

UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería

**Estudio y Análisis de prototipo de parqueaderos inteligentes en zona azul
del Distrito Metropolitano de Quito**

**Yoder Alexis Macas Galarza
Pablo Sebastián García Guevara**

Alberto Sánchez, Ph.D., Director de Tesis

Tesis de grado presentada como requisito
para la obtención del título de Ingeniero Electrónico

Quito, mayo de 2015

Universidad San Francisco de Quito

Colegio de Ciencias e Ingenierías

HOJA DE APROBACIÓN DE TESIS

**Estudio y Análisis de prototipo de parqueaderos inteligentes en zona azul del Distrito
Metropolitano de Quito**

Yoder Alexis Macas Galarza

Pablo Sebastián García Guevara

Alberto Sánchez, Ph.D.,
Director de Tesis

Diego Benítez, Ph.D.,
Miembro del Comité de Tesis

Omar Aguirre, MSc.,
Miembro del Comité de Tesis

Ximena Córdova, Ph.D.,
Decana de la Escuela de Ingeniería del Colegio de Ciencias e Ingeniería

Quito, mayo de 2015

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma: _____

Nombre: Yoder Alexis Macas Galarza

C. I.: 1714833942

Firma: _____

Nombre: Pablo Sebastián García Guevara

C. I.: 1716388184

Lugar: Quito, Ecuador

Fecha: mayo, 2015

DEDICATORIA

Dedicamos este proyecto principalmente a nuestros padres que son los guías de cada acto que realizamos y han sido el apoyo fundamental en nuestra carrera. A nuestros hermanos que siempre han estado presentes para ayudarnos y ser el incentivo para seguir adelante.

.

AGRADECIMIENTOS

Agradecemos a Dios por la vida y por permitirnos llegar hasta este momento tan trascendental de nuestra formación profesional.

A nuestros padres que con cariño y consejos hicieron todo por motivarnos y lograr nuestros sueños.

A nuestras novias que fueron la compañía fundamental y el apoyo incondicional.

A todos los profesionales, principalmente a Alberto Sánchez, nuestro tutor que dando parte de su valioso tiempo orientó con su conocimiento y experiencia para que tenga éxito este proyecto.

A la Universidad San Francisco de Quito por prepararnos para los retos que pone la vida.

RESUMEN

Este proyecto describe el desarrollo y funcionamiento de un prototipo, que permitirá la detección automática del estado de los parqueaderos en la ciudad de Quito, específicamente en la zona azul. Este prototipo consiste de una cámara de conexión USB acoplada a un servomotor, que permite girar a la cámara para adquirir la imagen de varios parqueaderos, los que se encuentran conectados al dispositivo MyRIO de National Instruments para controlar en tiempo real la adquisición y procesamiento de imágenes que se realiza en dos procesos: segmentación y comparación de las mismas.

El proceso de segmentación que consiste en determinar la posición de cada estacionamiento en una imagen panorámica, se realiza una sola vez después de que se coloca el prototipo en un lugar fijo, en un poste o cierto lugar de altura. Este movimiento determina el número de parqueaderos disponibles que se encuentran dentro del rango visual alcanzado por la combinación del uso de la cámara y el servo. Con este paso se obtiene las coordenadas de los pixeles de largo y ancho de cada uno de los parqueaderos que se detectan, permitiendo recortar la imagen y comparar uno por uno y determinar su estado ya sea libre u ocupado.

La fase de comparación se realiza constantemente y en tiempo real. Esta toma los datos obtenidos durante el proceso de segmentación y adquiere las imágenes de los parqueaderos, las recorta y las compara con las imágenes obtenidas en una iteración anterior del proceso. Al comparar se determina si existe un cambio de estado.

Esta información es guardada en un arreglo, que es transmitido hacia un servidor SQL y añadida a una base de datos, la misma que se utiliza para mostrar el estado del parqueadero en una página web.

ABSTRACT

This report describes the development and operation of a prototype for automatically detecting the parking spot state in the 'zona azul' that is deployed in Quito. The prototype consists of a USB camera coupled to a servo motor, which allows the camera to acquire different parking spots images by rotating. Both the servo motor and the camera are connected to a MyRIO device from National Instruments. This device works in real time by controlling the camera and the servo motor to acquire and process images in order to perform two tasks: segmentation and comparison of the images.

The segmentation process, identifies the position of each parking lot on a panoramic image, and it is performed only once after the prototype is placed in a fixed location, this should be a pole or a high place. This process determines the number of parking lots that are within the visual range obtained by combining the use of the camera and the servo motor, and obtains the coordinates of the pixels in length and width of each of the parking lots that are detected, this will allow us to crop the image and compare parking lots one by one and determine their individual status.

The comparison step is performed continuously and in real time. This process takes the data obtained during the segmentation and acquires images of the parking spots, then crops it and compares the images obtained in a previous iteration of the comparison process. By comparing the images it determines whether its status has changed or not.

The information obtained from the comparison is stored in an array that is transmitted to an SQL server and this information is added to a database, which is used to display the information about the status of the parking spaces in a web page.

TABLA DE CONTENIDO

Resumen	7
Abstract	8
INTRODUCCIÓN	14
Descripción del Proyecto.....	14
Antecedentes	14
Justificación.....	16
Objetivo General	17
Objetivos Específicos	17
Presentación.....	18
Logros Alcanzados	19
MARCO TEÓRICO	20
Revisión Bibliográfica.....	20
Adquisición digital de imágenes.....	22
Procesamiento de imágenes	25
Hardware de procesamiento de imágenes.....	32
Lenguajes de descripción de hardware HDL.....	34
Transmisión de datos	35
Resumen del Capítulo	36
METODOLOGÍA Y DISEÑO	38
Justificación de la metodología seleccionada.....	38
Adquisición de Imágenes.....	38
Sistema de Movimiento Rotacional.....	45
Procesamiento de Imagen.....	49
Descripción del Sistema de Comunicación e integración en red	57
Manejo de información y Base de Datos	63
Interfaces del cliente	72
Resumen del Capítulo	79
ANÁLISIS DE RESULTADOS.....	81
Determinación Cámara Web.....	81
Determinación Proceso de adquisición de imágenes	84
Caracterización Sistema de Movimiento Rotacional.....	86
Comprobación y optimización del proceso de corte de imagen	87
Caracterización Sistema de Comparación.....	95
Pruebas de confiabilidad del sistema de Comparación	99
Optimización Sistema de Comunicación MyRIO servidor SQL.....	103
Resumen del Capítulo	104
CONCLUSIONES	106
REFERENCIAS	111
ANEXO A: Antecedentes	114

ANEXO B: Análisis de Resultados	117
ANEXO C: Programas de prueba en Labview	127
ANEXO D: Programa De Segmentación	128
ANEXO E: Programa De Comparación.....	134

TABLAS

Tabla 1.- Resumen matrices de filtros gradiente Horizontal, Vertical Prewitt y Sobel	28
Tabla 2.- Tabla de verdad lógica de comparación.....	56
Tabla 3.- Tabla de Verdad lógica de comparación con estado anterior	56
Tabla 4.- Tabla de Direccionamiento IP Red “MyRIO”	59
Tabla 5.- Parámetros de acceso servidor MySQL	66
Tabla 6.- Variables Independientes análisis confiabilidad del sistema	87
Tabla 7.- Muestras de grupo de variaciones a diferentes horas del día y estado climáticos	87
Tabla 8.- Resultado Mathscript muestra "Nublado"	88
Tabla 9.- Resultado Mathscript muestra “Sol sin sombra”	89
Tabla 10.- Resultado Mathscript muestra “Sol con sombra”	90
Tabla 11.- Resultado Mathscript muestra "Lluvia"	91
Tabla 12.- Resultado Mathscript muestra "Neblina"	92
Tabla 13.- Resultado Mathscript muestra "Tarde”	93
Tabla 14.- Tamaño promedio en ancho y largo de los parqueaderos tomando en cuenta los casos con éxito (Nublado, Sol sin Sombra, lluvia, Tarde).....	94
Tabla 15.- Variación promedio en muestra "nublado” utilizando imágenes escala de grises	95
Tabla 16.- Variación promedio en muestra "nublado” utilizando detección de bordes	96
Tabla 17.- Límites rango de tolerancia del sistema	102
Tabla 18.- Porcentaje de Variación del sistema frente al cambio o no de estados.....	103
Tabla 19.- Características de tarjeta de red inalámbrica NI MyRio1900	114
Tabla 20.- Número de vehículos motorizados matriculados según provincias año 2010 .	114
Tabla 21.- Parámetro de Imagen cámara Logitech C160	117
Tabla 22.- Parámetros de Imagen cámara Genius Facecam 1000x	118
Tabla 23.- Tiempos para captura de imagen vs. Métodos de apertura	118
Tabla 24.- Variación promedio de muestras (100% y 60%)	123
Tabla 25.- Porcentajes de variación análisis tiempo.....	125
Tabla 26.- Porcentajes de variación análisis clima.....	125

FIGURAS

Figura 1.- Geometría Proyectiva de una vista	22
Figura 2.- Transformación euclidiana de coordenadas.....	24
Figura 3.- Modelo RGB para percepción de colores	26
Figura 4.- Imagen de muestra Barbara.tif (izquierda) e imagen de muestra aplicada filtro de Diferencias Vertical con tolerancia=0.1 (derecha)	27
Figura 5.- Transformación Isométrica	29
Figura 6.- Component Labeling	30
Figura 7.- Thresholding para eliminación de ruido	30
Figura 8.- Expansión de imágenes.....	31
Figura 9.- Cámara Genius facevam 1000x (izq) y Cámara Logitech C160 (der)	40
Figura 10.- Conexión de la cámara al puerto USB de NI MyRIO 1900	40
Figura 11.- Programa para abrir la cámara	40
Figura 12.- Módulo IMAQdx Open Camera.vi.....	41
Figura 13.- Módulo Propiedades de Node.....	41
Figura 14.- Módulo IMAQdx Configure Grab.vi.....	41
Figura 15.- Módulo IMAQ Create.....	42
Figura 16.- Módulo IMAQdx Grab.vi	42
Figura 17.- Módulo IMADx Close Camera.vi (izq) Módulo Imaq Dispose (der)	42
Figura 18.- Módulo Vision Acquisition Express.....	43
Figura 19.- Selección de cámara usando Vision Acquisition Express	43
Figura 20.- Alternativas de adquisición de imágenes en módulo Express	44
Figura 21.- Parámetros de Adquisición	44
Figura 22.- Módulo Transformación Imagen a arreglo	45
Figura 23.- Salidas del Dispositivo NI MyRio1900	46
Figura 24.- Conexión del Servomotor	47
Figura 25.- Programa para controlar Servomotor desde Labview 2014	47
Figura 26.- Módulo de Apertura del canal PWM.....	48
Figura 27.- Modulo de generación de señal PWM.....	48
Figura 28.- Módulo Reset myRIO.vi.....	49
Figura 29.- Foto de parqueadero de inicio del proceso	49
Figura 30.- Aplicación de filtros de extracción de color azul y escala de grises.....	50
Figura 31.- Imagen con Detección de Bordes	50
Figura 32.- Imagen con Transformada Euclidiana	51
Figura 33.- Detección de bordes con threshold de corrección	52
Figura 34.- Imagen con dilatación de líneas.....	52
Figura 35.- Imagen Total	53
Figura 36.- Mathscript de corte de las imágenes	53
Figura 37.- Imagen completa fase comparación.....	54
Figura 38.- Corte de imagen a comparar	54
Figura 39.- Corte de imagen al 100%	55
Figura 40.- Corte de imagen al 60%	55
Figura 41.- Topología de Red Local.....	58

Figura 42.- Interfaz Web de administración de Router para la creación de Red MyRio	59
Figura 43.- Configuración direccionamiento estático NI MyRIO 1900.....	60
Figura 44.- Módulo TCP Listen.vi	61
Figura 45.- Módulo TCP Write.vi (izq.) Módulo TCP Close Connctio.vi (der)	61
Figura 46.- Diseño de Servidor TCP/IP	62
Figura 47.- Módulo TCP Open Connection.vi (izq.) Módulo TCP Read.vi (der)	63
Figura 48.-Diseño de Cliente TCP/IP.....	63
Figura 49.- Pantalla de instalación de XAMPP.....	64
Figura 50.- Ventana del Panel de Control XAMPP	65
Figura 51.- Verificación de levantamiento de Servidor SQL en cmd	66
Figura 52.- Diagrama EER base de datos sql_tesis	67
Figura 53.- Configuración de columnas en tablas de base de datos sql_tesis	67
Figura 54.- Código de creación Base de datos sql_tesis	68
Figura 55.- Tabla de comandos para actualizar base de datos.....	69
Figura 56.- Demostración de la creación de la base de datos sql_tesis a través de ventana cmd.....	69
Figura 57.- Comunicación MyRIO-Servidor SQL en texto plano	70
Figura 58.- Programa del archivo parqueadero.bat	71
Figura 59.- Programa tesis.sql	71
Figura 60.- Tarea programada actualización_tesis	72
Figura 61.- Página Web Smart Parking	73
Figura 62.- Directorio archivos página Web en servidor web Apache	73
Figura 63.- Código archivo index.php.....	74
Figura 64.- Código archivo script.js	75
Figura 65.- Código función initialize del archivo script.js	76
Figura 66.- Condiciones de cada estado de parqueadero.....	76
Figura 67.- Declaración de marcador para cada parqueadero en programa script.js.....	77
Figura 68.- Programa conexion_servidor.php	78
Figura 69.- Programa conexion_servidor.php	78
Figura 70.- Pagina web desde dispositivo móvil.....	79
Figura 71.-Cálculo de ángulo de Visión de una cámara.....	82
Figura 72.- Imagen Cámara Logitech C160	83
Figura 73.- Imagen cámara Genius Facecam 1000x	84
Figura 74.- Pruebas apertura de cámara ambiente controlado.....	84
Figura 75.- Imagen apertura de cámara con incidencia directa del sol	85
Figura 76.- Imagen capturada con Vision Acquisition Express.....	85
Figura 77.- Prueba de comparación clima nublado	89
Figura 78.- Prueba de comparación clima sol sin sombra.....	90
Figura 79.- Prueba de comparación muestra “Lluvia ligera”	92
Figura 80.- Prueba de comparación muestra “Tarde”	94
Figura 81.- Análisis de Variación del indicador en diferentes horas del día.....	101
Figura 82.- Análisis de Variación del indicador en diferentes condiciones climáticas.....	102
Figura 83.- Variación Acumulada de ventas Vehiculares de Dic-2013 a Dic-2014 (AEADE, 2015)	114
Figura 84.- Servomotor marca Standard Hitec HS-311	118
Figura 85.- Adquisición de imagen análisis tiempo a las 19:00pm muestra “Tarde”	119
Figura 86.- Comparación análisis de segmentación	119
Figura 87.- Análisis de segmentación muestra "lluvia"	120

Figura 88.- Análisis de segmentación muestra "neblina"	120
Figura 89.- Comparación imágenes en escala de grises	120
Figura 90.-Comparación de imágenes con detección de bordes	121
Figura 91.- Imágenes cortadas al 100%.....	121
Figura 92.- Imágenes cortadas al 60%.....	122
Figura 93.- Fase comparación análisis clima muestra "Sol con sombra"	122
Figura 94.- Fase comparación 60% muestra "Sol con Sombra"	124
Figura 95.- Fase comparación análisis tiempo muestra "Medio Día"	124
Figura 96.- Límites de tolerancia de la variación en Tiempo y Clima	126
Figura 97.-Comunicación MyRIO-Servidor SQL en texto .xls	127
Figura 98.- Movimiento de servomotor lado izquierdo Programa Segmentación	128
Figura 99.- Adquisición de imagen lado izquierdo - Programa de Segmentación	128
Figura 100.-Movimiento de servomotor lado derecho – Programa de Segmentación	129
Figura 101.- Adquisición de imagen lado derecho - Programa Segmentación	129
Figura 102.- Inicialización de arreglo imagen total - Programa Segmentación	130
Figura 103.- Construcción de arreglo con imágenes antes capturadas - Programa Segmentación.....	131
Figura 104.- Visualización arreglo Imagen Total - Programa Segmentación	131
Figura 105.- Segmentación de arreglo Imagen Total -Programa Segmentación.....	132
Figura 106.- Inicialización de arreglos donde se encuentran las medidas de los parqueaderos cortados (ancho y largo) - Programa Segmentación.....	132
Figura 107.- Script de programa Segmentación en el Mathscript de LabVIEW.....	133
Figura 108.- Movimiento de servomotor lado izquierdo - Programa Comparación	134
Figura 109.- Adquisición de imagen lado izquierdo - Programa de Comparación	134
Figura 110.- Movimiento de servomotor lado derecho - Programa Comparación.....	135
Figura 111.- Adquisición de imagen lado derecho - Programa de Comparación.....	135
Figura 112.- Construcción de arreglo con imágenes antes capturadas - Programa Comparación	136
Figura 113.- Visualización de Imagen total - Programa Comparación	136
Figura 114.- Corte de imágenes para comparar - Programa Comparación	137
Figura 115.- Resta de imágenes - Programa de Comparación	137
Figura 116.- Algoritmo de Comparación - Programa de Comparación	138
Figura 117.- Envío de datos protocolo TCP/IP - Programa de Comparación	138
Figura 118.- Script del programa corte en Mathscript de LabVIEW	139

INTRODUCCIÓN

Descripción del Proyecto

El presente proyecto tiene como finalidad proponer una alternativa tecnológica para resolver la realidad emergente que sufre la Ciudad de Quito, relacionada con la movilidad y búsqueda de puntos de parqueo. El proyecto consiste en elaborar un dispositivo electrónico prototipo que procesa imágenes para detectar si un parqueadero está o no disponible en el Sistema de Estacionamiento “Quito Zona Azul”. De igual manera, esta información es enviada mediante comunicación TCP/IP a un servidor, donde posteriormente una Aplicación Móvil en tiempo real, puede consultar el estado de un parqueadero específico u obtener el listado de parqueaderos disponibles alrededor de su ubicación geográfica actual. Con esta solución tecnológica, se espera poder reducir el tiempo de búsqueda de un parqueadero y además obtener una reducción sustancial en el consumo de combustibles y la emisión de gases.

Antecedentes

El incremento acelerado del parque automotor de la ciudad de Quito, la densidad poblacional creciente y la reducción notable de la capacidad de la vía por usos inadecuados como el uso del vía para estacionamientos no permitidos, son elementos que motivan el desarrollo y uso de tecnologías para eliminar la mayor incidencia de congestión vehicular relacionada a la búsqueda de parqueaderos. Este es un problema que no solo se presenta en Ecuador sino en todo el mundo, ya que a medida que la población de las ciudades aumenta, también crecen los servicios y productos que los ciudadanos requieren. En tal virtud, entre los problemas que sufren las grandes ciudades está la movilidad, y en especial

la búsqueda de parqueaderos. Según C. Edward en su artículo “New way of working are needed to make smart cities a reality” alrededor del 30% del tráfico en las ciudades es causado por búsqueda de estacionamientos. (Edward, 2013)

El último censo realizado en el año 2010, estadísticas del INEC reflejan el aumento demográfico del DMQ en un 58,82% con respecto al año 1990 (Censo, 2011). Así como la Agencia Nacional de Tránsito registra que el año 2013 contó con la matriculación de 387.858 vehículos en la provincia de Pichincha (Tabla 20) (INEC, 2013). Para el año 2014 según la Asociación de Empresas Automotrices del Ecuador (AEADE), las ventas de vehículos ascienden a 120.015 unidades, que comparada con el año 2013 (113.812 unidades) se observa un incremento de 5.45% (Figura 83) (AEADE, 2015). Datos que permiten observar como la ciudad requiere de una realidad emergente de diseño y planificación de la movilidad.

El Distrito Metropolitano de Quito mediante Ordenanza Metropolitana N° 0221, publicada en el Registro Oficial N° 195 de 22 de octubre de 2007, implementa el Sistema de Estacionamiento Rotativo Tarifado “Quito Zona Azul” que permite el control y manejo de los espacios públicos como puestos de parqueo (Quito, Ordenanza Metropolitana Registro Oficial N195, 2007). Es así que, 7.471 plazas fueron creadas paulatinamente en cinco zonas administradas a través de distribuidores privados que operan hasta la actualidad. (Quito, Sesión Ordinaria , 2012).

Referente a estos antecedentes y siguiendo la tendencia tecnológica de las ciudades inteligentes, nace la idea de implementar una solución especializada que permita detectar e informar fácilmente los puestos de parqueo vacíos, reduciendo de esta forma el tiempo de búsqueda de parqueo. Para este proyecto se emplea la detección de objetos mediante el procesamiento de imágenes que determinan si un parqueadero está o no ocupado. Esta

tecnología no solo mejora el tráfico en la ciudad, sino también reduce los índices de contaminación contribuyendo a una mejor calidad de vida de los ciudadanos.

Justificación

En la actualidad el aumento de la población en las ciudades, así como las necesidades de sus habitantes son ineludibles. Es así que según I. Tossel, alrededor del 60 % del PIB mundial es concebido por la producción de las ciudades (Tossel, 2014). De igual forma, para el año 2040 el 65% de la población total se encontrará viviendo en el sector urbano, afectando no solo a nivel social sino también a nivel económico y hasta ambiental (Tossel, 2014). Por lo que, la evolución de tecnologías emergentes será la que mejore la situación actual con respecto a la interconexión del ser humano con los dispositivos electrónicos.

Dentro de este contexto, la tendencia de las Ciudades Inteligentes, permiten el manejo de la información de forma centralizada y abierta, la creación de nuevos negocios y fuentes de trabajo, la reducción de costos y sobretodo una mejor sostenibilidad económica, social y ambiental a los ciudadanos (Villanueva, 2014). Creando redes sensoriales y automatizando procesos a lo largo de las ciudades que permiten solucionar problemas relacionados al Transporte, Manejo de Desechos, Contaminación ambiental y Parqueaderos (Danaraj, 2014) De acuerdo al avance de las ciudades y al crecimiento del patio automotor a nivel mundial, el transportarse se ha convertido en un desperdicio de recursos y tiempo. Más aún a la hora de buscar un estacionamiento en las horas pico de las zonas con altos índices de congestionamiento.

Para solucionar este mal que aqueja a la mayoría de ciudades y en este caso al Distrito Metropolitano de Quito, se presenta la implementación de un prototipo de

parqueadero inteligente en el Sistema de Estacionamiento Rotativo Tarifado “Quito Zona Azul” mediante el procesamiento de imágenes. Utilizando cámara Web se adquieren las imágenes, y luego de un procesamiento posterior se indicarán los puestos libres de parqueo. A su vez, a través de una aplicación web se podrá consultar el estado y la ubicación exacta del parqueadero disponible en tiempo real. Esta solución permite reducir el tiempo de búsqueda, el consumo de combustible, la congestión de tráfico en puntos claves de la ciudad y un porcentaje de gases contaminantes. Aproximadamente alrededor de 33,5 toneladas de gases son emitidos anualmente por la cantidad de automóviles que buscan un estacionamiento. (Smart- Parking Latin America, 2012).

Objetivo General

Desarrollar un sistema que permita determinar si un lugar de estacionamiento se encuentra ocupado o no, procesando una imagen capturada desde una cámara digital. Los datos serán desplegados de forma abierta en un portal para la posterior creación de aplicaciones.

Objetivos Específicos

- Explorar la tecnología de hardware MyRIO para determinar su eficiencia en la adquisición y procesamiento de imágenes.
- Determinar las condiciones necesarias de adquisición de imagen para elegir el tipo de cámara adecuada a utilizar.
- Encontrar el mejor algoritmo que permita la adquisición, análisis y procesamiento para fijar el cambio de estado del parqueadero.

- Probar el sistema en diferentes tipos de condiciones tanto ambientales como de iluminación.
- Determinar un rango de tolerancia para el reconocimiento de un estado al otro.

Presentación

Este proyecto se encuentra organizado en cinco capítulos. El **Capítulo 1 Introducción** presenta la justificación y planteamiento de los antecedentes que dan como resultado la necesidad de crear un sistema que facilite la búsqueda de parqueaderos en la zona azul de Quito. Además se describen los objetivos a alcanzarse a través de este proyecto. El **Capítulo 2 Marco Teórico** aborda todos los fundamentos teóricos que se necesitan para comprender el funcionamiento del prototipo que se desarrolló en esta tesis. De igual manera se realiza la revisión bibliográfica relacionada con los avances actuales en temas sobre ciudades inteligentes, procesamiento de imágenes y parqueaderos inteligentes. El **Capítulo 3 Metodología** aborda de manera detallada cada uno de los pasos y actividades que se realizaron para el prototipo. Se describen los pasos de adquisición de imágenes, uso del sistema de movimiento rotacional, procesamiento, segmentación y comparación de imágenes. Además se describe el tipo de comunicación que se establece entre el hardware de procesamiento y el servidor de manejo de base de datos. También se describe el proceso de levantamiento del servidor SQL y la creación de una página web para implementar una interfaz con el cliente. En el **Capítulo 4 Resultados** se presenta de manera extensa las pruebas que se efectuaron para escoger las mejores opciones de cámara, sistema de movimiento rotacional y rango de tolerancia para la comparación a

utilizarse. Además se muestran los resultados de las pruebas realizadas en diferentes escenarios, como condiciones climáticas y horas del día en las fases de comparación y segmentación, para determinar la robustez del sistema. Finalmente en el **Capítulo 5 Conclusiones** se presenta las resoluciones que se obtienen a partir de los resultados obtenidos en el capítulo anterior. Además se añade un comentario acerca de la factibilidad del proyecto y recomendaciones para avances futuros.

Logros Alcanzados

- Se logró seleccionar los mejores dispositivos para la implementación del prototipo.
- Se logró realizar un proceso de corte automático de la imagen independiente para determinar el número y tamaño de parqueaderos que lo contenga.
- Se logró utilizar un algoritmo de comparación sencillo, eficiente y robusto para establecer el cambio de estado en los estacionamientos.
- Se logró desarrollar, implementar y probar el prototipo con resultados favorables.
- Se determinaron las mejores condiciones de tiempo y clima para implementar el proceso de segmentación y comparación.
- Se establecieron los límites de funcionamiento del prototipo desarrollado.
- Se construyó una interfaz entre el hardware de procesamiento y el servidor de manejo de bases de datos.
- Se construyó una página web para presentar la información del estado de los parqueaderos a los clientes.

MARCO TEÓRICO

Revisión Bibliográfica

Investigaciones sobre la importancia y avances relacionados a las ciudades inteligentes

La nueva tendencia de las Ciudades Inteligentes en Europa y América Latina según (Tossel, 2014) en el artículo “the RISE of the SMART CITY” se ve reflejado por el rápido crecimiento del desarrollo e implementación de este concepto. La idea principal radica en sensorizar una ciudad en la mayor parte de servicios y productos que los ciudadanos consumen y no tienen control. Si bien el concepto de Ciudades Inteligentes no solamente se basa en solucionar problemas ya presentes en la ciudadanía, sino también apoyar al desarrollo de medidas que mejoren la eficiencia y ahorro de recursos como el agua y la energía eléctrica.

Entre los beneficios según (López, 2014) una Smart City principalmente provee a sus habitantes de una mejor calidad de vida, gestionando eficientemente los recursos y, por ende, creando un sistema de sostenibilidad dentro de la ciudad. Este principio se entiende mejor, siempre y cuando la tecnología que se implemente para solucionar un mal común, sea más fácil que hacerla sin ella. Como indica (Edwards, 2013) en su artículo “NEW WAYS OF WORKING ARE NEEDED TO MAKE SMART CITIES A REALITY” alrededor del 70% de la población de cada región vivirá dentro de las ciudades, es decir que aumentará la demografía sin que estas crezcan. Es por esto que, las Ciudades Inteligentes son una solución inminente para las nuevas necesidades de las ciudades del futuro. El impulso de crear Ciudades Inteligentes no es limitar el consumo de los recursos o disponibilidad de los servicios, sino regularizar para que no exista un desperdicio de los mismos (Tossel, 2014). Si bien la sensorización de la ciudad es uno de los principales

factores, la red inalámbrica y la velocidad de transmisión de datos son ejes fundamentales para la intercomunicación de estos sistemas. Recientemente Intel en su última presentación (Burt, 2014) exhibió su nuevo chip que soporta los protocolos del Internet de las cosas. Con los nuevos protocolos y la implementación IPv6, las aplicaciones que actúan en las Ciudades Inteligentes serían soportadas ya que el número de direcciones IP así como la velocidad de transmisión aumentarían drásticamente.

Existen grandes proyectos innovadores como Smart Parking que se basa en identificar los parqueaderos disponibles a través de una aplicación móvil mediante el uso de una red de sensores colocados sobre el pavimento, que informa mientras se conduce por la ciudad (FAULKNER, 2014) ; Smart Light donde se manipulan variedad de lámparas con diferentes tipos de tecnologías que reduce el consumo de energía en todo el alumbrado público (Rutkin, 2014) hasta Smart Water que regula el consumo de este recurso gracias a indicadores localizados en cada domicilio (Tossel, 2014)

Investigaciones sobre la implementación de procesamiento de imágenes para Sistema de Parqueo Inteligente

La iniciativa de implementar los parqueaderos inteligentes que detecten si un sitio de estacionamiento está o no ocupado, es mediante la utilización de una cámara y el procesamiento de imágenes. Según (Funck S, 2004) entre las opciones para optimizar este sistema inteligente de localización de un espacio de parqueo, es el uso del circuito cerrado de televisión ya instalado en la mayoría de establecimientos. Lo importante es que, en el proceso de adquisición de imágenes, los componentes de la ventana del navegador no demanden una actividad compleja como utilizar imágenes de color o recursos con alta resolución, ya que solo se necesita la obtención de fotogramas estáticos. Según (X. Ying, 2014) la complejidad radica en la velocidad de los fotogramas estáticos que se adquiere de

un video. Para esto se tiene que tomar en cuenta las condiciones de la cámara a utilizar, si se coloca en posición estática, el algoritmo será diferente frente a una cámara con sistema giratorio con la que aumenta el ángulo de visión. Debido a esto, el procesamiento y el algoritmo se complican por cuanto se debe analizar el sincronismo y los tiempos de validación. (W. Genshun, 2014)

Adquisición digital de imágenes

La adquisición digital de imágenes es un proceso que resulta de transformar la proyección de una imagen espacial 3D a un plano 2D, y convertirlo en datos digitales para su posterior manipulación. Este fundamento se basa en la geometría Proyectiva, que considera la perspectiva de enfoque a uno de los ejes espaciales. (Richard Hartley, 2003).

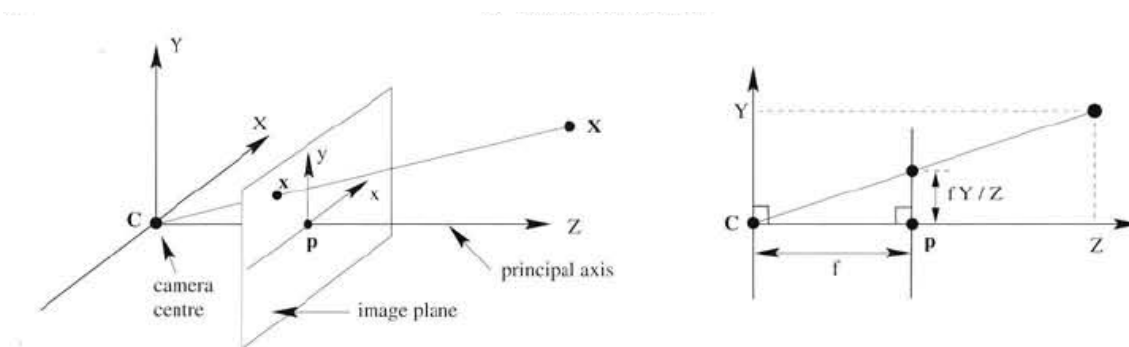


Figura 1.- Geometría Proyectiva de una vista
Fuente (Richard Hartley, 2003)

Actualmente las cámaras digitales manejan tecnología para adquirir imágenes “Charge Coupled Device” (CCD), que generan pulsos eléctricos debido a la luz captada por el lente. (Bell Labs, 2013) Para el caso del uso de cámaras, la geometría planar utiliza el punto central del lente como el eje de perspectiva donde se proyecta la imagen que se observa en la Figura 1. A este proceso se le denomina “Modelo de cámara” y contiene dos

subgrupos, el primero dependiendo de la posición del punto de perspectiva se clasifica en “Centro Finito” y “Centro al Infinito”, y el segundo que descompone la matriz fundamental para determinar puntos geométricos del mismo llamado “Modelo de cámara proyectiva”. (Richard Hartley, 2003).

Modelo Basic Pinhole

Es parte del modelo de cámaras con centro finito que basa su proyección en Coordenadas Homogéneas. Este modelo considera el centro de la proyección como el origen del sistema Euclidiano de coordenadas. Como se muestra en la ecuación (1), la matriz homogénea de proyección de 3x4, se compone con una diagonal (f, f, 1). La forma simplificada se observa en la ecuación (2) donde X representa el espacio real en tres dimensiones, P la matriz de proyección de cámara y x la representación de un punto en la imagen en coordenadas homogéneas. (Richard Hartley, 2003)

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

$$x = PX \quad (2)$$

Para completar el modelo Basic Pinhole, la matriz de cámara P se descompone en la matriz K que depende de los parámetros de calibración de la cámara, y la matriz R que contiene los índices de rotación y translación de los ejes de referencia. Para definir la matriz K se tiene en cuenta que no siempre el origen de coordenadas del plano x coincide con las de la cámara. La matriz K se muestra en la ecuación (3). (Richard Hartley, 2003)

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Del mismo modo, la relación entre las coordenadas del centro de la cámara y del espacio 3D, puede variar dependiendo de cuál sistema se toma como el eje principal. Para representar esta relación, se considera una transformación en rotación y traslación de las coordenadas como se muestra en la Figura 2.

$$X_{cam} = R(X - C) \quad (4)$$

Donde las coordenadas de la cámara son trasladadas un factor C , que es la posición de origen de la cámara y rotada dentro de una matriz R como se muestra en la ecuación (4).

