solución se puede realizar un cascade con personas con mascarillas y de este modo contarlas al momento que estas se suban y bajen del autobús.

3.5. Resultados

En la Raspberry PI 4 principal, se guarda un archivo Excel con los datos de las personas que se subieron y bajaron durante todo el trayecto del autobús en un día normal de trabajo, en la Figura 61, se puede visualizar los datos de forma gráfica viendo como fue el aforo del autobús durante toda la jornada de trabajo.

Figura 62

Pruebas del Prototipo en una jornada de trabajo



Nota. Elaboración Propia.

En la gráfica se puede observar que el autobús llegó a su aforo máximo en dos ocasiones durante toda la jornada de trabajo esto quiere decir que se bloqueó dos veces la puerta de entrada, y el resto del tiempo el autobús no llegó a su aforo máximo, esto quiere decir que el autobús cumple con el aforo respectivo para que las personas viajen tranquilas a su lugar de destino.

3.5.1. Funcionalidad del Sistema de Conteo

Se puede decir que después de las pruebas realizadas el sistema conteo de personas es de un 90% debido ya que estuve en el autobús durante toda la jornada de trabajo y puede evidenciar que en ocasiones se subieron personas con mascarillas, ocasionado que el sistema no detecte los rostros y fallar en el monitoreo de aforo en el autobús, pero como sucedió en una jornada donde el autobús no estaba a su máxima capacidad pasaron desapercibidas.

3.5.2. Funcionalidad del Control de Aforo

La funcionalidad del control de aforo en la puerta de entrada fue en dos ocasiones impidiendo que se suban más de las 50 personas en el autobús, pero como las personas se iban quedando en el transcurso la puerta nuevamente se desbloqueó y permitió que ingresen más personas en el autobús, esto quiere decir que el control de aforo es un 100 % debido a que o estuve en el autobús y pude observar que la puerta se bloqueó e impidiendo el ingreso de personas cuando supera el aforo.

CONCLUSIONES

Después de describir los mecanismos y métodos, se puede concluir que el mecanismo más factible para implementar en un autobús es mediante visión artificial siendo el método más adecuado para realizar el conteo de personas el Haar-Cascade, ya que permite un reconocimiento de los rostros, pero sin reconocer otros objetos.

En el diseño del prototipo se pudo determinar que los componentes seleccionados como Raspberry PI 4 de 8 RAM ayudó con un procesamiento más rápido para la detección con las cámaras, de esta manera, se pudo realizar el conteo de las personas que van a ingresar y salir de la unidad de transporte. Además, el prototipo tiene una pantalla HMI y de dos matrices de leds, las cuales facilitaron al conductor y usuarios conocer la cantidad de aforo a tiempo real.

Durante las pruebas realizadas en el autobús, se demostró que existe un monitoreo del aforo en la unidad de transporte, además, las dos cámaras detectaron de forma clara los rostros de las personas, permitiendo que cuando llegue al aforo máximo la puerta no se abra e impidiendo el ingreso de más personas, siendo la única forma que pueda subirse otro pasajero hasta que otro se baje de la unidad para que existan nuevamente lugares disponibles. Esto permitió que exista una mejor interacción con los pasajeros y el conductor del autobús.

RECOMENDACIONES

Verificar las nuevas tecnologías utilizadas en el procesamiento y detección de imagen para la visión artificial, además se recomienda probar con otras cámaras más avanzada para tener una mejor calidad de video y detectar a las personas más rápido, además utilizar cámaras que puedan ver con baja luminosidad y así tener un conteo sin la necesidad de utilizar la luz del autobús en las noches.

Se recomienda utilizar o investigar más acerca de los sensores de distancia debido a que en ocasiones estos sensores activaban la cámara sin haber una persona subiendo o bajando del autobús, se puede ocupar otro sensor existente en el mercado y verificar que no existan este tipo de errores.

Por último, se puede seguir ampliando el campo de la visión artificial o seguir innovando el prototipo, de este modo en un futuro no solo realizar el conteo de personas con detención de rostro sino identificar si se trata de un niño, joven, señor o adulto mayor y de esta manera tener un registro más claro de cuánto dinero genera un autobús durante toda una jornada de trabajo.

REFERENCIAS

- 330ohms. (05 de Abril de 2016). *330ohms*. Obtenido de 330ohms: <https://blog.330ohms.com/2016/04/05/arduino-mini-micro-y-nano-que-tan-diferentes-son-2/>
- A3M. (2019). *Identificación personal control de acceso*. España: A3M.
- Airwork. (1 de Julio de 2022). *Canales sectoriales*. Obtenido de Canales sectoriales: <https://www.interempresas.net/Robotica/Articulos/393575-Airwork-lanza-la-electrovalvula-neumatica-de-control-directo-serie-VF.html>
- AIS. (9 de Febrero de 2019). *VISION SYSTEMS*. Obtenido de VISION SYSTEMS: <https://www.aisvision.com/es/vision-artificial/camaras-de-vision-artificial/camaras-vision-artificial-series/>
- Alfonso, B. (24 de Julio de 2018). *SYSCOM*. Obtenido de SYSCOM: <https://www.syscomblog.com/2018/07/conteo-de-pasajeros.html>
- Alonso, R. (04 de Junio de 2020). *Hard Zone*. Obtenido de Hard Zone: <https://hardzone.es/tutoriales/compras/que-modelo-raspberry-pi-comprar/>
- Alonso, R. (17 de Julio de 2020). *Hard Zone*. Obtenido de Hard Zone: <https://hardzone.es/reportajes/que-es/soc-caracteristicas-hardware/>
- Alvarez, M. (19 de Noviembre de 2003). *Desarrolloweb*. Obtenido de Desarrolloweb: <https://desarrolloweb.com/articulos/1325.php#:~:text=Python%20es%20un%20lenguaje%20de,red%20o%20incluso%2C%20p%C3%A1ginas%20web.>
- ANT. (2017). *Listado de Dispositivos de Control Homologados*. Ecuador: Agencia Nacional de. Obtenido de <https://www.ant.gob.ec/servicios/homologacion-de-taximetros/listado-de-taximetros-homologados/>
- ANT, S. G. (2016). *Resolución No. 085-DIR-2016-ANT REFORMA A LAS RESOLUCIONES No. 002-DIR-2014-ANT QUE CONTIENE EL 'INSTRUCTIVO PARA EL CONTROL Y EVALUACIÓN DEL SISTEMA DE CAJA COMÚN EN LAS OPERADORAS DE TRANSPORTE PÚBLICO TERRESTRE' Y No. 003-DIR-2014-ANT QUE CONTIENE*. Ecuador: ANT. Obtenido de <http://www.ant.gob.ec/index.php/transito-7/resoluciones-2016/file/3758-resolucion-no-085-dir-2016-ant>
- Arduino. (2022). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/>

autoprotegido. (17 de Enero de 2015). *autoprotegido*. Obtenido de autoprotegido: <http://autoprotegido.com/contacto.html>

AV Electronics. (2022). Obtenido de AV Electronics: <https://avelectronics.cc/producto/modulo-matriz-led/>

AV Electronics. (2022). Obtenido de AV Electronics: <https://avelectronics.cc/producto/modulo-rele-1-canal/>

AZERONA. (2022). *Ubuy*. Obtenido de Ubuy: <https://www.ubuy.com.tw/en/product/1BRCPX2O-pixel-led-panels-digital-led-module-indoor-led-display-screen-rgb-matrix-led-board-p3-19296mm>

Azuela, S., & Humberto, J. (2013). *Visión artificial: Rasgos descriptores para el reconocimiento de objetos*. Madrid.

bcnvision. (12 de Septiembre de 2017). *Grupo bcnvision blog*. Obtenido de Grupo bcnvision blog: <https://www.bcnvision.es/blog-vision-artificial/componentes-vision-artificial/>

BEA. (2018). *Barras Contadoras de Pasajeros*. Mexico: Sistemas BEA. Obtenido de Sistemas BEA.

BEA-RT. (2018). *Contador de Pasajeros*. Mexico: Sistema BEA.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV*. United States of America: O'REILLY.

Bravo, F. (15 de Mayo de 2020). *ECOMMENCER NEWS*. Obtenido de ECOMMENCER NEWS: <https://www.ecommercenews.pe/lanzamientos-ecommerce/2020/app-pagabus.html>

Caballero, R. (2017). *APLICACIÓN PRÁCTICA DE LA VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO DE ROSTROS EN UNA IMAGEN, UTILIZANDO REDES NEURONALES Y ALGORITMOS DE RECONOCIMIENTO DE OBJETOS DE LA BIBLIOTECA OPENCV*. Bogota: Repositorio Universitario Universidad Distrital Francisco José de Caldas.

Centinkaya, H., & Akcay, M. (2015). *People Counting at Campuses*". Procedia.

Comercio, E. (2014). *Especificaciones Técnicas que debe tener un bus*. Recuperado el 19 de Enero de 2022, de especiales el comercio: <https://especiales.elcomercio.com/2014/09/buses-especificaciones/index.php>

Comercio, E. (8 de Enero de 2015). *La caja común ya funciona en cuatro ciudades*. Obtenido de La caja común ya funciona en cuatro ciudades:

<https://www.elcomercio.com/actualidad/ecuador/ecuador-caja-comun-funciona-cuatro.html>

Control de Accesos. (2 de Agosto de 2022). Obtenido de Control de Accesos: <https://controlaccesosistemas.com/conoce-los-tipos-de-torniquetes-de-acceso-que-existen/>

Currás, M., & Lola, T. (13 de 03 de 2012). *Academia.edu*. Obtenido de Academia.edu: https://www.academia.edu/24190825/Detecci%C3%B3n_de_Bordes_Vi%20si%C3%B3n_Artificial_13_03_2012

Derek, A. (30 de Septiembre de 2020). *eurosoft*. Obtenido de eurosoft: <https://solucioneseurosoft.com/control-acceso/control-de-acceso-con-torniquetes/>

Derek, A. (20 de Julio de 2020). *eurosoft*. Obtenido de eurosoft: <https://solucioneseurosoft.com/eurosoft/las-mejores-tarjetas-de-acceso/>

Dogán, I. (2008). *Programacion de microcontroladores PIC*. España: Marcombo. Obtenido de <https://elibro.net/es/lc/utnorte/titulos/45918>

Electronilab. (2022). Obtenido de Electronilab: <https://avelectronics.cc/producto/modulo-matriz-led/>

Fabara, S. (2 de Julio de 2020). *Xataka foto*. Obtenido de Xataka foto: <https://www.xatakafoto.com/actualidad/asi-se-revive-camara-analogica-juguete-utilizando-modulo-camara-alta-calidad-raspberry-pi-4>

Fernandez, Y. (23 de Septiembre de 2022). *Xataka*. Obtenido de Xataka: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Flores, F. (22 de Julio de 2022). *OpenWebinars*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

García, E. (2012). *Visión artificial*. Catalunya: UOC.

GCFGlobal. (s.f.). Obtenido de GCFGlobal: <https://edu.gcfglobal.org/es/informatica-basica/el-monitor-o-pantalla/1/>

Gervasoni, L., Dámato, J., Barbuzza, R., & Vénere, M. (2014). *UN MÉTODO EFICIENTE PARA LA SUSTRACCIÓN DE FONDO EN*. Argentina: Asociación Argentina de Mecánica Computacional.

Gon, U. (2010). *DETECCION DE PERSONAS A PARTIR DE VISION*. Madrid: CARLOS III DE MADRID,.

- González, A. (2006). *Técnicas y algoritmos básicos de visión artificial*. España: Universidad de La Rioja, Grupo de Investigación EDMANS.
- Guerrero, J. (21 de Septiembre de 2014). *PLUSELECTRIC*. Obtenido de PLUSELECTRIC: <https://pluselectric.wordpress.com/2014/09/21/arduino-uno-especificaciones-y-caracteristicas/>
- Hernández, E., Cabrera, A., & Sánchez, S. (2012). IMPLEMENTACIÓN HÍBRIDA HARDWARE SOFTWARE DEL ALGORITMO DE DETECCIÓN DE ROSTROS DE VIOLA-JONES SOBRE FPGA. *SciELO*, 114-124.
- ICOGtv. (23 de Septiembre de 2020). *Escuela Geología Professional*. Obtenido de Escuela Geología Professional: <https://www.icog.es/cursos/index.php/introduccion-a-la-programacion-en-python-para-geologos/>
- Industrias GSL. (22 de Junio de 2021). Obtenido de Industrias GSL: <https://industriagsl.com/blogs/automatizacion/regulador-de-voltaje>
- INEC. (2014). *Anuario de Estadísticas de Transportes 2014*. Ecuador: Ecuador en Cifras,.
- Java. (2022). *Java*. Obtenido de Java: https://www.java.com/es/download/help/whatis_java.html
- Kaewtrakulpong, P., & Bowden, R. (2001). *An Improved Adaptive Background Mixture*.
- Lowi. (28 de Junio de 2017). Obtenido de Lowi: <https://www.lowi.es/blog/como-funcionan-los-sensores-de-huella-dactilar/>
- Lozano, G., & Orduz, J. (2018). *DISEÑO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA REVISIÓN DEL NIVEL DE LLENADO DE BEBIDAS EMBOTELLADAS*. Barranquilla: FACULTAD DE INGENIERÍAS INGENIERÍA MECATRÓNICA.
- Lozano, R. (29 de Agosto de 2021). *Talos Electronics*. Obtenido de Talos Electronics: <https://www.taloselectronics.com/blogs/tutoriales/como-usar-pantalla-nextion-con-arduino#:~:text=Nextion%20es%20una%20pantalla%20touch,con%20cualquier%20microcontrolador%20mediante%20UART.>
- Lucas, J. (04 de 09 de 2019). *OpenWebinars*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-c/>
- Maxim. (2022). *Serially Interfaced, 8-Digit, LED Display Drivers*. Maxim.

Maza, G. (2017). *PROCESAMIENTO DE IMÁGENES USANDO OPENCV APLICADO EN RASPBERRY PI PARA LA CLASIFICACIÓN DEL CACAO*. Perú: FACULTAD DE INGENIERÍA Departamento de Ingeniería Mecánico-Eléctrica.

Morales, S. (2017). *SISTEMA DE VISIÓN ARTIFICIAL CON MONITOREO WEB PARA CONTEO*. Ambato: CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES.

Moreno, A., Sánchez, Á., & Vélez, J. (2003). *VISIÓN POR COMPUTADOR*. Madrid: Dykinson.

Mouser Electronics. (11 de Marzo de 2022). Obtenido de Mouser Electronics: <https://www.mouser.ec/new/raspberry-pi/raspberry-pi-3-bplus/>

movilflix. (2018). Obtenido de movilflix: <https://movilflix.com/que-es-movilflix/>

Naylamp Mechatronic. (2021). Obtenido de Naylamp Mechatronic: <https://naylampmechatronics.com/ardusystem-tarjetas/88-arduboard-nano-ch340g-mini-usb.html>

NAYLAMP MECHATRONICS. (s.f.). Obtenido de NAYLAMP MECHATRONICS: <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>

NeoDen, H. (14 de Abril de 2020). *NeoDen*. Obtenido de NeoDen: <http://www.neodenpnp.com/news/led-dot-matrix-display-36229286.html>

NEXTION. (s.f.). Obtenido de https://nextion.tech/nextion-editor/#_section1

NTE, & INEN. (2009). *VEHICULOS AUTOMOTORES. CARROCERIAS DE BUSES. REQUISITOS.1323*. Quito.

