



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA  
COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**TEMA: SISTEMA DE DETECCIÓN DE PLAZAS DE APARCAMIENTO  
DISPONIBLE, MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES  
Y TECNOLOGÍA BOT**

**AUTOR: FONSECA PAZMIÑO, JEFFERSON VLADIMIR**

**DIRECTORA: ING. GUERRERO IDROVO, ROSA GRACIELA**

**SANGOLQUÍ**

**2020**

## ÍNDICE DE CONTENIDOS

<b>ÍNDICE DE CONTENIDOS .....</b>	<b>ii</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>v</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>vi</b>
<b>CERTIFICADO DEL DIRECTOR.....</b>	<b>1</b>
<b>AUTORÍA DE RESPONSABILIDAD .....</b>	<b>2</b>
<b>AUTORIZACIÓN .....</b>	<b>3</b>
<b>DEDICATORIA .....</b>	<b>4</b>
<b>AGRADECIMIENTO .....</b>	<b>5</b>
<b>CAPÍTULO I .....</b>	<b>6</b>
<b>PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>6</b>
<b>1.1 Antecedentes .....</b>	<b>6</b>
<b>1.2 Contexto del problema .....</b>	<b>7</b>
<b>1.3 Pregunta de Investigación .....</b>	<b>10</b>
<b>1.4 Objetivos .....</b>	<b>10</b>
1.4.1 Objetivo General .....	10
1.4.2 Objetivos Específicos.....	10
<b>1.5 Justificación .....</b>	<b>11</b>
<b>1.6 Alcance .....</b>	<b>11</b>
<b>1.7 Hipótesis .....</b>	<b>12</b>
1.7.1 Estadística - Cuantitativa.....	12
<b>1.8 Metodología de la Investigación.....</b>	<b>12</b>
<b>CAPÍTULO II.....</b>	<b>14</b>
<b>TRABAJOS RELACIONADOS Y MARCO TEÓRICO.....</b>	<b>14</b>
<b>2.1 Estado del Arte .....</b>	<b>14</b>
2.1.1 Definición para la búsqueda.....	15
2.1.1.1 Definición de las preguntas de investigación .....	16
2.1.1.2 Alcance de la revisión .....	16
2.1.1.3 Criterios de inclusión y exclusión .....	18
2.1.1.4 Cadena de búsqueda .....	18
2.1.2. Ejecución de la búsqueda .....	19
2.1.2.1 Selección estudios primarios.....	19
2.1.2.2 Definición criterios de análisis .....	20
2.1.3 Trabajos Relacionados .....	21
2.1.3.1 Análisis de los trabajos.....	21

<b>2.2 Marco Teórico .....</b>	<b>42</b>
2.2.1 Procesamiento digital de imágenes .....	42
2.2.2 Imagen digital.....	43
2.2.2.1 Tipo básico de imágenes digitales.....	43
2.2.2.1.1 Imágenes de mapa de bits o rasters .....	44
2.2.2.1.2 Imágenes vectoriales .....	44
2.2.3 Técnicas de Procesamiento Digital de Imágenes .....	45
2.2.3.1 Adquisición .....	45
2.2.3.2 Realce y mejora.....	47
2.2.4 Segmentación .....	50
2.2.5 Extracción de características .....	51
2.2.6 Tecnología Bot .....	52
2.2.7 Sistemas empotrados con Raspberry.....	53
2.2.8 Aplicativos en Telegram con tecnología Bot .....	53
2.2.9 Métodos para el Procesamiento digital de Imágenes .....	54
2.2.9.1 Error cuadrático medio.....	54
2.2.9.2 Medida de similitud estructural.....	55
2.2.10 OpenCV.....	56
<b>CAPÍTULO III .....</b>	<b>57</b>
<b>DESARROLLO E IMPLEMETACIÓN.....</b>	<b>57</b>
<b>3.1 Diseño .....</b>	<b>57</b>
3.1.1 Arquitectura.....	57
3.1.2 Diseño del Bot en Telegram.....	58
3.1.2.1 Creación del Bot.....	58
3.1.2.2 Maquetación del Bot .....	59
3.1.3 Interacción.....	62
<b>3.2 Implementación .....</b>	<b>64</b>
3.2.1 Instalación OpenCV .....	64
3.2.2 Captura de imagen.....	65
3.2.3 Reconocimiento imagen.....	66
3.2.4 API .....	68
3.2.5 Peticiones y respuestas .....	68
3.2.6 Base de Datos e Imágenes.....	69
3.2.6.1 Modelo Base de Datos.....	69
3.2.6.2 Adquisición de Imágenes .....	70
3.2.7 Pruebas Individuales .....	72
3.2.7.1 Definición experimento.....	72
3.2.7.2 Prueba desde las 7 a.m. ....	73
3.2.7.3 Prueba desde las 2 p.m. ....	77
<b>CAPÍTULO IV .....</b>	<b>82</b>
<b>EVALUACIÓN.....</b>	<b>82</b>
<b>4.1 Instrumentos .....</b>	<b>82</b>

<b>4.2 Pruebas del Modelo</b> .....	<b>86</b>
4.2.1 Escenario I.....	86
<b>Objetivo</b> .....	86
<b>Tareas</b> .....	86
4.2.2 Escenario II .....	87
<b>Objetivo</b> .....	87
<b>Tareas</b> .....	87
<b>4.3 RESULTADOS</b> .....	<b>88</b>
<b>Escenario I</b> .....	88
<b>Tiempos</b> .....	88
<b>Escenario II</b> .....	89
<b>Tiempos</b> .....	89
<b>Encuesta</b> .....	91
<b>4.4 Análisis Resultados</b> .....	<b>93</b>
<b>CAPÍTULO V</b> .....	<b>95</b>
<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>95</b>
<b>5.1 Conclusiones</b> .....	<b>95</b>
<b>5.2 Recomendaciones</b> .....	<b>96</b>
<b>5.3 Trabajos Futuros</b> .....	<b>96</b>
<b>ANEXOS</b> .....	<b>97</b>
<b>REFERENCIAS</b> .....	<b>101</b>

**ÍNDICE DE TABLAS**

<b>Tabla 1</b>	Cadenas de búsqueda en base de datos .....	17
<b>Tabla 2</b>	Resultados numéricos por base de datos .....	17
<b>Tabla 3</b>	Esquema de caracterización de estudios primarios .....	36
<b>Tabla 4</b>	Descripción del parqueadero de la Figura 30.....	74
<b>Tabla 5</b>	Descripción del parqueadero de la Figura 30(b) .....	74
<b>Tabla 6</b>	Descripción del parqueadero de la Figura 30(c) .....	75
<b>Tabla 7</b>	Descripción del parqueadero de la Figura 30(d) .....	75
<b>Tabla 8</b>	Descripción del parqueadero de la Figura 30(e) .....	77
<b>Tabla 9</b>	Descripción del parqueadero de la Figura 30(f).....	77
<b>Tabla 10</b>	Descripción del parqueadero de la Figura 31 .....	79
<b>Tabla 11</b>	Descripción del parqueadero de la Figura 31(b) .....	79
<b>Tabla 12</b>	Descripción del parqueadero de la Figura 31(c) .....	80
<b>Tabla 13</b>	Descripción del parqueadero de la Figura 31(d) .....	80
<b>Tabla 14</b>	Descripción del parqueadero de la Figura 31(e) .....	81
<b>Tabla 15</b>	Encuesta Percepción sobre el uso del Parqueadero.....	84
<b>Tabla 16</b>	Encuesta Usabilidad del Sistema CarParkEspe.....	91

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Proceso para la obtención del mapeo sistemático.....	15
<b>Figura 2.</b> Gráfico de búsqueda de estudios primarios .....	20
<b>Figura 3.</b> Gráfico de burbuja del mapeo sistemático.....	38
<b>Figura 4.</b> Gráfico de anillo de estudios primarios .....	38
<b>Figura 5.</b> Total artículos publicados por año y base de datos .....	39
<b>Figura 6.</b> Arquitectura del Sistema.....	58
<b>Figura 7.</b> BotFather .....	59
<b>Figura 8.</b> Inicio .....	60
<b>Figura 9.</b> Petición para la disponibilidad de plazas.....	61
<b>Figura 10.</b> Comando de ayuda .....	62
<b>Figura 11.</b> Comando de locación del parqueadero .....	62
<b>Figura 12.</b> Diagrama de flujo del sistema de parqueadero .....	64
<b>Figura 13.</b> Conexión de RPI3 y cámara NoIR .....	65
<b>Figura 14.</b> Configuración habilitar cámara .....	65
<b>Figura 15.</b> Código para captura de imagen .....	66
<b>Figura 16.</b> Código para carga imágenes .....	66
<b>Figura 17.</b> Código para segmentación y redimensión imágenes.....	67
<b>Figura 18.</b> Código para algoritmo SSIM y deserción de plazas.....	67
<b>Figura 19.</b> Comandos para Telepot .....	68
<b>Figura 20.</b> Código para petición “/plaza” .....	68
<b>Figura 21.</b> Respuesta “/plaza” .....	68
<b>Figura 22.</b> Código para petición “/ayuda” y respuesta.....	69
<b>Figura 23.</b> Código para petición “/descripcion” y respuesta.....	69
<b>Figura 24.</b> Código para petición desconocida y respuesta .....	69
<b>Figura 25.</b> Modelo Conceptual de la BD .....	70
<b>Figura 26.</b> Directorio de la Plantilla.....	70
<b>Figura 27.</b> Directorio de imágenes capturadas con la cámara.....	71
<b>Figura 28.</b> Directorio de imágenes procesadas .....	71
<b>Figura 29.</b> Plazas del Parqueadero .....	72
<b>Figura 30.</b> Respuesta de plazas .....	73
<b>Figura 31.</b> Respuesta de plazas .....	78
<b>Figura 32.</b> Escenario 1, resultado tiempos para encontrar plazas libres (min:s:ms) .....	89
<b>Figura 33.</b> Escenario 2, resultado tiempos para encontrar plazas libres (min:s:ms) .....	90
<b>Figura 34.</b> Resultado tiempos Petición-Respuesta CarParkEspe (s:ms) .....	90



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “**SISTEMA DE DETECCIÓN DE PLAZAS DE APARCAMIENTO DISPONIBLE, MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES Y TECNOLOGÍA BOT**”, fue realizado por el señor **Fonseca Pazmiño, Jefferson Vladimir** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

**Sangolquí, 16 de enero de 2020**



**Ing. Guerrero Idrovo, Rosa Graciela**

C.C.: 1720513322

**DIRECTORA**



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**AUTORÍA DE RESPONSABILIDAD**

Yo, **Fonseca Pazmiño Jefferson Vladimir**, declaro que el contenido, ideas y criterios del trabajo de titulación: **SISTEMA DE DETECCIÓN DE PLAZAS DE APARCAMIENTO DISPONIBLE, MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES Y TECNOLOGÍA BOT** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

**Sangolquí, 16 de enero de 2020**

---

**Fonseca Pazmiño Jefferson Vladimir**

C.C.: 1718258393



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**AUTORIZACIÓN**

*Yo, **Fonseca Pazmiño Jefferson Vladimir** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **SISTEMA DE DETECCIÓN DE PLAZAS DE APARCAMIENTO DISPONIBLE, MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES Y TECNOLOGÍA BOT** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.*

**Sangolquí, 16 de enero de 2020**

A handwritten signature in blue ink, which appears to read 'Fonseca Pazmiño Jefferson Vladimir', is written over a horizontal line.

**Fonseca Pazmiño Jefferson Vladimir**

C.C.: 1718258393

## **DEDICATORIA**

A mi madre Maribel Pazmiño y mi padre Walberto Fonseca, por brindarme su amor incondicional, su preocupación por mi bienestar, su tiempo, sus consejos y reprimendas que han forjado una buena persona. Siempre serán fuente de inspiración y mi camino a seguir.

A mi hijo Sebastian por el amor que le tengo, por ser el pilar fundamental de mi vida, mi mayor motivo de superación y por el que lucho día tras día para ofrecerle un futuro mejor.

A mi abuelito Moisés que desde el cielo me cuida con su bendición, recuerdo su bondad y cariño en mi hermana y en mí.

A mi abuelita Digna que con su amor incondicional siempre permanece pendiente de nosotros. Nos alegra con su personalidad y carisma, y que a pesar de los problemas siempre nos cuida y nos aconseja.

A mi abuelita Sara y abuelito Rafael que desde siempre nos han brindado su amor, apoyo, consejos, siempre nos reciben con las puertas abiertas y permanecen pendientes de nosotros.

## **AGRADECIMIENTO**

En primer lugar, agradezco a Dios por cubrirme con su manto de amor, cuidarme y darme la sabiduría necesaria en mi trayecto por cumplir mi tan anhelado objetivo dentro de la Universidad.

Agradezco de todo corazón a mis padres por su amor incondicional, por su protección, ayuda y respaldo para comenzar, continuar y finalizar esta etapa de mi vida profesional.

A mi hermana Tania por los consejos brindados, por el apoyo prestado en momentos necesarios, y porque a pesar de las riñas siempre ha velado por el bienestar de mi hijo.

A mis abuelitos, tíos, primas, que me han apoyado desde el inicio y han visto en mi un ejemplo a seguir, por su preocupación y apoyo moral.

A la Ingeniera Graciela por su dedicación, tiempo y por orientarme con sus conocimientos durante el desarrollo de este trabajo.

Y en general a todas las personas que de una u otra manera contribuyeron para el desarrollo de mi vida profesional.

¡Gracias!

# CAPÍTULO I

## PLANTEAMIENTO DEL PROBLEMA

### 1.1 Antecedentes

La evolución y crecimiento tecnológico de las últimas dos décadas ha experimentado las posibilidades de la digitalización a gran velocidad, distribución de la información y el conocimiento a través de Internet y aún más cercanos a través de la telefonía móvil (Cabrera, 2010). La utilización de las tecnologías de la información y la comunicación (TIC) se ha expandido por todo el mundo. “El rápido progreso de estas tecnologías brinda oportunidades sin precedentes para alcanzar niveles más elevados de desarrollo. La capacidad de las TICs para reducir muchos obstáculos tradicionales, especialmente el tiempo y la distancia, posibilitan, por primera vez en la historia, el uso potencial de estas tecnologías en beneficio de millones de personas en todo el mundo” (Cumbre Mundial sobre la Sociedad de la Información, 2003).

El surgimiento de las tecnologías de la información y la comunicación se puede presentar a partir de dos hechos significativos: las personas empezaron a tener disponibilidad a internet en sus hogares a finales del siglo XX y el surgimiento de los móviles inteligentes o Smartphone a mediados de la primera década del siglo XXI (Rojo, 2018). La integración de estas tecnologías y el desarrollo constante de aplicaciones móviles ha permitido llevar en nuestro dispositivo decenas de aplicaciones como enciclopedias digitales, cientos de archivos multimedia como música, imágenes, videos, documentos; comunicación con muchos lugares del mundo, decenas de juegos y aplicaciones móviles como el GPS, que se usan diariamente.

Por otro lado, los conductores suelen enfrentarse diariamente con la ardua tarea de encontrar una plaza de aparcamiento. En el estudio realizado por IBM denominado “IBM Global Parking

Survey: Drivers Share Worldwide Parking Woes” (Armonk, 2011), fueron encuestados 8.042 viajeros en 20 ciudades en los seis continentes, donde la primera encuesta de estacionamiento que realizaron se encontró que los conductores en las 20 ciudades enfrentan diariamente una lucha por encontrar un espacio de aparcamiento. En el 2011 seis de cada 10 conductores abandonaron su búsqueda de una plaza al menos una vez, y que más de una cuarta parte tuvieron alguna discusión con otro automovilista por espacio de aparcamiento. Además de la congestión generada por el desplazamiento automovilístico y la búsqueda de parqueadero y los accidentes, los informes estimaron que más del 30% del tráfico en una ciudad es causada por conductores que buscan una plaza de estacionamiento. Por otro lado, los sistemas de estacionamiento ineficientes no solo provocan congestión y mayor emisión de carbono, sino que también desperdician el tiempo de los conductores, generan pérdida de productividad, oportunidades económicas y pueden llevar a servicios urbanos ineficientes (Armonk, 2011). Otro estudio, llevado a cabo por BOSCH (Robert, 2016), mostró que buscar una plaza de aparcamiento en Alemania representa aproximadamente el 30% del tráfico urbano. Las frenadas repentinas, conducir muy despacio en un atasco, maniobras esquivas, tráfico muy denso y conducir por zonas desconocidas son algunas de las situaciones más estresantes durante la búsqueda de aparcamiento. Es por ello, que se han intentado diseñar aplicaciones que pretenden simplificar dicho proceso del diario vivir de los usuarios.

## **1.2 Contexto del problema**

Los vehículos matriculados en el Ecuador entre el año 2008 y 2016 tuvo un incremento anual del 10.6%, según el Instituto Nacional de Estadística y Censos (INEC, 2016). En el año 2014 se registraron 1.752.712 vehículos matriculados donde el 94.5 % se trataba de vehículos particulares, mientras que en el año 2015 fueron registrados 1.925.368 vehículos matriculados,

donde las provincias de Pichincha, Manabí, Guayas, Azuay, concentraban el 58.7% de vehículos. En el año 2016, el número de vehículos era de 2.056.213, donde Pichincha, Manabí, Guayas, Los Ríos concentraban el 71.7% de vehículos, mientras que el año 2017 se registró 2.237.264 vehículos matriculados.

Entre los múltiples problemas que conlleva el aumento vehicular como los accidentes, la congestión, los atascos, cabe destacar que existe una demanda mayor del parqueadero automotor sobre todo en las zonas céntricas, por lo que los conductores deben afrontar la falta de parqueo en la metrópoli, perder tiempo en la búsqueda de plazas y terminar por aceptar pagar las tarifas en las zonas comerciales.

El incremento de vehículos implica un aumento de tráfico en la búsqueda del parque automotor, la falta de gestión en los parqueaderos de la ciudad, que conlleva a la pérdida de tiempo del conductor. En el 2015, se matricularon 1.925.368 vehículos motorizados en Ecuador, 57% más que lo registrado en el 2010 cuando la cifra llegó a 1.226.349, según los últimos datos del Anuario de Transportes 2015, publicado por el Instituto Nacional de Estadística y Censos (INEC, 2016). Pichincha fue la provincia con más vehículos matriculados con 492.568, seguido de Guayas con 362.857 y Manabí con 152.231. La mayor concentración de vehículos motorizados matriculados en el 2015, por marca, la lideró Chevrolet con 554.042. A nivel nacional del total de vehículos matriculados el 28,1% tiene más de doce años (2004 y años anteriores); mientras que aquellos vehículos que tienen de uno a 5 años (modelos 2009 a 2016), representan el 55,1% del total; es decir que el parque automotor está compuesto, en mayor proporción por vehículos nuevos.

Por tanto, encontrar aparcamiento en las ciudades se está convirtiendo en un problema nacional. Uno de los múltiples problemas que confronta la ciudad de Quito es el de la falta de estacionamientos en comparación a la demanda de vehículos que circulan por el centro y por

distritos de la periferia. Además, la búsqueda de parqueo ha originado malestar a los propietarios, ya que estos tienen que aparcar sus unidades de transporte lejanos de sus lugares de trabajo, realizar trámites, compras, etc. Formando una mayor demanda de espacios para estacionarse, e implantando así la necesidad de reglamentar el estacionamiento en las calles.

En el caso particular de las Universidades, los docentes, alumnos y personal administrativo se encuentra en la necesidad de buscar parqueadero en la zona más cercana de su lugar de trabajo o estudio, de tal manera que es necesario tomar medidas pertinentes que ayuden a obtener un manejo adecuado del parqueadero. Para esto, es necesario evaluar la relación existente entre la infraestructura, manejo del parqueadero y las actividades de los usuarios, donde el primero debe proporcionar un ambiente con condiciones adecuadas que faciliten el acceso vehicular.

Es por estos motivos que se requiere de brindar soluciones para los problemas asociados a la movilidad. Una posible solución al problema de las plazas de parqueo en las calles podría ser por medio de estudios de tránsito que identifiquen el comportamiento de la demanda de estacionamientos a partir de trabajos de campo como el estudio de placas tratando siempre de cumplir con los requisitos mínimos de estacionamiento para la habilitación de más espacios. Otra posible solución sería el encontrar un sistema de búsqueda de parqueadero en la ciudad, evitando al usuario la pérdida de tiempo y energía.

Actualmente existen algunos centros comerciales como Plaza Imperial en España que implementan sistemas que mediante pantallas que indican el número de plazas libres en algunas zonas específicas, además cuentan con sensores y luces que cambian de color para indicar si la plaza está disponible, sin embargo, estos sistemas tienen algunas deficiencias ya que el usuario debe pasar por las pantallas para conocer la información y en ese transcurso pueden existir plazas de parqueo libres que el usuario no puede conocer con exactitud. Además, el sistema requiere de

sensores por módulos de plaza por lo que su costo puede ser elevado y su instalación compleja comparada con otras soluciones inalámbricas.

### **1.3 Pregunta de Investigación**

Q1: ¿Qué porcentaje de optimización se obtiene al utilizar la técnica de procesamiento digital de imagen seleccionada y la tecnología Bot en la detección de plazas de aparcamiento disponible?

### **1.4 Objetivos**

#### **1.4.1 Objetivo General**

Diseñar un sistema de visualización de zonas disponibles de aparcamiento en una zona determinada del estacionamiento de la Universidad de las Fuerzas Armadas ESPE, basado en procesamiento de imágenes y el uso de tecnología Bot.

#### **1.4.2 Objetivos Específicos**

- Analizar mediante trabajos relacionados las herramientas que se van a utilizar en el desarrollo del sistema mediante la revisión de trabajos realizados con técnicas empleadas para el procesamiento de imágenes y tecnología Bot para la recepción y envío de imágenes al usuario.
- Diseñar un modelo que permita la captura de imágenes y posteriormente el procesamiento de la misma, para que mediante algoritmos de procesamiento indique si la plaza de estacionamiento se encuentra disponible.

- Desarrollar un Bot y un sistema empotrado que permita la captura, procesamiento y envío de imágenes al usuario.
- Realizar la evaluación, validación y divulgación de los resultados obtenidos.