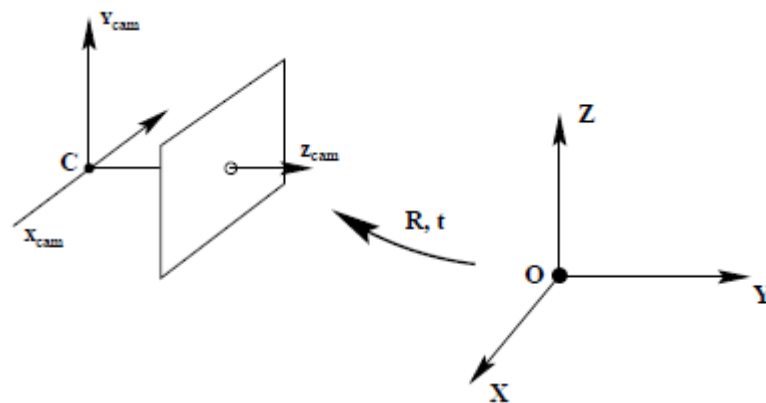


Figura 2.- Transformación euclidiana de coordenadas
Fuente (Richard Hartley, 2003)

Por último considerando estas modificaciones, la matriz P se resume como la ecuación (5).

$$x = KR(I - C) \quad (5)$$

Procesamiento de imágenes

El procesamiento digital de imágenes aparece tardíamente en la historia de la computación, debido a que se necesitaba desarrollar el hardware específico y los sistemas operativos que permitieran manipular las imágenes. Incluso después de tener las herramientas adecuadas, se debía optimizar los algoritmos y las técnicas, para conseguir un mejor manejo del hardware y obtener excelentes y más rápidos resultados. En la actualidad existen muchos programas, que permiten manipular las imágenes y procesarlas de diferentes maneras utilizando distintos algoritmos, algunos de los cuales están documentados y otros no. (Ekstrom, 1984)

Bitmaps (mapas de bits)

Constituyen la forma más básica de representar una imagen digital en la memoria de una computadora. Está formado por filas de píxeles que contienen el valor que fija el color. El valor contiene 3 números de 8 bits que determinan la intensidad de cada uno de los colores primarios (Rojo, Verde y Azul). Cualquier color visible por el ojo humano se puede simbolizar de esta manera. Esta técnica se conoce como codificación RGB y representa las siglas de los 3 colores primarios en inglés. Teniendo en cuenta la capacidad de almacenamiento que representa un byte desde el punto de vista de la computación, cada color contiene 8 bits ya que el ojo humano es capaz de diferenciar un color, más de 256 niveles de intensidad. (Jähne, 2005).

Percepción del color

La percepción digital del color radica en los receptores de luz cromática que tiene el ojo humano. La luz cromática visible para el ser humano, se expande en el espectro electromagnético con longitudes de onda entre 400 a 700 nm (Rafael C. Gonzalez, 2002).

Los factores para describir sus cualidades son la radiancia, iluminancia y brillo, que dependiendo de su valor determinan un color específico. Para la interpretación estandarizada de los colores se utiliza el modelo RGB, el cual indica que cada color se representa por una cantidad del espectro de los colores primarios rojo, verde y azul. Es decir una imagen a color se compone de tres imágenes monocromáticas con 256 niveles de intensidad. Un ejemplo se observa en la Figura 3.

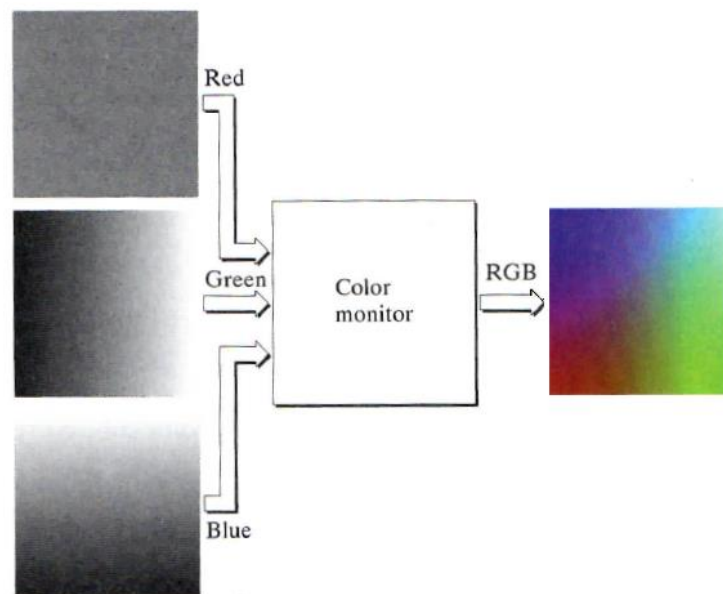


Figura 3.- Modelo RGB para percepción de colores
Fuente (Rafael C. Gonzalez, 2002)

Detección de Bordes

La detección de bordes es un campo en el procesamiento de imágenes que se encarga de revelar los cambios de intensidad lumínica. En el análisis de imágenes el procedimiento es utilizando diferentes filtros digitales tanto isométrico como asimétrico. Esta técnica en especial es utilizada para el reconocimiento de objetos, detección de profundidad y tamaño. Existen varios tipos de filtros digitales dependiendo del método matemático empleado, tales como primera y segunda derivada, el gradiente y el laplaciano.

La efectividad radica en el método matemático que se emplea, así como la tolerancia que se selecciona.

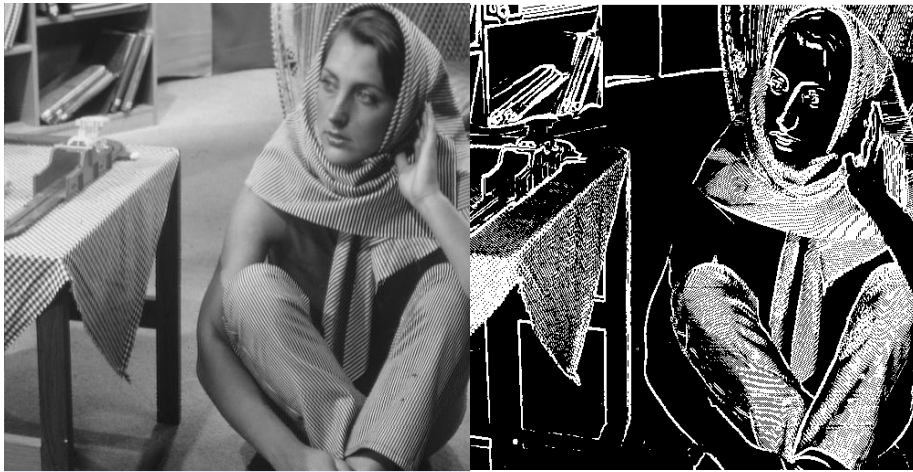


Figura 4.- Imagen de muestra Barbara.tif (izquierda) e imagen de muestra aplicada filtro de Diferencias Vertical con tolerancia=0.1 (derecha)
Fuente: Creación propia

Filtro Gradiente

Es un filtro digital utilizado con mayor frecuencia para la detección de bordes, que emplea como método matemático la primera derivada de la función. Considerando que la imagen digital es una función bidimensional, se emplea el gradiente como se muestra en la ecuación (6).

$$\nabla(f(x, y)) = \frac{\partial f(x, y)}{\partial x} \mathbf{i} + \frac{\partial f(x, y)}{\partial y} \mathbf{j} \quad (6)$$

El algoritmo práctico para su implementación maneja los filtros Horizontales y Verticales (ver Figura 4) con diferencias centradas por separado, para luego utilizarlos en cálculo de la magnitud del gradiente. Entre los operadores que manejan este tipo de filtros se encuentran Prewitt, que tiene mejoras en presencia de ruido y Sobel que posee mayor sensibilidad en los bordes diagonales. La Tabla 1 resume las matrices de los filtros de diferencias centradas, filtro Prewitt y Sobel.

Tabla 1.- Resumen matrices de filtros gradiente Horizontal, Vertical Prewitt y Sobel

Filtro de Diferencias Centradas	Matriz
Filtro Horizontal	$H = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Filtro Vertical	$V = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
Filtro Prewitt	$H = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ $V = \frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
Filtro Sobel	$H = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ $V = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$

Fuente: Creación propia con información suministrados por

Transformación Euclidiana

Las transformaciones geométricas permiten modificar parámetros de la imagen como área, longitud y ángulo. Entre estas se encuentra la transformada euclidiana o isométrica, que considera distancias iguales tanto en la imagen origen como en la transformada. Es decir la distancia entre dos puntos de la imagen original, se mantiene

igual que en la transformada, considerando modificaciones en translación, rotación y reflexión. La ecuación (7) indica la matriz H de la transformada. (Mery, 2004)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \epsilon \cos(\theta) & -\sin(\theta) & t_x \\ \epsilon \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (7)$$

Donde θ muestra el ángulo de rotación, t_x y t_y las coordenadas de traslación y el coeficiente ϵ indica reflexión si es igual a 1. En la Figura 5 se muestra la transformación isométrica en dos dimensiones.

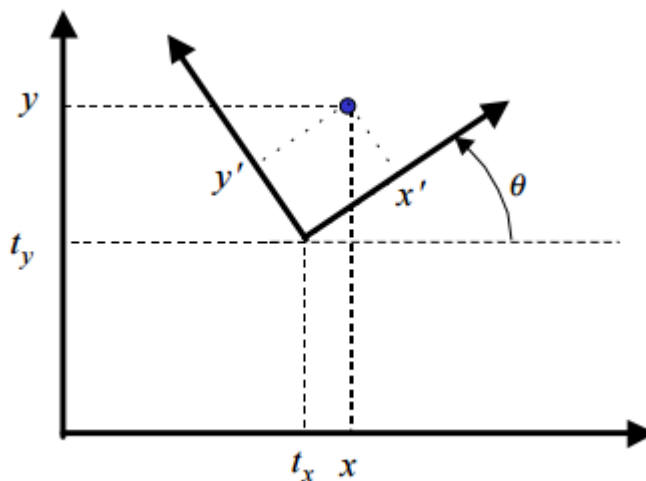


Figura 5.- Transformación Isométrica
Fuente: (Mery, 2004)

Component Labeling

Se utiliza en el procesamiento de imágenes para categorizar y encontrar las regiones que componen una imagen digital. El mecanismo se basa en categorizar las regiones de una imagen, dependiendo de la relación con sus vecinos. Este algoritmo denominado “Algoritmo secuencial” localiza una región específica de la imagen y asigna una etiqueta. De acuerdo a sus proporciones de tamaño, orden de localización y

condiciones de los vecindarios, se establece un valor numérico. (Ramesh Jain, 1995) En la Figura 6 se observa el proceso.

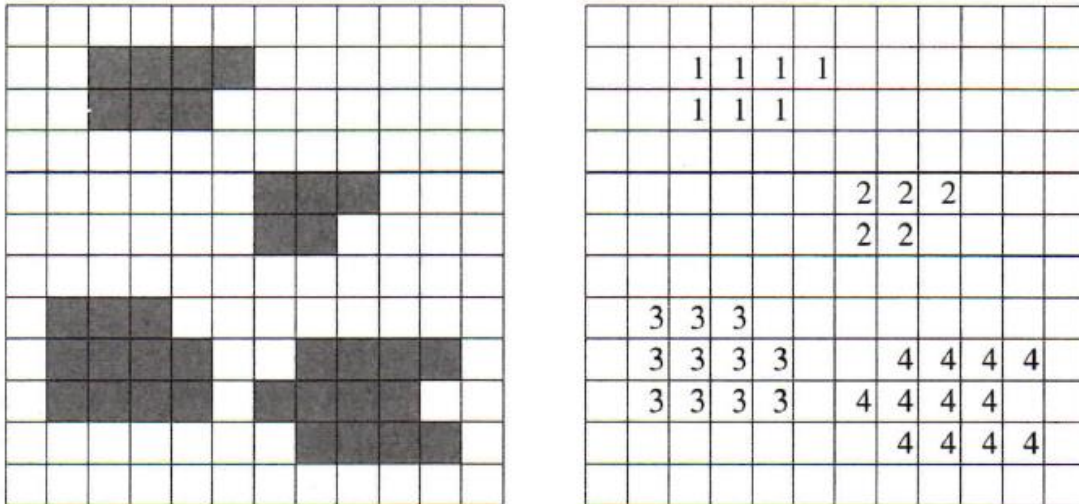


Figura 6.- Component Labeling
Fuente: (Ramesh Jain, 1995)

Entre las utilidades de este mecanismo se encuentra la eliminación de ruido dentro de una imagen. Utilizando un umbral de tolerancia (Thresholding) se define las etiquetas de menor densidad en la foto y se las detecta para ser eliminadas como se observa en la Figura 7.

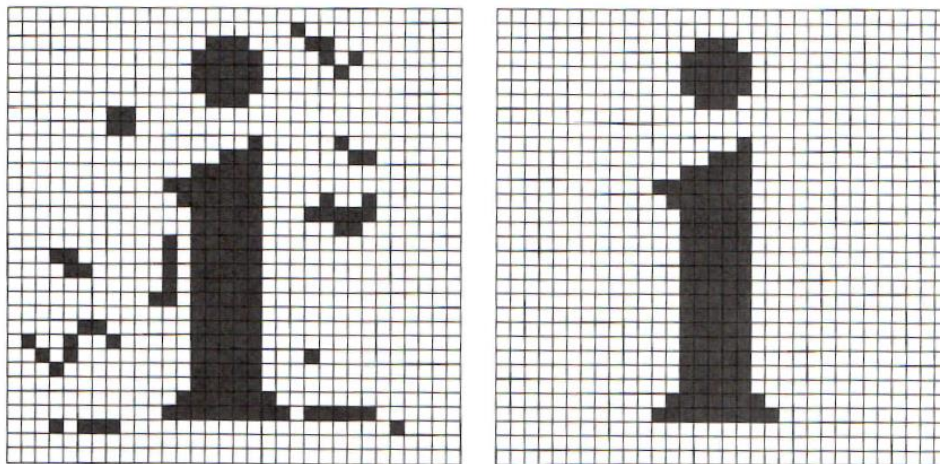


Figura 7.- Thresholding para eliminación de ruido
Fuente: (Ramesh Jain, 1995)

Expansión de la Imagen

Entre las herramientas de binarización que se utiliza, está la expansión de una sección de la imagen. Una vez realizado el thresholding, el proceso es más óptimo debido a que las partes con mayor densidad de intensidad son más reconocibles. La expansión consiste en cambiar un pixel de un nivel bajo a un nivel alto (0-255) si cualquier vecino se encuentra en un nivel alto; es decir si el vecino es etiquetado como zona relevante de la imagen (Ramesh Jain, 1995). El ejemplo se muestra en la Figura 8.

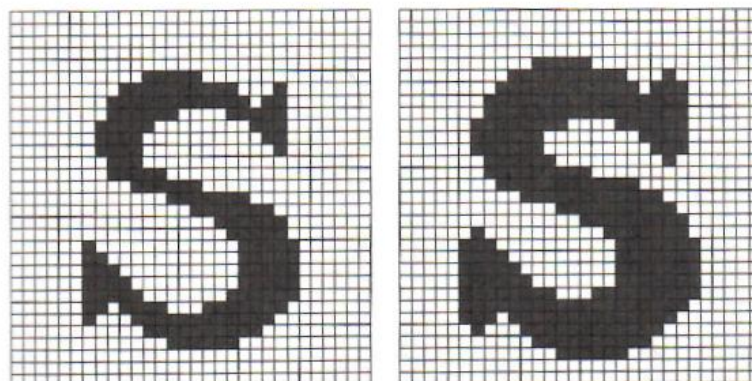


Figura 8.- Expansión de imágenes
Fuente: (Ramesh Jain, 1995)

Correlación y Autocorrelación

La correlación de dos variables estadísticas, es el método para determinar el grado de similitud y de relación que tiene al cambio simultáneo en ellas. Además informa cuan proporcionales son los datos de las variables A con respecto a B y viceversa. Esta información es utilizada para el procesamiento de imágenes y modelos estadísticos. (Vinay K. Ingle, 2012) La correlación se describe como la ecuación (8).

$$\phi_{xy}(\mathbf{t}) = \int_{-\infty}^{\infty} \mathbf{x}(\mathbf{t} + \boldsymbol{\tau})\mathbf{y}(\boldsymbol{\tau})d\boldsymbol{\tau} \quad (8)$$

Donde $y(t)$ y $x(t)$ son las funciones reales y $\phi_{xy}(t)$ la correlación. De la misma forma es posible reemplazar de la ecuación (8), $y(t)$ por $x(t)$ obteniendo la correlación de la misma función, lo que se denomina Autocorrelación que se describe en la ecuación (9).

$$\phi_{xx}(\mathbf{t}) = \int_{-\infty}^{\infty} \mathbf{x}(\mathbf{t} + \boldsymbol{\tau})\mathbf{x}(\boldsymbol{\tau})d\boldsymbol{\tau} \quad (9)$$

Tanto la Correlación como la Autocorrelación poseen propiedades de simetría. Es así que $\phi_{xx}(t) = \phi_{yy}(t)$ siendo funciones pares y $\phi_{xy}(t) = \phi_{yx}(-t)$ siendo impares.

(Alan V. Oppenheim, 1998)

Hardware de procesamiento de imágenes.

FPGA

Son chips de silicio reprogramables. Están constituidos de bloques digitales pre construidos que permiten al usuario cambiar la configuración de los mismos, de manera que pueda implementar distintas funciones personalizadas, sin necesidad de manipular el hardware de forma física. Se desarrolla el software compilado en un archivo de configuración, el que contiene información de cómo se deben conectar y comportar los distintos bloques del FPGA. En la actualidad se ha facilitado mucho la programación de estos dispositivos ya que se cuenta con compiladores que permiten transformar lenguajes de alto nivel como C o C++ e interfaces gráficas en programas de configuración para los FPGA's.

La adopción de chips FPGA en la industria ha sido impulsada por el hecho de que los FPGAs combinan lo mejor de los ASICs y de los sistemas basados en procesadores. Ofrecen velocidades temporizadas por hardware y fiabilidad, pero sin requerir altos

volúmenes de recursos para compensar el gran gasto que genera un diseño personalizado de ASIC. El silicio reprogramable tiene la misma capacidad de ajustarse en un software que se ejecuta en un sistema basado en procesadores, pero no está limitado por el número de núcleos disponibles.

A diferencia de los procesadores, los FPGAs llevan a cabo diferentes operaciones de manera paralela, por lo que éstas no necesitan competir por los mismos recursos. Cada tarea de procesos independientes se asigna a una sección dedicada del chip, y puede ejecutarse de manera autónoma sin ser afectada por otros bloques de lógica. Como resultado, el rendimiento de una parte de la aplicación no se ve afectado cuando se agregan otros procesos. (Simpson, 2010)

National Instruments MyRIO 1900

Es un hardware desarrollado por National Instrument que se basa en la tecnología RIO de NI con I/O reconfigurable capaz de funcionar paralelamente. MyRIO utiliza un chip Zynq de Xilinx que posee un procesador ARM dual-core y un FPGA que asegura una alta cantidad de procesamiento para el análisis de datos de entradas análogas repartidas en todo el hardware. Además el FPGA posee diferentes facilidades de programación y configuración de acuerdo a las necesidades requeridas (National Instrument, 2014) Posee 40 I/O digitales, 10 entradas y 6 salidas analógicas que permiten controlar y procesar gran cantidad de entradas como sean posibles. El sistema de programación es con LabVIEW que se adapta fácilmente a diferentes niveles de programación. (National Instrument, 2014). Además el hardware es lo más importante ya que el entorno gráfico y la gran elección de herramientas pre desarrolladas permiten un rápido desarrollo de aplicaciones.

Lenguajes de descripción de hardware HDL.

VHDL

VHDL para el acrónimo de “Very High speed integrated circuit hardware Description Language” es el lenguaje estándar industrial que describe circuitos digitales normado por IEEE 1164. En un inicio fue un lenguaje desarrollado por el Departamento de Defensa del Gobierno de los Estados Unidos para describir la estructura y el funcionamiento del Hardware. Actualmente, es utilizado para sintetizar, documentar y simular circuitos internos y para programar FPGA’s. El uso de la herramienta de compilación de VHDL convierte el código fuente en un circuito lógico tomando en cuenta a los BITS como los objetos de datos principales de este lenguaje. (Stephen Brown, 2006)

Como en todo lenguaje de programación, los primeros pasos son declarar las variables de entrada y salida que se denominan “declaración de entidad” así como las arquitecturas. Del mismo modo, es importante la utilización de Bibliotecas al momento que el código demande mayor cantidad de funciones específicas. (David Money Harris, 2013)

LabVIEW

Es una plataforma de programación gráfica de la compañía National Instruments desarrollada en 1986. Su utilización va desde procesos industriales, proyectos académicos y de investigación. La mayor parte de su programación fue desarrollada en lenguaje G, C y C++, lo que permitió una programación gráfica en diagramas de bloques convirtiéndolo en lenguaje de máquina. (National Instruments , 2015) Inicialmente fue elaborada para correr en sistema operativo Mac por su alta demanda gráfica como lenguaje de máquina, pero

actualmente está disponible para la mayoría de sistemas operativos del mercado. Su interfaz gráfica ayuda a la manipulación de sistemas complejos y consta con un sistema flexible a los cambios que requieren proyectos de ingeniería altamente variantes.

Esta plataforma cuenta con bibliotecas que facilitan la adquisición de datos, análisis y presentación de resultados. Además LabVIEW soporta simulaciones y ejecutables en Instrumentos Virtuales. La estructura principal de este lenguaje se basa en tres elementos: el panel frontal, el diagrama de bloques y los íconos. (Jeffrey Travis, 2006)

Transmisión de datos

Protocolo TCP-IP

El protocolo TCP/IP es un conjunto de dos protocolos de conexión de redes, el uno es Transmission Control Protocol (TCP) y el segundo Internet Protocol (IP). El objetivo de utilizar este protocolo es que puede conectar ordenadores de diferentes tipos que se encuentren a grandes distancias geográficas. Los diferentes tipos de ordenadores que este protocolo puede conectar son: PCs, minis y mainframes que ejecutan sistemas operativos diferentes sobre redes de área local y de área extensa o corporativa (G. Bermudez, 2003). Escrito de otra manera, el objetivo del protocolo TCP/IP es asignar una única dirección IP a cada ordenador de la red, y de esta manera se logra intercambiar los datos entre los mismos. Este protocolo TCP/IP tiene un buen funcionamiento sin importar el medio, es decir que puede ser una conexión ADSL, una red Ethernet o una fibra óptica. Además los estándares de protocolo TCP/IP son desarrollados independientemente del hardware de los ordenadores o del sistema operativo.

IEEE 802.11

La especificación IEEE 802.11 define las características de una red de área local inalámbrica (WLAN). Generalmente este estándar va de la mano con la certificación Wi-Fi que garantiza que todos los dispositivos 802.11 utilizados sean compatibles entre sí.

Con este patrón, se puede crear redes de área local inalámbrica, que permiten utilizar computadores portátiles, de escritorio, Smartphone entre otras, ya que la cobertura de la red varía entre 50 a 200 metros dependiendo del equipo y del ambiente en el que se desenvuelva. Debido a la facilidad de la implementación de una red Wi-Fi y la gran cantidad de usuarios que pueden conectarse a las mismas, este tipo de redes se están popularizando. (Cooklev, 2004)

Resumen del Capítulo

La información presentada en este capítulo constituye la base teórica que sustenta el proyecto. Se mostró los avances relacionados a la efectividad de los procesos de las ciudades inteligentes probados en diferentes ciudades de Europa y Estados Unidos. De igual manera, se presentan los adelantos sobre el procesamiento de imágenes hasta la actualidad, y la implementación de esta herramienta para los parqueaderos inteligentes. Además se explica como a través del modelo Basic Pinhole se interpreta un espacio 3D en un plano 2D. Por otro lado se muestra también el funcionamiento de los filtros para detección de bordes y transformaciones geométricas manipulando la matriz de cámara.

En cuanto al Hardware y al lenguaje de programación, se presentó las posibles alternativas más viables para el desarrollo del proyecto. Entre las que se destacan NI

MyRIO 1900 por su alto desempeño y LabVIEW por su estructura basada en gráficos que generan el lenguaje de máquina. Por último se presentan los protocolos de comunicación Ethernet y WiFi que fundamentan la comunicación tanto del hardware a los servidores como hacia la interfaz del cliente.

METODOLOGÍA Y DISEÑO

Justificación de la metodología seleccionada

La metodología a utilizarse en el prototipo de parqueaderos inteligentes en la zona azul del Distrito Metropolitano de Quito, es el análisis cuantitativo de resultados a través de simulaciones, pruebas en campo y experimentos que permitan obtener un mejor resultado en cada una de las actividades. Además recurriendo a la información obtenida en la revisión bibliográfica y en el marco teórico, se crean diferentes algoritmos y mecanismos de realización que fundamentan los resultados.

Adquisición de Imágenes

Requerimientos del proyecto

Antes de escoger el tipo y modelo de cámara que se va a utilizar para la determinación de si un parqueadero está ocupado o no, se fijó las especificaciones mínimas que debe cumplir la cámara para este propósito, de manera que se optimice los costos sin afectar la calidad del sistema y su estabilidad dando un soporte robusto al dispositivo. La cámara a utilizar debe capturar las imágenes en escala de grises, puesto que no es necesario establecer los colores de la imagen sino encontrar las formas que la componen. Esto se debe a que la comparación de imágenes para determinar el cambio de estado en un parqueadero, los colores no incide mayormente, de hecho solo complican el algoritmo. Sin embargo cuando la cámara adquiere imágenes a color se puede procesar la imagen en escala de grises.

De la misma manera se determinó que la cámara no debe ser de alta definición ya que, a mayor definición mayor capacidad de memoria ocupa la imagen al guardarla y más

grande la capacidad computacional para procesarla. Además si la cámara tiene demasiada resolución al momento de adquirir las imágenes de los parqueaderos sobre las calles, se puede observar con mayor detalle orificios y defectos sobre la calzada, parámetros que no interesan en el proyecto.

Selección de la Cámara

Después de determinar que la cámara no necesariamente tiene que capturar las imágenes a color ni debe ser de alta definición, se fija las especificaciones de la cámara dadas por el dispositivo que en este caso estará conectada al MyRIO 1900 de National Instruments. Este proceso de conexión tiene sus dificultades, por lo que se necesita conocer el tipo de interfaces con las que se cuenta para enlazar la cámara. Para este caso las dos interfaces principales son WiFi que permite conectar el dispositivo a una red donde se puede alcanzar una cámara IP, y otra utilizando el puerto USB con el que se accede a la cámara web.

Debido a que el costo de una cámara web es menor al de una cámara IP y en el mercado local se consigue fácilmente con todas las especificaciones del proyecto, se decide utilizar una cámara web que se conecte a una interfaz USB. Una vez escogido el tipo de dispositivo a utilizar fue necesario elegir el modelo, para lo cual se investigó si el MyRIO trabaja con algún tipo específico de cámaras web acudiendo al manual del dispositivo en el que simplemente dice que las cámaras deben cumplir con el protocolo UVC (USB video device), la mayoría de cámaras web cumplen con este protocolo. De este modo se escogió dos modelos de cámaras Logitech C160 y Genius facecam 1000x, las que se muestran en la Figura 9.



Figura 9.-Cámara Genius facecam 1000x (izq) y Cámara Logitec C160 (der)

Conexión y Configuración

Para abrir la cámara en el dispositivo se conecta en el puerto USB del MyRIO como se muestra en la Figura 10.



Figura 10.- Conexión de la cámara al puerto USB de NI MyRIO 1900

Conectada se procede a programar el dispositivo para que abra la cámara y adquiera las imágenes de la misma (ver Figura 10).

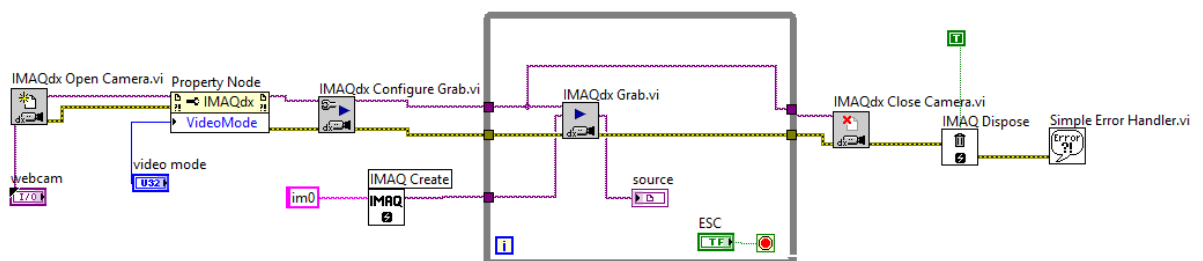


Figura 11.- Programa para abrir la cámara

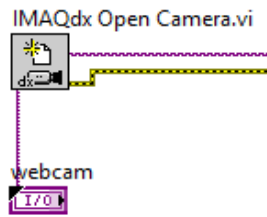


Figura 12.- Módulo IMAQdx Open Camera.vi

El programa está formado por 7 componentes los cuales se explican a lo largo del capítulo. Este componente (ver Figura 12) le permite al dispositivo acceder a su puerto USB y cámara web conectada al mismo, sin embargo es necesario especificar a qué cámara se desea acceder, por lo que cada vez que se conecte una nueva cámara se asigna un nombre como cam0 o cam1 y con esto abre una sesión donde se accede a la información de la cámara.

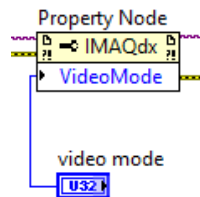


Figura 13.- Módulo Propiedades de Node

En la Figura 13 se muestra el componente que permite al dispositivo manejar la información que recibe de la cámara como una imagen, para ello necesita recibir los datos de la sesión creada al abrir la cámara.

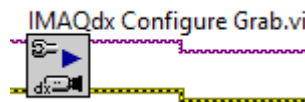


Figura 14.- Módulo IMAQdx Configure Grab.vi

El componente IMAQdx Configure Grab.vi que se muestra en la Figura 14, le permite al dispositivo guardar las imágenes una a continuación de otra en forma de video, también necesita la información de la sesión creada al iniciar la cámara.

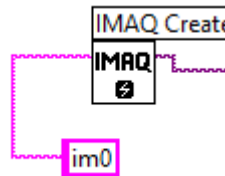


Figura 15.- Módulo IMAQ Create

Este componente (ver Figura 15) crea un espacio de memoria en el dispositivo, guardando las imágenes que va adquiriendo la cámara conectada.

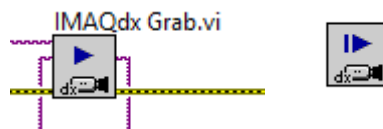


Figura 16.- Módulo IMAQdx Grab.vi

Los dispositivos de la Figura 16 permiten, grabar las imágenes continuamente en forma de video, y guardar las imágenes una por una mostrando a su salida. Ambas necesitan recibir la información del inicio de sesión con la cámara y la dirección del espacio de memoria creado anteriormente.

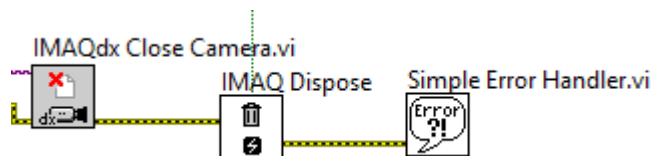


Figura 17.- Módulo IMAdx Close Camera.vi (izq) Módulo Imaq Dispose (der)

Finalmente los componentes que se muestran en la Figura 17, permiten cerrar la sesión de la cámara y eliminar los espacios de memoria creados para las imágenes adquiridas por la misma.

Por otro lado existe otra manera distinta de abrir la cámara en forma sencilla y confiable. Esto es, utilizando el asistente de Vision Acquisition Express (ver Figura 18) que permite acceder con mayor facilidad a la cámara y tener más control sobre la misma.

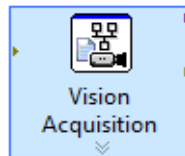


Figura 18.- Módulo Vision Acquisition Express

Este módulo permite primero elegir la cámara que se utilizará a lo largo del proceso. En ella se despliega una lista de cámaras disponibles que se encuentran directamente conectadas al MyRIO. Dependiendo de cuantas cámaras se hayan conectado, cada una tendrá un nombre del tipo cam # como se muestra en la Figura 19

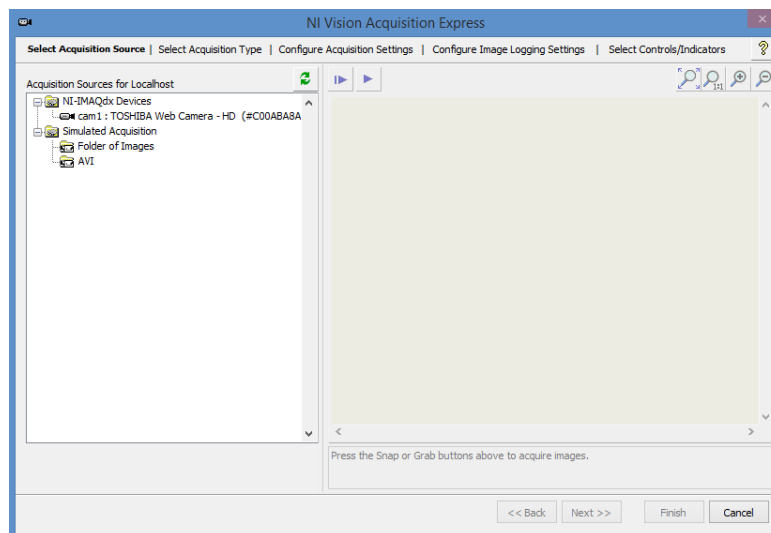


Figura 19.- Selección de cámara usando Vision Acquisition Express

Luego se elige el tipo de adquisición entre: procesar adquiriendo una sola o varias imágenes, o adquirir un video continuo o varias imágenes y procesarles en tiempo real. El resumen de las alternativas se muestra en la Figura 20.