OpenCv. (19 de Julio de 2022). *OpenCV*. Obtenido de OpenCv: https://docs.opencv.org/4.x/d8/d38/tutorial_bgsegm_bg_subtraction.html

PcDuino. (2022). *LinkSprite*. Obtenido de LinkSprite: <https://www.linksprite.com/linksprite-pcduino/>

Perez, J. (2021). *Definicion.de*. Obtenido de Definicion.de: <https://definicion.de/electrovalvula/>

Raspberry. (2022). *Raspberry*. Obtenido de Raspberry: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

Reiz, M. (23 de Enero de 2011). *Visión Artificial Para Todos*. Obtenido de Visión Artificial Para Todos: <https://visionartificialparatodos.wordpress.com/2011/01/23/camaras-para-vision-artificial/>

- Rodríguez, K. (2017). *Análisis e Implementación del algoritmo de detección fácil de Viola-Jones*. Ibarra: Repositorio Universitario Universidad Técnica del Norte.
- Roger. (31 de Mayo de 2021). *rfid*. Obtenido de rfid: <https://www.rfidfuture.com/es/difference-between-a-proximity-card-and-vicinity-card.html>
- Secretaria General ANT. (2014). *Resolución No 002-DIR-ANT INSTRUCTIVO PARA EL CONTROL Y EVALUACIÓN DEL SISTEMA DE CAJA COMÚN EN LAS OPERADORAS DE TRANSPORTE PÚBLICO TERRESTRE*. Ecuador: Agencia Nacional de Tránsito. Obtenido de <http://www.ant.gob.ec/index.php/transito-7/resoluciones-2014/file/2027-resolucion-no-002-instructivo-para-el-control-y-evaluacion-del-sistema-de-caja-comun-en-las-operadoras-de-transporte-publico-terrestre>
- Sensing*. (s.f.). Obtenido de Sensing: <https://sensores-de-medida.es/medicion/sensores-y-transductores/sensores-de-distancia/#:~:text=Los%20sensores%20de%20distancia%2C%20transductores,o%20medida%20de%20desplazamiento%20lineal.>
- SmartCounter. (03 de Marzo de 2019). *Proconsi*. Obtenido de Proconsi: <https://www.proconsi.com/sistema-de-conteo-de-pasajeros-smartcounter>
- Solano, G. (2 de Septiembre de 2019). *OMES*. Obtenido de OMES: <https://omes-va.com/leerimagen/>
- Solano, G. (14 de Noviembre de 2020). *OMES*. Obtenido de OMES: <https://omes-va.com/sustraccion-de-fondo-opencv-python/>
- Torres, H. (s.f.). *HETPRO*. Obtenido de HETPRO: <https://hetpro-store.com/TUTORIALES/pantalla-nextion-arduino/>
- UNIT Electronics*. (s.f.). Obtenido de UNIT Electronics: <https://uelectronics.com/categoria-producto/optoelectronica/display/>
- Valle, L. (12 de Junio de 2018). *Programarfacil*. Obtenido de Programarfacil: <https://programarfacil.com/blog/vision-artificial/deteccion-de-movimiento-con-opencv-python/#:~:text=Substracci%C3%B3n%20con%20fotogramas%20anteriores&text=La%20t%C3%A9cnica%20consiste%20en%20tomar,la%20velocidad%20de%20los%20objetos.>
- VARTA. (s.f.). Obtenido de VARTA: <https://www.varta-automotive.es/es-es/soporte-para-baterias-varta/conceptos-basicos-sobre-baterias/como-funciona-una-bateria>

Veriddica. (30 de Marzo de 2022). *Veriddica*. Obtenido de Veriddica:
<https://veriddica.com/tipos-de-tarjetas-sin-contacto>

VIEWNEXT. (16 de Octubre de 2019). Obtenido de VIEWNEXT:
<https://www.viewnext.com/watson-funcionalidades/>

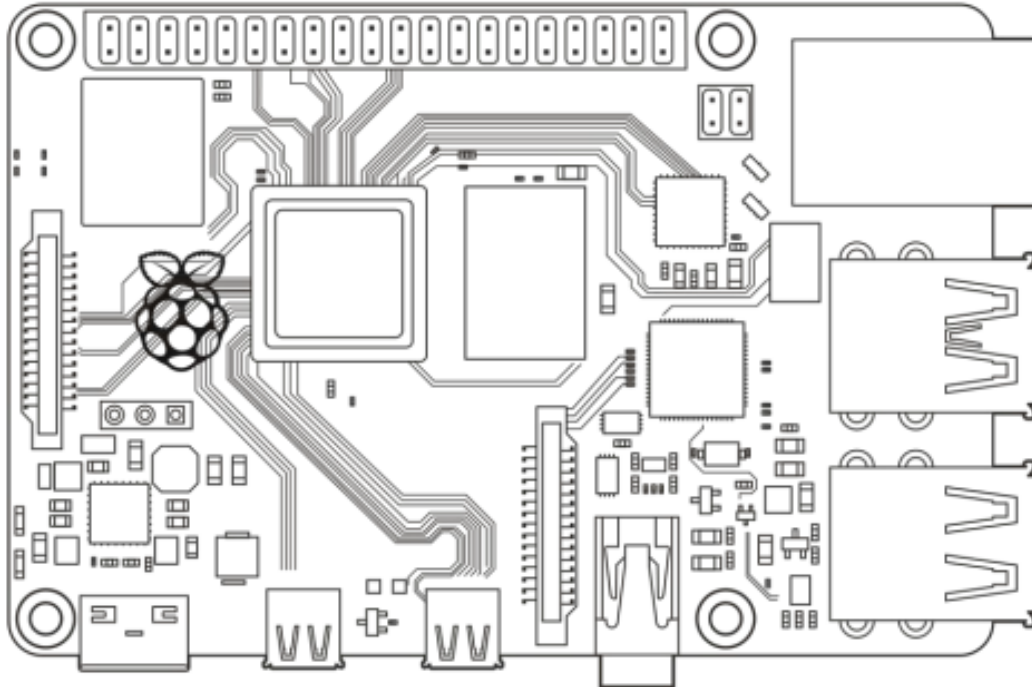
VIEWNEXT. (16 de Octubre de 2019). Obtenido de VIEWNEXT:
<https://www.viewnext.com/watson-funcionalidades/>

vordeo. (2022). Obtenido de vordeo: https://vordeo.com/shop/electronic-components-supplies/module-board/display-screen/nextion-enhanced-nx4832k035-3-5-inch-hmi-intelligent-smart-usart-uart-serial-touch-screen-tft-lcd-module/?gclid=Cj0KCQjwxIOXBhCrARIsAL1QFCaCZ5qOYSez_uTqwTAOx3uLJ5Z1K8OdkWFa

yaesta.com. (2022). Obtenido de yaesta.com: <https://www.yaesta.com/camara-web-1080p-hd-streaming-usb/p>

ANEXOS

Hojas de datos de la Raspberry PI 4



DESCRIPCIÓN GENERAL

Raspberry Pi 4 Model B es el último producto de la popular gama de ordenadores Raspberry Pi. Presenta un incremento espectacular de la velocidad del procesador, el rendimiento multimedia, la memoria y la conectividad en comparación con la Raspberry Pi 3 Model B+ de la generación anterior, todo ello sin perder la compatibilidad con versiones anteriores y manteniendo un consumo de energía similar. Para el usuario final, la Raspberry Pi 4 Model B ofrece el rendimiento de un equipo de sobremesa comparable al de los sistemas PC x86 de nivel básico.

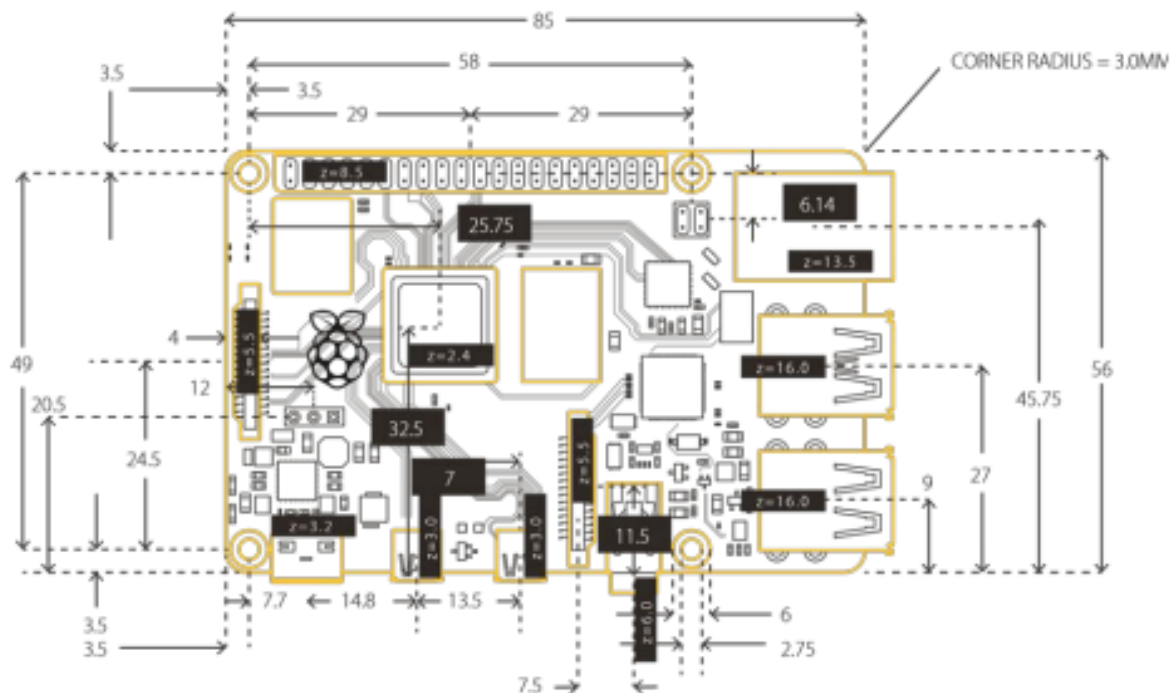
Entre las principales características de este producto, destaca su procesador de cuatro núcleos y 64 bits, su compatibilidad con dos pantallas en resoluciones de hasta 4K a través de los dos puertos micro HDMI, su hardware para decodificación de vídeo de hasta 4Kp60, hasta 4 GB de RAM, LAN inalámbrica de doble banda 2,4/5,0 GHz, Bluetooth 5.0, Gigabit Ethernet, USB 3.0 y compatibilidad con PoE (a través de un complemento HAT para PoE independiente).

La LAN inalámbrica de doble banda y el Bluetooth cuentan con certificación de conformidad modular, lo que permite incorporar la placa al diseño de productos finales con un número de pruebas de conformidad bastante menor, lo que mejora tanto el coste como el tiempo de comercialización.

ESPECIFICACIONES

Procesador:	Broadcom BCM2711, Cortex-A72 de cuatro núcleos (ARM v8) con SoC de 64 bits a 1,5 GHz
Memoria:	LPDDR4 de 1 GB, 2 GB o 4 GB (según el modelo)
Conectividad:	inalámbrica IEEE 802.11b/g/n/ac de 2,4 GHz y 5,0 GHz LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 puertos USB 3.0 2 puertos USB 2.0
GPIO:	Conector macho GPIO de 40 contactos estándar (completamente compatible con versiones de placas anteriores)
Vídeo y sonido:	2 puertos micro HDMI (compatibilidad con hasta 4Kp60) Puerto para pantallas MIPI DSI de 2 vías Puerto para cámaras MIPI CSI de 2 vías Puerto para vídeo compuesto y audio estéreo de 4 polos
Multimedia:	H.265 (decodificación en 4Kp60) H.264 (decodificación en 1080p60, codificación en 1080p30), OpenGL ES, gráficos 3.0
Compatibilidad con tarjetas SD:	Ranura para tarjetas microSD para la carga del sistema operativo y el almacenamiento de datos
Alimentación de entrada:	Dc de 5 V a través de un conector USB-C (mínimo 3 A*) Dc de 5 V a través de un conector macho GPIO (mínimo 3 A*) Compatible con alimentación a través de Ethernet (PoE)
Entorno:	(requiere un HAT para PoE independiente)
Conformidad:	Temperatura de funcionamiento: 0-50 °C
Vida útil de producción:	Para consultar la lista completa de productos locales y regionales homologados, visita https://www.raspberrypi.org/documentation/hardware/raspberrypi/conformity.md La Raspberry Pi 4 Model B se continuará produciendo como mínimo hasta enero de 2026.

ESPECIFICACIONES FÍSICAS



Advertencias

- Este producto solo debe conectarse a una fuente de alimentación externa con una potencia nominal de 5 V/3 A dc o 5,1 V/3 A dc como mínimo¹. Cualquier fuente de alimentación externa que se utilice con la Raspberry Pi 4 Model B deberá cumplir las normativas y estándares aplicables en el país donde se va a utilizar.
- Este producto debe utilizarse en un entorno con buena ventilación y, si se utiliza dentro de una carcasa, esta no debe estar cubierta. El producto debe colocarse sobre una superficie no conductora, estable y plana, y no debe entrar en contacto con elementos conductores.
- La conexión de dispositivos incompatibles al conector GPIO puede afectar a la conformidad, provocar daños en la unidad e invalidar la garantía.
- Todos los periféricos que se utilicen con este producto deben cumplir los estándares pertinentes del país donde se van a usar y marcarse correctamente para garantizar el cumplimiento de los requisitos de seguridad y rendimiento. Estos artículos incluyen, entre otros, los teclados, monitores y ratones que se utilicen con la Raspberry Pi.
- Cuando se conecten periféricos que no incluyan cable o conector, el cable o conector utilizados deben presentar un aislamiento y funcionamiento adecuados para asegurar que se cumplen los requisitos de rendimiento y seguridad pertinentes.

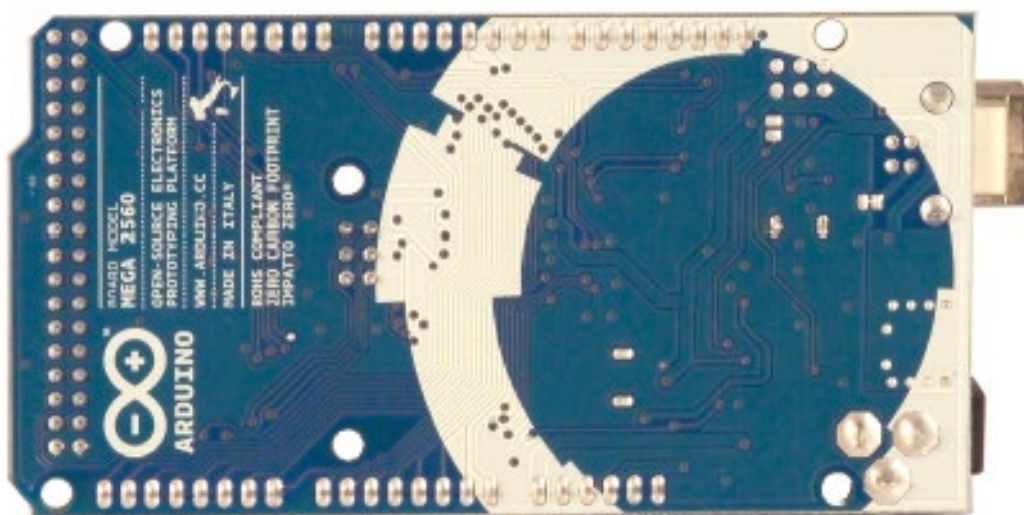
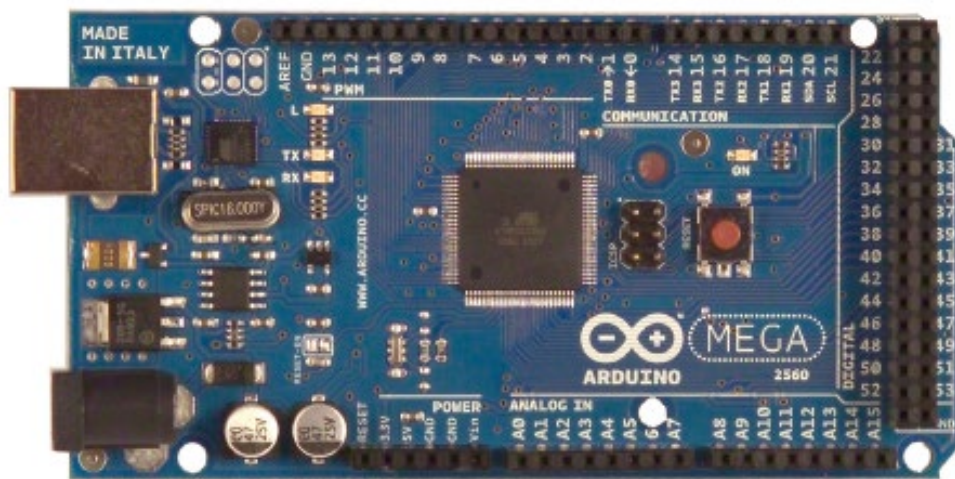
Instrucciones de seguridad

- El dispositivo no debe exponerse al agua o a la humedad ni colocarse sobre una superficie conductora cuando esté en funcionamiento.
- El dispositivo no debe exponerse a fuentes de calor. La Raspberry Pi 4 Model B se ha diseñado para ofrecer un funcionamiento fiable a temperaturas ambiente normales.