## **1.5 Justificación**

La Universidad de las Fuerzas Armadas ESPE, no cuenta con un sistema de alerta temprana que preste el servicio de visualización de zonas de aparcamiento, es por ello por lo que se ha tomado la decisión de desarrollar el sistema de visualización de plazas de aparcamiento, mismo que prestará dicho servicio de forma oportuna, clara y eficaz para la comunidad universitaria.

Con el desarrollo e implementación del Sistema de detección de plazas libres mediante procesamiento de imágenes y tecnología Bot, se optimizó el proceso en la búsqueda de lugares al minimizar el tiempo de búsqueda de plazas libres versus al método tradicional de ubicar de forma física los lugares de estacionamiento sobre todo en horarios en el que el índice de tráfico vehicular aumenta.

## **1.6 Alcance**

El desarrollo del presente proyecto se enfocó en verificar el algoritmo más adecuado de procesamiento de imágenes, mediante el análisis de estudios relacionados y la realización del BOT que conjuntamente permita la acepción, procesamiento y envío de la imagen de la plaza de aparcamiento disponible al usuario. Dicho sistema implementó en un sector del parqueadero cercano al departamento de Ciencias de la Computación de la Universidad de las Fuerzas Armadas ESPE.

El tiempo en el que se desarrolló el presente proyecto de investigación es de 12 meses.

## **1.7 Hipótesis**

La hipótesis o idea a defender del presente proyecto se plantea de la siguiente manera:

H1: La implementación de sistema empotrado en un lugar específico del parqueadero y el procesamiento digital de la imagen capturada conjuntamente con el programa informático Bot desarrollado, reduce el tiempo en el que el usuario pueda encontrar plazas disponibles de estacionamiento.

### **1.7.1 Estadística - Cuantitativa**

El método cuantitativo se utilizó para analizar los datos obtenidos de las encuestas y tener información específica de la realidad. H1:” El tiempo de estacionamiento está relacionado con la disponibilidad de plazas libres.” Explicación: La Hipótesis indica que, cuando una variable aumenta la otra disminuye, y si esta disminuye, la otra aumenta.

## **1.8 Metodología de la Investigación**

Para el desarrollo del presente proyecto fue necesario la aplicación de varias metodologías que guíen las fases para la obtención de la solución: Metodología de la Investigación Bibliográfica, Investigación Descriptiva y Metodología Design-Science.

La fase inicial del proyecto es el proceso investigativo en la que se proporcionó el conocimiento de las investigaciones ya existentes, de un modo sistemático, a través de una amplia búsqueda de: información, conocimientos y estudios relacionados con la temática (Rivas, 1994). Es decir, se debe realizar un estudio bibliográfico en bases digitales sobre el procesamiento de imágenes, las técnicas y patrones de procesamiento, los algoritmos de procesamiento y la

tecnología Bot, así como, los estudios relacionados sobre el tema de investigación, de manera que se conozca el estado de los estudios realizados y con ello orientar al proyecto al desarrollo de un modelo que sustente el problema mencionado.

La segunda fase empleó la investigación descriptiva con el propósito de orientar a la extracción de los estudios realizados; las características con mayor incidencia en el procesamiento de imágenes durante los últimos seis años con el fin de conocer la situación actual y real sobre las diferentes técnicas y algoritmos de procesamiento digital y el uso e implementación de la tecnología Bot.

En la fase final se aplicó la metodología Design-Science (DS) en sistemas de información la cual está orientada a la creación de artefactos exitosos (Pffers,2018). Siguiendo los siguientes pasos: identificación y motivación del problema, definición de los objetivos para una solución, diseño y desarrollo, demostración, evaluación y comunicación.

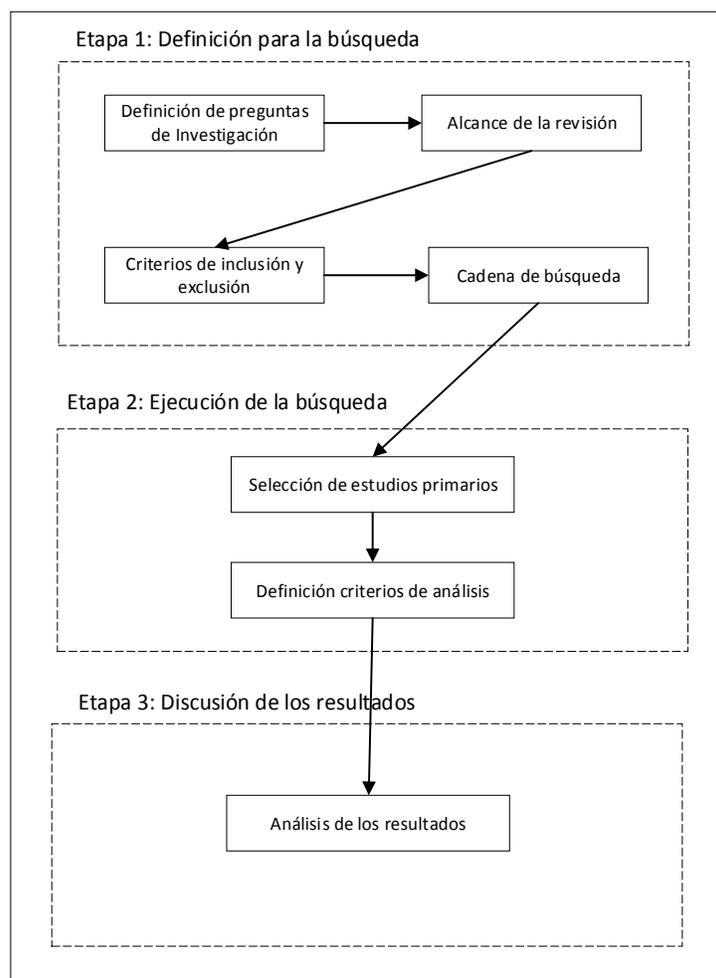
## CAPÍTULO II

### TRABAJOS RELACIONADOS Y MARCO TEÓRICO

#### 2.1 Estado del Arte

En el presente capítulo se examinó diferentes artículos relacionados con la temática de la técnica de procesamiento digital en una placa computadora de bajo coste como la Raspberry Pi, con el objetivo de analizar el estado del arte sobre procesamiento digital en la interpretación, procesamiento y envío de imágenes en un ordenador de placa reducida (SBC), obtenidos mediante la metodología común de los estudios de mapeos sistemáticos (SMS). Con la finalidad de mostrar una perspectiva general en el área de interés, a través del mapeo sistemático basado en el modelo propuesto por (Carrizo Moreno, 2015) y (Petersen, Feldt, Mujtaba, & Mattsson, 2008).

El modelo se define en tres bloques principales de la investigación como se muestra en la Figura 1. (1) Se define la pregunta de investigación, el alcance de la revisión, los criterios de inclusión/exclusión, y posteriormente la cadena de búsqueda. (2) La ejecución de la búsqueda, donde se define la selección de trabajos primarios y la de los criterios de análisis. (3) Por último, en la fase tres se definen los esquemas de caracterización y se analizan los resultados.



**Figura 1.** Proceso para la obtención del mapeo sistemático

### 2.1.1 Definición para la búsqueda

En la fase inicial del proceso de estudio de la revisión sistemática, debido a la gran diversidad de trabajos en el procesamiento digital y tecnología Bot, y el poco enfoque hacia qué temas de interés son los más sobresalientes en el estudio que antecede se plantaron las preguntas de investigación, seguido de la especificación del alcance de la revisión, seguido de los criterios de inclusión y exclusión, y finalmente la formulación de la cadena de búsqueda.

### 2.1.1.1 Definición de las preguntas de investigación

Para el presente estudio, se definieron las siguientes tres preguntas de investigación:

RQ1: ¿Existen estudios, investigaciones o trabajos relacionados que utilicen técnicas de procesamiento digital en Raspberry Pi? ¿Cuántos? ¿Qué técnicas utilizan?

RQ2: ¿Cómo ha incrementado el desarrollo de estudios en relación al tiempo?

RQ3: ¿Cuáles son las recomendaciones obtenidas tras la revisión sistemática de los trabajos?

Dichas preguntas presentan sus respuestas en la sección de resultados.

### 2.1.1.2 Alcance de la revisión

Este estudio se enfoca en la investigación de literatura mediante motores de búsqueda a través de librerías digitales con la búsqueda de información automatizada. Las bases de datos indexadas utilizadas en este estudio son las siguientes: ACM, SCOPUS, IEEE, desde 2013 al 2019. Se seleccionó este rango de fechas con el fin de conocer las posibles soluciones que han desarrollado durante los últimos seis años.

Las cadenas de búsqueda empleadas a través de las librerías digitales se especifican en la Tabla 1.

**Tabla 1***Cadenas de búsqueda en base de datos*

<b>Base de Datos</b>	<b>Cadena de Búsqueda</b>
<b>ACM</b>	recordAbstract:(+digital image processing technique +raspberry pi) OR (+algorithm process images +raspberry pi)
<b>SCOPUS</b>	TITLE-ABS-KEY ( ( image AND digital AND processing AND raspberry AND pi ) OR ( algorithm AND process AND images ) AND raspberry AND pi )
<b>IEEE</b>	((("All Metadata": digital processing) AND "All Metadata":image digital processing) AND "All Metadata":raspberry pi)

La cadena de búsqueda fue ajustada a los formatos correspondiente de cada base de datos. La búsqueda expulso un resultado de 58 documentos en ACM, 129 documentos en SCOPUS, y 45 documentos en IEEE. Los resultados numéricos se muestran en la Tabla 2.

**Tabla 2***Resultados numéricos por base de datos*

<b>Base de Datos</b>	<b>Numérico de estudios</b>
<b>ACM</b>	58
<b>SCOPUS</b>	129
<b>IEEE</b>	45
<b>Total</b>	232

### 2.1.1.3 Criterios de inclusión y exclusión

Con el objetivo de filtrar los estudios más relevantes y reducir el número de estudios obtenidos (232), es necesario descartar los estudios con ruido que podrían estar dentro de la búsqueda que antecede. Esto hace necesario definir los criterios de inclusión y exclusión.

Comenzando con los criterios de inclusión:

I1. Trabajos que solo guarden relación con el procesamiento digital de imágenes en Raspberry Pi.

I2. Publicaciones científicas que relacionen técnicas y algoritmos computacionales que permitan realizar el procesamiento de imágenes.

I3. Trabajos publicados entre 1 de enero de 2013 hasta el 1 de septiembre de 2019.

A continuación, los criterios de exclusión:

E1. Estudios no realizados en inglés.

E2. Publicaciones que no se puedan acceder por completo al contenido.

E3. Documentos no formales dentro de las bases indexadas.

E4. Documentos duplicados en otras bases de datos.

### 2.1.1.4 Cadena de búsqueda

En primera instancia al aplicar las cadenas de búsqueda se obtuvo un resultado de 232 trabajos. Después de aplicar los criterios de inclusión/exclusión se obtuvo un resultado menor de

trabajos con un valor numérico de 69 trabajos. Por consiguiente, a partir de los siguientes filtros de revisión que se detallan a continuación, se seleccionaron los estudios primarios.

### **Filtro 1**

F 1.1 Título: se inspecciona los títulos de los artículos resultado del motor de búsqueda.

F 1. 2 Abstract: se realiza la lectura de cada abstract o resumen de los artículos inspeccionados.

### **Filtro 2**

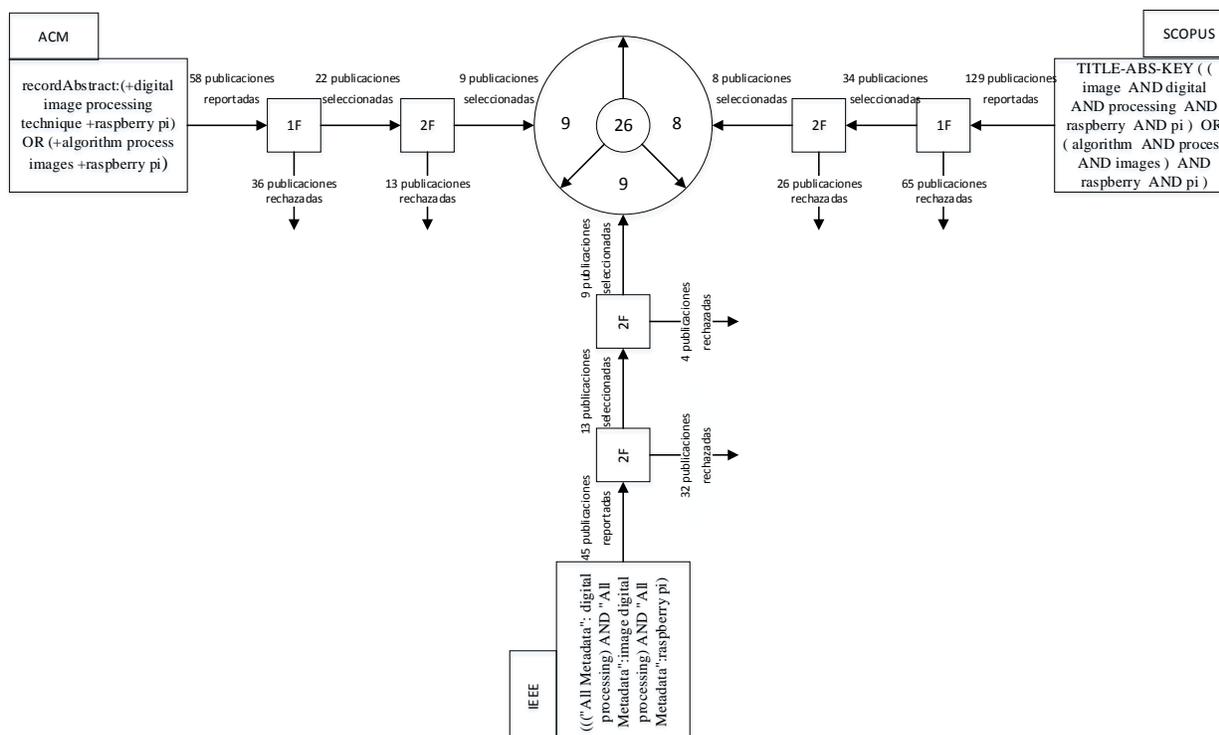
F 2.1 Documento completo: se procede a revisar y leer los artículos completos que pasaron el primer filtro.

#### 2.1.2 Ejecución de la búsqueda

En esta fase se procede a la ejecución de los filtros de revisión que anteceden, de tal manera que se obtuvieron 74 estudios primarios, para después definir los criterios que servirán de guía para evaluar y comparar los estudios entre sí.

##### 2.1.2.1 Selección estudios primarios

Una vez aplicados los filtros se obtuvo la cantidad de 26 estudios primarios cuyo resultado se puede identificar en la Figura 2.



**Figura 2.** Gráfico de búsqueda de estudios primarios

### 2.1.2.2 Definición criterios de análisis

Para poder realizar el análisis de los estudios primarios que se seleccionaron, se definió una gama de criterios que sirven de base para evaluar y comparar los estudios entre sí. Dichos criterios surgen con la finalidad de esclarecer y generar una vista panorámica de los estudios primarios seleccionados.

Los criterios determinados en este trabajo son los siguientes:

- Año: se identifica el año de publicación del estudio, de tal manera que su contexto quede más claro y como ha ido incrementando o disminuyendo el número de trabajos relacionados con la temática con el transcurso del tiempo.

- Tipo de documento: los tipos de documentos que se deben considerar son: conference paper y article, debido al impacto que poseen dichos estudios a nivel mundial.
- Foco: en este criterio se determinó tres categorías: imágenes, proceso, producto. En las imágenes, el contenido que indica la calidad, tamaño, resolución de la imagen para el procesamiento. En el proceso, el contenido referente al método de extracción de imagen, algoritmos, métodos de procesamiento digital. En el producto, se incluye el desempeño, la eficiencia, calidad del producto en el cumplimiento del objetivo.
- Área de aplicación: se determinó áreas como la medicina, deporte, docencia, ocio, biotecnología. Si hay un área no muy clara se definió como otras.

### 2.1.3 Trabajos Relacionados

En esta sección se realiza la búsqueda, lectura comprensiva y análisis de los documentos relacionados con el tema de investigación.

#### 2.1.3.1 Análisis de los trabajos

Con el objetivo de poder responder las preguntas de investigación planteadas anteriormente, se realiza la lectura comprensiva de los documentos primarios:

**RQ1: ¿Existen estudios, investigaciones o trabajos relacionados que utilicen técnicas de procesamiento digital de imágenes en Raspberry Pi? ¿Cuántos? ¿Qué herramientas utilizan?**

Desde el 1 de enero del año 2013 hasta el 1 de septiembre de 2019, se han publicado 232 artículos relacionados a la temática de procesamiento digital de imágenes y la utilización de Raspberry Pi. Sin embargo, una vez determinados y aplicados los criterios de inclusión y exclusión

se obtiene 26 artículos enfocados en la temática requerida donde emplean técnicas de procesamiento digital de imágenes y la utilización de Raspberry Pi. Los documentos se clasificaron en 9 áreas que se detallan en los siguientes párrafos.

## **Medicina**

El siguiente artículo se enfoca en la extracción de las características de la pupila del ojo dentro de las diferentes niveles de intensidad de las imágenes del ojo mediante el seguimiento de la pupila en tiempo real con el objetivo de analizar el comportamiento y la emoción de los alumnos en una plataforma de aprendizaje virtual como el cansancio o el nivel de atención (Deivanayagi, Nandhini Sri, Kalai Priya, & Aarthi, 2019). La cámara web captura el movimiento del ojo y se almacena en una Raspberry Pi, que realiza el procesamiento de imágenes en tres etapas: a) preprocesamiento, la imagen se transforma la imagen RGB en una imagen en escala de grises para después localizar la región ocular tomando como referencia la posición y postura de la cara; b) segmentación, utilizando el algoritmo Random Walker se obtiene la segmentación de la pupila del ojo, mediante el método de segmentación de imagen de etiquetas múltiples; y c) detección de la bola del ojo, previo a encontrar el ojo ocular es necesario rastrear el centro del iris realizado por la cascada de Haar mediante una secuencia de video.

En el mismo ramo, se propone un sistema de detección de cataratas basado en la placa Raspberry Pi utilizando una red neuronal convolucional profunda con GoogleNet mediante el aprendizaje de transferencia y el procesamiento de imagen digital en Matlab, basado en el sistema de clasificación de opacidades de lente III, en el lenguaje Python (Karamihan et al., 2019). La imagen capturada del ojo del paciente se utiliza como datos de entrada para la fase de procesamiento del sistema, la cual es el algoritmo de aprendizaje que realiza la Raspberry PI a

través del paradigma de procesamiento de imágenes digitales MATLAB y la red neuronal convolucional profunda con GoogLeNet, dando como resultado la imagen analizada con el tipo de catarata, el grado, la dureza donde se muestran los detalles generados en una pantalla LCD.

El siguiente trabajo describe un sistema biométrico multimodal (Sumathi, Poornima, & Haripriya, 2016) mediante el reconocimiento de la impresión de la palma de la mano del neonatal utilizando OpenCV en Raspberry. Por lo que permite identificar una herramienta de procesamiento digital útil en el desarrollo del presente. El reconocimiento de imagen se realiza utilizando la biblioteca OpenCV de código abierto en la plataforma Linux utilizando CMake, g ++, makefile. En la misma línea, se desarrolló un sistema que utiliza procesamiento de imágenes digitales a través de OpenCv, Python y la prueba Hirschberg (Yumang, Marquez, Paglinawan, Yamson, & Cuevas, 2019), como base para el algoritmo HOCORC para determinar si los pacientes padecen estrabismo y el tipo. El sistema es probado utilizando una Raspberry pi usando como complemento una Raspberry pi cámara de 5 megapíxel para la captura de imágenes.

Otro estudio, utiliza el procesamiento digital de imágenes como herramienta para reducir la complejidad en el análisis de la lengua (Velasco, Rojas, Ramos, Muaña, & Salazar, 2019). Se desarrolló un firmware capaz de extraer el color natural de la lengua y entrar una lista de síntomas posibles en el organismo interno del paciente. El proceso se divide en cuatro etapas: a) adquisición de las imágenes a través de la cámara Raspberry Pi, b) segmentación de imágenes, una vez separada la lengua de su fondo en escala de grises, la imagen se modifica a una imagen binaria a través del umbral, c) segmentación del cuerpo y pelaje, segmentación del recubrimiento de la lengua para identificar por completo el color del cuerpo y el revestimiento de la lengua, d) clasificación de colores, método por el cual se requiere la conversión del color RGB a color HSV (Tono, Saturación,

Color) y, e) red neuronal artificial, usando el lenguaje Python y las bibliotecas SciPy y Sci-kit, para el análisis de píxeles y la segmentación de imágenes y clasificación de colores.

El siguiente documento, con el objetivo de mejorar la comprensión de colores de una persona daltónica mediante el desarrollo de un prototipo con función de calibración personalizada, donde el dispositivo se ajusta a la deficiencia de la persona afectada, mediante el diseño del sistema de visión electrónica (Ostia et al., 2019). El proceso se resume en tres etapas: 1) la cámara usb genérica recopila la información de la imagen y la transmite a la Raspberry Pi, 2) el dispositivo contiene los procesos necesarios para identificar la deficiencia de visión color del individuo mediante la prueba de color Ishihara y 3) el proceso para que la imagen compensada de color se muestre en la pantalla LCD.

## **Educación**

Desarrollo de un juguete interactivo denominado Moya (Ahn et al., 2018), con el objetivo de que los niños desarrollen sus habilidades lingüísticas mediante la interacción entre el niños y el robot. Una de las actividades que ofrece es cuando el niño señala un objeto, el robot transmite el nombre del objeto por sonido e imágenes en el lenguaje seleccionado. Por un lado, la detección de objetos lo hace implementando la captación de video, con el objetivo de captar el objeto señalado y girar hacia él. Para ello se utilizó la arquitectura MobileNet que permite el reconocimiento de imágenes en tiempo real en hardware de recursos limitados (Juan Diego Polo, 2017). Mediante la captación de imágenes con la cámara de la Raspberry Pi captura la imagen con una resolución de 2592 x 1944 píxeles y mediante OpenCv reduce la resolución a 480 x 730 píxeles, para mejorar el procesamiento.