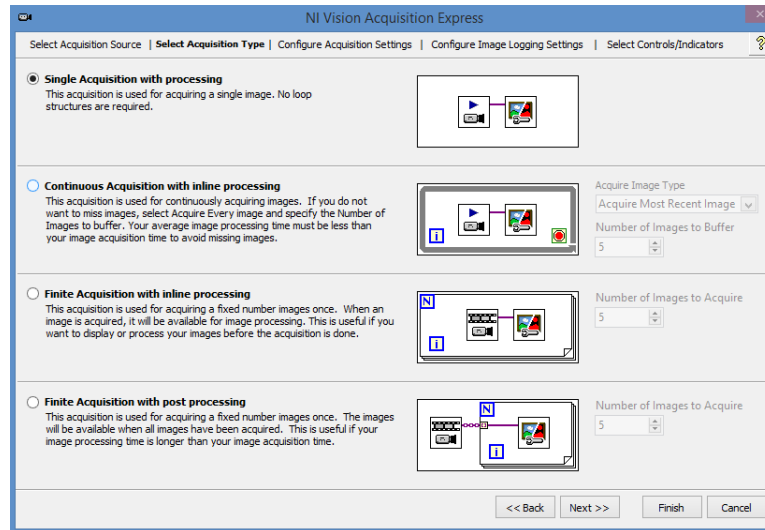


Figura 20.- Alternativas de adquisición de imágenes en módulo Express

Al elegir el tipo de procesamiento, se genera automáticamente el entorno en el que se trabaja, ya sea lazo while, lazo for o ninguno. Por último se elige la resolución y velocidad de captura de la cámara, en este caso es 640x480 10 fps como se muestra en la Figura 21.

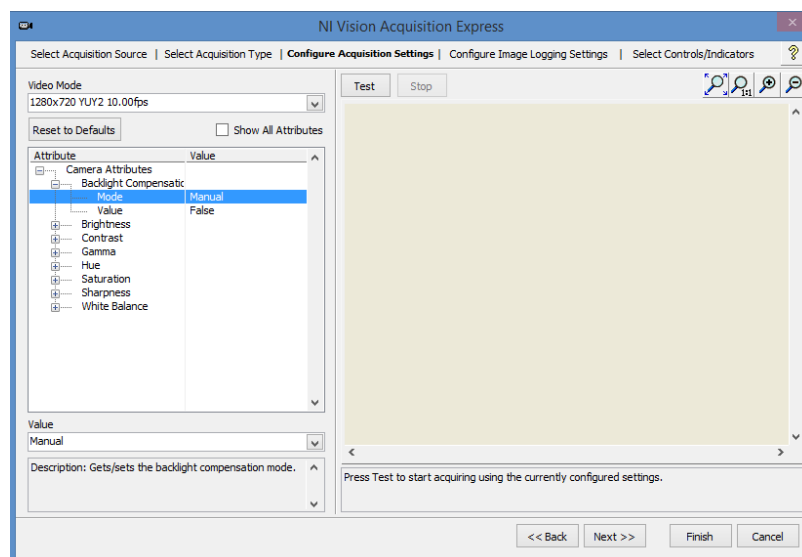


Figura 21.- Parámetros de Adquisición

Almacenamiento de Imagen

Ya que el espacio de memoria creado para las imágenes será reutilizado cada vez que se adquiriera una nueva imagen, es necesario guardar el ícono en otro lugar, de manera que más adelante permita acceder a la misma y se pueda procesar, segmentar y comparar para determinar el estado actual del parqueadero, para lo cual se guarda la imagen en forma de arreglo de dos dimensiones como se muestra en la Figura 22.

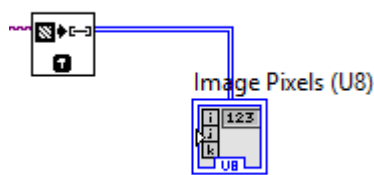


Figura 22.- Módulo Transformación Imagen a arreglo

Este proceso toma la imagen obtenida por la cámara y la transforma en un arreglo de números que guarda la información de la imagen capturada.

Sistema de Movimiento Rotacional

Especificaciones de servomotor

Con el fin de maximizar el número de parqueaderos que la cámara puede analizar, se decide utilizar un servomotor para que, a través de un movimiento rotacional logre capturar la imagen de un mayor número de parqueaderos.

Existen 3 tipos de servomotores disponibles en el mercado AC, DC y de imanes permanentes (Firoozian, 2008). Debido a que MyRIO cuenta con salidas de alimentación DC se decide trabajar con un servomotor DC.

Los servomotores DC están compuestos por un motor de corriente continua que produce movimiento en el servo al aplicar voltaje entre sus dos terminales, engranajes reductores que convierten la velocidad de giro en torque y circuito del control que se

salida PWM0 en el pin 27 donde se conecta el cable de control. El negativo se conecta al pin 6 que corresponde a la tierra analógica y el cable de alimentación en el pin 1 de +5V como se indica en la Figura 24.

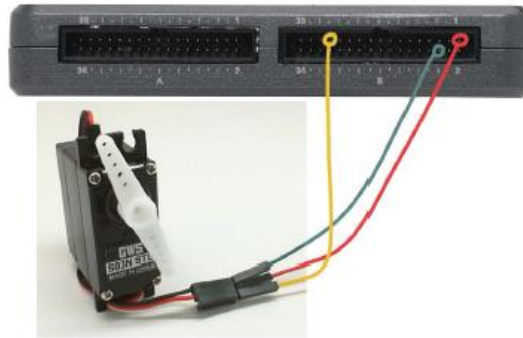


Figura 24.- Conexión del Servomotor

Una vez conectado el servo al dispositivo, se crea en LabVIEW el programa que permita al MyRIO controlar el movimiento del servo. El programa se muestra en la Figura 25.

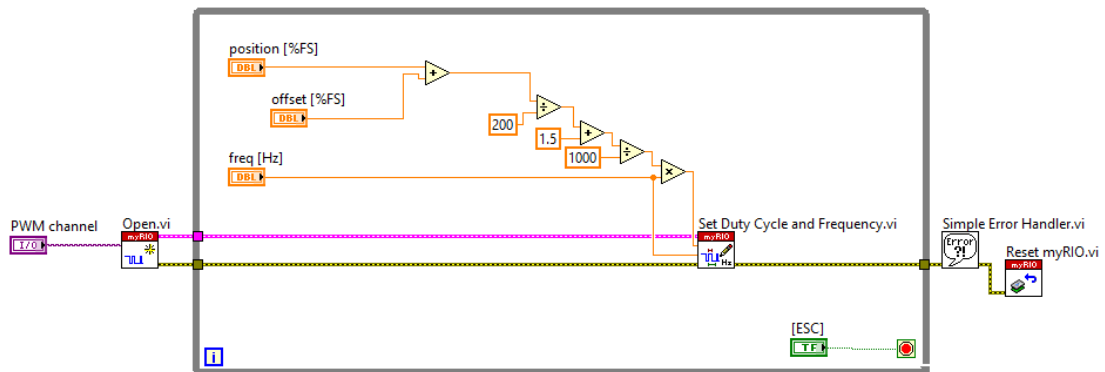


Figura 25.- Programa para controlar Servomotor desde Labview 2014

Cada módulo del programa tiene su uso específico como se indica en las siguientes Figuras. Se inicializa el MyRIO para generar señales PWM a través de un canal específico, en nuestro caso PWM0 B como muestra la Figura 26.

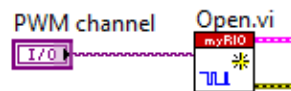


Figura 26.- Módulo de Apertura del canal PWM

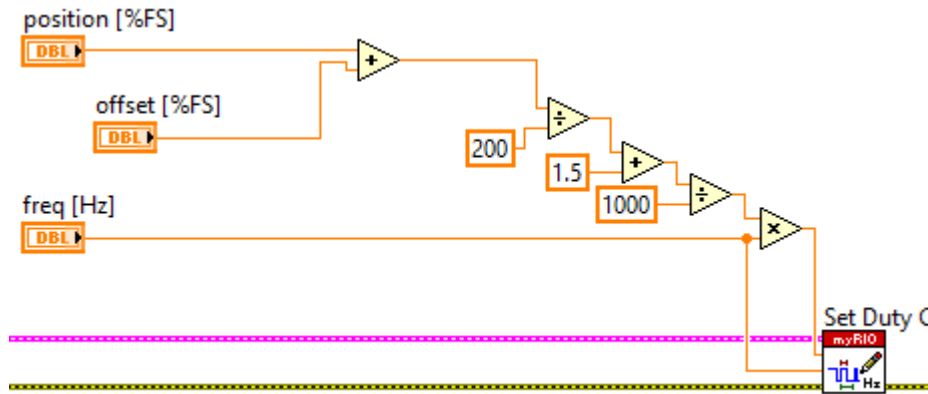


Figura 27.- Modulo de generación de señal PWM

El módulo de la Figura 27 se encarga de generar la señal PWM que controlará el movimiento del servomotor, para lo cual necesita el ciclo de trabajo y la frecuencia en Hz, que en este caso es 50 [Hz].

$$\text{PWM} = \frac{\left(\frac{\text{posicion deseada}}{200} + 1.5\right)}{1000} * \text{frecuencia} \quad (10)$$

Para determinar el ciclo de trabajo se debe tomar en cuenta el ancho del pulso. Si es de 1,5 ms, el servo se moverá a una posición vertical 0° , si queremos mover un ángulo de -90° será de 0.5 ms y si queremos mover $+90^\circ$ el ancho del pulso estará dado por 2.5 ms (Firoozian, 2008). Para determinar el ciclo de trabajo la fórmula está dada en la ecuación (10), la que utiliza el programa de la Figura 27. Se finaliza con el módulo de la Figura 28 que resetea todas las salidas del dispositivo.



Figura 28.- Módulo Reset myRIO.vi

Procesamiento de Imagen

Segmentación de Imagen

Una vez adquiridas las imágenes de los parqueaderos y para que se sea más fácil el corte de las mismas, se procesan antes de guardar en forma de arreglo. Al no utilizar toda la foto para la comparación, sino solamente la parte donde se encuentran ubicados los parqueaderos, se utiliza una serie de filtros y transformaciones geométricas que permiten observar el proceso de la imagen a segmentarla. A continuación se muestra en la Figura 29 la foto del parqueadero de inicio del proceso.

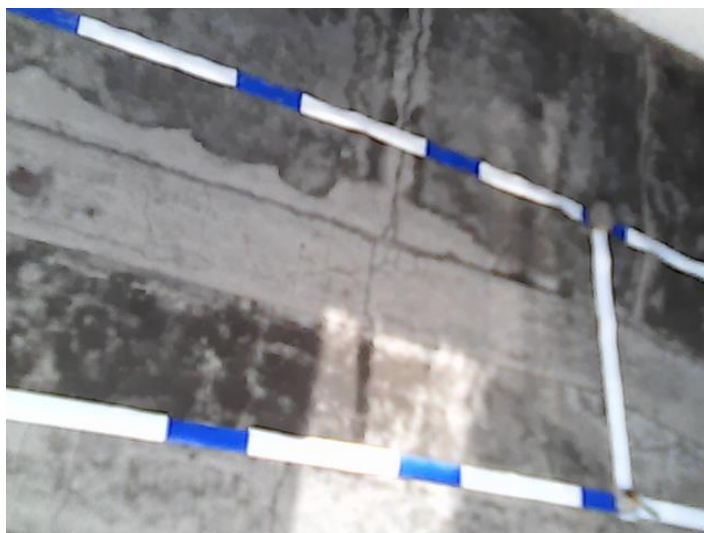


Figura 29.- Foto de parqueadero de inicio del proceso

Se utiliza un filtro que extrae el color azul de la imagen con dos objetivos, primero evitar que se diferencie entre el blanco y el azul de las líneas del parqueadero y parezca

una sola línea continua de un solo color. Segundo obtener una imagen en escala de grises que es más fácil manipular el resultado. (Ver Figura 30)

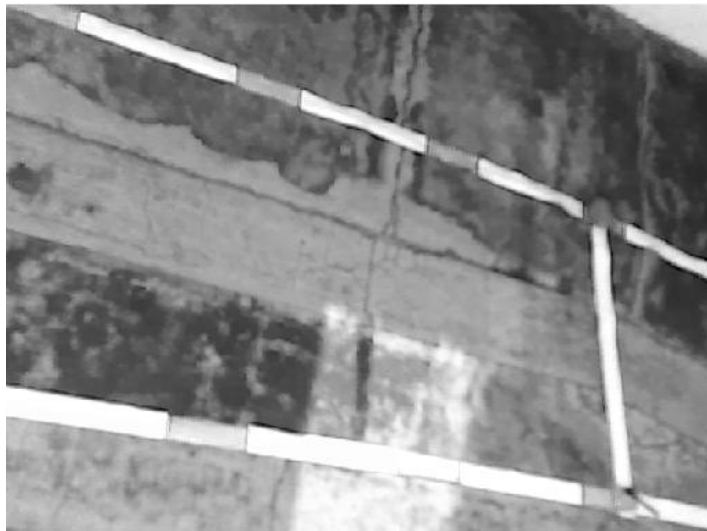


Figura 30.- Aplicación de filtros de extracción de color azul y escala de grises

Luego para detectar las líneas que dividen los parqueaderos tanto horizontal como vertical se utiliza un filtro de detección de bordes Prewitt.

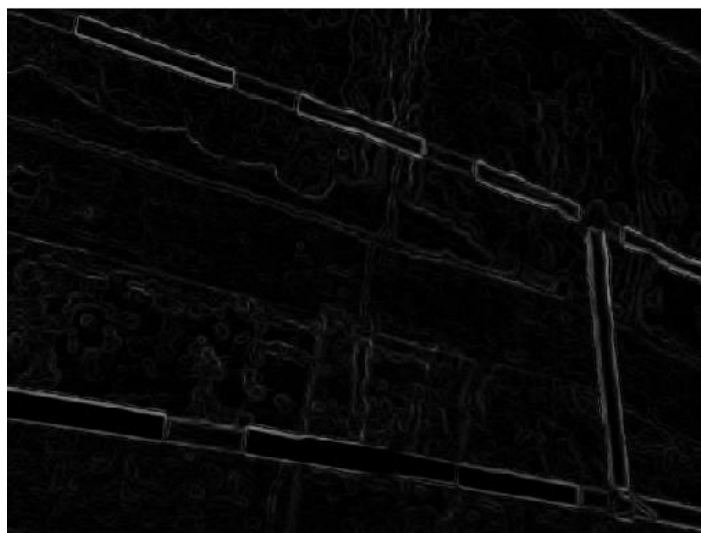


Figura 31.- Imagen con Detección de Bordes

En esta Figura 31 se observa claramente los bordes que dividen los parqueaderos. Sin embargo para mejorar aún más el resultado, se aplica una transformación geométrica que permite a las líneas horizontales que parecen estar inclinadas con respecto al borde de

la imagen, se encuentren rectas. Para ello se gira a la imagen como se muestra en la Figura 32.



Figura 32.- Imagen con Transformada Euclidiana

La imagen parece estar menos inclinada que antes, pero se nota algunos de los defectos de la calzada como grietas y huecos que al momento de hacer la segmentación de la imagen pueden generar problemas. Por lo que se aplica un valor de threshold que logra omitir los detalles que no son notorios y que para este caso no son útiles. Este valor se conoce como threshold de corrección de fondo y saca valores por secciones logrando corregir imperfecciones producidas por el fondo de la imagen. El resultado se muestra en la Figura 33

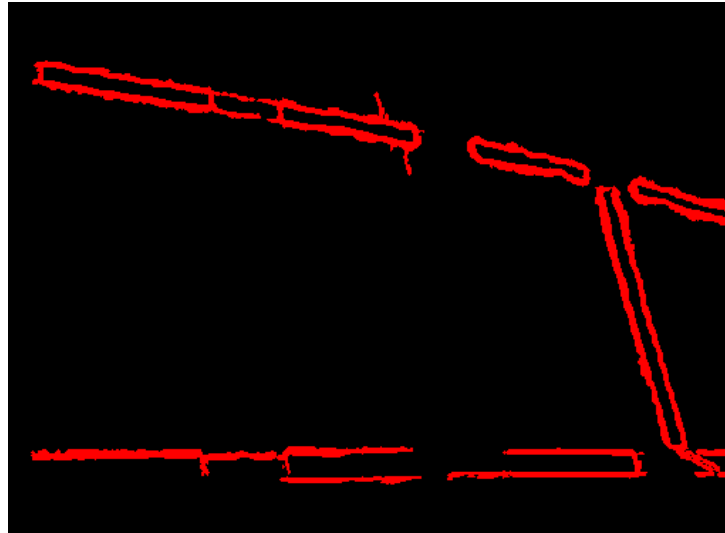


Figura 33.- Detección de bordes con threshold de corrección

Las líneas en rojo, para resaltar las partes de la detección de bordes que cumplieron con el threshold, claramente se observa como las imperfecciones del fondo de la imagen, producidas por el estado de la calzada se eliminan completamente. Sin embargo, se perdió parte de las líneas que eran muy tenues. Por último, para facilitar la detección de las mismas se dilata la imagen como muestra la Figura 34.

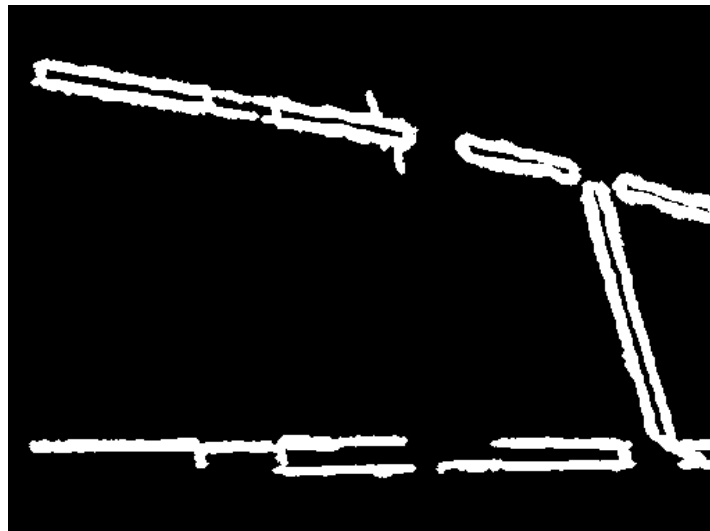


Figura 34.- Imagen con dilatación de líneas

El procedimiento de ensanchar dilatando la imagen, permite que las líneas de división de los parqueaderos sean nítidas. Las mismas transformaciones geométricas y

filtros se aplican a la otra imagen capturada por la cámara, y se las pone una a lado de otra como una foto panorámica (ver Figura 35).

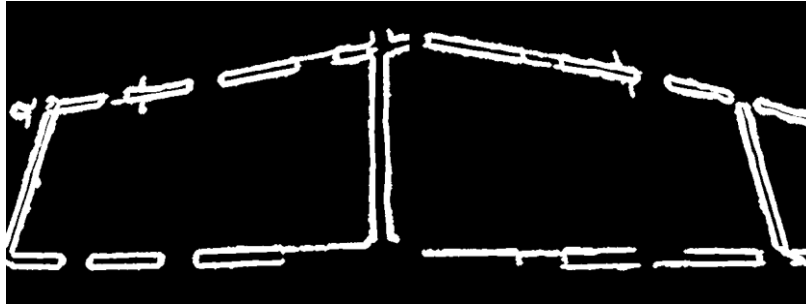


Figura 35.- Imagen Total

Una vez que se tiene la imagen completa, se guarda en forma de arreglo y se pasa a través de un nodo Mathscript en LabVIEW. Aquí la imagen se procesa y detecta el sitio de las líneas blancas, que delimitan la ubicación de los parqueaderos dentro de la foto, generando tres arreglos que definen alto, ancho y cantidad de parqueaderos que existen en la imagen.(Figura 36)

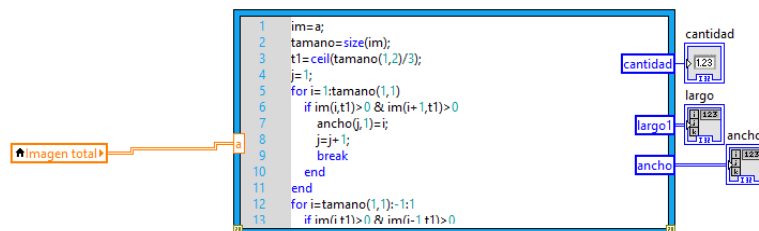


Figura 36.- Mathscript de corte de las imágenes

Con estos datos, se corta la imagen de manera que el momento de confrontar, solo se compare la parte de la imagen que se requiere.

Comparación de Imágenes

La comparación de imágenes parte de los datos obtenidos en el proceso de corte de la imagen, inmediatamente vuelve a capturar con la ayuda de la cámara y se procesa. Para empezar, se saca el plano de intensidad de la imagen obteniendo una nueva en escala de

grises que permite comparar de mejor manera. A continuación se gira la imagen igual que se lo hizo en la anterior fase, a fin de que las coordenadas que se obtuvieron en la imagen segmentada coincidan con las de esta fase. Por último se vuelve a poner las imágenes una a lado de otra, para que se pueda segmentar la imagen antes comparada (ver Figura 37).

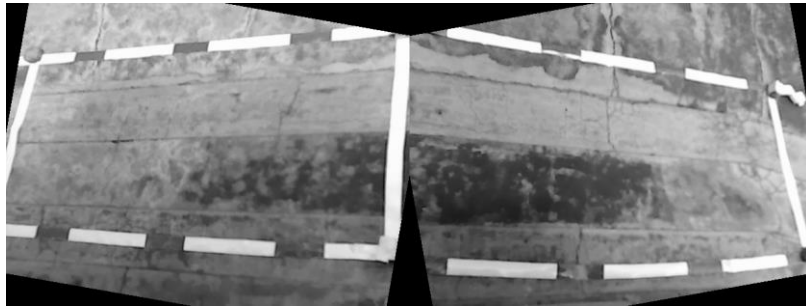


Figura 37.- Imagen completa fase comparación

Con este procedimiento, la imagen guardada se pasa en forma de arreglo, y a través de un nodo Mathscript en LabVIEW esta se corta en cada uno de los parqueaderos que se comparan. Luego se guarda a cada una de las imágenes en un arreglo de tres dimensiones, como indica la Figura 38.

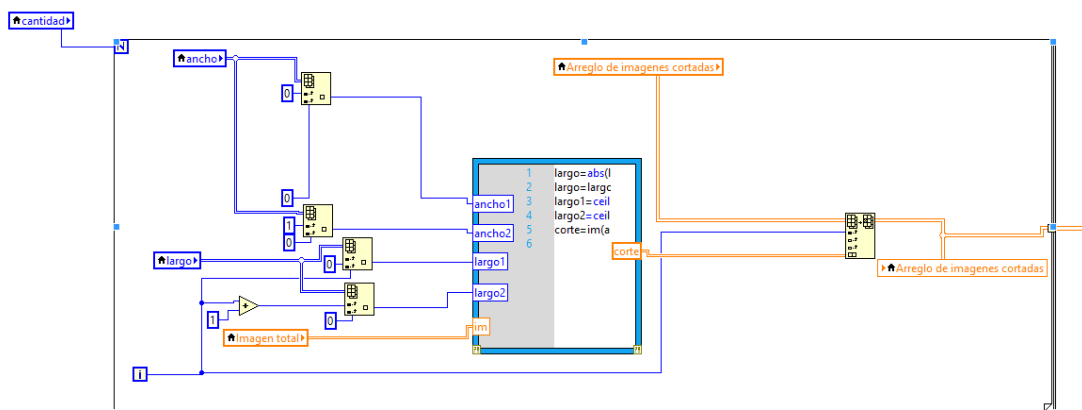


Figura 38.- Corte de imagen a comparar

Como resultado de este proceso se obtuvo dos tipos de tamaños de imagen, el primero con un recorte del 100% de la dimensión del parqueadero calculado (ver Figura 39) y el otro utilizando el 60% de la parte central de la imagen (ver Figura 40).

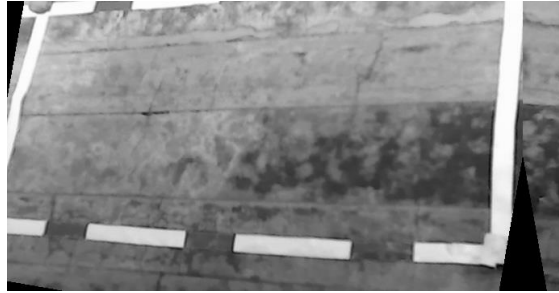


Figura 39.- Corte de imagen al 100%

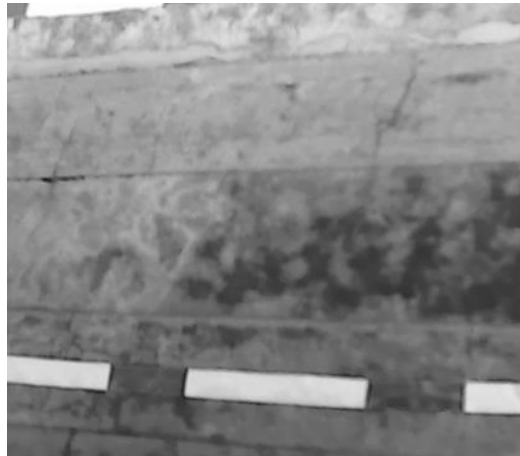


Figura 40.- Corte de imagen al 60%

Para concluir el proceso, cada imagen es restada de su anterior, del resultado se obtiene el valor absoluto y el valor promedio. De esta diferencia se establece cuánto cambia una imagen comparada con la otra. Si la comparación, supera el rango de tolerancia el estado del parqueadero tiene o no un cambio. Si hay variación, se produce un cambio de estado sea de vacío a ocupado o viceversa. Esto se hace mediante la siguiente lógica booleana donde 0 significa parqueadero vacío y 1 parqueadero ocupado en el campo “estado actual”. Por otro lado si el campo “cambio actual” está en 0, significa que en el parqueadero existe un cambio y en 1 que no (ver Tabla 2).

Tabla 2.- Tabla de verdad lógica de comparación

estado actual	cambio actual	nuevo estado
0	0	0
0	1	1
1	0	1
1	1	0

Fuente: Creación propia

Cuando el auto no alcance a estacionarse en el tiempo que se procesa un frame, se toman en cuenta el frame anterior para la lógica de comparación como se muestra en la

Tabla 3.

Tabla 3.- Tabla de Verdad lógica de comparación con estado anterior

estado actual	cambio actual	cambio anterior	nuevo estado
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Fuente: Creación propia

El sistema está diseñado para detectar cambios bruscos sobre el área del parqueadero es decir que si un peatón un animal cruza sobre la misma no se lo toma como un cambio debido a que la interferencia de los mismo no supera el valor de tolerancia necesario para detectar un cambio de estado en el parqueadero.

Descripción del Sistema de Comunicación e integración en red

Especificaciones de protocolo

El sistema de comunicación del proyecto, se basa en el protocolo TCP/IP que envía la información mediante paquetes a través de un inicio de sesión. En este caso, se realiza el envío de datos mediante el protocolo IEEE 802.11 que utiliza las tarjetas de red inalámbricas instaladas tanto en el Servidor como en MyRIO 1900. Los parámetros tomados en cuenta para el diseño de la red local son: el protocolo de comunicación que maneja la tarjeta NI MyRIO, la localización de los host, la calidad de servicio que requiere el proceso, la flexibilidad y costos.

En la topología de Red que se observa en la Figura 41, se presenta un Router inalámbrico marca TP-LINK con dirección MAC 00-25-86-CD-12-74 que permite enrutar los datos desde el MyRIO al servidor. Este funciona en las bandas de frecuencias de radio UHF de 2,4 GHz y SHF de 5 GHz y accede a la configuración de los modos 802.11g y 802.11b. En razón de no existir equipos conectados a la red, el servicio de DHCP que ofrece el Router se limita a 10 direcciones disponibles. La asignación de direcciones IP tanto en la tarjeta como en el servidor se colocan de forma estática como describe la Tabla 4.

Mediante la interfaz web de administración del Access Point, se crea una Red local inalámbrica con su Identificador de conjunto de Servicios Básicos (SSID) de nombre “MyRio”. Para proteger el acceso a la Red, se pide una autenticación por medio de una clave de acceso, configurada con una seguridad de tipo WPA2-PSK. Por último se especifica la versión del protocolo en 802.11g que transmite una velocidad máxima de

hasta 54Mb/s en frecuencia UHF así como en el canal 1. El diagrama de configuración se muestra en la Figura 41.

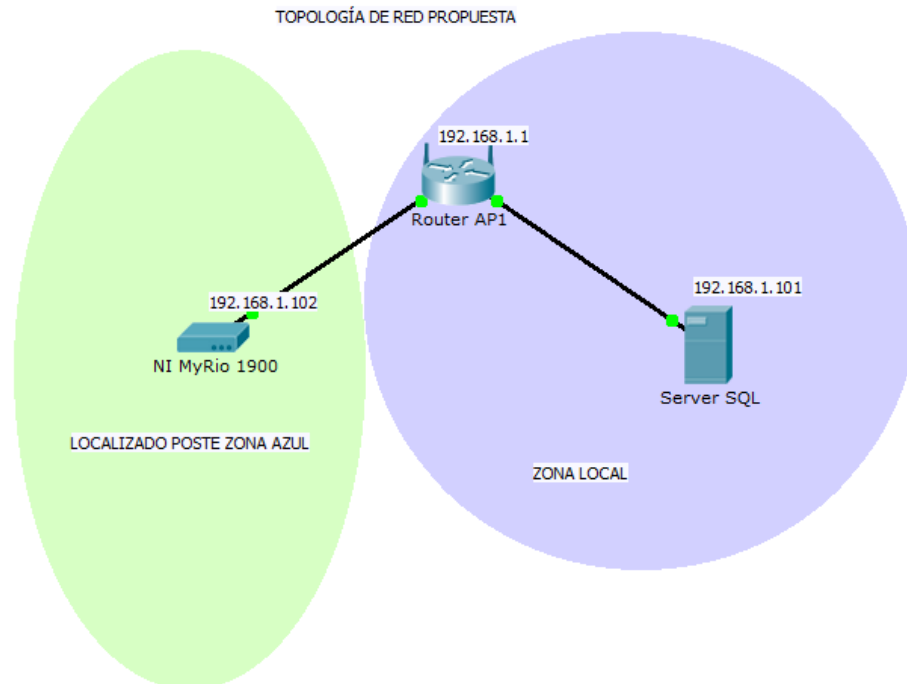


Figura 41.- Topología de Red Local
Fuente: Creación Propia

Una vez levantada la red, la tarjeta procesa las imágenes y genera una tabla de actualización del estado de cada parqueadero, que se transmite mediante un servidor TCP/IP programado con lenguaje G en LabVIEW. A su vez el servidor conectado a la red inalámbrica, levanta un servicio de cliente TCP/IP que en forma similar es programado en LabVIEW. Debido a que la red está configurada solo para servir al proyecto, y la comunicación no requiere mayor utilización de pórticos designados con el direccionamiento, se escoge el puerto 8080. Finalmente la Figura 42 demuestra la configuración de la Red mediante la interfaz web del AP.

Tabla 4.- Tabla de Direccionamiento IP Red “MyRIO”

Dispositivo	Dirección IP	Default Gateway
RouterAP1	192.168.1.1	N/A
Tarjeta NI MyRIO 1900	192.168.1.102	192.168.1.1
Servidor SQL	192.168.1.101	192.168.1.1

Fuente: Creación Propia en base a topología de Figura 41

Con fines demostrativos, los programas del servidor y cliente del protocolo TCP/IP no contienen todos los elementos utilizados en el proyecto final, sino los necesarios para realizar la comunicación.

54M
Wireless Router
Model No.: TL-WR340G
TL-WR340GD

- Status
- Basic Settings ---
- Quick Setup
- Network
- Wireless
 - Wireless Settings
 - MAC Filtering
 - Wireless Statistics
- Advanced Settings ---
- DHCP
- Forwarding
- Security
 - Static Routing
 - IP & MAC Binding
 - Dynamic DNS
- Maintenance ---
- System Tools

Wireless Settings

SSID:

Region:

Warning: Ensure you select a correct country to conform local law. Incorrect settings may cause interference.

Channel:

Mode:

Enable Wireless Router Radio

Enable SSID Broadcast

Enable Bridges

Enable Wireless Security

Security Type:

Security Option:

Encryption:

PSK Passphrase:

(The Passphrase is between 8 and 63 characters long)

Group Key Update Period: (in second, minimum is 30, 0 means no update)

Figura 42.- Interfaz Web de administración de Router para la creación de Red MyRio

Especificaciones y configuración de servidor

Mediante el diseño de red propuesto, la tarjeta NI MyRIO 1900 es configurada como servidor, siguiendo la topología y tabla de direccionamiento. Las características de la tarjeta se describen en la Tabla 19 en el apartado de Anexos. Además en la interfaz web del NI MyRio1900 se asigna la dirección IP en 192.168.1.102 con puerta de enlace como se observa en la Figura 43.