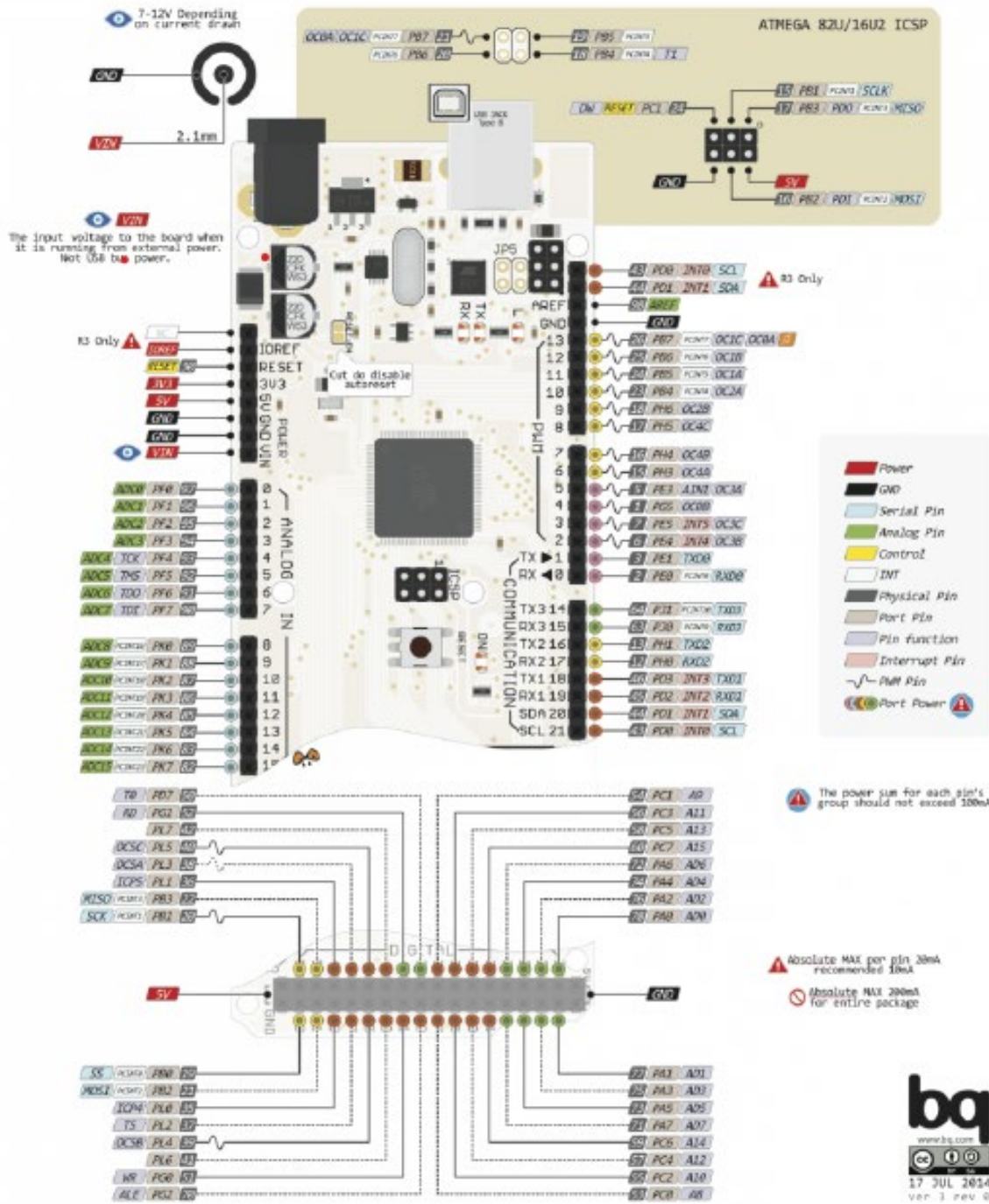
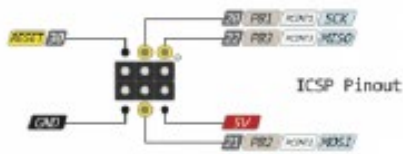
¹ Se puede utilizar una fuente de alimentación de 2,5 A de buena calidad si los periféricos USB conectados a la placa consumen menos de 500 mA en total.

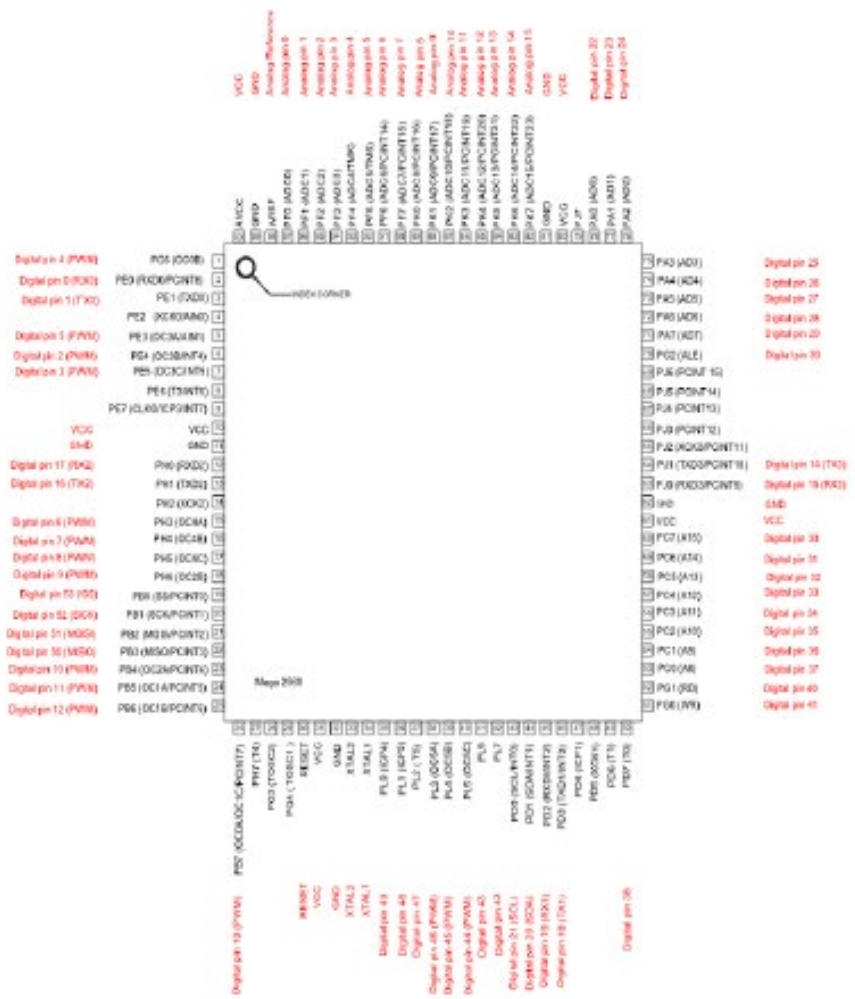
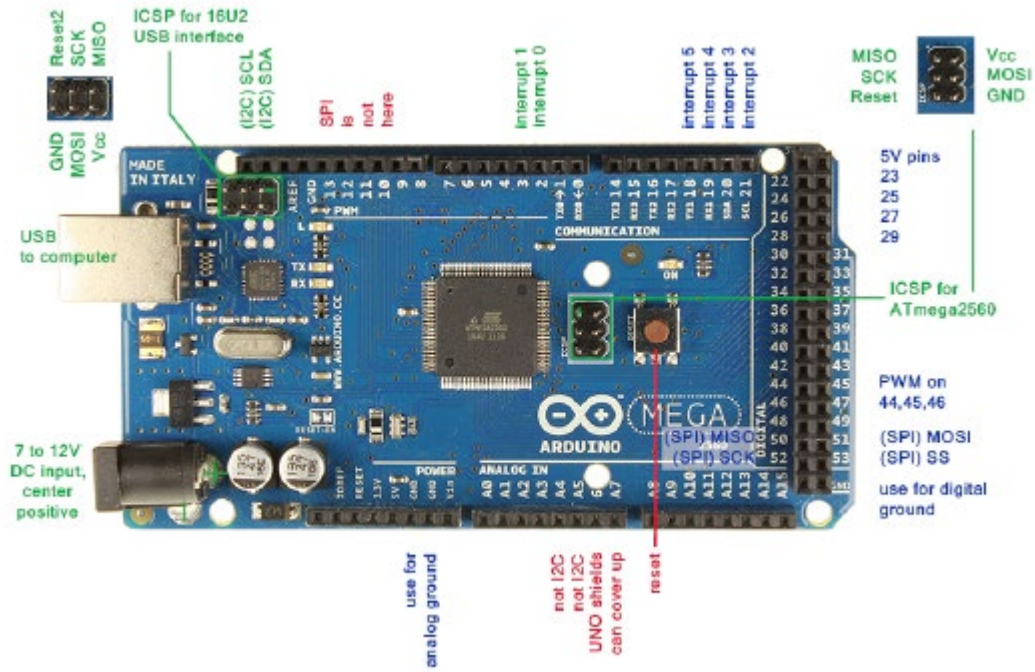


Arduino Mega 2560 Datasheet



MEGA PINOUT





Hoja de datos Pantalla HMI

NX8048T050

Descripción general

Modelos Nextion

Especificaciones

Características Electrónicas

Ambiente de trabajo y parámetro de confiabilidad

Rendimiento de las interfaces

Características de la memoria

Dimensiones del producto



Descripción general

Nextion es una solución perfecta de interfaz hombre-máquina (HMI) que proporciona una interfaz de control y visualización entre un ser humano y un proceso, máquina, aplicación o dispositivo. Nextion se aplica principalmente al IoT o al campo de la electrónica de consumo. Es la mejor solución para reemplazar el tradicional tubo LCD y LED Nixie. Con el software Nextion Editor (descarga oficial), los usuarios pueden crear y diseñar sus propias interfaces para la pantalla Nextion.

El paquete incluye: pantalla Nextion, cable de conexión, placa de prueba de fuente de alimentación.

Vaya de compras: NX8048T050 (IM150416006)

Nota: la pequeña placa de prueba de la fuente de alimentación y el cable de conexión dentro del paquete le permiten comprobar si la fuente de alimentación es suficiente o no. Vea la imagen a continuación sobre cómo usarlo.

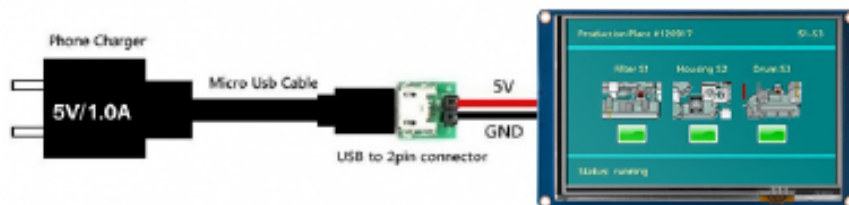
Caution:

Working under insufficient power supply condition will damage the Nextion model easily.

Blurred screen? Flashing? You may be suffering from power shortages. Power off at the first possible moment. No more repeated attempts to damage your Nextion model.

A small connector is included in the package. Please try to power Nextion with your phone charger through the connector to check if Nextion works well.

A high quality usb cable is required.



Modelos Nextion

Tipo de siguiente	Serie básica
Modelos Nextion	NX8048T050_011N (N: sin contacto)
	NX8048T050_011R (R: pantalla táctil resistiva)





Especificaciones

	Datos	Descripción
Color	64K 65536 colores	16 bits 565, 5R-6G-5B
Tamaño del diseño	133,5 (largo) × 84 (ancho) × 5 (alto)	NX8048T050_011N
	133,5 (largo) × 84 (ancho) × 6,2 (alto)	NX8048T050_011R
Área Activa (AA)	118,50 mm (largo) × 77,55 mm (ancho)	
Área visual (VA)	108 mm (largo) × 64,8 mm (ancho)	
Resolución	800×480 píxeles	También se puede configurar como 480 × 800
Tipo de toque	Resistador	
toques	> 1 millón	
Iluminar desde el fondo	CONDUJO	
Vida útil de la retroiluminación (promedio)	>30,000 Horas	
Brillo	250 nits (NX8048T050_011N)	0% a 100%, el intervalo de ajuste es 1%
	230 nits (NX8048T050_011R)	0% a 100%, el intervalo de ajuste es 1%
Peso	108 g (NX8048T050_011N)	
	126 g (NX8048T050_011R)	



Logitech®

HD Webcam C270

Videoconferencias en alta definición fáciles para todos

- Videoconferencias HD de 720p en la mayoría de aplicaciones de mensajería instantánea y Logitech Vid™ HD
- Fotos de 3 megapíxeles (mejora por software)
- Tecnología Logitech® RightLight™
- Tecnología Logitech® RightSound™



Logitech® HD Webcam C270. Videoconferencias en alta definición fáciles para todos. Disfrutará de videoconferencias HD de 720p en la mayoría de las aplicaciones de mensajería instantánea y en Logitech Vid™ HD, el medio gratis, rápido y fácil para mantener el contacto visual con sus seres queridos. Mande una foto a sus familiares y amistades cuando no tenga tiempo para hablar o grabar un video. Haga una foto de 3 megapíxeles (mejora por software) con un solo clic. La cámara Web se ajusta a las malas condiciones de iluminación de manera automática para producir la mejor imagen posible, gracias a la tecnología Logitech RightLight™. Además, el ruido de fondo no tiene por qué estropear sus videoconferencias. La tecnología Logitech RightSound™ produce conversaciones nítidas, para que puedan oírle.