En el siguiente proyecto, los autores tienen como objetivo implementar un sistema de detección y reconocimiento facial a través de una red neuronal artificial mediante grabaciones de video. Las grabaciones son captadas a través de una cámara pi que mediante el módulo de Raspberry Pi primero transformará el video a un formato mp4 y se comunicará y enviará al servidor a través de una carpeta compartida para que el video sea procesado y determinar el estado de asistencia del alumno. El video se lee cuadro por cuadro, pero para ahorrar tiempo y recursos, no todos los marcos se leen para la extracción de rostros, sino que las caras marcadas se recortaran individualmente para crear las imágenes de caras por estudiante. La red neuronal artificial multicapa (ANN) extraerá las características faciales de las imágenes para posteriormente ser comparadas con las características faciales almacenadas en la base de datos, para finalmente elegir la mayor similitud de la imagen facial del estudiante a procesar (Castillo, Catanaoan, Obliopas, & Linsangan, 2018).

Uno de los objetivos del siguiente documento, describe el desarrollo y la evaluación de un proyecto que permite a los estudiantes desarrollar una aplicación para el procesamiento de imágenes digitales, en donde en la configuración debe incluir una Raspberry Pi y programas de procesamiento de imágenes. Para implementar los algoritmos de procesamiento de imágenes, los estudiantes deben utilizar los lenguajes de programación Octave y Python, estudiar algunos programas de procesamiento de imágenes con el objetivo de extraer los puntos destacados de una imagen, y escribir su programa de procesamiento de imágenes con el lenguaje de programación Python. Los procesos son los siguientes: importación de los módulos SimpleCV, definición del tamaño de la imagen, definición de la ventana de visualización y de la variable que tendrá la imagen matricial suministrada por la cámara, transformar la imagen en formato binario, dilatación y erosión. Y finalmente los estudiantes deben procesar las imágenes obtenidas de la cámara web con

el objetivo de obtener los puntos sobresalientes visualmente, unirlos entre dos imágenes y posteriormente comparar la carga computacional de SURF con las características robustas aceleradas y SIFT con la transformación de características invariantes de escala (Marot & Bourennane, 2017).

## **Seguridad**

El proyecto propone un sistema para eliminar la posibilidad de robo de tarjetas e identidad con el uso del sistema biométrico mediante el reconocimiento dactilar con mejora de imagen de transformación rápida de Fourier (Dimaunahan, Ballado, & Cruz, 2017). El prototipo se lo realizó en un cajero automático equipado con un escáner de huellas digitales, el cual estaba conectado a una Raspberry Pi que almacenaba temporalmente las imágenes para enviarlas al servidor de base de datos que estaba conectado en red de forma inalámbrica. Este estudio es interesante puesto que la Raspberry Pi cumple el papel de recibir las imágenes y enviarlas al servidor para que las procese y devuelva la respuesta a la Raspberry Pi, minimizando la carga del proceso en el dispositivo.

El siguiente artículo presenta un sistema de seguridad biométrica que permite el reconocimiento de los usuarios a través del procesamiento digital de imágenes de las venas del dorso de la mano capturadas en el espectro infrarrojo, utilizando la placa Raspberry Pi, una serie de integrados como pin de propósito general o GPIO, una pantalla LCD y servomotores, que conjuntamente permiten desarrollar un sistema de seguridad que monitorea e identifica los usuarios para la seguridad de información y espacios (Herrera, León, Londoño, & Vargas, 2017). Los procedimientos y la solución propuesta son: a) implementación de ondas infrarrojas, cuando se escanea la imagen de los rayos infrarrojos, el patrón de sangre que contienen la hemoglobina desoxigenada es visible y se manifiesta en forma de líneas oscuras que son interpretadas por el

dispositivo biométrico en el patrón de los vasos sanguíneos de la parte superior de la mano, los cuales son comparados con el patrón de los vasos sanguíneos previamente registrados del individuo, b) distancia de Hamming, definido como el número de bits que debe cambiarse para transformar un código válido de palabra en otra palabra válida (Herrera et al., 2017). Luego de convertir la imagen para obtener las características se toman valores que muestra la binarización y se ubican en un vector para posteriormente comparar la información de la imagen capturada con plantillas previamente registradas, para aplicar de este modo la distancia de Hamming, c) conversión a escala de grises, la imagen se convierte en una imagen de escala de grises para poder interpretar la imagen sin perder las propiedades de forma, d) suavizado de imagen, se aplica el filtro gaussiano para minimizar ruido por iluminación y ruido por el rendimiento de la cámara, e) binarización de imágenes, técnica de procesamiento de imágenes que convierte la imagen de entrada en una imagen digital formada por valores de 0 a 255, f) morfología, permite rellenar regiones, eliminar bordes, engrosar o adelgazar, entre otros, por lo que la imagen previamente debe convertirse a escala de grises; y g) prototipo de la estructura física, caja que impide el paso de luz ambiental para hacer posible la visualización de los vasos sanguíneos posterior de la mano, mediante la utilización de luz infrarroja colocada en la palma.

En el siguiente estudio, se pretende demostrar la capacidad de los algoritmos de procesamiento de imágenes en una plataforma informática pequeña. El proyecto se lo desarrollo para el reconocimiento de señales viales basado en un sistema integrado que lee y reconoce señales de velocidad (Bilgin & Robila, 2016). El proyecto utiliza el módulo de reconocimiento Tesseract OCR, siendo en términos de precisión, uno de los principales motores de reconocimiento de caracteres. Puede detectar letras en varias formas de imágenes, utiliza la biblioteca Leptonica de C de código abierto. Además, el autor indica que los métodos de captura encontrados en OpenCV

(captura, captura continua, secuencia de captura) deben considerarse de acuerdo con su uso y habilidades. En este caso, se eligió el método captura secuencial ya que indica que es el método más rápido.

En el siguiente documento se propone un desarrollo que implementa un sistema de bloqueo de puerta computarizado utilizando el reconocimiento facial a través de imágenes fotográficas capturadas por una cámara digital. Utilizando una placa Raspberry Pi y una cámara digital se captura la imagen del usuario a ingresar y se la envía al servidor, donde se realiza el reconocimiento facial. El reconocimiento facial es implementado por OpenCV, mediante la utilización de algoritmos de reconocimiento basados en PCA (análisis de componentes principales) (Hwang, Nam, Lee, & Jang, 2015).

## **Botánica**

El siguiente proyecto se enfoca en el desarrollo de un dispositivo portátil capaz de identificar plantas mediante el procesamiento digital de imágenes en una Raspberry Pi a través de las venas de las hojas, utilizando una base de datos de biometría (Selda, Ellera, Cajayon, & Linsangan, 2017). Uno de los objetivos del estudio es recopilar información de la base de datos con la utilización de los algoritmos SIFT (Scale-Invariant Feature Transform) y SVM (Support Vector Machines) para interpretación de los datos con el uso de matrices. El procesamiento de imagen lo hace mediante las siguientes etapas: a) a través del método de agrupamiento k.mean para la agrupación de colores de la hoja, promediando la intensidad más cercana de la hoja para formar un color, b) eliminación y cambio de color a negro del color que no se encuentre en el rango RGB, c) cambio de color de la imagen a escala de grises para ayudar en su procesamiento d) enmascaramiento a nivel de bits, e) una vez que la imagen se encuentre en formato binario

interpretar el gradiente de intensidad de la misma y reducción de ruido, f) ciclo repetitivo con una nueva imagen donde se comparara con la matriz de entrada SIFT y la información de la matriz almacenada en la base de datos. Finalmente, los investigadores (Selda et al., 2017) recomiendan para mejorar el procesamiento de imágenes, captar las imágenes con buenas condiciones de luz, distancia razonable en la toma de imagen entre el objeto y la cámara, filtración de ruido y finalmente utilizar el último Raspberry Pi debido a los recursos limitados de hardware.

El siguiente artículo, habla de la clasificación de las plantas según su uso medicinal, utilizando un procesador Raspberry Pi en tiempo real con el análisis de la imagen de la hoja. Dicha imagen se captura con una cámara digital de alta resolución sobre un fondo blanco en una superficie plana y se almacena en la memoria interna del procesador. Menciona que se debe garantizar la calidad y autenticidad de las hojas para la preparación de medicamentos a base de hierbas. El primer paso para el procesamiento de la imagen es la conversión: convierte la imagen a una imagen en escala de grises, mejorando la imagen para obtener una mejor claridad en la región de interés. El segundo paso es la operación morfológica, aplica un elemento estructurante a una imagen de entrada generando una imagen de salida del mismo tamaño. Las actividades básicas de la operación morfológica es la dilatación, proceso de agregar pixeles a los límites del objeto en una imagen, mientras que la erosión es el proceso adverso al eliminar pixeles en los límites (Vijayashree & Gopal, 2017).

El siguiente artículo, realiza la extracción de imagen del índice de vegetación de diferencia normalizada (NDVI) de una tierra agrícola, mediante un cuadricóptero autónomo con una cámara de filtro modificada (Daroya & Ramos, 2018). Utiliza el enfoque basado en características clave en las imágenes como base para las superposiciones. Los enfoques mencionados basados en características son la función robusta acelerada (SURF) y la transformación de características

invariantes de escala (SIFT); en términos de resolución de imagen, iluminación, rotación, y escala, el algoritmo SIFT puede ser más robusto, especialmente cuando procesa imágenes que tienen características rotativas y condiciones de iluminación variables. Mientras que SURF es más robusto y menos pesado informáticamente proporcionando una resistencia parecida a la variación de la imagen.

## **Biometría**

El siguiente artículo, busca la coincidencia de patrones venosos, al analizar los patrones de los vasos sanguíneos visibles en la superficie de la mano a través de iluminación infrarroja (Linsangan, Estacio, & Torres, 2019). Presenta una comparativa de entre ORB (Oriented FAST and Rotated BRIEF) y SIFT (Scale Invariant Feature Transform) como algoritmos descriptores para el reconocimiento de la vena dorsal de la mano. En la parte del hardware utiliza un dispositivo portátil de escaneo de imágenes con la Raspberry Pi, luces infrarrojas, cámara CCD (dispositivo de carga acoplada) sensible a infrarrojos; y bibliotecas OpenCV para la programación. Las imágenes son procesadas previamente usando CLAHE (Contrast limited adaptive histogram equalization) para mejorar el nivel de visibilidad de la imagen o video (Yadav, Maheshwari, & Agarwal, 2014), y finalmente se usan como clasificadores de imágenes se usan Brute Force Matcher y FLANN, que implementan el cálculo de distancia entre puntos de rasgos de la imagen. Como resultado se obtuvo que ORB ofrece mejor rendimiento que SIFT.

El concepto del siguiente estudio, propone un sistema denominado EagleEye (Zhao, Liang, Zhao, Tang, & Wang, 2019) para mejorar el rendimiento en tiempo real en dispositivos integrados de recursos limitados de hardware. EagleEye presenta un buen equilibrio entre alta precisión y velocidad rápida en Raspberry Pi 3 B +. El estudio, habla sobre cinco estrategias para diseñar

detectores faciales efectivos con buen equilibrio de precisión y velocidad de operatividad. Las primeras dos estrategias ayudan a construir un detector facial de baja complejidad de cálculo, pero con suficiente capacidad. Y las tres siguientes estrategias, mejoran notablemente la precisión del detector sin agregar costos excesivos.

En la siguiente investigación, la técnica de reconocimiento facial se utiliza para detectar y contar el número de pasajeros en un vehículo eléctrico a través de una cámara web conectada a la Raspberry Pi 2 B (Wittayaprapakorn & Yooyativong, 2017). Cuando el vehículo eléctrico sale del estacionamiento, la cámara captura las imágenes de los rostros de los pasajeros que van en los asientos. Después las imágenes se ajustan y mejoran para reducir el ruido. Las imágenes son enviadas al servidor mediante comunicación 3G, donde son procesadas mediante tecnología de detección de rostros, el número de pasajeros y las vacantes en el vehículo eléctrico. Este sistema utiliza las librerías OpenCV para analizar y procesar los datos y luego calcular la vacante del vehículo eléctrico utilizando la información de detección de rostro máximo. Las características de Haar-likes es una técnica popular para la detección de rostros humanos, la técnica fue propuesta por Paul Viola y Michael Jones en el año 2001 (Viola & Jones, 2001). Las características de los algoritmos de Haar-like son separar la imagen desde la imagen de entrada a la ventana secundaria y escaneo facial. Usan una técnica de imagen integral para encontrar la suma de píxeles dentro de la imagen y después usan el detector que puede cambiar el tamaño y la posición para encontrar las áreas blancas y negras. El siguiente paso es llamar a AdaBoost, proceso para la clasificación de datos al aumentar el peso a la clasificación de una cara hasta detectar la mejor cara, y la última etapa es el clasificador en cascada, es decir, separa la imagen a la ventana secundaria y comprueba en la ventana secundaria para encontrar la cara. Si la ventana secundaria tiene la posibilidad de tener el rostro facial, pasa al siguiente clasificador aumentando el peso hasta detectar la mejor cara.

El siguiente estudio, se enfoca en la compresión de la imagen, debido a que inicialmente tiene gran cantidad de datos digitales por lo que es necesario reducir el volumen para su transmisión y preservación (Sahitya, Lokesha, & Sudha, 2017).

Este punto es importante en nuestro estudio, puesto que el procesamiento digital de imágenes debe realizarse con imágenes con el menor ruido y volumen posible para poder realizarlo en el menor tiempo posible.

La compresión de imágenes las realiza utilizando la placa Raspberry Pi que ayuda a preservar gran cantidad de imágenes y su calidad. Además, permite la implementación del método de compresión, transformada discreta bidimensional del coseno, más utilizado para dar el formato JPEG utilizando la biblioteca OpenCV. Y el algoritmo de compresión JPEG se utiliza para comprimir las imágenes fijas a color que dan buena impresión visual después de haber sido comprimida (Sahitya et al., 2017).

## **Robótica**

En el artículo, un robot humanoide es convertido en un robot de juego de fútbol con sistema de visión (Krishnan, Aswath, & Udupa, 2014). El robot está integrado con una cámara CMOS VGA conectada a una Raspberry Pi en la plataforma Linux utilizando la librería OpenCV y los algoritmos escritos en C++. Una vez se captura la imagen con la cámara, se envía a la Raspberry Pi para su procesamiento en función de los algoritmos programados, este dispositivo está interconectado con el controlador CM530 donde se programan los movimientos de los servos.

En el siguiente artículo, se analiza el desarrollo de un sistema de detección de piezas cilíndricas para ayudar a un brazo robótico industrial, mediante tecnologías de bajo costo como son, el módulo Picamara, la placa Raspberry Pi, y la biblioteca de código abierto OpenCv, para el

desarrollo del sistema de visión y el procesamiento digital de imágenes (Araujo, Araujo, Silva, & Regis, 2019). En este trabajo se empleó dos algoritmos, el primero para la captura y rectificación de la imagen siguiendo las siguientes etapas: envío de parámetros, cálculo de homografía, rectificación de imagen, segmentación de imágenes; y el segundo para la detección de piezas cilíndricas colocadas en la plataforma de producción.

Por un lado, el algoritmo empleado en la calibración de imagen sigue el método tradicional de OpenCV, el cual requiere solo una calibración estándar plana, donde se capturan 30 imágenes del patrón de calibración. Como resultado del proceso, se obtienen parámetros usados para la distorsión y el cálculo de la matriz homografía a través del DLT (Transporte Lineal Directo), encargado de la conversión del sistema de coordenadas en píxeles de la imagen para el sistema de coordenadas cartesianas en milímetro. La matriz homografía y los coeficientes de distorsión se envían al algoritmo principal para que se ejecute la corrección del sistema de coordenadas y la distorsión. En el siguiente paso, se realiza la bancarización de la imagen a través del algoritmo umbral, que por pruebas empíricas, dio como resultado un valor umbral de 50 en una escala de 0 a 255 como valor umbral óptimo en condiciones de iluminación regulares (Araujo et al., 2019).

### **Agropecuario**

La presente investigación, propone aumentar el control de calidad, identificando huevos de gallina adecuados para la incubación, utilizando algoritmo de procesamiento de imágenes (Fernández-S et al., 2019). El diseño es un cubo en una base estructural, contenido por una matriz led en su parte inferior para posicionar los huevos por transiluminación y en la parte superior, un sistema mecánico que permite el deslizamiento de una cámara digital para realizar los respectivos análisis. Este sistema se desarrolló utilizando una Raspberry Pi 3, como unidad de control central,

una cámara, el sistema operativo Raspbian, el lenguaje de programación Python para el desarrollo de los algoritmos de procesamiento de imágenes y la librería de OpenCV. En la placa Raspberry, se implementó servidores web y una base de datos para el almacenamiento de la información obtenida en los análisis.

Por un lado, basado en algunos métodos de OpenCV para el procesamiento de imagen se sigue las siguientes pautas: 1) conversión a escala de grises, utilizando el método `cvtColor ()` que recibe como parámetros la imagen a convertir y el código de conversión de color para obtener una imagen con las mismas dimensiones en escala de grises, 2) reducción de ruido mediante filtro gaussiano, utilizando el método `GaussianBlur ()` que recibe como parámetros la imagen en escala de grises, el tamaño del núcleo y el método de extrapolación de píxeles. Una vez aplicado el filtro gaussiano, se obtienen una matriz basa en la desviación estándar, 3) detección de contornos, utiliza el método `findContours ()` que recibe como parámetros la imagen binaria, la relación de contornos y el método de aproximación, 4) segmentación, es necesario segmentar la imagen procesada para realizar un análisis independiente 5) comparación de descriptores de forma, se utiliza el método `Canny ()`, que recibe como parámetros la imagen filtrada y los límites mínimo y máximo para el procedimiento de histéresis. Finalmente, se aplica el método `matchShapes ()` que recibe como parámetro dos imágenes para comparar en escala de grises y retorna el valor de similitud o métrico. Por otro lado, en la Raspberry Pi 3 se carga una base de datos administrada por phpMyAdmin y MySQL la información adquirida después de analizar el producto y además se implementan servidores web centralizados (Fernández-S et al., 2019).

## **IoT**

En este estudio, se propone un sistema digital para el procesamiento automático de video basado en la concepción del Internet de las cosas usando Raspberry Pi. En el modelo, se utilizó dos Raspberry Pi una como master y otra como esclavo dependiendo de cuál se utilice primero, ambas con su respectiva dirección IP. Toda la configuración está basada en IoT, si se reproduce el botón de video en la GUI de Raspberry Pi, comienza a grabar la cámara de la Raspberry Pi master y además configura la IP del esclavo y automáticamente el procesamiento de video se realizará utilizando el código OpenCV y al completar el procesamiento del video se reproducirá en la cámara de la Raspberry Pi esclavo y viceversa. Por otro lado, el autor indica que este modelo es rentable y a medida de que el requisito aumenta se puede unir más Raspberry Pi y formar un clúster (Mande & Lakhe, 2018).

## **Civil**

El siguiente artículo, presenta pruebas preliminares del análisis de un nuevo sistema fotogramétrico de bajo costo para el modelado 4D de áreas a gran escala para aplicaciones en el campo de la ingeniería civil. El sistema consta de cinco unidades independientes, cada una de las cuales está compuesta por una Raspberry Pi 2 B conectada a un módulo de picamara (8mp), alimentada por un panel solar de 10 W. Por un lado, la adquisición de imágenes automática se hace a través scripts de Python y la librería OpenCV. Las imágenes son capturas en diferentes momentos del día y enviadas al servidor FTP desde donde se puede acceder para el procesamiento. La adquisición y carga automática de la cámara se la realiza en el servidor FTP utilizando scripts en Python. Las cinco unidades se encuentran sincronizadas para recoger imágenes simultáneamente (Santise, Thoeni, Roncella, Sloan, & Giacomini, 2017).

A continuación, un resumen de la identificación de los estudios primarios que se analizaron en la Tabla 3, donde se puede visualizar los autores, el año de publicación, tipo de documento, foco y área de conocimiento.