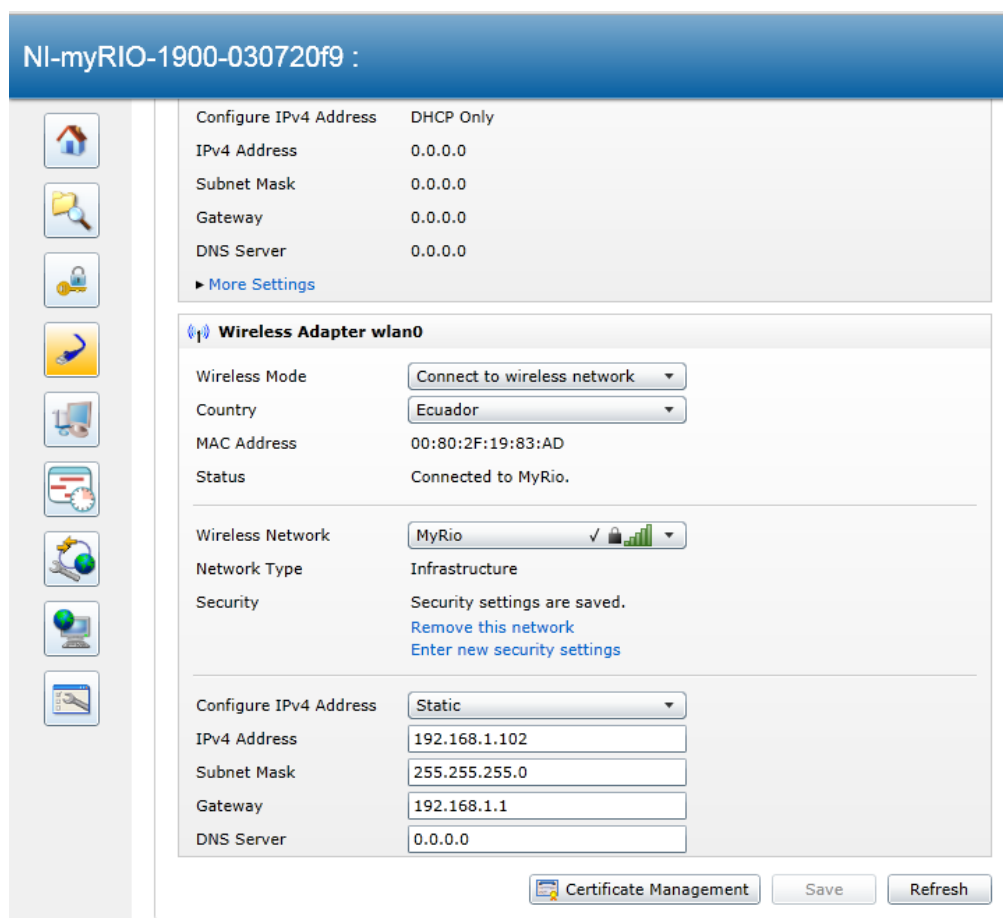


Figura 43.- Configuración direccionamiento estático NI MyRIO 1900

Para la configuración del servidor TCP/IP en la plataforma LabVIEW, es necesario diferentes objetos que se encuentran almacenados en la biblioteca de “Comunicación de Datos/Protocolos/TCP”. Estos sirven para iniciar la comunicación, escribir en el protocolo TCP y cerrarla. Es importante mencionar que tanto el servidor como el cliente son

dependientes el uno del otro, puesto que el cliente utiliza constantes del controlador donde se escribe en el servidor.

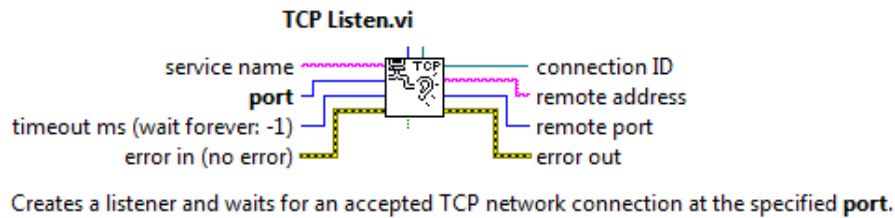


Figura 44.- Módulo TCP Listen.vi
Fuente: Context Help de Labview 2014

El objeto inicial que se muestra en la Figura 44 es “TCP Listen”. Este permite escuchar flujo de información a través del protocolo TCP/IP identificado por un nombre de servicio y puerto de salida. Luego se coloca la función “TCP Write” que se escribe sobre la sesión ya establecida. Como se observa en la Figura 45, entre las salidas de este objeto está el tamaño de bytes escritos, que arrojan datos importantes para conocer el estado del envío de información. Finalmente se cierra la comunicación con el objeto “TCP Close Connection” (ver Figura 45).

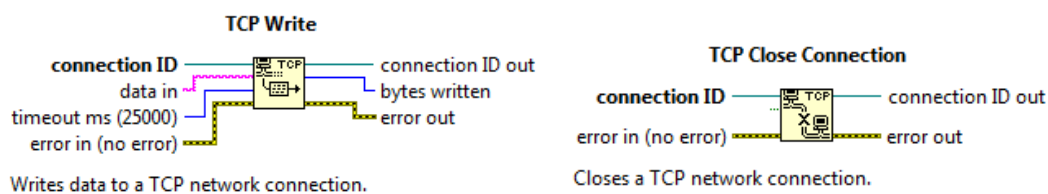


Figura 45.- Módulo TCP Write.vi (izq.) Módulo TCP Close Connctio.vi (der)

Fuente: Context Help de Labview 2014

En la Figura 46 se muestra el esquema completo de la programación del servidor. Este arranca con el inicio de sesión, detallando el nombre del servicio como “Servidor”, el puerto 8080 y el timeout en 60000 ms. Posteriormente se detalla el control y la

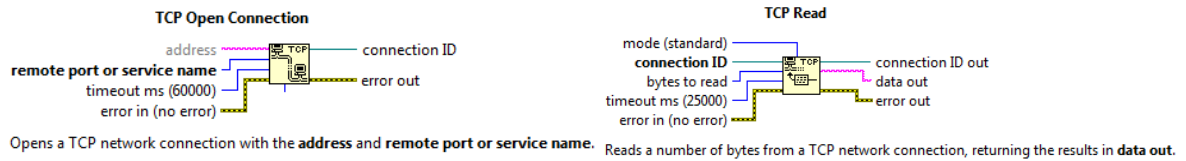


Figura 47.- Módulo TCP Open Connection.vi (izq.) Módulo TCP Read.vi (der)
Fuente: Context Help de Labview 2014

El cliente TCP/IP inicia su proceso con la apertura de la conexión, en donde se asigna la dirección del servidor en 192.168.1.102 con el puerto remoto 8080. Seguidamente en el objeto “TCP Read” se define la cantidad de bytes a leerse estableciendo los parámetros del proceso. Para luego, a través de un “Type Cast” detallada por la constante del control del servidor, logra recuperar la información enviada. El proceso finaliza con el cierre de la comunicación en el objeto “TCP Close Connection”. El esquema completo del cliente se muestra en la Figura 48.

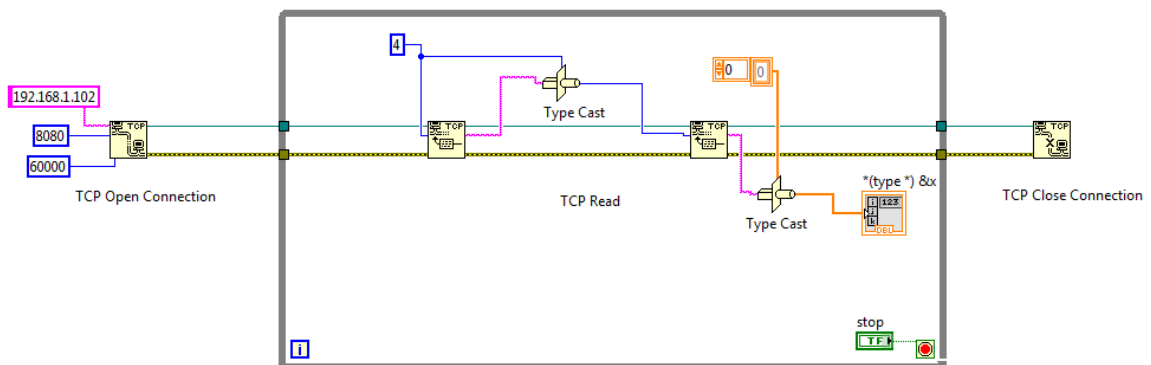


Figura 48.-Diseño de Cliente TCP/IP

Manejo de información y Base de Datos

Especificaciones del Servidor SQL

Una vez establecida la conexión mediante el protocolo TCP/IP, es necesario implementar un servidor SQL, con el fin de almacenar las tablas de actualización de los

estados de los parqueaderos; para que, después mediante un interfaz remota pueda ser esta consultada. Para el levantamiento del servidor se utiliza la herramienta XAMPP que es un desarrollador PHP distribuido por Oracle. La ventaja de este software es su compatibilidad con los diferentes sistemas operativos y la opción de descargar una versión gratuita suficientemente completa para los requerimientos del proyecto. Este se compone de tres servicios de los cuales se requiere MySQL y APACHE para el proyecto.

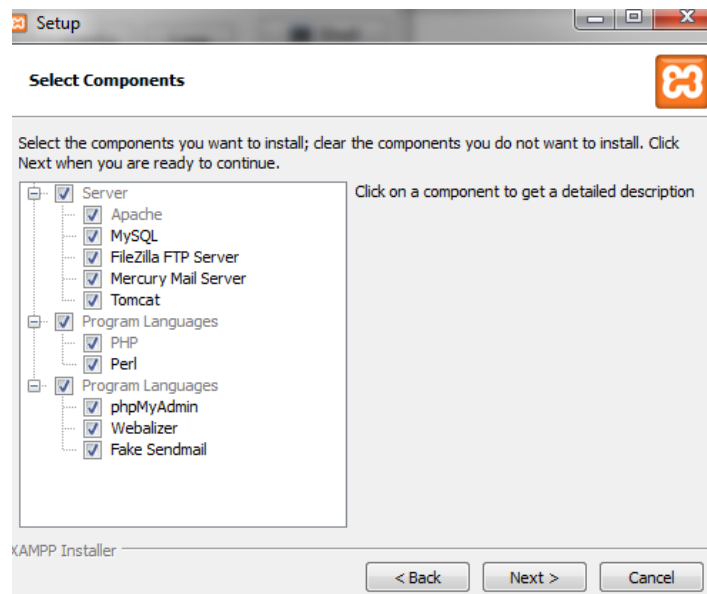


Figura 49.- Pantalla de instalación de XAMPP

Levantamiento de Servidor SQL

Se inicia el proceso eligiendo a una computadora con Windows 7 Ultimate de 64 bits para levantar los servidores. Además se requiere descargar el software en la página oficial “https://www.apachefriends.org/es/download_success.html” teniendo en cuenta la descripción del sistema operativo en el que se corre. Siguiendo el orden del proceso, se ejecuta el instalador donde se selecciona los servicios a levantar como se observa en la Figura 49. Para poder efectuar una correcta instalación, es necesario desactivar el antivirus y descargar pequeños programas que el mismo instalador indica. Realizado lo anterior, el

sistema crea un acceso rápido donde se despliega el Panel de Control (Figura 50), en el cual se configura, detiene o inicia el servidor.

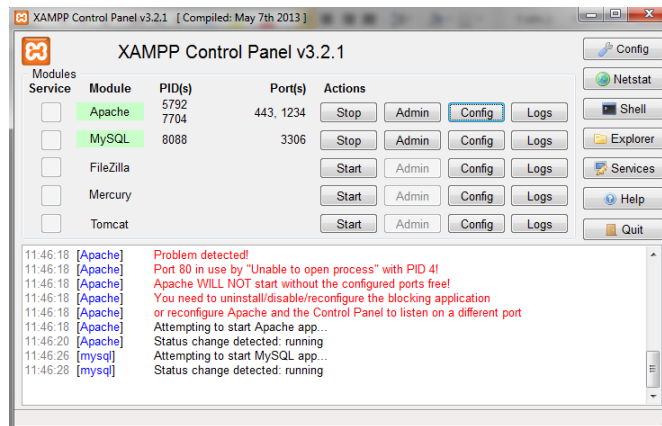


Figura 50.- Ventana del Panel de Control XAMPP

En un inicio existe conflicto en la asignación del puerto 80 en Apache, debido a que Windows utiliza este puerto para diversos procesos HTTP. Por lo tanto, es necesario cambiarlo mediante el archivo de configuración “httpd.conf” en el panel de control de XAMPP. En el comando lisen se cambia por un puerto diferente, en este caso se asiga el puerto 1234. Para comprobar si los servicios se han levantado correctamente se accede mediante un browser a la dirección <http://localhost:1234/xampp/>. En la interfaz web que se despliega, existe un menú donde se consulta el estado de la conexión así como se configura la contraseña de acceso al servidor.

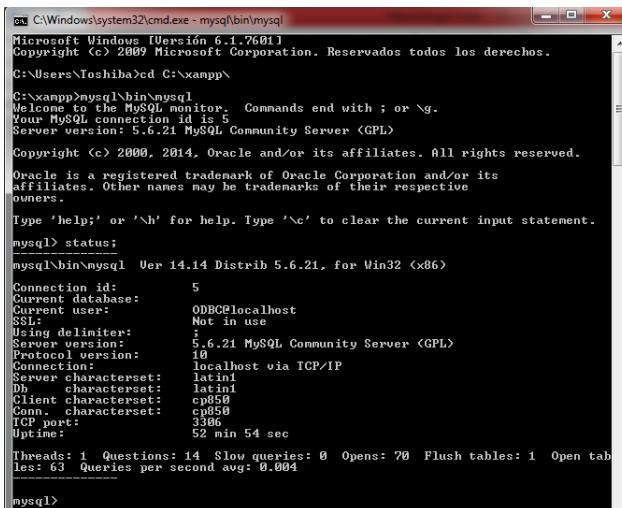


Figura 51.- Verificación de levantamiento de Servidor SQL en cmd

Por medio de la pantalla de comandos MS-DOS, se ingresa al servidor para verificar las configuraciones realizadas como indica la Figura 51. Los mecanismos de autenticación para acceder al servidor SQL se definen por defecto a excepción de la clave de acceso que se lo detalla en la interfaz web. El resumen de los parámetros se muestra en la Tabla 5

Tabla 5.- Parámetros de acceso servidor MySQL

Descripción	Parámetro
Dirección de acceso	http://localhost:1234/xampp/
Puerto de acceso	3306
Usuario	Root
Contraseña	12345

Fuente: Creación propia

Creación de Base de Datos

Si bien el servidor SQL se levanta con éxito, es necesario crear la base de datos que se actualice constantemente. Mediante la herramienta MySQL Workbench 6.2 C se diseña

la base de datos. Este programa permite enlazar con mayor facilidad a la plataforma del servidor MySQL. Además utiliza diagramas para la construcción de los modelos de base de datos, con el fin de mantener la jerarquía y mejorar su administración.

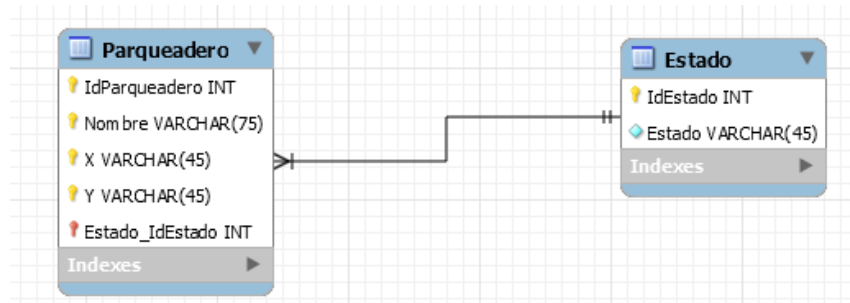


Figura 52.- Diagrama EER base de datos sql_tesis

Para el diseño de la base de datos, se inicia creando una plantilla donde se define los campos y la estructura en sí que tendrán las tablas. De acuerdo a las necesidades del proyecto se crea un esquemático EER con el nombre de sql_tesis que contiene dos tablas como se muestra en la Figura 52.

La imagen muestra la configuración de columnas en las tablas de base de datos sql_tesis. Se muestran dos ventanas de configuración de tablas: 'Estado - Table' y 'Parqueadero - Table'.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
IdEstado	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Estado	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
IdParqueadero	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Nombre	VARCHAR(75)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
X	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Y	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estado_IdEstado	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 53.- Configuración de columnas en tablas de base de datos sql_tesis

Por un lado, la tabla estado se compone de dos columnas definidas. La primera un Id que representa el identificador y la segunda que define el nombre. Por otro lado la tabla parqueadero se compone de la columna IdParqueadero que representa el identificador numérico de cada uno de los campos que se asignen a la tabla, la columna Nombre donde

se coloca la designación del campo, “Y - X” indican las coordenadas tanto en latitud como en longitud que después la interfaz del cliente requerirá y finalmente la columna Estado_IdEstado es una columna heredada de la tabla estado. En la Figura 53 se muestra la creación de las diferentes columnas de las dos tablas.

```

1  -- MySQL Workbench Forward Engineering
2  ● SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
3  ● SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
4  ● SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
5  ● CREATE SCHEMA IF NOT EXISTS `sql_tesis` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci ;
6  ● USE `sql_tesis` ;
7  /*Creación de tabla estado*/
8  -----
9  ● CREATE TABLE IF NOT EXISTS `sql_tesis`.`Estado` (
10     `IdEstado` INT NOT NULL AUTO_INCREMENT,
11     `Estado` VARCHAR(45) NOT NULL,
12     PRIMARY KEY (`IdEstado`),
13     UNIQUE INDEX `IdEstado_UNIQUE` (`IdEstado` ASC),
14     UNIQUE INDEX `Estado_UNIQUE` (`Estado` ASC)
15     ENGINE = InnoDB;
16  /*Creación de tabla parqueadero*/
17  ● CREATE TABLE IF NOT EXISTS `sql_tesis`.`Parqueadero` (
18     `IdParqueadero` INT NOT NULL AUTO_INCREMENT,
19     `Nombre` VARCHAR(75) NOT NULL,
20     `X` VARCHAR(45) NOT NULL,
21     `Y` VARCHAR(45) NOT NULL,
22     `Estado_IdEstado` INT NOT NULL,
23     PRIMARY KEY (`IdParqueadero`, `Nombre`, `X`, `Y`, `Estado_IdEstado`),
24     UNIQUE INDEX `IdParqueadero_UNIQUE` (`IdParqueadero` ASC),
25     INDEX `fk_Parqueadero_Estado_idx` (`Estado_IdEstado` ASC),
26     CONSTRAINT `fk_Parqueadero_Estado`
27     FOREIGN KEY (`Estado_IdEstado`)
28     REFERENCES `sql_tesis`.`Estado` (`IdEstado`)
29     ON DELETE NO ACTION
30     ON UPDATE NO ACTION)
31     ENGINE = InnoDB;
32  ● SET SQL_MODE=@OLD_SQL_MODE;
33  ● SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
34  ● SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Figura 54.- Código de creación Base de datos sql_tesis

MySQL Workbench compila y convierte el diagrama sql_tesis en lenguaje para el servidor. El código que arroja el programa se muestra en la Figura 54. Realizado lo anterior, se accede al servidor mediante cmd y se ingresa el código. Posteriormente se agrega los campos con los comandos set, así como se visualiza lo escrito con el comando select. Un resumen de los comandos utilizados se muestra en la Figura 55.

```

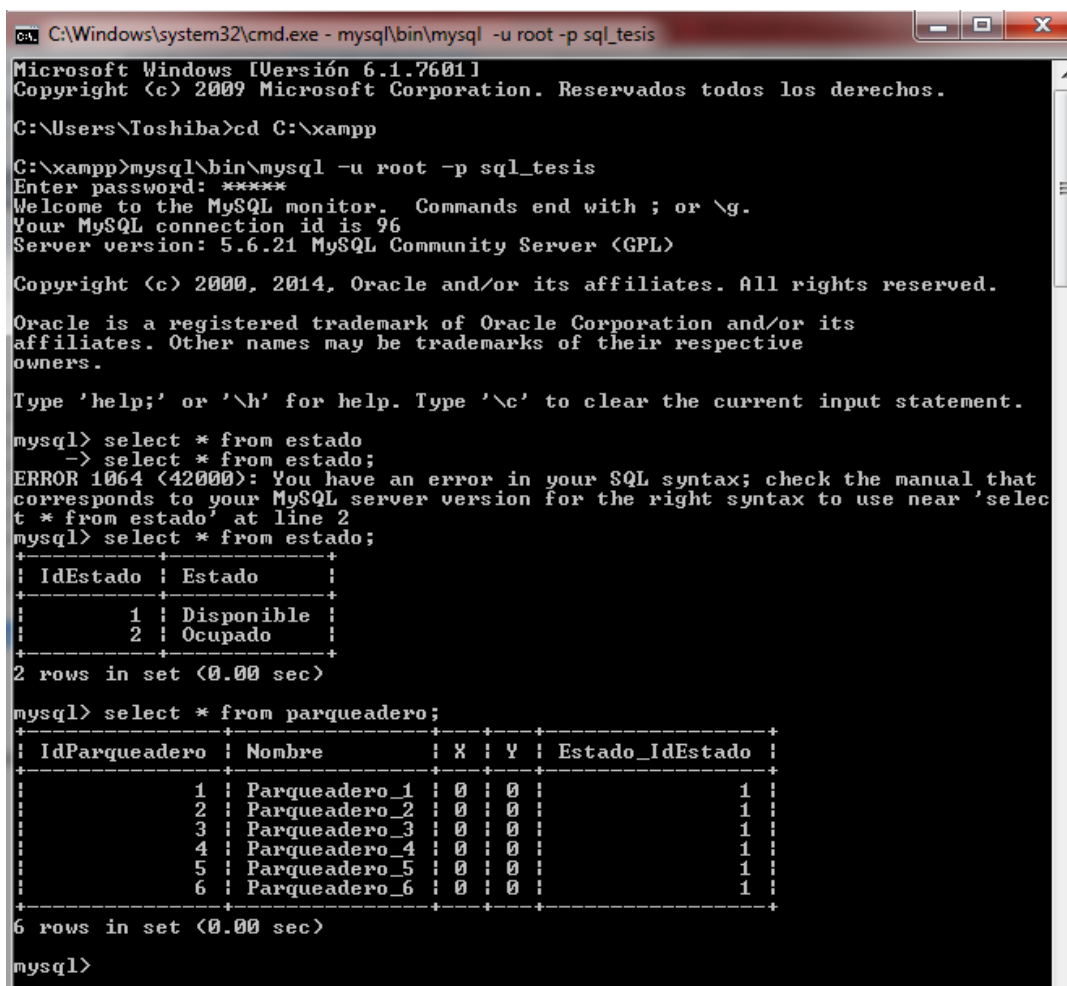
INSERTAR DATOS TABLA PARQUEADERO
insert into parqueadero(Nombre,X,Y,Estado_IdEstado) values ('Parqueadero_1',0,0,1);
insert into parqueadero(Nombre,X,Y,Estado_IdEstado) values ('Parqueadero_2',0,0,1);
insert into parqueadero(Nombre,X,Y,Estado_IdEstado) values ('Parqueadero_3',0,0,1);
insert into parqueadero(Nombre,X,Y,Estado_IdEstado) values ('Parqueadero_4',0,0,1);
insert into parqueadero(Nombre,X,Y,Estado_IdEstado) values ('Parqueadero_5',0,0,1);
insert into parqueadero(Nombre,X,Y,Estado_IdEstado) values ('Parqueadero_6',0,0,1);

INSERTAR DATOS TABLA PARQUEADERO
insert into estado(Estado) values ('disponible');
insert into estado(Estado) values ('Ocupado');

```

Figura 55.- Tabla de comandos para actualizar base de datos.

Se da por finalizado el proceso cuando la ventana cmd muestra las tablas parqueadero y estado, con sus respectivos contenidos a través del comando select. En la Figura 56 se presenta el proceso.



```

ca: C:\Windows\system32\cmd.exe - mysql\bin\mysql -u root -p sql_tesis
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Toshiba>cd C:\xampp

C:\xampp>mysql\bin\mysql -u root -p sql_tesis
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 96
Server version: 5.6.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from estado
-> select * from estado;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'selec
t * from estado' at line 2
mysql> select * from estado;
+-----+-----+
| IdEstado | Estado |
+-----+-----+
|         1 | Disponible |
|         2 | Ocupado |
+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from parqueadero;
+-----+-----+-----+-----+-----+-----+
| IdParqueadero | Nombre | X | Y | Estado_IdEstado |
+-----+-----+-----+-----+-----+-----+
|         1 | Parqueadero_1 | 0 | 0 | 1 |
|         2 | Parqueadero_2 | 0 | 0 | 1 |
|         3 | Parqueadero_3 | 0 | 0 | 1 |
|         4 | Parqueadero_4 | 0 | 0 | 1 |
|         5 | Parqueadero_5 | 0 | 0 | 1 |
|         6 | Parqueadero_6 | 0 | 0 | 1 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>

```

Figura 56.- Demostración de la creación de la base de datos sql_tesis a través de ventana cmd

Comunicación entre MyRIO 1900 y Servidor SQL utilizando texto plano

La comunicación entre los dos dispositivos es vital para el alcance de la interfaz del cliente, ya que de esto depende la actualización del estado de cada parqueadero en la base de datos. Para esto se utiliza el cliente TCP/IP programado en LabVIEW (ver Figura 48)

que guarda la información en un texto plano. De igual forma existe otra alternativa que utilizando la misma comunicación TCP/IP genera un archivo.xls como se observa en la Figura 97 (Anexo C). Este archivo es almacenado en el directorio del servidor para su posterior manejo. El archivo plano denominado “actualización.txt” contiene el identificador del parqueadero. En la Figura 57 se observa el proceso de comunicación del cliente TCP/IP con la creación de los textos planos.

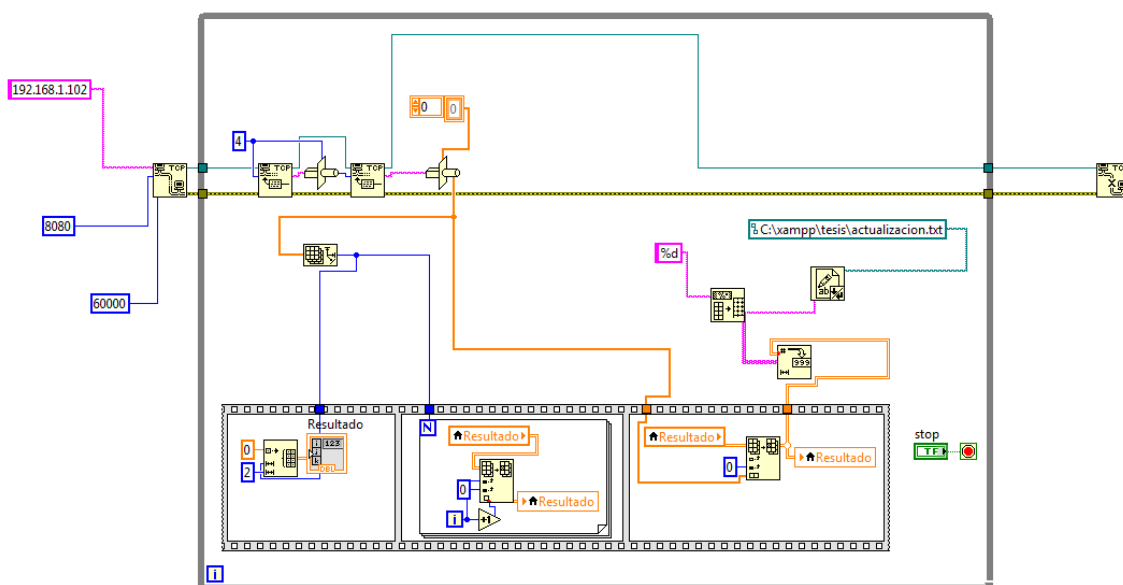
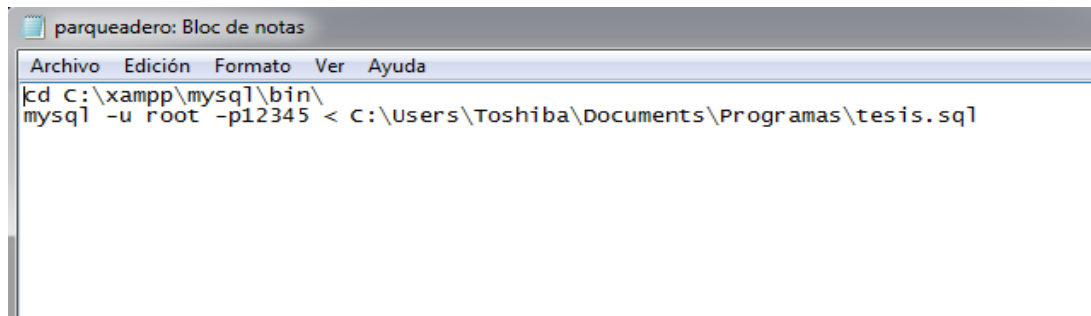


Figura 57.- Comunicación MyRIO-Servidor SQL en texto plano

Establecida la conexión indicada anteriormente, se genera el texto “actualizacion.txt” que a través de un archivo .bat se accede a la base de datos sql_tesis y se reemplazan los datos. El directorio que permite acceder a la base de datos se encuentra incluido dentro del ejecutable denominado “parqueadero.bat” como se demuestra en la Figura 58, donde se ingresa el usuario, contraseña y el url del archivo .sql que tiene las sentencias de actualización.



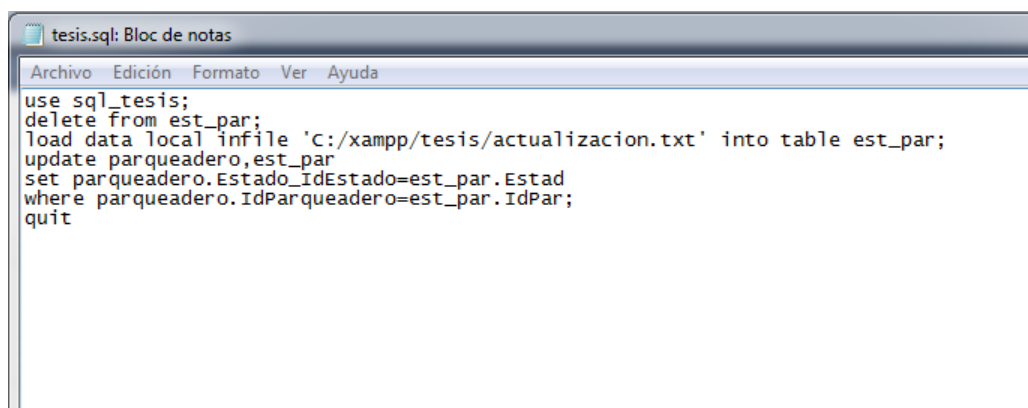
```

parqueadero: Bloc de notas
Archivo Edición Formato Ver Ayuda
cd C:\xampp\mysql\bin\
mysql -u root -p12345 < C:\Users\Toshiba\Documents\Programas\tesis.sql

```

Figura 58.- Programa del archivo parqueadero.bat

Para continuar con el proceso, dentro de la base de datos `sql_tesis`, es importante la creación de una nueva tabla llamada `est_par`, que sirve para cargar en sus campos el archivo plano “actualización”. Esta tabla se compone de `IdPar` y `Estad` que contiene dos columnas de características similares del archivo `actualización.txt` que inserta la información actualizando los campos de `IdParqueadero` y `Estado_IdEstado` de la tabla `parqueadero` con los campos de la tabla `est_par`.