REQUISITOS MÍNIMOS DEL SISTEMA

Windows® XP (SP2 o superior), Windows Vista® o Windows® 7 (32 bits o 64 bits)

Requisitos básicos:

- 1 GHz
- 512 MB de RAM o más
- Disco duro con 200 MB de espacio libre
- Conexión a Internet
- Puerto USB 1.1 disponible (se recomienda 2.0)

Para videoconferencias y grabaciones HD fluidas de 720p:

- Intel® Core™2 Duo a 2,4 GHz
- 2 GB de RAM
- Disco duro con 200 MB de espacio libre
- Puerto USB 2.0
- Velocidad de carga de 1 Mbps o superior
- Resolución de pantalla de 1280 x 720



Logitech Vid™



Logitech® HD Webcam C270



ESPECIFICACIONES TÉCNICAS

- Videoconferencias HD (1280 x 720 píxeles) con el sistema recomendado
- Captura de video: hasta 1280 x 720 píxeles
- Fotos: hasta 3 megapíxeles (mejora por software)
- Micrófono integrado con tecnología Logitech RightSound™
- Certificación USB 2.0 de alta velocidad (se recomienda)
- Clip universal para monitores LCD, CRT o portátiles

Software de cámara Web Logitech:

- Logitech Vid™ HD
- Tecnología Logitech RightLight™
- Captura de video y fotos

Funciona con la mayoría de las aplicaciones de mensajería instantánea

ESPECIFICACIONES DE PAQUETE

	Primary Pack (producto individual)	Master Shipper (paquete completo)
Nº de referencia WEB	960-000582	n/a
Código de barras	5099206023758 (EAN-13)	50992060237517 (SCC-14)
Nº de referencia CENTRAL	960-000635	n/a
Código de barras	5099206023802 (EAN-13)	50992060238019 (SCC-14)
Nº de referencia CSR	960-000636	n/a
Código de barras	5099206023819 (EAN-13)	50992060238118 (SCC-14)
Peso	226.8g	2.147 kg
Longitud	20.95cm	31.50cm
Ancho	7.62 cm	31.06cm
Alto	15.24 cm	22.40 cm
Volumen	2.433dm ³	0.02230m ³
1 Primary Pack (producto individual)	1	n/a
1 Intermediate Pack (paquete de cartón)	0	n/a
1 Master Shipper (caja de cartón)	8	1
1 EUR Pallet	432	54
1 Container de 6m	10080	1260
1 Container de 12m	21280	2660
1 Container de 12m HQ	23408	2926

ESTA CAJA CONTIENE

- Cámara Web con cable de 1,5 m
- Auriculares
- Software de cámara Web Logitech con Logitech Vid™ HD
- Documentación del usuario

Logitech® Vid™. El medio gratis, rápido y fácil para realizar videoconferencias con su cámara Web Logitech.

www.logitech.com

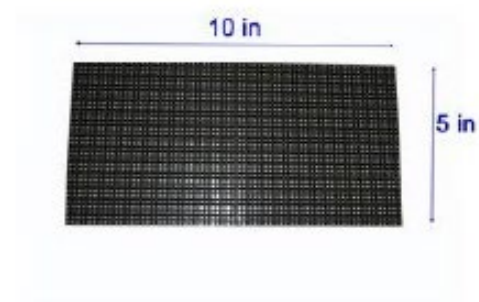
©2010 Logitech. Logitech, el logotipo de Logitech y las demás marcas de Logitech son propiedad de Logitech y pueden estar registradas. Las demás marcas comerciales pertenecen a sus respectivos propietarios.

Hoja de datos Matriz de Leds

Indoor LED matrix module
Not waterproof!
Size: 256x128 mm(10x5 in)
Resolution:64 x 32
Pixel Pitch: 4 mm
Controller Interface: HUB75(See picture 2 for pin definition)
Drive IC: Universal constant current
Input voltage: DC 5V
Scan pattern 1/16 scan
Max brightness: 1000 nits
Max power consumption: 20 W
Supports Open source project from GitHub:
PxMatrix - LED matrix panel driver for ESP8266, ESP32 and ATMEL

Package list:

1. LED matrix module
2. Power wire
3. Data cable



Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

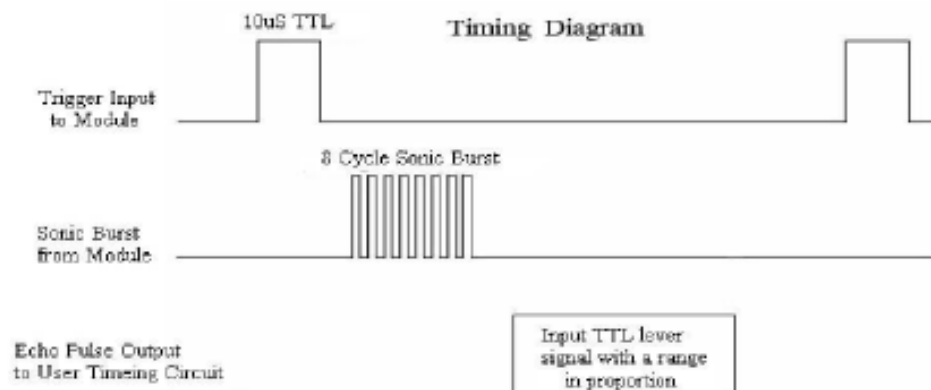
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 μ s pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{s} / 58 = \text{centimeters}$ or $\mu\text{s} / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Código Raspberry Principal

```
import RPi.GPIO as GPIO
import time
import cv2
import sqlite3
import socket
import threading
import serial
import xlsxwriter
from datetime import date
import json
import subprocess
import os

archivo = '/home/bus1/Documents/datos.json'

# Cambiar los permisos del archivo a lectura y escritura para todos los usuarios
os.chmod(archivo, 0o666)

# Ejecuta el comando para establecer los permisos del puerto
subprocess.run(['sudo', 'chmod', 'a+rw', '/dev/ttyS0'])
countG=0#perosnas que estan en el bus
count=0#personas que ingresaron
count2=0#personas que salieron
aforo1=50#aforo
gent=0#acientos disponibles
puerto_serie = serial.Serial('/dev/ttyACM0', 9600)#matrix 1

# Configuración del servidor
host = '169.254.212.26' # Dirección IP de la Raspberry Pi servidor
port = 8000 # Puerto para la comunicación
# Crear un socket TCP/IP
servidor = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# variable global acumulador de datos del socket
datos= ""
# Vincular el socket al host y puerto

# Conectar a la base de datos (o crearla si no existe)
conn = sqlite3.connect('autobus.db')

# Crear un cursor
c = conn.cursor()

# Crear tabla
c.execute('''
    CREATE TABLE IF NOT EXISTS pasajeros (
```



```

        id INTEGER PRIMARY KEY,
        timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
        personas_en_el_bus INTEGER,
        personas_al_dia INTEGER,
        aforo TEXT)
'''

# Configuración del sensor de ultrasonido
GPIO.setmode(GPIO.BCM)
TRIG = 23
ECHO = 24
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)

#led de aforo
AFOR = 19
GPIO.setup(AFOR,GPIO.OUT)
#Led deteccion cam1
CAM1 = 20
GPIO.setup(CAM1,GPIO.OUT)
# Configuración de la detección de rostros
face_cascade =
cv2.CascadeClassifier('/home/bus1/Documents/haarcascade_frontalface_alt.xml')
video_capture = cv2.VideoCapture(0)

# Contador de personas
count = 0
count2 = 0
countG = 0

# Variable de bloqueo
lock = False

def jsondato():
    global aforo1
    # Abre el archivo json para lectura
    with open('/home/bus1/Documents/datos.json', 'r') as f:
        data = json.load(f) # Carga los datos del archivo

    # Supongamos que el archivo contiene un objeto json similar a este:
    {"dato": "123"}
    numero_str = data['dato1'] # Accede al dato que quieres (aquí suponemos
que la clave es 'dato')

    # Convierte la cadena de texto a un entero
    aforo1 = int(numero_str)

    print(aforo1) # Imprime 123

```

```
jsondato()
```

```
def task1():
    host = '169.254.212.26' # Dirección IP de la Raspberry Pi servidor
    port = 8000 # Puerto para la comunicación

    while True:
        try:
            servidor.bind((host, port))
            break
        except socket.error as e:
            if e.errno == 98 or e.errno == 48: # Address already in use
                print("El puerto está en uso. Esperando y reintentando en 5
segundos...")
                time.sleep(5)
            else:
                raise
    #servidor.bind((host, port))
    servidor.listen(1)
    global datos
    global count2
    while True:
        cliente, direccion = servidor.accept()
        print('Conexión establecida desde:', direccion)
        datos = cliente.recv(1024).decode()
        print(datos)

        if datos == 'A':
            count2+=1
            print(count2)
```

```
thread1 = threading.Thread(target=task1)
```

```
def task2():

    # Abre la comunicación serial
    ser = serial.Serial(
        port='/dev/ttyS0', # Cambia esto al puerto correcto si es necesario
        baudrate = 9600, # Cambia esto a la velocidad en baudios correcta si
es necesario

    )
    escape='\xff'.encode('iso-8859-1')
    # Define los valores iniciales de los contadores
    contador1 = 0
    contador2 = 0
    contador3 = 0
```

```

global countG
global count
global count2
global aforo1

try:
    while True:
        # Incrementa los contadores
        contador1 += 1
        contador2 += 1
        contador3 += 1

        # Construye los comandos
        comando1 = 'ContG.val=' + str(countG)
        comando2 = 'Cont.val=' + str(count)
        comando3 = 'Cont2.val=' + str(count2)
        comando4 = 'Aforo.val=' + str(aforo1)

        print(comando1)

        # Envia los comandos a la pantalla Nextion
        ser.write(comando1.encode())
        ser.write(escape)
        ser.write(escape)
        ser.write(escape)
        ser.flush()
        time.sleep(0.5)

        ser.write(comando2.encode())
        ser.write(escape)
        ser.write(escape)
        ser.write(escape)
        ser.flush()
        time.sleep(0.5)

        ser.write(comando3.encode())
        ser.write(escape)
        ser.write(escape)
        ser.write(escape)
        ser.flush()
        time.sleep(0.5)

        ser.write(comando4.encode())
        ser.write(escape)
        ser.write(escape)
        ser.write(escape)
        ser.flush()
        time.sleep(0.5)

```

```

        # Espera un segundo antes de la próxima actualización
        ser.close()
        time.sleep(1)
        ser.open()
    except KeyboardInterrupt:
        # Cuando se presiona Ctrl+C, cierra la conexión serial
        ser.close()

thread2 = threading.Thread(target=task2)

if not thread1.is_alive():
    # Iniciar el hilo
    thread1.start()
else:
    print("El hilo ya ha sido iniciado")

thread2 = threading.Thread(target=task2)

if not thread2.is_alive():
    # Iniciar el hilo
    thread2.start()
else:
    print("El hilo ya ha sido iniciado")

def aforo():
    global countG
    global aforo1
    if(countG>=aforo1):
        print("aforo lleno")
        countG=aforo1
        GPIO.output(AFOR, False)
    else:
        GPIO.output(AFOR, True)

def generar_excel():
    conn = sqlite3.connect('autobus.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM pasajeros")
    rows = cursor.fetchall()

    # Generar nombre del archivo con la fecha actual
    today = date.today()
    filename = today.strftime("%Y-%m-%d") + '.xlsx'

    # Crear un nuevo libro de trabajo y añadir una hoja
    workbook = xlswriter.Workbook(filename)

```

```

worksheet = workbook.add_worksheet()

# Escribir la cabecera
for i, column_name in enumerate(cursor.description):
    worksheet.write(0, i, column_name[0])

# Escribir los datos
for i, row in enumerate(rows):
    for j, value in enumerate(row):
        worksheet.write(i+1, j, value)

# Cerrar el libro de trabajo
workbook.close()

# Cerrar la conexión a la base de datos
conn.close()

while True:

    #Verificar la presencia de una persona mediante el sensor de ultrasonido
    GPIO.output(TRIG, False)
    time.sleep(2)

    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)

    while GPIO.input(ECHO)==0:
        pulse_start = time.time()

    while GPIO.input(ECHO)==1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start

    distance = pulse_duration * 17150
    distance = round(distance, 2)
    #print(distance)

    if distance <= 100: # Si la distancia es menor a un metro
        # Intentar detectar un rostro
        ret, frame = video_capture.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))

```

```

        # Si se ha detectado un rostro y no estamos en estado de bloqueo,
        incrementar el contador y agregarlo a la base de datos
        if len(faces) > 0 and not lock:
            count += 1

            #print('llego')

            # Conectar a la base de datos
            conn = sqlite3.connect('autobus.db')
            c = conn.cursor()

            # Insertar un nuevo registro
            c.execute("INSERT INTO pasajeros (personas_en_el_bus,
            personas_al_dia, aforo) VALUES (?, ?, ?)", (countG, count, aforo1))
            # Guardar (commit) los cambios
            conn.commit()
            # Cerrar la conexión a la base de datos
            conn.close()
            # Activar el bloqueo
            lock = True
            GPIO.output(CAM1, False)

        elif len(faces) == 0:
            # Si no se detecta ningún rostro, desactivar el bloqueo
            lock = False
            GPIO.output(CAM1, True)

countG =count-count2
if countG<0:
    countG=0
gent=aforo1-countG
if gent<0:
    gent=0

#print(f"Personas en el bus: {countG}")
mensaje= str(gent) + " de " + str(aforo1)
print(gent)
puerto_serie.write(str(mensaje).encode())
aforo()
generar_excel()

GPIO.cleanup()
video_capture.release()
cv2.destroyAllWindows()

```

Código Raspberry Secundario

```
import RPi.GPIO as GPIO
import time
import cv2
import sqlite3
import socket

host="169.254.212.26"
port=8000
def start_client():
    while True:
        try:
            cliente = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            cliente.connect((host, port))
            cliente.send(b'A')
            break
        except socket.error as e:
            print("Error de conexión:", e)
            print("Reintentando en 5 segundos...")
            cliente.close()
            time.sleep(5)

    print("Conexión establecida con el servidor")

    # Lógica del cliente...

    # Cerrar la conexión
    cliente.close()
# Configuración del sensor de ultrasonido
GPIO.setmode(GPIO.BCM)
TRIG = 23
ECHO = 24
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)

# Led cam2
CAM2 = 20
GPIO.setup(CAM2,GPIO.OUT)

# Configuración de la detección de rostros
face_cascade =
cv2.CascadeClassifier('/home/bus2/Documentos/haarcascade_frontalface_alt.xml')
video_capture = cv2.VideoCapture(0) # cámara para las personas que bajan

# Contador de personas
count = 0
```

```

# Variables de bloqueo
lock = False

while True:
    # Verificar la presencia de una persona mediante el sensor de ultrasonido
    GPIO.output(TRIG, False)
    time.sleep(2)

    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)

    while GPIO.input(ECHO)==0:
        pulse_start = time.time()

    while GPIO.input(ECHO)==1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start

    distance = pulse_duration * 17150
    distance = round(distance, 2)
    print(distance)

    if distance <= 100: # Si la distancia es menor a un metro
        # Intentar detectar un rostro
        ret, frame = video_capture.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))
        print("llego")
        # Si se ha detectado un rostro y no estamos en estado de bloqueo,
decrementar el contador
        if len(faces) > 0 and not lock:
            count -= 1
            lock = True
            GPIO.output(CAM2, False)
            time.sleep(2)
            GPIO.output(CAM2, True)
            # Envía un mensaje a la otra Raspberry Pi para que disminuya su
contador
            print("detecto")
            #sockets()
            start_client()
            time.sleep(1)
        elif len(faces) == 0:
            lock = False