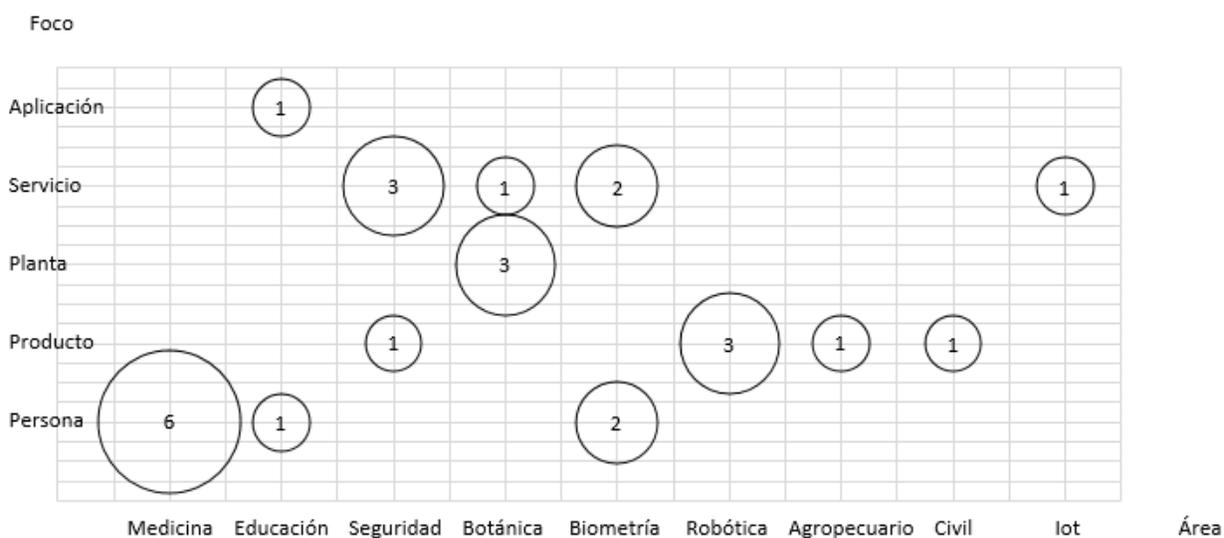
**Tabla 3**

*Esquema de caracterización de estudios primarios*

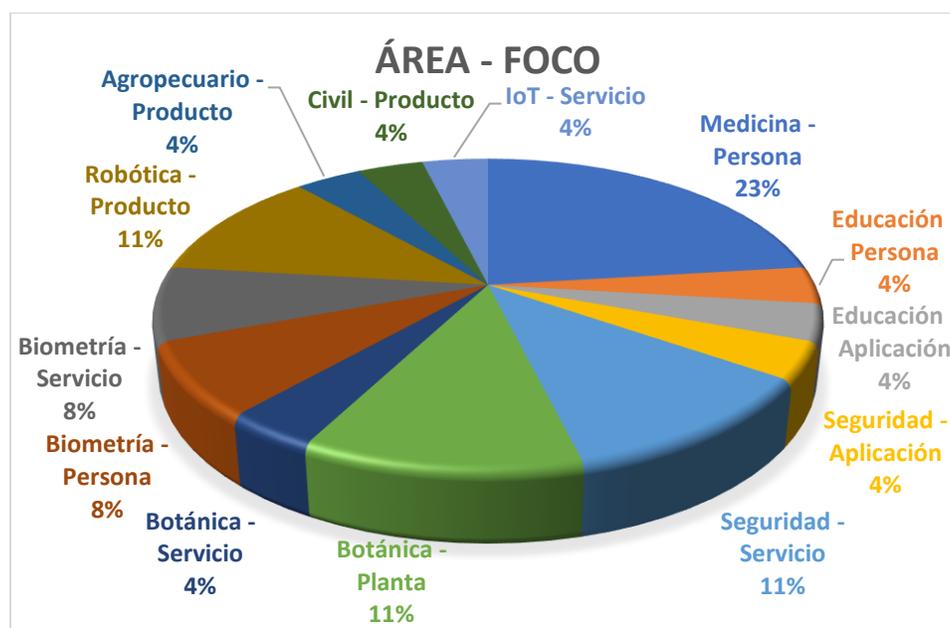
<b>Autores</b>	<b>Año</b>	<b>Tipo</b>	<b>Foco</b>	<b>Área</b>
(S. Sumathi, R. Poornima, T. Haripriya)	2016	C. Paper	Persona	Medicina
(Yumang, Analyn & Marquez, Daniel & Paglinawan, Arnold & Yamson, Kenneth & Cuevas, Edrick)	2019	C. Paper	Persona	Medicina
(Ji Yoon Ahn, Dong Wan Kim, Yong Hyeon Lee, Woori Kim, Jeong Kuk Hong, Yeonbo Shim, Jin Ho Kim, Juhyun Eune, and Seong-Woo Kim)	2018	C. Paper	Persona	Educación
(Ericson D. Dimaunahan, Alejandro H. Ballado, Jr., Febus Reidj G. Cruz)	2017	C. Paper	Producto	Seguridad
(Jesse Dave S. Selda, Roi Martin R. Ellera, Leandro C. Cajayon, II, Noel B. Linsangan)	2017	C. Paper	Planta	Botánica
(Noel B. Linsangan, Rosalina R. Estacio, Jumelyn L. Torres)	2019	C. Paper	Persona	Biometría
(Arjun B. Krishnan, S. Aswath, Ganesha Udupa)	2014	C. Paper	Producto	Robótica
(Ranielle Vincent C. Castillo, Mel-john T. Catanaoan, John Ferdinand E. Obliopas, Noel B. Linsangan)	2018	C. Paper	Producto	Robótica
(Jaejoon Hwang, Yoojin Nam, Sangheon Lee, Giljin Jang)	2015	C. Paper	Servicio	Seguridad
(Karamihan, K.C., Agustino, I.D.F., Bionesta, R.B.B., (...), Arellano, S.V.E., Esguerra, P.A.M.)	2019	Article	Persona	Medicina
(Deivanayagi, S., Nandhini Sri, V.G., Kalai Priya, P., Aarthi, G.)	2019	Article	Persona	Medicina

(Velasco, J., Rojas, J., Ramos, J.P., Muaña, H.M., Salazar, K.L.)	2019	Article	Persona	Medicina
(Zhao, X., Liang, X., Zhao, C., Tang, M., Wang, J.)	2019	Article	Persona	Biometría
(Fernández-S, Á, Salazar-L, F., Jurado, M., (...), Moreno-P, R., Buele, J.)	2019	C. Paper	Producto	Agropecuario
(Daroya, R., Ramos, M.)	2018	C. Paper	Planta	Botánica
(Santise, M., Thoeni, K., Roncella, R., Sloan, S.W., Giacomini, A.)	2017	C. Paper	Producto	Civil
(Sahitya, S., Loksha, H., Sudha, L.K.)	2017	C. Paper	Servicio	Biometría
(Bárbara Araújo ; Julierme Araújo ; Rafael Silva ; Danilo Regis)	2019	C. Paper	Producto	Robótica
(Rangel Daroya ; Manuel Ramos)	2017	C. Paper	Planta	Botánica
(Julien Marot ; Salah Bourennane)	2017	C. Paper	Aplicación	Educación
(T. Vijayashree ; A. Gopal)	2017	C. Paper	Servicio	Botánica
(Wilder D. Herrera ; Over L. León ; Williams C. Londoño ; Javier A. Vargas)	2017	C. Paper	Servicio	Seguridad
(Vaishnavi Mande ; Mandar Lakhe)	2018	C. Paper	Servicio	IoT
(Conrado F. Ostia ; Dionis A. Padilla ; Febus Reidj ; G. Cruz ; Roy Jayson D. Galang ; Anne Shirley M. Josafat ; Edrick F. Victoria)	2019	C. Paper	Persona	Medicina
(Janewit Wittayaprapakorn ; Thongchai Yooyativong)	2017	C. Paper	Servicio	Biometría
(Enis Bilgin ; Stefan Robila)	2016	C. Paper	Servicio	Seguridad

En la Figura 3 se puede observar un resumen del esquema de caracterización de los estudios de la Tabla 3, en la cual se puede observar la clasificación en áreas y focos que sirvieron de base para poder realizar el gráfico (Figura 4) y su posterior análisis.



**Figura 3.** Gráfico de burbuja del mapeo sistemático



**Figura 4.** Gráfico de anillo de estudios primarios

En la Figura 4, se realizó un resumen agrupado por Área y Foco de los estudios analizados, donde se puede constatar que la gran parte de trabajos se centran en el área de Medicina y la categoría Persona con el 23% de estudios que pretenden mejorar y facilitar el diagnóstico del

especialista médico. En segundo lugar, con el 11% se encuentran tres resultados: a) Seguridad-Servicio con estudios que ofrecen el servicio de reconocimiento facial para brindar seguridad en diferentes áreas, b) Botánica-Planta, con estudios que se enfocan en el reconocimiento digital de la hoja para reconocer la planta en el área de la botánica y, c) Robótica-Producto con estudios que ofrecen productos de robótica que realizan el procesamiento de imágenes para realizar alguna actividad. Y finalmente estudios con el 4% en IoT-Servicio, Botánica-Servicio, Agropecuario-Producto, y Civil-Producto.

## RQ2: ¿Cómo ha incrementado el desarrollo de estudios en relación al tiempo?

Para saber si existe alguna relación entre los trabajos que utilizan técnicas de procesamiento digital en Raspberry Pi y el tiempo en que se han realizado las publicaciones, con la finalidad de examinar si existe una tendencia temporal de las publicaciones. Para ello, se ha realizado el gráfico (Figura) con el total de artículos analizados.



**Figura 5.** Total artículos publicados por año y base de datos

La Figura 5 indica el número de estudios de la revisión sistemática entre el año 2013 al 2019. Se visualiza que con el transcurso de los años han aumentado los estudios moderadamente entre el año 2017 y 2019. Además, existe un indicio de aumento progresivo de publicaciones en las bases de datos (ACM, SCOPUS, IEEE) en los años 2018 y 2019, siendo el año 2018 con 63 publicaciones el año en el que hubo más auge sobre el procesamiento digital de imágenes en placas de hardware limitado. En resumen, este análisis nos indica que el número de publicaciones referentes a la temática ha ido aumentando proporcionalmente en el transcurso de los años, volviéndose un foco de interés dentro de la comunidad científica.

### **RQ3: ¿Cuáles son las recomendaciones obtenidas tras la revisión sistemática de los trabajos?**

En el presente proceso de investigación, se enfrentan varios desafíos técnicos, debido al número bajo de publicaciones que emplean limitadas técnicas de procesamiento digital en placas de hardware limitado como la Raspberry Pi, por lo que para realizar el procesamiento digital de imágenes deben utilizarse técnicas y algoritmos que no requieran un nivel alto de recursos.

La mayor parte de artículos utilizan la librería OpenCV en la Raspberry Pi para realizar parte de la técnica de procesamiento de imágenes respaldados de la utilización de un servidor web y el modulo para la cámara Pi propia de Raspberry Pi. Además, utilizan las últimas versiones hardware de Raspberry Pi para tener mayores recursos.

El campo donde se puede aplicar las técnicas de procesamiento digital ha ido aumentando proporcionalmente con el transcurso de los años. Siendo aplicadas en áreas como la medicina, la robótica, la biometría, la seguridad, entre otras.

En la parte del hardware se empleará una Raspberry Pi en su versión 3 con memoria Ram de 1 GB LPDDR2 SDRAM, con una tarjeta de memoria de 16GB, con puerto CSI para conectar una cámara y conexión HDMI. También, se empleará la cámara NoIR con resolución de 5mpx, con módulos de infrarrojos externos (si fuesen necesarios).

En la parte del software se instalará el sistema operativo Raspbian en la Raspberry Pi, la programación se realizará en Python, debido a la compatibilidad con el API de Telegram, los módulos de la cámara, y las librerías de procesamiento de imágenes.

Para el almacenamiento de las imágenes se utilizará la tarjeta de memoria y una memoria externa (si fuera necesario). Para el almacenamiento de las peticiones y resultados se empleará la base de datos MySql y para su administración la herramienta phpMyAdmin.

Las técnicas empleadas para el procesamiento digital que se utilizaran son:

- Eliminación y cambio de color a negro dentro del rango RGB
- Conversión a escala de grises
- Segmentación y comparación

## 2.2 Marco Teórico

### 2.2.1 Procesamiento digital de imágenes

El procesamiento digital de imágenes (PDI) es el conjunto de técnicas y procesos para extraer, analizar información contenida en una imagen. El objetivo principal del análisis digital es el facilitar la interpretación de la información contenida en las imágenes. El sistema de análisis tiene como parámetro de entrada una imagen, y como resultado normalmente una salida numérica relacionada al contenido de la imagen ingresada (Domínguez Torres, 1996).

Actualmente, el PDI es un área de investigación específica en informática, relacionada con el procesamiento digital de señales. Esto se debe a que el PDI es una forma particular de procesamiento digital de señales en dos o tres dimensiones.

El estudio del PDI se basa en dos áreas de aplicación principal:

- a) Mejoramiento de la calidad: información contenida en una imagen con el objetivo que dicha información sea interpretada por las personas.
- b) Procesamiento de los datos: procesamiento de la información contenida en un escenario a través de una máquina capaz de recibir, interpretar y comprender dichos datos.

Las aplicaciones de PDI han ido creciendo día tras día, con ello también el número de áreas de aplicación del PDI, a continuación, una lista de algunas áreas donde se puede aplicar:

- Lectura de documentos
- Cartografía y SIG
- Crecimiento de bacterias
- Modelación geológica
- Imágenes médicas
- Navegación autónoma

- Reconocimiento de objetos
- Astronomía
- Percepción remota
- Seguridad fotográfica
- Mapas climatológicos
- Impresión de huellas digitales
- Metamorfosis de imágenes
- Transferencia y comparación de documentos

### 2.2.2 Imagen digital

Una imagen digital es la representación bidimensional de una imagen empleando bits, es decir, que los elementos que lo conforman solo podrán tener valores formados por las combinaciones de ceros y unos. Según (Gómez, 2015), entre las características de la imagen digital se puede destacar: píxel, resolución, tamaño, tipos y formatos de imagen, profundidad de color, compresión, modelo de color, relación aspecto, entre otros.

#### 2.2.2.1 Tipo básico de imágenes digitales

La información digital al ser discontinua toda imagen de este tipo debe estar dividida en unidades identificables que contengan cada una un conjunto de información determinada (Duarte Aguirre & González Roque , 2017), siguiendo este principio existen dos tipos de imágenes digitales:

- Imágenes de mapa de bits o rasters.
- Imágenes vectoriales.

#### 2.2.2.1.1 Imágenes de mapa de bits o rasters

Las imágenes de mapa de bits se tratan de imágenes compuestas de píxeles dispuestos en una retícula y que cada píxel tiene uno o más valores que describen su color (Martín, 2009). Toda imagen de mapas de bits tiene un número invariable de píxeles a determinada resolución según su tamaño, y este varía si varía su resolución.

Como ejemplos de mapas de bits se puede mencionar: fotografías, imágenes captadas por un escáner, elementos gráficos, entre otros.

##### Características

- Estas imágenes son fiel representación de la realidad.
- Son fácilmente manipulables.
- No es posible ampliar un mapa de bits sin afectar su calidad de resolución.
- Ocupan gran capacidad de espacio para su almacenamiento.

#### 2.2.2.1.2 Imágenes vectoriales

Las imágenes vectoriales se tratan de gráficos conformados por una serie de puntos que se encuentran conectados por líneas y curvas por medio de elementos geométricos definidos como vectores.

##### Características

- Estos gráficos se caracterizan por sus contornos perfectos.
- Los efectos que se apliquen a estas imágenes pueden ser rectificadas, ya que son editables.
- Es posible modificar el tamaño de los componentes de la imagen vectorial sin que se produzca pérdida de información.
- Se almacena en archivos compactos.

### 2.2.3 Técnicas de Procesamiento Digital de Imágenes

Se ha logrado un gran avance en el desarrollo productos que utilizan técnicas para el procesamiento digital de imágenes en los últimos años. Principalmente, el proceso digital de imágenes se puede dividir en las siguientes técnicas (Trejos & Guzmán, s. f.):

- Adquisición
- Realce y mejora
- Segmentación
- Extracción de características

#### 2.2.3.1 Adquisición

Se define como la obtención de la imagen tanto como captura por un dispositivo digital como una cámara digital o la digitalización por medio de escáner. La adquisición se realiza mediante cuatro procesos (Trejos & Guzmán, n.d.):

- a) **El proceso de formación de imágenes:** Proceso mediante el cual la información luminosa en 3D es proyectada en un plano 2D (imagen digital), mediante dispositivos como las cámaras digitales.
- b) **Representación de imágenes digitales:** la intensidad luminosa  $f(x,y)$  se puede caracterizar por dos componentes: la cantidad de luz incidente y la cantidad de luz reflejada por los objetos de la escena. Como se mencionó anteriormente se puede dividir en dos grandes grupos:
  - a. Imágenes de mapa de bits o rasters

- i. Imagen binaria: donde cada celda de la matriz de una imagen binaria está representada por un píxel con la información mínima de un bit. Cuando bit es 0, el color es negro; y si el bit es 1, el color es blanco.
  
- ii. Imagen a escala de grises: donde cada celda de la matriz de la imagen en escala de grises se representa por 1 píxel equivalente a 1 byte, permitiendo 256 niveles de gris. Desde el 0 que representa el color negro, hasta el 255, con el color blanco.
  
- iii. Imagen a color: donde cada celda de la matriz de la imagen a color está representada por 1 píxel equivalente a 3 bytes. En este caso, cada píxel consta de 3 valores (rojo, azul, verde), cada color equivale a un byte y tiene posibles 16.7 millones de colores.

b. Imágenes vectoriales

- c) **Dispositivos de captura**: una imagen digital puede ser capturada por dos tipos de dispositivos, la cámara fotográfica digital o el escáner. Existen muchos tipos de cámaras digitales según su propósito, su resolución, calidad de imagen. Así, como existen diferentes tipos de escáner, como planos, de microfilm, etc.

**d) Formatos de almacenamiento:** formato estándar de organizar y almacenar los datos que presentan la imagen. Existen diferentes formatos según la necesidad de almacenamiento. Algunos de los más utilizados son: jpg, gif, bmp, tiff.

### 2.2.3.2 Realce y mejora

El realce o mejora de imágenes es una de las principales técnicas de procesamiento digital de imágenes. Se utiliza para facilitar un análisis computarizado de una aplicación específica, una vez digitalizada la imagen, es posible mejorar su apariencia visual, su calidad o transformarla. Esta técnica comprende un conjunto de operaciones que tienden a mejorar la calidad visual de la imagen. Permitiendo realizar características de contraste y brillo de una imagen, reducir el ruido, o intensificar o reducir detalles contenidos en la misma. Para ello se puede emplear dos técnicas: realce en el dominio espacial y realce en el dominio de la frecuencia (Aldalur, 2002).

### 2.2.3.3 *Técnicas de realce en el Dominio Espacial*

#### 2.2.3.3.1 Realce punto a punto

Son aquellas técnicas donde las operaciones se realizan sobre el valor del píxel, sin tener en cuenta los píxeles aledaños. Estas transformaciones se agrupan en dos principales técnicas, **transformaciones de intensidad simples y procesamiento de histograma (Aldalur, 2002).**

#### **a) Transformaciones de intensidad simples**

Se utiliza para mejorar condiciones de bajo contraste, baja luminosidad o demasiada claridad. En estas técnicas de realce, el valor de  $g$  en el punto  $(x, y)$  depende del valor de  $f$  en ese punto. Por tanto, la función  $T$  que define la operación sobre el píxel quedando la ecuación de la siguiente forma (1):

$$f(x, y) = T(g(x, y)) \quad (1)$$

Ese operador es conocido como **función de transformación de nivel de gris**. Donde las operaciones más frecuentes son:

- i. **Imagen en escala de grises:** consiste en obtener una imagen monocroma (definida en dos colores) a partir de una imagen en color. Si se emplea el RGB se puede utilizar la siguiente ecuación para obtener la señal lumínica (2).

$$Y = 0.299R + 0.587G + 0.114B \quad (2)$$

- ii. **Negativo de imágenes:** consiste en invertir los niveles de blanco a negro con el objetivo de obtener el negativo de la imagen de entrada. Particularmente, si la señal está en la escala de grises comprendida en el rango  $[0, L-1]$ , siendo 0 el nivel de negro y  $L-1$  el nivel de blanco, la transformación lineal se representaría por la siguiente ecuación (3).

$$s = (L - 1) - r \quad (3)$$

Donde  $s$  y  $r$  son variables indicadoras del nivel de gris de  $f(x, y)$

Los negativos de imágenes son útiles para representar imágenes médicas como las radiografías tomadas con Rayos X.

- iii. **Aumento del contraste:** debido a diferentes causas en las imágenes con poco contraste se puede utilizar esta técnica que consiste en incrementar el rango dinámico de gris de la imagen.

- iv. **Comprensión del rango dinámico:** se suele dar esta anomalía cuando el rango dinámico de una imagen procesada excede la capacidad del dispositivo de presentación, en este caso, aparecerá en la pantalla sólo las partes más brillantes de la imagen. Para solucionarlo, será necesario comprimir el rango dinámico de los valores de cada píxel (4).

$$|s = c \log(1 + |r|) \quad (4)$$

Donde c es un factor de escala.

- v. **Fraccionamiento del nivel de gris:** se utiliza para destacar un rango específico del nivel de gris de una imagen. Entre las formas para lograrlo, se puede adjuntar un valor alto a todos los niveles de gris del rango de interés y un valor bajo a los restantes y preservando el fondo.

## b) **Procesado de histograma**

Proceso por el que se normaliza la imagen de tal manera que el valor de gris más pequeño es llevado a 0 y el máximo a 255. Para poder variar el aspecto de las imágenes se utiliza el histograma, que es una función discreta que representa el número de píxeles en la imagen a aplicar, en relación de los niveles de intensidad, g. Probabilidad P(g) de ocurrencia de un determinado nivel g (5):

$$P(g) = N(g) / M \quad (5)$$

Donde M es el número de píxeles de la imagen y N(g) es el número de píxeles en el nivel n de intensidad g.

#### 2.2.3.3.2 Realce con operaciones entre vecinos

Técnicas cuyas operaciones son realizadas sobre un píxel tomando en cuenta sus píxeles vecinos. Se utiliza cuando se busca reducir ruido, detectar bordes o generar algún tipo de efecto artístico. Las operaciones entre vecinos son también conocidas como filtrado espacial

- a) **Filtrado espacial:** modifica el valor de cada píxel de acuerdo con los valores de los píxeles que lo rodean.
- b) **Filtrado por bajas:** se obtiene el promediado de los píxeles contenidos en el entorno de la máscara utilizada. Básicamente consiste en remplazar el valor de cada píxel en una imagen por el promedio de los niveles de gris del vecindario definido por la máscara.
- c) **Filtrado de mediana:** consiste en reemplazar el valor del píxel central por la mediana de los niveles de gris del vecindario de ese píxel.

#### 2.2.3.4 Técnica de realce en el Dominio de la Frecuencia

Hasta ahora la imagen digital se consideró como una matriz cuyos píxeles representan un valor proporcional al nivel de brillo de la imagen. Sin embargo, la misma imagen puede ser considerada como una función no periódica, definiéndose en otro espacio bidimensional cuyos ejes vengan determinados por la frecuencia y amplitud para cada dirección de la imagen.

#### 2.2.4 Segmentación

En esta etapa, la imagen se fracciona con la finalidad de clasificar las partes que componen. Dichas partes se denominan zonas o regiones de la imagen, las cuales están compuestas por diversas características. Conforme a estas características se seleccionan las zonas de interés para las técnicas a aplicar.

Los atributos básicos de segmentación de una imagen son: los componentes de color en imágenes, la luminancia en imágenes monocromáticas, textura, forma, etc.

La segmentación automática es una de las fases más complicadas del procesamiento de imágenes, en esta se determina si el análisis de la imagen es exitoso o no, y muy pocas veces se logra una solución confortable.

Dentro de la etapa de segmentación encuentran muchas técnicas, las cuales se puede agrupar dentro de estos tres grupos:

- Detección de bordes
- Umbralización
- Basado en regiones

#### 2.2.5 Extracción de características

La técnica consiste en extraer las características de la imagen, como los colores, el tamaño, área, forma, perímetros, así como también conocer las características topográficas como la representación del tamaño de la superficie y objetos geográficas.

Una peculiaridad es que no existe un método o algoritmo general para esta técnica, sino que con la aplicación de varios métodos se logra la extracción de características de la imagen.

Local Strokedirection features (LSD) es uno de los algoritmos utilizados en esta técnica que hace parte del procesamiento digital de imágenes. Consiste en procesar cada píxel (uno por uno); los pixeles repetidos en cada una de las cuatro direcciones se agrupan para después normalizarlos.