```

tesis.sql: Bloc de notas
Archivo Edición Formato Ver Ayuda
use sql_tesis;
delete from est_par;
load data local infile 'c:/xampp/tesis/actualizacion.txt' into table est_par;
update parqueadero,est_par
set parqueadero.Estado_IdEstado=est_par.Estad
where parqueadero.IdParqueadero=est_par.IdPar;
quit

```

Figura 59.- Programa tesis.sql

El paso siguiente es realizar una tarea programada, donde constantemente cada minuto se ejecuta el archivo “parqueadero.bat”. Para ello es necesario acceder al panel de control, crear el proceso y vincular la dirección del ejecutable como se muestra en la Figura 60.

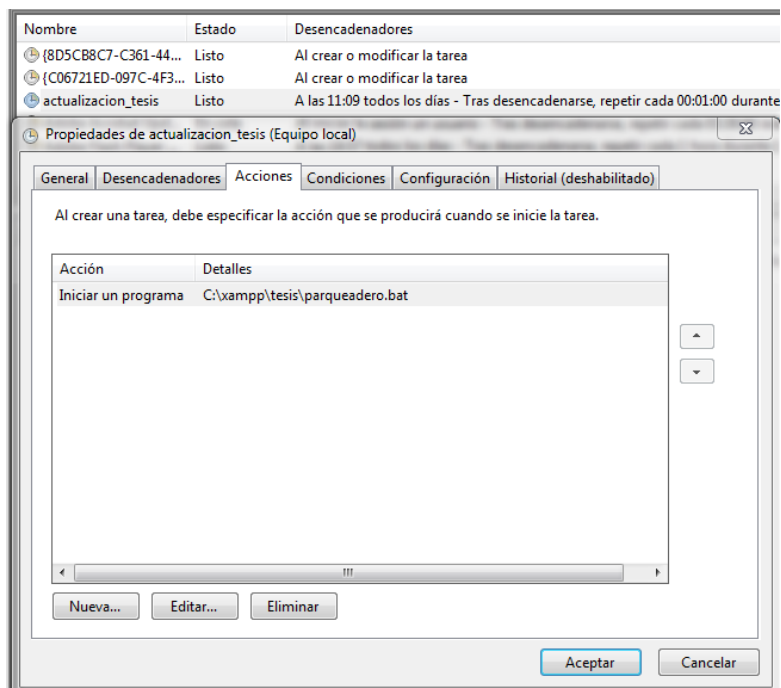


Figura 60.- Tarea programada actualización_tesis

Interfaces del cliente

Después de levantar el servidor SQL, la elección de las interfaces para el cliente se abre en varias opciones. Elección que permitirá que la interfaz que se elija acceda únicamente a la base de datos actualizada en el SQL, sin intervenir con los procesos que sucedan del lado del MyRIO. Esta topología aumenta la seguridad del sistema y garantiza independencia de los procesos, tanto para el cliente como para el hardware de procesamiento.

Creación Página Web

Las páginas web debido al crecimiento en el uso de internet así como la familiarización con esta tecnología, hacen que esta interfaz sea muy amigable para el cliente. Existen varias herramientas con el servidor que se utiliza (XAMPP) ya que permiten levantar de forma local páginas web escritas en lenguaje HTML. Para editar se

utiliza el programa de edición de texto Sublime Text que admite manipular los lenguajes en HTML y Java Script.

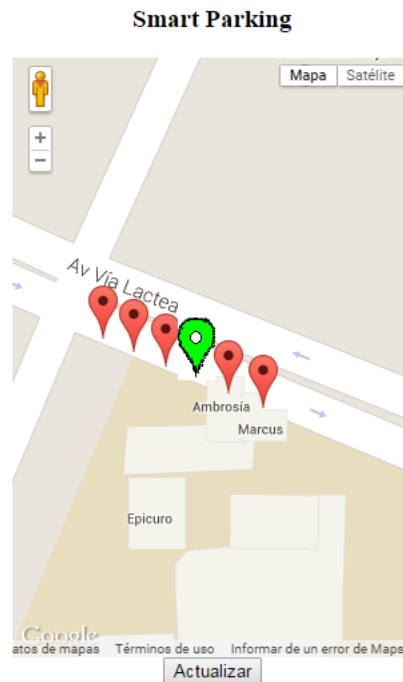


Figura 61.- Página Web Smart Parking

Para la creación de la página web se implantan tres archivos escritos en lenguaje php y uno en java script. Todos los archivos, así como los objetos y fotos a utilizarse en la página web, se guardan en un directorio dentro del servidor web Apache. Como se indica en la Figura 62 el directorio es C:\xampp\htdocs\tesis.

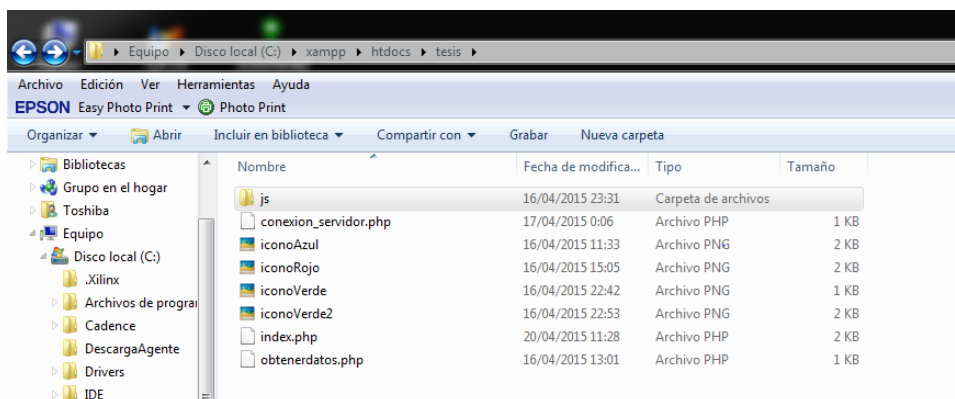
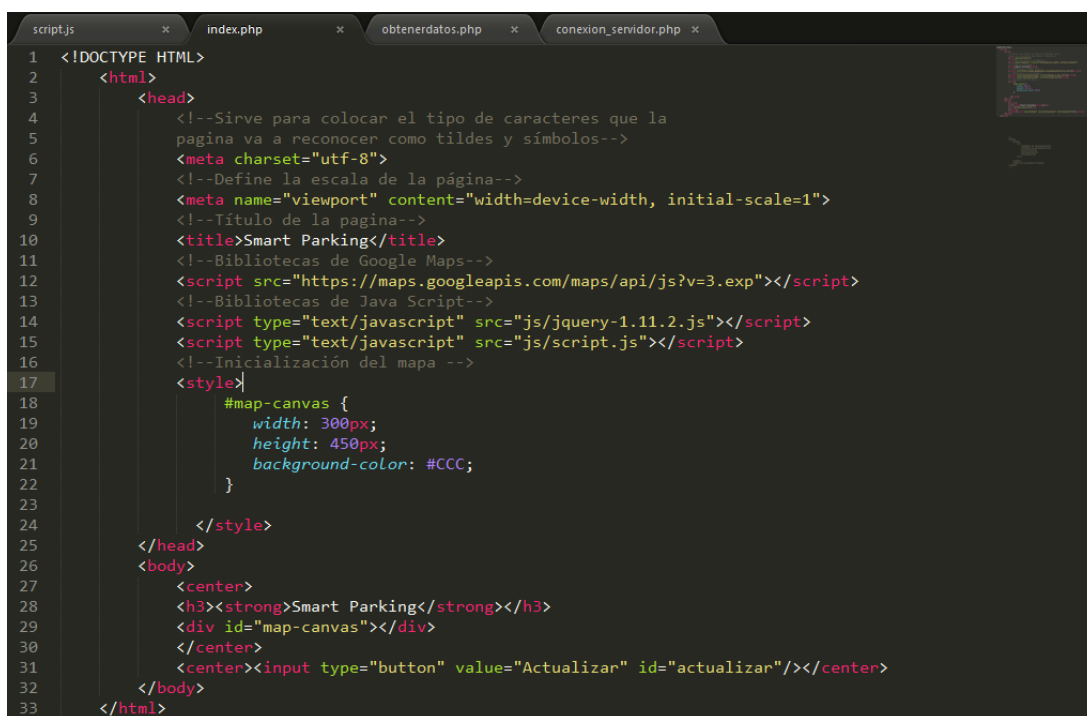


Figura 62.- Directorio archivos página Web en servidor web Apache

El archivo que da forma a la página es index.php escrito en lenguaje HTML. Este contiene la estructura final que tendrá la interfaz con el cliente como se demuestra en la Figura 63. La organización del archivo HTML consta de tres partes. Se inicia con el head donde se ponen los parámetros de programación como el tipo de caracteres a utilizar, la escala de la página, las bibliotecas y el título de inicio. Además se incorpora los objetos, en este caso el mapa de Google maps. Para la utilización de los comandos de Java Script es necesario llamar a las bibliotecas jquery-1.11.2 y angular.min que en igual forma se encuentran en el directorio de la Figura 62. La segunda parte es el body donde se escribe el código para indicar los elementos y el diseño en la pantalla de la página web.



```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <!--Sirve para colocar el tipo de caracteres que la
5 pagina va a reconocer como tildes y símbolos-->
6 <meta charset="utf-8">
7 <!--Define la escala de la página-->
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <!--Título de la página-->
10 <title>Smart Parking</title>
11 <!--Bibliotecas de Google Maps-->
12 <script src="https://maps.googleapis.com/maps/api/js?v=3.exp"></script>
13 <!--Bibliotecas de Java Script-->
14 <script type="text/javascript" src="js/jquery-1.11.2.js"></script>
15 <script type="text/javascript" src="js/script.js"></script>
16 <!--Inicialización del mapa -->
17 <style>
18 <div id="map-canvas" style="width: 300px; height: 450px; background-color: #CCC;
19 }
20 }
21 }
22 }
23 }
24 </style>
25 </head>
26 <body>
27 <center>
28 <h3><strong>Smart Parking</strong></h3>
29 <div id="map-canvas"></div>
30 </center>
31 <center><input type="button" value="Actualizar" id="actualizar"/></center>
32 </body>
33 </html>

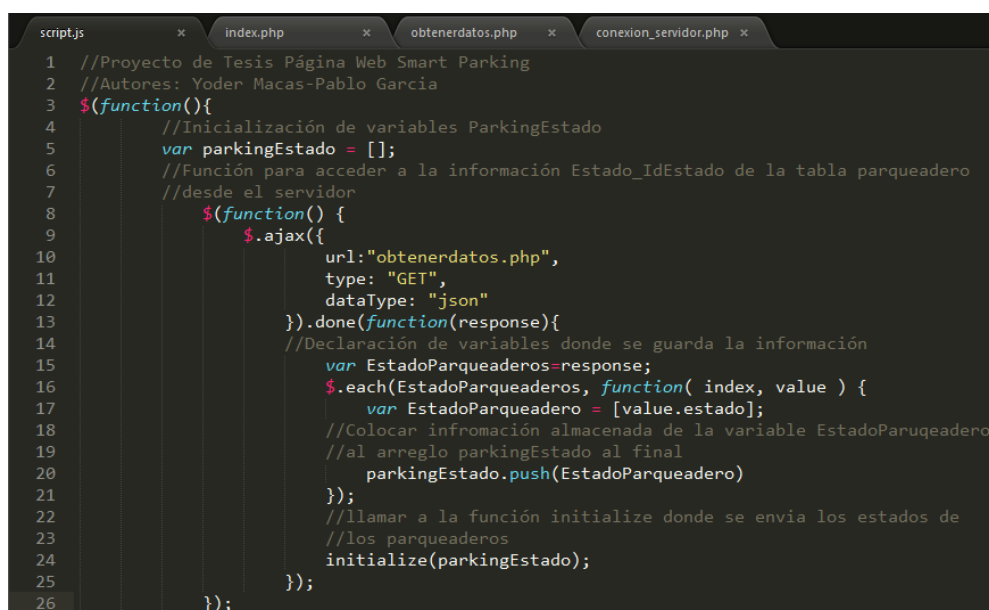
```

Figura 63.- Código archivo index.php

Se utilizan los comandos center, h3, y strong para definir del título principal la posición, el tamaño y el estilo respectivamente. Asimismo con el indicador “div” se coloca

el objeto anteriormente inicializado, y con la etiqueta input se crea un botón que actualiza la página cada vez que se lo acciona.

La programación de los objetos se realiza en un segundo archivo con nombre script.js. A diferencia del primero, este es escrito en lenguaje java script y contiene una estructura orientada a objetos. La Figura 64 muestra la configuración del archivo.



```

1 //Proyecto de Tesis Página Web Smart Parking
2 //Autores: Yoder Macas-Pablo Garcia
3 $(function(){
4     //Inicialización de variables ParkingEstado
5     var parkingEstado = [];
6     //Función para acceder a la información Estado_IdEstado de la tabla parqueadero
7     //desde el servidor
8     $(function() {
9         $.ajax({
10            url: "obtenerdatos.php",
11            type: "GET",
12            dataType: "json"
13        }).done(function(response){
14            //Declaración de variables donde se guarda la información
15            var EstadoParqueaderos=response;
16            $.each(EstadoParqueaderos, function( index, value ) {
17                var EstadoParqueadero = [value.estado];
18                //Colocar infromación almacenada de la variable EstadoParqueadero
19                //al arreglo parkingEstado al final
20                parkingEstado.push(EstadoParqueadero)
21            });
22            //llamar a la función initialize donde se envia los estados de
23            //los parqueaderos
24            initialize(parkingEstado);
25        });
26    });

```

Figura 64.- Código archivo script.js

Este se resume en tres funciones. La primera permite a través de un URL ejecutar un programa llamado “obtenerdatos.php” (ver Figura 69) que efectúa la consulta a la base de Datos en el servidor SQL. En esta función se crea un arreglo con los datos obtenidos del campo Estado_IdEstado de la tabla sql_tesis. Una vez que el arreglo es construido se lo envía a la segunda función denominada initialize como se muestra en la Figura 65.

```

27 //Función de inicialización del google maps y adquisición de marcadores dependiendo
28 //de la longitud y latitud
29
30 function initialize(a) {
31 //Se coloca la dirección inicial donde comienza google maps al igual que
32 //el zoom de inicio y el tipo de mapa
33 var coordenasUSFQ = new google.maps.LatLng(-0.195418, -78.435832 );
34 var mapOptions = {
35     zoom: 19,
36     center: coordenasUSFQ,
37     mapTypeId: google.maps.MapTypeId.ROADMAP,
38 }
39 var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
40 //Se inicializa las coordenadas de latitud y longitud de los marcadores
41 var myLatLng1 = new google.maps.LatLng(-0.195379, -78.436053);
42 var myLatLng2 = new google.maps.LatLng(-0.195407, -78.435990);
43 var myLatLng3 = new google.maps.LatLng(-0.195435, -78.435927);
44 var myLatLng4 = new google.maps.LatLng(-0.195463, -78.435864);
45 var myLatLng5 = new google.maps.LatLng(-0.195491, -78.435801);
46 var myLatLng6 = new google.maps.LatLng(-0.195519, -78.435731);
47 //Se inicializan las variables que tiene la animación y los iconos de a mostrarse
48 //dependiendo del estado del parqueadero
49 var image1 = 'iconoVerde2.png';
50 var image2 ;
51 var imagenp1;
52 var imagenp2;
53 var imagenp3;
54 var imagenp4;
55 var imagenp5;
56 var imagenp6;
57 var anim1 = google.maps.Animation.DROP;
58 var anim2 ;

```

Figura 65.- Código función initialize del archivo script.js

En esta función, el mapa de google arranca colocando las coordenadas de inicio, el zoom y el tipo del mapa. Luego se declaran las posiciones en longitud y latitud de cada uno de los parqueaderos para direccionarle a un marcador. Debido a que el proyecto utiliza solamente una cámara, la posición de cada parqueadero se determina en un lugar específico, haciendo que éste fuera en la Avenida Vía Láctea lateral a la Universidad. Seguido de esto, cada marcador utiliza las variables de animación e íconos que permiten visualizar las condiciones del estado de cada parqueadero (ver Figura 66).

```

58 //Se declara las condiciones de los iconos dependiendo del estado así como la animación
59
60 if ( a[0] == 'Ocupado'){
61     var animacionp1 = anim2;
62     imagenp1 = image2;
63 } else {
64     animacionp1=anim1;
65     imagenp1 = image1;
66 }
67 if ( a[1] == 'Ocupado'){
68     var animacionp2 = anim2;
69     imagenp2 = image2;
70 } else {
71     animacionp2=anim1;
72     imagenp2 = image1;
73 }
74 if ( a[2] == 'Ocupado'){
75     var animacionp3 = anim2;
76     imagenp3 = image2;
77 } else {
78     animacionp3=anim1;
79     imagenp3 = image1;
80 }
81 if ( a[3] == 'Ocupado'){
82     var animacionp4 = anim2;
83     imagenp4 = image2;
84 } else {
85     animacionp4=anim1;
86     imagenp4 = image1;
87 }
88 if ( a[4] == 'Ocupado'){
89     var animacionp5 = anim2;
90     imagenp5 = image2;
91 } else {
92     animacionp5=anim1;
93     imagenp5 = image1;
94 }
95 if ( a[5] == 'Ocupado'){

```

Figura 66.- Condiciones de cada estado de parqueadero

Luego se crea cada marcador con sus diferentes etiquetas y variables antes definidas como muestra la Figura 67. La tercera función es para activar las animaciones que utilizan los estados disponibles.

```

97 //Creacion de los marcadores con su respectiva localización, animacion e icono
98 var marker1 = new google.maps.Marker({
99   position: myLatLng1,
100   map: map,
101   title: 'P1',
102   //draggable:true,
103   animation: animacionp1,
104   icon: imagenp1,
105 });
106 var marker2 = new google.maps.Marker({
107   position: myLatLng2,
108   map: map,
109   title: 'P2',
110   animation: animacionp2,
111   icon: imagenp2,
112 });
113 var marker3 = new google.maps.Marker({
114   position: myLatLng3,
115   map: map,
116   title: 'P3',
117   animation: animacionp3,
118   icon: imagenp3,
119 });
120 var marker4 = new google.maps.Marker({
121   position: myLatLng4,
122   map: map,
123   title: 'P4',
124   animation: animacionp4,
125   icon: imagenp4,
126 });
127 var marker5 = new google.maps.Marker({
128   position: myLatLng5,
129   map: map,
130   title: 'P5',
131   animation: animacionp5,
132   icon: imagenp5,
133 });
134 var marker6 = new google.maps.Marker({
135   position: myLatLng6,
136   map: map,
137   title: 'P6',
138   animation: animacionp6,
139   icon: imagenp6,
140 });

```

Figura 67.- Declaración de marcador para cada parqueadero en programa script.js

Para acceder al servidor y obtener la información de la base de datos sql_tesis, es necesario de dos archivos php que inicien la sesión y que envíen la sentencia en lenguaje sql. Primero se accede al servidor creando la conexión, donde se declara la dirección del servidor, el usuario, la contraseña y la tabla en la cual se va a trabajar. Además se coloca un mensaje de error si existe algún inconveniente con la base de datos. Y luego se define el formato de los caracteres para establecer la conexión (ver Figura 68).

```

1 <?php
2 //Se inicia la conexion con la base de datos en el lado del servidor
3 $conexcion= new mysqli ("localhost","root","12345","sql_tesis");
4 // realiza la peticion para realizar las consultas
5 if(mysqli_connect_errno()){
6     // llamo al error del servidor
7     echo ("Error de conexión".mysqli_connect_error());
8     exit();
9 }
10 // definir los caracteres que traen desde el servidor
11 mysqli_set_charset($conexcion,"utf8");
12 //utf8 es el tipo de
13
14
15
16
17
18 ?>

```

Figura 68.- Programa conexion_servidor.php

Para concluir se crea un archivo php donde utilizando el programa de conexión, envía la sentencia Select que trae los campos de la tabla sql_tesis.sql. Esto se muestra en la Figura 69.