```



```
print("final")
```

Código App cambio de aforo y rutas

```
# Importa los módulos necesarios
import sys
import json
import serial
from PyQt5.QtWidgets import QApplication, QMainWindow, QPushButton, QTextEdit

puerto_serie = serial.Serial('/dev/ttyUSB0', 9600)#matrix 1

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setGeometry(50, 50, 500, 600)
        self.setWindowTitle("APP SETUP CAB")

        self.textEdit1 = QTextEdit(self)
        self.textEdit1.setGeometry(50, 50, 200, 50)

        lf.button = QPushButton("Enviar AFORO", self)
        self.button.setGeometry(290, 50, 200, 50)
        self.button.clicked.connect(self.button_clicked)

        self.textEdit2 = QTextEdit(self)
        self.textEdit2.setGeometry(50, 250, 200, 50)

        self.textEdit3 = QTextEdit(self)
        self.textEdit3.setGeometry(50, 340, 200, 50)

        self.textEdit4 = QTextEdit(self)
        self.textEdit4.setGeometry(50, 430, 200, 50)

        self.textEdit5 = QTextEdit(self)
        self.textEdit5.setGeometry(50, 520, 200, 50)

        self.button2 = QPushButton("Cambio Rutas", self)
        self.button2.setGeometry(290, 250, 200, 50)
        self.button2.clicked.connect(self.button2_clicked)

    def button_clicked(self):
        data = {
            'dato1': self.textEdit1.toPlainText()
```

```

    }
    with open('datos.json', 'w') as f:
        f.write(json.dumps(data))

def button2_clicked(self):
    var= self.textEdit2.toPlainText()
    var2= self.textEdit3.toPlainText()
    var3= self.textEdit4.toPlainText()
    var4= self.textEdit5.toPlainText()
    print(var)
    mensaje= str(var)+'\n'+str(var2)+'\n'+str(var3)+'\n'+str(var4)
    puerto_serie.write(str(mensaje).encode())

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = MainWindow()
    window.show()
    sys.exit(app.exec_())

```

Código Arduino Mega Aforo

```

#include <DFRobot_RGBMatrix.h>
// Hardware-specific library

//variables for all DFRobot RGB Matrix programs
#define OE      9
#define LAT     10
#define CLK     11
#define A       A0
#define B       A1
#define C       A2
#define D       A3
#define E       A4
#define WIDTH   64
#define HEIGHT  32

DFRobot_RGBMatrix matrix(A, B, C, D, E, CLK, LAT, OE, false, WIDTH, HEIGHT);

uint16_t textnumero,textmensaje;//the text color
const uint16_t BACK = matrix.Color333(0, 0, 0);//the background color is black
String texto;
String dis="Espacios";
String res="restantes";
int cap;

void setup() {
    matrix.begin();

```

```

textnumero = matrix.Color333(7, 7, 7); //set text color to white
textmensaje = matrix.Color333(4, 0, 7); //set text color to white
// fill with the background color
matrix.fillScreen(BACK);
// format text and display before starting motion
Serial.begin (9600);
delay(2000);
pinMode(8,OUTPUT);
pinMode(7,OUTPUT);
pinMode(6,OUTPUT);
pinMode(5,OUTPUT);
}

void loop() {
  //clear previous text
  if(Serial.available() > 0){
    texto=Serial.readString();
    cap=texto.toInt();
    Serial.println(cap);
  }
  if(cap==0)
  {
    matrix.fillScreen(BACK);
    //matrix.setTextSize(2);
    matrix.setCursor(1, 4);
    matrix.setTextColor(textnumero);
    matrix.print("No Existen");
    matrix.setCursor(2,16);
    matrix.setTextColor(textmensaje);
    matrix.println(dis);
    matrix.setCursor(2,24);
    matrix.setTextColor(textmensaje);
    matrix.println(res);
  }
  else
  {
    matrix.fillScreen(BACK);
    //matrix.setTextSize(2);
    matrix.setCursor(4, 4);
    matrix.setTextColor(textnumero);
    matrix.print(texto);
    matrix.setCursor(2,16);
    matrix.setTextColor(textmensaje);
    matrix.println(dis);
    matrix.setCursor(2,24);
    matrix.setTextColor(textmensaje);
    matrix.println(res);
  }
}

```

```

if (cap==0)
{
    digitalWrite(8, HIGH);
    digitalWrite(7, HIGH);
    digitalWrite(6, LOW);
    digitalWrite(5, LOW);
}
else
{
    digitalWrite(8, LOW);
    digitalWrite(7, LOW);
    digitalWrite(6, HIGH);
    digitalWrite(5, HIGH);
}
}
}

```

Código Arduino Mega Rutas

```

#include <DFRobot_RGBMatrix.h>
#include <Separador.h>
// Hardware-specific library

//variables for all DFRobot RGB Matrix programs
#define OE      9
#define LAT    10
#define CLK    11
#define A      A0
#define B      A1
#define C      A2
#define D      A3

#define E      A4
#define WIDTH  64
#define HEIGHT 32

DFRobot_RGBMatrix matrix(A, B, C, D, E, CLK, LAT, OE, false, WIDTH, HEIGHT);

uint16_t text1,text2,text3,text4;//the text color
const uint16_t BACK = matrix.Color333(0, 0, 0);//the background color is black
String d1="JATUNRUMMI";
String d2="ATUNTAQUI";
String d3="MERCADO";
String d4="DOLOROSA";

Separador s;
String datos;
//String d1,d2,d3,d4;

```

```

void setup() {
  matrix.begin();
  text1 = matrix.Color333(0, 0, 7); //set text color
  text2 = matrix.Color333(0, 7, 0); //set text color
  text3 = matrix.Color333(7, 0, 0); //set text color
  text4 = matrix.Color333(0, 7, 7); //set text color

  // fill with the background color
  matrix.fillScreen(BACK);
  // format text and display before starting motion
  Serial.begin (9600);
  delay(2000);
}

void loop() {
  if(Serial.available())
  {
  datos= Serial.readString();
  d1= s.separa(datos, '-',0);
  d2= s.separa(datos, '-',1);
  d3= s.separa(datos, '-',2);
  d4= s.separa(datos, '-',3);

  Serial.println("d1 "+ d1);
  Serial.println("d2 "+ d2);
  Serial.println("d3 "+ d3);
  Serial.println("d4 "+ d4);

  }

  //change color for next text
  matrix.fillScreen(BACK);
  //matrix.setTextSize(2);
  matrix.setCursor(0, 0);
  matrix.setTextColor(text1);
  matrix.println(d1);
  matrix.setCursor(0, 8);
  matrix.setTextColor(text2);
  matrix.println(d2);
  matrix.setCursor(0, 16);
  matrix.setTextColor(text3);
  matrix.println(d3);
  matrix.setCursor(0, 24);
  matrix.setTextColor(text4);
  matrix.println(d4);
  delay(3000);
}

```

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERIA EN CIENCIAS APLICAS
CARRERA DE ELECTRICIDAD



MANUAL DE USUARIO DE LA IMPLEMENTACIÓN DE UN PROTOTIPO
DE SISTEMA DE CONTEO DE PASAJERO EN BUSES URBANOS PARA
CONTROL DE AFORO



Objetivo

El presente manual de usuario tiene la finalidad de ser una guía de operación del prototipo, permitiendo al lector de este obtener destrezas y conocimientos necesarios para su funcionamiento y ser una herramienta de consulta a la cual el docente pueda acudir en cualquier momento.

Instrucciones de seguridad



ATENCIÓN: Advertencia y Seguridad.

1. Antes de manipular el dispositivo lea el manual de usuario y consérvelo para futuras consultas.
2. Solicitar la autorización de uso respectiva al docente.
3. Asegúrese que el prototipo se encuentre en un lugar fijo.
4. Limpiar cualquier suciedad que se pueda encontrar sobre o alrededor de los componentes electrónicos.
5. Verificar el estado de conductores y conexiones del prototipo.
6. Ser cuidadoso al momento de manipular el prototipo ya que puede provocar un cortocircuito y dañar ciertos componentes.
7. Verificar el correcto funcionamiento de los equipos de alimentación DC.

Introducción

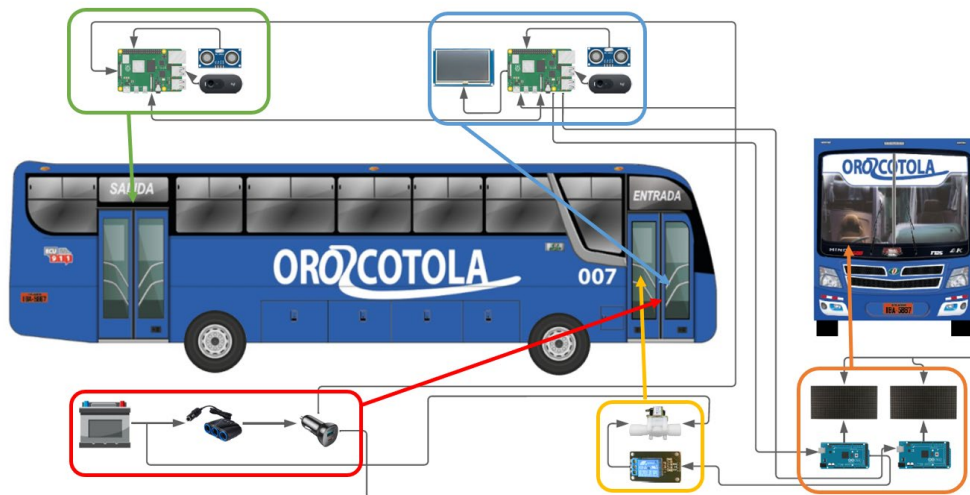
El prototipo de un sistema de conteo en buses urbanos con control de aforo, se diseñó y elaboró con el fin de ser una herramienta, para que los conductores no lleven una cantidad excesiva de personas en el autobús. El manual de usuario fue elaborado como un material de apoyo para la enseñanza de la visión artificial, para las personas que deseen aprender sobre el tema, la estructuración del manual se ha enfocado para lograr orientar el funcionamiento del prototipo mediante la explicación de las conexiones, elementos y demás factores que intervienen en él.

Elementos y distribución en el autobús

En esta sección se describen los componentes y elementos electrónicos que tiene el prototipo, con sus respectivas conexiones, con la finalidad de que el usuario pueda identificar con facilidad cada elemento de esta manera sea más fácil de utilizar y realizar el mantenimiento. En la Figura 1 muestra la distribución de sus respectivos elementos y las conexiones necesarias para su uso en el autobús.

Figura 1

Distribución de compones y elementos en el autobús



Nota. Elaboración Propia

Puerta de entrada

Esta parte del prototipo se encarga de contar a las personas que se van a subir al autobús, mediante la detección con el sensor ultrasonido y la cámara, además cuenta con una pantalla HMI, que muestra el total de pasajeros actuales, total de pasajeros durante todo el día, aforo, estado del autobús, indicadores del estado de la puerta y paro de emergencia.

En la Figura 2 se puede observar como es el diseño de la caja principal que se encuentra a la puerta de entrada.

Figura 2

Caja que se coloca en la puerta de entrada



Nota. Elaboración Propia

A continuación en la Figura 3 se puede observar como se visualiza la pantalla HMI encendida y como muestra el estado del autobús al momento de encenderle.

Figura 3

Interfaz de la pantalla HMI



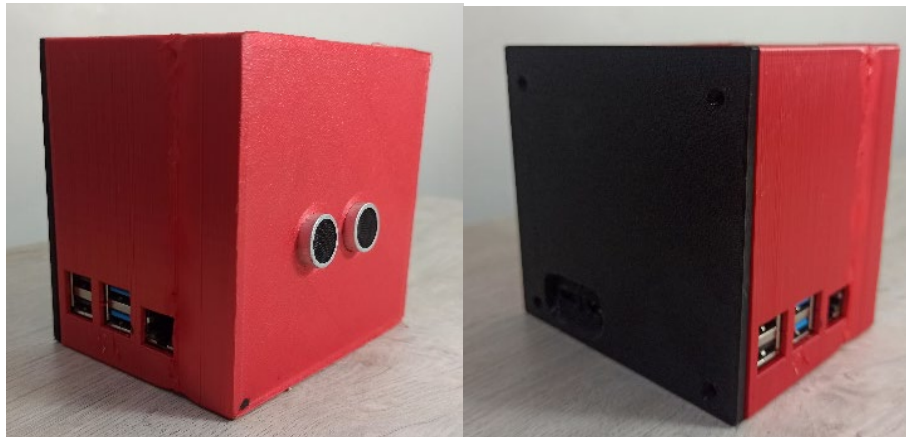
Nota. Elaboración Propia

Puerta de salida

Esta parte del prototipo se encarga de contar a las personas que salen del autobús con la finalidad de tener un monitoreo constante de cuantas personas salen de la unidad de transporte. En la Figura 4 se puede observar cómo es el diseño de la caja secundaria que se encuentra a la puerta de salida.

Figura 4

Caja que se coloca en la puerta de salida



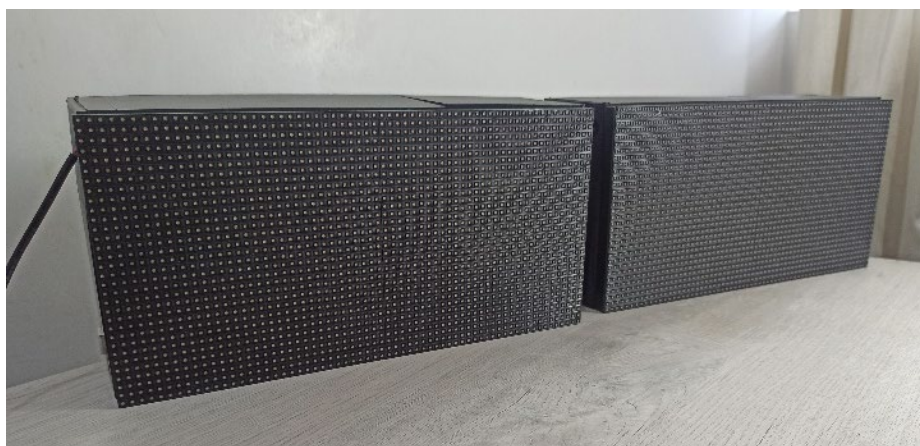
Nota. Elaboración Propia

Matrices Leds

Existen dos matrices leds una se encarga de mostrar la cantidad de personas que están en el autobús y la otra se encarga de mostrar las rutas por donde se movilizara el autobús. En la Figura 5 se puede observar cómo es el diseño de las cajas que indicaran las rutas y aforo.

Figura 6

Cajas indicadoras de rutas y aforo



Nota. Elaboración Propia

A continuación en la Figura 7 se puede observar como se visualiza las Matrices leds encendida y como muestra el estado del autobus al momento de encenderle.

Figura 7

Matrices de leds encendidas



Nota. Elaboración Propia

Bloqueo de puerta

Debido a que el autobús disponía de una válvula de 3 vías, de accionamiento manual para abrir y cerrar la puerta, se instaló una electroválvula para impedir el desfogue del aire y de esta manera bloquear la puerta de ingreso. En la Figura 7 se puede observar que tipo de electroválvula se utilizó para impedir el paso de aire.