La normalización consiste en acomodar la forma, el tamaño y la posición de los píxeles con el objetivo de minimizar la variación entre los grupos de la misma clase.

Dado que una imagen contiene gran cantidad de datos, es necesario definir las características más importantes para extraerlas y obtener la mayor cantidad de información de esta.

Por lo tanto, se puede tomar en cuenta:

- El tiempo y costo de extracción no debe ser desmesurado.
- La ubicación debe ser fija o muy precisa y el error cometido bajo.
- Deben ser características estables.
- Deben contener gran cantidad de datos de los cuales sea posible obtener información de la forma de la imagen.
- La geometría de la imagen que define la forma de esta es una de las características más importantes.
- La geometría de la imagen.

#### 2.2.6 Tecnología Bot

Los Bots son asistentes virtuales destinados para hacer a las personas la vida un poco más fácil. Son inteligencias virtuales diseñadas para poder interactuar con los usuarios de forma similar a como lo haría un ser humano y realizar tareas de forma autónoma. La clave de su comportamiento y reacción reside en su capacidad para aprender las respuestas y comportamientos de las personas (Raya, 2016).

Los Bots que se encuentran en un entorno controlado como una conversación en chat, pueden resultar útiles para guiar, entretener, asesorar o ayudar a cada usuario que solicite usarlos.

Esta tecnología ya se encuentra implementada en aplicaciones independientes y diseñadas para dichos Bots, en aplicaciones conocidas como las redes sociales de Facebook, Skype, WhatsApp, Messenger, Telegram, entre otros.

### 2.2.7 Sistemas empotrados con Raspberry

Una Raspberry es una placa base de 85 x 54 milímetros en el que se aloja un chip Broadcom BCM2835 con procesador ARM hasta a 1 GHz de velocidad, GPU Video Core IV y hasta 1 GB de memoria RAM. La fundación de Raspberry Pi pone a disposición desde su página web Raspbian, una distribución de Linux basada en Debían, pero también podemos recurrir a muchas de las distribuciones específicas que la comunidad de usuarios ha desarrollado para diversos fines (AlbertoCastro, 2014). Es una placa que soporta varios componentes necesarios en un ordenador común, que puede ser utilizado por muchas de las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos (Raspberrypi, 2018).

### 2.2.8 Aplicativos en Telegram con tecnología Bot

Telegram Messenger, es una aplicación de mensajería instantánea gratuita, que fue desarrollada en el 2013. Sin embargo desde el 2014 fue cuando comenzó a tener acogida (Jimenez, 2018), llegando a ser una de las aplicaciones con mayor número de descargas. La aplicación se puede utilizar en celulares, tablets y ordenadores. Se encuentra disponible para los siguientes sistemas operativos: Android, IOS, macOS, Windows, GNU/Linux.

Telegram ha logrado popularidad debido a que cada vez se diferencia más de su principal competidor, WhatsApp. Una de las diferencias principales es que Telegram utiliza la arquitectura propia MTProto, con almacenamiento en la nube para los mensajes, Bot, canales y supergrupos. Además, el usuario es el que decide si quiere descargar los archivos en el dispositivo para ahorrar memoria o almacenarlos en la nube.

Otra característica es que, al ser de código abierto, los desarrolladores pueden crear sus propias aplicaciones dentro de Telegram. Y se puede llegar a tener un supergrupo con más de 10000 integrantes.

Uno de los puntos fuertes de Telegram, que lo diferencia de otros es el poder desarrollar un Bot: programa con el que se puede interactuar dentro de la aplicación Telegram como si fuese una persona (Aguirre Juárez, Reguilón Rodríguez, & Román Urdiales, 2017).

## 2.2.9 Métodos para el Procesamiento digital de Imágenes

### 2.2.9.1 Error cuadrático medio

El error cuadrático medio (ECM) mide el promedio de los errores al cuadrado entre el estimador y lo que se estima. ECM mide la distancia una línea de regresión de un conjunto de puntos. Lo hace tomando las distancias desde los puntos a la línea de regresión (las distancias son los “errores”) y cuadrándolos. Para eliminar cualquier signo negativo es necesario la cuadratura. Por lo tanto, se llama error cuadrático medio cuando se encuentra el promedio de un conjunto de errores. La ecuación es la siguiente (6):

$$ECM = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (6)$$

donde:

- $n$  es el número de puntos de datos
- $Y_i$  representa los valores observados
- $\hat{Y}_i$  representa los valores predichos

El ECM cuanto más cercano a 0, indica una mayor similitud.

### 2.2.9.2 Medida de similitud estructural

La similitud estructural índice (SSIM) es un método para medir la calidad de imagen. Se diseñó para mejorar los métodos tradicionales como pico de relación señal a ruido (PSNR) y error cuadrático medio (ECM)

Por un lado, SSIM intenta modelar el cambio percibido en la información estructural de la imagen, mientras que ECM está estimando los errores percibidos. Por otro lado, SSIM se utiliza para comparar dos ventanas de la imagen en lugar de la imagen completa como ECM.

La fórmula SSIM se base en tres mediciones de comparación entre las muestras de  $y$ : luminancia, contraste, y estructura. La ecuación es la siguiente (7):

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (7)$$

Donde:

- $\mu_x$  es la media de  $x$
- $\mu_y$  es la media de  $y$
- $\sigma_x^2$  es la varianza de  $x$
- $\sigma_y^2$  es la varianza de  $y$
- $\sigma_{xy}$  es la covarianza de  $xy$

- $c_1 = (k_1 L)^2$ , dos variables para estabilizar la división con un denominador débil:  $c_2 = (k_2 L)^2$
- $L$  es el rango dinámico de los valores de píxel:  $2^{\#bits\ por\ pixel} - 1$
- $k_1 = 0.01$  y por defecto  $k_2 = 0.03$

El SSIM cuanto más cercano a 1, indica una mayor similitud.

### 2.2.10 OpenCV

OpenCV es una librería de software de aprendizaje por computadora y visión por computadora de código abierto. Fue creado para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de máquinas en los productos comerciales. Con más de 2500 algoritmos optimizados en la librería, que se pueden utilizar para detección y reconocimiento facial, identificación de objetos, clasificaciones de acciones humanas en videos, detectar objetos en movimiento, segmentación, entre otros (OpenCV, 2019).

## CAPÍTULO III

### DESARROLLO E IMPLEMENTACIÓN

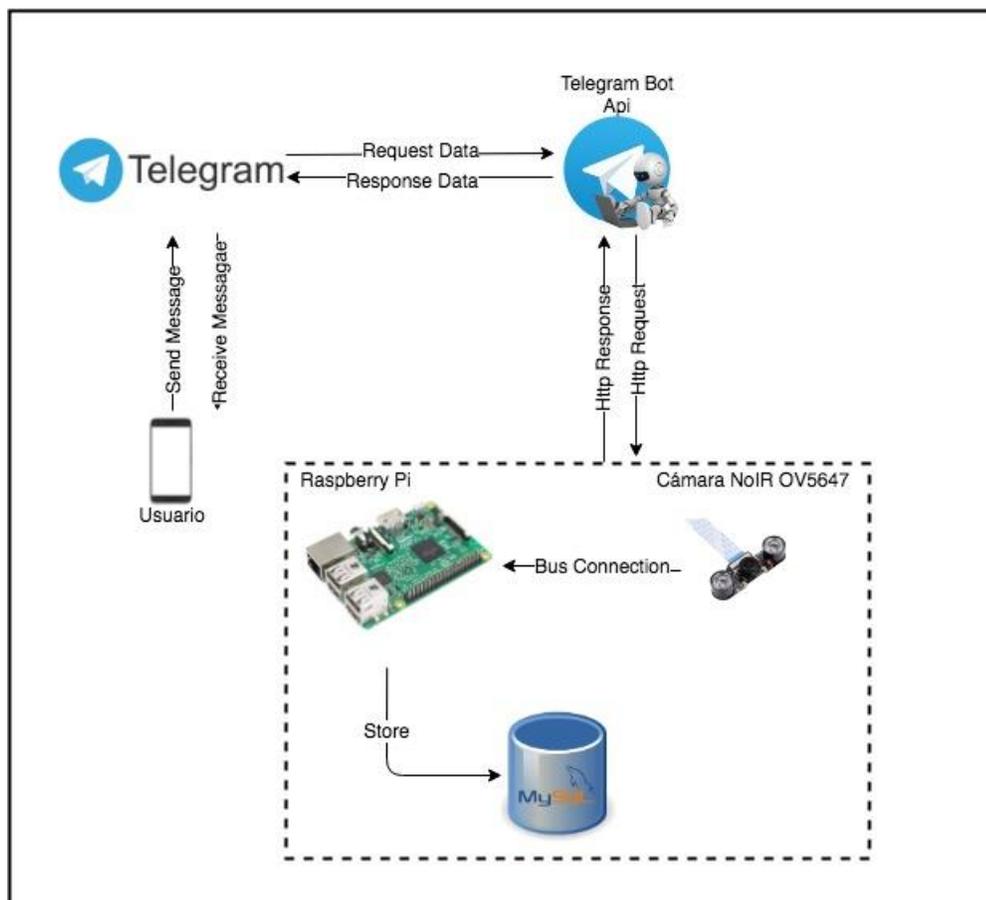
En este capítulo se describe el proceso llevado a cabo para el desarrollo del sistema con las siguientes actividades que son el diseño y la implementación de un sistema de detección de plazas de aparcamiento disponible. Donde primero se recibe la petición del usuario a través de un Bot previamente programado en Python y utilizando la API de Telegram, para posteriormente capturar la imagen del parqueadero y realizar el procesamiento digital de la misma, obteniendo las zonas disponibles del parqueadero. Finalmente, la imagen procesada en la cual se marcan las zonas libres del parqueadero se retorna al usuario por medio del Bot en Telegram.

#### **3.1 Diseño**

##### **3.1.1 Arquitectura**

La Figura 6 muestra la arquitectura general del diseño de la aplicación. El usuario realiza la petición a través del Bot en la cuenta de Telegram. Con el token del Bot la Raspberry puede acceder al API de Telegram (Telepot) desde el script en Python ejecutado en dicho dispositivo. Mediante las peticiones que realiza el usuario y las instrucciones programadas en el script para el Bot se procesa las peticiones y se retorna las respuestas correspondientes según la petición. La selección de respuesta se basa en reglas, es decir a través de condiciones simples implementadas en el código del script.

En la Raspberry Pi se configuró la base de datos relacional MySQL gestionada a través de la herramienta phpMyAdmin. La base de datos local almacena en una tabla los logs de la petición de disponibilidad de plazas.



**Figura 6.** Arquitectura del Sistema

### 3.1.2 Diseño del Bot en Telegram

#### 3.1.2.1 Creación del Bot

Para poder crear el Bot es necesario tener habilitada una cuenta en Telegram, buscar a @BotFather en los contactos de la cuenta como se muestra en la Figura 7. Mediante las siguientes instrucciones se procede a realizar la creación del Bot:

1. /start: este comando se ejecuta por defecto al ingresar por primera vez en el Bot, para iniciar una conversación.

2. /newbot: comando para crear un nuevo Bot. A continuación, se ingresa el nombre del Bot que se quiere crear, y el nombre de usuario, siguiendo los estándares de la creación del mismo: como el nombre no puede empezar con números, el nombre de usuario que no esté registrado y la terminación del nombre con ‘Bot’. En este caso el nombre del Bot es ‘CarParkEspeBot’ y el nombre de usuario: ‘CarParkEspe’.
3. Una vez creado el Bot, retorna un Token para acceder al Http API. Dicho Token permite autenticarse con el API de Telegram e interactuar con el Bot programado desde Python.

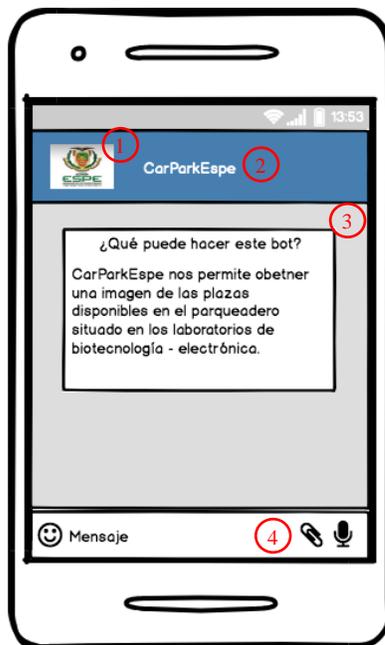


**Figura 7.** BotFather

### 3.1.2.2 Maquetación del Bot

El usuario puede realizar las peticiones a través del Bot “CarParkEspe” desde su dispositivo móvil en la plataforma Telegram. Las peticiones son recibidas y enviadas a través del API de Telegram para ser procesadas por la Raspberry Pi a través del script desarrollado en Python.

En la Figura 8, se observa el mensaje de la descripción de lo que hace el bot. Con un límite de 512 caracteres, será el texto que visualicen los usuarios cuando inicien la conversación con el Bot, bajo el título de “¿Qué puede hacer este bot?”



*Figura 8.* Inicio

En la Figura 8, se puede observar 4 secciones principales. La sección 1 representa la foto de perfil, mientras que la sección 2 indica el nombre del Bot. La sección 3 es el menú de comandos donde se visualiza las tanto las peticiones realizadas por el usuario como las respuestas del Bot. Y finalmente, la sección 4 donde el usuario ingresa los mensajes.

El siguiente comando “/plaza”, retorna la disponibilidad de plazas en la zona donde se encuentra el sistema que captura la imagen del parqueadero a través de una foto con la cámara NoIr. La cual será procesada en la Raspberry Pi mediante el procesamiento de imágenes. El diseño del reconocimiento de imágenes se base principalmente en la comparación de la información de dos imágenes que mediante un score resultante se discierne si estas son iguales o no. Es decir, la imagen original es la de la plaza disponible, mientras que la segunda imagen a comparar es la plaza que captura la cámara pi en el momento de la petición de plazas. De tal manera, que si el score de la comparación es relativamente alto (cercano a 1), la plaza será tomada como disponible y se

enmarcará la plaza con un rectángulo de color verde para facilitar al usuario su detección, además mediante un mensaje al Bot se le indicará el número de plazas disponibles en dicho sector. Se puede observar el resultado en la Figura 9.



**Figura 9.** Petición para la disponibilidad de plazas

En la Figura 10, se puede observar el comando “/opciones” que retorna que comandos están disponibles para el Bot. En este caso existe el comando “/plazas” que indica la disponibilidad de plazas como se mencionó anteriormente y el comando de “/descripción” que nos informa de la locación del parqueadero donde se encuentra el sistema (Figura 11).



**Figura 10.** Comando de ayuda

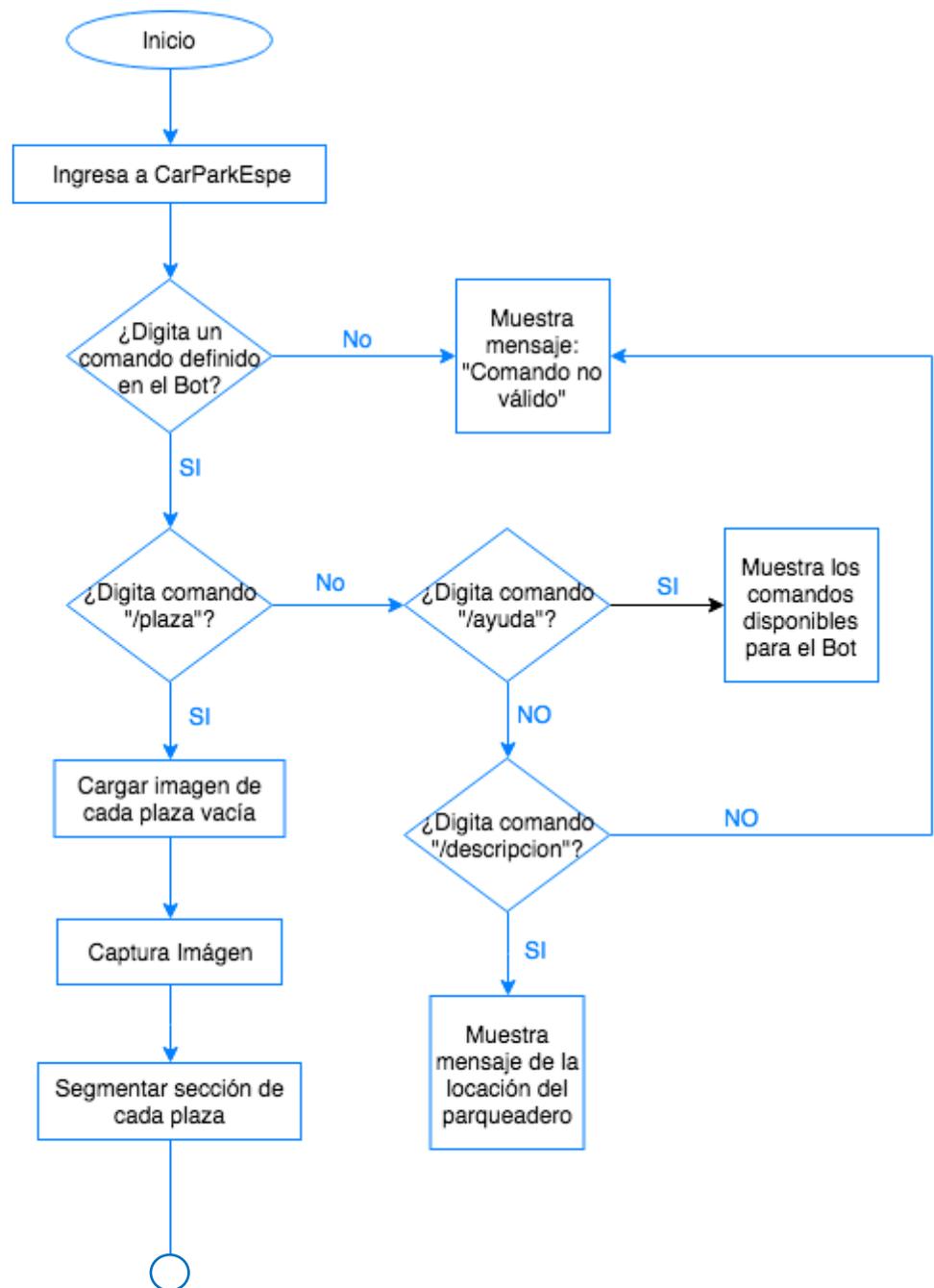


**Figura 11.** Comando de locación del parqueadero

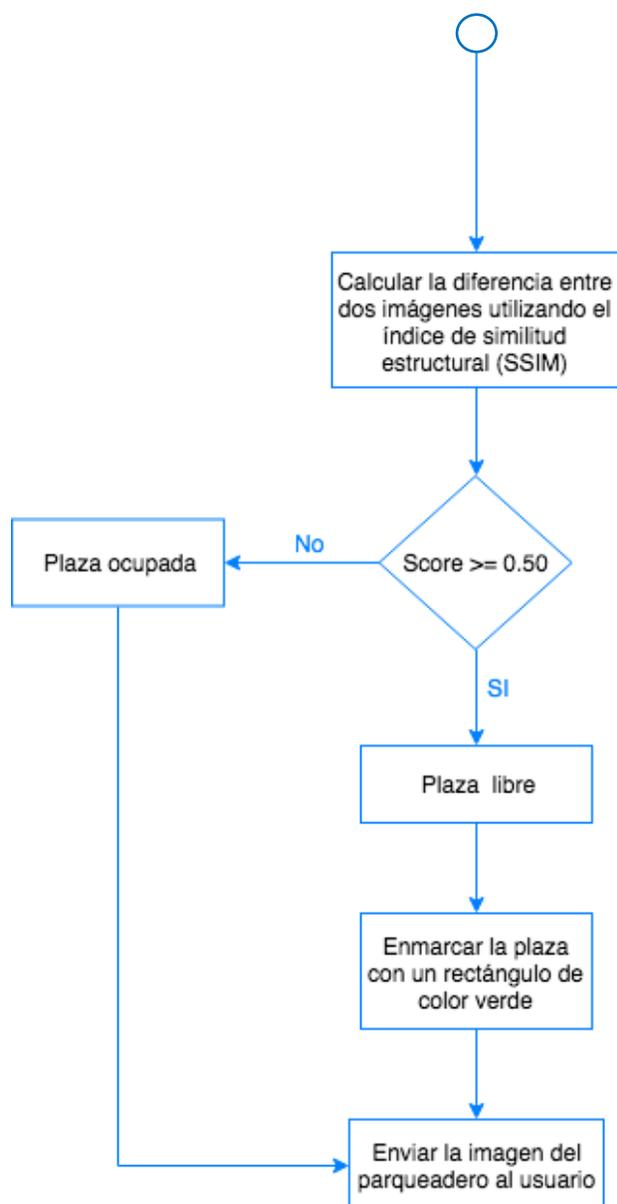
### 3.1.3 Interacción

La Figura 12 muestra el diagrama de flujo del sistema, al ingresar al Bot “CarParkEspe”. Si el usuario digita un comando correcto, en este caso “/plaza”, se carga automáticamente las imágenes almacenadas de cada plaza vacía guardadas en la memoria del dispositivo. Una vez cargadas las imágenes se captura una imagen del parqueadero para posteriormente recortas y definir las secciones de los estacionamientos, con esas secciones y las imágenes previamente cargadas al inicio se calcula el índice de similitud estructural (SSIM, de sus siglas en inglés “Structural Similarity Index”) entre las dos imágenes en escala de grises por plaza. El score resultante representa el SSIM entre las dos imágenes de entrada (plaza vacía y plaza en el momento de la captura de la imagen). Este valor puede resultar en el rango  $[-1, 1]$ , donde el valor de 1 es una similitud perfecta. En el caso de que el score sea mayor a 0.5 se determina que la plaza está libre,

por lo que se enmarcará la plaza con un rectángulo de color verde en la imagen capturada para retornarla al usuario.



CONTINÚA 



**Figura 12.** Diagrama de flujo del sistema de parqueadero

## 3.2 Implementación

### 3.2.1 Instalación OpenCV

Se implementó OpenCV, por un lado, porque es una librería libre de visión artificial, está disponible para múltiples plataformas, en este caso Linux, y porque cuenta con soporte para Python.