```

1 <?php
2 include "conexion_servidor.php";
3
4 if($pResultado = $conexcion->query("SELECT IdParqueadero,Nombre,X,Y,estado.Estado as estado from parqueadero inner join estado on
5 | parqueadero.Estado_IdEstado=estado.IdEstado order by IdParqueadero;")){
6     $rows = array();
7     while($pFila = $pResultado->fetch_assoc()){
8         $rows[] = $pFila;
9     }
10 $pResultado->free();
11 // IMPRIMIR
12 echo json_encode($rows);
13 }
14 $conexcion->close();
15
16
17 ?>

```

Figura 69.- Programa conexion_servidor.php

Una vez guardados los programas en la carpeta que corre el servidor, se accede a la dirección del servidor especificando el puerto y la carpeta. De acuerdo a la topología diseñada en la Figura 41, cualquier host que se conecte a esta red tendrá acceso a la página web. Como se muestra en la Figura 70, desde un dispositivo móvil se accede a la página web apuntando a la dirección del servidor.

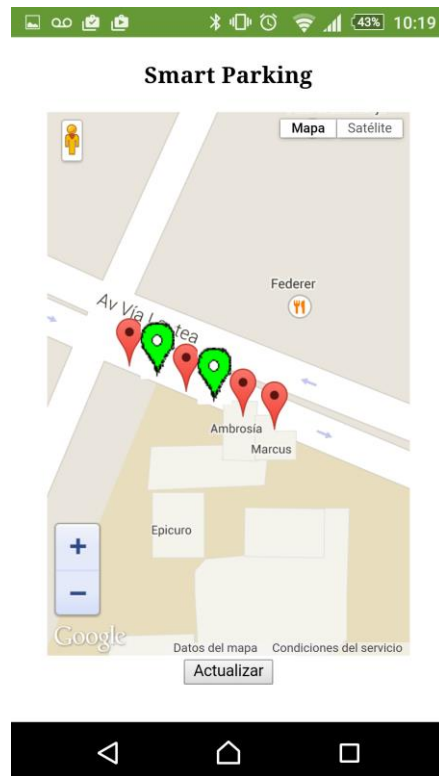


Figura 70.- Pagina web desde dispositivo móvil

Resumen del Capítulo

Se estudió paso a paso la estructura técnica del proyecto así como la configuración de la interfaz del cliente. En un comienzo se mostró los requerimientos del proyecto relacionado a la adquisición de imágenes, así como la selección de la cámara. Se detallaron los programas de apertura y de almacenamiento de la imagen escrito en LabVIEW. En la parte del sistema de movimiento rotacional, se mostró la conexión del servomotor así como la manipulación de la señal PWM desde el hardware. Para el procesamiento de imagen, se indicó la estructura propuesta para el desarrollo del proyecto. Por una parte, en la segmentación de la imagen se explicó el tratamiento que sufre cada frame a través de filtros para detectar automáticamente las dimensiones del parqueadero. El proceso resultó

de la utilización de filtros de detección de bordes, rotación de imagen, extracción de color y transformaciones geométricas. Por otra parte en la fase de comparación, se almacenó la imagen en un arreglo que a través de un registro, creaba una memoria donde se restan las dos imágenes para obtener una variación promedio y determinar si existe un cambio o no.

Además la integración de red, manejo de datos y sistemas de comunicación constituyeron una parte importante dentro de este capítulo. Se explicó el levantamiento de un servidor SQL que se utilizará para actualizar la tabla de estados de cada estacionamiento, así como la integración del sistema en una red local a través de los protocolos IEEE802.11 y Ethernet. En el proceso de comunicación se generó un archivo plano que a través de comandos sql, actualizan la base de datos mediante una tarea programada. Por último se mostró los comandos html y java script para la creación de la página web.

ANÁLISIS DE RESULTADOS

Antes de comenzar a desarrollar este capítulo, se debe mencionar que los lineamientos se encuentran enmarcados dentro de los parámetros establecidos por la universidad. Los resultados del estudio se hallan en tablas gráficas y diagrama que permiten visualizar todo el proceso que ha tenido el proyecto.

El trabajo realizado relaciona los datos y resultados que se encontró en la investigación, con la información de la base teórica y los antecedentes, por lo que se irá interpretando, analizando y explicando conforme el avance del estudio en cada uno de los procesos.

Determinación Cámara Web

Comprobación de calidad de imagen

Como dijimos anteriormente la resolución de la imagen no es de gran importancia ya que estamos interesados en los detalles grandes de la imagen, sin embargo al revisar la resolución de ambas cámaras se vio que pueden trabajar bajo los parámetros de las Tabla 21 y Tabla 22 (Anexo B).

Las tablas descritas, nos permiten observar que 640x480, es el valor que se encuentra en el rango ni alto ni muy bajo, y es la relación que mejor se ajusta a la “relación largo ancho de los parqueaderos de la zona azul” por cuanto reconoce a todo el parqueadero, como se observa en la

Figura 93. Los frames que capturan la imagen por segundo, no afectan al proyecto ya que para capturar solo se necesita uno de ellos, en este caso se utiliza el más bajo, es decir 5fps, en caso de utilizar una cantidad de frames adquiridos por segundo más alta no afectaría el resultado final de las adquisiciones y comparaciones de imágenes, pero se le estaría exigiendo a la tarjeta MyRio 1900 un mayor procesamiento inútil para nuestro propósito.

Caracterización de ángulo de visión

La cámara se ubicó en un lugar específico para caracterizar el ángulo de visión, en posición vertical se tomó una foto que permitió medir el largo que es capaz de capturarla, después se midió la altura a la que está posicionado el lente. El ángulo se calcula como se muestra en la ecuación (11).

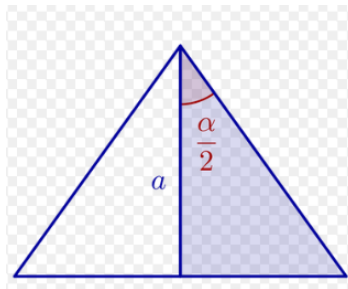


Figura 71.-Cálculo de ángulo de Visión de una cámara

$$\alpha = 2 \tan^{-1} \left(\frac{b}{2a} \right) \quad (11)$$

Como mencionamos anteriormente utilizamos 2 modelos distintos de cámaras la Cámara Logitech C160 y la Genius Facecam 1000x, para ambas se utilizó la resolución de 640x480. Para la primera se hizo la prueba sobre papel con la cámara colocada a una altura de 43.5 cm y se obtuvo la siguiente imagen (ver Figura 72).

El largo de esta imagen fue de 32.6 cm, lo que dio como resultado un ángulo de visión de 41.08° , para la cámara Genius Facecam 1000x se realizó la prueba sobre el lugar, donde se hicieron las pruebas finales y se la colocó a una altura de 6.02 m obteniendo la siguiente imagen (ver Figura 73).

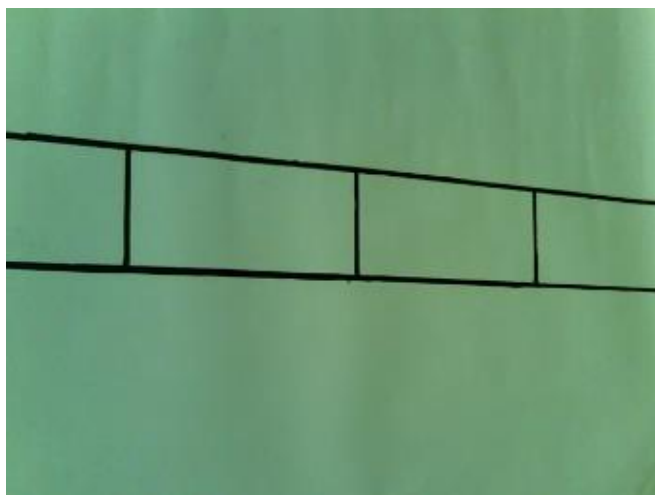


Figura 72.- Imagen Cámara Logitech C160

Esta imagen tiene un largo de 4.7 m lo que da como resultado un ángulo de visión de 42.64° . Como podemos ver la Genius Facecam 1000x tiene un ángulo de visión mayor que la Logitech C160 por lo que finalmente se escogió la cámara Genius para continuar con el proyecto.

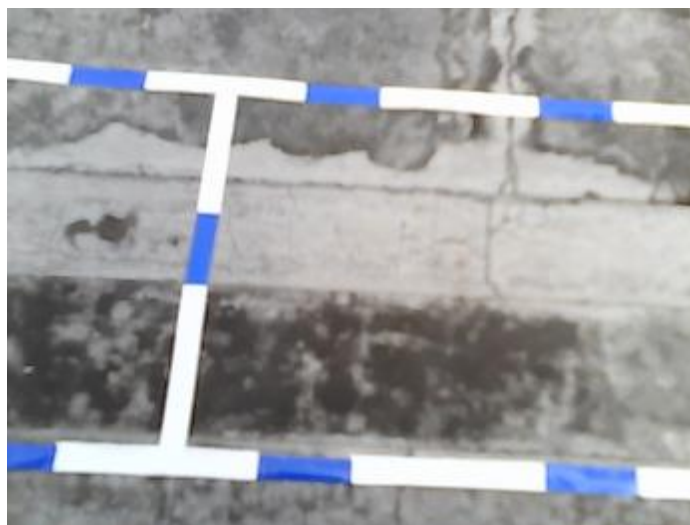


Figura 73.- Imagen cámara Genius Facecam 1000x

Determinación Proceso de adquisición de imágenes

Pruebas de apertura de cámara

La prueba fue efectuada sobre papel, en un lugar donde el sol no llega directamente, el procedimiento se realizó con los mismos parámetros antes mencionados y su resultado fue similar (ver Figura 74).

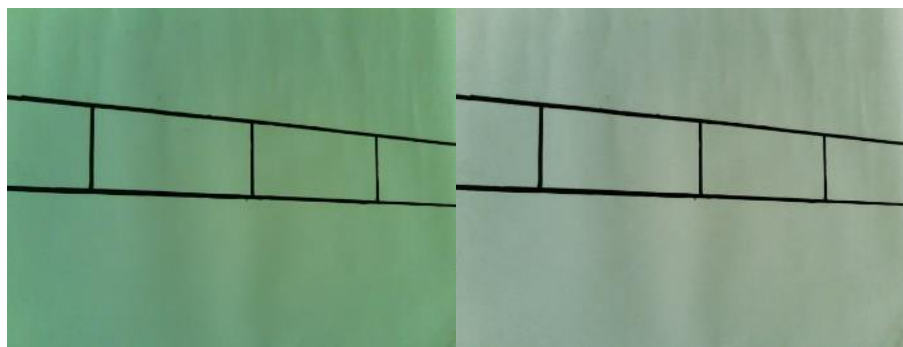


Figura 74.- Pruebas apertura de cámara ambiente controlado

Sin embargo cuando se probó con incidencia directa del sol, el resultado de ambas fue el mismo, no se veía más que un brillo blanco fuerte en la imagen como se muestra en la Figura 75.

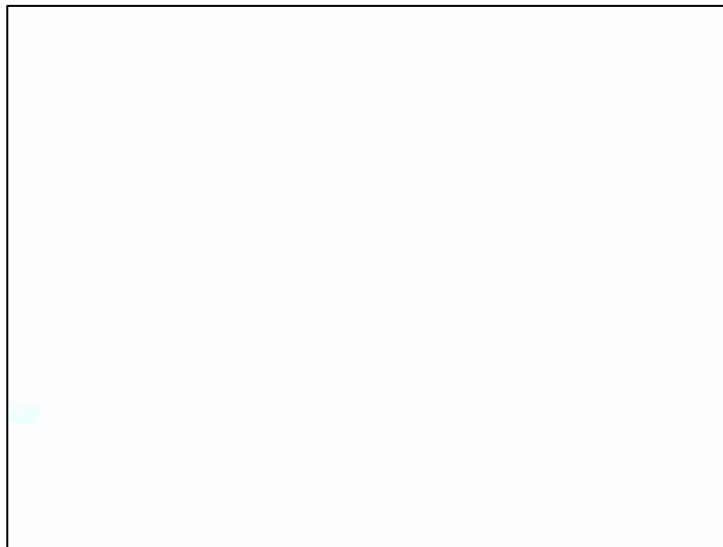


Figura 75.- Imagen apertura de cámara con incidencia directa del sol

Esto sucede por cuanto, la cámara para empezar a capturar correctamente las imágenes que observa, se demora un tiempo hasta acostumbrarse al brillo del exterior. La

Tabla 23 del Anexo B demuestra la diferencia entre el tiempo que se tarda la cámara al abrirse y empezar a capturar imágenes mediante los dos métodos. Como se aprecia es mejor abrir la cámara con la ayuda del Vision Aquisition Express que se demora menos tiempo en empezar a adquirir las imágenes correctamente como se muestra en la Figura 76.



Figura 76.- Imagen capturada con Vision Adquisition Express

Caracterización Sistema de Movimiento Rotacional

Pruebas Servomotor estándar

Para empezar las pruebas se utilizó un servo estándar, con el cual se probó el programa que controla la posición del servomotor mediante el uso de señales PWM enunciado anteriormente. Sin embargo, el servo con este programa dio como resultado un movimiento rotacional continuo desde la posición actual del servo hacia la nueva posición (la que es elegida por el usuario); es decir, el servo se mantenía en movimiento a menos que se deje de enviar la señal PWM. Esto hacía imposible capturar una imagen sin correr el riesgo de que esta salga movida y resulte imposible reconocer correctamente los objetos que se encuentran en la imagen. Incluso, variando la frecuencia a la que se mueve el servo para que el movimiento sea más lento, no se tenía el tiempo suficiente para capturar la imagen. Este resultado fue debido a que el servomotor era de movimiento continuo por lo que, mientras esté activo no mantiene una posición fija.

Pruebas Servomotor marca Hitec HS-311

Después se hizo pruebas con un servo normal que no es de movimiento continuo, al utilizar éste con el programa antes mencionado, se obtuvo los resultados esperados que fue, cambiar la posición del servomotor del actual a la deseada, permaneciendo en la misma hasta que se vuelva a cambiar. Es decir ahora es posible adquirir una imagen, ya que el servo permanece inmóvil hasta que nosotros demos la orden de cambio, en consecuencia tenemos el total control del tiempo de permanencia de la cámara en la posición deseada para adquirir la imagen (ver Figura 84).

Comprobación y optimización del proceso de corte de imagen

Con el fin de responder a las preguntas de la investigación relacionadas a la confiabilidad del sistema frente a cambios en el entorno de prueba, se identificaron cuatro variables independientes (ver Tabla 6) que se tendrá en cuenta para su posterior manipulación. Se crea dos grupos de investigación que corresponden a las variables P1-P2 y dos subgrupos (P3 y P4) analizados dentro de cada grupo. El indicador cuantitativo de medición corresponde al valor numérico de la diferencia de las dos imágenes como se mostró en la metodología de comparación del sistema. De igual forma se analizarán las imágenes obtenidas en la fase de segmentación y comparación.

Tabla 6.- Variables Independientes análisis confiabilidad del sistema

Simbología	Variable Independiente
P1	Variaciones a diferentes horas del día
P2	Variaciones con diferente condiciones climáticas
P3	Variaciones con cambio de estado
P4	Variaciones sin cambio de estado

Fuente: Creación Propia

Para el análisis se obtuvo cuatro muestras de tiempo y cinco muestras de clima como se resume en la Tabla 7.

Tabla 7.- Muestras de grupo de variaciones a diferentes horas del día y estado climáticos

Muestra	Hora del Día	Estado Climático
M1	Media mañana (9:15 am)	Sol con Sombra
M2	Medio día (12:30 pm)	Sol sin Sombra

M3	Tarde (5:30 pm)	Nublado
M4	Noche (7:00 pm)	Lluvia
M5		Neblina

Fuente: Creación Propia

Pruebas con clima nublado

Para esta prueba se corrió el programa de segmentación en clima nublado y a diferentes horas del día entre las 8 am y 6 pm. Los resultados de una de las muestras realizadas al segmentar la imagen, se reflejan a en la imagen Seg 1 de la Figura 86. (Anexo B). Como se observa, las líneas de división de los parqueaderos se ven de manera casi completa salvo por un segmento en la esquina derecha. Además en la imagen hay poca presencia de manchas, es decir líneas o puntos blancos producidos por las imperfecciones de la calzada. A continuación en la Tabla 8 se muestran los resultados obtenidos por el Mathscript de segmentación al ingresar la imagen anterior.

Tabla 8.- Resultado Mathscript muestra "Nublado"

Estado Climático Nublado		
Cantidad	Ancho	Largo
2	47	1
	404	598
		1240

Fuente: Creación Propia

Como podemos ver la cantidad de parqueaderos es correctamente determinado. Ingresando estos datos en el programa de comparación, se obtienen las dos siguientes imágenes cortadas como indica la Figura 77. Con las imágenes cortadas correctamente,

podemos decir que para la fase de segmentación, un clima nublado sin mucho sol, es un escenario adecuado.

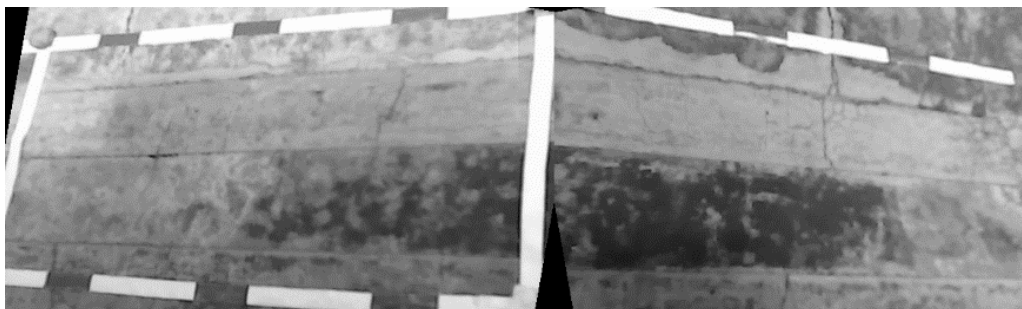


Figura 77.- Prueba de comparación clima nublado

Pruebas de clima con Sol sin Sombra

Para esta prueba se corrió el programa de segmentación a diferentes horas del día entre las 12 y 17:30 pm cuando el clima era soleado y no existían sombras que afecten a la imagen capturada. En la Figura 86 en la imagen Seg 2 se indica el resultado de segmentación para este caso. En esta imagen las líneas de división de los parqueaderos se ven de manera casi completa, excepto por un segmento grande en el centro hacia la izquierda de la imagen. Sin embargo existe mayor cantidad de manchas que para la prueba con clima nublado. A continuación la Tabla 9 expone los resultados obtenidos por el Mathscript de segmentación al ingresar la imagen.

Tabla 9.- Resultado Mathscript muestra “Sol sin sombra”

Estado Climático Sol sin sombra		
Cantidad	Ancho	Largo
2	39	4
	412	606
		1217

Fuente: Creación Propia

Como podemos observar la cantidad de parqueaderos es correctamente determinado. Ingresando estos datos en el programa de comparación, se obtienen las dos siguientes imágenes cortadas de la Figura 78. Estas se encuentran cortadas correctamente, es decir la segmentación se puede hacer durante un tiempo soleado si no existen muchas sombras sobre la imagen.

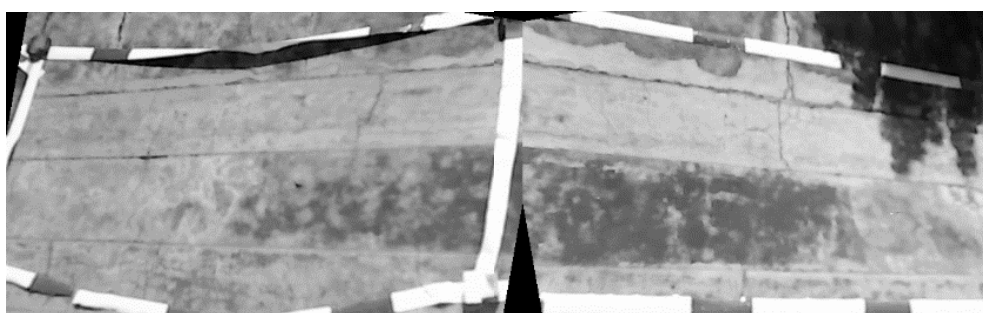


Figura 78.- Prueba de comparación clima sol sin sombra

Pruebas de clima con Sol con Sombra

Para esta prueba se corrió el programa de segmentación a diferentes horas del día entre las 8:00 y 11:30 am, cuando el tiempo era soleado y existían sombras que afecten a la imagen capturada. En la Figura 86, la imagen Seg 4 muestra un ejemplo de los resultados. En ésta, las líneas de división de los parqueaderos no se ven en la parte superior izquierda, donde se observa el contorno de la sombra. A continuación la Tabla 10 indica los resultados obtenidos por el Mathscript de segmentación al ingresar la imagen Seg4.

Tabla 10.- Resultado Mathscript muestra “Sol con sombra”

Estado Climático Sol con sombra		
Cantidad	Ancho	Largo
1	101	3

	401	603
--	-----	-----

Fuente: Creación Propia

La cantidad de parqueaderos está equivocada, por lo que se puede determinar que durante un tiempo soleado con muchas sombras, no se puede ejecutar correctamente la segmentación.

Pruebas de clima con Lluvia Ligera

Para esta prueba se corrió el programa de segmentación durante una lluvia ligera. En la Figura 87 del Anexo B se muestra un ejemplo de los resultados. En esta imagen las líneas de división de los parqueaderos no se distinguen en la esquina inferior derecha, además no se observan manchas sobre la imagen. La Tabla 11 refleja los resultados obtenidos del Mathscript.

Tabla 11.- Resultado Mathscript muestra "Lluvia"

Estado Climático Lluvia Ligera		
Cantidad	Ancho	Largo
2	62	16
	415	587
		1260

Fuente: Creación Propia

Como se ve, la cantidad de parqueaderos es correcta. El resultado después de ingresar al programa de comparación se indica en la Figura 79, donde se observa que las imágenes están cortadas correctamente. En tal virtud, se puede concluir que bajo estas condiciones de lluvia ligera, la segmentación funciona correctamente.

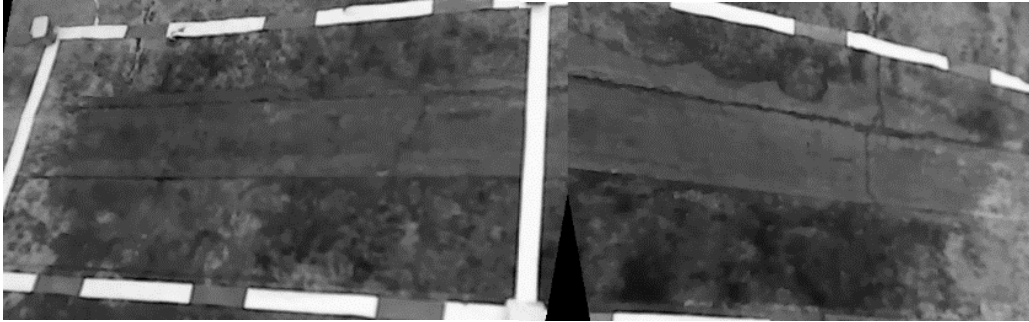


Figura 79.- Prueba de comparación muestra “Lluvia ligera”

Pruebas durante tiempo con Neblina

Para esta prueba, se corrió el programa de segmentación utilizando una máquina de humo que generó condiciones parecidas a las de un día con neblina. La imagen se demuestra en la Figura 88. (Anexo B) donde no se distinguen las líneas de los parqueaderos y la presencia de manchas. En la Tabla 12 se indican los resultados obtenidos por el Mathscript.

Tabla 12.- Resultado Mathscript muestra "Neblina"

Estado Climático Neblina		
Cantidad	Ancho	Largo
36	12	1
	479	31
		...

Fuente: Creación Propia

Como era de esperarse, el proceso de segmentación arrojó resultados negativos relacionados a la cantidad de parqueaderos. Esto pudo determinar que en condiciones de neblina el sistema no responde favorablemente.

Pruebas en horas de la tarde

Para esta prueba, se corrió el programa de segmentación a diferentes horas de la tarde entre las 17:45 y 18:30 pm, donde existía poca iluminación natural. La última imagen (Seg 3) de la Figura 86 muestra un ejemplo de los resultados. En ésta, las líneas de división se ven segmentadas, ya que el color azul no se detecta correctamente por la falta de luz. De igual manera no se percibe manchas en la imagen. En la Tabla 13 se encuentran los resultados del programa de segmentación para este caso.

Tabla 13.- Resultado Mathscript muestra "Tarde"

Estado hora del día "Tarde"		
Cantidad	Ancho	Largo
2	36	18
	377	620
		1243

Fuente: Creación Propia

Ingresado estos datos en el programa de comparación, se obtuvo las dos siguientes imágenes cortadas en la Figura 80. Como se aprecia las imágenes están cortadas correctamente, es decir la segmentación se puede hacer durante la tarde.



Figura 80.- Prueba de comparación muestra “Tarde”

Pruebas durante la Noche

Esta prueba se corrió a diferentes horas de la noche entre las 18:45 y 21:00 pm con el programa de segmentación. Este entorno se presenta sin luz solar y el resultado fue una imagen completamente oscura (ver Figura 85).

En consecuencia, en la fase de segmentación, el factor que produce mayor problema es la sombra generada por el cambio de la luz solar sobre los objetos que se encuentran en el entorno del parqueadero; así como la neblina que no permite distinguir ninguna forma. Tomando en cuenta los casos donde la cantidad de parqueaderos fue reconocida exitosamente (dos) se sacó el valor promedio de ancho y largo como se resume en la Tabla 14.

Tabla 14.- Tamaño promedio en ancho y largo de los parqueaderos tomando en cuenta los casos con éxito (Nublado, Sol sin Sombra, lluvia, Tarde)

Tamaño promedio de parqueaderos		
Cantidad	Ancho	Largo
2	356	593
		637

Fuente: Creación Propia

Caracterización Sistema de Comparación

Estas pruebas se realizaron con el fin de demostrar la mejor opción para determinar si un parqueadero está ocupado o no, entre imágenes en escala de grises o en blanco y negro con detección de bordes. El entorno de prueba fue un día nublado.

Análisis utilizando imágenes en escala de grises

La

Figura 89 (Anexo B) muestra un ejemplo de las imágenes obtenidas al comparar un parqueadero vacío y un lleno con imágenes procesadas en escala de grises. Al observar la

Tabla 15, las variaciones que se obtienen al permanecer en un mismo estado son alrededor del 5.37% mientras que al pasar de un estado al otro de 23.5%. Por lo que, fácilmente resulta obtener una tolerancia capaz de determinar si existe o no un cambio de estado.

Tabla 15.- Variación promedio en muestra "nublado" utilizando imágenes escala de grises

Estado Anterior	Estado actual	Variación Promedio
Vacío	Vacío	0,0306
Vacío	Ocupado	0,235
Ocupado	Ocupado	0,0171
Ocupado	Vacío	0,235

Fuente: Creación Propia

Análisis utilizando detección de bordes en imágenes

Para estas pruebas, la

Figura 90 (Anexo B) muestra un ejemplo de las imágenes obtenidas al comparar un parqueadero vacío y un lleno, con las imágenes procesadas en blanco y negro después de realizar una detección de bordes.

Tabla 16.- Variación promedio en muestra "nublado" utilizando detección de bordes

Estado Anterior	Estado actual	Variación Promedio
Vacío	Vacío	0,0419
Vacío	Ocupado	0,0625
Ocupado	Ocupado	0,0275
Ocupado	Vacío	0,0625

Fuente: Creación Propia

En la Tabla 16 se observa las variaciones en los dos casos de estudio. Por un lado al permanecer en un mismo estado se determinó un porcentaje del 3.83%, mientras que al pasar de un estado al otro del 6.25%. En consecuencia se puede visualizar que la determinación de una tolerancia que establece si existe un cambio de estado es mucho más compleja. Esto se debe a que todo el fondo de la imagen es negro y los cambios sobre ella solo se producen en los pixeles blancos, lo que hace que la variación entre un mismo estado o un cambio de estado sea muy cercana. Por tal razón, la mejor alternativa es la comparación de imágenes en escala de grises.

Pruebas con lógica booleana con el estado actual de parqueadero

En estas pruebas para determinar si existe un cambio de estado, se tomó una tolerancia del 15% observando buenos resultados, ya que si un parqueadero cambiaba de estado se lo determinaba correctamente. Sin embargo, cuando un parqueadero estaba vacío y un auto empezaba a estacionarse, parte del vehículo se encontraba en el área de estacionamiento y la cámara capturaba esta imagen. En ocasiones, se determinaba un cambio de estado registrándole al parqueadero como ocupado, pero más adelante cuando el vehículo se estacionaba completamente, volvía a contabilizarse como cambio de estado. Es así que, pese a que el vehículo se encontraba en el parqueadero el programa registraba al estacionamiento vacío a pesar de estar ocupado.

Lo mismo ocurría cuando un vehículo salía de un estacionamiento y en un inicio el programa lo reconocía como lleno. Si al momento de salir, la imagen capturaba una porción del vehículo afuera y una adentro, el programa cambiaba de estado. Por lo que al salir el vehículo completamente, el espacio se encontraba ocupado pese a estar vacío. El mayor de los problemas era que una vez que esto sucedía, se generaba un error que permanecía hasta reiniciar el programa, por lo que los datos ya no eran confiables. En tal virtud, era necesario buscar otra solución. Teniendo en cuenta la tabla de verdad descrita en Tabla 2 se obtuvo la ecuación lógica (12).

$$Y = A\bar{B} + \bar{A}B \quad (12)$$

Donde “A” es el Estado Actual, “B” el Cambio de estado y “Y” el Nuevo estado.

Pruebas con lógica booleana con el estado actual y estado anterior de parqueadero

Para evitar los errores antes mencionados, se decidió tomar en cuenta el estado anterior y el estado actual del parqueadero. Por lo que, si el programa en una iteración

detectaba un cambio de estado de ocupado a vacío o viceversa, el parqueadero era registrado con un nuevo estado. Pero si en la siguiente iteración, volvía a detectarse un cambio de estado, este no se registraba en el programa. En estas pruebas también se utilizó una tolerancia del 15% y los resultados fueron mejores, ya que no se introducían cambios de estado incorrectos al programa. La ecuación lógica resultante a partir de la Tabla 3 se muestra en la ecuación (13).

$$Y = \bar{A}\bar{B}\bar{C} + A\bar{B} + AC \quad (13)$$

Donde “A” es el Estado Actual, “B” el Cambio de estado actual “C” el Cambio de estado anterior y “Y” el Nuevo estado.

Pruebas con segmentación del 100% y 60% de la imagen del parqueadero

Todas las pruebas anteriores de comparación, arrojaron resultados positivos con imágenes recortadas al 100%. Sin embargo debido a la inclinación que tiene la cámara al moverse con el servo, la parte delantera y trasera del auto en ocasiones aparecían en la imagen recortada de los parqueaderos vecinos, afectando así el resultado. Dependiendo de la tolerancia seleccionada, la luminancia del día y el color del carro entre otras, podría provocar un cambio de estado en los parqueaderos adyacentes produciendo así un error no deseado. Por esta razón se decidió utilizar solo un 60% de la imagen recortada del parqueadero en su parte central, evitando así, que los estacionamientos adyacentes sean afectados por un cambio de estado en un parqueadero. En las Figuras

Figura 91 y

Figura 92 del Anexo B se muestran la comparación de las imágenes segmentadas a un 100% y 60%.

Para estar seguros de cuanto afecta este cambio en las variaciones, la Tabla 24 del Anexo B presenta la comparación de los resultados del indicador en imágenes a un 60% y 100%. Se pudo observar que las variaciones al 60% son ligeramente mayores con respecto al otro. Por lo que es factible la utilización de este parámetro para así evitar los problemas de interferencia entre el cambio de estado de un parqueadero a otro.

Pruebas de confiabilidad del sistema de Comparación

En este análisis, se tomó en cuenta los resultados de tamaño de parqueadero de la Tabla 14, a pesar de que las muestras “Neblina” y “Sol con Sombra” de la fase de segmentación no generaron buenos resultados. Entre los datos que se obtuvieron en el análisis de tiempo fue que el sistema propuesto tiene limitaciones en cuanto a las horas de funcionamiento. Es por eso que, como se observa en la Figura 85 (Anexo B) correspondiente a la toma de las 19:00 pm, el programa no es capaz de adquirir ninguna imagen por falta de luz que requiere la cámara.

Pruebas de Variación del indicador en diferentes horas del día y estado climático

Las fotos de la fase de comparación que generaron particularidades son las resultantes de la muestra “Sol con sombra” y “Medio día” de los análisis clima y tiempo respectivamente. La

Figura 93 (Anexo B) muestra las dos fotos que son tomadas una seguida de la otra de la misma muestra “Sol con Sombra”. Aquí, se distingue como el rápido e inesperado cambio de la posición de la sombra, altera los resultados tanto en las dos fases: segmentación (ver Figura 86) y comparación (ver Figura 94). Asimismo con respecto a la

Figura 93, la parte alumbrada de la foto de la izquierda, tiene mayor intensidad frente a la del lado derecho. Por otro lado, la Figura 95 (Anexo B) donde se analiza la muestra “Medio Día” se observa como la luz solar reflejada en el auto, cambia la intensidad en la imagen, siendo estas tomadas una tras otra. Esto nos permite concluir que siendo la misma muestra y una predecesora de la otra, las condiciones de posición e intensidad del sol son altamente cambiantes.