Figura 7

Electroválvula utilizada para el bloqueo de la puerta del autobús



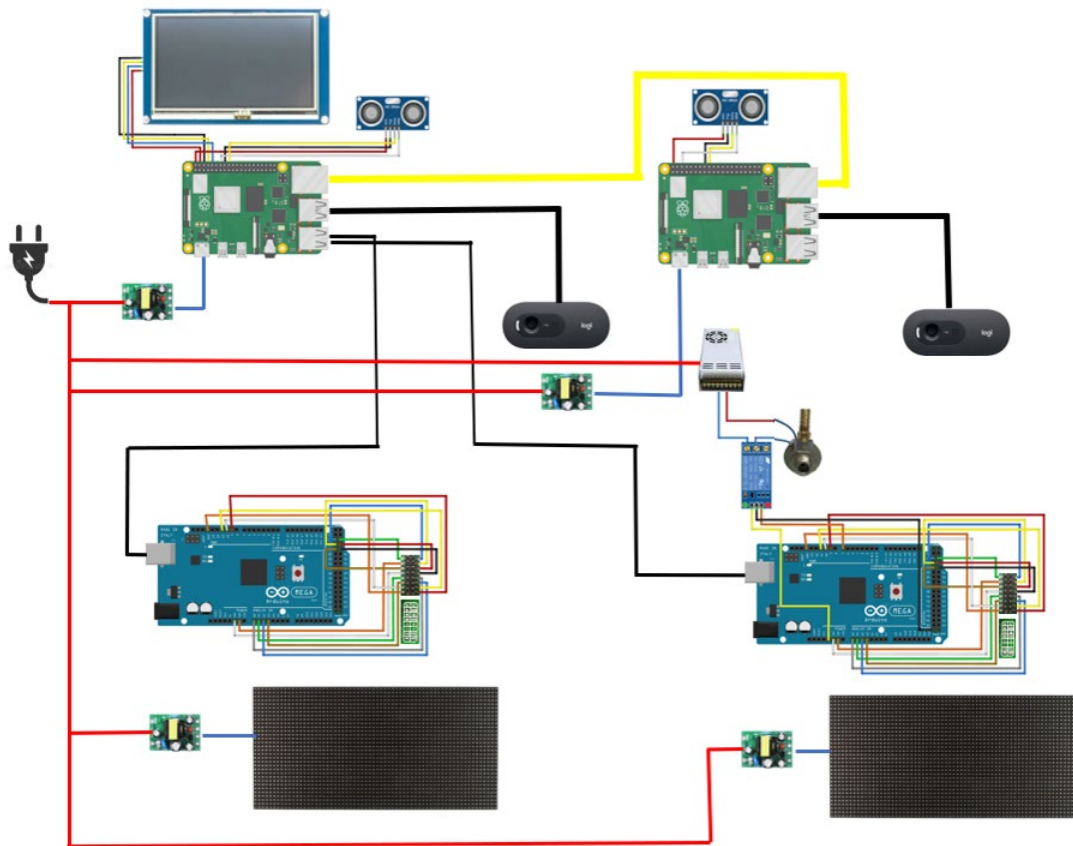
Nota. Elaboración Propia

Diagrama de conexiones general

En este diagrama se puede observar cómo es la conexión de todos los elementos y equipos del prototipo como se observa en la Figura 8

Figura 8

Diagrama de conexiones general



Nota. Elaboración Propia

Encendido del prototipo

Primero se debe energizar el sistema conectándolo a un tomacorriente de 120V. después aplastar el interruptor de encendido que se muestra en la Figura 8, se debe esperar de 1 a 2 minutos hasta que el sistema se ejecute solo y pueda empezar con el monitoreo de las personas que entran y salen de la unidad de transporte.

Figura 8



Para darnos cuenta de que el prototipo esté encendido y funcionando correctamente, se debe observar que en la matriz de leds aparezca “50 de 50”, caso contrario, el prototipo no se pudo ejecutar correctamente, y si sucede eso debemos conectar una pantalla al puerto micro-HDMI, para poder visualizar donde está el error en el prototipo.

Funcionamiento

El funcionamiento de la maqueta es el siguiente: una persona debe ponerse delante de la cámara de 1 a 3 segundos y después retirarse si la luz indicada se enciende, al momento que una persona ponga su rostro delante de la cámara de la puerta de entrada el número que muestra en la matriz de leds se reduce de 50 a 49 dependiendo del número del aforo de la unidad de transporte. De igual forma, cuando una persona ponga su rostro delante de la cámara de la puerta de salida el número de la matriz de leds aumenta de 49 a 50, y en la pantalla HMI mostrara el número total de pasajeros actuales, el total de pasajeros que entraron y el total de pasajeros que salieron, además el aforo del autobús, también el estado del autobús en porcentaje como se muestra la Figura 9.

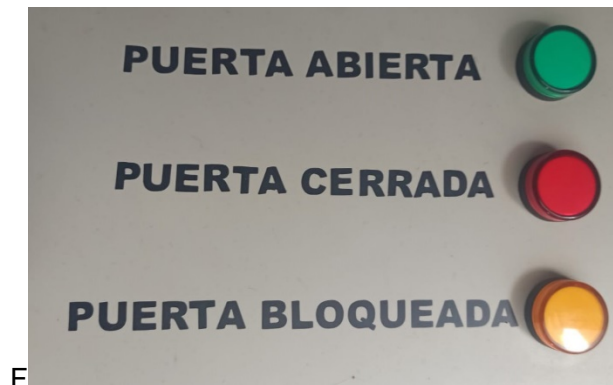
Figura 9



Además, cuando la matriz de leds aparezca el mensaje ya no existen lugares restantes se bloqueará la electroválvula y no registrara mas personas que quieran ingresar al autobús. También existe un paro de emergencia que está es un interruptor que desbloquea la electroválvula que se puede observar en la Figura 8. Estos paros se colocaron solo para aplastarlos en caso de que exista algún tipo de emergencia en el autobús caso contrario no se deben aplastar debido a que el prototipo dejará de funcionar correctamente.

También se muestran 3 luces indicadoras de color verde rojo y amarillalo como se puede observar en la Figura 10. Estas luces se pueden encender con el interruptor de accionamiento de la puerta que se muestra en la Figura 8.

Figura 10



Nota. Elaboración Propia

Por último, existe el accionamiento del cilindro neumático como se puede apreciar en la Figura 11, este accionamiento solo permite abrir o cerrar la válvula solo cuando existen lugares disponibles caso contrario la electroválvula se bloquea y no permite el paso del aire para que el accionamiento funcione. También en la Figura 8 está el encendido del compresor para efectuar el funcionamiento de este sistema de aire.

Figura 11



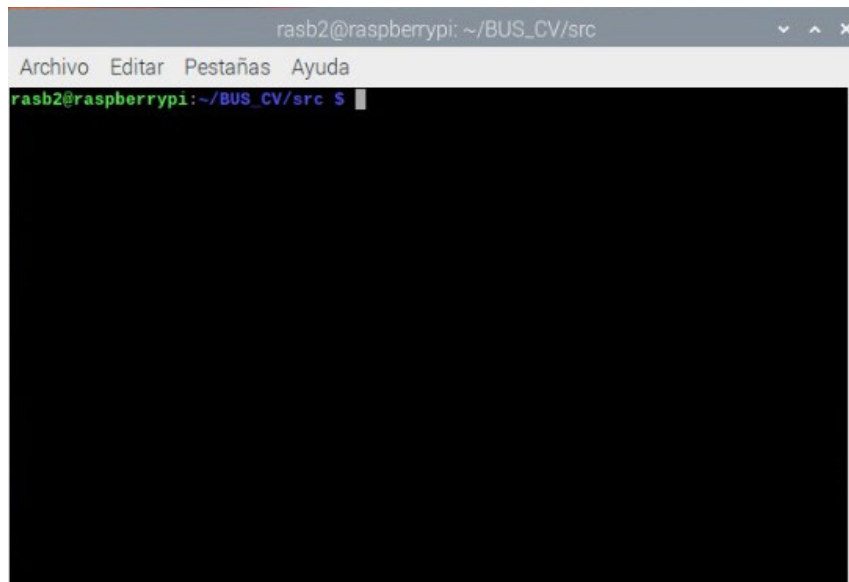
Nota. Elaboración Propia

Cambio del aforo y rutas

Para poder realizar el cambio de aforo se debe conectar una pantalla al puerto micro HDMI, un mouse y un teclado a los puertos USB de las raspberry pi 4 de la puerta de entrada, una vez conectado se enciende el prototipo, cuando el prototipo esté encendido se podrá observar una pantalla negra como se observa en la Figura 12.

Figura 12

Pantalla negra para escribir comandos

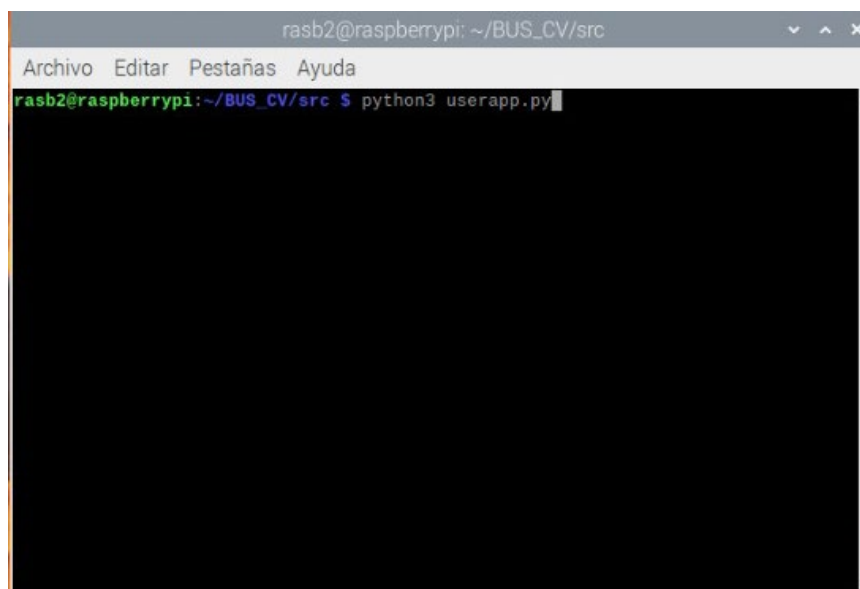


Nota. Elaboración Propia

Donde se debe poner el comando “python3 jasonapp.py” como se puede apreciar en la Figura 13, una vez puesto el comando se debe aplastar la tecla ENTER.

Figura 13

Pantalla negra con el comando

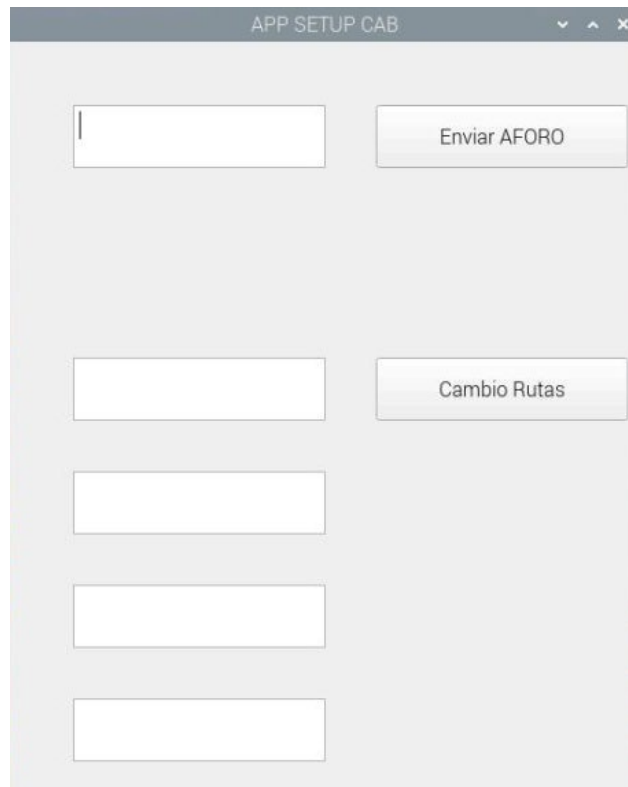


Nota. Elaboración Propia

Después de poner el comando se visualiza la interfaz que se puede apreciar en la Figura 14, donde se procede a cambiar el aforo y rutas del autobús.

Figura 14

Interfaz para cambiar el aforo y rutas



Nota. Elaboración Propia

Para cambiar el aforo se debe tomar en cuenta que solo se puede colocar números y si se coloca letras no se hará el cambio del aforo, además, en las rutas solo se puede escribir 11 letras máximo, debido a que si se escribe más no aparecerá de una forma visible en la matriz. Una vez hecho los cambios respectivos y siguiendo las indicaciones, se da un clic en el botón Aceptar y Añadir para guardar los cambios.

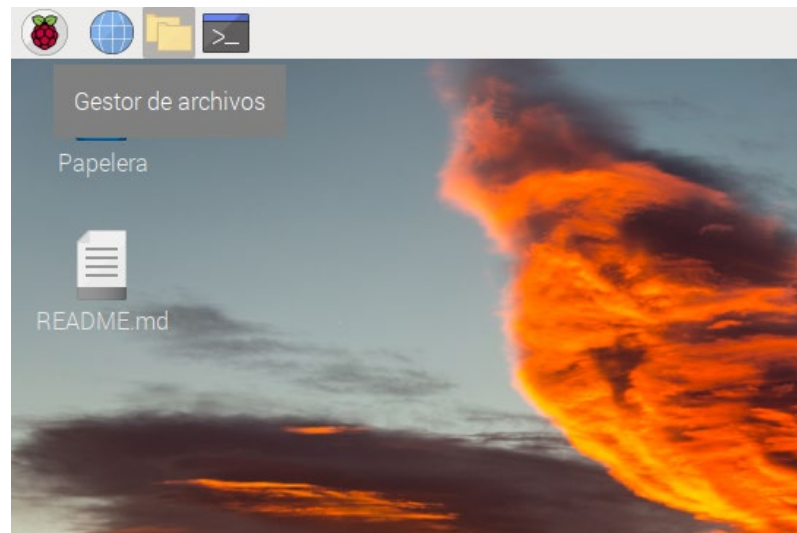
Después se da un clic en la X para cerrar la interfaz y se procede a colocar el comando “sudo reboot now” para reiniciar la raspberry pi 4 y así empezar el monitoreo del aforo del autobús con los cambios realizados, este cambio solo durara un día y lo podemos cambiar las veces que queramos, pero si queremos que el cambio sea permanente para todos los días, se debe realizar el cambio en el código, para realizar este cambio revisar el manual técnico.

Visualización del Registro

Para visualizar el archivo donde está el registro de las personas que ingresaron al autobús durante todo el día, debemos ingresar al gestor de archivos, en la Figura 15 se observa cual es el gestor de archivos, debemos dar un clic en el icono de las carpetas y entramos a la interfaz del gestor de archivos.

Figura 15

Gestor de archivos

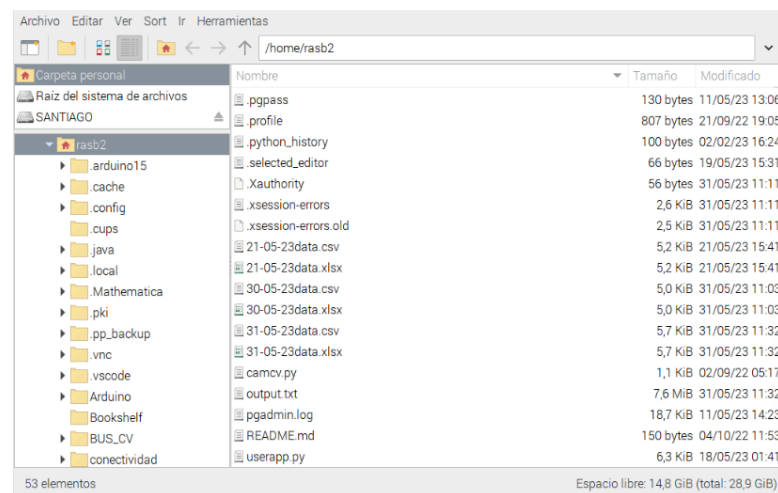


Nota. Elaboración Propia

Una vez adentro debemos buscar el archivo Excel con la fecha del registro que queramos visualizar, el nombre del archivo se guarda con el siguiente formato (dia-mes-añodada.xlsx) como se puede observar en la Figura 16. Una vez encontrada la fecha damos clic.