Se instaló OpenCV Python versión 3.4.1 en Raspberry Pi 3 con el sistema operativo Raspbian.

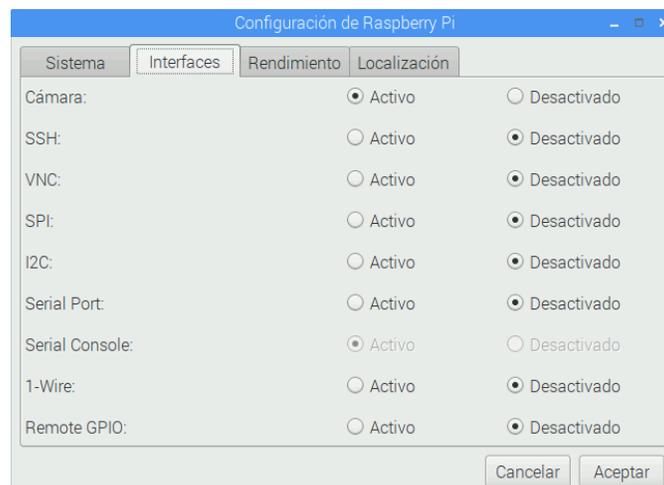
### 3.2.2 Captura de imagen

Para poder realizar la captura de la imagen se necesita habilitar la cámara, donde la cámara NoIR se conecta al slot correspondiente de la RPI3 apagada (Figura 13).



**Figura 13.** Conexión de RPI3 y cámara NoIR

Una vez conectada la cámara, se procede a encender la RPI3 y desde la configuración de la Raspberry Pi, en la pestaña Interfaces se habilita la opción de Cámara, y se reinicia el sistema (Figura 14).



**Figura 14.** Configuración habilitar cámara

A continuación, el código para la captura de una foto. Donde pasado 1 segundo, toma la foto y se almacena en el directorio “/home/pi/Desktop/CarParkEspe” (Figura 15).

```
import time,datetime
from picamera import PiCamera

def capturarPlaza():
    resultado = ""

    try:
        camara = PiCamera()
        time.sleep(1)
        camara.capture('estacionamiento.jpg')
        img_rgb =
cv2.imread('/home/pi/Desktop/CarParkEspe/estacionamiento.jpg')
        cv2.imwrite("/home/pi/Desktop/CarParkEspe/estacionamiento.jpg",
img_rgb)

        resultado = "/home/pi/Desktop/CarParkEspe/estacionamiento.jpg"

    except Exception as e:
        print(e)

    finally:
        camara.close()

return resultado
```

**Figura 15.** Código para captura de imagen

### 3.2.3 Reconocimiento imagen

En primera instancia se carga la imagen con el estacionamiento vacío y la imagen del estacionamiento en el momento de la petición (Figura 16).

```
def identificarPlazas():
    start_time = time_run()
    plazas = {}

    plazas[0] = cv2.imread('/home/pi/Desktop/CarParkEspe/Plantillas/plaza_1.jpg')
    parqueadero = cv2.imread('/home/pi/Desktop/CarParkEspe/estacionamiento.jpg')
```

**Figura 16.** Código para carga imágenes

Se segmenta cada una de las plazas a evaluar y se establece un tamaño de 100 x 100. A continuación, se convierte la imagen original del espacio de color BGR a gris (Figura 17).

```

def identificarPlazas():

    #Segmentos Plazas a evaluar
    y = [600,1125]
    segmento_plazas = {}
    segmento_plazas[0] = parqueadero[y[0]:y[1], y[0]:1155]
    segmento_plazas[1] = parqueadero[y[0]:y[1], 1155:1710]
    segmento_plazas[2] = parqueadero[y[0]:y[1], 1610:2165]

    coordenadas_plazas = {}
    coordenadas_plazas[0] = [[600,600],[1155,1125]]
    coordenadas_plazas[1] = [[1155,600],[1710,1125]]
    coordenadas_plazas[2] = [[1610,600],[2165,1125]]

    #Establecemos un tamaño de imagen para comparar (100x100)
    segmento_plazas[0] = cv2.resize(segmento_plazas[0],(0,0), fx=0.18,fy=0.19)
    segmento_plazas[1] = cv2.resize(segmento_plazas[1],(0,0), fx=0.18,fy=0.19)
    segmento_plazas[2] = cv2.resize(segmento_plazas[2],(0,0), fx=0.18,fy=0.19)

```

**Figura 17.** Código para segmentación y redimensión imágenes

En la Figura 18 se calcula el índice de similitud estructural (SSIM) entre las dos imágenes en escala de grises. Con la función `compare_ssim` de `scikit-image`, se calcula una imagen de puntaje y diferencia (`score`, `diff`). Para que finalmente, las comparaciones con un `score` mayor o igual a 0.50 sean marcadas como disponibles mediante la interpretación de un rectángulo verde sobre la plaza.

```

def identificarPlazas():
    cont = 0
    plazas_libres = 0

    for i in segmento_plazas :
        cont = cont + 1
        (score,diff) = compare_ssim(grayPlaza,grayParqueadero[i], full=True)
        if score > 0.5:
            plazas_libres = plazas_libres + 1

    cv2.rectangle(parqueadero, (coordenadas_plazas[i][0][0]+80, coordenadas_plazas[i][0][1]), (coordenadas_plazas[i][1][0]-80, coordenadas_plazas[i][1][1]), (36,255,12), 3)

    parqueadero = cv2.resize(parqueadero,(0,0), fx=0.18,fy=0.19)
    cv2.imwrite("/home/pi/Desktop/CarParkEspe/plazas_libres.jpg", parqueadero)

    resultado = "/home/pi/Desktop/CarParkEspe/plazas_libres.jpg"

    return resultado

```

**Figura 18.** Código para algoritmo SSIM y deserción de plazas

### 3.2.4 API

Para poder comunicarse la Raspberry Pi con Telegram Bot API es necesario instalar el paquete Telepot (Figura 19).

```
sudo apt-get install python-pip
sudo pip install telepot
```

*Figura 19.* Comandos para Telepot

### 3.2.5 Peticiones y respuestas

#### Petición “/plaza”

Cuando el usuario solicita mediante el comando “/plaza” (Figura 20) la disponibilidad de zonas libres de estacionamiento el Bot retorna la imagen capturada en ese momento con las zonas libres enmarcadas en un rectángulo de color verde (Figura 21).

```
elif comando == '/plaza':
    telegram_bot.sendMessage (chat_id, str("Su petición esta siendo
procesada..."))
    start_time = time_run()
    valor = capturarPlaza()
    print (valor)
    elapsed_time = time_run() - start_time
    print('Tiempo: ' + str(elapsed_time))
    telegram_bot.sendPhoto (chat_id, photo = open(str(valor), 'rb'))
```

*Figura 20.* Código para petición “/plaza”



*Figura 21.* Respuesta “/plaza”

## Petición “/ayuda”

Cuando el usuario solicite mediante el comando “/ayuda” las opciones disponibles que estén condicionadas en el bot, retornará la información (Figura 22).

```
elif comando == '/ayuda':
    telegram_bot.sendMessage (chat_id, str("Por favor seleccione una de
    las siguientes opciones: \n\n /descripcion - indica el lugar donde se
    encuentra el sistema de detección de plazas. \n /plaza - realiza la petición
    para obtener la imagen con las plazas disponibles."))
```

**Figura 22.** Código para petición “/ayuda” y respuesta

## Petición “/descripcion”

Cuando el usuario solicite mediante el comando “/descripcion” el detalle de la locación donde se encuentre ubicado el parqueadero con el sistema de detección de plazas. (Figura 23).

```
elif comando == '/descripcion':
    telegram_bot.sendMessage (chat_id, str("El siguiente Bot nos permite
    obtener una imagen de las plazas disponibles en el parqueadero situado en los
    laboratorios de biotecnología - electrónica a través de la petición /plaza"))
```

**Figura 23.** Código para petición “/descripcion” y respuesta

## Peticiones desconocidas

Cuando el usuario realice una petición desconocida que no estén dentro de las reglas del bot, este retorna un mensaje de información (Figura 24).

```
else:
    telegram_bot.sendMessage (chat_id, str("Comando no válido"))
```

**Figura 24.** Código para petición desconocida y respuesta

### 3.2.6 Base de Datos e Imágenes

#### 3.2.6.1 Modelo Base de Datos

En el presente trabajo se empleó una base de datos relacional gestionada en MySQL, con la tabla “Logs”, la cual registra las siguientes variables: a) id como representador único de cada

registro, b) `id_usuario` donde se almacena el nombre de usuario de Telegram, c) `primer_nombre` y d) `primer_apellido` (opcional) del usuario con el que se registra la cuenta en la plataforma Telegram, e) `comando` donde se almacena la petición del usuario, f) `respuesta` que guarda la dirección y nombre de la imagen que se retorna al usuario, g) `score` que representa los valores de similitud entre la imagen previamente cargada y la imagen capturada en el momento, h) `tiempo_respuesta` que representa el tiempo de respuesta desde que el usuario realiza la petición hasta que recibe la respuesta. Y finalmente, i) `created_at` que guarda la fecha y hora de la inserción del dato en la base (Ver Figura 25).

LOGS			
<u>id</u>	<pi>	Integer	<M>
id_usuario		Variable characters (25)	
primer_nombre		Variable characters (50)	
primer_apellido		Variable characters (50)	
comando		Variable characters (25)	
respuesta		Text	
score		Text	
tiempo_respuesta		Text	
created_at		Timestamp	
Identifier_1	<pi>		

**Figura 25.** Modelo Conceptual de la BD

### 3.2.6.2 Adquisición de Imágenes

Se implementó un repositorio local donde se almacena la imagen original de la plaza libre (Figura 26), la cual servirá de base para poder realizar la comparación con todas las plazas de aparcamiento cuando la petición sea solicitada.

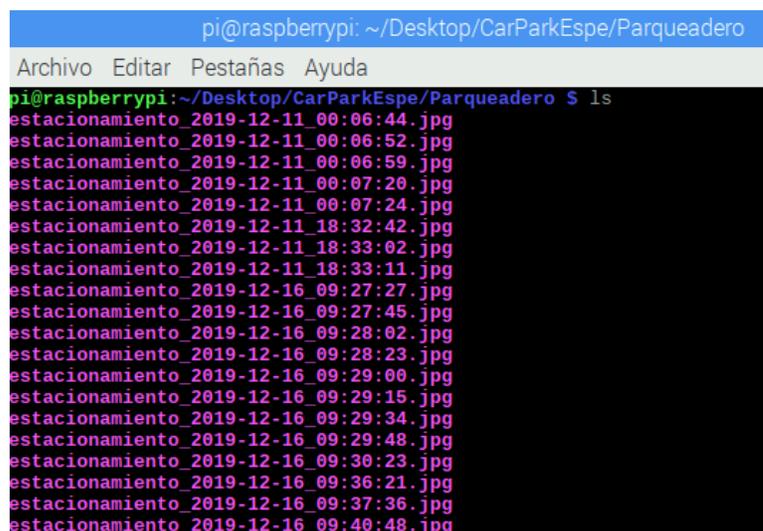
```

pi@raspberrypi: ~/Desktop/CarParkEspe/Plantillas
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~/Desktop/CarParkEspe/Plantillas $ ls
plantilla.png

```

**Figura 26.** Directorio de la Plantilla

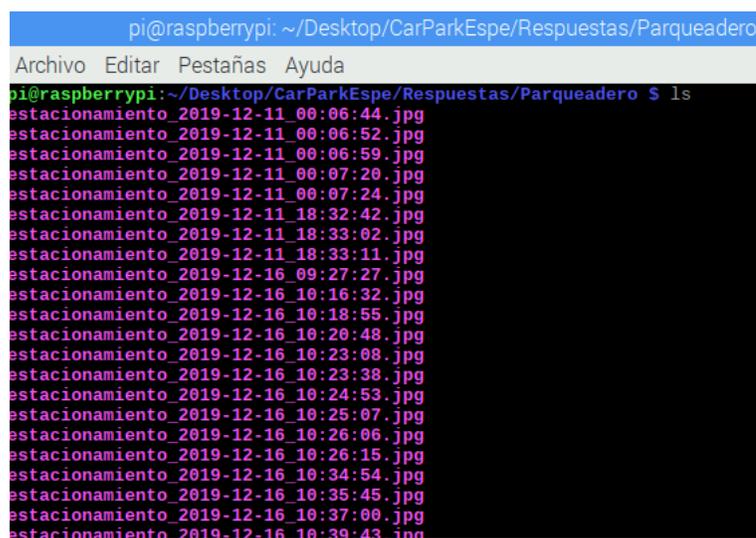
En otro directorio, se encuentran las imágenes que captura la cámara cuando se realiza la petición (Figura 27), y las que se empelarán para poder realizar la comparación por segmentación de plazas vs la imagen original de plaza libre.



```
pi@raspberrypi: ~/Desktop/CarParkEspe/Parqueadero
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~/Desktop/CarParkEspe/Parqueadero $ ls
estacionamiento_2019-12-11_00:06:44.jpg
estacionamiento_2019-12-11_00:06:52.jpg
estacionamiento_2019-12-11_00:06:59.jpg
estacionamiento_2019-12-11_00:07:20.jpg
estacionamiento_2019-12-11_00:07:24.jpg
estacionamiento_2019-12-11_18:32:42.jpg
estacionamiento_2019-12-11_18:33:02.jpg
estacionamiento_2019-12-11_18:33:11.jpg
estacionamiento_2019-12-16_09:27:27.jpg
estacionamiento_2019-12-16_09:27:45.jpg
estacionamiento_2019-12-16_09:28:02.jpg
estacionamiento_2019-12-16_09:28:23.jpg
estacionamiento_2019-12-16_09:29:00.jpg
estacionamiento_2019-12-16_09:29:15.jpg
estacionamiento_2019-12-16_09:29:34.jpg
estacionamiento_2019-12-16_09:29:48.jpg
estacionamiento_2019-12-16_09:30:23.jpg
estacionamiento_2019-12-16_09:36:21.jpg
estacionamiento_2019-12-16_09:37:36.jpg
estacionamiento_2019-12-16_09:40:48.jpg
```

*Figura 27.* Directorio de imágenes capturadas con la cámara

Y en otro directorio, se encuentran las imágenes procesadas con el resultado visual de que plazas se encuentran libres, Figura 28.



```
pi@raspberrypi: ~/Desktop/CarParkEspe/Respuestas/Parqueadero
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~/Desktop/CarParkEspe/Respuestas/Parqueadero $ ls
estacionamiento_2019-12-11_00:06:44.jpg
estacionamiento_2019-12-11_00:06:52.jpg
estacionamiento_2019-12-11_00:06:59.jpg
estacionamiento_2019-12-11_00:07:20.jpg
estacionamiento_2019-12-11_00:07:24.jpg
estacionamiento_2019-12-11_18:32:42.jpg
estacionamiento_2019-12-11_18:33:02.jpg
estacionamiento_2019-12-11_18:33:11.jpg
estacionamiento_2019-12-16_09:27:27.jpg
estacionamiento_2019-12-16_10:16:32.jpg
estacionamiento_2019-12-16_10:18:55.jpg
estacionamiento_2019-12-16_10:20:48.jpg
estacionamiento_2019-12-16_10:23:08.jpg
estacionamiento_2019-12-16_10:23:38.jpg
estacionamiento_2019-12-16_10:24:53.jpg
estacionamiento_2019-12-16_10:25:07.jpg
estacionamiento_2019-12-16_10:26:06.jpg
estacionamiento_2019-12-16_10:26:15.jpg
estacionamiento_2019-12-16_10:34:54.jpg
estacionamiento_2019-12-16_10:35:45.jpg
estacionamiento_2019-12-16_10:37:00.jpg
estacionamiento_2019-12-16_10:39:43.jpg
```

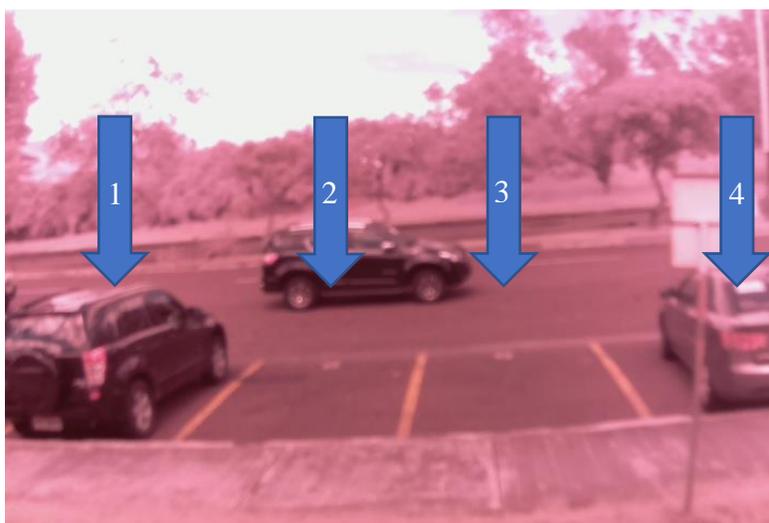
*Figura 28.* Directorio de imágenes procesadas

### 3.2.7 Pruebas Individuales

#### 3.2.7.1 Definición experimento

Como punto de partida para el desarrollo, se realizó la arquitectura general del diseño de la aplicación con la que se trabajaría, para continuar con la creación del Bot en Telegram a través de su maquetación e interacción mediante el diagrama de flujo del sistema. Posteriormente, para la implementación del sistema se seleccionó como herramienta OpenCV para el procesamiento de imágenes y compatibilidad con el API de Telegram. Además, para poder registrar la información para analizar se empleó la base de datos relacional MySQL y el repositorio local para almacenar las imágenes de las plazas para su posterior procesamiento, la plantilla para poder realizar las comparaciones, y las imágenes procesadas. Posteriormente, se procede a realizar las pruebas individuales para validar el funcionamiento del sistema.

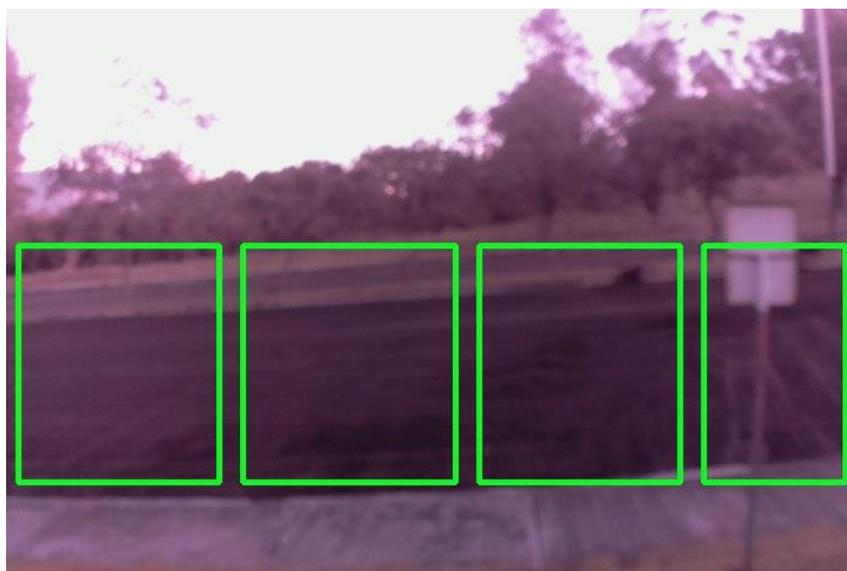
En la siguiente sección se realizó las pruebas en dos tiempos, considerando las horas de más afluencia vehicular. En la Figura 29, se puede visualizar las cuatro plazas de aparcamiento que se consideraron para la detección de plazas libres del parqueadero situado en los laboratorios de biotecnología-electrónica de la Universidad de las Fuerzas Armadas ESPE.



**Figura 29.** Plazas del Parqueadero

### 3.2.7.2 Prueba desde las 7 a.m.

A continuación, se puede visualizar la secuencia de imágenes de la Figura 30 (a-e) desde las 7 a.m. hasta las 13:59, donde muestra los diferentes casos de cada una de las plazas del parqueadero delimitado y sus respectivas tablas.