Para los resultados relacionados a la variación promedio de la diferencia de imágenes en la fase de comparación, se recolectó dos datos de cada muestra y por cada subgrupo de las variables P3 y P4 (ver Tabla 6). Los resultados del análisis de tiempo se muestran en la Figura 81 y Tabla 25 (Anexo B) así como los del análisis de clima en la Figura 82 y la Tabla 26 (Anexo B). Con estos datos se puede observar que las muestras “Media Mañana” y “Sol con Sombra” son las que presentan mayor fluctuación del porcentaje de variación. La muestra tomada en lluvia arrojó un porcentaje mayor del 83% de variación cuando no existe cambio de estado. De igual manera, las muestras “Sol sin Sombra” y “Medio Día” por sus bajos porcentajes de variación y por la separación de las dos curvas que reflejan un cambio de estado, presentan las condiciones más favorables a las que responde nuestro sistema.

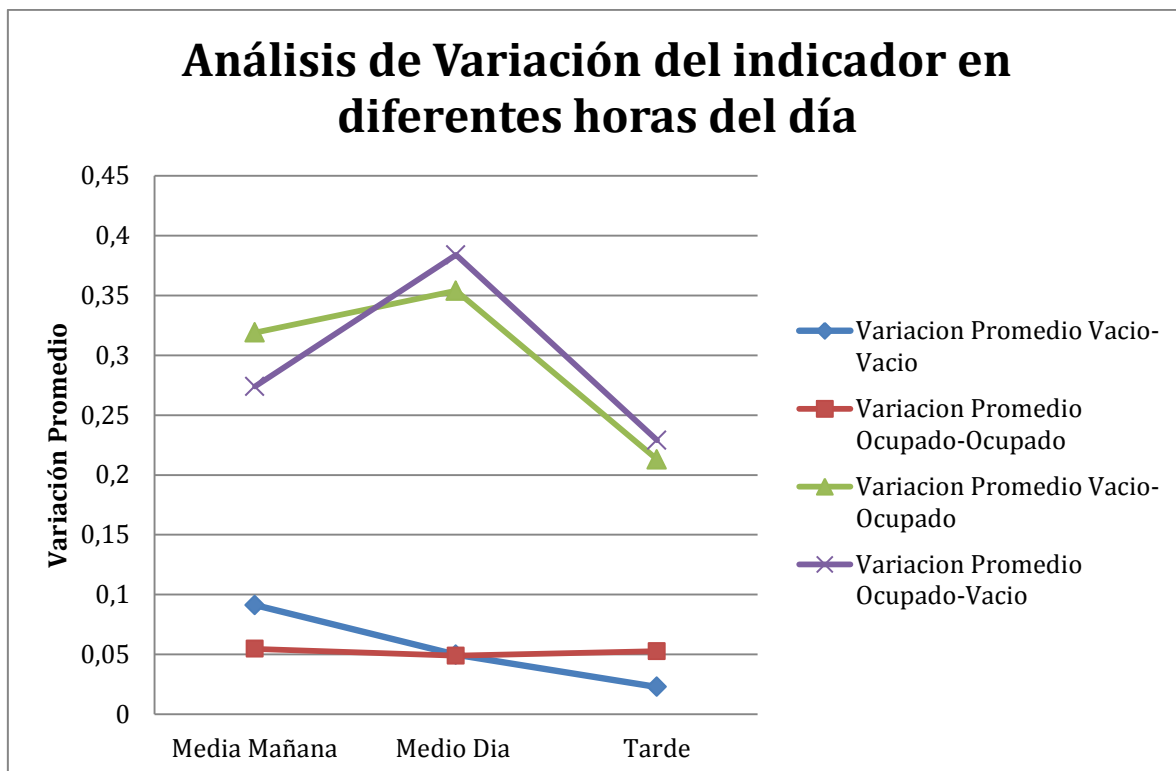


Figura 81.- Análisis de Variación del indicador en diferentes horas del día

Fuente: Creación Propia

El análisis tanto de tiempo como de clima se realizó en muestras distintas, sin existir una dependencia entre ellas. Sin embargo teniendo en cuenta las condiciones climáticas actuales de la ciudad de Quito, se observó una relación entre las dos. Por un lado, en el análisis de tiempo el pico más alto en todos los estados del parqueadero se encuentra en la muestra “Medio día”. De igual forma, en el análisis de clima los valores más altos del indicador se localizan en la muestra “Sol sin sombra”. Consecuentemente la relación ocurre debido a que, en la mayoría de casos al medio día en la ciudad de Quito, existe presencia de sol. Además su posición perpendicular no genera sombra como en nuestro caso de estudio, haciéndole a este el escenario con mejores resultados. Lo que no sucede con la muestra “neblina” que es la que presentó los peores resultados (ver Figura

82), ya que ni en la fase de segmentación se pudo conseguir la división correcta de los parqueaderos.

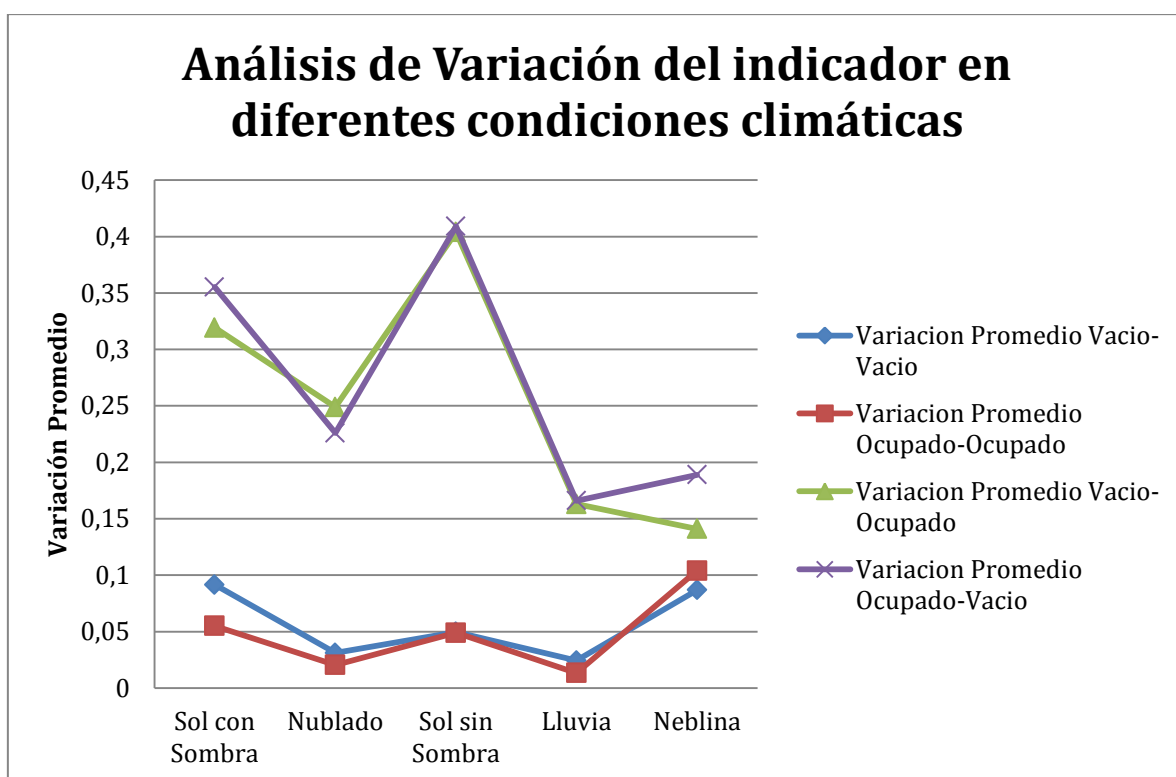


Figura 82.- Análisis de Variación del indicador en diferentes condiciones climáticas

Fuente: Creación Propia

Finalmente, para determinar los límites del rango de tolerancia del sistema, en cada uno de los análisis, se tomó los valores mínimos y máximos de las muestras, con excepción de “neblina”. Ya que esta acción abarca la mayor cantidad de variaciones del indicador en todos los casos de análisis.

Tabla 17.- Límites rango de tolerancia del sistema

Límite	Valor
Límite máximo superior	0,4090
Límite mínimo superior	0,1630

Límite máximo inferior	0,0915
Límite mínimo inferior	0,0133

Fuente: Creación Propia

La Figura 96 (Anexo B) resume por separado los límites para el estudio de tiempo y clima. De igual forma los rangos finales de funcionamiento de nuestro sistema se resumen en la Tabla 17, así como el porcentaje de variaciones que determina si existe un cambio o no (ver Tabla 18).

Tabla 18.- Porcentaje de Variación del sistema frente al cambio o no de estados

	Porcentaje de Variación
Cambio de estado	8,12%
Sin cambio de estado	46,23%

Fuente: Creación Propia

Optimización Sistema de Comunicación MyRIO servidor SQL

Pruebas de generación del archivo de actualización en formato .xls y .txt

Las pruebas efectuadas con este tipo de archivo, si bien generaron menor procesamiento al correr el programa en LabVIEW (ver Figura 97) nos dimos cuenta que por parte del servidor SQL, la comunicación se complicaba. La forma de actualizar la información en el servidor, es mediante comandos en lenguaje sql. Por lo que apuntar a un archivo .xls y más a las columnas y filas específicas donde se escribía la información,

constituyó un esfuerzo innecesario, puesto que este procesamiento podía demorar el tiempo de consulta y respuesta a la interfaz del usuario.

Por otro lado, la alternativa de generar un archivo .txt como medio de actualización al servidor, fue una manera más segura, ya que al ser un archivo simple, ofrecía mayores alternativas de comunicación al momento de apuntarlo, sin tener problemas de autenticación o permisos de acceso. En la parte del programa de LabVIEW (ver Figura 57) fue necesario colocar un módulo que separare la información con un TAB, ya que en el texto plano no se reconoce espacios ni posiciones de la información enviada. Finalmente determinamos que la opción de generar un texto plano .txt como medio de actualización era la más idónea.

Resumen del Capítulo

Con los fundamentos teóricos y la metodología del proyecto establecidos, se realizaron las pruebas de confiabilidad, estabilidad y comprobación que el proyecto requería. En un inicio se determinó la mejor alternativa de cámara web, teniendo en cuenta la resolución y el ángulo de visión. A continuación, se realizó pruebas de apertura de cámara para determinar los tiempos de respuesta al tomar una foto, y pruebas en diferentes marcas de servomotores dependiendo del sentido y tipo de movimiento.

Paso siguiente fue la comprobación de la fase de segmentación y comparación, utilizando pruebas en diferentes condiciones climáticas y horas del día. Con esto se, determinó el tamaño promedio del parqueadero segmentado y las condiciones específicas donde el sistema responde de mejor manera. Se realizaron pruebas con diferentes tipos de imágenes (escala de grises y detección de bordes) así como diferentes tamaños de

segmentación (100% y 60%). Asimismo se evaluó el algoritmo de comparación teniendo en cuenta por un lado el cambio de estado actual y por otro, el cambio de estado anterior. Con esto se determinó un rango de validación que el sistema soporta y responde favorablemente. Finalmente se realizó pruebas para optimizar el sistema de comunicación entre MyRIO y servidor SQL generando archivos de actualización en formato .xls y .txt

CONCLUSIONES

Entre las conclusiones que se derivan del presente proyecto, hay que resaltar que la ciudad de Quito ante el crecimiento acelerado del parque automotor, no está aprovechando adecuadamente el uso inteligente de la información que la tecnología ofrece, por lo que, se debe fomentar la innovación, buscando estrategias que permitan la sostenibilidad de la ciudad en cuanto a tecnología se refiere. El Distrito Metropolitano de Quito debe agotar esfuerzos para mejorar las necesidades del tráfico y estar en el lineamiento del medio ambiente, especialmente en la baja de emisión de carbono que produce la búsqueda de un espacio libre al momento de parquear un vehículo, y con esto dar una mejor condición de bienestar a la población.

Referente al análisis de resultados para determinar el tipo de cámara a utilizarse, la resolución de mejor alternativa fue la de 640x480 pixeles debido a que, esta relación de dimensión era la que mejor se ajustaba al tamaño del parqueadero de la zona azul, cabe destacar que esta es la resolución óptima para nuestras pruebas ya que en caso de que existan variaciones en cuanto al tipo de cámara o la altura a la que se ubica la misma esta resolución podría cambiar. Además por ser lo adecuadamente baja no complica el procesamiento de las imágenes y suficientemente alta que no deja de lado detalles importantes como las líneas de división de los parqueaderos, que son clave para la segmentación y comparación de las mismas. De igual forma, la mejor opción fue la Genius facecam 1000x por tener un ángulo de visión de 42.64° .

Asimismo es necesario informar, que las pruebas efectuadas en un espacio abierto ocasionaron varios problemas que aquellas realizadas en un ambiente controlado, especialmente con la apertura de la cámara. Para esto se utilizó el método Vision

Adquisition Express, que tenía un tiempo de espera para ajustarse a las condiciones de luz de 1.4 segundos, antes de empezar a capturar las imágenes correctamente.

Por otro lado para el sistema de movimiento rotacional, el servomotor que produjo los resultados esperados fue el Standard Hitec HS-31, ya que este permanecía estático en la posición deseada hasta recibir la orden de cambio. Esto permitía capturar las imágenes de manera óptima para que las mismas puedan ser posteriormente procesadas y comparadas.

Referente a la fase de segmentación, la muestra de sol y sombra fue con la que mejor resultados se obtuvo, ya que el sol permitía resaltar de excelente manera las líneas de división de los parqueaderos, y no con los problemas que se obtuvo con la muestra de sombras, como era de esperar. La muestra realizada con neblina fue la que peores resultados produjo, puesto que no permitió visualizar en absoluto las líneas de división de los estacionamientos, en tal virtud la segmentación que esta muestra produjo fue totalmente errónea. Es decir con todas las muestras realizadas el tamaño promedio de parqueadero fue del 356x615 pixeles.

En cambio en la fase de comparación se decidió utilizar una imagen en escala de grises, debido a que, con esta se obtenía una mayor diferencia entre las variaciones promedio el momento de determinar un cambio de estado en el estacionamiento. Al comparar las imágenes en blanco y negro donde solo existen dos posibles colores y tomando en cuenta que el fondo es negro, las variaciones producidas eran muy cercanas, cuando se producía o no un cambio de estado, por lo que al comparar este tipo de imágenes se corría un gran riesgo de tener presencia de errores, además el cálculo de la tolerancia se tornaba muy complejo.

En lo que se refiere al algoritmo de comparación, se observó que si tomamos en cuenta el cambio de estado actual puede generar un error al momento de estacionar un

vehículo o retirarlo, y este afectar en los siguientes resultados; por lo que, el algoritmo que utiliza el cambio de estado actual y el anterior logra eliminar el problema sin complicar el algoritmo. Para el tamaño de la imagen a comparar se determinó que lo mejor era utilizar solo un 60 % de la parte central del parqueadero, ya que al manejar la totalidad del parqueadero para la comparación, existía incidencia de los parqueaderos contiguos sobre los resultados de la misma.

En las pruebas de eficiencia de la fase de comparación, se encontró que la muestra sol sin sombra y medio día, son las condiciones en las que el sistema funciona de mejor manera. Asimismo la fase de segmentación en la muestra neblina generó los peores resultados. La relación que se hizo entre los resultados del análisis de tiempo y clima descubrió que en el medio día existe un clima de sol sin sombra debido a la ubicación geográfica de la ciudad de Quito.

Por otro lado la relación de las curvas de cambio o permanencia de estado mantienen su proporción, destacando el 46 % de variación como porcentaje mayor en el de permanencia de estado frente al 8% cuando existe un cambio de estado, esto nos permite indicar que es óptimo nuestro sistema cuando el rango de variación para cambio de estado es bajo. Se determinó también que cuando no existe un cambio de estado, las variaciones promedio fluctúan entre 0.013 y 0.092, mientras que, cuando existe un cambio de estado estas fluctúan entre 0,163 y 0,409. Por lo que se concluye que la tolerancia de 0,15 que se utilizó para las pruebas era lo correcto. Por último en la fase de comparación se determina que es necesario que exista cierto nivel de luminosidad para permitir la correcta comparación, este nivel de luz es el que existe generalmente entre las 8:00 y 18:00 horas, que es el horario de funcionamiento de la zona azul en Quito.

En el lado de la comunicación se encontró que el archivo de actualización de la base de datos SQL debe ser en texto plano, ya que la compatibilidad entre el servidor SQL y el archivo de texto plano es mayor que la relación con un archivo Excel. Además por efecto de minimizar costos se destaca que se utilizó software libre para la creación del servidor SQL y la página web.

Finalmente, hay que mencionar que a pesar de que el proyecto tiene resultados favorables, existen algunas recomendaciones que se pueden hacer con el objeto de mejorar la viabilidad para masificar el proyecto a futuro. Primero se requiere ubicar la cámara a una mayor altura y con un ángulo de visión más amplio con la finalidad de aumentar el campo de visión y la cantidad de parqueaderos a ser comparados con una sola cámara. Segundo, los altos costos que representan la utilización de un hardware como es MyRIO 1900, y con el objeto de producir a mayor escala el proyecto, los procesos de control de cámara, servo, segmentación y comparación deben trasladarse a un hardware de menor costo como un FPGA o un DSP. Sea cual sea el hardware que utilicemos para la segmentación y comparación una buena alternativa es alimentarlo mediante el uso de paneles solares debido a la complicación de alimentar al dispositivo que se encuentra a tal altura y el costo que esto implicaría.

Por otro lado ya existen en el mercado opciones que solucionan el problema de la detección del estado de los parqueaderos como la ya mencionada SMART PARKING que funciona mediante el uso de sensores sobre el pavimento que permiten detectar la presencia o no de un vehículo, esta solución ya está siendo implementada en algunas ciudades de Inglaterra con muy buenos resultados. Sin embargo en cuanto a costos esto representaría un problema en un país como el nuestro donde tendríamos que importar ya sea los sensores armados o las piezas para armar los sensores. Por esta razón una solución

viable sería controlar los servomotores con arduinos y transmitir la información obtenida por las cámaras a un servidor que se encargaría de hacer la segmentación y comparación de las imágenes no solo de una sino de varias cámaras a la vez, de esta forma podríamos reducir costos y aumentar la eficiencia y confiabilidad del sistema.

REFERENCIAS

- AEADE. (2015). AEADE Asociación de Empresas automotrices del Ecuador. Retrieved 03 05, 2015, from http://www.aeade.net/web/index.php?option=com_content&view=article&id=145&Itemid=80
- Alan V. Oppenheim, A. S. (1998). Señales y Sistemas . Prentice Hall Hispanoamericana.
- Allison, R. (2006). The Boston Massacre. Massachusetts: Series Editor.
- Allison, R. J. (2007). The Boston Tea Party. Massachusetts: Commonwealth Editions.
- Bell Labs. (2013). Experimental Verification of the Charge Couple Device Concept.
- Berkin/Miller. (2008). Making America A history of the United states. Boston: Houghton Mifflin Company.
- Burnett, B. (2004). The Continetal Congress. New York: The rosen Publishing Group.
- Burt, J. (2014). Intel-Partners Show Off Range of Chip Maker's IoT Technologies., (pp. Eweek, 1.).
- Censo, I. N. (2011). Resultados del Censo 2010 de Población y Vivienda en el Ecuador. Quito.
- Colonial Hilliamsburg. (2010, Febrero). A summary of the 1765 Stamp Act . Retrieved 03 14, 2014, from <http://www.history.org/history/teaching/tchcrsta.cfm>
- Cooklev, T. (2004). Wireless Communication Standards: A Study of IEEE 802.11, 802.15, 802.16. San Francisco: IEEE.
- Danaraj, J. (2014). The Internet of Evrything for Cities. Toward new models for Smart Communities. Washington DC.
- David Money Harris, S. L. (2013). Digital Design and Computer Architecture. Elsevier.
- Edward, C. (2013). New Way of working are needed to make smart cities a reality. Engineering & Technology.
- Edwards, C. (2013). NEW WAYS OF WORKING ARE NEEDED TO MAKE SMART CITIES A REALITY. Engineering & Technology 17509637, 8(11), 24-25.
- Ekstrom, M. P. (1984). Digital Image Processing Techniques. Orlando: Elsevier.
- Ellis, G. E. (1975). Battle of Hill. George.

- FAULKNER, M. C. (2014). Community Sense and Response Systems: Your Phone as Quake Detector. *Communications Of The ACM*. (doi:10.1145/2622628.2622633), 66-75.
- Firoozian, R. (2008). *Servo Motors and Industrial Control Theory*. New York: Springer.
- Funck S, M. N. (2004). Determining car-park occupancy from single images. *International Symposium on Intelligent Vehicles*, 325–328.
- G. Bermudez, J. R. (2003). Protocolo TCP/IP. Retrieved Noviembre 11, 2014, from <http://protocolotcpip.galeon.com/>
- INEC. (2013). *Anuario de Estadísticas de Transporte 2013*. Quito.
- Jähne, B. (2005). *Digital Image Processing*. New York: Springer-Verlag.
- Jedson, L. (2009). *The Treaty of Paris, 1783*. New York: The Rosen Publishing Group.
- Jeffrey Travis, J. K. (2006). *LabVIEW for Everyone*. Prentice Hall.
- Kight, P. (2003). *Conspiracy Theories in America History*. Santa Baraba.
- López, J. &. (2014). Universidad e innovación hacia la ciudad inteligente. (Spanish). *De Obras Públicas* , 161(3550), 49-54.
- Mery, D. (2004). *Visión por Computadora*. Santiago de Chile: Universidad Católica de Chile.
- National Instrument. (2014). My Rio . Retrieved 11 10, 2014, from <http://sine.ni.com/nips/cds/view/p/lang/es/nid/211694>
- National Instruments . (2015). Software de Desarrollo de Sistemas NI LABVIEW. Retrieved 04 09, 2015, from <http://www.ni.com/labview/esa/>
- Nell, W. C. (1970). *The colored patriots of the American Revolution* . Boston.
- Quito, C. M. (2007, 10 22). Ordenanza Metropolitana Registro Oficial N195. Quito, Pichincha.
- Quito, C. M. (2012). Sesión Ordinaria . Acta 05-O de Consejo Metropolitano , (p. 20). Quito .
- Rafael C. Gonzalez, R. E. (2002). *Digital Image Processing* . New Jersey: Prentice Hall.
- Ramesh Jain, R. K. (1995). *Machine Vision*. Boston: McGraw - Hill .
- Richard Hartley, A. Z. (2003). *Multiple View Geometry in computer vision*. Cambridge: Cambridge University.

- Russell, D. L. (1995). *The American Revolution in the southern colonies*. New York: MacFarland Company Inc.
- Rutkin, A. (2014). Bright lights, smart city. *New Scientist*(223(2981)), 17.
- Simpson, P. (2010). *FPGA Design*. Toyota: Springer.
- Smart- Parking Latin America. (2012). *Smart-Parking*. Retrieved 11 10, 2014, from <http://my-smartparking.com/>
- Stephen Brown, Z. V. (2006). *Fundamentos de lógica digital con diseño VHDL*. México DF: Mc-Graw Hill.
- The history. (n.d.). *Twownshend Acts*. Retrieved 03 14, 2014, from <http://www.history.com/topics/american-revolution/townshend-acts>
- Tobin, S. M. (2010). *DC Servos : Application and Design with MATLAB®*. Boca Raton: CRC Press.
- Tossel, I. (2014). *The Rise of the Smart City*. Canadian Business.
- Villanueva, L. &. (2014). *Universidad e innovación hacia la ciudad intelignete*. *Revista de Obras publicas*.
- Vinay K. Ingle, J. G. (2012). *Digital Signal Processing using Matlab*. Cengage Learning.
- W. Genshun, S. X. (2014). *An Improved EZW Algorithm and Its Application in Intelligent Transportation Systems*. *Engineering Letters*, 22(2), 63-69.
- X. Ying, Y. B. (2014). *Study on the optimization algorithm of sediment particle Imshange*. *Journal Of Chemical & Pharmaceutical Research*.(6(7)), 481-490.

ANEXO A: ANTECEDENTES



Figura 83.- Variación Acumulada de ventas Vehiculares de Dic-2013 a Dic-2014 (AEADE, 2015)

Tabla 19.- Características de tarjeta de red inalámbrica NI MyRio1900

Característica	Parámetro
Modo de Protocolo	IEEE 802.11 b,g,n
Banda de Frecuencia	ISM 2.4 GHz
Ancho de Canal	20MHz
Canales	USA 1-11
TX Power	+10 dBm max (10mW)
Rango de Alcance	150 metros
Directividad de Antena	Omnidireccional
Seguridad	WPA,WPA2

Fuente: Creación Propia en base a Especificaciones técnicas NI myRIO 1900

Tabla 20.- Número de vehículos motorizados matriculados según provincias año 2010

Provincia	Total
Azuay	99.913
Bolívar	13.143
Cañar	40.092
Carchi	19.318
Chimborazo	43.511
Cotopaxi	55.015
El Oro	73.898
Esmeraldas	41.635
Galápagos	825
Guayas	437.138
Imbabura	45.860
Loja	41.579
Los Ríos	82.918
Manabí	147.807
Morona Santiago	7.692
Napo	5.461
Orellana	12.245
Pastaza	8.291
Pichincha	387.858
Santa Elena	13.598
Santo Domingo	32.693
Sucumbíos	21.087

Tungurahua	80.694
Zamora Chinchipe	5.615
TOTAL	1.717.886

Fuente: Creación propia con datos suministrados por la Agencia Nacional de Tránsito

ANEXO B: ANÁLISIS DE RESULTADOS

Tabla 21.- Parámetro de Imagen cámara Logitech C160

Logitech C160		
Resolución	Relación largo ancho	Frames por segundo
160x90	1.778	30,25,20,15,10,5
160x100	1.6	30,25,20,15,10,5
160x120	1.333	30,25,20,15,10,5
176x144	1.222	30,25,20,15,10,5
320x180	1.778	30,25,20,15,10,5
320x200	1.6	30,25,20,15,10,5
320x240	1.333	30,25,20,15,10,5
352x288	1.222	30,25,20,15,10,5
640x360	1.778	30,25,20,15,10,5
640x400	1.6	30,25,20,15,10,5
640x480	1.333	30,25,20,15,10,5

Fuente: Creación Propia

Tabla 22.- Parámetros de Imagen cámara Genius Facecam 1000x

Genius Facecam 1000x		
Resolución	Relación largo ancho	Frames por segundo
160x120	1.333	30,25,20,15,10,5
176x144	1.222	30,25,20,15,10,5
320x240	1.333	30,25,20,15,10,5
352x288	1.222	30,25,20,15,10,5
640x360	1.778	30,25,20,15,10,5
640x480	1.333	30,25,20,15,10,5
1280x720	1.778	30,25,20,15,10,5

Fuente: Creación Propia

Tabla 23.- Tiempos para captura de imagen vs. Métodos de apertura

Método de apertura	Tiempo para empezar a capturar correctamente
Método manual	2.1 s
Vision Adquisition Express	1.4 s

Fuente: Creación Propia



Figura 84.- Servomotor marca Standard Hitec HS-311



Figura 85.- Adquisición de imagen análisis tiempo a las 19:00pm muestra "Tarde"

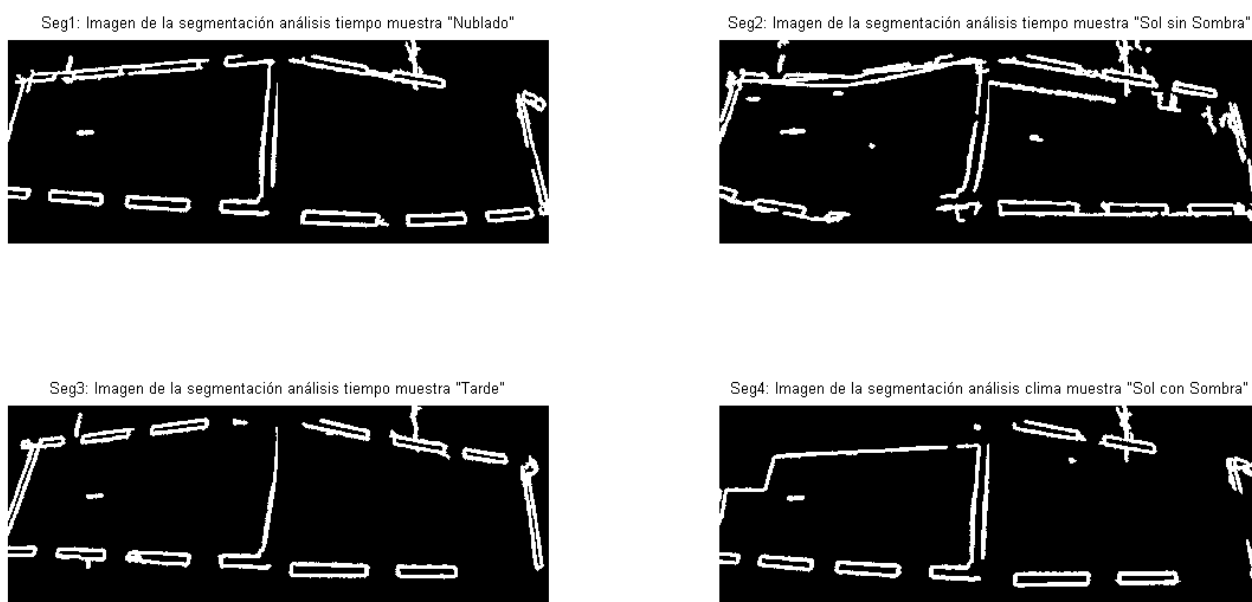


Figura 86.- Comparación análisis de segmentación

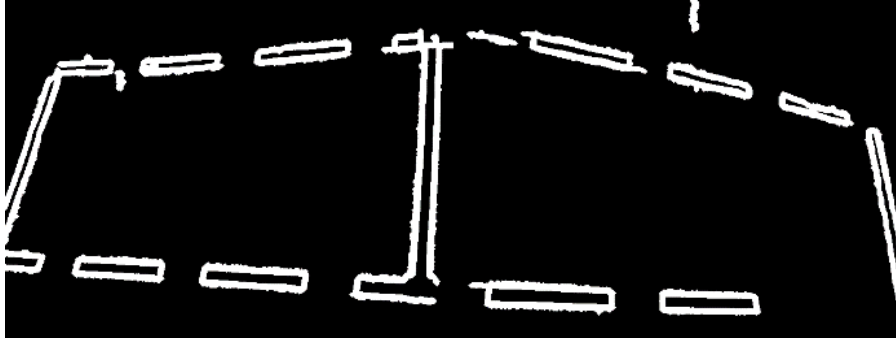


Figura 87. - Análisis de segmentación muestra "lluvia"

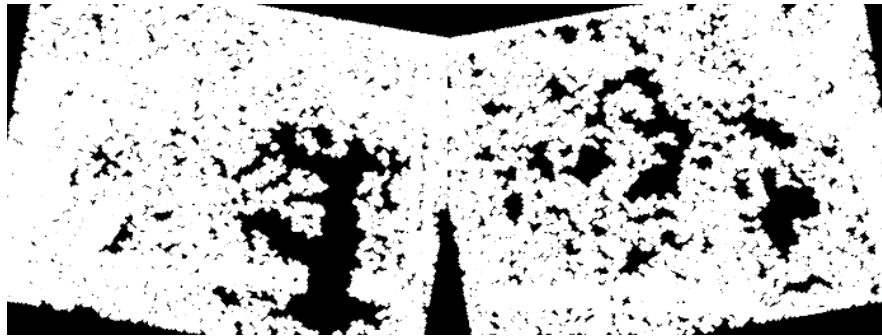


Figura 88.- Análisis de segmentación muestra "neblina"



Figura 89.- Comparación imágenes en escala de grises

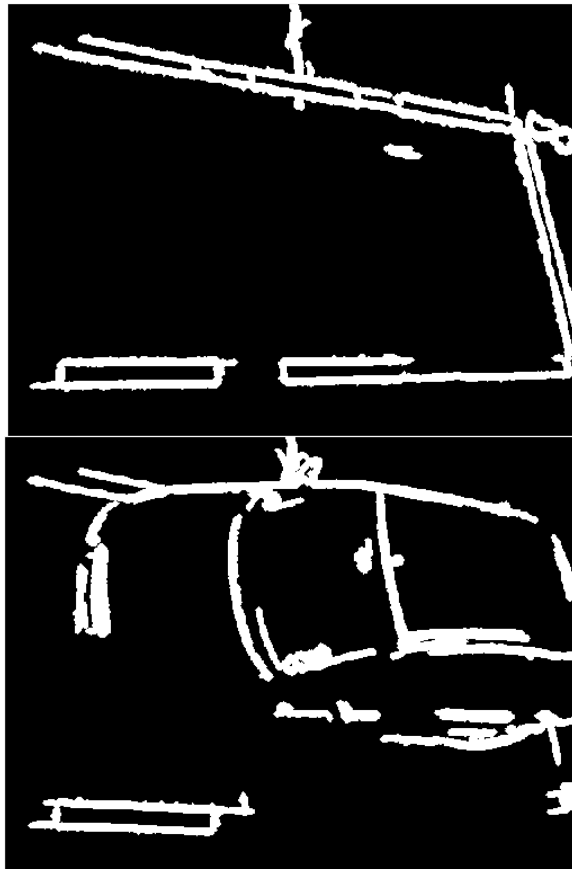


Figura 90.-Comparación de imágenes con detección de bordes



Figura 91.- Imágenes cortadas al 100%

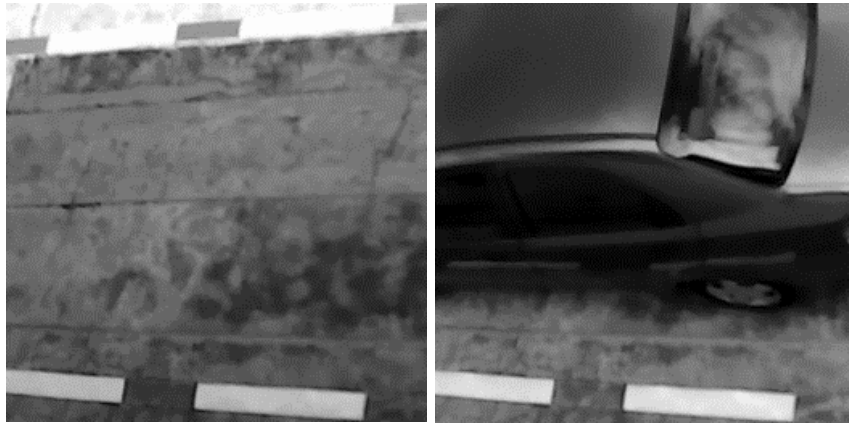


Figura 92.- Imágenes cortadas al 60%

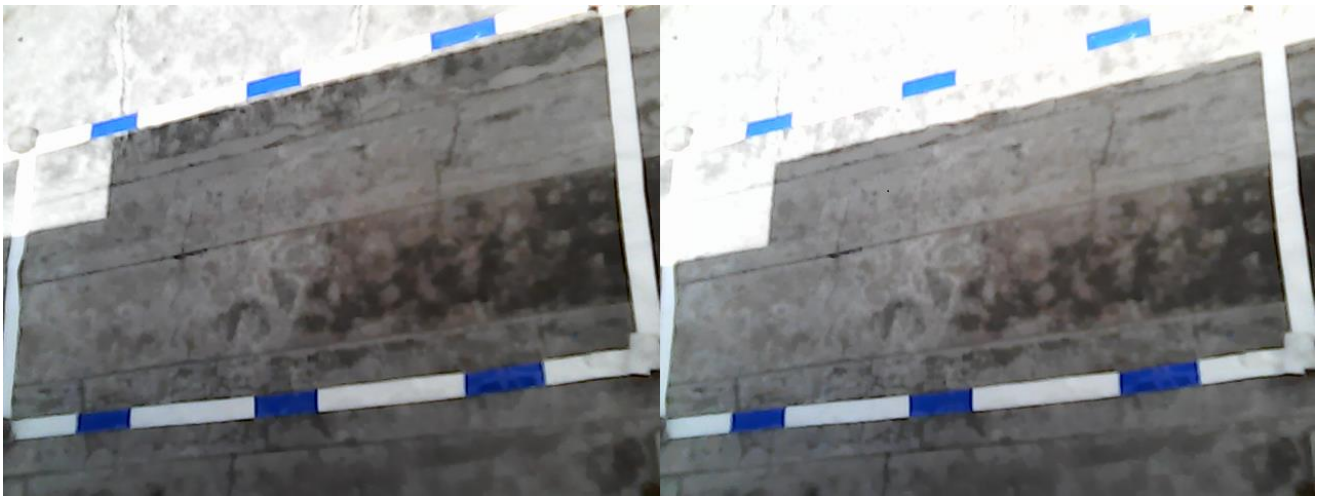


Figura 93.- Fase comparación análisis clima muestra "Sol con sombra"

Tabla 24.- Variación promedio de muestras (100% y 60%)

Estado Actual	Estado Nuevo	Muestra	Variación Promedio 100%	Variación Promedio 60%
Vacío	Vacío	Sol con Sombra	0,1050	0,0915
Vacío	Ocupado	Sol con Sombra	0,2870	0,3193
Ocupado	Ocupado	Sol con Sombra	0,0651	0,0550
Ocupado	Vacío	Sol con Sombra	0,2870	0,3553
Vacío	Vacío	Nublado	0,0306	0,0311
Vacío	Ocupado	Nublado	0,2350	0,2490
Ocupado	Ocupado	Nublado	0,0171	0,0204
Ocupado	Vacío	Nublado	0,2350	0,2260
Vacío	Vacío	Sol sin Sombra	0,0524	0,0497
Vacío	Ocupado	Sol sin Sombra	0,3560	0,4040
Ocupado	Ocupado	Sol sin Sombra	0,0130	0,0489
Ocupado	Vacío	Sol sin Sombra	0,0444	0,4090
Vacío	Vacío	Tarde	0,0240	0,0228
Vacío	Ocupado	Tarde	0,1960	0,2130
Ocupado	Ocupado	Tarde	0,0505	0,0526
Ocupado	Vacío	Tarde	0,1960	0,2130
Vacío	Vacío	Lluvia	0,0244	0,0243
Vacío	Ocupado	Lluvia	0,1590	0,1630
Ocupado	Ocupado	Lluvia	0,0129	0,0133
Ocupado	Vacío	Lluvia	0,1700	0,1660

Vacío	Vacío	Neblina	0,0826	0,0868
Vacío	Ocupado	Neblina	0,1410	0,1410
Ocupado	Ocupado	Neblina	0,0960	0,1040
Ocupado	Vacío	Neblina	0,1860	0,1890

Fuente: Creación Propia

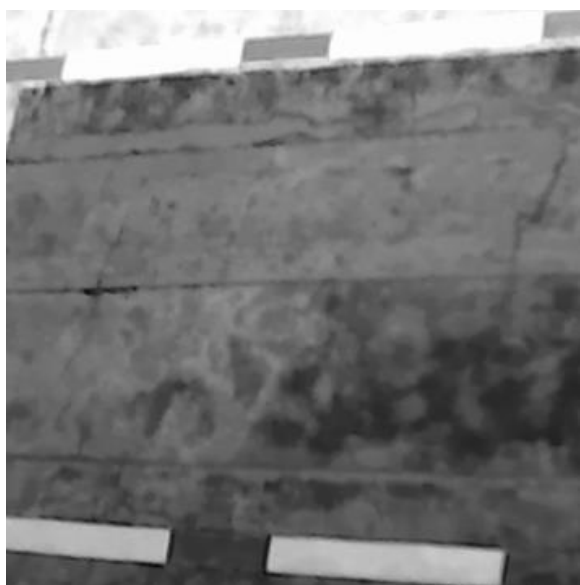


Figura 94.- Fase comparación 60% muestra "Sol con Sombra"



Figura 95.- Fase comparación análisis tiempo muestra "Medio Día"

Tabla 25.- Porcentajes de variación análisis tiempo

Muestra	% de variación con cambio de estado	% de variación sin cambio de estado
Media Mañana	12,42%	66,73%
Medio Día	7,81%	1,64%
Tarde	6,99%	56,65%
Promedio	10,41%	41,67%

Fuente: Creación Propia

Tabla 26.- Porcentajes de variación análisis clima

Muestra	% de variación con cambio de estado	% de variación sin cambio de estado
Sol con Sombra	10,13%	66,30%
Nublado	10,18%	52,45%
Sol sin Sombra	1,22%	1,64%
Lluvia	1,81%	82,71%
Promedio	5,38%	50,79%

Fuente: Creación Propia

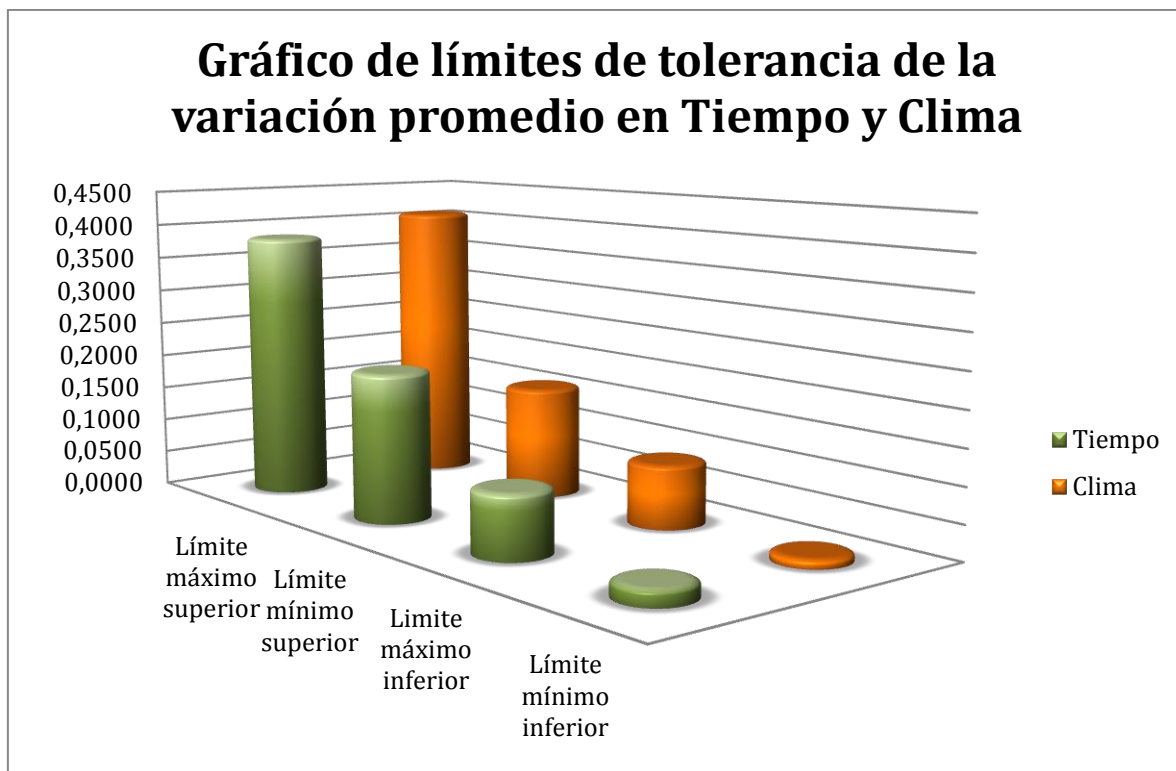


Figura 96.- Límites de tolerancia de la variación en Tiempo y Clima

Fuente: Creación Propia

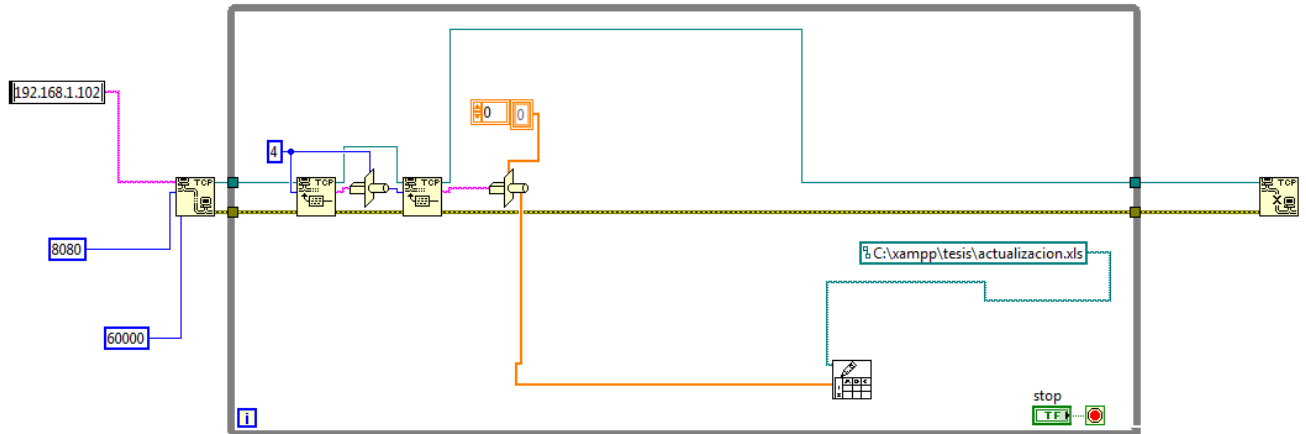
ANEXO C: PROGRAMAS DE PRUEBA EN LABVIEW

Figura 97.-Comunicación MyRIO-Servidor SQL en texto .xls

ANEXO D: PROGRAMA DE SEGMENTACIÓN

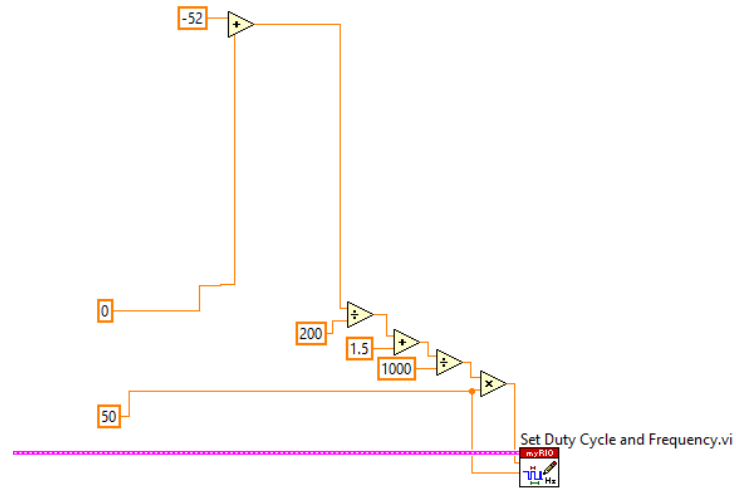


Figura 98.- Movimiento de servomotor lado izquierdo Programa Segmentación

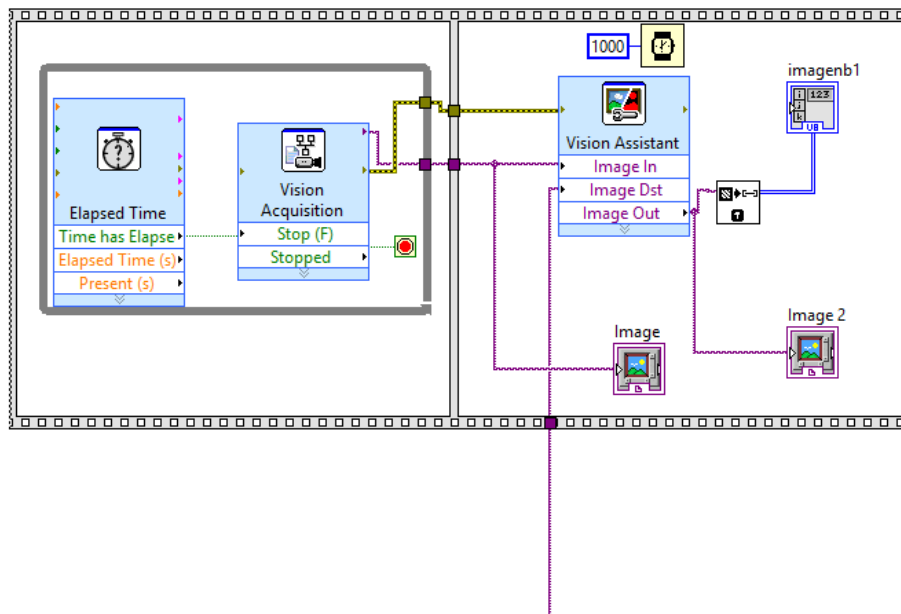


Figura 99.- Adquisición de imagen lado izquierdo - Programa de Segmentación

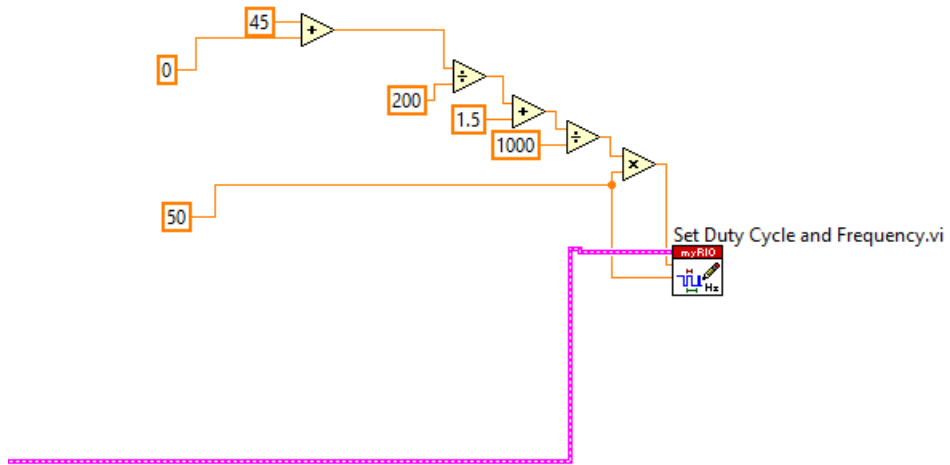


Figura 100.-Movimiento de servomotor lado derecho – Programa de Segmentación

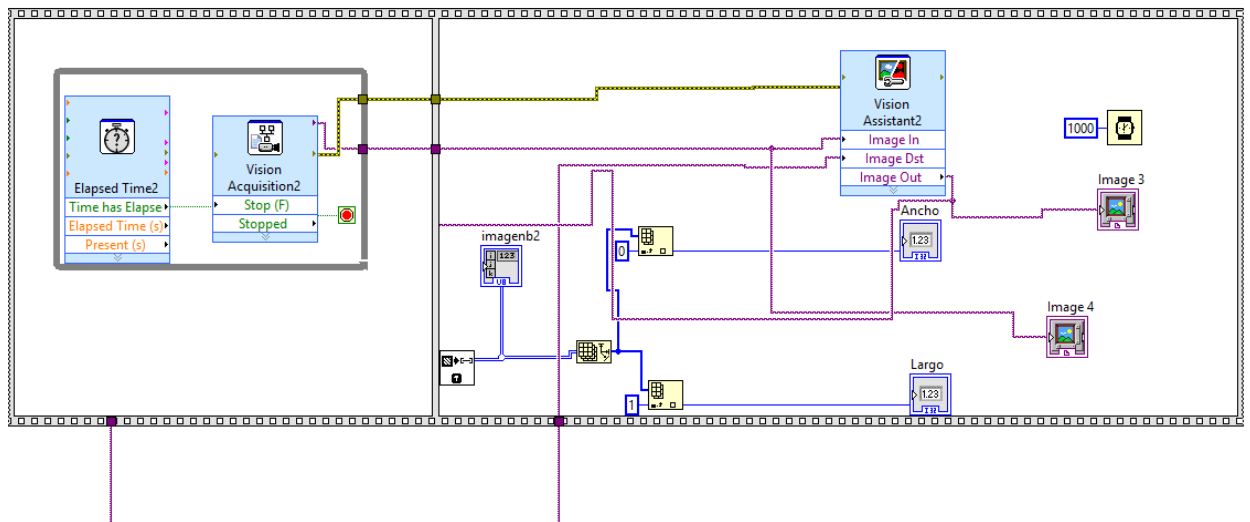


Figura 101.- Adquisición de imagen lado derecho - Programa Segmentación

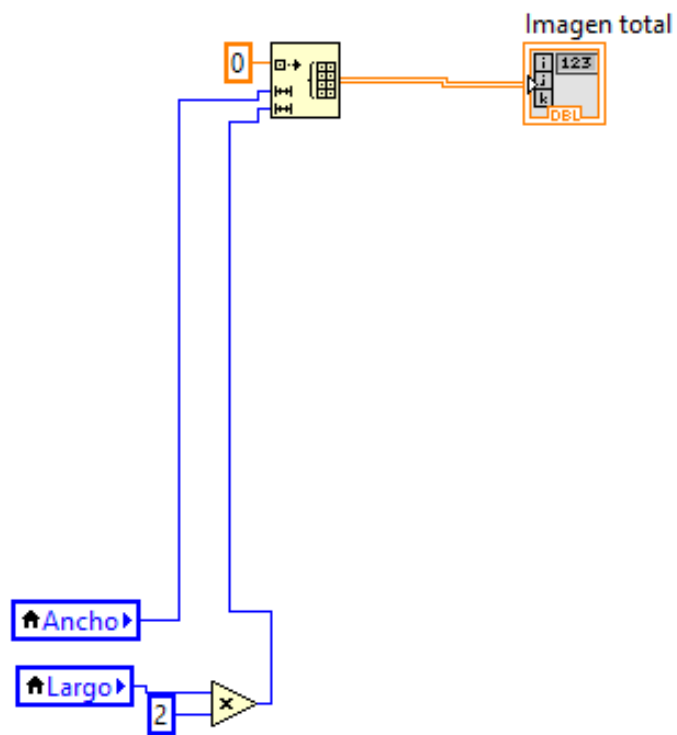


Figura 102.- Inicialización de arreglo imagen total - Programa Segmentación

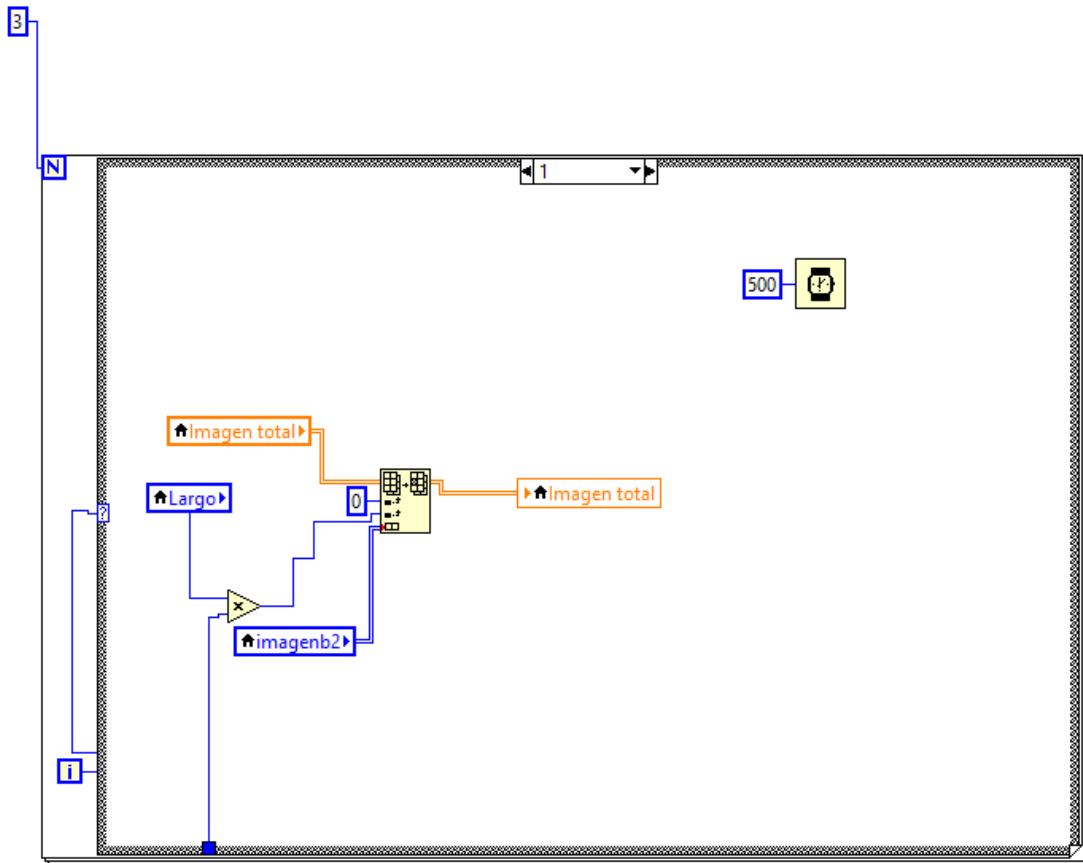


Figura 103.- Construcción de arreglo con imágenes antes capturadas - Programa Segmentación

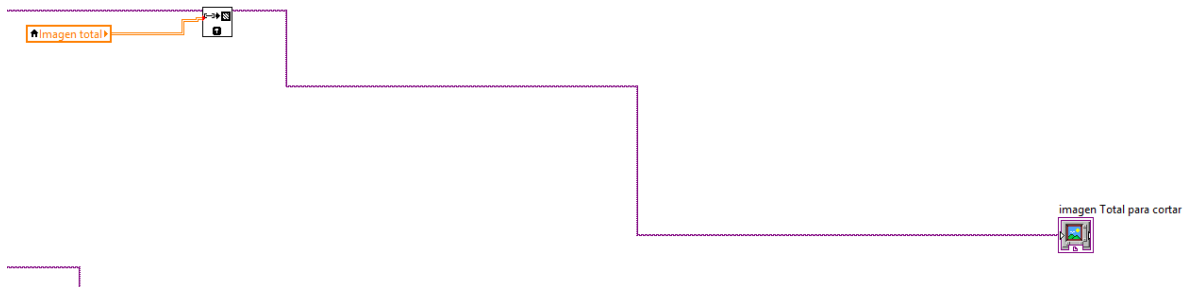


Figura 104.- Visualización arreglo Imagen Total - Programa Segmentación

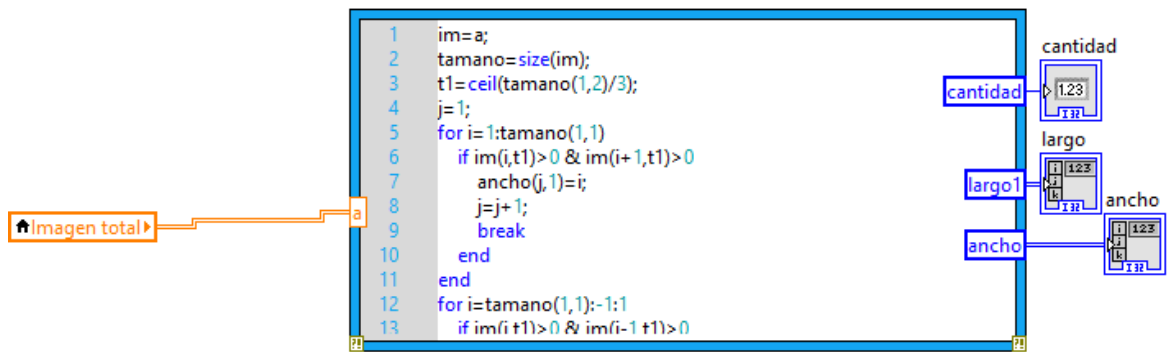


Figura 105.- Segmentación de arreglo Imagen Total -Programa Segmentación

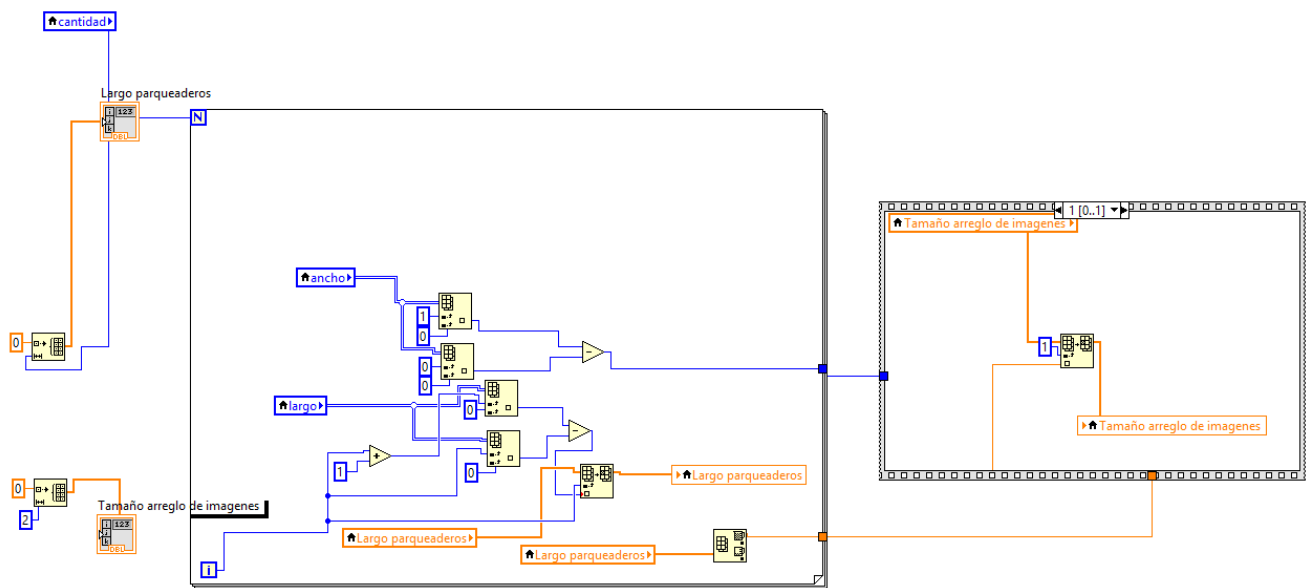


Figura 106.- Inicialización de arreglos donde se encuentran las medidas de los parqueaderos cortados (ancho y largo) - Programa Segmentación