Figura 16

Interfaz del gestor de archivos



Nota. Elaboración Propia

Después se abrirá el archivo Excel como se puede observar en la Figura 17.

Figura 17

Archivo Excel con los registros

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	18/05/23	15:43:21	6	6	50															
2	18/05/23	15:43:22	6	6	50															
3	18/05/23	15:43:23	6	6	50															
4	18/05/23	15:43:24	6	6	50															
5	18/05/23	15:43:25	6	6	50															
6	18/05/23	15:43:26	6	6	50															
7	18/05/23	15:43:27	6	6	50															
8	18/05/23	15:43:28	6	6	50															
9	18/05/23	15:43:29	6	6	50															
10	18/05/23	15:43:30	6	6	50															
11	18/05/23	15:43:31	6	6	50															
12	18/05/23	15:43:32	6	6	50															
13	18/05/23	15:43:33	6	6	50															
14	18/05/23	15:43:34	6	6	50															
15	18/05/23	15:43:35	8	8	50															
16	18/05/23	15:43:36	8	8	50															
17	18/05/23	15:43:37	8	8	50															
18	18/05/23	15:43:38	8	8	50															
19	18/05/23	15:43:39	11	11	50															
20	18/05/23	15:43:40	11	11	50															
21	18/05/23	15:43:41	11	11	50															
22	18/05/23	15:43:42	11	11	50															
23	18/05/23	15:43:43	15	15	50															
24	18/05/23	15:43:44	19	19	50															
25	18/05/23	15:43:45	20	20	50															
26	18/05/23	15:43:46	20	20	50															
27	18/05/23	15:43:47	20	20	50															

Nota. Elaboración Propia

Existen 4 columnas en la primera columna se encuentra la fecha y la hora del registro, en la tercera columna esta los pasajeros actuales, en la cuarta columna está el número de pasajeros durante todo el día y en la última columna está el aforo, aquí nos podemos dar cuenta si el aforo fue modificado durante el recorrido del autobús.

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERIA EN CIENCIAS APLICAS
CARRERA DE ELECTRICIDAD



MANUAL TÉCNICO LA IMPLEMENTACIÓN DE UN PROTOTIPO DE
SISTEMA DE CONTEO DE PASAJERO EN BUSES URBANOS PARA CONTROL
DE AFORO



Objetivo

El presente manual técnico tiene la finalidad de ser una guía de las posibles fallas que pueda sufrir el prototipo, permitiendo al lector de este obtener las soluciones necesarias para su funcionamiento y ser una herramienta de consulta a la cual el docente pueda acudir en cualquier momento.

Instrucciones de seguridad



ATENCIÓN: Advertencia y Seguridad.

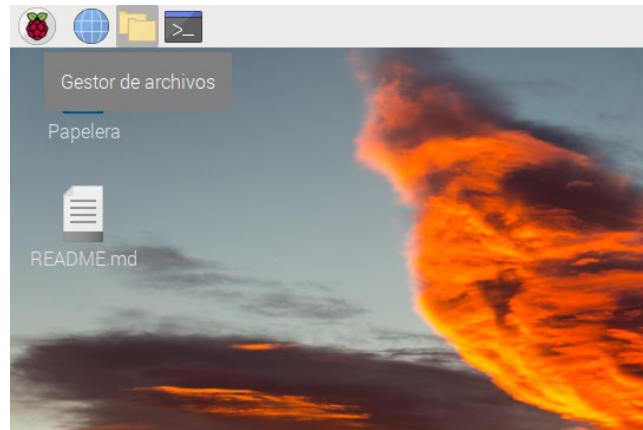
1. Antes de manipular el dispositivo lea el manual y consérvelo para futuras consultas.
2. Solicitar la autorización de uso respectiva al docente.
3. Asegúrese que el prototipo se encuentre en un lugar fijo.
4. Limpiar cualquier suciedad que se pueda encontrar sobre o alrededor de los componentes electrónicos.
5. Verificar el estado de conductores y conexiones del prototipo.
6. Ser cuidadoso al momento de manipular el prototipo ya que puede provocar un cortocircuito y dañar ciertos componentes.
7. Verificar el correcto funcionamiento de los equipos de alimentación DC.

Cambio de rutas predeterminadas

De igual forma si queremos dejar las rutas predeterminadas debemos ingresar al gestor de archivos como se muestra en la Figura 1 a continuación.

Figura 1

Gestor de archivos

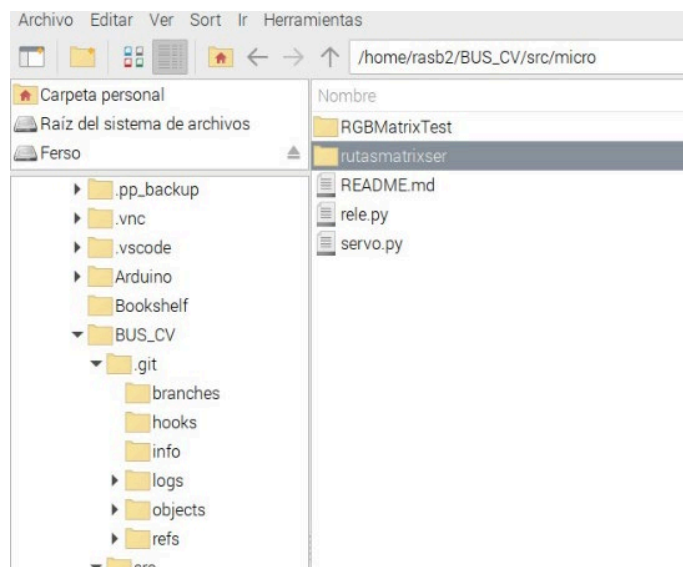


Nota. Elaboración Propia

Buscamos la Carpeta “Documentos”, después encontramos la carpeta “rutasmatrixser” damos clic en esa carpeta como se observa en la Figura 2.

Figura 2

Gestor de archivos

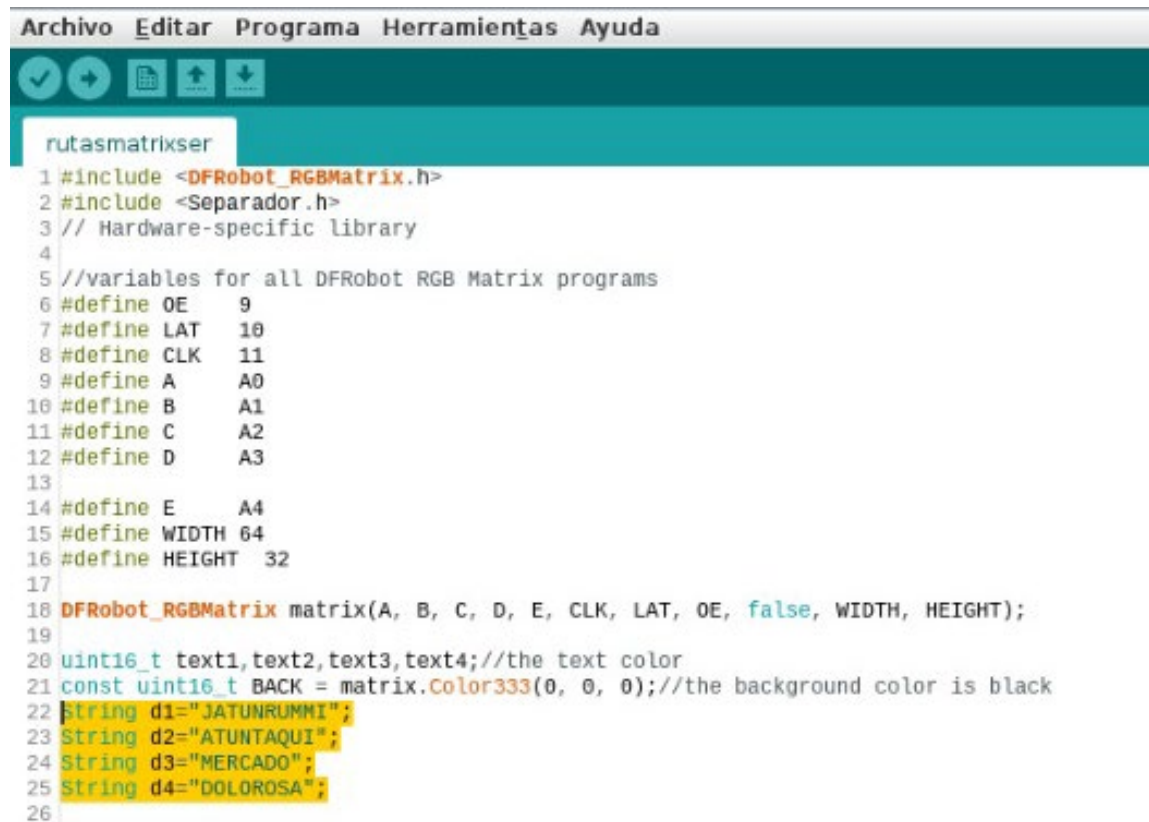


Nota. Elaboración Propia

Dentro de esa carpeta encontramos el código “rutasmatrixser.ino” aplastamos ahí y se abre el programa IDE Arduino como se observa en la Figura 3.

Figura 3

Interfaz del programa Arduino



```
Archivo  Editar  Programa  Herramientas  Ayuda
rutasmatrixser
1 #include <DFRobot_RGBMatrix.h>
2 #include <Separador.h>
3 // Hardware-specific library
4
5 //variables for all DFRobot RGB Matrix programs
6 #define OE      9
7 #define LAT     10
8 #define CLK     11
9 #define A       A0
10 #define B       A1
11 #define C       A2
12 #define D       A3
13
14 #define E       A4
15 #define WIDTH   64
16 #define HEIGHT  32
17
18 DFRobot_RGBMatrix matrix(A, B, C, D, E, CLK, LAT, OE, false, WIDTH, HEIGHT);
19
20 uint16_t text1, text2, text3, text4; //the text color
21 const uint16_t BACK = matrix.Color333(0, 0, 0); //the background color is black
22 String d1="JATUNRUNMI";
23 String d2="ATUNTAQUI";
24 String d3="MERCADO";
25 String d4="DOLOROSA";
26
```

Nota. Elaboración Propia

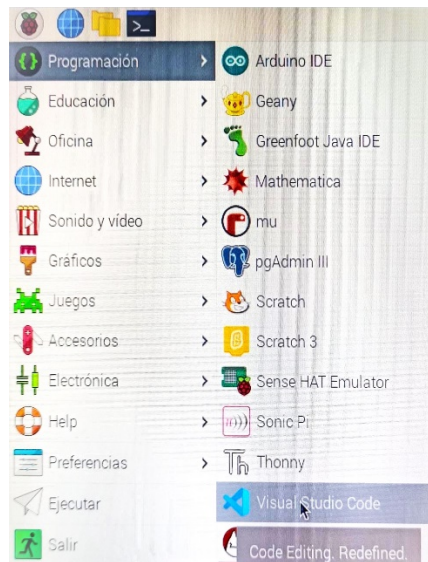
Para cambiar y dejar las rutas predeterminadas en las líneas de la 22 hasta 25 después de las comillas colocamos las rutas que van a quedar predeterminadas y subimos el código al Arduino mega. Debemos recordar que las rutas no deben superen las 11 letras para que se pueda visualizar de mejor manera en las matrices de leds. Después se procede a colocar el comando “sudo reboot now” para reiniciar la raspberry pi 4 y así empezar mostrando las nuevas rutas en la matriz de leds.

Cambio de distancia del sensor ultrasonido

Para el cambio de la distancia del sensor debemos entrar al programa Visual Studio Code. Para ingresar al programa primero se debe dar un clic en la esquina superior izquierda en el símbolo de la empresa raspberry como se puede observar en la Figura 4, después nos dijimos a programación y damos clic en el programa Visual Studio Code.

Figura 4

Interfaz de la raspberry pi 4



Nota. Elaboración Propia

Una vez dentro del programa nos dirigimos a la carpeta “raspberry” y buscamos el código con el nombre “controlador.py” como se muestra en la Figura 5 a continuación.

Figura 5

Interfaz del programa Visual Studio Code

```
253     if distance <= 100: # Si la distancia es menor a un metro
254         # Intentar detectar un rostro
255         ret, frame = video_capture.read()
256         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
257         faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
258
259         # Si se ha detectado un rostro y no estamos en estado de bloqueo, incrementar el contador y agregarlo a la base de datos
260         if len(faces) > 0 and not lock:
261             count += 1
262
263             #print('llego')
264
265             # Conectar a la base de datos
266             conn = sqlite3.connect('autobus.db')
267             c = conn.cursor()
268
269             # Insertar un nuevo registro
270             c.execute("INSERT INTO pasajeros (personas_en_el_bus, personas_al_dia, aforo) VALUES (?, ?, ?)", (countG, count, aforo1))
271             # Guardar (commit) los cambios
272             conn.commit()
273             # Cerrar la conexión a la base de datos
274             conn.close()
275             # Activar el bloqueo
276             lock = True
277             GPIO.output(CAM1, False)
278
279         elif len(faces) == 0:
280             # Si no se detecta ningún rostro, desactivar el bloqueo
281             lock = False
282             GPIO.output(CAM1, True)
283
284         countG = count - count2
285         if countG < 0:
286             countG = 0
287         gent = aforo1 - countG
288         if gent < 0:
289             gent = 0
290
```

Nota. Elaboración Propia

En la línea 253 se procede a cambiar la distancia del sensor ultrasonido, pero para que la raspberry funcione con el nuevo dato debemos reiniciarla comando “sudo reboot now”, una vez hecho esto funcionara con la nueva distancia.

Posibles fallas del Prototipo

El prototipo puede presentar algunas fallas, debido a esto se realizó el manual técnico para solucionar las fallas que pueda presentar el prototipo.

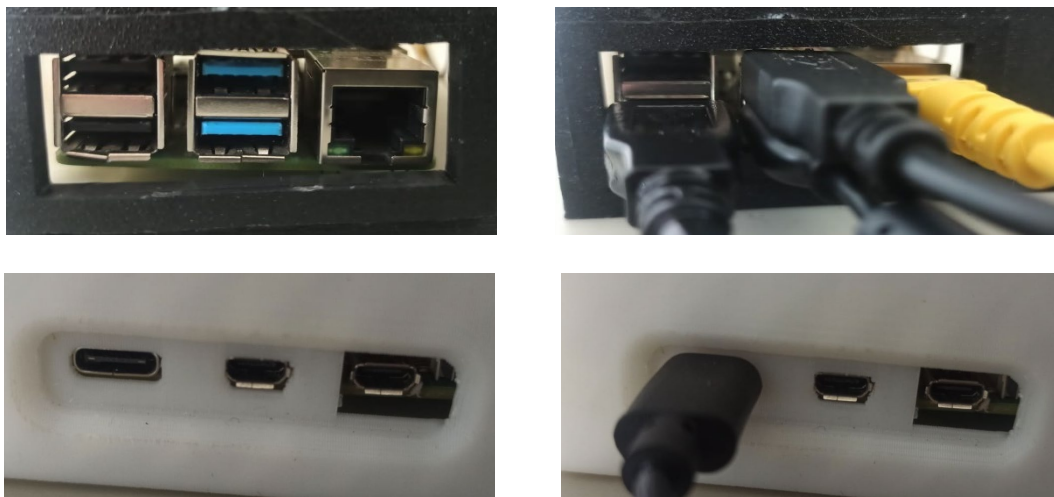
Desconexiones externas

En ocasiones se pueden desconectar los puertos que están conectados entre las raspberry pi 4, los Arduino y las cámaras debido a esto, a continuación, se indica como se debe conectar todos los equipos.