*Figura 30.* Respuesta de plazas



*Figura 31(b).* Respuesta de plazas

**Tabla 4**  
*Descripción del parqueadero de la Figura 30*

N ° de Plaza	Plaza Libre
<b>P1</b>	X
<b>P2</b>	X
<b>P3</b>	X
<b>P4</b>	X
<b>Plazas libres</b>	4

**Tabla 5**  
*Descripción del parqueadero de la Figura 30(b)*

N ° de Plaza	Plaza Libre
<b>P1</b>	X
<b>P2</b>	
<b>P3</b>	
<b>P4</b>	
<b>Plazas libres</b>	1



*Figura 30(c). Respuesta de plazas*



*Figura 30(d). Respuesta de plazas*

**Tabla 6**

*Descripción del parqueadero de la Figura 30(c)*

N ° de Plaza	Plaza Libre
<b>P1</b>	
<b>P2</b>	X
<b>P3</b>	X
<b>P4</b>	
<b>Plazas libres</b>	2

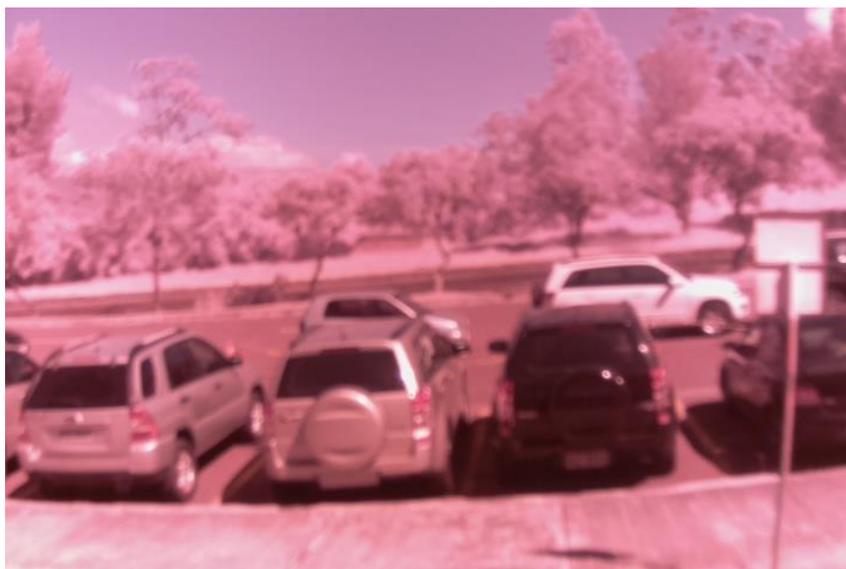
**Tabla 7**

*Descripción del parqueadero de la Figura 30(d)*

N ° de Plaza	Plaza Libre
<b>P1</b>	
<b>P2</b>	
<b>P3</b>	X
<b>P4</b>	
<b>Plazas libres</b>	1



*Figura 30(e).* Respuesta de plazas



*Figura 30(f).* Respuesta de plazas

**Tabla 8**  
*Descripción del parqueadero de la Figura 30(e)*

N ° de Plaza	Plaza Libre
<b>P1</b>	X
<b>P2</b>	
<b>P3</b>	
<b>P4</b>	X
<b>Plazas libres</b>	2

**Tabla 9**  
*Descripción del parqueadero de la Figura 30(f)*

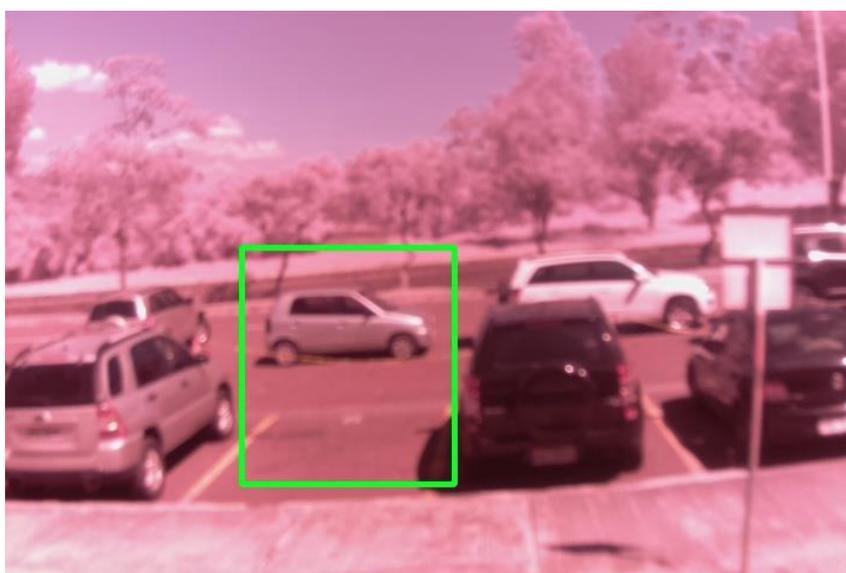
N ° de Plaza	Plaza Libre
<b>P1</b>	
<b>P2</b>	
<b>P3</b>	
<b>P4</b>	
<b>Plazas libres</b>	0

### 3.2.7.3 Prueba desde las 2 p.m.

En las siguientes pruebas, se puede visualizar la secuencia de imágenes de la Figura 31 (a-d) desde las 13:59 a.m. hasta las 17: 00, donde muestra los diferentes casos de cada una de las plazas del parqueadero delimitado y sus respectivas tablas.



*Figura 31. Respuesta de plazas*



*Figura 31(b). Respuesta de plazas*

**Tabla 10***Descripción del parqueadero de la Figura 31*

N ° de Plaza	Plaza Libre
<b>P1</b>	X
<b>P2</b>	
<b>P3</b>	
<b>P4</b>	
<b>Plazas libres</b>	1

**Tabla 11***Descripción del parqueadero de la Figura 31(b)*

N ° de Plaza	Plaza Libre
<b>P1</b>	
<b>P2</b>	X
<b>P3</b>	
<b>P4</b>	
<b>Plazas libres</b>	1

*Figura 31(c). Respuesta de plazas*



*Figura 31(d). Respuesta de plazas*

**Tabla 12**

*Descripción del parqueadero de la Figura 31(c)*

N ° de Plaza	Plaza Libre
<b>P1</b>	
<b>P2</b>	X
<b>P3</b>	X
<b>P4</b>	
<b>Plazas libres</b>	2

**Tabla 13**

*Descripción del parqueadero de la Figura 31(d)*

N ° de Plaza	Plaza Libre
<b>P1</b>	
<b>P2</b>	
<b>P3</b>	X
<b>P4</b>	
<b>Plazas libres</b>	1



*Figura 31(e). Respuesta de plazas*

**Tabla 14**

*Descripción del parqueadero de la Figura 31(e)*

N ° de Plaza	Plaza Libre
<b>P1</b>	
<b>P2</b>	X
<b>P3</b>	
<b>P4</b>	X
<b>Plazas libres</b>	2

## CAPÍTULO IV

### EVALUACIÓN

El sistema de detección de plazas de aparcamiento disponible, mediante procesamiento digital de imágenes y tecnología Bot, ha sido evaluado en la Universidad de las Fuerzas Armadas ESPE, campus Sangolquí (Ecuador).

En este capítulo se describe la evaluación del sistema de detección de plazas libres “CarParkEspe”, desarrollado con el propósito de diseñar un sistema de visualización de plazas disponibles de estacionamiento en la zona de parqueadero de los laboratorios de Biotecnología – Electrónica de la institución.

CarParkEspe busca minimizar el tiempo que el usuario con vehículo emplea para encontrar una plaza de aparcamiento por el método tradicional de búsqueda de plazas libres de forma tradicional, mediante la utilización del sistema.

#### 4.1 Instrumentos

En el desarrollo del presente trabajo de titulación, se utilizó dos herramientas principales para determinar la eficiencia del sistema de parqueadero: encuestas y tiempos.

Para la elaboración de la encuesta se utilizó la metodología cuantitativa para poder generalizar los datos que se observan del universo general de la población a partir de una muestra pequeña mediante preguntas genéricas.

Para el desarrollo de la encuesta se utilizó la herramienta de Formularios de Google y para determinar el tamaño de la muestra se utilizó la fórmula de tamaño muestral de una población conocida (8):

$$n = \frac{k^2 * p * q * N}{(e^2 * (N - 1)) + k^2 * p * q} \quad (8)$$

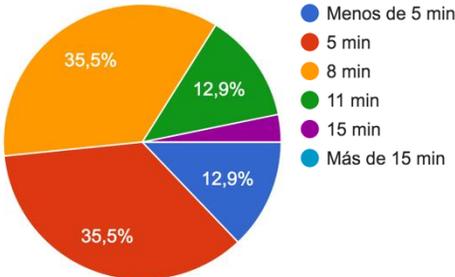
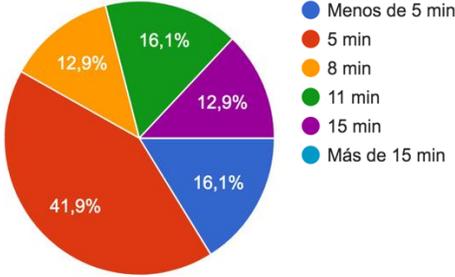
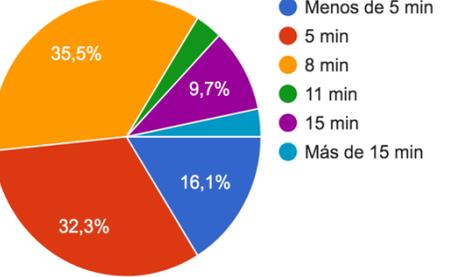
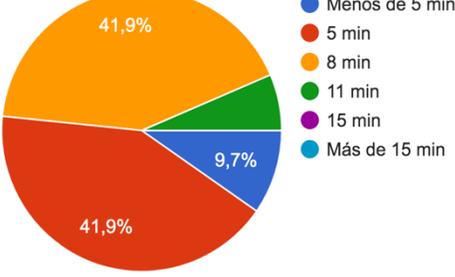
Donde:

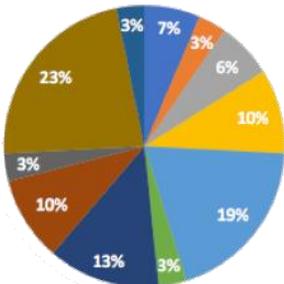
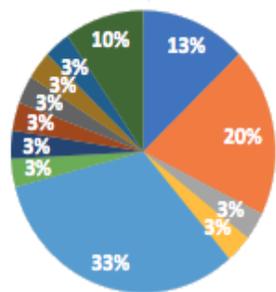
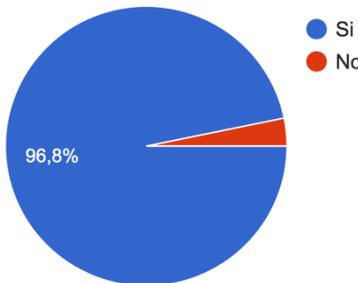
- $k$  es una constante que depende del nivel de confianza
- $N$  es el tamaño de la población
- $p$  es proporción esperada (se suele suponer que es 0.5)
- $q$  es igual a  $1 - p$
- $e$  es el error muestral deseado.
- $n$  es número de muestras que se van a realizar (en este caso el número de encuestas que se van a realizar)

Para realizar la encuesta a los estudiantes sobre la percepción del usuario sobre el uso del parqueadero de la institución (Anexo 1) sobre 275 estudiantes de la carrera de Software ( $N$ ), en la que se quiere un nivel de confianza del 90% que determina  $k=1.65$ , deseando un error muestral del 14% ( $e$ ) y considerando que estarán de acuerdo el 50% ( $p=q=0.5$ ), se necesita una muestra de 31 estudiantes.

Al realizar las encuestas se obtuvieron los siguientes resultados:

**Tabla 15**  
Encuesta Percepción sobre el uso del Parqueadero

Pregunta	Resultado	Observación
<p>1. ¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 7 am? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad.</p>	 <p>Menos de 5 min 5 min 8 min 11 min 15 min Más de 15 min</p>	<p>Al 70,10% de los usuarios le toma entre 5 - 8 minutos encontrar una plaza sobre las 7 am.</p>
<p>2. ¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 14:00? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad.</p>	 <p>Menos de 5 min 5 min 8 min 11 min 15 min Más de 15 min</p>	<p>Al 41,9% de los usuarios le toma 5 minutos encontrar una plaza sobre las 14:00.</p>
<p>3. ¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 15:00? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad.</p>	 <p>Menos de 5 min 5 min 8 min 11 min 15 min Más de 15 min</p>	<p>Al 35,5% de los usuarios le toma 8 minutos encontrar una plaza sobre las 15:00, seguido del 32,3% que le toma 5 minutos.</p>
<p>4. ¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 17:00? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad.</p>	 <p>Menos de 5 min 5 min 8 min 11 min 15 min Más de 15 min</p>	<p>Al 83,8% de los usuarios le toma entre 5 - 8 minutos encontrar una plaza sobre las 17:00.</p>

<p>5. ¿Qué tiempo estima que tarda el usuario en encontrar una plaza de aparcamiento libre desde que ingresa a la institución en horas pico?</p>	 <table border="1"> <thead> <tr> <th>Time Range</th> <th>Percentage</th> </tr> </thead> <tbody> <tr><td>5 min</td><td>3%</td></tr> <tr><td>6 min</td><td>7%</td></tr> <tr><td>7 min</td><td>3%</td></tr> <tr><td>8 min</td><td>6%</td></tr> <tr><td>10 min</td><td>10%</td></tr> <tr><td>15 min</td><td>19%</td></tr> <tr><td>10 a 15 min</td><td>3%</td></tr> <tr><td>Más de 15 min</td><td>3%</td></tr> <tr><td>20 min</td><td>23%</td></tr> <tr><td>30 min</td><td>3%</td></tr> </tbody> </table>	Time Range	Percentage	5 min	3%	6 min	7%	7 min	3%	8 min	6%	10 min	10%	15 min	19%	10 a 15 min	3%	Más de 15 min	3%	20 min	23%	30 min	3%	<p>El 23% de los usuarios estima que le tomaría 20 minutos encontrar una plaza en horas pico.</p>				
Time Range	Percentage																											
5 min	3%																											
6 min	7%																											
7 min	3%																											
8 min	6%																											
10 min	10%																											
15 min	19%																											
10 a 15 min	3%																											
Más de 15 min	3%																											
20 min	23%																											
30 min	3%																											
<p>6. ¿Qué tiempo estima que tarda el usuario en encontrar una plaza de aparcamiento libre desde que ingresa a la institución en horas de menor afluencia?</p>	 <table border="1"> <thead> <tr> <th>Time Range</th> <th>Percentage</th> </tr> </thead> <tbody> <tr><td>2 min</td><td>13%</td></tr> <tr><td>3 min</td><td>20%</td></tr> <tr><td>4 min</td><td>3%</td></tr> <tr><td>Menos 5 min</td><td>3%</td></tr> <tr><td>5 min</td><td>33%</td></tr> <tr><td>6 min</td><td>3%</td></tr> <tr><td>5 a 8 min</td><td>3%</td></tr> <tr><td>8 min</td><td>3%</td></tr> <tr><td>9 min</td><td>3%</td></tr> <tr><td>Entre 5 y 10 min</td><td>3%</td></tr> <tr><td>Menos 10 min</td><td>10%</td></tr> <tr><td>10 min</td><td>3%</td></tr> </tbody> </table>	Time Range	Percentage	2 min	13%	3 min	20%	4 min	3%	Menos 5 min	3%	5 min	33%	6 min	3%	5 a 8 min	3%	8 min	3%	9 min	3%	Entre 5 y 10 min	3%	Menos 10 min	10%	10 min	3%	<p>El 33% de los usuarios estima que le tomaría 5 minutos encontrar una plaza en horas de menor afluencia.</p>
Time Range	Percentage																											
2 min	13%																											
3 min	20%																											
4 min	3%																											
Menos 5 min	3%																											
5 min	33%																											
6 min	3%																											
5 a 8 min	3%																											
8 min	3%																											
9 min	3%																											
Entre 5 y 10 min	3%																											
Menos 10 min	10%																											
10 min	3%																											
<p>7. ¿Le gustaría tener un sistema de aparcamiento en la Universidad de las Fuerzas Armadas ESPE?</p>	 <table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr><td>Si</td><td>96,8%</td></tr> <tr><td>No</td><td>3,2%</td></tr> </tbody> </table>	Response	Percentage	Si	96,8%	No	3,2%	<p>Al 96,8% de los usuarios le gustaría tener un sistema de aparcamiento en la Universidad de las Fuerzas Armadas ESPE.</p>																				
Response	Percentage																											
Si	96,8%																											
No	3,2%																											

## 4.2 Pruebas del Modelo

Para realizar el proceso de prueba se propuso dos escenarios.

### 4.2.1 Escenario I

#### **Objetivo**

Determinar el tiempo promedio que le toma al usuario en encontrar una plaza de aparcamiento con el método tradicional (sin la implementación del Bot), en la zona de parqueadero de los laboratorios de Biotecnología – Electrónica de la Universidad de la Fuerzas Armadas ESPE, extensión Salgolquí.

#### **Tareas**

##### 1. Dirigirse al Hyper Market

Los usuarios se dirigen al Hyper Market situado entre la Universidad de las Fuerzas Armadas ESPE y el centro comercial San Luis. Con la finalidad, de tomar como punto de partida dicha localización.

##### 2. Dirigirse a los Laboratorio de Biotecnología – Electrónica

Desde el punto de partida, el usuario se dirige a la zona de parqueadero de los laboratorios de Biotecnología – Electrónica de la Universidad de la Fuerzas Armadas ESPE, con la finalidad de encontrar una plaza libre de estacionamiento.

##### 3. Tomar tiempos de búsqueda de plaza libre

El usuario toma el tiempo que le toma encontrar una plaza libre de aparcamiento desde que sale del Hyper Market hasta que llega al parqueadero de los laboratorios de Biotecnología – Electrónica y encuentra una plaza libre.

#### 4.2.2 Escenario II

##### **Objetivo**

Determinar el tiempo promedio que le toma al usuario en encontrar una plaza de aparcamiento con el sistema de detección de plazas libres mediante procesamiento digital de imágenes y tecnología BOT, en la zona de parqueadero de los laboratorios de Biotecnología – Electrónica de la Universidad de la Fuerzas Armadas ESPE, extensión Salgolqui.

##### **Tareas**

1. Dirigirse al Hyper Market

Los usuarios se dirigen al Hyper Market situado entre la Universidad de las Fuerzas Armadas ESPE y el centro comercial San Luis.

2. Iniciar el Bot con el comando “/plaza”

Cuando el usuario se encuentra en el punto de partida, ingresa a su aplicación de Telegram y buscan el contacto “CarParkBot”. Una vez iniciada la conversación con el Bot, el usuario digita el comando “/plaza”, por lo que la petición será procesada por la Raspberry Pi y retornará la respuesta al Bot. Donde el usuario recibirá un mensaje con el número de plazas disponibles, seguido de la imagen que muestra las plazas disponibles gráficamente enmarcadas en un rectángulo de color verde las plazas que estuviesen libres.

3. Dirigirse a los Laboratorio de Biotecnología – Electrónica

Desde el punto de partida, el usuario se dirige a la zona de parqueadero de los laboratorios de Biotecnología – Electrónica de la Universidad de la Fuerzas Armadas ESPE, con la información resultante de la petición realizada al Bot.

4. Tomar tiempos de búsqueda de plaza libre

El usuario toma el tiempo que le toma encontrar una plaza libre de aparcamiento desde que sale del Hyper Market hasta que llega al parqueadero de los laboratorios de Biotecnología – Electrónica y encuentra una plaza libre con la utilización del sistema de detección de plazas libres mediante procesamiento digital de imágenes y tecnología BOT.

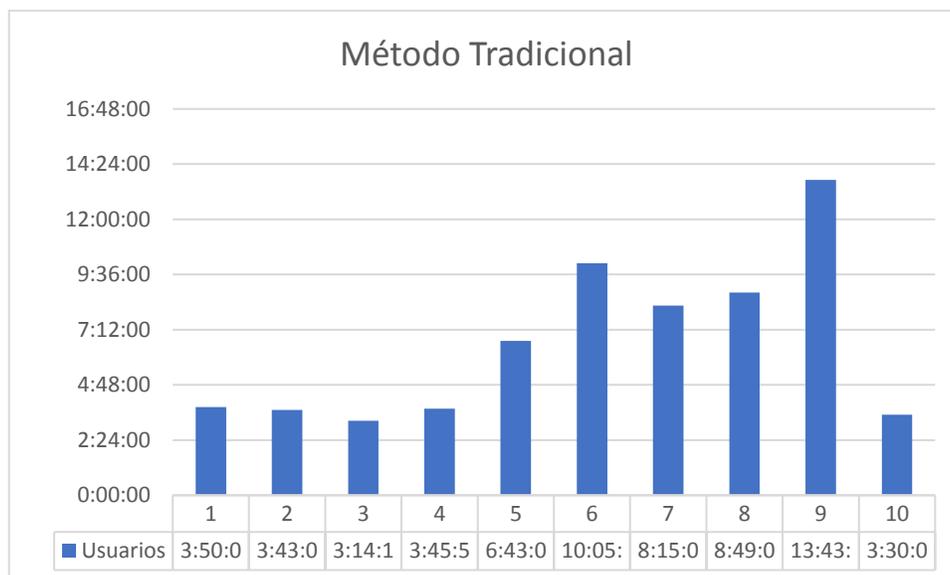
#### 4.3 RESULTADOS

##### **Escenario I**

##### **Tiempos**

En el método tradicional participaron 10 estudiantes de la Universidad de las Fuerzas Armadas ESPE con automóvil, para la realización de las pruebas de evaluación, en varias sesiones.

El promedio de los tiempos obtenidos para localizar la plaza de aparcamiento libre en el parqueadero situado en los laboratorios de Biotecnología – Electrónica fue de 6 minutos, 33 segundos, 48 milisegundos. En la Figura 32, se puede observar los datos obtenidos de los participantes en minutos, segundos y milisegundos.



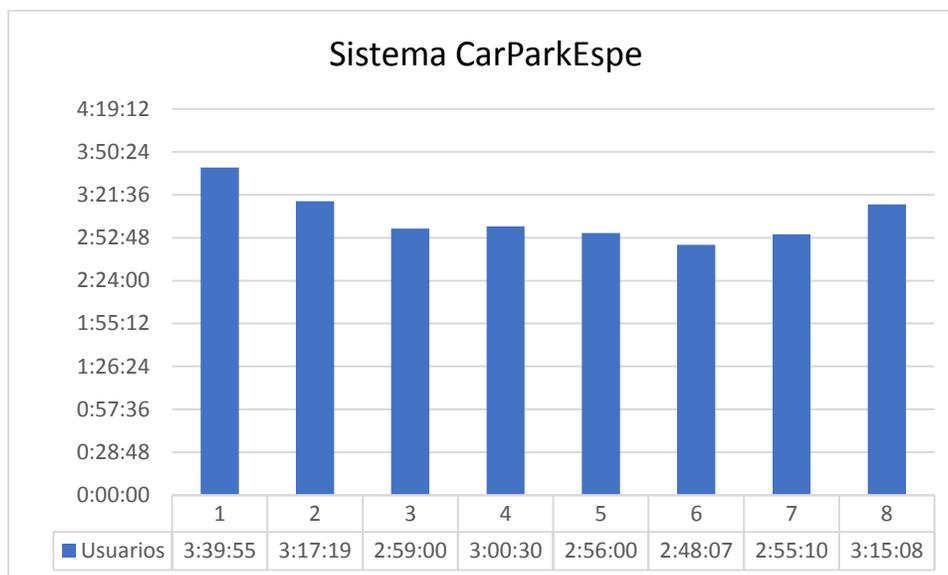
**Figura 32.** Escenario 1, resultado tiempos para encontrar plazas libres (min:s:ms)

## Escenario II

### Tiempos

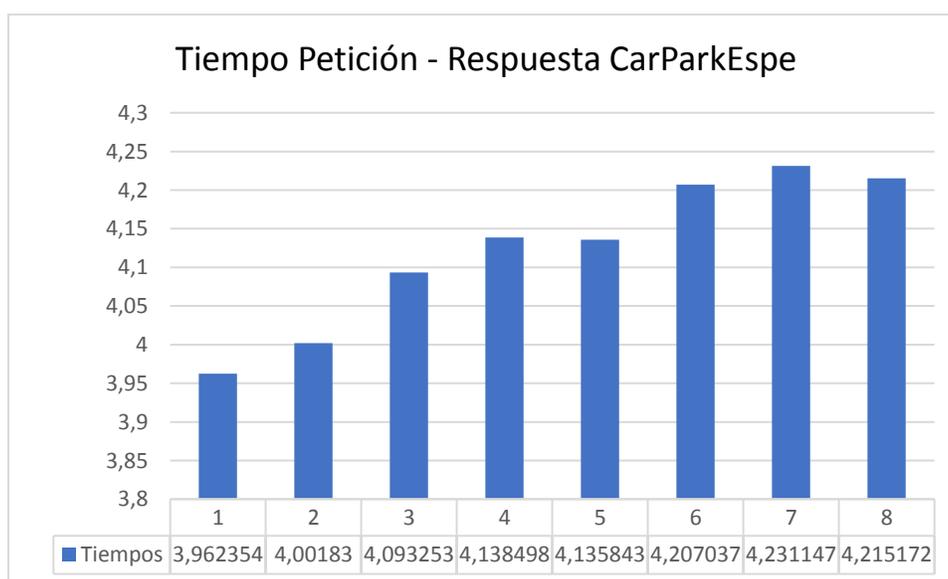
Con el sistema de detección de plazas libres CarParkEspe participaron 8 estudiantes de la Universidad de las Fuerzas Armadas ESPE con automóvil, para la realización de las pruebas de evaluación, en varias sesiones.