```

1      %Programa de segmentación
2      %Proyecto de Tesis
3      % Universidad san Francisco De Quito
4      % P. García - Y. Macas
5      im=a;tamano=size(im);t1=ceil(tamano(1,2)/3);j=1;
6      for i=1:tamano(1,1)
7          if im(i,t1)>0 & im(i+1,t1)>0
8              ancho(j,1)=i;
9              j=j+1;
10             break
11         end
12     end
13     for i=tamano(1,1):-1:1
14         if im(i,t1)>0 & im(i-1,t1)>0
15             ancho(j,1)=i;
16             j=j+1;
17             break
18         end
19     end
20     imout=a(ancho(1,1):ancho(2,1),:);j=1;t2=ceil(ancho(1,1)+(ancho(2,1)-ancho(1,1))/2);
21     for i=1:tamano(1,2)
22         if im(t2,i)>0 & im(t2,i+1)>0
23             largo(j,1)=i;
24             j=j+1;
25             break
26         end
27     end
28     for i=tamano(1,2):-1:1
29         if im(t2,i)>0 & im(t2,i-1)>0
30             w=i;
31             break
32         end
33     end
34     q=largo(1,1);largo1(1,1)=largo(1,1);
35     while (q+40)<w
36         for i=q:tamano(1,2)
37             if im(t2,i)>0 & im(t2,i+1)>0 & (i-q)>40
38                 largo(j,1)=i;
39                 j=j+1;
40                 break
41             end
42         end
43         q=largo(j-1);
44     end
45     c=length(largo)-1;
46     for k=1:c
47         l(k,1)=largo(k+1,1)-largo(k,1);
48     end
49     v=sum(l)/c;q=largo(1,1);j=2;
50     while (q+40)<w
51         q
52         for i=q:tamano(1,2)
53             if im(t2,i)>0 & im(t2,i+1)>0 & (i-q)>ceil(v*0.6)
54                 largo1(j,1)=i;
55                 j=j+1;
56                 break
57             end
58         end
59         q=largo1(j-1);
60     end
61     largo1(j-1,1)=w;cantidad=j-2;

```

Figura 107.- Script de programa Segmentación en el Mathscript de LabVIEW

ANEXO E: PROGRAMA DE COMPARACIÓN

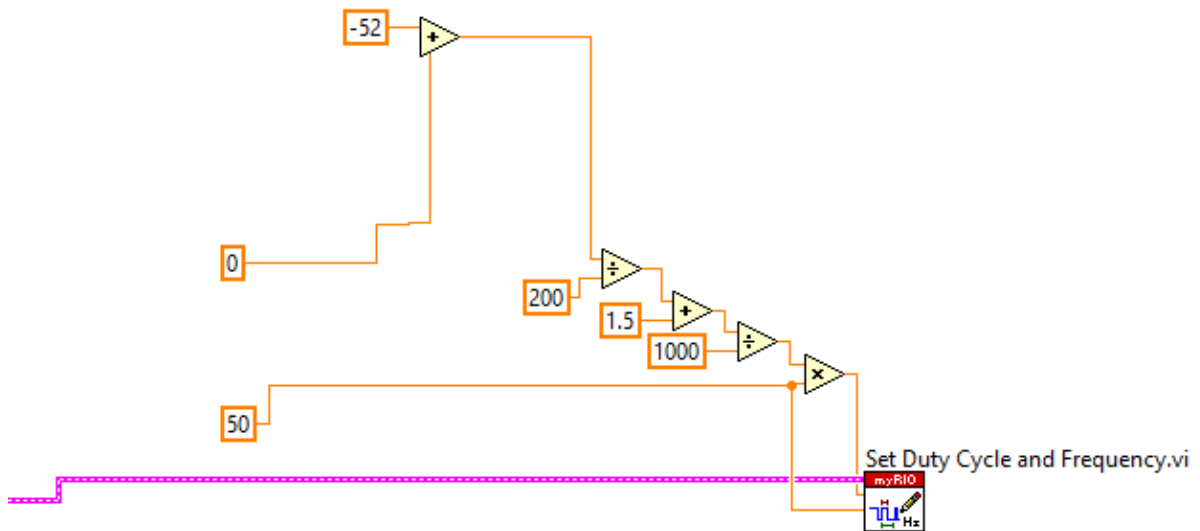


Figura 108.- Movimiento de servomotor lado izquierdo - Programa Comparación

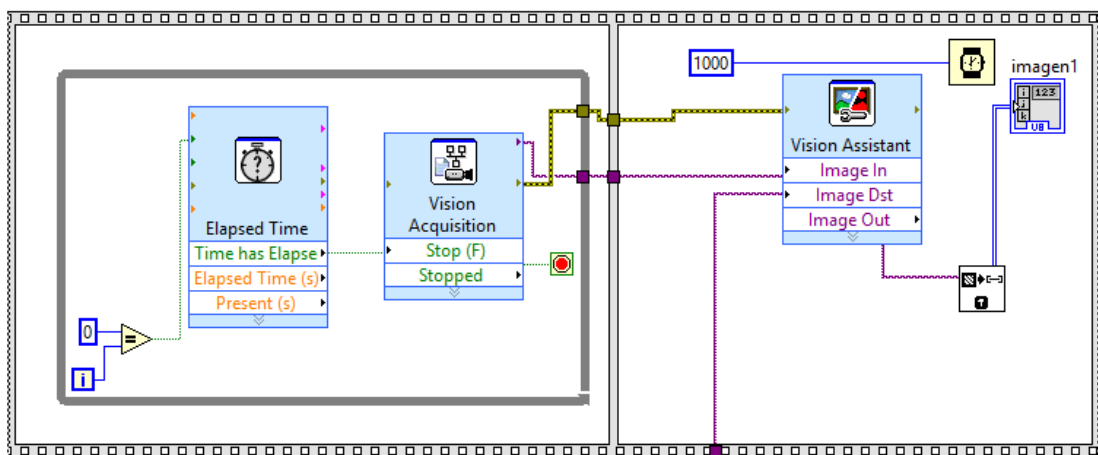


Figura 109.- Adquisición de imagen lado izquierdo - Programa de Comparación

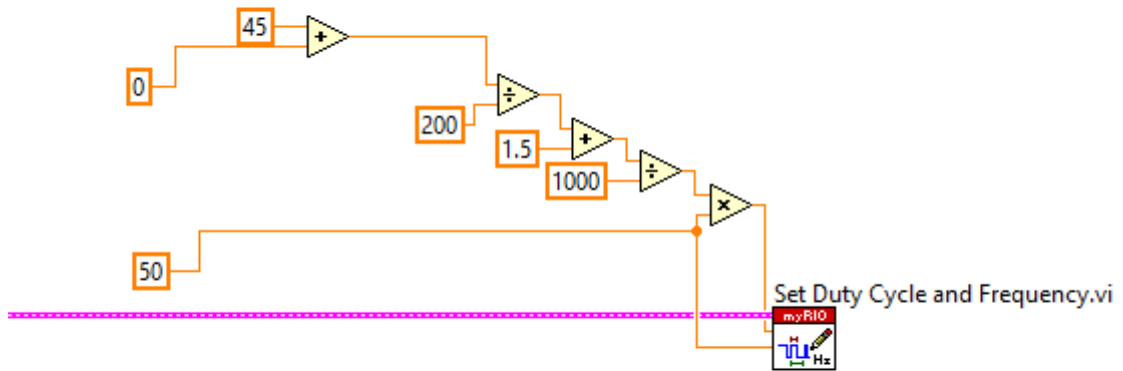


Figura 110.- Movimiento de servomotor lado derecho - Programa Comparación

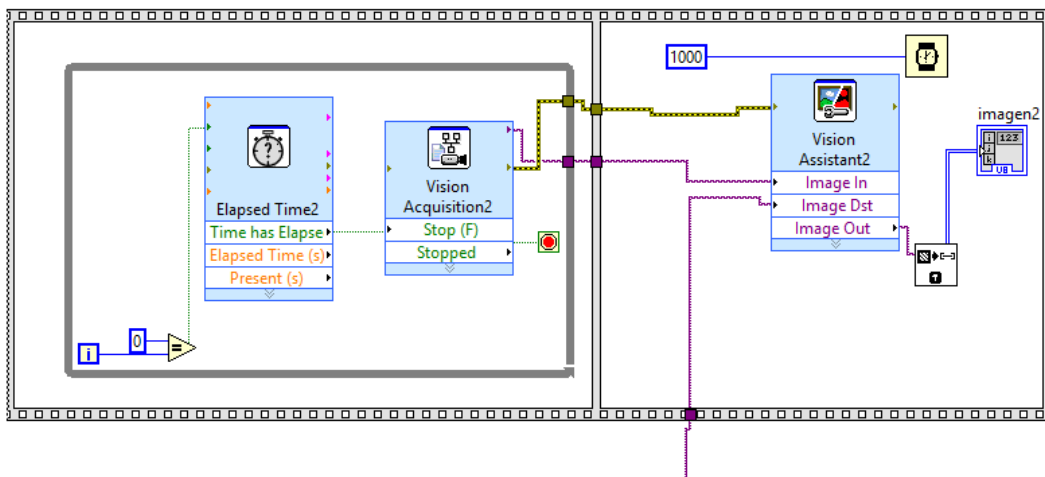


Figura 111.- Adquisición de imagen lado derecho - Programa de Comparación

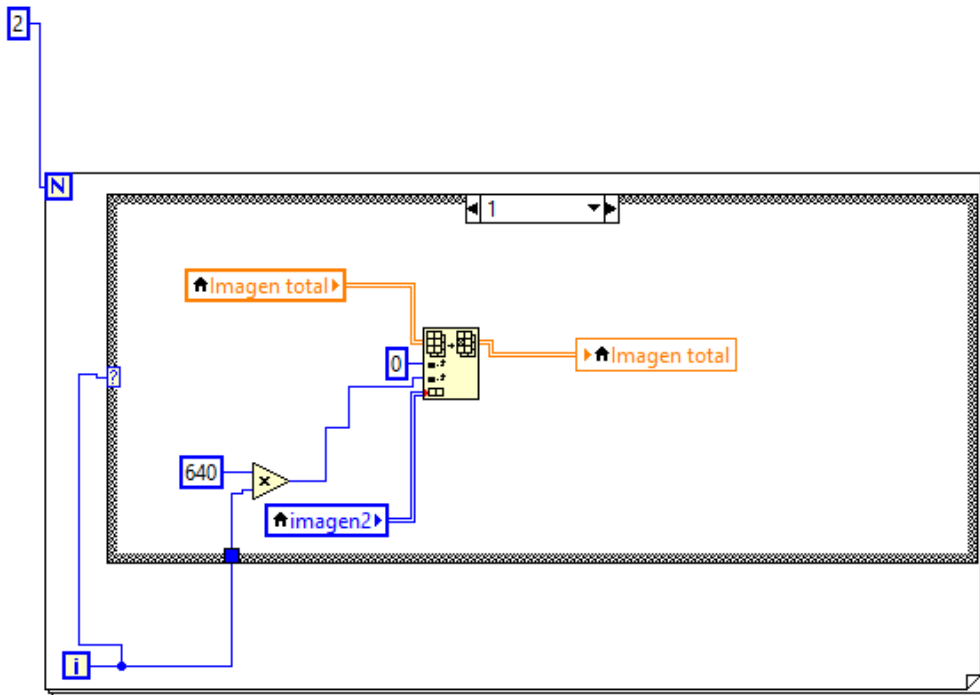


Figura 112.- Construcción de arreglo con imágenes antes capturadas - Programa Comparación

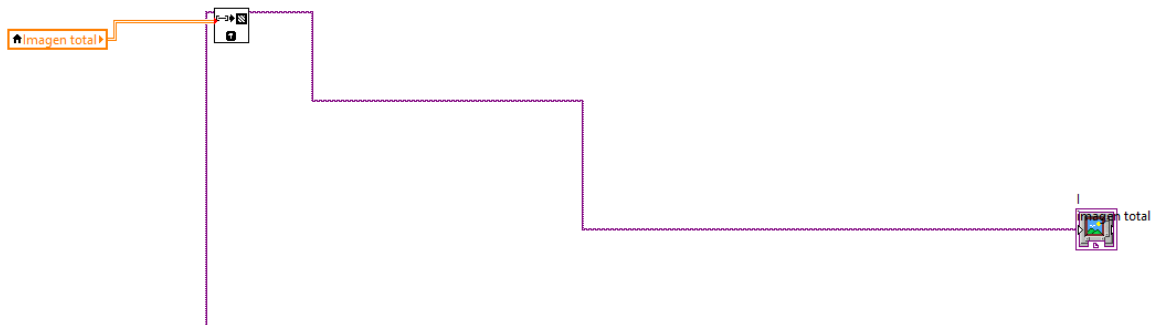


Figura 113.- Visualización de Imagen total - Programa Comparación

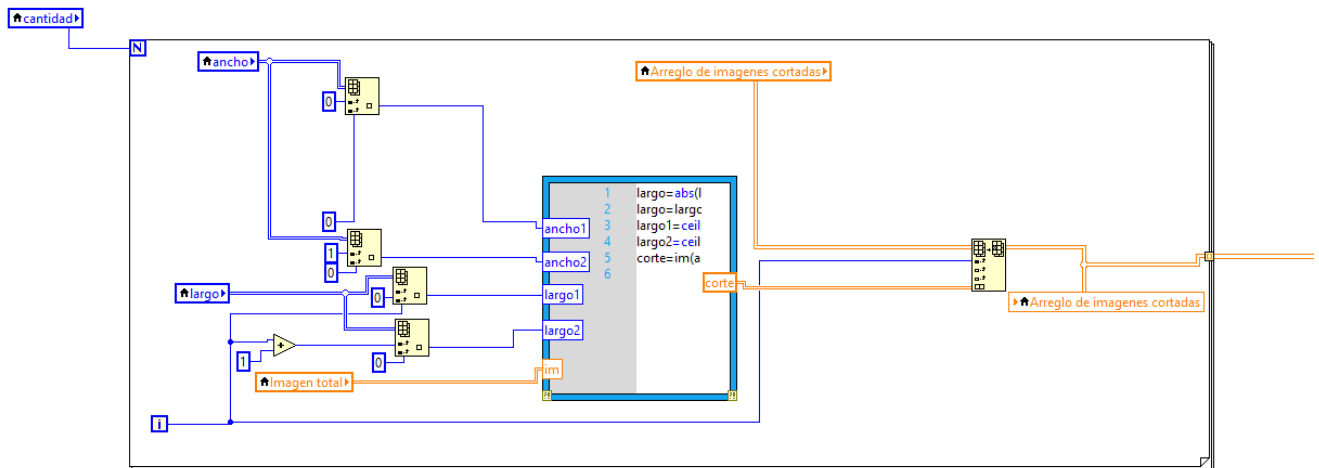


Figura 114.- Corte de imágenes para comparar - Programa Comparación

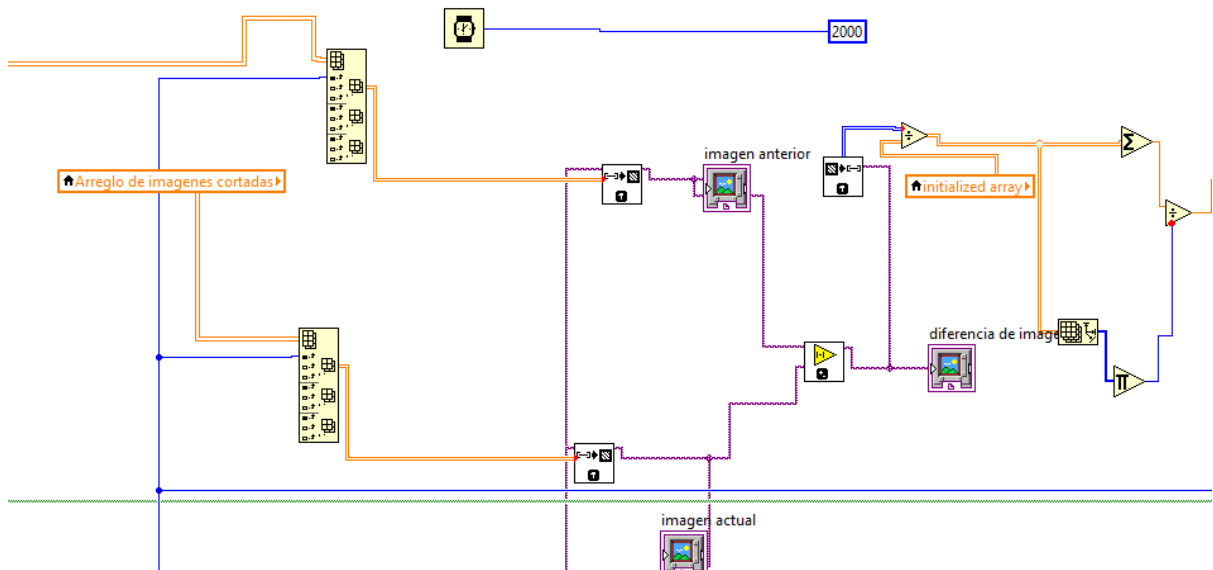


Figura 115.- Resta de imágenes - Programa de Comparación

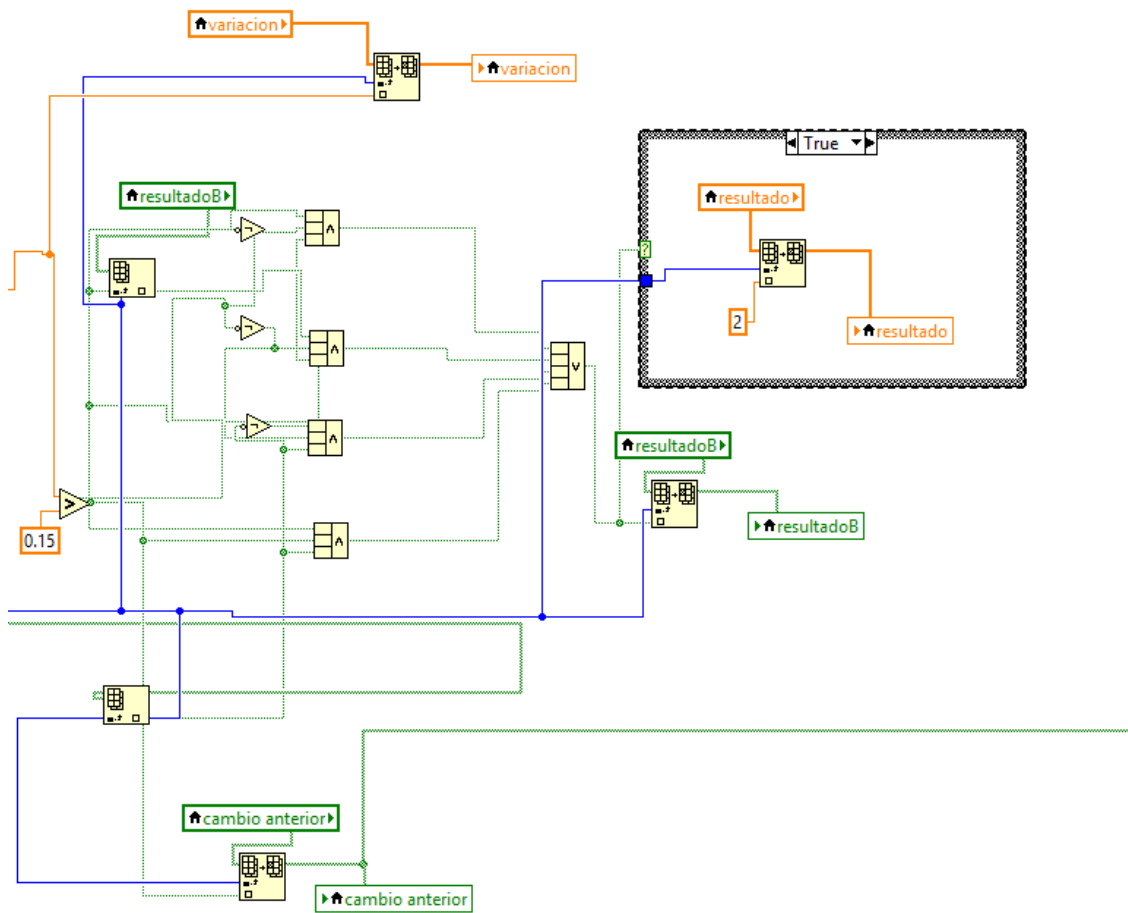


Figura 116.- Algoritmo de Comparación - Programa de Comparación

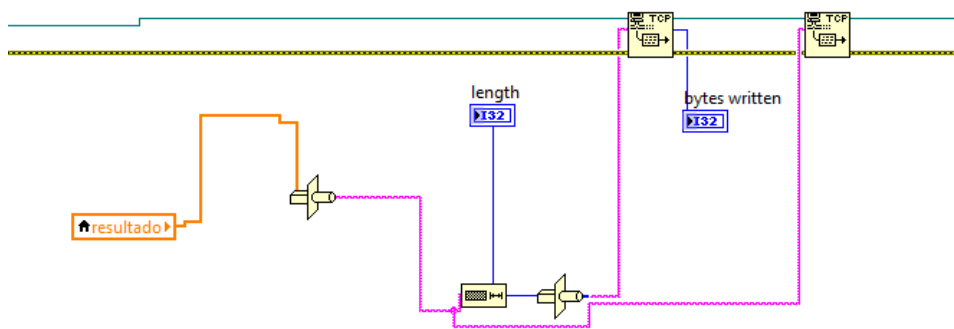


Figura 117.- Envío de datos protocolo TCP/IP - Programa de Comparación

```
1 %Programa de corte
2 %Proyecto de Tesis
3 % Universidad san Francisco De Quito
4 % P. García - Y. Macas
5 largo=abs(largo2-largo1);
6 largo=largo*0.2;
7 largo1=ceil(largo1+largo);
8 largo2=ceil(largo2-largo);
9 corte=im(ancho1:ancho2,largo1:largo2);
```

Figura 118.- Script del programa corte en Mathscript de LabVIEW