La caja principal de la puerta de entrada debe estar conectadas las 2 matrices de leds, mediante cables USB tipo A/B a la raspberry pi 4, además, debe estar conectada la cámara a un puerto USB de la raspberry pi 4. También debe estar conectado el cable de red entre las 2 raspberrys y por último debe estar conectado la fuente de alimentación al puerto tipo C de la raspberry pi 4 como muestra la Figura 6 a continuación.

Figura 6

Conexión raspberry pi 4 principal



Nota. Elaboración Propia

En la Tabla 1 a continuación se puede observar como es la conexión entre los elementos.

Tabla 1

Conexión de puertos caja principal

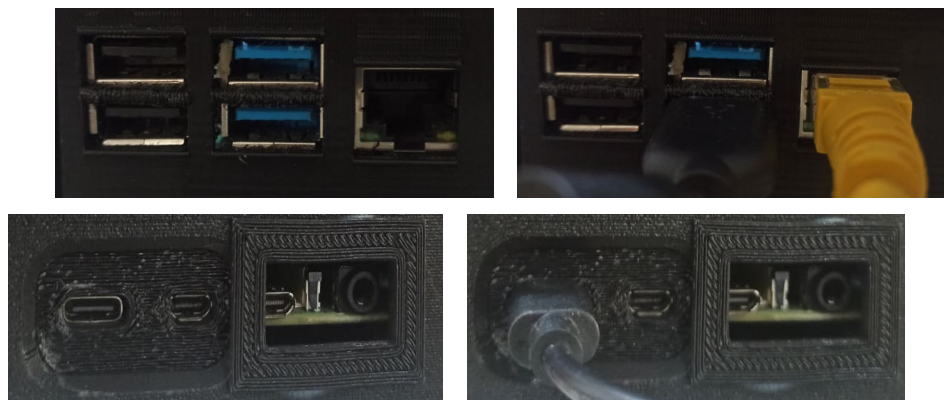
Com ponente	Pue rtos	Ras pberry PI 4 S	Ar duino Mega	Ar duino Mega	C ámara	F uente 5V
Rasp berry PI 4 P	Tipo C					X
	USB tipo A				X	
	USB tipo A		X			
	USB tipo A				X	
	ETH ERNET	X				

Nota. Elaboración Propia

La caja secundaria debe estar conectada el cable de red y la cámara debe estar conectada a un puerto USB de la raspberry pi 4 y por último debe estar conectado la fuente de alimentación al puerto tipo C de la raspberry pi 4 como muestra la Figura 7.

Figura 7

Conexión raspberry pi 4 secundaria



Nota. Elaboración Propia

En la Tabla 2 se puede observar cómo es la conexión entre los elementos.

Tabla 2

Conexión de puertos caja secundaria

Componen te	Puertos	Ras pberry PI 4 S	Cáma ra	Fuen te 5V

	Tipo C		X
Raspberry PI 4 secundaria	USB tipo A ETHERN ET	X	

Nota. Elaboración Propia

Las matrices de leds deben estar conectadas a una fuente de alimentación y a la raspberry principal como muestra la Figura 8.

Figura 8

Conexión Matrices de leds



Nota. Elaboración Propia

En la Tabla 3, se puede observar cómo es la conexión entre los elementos.

Tabla 3

Conexión de puertos Matrices de Leds

Componente	Puertos	Raspberry PI 4 Principal	Fuente 5V
Arduino mega	USB tipo A	X	
Matriz de leds	Directo		X

Nota. Elaboración Propia

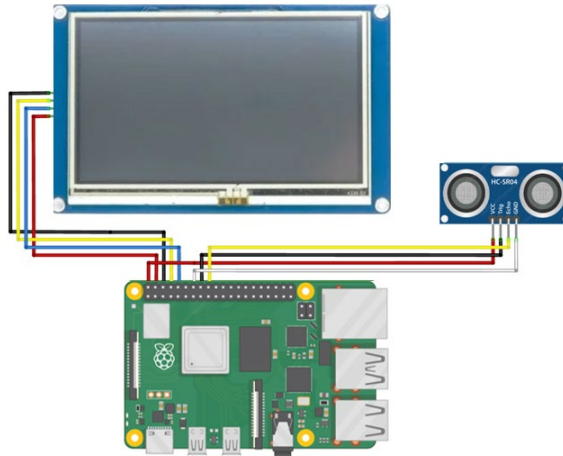
Desconexiones internas

En ocasiones se pueden desconectar los pines que están conectados dentro de las raspberry pi 4, los Arduino y las matrices de leds debido a esto a continuación se indica como se debe conectar todos los pines en caso de una desconexión.

La caja principal la raspberry pi 4 esta conectada con un sensor ultrasonido y una apantalla HMI Nextion de 5 pulgadas en la Figura 9 se puede observar de forma grafica como es la conexión entre estos elementos de la caja principal.

Figura 9

Conexión interna de la caja principal



Nota. Elaboración Propia

A continuación, en la Tabla 4 se observa cómo es la conexión de cada pin en caso de alguna desconexión interna de las cajas. Además, cada cable esta con su respectivo nombre en caso de existir alguna desconexión.

Tabla

12

Conexión de pines caja principal

Componentes	Pines	Color del cable	Pines Raspberry PI 4
Pantalla Nextion NX8048T050	+ 5V	rojo	4 (5V power)
	RX	amarillo	8 (GPIO 14 TXD)
	TX	azul	10 (GPIO 15 RTX)
	GND	negro	6 (Ground)

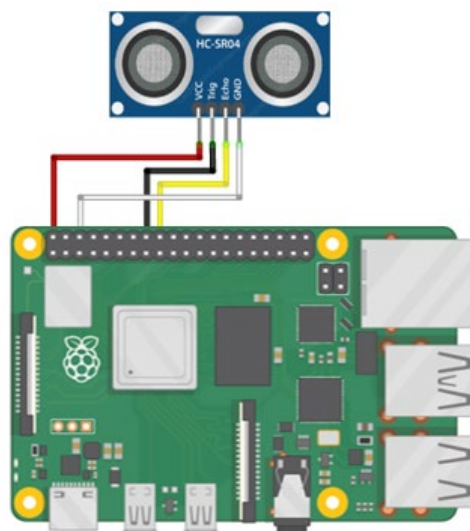
Sensor Ultrasonido HC-SR04	VCC	rojo	2 (5V power)
	Tring	negro	16 (GPIO 23)
	Echo	amarillo	18 (GPIO 24)
	GND	blanco	14 (Ground)

Nota. Elaboración Propia

La caja secundaria la raspberry pi 4 está conectada con un sensor ultrasonido en la cual se observa en la Figura 10 forma gráfica como es la conexión entre estos elementos de la caja principal.

Figura 10

Conexión interna de la caja secundaria



Nota. Elaboración Propia

En la Tabla 5 a continuación se observa cómo es la conexión de cada pin en caso de alguna desconexión interna de las cajas. Además, cada cable esta con su respectivo nombre en caso de existir alguna desconexión.

Tabla 5

Conexión de pines caja secundaria

Componentes	Pines	Color del cable	Pines Raspberry PI 4
--------------------	--------------	------------------------	-----------------------------

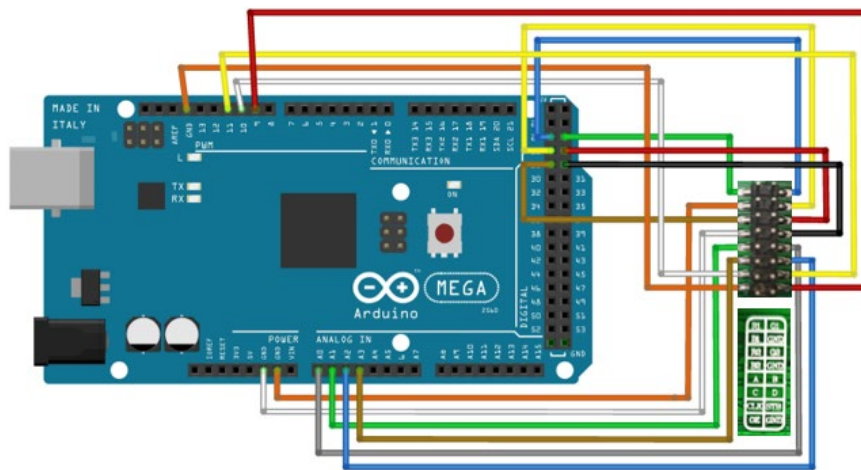
Sensor Ultrasonido HC-SR04	VCC	rojo	2 (5V power)
	Tring	negro	16 (GPIO 23)
	Echo	amarillo	18 (GPIO 24)
	GND	blanco	14 (Ground)

Nota. Elaboración Propia

Las cajas de las rutas el Arduino mega está conectada a una matriz de led de 64 x 32. En la Figura 11 se puede observar de forma gráfica como es la conexión entre estos elementos de la caja principal.

Figura 11

Conexión interna de la caja de las rutas



Nota. Elaboración Propia

A continuación, en la Tabla 6 se observa cómo es la conexión de cada pin en caso de alguna desconexión interna de las cajas. Además, cada cable está con su respectivo nombre en caso de existir alguna desconexión.

Tabla 6

Conexión de pines caja de las rutas

Componentes	Pines	Color del Cable	Pines Arduino Mega
	R1	verde	25

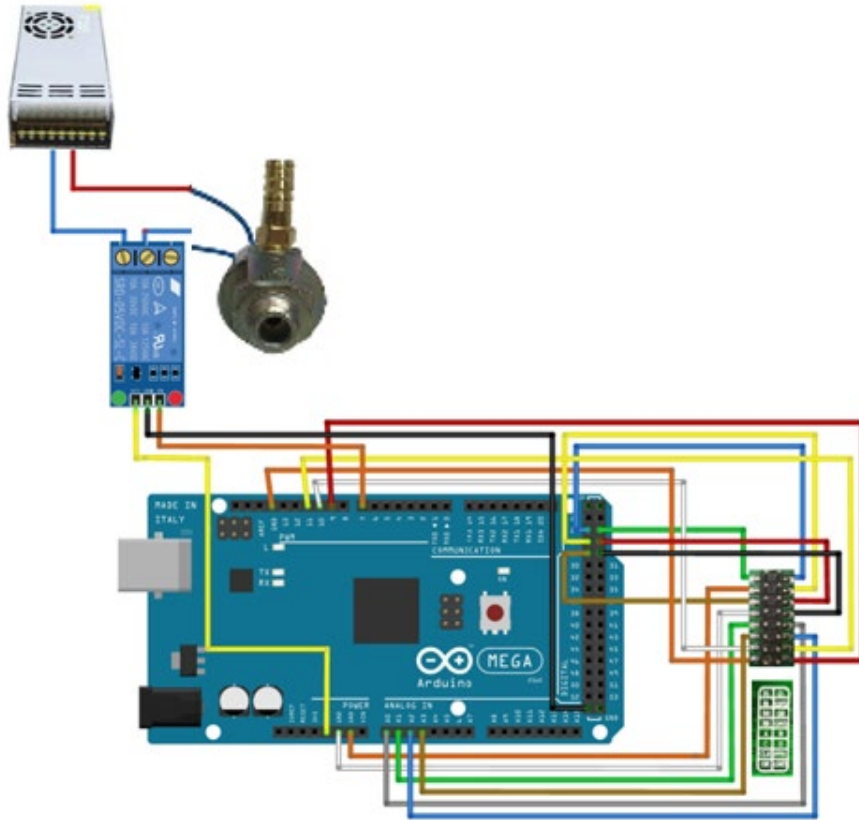
		B1	naranja	GND
		R2	café	28
		B2	blanco	GND
Matriz	Led	A	lila	A1
64x32		C	verde	A3
		CLK	naranja	10
		OE	café	GND
		G1	azul	24
		GND	amarillo	26
		G2	rojo	27
		GND	negro	29
		B	gris	A0
		D	azul	A1
		STB –		
		LAT	amarillo	11
		GND	rojo	9

Nota. Elaboración Propia

En la caja del aforo, el Arduino mega está conectada a una matriz de led de 64 x 32 en la Figura 12 se puede observar de forma gráfica como es la conexión entre estos elementos de la caja principal.

Figura 12

Conexión interna de la caja del aforo



Nota. Elaboración Propia

A continuación, en la Tabla 7 se observa cómo es la conexión de cada pin en caso de alguna desconexión interna de las cajas. Además, cada cable está con su respectivo nombre en caso de existir alguna desconexión.

Tabla 7

Conexión de pines caja de caja del aforo

Componentes		Pines	Color del Cable	Pines Arduino Mega
Matriz 64x32	Led	R1	verde	24
		B1	naranja	26
		R2	café	27
		B2	blanco	29
		A	verde	A0
		C	café	A2
		CLK	blanco	11
		OE	naranja	9
	G1	azul	25	

	GND	amarillo	GND
	G2	rojo	28
	GND	negro	GND
	B	gris	A1
	D	azul	A3
	STB –		
	LAT	amarillo	10
	GND	rojo	GND
	VCC	amarillo	5V
Relé	GND	negro	GND
	IN	naranja	7

Nota. Elaboración Propia

Configuración IP

La dirección IP viene configurada para funcionar con el cable de red. En caso de que se desconfigure se debe entrar mediante el comando “ifconfig” como se muestra en la Figura 13 y revisar el etho.

Figura 13

Comando para revisar las direcciones IP

```

rasb2@raspberrypi: ~/BUS_CV
Archivo  Editar  Pestañas  Ayuda
rasb2@raspberrypi:~/BUS_CV/src $ ifconfig

```

Nota. Elaboración Propia

En la raspberry principal debe estar la dirección IP 169.254.239.172 como se muestra en la Figura 14.

Figura 14

Dirección IP raspberry principal


```
rasb2@raspberrypi: ~/BUS_CV/src
Archivo Editar Pestañas Ayuda
rasb2@raspberrypi:~/BUS_CV/src $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 169.254.238.172 netmask 255.255.0.0 broadcast 169.254.255.255
    inet6 fe80::dea6:32ff:fe4e:d503 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:4e:d5:03 txqueuelen 1000 (Ethernet)
    RX packets 22 bytes 3493 (3.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 59 bytes 7252 (7.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2640 bytes 821296 (802.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2640 bytes 821296 (802.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Nota. Elaboración Propia

En la raspberry secundaria debe estar la dirección IP 169.254.212.26 como se muestra en la Figura 1.5

Figura 15

Dirección IP raspberry secundaria

```
rasb2@raspberrypi: ~/BUS_CV/src
Archivo Editar Pestañas Ayuda
rasb2@raspberrypi:~/BUS_CV/src $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 169.254.212.26 netmask 255.255.0.0 broadcast 169.254.255.255
    inet6 fe80::e65f:1ff:fe2f:31b6 prefixlen 64 scopeid 0x20<link>
    ether e4:5f:01:2f:31:b6 txqueuelen 1000 (Ethernet)
    RX packets 76 bytes 7529 (7.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 48 bytes 7180 (7.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 320 bytes 35420 (34.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 320 bytes 35420 (34.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Nota. Elaboración Propia