El promedio de los tiempos obtenidos para localizar la plaza de aparcamiento libre en el parqueadero situado en los laboratorios de Biotecnología – Electrónica fue de 3 minutos, 6 segundos, 24 milisegundos. En la Figura 33, se puede observar los datos obtenidos de los participantes en minutos, segundos y milisegundos.



**Figura 33.** Escenario 2, resultado tiempos para encontrar plazas libres (min:s:ms)

Por otro lado, el promedio de tiempos desde que el usuario realiza la petición al Bot hasta que obtiene respuesta sobre las plazas libres de estacionamiento es 4 segundos, 12 milisegundos. En la Figura 34, se puede observar los datos obtenidos de los participantes en segundos y milisegundos.



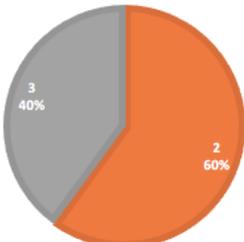
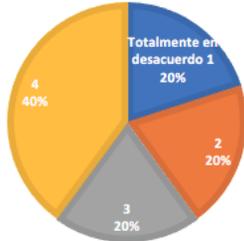
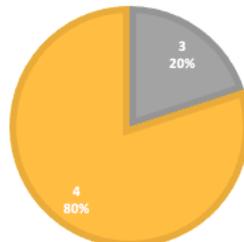
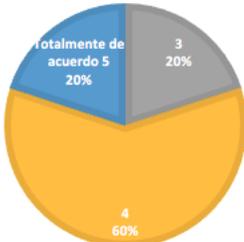
**Figura 34.** Resultado tiempos Petición-Respuesta CarParkEspe (s:ms)

## Encuesta

Se evaluó la usabilidad de la herramienta (Anexo 2) con el objetivo de saber cuán eficiente y satisfactorio es el sistema CarParkEspe para el usuario, con las preguntas de las Tabla 14.

**Tabla 16**  
*Encuesta Usabilidad del Sistema CarParkEspe*

Pregunta	Resultado	Observación
1. Encuentra llamativo el sistema de detección de plazas "CarParkEspe" implementado en el estacionamiento de los Laboratorios de Electrónica de la Universidad de las Fuerzas Armadas ESPE.	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">■ Totalmente en desacuerdo 1 ■ 2 ■ 3 ■ 4 ■ Totalmente de acuerdo 5</p>  <p style="text-align: center;">Totalmente de acuerdo 5 60%</p> <p style="text-align: center;">4 40%</p>	El 60% de los usuarios está totalmente de acuerdo que encuentra llamativo el sistema y el 40% está de acuerdo.
2. Piensa que le gustaría interactuar con frecuencia con el sistema de detección de plazas "CarParkEspe" implementado en el estacionamiento de los Laboratorios de Electrónica de la Universidad de las Fuerzas Armadas ESPE.	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">■ Totalmente en desacuerdo 1 ■ 2 ■ 3 ■ 4 ■ Totalmente de acuerdo 5</p>  <p style="text-align: center;">Totalmente de acuerdo 5 40%</p> <p style="text-align: center;">4 60%</p>	El 60% de los usuarios está de acuerdo le gustaría interactuar con frecuencia con el sistema y el 40% totalmente de acuerdo.

<p>3. Considera que la interacción con el sistema de detección de plazas "CarParkEspe" presenta complejidades al momento de ingresar los comandos.</p>	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">■ Totalmente en desacuerdo 1 ■ 2 ■ 3 ■ 4 ■ Totalmente de acuerdo 5</p>  <table border="1" style="display: none;"> <thead> <tr> <th>Respuesta</th> <th>Porcentaje</th> </tr> </thead> <tbody> <tr> <td>2 (Totalmente en desacuerdo)</td> <td>60%</td> </tr> <tr> <td>3 (Neutral)</td> <td>40%</td> </tr> </tbody> </table>	Respuesta	Porcentaje	2 (Totalmente en desacuerdo)	60%	3 (Neutral)	40%	<p>El 60% de los usuarios está en desacuerdo que la interacción con el sistema presenta complejidades al momento de ingresar los comandos y el 40% es neutral.</p>				
Respuesta	Porcentaje											
2 (Totalmente en desacuerdo)	60%											
3 (Neutral)	40%											
<p>4. Considera que el uso e interacción con el sistema de detección de plazas "CarParkEspe" requiere de experticia previa.</p>	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">■ Totalmente en desacuerdo 1 ■ 2 ■ 3 ■ 4 ■ Totalmente de acuerdo 5</p>  <table border="1" style="display: none;"> <thead> <tr> <th>Respuesta</th> <th>Porcentaje</th> </tr> </thead> <tbody> <tr> <td>1 (Totalmente en desacuerdo)</td> <td>20%</td> </tr> <tr> <td>2 (en desacuerdo)</td> <td>20%</td> </tr> <tr> <td>3 (Neutral)</td> <td>20%</td> </tr> <tr> <td>4 (Totalmente de acuerdo)</td> <td>40%</td> </tr> </tbody> </table>	Respuesta	Porcentaje	1 (Totalmente en desacuerdo)	20%	2 (en desacuerdo)	20%	3 (Neutral)	20%	4 (Totalmente de acuerdo)	40%	<p>El 40% de los usuarios está de acuerdo que el uso e interacción con el sistema requiere de experticia previa y el 20% está totalmente en desacuerdo.</p>
Respuesta	Porcentaje											
1 (Totalmente en desacuerdo)	20%											
2 (en desacuerdo)	20%											
3 (Neutral)	20%											
4 (Totalmente de acuerdo)	40%											
<p>5. Considera que las distintas funciones que presenta el sistema de detección de plazas "CarParkEspe" están bien integradas.</p>	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">■ Totalmente en desacuerdo 1 ■ 2 ■ 3 ■ 4 ■ Totalmente de acuerdo 5</p>  <table border="1" style="display: none;"> <thead> <tr> <th>Respuesta</th> <th>Porcentaje</th> </tr> </thead> <tbody> <tr> <td>3 (Neutral)</td> <td>20%</td> </tr> <tr> <td>4 (Totalmente de acuerdo)</td> <td>80%</td> </tr> </tbody> </table>	Respuesta	Porcentaje	3 (Neutral)	20%	4 (Totalmente de acuerdo)	80%	<p>El 80% de los usuarios está de acuerdo que las distintas funciones que presenta el sistema están bien integradas, el 20% es neutral.</p>				
Respuesta	Porcentaje											
3 (Neutral)	20%											
4 (Totalmente de acuerdo)	80%											
<p>6. Considera que los elementos multimedia presentes dentro del sistema de detección de plazas "CarParkEspe" transmiten información realmente útil.</p>	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">■ Totalmente en desacuerdo 1 ■ 2 ■ 3 ■ 4 ■ Totalmente de acuerdo 5</p>  <table border="1" style="display: none;"> <thead> <tr> <th>Respuesta</th> <th>Porcentaje</th> </tr> </thead> <tbody> <tr> <td>3 (Neutral)</td> <td>20%</td> </tr> <tr> <td>4 (Totalmente de acuerdo)</td> <td>60%</td> </tr> <tr> <td>5 (Totalmente de acuerdo)</td> <td>20%</td> </tr> </tbody> </table>	Respuesta	Porcentaje	3 (Neutral)	20%	4 (Totalmente de acuerdo)	60%	5 (Totalmente de acuerdo)	20%	<p>El 60% de los usuarios está de acuerdo que los elementos multimedia presentes dentro del sistema transmiten información realmente útil, el 20% está totalmente de acuerdo y el otro 20% es neutral.</p>		
Respuesta	Porcentaje											
3 (Neutral)	20%											
4 (Totalmente de acuerdo)	60%											
5 (Totalmente de acuerdo)	20%											

<p>7. Considera que el sistema de detección de plazas "CarParkEspe" le transmite confianza en la información que presenta y seguridad al momento de interactuar con él.</p>	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">Totalmente de acuerdo 5 20%</p> <p style="text-align: center;">4 60%</p> <p style="text-align: center;">3 20%</p>	<p>El 60% de los usuarios está de acuerdo que el sistema transmite confianza en la información que presenta y seguridad al momento de interactuar con él; el 20% está totalmente de acuerdo y el otro 20% es neutral.</p>
<p>8. Considera que el sistema de detección de plazas "CarParkEspe" permite ser utilizado por docentes, personal administrativo, y alumnos de la Universidad de las Fuerzas Armadas ESPE.</p>	<p style="text-align: center;"><b>ENCUESTA USABILIDAD</b></p> <p style="text-align: center;">Totalmente de acuerdo 5 60%</p> <p style="text-align: center;">4 40%</p>	<p>El 60% de los usuarios está totalmente de acuerdo que el sistema permite ser utilizado por docentes, personal administrativo, y alumnos de la Universidad de las Fuerzas Armadas ESPE y el 40% está de acuerdo.</p>

#### 4.4 Análisis Resultados

Los resultados obtenidos muestran qué con el método Tradicional, el promedio para localizar una plaza de aparcamiento libre en el parqueadero situado en los laboratorios de Biotecnología – Electrónica es de 6 minutos, 33 segundos, 48 milisegundos, mientras que, con el sistema de detección de plazas libres mediante procesamiento digital de imágenes y tecnología BOT es de 3 minutos, 6 segundos, 24 milisegundos. Por lo que, el tiempo que disminuye para encontrar una plaza de aparcamiento en dicho sector es de 3 minutos, 27 segundos, 24 milisegundos. Obteniendo una optimización del 47,33% para encontrar una plaza libre en el parqueadero situado en los laboratorios de Biotecnología – Electrónica de la Universidad de las Fuerzas Armadas ESPE, sector Sangolqui.

A través de la información obtenida de la encuesta previa sobre la percepción del usuario sobre el uso del parqueadero de la institución, se puede destacar que al 96,8% de los usuarios le gustaría tener un sistema de aparcamiento en la Universidad de las Fuerzas Armadas ESPE. Esto se debe a que al usuario le toma entre 6 a 8 minutos promedio encontrar una plaza libre de estacionamiento.

Y mediante la información obtenida de la encuesta de usabilidad del Sistema CarParkEspe se obtuvo que a 100% de los usuarios le gustaría interactuar con frecuencia con el sistema, por lo que se determinó la experiencia del usuario respecto a la usabilidad del sistema es eficiente y satisfactorio.

## CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones

Mediante los resultados obtenidos del análisis del Estado del Arte, se determinó utilizar como herramientas para el desarrollo del presente trabajo de titulación, Python como lenguaje de programación y la librería OpenCV para el desarrollo del algoritmo de procesamiento de imágenes.

El modelo que permite la captura de imágenes lo hace a través de la cámara NoIR OV5647 conectada a la Raspberry Pi. Mientras, que el procesamiento de imágenes se basa en la comparación de información de dos imágenes mediante el algoritmo SSIM, que calcula el índice de similitud estructural para la imagen asociada a la matriz A, utilizando P como imagen de referencia. Siendo (A) y (P) matrices del mismo orden, donde (A) corresponde a la imagen de cada plaza a determinar cómo libre u ocupada, y (P) corresponde a la imagen tomada como plantilla.

El desarrollo del Bot se lo realizó mediante la plataforma de Telegram a través de su Bot padre denominado “BotFather” que permitió la creación del Bot “CarParkEspe”. Se utilizó la plataforma de Telegram porque permite acceder gratuitamente a su API pública para desarrolladores e implementar el Bot con la funcionalidad requerida para el presente trabajo.

Mediante la evaluación, se determinó que los resultados fueron significativos. Con el sistema de detección de plazas CarParkEspe, el tiempo de petición – respuesta del Bot sobre las plazas libres fue 4,12 segundos promedio, y los tiempos para localizar la plaza de aparcamiento libre en el parqueadero situado en los laboratorios de Biotecnología – Electrónica fue de 3 minutos, 6 segundos, 24 milisegundos promedio. Mientras, que por el método tradicional fue de 6 minutos, 33 segundos, 48 milisegundos promedio.

Por lo tanto, el sistema de detección de plazas CarParkEspe es viable ya que mediante la evaluación demostró ser factible al optimizar el tiempo de búsqueda de plazas en un 47,33% al método tradicional.

## **5.2 Recomendaciones**

Se sugiere emplear las últimas versiones de Raspberry Pi con mejor capacidad hardware para obtener mayores recursos para el procesamiento de imágenes y evitar que se ralentice.

## **5.3 Trabajos Futuros**

Para reducir la carga de trabajo de la Raspberry se propone implementar el trabajo de procesamiento de imágenes en AWS, de tal manera que la Raspberry junto con la cámara capturaría las imágenes y previamente procesadas en AWS las enviaría al usuario a través del API de Telegram.

Para una próxima propuesta se podría realizar un Bot en una plataforma que utilicen más usuarios como es la plataforma de Facebook Messenger, que también dispone de un API gratuito denominado Messenger API. Sin embargo, su desarrollo implicaría una mayor complejidad y conocimientos previos en el área por parte del desarrollador.

## ANEXOS

### Anexo 1. Encuesta sobre la percepción del uso del parqueadero

#### Percepción sobre el uso del Parqueadero

Encuesta sobre la percepción del usuario en el uso del parqueadero de la Universidad de las Fuerzas Armadas ESPE.

\*Obligatorio

1. Dirección de correo electrónico \*

---

2. Nombres \*

---

3. Apellidos \*

---

4. ¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 7 am? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad. \*

Marca solo un óvalo.

- 3 min
- 5 min
- 8 min
- 11 min
- 15 min
- Más de 15 min

5. ¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 14:00? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad. \*

Marca solo un óvalo.

- Menos de 5 min
- 5 min
- 8 min
- 11 min
- 15 min
- Más de 15 min

6. **¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 15:00? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad.**

*Marca solo un óvalo.*

- Menos de 5 min  
 5 min  
 8 min  
 11 min  
 15 min  
 Más de 15 min

7. **¿Qué tiempo le toma encontrar una plaza de aparcamiento cuando ingresa a la institución sobre las 17:00? Tomando en cuenta que se encuentra en la entrada de ingreso de la Universidad.**

*Marca solo un óvalo.*

- Menos de 5 min  
 5 min  
 8 min  
 11 min  
 15 min  
 Más de 15 min

8. **¿Qué tiempo estima que tarda el usuario en encontrar una plaza de aparcamiento libre desde que ingresa a la institución en horas pico? \***
- 

9. **¿Qué tiempo estima que tarda el usuario en encontrar una plaza de aparcamiento libre desde que ingresa a la institución en horas de menor afluencia? \***
- 

10. **¿Le gustaría tener un sistema de aparcamiento en la Universidad de las Fuerzas Armadas ESPE? \***

*Marca solo un óvalo.*

- Si  
 No

## Anexo 2. Estudio de usabilidad del Sistema “CarParkEspe”

### Estudio de usabilidad del sistema “CarParkEspe” para la detección de plazas de aparcamiento en la Universidad de las Fuerzas Armadas ESPE

Estudio de usabilidad del sistema de detección de plazas "CarParkEspe" en la Universidad de las Fuerzas Armadas ESPE (Zona: Laboratorios de Electrónica)

\*Obligatorio

1. Dirección de correo electrónico \*

---

2. Nombres \*

---

3. Apellidos \*

---

4. 1. Encuentra llamativo el sistema de detección de plazas "CarParkEspe" implementado en el estacionamiento de los Laboratorios de Electrónica de la Universidad de las Fuerzas Armadas ESPE. \*

Marca solo un óvalo.

1      2      3      4      5

Totalmente en desacuerdo      Totalmente de acuerdo

5. 2. Piensa que le gustaría interactuar con frecuencia con el sistema de detección de plazas "CarParkEspe" implementado en el estacionamiento de los Laboratorios de Electrónica de la Universidad de las Fuerzas Armadas ESPE. \*

Marca solo un óvalo.

1      2      3      4      5

Totalmente en desacuerdo      Totalmente de acuerdo

6. 3. Considera que la interacción con el sistema de detección de plazas "CarParkEspe" presenta complejidades al momento de ingresar los comandos. \*

Marca solo un óvalo.

1      2      3      4      5

Totalmente en desacuerdo      Totalmente de acuerdo

7. 4. Considera que el uso e interacción con el sistema de detección de plazas "CarParkEspe" requiere de experticia previa. \*

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

8. 5. Considera que las distintas funciones que presenta el sistema de detección de plazas "CarParkEspe" están bien integradas. \*

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

9. 6. Considera que los elementos multimedia presentes dentro del sistema de detección de plazas "CarParkEspe" transmiten información realmente útil. \*

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

10. 7. Considera que el sistema de detección de plazas "CarParkEspe" le transmite confianza en la información que presenta y seguridad al momento de interactuar con él. \*

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

11. 8. Considera que el sistema de detección de plazas "CarParkEspe" permite ser utilizado por docentes, personal administrativo, y alumnos de la Universidad de las Fuerzas Armadas ESPE. \*

Marca solo un óvalo.

	1	2	3	4	5	
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

Recibir una copia de mis respuestas

Con la tecnología de  
 Google Forms

## REFERENCIAS

Aguirre Juárez, A., Reguilón Rodríguez, D., & Román Urdiales, A. (Septiembre de 2017).

*Telegram Bot: News Bot*. Obtenido de

<https://eprints.ucm.es/45196/1/MemoriaNewsBot.pdf>

Armonk, N. (2011, Septiembre 28). *IBM Global Parking Survey: Drivers Share Worldwide*

*Parking Woes*. Obtenido de <https://www-03.ibm.com/press/us/en/pressrelease/35515.wss>

Carrizo Moreno, D. (2015). Atributos contextuales influyentes en el proceso de educación de

requisitos: una exhaustiva revisión de literatura. *Ingeniare. Revista chilena de ingeniería.*,

23(2), 208-218.

Dangi, V., Parab, A., Pawar, K., & Rathod, S. (s.f.). *Image Processing Based Intelligent Traffic*

*Controller*. Obtenido de

[http://interscience.ac.in/URJA/journals/urja\\_vol1no1/urja\\_paper4.pdf](http://interscience.ac.in/URJA/journals/urja_vol1no1/urja_paper4.pdf)

Duarte Aguirre, O., & González Roque, J. (Enero de 2017). *Aplicaciones de Procesamiento*

*Digital de Imágenes y Videos con Propósitos Académicos*. Obtenido de

<http://ribuni.uni.edu.ni/1355/1/80756.pdf>

Flores, P., & Braun, J. (9 de Agosto de 2011). *Algoritmo RANSAC: fundamento teorico*.

Obtenido de

<http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/Fundamento>

RANSAC.pdf

Flores, P., & Braun, J. (9 de Agosto de 2011). *Algoritmo SIFT: fundamento teorico*. Obtenido de

<http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/Fundamento>

SIFT.pdf

- Gómez, F. (23 de Marzo de 2015). *Diseño y producción audiovisual*. Obtenido de <https://www.deustoformacion.com/blog/disenio-produccion-audiovisual/8-caracteristicas-imagen-digital-que-debes-conocer>
- Jimenez, B. R. (27 de Enero de 2018). *Telegram, una aplicación en crecimiento e imparable*. Obtenido de <https://andro4all.com/2018/01/crecimiento-telegram-cifras-imparable-busquedas>
- Magaña, J., José , A., & Rafael, C. (2015). Pruebas del algoritmo surf para la elección de imágenes de referencia. *Revista Electrónica de Divulgación de la Investigación Vol. 9*.
- Martín, H. F. (25 de Marzo de 2009). *Mapa de bits y Objetos vectoriales*. Obtenido de <https://docplayer.es/53254741-Mapa-de-bits-y-objetos-vectoriales.html>
- OpenCV. (2019). *OpenCV*. Obtenido de <https://opencv.org/>
- Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (Junio de 2008). Systematic Mapping Studies in Software Engineering. *In Ease*, 68-77.
- Raya, A. (3 de Mayo de 2016). *Bots: todo lo que hay que saber para entender la nueva tecnología de moda*. Obtenido de <https://www.tecnologiadetuuatue.elcorteingles.es/software/que-son-los-bots/>
- Robert, B. (2016, Abril 12). *The future of parking Bosch looks to shake up the market with revolutionary ideas*. Obtenido de [https://www.bosch-presse.de/pressportal/de/media/migrated\\_media/automatisiertes\\_parken\\_2016/automatisiertes\\_parken\\_2016\\_7600\\_en-us.pdf](https://www.bosch-presse.de/pressportal/de/media/migrated_media/automatisiertes_parken_2016/automatisiertes_parken_2016_7600_en-us.pdf)
- Sumathi, S., Poornima, R., & Haripriya, T. (29 de Noviembre de 2016). Implementation of Multimodal Neonatal Identification Using Raspberry Pi 2. *Proceedings of the 2nd International Conference on Communication and Information Processing*, (págs. 88-92).

Zambrano, I. M. (Julio de 2011). *Estimación de los parámetros de un modelo haciendo uso de correspondencias con incertidumbre*. Obtenido de <https://ciimat.repositorioinstitucional.mx/jspui/bitstream/1008/255/2/TE%20388.pdf>